ONLINE ISSN 2189-8871

TECHNICAL REPORTS OF THE METEOROLOGICAL RESEARCH INSTITUTE No.87

Reference Manual for the Meteorological Research Institute Community Ocean Model version 5 (MRI.COMv5)

BY

Kei Sakamoto, Hideyuki Nakano, Shogo Urakawa, Takahiro Toyoda
Yuma Kawakami, Hiroyuki Tsujino, and Goro Yamanaka,
(Department of Atmosphere, Ocean, and Earth System Modeling Research)

気象研究所技術報告
第 87 号

気象研究所共用海洋モデル第５版 (MRI.COMv5) 解説

坂本圭・中野英之・浦川昇吾・豊田隆寛
川上雄真・辻野博之・山中吾郎
（気象研究所全球大気海洋研究部）

気象研究所

METEOROLOGICAL RESEARCH INSTITUTE, JAPAN
February 2023

METEOROLOGICAL RESEARCH INSTITUTE

1-1 Nagamine, Tsukuba, Ibaraki, 305-0052 Japan

TECHNICAL REPORTS OF THE METEOROLOGICAL RESEARCH INSTITUTE

The *Technical Reports of the Meteorological Research Institute* has been issued at irregular intervals by the Meteorological Research Institute (MRI) since 1978 as a medium for the publication of technical report including methods, data and results of research, or comprehensive report compiled from published papers. The works described in the *Technical Reports of the MRI* have been performed as part of the research programs of MRI.

気象研究所技術報告　第87号　令和5年2月
TECHNICAL REPORTS OF THE METEOROLOGICAL
RESEARCH INSTITUTE No.87, February 2023
doi:10.11483/mritechrepo.87

Reference Manual for the Meteorological Research Institute Community Ocean Model version 5 (MRI.COMv5)

気象研究所共用海洋モデル第5版 (MRI.COMv5) 解説

SAKAMOTO Kei, NAKANO Hideyuki, URAKAWA Shogo, TOYODA Takahiro
KAWAKAMI Yuma, TSUJINO Hiroyuki, and YAMANAKA Goro,
(Department of Atmosphere, Ocean, and Earth System Modeling Research)

坂本圭・中野英之・浦川昇吾・豊田隆寛
川上 雄真・辻野博之・山中吾郎
（気象研究所 全球大気海洋研究部）

Preface

It has long been recognized that the role of the ocean in the earth's climate system is conclusively important in such issues as the global warming, the long-term variability in the air-sea coupled system like ENSO, associated meteorological extreme phenomena, and so forth. In these situations, modeling of the ocean has become an indispensable method of studying the climate variability and predicting its future state as well as studying the mechanisms of the oceanic variability itself.

The Meteorological Research Institute (MRI) developed its original, general-purpose numerical ocean model, the Meteorological Research Institute Community Ocean Model (MRI.COM), in the early 2000s for both the research work in MRI and operational work in the Japan Meteorological Agency (JMA) by combining two ocean models developed for their research work. The ocean modeling activities are maintained under MRI research programs "Development of a high-resolution (eddy-resolving) ocean general circulation model and study on formation, maintenance, and variation mechanisms of water masses based on the model" (fiscal years 2003 through 2007), "Development of an ocean environmental model and assimilation system and study on variation mechanisms of the ocean environment – feasibility study –" (fiscal year 2008), "Development of ocean environmental forecasting methods" (fiscal years 2009 through 2013), "Research on advanced ocean models" (fiscal year 2014 through 2018), and "Development of ocean prediction technology for multi-scale" (a sub-task of "Research on earth system and ocean modeling", fiscal year 2019 through 2023). The MRI.COM was revised as its fifth version. The present publication is the new MRI.COM manual corresponding to the MRI.COM (version 5). The manual has been revised from the previous version published in 2017.

The ocean modeling study in MRI began in the late 1970s for investigating the variability of the Kuroshio south of Japan. First, an ocean model with the primitive-equation system developed by former Prof. K. Takano in UCLA, USA, was introduced. Another ocean model was then introduced slightly later in 1981. It was similar to the former but developed by an ocean research group in the University of Tokyo. Since that time, the two ocean models with different codes have been improved in parallel in MRI for various purposes. The former model from UCLA has been vigorously optimized to exhibit a high computational efficiency in vector machines, and has been used in experiments with long-term integrations. The latter model from the University of Tokyo incorporated many options from the early stage, such as a surface mixed layer model, an isopycnal diffusion scheme, and a simple sea ice process, for various research and operational purposes.

In the early 1990s, the first coupled ocean-atmosphere model experiment was conducted through cooperation between the Oceanographic Research Department and the Climate Research Department (at the time), MRI, to simulate El Nino phenomenon. Since then, construction of a climate model synthesizing atmosphere, ocean, sea ice, and land surface has been strongly desired both for research and operational work associated with climate warming projection and seasonal forecasts, including the ENSO cycle prediction. To this end, development of a new, general-purpose ocean model, MRI.COM, which could provide the oceanic part of the synthetic climate model, has been initiated based on the two ocean models used so far to achieve efficiency in model improvement and management and to integrate their merits. In designing the new model, the main frame of the former model and the various physical options of the latter model were transferred to the new model, and many newly developed physical processes and schemes were added. MRI.COM was born in this way, and a manual describing the first version was published in 2005.

Since then, the model has been continuously updated through further improvements in physical processes and addition of new processes. One of the most pronounced improvements in MRI.COM (version 5) is the introduction of a new high-precision time integration scheme, which improves computational stability and calculation speed. In addition, improvements on the coupling process with the atmospheric model and the sea ice model, such as introduction of a bulk formula consistent with the atmospheric model and refinement of some sea ice processes, have been made. This makes it possible for an ocean model to seamlessly handle centennial climate change and short-term fluctuations related to coastal disaster prevention. Recent achievements have been fruited in this manual.

MRI.COM has been developed along with its own usage as a part of the climate model and the ocean data assimilation system in MRI, as well as a stand-alone ocean model. MRI.COM has already been in operation at JMA for over 10 years, contributing to providing reliable information such as seasonal prediction and ocean prediction. It also makes an important contribution to the prediction of global warming as the ocean part of the Earth system model, and is now one of the fundamental models of JMA. Based on our experiences, we believe MRI.COM is one of the best ocean models in the world.

In closing, we would like to express our deep gratitude to the great efforts and cooperation of those involved in the development of ocean models over the years. We hope MRI.COM and the present manual will contribute to research work in the fields of climatology, oceanography, and environmental sciences in domestic and foreign institutions as well as to the research and operational work in MRI and JMA.

Goro Yamanaka
Director
Department of Atmosphere, Ocean, and Earth System Modeling Research

序

地球温暖化や ENSO 等の大気海洋結合系の長期変動やそれに伴う異常気象等、地球の気候システムにおける、海洋の役割の重要性は古くから認識されてきた。このような状況の下、海洋数値モデリングは、海洋自体の変動メカニズムの解明のみならず、気候変動の研究やその将来予測を行うにあたっての不可欠な手段となっている。

気象研究所では、2000 年代前半にそれまで研究業務に開発・使用されてきた二種類の海洋モデルを統合して、高い汎用性を持つ気象研究所共用海洋モデル (MRI.COM) を独自に開発し、気象研究所の研究業務や気象庁における現業運用に供してきた。その後も海洋モデル開発は、気象研究所経常研究、「高解像度（渦解像）海洋大循環モデルの開発とそれによる水塊の形成、維持、及び変動機構の解明」（平成 15～19 年度）、「海洋環境モデル・同化システムの開発と海洋環境変動機構の解明に関する研究―フィージビリティ・スタディ― 」（平成 20 年度）、「海洋環境の予測技術の開発」（平成 21～25 年度）、「海洋モデルの高度化に関する研究」（平成 26～30 年度）、「地球システム・海洋モデリングに関する研究」（令和元～5 年度）の副課題 2「マルチスケールに対応した海洋予測技術の開発に関する研究」において継続されている。今回これまでの海洋モデルよりも性能を向上させた MRI.COM（第 5 版）を開発したことから、本技術報告は新しいバージョンの海洋モデルの英語版解説書として、これまでの解説書（2017 年出版）の内容を改訂して出版するものである。

気象研究所における海洋モデリング研究は、海洋変動のメカニズムを解明することを目的に 1970 年代の終わり頃に開始された。その初期段階で、米国 UCLA で高野健三教授により開発されたプリミティブ方程式系モデルが導入された。一方、その直後に東京大学の海洋グループによって開発された別のプリミティブ方程式系モデルも導入された。それ以降、海洋モデルとしてコードの全く異なる 2 系列のモデルが併存し、それぞれのモデルに独自の改良が加えられ、目的に応じて利用に供されてきた。UCLA 系列モデルの特徴は、当時のベクトル計算機に適合させた計算効率性の高さであり、水平的高解像度実験や全球深層循環実験等に使用された。一方、東大系列モデルの特徴は、海面混合層や等密度面拡散、海氷過程といった多彩な物理過程をオプションとして含んでいることであり、ENSO や中層水形成等、表層・中層の時間変動性をターゲットとする種々の研究や気象業務にも幅広く用いられた。

1990 年代初期、エルニーニョ現象再現のための初めての大気・海洋結合モデル実験が、海洋研究部と気候研究部 (いずれも当時) との共同研究として行われて以来、ENSO サイクルはもちろんのこと、地球温暖化予測、季節予報等に関連した研究および気象業務での利用にとって、大気・海洋・海氷・陸域等を総合した気候モデル構築の必要性が急速に高まってきた。このため、モデル開発・管理の効率化とそれぞれのモデルの長所の統合を目的として、従来の 2 系列の海洋モデルをもとに広範な種々の目的に供し得る新たな汎用的海洋モデルを開発することとした。2 系列モデルの統合に当たっては、海洋モデルとしての大枠は UCLA 系列のものを用い、東大系列の多彩な物理過程モデルを融合させるとともに、最新の物理過程やスキームを取り入れることとした。このようにして MRI.COM が誕生し、2005 年にその第 1 版の内容を記述した解説書が作成された。

その後も物理プロセスのさらなる改良と新たなプロセスの付加が継続的に行われ、MRI.COM は改良を重ねて現在に至っている。MRI.COM(第 5 版) について特筆すべきは、高精度時間積分スキームを新たに導入したことである。これにより、計算安定性が向上するとともに、高速化が可能になった。また、大気モデルと整合的なバルク式の導入や海氷プロセスの高度化を行うなど、大気モデルや海氷モデルとの結合過程を改善し、100 年先の気候変動から防災に係る短期変動までをシームレスに扱える海洋モデルとなっている。本解説書には最近の成果が結実している。

MRI.COM はすでに海洋モデル単独実験のみならず、気候モデル実験の海洋パートおよび海洋データ同化システムのモデルパートとしても数多くの研究上の実績を積み上げてきたもので、気象庁ですでに 10 年以上現業運用され、季節予報や海況予報などの確度の高い情報発信に貢献してきた。また、地球システムモデルの海洋部分として地球温暖化予測にも重要な貢献を行っており、今や MRI.COM は気象庁の基盤モデルの一つとなっている。その経験から、本モデルは世界に幾つかある他の海洋モデルに十分伍して行ける性能を持っていると確信している。

終わりに、長年にわたる海洋モデル開発関係者の多大な努力と協力に深く感謝の意を表する。今後、本モデルと解説書が気象庁と気象研究所における気象業務や研究活動のみならず、日本国内、諸外国における気候、海洋、環境科学の研究の推進に大きく貢献することを祈念している。

全球大気海洋研究部長
山中 吾郎

Abstract

About this manual and MRI.COM

This technical report is a manual of the Meteorological Research Institute Community Ocean Model (MRI.COM). MRI.COM is an ocean general circulation model developed and maintained at the Meteorological Research Institute (MRI) of the Japan Meteorological Agency (JMA). It has been used for studying large scale oceanic phenomena and as the oceanic part of the coupled climate models developed at MRI. Recently, it is expanding its coverage to coastal small scale phenomena.

Development of MRI.COM began around 2000. MRI.COM version 1 was created by integrating the two ocean models that had been used by MRI until then. Version 2 (early 2000s) was intended for use in the operational forecasting system in JMA. A Japanese mannual for this version was published in 2005 (Ishikawa et al., 2005), and eventually became the prototype of this manual. Version 3 was intended for use as an oceanic component of the Earth System Model of MRI (MRI-ESM1; Yukimoto et al., 2011). MRI used MRI-ESM1 to participate in the international project "Coupled Model Intercomparison Project Phase 5 (CMIP5)", so a detailed description of its oceanic component was prepared in English (Tsujino et al., 2010). Version 4 (2015) was intended for use as the oceanic part of the next operational forecasting systems of JMA as well as a new Earth System Model of MRI (MRI-ESM2) prepared for the Coupled Model Intercomparison Project Phase 6 (CMIP6). This version also renewed the vertical coordinate system (adopting the so-called z^* coordinate system (Adcroft and Campin, 2004)). For a detailed history of model development, see "Abstract" in the MRI.COMv4 manual (Tsujino et al., 2017).

This report is a manual for MRI.COM version 5, which was released in 2021. This version contains various improvements made for the "Japanese Coastal Ocean Monitoring and Forecasting System", that has been in operation at JMA since 2020. The main improvements are as follows.

- Renewal of time integration scheme from Leapfrog Matsuno method to Leapfrog Adams Moulton method. As a result, model stability and time accuracy have been improved, and calculation costs have been reduced.
- Various devices for speeding up, such as introduction of OpenMP, use of MPI-IO, reduction of communication in nesting experiments, etc.
- Addition of options to improve model stability
- Improvement of coupling with atmospheric model and sea ice model, such as introduction of bulk formula consistent with the JMA's operational atmosphere general circulation model, refinement of sea ice processes, etc.
- Enhancement of analysis function for modeling research, such as isopycnal analysis, particle tracking, netCDF output, etc.
- Many other improvements, such as introduction of physical schemes, parameter adjustment to improve accuracy, and so on.

Note that the purpose of this manual is to present a detailed description of a particular model system. The mathematical expressions of processes, the parameterization methods, and the numerical algorithms presented here follow those adopted in the latest code. They are largely state-of-the-art, but this does not necessarily mean that they are the complete reflections of physical, mathematical, and numerical integrity. Every method is subject to possible sophistication. We welcome critical comments and suggestions from any reader or user, which are necessary for further improvement.

For a more general or detailed description of OGCMs, please refer to textbooks by Griffies (2004) and Kantha and Clayson (2000). The former thoroughly describes the fundamentals of OGCMs, and the latter concisely summarizes the modeling of various oceanic processes such as tide and sea ice.

Development management

MRI.COM has been developed over almost 20 years, and, as a result, the program code has become larger and more complicated. Since it is now used for many projects, various departments and people of MRI and JMA are involved. To develop the model efficiently and maintain stability through widespread usage, we adopted tools and methods used in modern software development and redesigned our development management system. For example, a version control system of the code, Git, has been introduced, which enables multiple developers to work on multiple tasks in parallel. A project management system, Redmine, is also used for all development members to share the development situation. Based on these tools, we can share information and perform mutual checks daily as a development team. This greatly contributes to improvement of code quality. See Sakamoto et al. (2018) for more information on our development management.

Organization

Chapter 1 introduces OGCMs and MRI.COM. It also presents the classification of OGCMs and the status of MRI.COM with respect to the state-of-the-art OGCMs.

Part I describes the model configuration. Governing equations are derived in Chapter 2, and the spatial grid arrangement and grid area and volume calculation methods are explained in Chapter 3. The method of time integration is explained in Chapter 4.

Part II describes the solution procedures of diagnostic equations. Chapter 5 presents definition of the equation of state of sea water. Chapter 6 presents definition of the continuity equations for unit grid cells.

Part III describes the solution procedures of momentum equation. The method of solving the barotropic and the baroclinic part of the momentum equation are presented in Chapters 7 and 8, respectively.

Part IV describes the method of solving the advection-diffusion equation for tracers. After summarizing the equation in Chapter 9, Chapter 10 presents tracer advection schemes. Subgrid-scale parameterizations for horizontal and vertical mixing are explained in Chapters 11 and 12, respectively. Chapter 13 explains usage of the tracer package.

Part V describes boundary processes. Surface fluxes are presented in Chapter 14. Turbulence closure models that expresses small-scale processes in the sea surface mixed layer are presented in Chapter 15. Bottom boundary layer parameterization is explained in Chapter 16.

Part VI outlines coupling with other models. Combination with the sea ice model is described in Chapter 17, and combination with the tidal model in Chapter 18. Biogeochemical models are presented in Chapter 19, and some passive tracer models such as chlorofluorocarbons are described in Chapter 20. The program code for these models is included in MRI.COM and is being developed along with the physical model. Chapter 21 describes how to set up MRI.COM for the atmosphere-ocean general circulation model (AOGCM) experiment. Finally, how to construct and run a "nested-grid model" that combines multiple MRI.COM models is presented in Chapter 22.

Part VII contains miscellaneous topics. Basics of the finite difference method are presented in Chapter 23, general orthogonal curvilinear coordinates and related calculus are introduced in Chapter 24, and finally user's guide to construct and run a model is presented in Chapter 25.

Each chapter is almost independent from other chapters. Thus the readers might be able to understand the contents of each chapter without referring to other chapters. However, reading Part I will give the readers the background to help understand the remainder of this manual.

In this manual, we tried to derive the formula that directly corresponds to the program code. On the other hand, the description about the code itself is kept to a minimum. See the Japanese website `https://mri-ocean.github.io/mricom/` for commentary of the code. The content of this document is basically in line with version 5.0 released in 2021, though some content was developed after version 5.1.

The following are some comments about the notations used throughout this manual. The characters and expressions in `Courier` fonts are adopted from program codes. The subscripts and indices used in discrete equations are intended to express staggered grid arrangements. They do not necessarily correspond to the array indices in program codes.

概 要

MRI.COM と本解説書について

本技術報告は、気象研究所共用海洋モデル (MRI.COM) の解説書である。MRI.COM は気象庁気象研究所で開発、維持されてきた海洋大循環モデルである。本モデルは、大きなスケールの海洋現象に関する研究や気象研究所で開発された気候モデルの海洋部分として使用されてきた。最近では、その適用可能範囲を沿岸の小スケール現象に広げつつある。

MRI.COM は 2000 年頃に開発が開始された。それまで気象研究所で使用されてきていた 2 系統の海洋モデルを統合することにより、基本型であるバージョン 1 が作成された。バージョン 2（2000 年代前半）は、気象庁の現業システムへの供用を目的とした。このバージョンに対する日本語の詳細な解説書が 2005 年に出版され (Ishikawa et al., 2005)、本解説書の原型となっている。バージョン 3 は、気象研究所地球システムモデル (MRI-ESM1; Yukimoto et al., 2011) の海洋部分への供用を主な目的としたものである。気象研究所は MRI-ESM1 を用いて、国際的なプロジェクトである「結合モデル相互比較プロジェクト第 5 フェーズ（CMIP5）」に参加した。このため、解説書も英語で改訂された (Tsujino et al., 2010)。バージョン 4 は 2015 年にリリースされた。これは気象庁における現業海洋予測システム、並びに結合モデル相互比較プロジェクト第 6 フェーズ（CMIP6）用の新地球システムモデル（MRI-ESM2）の海洋部分としての供用を意図したものである。このバージョンでは鉛直座標系の刷新 (いわゆる z^* coordinate 系 (Adcroft and Campin, 2004) の採用) も行われた。モデル開発の詳しい歴史については、MRI.COMv4 マニュアル (Tsujino et al., 2017) の「概要」を参照されたい。

本解説書は、2021 年にリリースされたバージョン 5 の解説書である。このバージョンには、2020 年から気象庁で現業運用されている「日本沿岸海況監視予測システム」のために施された様々な改善が含まれている。主要な改良点は以下のとおりである。

- Leapfrog Matsuno method から Leapfrog Adams Moulton method への時間積分スキームの刷新。これにより、保存性の向上、時間精度の向上、計算コストの削減が実現された。
- 様々な高速化の導入 - OpenMP 導入、MPI-IO の利用、ネスティング通信軽量化など。
- 安定性向上のためのオプション
- 大気モデル及び海氷モデルとの結合の改善 - 気象庁現業大気モデルと整合するバルク式の導入や、海氷プロセスの高度化など。
- モデリング研究のための解析機能の強化 - 等密度面解析、粒子追跡、netCDF 出力など
- その他、多くの物理スキームの導入や、精度向上のための調整など多数の修正

読者には、本解説書の目的が、特定のモデルシステムに関する詳細な説明を与えることである点に注意していただきたい。本解説書に示す、海洋に生じる現象の数学的表現、パラメタリゼーションの方法、数値アルゴリズムは、最新のプログラムコードに則ったものである。これらは概ね最先端の知見を反映したものであるが、それがすなわち物理的、数学的、計算機科学的手法の完全性を表現していることを意味するわけではない。あらゆる手法も改良の対象となり得る。従って、著者らは読者ならびに利用者からの忌憚なきコメントや助言を歓迎する。これらはモデルのさらなる改良には不可欠なものである。

海洋大循環モデルに関するより包括的で詳細な解説に興味のある読者は Griffies (2004) や Kantha and Clayson (2000) による教科書を参照することを勧める。前者には海洋大循環モデルの原理的な面についての詳細な記述が、後者には潮汐や海氷といった、様々な海洋現象のモデリングの関する知見が簡潔にまとめられている。

開発管理

MRI.COM は 開発が始まってから 20 年が経過し，気象研究所と気象庁の様々な部門で利用されるようになるとともに，ソースコードの大規模化・複雑化が進んだ。このような状況の下でも，バグの混入や意図しない影響を抑えながらモデルを効率的に開発するため，現代的なソフトウェア開発で用いられるツールと手法を取り入れ，開発管理を

行っている。例えば、ソースコードのバージョン管理ツール「Git」が導入され、複数の開発者が複数の課題に同時に取り組む並行開発が可能になった。プロジェクト管理システム「Redmine」の導入により、開発状況は開発者全員で共有されている。これらのツールを基盤として開発チーム内の情報共有とコードレビューが日常的に行われており，コード品質の維持に寄与している。我々の開発管理の詳細については、Sakamoto et al. (2018) を参照せよ。

本解説書の構成

第 1 章では、海洋大循環モデルと MRI.COM を紹介する。海洋大循環モデルの分類と、最先端の海洋モデルに対する MRI.COM の位置づけについて述べる。

第 I 部では、モデルの基本設定について述べる。第 2 章では支配方程式の定式化を行う。空間格子配置、及び単位格子の面積や体積計算方法の解説を第 3 章で行う。時間積分の手法を第 4 章で説明する。

第 II 部では、診断式について説明を行う。第 5 章は海水の状態方程式の定義を示す。第 6 章は単位格子に対する連続方程式の差分式の定義を示す。

第 III 部ではモデルの運動方程式の解説を行う。運動方程式の順圧成分と傾圧成分の解法を第 7 章と第 8 章でそれぞれ述べる。

第 IV 部ではトレーサーに対する移流拡散方程式の解法を述べる。第 9 章で方程式の概要をまとめたあと、トレーサー移流スキームについて第 10 章で解説する。サブグリッドスケールの水平方向と鉛直方向の混合に対するパラメタリゼーションについて、第 11 章および第 12 章でそれぞれ解説する。第 13 章はトレーサーパッケージの使用法を説明する。

第 V 部では境界プロセスの解説を行う。海面フラックスの取り扱いを第 14 章で、海面混合層内の小スケール過程を表現する「乱流クロージャーモデル」を第 15 章で説明する。海底境界層モデルを第 16 章で説明する。

第 VI 部では他モデルとの結合について概説する。第 17 章では海氷モデルとの、第 18 章では潮汐モデルとの結合を解説する。生物地球化学モデルを第 19 章で、クロロフルオロカーボンなどいくつかのパッシブトレーサー・モデルを第 20 章で解説する。これらのモデルのソースコードは MRI.COM 内に同梱され、一体となって開発されている。第 21 章では大気海洋結合モデル実験時の、MRI.COM の設定方法を記述する。最後に、第 22 章では、複数の MRI.COM モデルを結合した「入れ子モデル」の作成と使用法について解説する。

第 VII 部では上記に分類できない項目について取り上げる。差分解法の基本について第 23 章で、一般直交曲線座標とそれに関連した計算法について第 24 章で、最後にモデルの作成と実行方法の解説を第 25 章で行う。

各章はほぼ他の章から独立している。そのため、読者は他の章を参照しなくとも各章の内容を理解できるはずである。但し、第 I 部を読んでおくと、それが背景的知識となって、本解説書の残りの部分の理解が容易になるので、参考にしていただきたい。

本解説書では、プログラムコードと直接に対応した式の導出を心がけた。一方で、プログラムコード自体に関する記述は最小限にとどめている。コーディングに関する説明については、日本語の web サイト `https://mri-ocean.github.io/mricom/` を参照してほしい。本書の内容は、2021 年リリースの version 5.0 に基本的に即している。ただし、version 5.1 以降に開発された内容も一部含まれていることに注意してほしい。

最後に本解説書で用いる表記法についての注意点を述べる。本文中タイプライター (`Courier`) 活字体が用いられている部分はプログラムコードからの抜粋である（綴り間違えなどではない）。差分式に現れる添え字や指数はスタガード（千鳥状）格子配置を表現するように意図している。しかし必ずしもプログラムコードの配列番号とは対応していないので注意願いたい。

Contents

Chapter 1

OGCMs and MRI.COM

This chapter outlines the ocean general circulation models (OGCM) and the status of MRI.COM. See "Organization" in Abstract for an introduction to this manual.

1.1 What do OGCMs cover?

OGCMs have been supposed to simulate relatively large-scale phenomena such as global-scale thermohaline circulations, basin-scale wind-driven circulations, and mesoscale eddies (Figure 1.1). Small-scale processes that are either unresolved or neglected might be incorporated in some forms of subgrid-scale (SGS) parameterizations. The current basin or global scale OGCMs may cover phenomena from thermohaline circulations to mesoscale eddies, but it is still almost impossible to conduct a simulation long enough to achieve a quasi-steady state of a thermohaline circulation with a horizontal resolution ($\sim$ several km) that is sufficiently high to resolve mesoscale eddies, even with the present computation resources. For these reasons, the standard practice in the ocean model community is to use a low horizontal resolution (a few hundred kilometers) model to study global thermohaline circulations and to use a limited-domain model to study an eddying ocean. Some research projects seek to conduct a several-hundred-year integration of a high resolution model that resolves mesoscale eddies using enormous resources, but such a resource is not available to everyone.

In recent years, the OGCMs are used to directly simulate some of the previously unresolved small scale phenomena. These include submesoscale currents (McWilliams, 2016) and the generation of internal gravity waves due to the interactions of tides and mesoscale currents with topographic features. Because large scale features such as mesoscale eddies, boundary currents, and oceanic fronts are prerequisite for the spontaneous emergence of these fine-scale features, an approach of embedding (nesting) subdomains with refined grids in a basin scale model is often taken. Use of a common OGCM code is preferable in these approaches.

1.2 Classification of OGCMs

Most OGCMs used by ocean research scientists and by operational centers for forecasting climate and oceanic states numerically solve almost the same set of equations for the Boussinesq and hydrostatic ocean. The fundamental equations consist of the momentum equation for continuous fluid, the advection-diffusion equation for temperature and salinity, the equation of state of sea water, and the mass conservation equation, collectively called primitive equations (Chapter 2). If necessary, equations for additional processes such as surface mixed-layer physics, sea ice, and bottom boundary layer physics are added.

Most OGCMs applied to ocean-climate studies adopt the finite difference or finite volume method on structured mesh to discretize the equations. Recently, the finite element method on horizontally unstructured mesh is also applied to both coastal ocean simulation and global ocean simulation (e.g., Chen et al., 2011; Danilov, 2013). MRI.COM takes the former approach. The spectral approach widely used by atmospheric models would have difficulty treating lands that completely block ocean circulation in the zonal direction, and thus this approach is not usually adopted in general-purpose ocean models.

Ocean models adopting horizontally structured mesh are classified by how they discretize the vertical direction. The choice of the vertical coordinate leads to fundamental differences among the models. There are three classes: z-coordinate models or z-models adopting depth as the vertical coordinate, σ-coordinate or terrain following models adopting fractional depth between the sea surface and the sea floor as the vertical coordinate, and ρ-coordinate or isopycnal models adopting isentropic surfaces (iso-potential density surfaces) as the vertical coordinate. Each class has its advantages and disadvantages and recent efforts are directed toward adopting generalized vertical coordinates, i.e., remedying each model's disadvantages by using advantages of other classes. Readers are referred to the book by Griffies (2004) for more general discussion about OGCMs.

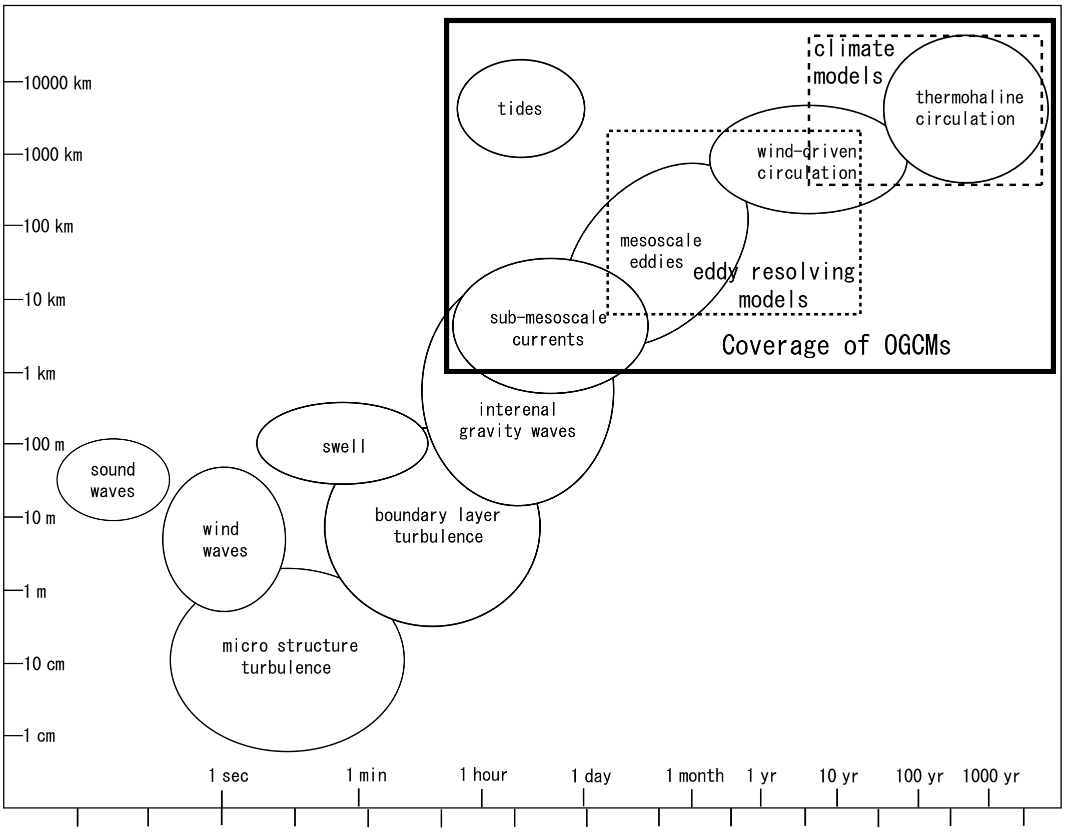

Figure1.1 Various oceanic phenomena in their space and time scales and coverage of the ocean general circulation model. the figure of oceanic phenomena is adopted from von Storch and Zwiers (2001), but some phenomena are added.

1.2.1 Z-coordinate models (z-models)

The first z-coordinate general circulation model was developed by Dr. Kirk Bryan and his colleagues at the Geophysical Fluid Dynamics Laboratory (GFDL) in the 1960's (e.g., Bryan, 1969). This model is sometimes referred to as the Bryan-Cox-Semtner model or GFDL model. The z-models utilize the character of the ocean that the local pressure is expressed as a function of depth by zero-order approximation, which makes implementing the equation of state straightforward. Implementation of bottom topography and drawing of results are also straightforward. The models of this class are most widely used in the community because of their versatility. Such models were first used as components of coupled atmosphere-ocean models.

The descendant of the GFDL model is called the Modular Ocean Model (MOM; Griffies, 2012) and is one of the most widely used z-models. Massachusetts Institute of Technology general circulation model (MITgcm; Marshall et al., 1997) and Nucleus for European Modelling of the Ocean (NEMO; Madec and the NEMO team, 2008) are also widely used. Most climate centers participating in climate model intercomparison projects use the z-coordinate models; MRI.COM also adopts z-coordinate. In Japan, the Center for Climate System Research Ocean Component Model (COCO) at the University of Tokyo (Hasumi, 2006) and the Research Institute of Applied Mechanics Ocean Model (RIAMOM) are also in this class.

The major disadvantages of this class of models are as follows:

- The vertical resolution in shallow seas and near the sea floor tends to be low and the processes that arise near the coast and the sea floor tend to be poorly reproduced.
- Numerical inaccuracy in the tracer transport algorithm immediately leads to spurious diapycnal mixing of the transported properties, while the diapycnal mixing is supposed to be very small in the ideal ocean interior.

The first disadvantage is expected to be remedied by the σ-models and the second by the ρ-models. However, z-model's disadvantages are not completely overcome; these substitutes have their own difficulties.

1.2.2 Sigma-coordinate models (σ-models)

The first sigma-coordinate model was developed by Dr. George Mellor and his colleagues at Princeton University. Since the number of vertical grid points is invariable throughout the model domain, σ-models are widely used for coastal ocean simulations.

Two major sigma-models are widely used in the community: The Princeton Ocean Model (POM; Mellor, 2004) and the Regional Ocean Modeling System (ROMS; Shchepetkin and McWilliams, 2003; 2005).

The major disadvantages of this class of models are as follows:

- An accurate representation of the horizontal pressure gradient is difficult near steeply sloping bottom topography.
- The lateral mixing along the same vertical layer near the continental slope region might lead to mixing of the shoreward light water and the seaward dense water.

These problems might prohibit using σ-models in long-term integrations of the global ocean.

1.2.3 Isopycnal-coordinate models (ρ-models)

The first isopycnal-coordinate model was developed by Dr. Rainer Bleck at the University of Miami. The development of this class of models is based on the fact that sea water moves along isopycnal surfaces in the interior. Thus, the character of a water mass is well maintained in the ocean interior. Since many theoretical studies of physical oceanography use an isopycnal-coordinate framework, the ρ-models have the great advantage of providing good correspondence between theory and numerical models.

A major ρ-model widely used in the community is the Miami Isopycnic Coordinate Ocean Model (MICOM; Bleck and Boudra, 1986) developed at the University of Miami.

The major disadvantages of this class of models are as follows:

- Implementation of surface mixed layer models into a ρ-model is in itself inappropriate.
- Since the density levels are prescribed, this class of models might not be appropriate for studying a drastic climate change that could lead to great variations in density of major water masses.

The Hybrid Coordinate Ocean Model (HYCOM; Bleck et al., 2002) has been developed in an effort to remedy some of these disadvantages.

1.3 About MRI.COM

MRI.COM is a z-coordinate model and uses the finite volume method on structured mesh to discretize the governing equations. The horizontal grid arrangement is Arakawa's B-grid (Arakawa, 1972). Coast lines are defined by the periphery of the grid cell centered by the velocity point, i.e., the lines connecting the tracer points. This arrangement is suitable for the discrete expressions for the side boundary conditions for velocity, and transport through a narrow passage can be achieved with a single grid cell. The time finite-difference scheme has been renewed from Leapfrog + Matsuno scheme to Leapfrog Adams-Moulton (LFAM3), a third-order predictor-corrector scheme, at MRI.COMv5 (Chapter 4).

Though the program code should ideally use MKS units, MRI.COM basically uses cgs units for historical reasons. The sea ice model, the mixed-layer model, some surface bulk formulae, and biogeochemical models are coded in MKS units and converted into cgs units before their outputs are used by the main part. Details are described in each chapter.

1.4 Future of OGCMs and MRI.COM

As OGCMs acquire scientific and numerical integrity, they expand the area of their usage and begin to fulfill social as well as scientific needs. The developer of an OGCM thus has the responsibility to ensure its scientific and numerical integrity and to acknowledge its limitations. Feedback from users of various fields and continuous efforts to overcome limitations will certainly improve the OGCMs.

Innovative ideas are still being published in basic frameworks of OGCMs as well as improvements in various physical schemes. Model development centers around the world continue to improve their OGCMs by incorporating them. For example, the vertical coordinate scheme called the Arbitrary Lagrangian-Eulerian method (ALE), which mixes multiple vertical coordinate systems described in Section 1.2 in a model, has the potential to dramatically improve a long-standing problem of false numerical diffusion. In the development of MRI.COM, we need to keep up with the progress in research about OGCM.

The advance of OGCMs has kept pace with that of super-computers. The mainstream of super computing is shifting from vector computation to massively parallel computation with distributed memory. On the other hand, general-purpose computing using inexpensive Graphics Processing Units (so-called GPGPU) is also becoming widespread in the field of high-performance computing. As the computer architecture advances, our program code must be constantly improved. With increasing computing power, ever higher resolution simulations will be achieved. The result will be more strongly affected by how subgrid-scale processes are parameterized and thus subgrid-scale schemes should be selected carefully.

To continue to stand as a multi-purpose model, an OGCM should be easily coupled with other component models and data assimilation schemes. Other component models include wave, river, iceberg, and ice-shelf models. In the interaction with these models, physical quantities should be exchanged by following conservative lows. Having an interface to universal couplers and an adjoint code should be mandatory for such a multi-purpose OGCM.

These are the main subjects for developing MRI.COM in the coming years.

Part I

Configuration

Chapter 2

Governing Equations

In this chapter, the governing equations for the general ocean circulation are formulated. These equations are usually called primitive equations. A discrete form of these equations is also presented to explain the fundamental solution methods. The detailed numerical methods are presented in later chapters.

2.1 Formulation

2.1.1 Coordinate System

The fundamental purpose of developing this ocean circulation model (MRI.COM) is to use it for realistic simulations of oceanic circulations in various circumstances. To achieve this, we must carefully choose a coordinate system before formulating the governing equations.

In the lateral direction, the governing equations need to be formulated on a sphere. Originally, spherical coordinates were adopted, and the equations were discretized on a geographical (latitude-longitude-depth) grid. A problem arises for a global model because the North Pole is a singular point in the geographic coordinate system. Since the zonal grid widths within five latitudinal degrees from the Pole become less than a tenth of those in middle to low latitudes, a short time step is required owing to the limitation of the Courant-Friedrichs-Lewy (CFL) condition.[*] This becomes a burden on performing long-term integration.

One simple means to remove this North Pole singularity is to shift both poles to land. In this case, one may use the spherical coordinate program codes without major modification by only adjusting the Coriolis parameter. Unfortunately, there are not many pairs of points on land that are symmetric about the Earth's center.[†] Even if the most ideal pair with poles on Greenland and Antarctica (near the Ross Sea) was chosen, it is only 5 degrees from the coastline to the newly shifted pole. One might also be concerned that the Equator is not represented as a line on the shifted grid arrangement.[‡]

To resolve these issues, the model equations are represented on generalized orthogonal coordinates instead of spherical coordinates. Users can select the coordinate system according to their purposes. For example, the resolution of a target region can be intentionally enhanced by placing a pole of the distorted grid near the target region. Of course, a regional model without the North Pole may be arranged on geographic coordinates since spherical coordinates are one form of generalized orthogonal coordinates. Now our model equations are formulated on generalized orthogonal coordinates. Chapter 24 summarizes the concepts and calculus related to generalized orthogonal coordinates.

In the vertical direction, the depth coordinate was adopted from the first stage of the development. No attempt has been made to apply other options such as terrain following or density coordinates. In the earliest stage, the sea surface was assumed to have a rigid-lid on it. Then, the sea surface was allowed to move freely. When the free surface was first introduced, the movement of sea surface was absorbed in the first layer of the model. The problem with this treatment is that it is not possible to take the first layer thickness thinner than about 4 meters whereas a finer vertical resolution is required near the sea surface. This is because the contrast of mean sea level in the global ocean may reach 3 meters. To resolve this problem, the upper several layers were allowed to undulate following the sea surface evolution as in the sigma-coordinate model (Hasumi, 2006). A problem with this approach is that there is a transition in the vertical coordinate, which would make analytical treatment awkward in some situations. For MRI.COM version 4, we have adopted a vertically rescaled height coordinate system, where a sea level undulation is reflected throughout the water column (Adcroft and Campin, 2004). This vertical coordinate is named z^* coordinate.

In this section, we first formulate the governing equations on Cartesian coordinates for brevity. Then a coordinate transformation in the lateral direction is applied to the governing equations. Approximations and boundary conditions

[*] The time step, Δt, needs to satisfy $v\Delta t/\Delta x \leq 1$, where v is velocity and Δx is the grid width.

[†] Greenland and Antarctica, China and Argentina, Kalimantan and Columbia.

[‡] If the grid size is fine enough, the Kelvin wave in the shifted-pole model will propagate along the Equator as the theory suggests.

are discussed using equations on generalized orthogonal coordinates. Equations are further transformed to introduce generalized vertical coordinates. Readers are referred to Griffies and Adcroft (2008) for the thorough discussion on the formulation of primitive equations for ocean circulation models.

2.1.2 Primitive equations in Cartesian coordinates

a. General governing equations

Evolution of state variables of ocean circulation $(\mathbf{v}, \rho, \theta, S, p)$, where $\mathbf{v} = (u, v, w)$ is the velocity vector, ρ is density, θ is potential temperature, S is salinity, and p is pressure, is obtained by solving the following simultaneous equations written in Cartesian coordinates.

Momentum equation in a vector form is

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v}\cdot\nabla)\mathbf{v} + 2\Omega \times (\rho\mathbf{v}) = -\rho\nabla\Phi - \nabla p + \nabla\cdot\tau, \tag{2.1}$$

where Ω is the rotation vector of the Earth, Φ is the geopotential, τ is the frictional stress tensor.

Mass conservation equation is

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0. \tag{2.2}$$

Equations for potential temperature and salinity are

$$\rho\frac{\partial \theta}{\partial t} + \rho\mathbf{v}\cdot\nabla\theta = -\nabla\cdot(\rho\mathbf{F}^{\theta}) + \rho Q^{\theta} \tag{2.3}$$

and

$$\rho\frac{\partial S}{\partial t} + \rho\mathbf{v}\cdot\nabla S = -\nabla\cdot(\rho\mathbf{F}^{S}) + \rho Q^{S}, \tag{2.4}$$

where $\mathbf{F}^{\theta,S}$ are tracer fluxes due to subgrid-scale transport and mixing parameterizations and $Q^{\theta,S}$ are sources of tracers due to nudging, convective adjustment (Section 12.2), shortwave absorption (Section 14.3), etc. Here, tracer concentration is expressed as concentration per unit mass of sea water.

Equation of state of sea water determines in situ density of sea water. Density is given as a function of potential temperature, salinity, and pressure:

$$\rho = \rho(\theta, S, p). \tag{2.5}$$

The equation of state is usually given as a polynomial fit to the available measurements. A detailed description of this will be presented in Chapter 5.

The above set is the most general set of equations governing the evolution of oceanic state.

b. Boussinesq approximation

Because the density of sea water varies only by 5% throughout the water column and the horizontal density variations are less than 1%, most ocean general circulation models use the Boussinesq approximation. In the Boussinesq approximation, the density (ρ) in the non-linear product of density times velocity $(\rho\mathbf{v})$ that appears in the momentum equation (2.1) is replaced by a reference density (ρ_0). The momentum equation becomes

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\nabla)\mathbf{v} + 2\Omega\times\mathbf{v} = -\frac{\rho}{\rho_0}\nabla\Phi - \frac{1}{\rho_0}\nabla p + \frac{1}{\rho_0}\nabla\cdot\tau. \tag{2.6}$$

Note that the in-situ density (ρ) is retained for the geopotential term.

Further, the sea water is treated as incompressible. Mass conservation equation (2.2) becomes the volume conservation equation:

$$\nabla\cdot\mathbf{v} = 0. \tag{2.7}$$

The tracer concentrations are now concentration per unit volume instead of unit mass:

$$\frac{\partial \theta}{\partial t} + \mathbf{v}\cdot\nabla\theta = -\nabla\cdot\mathbf{F}^{\theta} + Q^{\theta}, \tag{2.8}$$

and

$$\frac{\partial S}{\partial t} + \mathbf{v}\cdot\nabla S = -\nabla\cdot\mathbf{F}^{S} + Q^{S}. \tag{2.9}$$

The above expression is the most general form under the Boussinesq approximation. This form is used to formulate an oceanic non-hydrostatic model and a quasi-hydrostatic model (Marshall et al., 1997).

2.1.3 Primitive equations in generalized orthogonal coordinates

We consider the momentum equation first. On a generalized orthogonal coordinate system (μ, ψ, r) whose unit vectors are $\mathbf{e}_\mu$, $\mathbf{e}_\psi$, and $\mathbf{e}_r$, the momentum equation for velocity $\mathbf{v} = u\mathbf{e}_\mu + v\mathbf{e}_\psi + w\mathbf{e}_r$, where $u = h_\mu \dot{\mu}$, $v = h_\psi \dot{\psi}$, $w = h_r \dot{r}$, is represented by

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u + f_\psi w - f v = -\frac{1}{\rho_0 h_\mu}\frac{\partial p}{\partial \mu} - \frac{v}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial \psi}u - \frac{\partial h_\psi}{\partial \mu}v\right) - \frac{w}{h_r h_\mu}\left(\frac{\partial h_\mu}{\partial r}u - \frac{\partial h_r}{\partial \mu}w\right) + F^\mu_{\text{fric}}, \tag{2.10}$$

$$\frac{\partial v}{\partial t} + \mathbf{v} \cdot \nabla v + f u - f_\mu w = -\frac{1}{\rho_0 h_\psi}\frac{\partial p}{\partial \psi} - \frac{w}{h_\psi h_r}\left(\frac{\partial h_\psi}{\partial r}v - \frac{\partial h_r}{\partial \psi}w\right) - \frac{u}{h_\mu h_\psi}\left(\frac{\partial h_\psi}{\partial \mu}v - \frac{\partial h_\mu}{\partial \psi}u\right) + F^\psi_{\text{fric}}, \tag{2.11}$$

$$\frac{\partial w}{\partial t} + \mathbf{v} \cdot \nabla w + f_\mu v - f_\psi u = -\frac{1}{\rho_0 h_r}\frac{\partial p}{\partial r} - \frac{\rho}{\rho_0}\frac{\partial \Phi}{\partial r} - \frac{u}{h_r h_\mu}\left(\frac{\partial h_r}{\partial \mu}w - \frac{\partial h_\mu}{\partial r}u\right) - \frac{v}{h_\psi h_r}\left(\frac{\partial h_r}{\partial \psi}w - \frac{\partial h_\psi}{\partial r}v\right) + F^r_{\text{fric}}, \tag{2.12}$$

where h_μ, h_ψ, and h_r are scale factors, which measure the width in the original coordinate of the unit length in the new coordinate. Metric terms appear on the r.h.s.. $\mathbf{F}_{\text{fric}}$ is frictional force obtained as the divergence of frictional stress tensor. The radial distance from the Earth's center is represented by r and the gravitational acceleration is in the negative direction of r.

The Coriolis force is represented by

$$2\Omega \times \mathbf{v} = (2\Omega_\psi w - 2\Omega_r v)\mathbf{e}_\mu + (2\Omega_r u - 2\Omega_\mu w)\mathbf{e}_\psi + (2\Omega_\mu v - 2\Omega_\psi u)\mathbf{e}_r, \tag{2.13}$$

where $\Omega = \Omega_\mu \mathbf{e}_\mu + \Omega_\psi \mathbf{e}_\psi + \Omega_r \mathbf{e}_r$ is the rotation vector of the Earth. We designate $f_\mu = 2\Omega_\mu$, $f_\psi = 2\Omega_\psi$, and $f = f_r = 2\Omega_r$.

We apply the following two approximations which are relevant to the momentum equation.

a. Shallow ocean approximation

Shallow ocean approximation employs the fact that the vertical thickness of the ocean is far smaller than the radius of the Earth. Since the vertical scale of motion of a water particle is far smaller than the Earth's radius (a), the radial distance r as a scalar quantity is replaced by the Earth's radius a. The new vertical coordinate (z) is the distance (positive upward) from the geoid (sea surface height in the state of rest) and $\partial/\partial r$ is replaced by $\partial/\partial z$. We set a constant vertical scale factor $h_r (= h_z = 1)$. Horizontal scale factors are independent of vertical coordinate ($\partial h_\mu/\partial r = \partial h_\psi/\partial r = 0$). As a result, to conserve angular momentum under this approximation, we drop the metric terms that involve w for the horizontal components and all the metric terms for the vertical components.

We also assume that the gravitational acceleration is constant (g). This assumption results in a specific expression of geopotential as $\Phi = gz$.

b. Hydrostatic approximation

For horizontal motions with a scale larger than a few kilometers, hydrostatic balance is maintained in the vertical direction. The vertical momentum equation becomes:

$$0 = -\frac{\partial p}{\partial z} - \rho g. \tag{2.14}$$

By hydrostatic approximation, we must drop all the remaining Coriolis terms that do not involve f to conserve angular momentum (Phillips, 1966).

We also separately treat horizontal and vertical strain for calculating frictional stresses. The vertical stress is usually parameterized as the vertical diffusion of momentum:

$$\mathbf{F}_{\text{fric}} = \frac{1}{\rho_0}\nabla \cdot \tau = \frac{\partial}{\partial z}\left(\nu_V \frac{\partial \mathbf{u}}{\partial z}\right) + \frac{1}{\rho_0}\nabla \cdot \tau_{\text{horizontal strain}}, \tag{2.15}$$

where ν_V is the vertical viscosity (essentially vertical diffusion of momentum).

c. Equations solved by a standard version of MRI.COM

With the above approximations, the momentum equation is now written as:

$$\begin{aligned}\frac{\partial u}{\partial t}&+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi uu)}{\partial\mu}+\frac{\partial(h_\mu vu)}{\partial\psi}\right\}+\frac{\partial(wu)}{\partial z}+\frac{v}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial\psi}u-\frac{\partial h_\psi}{\partial\mu}v\right)-fv\\&=-\frac{1}{\rho_0 h_\mu}\frac{\partial p}{\partial\mu}+\frac{1}{\rho_0}(\nabla\cdot\tau_{\text{horizontal strain}})_u+\frac{\partial}{\partial z}\left(\nu_V\frac{\partial u}{\partial z}\right),\end{aligned}\tag{2.16}$$

$$\begin{aligned}\frac{\partial v}{\partial t}&+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi uv)}{\partial\mu}+\frac{\partial(h_\mu vv)}{\partial\psi}\right\}+\frac{\partial(wv)}{\partial z}+\frac{u}{h_\mu h_\psi}\left(\frac{\partial h_\psi}{\partial\mu}v-\frac{\partial h_\mu}{\partial\psi}u\right)+fu\\&=-\frac{1}{\rho_0 h_\psi}\frac{\partial p}{\partial\psi}+\frac{1}{\rho_0}(\nabla\cdot\tau_{\text{horizontal strain}})_v+\frac{\partial}{\partial z}\left(\nu_V\frac{\partial v}{\partial z}\right).\end{aligned}\tag{2.17}$$

Continuity equation is written as:

$$\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi u)}{\partial\mu}+\frac{\partial(h_\mu v)}{\partial\psi}\right\}+\frac{\partial w}{\partial z}=0.\tag{2.18}$$

The equations for potential temperature and salinity are written as:

$$\frac{\partial\theta}{\partial t}=-\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi u\theta)}{\partial\mu}+\frac{\partial(h_\mu v\theta)}{\partial\psi}\right\}-\frac{\partial(w\theta)}{\partial z}-\nabla\cdot\mathbf{F}^\theta+Q^\theta,\tag{2.19}$$

$$\frac{\partial S}{\partial t}=-\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi uS)}{\partial\mu}+\frac{\partial(h_\mu vS)}{\partial\psi}\right\}-\frac{\partial(wS)}{\partial z}-\nabla\cdot\mathbf{F}^S+Q^S.\tag{2.20}$$

2.1.4 Boundary conditions

a. Momentum equation

■ Upper surface ($z=\eta$): At the sea surface, momentum flux enters the ocean in the form of wind stress (or stress from sea ice where sea ice exists):

$$\nu_V\left.\frac{\partial(u,v)}{\partial z}\right|_{z=\eta}=\frac{(\tau^\mu,\tau^\psi)}{\rho_0}.\tag{2.21}$$

Note that z is defined positive upward (the surface wind stress is positive into the ocean).

In the model algorithm, this is treated as a body force to the first level velocity. See Chapter 14 for details.

Surface fresh water flux is assumed to have zero velocity.

■ Bottom ($z=-H$): Bottom friction (τ_b^μ in zonal and τ_b^ψ in meridional direction) exists at the sea floor ($z=-H(\mu,\psi)$).

$$\nu_V\left.\frac{\partial(u,v)}{\partial z}\right|_{z=-H(\mu,\psi)}=-\frac{(\tau_b^\mu,\tau_b^\psi)}{\rho_0}.\tag{2.22}$$

In the model algorithm, this is treated as a body force to the bottom level (k = kbtm) velocity. See Chapter 8 for details.

■ Side wall: No slip condition is applied ($u=v=0$).

b. Temperature and Salinity

■ Upper surface ($z=\eta$): At the sea surface, heat and fresh water are exchanged with atmosphere and sea ice. Salt is also exchanged below sea ice. All these exchanges are expressed as surface fluxes and become surface boundary conditions.

The surface boundary conditions for temperature and salinity are expressed as follows:

$$\kappa_V\left.\frac{\partial\theta}{\partial z}\right|_{z=\eta}=F^\theta_{\text{surf}},\tag{2.23}$$

$$\kappa_V\left.\frac{\partial S}{\partial z}\right|_{z=\eta}=F^S_{\text{surf}},\tag{2.24}$$

where surface flux (F^α_{surf}) is defined positive downward (positive into the ocean).

■ Bottom ($z = -H$): At the sea floor ($z = -H(\mu, \psi)$), geothermal heating ($F^{\theta}_{\text{bottom}}$) and sediment trap ($F^{S}_{\text{bottom}}$) may affect temperature and salinity:

$$-\kappa_V \frac{\partial \theta}{\partial z}\bigg|_{z=-H(\mu,\psi)} = F^{\theta}_{\text{bottom}}, \quad -\kappa_V \frac{\partial S}{\partial z}\bigg|_{z=-H(\mu,\psi)} = F^{S}_{\text{bottom}}, \tag{2.25}$$

where bottom flux ($F^{\alpha}_{\text{bottom}}$) is defined positive upward (positive into the ocean).

■ Side wall: For any tracer, the adiabatic condition is applied at the side wall:

$$\frac{\partial \theta}{\partial n} = 0, \quad \frac{\partial S}{\partial n} = 0, \tag{2.26}$$

where n denotes the direction perpendicular to the wall. River discharge is expressed as a part of the surface fresh water flux.

c. Continuity equation

■ Upper surface ($z = \eta$): At the sea surface, vertical velocity is generated because a water parcel moves following the freely moving sea surface. Surface fresh water flux is explicitly incorporated into the boundary condition for the continuity equation.

$$w = \frac{d\eta}{dt} - (P - E + R + I) = \frac{\partial \eta}{\partial t} + u\frac{1}{h_\mu}\frac{\partial \eta}{\partial \mu} + v\frac{1}{h_\psi}\frac{\partial \eta}{\partial \psi} - (P - E + R + I), \tag{2.27}$$

where P is precipitation, E is evaporation, R is river discharge, and I is fresh water exchange with sea ice component.

■ Bottom ($z = -H$): At the sea floor, vertical velocity is generated because the water parcel moves following the bottom topography:

$$w = -\left(u\frac{1}{h_\mu}\frac{\partial H}{\partial \mu} + v\frac{1}{h_\psi}\frac{\partial H}{\partial \psi}\right). \tag{2.28}$$

■ Evolution equation for sea surface height (η): Using these boundary conditions, we obtain evolution equation for sea surface height η by vertically integrating the continuity equation (2.18),

$$\frac{\partial \eta}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial (h_\psi (H+\eta)\langle u \rangle)}{\partial \mu} + \frac{\partial (h_\mu (H+\eta)\langle v \rangle)}{\partial \psi}\right\} = P - E + R + I, \tag{2.29}$$

where $\langle (u, v) \rangle = \frac{1}{H+\eta}\int_{-H}^{\eta} (u, v) dz$.

d. Mixing at the surface boundary layer

Near the sea surface, strong vertical mixing may occur for stably stratified situations because of active turbulence. These processes occur on a small scale (< several meters) and cannot be resolved in a large scale hydrostatic model with typical horizontal and vertical resolutions. These processes are parameterized as enhanced vertical viscosity and diffusivity. The user chooses whether to set a high vertical viscosity and diffusivity *a priori* or to use a surface boundary layer model. MRI.COM supports several surface boundary layer models. In the surface boundary layer models, vertical viscosity and diffusivity are calculated every time step. See Chapter 15 for details.

2.1.5 Generalized vertical coordinates

From MRI.COM version 4, the standard choice for the vertical coordinate is z^*, which was first introduced by Adcroft and Campin (2004). Before showing a specific expression of governing equations in z^* coordinate, we consider generalized vertical coordinates (s). Note that generalized vertical coordinates employed by ocean models are not orthogonal. Horizontal velocities are not perpendicular to the vertical coordinate but perpendicular to the local gravitational field. The generalized vertical coordinate surface $s = s(\mu, \psi, z, t)$ is expressed as a smooth function of the original coordinate and time. We introduce a new scale factor z_s, which measures the thickness in the original depth coordinate of the unit length in the new coordinate:

$$z_s \equiv \frac{\partial z}{\partial s}\bigg|_{\mu\psi t}. \tag{2.30}$$

We further introduce vertical velocity $\dot{s}$ in generalized vertical coordinates.

Using a transformation rule presented by Adcroft and Campin (2004), we write the governing equations of the variables $(\mathbf{v}, \rho, \theta, S, p)$ in generalized vertical coordinates as follows:

$$
\begin{aligned}
&\frac{\partial (z_s u)}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial (z_s h_\psi u u)}{\partial \mu} + \frac{\partial (z_s h_\mu v u)}{\partial \psi}\right\} + \frac{\partial (z_s \dot{s} u)}{\partial s} + z_s \frac{v}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial \psi} u - \frac{\partial h_\psi}{\partial \mu} v\right) - z_s f v \\
&\qquad = -z_s \frac{1}{\rho_0 h_\mu}\frac{\partial p}{\partial \mu} - z_s \frac{\rho}{\rho_0 h_\mu}\frac{\partial (gz)}{\partial \mu} + z_s \frac{1}{\rho_0}(\nabla \cdot \tau_{\text{horizontal strain}})_u + z_s \frac{1}{z_s}\frac{\partial}{\partial s}\left(\frac{\nu_V}{z_s}\frac{\partial u}{\partial s}\right), \qquad (2.31)
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\partial (z_s v)}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial (z_s h_\psi u v)}{\partial \mu} + \frac{\partial (z_s h_\mu v v)}{\partial \psi}\right\} + \frac{\partial (z_s \dot{s} v)}{\partial s} + z_s \frac{u}{h_\mu h_\psi}\left(\frac{\partial h_\psi}{\partial \mu} v - \frac{\partial h_\mu}{\partial \psi} u\right) + z_s f u \\
&\qquad = -z_s \frac{1}{\rho_0 h_\psi}\frac{\partial p}{\partial \psi} - z_s \frac{\rho}{\rho_0 h_\psi}\frac{\partial (gz)}{\partial \psi} + z_s \frac{1}{\rho_0}(\nabla \cdot \tau_{\text{horizontal strain}})_v + z_s \frac{1}{z_s}\frac{\partial}{\partial s}\left(\frac{\nu_V}{z_s}\frac{\partial v}{\partial s}\right), \qquad (2.32)
\end{aligned}
$$

$$
\rho \frac{\partial (gz)}{\partial s} + \frac{\partial p}{\partial s} = 0, \qquad (2.33)
$$

$$
\frac{\partial z_s}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial (z_s h_\psi u)}{\partial \mu} + \frac{\partial (z_s h_\mu v)}{\partial \psi}\right\} + \frac{\partial (z_s \dot{s})}{\partial s} = 0, \qquad (2.34)
$$

$$
\frac{\partial (z_s \theta)}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial (z_s h_\psi u \theta)}{\partial \mu} + \frac{\partial (z_s h_\mu v \theta)}{\partial \psi}\right\} + \frac{\partial (z_s \dot{s} \theta)}{\partial s} = -z_s \nabla \cdot \mathbf{F}_\theta + z_s Q^\theta, \qquad (2.35)
$$

$$
\frac{\partial (z_s S)}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial (z_s h_\psi u S)}{\partial \mu} + \frac{\partial (z_s h_\mu v S)}{\partial \psi}\right\} + \frac{\partial (z_s \dot{s} S)}{\partial s} = -z_s \nabla \cdot \mathbf{F}_S + z_s Q^S, \qquad (2.36)
$$

and

$$
\rho = \rho(\theta, S, p). \qquad (2.37)
$$

Evolution equation for sea surface height is obtained by vertically integrating the continuity equation (2.34) and considering the boundary condition. It has the same form as in the depth coordinate system:

$$
\frac{\partial (\eta + H)}{\partial t} + \frac{1}{h_\mu h_\psi}\left[\frac{\partial (h_\psi U)}{\partial \mu} + \frac{\partial (h_\mu V)}{\partial \psi}\right] = P - E + R + I, \qquad (2.38)
$$

where

$$
(U, V) = \int_{s(z=-H)}^{s(z=\eta)} [z_s (u, v)] ds. \qquad (2.39)
$$

2.1.6 z^* coordinate

a. Definition and boundary condition

Definition of the new vertical coordinate z^* is as follows:

$$
z^* = \sigma(\mu, \psi, z, t) H(\mu, \psi) = \frac{z - \eta(\mu, \psi, t)}{H(\mu, \psi) + \eta(\mu, \psi, t)} H(\mu, \psi), \qquad (2.40)
$$

where $z = -H(\mu, \psi)$ is sea floor and η is the free surface displacement. σ is a symbol of the conventional terrain following vertical coordinate. In z^* coordinate, σ is further scaled by sea floor depth $H(\mu, \psi)$, which makes z^* coordinate more similar to the depth coordinate rather than the terrain following coordinate.

The scaling factor $z_s = \frac{\partial z}{\partial z^*}$ is

$$
\frac{\partial z}{\partial z^*} = \frac{H + \eta}{H}. \qquad (2.41)
$$

The governing equations in z^* vertical coordinate are in the same form as equations (2.31) through (2.38), with z_s replaced by $z_{z^*} = \frac{H+\eta}{H}$.

The vertical velocity in this coordinate system is expressed as w^*, instead of $\dot{s}$. Here, w^* has the following relation with the vertical velocity w of the depth coordinate z,

$$
w^* \equiv D_t z^* = \frac{H}{H + \eta}\left(w - \left(1 + \frac{z^*}{H}\right) D_t \eta + \frac{z^* \eta}{H^2} \mathbf{v}_h \cdot \nabla H\right), \qquad (2.42)
$$

where D_t represents the material time derivative operator in any coordinate system,

$$D_t \equiv \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla = \frac{\partial}{\partial t} + \frac{u}{h_\mu}\frac{\partial}{\partial \mu} + \frac{v}{h_\psi}\frac{\partial}{\partial \psi} + \frac{w}{z_s}\frac{\partial}{\partial s}. \tag{2.43}$$

Horizontal boundary conditions are unchanged by this coordinate transformation. Vertical boundary conditions need some discussion.

Sea floor ($z = -H$) in z^* coordinate is also $-H$.

$$z = -H(\mu, \psi) \quad \Rightarrow \quad z^* = -H(\mu, \psi). \tag{2.44}$$

Thus the kinematic boundary condition at sloping bottom is

$$w_{z=-H} = -\mathbf{v}_h \cdot \nabla H \quad \Rightarrow \quad w^*_{z^*=-H} = -\mathbf{v}_h \cdot \nabla H, \tag{2.45}$$

where $\mathbf{v}_h = (u, v, 0)$ is the horizontal component of velocity at $z = -H(\mu, \psi)$.

Sea surface ($z = \eta$) in z^* coordinate is

$$z = \eta(x, y, t) \quad \Rightarrow \quad z^* = 0. \tag{2.46}$$

Sea surface is fixed in time in z^* frame. In other words, model domain and grid cells are logically fixed in time. We do not have to concern about the moving sea surface and vanishing of the first layer thickness [§].

The kinematic boundary condition at the sea surface is

$$w_{z=\eta} = D_t\eta - (P - E + R + I) \quad \Rightarrow w^*_{z^*=0} = -\frac{H}{H+\eta}(P - E + R + I). \tag{2.47}$$

For example, precipitation ($P > 0$) penetrates the ocean as a scaled downward vertical velocity.

b. Pressure gradient term

Horizontal momentum equations in generalized vertical coordinates (2.31), (2.32) involve both pressure gradient and geopotential gradient term. Pressure gradient error appears when these terms do not cancel each other. That said, pressure gradient error is not a big issue for z^* coordinate because horizontal gradient of a constant z^*-surface is usually very small. However, source of errors must be kept as small as possible. For this purpose, we first separate density into constant and its deviation

$$\rho = \rho_0 + \rho'(x, y, z^*, t). \tag{2.48}$$

Pressure is also separated in the same manner ($p = p_0 + p'$) and the hydrostatic relation (2.33) is separated into two equations

$$\partial_{z^*} p_0 = -g\left(\frac{H+\eta}{H}\right)\rho_0 \ \text{ and } \ p_0(z^* = 0) = 0, \tag{2.49}$$

$$\partial_{z^*} p' = -g\left(\frac{H+\eta}{H}\right)\rho' \ \text{ and } \ p'(z^* = 0) = p_a, \tag{2.50}$$

where p_a is atmospheric pressure[¶]. Specific expression for p_0 is obtained by integrating (2.49)

$$p_0(z^*) = p_0(x, y, z, t) = -g\rho_0\frac{H+\eta}{H}z^* = g\rho_0(\eta - z) = g\rho_0\eta - \rho_0\Phi. \tag{2.51}$$

Rewriting pressure gradient term by using this, we have

$$\frac{1}{\rho_0}\nabla_{z^*}(p_0 + p') + \frac{\rho}{\rho_0}\nabla_{z^*}\Phi = g\nabla\eta - \nabla_{z^*}\Phi + \frac{1}{\rho_0}\nabla_{z^*}p' + \frac{\rho_0+\rho'}{\rho_0}\nabla_{z^*}\Phi \tag{2.52}$$

$$= g\nabla\eta + \frac{1}{\rho_0}\nabla_{z^*}p' + \frac{\rho'}{\rho_0}\nabla_{z^*}\Phi. \tag{2.53}$$

[§] However, sea surface is not allowed to touch the see floor ($H + \eta \leq 0$), which is a local problem. This is a restrictive condition inherent to this coordinate system.

[¶] Ice-loading effect (e.g., Campin et al., 2008) has not been included in MRI.COM

It is noticed that time-independent terms are removed. Because geopotential is $\Phi = gz$, the momentum equation is expressed as

$$\partial_t \mathbf{v}_h + \mathbf{v}_h \cdot \nabla_{z^*} \mathbf{v}_h + w^* \partial_{z^*} \mathbf{v}_h + f\hat{\mathbf{z}} \times \mathbf{v}_h + g\nabla\eta + \frac{1}{\rho_0}\nabla_{z^*} p' + \frac{g\rho'}{\rho_0}\nabla_{z^*} z = \mathbf{F} \tag{2.54}$$

Perturbation pressure is obtained by integrating (2.50) as

$$p'(z^*) = p_a + g\int_{z^*}^{0} \rho' z_{z^*} dz^* = p_a + g\int_{z(z^*)}^{\eta} \rho' dz. \tag{2.55}$$

Then (2.54) becomes

$$\begin{aligned} &\partial_t \mathbf{v}_h + \mathbf{v}_h \cdot \nabla_{z^*} \mathbf{v}_h + w^* \partial_{z^*} \mathbf{v}_h + f\hat{\mathbf{z}} \times \mathbf{v}_h \\ &\quad + \underbrace{\frac{1}{\rho_0}\nabla_{z^*}(p_a + g\rho_0\eta)}_{\text{fast}} + \underbrace{\frac{1}{\rho_0}\nabla_{z^*}\left(g\int_{z(z^*)}^{\eta} \rho' dz\right) + \frac{g\rho'}{\rho_0}\nabla_{z^*} z}_{\text{slow}} = \mathbf{F}. \end{aligned} \tag{2.56}$$

We separate the pressure gradient term into barotropic (fast) and baroclinic (slow) component in preparation for the split-explicit solution method for equations of motion.

2.2 Numerical Methods

2.2.1 Discretization and finite volume method

To solve the primitive equations formulated in the previous section, the equations are projected on a discrete lattice and then advanced for a discrete time interval.

Because primary choice of the vertical coordinate of MRI.COM is z^*, a logically fixed (but actually moving) Eulerian lattice is arranged. A detailed description of the grid arrangement is given in Chapter 3. The equations are then volume integrated over a discrete model grid cell. This approach is called a finite volume approach or sometimes a flux form expression in this manual.

A vertically integrated expression for the primitive equations is useful for describing the solution procedure. These are called semi-discrete equations (Griffies, 2004). The body force and metric terms will be simply multiplied by the grid width. The material transport and subgrid-scale flux terms need some attention.

In this section the vertical coordinate of z^* is written as a general symbol s. The material transport of any quantity α that commonly appears in the prognostic equations,

$$\frac{\partial(z_s\alpha)}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi z_s u\alpha)}{\partial\mu} + \frac{\partial(h_\mu z_s v\alpha)}{\partial\psi}\right\} + \frac{\partial(z_s\dot{s}\alpha)}{\partial s} \tag{2.57}$$

is vertically integrated over a $(k-\frac{1}{2})$-th grid cell bounded by s_{k-1} and s_k to give

$$\begin{aligned} &\int_{s_k}^{s_{k-1}} \frac{\partial(z_s\alpha)}{\partial t} ds + \int_{s_k}^{s_{k-1}} \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi z_s u\alpha)}{\partial\mu} + \frac{\partial(h_\mu z_s v\alpha)}{\partial\psi}\right\} ds + \int_{s_k}^{s_{k-1}} \frac{\partial(z_s\dot{s}\alpha)}{\partial s} ds \\ &= \frac{\partial}{\partial t}\left(\int_{s_k}^{s_{k-1}} (z_s\alpha) ds\right) + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial\mu}\left(\int_{s_k}^{s_{k-1}} h_\psi z_s u\alpha\, ds\right) + \frac{\partial}{\partial\psi}\left(\int_{s_k}^{s_{k-1}} h_\mu z_s v\alpha\, ds\right)\right\} \\ &\quad - \left(z_s\frac{\partial s_{k-1}}{\partial t} + \frac{z_s u(s_{k-1})}{h_\mu}\frac{\partial s_{k-1}}{\partial\mu} + \frac{z_s v(s_{k-1})}{h_\psi}\frac{\partial s_{k-1}}{\partial\psi} - z_s\dot{s}(s_{k-1})\right)\alpha(s_{k-1}) \\ &\quad + \left(z_s\frac{\partial s_k}{\partial t} + \frac{z_s u(s_k)}{h_\mu}\frac{\partial s_k}{\partial\mu} + \frac{z_s v(s_k)}{h_\psi}\frac{\partial s_k}{\partial\psi} - z_s\dot{s}(s_k)\right)\alpha(s_k). \end{aligned} \tag{2.58}$$

The first line on the r.h.s. is expressed in a semi-discrete form as

$$\frac{\partial}{\partial t}\left(\Delta z\alpha\right)_{k-\frac{1}{2}} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial\mu}\left(h_\psi\Delta z u\alpha\right)_{k-\frac{1}{2}} + \frac{\partial}{\partial\psi}\left(h_\mu\Delta z v\alpha\right)_{k-\frac{1}{2}}\right\}, \tag{2.59}$$

where any quantity is assumed to have a uniform distribution within a grid cell.

Using $\dot{s} \equiv D_t s$ and (2.43), the last two lines are reduced to the difference between vertical advective fluxes:

$$\text{the last two lines of (2.58)} = z_s\dot{s}(s_{k-1})\alpha(s_{k-1}) - z_s\dot{s}(s_k)\alpha(s_k). \tag{2.60}$$

For the sea surface ($k = 1; s_0 = 0$) and the bottom ($k = kbtm; s_{kbtm} = -H$), kinematic conditions (2.47) and (2.45) are used to give

$$\text{the last two lines of (2.58)} = -(P - E + R + I)\alpha(0) - z_s\dot{s}(s_1)\alpha(s_1) \tag{2.61}$$

at the surface and

$$\text{the last two lines of (2.58)} = z_s\dot{s}(s_{kbtm-1})\alpha(s_{kbtm-1}) - 0 \tag{2.62}$$

at the bottom (bottom term vanishes to give no advective fluxes through the sea floor).

Similarly, the vertical integral of the divergence of the subgrid-scale fluxes gives

$$\begin{aligned}
&\int_{s_k}^{s_{k-1}} \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi z_s F_\mu)}{\partial\mu} + \frac{\partial(h_\mu z_s F_\psi)}{\partial\psi}\right\} ds + \int_{s_k}^{s_{k-1}} \frac{\partial F_s}{\partial s} ds \\
&\quad= \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial\mu}\Big(h_\psi \Delta z F_\mu\Big)_{k-\frac{1}{2}} + \frac{\partial}{\partial\psi}\Big(h_\mu \Delta z F_\psi\Big)_{k-\frac{1}{2}}\right\} \\
&\quad\quad - \left(\frac{F_\mu(s)}{h_\mu}\frac{\partial s}{\partial\mu} + \frac{F_\psi(s)}{h_\psi}\frac{\partial s}{\partial\psi} - F_s(s)\right)_{k-1} + \left(\frac{F_\mu(s)}{h_\mu}\frac{\partial s}{\partial\mu} + \frac{F_\psi(s)}{h_\psi}\frac{\partial s}{\partial\psi} - F_s(s)\right)_{k}.
\end{aligned} \tag{2.63}$$

In summary, the material transport and subgrid-scale flux parts are integrated for a vertical grid cell to give the semi-discrete expression on the r.h.s.,

$$\begin{aligned}
&\int_{s_k}^{s_{k-1}} \frac{\partial(z_s\alpha)}{\partial t} ds + \int_{s_k}^{s_{k-1}} \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi z_s u\alpha)}{\partial\mu} + \frac{\partial(h_\mu z_s v\alpha)}{\partial\psi}\right\} ds + \int_{s_k}^{s_{k-1}} \frac{\partial(z_s\dot{s}\alpha)}{\partial s} ds \\
&\quad + \int_{s_k}^{s_{k-1}} \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi z_s F_\mu)}{\partial\mu} + \frac{\partial(h_\mu z_s F_\psi)}{\partial\psi}\right\} ds + \int_{s_k}^{s_{k-1}} \frac{\partial F_s}{\partial s} ds \\
&= \frac{\partial}{\partial t}\Big(\Delta z\alpha\Big)_{k-\frac{1}{2}} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial\mu}\Big(h_\psi \Delta z u\alpha\Big)_{k-\frac{1}{2}} + \frac{\partial}{\partial\psi}\Big(h_\mu \Delta z v\alpha\Big)_{k-\frac{1}{2}}\right\} + z_s\dot{s}(s_{k-1})\alpha(s_{k-1}) - z_s\dot{s}(s_k)\alpha(s_k) \\
&\quad + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial\mu}\Big(h_\psi \Delta z F_\mu\Big)_{k-\frac{1}{2}} + \frac{\partial}{\partial\psi}\Big(h_\mu \Delta z F_\psi\Big)_{k-\frac{1}{2}}\right\} + F_s(s_{k-1}) - F_s(s_k).
\end{aligned} \tag{2.64}$$

The quantity

$$F_{\text{surf}}^{\alpha} = (P - E + R + I)\alpha(0) + \left(\frac{F_\mu(s_0)}{h_\mu}\frac{\partial s_0}{\partial\mu} + \frac{F_\psi(s_0)}{h_\psi}\frac{\partial s_0}{\partial\psi} - F_s(s_0)\right) = (P - E + R + I)\alpha(0) - F_s(0) \tag{2.65}$$

taken from (2.61) and (2.63) may be regarded as a surface forcing term and corresponds to the surface flux (positive downward) given in the previous section. The first term on the r.h.s. of (2.65) is the tracer transport by the fresh water flux, and the second term is the microstructure flux calculated by subgrid-scale parameterizations such as bulk formula. Similarly, geothermal heating may be incorporated as microstructure flux from the sea floor ($\mathbf{F}_h(-H) \cdot \nabla_h H + F_s(-H)$).

2.2.2 Mode splitting in the momentum equation

Here we consider to solve the momentum equation with hydrostatic and Boussinesq approximation. Equations are (2.31), (2.32), and (2.33). To integrate these equations in time, we should know the instantaneous vector field, pressure, and geopotential. For the vector field, we use one at the previous time level. We obtain the pressure field by integrating the hydrostatic equation vertically, in which the sea surface height ($z = \eta$) is needed. The sea surface height is also needed for geopotential. To obtain the surface height, we should solve vertically integrated continuity equation (2.38).

The rise and fall of the sea level causes external gravity waves whose phase speed is two orders of magnitude greater than that of other waves. This will impose tight limits on the time intervals due to the CFL condition. We want to separate or remove external gravity waves, because they are usually not important when a target phenomenon has a longer time scale.

Historically, external gravity waves were removed from the model by prohibiting the vertical movement of the sea surface (rigid-lid approximation). In this case, the vertically integrated equations result in a vorticity equation in the form of the Poisson equation, solved by relaxation methods. The surface pressure is then diagnostically obtained as the pressure pushing up the lid.

After the sea surface was allowed to move vertically, the problem of fast external modes was resolved by separating the barotropic mode from the baroclinic mode. The barotropic mode is solved explicitly with a short time step. The baroclinic mode can take a longer time step by reflecting a state of the barotropic mode in which high-frequency components are filtered out. Since this free surface option is more suitable for parallel computation than the rigid-lid approximation, only the free surface option is supported by MRI.COM. The essence of the split-explicit free surface formulation is explained below. See Chapter 7 for details.

a. Barotropic mode

Following (2.39), if we put

$$U = \int_{-H}^{0} z_s u ds = \sum_{k=1}^{N} u_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}, \quad V = \int_{-H}^{0} z_s v ds = \sum_{k=1}^{N} v_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}, \tag{2.66}$$

and separate fast and slow modes of the pressure gradient as in (2.56), then the vertically summed semi-discrete momentum equations are

$$\frac{\partial U}{\partial t} - fV = -\frac{(\eta + H)}{\rho_0 h_\mu} \frac{\partial (p_a + g\rho_0 \eta)}{\partial \mu} + X, \tag{2.67}$$

$$\frac{\partial V}{\partial t} + fU = -\frac{(\eta + H)}{\rho_0 h_\psi} \frac{\partial (p_a + g\rho_0 \eta)}{\partial \psi} + Y, \tag{2.68}$$

where p_a is the atmospheric pressure at sea surface. Density ρ has been separated into mean ρ_0 and perturbation ρ', and

$$\begin{aligned} X =& - \nabla_H \cdot \Big(\sum_{k=1}^{N} (\Delta z(u,\ v)u)_{k-\frac{1}{2}} \Big) - \sum_{k=1}^{N} \Big[\frac{v}{h_\mu h_\psi} \Big(\frac{\partial h_\mu}{\partial \psi} u - \frac{\partial h_\psi}{\partial \mu} v \Big) \Big]_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \\ & - \sum_{k=1}^{N} \Big[\frac{1}{\rho_0} \frac{1}{h_\mu} \int_{s_{k-\frac{1}{2}}}^{0} z_s g \rho_\mu ds' \Big] \Delta z_{k-\frac{1}{2}} - \frac{g}{\rho_0 h_\mu} \sum_{k=1}^{N} [\rho' z_\mu] \Delta z_{k-\frac{1}{2}} + \sum_{k=1}^{N} (\Delta z F^\mu_{\mathrm{horz}})_{k-\frac{1}{2}} + F^\mu_{\mathrm{surf}} \Delta z_{\frac{1}{2}} + F^\mu_{\mathrm{bottom}} \Delta z_{N-\frac{1}{2}} \\ & (\equiv \sum_{k=1}^{N} F_\mu), \end{aligned} \tag{2.69}$$

$$\begin{aligned} Y =& - \nabla_H \cdot \Big(\sum_{k=1}^{N} (\Delta z(u,\ v)v)_{k-\frac{1}{2}} \Big) + \sum_{k=1}^{N} \Big[\frac{u}{h_\mu h_\psi} \Big(\frac{\partial h_\mu}{\partial \psi} u - \frac{\partial h_\psi}{\partial \mu} v \Big) \Big]_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \\ & - \sum_{k=1}^{N} \Big[\frac{1}{\rho_0} \frac{1}{h_\psi} \int_{s_{k-\frac{1}{2}}}^{0} z_s g \rho_\psi ds' \Big] \Delta z_{k-\frac{1}{2}} - \frac{g}{\rho_0 h_\psi} \sum_{k=1}^{N} [\rho' z_\psi] \Delta z_{k-\frac{1}{2}} + \sum_{k=1}^{N} (\Delta z F^\psi_{\mathrm{horz}})_{k-\frac{1}{2}} + F^\psi_{\mathrm{surf}} \Delta z_{\frac{1}{2}} + F^\psi_{\mathrm{bottom}} \Delta z_{N-\frac{1}{2}} \\ & (\equiv \sum_{k=1}^{N} F_\psi). \end{aligned} \tag{2.70}$$

The vertically integrated continuity equation is given by

$$\frac{\partial \eta}{\partial t} + \frac{1}{h_\mu h_\psi} \Big(\frac{\partial (h_\psi U)}{\partial \mu} + \frac{\partial (h_\mu V)}{\partial \psi} \Big) = (P - E + R + I). \tag{2.71}$$

We solve these equations for U, V, and η with a short time step constrained by the phase speed of the external gravity waves.

On returning (U, V) to the baroclinic mode, the effect of high-frequency phenomena with time scale shorter than the baroclinic time step is filtered out by a weighted averaging, which is symbolically expressed as follows:

$$\langle U \rangle^{n+1} = U^n + \Delta t_{\mathrm{cl}} f \langle\langle V \rangle\rangle^{n+\frac{1}{2}} - \Delta t_{\mathrm{cl}} \Big\langle\Big\langle \frac{(\eta + H)}{\rho_0 h_\mu} \frac{\partial (p_a + \rho_0 g \eta)}{\partial \mu} \Big\rangle\Big\rangle^{n+\frac{1}{2}} + \Delta t_{\mathrm{cl}} X^n, \tag{2.72}$$

$$\langle V \rangle^{n+1} = V^n - \Delta t_{\mathrm{cl}} f \langle\langle U \rangle\rangle^{n+\frac{1}{2}} - \Delta t_{\mathrm{cl}} \Big\langle\Big\langle \frac{(\eta + H)}{\rho_0 h_\psi} \frac{\partial (p_a + \rho_0 g \eta)}{\partial \psi} \Big\rangle\Big\rangle^{n+\frac{1}{2}} + \Delta t_{\mathrm{cl}} Y^n, \tag{2.73}$$

where $\langle \cdot \rangle^{n+1} \equiv \sum_{m=1}^{M^*} b_m (\cdot)^{m-\frac{1}{2}}$ and $\langle\langle \cdot \rangle\rangle^{n+\frac{1}{2}} \equiv \sum_{m=1}^{M^*} b_m \sum_{m'=1}^{m} (\cdot)^{m'-\frac{1}{2}}$, with $m \in \{1, M^*\}$ representing the barotropic time level, n representing the baroclinic time level, Δt_{cl} being the baroclinic time step, and b_m being weighting factors explained in Chapter 7. The above equations are used to obtain absolute velocity with equations of the baroclinic mode described in this subsection (see below). See Sec. (7.7) for detais of the formulation.

b. Baroclinic mode

Velocity is decomposed into a barotropic component based on (2.39), $(\bar{u},\bar{v})$, and a baroclinic component, $(\hat{u},\hat{v})$, as follows:

$$u = \hat{u} + \bar{u}, \qquad v = \hat{v} + \bar{v}, \tag{2.74}$$

where

$$\bar{u} = \frac{1}{H}U, \qquad \bar{v} = \frac{1}{H}V. \tag{2.75}$$

By solving the baroclinc mode $(\hat{u},\hat{v})$, absolute velocity (u,v) at the next step is obtained.

First, we drop the fast pressure gradient term from the momentum equation, (2.31) and (2.32), since it is a barotropic component:

$$\begin{aligned}
&\frac{\partial(z_s u)}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi uu)}{\partial\mu}+\frac{\partial(z_s h_\mu vu)}{\partial\psi}\right\}+\frac{\partial(z_s\dot{s}u)}{\partial s}+z_s\frac{v}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial\psi}u-\frac{\partial h_\psi}{\partial\mu}v\right)-z_s f v\\
&\quad=-z_s\frac{1}{\rho_0 h_\mu}\frac{\partial}{\partial\mu}\left(g\int_{z(z^*)}^{\eta}\rho' dz\right)-z_s\frac{g\rho'}{\rho_0 h_\mu}\frac{\partial z}{\partial\mu}\\
&\qquad+z_s\frac{1}{\rho_0}(\nabla\cdot\tau_{\text{horizontal strain}})_u+z_s\frac{1}{z_s}\frac{\partial}{\partial s}\left(\frac{\nu_V}{z_s}\frac{\partial u}{\partial s}\right),
\end{aligned}\tag{2.76}$$

$$\begin{aligned}
&\frac{\partial(z_s v)}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi uv)}{\partial\mu}+\frac{\partial(z_s h_\mu vv)}{\partial\psi}\right\}+\frac{\partial(z_s\dot{s}v)}{\partial s}+z_s\frac{u}{h_\mu h_\psi}\left(\frac{\partial h_\psi}{\partial\mu}v-\frac{\partial h_\mu}{\partial\psi}u\right)+z_s f u\\
&\quad=-z_s\frac{1}{\rho_0 h_\psi}\frac{\partial}{\partial\psi}\left(g\int_{z(z^*)}^{\eta}\rho' dz\right)-z_s\frac{g\rho'}{\rho_0 h_\psi}\frac{\partial z}{\partial\psi}\\
&\qquad+z_s\frac{1}{\rho_0}(\nabla\cdot\tau_{\text{horizontal strain}})_v,+z_s\frac{1}{z_s}\frac{\partial}{\partial s}\left(\frac{\nu_V}{z_s}\frac{\partial v}{\partial s}\right).
\end{aligned}\tag{2.77}$$

Using these equations, provisional velocity (u',v') at the next step is calculated. In discrete forms,

$$\frac{u'_{k-\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}}-u^n_{k-\frac{1}{2}}\Delta z^n_{k-\frac{1}{2}}}{\Delta t_{\text{cl}}}=f[v_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}}+F^n_\mu, \tag{2.78}$$

$$\frac{v'_{k-\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}}-v^n_{k-\frac{1}{2}}\Delta z^n_{k-\frac{1}{2}}}{\Delta t_{\text{cl}}}=-f[u_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}}+F^n_\psi, \tag{2.79}$$

where

$$\begin{aligned}
F^n_\mu=&-\nabla_H\cdot\Big(\Delta z(u,\ v)u\Big)_{k-\frac{1}{2}}-\left[\frac{v}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial\psi}u-\frac{\partial h_\psi}{\partial\mu}v\right)\right]_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}\\
&-\left[\frac{1}{\rho_0}\frac{1}{h_\mu}\int_{s_{k-\frac{1}{2}}}^{0}z_s g\rho_\mu ds'\right]\Delta z_{k-\frac{1}{2}}-\frac{g}{\rho_0 h_\mu}\sum_{k=1}^{N}[\rho' z_\mu]\Delta z_{k-\frac{1}{2}}+(\Delta z F^\mu_{\text{horz}})_{k-\frac{1}{2}}-F^\mu_{\text{vert}\,k-1}+F^\mu_{\text{vert}\,k},
\end{aligned}\tag{2.80}$$

$$\begin{aligned}
F^n_\psi=&-\nabla_H\cdot\Big(\Delta z(u,\ v)v\Big)_{k-\frac{1}{2}}+\left[\frac{u}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial\psi}u-\frac{\partial h_\psi}{\partial\mu}v\right)\right]_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}\\
&-\left[\frac{1}{\rho_0}\frac{1}{h_\psi}\int_{s_{k-\frac{1}{2}}}^{0}z_s g\rho_\psi ds'\right]\Delta z_{k-\frac{1}{2}}-\frac{g}{\rho_0 h_\psi}\sum_{k=1}^{N}[\rho' z_\psi]\Delta z_{k-\frac{1}{2}}+(\Delta z F^\psi_{\text{horz}})_{k-\frac{1}{2}}-F^\psi_{\text{vert}\,k-1}+F^\psi_{\text{vert}\,k}.
\end{aligned}\tag{2.81}$$

A method for solving these equations is explained in Chapter 8.

Next, using (u',v'), baroclinic component $(\hat{u},\hat{v})$ and absolute velocity (u,v) at the next step are calculated. Summing over the whole water column gives

$$\frac{\sum_{k=1}^{N}(u'_{k-\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}})-\sum_{k=1}^{N}(u^n_{k-\frac{1}{2}}\Delta z^n_{k-\frac{1}{2}})}{\Delta t_{\text{cl}}}=f\sum_{k=1}^{N}[v_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}}+X^n, \tag{2.82}$$

$$\frac{\sum_{k=1}^{N}(v'_{k-\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}})-\sum_{k=1}^{N}(v^n_{k-\frac{1}{2}}\Delta z^n_{k-\frac{1}{2}})}{\Delta t_{\text{cl}}}=-f\sum_{k=1}^{N}[u_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}}+Y^n. \tag{2.83}$$

X^n and Y^n are removed by using the equations from the barotropic mode (2.72), (2.73) to give

$$\frac{\sum_{k=1}^{N}(\langle u\rangle^{n+1}\Delta z^{n+1}_{k-\frac{1}{2}}) - \sum_{k=1}^{N}(u'_{k-\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}})}{\Delta t_{\mathrm{cl}}} = f\langle\langle V\rangle\rangle^{n+\frac{1}{2}} - f\sum_{k=1}^{N}[v_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} - \left\langle\left\langle\frac{(\eta+H)}{\rho_0 h_\mu}\frac{\partial(p_a+\rho_0 g\eta)}{\partial\mu}\right\rangle\right\rangle^{n+\frac{1}{2}}, \tag{2.84}$$

$$\frac{\sum_{k=1}^{N}(\langle v\rangle^{n+1}\Delta z^{n+1}_{k-\frac{1}{2}}) - \sum_{k=1}^{N}(v'_{k-\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}})}{\Delta t_{\mathrm{cl}}} = -f\langle\langle U\rangle\rangle^{n+\frac{1}{2}} + f\sum_{k=1}^{N}[u_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} - \left\langle\left\langle\frac{(\eta+H)}{\rho_0 h_\psi}\frac{\partial(p_a+\rho_0 g\eta)}{\partial\psi}\right\rangle\right\rangle^{n+\frac{1}{2}}. \tag{2.85}$$

where $\langle u\rangle^{n+1} = \langle U\rangle^{n+1}/(\eta^{n+1}+H)$ and $\langle v\rangle^{n+1} = \langle V\rangle^{n+1}/(\eta^{n+1}+H)$. This is combined with (2.78) and (2.79) to give

$$\begin{aligned}\frac{\left(u'_{k-\frac{1}{2}} - \overline{u'}^{z} + \langle u\rangle^{n+1}\right)\Delta z^{n+1}_{k-\frac{1}{2}} - u^n_{k-\frac{1}{2}}\Delta z^n_{k-\frac{1}{2}}}{\Delta t_{\mathrm{cl}}} =& f\left[v_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}\right]^{n+\frac{1}{2}} - \overline{f[v]^{n+\frac{1}{2}}}^{z}\Delta z^{n+1}_{k-\frac{1}{2}} + f\langle\langle v\rangle\rangle^{n+\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}}\\ &- \frac{\Delta z^{n+1}_{k-\frac{1}{2}}}{\eta^{n+1}+H}\left\langle\left\langle\frac{(\eta+H)}{\rho_0 h_\mu}\frac{\partial(p_a+\rho_0 g\eta)}{\partial\mu}\right\rangle\right\rangle^{n+\frac{1}{2}} + F^n_\mu\end{aligned} \tag{2.86}$$

$$\begin{aligned}\frac{\left(v'_{k-\frac{1}{2}} - \overline{v'}^{z} + \langle v\rangle^{n+1}\right)\Delta z^{n+1}_{k-\frac{1}{2}} - v^n_{k-\frac{1}{2}}\Delta z^n_{k-\frac{1}{2}}}{\Delta t_{\mathrm{cl}}} =& - f\left[u_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}\right]^{n+\frac{1}{2}} + \overline{f[u]^{n+\frac{1}{2}}}^{z}\Delta z^{n+1}_{k-\frac{1}{2}} - f\langle\langle u\rangle\rangle^{n+\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}}\\ &- \frac{\Delta z^{n+1}_{k-\frac{1}{2}}}{\eta^{n+1}+H}\left\langle\left\langle\frac{(\eta+H)}{\rho_0 h_\psi}\frac{\partial(p_a+\rho_0 g\eta)}{\partial\psi}\right\rangle\right\rangle^{n+\frac{1}{2}} + F^n_\psi,\end{aligned} \tag{2.87}$$

where $\overline{(\ldots)}^{z}$ denotes the thickness weighted vertical average.

Since $\overline{f[v]^{n+\frac{1}{2}}}^{z}\Delta z^{n+1}_{k-\frac{1}{2}} \sim f\langle\langle v\rangle\rangle^{n+\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}}$, $\overline{f[u]^{n+\frac{1}{2}}}^{z}\Delta z^{n+1}_{k-\frac{1}{2}} \sim f\langle\langle u\rangle\rangle^{n+\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}}$, and the fourth terms on the r. h. s. are the surface pressure gradient, we may regard $u' - \overline{u'}^{z} + \langle u\rangle^{n+1}$ and $v' - \overline{v'}^{z} + \langle v\rangle^{n+1}$ as the real updated velocity for time level $n+1$, the baroclinic component is expressed as $\hat{u} = u' - \overline{u'}$ and $\hat{v} = v' - \overline{v'}$.

To summarize, we first solve for (u', v') using (2.78) and (2.79), and then compute the baroclinic component by $\hat{u} = u' - \overline{u'}$ and $\hat{v} = v' - \overline{v'}$. The absolute velocity is obtained by $u = \hat{u} + \bar{u}$ and $v = \hat{v} + \bar{v}$.

2.3 Appendix

2.3.1 Acceleration method

It usually takes several thousand years before the global thermohaline circulation reaches a steady state under (quasi-)steady forcing. The limiting factor for the time step is the phase speed of external gravity waves (~ 200 m/s) for the barotropic mode and that of internal gravity waves (~ a few m/s) for the baroclinic mode. A several-thousand-year integration will not be a workable exercise as long as we are restricted by this criterion in determining the time step. Bryan (1984) proposed a method to accelerate the ocean's convergence to equilibrium. In this method, the phase speed of waves is reduced by modifying the governing equations, and a thermally balanced state is quickly reached by reducing the specific heat.

Specifically, they are achieved by multiplying a constant to the tendency terms (α for momentum and γ for tracers) to increase inertia and to reduce specific heat. When a steady state is reached in these equations, we expect that the same balance as the undistorted equations will be obtained, because the only difference between the distorted and undistorted equations are tendency terms, which are expected to be zero in the steady state.

The modified momentum equation is given by

$$\begin{aligned}\alpha\frac{\partial u}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi uu)}{\partial\mu} + \frac{\partial(h_\mu vu)}{\partial\psi}\right\} + \frac{\partial(wu)}{\partial z} + \frac{v}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial\psi}u - \frac{\partial h_\psi}{\partial\mu}v\right) - fv&\\ = -\frac{1}{\rho_0 h_\mu}\frac{\partial p}{\partial\mu} + \frac{1}{\rho_0}(\nabla\cdot\tau_{\text{horizontal strain}})_u + \frac{\partial}{\partial z}\left(\nu_V\frac{\partial u}{\partial z}\right),&\end{aligned} \tag{2.88}$$

$$\begin{aligned}\alpha\frac{\partial v}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi uv)}{\partial\mu} + \frac{\partial(h_\mu vv)}{\partial\psi}\right\} + \frac{\partial(wv)}{\partial z} + \frac{u}{h_\mu h_\psi}\left(\frac{\partial h_\psi}{\partial\mu}v - \frac{\partial h_\mu}{\partial\psi}u\right) + fu&\\ = -\frac{1}{\rho_0 h_\psi}\frac{\partial p}{\partial\psi} + \frac{1}{\rho_0}(\nabla\cdot\tau_{\text{horizontal strain}})_v + \frac{\partial}{\partial z}\left(\nu_V\frac{\partial v}{\partial z}\right).&\end{aligned} \tag{2.89}$$

The modified temperature and salinity equations are given by

$$\gamma \frac{\partial \theta}{\partial t} = -\frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (h_\psi u \theta)}{\partial \mu} + \frac{\partial (h_\mu v \theta)}{\partial \psi} \right\} - \frac{\partial (w\theta)}{\partial z} - \nabla \cdot \mathbf{F}^\theta + Q^\theta, \tag{2.90}$$

$$\gamma \frac{\partial S}{\partial t} = -\frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (h_\psi u S)}{\partial \mu} + \frac{\partial (h_\mu v S)}{\partial \psi} \right\} - \frac{\partial (wS)}{\partial z} - \nabla \cdot \mathbf{F}^S + Q^S. \tag{2.91}$$

Here, equations are written in depth (z) coordinate for brevity.

These modifications are equivalent to changing time to $t' = t/\alpha$ and the Brunt-Vaisala frequency to $N'^2 = N^2\alpha/\gamma$. In this case, the equivalent depth for the n-th mode of the vertical mode decomposition becomes $H_n^{'} = H_n/\alpha$.

The dispersion relation for the free inertia-gravity waves becomes:

$$\omega^2 = \frac{f^2}{\alpha^2} + \left(\frac{gH_n}{\alpha}\right)(k^2 + l^2). \tag{2.92}$$

Since the angular frequency ω is inversely proportional to $\alpha^{1/2}$, the phase speed becomes low for large α. The model can be run with a long time step.

The dispersion relation for Rossby waves becomes:

$$\omega = -\beta k \left[\alpha(k^2 + l^2) + \frac{f^2}{gH_n}\right]^{-1}. \tag{2.93}$$

Again, a large α yields a low phase speed.

In standard practice, a value from several tens to a few hundred is used as α, a value of one is used near the sea surface, and a value about a tenth is used near the bottom as γ.

It should be noted that when α is too large, the model field is prone to baroclinic instability. Since this should not occur in nature, an integration of the model should be performed carefully by checking outputs during the integration.

2.3.2 Physical constants

On Table 2.1, we list basic physical constants used for MRI.COM. These are defined in `param.F90`. Physical constants or formulae used to calculate surface fluxes and sea ice processes are explained in Chapters 14 and 17, respectively.

Table2.1 Physical constants used in the model

	value	variable name in MRI.COM
radius of the Earth	6375.0×10^5 cm	`RADIUS`
acceleration due to gravity	$981.0\,\mathrm{cm \cdot s^{-2}}$	`GRAV`
angular velocity of the Earth's rotation	$\pi/43082.0\,\mathrm{radian \cdot s^{-1}}$	`OMEGA`
the absolute temperature of 0 °C	273.15 K	`TAB`
the average density of sea water	$1.036\,\mathrm{g \cdot cm^{-3}}$	`RO`
the specific heat of sea water	$3.99 \times 10^7\,\mathrm{erg \cdot g^{-1} \cdot K^{-1}}$ $(1.0\,\mathrm{erg \cdot g^{-1} \cdot K^{-1}}$ $= 1.0 \times 10^{-4}\,\mathrm{J \cdot kg^{-1} \cdot K^{-1}})$	`CP`

Chapter 3

Spatial grid arrangement

MRI.COM uses a structured grid, and the model ocean domain is defined as a three-dimensional aggregate of rectangular grid cells on model's logical coordinate system. (However, for cells just above the bottom, so-called partial cells are used, which can locally change their vertical thickness.) The horizontal and vertical lengths of the cells are regarded as the horizontal and vertical grid intervals, respectively. In MRI.COM, the grid interval can be varied spatially in each direction (variable grid interval). Note that the introduction of z^* coordinate results in the temporally variable actual vertical grid intervals, though the logical grids are fixed in time.

In this chapter, we first describe the horizontal grid arrangement (3.1) and the vertical grid arrangement (3.2), which are main features of MRI.COM. Next, the notation of the grid index and the model region are shown (3.3). How to calculate the horizontal area of each grid (3.4) and the thickness and volume in the z^* coordinate system (3.5) are also explained. Finally, we summarize how to set the model grids for users (3.6).

3.1 Horizontal grid arrangement

MRI.COM adopts somewhat unique horizontal grid arrangement, which is illustrated in Figure 3.1a. Horizontal components of velocity and bottom depth are defined at the center of the grid cell (×), and tracers such as temperature and salinity, density, and sea-surface height (SSH) are defined at the four corners of the cell (∘). Hereafter, for simplicity, the velocity point is referred to as the U-point; the grid cell centered on the U-point, the U-cell (Figure 3.1c); the tracer point as the T-point; and the grid cell centered on the T-point, the T-cell (Figure 3.1c). The T-cells are staggered from the U-cells by a half grid size and consists of partial cells along the coast lines (Figure 3.2). The coast lines are defined by the periphery of the U-cells, i.e., the lines connecting the T-points. This type of horizontal grid arrangement is called Arakawa's B-grid arrangement (Arakawa, 1972). Although Arakawa's B-grid arrangement is also used in MOM (NOAA-GFDL, USA) and COCO (AORI of U. Tokyo and JAMSTEC, Japan), the primary cell is the T-cell in those models and the coast lines are defined by the lines connecting the U-points (Figure 3.1b).

In the case of the variable grid size, the T-points are defined just at the centers of the T-cells as seen in Figure 3.1c, but the U-points are not at the centers of the U-cells. The U-points are arranged so that the U-cell boundary stands at the mid-point between two neighboring U-points.

See Section 3.6.2 for the placement and numbering of the grid at the edge of the model domain.

3.2 Vertical grid arrangement

A variable grid size is usually used for the vertical grid arrangement, i.e., fine near the surface and coarse at depth. As illustrated in Figure 3.3a, tracers (∘) and velocity (×) are defined at just the mid-depth level of the cell, and the vertical mass fluxes W (△, □) are defined at the boundary of the cell. There are two kinds of W, one for the T-cell (W^T; △) and another for the U-cell (W^U; □). Their horizontal locations are the T-points and the U-points depicted in Figure 3.1a.

In order to express the gentle bottom slope as smoothly as possible, the thickness of the deepest U-cell at each horizontal location is variable (so called partial cell), with a limitation that it must exceed a fraction of around 10 percent of the nominal thickness of the layer to avoid violating the vertical CFL condition (Figure 3.3b). Otherwise, as presented in Figure 3.3c, the gentle bottom slope is expressed by wide, flat bottoms and cliffs here and there with height of vertical grid size Δz. In these regions, the vertical velocity is concentrated at the cliffs, resulting in relatively strong fictitious horizontal currents there.

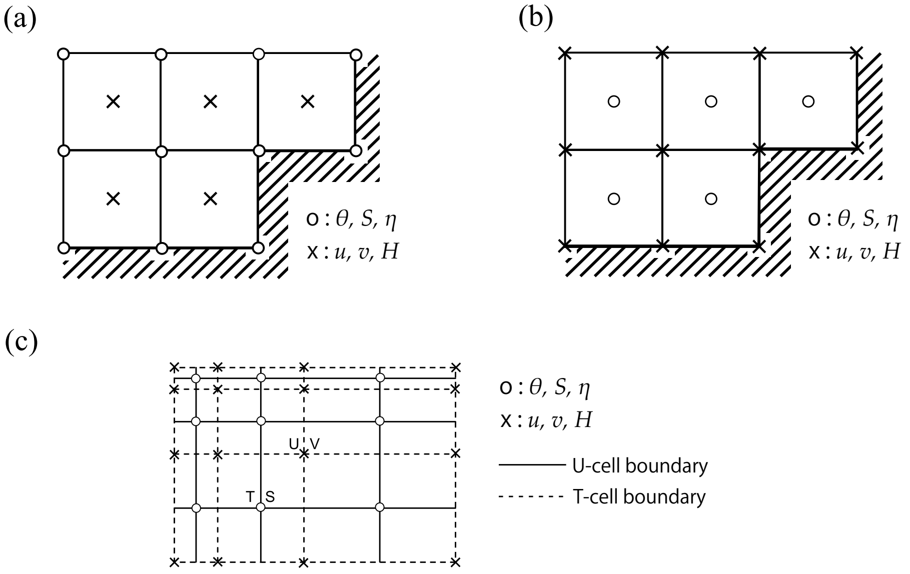

Figure3.1 Horizontal grid arrangement. (a) MRI.COM ($\circ$: θ, S, η, $\times$: u, v, H), (b) MOM and COCO ($\circ$: θ, S, η, H, $\times$: u, v), (c) Variable grid size in MRI.COM

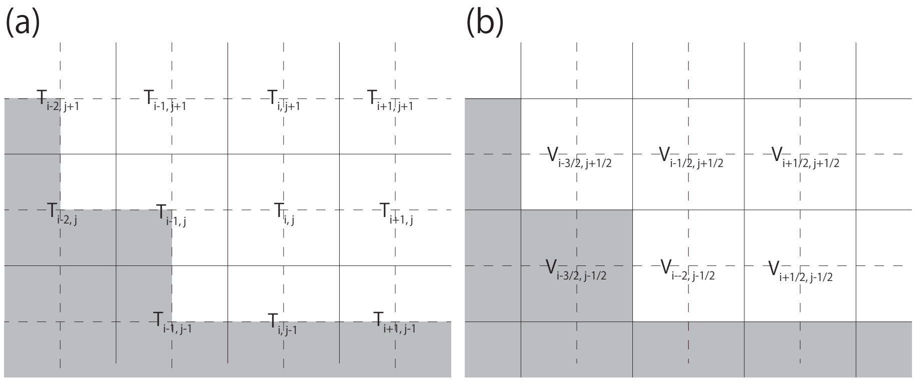

Figure3.2 Horizontal grid lattices in relation to topography. (a) Tracer lattice. (b) Velocity lattice. Land distribution (shade) is common for (a) and (b).

3.3 Grid indices and model domain

The conventions for indexing and the definitions of symbols used in finite difference expressions of the equations throughout this document are given here.

The actual distance corresponding to an increment of $\Delta\mu$ in the zonal direction of generalized orthogonal coordinates is expressed as follows:

$$\Delta x \equiv h_\mu \Delta\mu, \tag{3.1}$$

where h_μ is the scaling factor, the unit of which is [cm]. The unit of Δx is [cm], and $\Delta\mu$ is dimensionless. The actual meridional distance is defined similarly:

$$\Delta y \equiv h_\psi \Delta\psi. \tag{3.2}$$

The vertical distance is expressed by Δz (The unit is [cm]). For a discretized grid cell, the horizontal area is expressed by ΔA $(= \Delta x \Delta y)$ and the volume is expressed by ΔV $(= \Delta x \Delta y \Delta z)$.

The subscript indices expressing the horizontal grid position in the finite difference expression of the equations are usually integers for the T-points, i.e., (i, j) and are increased by a half for the U-points $(i + \frac{1}{2}, j + \frac{1}{2})$ (Figure 3.2). In some cases vice versa, with a notice.

In the vertical subscript index for the finite difference expression, the upper level of a grid cell, where the vertical mass flux is defined, is numbered as k ($k = 0$ being the sea surface), the levels of the T-point and U-point are numbered as $k - \frac{1}{2}$

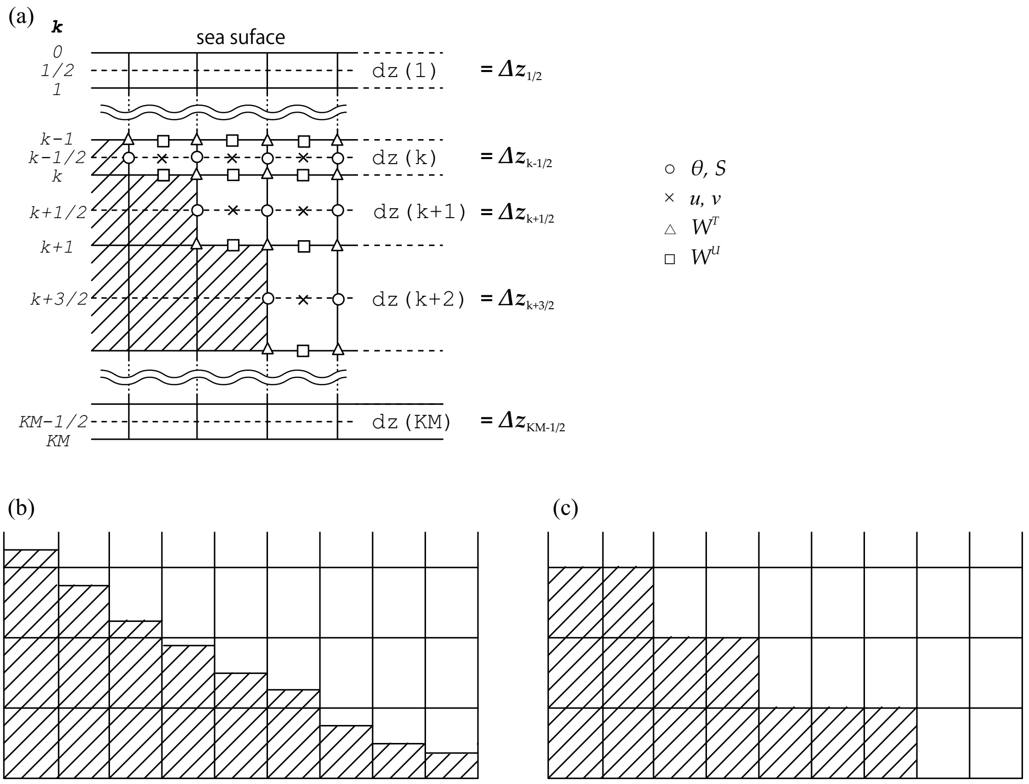

Figure3.3 Vertical grid arrangement. (a) Placement and numbering of vertical grid. (b) Smooth bottom slope with partial bottom cells. (c) Stair-like bottom slope.

(Figure 3.3a). In some cases, the T-point and U-point levels may be numbered as k, with a notice.

The model region is composed of U cells, and the numbers of horizontal grids (`imut`, `jmut`) are specified by the user (see Sec. 3.6). Note that this region includes the "halo region", which is the marginal region for the model calculation, in addition to the substantial "core region". The grid width of the halo region and the horizontal boundaries of the core region depend on the model settings as shown in Table 3.1. Figure 3.4a shows the model region of a closed basin surrounded by land, and Figure 3.4b shows that of a tri-polar global model. See Chapter 22 for more information on nested sub models.

Table3.1 Halo region and core region boundaries depending on model region settings.

model config	halo	west	east	south	north
closed basin (default)	1 grid-width land cell	$i = 2$	$i =$ `imut`	$j = 2$	$j =$ `jmut`
x-cyclic boundary condition (`CYCLIC`)	2 grids at east and west	$i = 3$	$i =$ `imut` $- 1$	$j = 2$	$j =$ `jmut`
tri-polor model (`TRIPOLAR` or `JOT` with `CYCLIC`)	2 grids at east and west, 1 at south, 3 at north	$i = 3$	$i =$ `imut` $- 1$	$j = 2$	$j =$ `jmut` $- 2$
nested sub model (`SUB` or `OFFNESTSUB`	2 grids at all boundaries	$i = 3$	$i =$ `imut` $- 1$	$j = 3$	$j =$ `jmut` $- 1$

3.4 Calculation of horizontal grid cell area and width

When equations are solved in MRI.COM, the temporal variations of physical quantities are calculated as a budget of their fluxes through the boundaries of the U-cells or T-cells (finite volume method). For this method, it is necessary to know the area and volume of the grid cells. These are numerically calculated for generalized orthogonal coordinate grids (Sec. 3.4.1), though they are analytically calculated if polar coordinate is chosen by user (Sec. 3.4.2). The outline of the lengths and areas calculated in this section is shown in the figure 3.5. Note that variable names in the program may be used even in the manual.

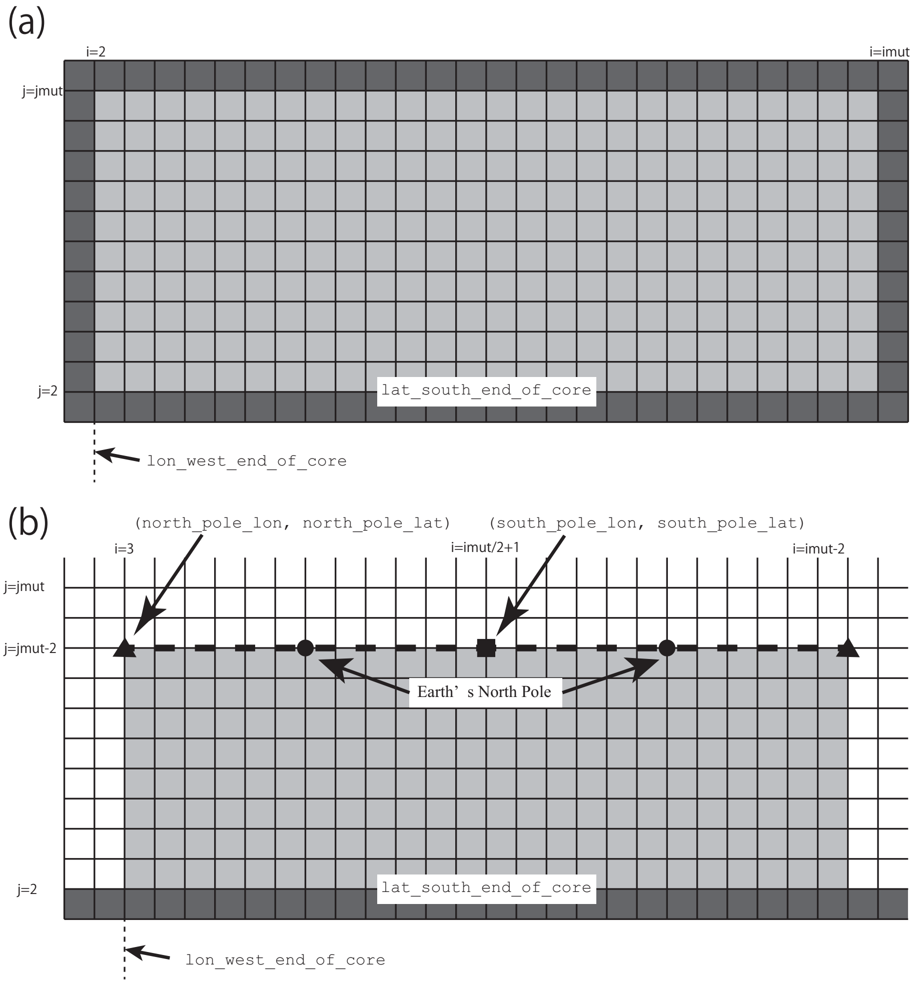

Figure3.4 The model domain and its halo region for (a) a model with closed domain, and (b) a tri-polar model with zonally cyclic (`CYCLIC`) and northern end folding (`TRIPOLAR` or `JOT`) conditions. The light shades are model's region of interest (core region). The dark shades are land that must be attached around the core region. The white cells in (b) are halo region. See Sec. 3.6 for the user specifications in the figure, such as `north_pole_lon`.

3.4.1 Generalized orthogonal coordinates

The longitude and latitude (λ, ϕ) of grid points on the sphere are defined by user as a function of the model coordinates (μ, ψ),

$$\lambda = \lambda(\mu, \psi), \quad \phi = \phi(\mu, \psi).$$

For example, the distance from a T-point $(\mu(i), \psi(j))$ to a point a half grid size to the east $(\mu(i+\frac{1}{2}), \psi(j))$ (variable name in the model: `dx_bl`; Figure 3.5a) is approximated numerically as follows taking $\mu_1 = \mu(i)$, $\mu_2 = \mu(i+\frac{1}{2})$, and $\psi_1 = \psi(j)$:

$$\sum_{m=1}^{M} L\Big[\lambda\Big(\mu_1 + (m-1)\delta\mu, \psi_1\Big), \phi\Big(\mu_1 + (m-1)\delta\mu, \psi_1\Big), \lambda\Big(\mu_1 + m\delta\mu, \psi_1\Big), \phi\Big(\mu_1 + m\delta\mu, \psi_1\Big)\Big]. \tag{3.3}$$

Here, $L[\lambda_1, \phi_1, \lambda_2, \phi_2]$ is the distance between the two points (λ_1, ϕ_1) and (λ_2, ϕ_2) on the sphere along a great circle and $\delta\mu = (\mu_2 - \mu_1)/M$ (divided by $M \sim 20$ between μ_1 and μ_2).

Similarly, a quarter grid area (`a_bl`; Figure 3.5b) surrounded by four points $(\mu(i), \psi(j))$, $(\mu(i+\frac{1}{2}), \psi(j))$, $(\mu(i+\frac{1}{2}), \psi(j+\frac{1}{2}))$, and $(\mu(i), \psi(j+\frac{1}{2}))$ is, taking $\psi_2 = \psi(j+\frac{1}{2})$ and $\delta\psi = (\psi_2 - \psi_1)/N$ (divided by $N \sim 20$ between ψ_1 and

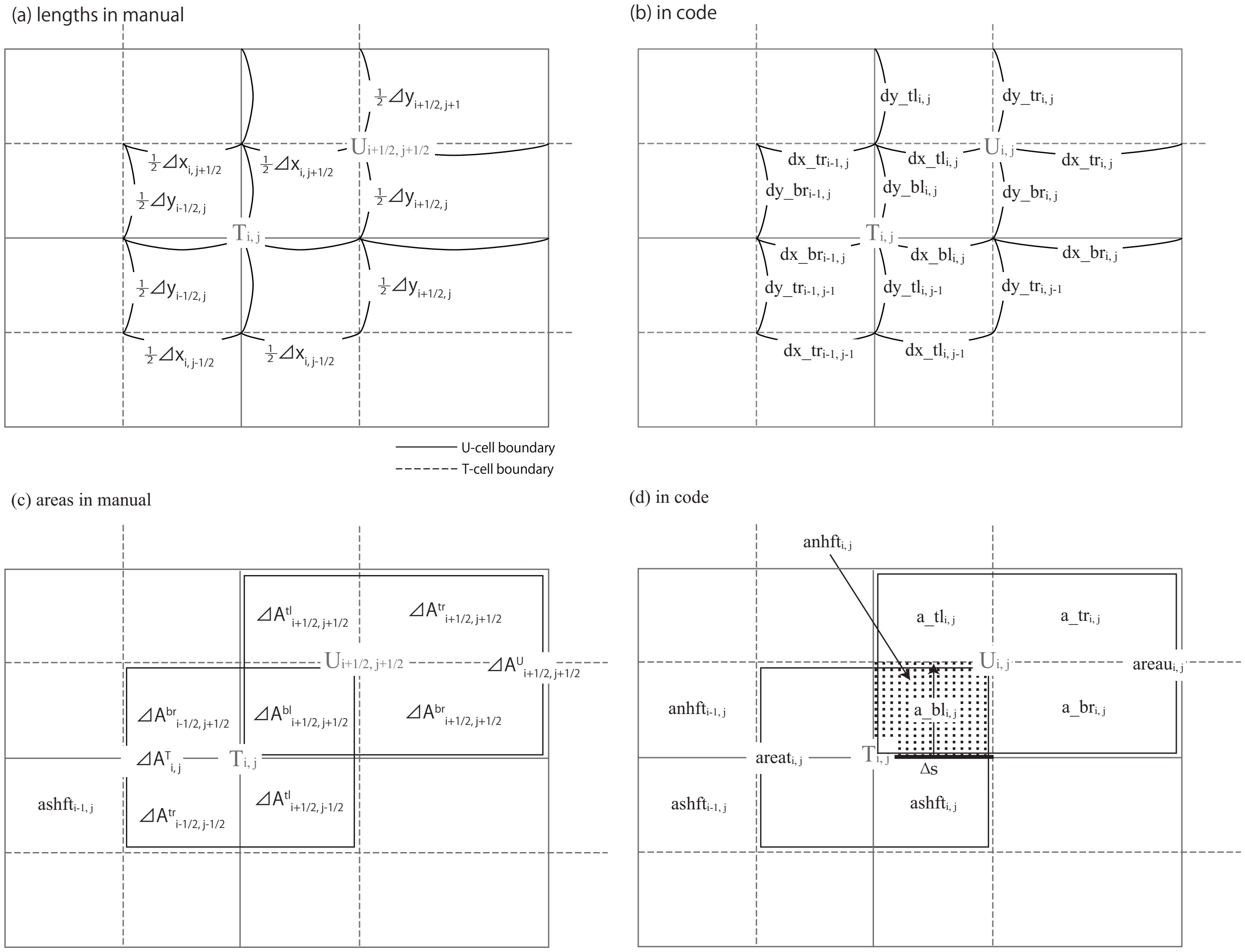

Figure3.5 Variables that define a grid unit for (a)(b) distance ($\Delta x, \Delta y$) and (c)(d) area (ΔA). The panels (a) and (c) show symbols and grid indices (i, j) following the convention described in Section 3.3. The panels (b) and (d) show variable names and indices in the code.

ψ_2), calculated as:

$$\begin{aligned}
\sum_{n=1}^{N}\sum_{m=1}^{M} L\Big[&\lambda\Big(\mu_1 + (m-1)\delta\mu, \psi_1 + \big(n-\frac{1}{2}\big)\delta\psi\Big), && \phi\Big(\mu_1 + (m-1)\delta\mu, \psi_1 + \big(n-\frac{1}{2}\big)\delta\psi\Big), \\
&\lambda\Big(\mu_1 + m\delta\mu, \psi_1 + \big(n-\frac{1}{2}\big)\delta\psi\Big), && \phi\Big(\mu_1 + m\delta\mu, \psi_1 + \big(n-\frac{1}{2}\big)\delta\psi\Big)\Big] \\
&\times \\
L\Big[&\lambda\Big(\mu_1 + \big(m-\frac{1}{2}\big)\delta\mu, \psi_1 + (n-1)\delta\psi\Big), && \phi\Big(\mu_1 + \big(m-\frac{1}{2}\big)\delta\mu, \psi_1 + (n-1)\delta\psi\Big), \\
&\lambda\Big(\mu_1 + \big(m-\frac{1}{2}\big)\delta\mu, \psi_1 + n\delta\psi\Big), && \phi\Big(\mu_1 + \big(m-\frac{1}{2}\big)\delta\mu, \psi_1 + n\delta\psi\Big)\Big].
\end{aligned} \tag{3.4}$$

As depicted in Figure 3.5, $(\texttt{a\_bl})_{i,j}$ is the area of the lower left quarter of the central U-point. Those for the lower right $(\texttt{a\_br})_{i,j}$, upper left $(\texttt{a\_tl})_{i,j}$, and upper right $(\texttt{a\_tr})_{i,j}$ quarters are obtained similarly.

The unit area centered on U-point ($(\texttt{areau})_{i,j}$) is then expressed as:

$$(\texttt{areau})_{i,j} = (\texttt{a\_bl})_{i,j} + (\texttt{a\_br})_{i,j} + (\texttt{a\_tl})_{i,j} + (\texttt{a\_tr})_{i,j}, \tag{3.5}$$

and the area centered on T-point ($(\texttt{areat})_{i,j}$) as

$$(\texttt{areat})_{i,j} = (\texttt{a\_bl})_{i,j} + (\texttt{a\_br})_{i-1,j} + (\texttt{a\_tl})_{i,j-1} + (\texttt{a\_tr})_{i-1,j-1}. \tag{3.6}$$

Following the conventions for indexing introduced in Section 3.3 (i and j are notated with a half-grid shift for U points), the above equations are expressed in later chapters as follows:

$$\Delta A^U_{i+\frac{1}{2},j+\frac{1}{2}} = \Delta A^{bl}_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta A^{br}_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta A^{tl}_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta A^{tr}_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{3.7}$$

$$\Delta A^T_{i,j} = \Delta A^{bl}_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta A^{br}_{i-\frac{1}{2},j+\frac{1}{2}} + \Delta A^{tl}_{i+\frac{1}{2},j-\frac{1}{2}} + \Delta A^{tr}_{i-\frac{1}{2},j-\frac{1}{2}}. \tag{3.8}$$

See Fig. 3.5.

3.4.2 Spherical coordinates

For grids in the spherical (geographic) coordinate system, we use more precise and computationally lighter analytical solutions. Let us examine the situation of T-cell divided in quarters (Figure 3.5). The area of the northeastern quarter (anhft, the same as that of the northwestern quarter) is obtained by the latitudinal integration of the thick line in Figure 3.5b, where $\Delta\phi = \phi(j+\frac{1}{2}) - \phi(j-\frac{1}{2})$.

Using the latitude of T-point $\phi(j)$, the zonal width of the grid unit for T-points $\Delta\lambda = \lambda(i+\frac{1}{2}) - \lambda(i-\frac{1}{2})$, and the Earth's radius a, the length of the thick line along the latitude circle (Δs) is expressed as:

$$\Delta s = a\frac{\Delta\lambda}{2}\cos\phi. \tag{3.9}$$

Integrating this in the latitudinal direction, we obtain the following.

$$\begin{aligned}
(\texttt{anhft})_{i,j} = &\int_{\phi}^{\phi+\frac{\Delta\phi}{2}} \Delta s a d\phi = \frac{a^2\Delta\lambda}{2}\int_{\phi}^{\phi+\frac{\Delta\phi}{2}} \cos\phi d\phi = \frac{a^2\Delta\lambda}{2}\left\{\sin\left(\phi+\frac{\Delta\phi}{2}\right) - \sin\phi\right\}\\
=&a^2\Delta\lambda\cos\left(\phi+\frac{\Delta\phi}{4}\right)\sin\frac{\Delta\phi}{4}\\
=&a^2\Delta\lambda\left(\cos\phi\cos\frac{\Delta\phi}{4} - \sin\phi\sin\frac{\Delta\phi}{4}\right)\sin\frac{\Delta\phi}{4}\\
=&a^2\Delta\lambda\cos\phi\cos\frac{\Delta\phi}{4}\sin\frac{\Delta\phi}{4}\left(1-\tan\phi\tan\frac{\Delta\phi}{4}\right)\\
=&\frac{a^2}{2}\Delta\lambda\cos\phi\sin\frac{\Delta\phi}{2}\left(1-\tan\phi\tan\frac{\Delta\phi}{4}\right).
\end{aligned} \tag{3.10}$$

Similarly, the area of the southeastern quarter of the T-cell (variable name in the model: ashft, the same as that of the southwestern quarter) is expressed as:

$$(\texttt{ashft})_{i,j} = \frac{a^2}{2}\Delta\lambda\cos\phi\sin\frac{\Delta\phi}{2}\left(1+\tan\phi\tan\frac{\Delta\phi}{4}\right). \tag{3.11}$$

At the north and south poles, where $\phi = \pm 90°$, we obtain the following by changing (3.10) and (3.11) to the following forms.

$$(\texttt{anhft})_{i,j} = \frac{a^2}{2}\Delta\lambda\sin\frac{\Delta\phi}{2}\left(\cos\phi - \sin\phi\tan\frac{\Delta\phi}{4}\right) \tag{3.12}$$

$$(\texttt{ashft})_{i,j} = \frac{a^2}{2}\Delta\lambda\sin\frac{\Delta\phi}{2}\left(\cos\phi + \sin\phi\tan\frac{\Delta\phi}{4}\right). \tag{3.13}$$

At the north pole:

$$(\texttt{anhft})_{i,j} = 0 \tag{3.14}$$

$$(\texttt{ashft})_{i,j} = \frac{a^2}{2}\Delta\lambda\sin\frac{\Delta\phi}{2}\tan\frac{\Delta\phi}{4}. \tag{3.15}$$

At the south pole:

$$(\mathtt{anhft})_{i,j} = \frac{a^2}{2} \Delta\lambda \sin\frac{\Delta\phi}{2} \tan\frac{\Delta\phi}{4} \tag{3.16}$$

$$(\mathtt{ashft})_{i,j} = 0. \tag{3.17}$$

In our model

$$(\mathtt{a\_bl})_{i,j} = (\mathtt{anhft})_{i,j}, \qquad (\mathtt{a\_br})_{i,j} = (\mathtt{anhft})_{i+1,j},$$
$$(\mathtt{a\_tl})_{i,j} = (\mathtt{ashft})_{i,j+1}, \quad (\mathtt{a\_tr})_{i,j} = (\mathtt{ashft})_{i+1,j+1},$$

and the areas of the grid cells centered on the U-points and T-points are calculated by (3.5) and (3.6).

3.5 Calculation of vertical cell thickness

In the z^* coordinate system, the depth coordinate z from the seafloor $-H$ to the sea level η is converted to the z^* coordinate from $-H$ to 0 according to (Eq. 2.40). As a result, in the z^* coordinate system, the physical thickness Δz [cm] and the physical volume ΔV of each grid change depending on $\eta(t)$. In the following, how to calculate them are shown in the order of quarter grid unit, U cell, and T cell. Finally, the depth anomaly induced by ascending and descending of $\eta(t)$ is also shown. In the volume calculation, the area ΔA calculated by the method described in the previous section is used.

a. Distribute sea level variation to the four sub-divided parts of a T-cell

When the sea level variation $\eta_{i,j}$ is known at a T-point, the thicknesses of four divided cells that comprise a T-cell (i.e. four cells, $\Delta A^{bl}_{i+\frac{1}{2},j+\frac{1}{2}}$, $\Delta A^{tl}_{i+\frac{1}{2},j-\frac{1}{2}}$, $\Delta A^{br}_{i-\frac{1}{2},j+\frac{1}{2}}$ and $\Delta A^{tr}_{i-\frac{1}{2},j-\frac{1}{2}}$, in Fig. 3.5) are determined as follows:

$$(\mathtt{dzu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = (\eta_{i,j} + H_{i+\frac{1}{2},j+\frac{1}{2}}) \frac{(\mathtt{dzu\_cnst})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}}{H_{i+\frac{1}{2},j+\frac{1}{2}}} \tag{3.18}$$

$$(\mathtt{dzu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} = (\eta_{i,j} + H_{i+\frac{1}{2},j-\frac{1}{2}}) \frac{(\mathtt{dzu\_cnst})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{H_{i+\frac{1}{2},j-\frac{1}{2}}} \tag{3.19}$$

$$(\mathtt{dzu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = (\eta_{i,j} + H_{i-\frac{1}{2},j+\frac{1}{2}}) \frac{(\mathtt{dzu\_cnst})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}}{H_{i-\frac{1}{2},j+\frac{1}{2}}} \tag{3.20}$$

$$(\mathtt{dzu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} = (\eta_{i,j} + H_{i-\frac{1}{2},j-\frac{1}{2}}) \frac{(\mathtt{dzu\_cnst})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{H_{i-\frac{1}{2},j-\frac{1}{2}}}, \tag{3.21}$$

where H is the depth of sea floor and `dzu_cnst` at U-points is the logical definition of vertical cell thickness in z^* frame, which does not vary in time. Volume of the four cells becomes

$$(\mathtt{volu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = \Delta A^{bl}_{i+\frac{1}{2},j+\frac{1}{2}} (\mathtt{dzu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \tag{3.22}$$

$$(\mathtt{volu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} = \Delta A^{tl}_{i+\frac{1}{2},j-\frac{1}{2}} (\mathtt{dzu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \tag{3.23}$$

$$(\mathtt{volu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = \Delta A^{br}_{i-\frac{1}{2},j+\frac{1}{2}} (\mathtt{dzu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \tag{3.24}$$

$$(\mathtt{volu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} = \Delta A^{tr}_{i-\frac{1}{2},j-\frac{1}{2}} (\mathtt{dzu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}. \tag{3.25}$$

Using these, the new volume of T-cells and U-cells is obtained. There is no leak in volume by using this method. Figure 3.6 illustrates the procedure.

b. U-cell

Thickness of a U-cell (`dzu`) is obtained by dividing U-cell's volume by U-cell's horizontal area. U-cell's volume is a sum of the four sub-divided cells whose volume varies following sea level variation on T-cells where they belong to. Using (3.22) through (3.25), we have,

$$\begin{aligned}(\mathtt{volu})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = {} & (\mathtt{volu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\mathtt{volu\_br})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \\ & + (\mathtt{volu\_tl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\mathtt{volu\_tr})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}.\end{aligned} \tag{3.26}$$

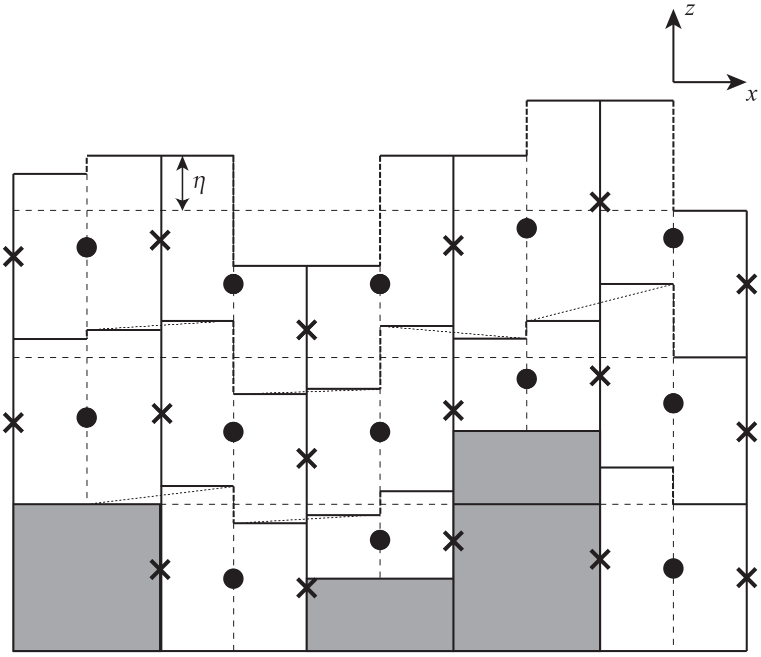

Figure3.6 Illustration of a vertical slice through a set of grid cells in the x-z plane for z^* coordinate. The center point in each cell (•) is a velocity point. The cross (×) is a tracer point.

Then the thickness is computed by

$$(\mathtt{dzu})_{i+\frac{1}{2}.j+\frac{1}{2},k-\frac{1}{2}} \equiv (\mathtt{volu})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \Big/ (\mathtt{areau})_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{3.27}$$

that is, the thickness of a U-cell is the average of the thickness of the four sub-divided T-cells.

c. T-cell

Volume of a T-cell is also calculated as a sum of the four sub-divided cells. Using (3.22) through (3.25), we have

$$\begin{aligned}(\mathtt{volt})_{i,j,k-\frac{1}{2}} = {}& (\mathtt{volu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\mathtt{volu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \\ &+ (\mathtt{volu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\mathtt{volu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}.\end{aligned} \tag{3.28}$$

Because horizontal cross section of a T-cell is not uniform vertically owing to the presence of partial cells, thickness of a T-cell cannot be determined identically. Here, the thickness `dzt_cnst` is determined as the difference between top and bottom face of the T-cell,

$$\begin{aligned}(\mathtt{dzt\_cnst})_{i,j,k-\frac{1}{2}} = \max\big(&(\mathtt{dzu\_cnst})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}, (\mathtt{dzu\_cnst})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}, \\ &(\mathtt{dzu\_cnst})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}, (\mathtt{dzu\_cnst})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}\big).\end{aligned} \tag{3.29}$$

d. Depth anomalies

We defined vertical cell thickness of four divided cells that comprise a T-cell by (3.18) through (3.21). In the same manner, we may define the anomalies of the actual depth at the center and bottom of the four divided cells, with $-\eta$ at the sea surface ($s = 0$). Note that depth is defined positive downward.

For the center,

$$(\mathtt{dpu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = -\frac{H_{i+\frac{1}{2},j+\frac{1}{2}} - (\mathtt{dpu\_cnst})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}}{H_{i+\frac{1}{2},j+\frac{1}{2}}}\eta_{i,j} \tag{3.30}$$

$$(\mathtt{dpu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} = -\frac{H_{i+\frac{1}{2},j-\frac{1}{2}} - (\mathtt{dpu\_cnst})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{H_{i+\frac{1}{2},j-\frac{1}{2}}}\eta_{i,j} \tag{3.31}$$

$$(\mathtt{dpu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = -\frac{H_{i-\frac{1}{2},j+\frac{1}{2}} - (\mathtt{dpu\_cnst})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}}{H_{i-\frac{1}{2},j+\frac{1}{2}}}\eta_{i,j} \tag{3.32}$$

$$(\mathtt{dpu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} = -\frac{H_{i-\frac{1}{2},j-\frac{1}{2}} - (\mathtt{dpu\_cnst})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{H_{i-\frac{1}{2},j-\frac{1}{2}}}\eta_{i,j} \tag{3.33}$$

and for the bottom,

$$(\mathtt{depu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k} = -\frac{H_{i+\frac{1}{2},j+\frac{1}{2}} - (\mathtt{depu\_cnst})_{i+\frac{1}{2},j+\frac{1}{2},k}}{H_{i+\frac{1}{2},j+\frac{1}{2}}}\eta_{i,j} \tag{3.34}$$

$$(\mathtt{depu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k} = -\frac{H_{i+\frac{1}{2},j-\frac{1}{2}} - (\mathtt{depu\_cnst})_{i+\frac{1}{2},j-\frac{1}{2},k}}{H_{i+\frac{1}{2},j-\frac{1}{2}}}\eta_{i,j} \tag{3.35}$$

$$(\mathtt{depu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k} = -\frac{H_{i-\frac{1}{2},j+\frac{1}{2}} - (\mathtt{depu\_cnst})_{i-\frac{1}{2},j+\frac{1}{2},k}}{H_{i-\frac{1}{2},j+\frac{1}{2}}}\eta_{i,j} \tag{3.36}$$

$$(\mathtt{depu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k} = -\frac{H_{i-\frac{1}{2},j-\frac{1}{2}} - (\mathtt{depu\_cnst})_{i-\frac{1}{2},j-\frac{1}{2},k}}{H_{i-\frac{1}{2},j-\frac{1}{2}}}\eta_{i,j}, \tag{3.37}$$

where `dpu_cnst` is the logical depth at the center of a U-cell (dashed line in Figure 3.3a) and `depu_cnst` is the logical depth at the bottom of a U-cell (solid line in Figure 3.3a).

The depth anomaly of a T-point is obtained as an area average. For example, at the center,

$$\begin{aligned}(\mathtt{dpt})_{i,j,k-\frac{1}{2}} = \Big\{&(\mathtt{a\_bl})_{i+\frac{1}{2},j+\frac{1}{2}}(\mathtt{dpu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\mathtt{a\_tl})_{i+\frac{1}{2},j-\frac{1}{2}}(\mathtt{dpu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \\ &+ (\mathtt{a\_br})_{i-\frac{1}{2},j+\frac{1}{2}}(\mathtt{dpu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\mathtt{a\_tr})_{i-\frac{1}{2}.j-\frac{1}{2}}(\mathtt{dpu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}\Big\}\Big/(\mathtt{areat})_{i,j,k+\frac{1}{2}}\end{aligned} \tag{3.38}$$

and for the bottom,

$$\begin{aligned}(\mathtt{dept})_{i,j,k} = \Big\{&(\mathtt{a\_bl})_{i+\frac{1}{2},j+\frac{1}{2}}(\mathtt{depu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k} + (\mathtt{a\_tl})_{i+\frac{1}{2},j-\frac{1}{2}}(\mathtt{depu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k} \\ &+ (\mathtt{a\_br})_{i-\frac{1}{2},j+\frac{1}{2}}(\mathtt{depu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k} + (\mathtt{a\_tr})_{i-\frac{1}{2}.j-\frac{1}{2}}(\mathtt{depu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k}\Big\}\Big/(\mathtt{areat})_{i,j,k+\frac{1}{2}}.\end{aligned} \tag{3.39}$$

In the above calculation, we use only full cells, that is, we do not include partial cells except for the bottom cell ($k = ktbtm$). This is reflected in the use of $(\mathtt{areat})_{i,j,k+\frac{1}{2}}$ instead of $(\mathtt{areat})_{i,j,k-\frac{1}{2}}$.

Using this, the variable thickness of a T-cell is calculated as follows:

$$(\mathtt{dzt})_{i,j,k-\frac{1}{2}} = (\mathtt{dept})_{i,j,k} - (\mathtt{dept})_{i,j,k-1} + (\mathtt{dzt\_cnst})_{i,j,k-\frac{1}{2}}. \tag{3.40}$$

3.6 Usage

How to set the model region and grid are explained here. Specifically, the horizontal coordinate system is selected in Sec. (3.6.1), and the region and grid are set in Sec. (3.6.2). Also, if a polar coordinate is not used, the user must provide the grid width and area directly in Sec. (3.6.3).

3.6.1 Horizontal coordinate system

For horizontal coordinate system, either spherical or generalized orthogonal curvilinear coordinates must be chosen.

a. Spherical coordinates

For a model that does not include the North Pole, spherical coordinates with geographical longitude-latitude axes will be a standard choice.

If spherical coordinates are chosen, model option `SPHERICAL` must be added to the list of options (the line start with `OPTIONS =`) specified in `configure.in`.

The geographical position of north and south pole should not be necessarily at the Earth's North and South Pole. If the model's north pole is displaced, this is specified by namelist `nml_poles` (Table 3.2).

Table3.2 Namelist `nml_poles` for `SPHERICAL` option. The geographical location of the model's south pole is determined automatically by the specification of the north pole.

variable name	units	description	usage
`north_pole_lon`	degree	geographical longitude of north pole	default is 0°
`north_pole_lat`	degree	geographical latitude of north pole	default is 90°N

b. Generalized orthogonal curvilinear coordinates

Singularity at the North Pole may be avoided by coordinate transformation within the framework of generalized orthogonal curvilinear coordinates.

Two options are available:

- Tripolar grid (`TRIPOLAR`) combining geographical lat-lon south of around 60°N and transformed coordinates to the north.
- Joukowski conversion (`JOT`) applied to the whole sphere of the Earth.

When either `TRIPOLAR` or `JOT` option is chosen, that option must be added to the list of options (the line start with `OPTIONS =`) specified in `configure.in`. In addition, namelist `nml_poles` must be specified, but in this case the geographical locations of the two new singular points instead of the Earth's North and South Pole should be given (Table 3.3). See Chapter 24 for details.

Table3.3 Namelist `nml_poles` for `TRIPOLAR` or `JOT` option. See also Figure 3.4b for the locations of the two singilar points to be specified.

variable name	units	description	usage
`north_pole_lon`	degree	geographical longitude of one of the two new singular points	specification is required
`north_pole_lat`	degree	geographical latitude of one of the two new singular points	specification is required
`south_pole_lon`	degree	geographical longitude of the other of the two new singular points	specification is required
`south_pole_lat`	degree	geographical latitude of the other of the two new singular points	specification is required

3.6.2 Model domain and grid

There are several options about the model region as shown in Table 3.1: closed basin (default), x-cyclic boundary condition(`CYCLIC`), tri-pole models (`TRIPOLAR`) and nested sub models (`SUB` or `OFFNESTSUB`). User specifies these options in `configure.in`.

The total numbers of grid points in the three directions (`imut`, `jmut`, and `km`) must be given to `configure.in`, too. The grid sizes (`imut` and `jmut`) must include halo cells. Minimum information that must be given to `configure.in` for compiling a global tri-polar grid model will look like as follows.

An example specification given to `configure.in` for a global tri-polar grid model

```
OPTIONS="TRIPOLAR CYCLIC"
IMUT=364
JMUT=368
KM=60
```

Further horizontal grid information is given to the model by namelist at run time. It is necessary to specify the western and southern end of model's core region and the X and Y axis grid spacing. They are given by namelist `nml_horz_grid` (Table 3.4). When the horizontal grid spacing is given by a file, the file is read by the model at run time as follows.

Format of horizontal grid spacing data (`file_dxdy_tbox_deg`).

```
real(8) :: dxtdeg(imut), dytdeg(jmut)
integer(4),parameter :: lun = 10
open(lun, file=file_dxdy_tbox_deg, form=unformatted )
read( lun ) i, j
if ( ( i == imut ).and.( j == jmut ) ) then
  read( lun ) dxtdeg
  read( lun ) dytdeg
end if
close(lun)
```

Here, `dxtdeg` and `dytdeg` are longitudinal and latitudinal width of T-cells, respectively (Figure 3.4c). Note that they can vary only in the direction of their own axis.

Table3.4 Namelist `nml_horz_grid`

variable name	units	description	usage
`lon_west_end_of_core`	model longitude in degree	the longitude of the western end of model's main region	specification is required
`lat_south_end_of_core`	model latitude in degree	the latitude of the southern end of model's main region	specification is required
`dx_const_deg`	degree	uniform X-axis grid spacing	default is zero
`dy_const_deg`	degree	uniform Y-axis grid spacing	default is zero
`file_dxdy_tbox_deg`	character	X and Y axis grid spacing are given by this file	If either X or Y axis grid spacing is not uniform, prepare this file.

The vertical water column is filled by U-cells with specified thickness. The start point of the vertical grid is always set to be zero (sea surface). The vertical grid index increases downward and the vertical grid width ($\Delta z_{k-\frac{1}{2}}$ = `dz(k)`) is given either by a file or a namelist. Which one to select is determined by namelist `nml_vert_grid` (Table 3.5) at run time. When the vertical grid spacing is given by a file, the file is read by the model at run time as follows.

Format of vertical grid spacing data (`file_dz_cm`).

```
real(8) :: dz(km)
integer(4),parameter :: lun = 10
open(lun, file=file_dz_cm, form=unformatted)
read(lun) k
if ( k == km ) then
  read(lun) dz
endif
close(lun)
```

3.6.3 Grid cell area and line elements

When spherical coordinates are chosen (`SPHERICAL`), grid cell area and line elements are calculated analytically by the model (Section 3.4.2). When generalized orthogonal curvilinear coordinates are chosen (`TRIPOLAR` or `JOT`), they should be read from file (Section 3.4.1). The file name must be given by namelist `nml_grid_scale` (Table 3.6).

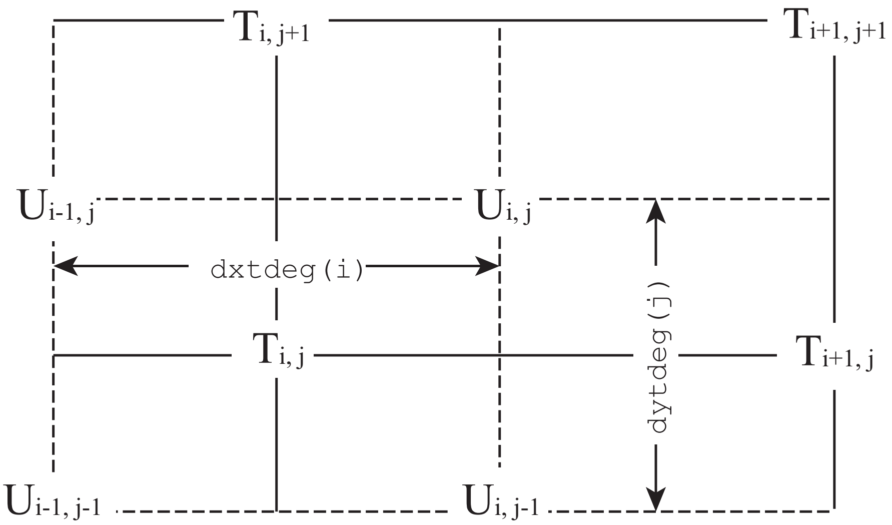

Figure3.7 Specification of the basic arrays (`dxtdeg` and `dytdeg`) that define grid spacing. Grid indices (i, j) follow array indices in program codes. The indices of the T-points in the vicinity of boundaries are shown in Fig. 3.4.

Table3.5 Namelist `nml_vert_grid`

variable name	units	description	usage
`file_dz_cm`	character	vertical grid spacing is given by this file	If the vertical grid spacing is not uniform, prepare this file. `dz_const_cm` must not be given.
`dz_const_cm`	cm	uniform vertical grid spacing	default is zero

Table3.6 Namelist `nml_grid_scale`

variable name	units	description	usage
`file_scale`	character	grid cell area and line elements are given by this file	specification is required for `TRIPOLAR` or `JOT`.

The file that contains the grid cell and line elements (see also Figure 3.5 for positions) is read by the model at run time as follows.

Format of grid cell area and line elements (file_scale; if not SPHERICAL)

```
real(8) :: a_bl(imut,jmut),  a_br(imut,jmut),  a_tl(imut,jmut),  a_tr(imut,jmut)
real(8) :: dx_bl(imut,jmut), dx_br(imut,jmut), dx_tl(imut,jmut), dx_tr(imut,jmut)
real(8) :: dy_bl(imut,jmut), dy_br(imut,jmut), dy_tl(imut,jmut), dy_tr(imut,jmut)
integer(4),parameter :: lun = 10
open(unit=lun, file=file_scale, form=unformatted)
read(unit=lun) a_bl  ! U-box area of bottom-left 1/4 grid
read(unit=lun) a_br  ! U-box area of bottom-right 1/4 grid
read(unit=lun) a_tl  ! U-box area of top-left 1/4 grid
read(unit=lun) a_tr  ! U-box area of top-right 1/4 grid
read(unit=lun) dx_bl ! U-box length of bottom-left 1/4 grid
read(unit=lun) dx_br ! U-box length of bottom-right 1/4 grid
read(unit=lun) dx_tl ! U-box length of top-left 1/4 grid
read(unit=lun) dx_tr ! U-box length of top-right 1/4 grid
read(unit=lun) dy_bl ! U-box length of bottom-left 1/4 grid
read(unit=lun) dy_br ! U-box length of bottom-right 1/4 grid
read(unit=lun) dy_tl ! U-box length of top-left 1/4 grid
read(unit=lun) dy_tr ! U-box length of top-right 1/4 grid
close(lun)
```

Chapter 4

Time integration

In this chapter, we discuss the time integration method for the primitive equations defined in Section 2.2. In Section 4.2, we discuss the leap-frog scheme, which is traditionally used in ocean models and has been used in MRI.COM until version 4.7. Section 4.3 describes the Leap frog Adams-Moulton (LFAM3) scheme, which was first used in ocean models by the Regional Ocean Modeling System (ROMS) and has become the only time integration scheme in MRI.COM since version 5.0.

4.1 Time integration flow

In MRI.COM, each component is calculated as follows.

1. Boundary Process (Part V) + Sea ice (Chapter 17)
2. **Evolution of tracers (Part IV)**
3. **Equation of state (Chapter 5)**
4. **Equation of motion [baroclinic component] (Chapter 8)**
5. **Equation of motion [barotropic component] (Chapter 7)**
6. Turbulence closure models (Chapter 15)
7. **Preparing for the next step**

However, the sea ice and mixed layer calculations are always done in forward differencing, independent of the time integration scheme of the ocean model. This is especially important for sea ice to satisfy conservation laws. For details, see section 17 for sea ice and section 15 for the mixed layer. Only for the remaining items, marked with **bold line**, the ocean model time integration scheme is applied.

4.2 Leap-frog method (LF)

For the time integration method of MRI.COM, the leap-frog method (LF) had been used until version 4.7.

Consider the following typical advection-diffusion equation (the first term on the right-hand side is advection and the second term on the right-hand side is diffusion)

$$\frac{\partial a}{\partial t} = \mathcal{A}(a) + \mathcal{D}(a) \tag{4.1}$$

Three time steps are used in solving this equation.

$$a^{n+1} = a^{n-1} + 2\Delta t \times \left\{\mathcal{A}(a^n) + \mathcal{D}(a^{n-1})\right\} \tag{4.2}$$

This method has long been the standard for ocean models because it is simple, energy-neutral, and one of the most computationally efficient method to describe wave equations. However, there are some problems. The accuracy is only $O(\Delta t^2)$. Since the time series of a^{2n+1} and a^{2n} are separated (red and blue lines in Fig. 4.1), it is difficult to satisfy the conservation between the two time series. Before this problem becomes serious, a Matsuno scheme consisting of two steps, forward scheme and backward scheme, is inserted once in the calculation of a few dozen steps in LF, so that one of the separated modes is discarded. The Matsuno scheme is also used to create the two time steps needed for LF at the start of integration. In addition, the accurate advection of tracers is often written in forward difference method, in which case the stability is halved. The issue of conservation is a particularly serious problem when fluxes are exchanged with other systems that are not running in LF, such as sea ice. In order to solve these problems, starting with 5.0, we will use the third order leap-frog Adams-Moulton (LFAM3), which is used in Regional Ocean Models (ROMS).

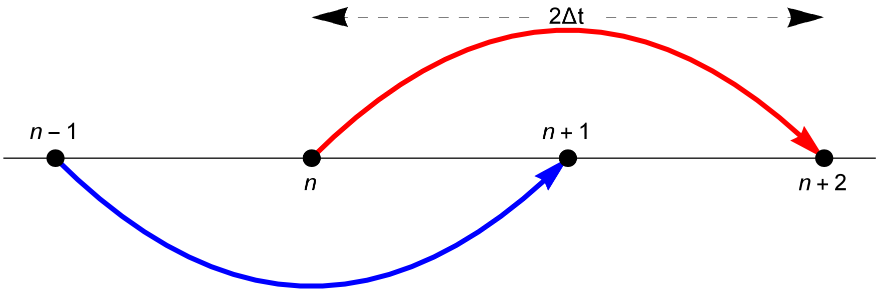

Figure4.1 Schematic description of the LF time integration.

4.3 Leap-frog Adams-Moulton (LFAM3) method

Consider the following one-dimensional wave equation,

$$\frac{\partial p}{\partial t} = -c\frac{\partial u}{\partial x}, \tag{4.3}$$

$$\frac{\partial u}{\partial t} = -c\frac{\partial p}{\partial x}. \tag{4.4}$$

where p is pressure, u is velocity, and c is the phase velocity of the wave. Applying the Fourier transformation with respect to space by placing $p(t,x) = \hat{p}_k e^{ikx}, u(t,x) = \hat{u}_k u^{ikx}$ and $\omega_k = kc$ yields,

$$\frac{\partial \hat{p}_k}{\partial t} = -i\omega_k \hat{u}_k, \tag{4.5}$$

$$\frac{\partial \hat{u}_k}{\partial t} = -i\omega_k \hat{p}_k. \tag{4.6}$$

For simplicity, we omit ^ and the subscript k, and denote the variable ζ^n at the time $t = n\Delta t$ of the variable ζ. To further simplify the equation, we introduce the dimensionless quantity $\alpha = \omega_k \Delta t$.

The LFAM3 for the Fourier transform of the 1D wave equation with respect to space is written as follows. These equations correspond to equations (2.38)-(2.41) in Shchepetkin and McWilliams (2005) (hereafter, SM2005) (or equations (4.1)-(4.2) in Shchepetkin and McWilliams (2009) (hereafter, SM2009)). The derivation of these equations is included in Chapter 23 of this manual (23.49–23.52).

LFAM3 is composed of the predictor substep:

$$p^{n+\frac{1}{2}} = \left(\frac{1}{2} - 2\gamma\right) p^{n-1} + \left(\frac{1}{2} + 2\gamma\right) p^n - i\alpha(1-2\gamma)u^n \tag{4.7}$$

$$u^{n+\frac{1}{2}} = \left(\frac{1}{2} - 2\gamma\right) u^{n-1} + \left(\frac{1}{2} + 2\gamma\right) u^n - i\alpha(1-2\gamma)\left[p^n + \beta\frac{2p^{n+1/2} - 3p^n + p^{n-1}}{1-2\gamma}\right] \tag{4.8}$$

and the corrector substep:

$$p^{n+1} = p^n - i\alpha u^{n+\frac{1}{2}} \tag{4.9}$$

$$u^{n+1} = u^n - i\alpha\left\{(1-\epsilon)p^{n+\frac{1}{2}} + \epsilon\left[\left(\frac{1}{2} - \gamma\right) p^{n+1} + \left(\frac{1}{2} + 2\gamma\right) p^n - \gamma p^{n-1}\right]\right\}, \tag{4.10}$$

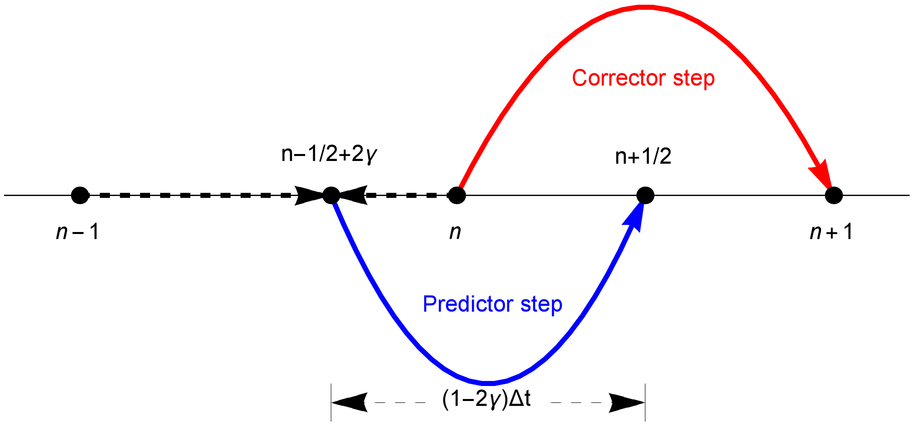

Figure4.2 Schematic description of the LFAM3 time integration. During the predictor substep, the initial data at $n-1$ and n steps are linearly interpolated to $n-\frac{1}{2}+2\gamma$ step and advanced to $n+\frac{1}{2}$ step by $(1-2\gamma)\Delta t$. During the corrector substep, the resultant values are used to compute the RHS at $n+\frac{1}{2}$ step to advance prognostic variables from n to $n+1$ steps. At the end of the corrector substep, the values at $n+\frac{1}{2}$ step are discarded.

where $\alpha = kc\Delta t$. The u corresponds to the flow velocity and p corresponds to the active tracer (temperature, salinity) related to the pressure. The predictor substep for $p^{n+\frac{1}{2}}$ can be interpreted as the time integration from $n-\frac{1}{2}+2\gamma$ step to $n = n-\frac{1}{2}$ step by $(1-2\gamma)\Delta t$ (Figure4.2).

During the corrector substep of finding p^{n+1}, the time integration for the advection is conducted from n to $n+1$step using velocity at $n+\frac{1}{2}$ step, which is compatible with high accurate tracer advection schemes such as Second Order Moments (SOM). During the corrector step of finding u^{n+1}, the values for the pressure gradient term is a linear combination of $n-1$, n, $n+\frac{1}{2}$, and $n+1$ steps. The accuracy of this wave equation can be increase to $O(\alpha^4)$. The accuracy drops when other terms such as viscosity are included, but it still maintain $O(\alpha^3)$. See Chapter19 for an evaluation of the accuracy of various time integration methods, including LF and LFAM3.

LFAM3 has the following advantages and disadvantages over the previously used LF.

- Advantages
  - The accuracy of the time scheme is increased from $O(\Delta t^2)$ to $O(\Delta t^3)$. In particular, it has $O(\Delta t^4)$ accuracy with respect to waves.
  - Computation cost is reduced by 2~30% (see below).
  - Tracer conservation is improved by removing computational mode in LF.
  - Time integration can be restarted from any time step.
- Disadvantages
  - A complete restart requires information at $n-2$, $n-1$, and n steps, making the restart process somewhat complicated and increasing file size. If restart files at $n-2$ and $n-1$ steps do not exist, the calculation starts by treating the restart file at n step as those at $n-2$ and $n-1$ steps.
  - It is always necessary to be aware of whether the current process is being performed during the predictor or corrector substep.

4.3.1 Time integration flow in case of LFAM3

LFAM3 has been the only time integration scheme used in MRI.COM since version 5.0. In this LFAM3, time integration is performed as follows. The items written in bold lines are performed by LFAM3.

Predictor substep

1. Boundary process + Sea ice : Time integration is done by forward differencing to compute $n+\frac{1}{2}$-step external forces.
2. **Evolution of tracers**: To find the pressure gradient at $n+\frac{1}{2}$ step, the tracer values are calculated using an advection scheme with low computational cost (4.7).
3. Equation of state (Section 5.1): Density is calculated from the tracer obtained above. Convective instability is also

checked.

4. **Equation of motion [baroclinic component]**: To find a velocity field at $n+\frac{1}{2}$ step, baroclinic velocity is calculated (4.8).
5. Equation of motion [barotropic component]: Barotropic velocity and sea-surface height at $n+\frac{1}{2}$ and n steps are calculated.
6. Preparation of the corrector substep.

Corrector substep

1. **Evolution of tracer**: Tracer values at $n+1$ step is calculated with an advection scheme with high accuracy (4.9).
2. Equation of state: Density is calculated from the tracer obtained above. Convective instability is also checked.
3. **Equation of motion [baroclinic component]**: Baroclinic velocity at $n+1$ step is calculated (4.10).
4. Turbulence closure models: Time integration is done by forward differencing to compute vertical diffusivity and viscosity at $n+1$ step.
5. Preparing for the next step. The values at $n+\frac{1}{2}$ step are discarded.

SM2005 shows that the maximum allowable time interval of LFAM3 is about 1.8 times as large as that of LF for the wave equation (subsection 23.6.2). If the computational cost of one substep for the predictor or the corrector is the same as the one step in LF, then LFAM3 allows time integration of only $1.8\Delta t$ with the same computational resources as time integration of $2\Delta t$ in LF. However, as mentioned above, the computational cost of LFAM3 is reduced by using less computationally demanding tracer advection scheme and omitting the turbulent closure calculation in the predictor substep, and by omitting the boundary and sea ice process, and barotropic velocity calculations in the corrector substep. Several test cases shows that the adoption of LFAM3 reduces the calculational cost by about 30% due to these effects, and the accuracy of the time integration becomes better.

The details of the calculation process of predictor and corrector substep in MRI.COM are described below.

4.3.2 Evolution of tracers during the predictor step

In the predictor step, the tracer at $n+\frac{1}{2}$ step is calculated only to find the density gradient at $n+\frac{1}{2}$ step. The tracer values at $n+\frac{1}{2}$ step are discarded after the density is obtained.

The predictor step (at $n+\frac{1}{2}$ step) for tracers is written in the form (Fig. 4.3)。

$$\theta^{n+\frac{1}{2}} = \left(\frac{1}{2}-2\gamma\right)\theta^{n-1} + \left(\frac{1}{2}+2\gamma\right)\theta^{n} + (1-2\gamma)\Delta t \cdot [\text{Right hand side of the tracer equation}(\text{RHS}_\theta)^n], \tag{4.11}$$

where, the RHS is the sum of θ-tendencies such as advection, diffusion, and surface fluxes. Corresponding to the above equation, the model variable in the b-series are created from linear combinations as described below, rather than using the past values as they are.

- trcv1b = $\left(\frac{1}{2}-2\gamma\right)\theta^{n-1}(\Delta V^T)^{n-1} + \left(\frac{1}{2}+2\gamma\right)\theta^{n}(\Delta V^T)^{n} = \left[\theta\Delta V^T\right]^b$
- trcbl = $\left(\frac{1}{2}-2\gamma\right)\theta^{n-1} + \left(\frac{1}{2}+2\gamma\right)\theta^{n} = \theta^b$
- trcl = θ^n
- trcal = $\theta^{n+\frac{1}{2}}$
- trcpl = θ^{n-1} (“p” means “Past”.)

Update of trcbl,trcl, and trcpl is conducted at the end of corrector substep or in rewrit_prepare. The trcpl should be stored because a new value of θ^{n-1} is needed in the expression (4.20) described below. It is also needed for creating the restart file. (As for u, the value at $n-1$ steps is not necessary for calculation, but necessary for restart.)

■Modification for the local conservation of tracers Since the barotropic velocity component (u_m) and sea level height (η) do not satisfy the continuity equation (eq. 6.19), the tracer field cannot satisfy its local conservation, that is, the spatially uniformity of tracer field at the beginning of the predictor substep does not hold at the end of the predictor substep. This situation is rewritten as,

$$\partial_t(\Delta V^T)_{i,j,k-\frac{1}{2}} \neq \frac{\Delta s_{i,j,k-\frac{1}{2}}}{H}\left((\texttt{ws\_total})_{i,j} + (\texttt{transport\_wflux})_{i,j}\right), \tag{4.12}$$

where ws_total is the total amount of water flux entering the grid due to the convergence of both barotropic flow and diffusion of sea level height, and transport_wflux is the freshwater flux at the surface. Since the value of $(\theta)^{n+1}$ is used only for the pressure gradient term and is discarded at the end of the corrector step, we need not be concerned about its total

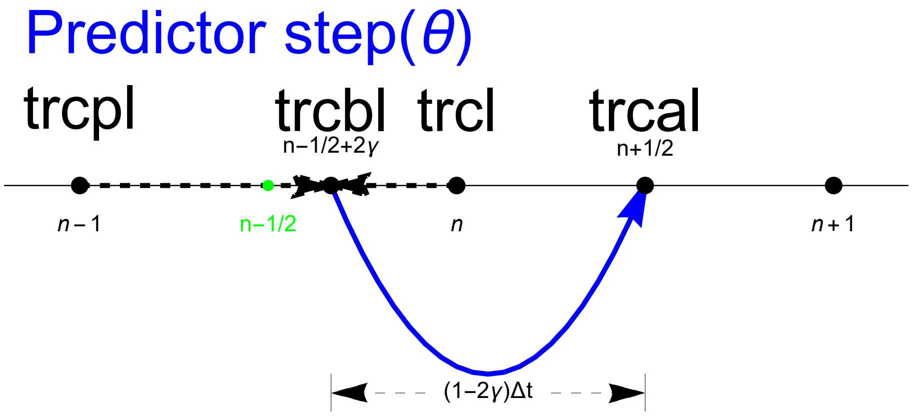

Figure4.3 Schematic of LFAM3 Time integration for MRI.COM. The beginning of the predictor step is a linear combination at $n-1$ and n steps.

conservation of it. On the other hand, the pressure gradient derived from it may have a bad influence on the velocity field when local conservation is not maintained. Therefore, we give up global conservation and consider adding a correction term that resolves the imbalance between the left and right sides of Eq. 4.12.

First, the time evolution of the tracer $\theta_{i,j,k-\frac{1}{2}}$ due to tracer advection is obtained in finite volume as follows.

$$\begin{aligned}\left[\theta\Delta V^T\right]^{n+1/2}_{i,j,k-\frac{1}{2}} &= \left[\theta\Delta V^T\right]^{b}_{i,j,k-\frac{1}{2}} \\ &+(1-2\gamma)\Delta t\left[U^T_{i-\frac{1}{2},j,k-\frac{1}{2}}\theta_{i-\frac{1}{2},j,k-\frac{1}{2}}-U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\theta_{i+\frac{1}{2},j,k-\frac{1}{2}}\right. \\ &V^T_{i,j-\frac{1}{2},k-\frac{1}{2}}\theta_{i,j-\frac{1}{2},k-\frac{1}{2}}-V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}\theta_{i,j+\frac{1}{2},k-\frac{1}{2}} \\ &\left.W^T_{i,j,k+1}\theta_{i,j,k}-W^T_{i,j,k-1}\theta_{i,j,k-1}\right]^n+\theta^b_{i,j,k-\frac{1}{2}}\times[\text{volume correction term}]\end{aligned} \tag{4.13}$$

$$\theta^{n+1/2}_{i,j,k-\frac{1}{2}} = \left[\theta\Delta V^T\right]^{n+1/2}_{i,j,k-\frac{1}{2}}/\left[\Delta V^T\right]^{n+1/2}_{i,j,k-\frac{1}{2}} \tag{4.14}$$

In this evolution of tracer equation, the following relationship must be satisfied when spatially uniform tracer fields remain to be spatially uniform after the advection.

$$\begin{aligned}\left[\Delta V^T\right]^{n+1/2}_{i,j,k-\frac{1}{2}} &= \left[\Delta V^T\right]^{b}_{i,j,k-\frac{1}{2}} \\ &+(1-2\gamma)\Delta t\left[U^T_{i-\frac{1}{2},j,k-\frac{1}{2}}-U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right. \\ &\left.V^T_{i,j-\frac{1}{2},k-\frac{1}{2}}-V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}+W^T_{i,j,k}-W^T_{i,j,k-1}\right]^n+\text{Volume correction term}\end{aligned} \tag{4.15}$$

$$\begin{aligned}&\left[U^T_{i-\frac{1}{2},j,k-\frac{1}{2}}-U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}+V^T_{i,j-\frac{1}{2},k-\frac{1}{2}}-V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}+W^T_{i,j,k}-W^T_{i,j,k-1}\right]^n \\ &\qquad=\left\{(\mathtt{ws\_total})_{i,j}+(\mathtt{transport\_wflux})_{i,j}\right\}\frac{\Delta s_{i,j,k-\frac{1}{2}}}{H}\end{aligned} \tag{4.16}$$

Using the above relationship derived from the diagnostic equation of the continuity equation, Eqs. (6.19) and (6.21), the volume change correction term is eventually obtained as follows. (In the corrector step, ΔV^T is obtained so that the correction term is zero.)

$$\begin{aligned}\text{Volume correction term} &= \left[\Delta V^T\right]^{n+1/2}_{i,j,k-\frac{1}{2}}-\left[\Delta V^T\right]^{b}_{i,j,k-\frac{1}{2}} \\ &-(1-2\gamma)\Delta t\left\{(\mathtt{ws\_total})_{i,j}+(\mathtt{transport\_wflux})_{i,j}\right\}\frac{\Delta s_{i,j,k-\frac{1}{2}}}{H}\end{aligned} \tag{4.17}$$

In the notation in the program, the correction term for satisfying local conservation in the predictor step, the last term in the r.h.s. of Eq. (4.13), is calculated as the quantity θ^b multiplied by this volume correction term as follows

$$\theta^b \times [\text{volume correction term}] = (\texttt{trcbl})_{i,j,k} \Big[(\texttt{volt\_vara})_{i,j,k} - (\texttt{volt\_varb})_{i,j,k} - (1-2\gamma)\Delta t \left\{ (\texttt{ws\_total})_{i,j} + (\texttt{transport\_wflux})_{i,j} \right\} (\texttt{thk\_ratio\_t})_{i,j,k} \Big] \tag{4.18}$$

A similar correction is proposed in SM2005 and corresponds to Eq. (4.8) in SM2005 and Eq. (4.3) in SM2009. However, in SM2005, the layer thickness is modified to achieve local conservation. In MRI.COM, local conservation is realized by modifying the tracer × volume because modifying the layer thickness is quite troublesome.

■Reduction of calculation costs The active tracer (temperature and salinity) values at $n+\frac{1}{2}$ is used only for calculating the pressure gradient through density and discarded after density is obtained. To reduce its computational costs, an advection scheme with low computational costs is used. Calculation of passive tracers are omitted in the predictor substep.

4.3.3 predictor step + baroclinic velocity

In the predictor substep, velocity is calculated as follows (Fig. 4.4).

$$u^{n+1/2} = \left(\frac{1}{2} - 2\gamma\right) u^{n-1} + \left(\frac{1}{2} + 2\gamma\right) u^n + (1-2\gamma)\Delta t \cdot [\text{the right hand side of the momentum equation } (\text{RHS}_u)^n] \tag{4.19}$$

Here, the density used to calculate the pressure gradient of the RHS_u^n is a linear combination of θ at $n+\frac{1}{2}$, n, and $n-1$ steps (θ_{filter}).

$$\theta_{filter} = \theta^n + \beta \frac{2\theta^{n+\frac{1}{2}} - 3\theta^n + \theta^{n-1}}{1-2\gamma} \tag{4.20}$$

Corresponding to the above equation, the model variables are expressed as follows.

- ubl = $\left(\frac{1}{2} - 2\gamma\right) u^{n-1} + \left(\frac{1}{2} + 2\gamma\right) u^n$
- ul = u^n
- ual = $u^{n+\frac{1}{2}}$

The update of ubl and ul is done at the end of the corrector substep in preparation for the next step.

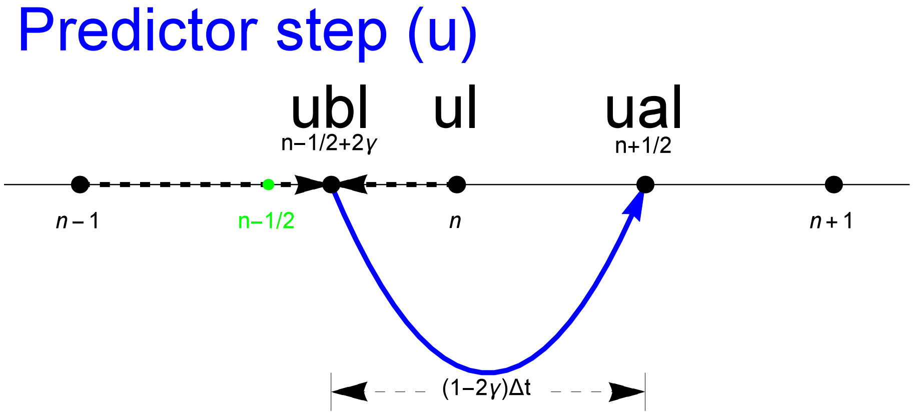

Figure4.4 Schematics of the LFAM3 time step in MRI.COM. The beginning of the predictor substep is at $n-1/2+2\gamma$ step, which is a linear combination of $n-1$ and n.

Since $u^{n+\frac{1}{2}}$ is used for the advection of tracers, RHS_u^n must be obtained with high precision. The ual obtained here is a provisional value and will be modified after barotropic velocity is obtained.

4.3.4 Barotropic velocity

After the calculation of baroclinic velocity in the predictor substep, the calculation of barotropic velocity is performed to obtain its values at $n+\frac{1}{2}$ step ($U^{n+\frac{1}{2}}$) and at $n+1$ steps (U^{n+1}) (Chapter 7). The sea level height at $n+1$ steps (η^{n+1}) can also be computed here, but in the model η^{n+1} is calculated using the continuous expression $(\eta^{n+1} - \eta^n)/\Delta t = \nabla \cdot U^{n+\frac{1}{2}}$ and $U^{n+\frac{1}{2}}$ in the corrector substep. Actually, the values obtained in the above two methods are identical except for the floating point error.

■Forcing for barotropic component of the equation of motionzul,zvl
The forcing (zul) should be evaluated at $n+\frac{1}{2}$ step. The component of external forcings such as wind stress are obtained at $n+\frac{1}{2}$ step. Since other components such as advection are obtained at n step, we make an extrapolation with respect to time to make them at $n+\frac{1}{2}$ step. There are several options for this extrapolation.

The first option is the 3rd-order Adams-Bashforth (AB3) method, which has $O(\Delta t^3)$ accuracy. *

$$\mathtt{zul}^{n+\frac{1}{2}} = \frac{23}{12}\mathtt{zul}^n - \frac{4}{3}\mathtt{zul}^{n-1} + \frac{5}{12}\mathtt{zul}^{n-2} \tag{4.21}$$

The second option is the quasi-2nd-order Adams-Bashforth (AB2ϵ) method, which does not have $O(\Delta t^2)$ accuracy, but slightly more stable than the AB3 method. This AB2ϵ method is used, for example, to estimate the advection term for the MITgcm. Usually 0.1 is used for ϵ_{AB}.

$$\mathtt{zul}^{n+\frac{1}{2}} = \left(\frac{3}{2} + \epsilon_{AB}\right)\mathtt{zul}^n - \left(\frac{1}{2} + \epsilon_{AB}\right)\mathtt{zul}^{n-1} \tag{4.22}$$

■Calculation of the barotropic component of the equation of motion The external forcings such as wind stress at the surface at $n+\frac{1}{2}$ step are added to the $\mathtt{zul}^{n+\frac{1}{2}}$ obtained above to yield a forcing for the barotropic component of the equation motion without time variation from n to $n+1$ steps. Using the subcycle velocity U^m and the time-filters a_m, b_m (Fig. 4.5) described in Chapter 3 of SM2005, the barotropic velocities are calculated as follows.

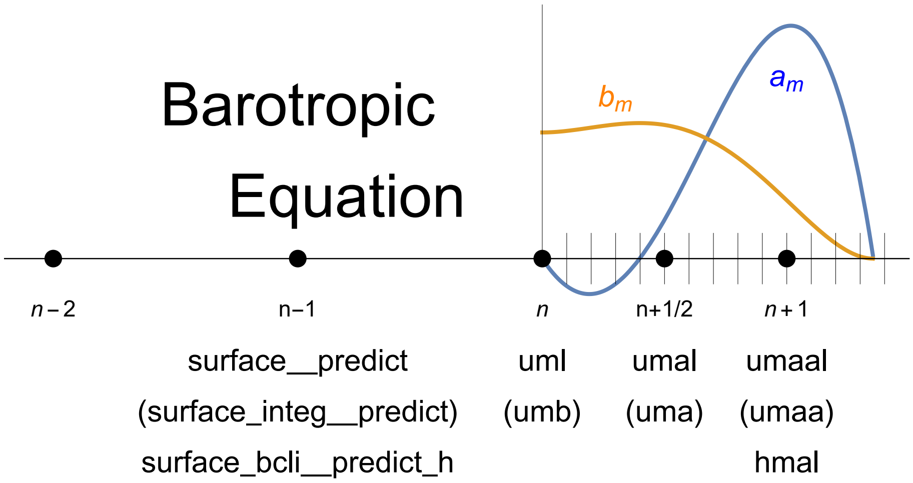

Figure4.5 Momentum equation for the barotropic component in MRI.COM in the case of LFAM3 method

$$U^{n+\frac{1}{2}} = \sum^{m} b_m U^m \tag{4.23}$$

$$U^{n+1} = \sum^{m} a_m U^m \tag{4.24}$$

$$\eta^{n+1} = \sum^{m} a_m \eta^m \tag{4.25}$$

(4.26)

* This AB3 method is used in ROMS other than UCLA version

Here,

$$\sum^{m} a_m = 1 \tag{4.27}$$

$$\sum^{m} m\Delta t_{tr} a_m = \Delta t_{cl} \tag{4.28}$$

$$b_{m'}\Delta t_{cl} = \sum_{m=m'}^{m} a_m \Delta t_{tr} \tag{4.29}$$

$$(\eta^{n+1} - \eta^n)/\Delta t_{cl} = \nabla \cdot U^{n+1/2} \tag{4.30}$$

$$(\eta^{m+1} - \eta^m)/\Delta t_{tr} = \nabla \cdot U^{m+1/2} \tag{4.31}$$

The $U^{n+\frac{1}{2}}$ is also used to find the sea level η^{n+1} at $n+1$ steps using the continuity equation. Therefore, the equation (4.12) holds for $n, n+1$ steps, and tracers advected by the flow velocity using this barotropic velocity field satisfy local conservation.

The values of $n+\frac{1}{2}$-step and $n+1$-step in LFAM3 correspond to n-step and $n+1$-step in LF. When LF was used in the previous MRI.COM, the barotropic velocity values at these times were used separately during the integration of LF from $n-1$-step to $n+1$-step and from n-step to $n+2$-step, respectively. In the case of LFAM3, the barotropic component of equation of motion is calculated only once, and the computational cost of the barotropic component of equation of motion is halved in MRI.COM compared to the case when LF is used.

4.3.5 Corrector substep

Corrector substep (Fig. 4.6) is calculated as follows. The calculation of the corrector substep is no different from that of a normal leap-frog method except for the pressure gradient term, which does not require any special processing such as changing the starting point of the time integration. Only the results of this step are used for subsequent calculations and historical output.

$$\theta^{n+1} = \theta^n + \Delta t \cdot [\text{The right hand side of the tracer equation } (\text{RHS}_\theta)^{n+\frac{1}{2}}] \tag{4.32}$$

$$u^{n+1} = u^n + \Delta t \cdot [\text{The right hand side of the momentum equation } (\text{RHS}_u)^{n+\frac{1}{2}}] \tag{4.33}$$

Here, density used to calculate the pressure gradient of RHS_u^n is a linear combination of θ (θ_{filter}).

$$\theta_{filter} = (1-\epsilon)\theta^{n+\frac{1}{2}} + \epsilon\left[\left(\frac{1}{2}-\gamma\right)\theta^{n+1} + \left(\frac{1}{2}+2\gamma\right)\theta^n - \gamma\theta^{n-1}\right] \tag{4.34}$$

The tracer equation is compatible with high precision advection scheme such as SOM because it uses only the n-step tracer values to find the $n+1$-step values by forward integration. Since the $n+\frac{1}{2}$ value is discarded after the corrector substep, only the values at corrector substep needs to be considered for conservation of the entire system, including the interaction with sea ice and external forces.

4.3.6 Summary of predictor and corrector subcycles

The calculation of the integral from n to $n+1$ steps when using LFAM3 in MRI.COM is summarized in Fig. 4.7. The renaming of arrays such as `trcl` is conducted in `subroutine rewrit` (Fig. 4.7(c) and (e)).

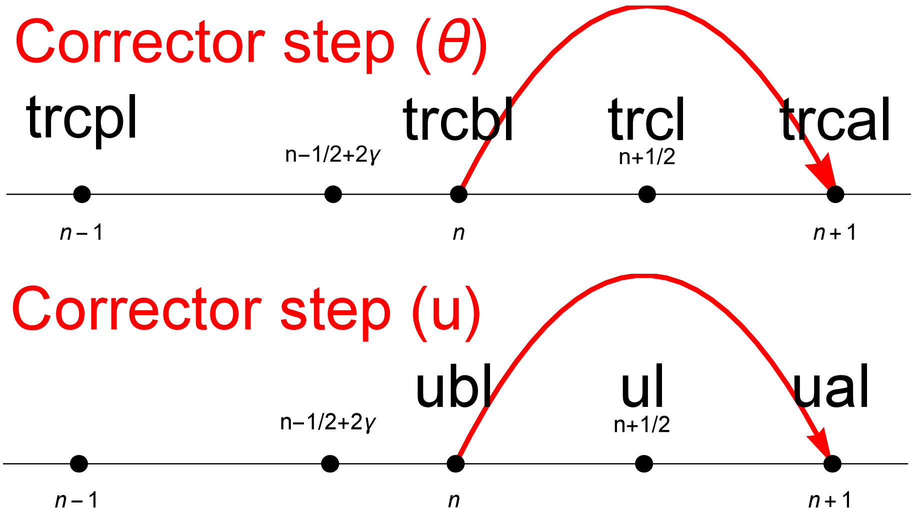

Figure4.6　Time integration of the corrector substep

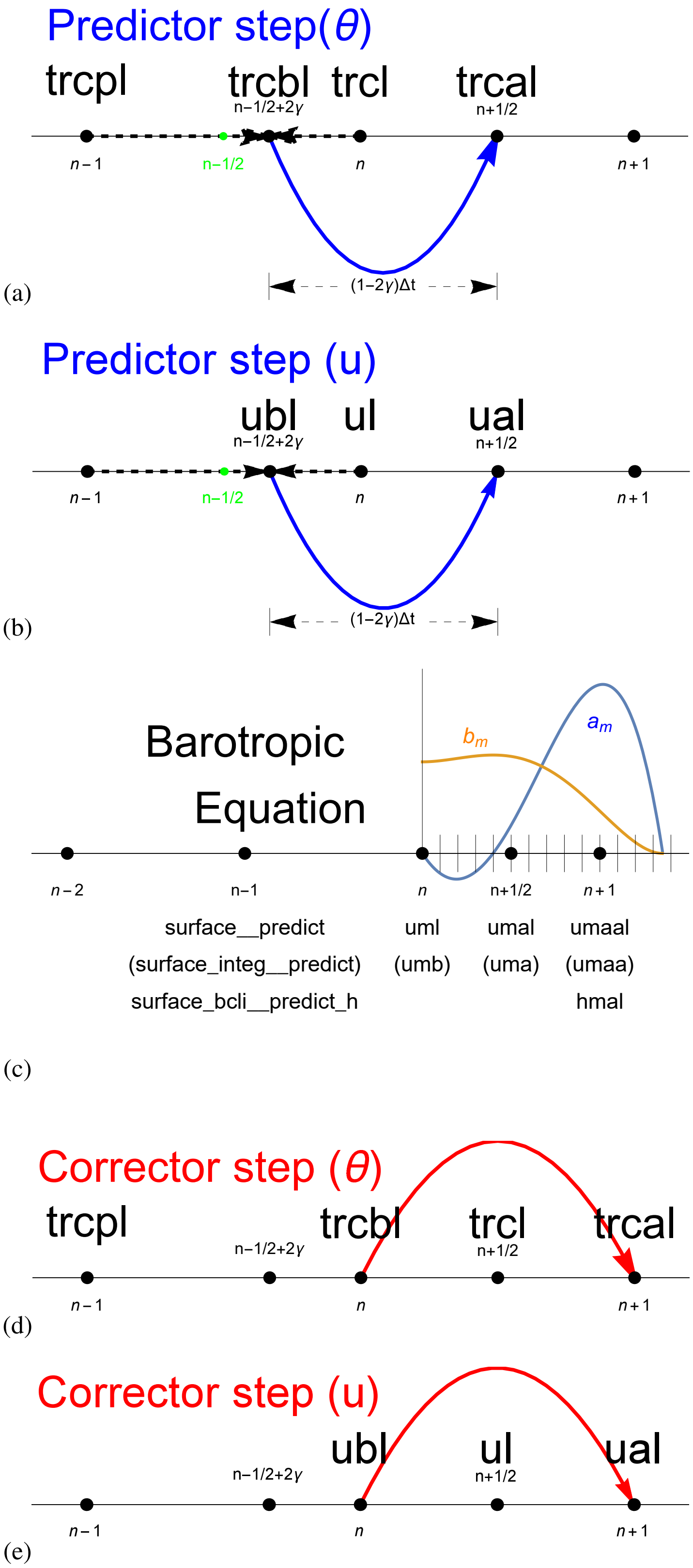

Figure4.7 Summary of the LFAM3 time integration method in MRI.COM. (a,b) The starting point of the predictor step is $n-\frac{1}{2}+2\gamma$ step, which is a linear combination of $n-1$ and n steps. Time integration is conducted from $n-\frac{1}{2}+2\gamma$ step to $n+\frac{1}{2}$ step by $(1-2\gamma)\Delta t$. (d,e) In the corrector substep, time integration is conducted from n step to $n+1$ step using the value at $n+\frac{1}{2}$ step. The values at $n+\frac{1}{2}$ step are discarded after the corrector substep.

Part II

Diagnostic Equations

Chapter 5

Equation of State

The *in situ* density is needed to calculate the pressure gradient term in the momentum equation. As indicated in (2.5), the equation of state represents *in situ* density as a function of pressure, temperature, and salinity. Here we present the specific form of the equation of state. We still adhere to the 1980 International Equation of State of Seawater (EOS-80; UNESCO, 1981). Thermodynamic Equation Of Seawater - 2010 (TEOS-10; IOC, SCOR, and IAPSO, 2010) has not been adopted.

5.1 Basics of the equation of state

The standard equation of state provided by UNESCO (1981) represents density (kg m^{-3}) as a function of *in situ* temperature[*] t (°C), salinity S (practical salinity scale (pss) ~ parts per thousand (ppt)), and pressure P (bar). Note that *in situ* temperature is used, not potential temperature. Density (ρ_w) of pure water ($S = 0$) under sea level pressure is given as a function of temperature (t):

$$\begin{aligned} \rho_w(t) \;=\; & 999.842594 + 6.793952 \times 10^{-2} t - 9.095290 \times 10^{-3} t^2 \\ & + 1.001685 \times 10^{-4} t^3 - 1.120083 \times 10^{-6} t^4 + 6.536332 \times 10^{-9} t^5. \end{aligned} \tag{5.1}$$

Density at the sea surface ($\rho_{\text{surf}} = \rho(t, S, 0)$) is expressed using sea surface temperature and salinity:

$$\begin{aligned} \rho_{\text{surf}} \;=\; & \rho_w \\ & + (0.824493 - 4.0899 \times 10^{-3} t + 7.6438 \times 10^{-5} t^2 - 8.2467 \times 10^{-7} t^3 + 5.3875 \times 10^{-9} t^4) S \\ & + (-5.72466 \times 10^{-3} + 1.0227 \times 10^{-4} t - 1.6546 \times 10^{-6} t^2)\, S^{\frac{3}{2}} \\ & + 4.8314 \times 10^{-4} S^2. \end{aligned} \tag{5.2}$$

The *in situ* density ($\rho = \rho(t, S, P)$) is converted from ρ_{surf} using the following equation,

$$\rho = \rho_{\text{surf}} / (1 - P/K), \tag{5.3}$$

where $K(S, t, P)$ is the secant bulk modulus. Its value at pure water, K_w, is given by

$$K_w \;=\; 19652.21 + 1.484206 \times 10^2 t - 2.327105 t^2 + 1.360477 \times 10^{-2} t^3 - 5.155288 \times 10^{-5} t^4. \tag{5.4}$$

The value at the sea surface (K_0) is given by

$$\begin{aligned} K_0 \;=\; & K_w + (54.6746 - 0.603459 t + 1.09987 \times 10^{-2} t^2 - 6.1670 \times 10^{-5} t^3) S \\ & + (7.944 \times 10^{-2} + 1.6483 \times 10^{-2} t - 5.3009 \times 10^{-4} t^2)\, S^{\frac{3}{2}}, \end{aligned} \tag{5.5}$$

[*] EOS-80 provided by UNESCO (1981) was based on temperature on the International Practical Temperature Scale of 1968 (IPTS-68; t_{68}) and the Practical Salinity Scale 1978 (PSS-78). After that the International Committee for Weights and Measures adopted a new temperature scale (the International Temperature Scale of 1990 (ITS-90); t_{90}). It has been recommended that t_{90} should be converted to t_{68} by a relation $t_{68} = 1.00024 * t_{90}$ when it is used for EOS-80 (Saunders, 1990). However, this conversion has not been applied for temperature in any operation of MRI.COM.

Table5.1 Coefficients for the lapse rate of sea water (Eq. 5.8).

a_0	$+3.5803 \times 10^{-5}$	b_0	$+1.8932 \times 10^{-6}$	c_0	$+1.8741 \times 10^{-8}$	d_0	-1.1351×10^{-10}
a_1	$+8.5258 \times 10^{-6}$	b_1	-4.2393×10^{-8}	c_1	-6.7795×10^{-10}	d_1	$+2.7759 \times 10^{-12}$
a_2	-6.8360×10^{-8}			c_2	$+8.7330 \times 10^{-12}$	e_0	-4.6206×10^{-13}
a_3	$+6.6228 \times 10^{-10}$			c_3	-5.4481×10^{-14}	e_1	$+1.8676 \times 10^{-14}$
						e_2	-2.1687×10^{-16}

and the value at pressure P is given by

$$\begin{aligned}K =& K_0 \\ &+ P\,(3.239908 + 1.43713 \times 10^{-3} t + 1.16092 \times 10^{-4} t^2 - 5.77905 \times 10^{-7} t^3) \\ &+ P\,(2.2838 \times 10^{-3} - 1.0981 \times 10^{-5} t - 1.6078 \times 10^{-6} t^2)\, S \\ &+ P\,(1.91075 \times 10^{-4})\, S^{\frac{3}{2}} \\ &+ P^2\,(8.50935 \times 10^{-5} - 6.12293 \times 10^{-6} t + 5.2787 \times 10^{-8} t^2) \\ &+ P^2\,(-9.9348 \times 10^{-7} + 2.0816 \times 10^{-8} t + 9.1697 \times 10^{-10} t^2)\, S.\end{aligned} \tag{5.6}$$

When potential temperature (θ) is available, it should be converted to *in situ* temperature. The conversion equation is obtained as follows using the adiabatic lapse rate $\Gamma(t, S, P)$:

$$t(\theta_0, S, P) = \theta_0 + \int_{P_0}^{P} \Gamma(t, S, P')dP'. \tag{5.7}$$

A polynomial for the adiabatic lapse rate $\Gamma(t, S, P)$ is given by UNESCO (1983):

$$\begin{aligned}\Gamma(t, S, P) =& a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ &+ (b_0 + b_1 t)(S - 35) \\ &+ \{c_0 + c_1 t + c_2 t^2 + c_3 t^3 + (d_0 + d_1 t)(S - 35)\}\, P \\ &+ (e_0 + e_1 t + e_2 t^2)\, P^2.\end{aligned} \tag{5.8}$$

Coefficients are given on Table 5.1.

5.2 An equation of state used by MRI.COM

5.2.1 Formulation

Since potential temperature (θ) is a prognostic variable in MRI.COM, it is desirable that the potential temperature can be directly used for the equation of state in the model. If this is possible, it will not be necessary to convert between potential temperature and *in situ* temperature. Accordingly, the equation of state of MRI.COM is a polynomial of almost the same form as UNESCO, but it is now the function of θ, S, and P and has a modified set of parameters. The parameters are determined by the least square fit for a realistic range of potential temperature and salinity following the method of Ishizaki (1994).

In MRI.COM version 4, considering the treatment of brackish waters in coastal modeling as well as the practical demand of numerical stability, we slightly modified the polynomial form and revised the set of coefficients relative to the previous version. The process of determining coefficients will be described below. We allow salinity to take slightly negative values that could be caused by numerical errors. So we use a slightly wider range ($-2 \leq \theta \leq 40$ °C, $-2 \leq S \leq 42$ pss, and $0 \leq P \leq 1000$ bar) than the previous version.

First, density is calculated at the sea surface (potential density or σ_θ) using equations (5.1) and (5.2). We introduce a minor modification from $(S)^{\frac{3}{2}}$ to $(|S|)^{\frac{3}{2}}$. It has been confirmed that this modification leads to a smooth density profile at $S = 0$.

The pressure dependent part, or specific volume $K(\theta, S, P)$ is given by

$$\begin{aligned}K(\theta, S, P) =& e_1(P) + e_2(P)\theta + e_3(P)\theta^2 + e_4(P)\theta^3 + e_5(P)\theta^4 \\ &+ S(f_1(P) + f_2(P)\theta + f_3(P)\theta^2 + f_4(P)\theta^3) \\ &+ (|S|)^{\frac{3}{2}}(f_5(P) + f_6(P)\theta + f_7(P)\theta^2),\end{aligned} \tag{5.9}$$

where

$$\begin{aligned} e_1(P) &= ec_1 + (gc_1 + hc_1 P)P, & f_1(P) &= fc_1 + (gc_5 + hc_4 P)P, \\ e_2(P) &= ec_2 + (gc_2 + hc_2 P)P, & f_2(P) &= fc_2 + (gc_6 + hc_5 P)P, \\ e_3(P) &= ec_3 + (gc_3 + hc_3 P)P, & f_3(P) &= fc_3 + (gc_7 + hc_6 P)P, \\ e_4(P) &= ec_4 + gc_4 P, & f_4(P) &= fc_4, \\ e_5(P) &= ec_5, & f_5(P) &= fc_5 + gc_8 P, \\ & & f_6(P) &= fc_6, \\ & & f_7(P) &= fc_7. \end{aligned} \tag{5.10}$$

The set of coefficients in the above equation is computed using a least square fit and listed on Table 5.2. In the previous version, uniform 1.0 °C, 1.0 pss, 10 decibar bins were used for the least square fit in the range of $-2 \le \theta \le 40$ °C, $0 \le S \le 42$ pss, and $0 \le P \le 1000$ bar. But it is found that maximum error tends to occur in the range of low potential temperature and low salinity, and increasing the number of bins in this range is favorable for reducing the error (not shown). Therefore, potential temperature and salinity bins are changed from 1.0 °C and 1.0 pss to 0.1 °C and 0.1 pss for $-2 \le \theta \le 10$ °C and $-2 \le S \le 10$ pss, respectively. Using 151×151×101 combinations of the above range of potential temperature, salinity, and pressure, *in situ* temperature is first computed using (5.7). Density is then calculated by the UNESCO equations using *in situ* temperature and salinity. The above coefficients are determined using these data of density, potential temperature, salinity, and pressure by the least square method. They are given on Table 5.2.

Table5.2 Coeffients for the equation of state of sea water. See equation (5.10).

ec_1	19659.35	fc_1	52.85624	gc_1	3.185918	hc_1	$2.111\,102 \times 10^{-4}$
ec_2	144.5863	fc_2	$-3.128\,126 \times 10^{-1}$	gc_2	$2.189\,412 \times 10^{-2}$	hc_2	$-1.196\,438 \times 10^{-5}$
ec_3	−1.722523	fc_3	$6.456\,036 \times 10^{-3}$	gc_3	$-2.823\,685 \times 10^{-4}$	hc_3	$1.364\,330 \times 10^{-7}$
ec_4	$1.019\,238 \times 10^{-2}$	fc_4	$-5.370\,396 \times 10^{-5}$	gc_4	$1.715\,739 \times 10^{-6}$	hc_4	$-2.048\,755 \times 10^{-6}$
ec_5	$-4.768\,276 \times 10^{-5}$	fc_5	$3.884\,013 \times 10^{-1}$	gc_5	$6.703\,377 \times 10^{-3}$	hc_5	$6.375\,979 \times 10^{-8}$
		fc_6	$9.116\,446 \times 10^{-3}$	gc_6	$-1.839\,953 \times 10^{-4}$	hc_6	$5.240\,967 \times 10^{-10}$
		fc_7	$-4.628\,163 \times 10^{-4}$	gc_7	$1.912\,264 \times 10^{-7}$		
				gc_8	$1.477\,291 \times 10^{-4}$		

With the new set of coefficients, the maximum density error (thick dashed line) relative to the UNESCO equation is less than 1.6×10^{-3} kg m^{-3} when pressure is less than 5000 dbar (Figure 5.1). This maximum error is less than that (3×10^{-3} kg m^{-3}) of McDougall et al. (2003)'s equation of state. The error is also smaller than that using the previous version (version 3 and earlier) in this pressure range (thin dashed line). Though the relatively larger maximum errors are found when the pressure is greater than 8000 dbar, they occur only when S ≃ 40 pss or S ≃ 0 pss (not shown) and would not cause a serious problem. The standard deviation (thick solid line) is less than 1×10^{-3} kg m^{-3} when pressure is less than 8000 dbar.

5.2.2 Implementation

In MRI.COM, *in situ* density is computed by (5.3) using (5.2) for ρ_{surf} and (5.9) for K. For Pressure (P), a horizontally uniform value is used on each vertical level where densities are evaluated, instead of using the actual pressure at each location. This is based on the assumption that horizontal variation of pressure on a constant depth (z^*) surface does not affect the required accuracy of density.

In the program code of MRI.COM, the *in situ* density is evaluated at the depth where the tracer is defined and the depth of the boundary of tracer cells. In the standard case, pressure at the depth where tracer is defined ($\overline{P_{k-\frac{1}{2}}}$) and that at the depth of the cell boundary ($\overline{P_k}$) are calculated as follows:

$$10^6 \overline{P_{k-\frac{1}{2}}} = \rho_0 g z_{k-\frac{1}{2}}, \tag{5.11}$$

$$10^6 \overline{P_k} = \rho_0 g z_k, \tag{5.12}$$

where ρ_0 is the reference density, g is the acceleration due to gravity (Table 2.1), and $z_{k-\frac{1}{2}}$ and z_k are the actual depths in the state of rest. Note that a factor 10^6 converts pressure in cgs units (dyn cm^{-2}) to bar.

If CALPP option is chosen, the time variation of these horizontally uniform pressures is considered. In this case, pressures are evaluated as follows:

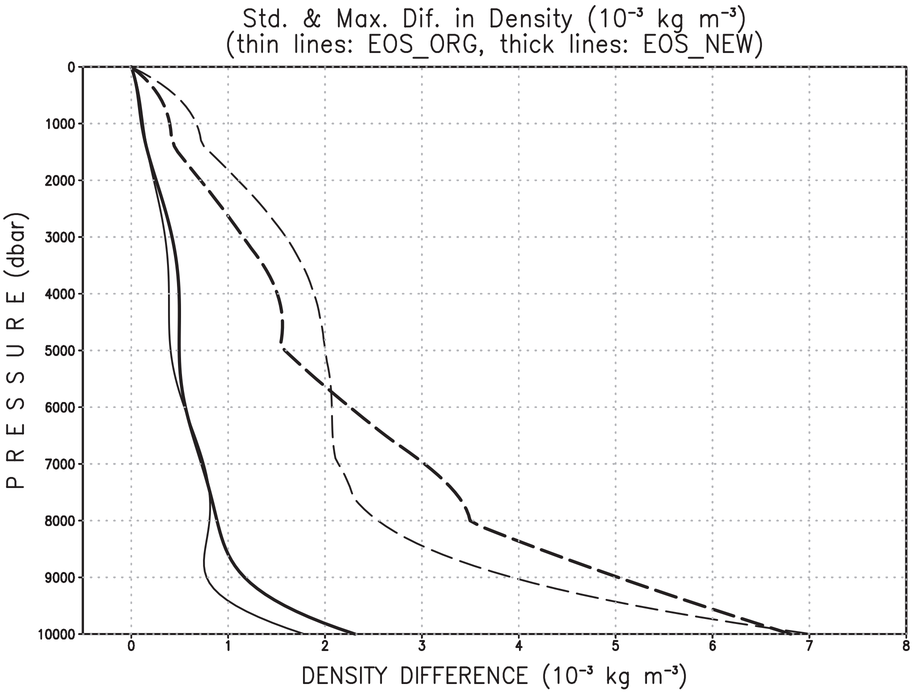

Figure5.1 Deviation from the UNESCO equation for the old (version 3 and earlier) and new (version 4) set of coefficients. Unit is $10^{-3}\,\mathrm{kg\,m^{-3}}$. Thin lines represent the equation of state using the old coefficients and thick lines represent those of the new ones. The solid (long dashed) lines represent the standard deviation (maximum error) from the UNESCO equation in the range of $-2^\circ \le \theta \le 10\,^\circ$C and $10 \le S \le 40$ pss.

$$10^6\overline{P_{\frac{1}{2}}} = \frac{1}{2} g \Delta z_{\frac{1}{2}} \overline{\rho_{\frac{1}{2}}}^{xy}, \tag{5.13}$$

$$10^6\overline{P_{k-\frac{1}{2}}} = 10^6 P_{\frac{1}{2}} + g \sum_{l=2}^{k} \Delta z_{l-1} \frac{\overline{\rho_{l-\frac{3}{2}}}^{xy} + \overline{\rho_{l-\frac{1}{2}}}^{xy}}{2} \quad (k \ge 2), \tag{5.14}$$

$$10^6\overline{P_k} = g \sum_{k=1}^{k} \Delta z_{k-\frac{1}{2}} \overline{\rho_{k-\frac{1}{2}}}^{xy}. \tag{5.15}$$

where $\overline{\rho_{k-\frac{1}{2}}}^{xy}$ represents horizontally averaged density at $(k-\frac{1}{2})$th level. In addition, the density averaged for the entire model domain $(\bar{\rho})$ is used as the reference density in the momentum equations instead of ρ_0.

These quantities are evaluated at a time step interval specified by namelist `nml_calpp` (Table 5.3) and should be stored in a restart file to be used in the next run (Table 5.4). In the restart file, horizontally averaged density at tracer levels $(\overline{\rho_{k-\frac{1}{2}}}^{xy})$, the density averaged for the entire model domain $(\bar{\rho})$, pressures used for the equation of state $(\overline{P_{k-\frac{1}{2}}}$ and $\overline{P_k})$ are stored. The restart file has the following format:

Format of the restart file for the equation of state (nmlrs_density).

```
read(8)       :: dmn(km), ddmna, pm(km+1), pd(km)
integer(4)    :: k, reck
real(8)       :: density_array(3*km+2)
character(14) :: date = '20010101000000'  !- for 0:00z1JAN2001
integer(4),parameter :: nu = 10 ! device number

open(nu, file='result/rs_density.'//date, form='unformatted', &
     & access='direct',recl=(3*km+2)*8)
read(nu, rec=1) density_array
close(nu)

reck = 0
do k = 1, km
  reck = reck + 1
  dmn(k) = density_array(reck) ! horizontally averaged density
end do
reck = reck + 1
ddmna = density_array(reck)    ! density averaged over the entire model domain
do k = 1, km + 1
  reck = reck + 1
  pm(k) = density_array(reck) ! pressure at the tracer cell boundaries
end do
do k = 1, km
  reck = reck + 1
  pd(k) = density_array(reck) ! pressure at the tracer levels
end do
```

Table5.3 namelist nml_calpp

variable name	units	description	usage
nstep_calpp_interval	1	time step interval with which horizontally averaged density and pressure are evaluated for the equation of state	required if CALPP

Table5.4 namelist nml_density_run

variable name	units	description	usage
l_rst_density_in	logical	.true. : Read restart files specified by nmlrs_density for the initial condition. .false. : Start condition is calculated by the 3-D density field of the initial state.	default = l_rst_in of nml_run_ini_state, see Table 25.5.

Chapter 6

Continuity equation

The mass (volume) fluxes, which are fundamental for estimating the advection of momentum and tracers, are calculated on the basis of the finite difference expression of the continuity equation. Owing to the use of staggered grid arrangement, the finite difference expression of the continuity equation differs for the T-cell and U-cell (Figure 6.1). In MRI.COM, the mass continuity for the T-cell is primary (Sec. 6.2), and that for the U-cell is derived from the former by an averaging operation (Sec. 6.3). By this, we can avoid spurious vertical mass fluxes for the U-cell continuity, which appear when the U-cell continuity is calculated independently of the T-cell continuity, with the largest error magnitude increasing as the grid size decreases (Webb, 1995).

To summarize the solution procedure, MRI.COM uses (6.12) to obtain W^T under the boundary condition (6.11). Then, W^U is obtained by (6.27) using W^T. These are used for advection of tracers and momentum.

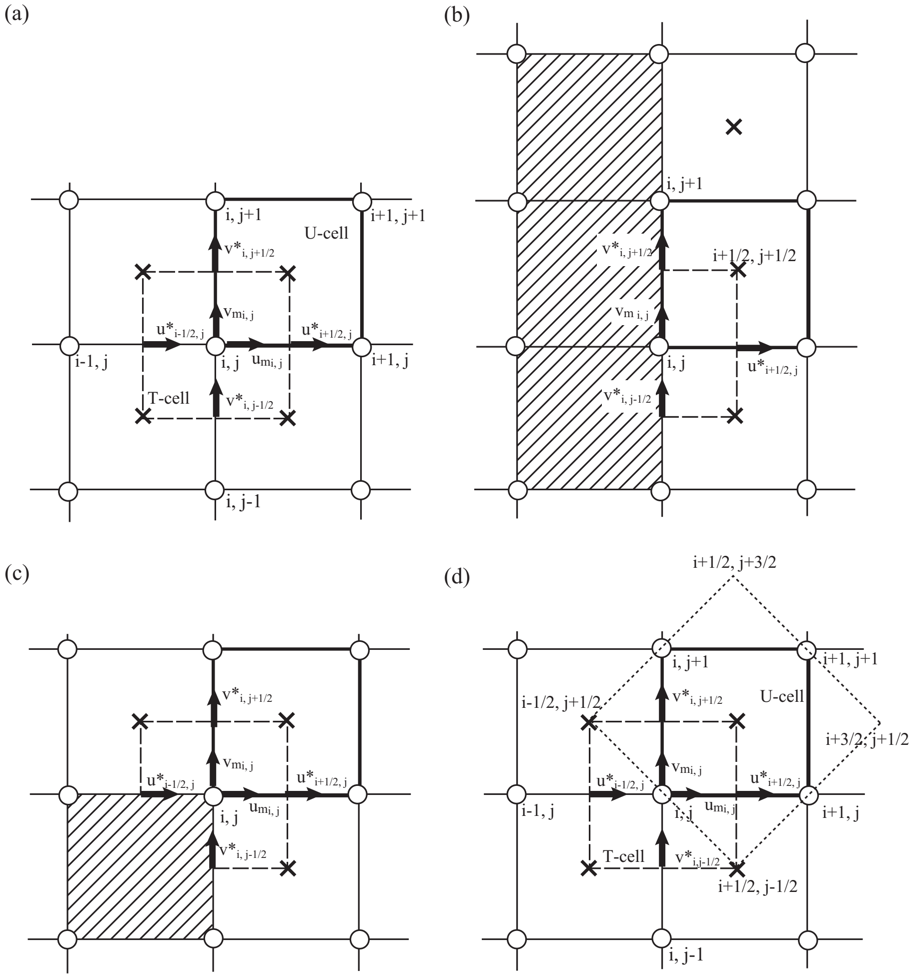

Figure6.1 Horizontal arrangement of variables for the continuity equation. (a) Relationship between T-cell and U-cell (standard form). (b),(c) Relationship between T-cell and U-cell near the coast. (d) Diagonal square grid cell and mass fluxes.

6.1 Semi-discrete form of the continuity equation

Among the related governing equations, the continuity equation (2.34),

$$\frac{\partial z_s}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial (z_s h_\psi u)}{\partial \mu} + \frac{\partial (z_s h_\mu v)}{\partial \psi}\right\} + \frac{\partial (z_s w^*)}{\partial s} = 0, \tag{6.1}$$

the free surface equation (2.38),

$$\frac{\partial (\eta + H)}{\partial t} + \frac{1}{h_\mu h_\psi}\left[\frac{\partial (h_\psi U)}{\partial \mu} + \frac{\partial (h_\mu V)}{\partial \psi}\right] = P - E + R + I, \tag{6.2}$$

and the tracer equations (2.35, 2.36)

$$\frac{\partial (z_s \theta)}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial (z_s h_\psi u \theta)}{\partial \mu} + \frac{\partial (z_s h_\mu v \theta)}{\partial \psi}\right\} + \frac{\partial (z_s w^* \theta)}{\partial s} = -z_s \nabla \cdot \mathrm{F}_\theta + z_s Q_\theta, \tag{6.3}$$

$$\frac{\partial (z_s S)}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial (z_s h_\psi u S)}{\partial \mu} + \frac{\partial (z_s h_\mu v S)}{\partial \psi}\right\} + \frac{\partial (z_s w^* S)}{\partial s} = -z_s \nabla \cdot \mathrm{F}_S + z_s Q_S, \tag{6.4}$$

must be mutually consistent in order to keep sign-definiteness of tracers. Hereafter, since the z^* coordinate system is used, $\dot{s}$ is expressed as w^* (Eq. 2.42). The purpose of this chapter is to explain how to integrate (6.1) vertically from the surface in order to obtain w^*. The surface boundary condition of Eq. (6.1) is

$$z_s w^*_{z^*=0} = -(P - E + R + I), \tag{6.5}$$

based on (2.47). The surface flux of a state variable caused by the surface fresh water transport (r.h.s. of the above) is treated as the surface boundary condition for the advection term of that state variable.

Before showing the finite difference equations for T-cell and U-cell, the vertical integration of (6.1) for the k-th vertical level is performed as follows:

$$(z_s w^*)_k = (z_s w^*)_{k-1} + \left[\Delta s_{k-\frac{1}{2}} (\partial_t z_s)\right] + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial\left(h_\psi \Delta z_{k-\frac{1}{2}} u_{k-\frac{1}{2}}\right)}{\partial \mu} + \frac{\partial\left(h_\mu \Delta z_{k-\frac{1}{2}} v_{k-\frac{1}{2}}\right)}{\partial \psi}\right\}, \tag{6.6}$$

where $\Delta z_{k-\frac{1}{2}}$ is the width of the $(k - \frac{1}{2})$-th layer and $\Delta s_{k-\frac{1}{2}}$ is the logical width of s of the $(k - \frac{1}{2})$-th layer. Note also that $\partial_t z_s$ is independent of depth as:

$$\partial_t z_s = \frac{\partial}{\partial t}\left(\frac{H + \eta}{H}\right) = \frac{\partial_t \eta}{H}, \tag{6.7}$$

using Eq. (2.41).

6.2 Finite difference expression for the T-cell continuity equation

In MRI.COM, vertical velocity is not computed but vertical transport is extensively used. The vertical transport for T-cell is mathematically expressed as

$$W^T \equiv z_s w^* \Delta A^T(s), \tag{6.8}$$

where $\Delta A^T(s)$ is the horizontal area of a T-cell[*].

We now integrate (6.6) over the T-cell, $\Delta A^T_{i,j,k-\frac{1}{2}}$:

$$\begin{aligned} W^T_{i,j,k} =& W^T_{i,j,k-1} + \Delta A^T_{i,j,k-\frac{1}{2}} (\partial_t \eta)_{i,j} \frac{\Delta s_{i,j,k-\frac{1}{2}}}{H} \\ & + U^{*T}_{i+\frac{1}{2},j,k-\frac{1}{2}} - U^{*T}_{i-\frac{1}{2},j,k-\frac{1}{2}} + V^{*T}_{i,j+\frac{1}{2},k-\frac{1}{2}} - V^{*T}_{i,j-\frac{1}{2},k-\frac{1}{2}}. \end{aligned} \tag{6.9}$$

Here, the horizontal transport across the T-cell is used for the horizontal divergence term as shown in Fig. 6.1(a),

$$U^{*T}_{i+\frac{1}{2},j} = \frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}})\Delta y_{i+\frac{1}{2},j}\Delta z, \quad V^{*T}_{i,j+\frac{1}{2}} = \frac{1}{2}(v_{i+\frac{1}{2},j+\frac{1}{2}} + v_{i-\frac{1}{2},j+\frac{1}{2}})\Delta x_{i,j+\frac{1}{2}}\Delta z. \tag{6.10}$$

[*] Horizontal cross section of a T-cell is treated as constant throughout the cell, that is, vertically bounding walls are assumed to move with the cell

However, if there are land cells, only the values of the sea cells are used as in Fig. 6.1(b) and (c). The surface boundary condition for $W^T_{i,j}$ is based on Eq. (6.5) as follows:

$$\begin{aligned} W^T_{i,j,0} =& z_s w^*_{s=0} \Delta A^T_{i,j,\frac{1}{2}} \\ =& -(P-E+R+I)_{i,j}\, \Delta A^T_{i,j,\frac{1}{2}}. \end{aligned} \tag{6.11}$$

With this condition and Eq. (6.9), W^T can be obtained in order from $k = 1$.

As a supplement to this subsection, consistency of Eq. (6.9) with the free surface equation is checked. We sum up (6.9) from the surface ($k = 1$) to the bottom ($k = ktbtm$) to obtain

$$\begin{aligned} W^T_{i,j,ktbtm} =& -(P-E+R+I)\Delta A^T_{i,j,\frac{1}{2}} + \Delta A^T_{i,j,\frac{1}{2}}(\partial_t \eta)_{i,j} + \frac{U_{i+\frac{1}{2},j+\frac{1}{2}} + U_{i+\frac{1}{2},j-\frac{1}{2}}}{2}\Delta y_{i+\frac{1}{2},j} \\ & - \frac{U_{i-\frac{1}{2},j+\frac{1}{2}} + U_{i-\frac{1}{2},j-\frac{1}{2}}}{2}\Delta y_{i-\frac{1}{2},j} + \frac{V_{i+\frac{1}{2},j+\frac{1}{2}} + V_{i-\frac{1}{2},j+\frac{1}{2}}}{2}\Delta x_{i,j+\frac{1}{2}} - \frac{V_{i+\frac{1}{2},j-\frac{1}{2}} + V_{i-\frac{1}{2},j-\frac{1}{2}}}{2}\Delta x_{i,j-\frac{1}{2}}, \end{aligned} \tag{6.12}$$

where U and V indicate vertically-integrated transports. Since

$$W^T_{i,j,ktbtm} = 0, \tag{6.13}$$

we have,

$$\Delta A^T_{i,j,\frac{1}{2}} \partial_t \eta_{i,j} = (\mathtt{ws})_{i,j} + (\mathtt{transport\_wflux})_{i,j}, \tag{6.14}$$

where

$$\begin{aligned} (\mathtt{ws})_{i,j} = -\Bigg(& \frac{U_{i+\frac{1}{2},j+\frac{1}{2}} + U_{i+\frac{1}{2},j-\frac{1}{2}}}{2}\Delta y_{i+\frac{1}{2},j} - \frac{U_{i-\frac{1}{2},j+\frac{1}{2}} + U_{i-\frac{1}{2},j-\frac{1}{2}}}{2}\Delta y_{i-\frac{1}{2},j} \\ & + \frac{V_{i+\frac{1}{2},j+\frac{1}{2}} + V_{i-\frac{1}{2},j+\frac{1}{2}}}{2}\Delta x_{i,j+\frac{1}{2}} - \frac{V_{i+\frac{1}{2},j-\frac{1}{2}} + V_{i-\frac{1}{2},j-\frac{1}{2}}}{2}\Delta x_{i,j-\frac{1}{2}} \Bigg) \end{aligned} \tag{6.15}$$

and

$$(\mathtt{transport\_wflux})_{i,j} = (P-E+R+I)_{i,j}\Delta A^T_{i,j,\frac{1}{2}}. \tag{6.16}$$

Thus, it is confirmed that an integrated form of (6.9) is consistent with the free surface equation (6.2).

Equation (6.14) can be used to describe the volume change of each cell, $\partial(\Delta V^T)/\partial t$.

$$\partial_t(\Delta V^T)_{i,j,k-\frac{1}{2}} = \Delta A^T_{i,j,k-\frac{1}{2}} \Delta s_{i,j,k-\frac{1}{2}} (\partial_t z_s)_{i,j}, \tag{6.17}$$

$$= \Delta A^T_{i,j,k-\frac{1}{2}} \Delta s_{i,j,k-\frac{1}{2}} \frac{\partial_t \eta_{i,j}}{H} \tag{6.18}$$

$$= \frac{\Delta s_{i,j,k-\frac{1}{2}}}{H}\left((\mathtt{ws})_{i,j} + (\mathtt{transport\_wflux})_{i,j} \right). \tag{6.19}$$

This relationship is used to predict the tracer value at the predictor step (Eq. 4.12).

6.3 Finite difference expression for the U-cell continuity equation

The finite difference expression of the continuity equation for a U-cell $(i+\frac{1}{2}, j+\frac{1}{2})$ is defined using those for T-cells (Figure 6.1(a, b, and c)). Mass convergence in a U-cell, MC^U, is given by

$$MC^U_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \equiv \frac{\partial_t(\Delta V^T)_{i,j,k-\frac{1}{2}}}{N_{i,j,k-\frac{1}{2}}} + \frac{\partial_t(\Delta V^T)_{i+1,j,k-\frac{1}{2}}}{N_{i+1,j,k-\frac{1}{2}}} + \frac{\partial_t(\Delta V^T)_{i,j+1,k-\frac{1}{2}}}{N_{i,j+1,k-\frac{1}{2}}} + \frac{\partial_t(\Delta V^T)_{i+1,j+1,k-\frac{1}{2}}}{N_{i+1,j+1,k-\frac{1}{2}}}, \tag{6.20}$$

based on (3.28). Here, $N_{i,j,k-\frac{1}{2}}$ is the number of sea grid cells around the T-point (i, j) in the $(k-\frac{1}{2})$th layer. Usually, $N = 4$ for T-cells away from land (Figure 6.1a), but $N < 4$ for the partial T-cells along coast lines (Figure 6.1b,c). This equation means that MC^U consists of the sum of the contributions from four surrounding T-cells, thinking that the following relation is obtained from Eq. (6.9):

$$\partial_t(\Delta V^T)_{i,j,k-\frac{1}{2}} = U^{*T}_{i-\frac{1}{2},j,k-\frac{1}{2}} - U^{*T}_{i+\frac{1}{2},j,k-\frac{1}{2}} + V^{*T}_{i,j-\frac{1}{2},k-\frac{1}{2}} - V^{*T}_{i,j+\frac{1}{2},k-\frac{1}{2}} + W^T_{i,j,k} - W^T_{i,j,k-1}. \tag{6.21}$$

The standard form of the mass continuity, which applies for U-cells $(i+\frac{1}{2}, j+\frac{1}{2})$ away from coast lines, is as follows:

$$MC^U_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \equiv \frac{1}{4}[\partial_t(\Delta V^T)_{i,j,k-\frac{1}{2}} + \partial_t(\Delta V^T)_{i+1,j,k-\frac{1}{2}} + \partial_t(\Delta V^T)_{i,j+1,k-\frac{1}{2}} + \partial_t(\Delta V^T)_{i+1,j+1,k-\frac{1}{2}}]. \tag{6.22}$$

Based on (6.7), (6.9) and (6.17), this can be rewritten as

$$\begin{aligned}\partial_t(\Delta V^U)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \simeq &\frac{1}{2}(U^U_{i,j} + U^U_{i,j+1}) - \frac{1}{2}(U^U_{i+1,j} + U^U_{i+1,j+1}) + \frac{1}{2}(V^U_{i,j} + V^U_{i+1,j}) - \frac{1}{2}(V^U_{i,j+1} + V^U_{i+1,j+1}) \\ &- W^U_{i+\frac{1}{2},j+\frac{1}{2},k-1} + W^U_{i+\frac{1}{2},j+\frac{1}{2},k}.\end{aligned} \tag{6.23}$$

Terms $U^U_{i,j}$, $V^U_{i,j}$, and $W^U_{i+\frac{1}{2},j+\frac{1}{2},k}$ are defined as follows:

$$U^U_{i,j} = u_{m\,i,j}\Delta y_{i,j}\Delta z, \quad V^U_{i,j} = v_{m\,i,j}\Delta x_{i,j}\Delta z, \tag{6.24}$$

$$u_{m\,i,j} = \frac{1}{2}(u^*_{i+\frac{1}{2},j} + u^*_{i-\frac{1}{2},j}), \quad v_{m\,i,j} = \frac{1}{2}(v^*_{i,j+\frac{1}{2}} + v^*_{i,j-\frac{1}{2}}), \tag{6.25}$$

$$u^*_{i+\frac{1}{2},j} = \frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}}), \quad v^*_{i,j+\frac{1}{2}} = \frac{1}{2}(v_{i+\frac{1}{2},j+\frac{1}{2}} + v_{i-\frac{1}{2},j+\frac{1}{2}}), \tag{6.26}$$

$$W^U_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{1}{4}(W^T_{i,j,k} + W^T_{i+1,j,k} + W^T_{i,j+1,k} + W^T_{i+1,j+1,k}). \tag{6.27}$$

Note that W^U is obtained by an averaging operation on W^T and also that the following equations are derived from (6.10) and (6.24) for the standard form of the U-cell continuity:

$$\begin{aligned}U^U_{i,j} &= \frac{1}{2}(U^{*T}_{i+\frac{1}{2},j} + U^{*T}_{i-\frac{1}{2},j}), \\ V^U_{i,j} &= \frac{1}{2}(V^{*T}_{i,j+\frac{1}{2}} + V^{*T}_{i,j-\frac{1}{2}}).\end{aligned} \tag{6.28}$$

The advecting velocity for momentum is based on (U^U, V^U, W^U). All the above relationships hold for the cases with variable grid sizes (Figure 3.1c).

Note that the r.h.s. of (6.23) does not give the exact volume tendency that would be obtained by following the definition of the volume of a U-cell (3.26). This results in a local violation of momentum conservation, though the momentum is globally conserved. We keep using the definition of the volume of a U-cell given by (3.26) because we do not want to use the simple average of the volumes of T-cells similar to (6.20) for defining the control volume of a U-cell. This comes from the need for a simple algebraic expression.

Finally, we discuss horizontal advection in this model from the perspective of advection classification. The r.h.s. of (6.23) expresses the convergence of mass fluxes along the horizontal coordinate axes and it is completed as far as the continuity equation is concerned. However, when the mass continuity is used to calculate the momentum advection, the r.h.s. of (6.23) is rewritten as follows to express the convergence of the diagonal mass fluxes to the coordinate axes, and is used together with its original form (6.23):

$$\begin{aligned}\partial_t(\Delta V^U)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \simeq &\frac{1}{2}(U^U_{i,j} + V^U_{i,j}) - \frac{1}{2}(U^U_{i+1,j+1} + V^U_{i+1,j+1}) + \frac{1}{2}(U^U_{i,j+1} - V^U_{i,j+1}) - \frac{1}{2}(U^U_{i+1,j} - V^U_{i+1,j}) \\ &- W^U_{i+\frac{1}{2},j+\frac{1}{2},k-1} + W^U_{i+\frac{1}{2},j+\frac{1}{2},k}\end{aligned} \tag{6.29}$$

Let us explain the meaning taking the first term on the r.h.s. of (6.29),

$$U^U_{i,j} + V^U_{i,j} = \frac{1}{2}\{(U^{*T}_{i-\frac{1}{2},j} + V^{*T}_{i,j-\frac{1}{2}}) + (U^{*T}_{i+\frac{1}{2},j} + V^{*T}_{i,j+\frac{1}{2}})\}, \tag{6.30}$$

as an example, where (6.28) is used. If the flow is horizontally nondivergent, the horizontal mass fluxes $U^{*T}_{i-\frac{1}{2},j}$ and $V^{*T}_{i,j-\frac{1}{2}}$ in the first term on the r.h.s. are expressed by streamfunction at two pairs of U-points, $(i-\frac{1}{2}, j+\frac{1}{2})$ and $(i-\frac{1}{2}, j-\frac{1}{2})$, and $(i-\frac{1}{2}, j-\frac{1}{2})$ and $(i+\frac{1}{2}, j-\frac{1}{2})$, respectively. Then, their sum corresponds to the net mass flux crossing the diagonal section connecting the two U-points $(i-\frac{1}{2}, j+\frac{1}{2})$ and $(i+\frac{1}{2}, j-\frac{1}{2})$ (Figure 6.1d). The second term on the r.h.s. expresses the same quantity, though the route is different. Thus, multiplying by a factor of two, the r.h.s. of (6.29) means the horizontal mass convergence in the diagonal square defined by four U-points $(i-\frac{1}{2}, j+\frac{1}{2})$, $(i+\frac{1}{2}, j-\frac{1}{2})$, $(i+\frac{3}{2}, j+\frac{1}{2})$, and $(i+\frac{1}{2}, j+\frac{3}{2})$. Multiplying by a factor of $\frac{1}{2}$, the r.h.s. of (6.29) itself means the horizontal mass convergence in the U-cell $(i+\frac{1}{2}, j+\frac{1}{2})$, whose area is a half of that of the diagonal square.

Taking the first four terms on the r.h.s. of (6.23) as $A_{i+\frac{1}{2},j+\frac{1}{2}}$ and those of (6.29) as $B_{i+\frac{1}{2},j+\frac{1}{2}}$, the standard form of the continuity equation for the U-cell used for the calculation of the momentum advection is generally expressed as:

$$\partial_t (\Delta V^U)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \simeq \alpha\ A_{i+\frac{1}{2},j+\frac{1}{2}} + \beta\ B_{i+\frac{1}{2},j+\frac{1}{2}} - W^U_{i+\frac{1}{2},j+\frac{1}{2},k-1} + W^U_{i+\frac{1}{2},j+\frac{1}{2},k}, \tag{6.31}$$

where

$$\alpha + \beta = 1. \tag{6.32}$$

As shown later in Chapter 8, $(\alpha, \beta) = (2/3, 1/3)$ for the generalized Arakawa scheme and $(\alpha, \beta) = (1/2, 1/2)$ for the standardized form derived from the continuity equation generalized for arbitrary bottom topography. What Webb (1995) proposed corresponds to $(\alpha, \beta) = (1, 0)$.

Part III

Equations of Motion

Chapter 7

Equations of motion (barotropic component)

MRI.COM predicts the barotropic and baroclinic components (modes) of the equation of motion separately using mode splitting (Sec. 2.2.2). In this chapter, we explain how to solve the barotropic component under the free-surface condtion with z^* vertical coordinate. This chapter contains an overview of mode splitting (Sec. 7.1), the governing equation of the barotropic model (Sec. 7.2), numerical solving techniques of the model (Sec. 7.3), prognostification of the state variables at the baroclinic time step (Sec. 7.4), and the time weighting factor (Sec. 7.5). In addition, a description of optional horizontal diffusion of sea surface height (Sec. 7.6) and a discussion about the effects of mode splitting on the equation of motion (Sec. 7.7) are included as appendices. Finally, we will briefly explain how to set the model (Sec. 7.8). The basic formulation in this chapter is the same as MRI.COMv4, though the representation of the time difference was partially changed with the introduction of the LFAM3 scheme in MRI.COMv5 (Sec. 4.3). See Chapter 6 of the MRI.COMv4 manual for barotropic models that were once used.

7.1 Overview of mode splitting

Before presenting the formulation of the barotropic model, an overview of the mode splitting process in MRI.COM is given. Figure 7.1 presents schematics of the time integration of the barotropic mode and the relation with the baroclinic mode. When the time integration of the baroclinic mode is performed from step n ($t = t_n$) to step $n+1$ ($t = t_{n+1} = t_n + \Delta t_{\mathrm{cl}}$), the corresponding time integration of the barotropic mode is carried out from step n to a step sometime beyond step $n+2$ with the barotropic time interval Δt_{tr} using the vertically integrated forcings (X, Y) at $t = t_n$ calculated in the program that solves the baroclinic mode. A weighted average of the barotropic mode over the integration period is used to represent the vertically integrated velocity $\boldsymbol{U}$ at $t = t_{n+1}$. On the other hand, sea level η is predicted from the continuity equation using $\boldsymbol{U}$ at the $n + \frac{1}{2}$ step, instead of the time-weighted average of the barotropic model (See 7.4.1). See Sec. 4.3 for the role of the barotropic model in the overall time integration scheme of MRI.COM.

7.2 Governing equations of the barotropic submodel

As described in Chapter 2, the prognostic variables in the free-surface barotropic model are the surface elevation (η) and the vertically integrated velocity (U and V) given by Eq. (2.66). The prognostic equations are obtained by integrating momentum and continuity equations vertically.

The momentum equations, Eqs. (2.67) and (2.68), are re-written:

$$\frac{\partial U}{\partial t} - fV = -\frac{(\eta + H)}{\rho_0 h_\mu}\frac{\partial (p_a + \rho_0 g \eta)}{\partial \mu} + X, \tag{7.1}$$

$$\frac{\partial V}{\partial t} + fU = -\frac{(\eta + H)}{\rho_0 h_\psi}\frac{\partial (p_a + \rho_0 g \eta)}{\partial \psi} + Y, \tag{7.2}$$

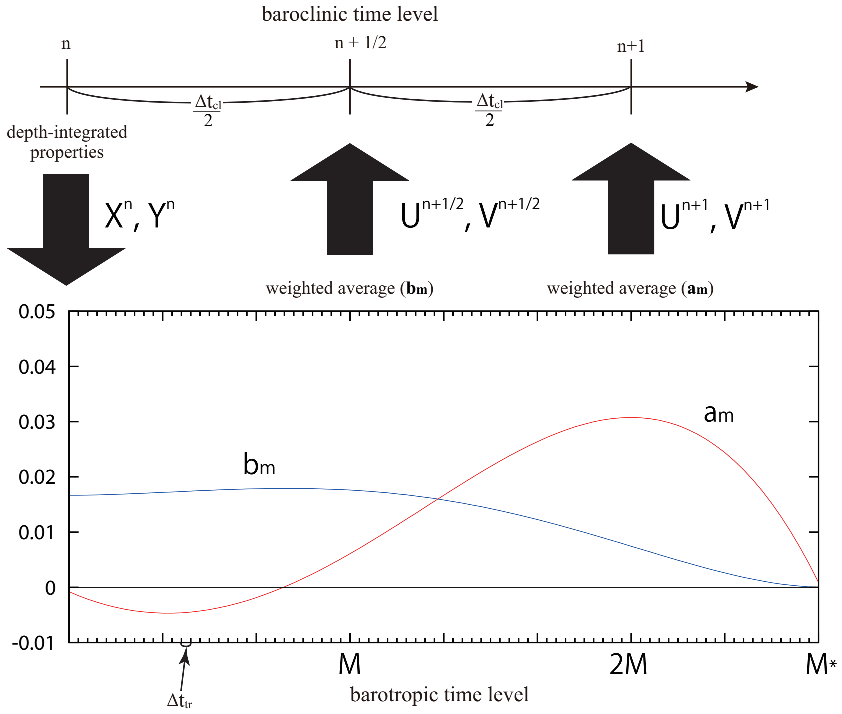

Figure7.1 Schematic figure of the time integration of the barotropic mode and its time-filtering procedure. The weights, a_m and b_m, are for t_{n+1} and $t_{n+\frac{1}{2}}$, respectively.

where

$$
\begin{aligned}
X \equiv \sum_{k=1}^{N} F_\mu = &-\nabla_H \cdot \left(\sum_{k=1}^{N} (\Delta z(u,\ v)u)_{k-\frac{1}{2}} \right) - \sum_{k=1}^{N} \Big[\frac{v}{h_\mu h_\psi} \left(\frac{\partial h_\mu}{\partial \psi} u - \frac{\partial h_\psi}{\partial \mu} v \right) \Big]_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \\
&- \sum_{k=1}^{N} \Big[\frac{1}{\rho_0} \frac{1}{h_\mu} \int_{s_{k-\frac{1}{2}}}^{0} z_s g \rho_\mu ds' \Big] \Delta z_{k-\frac{1}{2}} - \frac{g}{\rho_0 h_\mu} \sum_{k=1}^{N} [\rho' z_\mu] \Delta z_{k-\frac{1}{2}} + \sum_{k=1}^{N} (\Delta z F^{\mu}_{\text{horz}})_{k-\frac{1}{2}} + F^{\mu}_{\text{surf}} \Delta z_{\frac{1}{2}} + F^{\mu}_{\text{bottom}} \Delta z_{N-\frac{1}{2}}
\end{aligned} \tag{7.3}
$$

$$
\begin{aligned}
Y \equiv \sum_{k=1}^{N} F_\psi = &-\nabla_H \cdot \left(\sum_{k=1}^{N} (\Delta z(u,\ v)v)_{k-\frac{1}{2}} \right) + \sum_{k=1}^{N} \Big[\frac{u}{h_\mu h_\psi} \left(\frac{\partial h_\mu}{\partial \psi} u - \frac{\partial h_\psi}{\partial \mu} v \right) \Big]_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \\
&- \sum_{k=1}^{N} \Big[\frac{1}{\rho_0} \frac{1}{h_\psi} \int_{s_{k-\frac{1}{2}}}^{0} z_s g \rho_\psi ds' \Big] \Delta z_{k-\frac{1}{2}} - \frac{g}{\rho_0 h_\psi} \sum_{k=1}^{N} [\rho' z_\psi] \Delta z_{k-\frac{1}{2}} + \sum_{k=1}^{N} (\Delta z F^{\psi}_{\text{horz}})_{k-\frac{1}{2}} + F^{\psi}_{\text{surf}} \Delta z_{\frac{1}{2}} + F^{\psi}_{\text{bottom}} \Delta z_{N-\frac{1}{2}}.
\end{aligned} \tag{7.4}
$$

Note that the surface atmospheric pressure p_a is omitted in the remainder of this chapter. In MRI.COM, the surface atmospheric pressure is not included unless explicitly specified by SLP option.

The continuity equation, Eq. (2.71), is re-written:

$$
\frac{\partial \eta}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (h_\psi U)}{\partial \mu} + \frac{\partial (h_\mu V)}{\partial \psi} \right\} = (P - E + R + I), \tag{7.5}
$$

where P is precipitation (positive downward), E is evaporation (positive upward), R is the river discharge rate (positive into the ocean), and I is the mass exchange with the sea ice model (positive into the ocean).

7.3 Time integration on barotropic time levels

The barotropic model is solved under the governing equations, Eqs. (7.1), (7.2) and (7.5). We adopt the "Euler forward-backward" scheme, which is a stable and economical numerical scheme for linear gravity wave equations without advection

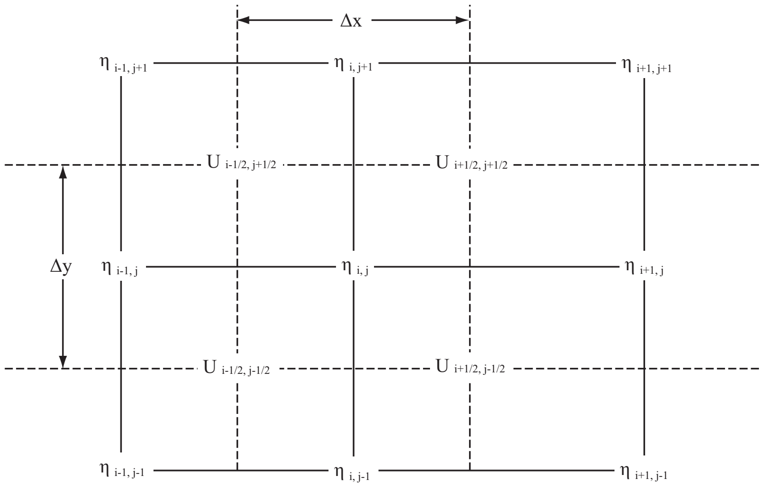

Figure7.2 Grid arrangement of the barotropic model

terms (Mesinger and Arakawa, 1976). This scheme is more stable than the leap-frog scheme. The time step can be doubled for the linear gravity wave equations. In this numerical scheme, either the continuity equation or the momentum equation is calculated first, and then the estimated values are used for calculating the remaining equations. In the procedure of MRI.COM, the surface elevation is first calculated using the continuity equation, Eq. (7.5); the calculated surface elevation is then used to calculate the pressure gradient terms of the momentum equations, Eqs. (7.1) and (7.2). (Killworth et al. (1991) recommended using the Euler backward (Matsuno) scheme for the free surface model except for the tidal problem. The Euler-backward scheme damps higher modes and is more stable. However, the computer burden increases considerably because this scheme calculates the equations twice for one time step. In MRI.COM, stable solutions are efficiently obtained by using the Euler forward-backward scheme because the time filter is applied for the barotropic mode.)

Figure 7.2 illustrates the grid arrangement of the barotropic submodel. The variable η is defined at T-points, and the variables U and V are defined at U-points. Forcing terms (X and Y) in Eqs. (7.1) and (7.2) are calculated in the subroutine for the baroclinic component and defined at U-points.

The finite-difference expression of the continuity equation (Eq. 7.5) is

$$\frac{(\eta'_{i,j}-\eta_{i,j})}{\Delta t_{\mathrm{tr}}} + \frac{1}{(h_\mu h_\psi)_{i,j}}\left[(\delta_\mu \overline{h_\psi U}^{\psi})_{i,j} + (\delta_\psi \overline{h_\mu V}^{\mu})_{i,j}\right] = (P-E+R+I)_{i,j}, \tag{7.6}$$

where the subscripts are labeled on the basis of T-points. The variable $\eta_{i,j}$ is located at T-points, and the variables $U_{i+\frac{1}{2},j+\frac{1}{2}}$ and $V_{i+\frac{1}{2},j+\frac{1}{2}}$ are located at U-points. (They are located at $(i+\frac{1}{2}, j+\frac{1}{2})$ on the basis of T-points; see Figure 3.3). The finite-differencing and averaging operators are defined as follows:

$$\begin{aligned}
\delta_\mu A_i &\equiv \frac{A_{i+\frac{1}{2}}-A_{i-\frac{1}{2}}}{\Delta\mu_i}, & \delta_\mu A_{i+\frac{1}{2}} &\equiv \frac{A_{i+1}-A_i}{\Delta\mu_{i+\frac{1}{2}}}, \\
\overline{A_i}^{\mu} &\equiv \frac{A_{i+\frac{1}{2}}+A_{i-\frac{1}{2}}}{2}, & \overline{A_{i+\frac{1}{2}}}^{\mu} &\equiv \frac{A_{i+1}+A_i}{2}.
\end{aligned} \tag{7.7}$$

The same applies to ψ. In the program codes, the above equation is multiplied by the area of a T-cell ($\Delta S_{T\,i,j} = (\Delta x \Delta y)_{i,j} = (h_\mu \Delta\mu h_\psi \Delta\psi)_{i,j}$):

$$\begin{aligned}
(\eta'_{i,j}-\eta_{i,j})\Delta S_{T\,i,j} = \Delta t_{\mathrm{tr}}\Big\{&(P-E+R+I)_{i,j}\Delta S_{T\,i,j} \\
&-\left(\Delta y_{i+\frac{1}{2},j}\overline{U}^{\psi}_{i+\frac{1}{2},j} - \Delta y_{i-\frac{1}{2},j}\overline{U}^{\psi}_{i-\frac{1}{2},j}\right) - \left(\Delta x_{i,j+\frac{1}{2}}\overline{V}^{\mu}_{i,j+\frac{1}{2}} - \Delta x_{i,j-\frac{1}{2}}\overline{V}^{\mu}_{i,j-\frac{1}{2}}\right)\Big\}.
\end{aligned} \tag{7.8}$$

Averaging operators are defined the same way as the previous ones. This equation is used to obtain the new surface elevation, $\eta'_{i,j}$.

After obtaining $\eta'_{i,j}$, the momentum equations, Eqs. (7.1) and (7.2), are solved. A longer time step can be used when the semi-implicit scheme is applied for the Coriolis terms in the momentum equations. Their finite-difference forms are

$$\frac{(U'_{i+\frac{1}{2},j+\frac{1}{2}} - U_{i+\frac{1}{2},j+\frac{1}{2}})}{\Delta t_{\mathrm{tr}}} - \frac{f(V'_{i+\frac{1}{2},j+\frac{1}{2}} + V_{i+\frac{1}{2},j+\frac{1}{2}})}{2} = -\frac{g(H_{i+\frac{1}{2},j+\frac{1}{2}} + \overline{\overline{\eta'}}^{\mu,\psi}_{i+\frac{1}{2},j+\frac{1}{2}})}{(h_\mu)_{i+\frac{1}{2},j+\frac{1}{2}}}\delta_\mu \overline{\eta'}^{\psi}_{i+\frac{1}{2},j+\frac{1}{2}} + X_{i+\frac{1}{2},j+\frac{1}{2}} \tag{7.9}$$

$$\frac{(V'_{i+\frac{1}{2},j+\frac{1}{2}} - V_{i+\frac{1}{2},j+\frac{1}{2}})}{\Delta t_{\mathrm{tr}}} + \frac{f(U'_{i+\frac{1}{2},j+\frac{1}{2}} + U_{i+\frac{1}{2},j+\frac{1}{2}})}{2} = -\frac{g(H_{i+\frac{1}{2},j+\frac{1}{2}} + \overline{\overline{\eta'}}^{\mu,\psi}_{i+\frac{1}{2},j+\frac{1}{2}})}{(h_\psi)_{i+\frac{1}{2},j+\frac{1}{2}}}\delta_\psi \overline{\eta'}^{\mu}_{i+\frac{1}{2},j+\frac{1}{2}} + Y_{i+\frac{1}{2},j+\frac{1}{2}}. \tag{7.10}$$

Next, we solve these equations for $U'_{i+\frac{1}{2},j+\frac{1}{2}}$ and $V'_{i+\frac{1}{2},j+\frac{1}{2}}$. Let the r.h.s. of the above equations be GX and GY. Multiplying both sides by Δt_{tr}, we have

$$(U'_{i+\frac{1}{2},j+\frac{1}{2}} - U_{i+\frac{1}{2},j+\frac{1}{2}}) - \frac{f\Delta t_{\mathrm{tr}}}{2}(V'_{i+\frac{1}{2},j+\frac{1}{2}} + V_{i+\frac{1}{2},j+\frac{1}{2}}) = \Delta t_{\mathrm{tr}} GX_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{7.11}$$

$$(V'_{i+\frac{1}{2},j+\frac{1}{2}} - V_{i+\frac{1}{2},j+\frac{1}{2}}) + \frac{f\Delta t_{\mathrm{tr}}}{2}(U'_{i+\frac{1}{2},j+\frac{1}{2}} + U_{i+\frac{1}{2},j+\frac{1}{2}}) = \Delta t_{\mathrm{tr}} GY_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{7.12}$$

leading to

$$U'_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{f\Delta t_{\mathrm{tr}}}{2}V'_{i+\frac{1}{2},j+\frac{1}{2}} = U_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{f\Delta t_{\mathrm{tr}}}{2}V_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta t_{\mathrm{tr}} GX_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{7.13}$$

$$V'_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{f\Delta t_{\mathrm{tr}}}{2}U'_{i+\frac{1}{2},j+\frac{1}{2}} = V_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{f\Delta t_{\mathrm{tr}}}{2}U_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta t_{\mathrm{tr}} GY_{i+\frac{1}{2},j+\frac{1}{2}}. \tag{7.14}$$

Letting the r.h.s. of the above equations be RX and RY, we may derive expressions for $U'_{i+\frac{1}{2},j+\frac{1}{2}}$ and $V'_{i+\frac{1}{2},j+\frac{1}{2}}$ as follows:

$$U'_{i+\frac{1}{2},j+\frac{1}{2}} = \left\{RX_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{f\Delta t_{\mathrm{tr}}}{2}RY_{i+\frac{1}{2},j+\frac{1}{2}}\right\} \Big/ \left\{1 + (\frac{f\Delta t_{\mathrm{tr}}}{2})^2\right\}, \tag{7.15}$$

$$V'_{i+\frac{1}{2},j+\frac{1}{2}} = \left\{RY_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{f\Delta t_{\mathrm{tr}}}{2}RX_{i+\frac{1}{2},j+\frac{1}{2}}\right\} \Big/ \left\{1 + (\frac{f\Delta t_{\mathrm{tr}}}{2})^2\right\}. \tag{7.16}$$

In the model, the time evolutions of η, U and V are obtained from $t = t_n$ (the initial value) to the barotropic time level M^* (Fig. 7.1) under the finite difference equations, Eqs. (7.8), (7.15) and (7.16).

7.4 Prognostication of state variables at the baroclinic time level

In the split-explicit method, the state at the next baroclinic time level is obtained by using the solution of the barotropic mode explained in the previous section. To do this, an appropriate time-filtering is necessary. The velocity at the baroclinic time level uses the time-filtered barotropic state as the depth averaged velocity.

7.4.1 Weighted averaging to obtain a barotropic state at the baroclinic time level

The main purpose of solving the barotropic submodel on proceeding from the baroclinic time level of $t = t_n$ is to obtain the barotropic state variables η, U, and V at $t = t_{n+1}$. To achieve this, starting from $t = t_n$, the governing equations of the barotropic submodel are integrated slightly beyond $t = t_{n+1}$ to compute the barotropic state variables as a weighted average of those at barotropic time levels.

First, the barotropic current (U^{n+1}, V^{n+1}) at $t = t_{n+1}$ is calculated by the following weighting average:

$$U^{n+1} = \sum_{m=1}^{M^*} a_m U^m, \quad V^{n+1} = \sum_{m=1}^{M^*} a_m V^m, \tag{7.17}$$

using a time-weighting factor a_m, where U^m and V^m are instantaneous state variables at the barotropic time level $t = t_m$.

On the other hand, the surface height at $t = t_{n+1}$, η^{n+1}, is computed by (7.5) using a central difference scheme as

$$\frac{\eta^{n+1} - \eta^n}{\Delta t_{\mathrm{cl}}} = -\nabla \cdot \boldsymbol{U}^{n+\frac{1}{2}} + (P - E + R + I), \tag{7.18}$$

where $\boldsymbol{U}^{n+\frac{1}{2}}$ at time step $n+\frac{1}{2}$ is obtained by the following weighting average:

$$U^{n+\frac{1}{2}} = \sum_{m=1}^{M^*} b_m U^m, \quad V^{n+\frac{1}{2}} = \sum_{m=1}^{M^*} b_m V^m. \tag{7.19}$$

This is to make the surface height equation be consistent with the continuity equations in a vertical column (Leclair and Madec, 2009). The time weighting factors, a_m and b_m, are given in Sec. 7.5.

7.4.2 Update of 3-D state variables at the baroclinic time level

With the surface elevation η at $t = t_{n+1}$, the height of the vertical column and the volume of the grid cells in the column are determined (see Chapter 3). Once the volume is determined, 3D state variables at $t = t_{n+1}$ are obtained. The velocity field is computed as follows:

- A provisional baroclinic velocity field (u'^{n+1}, v'^{n+1}) is computed using the tendency terms computed by the baroclinic module.
- The provisional velocity (u'^{n+1}, v'^{n+1}) is integrated over the whole column and the vertical mean of the provisional velocity is computed.
- The vertical mean velocity is subtracted from the provisional velocity first and the actual mean velocity based on (U^{n+1}, V^{n+1}) is added instead in order to yield the final velocity field (u^{n+1}, v^{n+1}).

As explained in Sec. 7.7, these operations correspond to the expression for the velocity at the $(n+1)$th time step in the tendency term (for example, $u'_{k-1/2} - \overline{u'}^z + \langle u \rangle^{n+1}$ of Eqs. (7.55) and (7.56)).

Because Eq. (7.18) is consistent with the continuity equation for the T-cell, computation of tendency of tracer due to advection is both conservative and constancy preserving. This method guarantees that tracer is conserved both globally and locally.

Note that body forcing for the uppermost layer such as wind-forcing and restoration of temperature and salinity to the prescribed values may act differently for grid cells with different width. The body force for the uppermost layer becomes

$$\left(\frac{\partial u}{\partial t}, \frac{\partial v}{\partial t}\right)\bigg|_{k=\frac{1}{2}} = \ldots + \frac{1}{\rho_0}\frac{(\tau_\mu, \tau_\psi)}{\Delta z_{\frac{1}{2}}}, \tag{7.20}$$

where (τ_μ, τ_ψ) are the wind stress at the surface (momentum flux), and $\Delta z_{\frac{1}{2}}$ is the variable thickness of the uppermost layer. Thus, the uppermost layer is more accelerated when this layer is thinner than the standard value.

When the restoring condition is applied at the surface, the corresponding temperature and salinity fluxes are

$$F_z^\theta = -\frac{1}{\gamma_\theta}(\theta - \theta^*)\Delta z_{\frac{1}{2}}, \qquad \frac{\partial \theta}{\partial t}\bigg|_{k=\frac{1}{2}} = \ldots + \frac{F_z^\theta}{\Delta z_{\frac{1}{2}}}, \tag{7.21}$$

$$F_z^S = -\frac{1}{\gamma_s}(S - S^*)\Delta z_{\frac{1}{2}}, \qquad \frac{\partial S}{\partial t}\bigg|_{k=\frac{1}{2}} = \ldots + \frac{F_z^S}{\Delta z_{\frac{1}{2}}}. \tag{7.22}$$

Thus, the temperature and salinity are more strongly restored to the prescribed values when this layer is thinner than the standard value.

7.5 Time-average weighting function

From MRI.COM version 4, we use a weighted average for the baroclinic time level, as explained by Shchepetkin and McWilliams (2005). Derivation of the weighting functions, a_m and b_m, for $\boldsymbol{U}$ in Eqs. (7.17) and (7.19), is briefly summarized in this section.

First, a weighting shape function to calculate variables at $t = t_{n+1}$ is denoted with $\{a_m\}$. This must satisfy discrete normalization and centroid conditions,

$$\sum_{m=1}^{M^*} a_m \equiv 1, \quad \sum_{m=1}^{M^*} m a_m \equiv 2M, \tag{7.23}$$

where $2M$ is the ratio between barotropic-baroclinic time step ($2M = \Delta t_{\mathrm{cl}}/\Delta t_{\mathrm{tr}}$), and M^* is the last index at which $a_m > 0$, where $2M \leq M^*$. The weighted averaging is designed so that aliasing between barotropic and baroclinic modes is suppressed as well. With a set of $\{a_m\}$, state variables at the baroclinic time level t_{n+1} is computed by Eq. (7.17) and

$$\eta^{n+1} = \sum_{m=1}^{M^*} a_m \eta^m. \tag{7.24}$$

To be consistent with the continuity equation at baroclinic time levels, the vertically integrated continuity equation must satisfy

$$\frac{\eta^{n+1} - \eta^n}{\Delta t_{\mathrm{cl}}} = -\nabla \cdot \mathbf{U}^{n+\frac{1}{2}}. \tag{7.25}$$

The flow field at $t = t_{n+\frac{1}{2}}$ can be determined accordingly. Assuming that the vertically integrated continuity equation is advanced in time as

$$\frac{\eta^{m+1} - \eta^m}{\Delta t_{\mathrm{tr}}} = -\nabla \cdot \mathbf{U}^{m+\frac{1}{2}}, \tag{7.26}$$

we may obtain the expression for η^m as

$$\eta^m = \eta^0 - \Delta t_{\mathrm{tr}} \sum_{m'=0}^{m-1} \nabla \cdot \mathbf{U}^{m'+\frac{1}{2}}. \tag{7.27}$$

Inserting this into Eq. (7.17) and after some manipulations, which is detailed in Shchepetkin and McWilliams (2005), we have

$$\eta^{n+1} = \eta^0 - \Delta t_{\mathrm{cl}} \nabla \cdot \sum_{m'=1}^{M^*} b_{m'} \mathbf{U}^{m'-\frac{1}{2}}, \tag{7.28}$$

where $\eta^0 = \eta^n$ and

$$b_{m'} = \frac{1}{2M} \sum_{m=m'}^{M^*} a_m. \tag{7.29}$$

By comparing Eqs. (7.25) and (7.28), vertically integrated flow field at $t = t_{n+\frac{1}{2}}$ is obtained as

$$U^{n+\frac{1}{2}} = \sum_{m=1}^{M^*} b_m U^{m-\frac{1}{2}}, \quad V^{n+\frac{1}{2}} = \sum_{m=1}^{M^*} b_m V^{m-\frac{1}{2}}. \tag{7.30}$$

We require that a set of $\{b_m\}$ satisfies

$$\sum_{m=1}^{M^*} b_m \equiv 1, \quad \sum_{m=1}^{M^*} m b_m \equiv M, \tag{7.31}$$

as well as Eq. (7.23).

Specific shape of the weighting function (a_m) for $\tau = m/2M$, $(1 \le m \le M^*)$ is given as follows:

$$A(\tau) = A_0 \left\{ \left(\frac{\tau}{\tau_0}\right)^p \left[1 - \left(\frac{\tau}{\tau_0}\right)^q\right] - r\frac{\tau}{\tau_0} \right\}, \tag{7.32}$$

where $p = 2$, $q = 2$, $r = 0.2346283$, and A_0 and τ_0 are chosen to satisfy normalization conditions (7.23) and (7.31) iteratively. The initial guess for A_0 and τ_0 is given as follows:

$$A_0 = 1, \quad \tau_0 = \frac{(p+2)(p+q+2)}{(p+1)(p+q+1)}. \tag{7.33}$$

Approximate shape of $\{a_m\}$ and $\{b_m\}$ are shown by red and blue lines of Figure 7.1.

7.6 Horizontal diffusivity of sea surface height (option)

The null mode, or checkerboard pattern, would appear in the sea surface height field commonly in the Arakawa B-grid models. For the purpose of suppressing this mode, a weak horizontal diffusion of sea surface height may be included in the vertically integrated continuity equation,

$$\frac{\partial \eta}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (h_\psi U)}{\partial \mu} + \frac{\partial (h_\mu V)}{\partial \psi} \right\} = \mathcal{D}(\eta) + (P - E + R + I) \tag{7.34}$$

where $\mathcal{D}$ is the diffusion operator (Chapter 12.9 in Griffies (2004)). The diffusion operator mixes a sea surface height in each direction of the model coordinates with the harmonic scheme. The specific form of the harmonic-type diffusivity is represented as follows:

$$\mathcal{D}(\eta) = \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial\mu}\left(\frac{h_\psi \kappa_H}{h_\mu}\frac{\partial\eta}{\partial\mu}\right) + \frac{\partial}{\partial\psi}\left(\frac{h_\mu \kappa_H}{h_\psi}\frac{\partial\eta}{\partial\psi}\right)\right\} \tag{7.35}$$

$$= \frac{1}{h_\mu h_\psi}\left\{\frac{\partial h_\psi F^\eta_\mu}{\partial\mu} + \frac{\partial h_\mu F^\eta_\psi}{\partial\psi}\right\} \tag{7.36}$$

where κ_H is the horizontal diffusion coefficients and the diffusive fluxes are represented by

$$\mathbf{F}^\eta = -\left(\frac{\kappa_H}{h_\mu}\frac{\partial\eta}{\partial\mu}, \frac{\kappa_H}{h_\psi}\frac{\partial\eta}{\partial\psi}\right). \tag{7.37}$$

Therefore, continuity equation may be rewritten as follows:

$$\frac{\partial\eta}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial[h_\psi(U+F^\eta_\mu)]}{\partial\mu} + \frac{\partial[h_\mu(V+F^\eta_\psi)]}{\partial\psi}\right\} = (P-E+R+I). \tag{7.38}$$

If the horizontal diffusivity is non-zero for sea surface height, $U^* = U + F^\eta_\mu$ and $V^* = V + F^\eta_\psi$ are treated as the vertically integrated transport velocity, which are divided by the column height to replace the vertical mean velocity based on (U, V) which have been contained in the 3-D velocity field used to advect momentum and tracers. It should be noted that additional restart files are necessary when horizontal diffusion is applied to SSH. This is because $(F^\eta_\mu, F^\eta_\psi)$ are needed to obtain sea surface height for the new time step. MRI.COM provides namelist `nmlrs_ssh_dflx_x` and `nmlrs_ssh_dflx_y` for this purpose.

7.7 The momentum equation modified by the mode splitting

In this appendix, the discretized momentum equation for the 3-D velocity field under the split-explicit method will be derived. This is to clarify the difference from the one for the 3-D velocity under a normal, fully explicit discretizing method. Let us consider reconstructing the flow field of (7.30) from the momentum equation for the barotropic submodel. In the following, we adopted a simple time stepping scheme (forward scheme) and conceptual notations for representing time levels. Because specific expressions will depend upon the choice of time stepping algorithm, the following equations do not necessarily correspond to the time stepping methods used in the model. But this discrepancy is not essential.

At barotropic time levels, momentum equations are advanced in time as follows:

$$U^{m+1} = U^m + \Delta t_{\mathrm{tr}} f V^{m+\frac{1}{2}} - \Delta t_{\mathrm{tr}}\left[\frac{(\eta+H)}{\rho_0 h_\mu}\frac{\partial(p_a+\rho_0 g\eta)}{\partial\mu}\right]^{m+\frac{1}{2}} + \Delta t_{\mathrm{tr}} X, \tag{7.39}$$

$$V^{m+1} = V^m - \Delta t_{\mathrm{tr}} f U^{m+\frac{1}{2}} - \Delta t_{\mathrm{tr}}\left[\frac{(\eta+H)}{\rho_0 h_\psi}\frac{\partial(p_a+\rho_0 g\eta)}{\partial\psi}\right]^{m+\frac{1}{2}} + \Delta t_{\mathrm{tr}} Y, \tag{7.40}$$

where $[\cdot]^{m+\frac{1}{2}}$ is an approximate value of $(\cdot)$ at the time level of $m+\frac{1}{2}$, which is dependent on the choice of the time stepping scheme.

Successive summation of (7.39) and (7.40) over $m' = [0, m-1]$ yields

$$U^m = U^0 + \Delta t_{\mathrm{tr}}\sum_{m'=0}^{m-1} f V^{m'+\frac{1}{2}} - \Delta t_{\mathrm{tr}}\sum_{m'=0}^{m-1}\left[\frac{(\eta+H)}{\rho_0 h_\mu}\frac{\partial(p_a+\rho_0 g\eta)}{\partial\mu}\right]^{m'+\frac{1}{2}} + m\Delta t_{\mathrm{tr}} X, \tag{7.41}$$

$$V^m = V^0 - \Delta t_{\mathrm{tr}}\sum_{m'=0}^{m-1} f U^{m'+\frac{1}{2}} - \Delta t_{\mathrm{tr}}\sum_{m'=0}^{m-1}\left[\frac{(\eta+H)}{\rho_0 h_\psi}\frac{\partial(p_a+\rho_0 g\eta)}{\partial\psi}\right]^{m'+\frac{1}{2}} + m\Delta t_{\mathrm{tr}} Y, \tag{7.42}$$

Applying the time averaging procedure of (7.30) yields

$$\langle U\rangle^{n+1} \equiv \sum_{m=1}^{M^*} b_m U^m \tag{7.43}$$

$$= U^0 + \Delta t_{\mathrm{tr}} \sum_{m=1}^{M^*} \left[b_m \sum_{m'=1}^{m} f V^{m'-\frac{1}{2}} \right] - \Delta t_{\mathrm{tr}} \sum_{m=1}^{M^*} \left\{ b_m \sum_{m'=1}^{m} \left[\frac{(\eta + H)}{\rho_0 h_\mu} \frac{\partial(p_a + \rho_0 g \eta)}{\partial \mu} \right]^{m'-\frac{1}{2}} \right\} + \sum_{m=1}^{M^*} m b_m \Delta t_{\mathrm{tr}} X, \tag{7.44}$$

$$\langle V\rangle^{n+1} \equiv \sum_{m=1}^{M^*} b_m V^m \tag{7.45}$$

$$= V^0 - \Delta t_{\mathrm{tr}} \sum_{m=1}^{M^*} \left[b_m \sum_{m'=1}^{m} f U^{m'-\frac{1}{2}} \right] - \Delta t_{\mathrm{tr}} \sum_{m=1}^{M^*} \left\{ b_m \sum_{m'=1}^{m} \left[\frac{(\eta + H)}{\rho_0 h_\psi} \frac{\partial(p_a + \rho_0 g \eta)}{\partial \psi} \right]^{m'-\frac{1}{2}} \right\} + \sum_{m=1}^{M^*} m b_m \Delta t_{\mathrm{tr}} Y, \tag{7.46}$$

which are rearranged to have a form

$$\langle U\rangle^{n+1} = U^0 + \Delta t_{\mathrm{cl}} f \langle\langle V\rangle\rangle^{n+\frac{1}{2}} - \Delta t_{\mathrm{cl}} \left\langle\left\langle \frac{(\eta + H)}{\rho_0 h_\mu} \frac{\partial(p_a + \rho_0 g \eta)}{\partial \mu} \right\rangle\right\rangle^{n+\frac{1}{2}} + \Delta t_{\mathrm{cl}} X^n, \tag{7.47}$$

$$\langle V\rangle^{n+1} = V^0 - \Delta t_{\mathrm{cl}} f \langle\langle U\rangle\rangle^{n+\frac{1}{2}} - \Delta t_{\mathrm{cl}} \left\langle\left\langle \frac{(\eta + H)}{\rho_0 h_\psi} \frac{\partial(p_a + \rho_0 g \eta)}{\partial \psi} \right\rangle\right\rangle^{n+\frac{1}{2}} + \Delta t_{\mathrm{cl}} Y^n, \tag{7.48}$$

where $\langle\langle\cdot\rangle\rangle^{n+\frac{1}{2}} \equiv \sum_{m=1}^{M^*} b_m \sum_{m'=1}^{m} (\cdot)^{m'-\frac{1}{2}}$.

On the other hand, the baroclinic momentum equation where surface pressure gradient term is dropped is

$$\frac{u'_{k-\frac{1}{2}} \Delta z^{n+1}_{k-\frac{1}{2}} - u^n_{k-\frac{1}{2}} \Delta z^n_{k-\frac{1}{2}}}{\Delta t_{\mathrm{cl}}} = f[v_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + F^n_\mu, \tag{7.49}$$

$$\frac{v'_{k-\frac{1}{2}} \Delta z^{n+1}_{k-\frac{1}{2}} - v^n_{k-\frac{1}{2}} \Delta z^n_{k-\frac{1}{2}}}{\Delta t_{\mathrm{cl}}} = -f[u_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + F^n_\psi. \tag{7.50}$$

This is vertically summed up to give

$$\frac{\sum_{k=1}^{N}(u'_{k-\frac{1}{2}} \Delta z^{n+1}_{k-\frac{1}{2}}) - \sum_{k=1}^{N}(u^n_{k-\frac{1}{2}} \Delta z^n_{k-\frac{1}{2}})}{\Delta t_{\mathrm{cl}}} = f \sum_{k=1}^{N} [v_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + X^n, \tag{7.51}$$

$$\frac{\sum_{k=1}^{N}(v'_{k-\frac{1}{2}} \Delta z^{n+1}_{k-\frac{1}{2}}) - \sum_{k=1}^{N}(v^n_{k-\frac{1}{2}} \Delta z^n_{k-\frac{1}{2}})}{\Delta t_{\mathrm{cl}}} = -f \sum_{k=1}^{N} [u_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + Y^n. \tag{7.52}$$

X^n and Y^n are removed from (7.47), (7.48), (7.51), and (7.52) to give

$$\frac{\sum_{k=1}^{N}(\langle u\rangle^{n+1} \Delta z^{n+1}_{k-\frac{1}{2}}) - \sum_{k=1}^{N}(u'_{k-\frac{1}{2}} \Delta z^{n+1}_{k-\frac{1}{2}})}{\Delta t_{\mathrm{cl}}} = f \langle\langle V\rangle\rangle^{n+\frac{1}{2}} - f \sum_{k=1}^{N} [v_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} - \left\langle\left\langle \frac{(\eta + H)}{\rho_0 h_\mu} \frac{\partial(p_a + \rho_0 g \eta)}{\partial \mu} \right\rangle\right\rangle^{n+\frac{1}{2}} \tag{7.53}$$

$$\frac{\sum_{k=1}^{N}(\langle v\rangle^{n+1} \Delta z^{n+1}_{k-\frac{1}{2}}) - \sum_{k=1}^{N}(v'_{k-\frac{1}{2}} \Delta z^{n+1}_{k-\frac{1}{2}})}{\Delta t_{\mathrm{cl}}} = -f \langle\langle U\rangle\rangle^{n+\frac{1}{2}} + f \sum_{k=1}^{N} [u_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} - \left\langle\left\langle \frac{(\eta + H)}{\rho_0 h_\psi} \frac{\partial(p_a + \rho_0 g \eta)}{\partial \psi} \right\rangle\right\rangle^{n+\frac{1}{2}}. \tag{7.54}$$

This is combined with (7.49) and (7.50) to give

$$
\begin{aligned}
\frac{\left(u'_{k-\frac{1}{2}} - \overline{u'}^{z} + \langle u \rangle^{n+1}\right) \Delta z^{n+1}_{k-\frac{1}{2}} - u^{n}_{k-\frac{1}{2}} \Delta z^{n}_{k-\frac{1}{2}}}{\Delta t_{\mathrm{cl}}} =& f \left[v_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \right]^{n+\frac{1}{2}} - f \overline{[v]^{n+\frac{1}{2}}}^{z} \Delta z^{n+1}_{k-\frac{1}{2}} + f \langle\langle v \rangle\rangle^{n+\frac{1}{2}} \Delta z^{n+1}_{k-\frac{1}{2}} \\
& - \frac{\Delta z^{n+1}_{k-\frac{1}{2}}}{\eta^{n+1} + H} \left\langle\left\langle \frac{(\eta + H)}{\rho_0 h_\mu} \frac{\partial (p_a + \rho_0 g \eta)}{\partial \mu} \right\rangle\right\rangle^{n+\frac{1}{2}} + F^{n}_{\mu} \qquad (7.55) \\
\frac{\left(v'_{k-\frac{1}{2}} - \overline{v'}^{z} + \langle v \rangle^{n+1}\right) \Delta z^{n+1}_{k-\frac{1}{2}} - v^{n}_{k-\frac{1}{2}} \Delta z^{n}_{k-\frac{1}{2}}}{\Delta t_{\mathrm{cl}}} =& - f \left[u_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \right]^{n+\frac{1}{2}} + f \overline{[u]^{n+\frac{1}{2}}}^{z} \Delta z^{n+1}_{k-\frac{1}{2}} - f \langle\langle u \rangle\rangle^{n+\frac{1}{2}} \Delta z^{n+1}_{k-\frac{1}{2}} \\
& - \frac{\Delta z^{n+1}_{k-\frac{1}{2}}}{\eta^{n+1} + H} \left\langle\left\langle \frac{(\eta + H)}{\rho_0 h_\psi} \frac{\partial (p_a + \rho_0 g \eta)}{\partial \psi} \right\rangle\right\rangle^{n+\frac{1}{2}} + F^{n}_{\psi}, \qquad (7.56)
\end{aligned}
$$

where $\overline{(...)}^{z}$ denotes the thickness weighted vertical average. This is the momentum balance for the total velocity field under the mode-splitting scheme. In comparison with (7.49) and (7.50), the correction terms appear in the tendency and Coriolis terms owing to the use of the split-explicit method. In the tendency term on the l.h.s., the vertically averaged velocity $(\overline{u'}^{z}, \overline{v'}^{z})$ is replaced by the prediction of the barotropic model $(\langle u \rangle^{n+1}, \langle v \rangle^{n+1})$. Likewise on the r.h.s., the Coriolis term takes a form in which vertical average for $[\mathbf{u}_{k+\frac{1}{2}}]^{n+\frac{1}{2}}$ is replaced by $\langle\langle \mathbf{u} \rangle\rangle^{n+\frac{1}{2}}$.

7.8 Usage

Behavior of the barotropic model at run time is specified by the three namelist blocks shown on Tables 7.1 through 7.3. For the initial condition of the model, restart files must be prepared for five variables: sea surface height, X- and Y-ward barotropic transports, and X- and Y-ward transports due to SSH diffusion.

In addition, following model options are available.

SLP: Sea surface is elevated/depressed according to surface atmospheric pressure
Give atmospheric sea-level pressure, p_a in (7.1) and (7.2), to the model by namelist `nml_force_data` (See Table 14.4).

FSVISC: Explicit viscosity is added to the barotropic momentum equation
Specify the horizontal viscosity coefficient by namelist `nml_barotropic_visc_horz`. See Section 7.3 for the time discretization method.

See `docs/README.Namelist` for namelist details.

Table7.1 Namelist `nml_barotropic_model` for the specifying basic features of the barotropic model (see Section 7.8)

variable	units	description	usage
dt_barotropic_sec	sec	time step interval	required

Table7.2 Namelist `nml_barotropic_run` for starting the barotropic model (see Section 7.8)

variable	units	description	usage
l_rst_barotropic_in	logical	read initial restart for sea surface and transport or not	default = l_rst_in
l_rst_barotropic_dflx_in	logical	read initial restart for diffusive flux of SSH or not (Section 7.6)	default = l_rst_barotropic_in

Table7.3 Namelist `nml_barotropic_diff` for specifying SSH diffusion (see Section 7.8)

variable	units	description	usage
ssh_diff_cm2ps	$\mathrm{cm}^2\ \mathrm{s}^{-1}$	diffusivity for SSH (Section 7.6)	default = zero

Chapter 8

Equations of motion (baroclinic component)

This chapter explains how to solve the baroclinic momentum equations (2.76) and (2.77), which are re-written:

$$\begin{aligned}
&\frac{\partial(z_s u)}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi u u)}{\partial \mu}+\frac{\partial(z_s h_\mu v u)}{\partial \psi}\right\}+\frac{\partial(z_s \dot{s} u)}{\partial s}+z_s\frac{v}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial \psi}u-\frac{\partial h_\psi}{\partial \mu}v\right)-z_s f v \\
&\quad=-z_s\frac{1}{\rho_0 h_\mu}\frac{\partial}{\partial \mu}\left(g\int_{z(z^*)}^{\eta}\rho' dz\right)-z_s\frac{g\rho'}{\rho_0 h_\mu}\frac{\partial z}{\partial \mu} \\
&\quad\quad+z_s\frac{1}{\rho_0}(\nabla\cdot\tau_{\text{horizontal strain}})_u+z_s\frac{1}{z_s}\frac{\partial}{\partial s}\left(\frac{\nu_V}{z_s}\frac{\partial u}{\partial s}\right),
\end{aligned} \tag{8.1}$$

$$\begin{aligned}
&\frac{\partial(z_s v)}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi u v)}{\partial \mu}+\frac{\partial(z_s h_\mu v v)}{\partial \psi}\right\}+\frac{\partial(z_s \dot{s} v)}{\partial s}+z_s\frac{u}{h_\mu h_\psi}\left(\frac{\partial h_\psi}{\partial \mu}v-\frac{\partial h_\mu}{\partial \psi}u\right)+z_s f u \\
&\quad=-z_s\frac{1}{\rho_0 h_\psi}\frac{\partial}{\partial \psi}\left(g\int_{z(z^*)}^{\eta}\rho' dz\right)-z_s\frac{g\rho'}{\rho_0 h_\psi}\frac{\partial z}{\partial \psi} \\
&\quad\quad+z_s\frac{1}{\rho_0}(\nabla\cdot\tau_{\text{horizontal strain}})_v,+z_s\frac{1}{z_s}\frac{\partial}{\partial s}\left(\frac{\nu_V}{z_s}\frac{\partial v}{\partial s}\right).
\end{aligned} \tag{8.2}$$

The advection terms are explained in Section 8.1, the pressure gradient terms in Section 8.2, the viscosity terms in Section 8.3 and the Coriolis terms in Section 8.4. Finally, Section 8.5 explains usage. The purpose of this chapter is to find the temporary velocity (u', v') shown by Eqs. (2.78) and (2.79). See Section 2.2.2 for the procedure to obtain the absolute velocity (u, v) based on (u', v').

One of the unique characteristics of MRI.COM's momentum advection terms is that there are oblique exchanges of momentum between U-cells that share only a corner. This scheme enables the flow field around and over the bottom topography to be naturally expressed. Furthermore, quasi-enstrophies, $(\partial u/\partial y)^2$ and $(\partial v/\partial x)^2$, for the U-cells away from land are conserved in calculating the momentum advection for horizontally non-divergent flows. The description of momentum advection in Section 8.1 is based on Ishizaki and Motoi (1999).

The discrete expressions for the viscosity terms in the momentum equations are based on generalized orthogonal coordinates. A harmonic operator is used as the default assuming a no-slip condition on the land-sea boundaries. A biharmonic operator (`VISBIHARM` option) and a parameterization of viscosity as a function of deformation rate (`SMAGOR` option) may also be used.

8.1 Advection terms

Chapter 6 demonstrated that the mass fluxes used for calculating momentum advection are identical to those for the mass continuity of the U-cell, and that they are obtained by an averaging operation (6.20) of those for the T-cell mass continuity (6.21). This is the preliminaries for constructing the general mass flux form over an arbitrary bottom and coastal topography. Its vertical part can express diagonally upward mass fluxes over bottom relief and its horizontal part can express horizontally diagonal mass fluxes along coast lines (Ishizaki and Motoi, 1999).

Here we explain how to obtain the mass fluxes used in the momentum advection and how to get the finite difference expression of the advection terms.

The horizontal subscript indices of variables are integers for the T-point (i, j), and therefore, $(i + \frac{1}{2}, j + \frac{1}{2})$ for the U-point. In the vertical direction, integer k is used for the level of the vertical mass fluxes and the level for the T- and U-points a half vertical grid size lower is expressed by $k + \frac{1}{2}$ (Figure 3.3(a)).

8.1.1 Vertical mass fluxes and its momentum advection

Following (6.20), the vertical mass flux at the upper surface, level k, of the U-cell $(i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$, $\overline{W}^U_{i+\frac{1}{2},j+\frac{1}{2},k}$, is defined by surrounding W^T as

$$\overline{W}^U_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{W^T_{i,j,k}}{N_{i,j,k+\frac{1}{2}}} + \frac{W^T_{i+1,j,k}}{N_{i+1,j,k+\frac{1}{2}}} + \frac{W^T_{i,j+1,k}}{N_{i,j+1,k+\frac{1}{2}}} + \frac{W^T_{i+1,j+1,k}}{N_{i+1,j+1,k+\frac{1}{2}}}, \tag{8.3}$$

where $N_{i,j,k+\frac{1}{2}}$ is the number of sea grid cells around the T-point $T_{i,j}$ in layer $k+\frac{1}{2}$.

On the other hand, the vertical mass flux at the bottom surface, level k, of the U-cell $(i+\frac{1}{2}, j+\frac{1}{2}, k-\frac{1}{2})$, $\mathcal{W}^U_{i+\frac{1}{2},j+\frac{1}{2},k}$ is defined as

$$\mathcal{W}^U_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{W^T_{i,j,k}}{N_{i,j,k-\frac{1}{2}}} + \frac{W^T_{i+1,j,k}}{N_{i+1,j,k-\frac{1}{2}}} + \frac{W^T_{i,j+1,k}}{N_{i,j+1,k-\frac{1}{2}}} + \frac{W^T_{i+1,j+1,k}}{N_{i+1,j+1,k-\frac{1}{2}}}. \tag{8.4}$$

Though W^T are continuous at the boundary of vertically adjacent T-cells, $\overline{W}^U_{i+\frac{1}{2},j+\frac{1}{2},k}$ and $\mathcal{W}^U_{i+\frac{1}{2},j+\frac{1}{2},k}$ seem to be discontinuous at the boundary when N are vertically different, for example, $N_{i,j,k+\frac{1}{2}} < N_{i,j,k-\frac{1}{2}}$, over the bottom relief. However, this apparent discrepancy can be consistently interpreted by introducing diagonally upward or downward mass fluxes as shown below.

a. One-dimensional variation of bottom relief

We first consider a case in which the bottom depth varies in one direction like a staircase and a barotropic current flows from the left to the right over the topography. Figure 8.1(a) indicates the mass continuity for T-cells to conserve barotropy in shallow regions (the U-points are just intermediate between T-points) . Figure 8.1(b) depicts the mass continuity for U-cells, derived from those for adjacent T-cells. Except for fluxes just on the bottom slope, each flux is obtained as a mean value of neighboring fluxes for T-cells. Just on the bottom slope, we must introduce a flux that flows along the slope to ensure mass continuity. The lowermost U-cells at the slope have nonzero vertical flux at the bottom. The barotropy of the flow and the distribution of vertical velocity are thereby kept reasonable for U-cell fluxes.

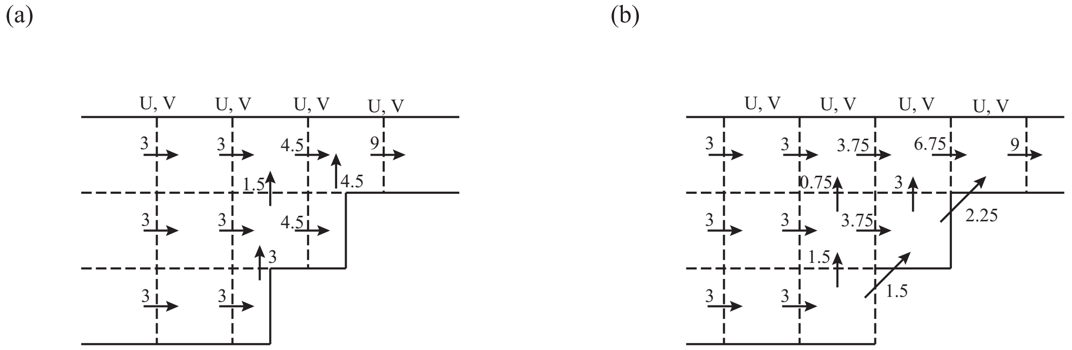

Figure8.1 (a) Two-dimensional mass fluxes for T-cells on a stair-like topography. (b) Two-dimensional mass fluxes for U-cells on the same topography.

b. Two-dimensional variation of bottom depth

The diagonally upward or downward mass fluxes introduced in the previous simple case are generalized for flows over bottom topography that varies two-dimensionally. For simplicity, we consider a two-layer case without losing generality. First, we consider three examples of bottom relief, and then generalize the results.

■Example 1 Consider a case in which all cells are sea cells in the upper and lower layers except for cell **d** in the lower layer (cell $\mathbf{d}_l$) (Figure 8.2). We use suffixes l and u to designate the lower and the upper layer. The central T-point and T-cell are represented by A. The vertical mass flux W^T should be continuous at the interface between cells A_l and A_u, though the area of cell A_l (3/4 measured in grid area units) differs from that of A_u (1 unit). Let us consider how this T-cell mass flux W^T should be distributed to the mass flux W^U of neighboring cells represented by **a**, **b**, **c**, and **d**. In the lower layer, W_T is shared by three cells, $\mathbf{a}_l$, $\mathbf{b}_l$, and $\mathbf{c}_l$, so the contribution of W^T to each W^U is $W^T/3$, but in the upper layer, it is $W^T/4$ because part of W_T should also be shared by W^U at cell $\mathbf{d}_u$. Here W^U at the bottom of cell $\mathbf{d}_u$ is no longer zero. Therefore, $W^T/4$ of the $W^T/3$ shared by each of the three lower sea grid cells $\mathbf{a}_l$, $\mathbf{b}_l$, and $\mathbf{c}_l$ is purely vertical, and the remaining $W^T/12$ (= $W^T/3 - W^T/4$) flows to cell $\mathbf{d}_u$ through the interface. Gathering these diagonal fluxes from the lower three cells, the total amount entering cell $\mathbf{d}_u$ is certainly $W^T/4$ (=$W^T/12 \times 3$). The advected momentum value should be the mean of those at the starting and ending cells of the flux, if the centered difference scheme is used, which is necessary to conserve the total kinetic energy.

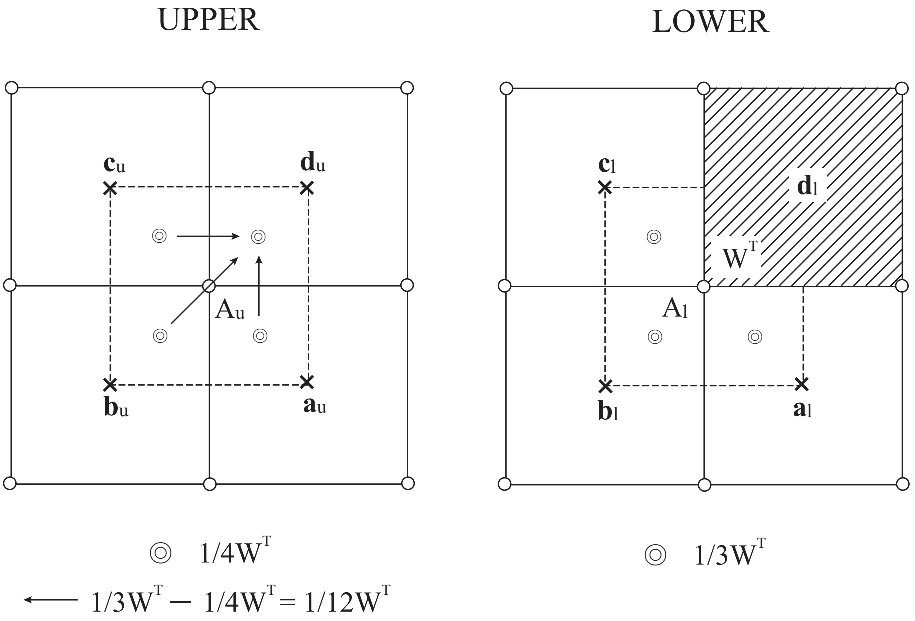

Figure8.2 First example of land-sea patterns, in which all four upper cells are sea cells, with three sea cells and one land cell in the lower layer.

■Example 2 Next, consider an example in which only $\mathbf{b}_l$ is a sea cell in the lower layer, and all four cells are sea cells in the upper layer (Figure 8.3). In the lower layer, W^T is shared only by $\mathbf{b}_l$ but in the upper layer, it is shared by all four cells. Therefore, $W^T/4$ of W^T at cell $\mathbf{b}_l$ is carried vertically upward and the remaining $3W^T/4$ is distributed to the other three cells in the upper layer ($\mathbf{a}_u$, $\mathbf{c}_u$, and $\mathbf{d}_u$), each receiving $W^T/4$.

■Example 3 A third example holds that the upper layer also has land area. In this example, cells $\mathbf{c}_l$, $\mathbf{d}_l$, and $\mathbf{d}_u$ are land cells and the others are sea cells (Figure 8.4). In the lower layer, W^T is shared by two cells ($\mathbf{a}_l$ and $\mathbf{b}_l$) while it is shared by three cells ($\mathbf{a}_u$, $\mathbf{b}_u$, and $\mathbf{c}_u$) in the upper layer. Therefore, from each of $\mathbf{a}_l$ and $\mathbf{b}_l$, $W^T/3$ of $W^T/2$ goes vertically upward and the remaining $W^T/6$ (= $W^T/2 - W^T/3$) goes diagonally upward to cell $\mathbf{c}_u$ with a total amount of $W^T/3$ (= $W^T/6 \times 2$).

c. Generalization

The relationship between the land-sea distribution and the vertically and diagonally upward fluxes stated above is generalized for an arbitrary land-sea distribution. Assume cell $\mathbf{d}_l$ is a land but cell $\mathbf{d}_u$ is a sea cell and consider the diagonally upward fluxes coming to cell $\mathbf{d}_u$. We take N_l as the number of sea cells around point A in the lower layer and N_u as the number in the upper layer ($1 \leq N_l < N_u \leq 4$). Each cell in the lower layer carries W^T/N_l, and W^T/N_u of it goes vertically upward. The remaining

$$W^T/N_l - W^T/N_u = W^T(N_u - N_l)/(N_l N_u) \tag{8.5}$$

should be distributed as diagonally upward fluxes to sea cells in the upper layer at which the lower layer is land. The number of such upper sea cells is $N_u - N_l$ including cell $\mathbf{d}_u$. Thus, each diagonally upward flux coming to cell $\mathbf{d}_u$ is

$$W^T(N_u - N_l)/(N_l N_u) \times 1/(N_u - N_l) = W^T/(N_l N_u). \tag{8.6}$$

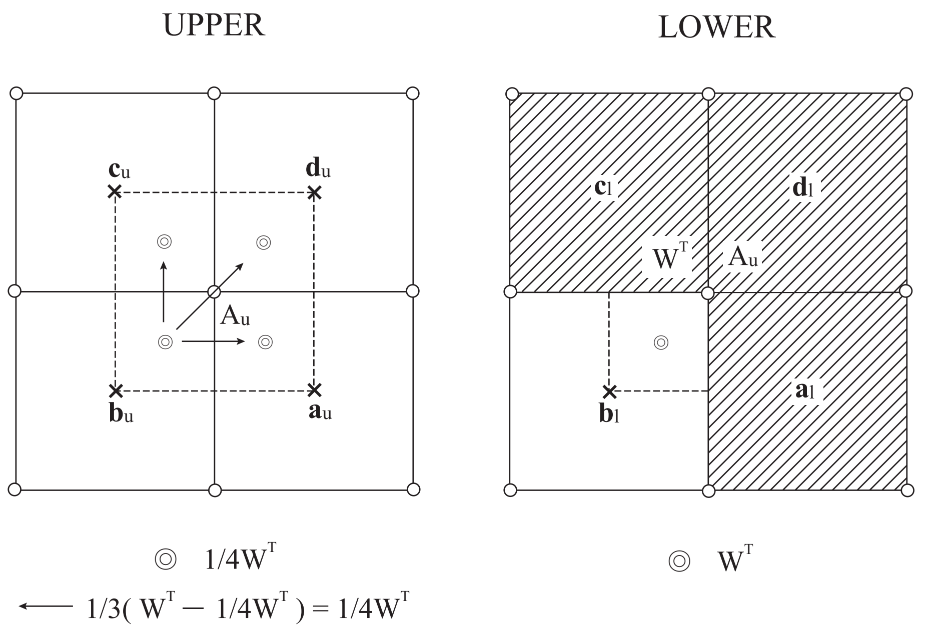

Figure8.3 Second example of land-sea patterns, in which all four upper cells are sea cells, with one sea cell and three land cells in the lower layer.

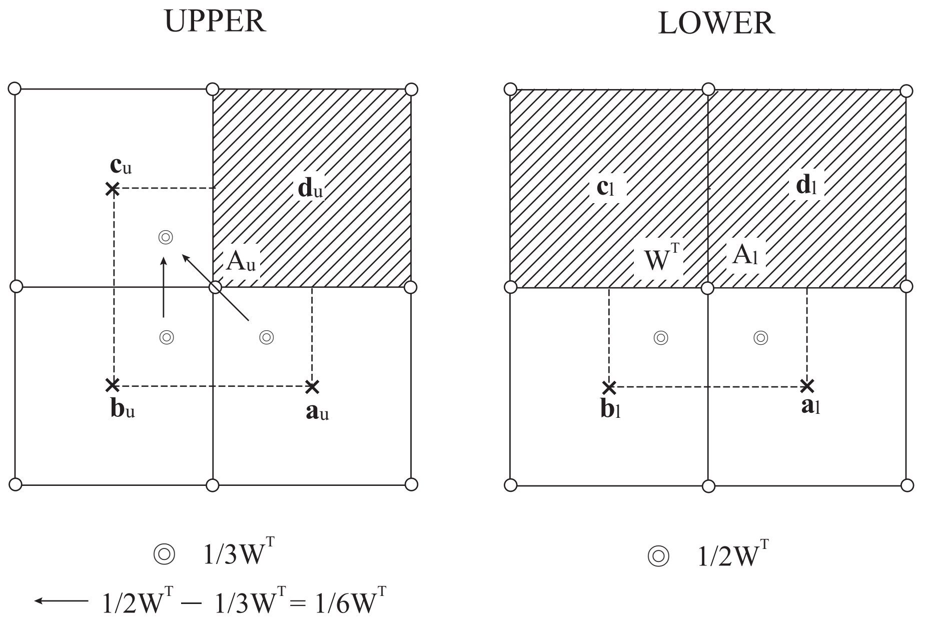

Figure8.4 Third example of land-sea patterns, in which one of the upper cells is a land cell, with two land and two sea cells in the lower layer.

The number of such fluxes coming to the cell $\mathbf{d}_u$ is N_l, so their total is

$$W^T/(N_l N_u) \times N_l = W^T/N_u. \tag{8.7}$$

Based on these discussions we understand the difference between (8.3) and (8.4).

We regard the name of each cell such as $\mathbf{a}_l$ also as a land-sea index. If we assume that $\mathbf{a}_l = 1(0)$ when cell $\mathbf{a}_l$ is a sea (land) cell, then the diagonally upward mass flux and momentum flux coming from cell $\mathbf{a}_l$ to cell $\mathbf{d}_u$ are

$$\mathbf{a}_l W^T/(N_l N_u) \quad \text{and} \quad \mathbf{a}_l W^T(u_{\mathrm{a}_l} + u_{\mathrm{d}_u})/(2N_l N_u), \tag{8.8}$$

where u_{a_l} and u_{d_u} are the velocity at cells $\mathbf{a}_l$ and $\mathbf{d}_u$, respectively. Purely vertical mass flux and momentum flux from cell

$\mathbf{a}_l$ to cell $\mathbf{a}_u$ are expressed as

$$\mathbf{a}_l W^T/N_u \quad \text{and} \quad \mathbf{a}_l W^T(u_{\mathrm{a}_l} + u_{\mathrm{a}_u})/(2N_u), \tag{8.9}$$

respectively, where u_{a_u} is the velocity at cell $\mathbf{a}_u$. Similar formulations apply to cells $\mathbf{b}_l$ and $\mathbf{c}_l$.

Mass and momentum fluxes for W^T at other T-points around cell $\mathbf{d}_u$ should be calculated similarly to complete vertically and diagonally upward momentum advections around cell $\mathbf{d}_u$. When $N_u = N_l$, diagonally upward fluxes need not be considered and only vertical fluxes (8.9) apply.

To summarize, the discrete expression for the vertical flux of zonal momentum that is transported into a U-cell at $(i+\frac{1}{2}, j+\frac{1}{2}, k-\frac{1}{2})$ through its bottom, $F_{\mathbf{L}_{i+\frac{1}{2},j+\frac{1}{2},k}}(u)$, is given as follows:

$$\begin{aligned}
F_{\mathbf{L}_{i+\frac{1}{2},j+\frac{1}{2},k}}(u) =& \frac{1}{2}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right) \times \left[\frac{W^T_{i,j,k}}{N_{i,j,k-\frac{1}{2}}} + \frac{W^T_{i+1,j,k}}{N_{i+1,j,k-\frac{1}{2}}} + \frac{W^T_{i,j+1,k}}{N_{i,j+1,k-\frac{1}{2}}} + \frac{W^T_{i+1,j+1,k}}{N_{i+1,j+1,k-\frac{1}{2}}}\right] \\
&+ \frac{e_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\left(1 - e_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right)}{2} \times \\
&\Big[\Big\{e_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}\right) + e_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}\right) \\
&+ e_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right)\Big\} \times \frac{W^T_{i,j,k}}{N_{i,j,k+\frac{1}{2}} N_{i,j,k-\frac{1}{2}}} \\
&+ \Big\{e_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}\right) + e_{i+\frac{3}{2},j-\frac{1}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{3}{2},j-\frac{1}{2},k+\frac{1}{2}}\right) \\
&+ e_{i+\frac{3}{2},j+\frac{1}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{3}{2},j+\frac{1}{2},k+\frac{1}{2}}\right)\Big\} \times \frac{W^T_{i+1,j,k}}{N_{i+1,j,k+\frac{1}{2}} N_{i+1,j,k-\frac{1}{2}}} \\
&+ \Big\{e_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right) + e_{i-\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i-\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}}\right) \\
&+ e_{i+\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}}\right)\Big\} \times \frac{W^T_{i,j+1,k}}{N_{i,j+1,k+\frac{1}{2}} N_{i,j+1,k-\frac{1}{2}}} \\
&+ \Big\{e_{i+\frac{3}{2},j+\frac{1}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{3}{2},j+\frac{1}{2},k+\frac{1}{2}}\right) + e_{i+\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}}\right) \\
&+ e_{i+\frac{3}{2},j+\frac{3}{2},k+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{3}{2},j+\frac{3}{2},k+\frac{1}{2}}\right)\Big\} \times \frac{W^T_{i+1,j+1,k}}{N_{i+1,j+1,k+\frac{1}{2}} N_{i+1,j+1,k-\frac{1}{2}}}\Big],
\end{aligned} \tag{8.10}$$

where $e_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}$ is the land-sea index for a U-cell (unity for sea and zero for land). Note that diagonally upward/downward momentum fluxes may only occur for a U-cell with the sea floor underneath. The momentum fluxes entering and leaving the cells at the lower level ($k = k + \frac{1}{2}$) are individually added or removed from the cell when calculating them.

8.1.2 Horizontal mass flux and its momentum advection

a. Horizontal mass fluxes

We next consider the generalization of the U-cell horizontal mass fluxes for arbitrary coast lines, in order to derive the horizontal momentum advection. To do this, we start with the generalization of the T-cell mass continuity (6.21). Assuming that $e_{i+\frac{1}{2},j+\frac{1}{2}}$ is a land-sea index (unity for sea and zero for land) for U-cell $(i+\frac{1}{2}, j+\frac{1}{2})$, the general formulae for $U^{*T}_{i+\frac{1}{2},j}$ and $V^{*T}_{i,j+\frac{1}{2}}$ are given as

$$\begin{aligned}
U^{*T}_{i+\frac{1}{2},j} =& \frac{1}{2}(e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u^*_{i+\frac{1}{2},j} \Delta y_{i+\frac{1}{2},j} \Delta z \\
V^{*T}_{i,j+\frac{1}{2}} =& \frac{1}{2}(e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) v^*_{i,j+\frac{1}{2}} \Delta x_{i,j+\frac{1}{2}} \Delta z,
\end{aligned} \tag{8.11}$$

where, $u^*_{i+\frac{1}{2},j}$ and $v^*_{i,j+\frac{1}{2}}$ are the zonal and meridional velocity at the eastern and northern side boundary of a T-cell (i, j) (Eq. 6.26). However, if there are any land cells, only the values at the sea cells are used as in the calculation of Eq. (6.10). Here we neglect the vertical subscript $(k - \frac{1}{2})$.

Substituting these formulae into the T-cell mass continuity (6.21), the X (zonal) component of the mass continuity for

U-cell $(i+\frac{1}{2}, j+\frac{1}{2})$ (6.20), $\mathrm{XMC}^{U}_{i+\frac{1}{2},j+\frac{1}{2}}$, multiplied by its own land-sea signature $e_{i+\frac{1}{2},j+\frac{1}{2}}$, is expressed as

$$\begin{aligned}
\mathrm{XMC}^{U}_{i+\frac{1}{2},j+\frac{1}{2}} = & e_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \Delta z \\
& \times \left\{ \frac{1}{N_{i,j}} [(e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}}) u^{*}_{i-\frac{1}{2},j} - (e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u^{*}_{i+\frac{1}{2},j}] \right. \\
& + \frac{1}{N_{i+1,j}} [(e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u^{*}_{i+\frac{1}{2},j} - (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}}) u^{*}_{i+\frac{3}{2},j}] \\
& + \frac{1}{N_{i,j+1}} [(e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}}) u^{*}_{i-\frac{1}{2},j+1} - (e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) u^{*}_{i+\frac{1}{2},j+1}] \\
& \left. + \frac{1}{N_{i+1,j+1}} [(e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) u^{*}_{i+\frac{1}{2},j+1} - (e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}}) u^{*}_{i+\frac{3}{2},j+1}] \right\} \\
= & e_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \Delta z \times \left\{ \left[\frac{1}{N_{i,j}} (e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}}) u^{*}_{i-\frac{1}{2},j} \right. \right. \\
& + \left(-\frac{1}{N_{i,j}} + \frac{1}{N_{i+1,j}} \right) (e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u^{*}_{i+\frac{1}{2},j} \\
& \left. - \frac{1}{N_{i+1,j}} (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}}) u^{*}_{i+\frac{3}{2},j} \right] \\
& + \left[\frac{1}{N_{i,j+1}} (e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}}) u^{*}_{i-\frac{1}{2},j+1} \right. \\
& + \left(-\frac{1}{N_{i,j+1}} + \frac{1}{N_{i+1,j+1}} \right) (e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) u^{*}_{i+\frac{1}{2},j+1} \\
& \left. \left. - \frac{1}{N_{i+1,j+1}} (e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}}) u^{*}_{i+\frac{3}{2},j+1} \right] \right\}.
\end{aligned} \tag{8.12}$$

Here, recalling

$$N_{i,j} = e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{8.13}$$

we have,

$$\begin{aligned}
& \left(-\frac{1}{N_{i,j}} + \frac{1}{N_{i+1,j}} \right) (e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) \\
& \qquad = \frac{1}{N_{i,j}} (e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}}) - \frac{1}{N_{i+1,j}} (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}})
\end{aligned} \tag{8.14}$$

and

$$\begin{aligned}
& \left(-\frac{1}{N_{i,j+1}} + \frac{1}{N_{i+1,j+1}} \right) (e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) \\
& \qquad = \frac{1}{N_{i,j+1}} (e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}}) - \frac{1}{N_{i+1,j+1}} (e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}}).
\end{aligned} \tag{8.15}$$

Thus, based on (6.24) and (6.25),

$$\begin{aligned}
\mathrm{XMC}^U_{i+\frac{1}{2},j+\frac{1}{2}} &= e_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \Delta z \Big\{ \Big[\frac{1}{N_{i,j}} (e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}})(u^*_{i-\frac{1}{2},j} + u^*_{i+\frac{1}{2},j}) \\
&\quad - \frac{1}{N_{i+1,j}} (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}})(u^*_{i+\frac{1}{2},j} + u^*_{i+\frac{3}{2},j}) \Big] \\
&\quad + \Big[\frac{1}{N_{i,j+1}} (e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}})(u^*_{i-\frac{1}{2},j+1} + u^*_{i+\frac{1}{2},j+1}) \\
&\quad - \frac{1}{N_{i+1,j+1}} (e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}})(u^*_{i+\frac{1}{2},j+1} + u^*_{i+\frac{3}{2},j+1}) \Big] \Big\} \\
&= e_{i+\frac{1}{2},j+\frac{1}{2}} \Big[e_{i-\frac{1}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i,j}} U^U_{i,j} + \frac{1}{N_{i,j+1}} U^U_{i,j+1} \Big) \\
&\quad - e_{i+\frac{3}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i+1,j}} U^U_{i+1,j} + \frac{1}{N_{i+1,j+1}} U^U_{i+1,j+1} \Big) + e_{i-\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} U^U_{i,j} \\
&\quad - e_{i+\frac{3}{2},j+\frac{3}{2}} \frac{1}{N_{i+1,j+1}} U^U_{i+1,j+1} + e_{i-\frac{1}{2},j+\frac{3}{2}} \frac{1}{N_{i,j+1}} U^U_{i,j+1} - e_{i+\frac{3}{2},j-\frac{1}{2}} \frac{1}{N_{i+1,j}} U^U_{i+1,j} \Big].
\end{aligned} \tag{8.16}$$

Adding the Y (meridional) component, YMC (the expression is omitted here), to the above formula, we obtain the horizontal part of the U-cell mass continuity, HMC, as follows:

$$\begin{aligned}
\mathrm{HMC}^U_{i+\frac{1}{2},j+\frac{1}{2}} &= \mathrm{XMC}^U_{i+\frac{1}{2},j+\frac{1}{2}} + \mathrm{YMC}^U_{i+\frac{1}{2},j+\frac{1}{2}} \\
&= e_{i+\frac{1}{2},j+\frac{1}{2}} \\
&\quad \times \Big[e_{i-\frac{1}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i,j}} U^U_{i,j} + \frac{1}{N_{i,j+1}} U^U_{i,j+1} \Big) - e_{i+\frac{3}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i+1,j}} U^U_{i+1,j} + \frac{1}{N_{i+1,j+1}} U^U_{i+1,j+1} \Big) \\
&\quad + e_{i+\frac{1}{2},j-\frac{1}{2}} \Big(\frac{1}{N_{i,j}} V^U_{i,j} + \frac{1}{N_{i+1,j}} V^U_{i+1,j} \Big) - e_{i+\frac{1}{2},j+\frac{3}{2}} \Big(\frac{1}{N_{i,j+1}} V^U_{i,j+1} + \frac{1}{N_{i+1,j+1}} V^U_{i+1,j+1} \Big) \\
&\quad + e_{i-\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} (U^U_{i,j} + V^U_{i,j}) - e_{i+\frac{3}{2},j+\frac{3}{2}} \frac{1}{N_{i+1,j+1}} (U^U_{i+1,j+1} + V^U_{i+1,j+1}) \\
&\quad + e_{i-\frac{1}{2},j+\frac{3}{2}} \frac{1}{N_{i,j+1}} (U^U_{i,j+1} - V^U_{i,j+1}) - e_{i+\frac{3}{2},j-\frac{1}{2}} \frac{1}{N_{i+1,j}} (U^U_{i+1,j} - V^U_{i+1,j}) \Big].
\end{aligned} \tag{8.17}$$

Assuming mass fluxes $M_{\mathbf{E}}, M_{\mathbf{N}}, M_{\mathbf{NE}}, M_{\mathbf{SE}}$ as follows:

$$\begin{aligned}
M_{\mathbf{E}_{i,j+\frac{1}{2}}} &= e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i,j}} U^U_{i,j} + \frac{1}{N_{i,j+1}} U^U_{i,j+1} \Big), \\
M_{\mathbf{N}_{i+\frac{1}{2},j}} &= e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i+\frac{1}{2},j-\frac{1}{2}} \Big(\frac{1}{N_{i,j}} V^U_{i,j} + \frac{1}{N_{i+1,j}} V^U_{i+1,j} \Big), \\
M_{\mathbf{NE}_{i,j}} &= e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} (U^U_{i,j} + V^U_{i,j}), \\
M_{\mathbf{SE}_{i,j}} &= e_{i-\frac{1}{2},j+\frac{1}{2}} e_{i+\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} (U^U_{i,j} - V^U_{i,j}),
\end{aligned} \tag{8.18}$$

then,

$$\begin{aligned}
\mathrm{HMC}^U_{i+\frac{1}{2},j+\frac{1}{2}} =& M_{\mathbf{E}_{i,j+\frac{1}{2}}} - M_{\mathbf{E}_{i+1,j+\frac{1}{2}}} + M_{\mathbf{N}_{i+\frac{1}{2},j}} - M_{\mathbf{N}_{i+\frac{1}{2},j+1}} \\
&+ M_{\mathbf{NE}_{i,j}} - M_{\mathbf{NE}_{i+1,j+1}} + M_{\mathbf{SE}_{i,j+1}} - M_{\mathbf{SE}_{i+1,j}}.
\end{aligned} \tag{8.19}$$

Here, $M_{\mathbf{E}}$ and $M_{\mathbf{N}}$ are axis-parallel mass fluxes, and $M_{\mathbf{NE}}$ and $M_{\mathbf{SE}}$ are horizontally diagonal ones (Figure 8.5).

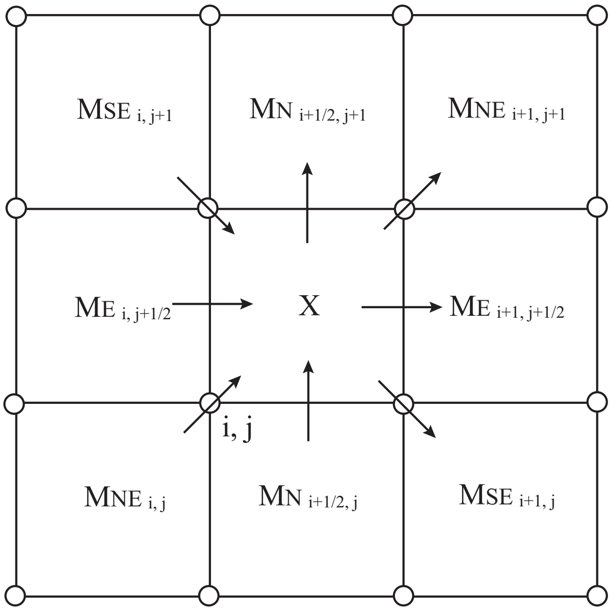

Figure8.5 Distribution of generalized mass fluxes for U-cell $(i + 1/2, j + 1/2)$

If we derive the formula for the standard case from (8.17) (all of N are 4),

$$\begin{aligned}\mathrm{HMC}^{U}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}&\left[\frac{1}{2}(U^{U}_{i,j} + U^{U}_{i,j+1}) - \frac{1}{2}(U^{U}_{i+1,j} + U^{U}_{i+1,j+1})\right.\\ &\left.+\frac{1}{2}(V^{U}_{i,j} + V^{U}_{i,j+1}) - \frac{1}{2}(V^{U}_{i+1,j} + V^{U}_{i+1,j+1})\right]\\ +\frac{1}{2}&\left[\frac{1}{2}(U^{U}_{i,j} + V^{U}_{i,j}) - \frac{1}{2}(U^{U}_{i+1,j+1} + V^{U}_{i+1,j+1})\right.\\ &\left.+\frac{1}{2}(U^{U}_{i,j+1} - V^{U}_{i,j+1}) - \frac{1}{2}(U^{U}_{i+1,j} - V^{U}_{i+1,j})\right].\end{aligned} \tag{8.20}$$

This expression means that the horizontal mass flux convergence is a mean of those of the axis-parallel mass fluxes (6.23) and of the diagonal ones (6.29). However, their weighting factors α and β are both $1/2$ in the present case, while $(\alpha, \beta) = (2/3, 1/3)$ for the generalized Arakawa scheme, which conserves quasi-enstrophy such as $(\delta v/\delta x)^2$ and $(\delta u/\delta y)^2$ in a horizontally non-divergent flow.

b. Horizontal momentum advection

For the standard case away from land, we have the freedom to choose weights $(\alpha : \beta)$ for averaging the convergences of the axis-parallel and the horizontally diagonal mass fluxes, as long as $\alpha + \beta = 1$, as seen in (8.20). In MRI.COM $\alpha = 2/3$ and $\beta = 1/3$ are chosen for the standard case so that the momentum advection terms lead to the generalized Arakawa scheme. In this case the zonal momentum advection term is expressed by convergence of the horizontal momentum fluxes

Table8.1 Definition of a land-sea index, the index identifying each case (column **A**), the coefficient of $U^U_{i,j}$ in the axis-parallel mass flux (column **B**), and the coefficient of $U^U_{i,j}$ in the horizontally diagonal mass flux (column **C**), for eight combinations of indices **a**, **b**, and **c** in Figure 8.2b (see the main text). Cell **d** is assumed to be a sea cell, and the momentum advection by means of $U^U_{i,j}$ into and from cell **d** is generalized. Note that $U^U_{i,j}$ is identically zero for cases 5 and 8 (**b** = **c** = 0).

CASE n	Land-sea index **a b c**	**A**	**B** Coefficient of $U^U_{i,j}$ (axis-parallel) +	**C** Coefficient of $U^U_{i,j}$ (horizontally-diagonal) ×
1	1 1 1	**abc**	1/3	1/6
2	1 1 0	**ab**(1 − **c**)	0	1/3
3	1 0 1	**ab**(1 − **b**)**c**	1/3	0
4	0 1 1	(1 − **a**)**bc**	1/3	1/3
5	1 0 0	−	0	0
6	0 1 0	(1 − **a**)**b**(1 − **c**)	0	1/2
7	0 0 1	(1 − **a**)(1 − **b**)**c**	1/2	0
8	0 0 0	−	0	0

as follows:

$$\begin{aligned}\mathrm{CAD}_{i+\frac{1}{2},j+\frac{1}{2}}(u) = \frac{2}{3}\Big[&\frac{1}{2}(u_{i-\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2}})\frac{1}{2}(U^U_{i,j} + U^U_{i,j+1})\\ &-\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{3}{2},j+\frac{1}{2}})\frac{1}{2}(U^U_{i+1,j} + U^U_{i+1,j+1})\\ &+\frac{1}{2}(u_{i+\frac{1}{2},j-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2}})\frac{1}{2}(V^U_{i,j} + V^U_{i+1,j})\\ &-\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{3}{2}})\frac{1}{2}(V^U_{i,j+1} + V^U_{i+1,j+1})\Big]\\ +\frac{1}{3}\Big[&\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i-\frac{1}{2},j-\frac{1}{2}})\frac{1}{2}(U^U_{i,j} + V^U_{i,j})\\ &-\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{3}{2},j+\frac{3}{2}})\frac{1}{2}(U^U_{i+1,j+1} + V^U_{i+1,j+1})\\ &+\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i-\frac{1}{2},j+\frac{3}{2}})\frac{1}{2}(U^U_{i,j+1} - V^U_{i,j+1})\\ &-\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{3}{2},j-\frac{1}{2}})\frac{1}{2}(U^U_{i+1,j} - V^U_{i+1,j})\Big]\end{aligned} \tag{8.21}$$

This scheme under Arakawa's B-grid arrangement conserves the quasi-enstrophies ($(\partial u/\partial y)^2$ and $(\partial v/\partial x)^2$) in a horizontally non-divergent flow.

To merge the generalized Arakawa scheme for the standard case into the general form of the horizontal mass flux expressed in Figure 8.5 and related momentum flux, let us examine the axis-parallel and horizontally diagonal mass flux associated with $U^U_{i,j}$, taking topography into account. Look at Figure 8.2b, where letters **a**, **b**, **c**, and **d** designate the land-sea index and names of U-cells. Cell **d** is assumed to be a sea cell (**d** = 1). We analyze two kinds of mass fluxes associated with $U^U_{i,j}$ under different combinations of **a**, **b**, and **c** (eight cases), as indicated in the first column in Table 8.1. Column (A) corresponds to an index, which is unity for its own combination and zero for all other combinations. Column (B) lists the coefficient of $U^U_{i,j}$ in the axis-parallel mass flux of the U-cell mass continuity (8.17). Column (C) indicates the coefficient of $U^U_{i,j}$ in the horizontally diagonal mass flux of (8.17).

The generalized coefficient of $U^U_{i,j}$ in the axis-parallel mass flux (c_1) is obtained by summing the product of **A** and **B** over the eight cases. Similarly, the generalized coefficient in the horizontally diagonal mass flux (c_2) is obtained by the summing the product of **A** and **C**. That is,

$$c_1 = \sum_{n=1}^{8} \mathbf{A}_n \mathbf{B}_n = \frac{1}{6}\mathbf{c}(\mathbf{ab} - \mathbf{a} - \mathbf{b}{+}3)$$

and

$$c_2 = \sum_{n=1}^{8} \mathbf{A}_n \mathbf{C}_n = \frac{1}{6}\mathbf{b}(3{-}\mathbf{a} - \mathbf{c}). \tag{8.22}$$

Then, the axis-parallel and the horizontally diagonal flux of zonal momentum (u) related with $U^U_{i,j}$, multiplied by the land-sea index **d**, are

$$\frac{\mathbf{d}}{2}(u_\mathbf{c} + u_\mathbf{d})c_1 U^U_{i,j} = \frac{1}{2}(u_\mathbf{c} + u_\mathbf{d})\frac{1}{6}\mathbf{cd}(\mathbf{ab} - \mathbf{a} - \mathbf{b}{+}3)U^U_{i,j},$$

and

$$\frac{\mathbf{d}}{2}(u_\mathbf{b} + u_\mathbf{d})c_2 U^U_{i,j} = \frac{1}{2}(u_\mathbf{b} + u_\mathbf{d})\frac{1}{6}\mathbf{bd}(3{-}\mathbf{a} - \mathbf{c})U^U_{i,j}, \tag{8.23}$$

respectively.

The resultant momentum fluxes are as follows:

$$\begin{aligned}
F_{\mathbf{E}_{i,j+\frac{1}{2}}}(u) &= \frac{1}{2}(u_{i-\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2}})M_{\mathbf{E}_{i,j+\frac{1}{2}}},\\
F_{\mathbf{N}_{i+\frac{1}{2},j}}(u) &= \frac{1}{2}(u_{i+\frac{1}{2},j-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2}})M_{\mathbf{N}_{i+\frac{1}{2},j}},\\
F_{\mathbf{NE}_{i,j}}(u) &= \frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i-\frac{1}{2},j-\frac{1}{2}})M_{\mathbf{NE}_{i,j}},\\
F_{\mathbf{SE}_{i,j}}(u) &= \frac{1}{2}(u_{i-\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}})M_{\mathbf{SE}_{i,j}},
\end{aligned} \tag{8.24}$$

where

$$\begin{aligned}
M_{\mathbf{E}_{i,j+\frac{1}{2}}} &= \frac{1}{6}(C_{\mathbf{XN}_{i,j}}U^U_{i,j} + C_{\mathbf{XS}_{i,j+1}}U^U_{i,j+1}),\\
M_{\mathbf{N}_{i+\frac{1}{2},j}} &= \frac{1}{6}(C_{\mathbf{YE}_{i,j}}V^U_{i,j} + C_{\mathbf{YW}_{i+1,j}}V^U_{i+1,j}),\\
M_{\mathbf{NE}_{i,j}} &= \frac{1}{6}C_{\mathbf{NE}_{i,j}}(U^U_{i,j} + V^U_{i,j}),\\
M_{\mathbf{SE}_{i,j}} &= \frac{1}{6}C_{\mathbf{SE}_{i,j}}(U^U_{i,j} - V^U_{i,j}),
\end{aligned} \tag{8.25}$$

and

$$\begin{aligned}
C_{\mathbf{XN}_{i,j}} &= e_{i+\frac{1}{2},j+\frac{1}{2}}e_{i-\frac{1}{2},j+\frac{1}{2}}(e_{i+\frac{1}{2},j-\frac{1}{2}}e_{i-\frac{1}{2},j-\frac{1}{2}} - e_{i+\frac{1}{2},j-\frac{1}{2}} - e_{i-\frac{1}{2},j-\frac{1}{2}} + 3),\\
C_{\mathbf{XS}_{i,j}} &= e_{i+\frac{1}{2},j-\frac{1}{2}}e_{i-\frac{1}{2},j-\frac{1}{2}}(e_{i+\frac{1}{2},j+\frac{1}{2}}e_{i-\frac{1}{2},j+\frac{1}{2}} - e_{i+\frac{1}{2},j+\frac{1}{2}} - e_{i-\frac{1}{2},j+\frac{1}{2}} + 3),\\
C_{\mathbf{YE}_{i,j}} &= e_{i+\frac{1}{2},j+\frac{1}{2}}e_{i+\frac{1}{2},j-\frac{1}{2}}(e_{i-\frac{1}{2},j+\frac{1}{2}}e_{i-\frac{1}{2},j-\frac{1}{2}} - e_{i-\frac{1}{2},j+\frac{1}{2}} - e_{i-\frac{1}{2},j-\frac{1}{2}} + 3),\\
C_{\mathbf{YW}_{i,j}} &= e_{i-\frac{1}{2},j+\frac{1}{2}}e_{i-\frac{1}{2},j-\frac{1}{2}}(e_{i+\frac{1}{2},j+\frac{1}{2}}e_{i+\frac{1}{2},j-\frac{1}{2}} - e_{i+\frac{1}{2},j+\frac{1}{2}} - e_{i+\frac{1}{2},j-\frac{1}{2}} + 3),\\
C_{\mathbf{NE}_{i,j}} &= e_{i+\frac{1}{2},j+\frac{1}{2}}e_{i-\frac{1}{2},j-\frac{1}{2}}(3 - e_{i-\frac{1}{2},j+\frac{1}{2}} - e_{i+\frac{1}{2},j-\frac{1}{2}}),\\
C_{\mathbf{SE}_{i,j}} &= e_{i-\frac{1}{2},j+\frac{1}{2}}e_{i+\frac{1}{2},j-\frac{1}{2}}(3 - e_{i+\frac{1}{2},j+\frac{1}{2}} - e_{i-\frac{1}{2},j-\frac{1}{2}}).
\end{aligned} \tag{8.26}$$

Finally, convergence of the horizontal momentum fluxes is written as

$$\begin{aligned}
\mathrm{CAD}_{i+\frac{1}{2},j+\frac{1}{2}}(u) =& F_{\mathbf{E}_{i,j+\frac{1}{2}}}(u) - F_{\mathbf{E}_{i+1,j+\frac{1}{2}}}(u) + F_{\mathbf{N}_{i+\frac{1}{2},j}}(u) - F_{\mathbf{N}_{i+\frac{1}{2},j+1}}(u)\\
&+ F_{\mathbf{NE}_{i,j}}(u) - F_{\mathbf{NE}_{i+1,j+1}}(u) + F_{\mathbf{SE}_{i,j+1}}(u) - F_{\mathbf{SE}_{i+1,j}}(u).
\end{aligned} \tag{8.27}$$

This is the discrete expression for the advection term of the zonal momentum $\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi uu)}{\partial\mu} + \frac{\partial(z_s h_\mu vu)}{\partial\psi}\right\}$ under the finite volume method (equations being integrated over a U-cell).

8.2 Pressure gradient term

In the split-explicit solution method, the pressure gradient term of the horizontal momentum equation is separated into barotropic (fast) and baroclinic (slow) terms (Eq. 2.56). Its form is

$$\underbrace{\frac{1}{\rho_0}\nabla_s(p_a + g\rho_0\eta)}_{\text{fast}} + \underbrace{\overbrace{\frac{1}{\rho_0}\nabla_s\left[g\int_{z(s)}^{\eta}\rho' dz\right]}^{\text{pressure perturbation}} + \overbrace{\frac{g\rho'}{\rho_0}\nabla_s z}^{\text{geopotential}}}_{\text{slow}}\,. \tag{8.28}$$

Here, we use symbol s to indicate z^* for brevity, and so ∇_s means ∇_{z^*}. These three terms correspond to the first, second, and third terms in the r.h.s. of (8.1) and (8.2). From the perspective of the momentum conservation, the finite difference of the pressure gradient terms should be expressed so that the pressure at the interface of adjacent cells is common, giving only boundary pressures after horizontal integration. This is not difficult for the barotropic mode. The baroclinic mode should be considered carefully. Among the baroclinic terms in (8.28), the former is called the pressure perturbation term and the latter is called the geopotential term.

Because sea-floor depth is defined at U-cells in MRI.COM, it is not necessary to consider the horizontal gradient of the sea floor in calculating the pressure gradient for a bottom U-cell unlike σ-coordinate models and possibly the depth-coordinate models that employ staggered grid arrangements different from MRI.COM. This makes the finite difference expression simple, and the pressure gradient error may be expected to be small. However, this simplicity does not hold for the bottom boundary layer (BBL). The treatment of the pressure gradient term for BBL is explained in Chapter 16.

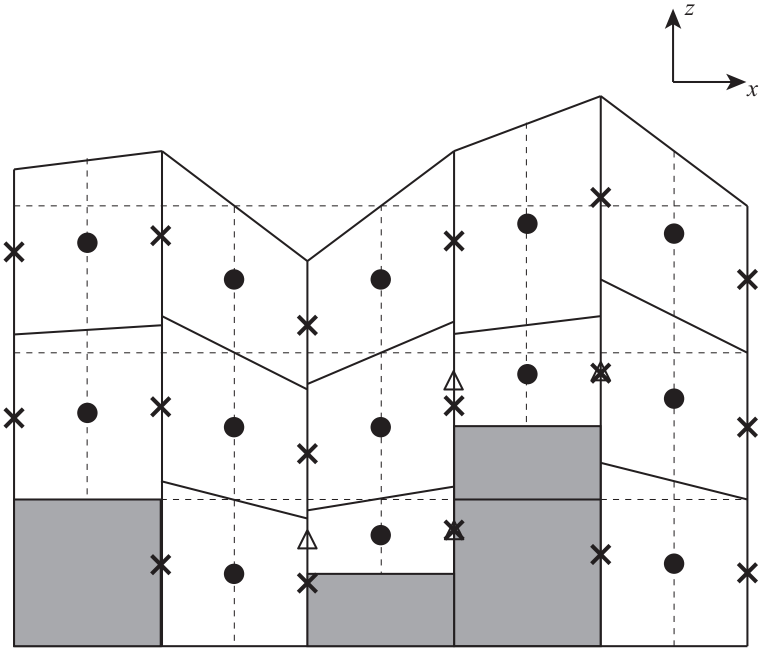

Figure8.6 Illustration of a vertical slice through a set of grid cells in the x-z plane for z^* coordinate. The center point in each cell (•) is a velocity point, and the pressure gradient term is calculated at this point. The cross (×) is a tracer point, and pressure is calculated here. The triangle (△), whose z^* value is that of the adjacent U-point of the bottom U-cell, is the hydrostatic pressure point which is used to evaluate pressure gradient for the bottom U-cell.

a. Discretization of the pressure perturbation term

Figure 8.6 shows the grid points on the x-z plane relevant to calculating pressure gradient terms. Pressures are defined on T-points. Because the actual vertical integration distances are different between adjacent T-points for z^* coordinate, it is appropriate to differentiate pressures after computing them at T-points rather than vertically integrating the pressure gradient as in z coordinate. However, the round-off error due to the differentiation of vertically integrated quantity would increase with depth. To avoid this, we add contribution from the integration over one vertical grid to the pressure gradient term at the upper vertical level, as explained below.

Introducing pressure perturbation p', the pressure perturbation term in Eq. (8.28) is expressed as

$$\frac{1}{\rho_0}\nabla_s\left[g\int_{z(s)}^{\eta}\rho' dz\right]=\frac{1}{\rho_0}\nabla_s p', \tag{8.29}$$

$$\text{where}\quad p'=g\int_{z(s)}^{\eta}\rho' dz. \tag{8.30}$$

A simple discrete expression would be given as follows ($1/\rho_0$ is omitted):

$$(\nabla_s p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}=\hat{\mathbf{x}}\frac{\frac{p'_{i+1,j+1,k-\frac{1}{2}}+p'_{i+1,j,k-\frac{1}{2}}}{2}-\frac{p'_{i,j+1,k-\frac{1}{2}}+p'_{i,j,k-\frac{1}{2}}}{2}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}+\hat{\mathbf{y}}\frac{\frac{p'_{i+1,j+1,k-\frac{1}{2}}+p'_{i,j+1,k-\frac{1}{2}}}{2}-\frac{p'_{i+1,j,k-\frac{1}{2}}+p'_{i,j,k-\frac{1}{2}}}{2}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}, \tag{8.31}$$

$$\text{where}\quad p'_{i,j,k-\frac{1}{2}}=g\sum_{l=1}^{k-1}(\rho')_{i,j,l-\frac{1}{2}}(\mathtt{dzt})_{i,j,l-\frac{1}{2}}+g(\rho')_{i,j,k-\frac{1}{2}}\frac{(\mathtt{dzt})_{i,j,k-\frac{1}{2}}}{2}. \tag{8.32}$$

However, because density perturbation (ρ') would generally take similar values in adjacent grids, the pressure perturbation (p') would also take similar values in adjacent grids. The round-off error caused by differentiation would increase with the depth. To avoid this round-off error, we use another discrete expression given as follows:

$$\begin{aligned}(\nabla_s p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}&=(\nabla_s p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{3}{2}}+\left\{(\nabla_s p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}-(\nabla_s p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{3}{2}}\right\}\\&=(\nabla_s p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{3}{2}}+\hat{\mathbf{x}}\frac{\frac{\Delta p'_{i+1,j+1,k-1}+\Delta p'_{i+1,j,k-1}}{2}-\frac{\Delta p'_{i,j+1,k-1}+\Delta p'_{i,j,k-1}}{2}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\\&\quad+\hat{\mathbf{y}}\frac{\frac{\Delta p'_{i+1,j+1,k-1}+\Delta p'_{i,j+1,k-1}}{2}-\frac{\Delta p'_{i+1,j,k-1}+\Delta p'_{i,j,k-1}}{2}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}},\end{aligned} \tag{8.33}$$

where

$$\Delta p'_{i,j,k-1}\equiv p'_{i,j,k-\frac{1}{2}}-p'_{i,j,k-\frac{3}{2}}=g(\rho')_{i,j,k-\frac{3}{2}}\frac{(\mathtt{dzt})_{i,j,k-\frac{1}{2}}}{2}+g(\rho')_{i,j,k-\frac{1}{2}}\frac{(\mathtt{dzt})_{i,j,k-\frac{1}{2}}}{2}. \tag{8.34}$$

b. Discretization of the geopotential term

The geopotential term is also discretized so that a round-off error is minimized. Considering that the time-dependent part of the actual depth $z=z(s)$ is due to the perturbation ($z'(s)$) from the state of rest ($z_0(s)$) caused by the sea level variation, taking differentiation for $z(s)$ at depth is not quite accurate owing to the numerical round-off. Then we separate the actual depth as $z(s)=z_0(s)+z'(s)$ and only use $z'(s)$ for horizontal gradient of geopotential, that is,

$$\frac{g\rho'}{\rho_0}\nabla_s z=\frac{g\rho'}{\rho_0}\nabla_s z'(s). \tag{8.35}$$

Here, we used the fact that the depth of the constant s-surface is flat at the state of rest, i.e. $\nabla_s z_0(s)=0$. (Again, we cannot use this simplicity for the bottom boundary layer. See Chapter 16 for details.) Then, the term is differentiated as

$$\begin{aligned}&\left[\frac{g\rho'}{\rho_0}\nabla_s z'\right]_{i+\frac{1}{2},j+\frac{1}{2}}\\&=\frac{g}{\rho_0}\left\{\left[\frac{\frac{\rho'_{i+1,j}+\rho'_{i,j}}{2}\frac{z'_{i+1,j}-z'_{i,j}}{\Delta x_{i+\frac{1}{2},j}}+\frac{\rho'_{i+1,j+1}+\rho'_{i,j+1}}{2}\frac{z'_{i+1,j+1}-z'_{i,j+1}}{\Delta x_{i+\frac{1}{2},j+1}}}{2}\hat{\mathbf{x}}\right]+\left[\frac{\frac{\rho'_{i,j+1}+\rho'_{i,j}}{2}\frac{z'_{i,j+1}-z'_{i,j}}{\Delta y_{i,j+\frac{1}{2}}}+\frac{\rho'_{i+1,j+1}+\rho'_{i+1,j}}{2}\frac{z'_{i+1,j+1}-z'_{i+1,j}}{\Delta y_{i+1,j+\frac{1}{2}}}}{2}\hat{\mathbf{y}}\right]\right\}.\end{aligned} \tag{8.36}$$

Here, we follow the expression presented in Section 3.3.1 of the reference manual of GFDL-MOM.

c. The sea-floor grid (except for bottom boundary layer)

Owing to the introduction of the partial cell, the grid width of the bottom U-cell may depend on the horizontal position. The pressure gradient term in (8.28) is the gradient of pressure on constant $s(z^*)$-surface. Thus, the pressure gradient force on a partial U-cell is evaluated using the pressure perturbation p' obtained by integrating to the depth where velocity is defined on z^* coordinate (the central depth of the bottom U-cell). These are the points with triangle (△) symbols in Figure 3.6.

To calculate the pressure gradient, the actual depth of the vertical level where velocity is defined for the bottom U-cell ($s = s^U_{k=kbtm}$) must be obtained at all four corner T-points of a U-cell. Here, these are determined using the ratio of the half width of a U-cell $\frac{1}{2}\Delta s^U$ to the width of a T-cell Δs^T:

$$z^U_{kbtm} = z^T_{kbtm-1} - \Delta z^T_{kbtm} \frac{\Delta s^U_{kbtm}}{2\Delta s^T_{kbtm}}. \tag{8.37}$$

The depth anomaly used for computing geopotential gradient anomaly may be obtained in the same way:

$$z'^U_{kbtm} = z'^T_{kbtm-1} - \Delta z'^T_{kbtm} \frac{\Delta s^U_{kbtm}}{2\Delta s^T_{kbtm}}. \tag{8.38}$$

The finite difference expression for horizontal gradient is formally the same as (8.33) and (8.36). However, the vertical integration to obtain p' is performed to the depth where velocity is defined for the bottom U-cell. Geopotential term is evaluated using a depth anomaly of $s(z^*)$-surface. Thus, on the same vertical level, the algorism is different depending on whether it is bottom cell or not.

8.3 Viscosity

The viscosity in an ocean general circulation model seeks to attenuate numerical noise rather than parameterizing the subgrid-scale momentum transport. The momentum advection scheme should conserve the total kinetic energy in the general three-dimensional flows and the total enstrophy in the two-dimensional flows. Therefore, spatially and temporally centered discretization should be used, although this inevitably produces near-grid-size noise accompanying numerical dispersion. In eddy-resolving models, the current velocity and the numerical noise are greater than those of eddy-less models. A biharmonic viscosity scheme has been widely used to reduce numerical noise while maintaining the eddy structure.

The viscosity term is represented by $\mathcal{V}$ and is calculated separately in the lateral and vertical directions, i.e., the fourth and fifth terms, respectively, on the r.h.s. of (8.1) and (8.2). For horizontal viscosity, the harmonic (default) or biharmonic (`VISBIHARM` option) scheme can be selected. Anisotropy of viscosity with respect to the flow direction can be applied (`VISANISO` option) when harmonic viscosity is chosen. The viscosity coefficient is a constant by default but can be determined as a function of local velocity gradients and grid-size (`SMAGOR` option).

For vertical viscosity, the harmonic scheme is used and the local coefficient is the larger one of the background constant and the value calculated from a turbulence closure scheme. A parameterization of bottom friction (Weatherly et al., 1980) is adopted at the lowest layer.

8.3.1 Horizontal viscosity

The specific form for harmonic viscosity is shown here. Horizontal tension D_T and shear D_S are defined as follows:

$$D_T = h_\psi \frac{\partial}{h_\mu \partial \mu}\left(\frac{u}{h_\psi}\right) - h_\mu \frac{\partial}{h_\psi \partial \psi}\left(\frac{v}{h_\mu}\right), \tag{8.39}$$

$$D_S = h_\psi \frac{\partial}{h_\mu \partial \mu}\left(\frac{v}{h_\psi}\right) + h_\mu \frac{\partial}{h_\psi \partial \psi}\left(\frac{u}{h_\mu}\right). \tag{8.40}$$

The viscosity terms are

$$\mathcal{V}_u = \frac{1}{h_\psi^2} \frac{\partial}{h_\mu \partial \mu}\left(h_\psi^2 \sigma_T\right) + \frac{1}{h_\mu^2} \frac{\partial}{h_\psi \partial \psi}\left(h_\mu^2 \sigma_S\right), \tag{8.41}$$

$$\mathcal{V}_v = \frac{1}{h_\psi^2} \frac{\partial}{h_\mu \partial \mu}\left(h_\psi^2 \sigma_S\right) - \frac{1}{h_\mu^2} \frac{\partial}{h_\psi \partial \psi}\left(h_\mu^2 \sigma_T\right), \tag{8.42}$$

where $\sigma_T = \nu_H D_T$ and $\sigma_S = \nu_H D_S$, and ν_H is the horizontal viscosity coefficient. The above representation of the viscous term was derived by Bryan (1969) and is consistent with Smagorinsky (1963).

If we take $h_\mu = 1, h_\psi = 1$, the coordinate system is Cartesian. In this case, the viscosity term is reduced to the Laplacian form if the viscosity coefficient is a constant. In the geographic coordinate system, where $(\mu, \psi) = (\lambda, \phi)$, $h_\lambda = a\cos\phi$, and $h_\phi = a$, tension and shear are

$$D_T = \frac{1}{a\cos\phi}\frac{\partial u}{\partial \lambda} - \frac{1}{a}\frac{\partial v}{\partial \phi} - \frac{v}{a}\tan\phi, \tag{8.43}$$

$$D_S = \frac{1}{a\cos\phi}\frac{\partial v}{\partial \lambda} + \frac{1}{a}\frac{\partial u}{\partial \phi} + \frac{u}{a}\tan\phi. \tag{8.44}$$

The viscosity terms in this case are

$$\mathcal{V}_u = \frac{1}{a\cos\phi}\frac{\partial}{\partial \lambda}\sigma_T + \frac{1}{a}\frac{\partial}{\partial \phi}\sigma_S - \sigma_S\frac{2\tan\phi}{a}, \tag{8.45}$$

$$\mathcal{V}_v = \frac{1}{a\cos\phi}\frac{\partial}{\partial \lambda}\sigma_S - \frac{1}{a}\frac{\partial}{\partial \phi}\sigma_T + \sigma_T\frac{2\tan\phi}{a}, \tag{8.46}$$

where the third term on the r.h.s. is called the metric term.

When biharmonic viscosity is used (VISBIHARM option), the above operation is repeated twice using a viscosity coefficient ν_{BH}. The terms $\mathcal{V}_u$ and $\mathcal{V}_v$ given by (8.41) and (8.42) are sign-reversed and substituted as u and v in equations (8.39) and (8.40). A biharmonic scheme dissipates noise only on scales near the grid size. This scale selectivity allows the explicitly represented eddies to survive without unphysical damping in eddy-resolving models, although we must note that a biharmonic operator produces overshootings and spurious oscillations of variables (Delhez and Deleersnijder, 2007). A biharmonic viscosity scheme is not suitable for coarse resolution models that cannot resolve mesoscale eddies.

A non-slip condition is used for the side boundaries of topography by default in MRI.COM. A free-slip condition assuming zero viscosity there is also available.

8.3.2 Horizontal anisotropic viscosity (VISANISO)

Smith and McWilliams (2003) proposed a method of making a harmonic viscosity scheme anisotropic in an arbitrary direction. Setting σ_T and σ_S in equations (8.41) and (8.42) to

$$\begin{pmatrix} \sigma_T \\ \sigma_S \end{pmatrix} = \left[\begin{pmatrix} \frac{1}{2}(\nu_0+\nu_1) & 0 \\ 0 & \nu_1 \end{pmatrix} + (\nu_0 - \nu_1) n_\mu n_\psi \begin{pmatrix} -2n_\mu n_\psi & n_\mu^2 - n_\psi^2 \\ n_\mu^2 - n_\psi^2 & 2n_\mu n_\psi \end{pmatrix}\right]\begin{pmatrix} D_T \\ D_S \end{pmatrix}, \tag{8.47}$$

where $\hat{\mathbf{n}} = (n_\mu, n_\psi)$ is a unit vector in an arbitrary direction and ν_0 (ν_1) is the viscosity coefficient parallel (perpendicular) to $\hat{\mathbf{n}}$. When VISANISO option is selected, $\hat{\mathbf{n}}$ is set to the direction of local flow in MRI.COM. Given the harmonic viscosity only in the direction of flow ($\nu_1 = 0$), the numerical noise is erased while the swift currents and eddy structures are maintained.

The following is a note on usage. The behavior of this scheme is specified at run time by namelist nml_visaniso (Table 8.2). The ratio ν_1/ν_0 should be given by cc0 (default value is 0.2). The ratio at the lateral boundary should be given by cc1 (default value is 0.5). When the variable flgvisequator is set as a positive number in the namelist, the ratio ν_1/ν_0 is tapered linearly from cc0 at the latitude flgvisequator (in degrees) to vis_factor_equator at the Equator. The ratio is not tapered when a negative number is set, and the default value of flgvisequator is -1.

Table8.2 Namelist nml_visaniso anisotropic horizontal viscosity

variable	units	description	usage
cc0	1	the factor of anisotropy in viscosity; perpendicular / parallel (ν_1/ν_0) to the flow	default = 0.2
cc1	1	the factor (ν_1/ν_0) at lateral boundary	default = 0.5
figvisequator	degree latitude	the factor is tapered toward the Equator; when |LAT| < flgvisequator [deg]	default = -1 (no tapering)
vis_factor_equator	1	the factor at Equator when flgvisequator > 0	default = 0.0

8.3.3 Smagorinsky parameterization for horizontal viscosity (SMAGOR option)

To give the necessary but minimum viscosity to reduce numerical noise, the viscosity coefficient is made proportional to the local deformation rate (SMAGOR option; Smagorinsky, 1963; Griffies and Hallberg, 2000). When this parameterization is used with the biharmonic scheme, the scale selectivity of the viscosity scheme becomes more effective.

Defining deformation rate $|D|$:

$$|D| = \sqrt{D_T^2 + D_S^2}, \tag{8.48}$$

the viscosity coefficients are set as follows:

$$\nu_H = \left(\frac{C\Delta_{\text{min}}}{\pi}\right)^2 |D|, \tag{8.49}$$

$$\nu_{BH} = \frac{\Delta_{\text{min}}^2}{8}\nu_H, \tag{8.50}$$

where C (smagor_scale) is a dimensionless scaling parameter set by considering numerical stability and Δ_{min} is the smaller of the zonal and meridional grid widths.

The parameter C should be selected to satisfy the following conditions.

- Restriction of grid Reynolds number:

$$\nu_H > U\frac{\Delta_{\text{min}}}{2}, \tag{8.51}$$

- Restriction on the width of the lateral boundary layer:

$$\nu_H > \beta\Delta_{\text{min}}^3, \tag{8.52}$$

- CFL condition:

$$\nu_H < \frac{\Delta_{\text{min}}^2}{2\Delta t}, \tag{8.53}$$

where $\beta = df/dy$ is the meridional gradient of the Coriolis parameter. Scaling the deformation rate $|D|$ by U/Δ_{min} gives the condition for stability: $C > \pi/\sqrt{2} \approx 2.2$ from (8.51) (Griffies and Hallberg, 2000).

Behavior of this scheme at run time is specified by namelist nml_smagor, whose components are listed on Table 8.3.

Table8.3 Namelist nml_smagor for Smagorinsky parameterization of horizontal viscosity coefficient

variable	units	description	usage
smagor_scale	1	scaling factor coefficient	default = −1.0
smagor_diff_ratio	1	ratio for diffusivity to scaling factor of viscosity (valid only when SMAGHD is specified)	required
l_smagor_nonzero_bg	logical	Use background viscosity as given by nml_baroclinic_visc_horz (Table 8.4)	default = .false.

8.3.4 Discretization of the horizontal viscosity term

Using the notations

$$\delta_\mu A_{i,j} \equiv \frac{A_{i+\frac{1}{2},j} - A_{i-\frac{1}{2},j}}{\Delta\mu}, \quad \delta_\psi A_{i,j} \equiv \frac{A_{i,j+\frac{1}{2}} - A_{i,j-\frac{1}{2}}}{\Delta\psi},$$
$$\delta_i A_{i,j} \equiv A_{i+\frac{1}{2},j} - A_{i-\frac{1}{2},j}, \quad \delta_j A_{i,j} \equiv A_{i,j+\frac{1}{2}} - A_{i,j-\frac{1}{2}},$$

and

$$\overline{A_{i,j}}^{\mu} \equiv \frac{1}{2}(A_{i-\frac{1}{2},j} + A_{i+\frac{1}{2},j}), \quad \overline{A_{i,j}}^{\psi} \equiv \frac{1}{2}(A_{i,j-\frac{1}{2}} + A_{i,j+\frac{1}{2}}),$$

deformation rates are discretized as follows:

$$
\begin{aligned}
D_{T\,i,j} &= \frac{h_{\psi i,j}}{h_{\mu i,j}} \overline{\delta_\mu \left(\frac{u}{h_\psi}\right)_{i,j}}^{\psi} - \frac{h_{\mu i,j}}{h_{\psi i,j}} \overline{\delta_\psi \left(\frac{v}{h_\mu}\right)_{i,j}}^{\mu}, \\
D_{S\,i,j} &= \frac{h_{\psi i,j}}{h_{\mu i,j}} \overline{\delta_\mu \left(\frac{v}{h_\psi}\right)_{i,j}}^{\psi} + \frac{h_{\mu i,j}}{h_{\psi i,j}} \overline{\delta_\psi \left(\frac{u}{h_\mu}\right)_{i,j}}^{\mu}.
\end{aligned}
\tag{8.54}
$$

Horizontal viscosity forces of (8.41) and (8.42) are discretized as follows:

$$
\begin{aligned}
F_{x\,i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}}} \\
&\times \left[\frac{1}{h^2_{\psi i+\frac{1}{2},j+\frac{1}{2}}} \delta_i \left(\Delta y \Delta z h_\psi^2 \overline{\nu_H D_T}^{\psi} \right)_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{1}{h_{\mu}{}^2_{i+\frac{1}{2},j+\frac{1}{2}}} \delta_j \left(\Delta x \Delta z h_\mu^2 \overline{\nu_H D_S}^{\mu} \right)_{i+\frac{1}{2},j+\frac{1}{2}} \right], \\
F_{y\,i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}}} \\
&\times \left[\frac{1}{h^2_{\psi i+\frac{1}{2},j+\frac{1}{2}}} \delta_i \left(\Delta y \Delta z h_\psi^2 \overline{\nu_H D_S}^{\psi} \right)_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{1}{h_{\mu}{}^2_{i+\frac{1}{2},j+\frac{1}{2}}} \delta_j \left(\Delta x \Delta z h_\mu^2 \overline{\nu_H D_T}^{\mu} \right)_{i+\frac{1}{2},j+\frac{1}{2}} \right].
\end{aligned}
\tag{8.55}
$$

The non-slip condition at the side boundaries of topography is discretized as follows. When the grid point $(i-\frac{1}{2}, j+\frac{1}{2})$ is defined as a (vertically partial) land (Figure 8.7a), the velocity gradients at the wall are calculated as follows:

$$
\begin{aligned}
\left(\frac{\partial u}{\partial x}\right)_{i,j+\frac{1}{2}} &= \frac{u_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x^-_{ij}}, \\
\left(\frac{\partial v}{\partial x}\right)_{i,j+\frac{1}{2}} &= \frac{v_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x^-_{ij}},
\end{aligned}
\tag{8.56}
$$

where Δx^-_{ij} is the length between the points $(i, j+\frac{1}{2})$ and $(i+\frac{1}{2}, j+\frac{1}{2})$. The contribution of this wall to the force is:

$$
\begin{aligned}
F^W_{x\;i+\frac{1}{2},j+\frac{1}{2}} &= -\frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}} h_{\psi}{}^2_{i+\frac{1}{2},j+\frac{1}{2}}} \Delta y_{i,j+\frac{1}{2}} \Delta\tilde{z}_{i,j+\frac{1}{2}} h_{\psi}{}^2_{i-\frac{1}{2},j} \nu_{H\,i-\frac{1}{2},j} \frac{u_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x^-_{i+\frac{1}{2},j+\frac{1}{2}}}, \\
F^W_{y\;i+\frac{1}{2},j+\frac{1}{2}} &= -\frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}} h_{\psi}{}^2_{i+\frac{1}{2},j+\frac{1}{2}}} \Delta y_{i,j+\frac{1}{2}} \Delta\tilde{z}_{i,j+\frac{1}{2}} h_{\psi}{}^2_{i-\frac{1}{2},j} \nu_{H\,i-\frac{1}{2},j} \frac{v_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x^-_{i+\frac{1}{2},j+\frac{1}{2}}},
\end{aligned}
\tag{8.57}
$$

where $\Delta\tilde{z}_{i,j+\frac{1}{2}}$ is the wall height.

8.3.5 Vertical viscosity

Only the harmonic scheme is considered. The vertical momentum flux is assumed to be proportional to the vertical gradient of velocity. For the upper part of a U-cell at the $(k-\frac{1}{2})$th vertical level, the momentum flux (positive upward) is calculated as follows:

$$
-\left(\nu_v \frac{\partial u}{\partial z}\right)_{k-1} = -\nu_{v\,k-1} \frac{u_{k-\frac{3}{2}} - u_{k-\frac{1}{2}}}{\Delta z_{k-1}},
$$

where $\Delta z_{k-1} = (\Delta z_{k-\frac{3}{2}} + \Delta z_{k-\frac{1}{2}})/2$, $\Delta z_{k-\frac{1}{2}}$ is the thickness of the U-cell (dzu), and ν_v is the vertical viscosity coefficient. Similarly, the momentum flux in the lower part of the U-cell is calculated as follows:

$$
-\left(\nu_v \frac{\partial u}{\partial z}\right)_{k} = -\nu_{v\,k} \frac{u_{k-\frac{1}{2}} - u_{k+\frac{1}{2}}}{\Delta z_{k}},
$$

where $\nu_{v\,k}$ is set to zero if the $(k+\frac{1}{2})$th level is the solid Earth. The bottom friction is calculated independently (see the next subsection). Also note that the variations of the grid thickness due to the partial bottom cell and the undulation of the sea surface are not considered when evaluating fluxes for simplicity.

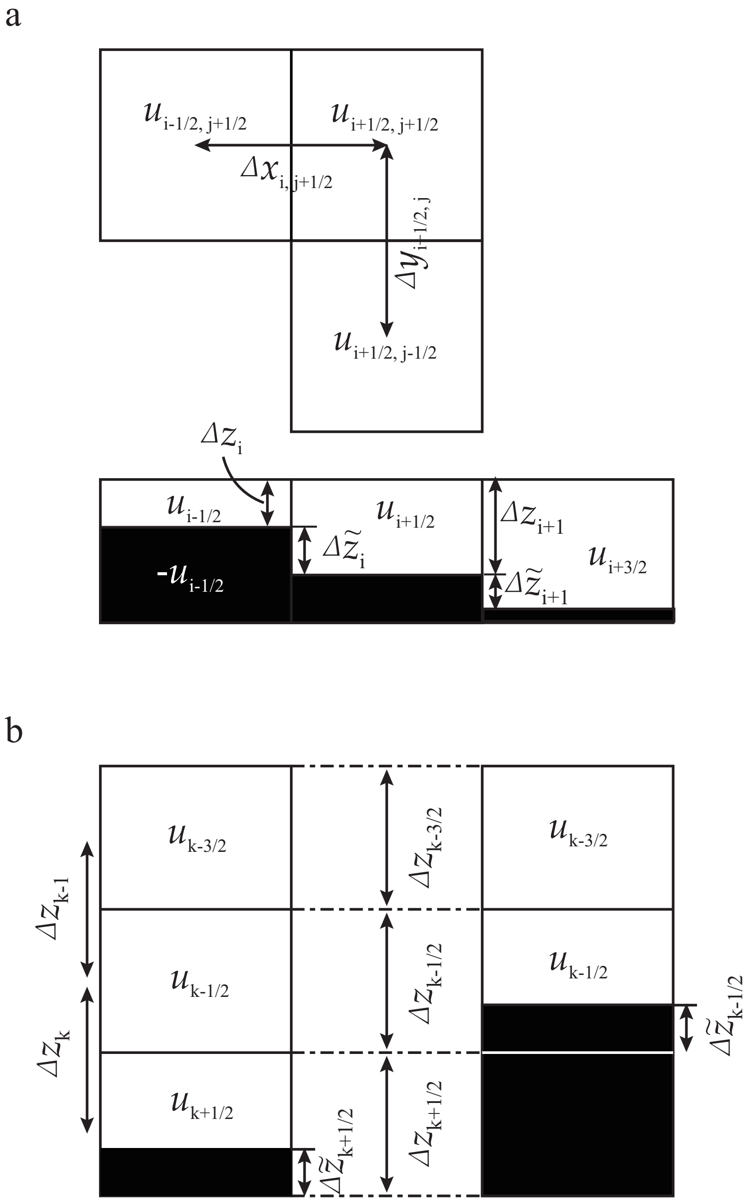

Figure8.7 (a) A schematic distribution of grids for horizontal viscosity. Upper: Plan view. Lower: Side view. The shadings denote solid earth. (b) A schematic distribution of grids for vertical viscosity. Side views. Left: The lower adjacent layer $(k + \frac{1}{2})$ has a sea bed. Right: The U-cell $(k - \frac{1}{2})$ has a sea bed.

To calculate viscosity, the divergence of the momentum flux is first calculated. The expression for the vertical viscosity term is

$$\frac{\partial}{\partial z}\left(\nu_v \frac{\partial u}{\partial z}\right)_{k-\frac{1}{2}} = \frac{\left(\nu_v \frac{\partial u}{\partial z}\right)_{k-1} - \left(\nu_v \frac{\partial u}{\partial z}\right)_k}{\widehat{\Delta z}_{k-\frac{1}{2}}} = \frac{\nu_{v\,k-1}(u_{k-\frac{3}{2}} - u_{k-\frac{1}{2}})}{\Delta z_{k-1}\widehat{\Delta z}_{k-\frac{1}{2}}} - \frac{\nu_{v\,k}(u_{k-\frac{1}{2}} - u_{k+\frac{1}{2}})}{\Delta z_k\widehat{\Delta z}_{k-\frac{1}{2}}} \tag{8.58}$$

where the variation of the grid thickness for the U-cell due to the partial bottom cell is now taken into account and is represented by $\widehat{\Delta z}$, that is, $\widehat{\Delta z}_{k-\frac{1}{2}} = \Delta z_{k-\frac{1}{2}} - \widetilde{\Delta z}_{k-\frac{1}{2}}$ (Figure 8.7(b)). Note that the first term on the r.h.s. of equation (8.58) is set to zero in calculating the viscosity term for the vertical level of $\frac{1}{2}$ $(k = 1)$. See section 14.1 for sea surface wind stress. For the vertical viscosity coefficient ν_v, MRI.COM uses the larger of the value predicted by a turbulent closure scheme and a background one.

8.3.6 Bottom friction

When a U-cell in the $(k-\frac{1}{2})$th layer contains solid earth (Figure 8.7(b) right), the stress from the lower boundary (τ_x^b, τ_y^b) is calculated following Weatherly (1972). The specific expression is as follows:

$$\begin{pmatrix} \tau_x^b \\ \tau_y^b \end{pmatrix} = -\rho_0 C_{\mathrm{btm}} \sqrt{u_{k-\frac{1}{2}}^2 + v_{k-\frac{1}{2}}^2} \begin{pmatrix} \cos\theta_0 & -\sin\theta_0 \\ \sin\theta_0 & \cos\theta_0 \end{pmatrix} \begin{pmatrix} u_{k-\frac{1}{2}} \\ v_{k-\frac{1}{2}} \end{pmatrix},$$

where C_{btm} is a dimensionless constant. Viscous stress at the lower boundary has a magnitude proportional to the square of the flow speed at the U-cell and an angle $(\theta_0 + \pi)$ relative to the flow direction.

In MRI.COM,

$$\begin{aligned} C_{\mathrm{btm}} &= 1.225 \times 10^{-3} \\ \theta_0 &= \pm\pi/18 \quad \mathrm{rad} \qquad (\equiv\ 10^\circ), \end{aligned}$$

where θ_0 is positive (negative) in the northern (southern) hemisphere. This value of C_{btm} is based on previous research about the ratio of the friction velocity, u^*, to the geostrophic current velocity in the internal region, V_g, (specifically, 0.035 is used after the median value of the range reported in Section 4a of Weatherly (1972)), and the relation equation, $C_{\mathrm{btm}} = (u^*/V_g)^2$. The variables are designated in the model as $C_{\mathrm{btm}} =$ `abtm`, $\cos\theta_0 =$ `bcs`, and $\sin(\pm\theta_0) =$ `isgn` $*$ `bsn`, where `isgn` = `1` in the northern hemisphere and `isgn` = `−1` in the southern hemisphere.

8.4 Coriolis term

Because MRI.COM employs Arakawa B-grid arrangement, both of the horizontal components of velocity are defined at the same point. Coriolis term is evaluated by using the middle time level for the leap-frog scheme and the start time level for the Euler-backward scheme.

8.5 Usage

Runtime behavior of the baroclinic mode in the default settings is specified by the five namelist blocks (some of them are optional) shown on Tables 8.4 through 8.8. For the initial condition of the 3D velocity field, restart file must be prepared for X- and Y-ward velocities.

In addition, following model options are available.

`BIHARMONIC` or `VISBIHARM`: Biharmonic horizontal viscosity is used instead of harmonic viscosity
Specify `visc_horz_cm4ps` [cm^{4}sec^{-1}] instead of `visc_horz_cm2ps` [cm^{2}sec^{-1}] in namelist `nml_baroclinic_visc_horz`. See Section 8.3.1 for detail.

`VISANISO`: Anisotropic viscosity coefficients are used
Specify the coefficients by `nml_visaniso` (Table 8.2). See Section 8.3.2 for detail.

`SMAGOR` or `SMAGHD`: Smagorinsky parameterization is used for horizontal viscosity
Specify `nml_smagor` (Table 8.3) for factors of the Smagorinsky parameterization. See Section 8.3.3 for detail.

`VIS9P`: Nine points are used in computing viscosity
Five points are used by default.

See `docs/README.Namelist` for namelist details.

Table8.4 Namelist `nml_baroclinic_visc_horz` (required) for the horizontal viscosity (see Section 8.5)

variable	units	description	usage
`visc_horz_cm2ps`	cm^2 sec^{-1}	horizontal viscosity (ν_H in Section 8.3.1)	required

Continued on next page

Table 8.4 – continued from previous page

variable	units	description	usage
file_visc_horz_2d	cm^2 sec^{-1}	2D distribution of viscosity	optional (This overwrites visc_horz_cm2ps.)
slip_factor	factor	0.d0 means no-slip condition, while 1.d0 slip (Section 8.3.1)	default = 0.d0 (only valid with VIS9P option)

Table8.5 Namelist nml_visc_vert_bg (optional) for vertical viscosity (see Section 8.5)

variable	units	description	usage
visc_vert_bg_cm2ps	cm^2 sec^{-1}	background vertical viscosity (ν_v in Section 8.3.5)	default = 1.d0

Table8.6 Namelist nml_bottom_friction (optional) for the bottom friction (see Section 8.5)

variable	units	description	usage
btmfrc_scale	1	$C_{\rm btm}$ in Section 8.3.6	default = 1.225d−3
btmfrc_angle_deg	degree	θ_0	default = 1.d1
file_btm_frc_2d	file name	2D distribution of $C_{\rm btm}$	optional (This overwrites btmfrc_scale)

Table8.7 Namelist nml_hvisc_add (optional) for additional horizonta viscosity (see Section 8.5)

variable	units	description	usage
l_hvisc_add_harmonic	logical	use additional harmonic viscosity or not	default = .false.
file_hvisc_harmonic	cm^2 sec^{-1}	2D distribution of viscosity	file name

Table8.8 Namelist nml_baroclinic_run (optional) for starting the baroclinic mode (see Section 8.5)

variable	units	description	usage
l_rst_baroclinic_in	logical	read initial restart for 3D velocity or not	default = l_rst_in

Part IV

Evolution of tracers

Chapter 9

Tracer advection-diffusion equation

In this chapter, we explain the advection-diffusion equation, which is the governing equation of the tracers including potential temperature and salinity.

9.1 The advection-diffusion equation

The equations for potential temperature and salinity, Eqs. (2.35) and (2.36), are re-written:

$$\frac{\partial(z_s\theta)}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi u\theta)}{\partial\mu}+\frac{\partial(z_s h_\mu v\theta)}{\partial\psi}\right\}+\frac{\partial(z_s\dot{s}\theta)}{\partial s}=-z_s\nabla\cdot\mathbf{F}_\theta+z_sQ^\theta, \tag{9.1}$$

$$\frac{\partial(z_sS)}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi uS)}{\partial\mu}+\frac{\partial(z_s h_\mu vS)}{\partial\psi}\right\}+\frac{\partial(z_s\dot{s}S)}{\partial s}=-z_s\nabla\cdot\mathbf{F}_S+z_sQ^S. \tag{9.2}$$

Representing θ and S with a general variable T, they become a common advection-diffusion equation,

$$\frac{\partial(z_sT)}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi uT)}{\partial\mu}+\frac{\partial(z_s h_\mu vT)}{\partial\psi}\right\}+\frac{\partial(z_s\dot{s}T)}{\partial s}=-z_s\nabla\cdot\mathbf{F}_T+z_sQ^T, \tag{9.3}$$

where Q^T indicates the source or sink term for T. If the advection term is expressed as $\mathcal{A}(T)$, the horizontal diffusion term $\mathcal{D}_H(T)$, and the vertical diffusion term $\mathcal{D}_V(T)$,

$$\frac{\partial(z_sT)}{\partial t}=\mathcal{A}(T)+\mathcal{D}_H(T)+\mathcal{D}_V(T)+z_sQ^T \tag{9.4}$$

$$\mathcal{A}(T)=-\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi uT)}{\partial\mu}+\frac{\partial(z_s h_\mu vT)}{\partial\psi}\right\}-\frac{\partial(z_s\dot{s}T)}{\partial s} \tag{9.5}$$

$$\mathcal{D}_H(T)=-\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi z_sF^T_\mu)}{\partial\mu}+\frac{\partial(h_\mu z_sF^T_\psi)}{\partial\psi}\right\} \tag{9.6}$$

$$\mathcal{D}_V(T)=-\frac{\partial F^T_z}{\partial s}. \tag{9.7}$$

Chapter 10, 11 and 12 explain the numerical treatments of $\mathcal{A}(T)$, $\mathcal{D}_H(T)$ and $\mathcal{D}_V(T)$, respectively. It should be noted that horizontal diffusion and vertical diffusion are treated together in the derivation of isopycnal diffusion (Sec. 11.3) as

$$\mathcal{D}_(T)=\mathcal{D}_H(T)+\mathcal{D}_v(T). \tag{9.8}$$

(In this case as well, diffusion is divided into $\mathcal{D}_H(T)$ and $\mathcal{D}_V(T)$ in the program code.)

Since the hydrostatic approximation is used, an unstable stratification should be removed somehow. Generally, we assume that vertical convection occurs instantaneously to remove unstable stratification. We call this convective adjustment, which is explained in Section 12.2. One might also choose to mix tracers by setting the local vertical diffusion coefficient to a large value such as 10 000 cm^2 s^{-1} where stratification is unstable. In this case, the tracer equation should be solved using the partial implicit method, which is described in Section 23.5.

9.2 Boundary conditions

The boundary conditions of tracers can also be represented in common. The sea surface and bottom boundary conditions for θ and S are shown by Eqs. (2.23), (2.24) and (2.25). These can be generally expressed as follows:

$$\kappa_V \frac{\partial T}{\partial z}\bigg|_{z^*=0} = F^T_{\mathrm{surf}}, \tag{9.9}$$

$$-\kappa_V \frac{\partial T}{\partial z}\bigg|_{z^*=-H} = F^T_{\mathrm{bottom}}, \tag{9.10}$$

where F^T_{surf} is the surface flux (positive downward), while F^T_{bottom} is the bottom flux (positive upward). At side walls, the boundary condition (Eq. 2.26) is given for the n direction, which is perpendicular to the wall, as

$$\frac{\partial T}{\partial n} = 0. \tag{9.11}$$

9.3 Various tracers

Tracers governed by the advection-diffusion equation are not only θ and S. Passive tracers such as CFCs and age tracers (Chapter 20), and state variables of biogeochemical models (Chapter 19) are also tracers. Though time evolution of them is obtained by the common governing equation, Eq. (9.3), the boundary conditions and Q^T are different. In addition, appropriate numerical schemes for advection and diffusion can change depending on tracer characteristics. MRI.COM allows individual flexible specifications for various tracers in the model. See Chapter 13 for details.

Chapter 10

Tracer advection schemes

This chapter describes the following tracer advection schemes available in MRI.COM.

- The weighted upcurrent scheme (Section 10.2)
- The Quadratic Upstream Interpolation for Convective Kinematics (QUICK; Leonard, 1979, Section 10.3)
- A combination of the QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979, Section 10.3) for vertical advection and the Uniformly Third-Order Polynomial Interpolation Algorithm (UTOPIA; Leonard et al., 1993, Section 10.4) for horizontal advection
- The Second Order Moment (SOM; Prather, 1986, Section 10.5) scheme
- The Piecewise Parabolic Method (PPM; Colella and Woodward, 1984, Section 10.6) scheme
- The Multidimensional Positive Definite Advection Transport Algorithm (MPDATA; Smolarkiewicz and Margolin, 1998, Section 10.7) scheme

The default advection scheme for MRI.COM is a weighted upcurrent scheme, and the others are optional. Different advection schemes can be used for individual tracers. They should be chosen from among compiled schemes at run time (Chapter 13).

10.1 Finite volume or flux form method

Removing terms except for the time tendency and advection terms from Eq. (9.4), we gain the following three-dimensional advection equation:

$$\frac{\partial(z_s T)}{\partial t} = \mathcal{A}(T), \tag{10.1}$$

or

$$\frac{\partial(z_s T)}{\partial t} = -\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi u T)}{\partial \mu} + \frac{\partial(z_s h_\mu v T)}{\partial \psi}\right\} - \frac{\partial(z_s \dot{s} T)}{\partial s}. \tag{10.2}$$

The finite difference form of (10.1) is given by first considering a control cell volume and then calculating fluxes through the cell faces, and setting their divergence and convergence to be the time change rate at the grid cell (Figure 10.1). In finite difference form, this is expressed as follows:

$$\begin{aligned} T^{n+1}_{i,j,k-\frac{1}{2}} \Delta V^{n+1}_{i,j,k-\frac{1}{2}} &= T^{n}_{i,j,k-\frac{1}{2}} \Delta V^{n}_{i,j,k-\frac{1}{2}} \\ &+ \Delta t \left\{FXA_{i-\frac{1}{2},j,k-\frac{1}{2}} - FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} + FYA_{i,j-\frac{1}{2},k-\frac{1}{2}} - FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} + FZA_{i,j,k} - FZA_{i,j,k-1}\right\}, \end{aligned} \tag{10.3}$$

where ΔV is the volume of the grid cell, and FXA, FYA, and FZA represent (*flux due to advection*) × (*area of the cell boundary*). The same consideration may be applied for the discretization of the diffusion term.

As explained in Chapter 3, ΔV varies with time and is given by (3.28),

$$\begin{aligned} \Delta V_{i,j,k-\frac{1}{2}} \equiv (\texttt{volt})_{i,j,k-\frac{1}{2}} &= (\texttt{volu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\texttt{volu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \\ &+ (\texttt{volu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\texttt{volu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}. \end{aligned} \tag{10.4}$$

The volume of the left-lower quarter of the T-cells at (i, j) (corresponding to the southwestern part in geographic coordinates) is represented by $(\texttt{volt\_tr})_{i-\frac{1}{2},j-\frac{1}{2}}$. Similarly, $(\texttt{volt\_tl})_{i+\frac{1}{2},j-\frac{1}{2}}$ is the right-lower (southeastern), $(\texttt{volt\_br})_{i-\frac{1}{2},j+\frac{1}{2}}$ is the left-upper (northwestern), and $(\texttt{volt\_bl})_{i+\frac{1}{2},j+\frac{1}{2}}$ is the right-upper (northeastern) quarter of a T-cell at (i, j).

Fluxes due to advection are given as follows:

$$FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} = U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} T_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{10.5}$$

$$FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} = V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} T_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{10.6}$$

$$FZA_{i,j,k} = W^T_{i,j,k} T_{i,j,k}, \tag{10.7}$$

where horizontal volume transport U^T and V^T are defined as follows,

$$U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta y_{i+\frac{1}{2},j}}{2}\left(u_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}\Delta z_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right), \tag{10.8}$$

$$V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{\Delta x_{i,j+\frac{1}{2}}}{2}\left(v_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\Delta z_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + v_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right). \tag{10.9}$$

Vertical volume transport W^T is then obtained by diagnostically solving (6.9) with (6.11). Moreover, the vertical velocity w, which is necessary for using QUICKEST, is calculated as follows (w is not needed except for QUICKEST):

$$W^T_{i,j,k} = w_{i,j,k} \times (\texttt{areat})_{i,j,k-\frac{1}{2}}, \tag{10.10}$$

where

$$\begin{aligned}(\texttt{areat})_{i,j,k-\frac{1}{2}} =& (\texttt{a\_tr})_{i-\frac{1}{2},j-\frac{1}{2}} \times (\texttt{aexl})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + (\texttt{a\_tl})_{i+\frac{1}{2},j-\frac{1}{2}} \times (\texttt{aexl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \\ &+ (\texttt{a\_br})_{i-\frac{1}{2},j+\frac{1}{2}} \times (\texttt{aexl})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\texttt{a\_bl})_{i+\frac{1}{2},j+\frac{1}{2}} \times (\texttt{aexl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}.\end{aligned} \tag{10.11}$$

An array `aexl` is set to be unity if the corresponding U-cell is a sea cell and set to be zero otherwise.

10.2 Weighted upcurrent scheme

In the weighted upcurrent scheme (e.g., Suginohara and Aoki, 1991; Yamanaka et al., 2000), the cell boundary value is determined by a weighted average of the upcurrent scheme and the centered finite difference scheme.

The upcurrent scheme employs the upstream value as the cell boundary value:

$$T^{upcurrent}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \begin{cases} T_{i,j,k-\frac{1}{2}}, & \text{if} \quad U_{i+\frac{1}{2},j,k-\frac{1}{2}} > 0, \\ T_{i+1,j,k-\frac{1}{2}}, & \text{if} \quad U_{i+\frac{1}{2},j,k-\frac{1}{2}} < 0. \end{cases} \tag{10.12}$$

The centered finite difference scheme uses the average between the two neighboring points of tracer as the cell boundary value:

$$T^{center}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{T_{i+1,j,k-\frac{1}{2}} + T_{i,j,k-\frac{1}{2}}}{2}. \tag{10.13}$$

Taking the weight of the upcurrent scheme to be α ($0 \le \alpha \le 1$), the tracer flux at the eastern face of a grid cell is expressed as follows:

$$\begin{aligned} FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} =& \alpha\left\{\frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} + \left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}{2}T_i + \frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} - \left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}{2}T_{i+1}\right\} \\ &+ (1-\alpha)U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\frac{T_{i+1,j,k-\frac{1}{2}} + T_{i,j,k-\frac{1}{2}}}{2} && (10.14) \\ =& U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\left\{\frac{1}{2}\left(1 + \alpha\frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}\right)T_i + \frac{1}{2}\left(1 - \alpha\frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}\right)T_{i+1}\right\} && (10.15) \\ =& U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\left\{\left[\frac{1}{2} + \frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}\left(\beta - \frac{1}{2}\right)\right]T_i + \left[\frac{1}{2} - \frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}\left(\beta - \frac{1}{2}\right)\right]T_{i+1}\right\}. && (10.16) \end{aligned}$$

Mainly for the sake of computational efficiency, we give a parameter $\beta = \frac{1}{2}(\alpha + 1)$ ($0.5 \le \beta \le 1$) instead of α at run time. Different parameters for horizontal and vertical directions may be given, which are listed on Table 10.1.

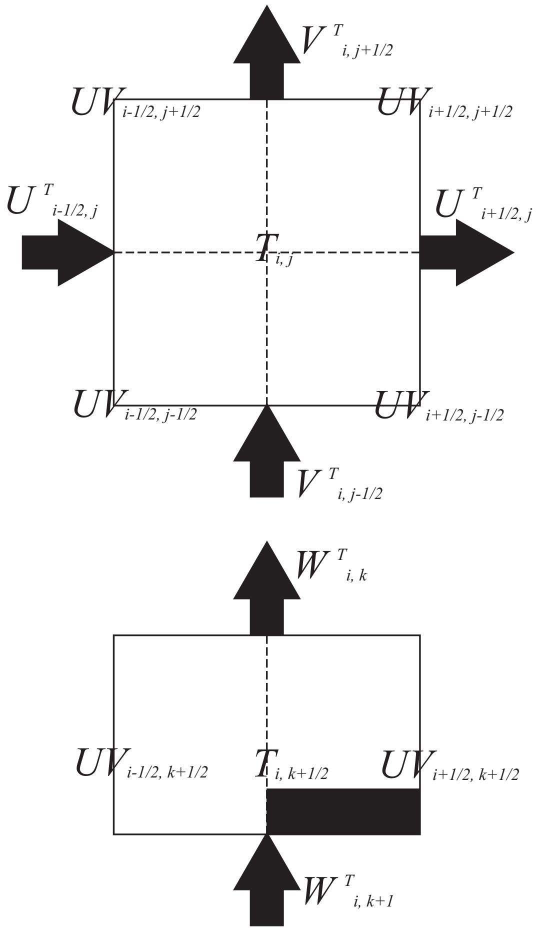

Figure10.1 Grid arrangement around TS-Box (Upper: Views from the upper, Below: views from the horizontal). Fluxes represented by an arrow are calculated.

Table10.1 Namelist `nml_tracer_adv` for weighted upcurrent scheme

variable name	units	description	usage
`weight_upcurrent_horz`	1	upstream-side weighting ratio for the horizontal advection	1.0: up-current, 0.5: centered difference
`weight_upcurrent_vert`	1	upstream-side weighting ratio for the vertical advection	1.0: up-current, 0.5: centered difference

10.3 QUICK and QUICKEST schemes

This section explains the QUICK and QUICKEST schemes. In the QUICK scheme, the cell boundary value is interpolated by a quadratic function, using three points, with one of them added from the upstream side (Figure 10.2).

Originally, cell boundary values in the QUICK scheme are given as follows:

$$T_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta x_i T_{i+1,j,k-\frac{1}{2}} + \Delta x_{i+1} T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+1} + \Delta x_i} - \frac{\Delta x_{i+1}\Delta x_i}{4} c_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{10.17}$$

$$T_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{\Delta y_j T_{i,j+1,k-\frac{1}{2}} + \Delta y_{j+1} T_{i,j,k-\frac{1}{2}}}{\Delta y_{j+1} + \Delta y_j} - \frac{\Delta y_{j+1}\Delta y_j}{4} d_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{10.18}$$

$$T_{i,j,k} = \frac{\Delta z_{k-\frac{1}{2}} T_{i,j+1,k+\frac{1}{2}} + \Delta z_{k+\frac{1}{2}} T_{i,j,k-\frac{1}{2}}}{\Delta z_{k+\frac{1}{2}} + \Delta z_{k-\frac{1}{2}}} - \frac{\Delta z_{k-\frac{1}{2}}\Delta z_{k+\frac{1}{2}}}{4} e_{i,j,k}, \tag{10.19}$$

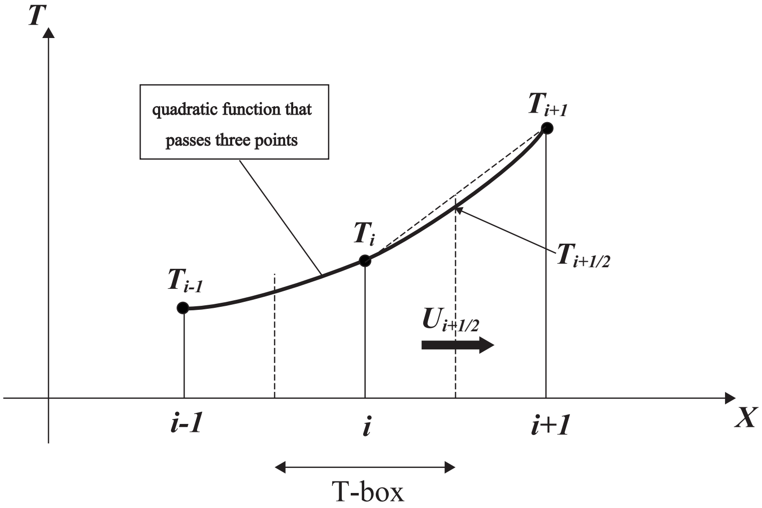

Figure10.2 Schematic for interpolation: T_i represents the value of a tracer at T-point with index i, and $T_{i+\frac{1}{2}}$ is the cell boundary value. In the QUICK scheme, $T_{i+\frac{1}{2}}$ is interpolated by a quadratic function that passes T_i and the neighboring T-point values, T_{i-1} and T_{i+1}.

where c, d, and e are defined depending on the direction of the mass flux as follows:

$$\begin{aligned}
c_{i+\frac{1}{2},j,k-\frac{1}{2}} &= \frac{\Delta x_i \delta_x \delta_x T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta x_i}^x} (\equiv c_p), && \text{if} \quad U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} > 0, \\
&= \frac{\Delta x_{i+1} \delta_x \delta_x T_{i+1,j,k-\frac{1}{2}}}{2\overline{\Delta x_{i+1}}^x} (\equiv c_m), && \text{if} \quad U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} < 0, \\
d_{i,j+\frac{1}{2},k-\frac{1}{2}} &= \frac{\Delta y_j \delta_y \delta_y T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta y_j}^y} (\equiv d_p), && \text{if} \quad V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} > 0, \\
&= \frac{\Delta y_{j+1} \delta_y \delta_y T_{i,j+1,k-\frac{1}{2}}}{2\overline{\Delta y_{j+1}}^y} (\equiv d_m), && \text{if} \quad V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} < 0, \\
e_{i,j,k} &= \frac{\Delta z_{k+\frac{1}{2}} \delta_z \delta_z T_{i,j,k+\frac{1}{2}}}{2\overline{\Delta z_{i,j,k+\frac{1}{2}}}^z} (\equiv e_p), && \text{if} \quad W^T_{i,j,k} > 0, \\
&= \frac{\Delta z_{k-\frac{1}{2}} \delta_z \delta_z T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta z_{i,j,k-\frac{1}{2}}}^z} (\equiv e_m), && \text{if} \quad W^T_{i,j,k} < 0.
\end{aligned} \tag{10.20}$$

The finite difference operators are defined as follows (definitions in y and z directions are the same):

$$\begin{aligned}
\delta_x A_i &\equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta x_i}, \quad \delta_x A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_i}{\Delta x_{i+\frac{1}{2}}}, \\
\overline{A_i}^x &\equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2}, \quad \overline{A_{i+\frac{1}{2}}}^x \equiv \frac{A_{i+1} + A_i}{2}.
\end{aligned} \tag{10.21}$$

Letting c_p, d_p, and e_p represent their values for positive velocity at the cell boundary and c_m, d_m, and e_m represent their values for negative velocity at cell boundary and taking

$$c_a = c_m + c_p \tag{10.22}$$

$$c_d = c_m - c_p \tag{10.23}$$

$$d_a = d_m + d_p \tag{10.24}$$

$$d_d = d_m - d_p \tag{10.25}$$

$$e_a = e_m + e_p \tag{10.26}$$

$$e_d = e_m - e_p, \tag{10.27}$$

we obtain

$$\begin{aligned}
FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} &= U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} \left[\frac{\Delta x_{i,j} T_{i+1,j,k-\frac{1}{2}} + \Delta x_{i+1,j} T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+1,j} + \Delta x_{i,j}} - \frac{\Delta x_{i+1,j} \Delta x_{i,j}}{8} c_{a\,i+\frac{1}{2},j,k-\frac{1}{2}} \right] \\
&+ |U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}| \frac{\Delta x_{i+1,j} \Delta x_{i,j}}{8} c_{d\,i+\frac{1}{2},j,k-\frac{1}{2}}, \qquad (10.28) \\
FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} &= V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} \left[\frac{\Delta y_{i,j} T_{i,j+1,k-\frac{1}{2}} + \Delta y_{i,j+1} T_{i,j,k-\frac{1}{2}}}{\Delta y_{i,j+1} + \Delta y_{i,j}} - \frac{\Delta y_{i,j+1} \Delta y_{i,j}}{8} d_{a\,i,j+\frac{1}{2},k-\frac{1}{2}} \right] \\
&+ |V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}| \frac{\Delta y_{i,j+1} \Delta y_{i,j}}{8} d_{d\,i,j+\frac{1}{2},k-\frac{1}{2}}, \qquad (10.29) \\
FZA_{i,j,k} &= W^T_{i,j,k} \left[\frac{\Delta z_{i,j,k-\frac{1}{2}} T_{i,j,k+\frac{1}{2}} + \Delta z_{i,j,k+\frac{1}{2}} T_{i,j,k-\frac{1}{2}}}{\Delta z_{i,j,k+\frac{1}{2}} + \Delta z_{i,j,k-\frac{1}{2}}} - \frac{\Delta z_{i,j,k+\frac{1}{2}} \Delta z_{i,j,k-\frac{1}{2}}}{8} e_{a\,i,j,k} \right] \\
&+ |W^T_{i,j,k-\frac{1}{2}}| \frac{\Delta z_{i,j,k+\frac{1}{2}} \Delta z_{i,j,k-\frac{1}{2}}}{8} e_{d\,i,j,k}. \qquad (10.30)
\end{aligned}$$

Equation (10.28) can be rewritten as

$$FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} \simeq U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} \tilde{T}_{i+\frac{1}{2},j,k-\frac{1}{2}} + A_Q \frac{\partial^3 T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\partial x^3}, \qquad (10.31)$$

where $\tilde{T}_{i+\frac{1}{2},j,k-\frac{1}{2}}$ is the value of T at the cell boundary interpolated by the cubic polynomial, and

$$A_Q = |U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}| \frac{\Delta x_{i+1} \Delta x_{i+\frac{1}{2}} \Delta x_i}{8}. \qquad (10.32)$$

Although the time integration for advection is done by the leap-frog scheme, the second term on the r.h.s. of (10.31) has a biharmonic diffusion form, and thus the forward scheme is used to achieve calculation stability (Holland et al., 1998).

A similar procedure is applied for the north-south and vertical directions.

The weighted up-current scheme is used for vertical direction if $w_{i,j,k-1} > 0$ and the T-point at $(i, j, k + \frac{1}{2})$ is below the bottom. The upstream-side weighting ratio is given by the user as the namelist parameter specified for the up-current scheme (Table 10.1).

The following paragraphs describe the specific expression and the accuracy of the QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979).

Consider a one-dimensional equation of advection for incompressible fluid

$$\frac{\partial T}{\partial t} + \frac{\partial}{\partial z}(wT) = 0, \qquad (10.33)$$

where w is a constant. Although the velocities are not uniform in the real three dimensional ocean, we assume a constant velocity for simplicity.

Following the notation of vertical grid points and their indices (Section 3.2), tracers are defined at the center $(k - \frac{1}{2})$ of the vertical cells and vertical velocities are defined at the top $(k - 1)$ and bottom (k) faces of the vertical cells. The following relation holds for the vertical grid spacing:

$$\Delta z_k = \frac{\Delta z_{k+\frac{1}{2}} + \Delta z_{k-\frac{1}{2}}}{2}. \qquad (10.34)$$

In QUICKEST, the distribution of tracer T is defined using the second order interpolations, and the mean value during a time step at the cell face (boundary of two adjacent tracer cells) is calculated. The coefficients for the second order interpolation are calculated first. A Taylor expansion of T about point z_k gives

$$T_{k-\frac{3}{2}} = c_0 + c_1 \left(\frac{\Delta z_{k-\frac{3}{2}}}{2} + \Delta z_{k-\frac{1}{2}}\right) + c_2 \left(\frac{\Delta z_{k-\frac{3}{2}}}{2} + \Delta z_{k-\frac{1}{2}}\right)^2 + O(\Delta z^3), \tag{10.35}$$

$$T_{k-\frac{1}{2}} = c_0 + c_1 \frac{\Delta z_{k-\frac{1}{2}}}{2} + c_2 \frac{\Delta z^2_{k-\frac{1}{2}}}{4} + O(\Delta z^3), \tag{10.36}$$

$$T_{k+\frac{1}{2}} = c_0 - c_1 \frac{\Delta z_{k+\frac{1}{2}}}{2} + c_2 \frac{\Delta z^2_{k+\frac{1}{2}}}{4} + O(\Delta z^3), \tag{10.37}$$

$$T_{k+\frac{3}{2}} = c_0 - c_1 \left(\frac{\Delta z_{k+\frac{3}{2}}}{2} + \Delta z_{k+\frac{1}{2}}\right) + c_2 \left(\frac{\Delta z_{k+\frac{3}{2}}}{2} + \Delta z_{k+\frac{1}{2}}\right)^2 + O(\Delta z^3). \tag{10.38}$$

Coefficients c_0, c_1, and c_2 can be solved using three of the four equations (10.35), (10.36), (10.37), and (10.38). The three upstream-side equations are chosen. When $w > 0$ ($w < 0$), equations (10.36), (10.37), and (10.38) ((10.35), (10.36), and (10.37)) are used. The solution is as follows:

$$c_0 = \frac{T_{k-\frac{1}{2}}\Delta z_{k+\frac{1}{2}} + T_{k+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}}{2\Delta z_k} - \frac{\Delta z_{k+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}}{4} c_2, \tag{10.39}$$

$$c_1 = \frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_k} - \frac{\Delta z_{k-\frac{1}{2}} - \Delta z_{k+\frac{1}{2}}}{2} c_2, \tag{10.40}$$

$$c_2 = \begin{cases} \frac{1}{\Delta z_k + \Delta z_{k+1}} \left(\frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_k} - \frac{T_{k+\frac{1}{2}} - T_{k+\frac{3}{2}}}{\Delta z_{k+1}} \right) & (w > 0), \\ \frac{1}{\Delta z_{k-1} + \Delta z_k} \left(\frac{T_{k-\frac{3}{2}} - T_{k-\frac{1}{2}}}{\Delta z_{k-1}} - \frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_k} \right) & (w < 0). \end{cases} \tag{10.41}$$

Next, equation (10.33) is integrated over one time step and one grid cell.

$$\int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz \frac{\partial T}{\partial t} = - \int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz \frac{\partial}{\partial z}(wT). \tag{10.42}$$

The r.h.s. of (10.42) can be written as

$$- \int_{t^n}^{t^{n+1}} dt (w_u T_u - w_l T_l), \tag{10.43}$$

where subscript u (l) denotes $z = z_{k-1}$ ($z = z_k$). Assuming that w does not depend on time,

$$\int_{t^n}^{t^{n+1}} dt T_l = \int_{-w_l \Delta t}^{0} [c_0^n + c_1^n \xi + c_2^n \xi^2 + O(\Delta z^3)] \frac{d\xi}{w_l}. \tag{10.44}$$

Thus expression (10.43) becomes

$$-\Delta t (w_u \overline{T_u^n} - w_l \overline{T_l^n}) + O(\Delta z^3 w \Delta t), \tag{10.45}$$

where

$$\begin{aligned} \overline{T_l^n} &= \frac{1}{w_l \Delta t} \int_{-w_l \Delta t}^{0} (c_0^n + c_1^n \xi + c_2^n \xi^2) d\xi \\ &= c_0^n - \frac{c_1^n}{2} w_l \Delta t + \frac{c_2^n}{3} w_l^2 \Delta t^2. \end{aligned} \tag{10.46}$$

Using up to the second order terms of a Taylor expansion, the l.h.s. of (10.42) can be written as follows:

$$\int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz \frac{\partial T}{\partial t} = \Delta z_{k-\frac{1}{2}} \left[T^{n+1}_{k-\frac{1}{2}} - T^n_{k-\frac{1}{2}} + \frac{\Delta z^2_{k-\frac{1}{2}}}{24} ({T_{zz}}^{n+1}_{k-\frac{1}{2}} - {T_{zz}}^{n}_{k-\frac{1}{2}}) + O(\Delta z^3) \right], \tag{10.47}$$

where

$$
\begin{aligned}
{T_{zz}}^{n+1}_{k-\frac{1}{2}} - {T_{zz}}^{n}_{k-\frac{1}{2}} &= \Delta t \left.\frac{\partial T_{zz}}{\partial t}\right|^{n}_{k-\frac{1}{2}} + O(\Delta t^2) \\
&= -\Delta t \frac{\partial^2}{\partial z^2}\left[\frac{\partial}{\partial z}(wT)\right]^{n}_{k-\frac{1}{2}} + O(\Delta t^2) \\
&= -\Delta t \frac{\partial}{\partial z}(wT_{zz})^{n}_{k-\frac{1}{2}} + O(\Delta t^2) \\
&= -\Delta t \left\{ \frac{w_u {T_{zz}}^{n}_{u} - w_l {T_{zz}}^{n}_{l}}{\Delta z_{k-\frac{1}{2}}} \right\} + O(w\Delta t\Delta z) + O(\Delta t^2). \qquad (10.48)
\end{aligned}
$$

The expression for the r.h.s. of (10.47) becomes

$$
\Delta z_{k-\frac{1}{2}} \left[T^{n+1}_{k-\frac{1}{2}} - T^{n}_{k-\frac{1}{2}} - \frac{\Delta z^2_{k-\frac{1}{2}}}{24} \Delta t \frac{w_u {T_{zz}}^{n}_{u} - w_l {T_{zz}}^{n}_{l}}{\Delta z_{k-\frac{1}{2}}} + O(\Delta z^3) \right] + O(w\Delta t\Delta z^4) + O(\Delta z^3 \Delta t^2). \qquad (10.49)
$$

Based on (10.45) and (10.49), the discretized forecasting equation is expressed as follows:

$$
T^{n+1}_{k-\frac{1}{2}} = T^{n}_{k-\frac{1}{2}} - \frac{\Delta t}{\Delta z_{k-\frac{1}{2}}} \left[w_u \overline{T^n_u} - w_l \overline{T^n_l} - \frac{\Delta z^2_{k-\frac{1}{2}}}{24} (w_u {T_{zz}}^{n}_{u} - w_l {T_{zz}}^{n}_{l}) \right] + O(\alpha \Delta z^4) + O(\Delta z^2 \Delta t^2), \qquad (10.50)
$$

where

$$
\begin{aligned}
\alpha &\equiv \frac{w\Delta t}{\Delta z} < 1, \qquad (10.51) \\
{T_{zz}}^{n}_{l} &= 2c_2 + O(\Delta z). \qquad (10.52)
\end{aligned}
$$

The accuracy of equation (10.50) is $\max(O(\Delta z^4), O(\Delta z^2 \Delta t^2))$.

Note that MRI.COM adopts a flux limiter proposed by Leonard et al. (1994), which prevents unrealistic extrema, for the vertical QUICKEST scheme.

10.4 UTOPIA for horizontal advection

The Uniformly Third Order Polynomial Interpolation Algorithm (UTOPIA; Leonard et al., 1993) is an advection scheme that can be regarded as a multi-dimensional version of QUICKEST. In UTZQADVEC option, horizontally two-dimensional advection is calculated using UTOPIA. Vertical advection is calculated separately using QUICKEST.

Since grid intervals could be variable in both zonal and meridional directions in MRI.COM, UTOPIA is formulated based on a variable grid interval. It is assumed that the tracer cell is subdivided by the borderlines of the velocity cells into four boxes with (almost) identical area.

Consider an equation of advection:

$$
\frac{\partial T}{\partial t} + \frac{1}{h_\mu h_\psi}\frac{\partial}{\partial \mu}(h_\psi u T) + \frac{1}{h_\mu h_\psi}\frac{\partial}{\partial \psi}(h_\mu v T) = 0. \qquad (10.53)
$$

Integrated over a tracer cell and for one time step,

$$
\int_{\psi_L-\Delta\psi_L/2}^{\psi_L+\Delta\psi_L/2} d\psi \int_{\mu_L-\Delta\mu_L/2}^{\mu_L+\Delta\mu_L/2} d\mu (\chi^{n+1} - \chi^n) = -\Delta t (u^n_r \overline{T^n_r} \Delta y_r - u^n_l \overline{T^n_l} \Delta y_l + v^n_u \overline{T^n_u} \Delta x_u - v^n_d \overline{T^n_d} \Delta x_d), \qquad (10.54)
$$

where $\chi \equiv h_\mu h_\psi T$ and $\overline{T^n_r}$ etc. on the r.h.s. are the face values described later. On the l.h.s. of (10.54), the second-order interpolation of χ is used to integrate the terms. The Taylor expansion of χ about L is given as follows (see Figure 10.3 for the label of the point):

$$
\chi = \chi_L + a_{10}(\mu - \mu_L) + a_{20}(\mu - \mu_L)^2 + a_{01}(\psi - \psi_L) + a_{02}(\psi - \psi_L)^2 + a_{11}(\mu - \mu_L)(\psi - \psi_L). \qquad (10.55)
$$

Then values at points E, W, N, and S are

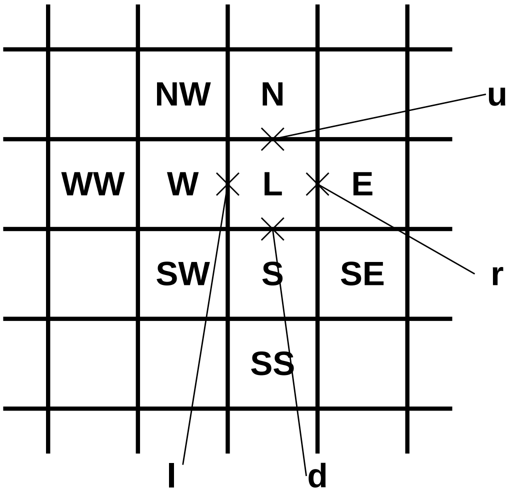

Figure10.3 Labels of tracer grid points (upper case characters) and faces (lower case characters).

$$\chi_E = \chi_L + a_{10}\Delta\mu_r + a_{20}\Delta\mu_r^2, \tag{10.56}$$

$$\chi_W = \chi_L - a_{10}\Delta\mu_l + a_{20}\Delta\mu_l^2, \tag{10.57}$$

$$\chi_N = \chi_L + a_{01}\Delta\psi_u + a_{02}\Delta\psi_u^2, \tag{10.58}$$

$$\chi_S = \chi_L - a_{01}\Delta\psi_d + a_{02}\Delta\psi_d^2, \tag{10.59}$$

where

$$\Delta\psi_u \equiv \frac{\Delta\psi_L + \Delta\psi_N}{2}, \tag{10.60}$$

$$\Delta\psi_d \equiv \frac{\Delta\psi_L + \Delta\psi_S}{2}, \tag{10.61}$$

$$\Delta\mu_r \equiv \frac{\Delta\mu_L + \Delta\mu_E}{2}, \tag{10.62}$$

$$\Delta\mu_l \equiv \frac{\Delta\mu_L + \Delta\mu_W}{2}. \tag{10.63}$$

Using these known values, the following parameters are obtained,

$$a_{10} = \frac{\Delta\mu_l \dfrac{\chi_E - \chi_L}{\Delta\mu_r} + \Delta\mu_r \dfrac{\chi_L - \chi_W}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}, \tag{10.64}$$

$$a_{20} = \frac{\dfrac{\chi_E - \chi_L}{\Delta\mu_r} - \dfrac{\chi_L - \chi_W}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}, \tag{10.65}$$

$$a_{01} = \frac{\Delta\psi_d \dfrac{\chi_N - \chi_L}{\Delta\psi_u} + \Delta\psi_u \dfrac{\chi_L - \chi_S}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}, \tag{10.66}$$

$$a_{02} = \frac{\dfrac{\chi_N - \chi_L}{\Delta\psi_u} - \dfrac{\chi_L - \chi_S}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{10.67}$$

Substituting (10.55) into the l.h.s. of (10.54) yields

$$\Delta\mu_L\Delta\psi_L\left[\chi_L^{n+1} - \chi_L^n + \frac{\Delta\mu_L^2}{12}(a_{20}^{n+1} - a_{20}^n) + \frac{\Delta\psi_L^2}{12}(a_{02}^{n+1} - a_{02}^n)\right]. \tag{10.68}$$

Using equation (10.53), the following approximation is allowed:

$$a_{20}^{n+1} - a_{20}^{n} = -\Delta t \left[\frac{h_{\psi r} u_r^n T_{\mu\mu r}^n - h_{\psi l} u_l^n T_{\mu\mu l}^n}{\Delta\mu_L} + \frac{h_{\mu u} v_u^n T_{\mu\mu u}^n - h_{\mu d} v_d^n T_{\mu\mu d}^n}{\Delta\psi_L} \right], \tag{10.69}$$

$$a_{02}^{n+1} - a_{02}^{n} = -\Delta t \left[\frac{h_{\psi r} u_r^n T_{\psi\psi r}^n - h_{\psi l} u_l^n T_{\psi\psi l}^n}{\Delta\mu_L} + \frac{h_{\mu u} v_u^n T_{\psi\psi u}^n - h_{\mu d} v_d^n T_{\psi\psi d}^n}{\Delta\psi_L} \right], \tag{10.70}$$

where $T_{\mu\mu r}^n$ is the value of the second order derivative at the right face **r**, whose expression is similar to that of c_{20} described later. Therefore, under a suitable approximation,

$$T_L^{n+1} = T_L^n - \frac{\Delta t}{\Delta S_L}(u_r^n \tilde{T_r^n} \Delta y_r - u_l^n \tilde{T_l^n} \Delta y_l + v_u^n \tilde{T_u^n} \Delta x_u - v_d^n \tilde{T_d^n} \Delta x_d), \tag{10.71}$$

where

$$\tilde{T_l^n} = \overline{T_l^n} - \frac{\Delta\mu_L^2}{24} T_{\mu\mu l}^n - \frac{\Delta\psi_L^2}{24} T_{\psi\psi l}^n, \tag{10.72}$$

$$\tilde{T_d^n} = \overline{T_d^n} - \frac{\Delta\mu_L^2}{24} T_{\mu\mu d}^n - \frac{\Delta\psi_L^2}{24} T_{\psi\psi d}^n. \tag{10.73}$$

Next, the expressions for $\overline{T_l^n}$ and $\overline{T_d^n}$ are required. The term $\overline{T_l^n}$ is the average over the hatched area of Figure 10.4, and the values of T^n are given as the second order interpolation about **l** of Figure 10.3. Similar operations will be used to obtain the expression for $\overline{T_d^n}$.

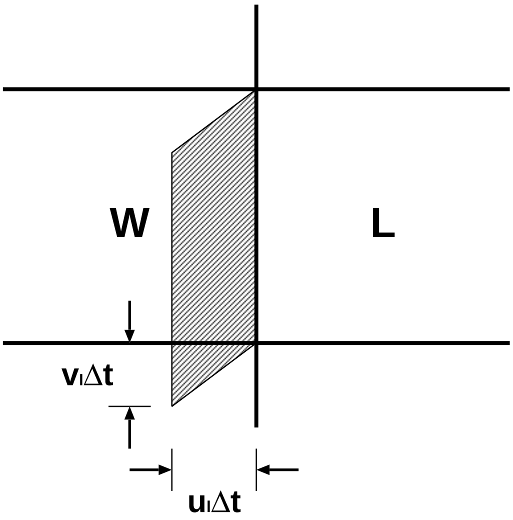

Figure10.4 Area used to average tracer values for the face **l**

First, Taylor expansions of T^n about **l** and **d** are written as follows:

$$T^n|_{\mathbf{l}} = c_{00} + c_{10}(\mu - \mu_l) + c_{20}(\mu - \mu_l)^2 + c_{01}(\psi - \psi_L) + c_{02}(\psi - \psi_L)^2 + c_{11}(\mu - \mu_l)(\psi - \psi_L), \tag{10.74}$$

$$T^n|_{\mathbf{d}} = d_{00} + d_{10}(\mu - \mu_l) + d_{20}(\mu - \mu_l)^2 + d_{01}(\psi - \psi_L) + d_{02}(\psi - \psi_L)^2 + d_{11}(\mu - \mu_l)(\psi - \psi_L). \tag{10.75}$$

The T^n values at eight points around **l** are,

$$T_L^n = c_{00} + c_{10}\frac{\Delta\mu_L}{2} + c_{20}\frac{\Delta\mu_L^2}{4}, \tag{10.76}$$

$$T_W^n = c_{00} - c_{10}\frac{\Delta\mu_W}{2} + c_{20}\frac{\Delta\mu_W^2}{4}, \tag{10.77}$$

$$T_E^n = c_{00} + c_{10}\left(\Delta\mu_L + \frac{\Delta\mu_E}{2}\right) + c_{20}\left(\Delta\mu_L + \frac{\Delta\mu_E}{2}\right)^2, \tag{10.78}$$

$$T_{WW}^n = c_{00} - c_{10}\left(\Delta\mu_W + \frac{\Delta\mu_{WW}}{2}\right) + c_{20}\left(\Delta\mu_W + \frac{\Delta\mu_{WW}}{2}\right)^2, \tag{10.79}$$

$$T_N^n = T_L^n + c_{01}\Delta\psi_u + c_{02}\Delta\psi_u^2 + c_{11}\frac{\Delta\mu_L}{2}\Delta\psi_u, \tag{10.80}$$

$$T_S^n = T_L^n - c_{01}\Delta\psi_d + c_{02}\Delta\psi_d^2 - c_{11}\frac{\Delta\mu_L}{2}\Delta\psi_d, \tag{10.81}$$

$$T_{NW}^n = T_W^n + c_{01}\Delta\psi_u + c_{02}\Delta\psi_u^2 - c_{11}\frac{\Delta\mu_W}{2}\Delta\psi_u, \tag{10.82}$$

$$T_{SW}^n = T_W^n - c_{01}\Delta\psi_d + c_{02}\Delta\psi_d^2 + c_{11}\frac{\Delta\mu_W}{2}\Delta\psi_d. \tag{10.83}$$

To obtain all six coefficients, six of these equations (points) are used. The equations are chosen according to the following flow direction.

$$u_l^n > 0, \quad v_l^n > 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{WW}, \mathbf{S}, \mathbf{NW}, \mathbf{SW} \tag{10.84}$$

$$u_l^n < 0, \quad v_l^n > 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{E}, \mathbf{N}, \mathbf{S}, \mathbf{SW} \tag{10.85}$$

$$u_l^n > 0, \quad v_l^n < 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{WW}, \mathbf{N}, \mathbf{NW}, \mathbf{SW} \tag{10.86}$$

$$u_l^n < 0, \quad v_l^n < 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{E}, \mathbf{N}, \mathbf{S}, \mathbf{NW} \tag{10.87}$$

From equations (10.76) and (10.77),

$$c_{00} = \frac{\Delta\mu_W T_L^n + \Delta\mu_L T_W^n}{2\Delta\mu_l} - c_{20}\frac{\Delta\mu_L\Delta\mu_W}{4}, \tag{10.88}$$

$$c_{10} = \frac{T_L^n - T_W^n}{\Delta\mu_l} - c_{20}\frac{\Delta\mu_L - \Delta\mu_W}{2}. \tag{10.89}$$

When $u_l^n > 0$, from (10.76) and (10.79),

$$c_{20} = \frac{\dfrac{T_L^n - T_W^n}{\Delta\mu_l} - \dfrac{T_W^n - T_{WW}^n}{\Delta\mu_{ll}}}{\Delta\mu_l + \Delta\mu_{ll}}, \tag{10.90}$$

$$\text{where} \quad \Delta\mu_{ll} \equiv \frac{\Delta\mu_W + \Delta\mu_{WW}}{2}.$$

Using equations (10.82) and (10.83),

$$c_{02} = \frac{\dfrac{T_{NW}^n - T_W^n}{\Delta\psi_u} - \dfrac{T_W^n - T_{SW}^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{10.91}$$

When $u_l^n < 0$, from (10.77) and (10.78),

$$c_{20} = \frac{\dfrac{T_E^n - T_L^n}{\Delta\mu_r} - \dfrac{T_L^n - T_W^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{10.92}$$

Using equations (10.80) and (10.81),

$$c_{02} = \frac{\dfrac{T_N^n - T_L^n}{\Delta\psi_u} - \dfrac{T_L^n - T_S^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{10.93}$$

When $v_l^n > 0$, from (10.81) and (10.83),

$$c_{01} = \frac{\Delta\mu_W(T_L^n - T_S^n) + \Delta\mu_L(T_W^n - T_{SW}^n)}{2\Delta\mu_l\Delta\psi_d} + c_{02}\Delta\psi_d, \tag{10.94}$$

$$c_{11} = \frac{T_{SW}^n - T_W^n - T_S^n + T_L^n}{\Delta\mu_l\Delta\psi_d}. \tag{10.95}$$

When $v_l^n < 0$, from (10.80) and (10.82),

$$c_{01} = \frac{\Delta\mu_W(T_N^n - T_L^n) + \Delta\mu_L(T_{NW}^n - T_W^n)}{2\Delta\mu_l\Delta\psi_u} - c_{02}\Delta\psi_u, \tag{10.96}$$

$$c_{11} = \frac{T_N^n - T_L^n - T_{NW}^n + T_W^n}{\Delta\mu_l\Delta\psi_u}. \tag{10.97}$$

Next, using equation (10.75), the T^n values at eight points around $\mathbf{d}$ are

$$T_L^n = d_{00} + d_{01}\frac{\Delta\psi_L}{2} + d_{02}\frac{\Delta\psi_L^2}{4}, \tag{10.98}$$

$$T_S^n = d_{00} - d_{01}\frac{\Delta\psi_S}{2} + d_{02}\frac{\Delta\psi_S^2}{4}, \tag{10.99}$$

$$T_N^n = d_{00} + d_{01}\left(\Delta\psi_L + \frac{\Delta\psi_N}{2}\right) + d_{02}\left(\Delta\psi_L + \frac{\Delta\psi_N}{2}\right)^2, \tag{10.100}$$

$$T_{SS}^n = d_{00} - d_{01}\left(\Delta\psi_S + \frac{\Delta\psi_{SS}}{2}\right) + d_{02}\left(\Delta\psi_S + \frac{\Delta\psi_{SS}}{2}\right)^2, \tag{10.101}$$

$$T_E^n = T_L^n + d_{10}\Delta\mu_r + d_{20}\Delta\mu_r^2 + d_{11}\frac{\Delta\psi_L}{2}\Delta\mu_r, \tag{10.102}$$

$$T_W^n = T_L^n - d_{10}\Delta\mu_l + d_{20}\Delta\mu_l^2 - d_{11}\frac{\Delta\psi_L}{2}\Delta\mu_l, \tag{10.103}$$

$$T_{SE}^n = T_S^n + d_{10}\Delta\mu_r + d_{20}\Delta\mu_r^2 - d_{11}\frac{\Delta\psi_S}{2}\Delta\mu_r, \tag{10.104}$$

$$T_{SW}^n = T_S^n - d_{10}\Delta\mu_l + d_{20}\Delta\mu_l^2 + d_{11}\frac{\Delta\psi_S}{2}\Delta\mu_l. \tag{10.105}$$

From equations (10.98) and (10.99),

$$d_{00} = \frac{\Delta\psi_S T_L^n + \Delta\psi_L T_S^n}{2\Delta\psi_d} - d_{02}\frac{\Delta\psi_L\Delta\psi_S}{4}, \tag{10.106}$$

$$d_{01} = \frac{T_L^n - T_S^n}{\Delta\psi_d} - d_{02}\frac{\Delta\psi_L - \Delta\psi_S}{2}. \tag{10.107}$$

When $v_d^n > 0$, from (10.98) and (10.101),

$$d_{02} = \frac{\dfrac{T_L^n - T_S^n}{\Delta\psi_d} - \dfrac{T_S^n - T_{SS}^n}{\Delta\psi_{dd}}}{\Delta\psi_d + \Delta\psi_{dd}}, \quad \text{where} \quad \Delta\psi_{dd} \equiv \frac{\Delta\psi_S + \Delta\psi_{SS}}{2}. \tag{10.108}$$

From (10.104) and (10.105),

$$d_{20} = \frac{\dfrac{T_{SE}^n - T_S^n}{\Delta\mu_r} - \dfrac{T_S^n - T_{SW}^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{10.109}$$

When $v_d^n < 0$, from (10.99) and (10.100),

$$d_{02} = \frac{\dfrac{T_N^n - T_L}{\Delta\psi_u} - \dfrac{T_L^n - T_S^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{10.110}$$

From (10.102) and (10.103),

$$d_{20} = \frac{\dfrac{T_E^n - T_L^n}{\Delta\mu_r} - \dfrac{T_L^n - T_W^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{10.111}$$

When $u_d^n > 0$, from (10.103) and (10.105),

$$\begin{aligned} d_{10} &= \frac{\Delta\psi_S(T_L^n - T_W^n) + \Delta\psi_L(T_S^n - T_{SW}^n)}{2\Delta\psi_d\Delta\mu_l} + d_{20}\Delta\mu_l, &(10.112)\\ d_{11} &= \frac{T_L^n - T_S^n - T_W^n + T_{SW}^n}{\Delta\psi_d\Delta\mu_l}. &(10.113)\end{aligned}$$

When $u_d^n < 0$, from (10.102) and (10.104),

$$\begin{aligned} d_{10} &= \frac{\Delta\psi_S(T_E^n - T_L^n) + \Delta\psi_L(T_{SE}^n - T_S^n)}{2\Delta\psi_d\Delta\mu_r} - d_{20}\Delta\mu_r, &(10.114)\\ d_{11} &= \frac{T_E^n - T_{SE}^n - T_L^n + T_S^n}{\Delta\psi_d\Delta\mu_r}. &(10.115)\end{aligned}$$

The value of $\overline{T_l^n}$ is the average of T^n over the hatched area of Figure 10.4. Defining

$$\xi_l^n = \frac{u_l^n}{h_{\mu l}}, \quad \eta_l^n = \frac{v_l^n}{h_{\psi l}}, \tag{10.116}$$

we have

$$\begin{aligned} \overline{T_l^n} = &\frac{1}{\xi_l^n \Delta t \Delta\psi_L}\left[\int_{\psi_L-\Delta\psi_L/2}^{\psi_L+\Delta\psi_L/2}\int_{\mu_l-\xi_l^n\Delta t}^{\mu_l} T^n d\mu d\psi\right.\\ &\left.+ \int_{\psi_L-\Delta\psi_L/2-\eta_l^n\Delta t}^{\psi_L-\Delta\psi_L/2}\int_{\mu_l-\xi_l^n\Delta t}^{\mu_l+\frac{\xi_l^n}{\eta_l^n}(\psi-\psi_L+\Delta\psi_L/2)} \{T^n(\psi) - T^n(\psi+\Delta\psi_L)\} d\mu d\psi\right]\\ = &c_{00} - \frac{1}{2}\eta_l^n\Delta t\, c_{01} + \left[\frac{1}{12}\Delta\psi_L^2 + \frac{1}{3}(\eta_l^n\Delta t)^2\right] c_{02}\\ &- \frac{1}{2}\xi_l^n\Delta t\, c_{10} + \frac{1}{3}(\xi_l^n\Delta t)^2 c_{20} + \frac{1}{3}\xi_l^n\eta_l^n\Delta t^2\, c_{11}. \end{aligned} \tag{10.117}$$

Although the ranges of integrals in (10.117) assume $u_l^n > 0$ and $v_l^n > 0$, the result is independent of the signs of u_l^n and v_l^n.

Similarly,

$$\begin{aligned} \overline{T_d^n} = &d_{00} - \frac{1}{2}\eta_d^n\Delta t\, d_{01} + \frac{1}{3}(\eta_d^n\Delta t)^2\, d_{02}\\ &- \frac{1}{2}\xi_d^n\Delta t\, d_{10} + \left[\frac{1}{12}\Delta\mu_L^2 + \frac{1}{3}(\xi_d^n\Delta t)^2\right] d_{20} + \frac{1}{3}\xi_d^n\eta_d^n\Delta t^2 d_{11}, \end{aligned} \tag{10.118}$$

where

$$\xi_d^n = \frac{u_d^n}{h_{\mu d}}, \quad \eta_d^n = \frac{v_d^n}{h_{\psi d}}. \tag{10.119}$$

Therefore,

$$\begin{aligned} \tilde{T_l^n} = &c_{00} - \frac{1}{2}\xi_l^n\Delta t\, c_{10} + \left[\frac{1}{3}(\xi_l^n\Delta t)^2 - \frac{\Delta\mu_L^2}{12}\right] c_{20}\\ &- \frac{1}{2}\eta_l^n\Delta t\, c_{01} + \frac{1}{3}(\eta_l^n\Delta t)^2 c_{02} + \frac{1}{3}\xi_l^n\eta_l^n\Delta t^2\, c_{11}, &(10.120)\\ \tilde{T_d^n} = &d_{00} - \frac{1}{2}\eta_d^n\Delta t\, d_{01} + \left[\frac{1}{3}(\eta_d^n\Delta t)^2 - \frac{\Delta\psi_L^2}{12}\right] d_{02}\\ &- \frac{1}{2}\xi_d^n\Delta t\, d_{10} + \frac{1}{3}(\xi_d^n\Delta t)^2 d_{20} + \frac{1}{3}\xi_d^n\eta_d^n\Delta t^2\, d_{11}. &(10.121)\end{aligned}$$

Finally, we describe how to derive the boundary conditions. Since the face values of the tracers are calculated through the second order interpolation, the value of a tracer at a point over land is sometimes necessary. For that case, the value should be appropriately decided by using the tracer values at the neighboring points in the sea. Since ocean models generally assume that there is no flux of tracers across land-sea boundary, the provisional value over land should be given so as not to create a normal gradient at the boundary.

When the face value of a tracer at boundary **l** is calculated, **W** and **L** are not land, but either **N** or **S** may be land, and either **NW** or **SW** may be land. When **N** or **S** is land, the land-sea boundary runs at the center of **L** in the zonal direction. It is reasonable to assume that the value of land grid **N** or **S** must not cause any meridional tracer gradient at **L** set by second order interpolation using the values at grids **N**, **L**, and **S**. Thus, we set

$$(T_N^n - T_L^n)\Delta\psi_d^2 = (T_S^n - T_L^n)\Delta\psi_u^2. \tag{10.122}$$

When **NW** or **SW** is a land grid, the following should be assumed.

$$(T_{NW}^n - T_W^n)\Delta\psi_d^2 = (T_{SW}^n - T_W^n)\Delta\psi_u^2 \tag{10.123}$$

When **WW** is a land grid,

$$(T_{WW}^n - T_W^n)\Delta\mu_l^2 = (T_L^n - T_W^n)\Delta\mu_{ll}^2. \tag{10.124}$$

When **E** is a land grid,

$$(T_E^n - T_L^n)\Delta\mu_l^2 = (T_W^n - T_L^n)\Delta\mu_r^2. \tag{10.125}$$

Similar boundary conditions are specified for face **d**.

Note that MRI.COM adopts a flux limiter proposed by Leonard et al. (1994), which prevents unrealistic extrema, for the UTOPIA scheme.

10.5 Second Order Moment (SOM) scheme

10.5.1 Outline

The Second Order Moment (SOM) advection scheme by Prather (1986) seeks to improve the accuracy by treating the tracer distribution within a grid cell, unlike the scheme that aims to calculate the tracer flux at the boundary of grid cells with high accuracy. It is assumed that the distribution of tracer f in a grid cell ($0 \le x \le X,\quad 0 \le y \le Y,\quad 0 \le z \le Z$; volume $V = XYZ$) can be represented using second order functions as follows:

$$f(x,y,z) = a_0 + a_x x + a_{xx}x^2 + a_y y + a_{yy}y^2 + a_z z + a_{zz}z^2 + a_{xy}xy + a_{yz}yz + a_{zx}zx. \tag{10.126}$$

Prather (1986) expressed the above as a sum of orthogonal functions $K_n(x,y,z)$;

$$f(x,y,z) = m_0K_0 + m_xK_x + m_{xx}K_{xx} + m_yK_y + m_{yy}K_{yy} + m_zK_z + m_{zz}K_{zz} + m_{xy}K_{xy} + m_{yz}K_{yz} + m_{zx}K_{zx}, \tag{10.127}$$

where the orthogonal functions are given as follows:

$$\begin{aligned}
K_0 &= 1,\\
K_x(x) &= x - X/2,\\
K_{xx}(x) &= x^2 - Xx + X^2/6,\\
K_y(y) &= y - Y/2,\\
K_{yy}(y) &= y^2 - Yy + Y^2/6,\\
K_z(z) &= z - Z/2,\\
K_{zz}(z) &= z^2 - Zz + Z^2/6,\\
K_{xy}(x,y) &= (x - X/2)(y - Y/2),\\
K_{yz}(y,z) &= (y - Y/2)(z - Z/2),\\
K_{zx}(z,x) &= (z - Z/2)(x - X/2),
\end{aligned} \tag{10.128}$$

and

$$\int K_mK_n dV = 0 \quad (m \ne n). \tag{10.129}$$

The constants for normalization are determined using

$$\int K_x^2 dV = VX^2/12, \quad \int K_{xx}^2 dV = VX^4/180,$$

$$\int K_y^2 dV = VY^2/12, \quad \int K_{yy}^2 dV = VY^4/180,$$

$$\int K_z^2 dV = VZ^2/12, \quad \int K_{zz}^2 dV = VZ^4/180,$$

$$\int K_{xy}^2 dV = VX^2Y^2/144, \quad \int K_{yz}^2 dV = VY^2Z^2/144, \quad \int K_{zx}^2 dV = VZ^2X^2/144.$$

Accordingly, the moments are set by the following expressions:

$$\begin{aligned}
S_0 &= \int f(x,y,z) K_0 dV = m_0 V, \\
S_x &= (6/X) \int f(x,y,z) K_x(x) dV = m_x VX/2, \\
S_{xx} &= (30/X^2) \int f(x,y,z) K_{xx}(x) dV = m_{xx} VX^2/6, \\
S_y &= (6/Y) \int f(x,y,z) K_y(y) dV = m_y VY/2, \\
S_{yy} &= (30/Y^2) \int f(x,y,z) K_{yy}(y) dV = m_{yy} VY^2/6, \\
S_z &= (6/Z) \int f(x,y,z) K_z(z) dV = m_z VZ/2, \\
S_{zz} &= (30/Z^2) \int f(x,y,z) K_{zz}(x) dV = m_{zz} VZ^2/6, \\
S_{xy} &= (36/XY) \int f(x,y,z) K_{xy}(x,y) dV = m_{xy} VXY/4, \\
S_{yz} &= (36/YZ) \int f(x,y,z) K_{yz}(y,z) dV = m_{yz} VYZ/4, \\
S_{zx} &= (36/ZX) \int f(x,y,z) K_{zx}(z,x) dV = m_{zx} VZX/4.
\end{aligned} \tag{10.130}$$

All these moments are transported with the upstream advection scheme. The procedure is carried out in one direction at a time. The second and third direction procedures use the results of the last procedure. For simplicity, we describe the change of each moment caused by an advection procedure in one direction (x) in a two dimensional plane (xy) in the following. You may replace (y, Y) with (z, Z) to derive a full set of formulas in the three dimensional space.

When velocity c in the x direction is positive, the right part of the grid cell,

$$X - ct \le x \le X, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{10.131}$$

is removed from the cell and added to the adjacent cell on the right during time interval t. This transported part is expressed using superscript T. The remaining part,

$$0 \le x \le X - ct, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{10.132}$$

is expressed by superscript R. New orthogonal functions K_n^T (K_n^R) are calculated in the part T (R) with the volume $V^T = ctYZ$ ($V^R = (X - ct)YZ$). The orthogonal functions are given as follows:

$$K_0^T = K_0^R = 1,$$

$$K_x^T = x - (2X - ct)/2, \quad K_x^R = x - (X - ct)/2,$$

$$K_{xx}^T = x^2 - (2X - ct)x + (X - ct)X + (ct)^2/6,$$

$$K_{xx}^R = x^2 - (X - ct)x + (X - ct)^2/6,$$

$$\begin{aligned} K_y^T &= K_y^R = y - Y/2, \\ K_{yy}^T &= K_{yy}^R = y^2 - Yy + Y^2/6, \\ K_{xy}^T &= [x - (2X - ct)/2](y - Y/2). \\ K_{xy}^R &= [x - (X - ct)/2](y - Y/2), \end{aligned} \tag{10.133}$$

The basic quantities for calculating the moments are

$$\begin{aligned} m_0^T &= m_0 + \bar{K}_x^T m_x + \bar{K}_{xx}^T m_{xx}, \\ m_x^T &= m_x + 2\bar{K}_x^T m_{xx}, \\ m_{xx}^T &= m_{xx}, \\ m_y^T &= m_y + \bar{K}_x^T m_{xy}, \\ m_{yy}^T &= m_{yy}, \\ m_{xy}^T &= m_{xy}, \end{aligned} \tag{10.134}$$

and

$$\begin{aligned} m_0^R &= m_0 + \bar{K}_x^R m_x + \bar{K}_{xx}^R m_{xx}, \\ m_x^R &= m_x + 2\bar{K}_x^R m_{xx}, \\ m_{xx}^R &= m_{xx}, \\ m_y^R &= m_y + \bar{K}_x^R m_{xy}, \\ m_{yy}^R &= m_{yy}, \\ m_{xy}^R &= m_{xy}, \end{aligned} \tag{10.135}$$

where $\bar{K}$ is the average of the new orthogonal function:

$$\bar{K}_x^T = (X - ct)/2, \quad \bar{K}_x^R = -ct/2,$$
$$\bar{K}_{xx}^T = (X - ct)(X - 2ct)/6, \quad \bar{K}_{xx}^R = ct(2ct - X)/6.$$

The moments in the right part to be transported to the adjacent cell are expressed as follows:

$$\begin{aligned} S_0^T &= \alpha[S_0 + (1-\alpha)S_x + (1-\alpha)(1-2\alpha)S_{xx}], \\ S_x^T &= \alpha^2[S_x + 3(1-\alpha)S_{xx}], \\ S_{xx}^T &= \alpha^3 S_{xx}, \\ S_y^T &= \alpha[S_y + (1-\alpha)S_{xy}], \\ S_{yy}^T &= \alpha S_{yy}, \\ S_{xy}^T &= \alpha^2 S_{xy}, \end{aligned} \tag{10.136}$$

where $\alpha = \alpha^T = ct/X = V^T/V$. The moments in the remaining part are expressed as follows:

$$\begin{aligned} S_0^R &= (1-\alpha)[S_0 - \alpha S_x - \alpha(1-2\alpha)S_{xx}] = S_0 - S_0^T, \\ S_x^R &= (1-\alpha)^2(S_x - 3\alpha S_{xx}), \\ S_{xx}^R &= (1-\alpha)^3 S_{xx}, \\ S_y^R &= (1-\alpha)(S_y - \alpha S_{xy}) = S_y - S_y^T, \\ S_{yy}^R &= (1-\alpha)S_{yy} = S_{yy} - S_{yy}^T, \\ S_{xy}^R &= (1-\alpha)^2 S_{xy}. \end{aligned} \tag{10.137}$$

As the final step of the procedure, the orthogonal functions and moments transported from the adjacent cell and those in the remaining part of the original cell are combined to create new united moments in the cell. Here, we consider combinations of moments in the transported cell of the i-th grid point ($S^T_{n,i}$) and those in the remaining cell of the (i+1)-th grid point ($S^R_{n,i+1}$). The calculation is terribly complex, and only the results are presented:

$$S_0 = S^R_{0,i+1} + S^T_{0,i},$$

$$S_x = \alpha S^T_{x,i} + (1-\alpha)S^R_{x,i+1} + 3[\alpha S^R_{0,i+1} - (1-\alpha)S^T_{0,i}],$$

$$S_{xx} = \alpha^2 S^T_{xx,i} + (1-\alpha)^2 S^R_{xx,i+1} + 5\{\alpha(1-\alpha)(S^R_{x,i+1} - S^T_{x,i}) - (1-2\alpha)[\alpha S^R_{0,i+1} - (1-\alpha)S^T_{0,i}]\},$$

$$S_y = S^R_{y,i+1} + S^T_{y,i},$$

$$S_{yy} = S^R_{yy,i+1} + S^T_{yy,i},$$

$$S_{xy} = \alpha S^T_{xy,i} + (1-\alpha)S^R_{xy,i+1} + 3[-(1-\alpha)S^T_{y,i} + \alpha S^R_{y,i+1}],$$

where

$$\alpha = \alpha^T_i = V^T_i/(V^T_i + V^R_{i+1}). \tag{10.138}$$

When velocity c in the x direction is negative, the left part of the grid cell,

$$0 \le x \le -ct, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{10.139}$$

is removed (transported) from the cell and added to the adjacent cell on the left during time interval t. This part is now expressed using superscript T. The remaining part,

$$-ct \le x \le X, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{10.140}$$

is expressed by superscript R. New orthogonal functions K^T_i (K^R_i) are calculated in the part T (R) with the volume $V^T = -ctYZ$ ($V^R = (X+ct)YZ$). The orthogonal functions are given as follows:

$$K^T_0 = K^R_0 = 1,$$

$$K^T_x = x + ct/2, \quad K^R_x = x - (X-ct)/2,$$

$$K^T_{xx} = x^2 + ctx + (ct)^2/6,$$

$$K^R_{xx} = x^2 - (X-ct)x + (X^2 - 4Xct + c^2t^2)/6,$$

$$K^T_y = K^R_y = y - Y/2, \tag{10.141}$$

$$K^T_{yy} = K^R_{yy} = y^2 - Yy + Y^2/6,$$

$$K^T_{xy} = (x + ct/2)(y - Y/2),$$

$$K^R_{xy} = [x - (X-ct)/2](y - Y/2).$$

The moments in the left part to be transported to the adjacent cell are expressed as follows:

$$\begin{aligned} S^T_0 &= \alpha[S_0 - (1-\alpha)S_x + (1-\alpha)(1-2\alpha)S_{xx}],\\ S^T_x &= \alpha^2[S_x - 3(1-\alpha)S_{xx}],\\ S^T_{xx} &= \alpha^3 S_{xx},\\ S^T_y &= \alpha[S_y - (1-\alpha)S_{xy}],\\ S^T_{yy} &= \alpha S_{yy},\\ S^T_{xy} &= \alpha^2 S_{xy}, \end{aligned} \tag{10.142}$$

where $\alpha = \alpha^T = -ct/X = V^T/V$. The moments in the remaining part are expressed as follows:

$$\begin{aligned}
S_0^R &= (1-\alpha)[S_0 + \alpha S_x - \alpha(1-2\alpha)S_{xx}] = S_0 - S_0^T, \\
S_x^R &= (1-\alpha)^2(S_x + 3\alpha S_{xx}), \\
S_{xx}^R &= (1-\alpha)^3 S_{xx}, \\
S_y^R &= (1-\alpha)(S_y + \alpha S_{xy}) = S_y - S_y^T, \\
S_{yy}^R &= (1-\alpha)S_{yy} = S_{yy} - S_{yy}^T, \\
S_{xy}^R &= (1-\alpha)^2 S_{xy}.
\end{aligned} \tag{10.143}$$

The following equations give combinations of moments in the remaining cell of the i-th grid point ($S_{n,i}^R$) and those in the transported cell of the (i+1)-th grid point ($S_{n,i+1}^T$):

$$S_0 = S_{0,i}^R + S_{0,i+1}^T,$$

$$S_x = \alpha S_{x,i+1}^T + (1-\alpha)S_{x,i}^R - 3[\alpha S_{0,i}^R - (1-\alpha)S_{0,i+1}^T],$$

$$S_{xx} = \alpha^2 S_{xx,i+1}^T + (1-\alpha)^2 S_{xx,i}^R + 5\{-\alpha(1-\alpha)(S_{x,i}^R - S_{x,i+1}^T) - (1-2\alpha)[\alpha S_{0,i}^R - (1-\alpha)S_{0,i+1}^T]\},$$

$$S_y = S_{y,i}^R + S_{y,i+1}^T,$$

$$S_{yy} = S_{yy,i}^R + S_{yy,i+1}^T,$$

$$S_{xy} = \alpha S_{xy,i+1}^T + (1-\alpha)S_{xy,i}^R - 3[-(1-\alpha)S_{y,i+1}^T + \alpha S_{y,i}^R],$$

where

$$\alpha = \alpha_{i+1}^T = V_{i+1}^T/(V_i^R + V_{i+1}^T). \tag{10.144}$$

The globally integrated tracer is conserved through these operations.

10.5.2 Flux limiter

Some limiters are necessary to guarantee that a tracer is positive (negative) definite.

a. Prather (1986)

Prather (1986) proposed to set limits for the moments related to the direction of advection. For instance, when the moments are advected in the x direction,

$$\begin{aligned}
S_0 &\geq 0, \\
S'_x &= \min[+1.5S_0, \max(-1.5S_0, S_x)], \\
S'_{xx} &= \min[2S_0 - |S'_x|/3, \max(|S'_x| - S_0, S_{xx})], \\
S'_{xy} &= \min[+S_0, \max(-S_0, S_{xy})].
\end{aligned} \tag{10.145}$$

It should be noted that the application of this limiter does not completely guarantee that the tracer will be positive (negative) definite.

b. Merryfield and Holloway (2003)

Flux limiters introduced by Morales Maqueda and Holloway (2006) (hereinafter MH06) extended that of Prather (1986) (hereinafter PR86).

Let us consider the advection in the x direction. As in PR86, MH06 consider the mean tracer distribution ($\bar{\tau}(x)$) in the x direction by integrating in the y and z direction within the grid cell (Equation (2.15) of MH06).

$$\begin{aligned}
\bar{\tau}(x) &= \frac{1}{YZ}\int_0^Y \int_0^Z dy\, dz\, \tau(x,y,z) \\
&= \frac{1}{V}\left[S_0 - S_x + S_{xx} + 2(S_x - 3S_{xx})\frac{x}{X} + 6S_{xx}\left(\frac{x}{X}\right)^2\right],
\end{aligned} \tag{10.146}$$

where x, y, z represent the position relative to the lower south-western corner of the grid cell and X, Y, and Z are grid widths in the x, y, and z directions, respectively. V is the volume of the grid cell.

First we consider the minimum value of $\bar{\tau}(x)$. When S_{xx} is negative (region I), the minimum is taken either at $x = 0$ or $x = X$ because $\bar{\tau}(x)$ is convex upward. Even when S_{xx} is positive, if the value of x that gives the global minimum of the quadratic function is not within $0 \leq x \leq X$ (region II), the minimum is taken either at $x = 0$ or $x = X$. Otherwise (region III), the minimum value is given as the global minimum of the quadratic function. These are summarized as follows:

$$V \min \bar{\tau}(x) = \begin{cases} S_0 - |S_x| + S_{xx}, & \text{if } S_{xx} \leq 0 \quad \text{(region I)}, \\ S_0 - |S_x| + S_{xx}, & \text{if } |S_x| \geq 3S_{xx} \geq 0 \quad \text{(region II)}, \\ S_0 - \frac{S_x^2 + 3S_{xx}^2}{6S_{xx}}, & \text{if } 3S_{xx} \geq |S_x| \quad \text{(region III)}. \end{cases} \tag{10.147}$$

The line of $V \min \bar{\tau}(x) = 0$ is plotted in the $(|S_x|, S_{xx})$ space in Figure 10.5. If a pair of $(|S_x|, S_{xx})$ resides in the region bounded by the blue line and the left vertical axis, the minimum value of $\bar{\tau}(x)$ is positive.

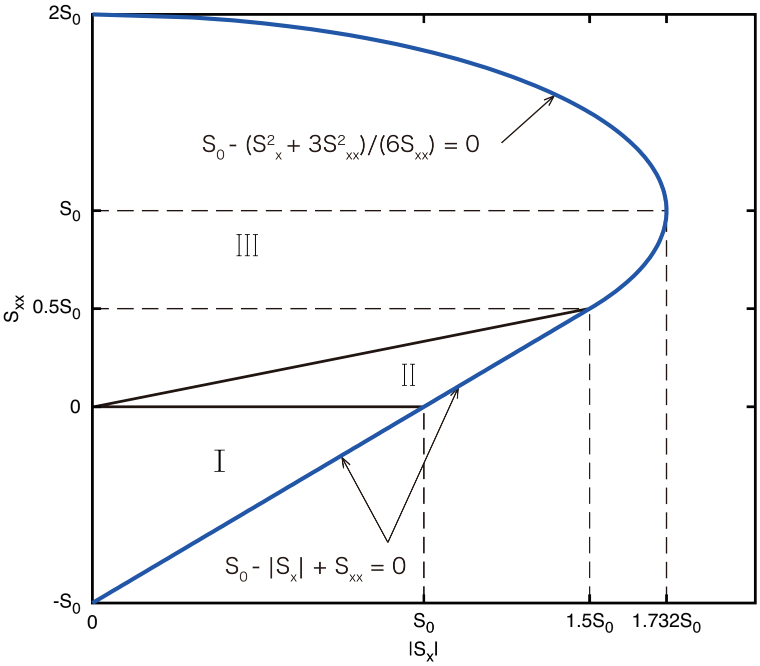

Figure10.5 Schematic to explain the flux limiter to avoid undershooting. This is a reproduction of Figure 3 of Morales Maqueda and Holloway (2006). See text for explanation.

PR86 proposed to modify the moments so that the minimum of $\bar{\tau}$ is positive. This is to improve the sign-definiteness of the tracer. The limiters of MH06 are also based on this strategy. They proposed to modify the moments in the following way,

$$\begin{aligned} S'_x &= \min\left[3^{1/2} S_0,\ \max\left[-3^{1/2} S_0,\ S_x\right]\right], \\ S'_{xx} &= \begin{cases} \min\left[S_0 + \left(S_0^2 - \frac{S'^2_x}{3}\right)^{1/2},\ \max\left[\left|S'_x\right| - S_0,\ S_{xx}\right]\right], & \text{if } \left|S'_x\right| < \frac{3}{2} S_0, \\ \min\left[S_0 + \left(S_0^2 - \frac{S'^2_x}{3}\right)^{1/2},\ \max\left[S_0 - \left(S_0^2 - \frac{S'^2_x}{3}\right)^{1/2},\ S_{xx}\right]\right], & \text{if } \left|S'_x\right| \geq \frac{3}{2} S_0, \end{cases} \\ S'_{xy} &= \min\left[S_0,\ \max[-S_0,\ S_{xy}]\right], \\ S'_{xz} &= \min\left[S_0,\ \max[-S_0,\ S_{xz}]\right]. \end{aligned} \tag{10.148}$$

Here, S'_x, S'_{xx}, S'_{xy} are modified moments and it is assumed that $S_0 \geq 0$. This is slightly different from PR86 because an approximation $\sqrt{3} \sim 1.5$ (see 10.145) was used in PR86.

This modification is explained using Figure 10.5. If a pair of $(|S_x|, S_{xx})$ resides in the region that gives negative minimum of $\bar{\tau}(x)$, $|S_x|$ is shifted along the horizontal axis so that it is less than $3^{1/2} S_0$. Then S_{xx} is shifted along the vertical axis so that the point is within the region that gives the minimum of $\bar{\tau}(x)$ to be positive. The modification to S_{xy}, S_{xz} is designed so that a negative value of the tracer is avoided by these moments.

MH06 proposed to extend the above modifications for setting the lower and upper bounds for the tracers to improve the monotonicity. The lower and upper limit for $\tau(x)$ are set as τ_a and τ_b, respectively ($\tau_a \le \bar{\tau}(x) \le \tau_b$). The setting of the lower limit is achieved by replacing S_0 with $S_0^* = S_0 - V\tau_a$ in the above modification. For the upper limit, we interpret the condition "the maximum of $\bar{\tau}(x)$ must be less than τ_b" as "the minimum of $-\bar{\tau}(x)$ must be more than $-\tau_b$." In other words, the condition is "the minimum of $\tau_b - \bar{\tau}(x)$ must be positive."

Using (10.146), $\tau_b - \bar{\tau}(x)$ may be expressed as follows:

$$\begin{aligned}\tau_b - \bar{\tau}(x) &= \frac{1}{V}\left[V\tau_b - S_0 + S_x - S_{xx} - 2(S_x - 3S_{xx})\frac{x}{X} - 6S_{xx}\left(\frac{x}{X}\right)^2\right] \\ &= \frac{1}{V}\left[S_0^* + S_x - S_{xx} - 2(S_x - 3S_{xx})\frac{x}{X} - 6S_{xx}\left(\frac{x}{X}\right)^2\right], \end{aligned} \tag{10.149}$$

where $S_0^* = V\tau_b - S_0$. Its minimum is obtained as follows:

$$V \min\left[\tau_b - \bar{\tau}(x)\right] = \begin{cases} S_0^* - |S_x| - S_{xx}, & \text{if } S_{xx} \ge 0 \quad \text{(region I)}, \\ S_0^* - |S_x| - S_{xx}, & \text{if } |S_x| \ge -3S_{xx} \ge 0 \quad \text{(region II)}, \\ S_0^* + \frac{S_x^2 + 3S_{xx}^2}{6S_{xx}}, & \text{if } -3S_{xx} \ge |S_x| \quad \text{(region III)}. \end{cases} \tag{10.150}$$

Following Figure 10.5, this is visualized as Figure 10.6.

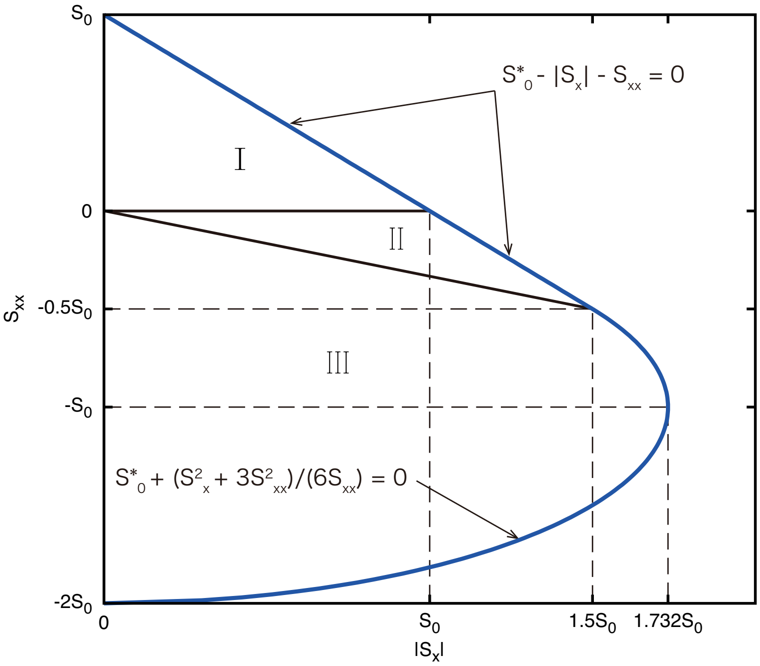

Figure10.6 Schematic to explain the flux limiter to avoid overshooting. The region in $(|S_x|, S_{xx})$ space that may set the upper bound on tracer distribution. The region bounded by the blue curve and the left vertical axis gives the tracer value that does not exceed the upper bound.

The modification for the moments corresponding to (10.148) is expressed as follows:

$$\begin{aligned} S'_x &= \min\left[3^{1/2}S_0^*,\ \max\left[-3^{1/2}S_0^*,\ S_x\right]\right], \\ S'_{xx} &= \begin{cases} \min\left[S_0^* - |S'_x|,\ \max\left[-S_0^* - \left(S_0^{*2} - \frac{S_x'^2}{3}\right)^{1/2},\ S_{xx}\right]\right], & \text{if } |S'_x| < \frac{3}{2}S_0^*, \\ \min\left[-S_0^* + \left(S_0^{*2} - \frac{S_x'^2}{3}\right)^{1/2},\ \max\left[-S_0^* - \left(S_0^{*2} - \frac{S_x'^2}{3}\right)^{1/2},\ S_{xx}\right]\right], & \text{if } |S'_x| \ge \frac{3}{2}S_0^*, \end{cases} \\ S'_{xy} &= \min\left[S_0^*,\ \max(-S_0^*,\ S_{xy})\right], \\ S'_{xz} &= \min\left[S_0^*,\ \max(-S_0^*,\ S_{xz})\right]. \end{aligned} \tag{10.151}$$

MH06 proposed three modification methods (Method A - C). MRI.COM adopts "Method B" of Morales Maqueda and Holloway (2006) as proposed by Merryfield and Holloway (2003), which lays emphasis on monotonicity and suppresses numerical diffusion as well. In this method, the minimum and the maximum value among the three grid cells $(i-1, i, i+1)$ are set as the lower and the upper limit, respectively. Moments are adjusted using Eqs. (10.148) and (10.151). This method does not guarantee exact monotonicity, but improves monotonicity of the zeroth moment.

10.5.3 Calculating SOM advection in MRI.COM

It should be noted that the coordinate system is not Cartesian in ocean models. Since the coordinate system covers a spherical surface, the x direction in a grid cell is not identical to that in the adjacent cell, for instance. Thus, the exact conservation of moments cannot be realized. In addition, a tracer-cell including solid earth (sea floor or lateral boundary) is not a cuboid, so the orthogonal functions cannot be defined precisely for such a cell. Nevertheless, the procedures described in the previous subsection can be carried out using the volume of seawater in the non-cuboid grid cell, and the zeroth moment S_0 (total amount of the tracer) is conserved.

As indicated in the expressions in the previous subsections, the volume-integrated moments (S_i) and the fraction of volume to be removed (α) are used in the SOM advection scheme. There are 10 moments for each tracer. The fraction α is calculated using volume transports, which are calculated in the subroutines `continuity__diagnose_horizontal` and `continuity__diagnose_vertical`. Following Prather (1986), the procedures in three directions are executed in order, not simultaneously. By default, the procedure in the meridional direction (`advec_y`) is called first, the zonal direction (`advec_x`) next, and lastly the vertical direction (`advec_z`). The order of operations in the horizontal direction may be flipped every time step if runtime option `lsomstrang` is set to be true (Strang splitting). This would improve the overall accuracy of the serially executed advection operations (Skamarock, 2006). The change in the tracer value caused by SOM advection is estimated in the subroutine `tracer__adv` and added to the variable `trcal` directly.

Usage

Model option `SOMADVEC` must be specified for compilation when the SOM advection scheme will be used for any tracer. Namelist `nml_somadv` is required (see Table 10.2) at run time. You must also specify which tracer will use this scheme as well as what kind of specifications will be used for that tracer. See Section 13.3.2 and Table 13.3 for details.

Table10.2 Variables defined in namelist `nml_somadv`

name	explanation	string or value
`lsomstrang`	The order of calling x and y direction is reversed every time step	`.true.` / `.false.`
`lsommonitor`	flag to monitor the conservation of the moments	`.true.` / `.false.`

10.6 Piecewise Parabolic Method (PPM) scheme

10.6.1 Outline

The Piecewise Parabolic Method (PPM; Colella and Woodward, 1984) scheme first expresses a local tracer distribution within each model grid cell as a quadratic equation. These tracer distributions are then advected in a Lagrangian manner and redistributed to the fixed model grid cells (Figure 10.7).

Here we consider a one-dimensional advection equation in the ξ direction and assume that we know the mean value of an arbitrary tracer in the j grid at time t^n (a_j^n). Then we consider a polynomial that represents the local distribution of the tracer in each grid cell and describes a set of these polynomials with $a(\xi)$. These polynomials satisfy the following equation:

$$a_j^n = \frac{1}{\Delta\xi_j}\int_{\xi_{j-\frac{1}{2}}}^{\xi_{j+\frac{1}{2}}} a(\xi)d\xi, \quad \Delta\xi_j = \xi_{j+\frac{1}{2}} - \xi_{j-\frac{1}{2}}, \tag{10.152}$$

where $\xi_{j-\frac{1}{2}}$ and $\xi_{j+\frac{1}{2}}$ denote cell boundaries between the $j-1$ and j grid cells and between the j and $j+1$ grid cells, respectively.

A solution of the advection equation at time $t^n + \Delta t$ is described as $a(\xi - u\Delta t)$ (Fig. 10.7b) with its initial distribution $a(\xi)$ (Fig. 10.7a). We can gain a_j^{n+1} by integrating this solution as follows:

$$a_j^{n+1} = \frac{1}{\Delta\xi}\int_{\xi_{j-\frac{1}{2}}}^{\xi_{j+\frac{1}{2}}} a(\xi - u\Delta t)d\xi. \tag{10.153}$$

The specific form of the solution depends on a choice of polynomials. The piecewise parabolic method uses a locally continuous quadratic polynomial in each grid cell as follows:

$$\begin{aligned} a(\xi) &= a_{L,j} + x(\Delta a_j + a_{6,j}(1-x)), \\ x &= \frac{\xi - \xi_{j-\frac{1}{2}}}{\Delta \xi_j}, \quad \xi_{j-\frac{1}{2}} \le \xi \le \xi_{j+\frac{1}{2}}. \end{aligned} \tag{10.154}$$

The coefficients of this polynomial are defined with a^n_j, $\lim_{\xi \to \xi_{j-\frac{1}{2}}} a(\xi) = a_{L,j}$ and $\lim_{\xi \to \xi_{j+\frac{1}{2}}} a(\xi) = a_{R,j}$ as follows:

$$\Delta a_j = a_{R,j} - a_{L,j}, \quad a_{6,j} = 6(a^n_j - \frac{1}{2}(a_{L,j} + a_{R,j})). \tag{10.155}$$

The edge values such as $a_{L,j}$ and $a_{R,j}$ are calculated according to the following procedures. First, we interpolate a tracer value at a cell interface using the surrounding cell averages. We use the indefinite integral of a as $A(\xi) = \int^{\xi} a(\xi', t^n) d\xi'$ for this interpolation. This integral at a cell boundary is calculated as follows:

$$A(\xi_{j+\frac{1}{2}}) = A_{j+\frac{1}{2}} = \sum_{k \le j} a^n_k \Delta \xi_k. \tag{10.156}$$

Then we find a quartic polynomial through $(A_{j+k+\frac{1}{2}}, \xi_{j+k+\frac{1}{2}})$, $(k = 0, \pm 1, \pm 2)$ and calculate a tracer value at the interface $a(\xi_{j+\frac{1}{2}}) = a_{j+\frac{1}{2}}$ by $a_{j+\frac{1}{2}} = dA/d\xi|_{\xi_{j+\frac{1}{2}}}$. The interpolated value $a_{j+\frac{1}{2}}$ is described as follows:

$$\begin{aligned} a_{j+\frac{1}{2}} =& a^n_j + \frac{\Delta \xi_j}{\Delta \xi_j + \Delta \xi_{j+1}} (a^n_{j+1} - a^n_j) + \frac{1}{\sum_{k=-1}^{2} \Delta \xi_{j+k}} \\ & \times \left\{ \frac{2\Delta \xi_{j+1} \Delta \xi_j}{\Delta \xi_j + \Delta \xi_{j+1}} \left[\frac{\Delta \xi_{j-1} + \Delta \xi_j}{2\Delta \xi_j + \Delta \xi_{j+1}} - \frac{\Delta \xi_{j+2} + \Delta \xi_{j+1}}{2\Delta \xi_{j+1} + \Delta \xi_j} \right] (a^n_{j+1} - a^n_j) \right. \\ & \left. - \Delta \xi_j \frac{\Delta \xi_{j-1} + \Delta \xi_j}{2\Delta \xi_j + \Delta \xi_{j+1}} \delta a_{j+1} + \Delta \xi_{j+1} \frac{\Delta \xi_{j+1} + \Delta \xi_{j+2}}{\Delta \xi_j + 2\Delta \xi_{j+1}} \delta a_j \right\}, \end{aligned} \tag{10.157}$$

where the mean slope of tracer at the j grid point δa_j is

$$\delta a_j = \frac{\Delta \xi_j}{\Delta \xi_{j-1} + \Delta \xi_j + \Delta \xi_{j+1}} \left[\frac{2\Delta \xi_{j-1} + \Delta \xi_j}{\Delta \xi_{j+1} + \Delta \xi_j} (a^n_{j+1} - a^n_j) + \frac{\Delta \xi_j + 2\Delta \xi_{j+1}}{\Delta \xi_{j-1} + \Delta \xi_j} (a^n_j - a^n_{j-1}) \right]. \tag{10.158}$$

This expression is also suitable for MRI.COM because its tracer point is defined at the center of each unit cell. The interpolation by Eq. (10.157) requires two effective tracer cells on both sides of the cell interface, respectively. For example, we need $[a_{j-1}, a_{j+2}]$ for $a_{j+\frac{1}{2}}$. If a land cell exists in this range, $a_{j+\frac{1}{2}}$ is calculated by distance weighted average of the adjacent cells.

This cell boundary value $a_{j+\frac{1}{2}}$ is usually modified by flux limiters described later to improve monotonicity, before it is used as the edge values $a_{L,j+1}$ and $a_{R,j}$. Because each edge value $a_{L,j+1}$ or $a_{R,j}$ is modified independently to each other, $a(\xi)$ is generally discontinuous at each cell boundary. If a lateral boundary of an ocean cell is completely in land (possible in MRI.COM; see Chapter 3) or a vertical cell interface is at the land-sea boundary, the edge value there is set to the cell average of the ocean cell.

Now we can calculate tracer fluxes at the cell boundaries and the cell average at the next time step with these edge values $a_{L,j}$ and $a_{R,j}$. Averages of advected tracer crossing lateral cell boundaries are calculated by the following equations:

$$f^a_{j+\frac{1}{2},L}(y) = \frac{1}{y} \int_{\xi_{j+\frac{1}{2}} - y}^{\xi_{j+\frac{1}{2}}} a(\xi) d\xi \tag{10.159}$$

$$f^a_{j+\frac{1}{2},R}(y) = \frac{1}{y} \int_{\xi_{j+\frac{1}{2}}}^{\xi_{j+\frac{1}{2}} + y} a(\xi) d\xi. \tag{10.160}$$

For the PPM scheme, these are described as follows:

$$\begin{aligned} f^a_{j+\frac{1}{2},L}(y) &= a_{R,j} - \frac{x}{2}\Big(\Delta a_j - \Big(1 - \frac{2}{3}x\Big) a_{6,j}\Big), \quad \text{for} \ \ x = \frac{y}{\Delta \xi_j} \\ f^a_{j+\frac{1}{2},R}(y) &= a_{L,j+1} + \frac{x}{2}\Big(\Delta a_{j+1} + \Big(1 - \frac{2}{3}x\Big) a_{6,j+1}\Big), \quad \text{for} \ \ x = \frac{y}{\Delta \xi_{j+1}} \end{aligned} \tag{10.161}$$

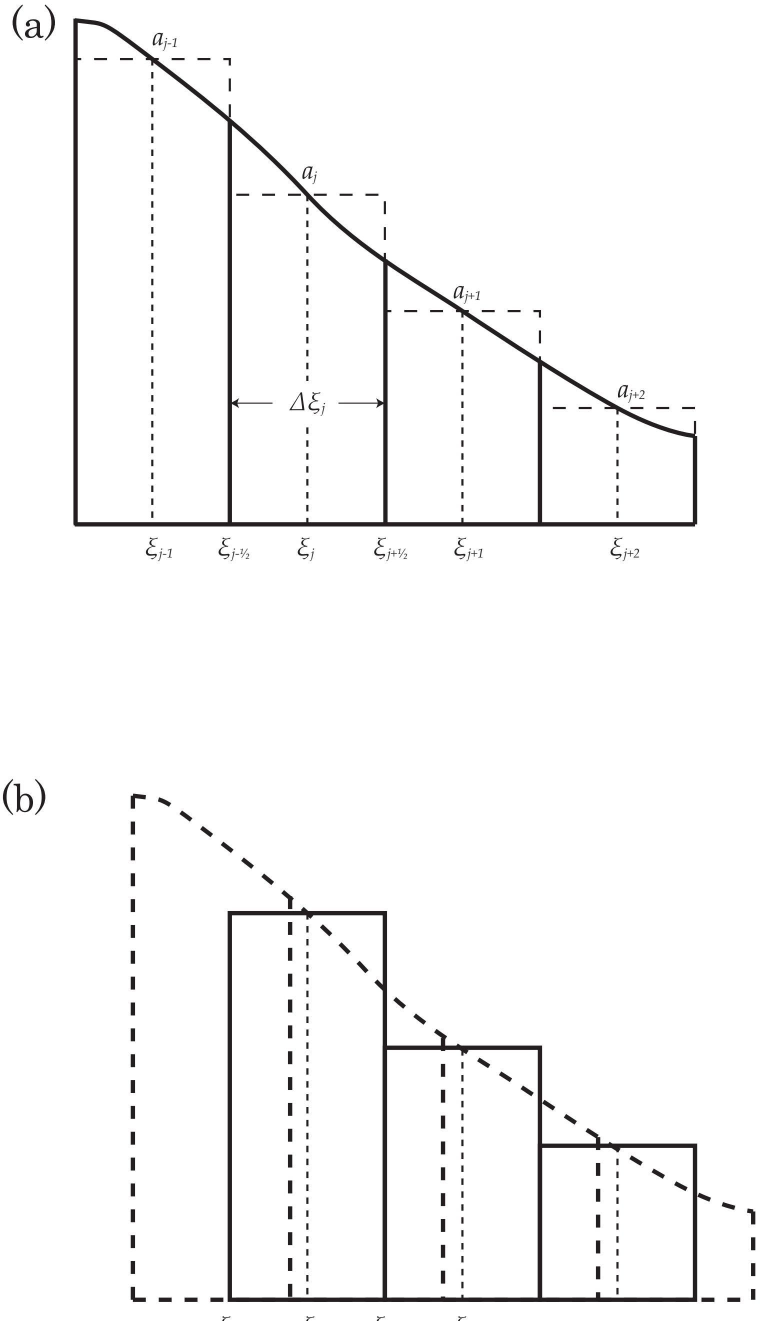

Figure10.7 Schematic of PPM scheme. (a) An initial state. (b) The model is advanced in time and the initial profile in (a) is shifted to the right.

If a lateral boundary is completely in land or a vertical cell interface is at the land-sea boundary, Δa_j and $a_{6,j}$ in Eq. (10.161) in the corresponding direction are forced to zero. So, the equations above are simplified to

$$
\begin{aligned}
f^a_{j+\frac{1}{2},L}(y) &= a_{R,j}, \\
f^a_{j-\frac{1}{2},R}(y) &= a_{L,j}.
\end{aligned}
\tag{10.162}
$$

The new cell average a_j^{n+1} is described in the flux form that guarantees the tracer conservation as follows:

$$
a_j^{n+1} = a_j^n + u\frac{\Delta t}{\Delta\xi_j}(\bar{a}_{j-\frac{1}{2}} - \bar{a}_{j+\frac{1}{2}}), \tag{10.163}
$$

where

$$\bar{a}_{j+\frac{1}{2}} = \begin{cases} f^a_{j+\frac{1}{2},L}(u\Delta t), & \text{if } \; u \geq 0 \\ f^a_{j+\frac{1}{2},R}(-u\Delta t), & \text{if } \; u \leq 0. \end{cases} \tag{10.164}$$

10.6.2 Flux limiter

The following two flux limiters are implemented to enhance monotonicity.

a. Colella and Woodward (1984)

The flux limiter proposed by Colella and Woodward (1984) modifies the tracer gradient in Eq. (10.158) as follows:

$$\begin{aligned} \delta_m a_j =& \min(|\delta a_j|, 2|a^n_{j+1} - a^n_j|, 2|a^n_j - a^n_{j-1}|)\mathrm{sgn}(\delta a_j) \\ & \text{if } \; (a^n_{j+1} - a^n_j)(a^n_j - a^n_{j-1}) > 0 \\ =& 0 \quad \text{otherwise} \end{aligned} \tag{10.165}$$

This allows us to express steeper gradients when we treat discontinuous tracer distributions.

The edge values $a_{L,j+1}$ and $a_{R,j}$ are normally assigned $a_{j+\frac{1}{2}}$. However, an interpolation function occasionally exceeds the range of $a_{L,j}$ and $a_{R,j}$ within the j-th grid cell. In that case, we make the following adjustments.

- Make the interpolation function a constant if a^n_j exceeds the range of $a_{L,j}$ and $a_{R,j}$.
- Adjust edge values to satisfy a condition to prevent overshoot of the interpolation function, which may occur when a^n_j is in the range of $a_{L,j}$ and $a_{R,j}$ but close enough to one side. The condition is $|\Delta a_j| \geq |a_{6,j}|$.

Specifically, edge values are modified as follows:

$$a_{L,j} \longrightarrow a^n_j, \quad a_{R,j} \longrightarrow a^n_j \quad \text{if} \quad (a_{R,j} - a^n_j)(a^n_j - a_{L,j}) < 0 \tag{10.166}$$

$$a_{L,j} \longrightarrow 3a^n_j - 2a_{R,j} \quad \text{if} \quad (a_{R,j} - a_{L,j})\left(a^n_j - \frac{1}{2}(a_{L,j} + a_{R,j})\right) > \frac{(a_{R,j} - a_{L,j})^2}{6} \tag{10.167}$$

$$a_{R,j} \longrightarrow 3a^n_j - 2a_{L,j} \quad \text{if} \quad -\frac{(a_{R,j} - a_{L,j})^2}{6} > (a_{R,j} - a_{L,j})\left(a^n_j - \frac{1}{2}(a_{L,j} + a_{R,j})\right) \tag{10.168}$$

Equation (10.167) is applied when $a_{R,j} \sim a^n_j$, and Eq. (10.168) is applied when $a_{L,j} \sim a^n_j$.

b. Lin et al. (1994)

The flux limiter proposed by Lin et al. (1994) adjust the tracer slope in Eq. (10.158) as follows:

$$\delta_m a_j = \min(|\delta a_j|, 2\delta a^{\min}_j, 2\delta a^{\max}_j)\,\mathrm{sgn}(\delta a_j), \tag{10.169}$$

where

$$\delta a^{\max}_j = \max(a_{j-1}, a_j, a_{j+1}) - a_j, \tag{10.170}$$

$$\delta a^{\min}_j = a_j - \min(a_{j-1}, a_j, a_{j+1}). \tag{10.171}$$

Edge values are modified as follows:

$$a_{L,j} \longleftarrow a_j - \min(|\delta_m a_j|, |a_{L,j} - a_j|)\,\mathrm{sgn}(\delta_m a_j), \tag{10.172}$$

$$a_{R,j} \longleftarrow a_j + \min(|\delta_m a_j|, |a_{R,j} - a_j|)\,\mathrm{sgn}(\delta_m a_j). \tag{10.173}$$

10.6.3 Extension to multiple dimensions with time-splitting algorithm

An advection scheme is commonly developed for one dimensional problems. There are two ways to extend it to multiple dimensions: one is based on a completely three-dimensional advection algorithm and the other is to apply the one-dimensional scheme to each dimensions in turn. The latter is called the time-splitting method. MRI.COM adopts the time-splitting method implemented in GFDL-MOM.

The following finite difference equations describe an one-dimensional advection equation for an arbitrary tracer ϕ and the equation of continuity in a generalized vertical coordinate r, which adopt the Euler forward scheme in time:

$$(z_r\phi)^{t+\Delta t} = (z_r\phi)^t - F_x(\phi^t) \tag{10.174}$$

$$(z_r)^{t+\Delta t} = (z_r)^t - F_x(I), \tag{10.175}$$

where z_r is cell thickness, the right superscript denotes time and the vector $I \equiv 1$. The second term in the right hand side F_x denotes flux divergence and is described as follows:

$$F_x(\phi) = (\Delta t/\Delta x)[f(\phi)_{x+\Delta x/2} - f(\phi)_{x-\Delta x/2}]. \tag{10.176}$$

The flux f is calculated by a product of a volume transport $(z_r u)_{x\pm\Delta x/2}$ and a tracer value $\phi_{x\pm\Delta x/2}$ at a cell boundary. The tracer values at cell boundaries are calculated by interpolation or integration. Both the tracer advection equation and the equation of continuity must use the same volume transports at cell boundaries.

The volume conservation equation (the equation of continuity) is extended to three dimensions as follows:

$$(z_r)^{t+\Delta t} = (z_r)^t - F_x(I) - F_y(I) - F_z(I). \tag{10.177}$$

The finite difference equation of the one-dimensional tracer advection equation (10.174) is extended to multiple dimensions according to the following time-splitting method implemented in GFDL-MOM.

$$(z_r)^{t+\Delta t} = (z_r)^t - F_x(I) - F_y(I) - F_z(I) \tag{10.178}$$

$$(z_r\phi)^* = (z_r\phi)^t - F_x(\phi^t) + F_x(I)\phi^t, \tag{10.179}$$

$$\phi^* = (z_r\phi)^*/(z_r)^t, \tag{10.180}$$

$$(z_r\phi)^{**} = (z_r\phi)^* - F_y(\phi^*) + F_y(I)\phi^t, \tag{10.181}$$

$$\phi^{**} = (z_r\phi)^{**}/(z_r)^t, \tag{10.182}$$

$$(z_r\phi)^{t+\Delta t} = (z_r\phi)^{**} - F_z(\phi^{**}) - \{F_x(I) + F_x(I)\}\phi^t, \tag{10.183}$$

$$\phi^{t+\Delta t} = (z_r\phi)^{t+\Delta t}/(z_r)^{t+\Delta t}. \tag{10.184}$$

A feature of this method is that $F_x(I)\phi^t$ and $F_y(I)\phi^t$ are added in the right hand side of Eqs. (10.179) and (10.181). Each additional term puts back (removes) a change in the tracer content due to volume flux divergence (convergence) in the corresponding direction with a tracer value ϕ^t at the initial time ($t = t$). This process improves numerical stability when the volume flux divergence/convergence in each direction is large. The total tracer content in the cell is conserved by adding $-F_x(I)\phi^t - F_y(I)\phi^t$ in the right hand side of Eq. (10.183). We gain the following equation by taking the sum of Eqs. (10.179), (10.181) and (10.183):

$$(z_r\phi)^{t+\Delta t} = (z_r\phi)^t - F_x(\phi^t) - F_y(\phi^*) - F_z(\phi^{**}). \tag{10.185}$$

This clearly shows the tracer content conservation.

10.6.4 Usage

Users must compile the model with the option `PPMADVEC` and specify `adv_scheme%name` = ”`ppm`” in namelist `nml_tracer_data` of tracers to be advected with the PPM advection scheme. Two flux limiters are available as already noted; the monotonic scheme proposed by Lin et al. (1994) and the original limiter of Colella and Woodward (1984) (Table 10.3). Users may also specify whether to use the Strang splitting method which changes the order of horizontal advections every time step and whether to output a special monitor for PPM scheme (Table 10.4).

10.7 MPDATA scheme

10.7.1 Outline

This section explains the MPDATA scheme. In the MPDATA scheme (e.g., Smolarkiewicz and Margolin, 1998), advection for T is first solved by using the original volume transport, U^T, V^T, W^T, to obtain a temporary value ($T^{(1)}$) using the upstream scheme. Using this temporary value, an anti-diffusive volume transport ($U^{(1)}$, $V^{(1)}$, $W^{(1)}$) is computed. Here, the superscript $^{(1)}$ means that it is the first approximation to the error to be subtracted. This set of transports is used to compute a value of the next time step using the upstream scheme starting from the above temporary value.

Table10.3 Namelist `nml_tracer_data`.

variable name	units	description	usage
adv_scheme%name	character	select advection scheme for that tracer	= "ppm"
adv_scheme%limiter_ppm_org	logical	Use correction introduced by Colella and Woodward (1984) for monotonicity	default = .false.
adv_scheme%limiter_ppm_lin	logical	Use correction introduced by Lin et al. (1994) for monotonicity	default = .true.

Table10.4 Namelist `nml_ppmadv`.

variable name	units	description	usage
lppmmonitor	logical	monitor domain integrated volume and tracer	default = .false.
lppmstrang	logical	use Strang splitting	default = .false.

Original dimensionless form of the anti-diffusive velocity given by Smolarkiewicz and Margolin (1998) is written as follows (their equation 16):

$$\mathcal{U}^{(1)} \equiv \frac{u^{(1)}\delta t}{\delta x} = |\mathcal{U}|(1-\mathcal{U})A^{(1)} - \mathcal{U}\mathcal{V}B^{(1)} - \mathcal{U}\mathcal{W}C^{(1)}, \tag{10.186}$$

$$\mathcal{V}^{(1)} \equiv \frac{v^{(1)}\delta t}{\delta y} = |\mathcal{V}|(1-\mathcal{V})B^{(1)} - \mathcal{V}\mathcal{W}C^{(1)} - \mathcal{V}\mathcal{U}A^{(1)}, \tag{10.187}$$

$$\mathcal{W}^{(1)} \equiv \frac{w^{(1)}\delta t}{\delta z} = |\mathcal{W}|(1-\mathcal{W})C^{(1)} - \mathcal{W}\mathcal{U}A^{(1)} - \mathcal{W}\mathcal{V}B^{(1)}, \tag{10.188}$$

where $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{W}$ are the dimensionless velocity based on the original flow field and $A^{(1)}$, $B^{(1)}$, and $C^{(1)}$ are defined as

$$A^{(1)} \equiv \left[\frac{\delta x}{2T}\frac{\partial T}{\partial x}\right]^{(1)}, \tag{10.189}$$

$$B^{(1)} \equiv \left[\frac{\delta y}{2T}\frac{\partial T}{\partial y}\right]^{(1)}, \tag{10.190}$$

$$C^{(1)} \equiv \left[\frac{\delta z}{2T}\frac{\partial T}{\partial z}\right]^{(1)}. \tag{10.191}$$

In MRI.COM, we formulate them by using the set of original volume transports U^T, V^T, W^T. Considering the definition of the volume transport (10.8) to (10.10), the dimensionless form of volume transports can be written as

$$\mathcal{U}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{2U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\Delta t}{\Delta x_{i+\frac{1}{2}j}\Delta y_{i+\frac{1}{2},j}\left(\Delta z_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right)}, \tag{10.192}$$

$$\mathcal{V}_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{2V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}\Delta t}{\Delta x_{i,j+\frac{1}{2}}\Delta y_{i,j+\frac{1}{2}}\left(\Delta z_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right)}, \tag{10.193}$$

$$\mathcal{W}_{i,j,k} = \frac{W^T_{i,j,k}\Delta t}{\Delta z_{i,j,k}(\texttt{areat})_{i,j,k+\frac{1}{2}}}. \tag{10.194}$$

The $A^{(1)}$ through $C^{(1)}$ terms at $(i+\frac{1}{2}, j, k-\frac{1}{2})$, which are needed to calculate $U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}$, are

$$A^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta x_{i+\frac{1}{2},j}\,\delta_x T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}{2\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x} + \epsilon} \tag{10.195}$$

$$B^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta y_{i+\frac{1}{2},j}\,\delta_y \overline{\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{y}}{2\overline{\overline{\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{y}}^{x}}^{y} + \epsilon} \tag{10.196}$$

$$C^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta z_{i+\frac{1}{2},j,k-\frac{1}{2}}\,\delta_z \overline{\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{z}}{2\overline{\overline{\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{z}}^{x}}^{z} + \epsilon} \tag{10.197}$$

where ϵ is a small number that prevents zero division when tracer values reach zero. The finite difference and averaging operators are defined as follows (definitions in y and z directions are the same):

$$\begin{aligned} \delta_x A_i &\equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta x_i}, \quad \delta_x A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_i}{\Delta x_{i+\frac{1}{2}}}, \\ \overline{A_i}^{x} &\equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2}, \quad \overline{A_{i+\frac{1}{2}}}^{x} \equiv \frac{A_{i+1} + A_i}{2}. \end{aligned} \tag{10.198}$$

The $A^{(1)}$ through $C^{(1)}$ terms should be estimated at $(i, j+\frac{1}{2}, k-\frac{1}{2})$ for $V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}}$ and (i, j, k) for $W^{(1)}_{i,j,k}$.

Then anti-diffusive volume transport is written as

$$U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|(1 - \mathcal{U}_{i+\frac{1}{2},j,k-\frac{1}{2}})A^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} - U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\overline{\overline{\mathcal{V}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{y} B^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} - U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\overline{\overline{\mathcal{W}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{z} C^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{10.199}$$

$$V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} = \left|V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}\right|(1 - \mathcal{V}_{i,j+\frac{1}{2},k-\frac{1}{2}})B^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} - V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}\overline{\overline{\mathcal{U}_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{x}}^{y} A^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} - V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}\overline{\overline{\mathcal{W}_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{y}}^{z} C^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{10.200}$$

$$W^{(1)}_{i,j,k} = \left|W^T_{i,j,k}\right|(1 - \mathcal{W}_{i,j,k})C^{(1)}_{i,j,k} - W^T_{i,j,k}\overline{\overline{\mathcal{U}_{i+\frac{1}{2},j,k}}^{x}}^{z} A^{(1)}_{i,j,k} - W^T_{i,j,k}\overline{\overline{\mathcal{V}_{i+\frac{1}{2},j,k}}^{y}}^{z} B^{(1)}_{i,j,k}. \tag{10.201}$$

10.7.2 Gauge transformation and flux limiter

Original MPDATA scheme is positive definite and cannot be used for tracers that take negative value. This can be avoided by adding some constant for the temporary value, $T^{(1)}$.

The monotonicity of MPDATA scheme can be obtained by using a non-oscillatory option proposed by Smolarkiewicz and Grabowski (1990). In this option, an anti-diffusive volume transport has the upper limit given as follows:

$$\left[U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}\right]^{mon} = \begin{cases} \min(1, \beta^{out}_{i,j,k-\frac{1}{2}}, \beta^{in}_{i+1,j,k-\frac{1}{2}})U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} & \text{if} \quad U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} > 0, \\ \min(1, \beta^{out}_{i+1,j,k-\frac{1}{2}}, \beta^{in}_{i,j,k-\frac{1}{2}})U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} & \text{if} \quad U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} < 0, \end{cases} \tag{10.202}$$

$$\left[V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}}\right]^{mon} = \begin{cases} \min(1, \beta^{out}_{i,j,k-\frac{1}{2}}, \beta^{in}_{i,j+1,k-\frac{1}{2}})V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} & \text{if} \quad V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} > 0, \\ \min(1, \beta^{out}_{i,j+1,k-\frac{1}{2}}, \beta^{in}_{i,j,k-\frac{1}{2}})V^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} & \text{if} \quad V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} < 0, \end{cases} \tag{10.203}$$

$$\left[W^{(1)}_{i,j,k}\right]^{mon} = \begin{cases} \min(1, \beta^{out}_{i,j,k+\frac{1}{2}}, \beta^{in}_{i,j,k-\frac{1}{2}})W^{(1)}_{i,j,k} & \text{if} \quad W^{(1)}_{i,j,k} > 0, \\ \min(1, \beta^{out}_{i,j,k-\frac{1}{2}}, \beta^{in}_{i,j,k+\frac{1}{2}})W^{(1)}_{i,j,k} & \text{if} \quad W^{(1)}_{i,j,k} < 0, \end{cases} \tag{10.204}$$

where

$$\beta^{in}_{i,j,k-\frac{1}{2}} \equiv \frac{T^{max}_{i,j,k-\frac{1}{2}} - T^{(1)}_{i,j,k-\frac{1}{2}}}{A^{in}_{i,j,k-\frac{1}{2}} + \epsilon}, \tag{10.205}$$

$$\beta^{out}_{i,j,k-\frac{1}{2}} \equiv \frac{T^{(1)}_{i,j,k-\frac{1}{2}} - T^{min}_{i,j,k-\frac{1}{2}}}{A^{out}_{i,j,k-\frac{1}{2}} + \epsilon}, \tag{10.206}$$

$$\begin{aligned} T^{max}_{i,j,k-\frac{1}{2}} = \max\Big(& T_{i,j,k-\frac{1}{2}}, \; T_{i+1,j,k-\frac{1}{2}}, \; T_{i-1,j,k-\frac{1}{2}}, \; T_{i,j+1,k-\frac{1}{2}}, \; T_{i,j-1,k-\frac{1}{2}}, \; T_{i,j,k+\frac{1}{2}}, \; T_{i,j,k-\frac{3}{2}}, \\ & T^{(1)}_{i,j,k-\frac{1}{2}}, \; T^{(1)}_{i+1,j,k-\frac{1}{2}}, \; T^{(1)}_{i-1,j,k-\frac{1}{2}}, \; T^{(1)}_{i,j+1,k-\frac{1}{2}}, \; T^{(1)}_{i,j-1,k-\frac{1}{2}}, \; T^{(1)}_{i,j,k+\frac{1}{2}}, \; T^{(1)}_{i,j,k-\frac{3}{2}} \Big), \end{aligned} \tag{10.207}$$

$$\begin{aligned} T^{min}_{i,j,k-\frac{1}{2}} = \min\Big(& T_{i,j,k-\frac{1}{2}}, \; T_{i+1,j,k-\frac{1}{2}}, \; T_{i-1,j,k-\frac{1}{2}}, \; T_{i,j+1,k-\frac{1}{2}}, \; T_{i,j-1,k-\frac{1}{2}}, \; T_{i,j,k+\frac{1}{2}}, \; T_{i,j,k-\frac{3}{2}}, \\ & T^{(1)}_{i,j,k-\frac{1}{2}}, \; T^{(1)}_{i+1,j,k-\frac{1}{2}}, \; T^{(1)}_{i-1,j,k-\frac{1}{2}}, \; T^{(1)}_{i,j+1,k-\frac{1}{2}}, \; T^{(1)}_{i,j-1,k-\frac{1}{2}}, \; T^{(1)}_{i,j,k+\frac{1}{2}}, \; T^{(1)}_{i,j,k-\frac{3}{2}} \Big), \end{aligned} \tag{10.208}$$

and A^{in} and A^{out} are the absolute values of the total incoming and outgoing advection flux at the T-box.

10.7.3 Usage

Model option `MPDATAADVEC` must be specified for compilation when MPDATA will be used for any tracer. At run time, you must specify which tracer will use this scheme as well as what kind of specifications will be used for that tracer. See Section 13.3.2 and Table 13.3 for details.

Chapter 11

SGS parameterization of lateral mixing of tracers

This chapter explains subgrid-scale parameterizations for horizontal mixing of tracers.

11.1 Introduction and Formulation

Historically, a harmonic diffusion operator is applied in each direction of the model coordinates to express mixing of tracers. In the real ocean, transport and mixing would occur dominantly along neutral (isopycnal) surfaces. Thus, horizontal mixing along a constant depth surface is generally inappropriate since neutral surfaces are generally slanting relative to a constant depth surface. Neutral physics schemes are devised as substitutes for the harmonic scheme in the horizontal direction, while the harmonic scheme continues to be used for vertical diffusion.

By default, the diffusion operator mixes a tracer in each direction of the model coordinates with the harmonic scheme. For horizontal diffusion, the biharmonic scheme can be used instead of the harmonic scheme. Using (24.22) and (24.23), the harmonic-type diffusivity is represented for the horizontal diffusion, Eq. (9.6), as follows:

$$
\begin{aligned}
\mathcal{D}_H(T) &= -\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi z_s F^T_\mu)}{\partial\mu} + \frac{\partial(h_\mu z_s F^T_\psi)}{\partial\psi}\right\} \\
&= \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial\mu}\left(\frac{h_\psi z_s \kappa_H}{h_\mu}\frac{\partial T}{\partial\mu}\right) + \frac{\partial}{\partial\psi}\left(\frac{h_\mu z_s \kappa_H}{h_\psi}\frac{\partial T}{\partial\psi}\right)\right\}, \qquad (11.1)
\end{aligned}
$$

where κ_H is the horizontal diffusion coefficient. When the biharmonic-type is selected for horizontal diffusion, the above Laplacian operation is repeated twice reversing its sign after the first operation.

When the neutral physics schemes are selected, the advection-diffusion equation for any tracer (T) is expressed as follows (Gent and McWilliams, 1990):

$$
\frac{DT}{Dt} + \nabla_H \cdot \left[T\frac{\partial}{\partial z}(\kappa_T \nabla_H \rho/\rho_z)\right] + \frac{\partial}{\partial z}\left[T\nabla_H \cdot (-\kappa_T \nabla_H \rho/\rho_z)\right] = \mathcal{D}(T) + Q^T, \qquad (11.2)
$$

where the first term on the r.h.s. is the isopycnal diffusion, whose form is given by

$$
\mathcal{D}(T) = \nabla \cdot (\kappa_I \mathbf{K} \nabla T), \qquad (11.3)
$$

where

$$
\begin{aligned}
\mathbf{K} &= \frac{1}{\rho_x^2+\rho_y^2+\rho_z^2}\begin{pmatrix} \rho_y^2+\rho_z^2 & -\rho_x\rho_y & -\rho_x\rho_z \\ -\rho_x\rho_y & \rho_x^2+\rho_z^2 & -\rho_y\rho_z \\ -\rho_x\rho_z & -\rho_y\rho_z & \rho_x^2+\rho_y^2 \end{pmatrix} \qquad (11.4) \\
&= \frac{1}{1+(\rho_x/\rho_z)^2+(\rho_y/\rho_z)^2}\begin{pmatrix} 1+(\rho_y/\rho_z)^2 & -(\rho_x/\rho_z)(\rho_y/\rho_z) & -\rho_x/\rho_z \\ -(\rho_x/\rho_z)(\rho_y/\rho_z) & 1+(\rho_x/\rho_z)^2 & -\rho_y/\rho_z \\ -\rho_x/\rho_z & -\rho_y/\rho_z & (\rho_x/\rho_z)^2+(\rho_y/\rho_z)^2 \end{pmatrix}, \qquad (11.5)
\end{aligned}
$$

(Redi, 1982). In the above, the Cartesian notation is used for simplicity. The subscript x represents $\partial/(h_\mu\partial\mu)$ y represents $\partial/(h_\psi\partial\psi)$, and z represents $\partial/(z_s\partial s)$, The isopycnal diffusion coefficient is represented by κ_I. Diapycnal diffusion is not considered here.

The second and third terms on the l.h.s. of (11.2) have the form of advection terms with a transport velocity vector

(u_T, v_T, w_T):

$$u_T \equiv \frac{\partial}{\partial z}\left(\kappa_T \frac{1}{h_\mu}\frac{\partial \rho}{\partial \mu} \Big/ \frac{\partial \rho}{\partial z}\right), \tag{11.6}$$

$$v_T \equiv \frac{\partial}{\partial z}\left(\kappa_T \frac{1}{h_\psi}\frac{\partial \rho}{\partial \psi} \Big/ \frac{\partial \rho}{\partial z}\right), \tag{11.7}$$

$$w_T \equiv -\frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial \mu}\left(\kappa_T \frac{h_\psi}{h_\mu}\frac{\partial \rho}{\partial \mu} \Big/ \frac{\partial \rho}{\partial z}\right) + \frac{\partial}{\partial \psi}\left(\kappa_T \frac{h_\mu}{h_\psi}\frac{\partial \rho}{\partial \psi} \Big/ \frac{\partial \rho}{\partial z}\right)\right\}, \tag{11.8}$$

(Gent and McWilliams, 1990). This velocity can be understood as the advection caused by the thickness diffusion of an isopycnal layer, where κ_T is often referred to as thickness diffusivity.

Note that these could be rewritten as

$$\mathcal{G}(T) = \nabla \cdot (\kappa_T \mathbf{A} \nabla T) \tag{11.9}$$

with

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & -\rho_x/\rho_z \\ 0 & 0 & -\rho_y/\rho_z \\ \rho_x/\rho_z & \rho_y/\rho_z & 0 \end{pmatrix}. \tag{11.10}$$

Comparing with (11.5), we notice that the isopycnal diffusion and the thickness diffusion terms are combined to yield a concise form (Griffies, 1998) and (11.2) can be rewritten as:

$$\frac{DT}{Dt} = \nabla \cdot \{(\kappa_I \mathbf{K} - \kappa_T \mathbf{A}) \nabla T\} + Q^T. \tag{11.11}$$

Three types of lateral diffusion, harmonic horizontal diffusion (default), biharmonic horizontal diffusion (`TRCBIHARM` option), and isopycnal diffusion (`ISOPYCNAL` option), are available in MRI.COM. When isopycnal diffusion (Redi, 1982) is selected, the parameterization scheme for eddy induced advection by Gent and McWilliams (1990) (GM scheme) is used with it (see Section 11.4 for details of this parameterization).

The following is a guide to selecting a horizontal diffusion scheme. Biharmonic diffusion is appropriate for a high resolution model that can resolve eddies because it is more scale-selective than harmonic diffusion and does not unnecessarily suppress disturbances in resolved scales. On the other hand, biharmonic diffusion is not recommended in eddy-less models because this would result in numerical instability. Harmonic horizontal diffusion is also not recommended because this scheme would cause unrealistic cross-isopycnal (diapycnal) mixing as mentioned above. Instead, both isopycnal diffusion and the GM scheme should be used there. Using an anisotropic GM scheme can maintain the meso-scale eddy structures and swift currents by restricting the direction of diffusion, and thus may be usable even for a high resolution model.

The finite difference expression of (11.11) is given by taking the finite volume approach as follows:

$$\begin{aligned} T^{n+1}_{i,j,k-\frac{1}{2}} \Delta V^{n+1}_{i,j,k-\frac{1}{2}} = {} & T^{n-1}_{i,j,k-\frac{1}{2}} \Delta V^{n-1}_{i,j,k-\frac{1}{2}} + 2\,\Delta t\,\{FXD_{i-\frac{1}{2},j,k-\frac{1}{2}} - FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} + FYD_{i,j-\frac{1}{2},k-\frac{1}{2}} - FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} \\ & + FZD_{i,j,k} - FZD_{i,j,k-1}\} + (\text{other terms}). \end{aligned} \tag{11.12}$$

Usually, fluxes due to diffusion are explicitly represented using a starting time level of the temporal discretization. However, when the flux is very large relative to the grid size and the time step interval, which would often occur for vertical fluxes, an implicit scheme is used. These issues are discussed in Chapter 23.

11.2 Horizontal diffusion

11.2.1 Laplacian diffusion

Harmonic horizontal diffusion assumes that the diffusion flux is a product of the gradient of tracer and the diffusion coefficient (κ_H). The finite difference expressions for the fluxes are given as follows:

$$FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} = -\,\kappa_H \Delta y_{i+\frac{1}{2},j} \overline{\Delta z_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{y} \delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{11.13}$$

$$FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} = -\,\kappa_H \Delta x_{i,j+\frac{1}{2}} \overline{\Delta z_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{x} \delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{11.14}$$

where

$$\delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}} \equiv \frac{T_{i+1,j,k-\frac{1}{2}} - T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j}}, \tag{11.15}$$

$$\delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}} \equiv \frac{T_{i,j+1,k-\frac{1}{2}} - T_{i,j,k-\frac{1}{2}}}{\Delta y_{i,j+\frac{1}{2}}}. \tag{11.16}$$

11.2.2 Biharmonic diffusion

Biharmonic horizontal diffusion (`TRCBIHARM` option) assumes that the diffusion flux is proportional to the gradient of the Laplacian of tracer. The finite difference expressions for the fluxes are given as follows:

$$FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} = \sqrt{\kappa_b}\,\Delta y_{i+\frac{1}{2},j}\overline{\Delta z_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{y}\,\delta_x\,[\mathsf{Lap}(T)]_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{11.17}$$

$$FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} = \sqrt{\kappa_b}\,\Delta x_{i,j+\frac{1}{2}}\overline{\Delta z_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{x}\,\delta_y\,[\mathsf{Lap}(T)]_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{11.18}$$

where κ_b is diffusivity coefficient and

$$\mathsf{Lap}(T)_{i,j,k-\frac{1}{2}} = \frac{\Delta x_{i,j}\Delta y_{i,j}}{\Delta V_{i,j,k-\frac{1}{2}}}(\delta_x\sqrt{\kappa_b}\,\overline{\Delta z_{i,j,k-\frac{1}{2}}}^{y}\,\delta_x T_{i,j,k-\frac{1}{2}} + \delta_y\sqrt{\kappa_b}\,\overline{\Delta z_{i,j,k-\frac{1}{2}}}^{x}\,\delta_y T_{i,j,k-\frac{1}{2}}). \tag{11.19}$$

11.2.3 Specification of coefficient

The diffusion coefficient of horizontal diffusion is specified using the namelists listed on Tables 11.1 and 11.2. This must be zero if `ISOPYCNAL` option is selected, unless the horizontal diffusion is applied intentionally. By using `SMAGHD` option, the diffusion coefficient can be given based on the horizontal viscosity coefficient according to the Smagorinsky scheme (Section 8.3.3).

Table11.1 Namelist `nml_tracer_diff_horz`

variable name	units	description	usage
`diff_horz_cm2ps`	cm^2 s^{-1}	Laplacian diffusion coefficient (κ_H)	if not `TRCBIHARM`
`diff_horz_cm4ps`	cm^4 s^{-1}	Biharmonic diffusion coefficient (κ_b)	if `TRCBIHARM`
`file_diff_horz_2d`	character	2D distribution of diffusion coefficient is read from this file	

Table11.2 Namelist `nml_grid_size_change_mix_coefs`

variable name	units	description	usage
`l_grid_size_change_mix_coefs`	logical	the given coefficient is multiplied by the fraction of the local grid size to 100 km.	

11.3 Isopycnal diffusion

In isopycnal diffusion, the diffusion flux is expressed by high diffusivity along the isopycnal surface κ_I, low diffusivity across the isopycnal surface κ_D, and their product with the gradient of tracer in each direction. Using diffusion tensor $\mathbf{K}$, each flux component is written as $F^m(T) = -K^{mn}\partial_n T$, and then

$$\mathbf{K} = \frac{\kappa_I}{1+S^2}\begin{pmatrix} 1+S_y^2+\epsilon S_x^2 & (\epsilon-1)S_xS_y & (1-\epsilon)S_x \\ (\epsilon-1)S_xS_y & 1+S_x^2+\epsilon S_y^2 & (1-\epsilon)S_y \\ (1-\epsilon)S_x & (1-\epsilon)S_y & \epsilon+S^2 \end{pmatrix}, \tag{11.20}$$

where

$$\mathbf{S} = (S_x, S_y, 0) = \left(-\frac{\dfrac{\partial \rho}{\partial x}}{\dfrac{\partial \rho}{\partial z}}, -\frac{\dfrac{\partial \rho}{\partial y}}{\dfrac{\partial \rho}{\partial z}}, 0 \right), \tag{11.21}$$

$$S = \mid \mathbf{S} \mid, \tag{11.22}$$

$$\epsilon = \frac{\kappa_D}{\kappa_I} \tag{11.23}$$

(Redi, 1982). This diffusion tensor is rewritten as the sum of horizontal diffusion tensor and a difference from the horizontal diffusion:

$$\mathbf{K} = \kappa_I \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \frac{\kappa_I}{1+S^2} \begin{pmatrix} (\epsilon-1)S_x^2 & (\epsilon-1)S_xS_y & (1-\epsilon)S_x \\ (\epsilon-1)S_xS_y & (\epsilon-1)S_y^2 & (1-\epsilon)S_y \\ (1-\epsilon)S_x & (1-\epsilon)S_y & \epsilon+S^2 \end{pmatrix}. \tag{11.24}$$

The finite difference method is based on Cox (1987) except for the small isopycnal slope approximation. The finite difference form of three components of the gradient of tracer in calculating the east-west component of flux $FXD_{i+\frac{1}{2},j,k-\frac{1}{2}}$ is given as follows (see Figure 11.1):

$$(\delta_x T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{11.25}$$

$$(\delta_y T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \overline{\delta_y T_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{xy}, \tag{11.26}$$

$$(\delta_z T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \overline{\delta_z T_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{xz}. \tag{11.27}$$

Similarly, the north-south component $FYD_{i,j+\frac{1}{2},k-\frac{1}{2}}$ is given by

$$(\delta_x T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \overline{\delta_x T_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{xy}, \tag{11.28}$$

$$(\delta_y T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{11.29}$$

$$(\delta_z T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \overline{\delta_z T_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{yz}, \tag{11.30}$$

and the vertical component $FZD_{i,j,k}$ is given by

$$(\delta_x T)_{i,j,k} = \overline{\delta_x T_{i,j,k}}^{xz}, \tag{11.31}$$

$$(\delta_y T)_{i,j,k} = \overline{\delta_y T_{i,j,k}}^{yz}, \tag{11.32}$$

$$(\delta_z T)_{i,j,k} = \delta_z T_{i,j,k}. \tag{11.33}$$

The density gradient in the calculation of each component of the diffusion tensor can be obtained by replacing T in the above equations with ρ. However, density is calculated at the reference level $k-\frac{1}{2}$ for the east-west and north-south components, and at the reference level k for the vertical component.

In addition, the upper bound on the isopycnal slope $S_{\max}$ (`slope_clip_iso` in namelist `nml_isopy_slope_clip`, Table 11.4) is set because a nearly vertical isopycnal slope and the resultant low horizontal diffusivity could cause numerical instability. If $|\mathbf{S}| > S_{\max}$, $\partial_z \rho$ in all flux components is replaced so as to satisfy $|\mathbf{S}| = S_{\max}$.

The vertical flux due to the third diagonal component of the diffusion tensor (11.20) is

$$FZD_{i,j,k} = (\mathtt{areat})_{i,j,k} \frac{\kappa_I(\epsilon+S^2)}{1+S^2} \delta_z T_{i,j,k}. \tag{11.34}$$

Thus the effective vertical diffusivity κ_{eff} is

$$\kappa_{\text{eff}} = \frac{\kappa_D + \kappa_I S^2}{1+S^2}. \tag{11.35}$$

For a steep isopycnal slope $S \sim 1/100$ and a canonical value of isopycnal diffusion coefficient $\kappa_I \sim 10^7\ \text{cm}^2\ \text{s}^{-1}$ and a typical value of diapycnal diffusion coefficient $\kappa_D \sim 10^{-1}\ \text{cm}^2\ \text{s}^{-1}$,

$$\kappa_{\text{eff}} \sim 10^3\ \text{cm}^2\ \text{s}^{-1}. \tag{11.36}$$

This is a fairly large value which warrants use of an implicit scheme (Section 23.5). In MRI.COM, this term is separated from other terms and solved with other vertical diffusion terms using an implicit method.

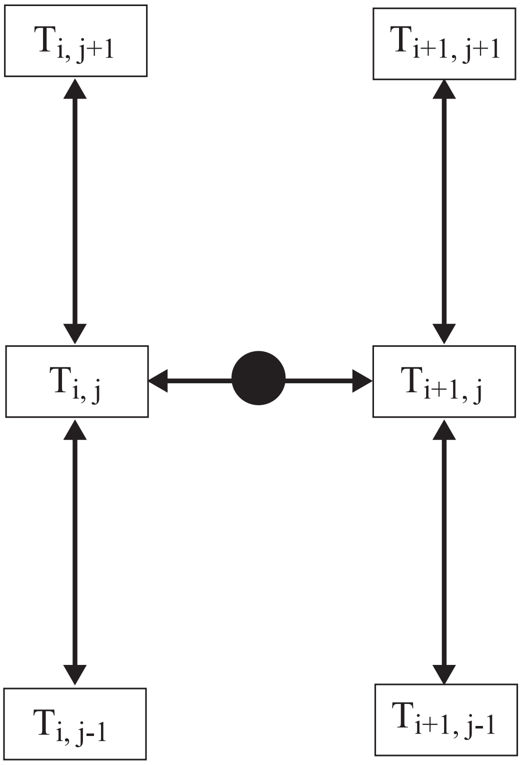

Figure11.1 The way of calculating the gradient at the circle $(i + \frac{1}{2}, j, k - \frac{1}{2})$ in isopycnal diffusion: the east-west gradient is indicated by an arrow through the circle, and the north-south gradient is given by averaging four arrows in the vertical direction.

Griffies et al. (1998) noted a problem in the finite difference expression of the isopycnal diffusion as implemented in the GFDL-model by Cox (1987). The problem is that the down-gradient orientation of the diffusive fluxes along the neutral directions does not necessarily guarantee the zero isoneutral diffusive flux of locally referenced density (e.g., potential temperature when it is the only active tracer). This is caused by the nature of the finite difference method and the non-linearity of the equation of state. Griffies et al. (1998) proposed a remedy, but this remains to be implemented in MRI.COM.

11.3.1 Tapering isopycnal diffusion tensor

We set an upper bound on the isopycnal slope used to evaluate isopycnal tracer diffusion terms in MRI.COM in order to prevent numerical instability around steep isopycnal slopes as noted above. Griffies (2004) shows that such slope clipping could lead to an unrealistically large tracer flux. Another method to prevent numerical instability is to introduce a tapering factor to isopycnal diffusion tensor by specifying `ISOTAPER` option. This tapering factor f is a product of two kinds of factors ($f = f_{\mathrm{steep}} f_{\mathrm{surface}}$) introduced in the subsequent sections. The equation 11.24 is rewritten as follows:

$$\mathbf{K} = f_{\mathrm{hdiag}} \kappa_I \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \frac{f \kappa_I}{1 + S^2} \begin{pmatrix} (\epsilon - 1) S_x^2 & (\epsilon - 1) S_x S_y & (1 - \epsilon) S_x \\ (\epsilon - 1) S_x S_y & (\epsilon - 1) S_y^2 & (1 - \epsilon) S_y \\ (1 - \epsilon) S_x & (1 - \epsilon) S_y & \epsilon + S^2 \end{pmatrix}. \tag{11.37}$$

By default, the factor f_{hdiag} is set to unity. This means that the isopycnal diffusion is rendered the horizontal diffusion as the factor f reduces to zero. One may apply the tapering factor f to the horizontal diagonal terms by setting `l_apply_hdiag = .true.` in namelist `nml_tracer_diff_isopy_taper` (Table 11.5). Note that the tapering factor is defined at each cell boundary where isopycnal diffusion flux is calculated and $\epsilon = \kappa_D / (f \kappa_I)$ in Eq. 11.37.

a. Around steep isopycnal slopes

Danabasoglu and McWilliams (1995) propose a factor that uses a hyperbolic tangent to exponentially taper isopycnal diffusion in steep slope regions as follows:

$$f_{\mathrm{steep}} = \frac{1}{2} \left\{ 1 + \tanh \left(\frac{S_{\mathrm{center}} - |\mathbf{S}|}{S_{\mathrm{width}}} \right) \right\}. \tag{11.38}$$

Two parameters S_{center} and S_{width}, which determine a transitional region, are given by `center_transition` and `width_transition` in namelist `nml_tracer_diff_isopy_taper`, respectively (Table 11.5).

b. Around the sea surface

The vertical displacement of a water parcel due to mesoscale eddy stirring (D) is approximately calculated as follows:

$$D = R|\mathbf{S}|, \tag{11.39}$$

where R is the internal deformation radius and $|\mathbf{S}|$ is the isopycnal slope. If the depth of the water parcel (d) is shallower than D, the boundary constrains its displacement and eddy diffusive fluxes (Griffies, 2004). The other factor f_{surface} proposed by Large et al. (1997) introduces this constraint to the isopycnal diffusion tensor and is calculated as follows:

$$f_{\mathrm{surface}} = \frac{1}{2}\left\{1 + \sin\pi\left(r - \frac{1}{2}\right)\right\}, \tag{11.40}$$

where $r = \max(0, \min(1, d/D))$. Eddy diffusivity is tapered off to zero toward the sea surface in the region where $0 \le d \le D$. In MRI.COM, the reference level of the depth d is set at the boundary layer depth (BLD): $d = -z - \mathrm{BLD}$. In MRI.COM, the surface mixed layer depth (MLD) is treated as the BLD (see also the next section). This means that the eddy diffusivity is tapered to zero within the mixed layer. The upper boundary of this tapering region can be changed to a constant level by setting `upper_level_isotaper_m` in namelist `nml_tracer_diff_isopy_taper` (Table 11.5).

11.3.2 Specification of coefficient

The diffusion coefficients of isopycnal diffusion and GM parameterization explained in the next section are specified using the namelist listed on Table 11.3. We can use different slope maximal limits for isopycnal diffusion and GM parameterization that are specified using the namelist listed on Table 11.4. Configurations of surface tapering for the isopycnal diffusion scheme with `ISOTAPER` option are specified using the namelist listed on Table 11.5.

Table11.3 Namelist `nml_tracer_diff_isopy`

variable name	units	description	usage
`diff_isopy_cm2ps`	$\mathrm{cm^2\,sec^{-1}}$	isopycnal diffusion coefficient	if `ISOPYCNAL`
`diff_diapy_cm2ps`	$\mathrm{cm^2\,sec^{-1}}$	diapycnal diffusion coefficient	if `ISOPYCNAL`
`diff_thick_cm2ps`	$\mathrm{cm^2\,sec^{-1}}$	coefficient of GM parameterization	if `ISOPYCNAL`

Table11.4 Namelist `nml_isopy_slope_clip`

variable name	units	description	usage
`slope_clip_iso`	1	maximum slope of isopycnal surface for isopycnal diffusion	if `ISOPYCNAL`
`slope_clip_gm`	1	maximum slope of isopycnal surface for GM parameterization	if `ISOPYCNAL`

Table11.5 Namelist `nml_tracer_diff_isopy_taper`

variable name	units	description	usage
`l_apply_hdiag`	logical	apply tapering factors to horizontal diagonal terms (default = `.false.`)	if `ISOTAPER`
`center_transition`	1	center of the transition region of tapering with hyperbolic tangent (default = 0.005)	if `ISOTAPER`
`width_transition`	1	width of the transition region of tapering with hyperbolic tangent (default = 0.001)	if `ISOTAPER`
`ai_min`	$\mathrm{cm^2\,sec^{-1}}$	lower limit of horizontal isopycnal diffusion coefficient when `l_apply_hdiag` = `.true.` (default = 0.0)	if `ISOTAPER`

Continued on next page

Table 11.5 – continued from previous page

variable name	units	description	usage
`upper_level_isotaper_m`	m	BLD for the sine taper (default = mixed layer depth (-1.0))	if `ISOTAPER`
`l_explicit_vdif`	logical	handle the vertical diffusion term explicitly (default = `.false.`)	if `ISOTAPER`

11.4 Gent and McWilliams parameterization for eddy-induced transport

11.4.1 General features

The Gent and McWilliams (1990) parameterization represents transports of tracers due to disturbances smaller than the grid size, assuming that a flux proportional to the gradient of the layer thickness exists along the isopycnal surface. The isopycnal diffusion stated above does not produce any flux when the isopycnal surface coincides with the isotherm and isohaline surface. This parameterization, however, produces fluxes in such a case, and acts to decrease the isopycnal slope.

Flux convergence due to diffusion is expressed as follows:

$$R(T) = \partial_m (J^{mn} \partial_n T) \tag{11.41}$$

Diffusion tensor J^{mn} is expressed as the sum of the symmetric component $K^{mn} = (J^{mn} + J^{nm})/2$ and the anti-symmetric component $A^{mn} = (J^{mn} - J^{nm})/2$. Isopycnal diffusion has the form of a symmetric diffusion tensor. Convergence of a skew flux caused by the anti-symmetric component $F^m_{\mathrm{skew}} = -A^{mn}\partial_n T$ is as follows:

$$\begin{aligned} R_A(T) &= \partial_m (A^{mn} \partial_n T) \\ &= \partial_m (A^{mn}) \partial_n T \\ &= \partial_n (\partial_m A^{mn} T), \end{aligned} \tag{11.42}$$

where $A^{mn}\partial_m\partial_n T = 0$ and $\partial_m\partial_n A^{mn} = 0$ are used. If we set a virtual velocity $u_*^n \equiv -\partial_m A^{mn}$, then the flux due to the anti-symmetric component could be regarded as the advection due to this virtual velocity $\mathbf{u}_*$. In this case, the flux is $\mathbf{F}_{\mathrm{adv}} = \mathbf{u}_* T$ and $R_A(T) = -\mathbf{u}_* \cdot \nabla T$ since $\mathbf{u}_*$ is divergence free by definition.

The Gent and McWilliams parameterization for eddy-induced transport velocity is given by

$$\mathbf{u}_* = -\frac{\partial}{\partial \rho}(\kappa_{\mathrm{GM}} \nabla_\rho h) \Big/ \frac{\partial h}{\partial \rho}, \tag{11.43}$$

where h is the depth of the neutral surface (ρ = const). This velocity is expressed in the depth coordinate as

$$\mathbf{u}_* = \left(-\partial_z(\kappa_{\mathrm{GM}} S_x),\ -\partial_z(\kappa_{\mathrm{GM}} S_y),\ \nabla_h \cdot (\kappa_{\mathrm{GM}} \mathbf{S})\right) \tag{11.44}$$

where

$$\mathbf{S} = (S_x, S_y, 0) = \left(-\rho_x/\rho_z, -\rho_y/\rho_z, 0\right). \tag{11.45}$$

(Gent et al., 1995).

Griffies (1998) showed that the tendency of a tracer due to this parameterization might be expressed using an anti-symmetric diffusion tensor $\mathbf{A}$

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & -\kappa_{\mathrm{GM}} S_x \\ 0 & 0 & -\kappa_{\mathrm{GM}} S_y \\ \kappa_{\mathrm{GM}} S_x & \kappa_{\mathrm{GM}} S_y & 0 \end{pmatrix}, \tag{11.46}$$

so that

$$\frac{\partial T}{\partial t} = \cdots \quad + \nabla \cdot (\mathbf{A} \nabla T). \tag{11.47}$$

The flux due to advection can be expressed using a vector streamfunction,

$$\Psi = \kappa_{\mathrm{GM}} \hat{\mathbf{z}} \times \mathbf{S} = (-\kappa_{\mathrm{GM}} S_y,\ \kappa_{\mathrm{GM}} S_x,\ 0), \tag{11.48}$$

which produces $\mathbf{u}_*$ in (11.44):

$$\mathbf{F}_{\mathrm{adv}} = \mathbf{u}_* T = T(\nabla \times \Psi).$$

The skew diffusive expression for the flux using tensor $\mathbf{A}$ in (11.46) is

$$\mathbf{F}_{\text{skew}} = -\mathbf{A}\nabla T = -(\nabla T) \times \Psi = \mathbf{F}_{\text{adv}} - \nabla \times (T\Psi).$$

Thus, the convergence of the flux expressed in tensorial form matches that of the advective expression. In other words, the Gent and McWilliams parameterization is realized by only adding $\mathbf{A}$ to the tensor of the isopycnal diffusion $\mathbf{K}$ (Griffies, 1998).

11.4.2 Dependency of coefficient on space and time

By default, the diffusivity coefficient for the Gent-McWilliams parameterization is constant both in time and space, whose value, $[\kappa_{\text{GM}}]_{\text{ref}}$, is given by `diff_thick_cm2ps` in namelist `nml_tracer_diff_isopy` (Table 11.3). However, it may be dependent on local horizontal grid size by specifying a namelist (see the next paragraph). Several parameterization may be used by choosing `GMVAR` option. User should specify one of `l_visbeck`, `l_eden`, `l_danabasoglu` to be `.true.` in namelist `nml_gmvar_select` (Table 11.8).

a. Simple scheme

If `l_grid_size_change_mix_coefs` = `.true.` in namelist `nml_grid_size_change_mix_coefs` (Table 11.7), the coefficient may be dependent on the horizontal grid size according to the following formula

$$\kappa_{\text{GM}} = [\kappa_{\text{GM}}]_{\text{ref}} \times \min(\Delta x, \Delta y)/(100\,\text{km}), \tag{11.49}$$

where Δx and Δy are local zonal and meridional grid sizes of a U-cell, respectively.

b. Visbeck et al. (1997)

To use the method proposed by Visbeck et al. (1997), specify `l_visbeck` = `.true.` in namelist `nml_gmvar_select`.

Visbeck et al. (1997) proposed to give the GM coefficient κ_{GM} as

$$\kappa_{\text{GM}} = \alpha \frac{M^2}{N} l^2, \tag{11.50}$$

where $\alpha = 0.015$,

$$M^2 = \frac{g}{\rho_0} \left| \nabla_H \rho \right|, \qquad N^2 = -\frac{g}{\rho_0} \frac{\partial \rho}{\partial z},$$

and l the horizontal length scale of the baroclinic zone, g acceleration of gravity, ρ_0 reference density.

Specifically in our model,

$$M^2 = \frac{g}{\rho_0 D} \left[\left(\int_{-H_1}^{-H_0} \frac{\partial \rho}{\partial x} dz \right)^2 + \left(\int_{-H_1}^{-H_0} \frac{\partial \rho}{\partial y} dz \right)^2 \right]^{\frac{1}{2}}, \tag{11.51}$$

and

$$N^2 = \frac{g\left[\sigma_0(H_1) - \sigma_0(H_0)\right]}{\rho_0 D}, \tag{11.52}$$

where $H_0 = 100$ m, $H_1 = 2000$ m, $D = H_1 - H_0$, and σ_0 is the potential density. Lower limit for N is set so that $N^2 \geq 10^{-9}\,\text{s}^{-1}$.

Using the following formula for the phase speed of the 1st baroclinic mode gravity wave (Sueyoshi and Yasuda, 2009)

$$c_1 = \frac{1}{\pi} \int_{-H_B}^{0} \left(-\frac{g}{\rho_0} \frac{\partial \sigma_0}{\partial z} \right)^{\frac{1}{2}} dz, \tag{11.53}$$

where H_B is the depth of sea floor, deformation radius λ_1 is calculated as follows:

$$\lambda_1 = \min\left(\left| \frac{c_1}{f} \right|, 4 \times 10^4 \right) \quad \text{m}. \tag{11.54}$$

Using a factor $r = 7$, GM coefficient is determined as follows:

$$\kappa_{\text{GM}} = \alpha \frac{M^2}{N} (r\lambda_1)^2. \tag{11.55}$$

Lower and upper limits for the coefficient are set as follows:

$$300 \leq \kappa_{\text{GM}} \leq 1500 \quad \text{m}^2\,\text{s}^{-1}. \tag{11.56}$$

c. Eden and Greatbatch (2008)

To use the method proposed by Eden and Greatbatch (2008), specify `l_eden = .true.` in namelist `nml_gmvar_select`.

Eden and Greatbatch (2008) and Eden et al. (2009) proposed that the thickness diffusivity is given by

$$\kappa_{\mathrm{GM}} = cL^2\sigma. \tag{11.57}$$

The eddy length scale L is given as the minimum of the Rossby radius L_r and Rhines scale L_{Rhi}. This choice for L was found to be consistent with independent estimates of eddy length scales from satellite observations and high-resolution model results (Eden, 2007) and with theoretical considerations (Theiss, 2004). L_{Rhi} is estimated from variables of the coarse resolution model as

$$L_{Rhi} = \frac{\sigma}{\beta} \tag{11.58}$$

(Eden and Greatbatch, 2008), while L_r is given by

$$L_r = \min\left[\frac{c_1}{|f|}, \sqrt{\frac{c_1}{2\beta}}\right], \tag{11.59}$$

where c_1 denotes the 1st baroclinic gravity wave speed calculated approximately as eq. (11.53). Considering the thermal wind relation in mid-latitudes, Eden and Greatbatch (2008) proposed that the inverse eddy time scale σ is given by

$$\sigma = f(\mathrm{Ri} + \gamma)^{-\frac{1}{2}}. \tag{11.60}$$

Here, $\mathrm{Ri} = N^2|\partial_z u_h|^{-2}$ denotes the local Richardson number. $\gamma(> 0)$ is introduced to prevent the singularity as $N \longrightarrow 0$, which acts effectively as an upper limit for σ and consequently for κ_{GM}. The default values of γ and c in eq. (11.57), are 200 and 2, respectively.

d. Danabasoglu and Marshall (2007)

To use the method proposed by Danabasoglu and Marshall (2007), `l_danabasoglu = .true.` in namelist `nml_gmvar_select`.

Danabasoglu and Marshall (2007), guided by Ferreira et al. (2005) and Ferreira and Marshall (2006), proposed to specify the vertical variation of κ_{GM} using

$$\kappa_{\mathrm{GM}} = \left[\frac{N^2}{N^2_{\mathrm{ref}}}\right][\kappa_{\mathrm{GM}}]_{\mathrm{ref}}, \tag{11.61}$$

where N is the local buoyancy frequency and $[\kappa_{\mathrm{GM}}]_{\mathrm{ref}}$ is the constant reference value of κ_{GM} within the surface diabatic layer. N_{ref} is the reference buoyancy frequency obtained just below the diabatic layer, in other words, the first stable N^2 below surface diabatic layer. The ratio N^2/N^2_{ref} is set to 1 for all shallower depths. Between the depth at which $N^2 = N^2_{\mathrm{ref}}$ and the ocean bottom, we also ensure that

$$N_{\min} \le \frac{N^2}{N^2_{\mathrm{ref}}} \le 1.0, \tag{11.62}$$

where $N_{\min}$ is the lower limit specified by the user (`ratio_bvf_min` in namelist `nml_gmvar_danabasoglu`).

11.4.3 Surface tapering

By default, no specific modification is applied to the eddy-induced transport velocity of the Gent-McWilliams parameterization near the surface and the bottom, except for limiting the isopycnal slope to a specified value (`slope_clip_gm` in namelist `nml_isopy_slope_clip`). This may result in too strong transport velocity in the first vertical level of the model (sea surface). The problem may be overcome by tapering the transport in the surface mixed layer, where the transport is made nearly or completely uniform in the vertical direction. This is realized by choosing either `SLIMIT` or `GMTAPER` option.

a. Simple scheme

By choosing `SLIMIT` option, the Gent-McWilliams coefficient (κ_{GM}) is linearly reduced from the value at the base of mixed layer to zero at the sea surface within the mixed layer,

$$\kappa_{\mathrm{GM}}(z) = \kappa_{\mathrm{GM}}(z = -\mathrm{MLD}) \times (-z/\mathrm{MLD}) \quad \text{for} \quad -\mathrm{MLD} \le z \le 0, \tag{11.63}$$

where MLD is the depth of the mixed layer. The MLD is defined as the level at which the local potential density is larger than the surface density by a specified value, given by the user (default value is 0.03 $\mathrm{kg\,m^{-3}}$).

b. Danabasoglu et al. (2008)

By choosing GMTAPER option, a practical scheme proposed by Danabasoglu et al. (2008) is used. This scheme modifies the Gent-McWilliams vector stream function for eddy induced transport velocity near the surface, aiming to implement a near-surface parameterization proposed by Ferrari et al. (2008). Concept of the near-surface parameterization is as follows (Danabasoglu et al., 2008):

- In the turbulent boundary layer (BL), the eddy-induced velocity is set parallel to the boundary and has no vertical shear, as expected in the mixed layer.
- There is an eddy diffusion of buoyancy along the boundary as well as along isopycnals.
- In the interior the parameterization satisfies the adiabatic constraint as in the original scheme.
- The two forms are matched through a transition layer that separates the quasi-adiabatic interior with isopycnically oriented eddy fluxes from the near boundary regions.

Two vertical length scales must be estimated to implement this parameterization: the boundary layer depth (BLD) and the transition layer thickness (TLT). Their sum is defined as the diabatic layer depth (DLD), over which the upper-ocean eddy fluxes depart from their interior formulas. In MRI.COM, the surface mixed layer depth (MLD) is treated as the BLD. The MLD is defined as the level at which the local potential density is larger than the surface density by a specified value, given by the user (default value is $0.03\,\mathrm{kg\,m^{-3}}$). The TLT is defined by the range of isopycnals that can be lifted into the boundary layer by eddy heaving, which is given by the product of the internal deformation radius (R) and the isopycnal slope ($|\mathbf{S}|$):

$$D = R|\mathbf{S}|. \tag{11.64}$$

Thus we calculate D at each grid point and the DLD is obtained as follows:

$$\mathrm{DLD} = \mathrm{BLD} + D. \tag{11.65}$$

Now the near-surface expression for the eddy-induced vector streamfunction is given in the following. The streamfunction is split into its boundary layer, Ψ_{BL}, and transition layer, Ψ_{TL}, expression as follows:

$$\Psi_{\mathrm{BL}} = \frac{\eta - z}{\eta + \mathrm{BLD}}\Psi_0 \quad \text{for} \quad -\mathrm{BLD} \le z \le \eta \tag{11.66}$$

and

$$\Psi_{\mathrm{TL}} = \left(\frac{z + \mathrm{BLD}}{\mathrm{TLT}}\right)^2 \Phi + \left(\frac{\eta - z}{\eta + \mathrm{BLD}}\right)\Psi_0 \quad \text{for} \quad -\mathrm{DLD} \le z < -\mathrm{BLD} \tag{11.67}$$

The two functions Ψ_0 and Φ are chosen such that Ψ and its vertical derivative are continuous across the base of BLD and the base of TLT. These constraints then yield

$$\Psi_0 = \frac{\eta + \mathrm{BLD}}{2(\eta + \mathrm{BLD}) + \mathrm{TLT}}(2\Psi_I + \mathrm{TLT}\partial_z\Psi_I) \tag{11.68}$$

and

$$\Phi = -\frac{\mathrm{TLT}}{2(\eta + \mathrm{BLD}) + \mathrm{TLT}}(\Psi_I + (\eta + \mathrm{DLD})\partial_z\Psi_I), \tag{11.69}$$

where Ψ_I is the interior eddy-induced streamfunction at the base of the transition layer given by the Gent-McWilliams parameterization,

$$\Psi_I = -\kappa_{\mathrm{GM}}\frac{\mathbf{z} \times \nabla_H \rho}{\partial_z \rho} \quad \text{at} \quad z = -\mathrm{DLD}. \tag{11.70}$$

In the implementation, to evaluate both Ψ_I and $\partial_z\Psi_I$ at $z = -\mathrm{DLD}$, Ψ_I are evaluated at the vertical grid points that straddle $z = -\mathrm{DLD}$.

Danabasoglu et al. (2008) also showed that the model solutions are not very sensitive to their transition layer thickness. Whether the transition layer is included or not may be specified by `l_transition_layer` in namelist `nml_gm_transition`.

11.4.4 Anisotropic Gent-McWilliams scheme

An anisotropic GM scheme (Smith and Gent (2004), `GMANISOTROP` option), which gives greater diffusivity only in the direction of the current vector, is also available. Using unit vector $\hat{\mathbf{n}} = (n_x, n_y)$ in an arbitrary direction, the two-dimensional anisotropic diffusion tensor is defined as follows:

$$\mathbf{K}_2 = \begin{pmatrix} L & M \\ M & N \end{pmatrix} = \begin{pmatrix} \kappa_A n_x^2 + \kappa_B n_y^2 & \kappa_B n_x n_y \\ \kappa_B n_x n_y & \kappa_B n_x^2 + \kappa_A n_y^2 \end{pmatrix}, \tag{11.71}$$

where κ_A is the diffusivity in the $\hat{\mathbf{n}}$ direction, and κ_B is that in the direction normal to $\hat{\mathbf{n}}$. This is applied to the anti-symmetric tensor in the Gent-McWilliams scheme, and the following expression is obtained (Smith and Gent, 2004),

$$\mathbf{A}' = \begin{pmatrix} 0 & 0 & -LS_x - MS_y \\ 0 & 0 & -MS_x - NS_y \\ LS_x + MS_y & MS_x + NS_y & 0 \end{pmatrix}. \tag{11.72}$$

In the choice of `GMANISOTROP` option, $\hat{\mathbf{n}}$ is set in the direction of the local horizontal velocity. The value of κ_A is read from namelist `nml_tracer_diff_isopy` (variable name `diff_thick_cm2ps`). The ratio of κ_B/κ_A is read from namelist `nml_gmanisotrop` (variable name `cscl_isotrop`). The default value of `cscl_isotrop` is set to 1/2.

11.4.5 Usage Summary

How to specify the overall behavior of the Gent-McWilliams parameterization is summarized as follows.

a. Model options

Model options related to the GM parameterization are listed on Table 11.6

Table11.6 List of model options related to GM parameterization.

option name	description	usage
GMVAR	Coefficient of GM parameterization is allowed to vary	specify nml_gmvar
SLIMIT	Linearly reduce the coefficient of GM parameterization from the bottom of the mixed layer to the sea surface	cannot be used with GMTAPER
GMTAPER	Taper GM vector stream function near the sea surface	cannot be used with SLIMIT, GMANISOTROP, AFC
GMANISOTROP	An-isotropic horizontal variation of GM parameterization	specify nml_gmanisotrop
AFC	Calculate additional flux by using horizontal gradients of density and velocity (Hirabara et al., 2010)	cannot be used with TRCBIHARM

b. Spatial dependency

The diffusion coefficient of GM parameterization may be grid size dependent by using the namelist listed on Table 11.7.

Table11.7 Namelist nml_grid_size_change_mix_coefs

variable name	units	description	usage
l_grid_size_change_mix_coefs	logical	the given coefficient is multiplied by the fraction of the local grid size to 100 km.	default = .false.

Overall behavior of GM parameterization with GMVAR option should be specified by using the namelists listed on Tables 11.8 through 11.12.

Table11.8 Namelist nml_gmvar_select for GMVAR

variable name	units	description	usage
l_visbeck	logical	use Visbeck et al. (1997)	choose only one of the three options
l_eden	logical	use Eden and Greatbatch (2008)	choose only one of the three options
l_danabasoglu	logical	use Danabasoglu and Marshall (2007)	choose only one of the three options

Table11.9 Namelist nml_gmvar_visbeck for GMVAR

variable name	units	description	usage
start_depth	cm	density gradients are averaged from start_depth	l_visbeck = .true.
base_depth	cm	density gradients are averaged to base_depth	l_visbeck = .true.
cscl_gmvar	1	parameter for GM diffusivity calculation	l_visbeck = .true.
upper_limit	$\mathrm{cm}^2\ \mathrm{sec}^{-1}$	upper limit of thickness diffusivity	l_visbeck = .true.
lower_limit	$\mathrm{cm}^2\ \mathrm{sec}^{-1}$	lower limit of thickness diffusivity	l_visbeck = .true.
lcalc_defrad	logical	flag whether deformation radius is calculated or not	l_visbeck = .true.
defrad_const	cm	upper limit of deformation radius when lcalc_defrad = .true. constant horizontal length scale when lcalc_defrad = .false.	l_visbeck = .true.
length_factor	1	[horizontal length scale] = [deformation radius] × length_factor	l_visbeck = .true.

Table11.10 Namelist `nml_gmvar_eden` for `GMVAR`

variable name	units	description	usage
c_EG	1	In this parameterization, Thickness diffusivity is parameterized as `c_EG` $\times L^2 \times$ `sigma_EG`.	`l_eden` = `.true.`
`gamma_EG`	1	`sigma_EG` = $f/(\mathrm{Ri} +$ `gamma_EG`)	`l_eden` = `.true.`
`upper_limit`	$\mathrm{cm}^2\,\mathrm{sec}^{-1}$	upper limit of thickness diffusivity	`l_eden` = `.true.`
`lower_limit`	$\mathrm{cm}^2\,\mathrm{sec}^{-1}$	lower limit of thickness diffusivity	`l_eden` = `.true.`

Table11.11 Namelist `nml_gmvar_danabasoglu` for `GMVAR`

variable name	units	description	usage
`ratio_bvf_min`	1	Lower bound for the squared buoyancy frequency relative to the reference value (default = 0.1)	`l_danabasoglu` = `.true.`
`ratio_bvf_max`	1	Upper bound for the squared buoyancy frequency relative to the reference value (default = 1.0)	`l_danabasoglu` = `.true.`

Table11.12 Namelist `nml_gm_transition`

variable name	units	description	usage
`l_transition_layer`	logical	include transition layer into diabatic layer (default = `.false.`)	effective when `l_danabasoglu` = `.true.`

c. Anisotropic scheme

Behavior of `GMANISOTROP` option should be specified using the namelist listed on Table 11.13.

Table11.13 Namelist `nml_gm_anisotrop`

variable name	units	description	usage
`cscl_isotrop`	1	factor for anisotropy in GM diffusivity	if `GMANISOTROP`

Chapter 12

SGS parameterization of vertical mixing of tracers

This chapter explains subgrid-scale parameterizations of vertical mixing of tracers.

12.1 Vertical diffusion

The following is the equation from the advection-diffusion equation (9.4) with only the time-varying term and vertical diffusion term ($\mathcal{D}_V(T)$),

$$\frac{\partial(z_s T)}{\partial t} = \mathcal{D}_V(T). \tag{12.1}$$

Vertical diffusion term takes the form of Laplacian and the vertical diffusion flux is proportional to the vertical gradient of tracer. The vertical diffusion, Eq. (9.6),

$$\mathcal{D}_V(T) = -\frac{\partial F_z^T}{\partial s} = \frac{\partial}{\partial s}\left(\frac{\kappa_V}{z_s}\frac{\partial T}{\partial s}\right). \tag{12.2}$$

Here, κ_V is the vertical diffusion coefficient.

These equations yield the finite difference form as follows:

$$T^{n+1}_{i,j,k-\frac{1}{2}}\Delta V^{n+1}_{i,j,k-\frac{1}{2}} = T^{n-1}_{i,j,k-\frac{1}{2}}\Delta V^{n-1}_{i,j,k-\frac{1}{2}} + 2\Delta t(FZD_{i,j,k} - FZD_{i,j,k-1}), \tag{12.3}$$

$$FZD_{i,j,k} = -\kappa_z(\mathtt{areat})_{i,j,k+\frac{1}{2}}\delta_z T_{i,j,k}, \tag{12.4}$$

where the use of $(\mathtt{areat})_{i,j,k+\frac{1}{2}}$ implies that the flux occurs only through the oceanic part of the grid interface and

$$\delta_z T_{i,j,k} \equiv \frac{T_{i,j,k-\frac{1}{2}} - T_{i,j,k+\frac{1}{2}}}{\Delta z_k}. \tag{12.5}$$

Note that, for simplicity, the change of the grid thickness at the bottom and fluctuations of the surface height are not considered in the grid distance Δz_k when calculating the gradient.

In most realistic simulations, a backward (implicit) scheme is used in the time integration (`VVDIMP` option; Section 23.5) because high diffusivity is expected owing to the choice of parameterizations needed for realistic simulations. Otherwise, a forward scheme is used.

12.1.1 Specification of coefficient

Background vertical diffusivity, which is horizontally uniform, a function of depth, and fixed in time, should be always given. Additionally, non-time-varying, a three dimensional distribution can be set by selecting `VMBG3D` option to incorporate locally enhanced mixing processes in the climatology induced by interaction between the bottom topography and tidal currents (e.g., St. Laurent et al., 2002). With this choice, three dimensional distributions for vertical diffusivity and viscosity should be prepared in advance. Tables 12.1 and 12.2 summarizes how to give background vertical diffusivity.

Table12.1 Namelist `nml_diff_vert_bg`. Specify only one of the two variables

variable name	units	description	usage
`diff_vert_bg_cm2ps`	$\mathrm{cm}^2\,\mathrm{s}^{-1}$	vertically uniform value of background vertical diffusivity	Usable only if a vertically uniform value is intended
`file_diff_vert_1d_cm2ps`		file having vertical 1D distribution	cannot be specified with the above

Table12.2 Namelist `nml_vmbg3d`. Specify when `VMBG3D` is selected

variable name	units	description	usage
`file_vmix_3d`		file having 3D distribution	
`imvm`		east-west data size	
`jmvm`		north-south data size	
`l_vmintpol`	logical	interpolate input data to model grid points	
`slatvm`		latitude of the southern end	only if `l_vmintpol` = `.true.`
`slonvm`		longitude of the western end	only if `l_vmintpol` = `.true.`
`dlatvm`		uniform grid spacing in the meridional direction	only if `l_vmintpol` = `.true.`
`dlonvm`		uniform grid spacing in the zonal direction	only if `l_vmintpol` = `.true.`

In addition to the static background profiles, the following processes give time-varying vertical diffusivity coefficients at every model time step.

- Surface mixed layer models (`TURBULENCE` option).
- Vertical component of isopycnal diffusion (`ISOPYCNAL` option).
- Enhanced diffusivity ($= 1.0\,\mathrm{m}^2\,\mathrm{s}^{-1} = 10^4\,\mathrm{cm}^2\,\mathrm{s}^{-1}$) where the stratification is unstable (`DIFAJS` option).
- Enhanced diffusivity around rivermouths to avoid too low salinity if the model receives river run-off option (`RUNOFF` option). This scheme is especially needed when positive salinity is not guaranteed by a tracer advection algorithm. See Table 12.3 for how to specify the mixing.

The vertical diffusion for "this" time step is taken as the largest of the above estimations.（合計ではない).

Table12.3 Namelist `nml_vmix_river`. Specify when `RUNOFF` is selected

variable name	units	description	usage
`l_enhance_vmix_rivmouth`	logical	diffusivity is enlarged around the rivermouth	default = `.false.`
`diff_max_vmixriv_cm2ps`	$\mathrm{cm}^2\,\mathrm{sec}^{-1}$	maximum value of the enlarged diffusivity ($= \kappa_{\mathrm{rivmax}}$)	default = $1 \times 10^4\,\mathrm{cm}^2\,\mathrm{sec}^{-1}$
`depth_max_vmixriv_cm`	cm	vertical diffusion is enlarged from surface to this depth, this is also used by subroutine `salinity_limit`	default = 30×10^2 cm
`para_vmixriv_1`	1	parameter for the enlarged vertical diffusion formula ($= a$). See below.	default = 10
`para_vmixriv_2`	1	parameter for the enlarged vertical diffusion formula ($= b$). Enlarged diffusion is calculated as $\kappa_{\mathrm{riv}} = \min(a^{\log_{10} W_{\mathrm{river}} + b}, \kappa_{\mathrm{rivmax}})$, where W_{river} is river discharge rate in $\mathrm{cm\,sec}^{-1}$	default = 7

12.2 Convective adjustment

Convective adjustment is realized by replacing the density (temperature and salinity) that is statically unstable (the upper density exceeds the lower density) in a water column with the averaged density between neighboring levels (neutralization), considering that interior convection occurs in that place. Most of the realistic phenomena represented by the convective adjustment include the developing mixed layer due to surface cooling during winter. Convective adjustment also includes the case in which dense bottom water flows out the sill and flows down along the slope. Moreover, the convective adjustment includes the practical effect that it suppresses disturbances caused by the numerical calculation error and smoothes the distribution.

In general, there are three numerical schemes for convective adjustment.

1. In the simplest one, adjustment is done for a pair of two neighboring levels, and then for a pair of another two neighboring levels. By repeating this procedure, it attempts to neutralize the density in the unstable part. This procedure is simple at each step, but it has a defect that the finite-time repetition does not necessarily guarantee reaching the complete averaged value. Therefore, this scheme is not used in MRI.COM.
2. In the second scheme, adjustment is done by assigning a high vertical diffusivity between the two levels that are statically unstable and by solving the vertical diffusion term using an implicit method. This method cannot remove the unstable condition completely in one procedure. However, it has good calculation efficiency for the case where the model has a high vertical diffusivity already due to the mixed layer or isopycnal diffusion schemes and thus needs an implicit method to solve it. In MRI.COM, this scheme is invoked by specifying `DIFAJS` option. The vertical diffusivity between the unstable grid points is set to 10^4 cm^2 s^{-1}. This scheme is the most standard used in ocean models worldwide.
3. In the third scheme, the unstable part is first neutralized. The stability at the top and bottom of the neutralized column is then reexamined. If the unstable condition remains, the part including the already-neutralized column is re-neutralized. This procedure continues until the instability at the top and bottom of the neutralized column disappears. This method can remove the unstable part completely and thus is called "Complete Convection," but it requires a number of iterations, the vertical level size minus one, at maximum. The third method, which is the default scheme in MRI.COM, is explained below (Ishizaki, 1997).

12.2.1 Algorithm

In order to minimize the judgment process ("IF" statement) and replace it by arithmetic calculation, this scheme defines two integer indices, α_k and β_k, at the layer boundaries, and six real variables TU_k, TL_k, SU_k, SL_k, VU_k, and VL_k, $(k = 1, KM-1)$, in addition to the vertical rows of temperature, salinity, and density T_k, S_k, R_k, $(k = \frac{1}{2}, KM - \frac{1}{2})$ (KM is the number of levels; see Figure 12.1). The level at the vertical boundary of a T-cell corresponds to the integer k. The index α_k indicates an unstable part within a water column: $\alpha_k = 1$ if it is unstable at the level between $k-\frac{1}{2}$ and $k+\frac{1}{2}$, and $\alpha_k = 0$ if it is neutral or stable. The index β_k memorizes the mixed part: $\beta_k = 1$ at the boundary where it is neutral as a result of mixing, and $\beta_k = 0$ elsewhere. Variables TU_k, SU_k, and VU_k and TL_k, SL_k, and VL_k are temperature, salinity and volume accumulated by multiplying α above the level k and below the level k, respectively, and are expressed by the following recursive relation.

$$
\begin{aligned}
VU_1 &= \alpha_1 V_{\frac{1}{2}}, \\
VU_2 &= \alpha_2 (V_{1+\frac{1}{2}} + \alpha_1 V_{\frac{1}{2}}) = \alpha_2 (V_{1+\frac{1}{2}} + VU_1), \\
&\cdots, \\
VU_k &= \alpha_k (V_{k-\frac{1}{2}} + VU_{k-1}), \\
&\cdots, \\
VU_{KM-1} &= \alpha_{KM-1} (V_{KM-1-\frac{1}{2}} + VU_{KM-2}),
\end{aligned}
\tag{12.6}
$$

and

$$
\begin{aligned}
VL_{KM-1} &= \alpha_{KM-1} V_{KM-\frac{1}{2}}, \\
VL_{KM-2} &= \alpha_{KM-2} (V_{KM-1-\frac{1}{2}} + \alpha_{KM-1} V_{KM-\frac{1}{2}}) = \alpha_{KM-2} (V_{KM-1-\frac{1}{2}} + VL_{KM-1}), \\
&\cdots, \\
VL_k &= \alpha_k (V_{k+\frac{1}{2}} + VL_{k+1}), \\
&\cdots, \\
VL_1 &= \alpha_1 (V_{1+\frac{1}{2}} + VL_2),
\end{aligned}
\tag{12.7}
$$

where $V_{k+\frac{1}{2}}$ denotes a volume of the cell at the level $k+\frac{1}{2}$. In a similar way, other quantities are expressed as follows:

$$
\begin{aligned}
&TU_1 = \alpha_1 T_{\frac{1}{2}} V_{\frac{1}{2}}, \quad TU_k = \alpha_k (T_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + TU_{k-1}), \\
&SU_1 = \alpha_1 S_{\frac{1}{2}} V_{\frac{1}{2}}, \quad SU_k = \alpha_k (S_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + SU_{k-1}), \\
&TL_{KM-1} = \alpha_{KM-1} T_{KM-\frac{1}{2}} V_{KM-\frac{1}{2}}, \quad TL_k = \alpha_k (T_{k+\frac{1}{2}} V_{k+\frac{1}{2}} + TL_{k+1}), \\
&SL_{KM-1} = \alpha_{KM-1} S_{KM-\frac{1}{2}} V_{KM-\frac{1}{2}}, \quad SL_k = \alpha_k (S_{k+\frac{1}{2}} V_{k+\frac{1}{2}} + SL_{k+1}),
\end{aligned}
\tag{12.8}
$$

where $T_{k+\frac{1}{2}}$ and $S_{k+\frac{1}{2}}$ are temperature and salinity at the level $k+\frac{1}{2}$.

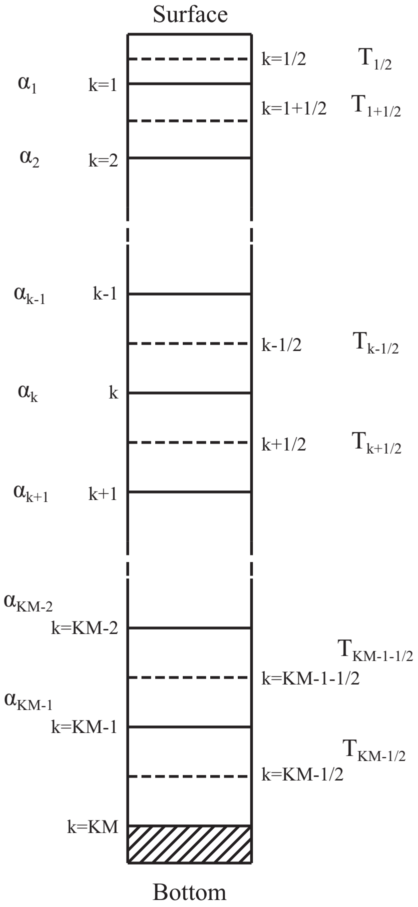

Figure12.1 Reference vertical grid points in Section 12.2

According to this definition, if $\alpha_k = 1$ and elsewhere 0, we get

$$
\begin{aligned}
VU_k + VL_k &= V_{k-\frac{1}{2}} + V_{k+\frac{1}{2}}, \\
TU_k + TL_k &= T_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + T_{k+\frac{1}{2}} V_{k+\frac{1}{2}}, \\
SU_k + SL_k &= S_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + S_{k+\frac{1}{2}} V_{k+\frac{1}{2}},
\end{aligned}
$$

indicating a volume and accumulated temperature and salinity in an unstable part and

$$
\begin{aligned}
TM_{k-\frac{1}{2},k+\frac{1}{2}} &= \frac{TU_k + TL_k}{VU_k + VL_k}, \\
SM_{k-\frac{1}{2},k+\frac{1}{2}} &= \frac{SU_k + SL_k}{VU_k + VL_k},
\end{aligned}
\tag{12.9}
$$

are volume averaged temperature and salinity, respectively.

If the level k constitutes a series of the unstable part, the same equation holds for the averaged temperature and salinity. For example, let $\alpha_{k-1} = \alpha_k = 1$ and $\alpha_{k-2} = \alpha_{k+1} = 0$,

$$
\begin{aligned}
VU_{k-1} + VL_{k-1} &= VU_k + VL_k \\
&= V_{k-1-\frac{1}{2}} + V_{k-\frac{1}{2}} + V_{k+\frac{1}{2}}, \\
TU_{k-1} + TL_{k-1} &= TU_k + TL_k \\
&= T_{k-1-\frac{1}{2}} V_{k-1-\frac{1}{2}} + T_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + T_{k+\frac{1}{2}} V_{k+\frac{1}{2}}, \\
SU_{k-1} + SL_{k-1} &= SU_k + SL_k \\
&= S_{k-1-\frac{1}{2}} V_{k-1-\frac{1}{2}} + S_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + S_{k+\frac{1}{2}} V_{k+\frac{1}{2}},
\end{aligned} \tag{12.10}
$$

and

$$
\begin{aligned}
TM_{k-1-\frac{1}{2},k+\frac{1}{2}} &= \frac{TU_{k-1} + TL_{k-1}}{VU_{k-1} + VL_{k-1}} = \frac{TU_k + TL_k}{VU_k + VL_k}, \\
SM_{k-1-\frac{1}{2},k+\frac{1}{2}} &= \frac{SU_{k-1} + SL_{k-1}}{VU_{k-1} + VL_{k-1}} = \frac{SU_k + SL_k}{VU_k + VL_k}.
\end{aligned} \tag{12.11}
$$

These are averages of the three layer, $k - 1 - \frac{1}{2}$, $k - \frac{1}{2}$, and $k + \frac{1}{2}$.

12.2.2 Numerical procedure

In summary, numerical procedures are summarized as follows.

[1] Density is calculated at the intermediate depth between adjacent levels using (A) upper level temperature and salinity and (B) lower level ones. If the density using (A) is larger than using (B), $\alpha(\alpha^1)$ is replaced by 1, otherwise by 0. At this stage, $\beta(\beta^0)$ is set to 0, where the superscript denotes the number of the iteration.

After this preprocessing, the following procedure (represented by n-th) is repeated until the instability is removed.

[2] Based on equations (12.6) to (12.8), VU, TU, SU, VL, TL, and SL are calculated using α^n for a water column that includes an unstable part.

[3] The vertical mean TM and SM are calculated for the unstable part using equation (12.9) and substituted for the original values of T and S. This change modifies the density at the intermediate depth in [1].

[4] The value of α^n is stored in β^n. $\beta^n = 1$ is set if $\alpha^n = 1$, or $\alpha^n = 0$ and $\beta^{n-1} = 1$, and otherwise $\beta^n = 0$. This is presented by the following:

$$\beta_k^n = \alpha_k^n + \beta_k^{n-1}(1 - \alpha_k^n). \tag{12.12}$$

[5] The static stability is judged only for $\beta_k^n = 0$. Let $\alpha_k^{n+1} = 1$ if statically unstable, and 0 otherwise. If there is no unstable part, the procedure for that water column is completed.

[6] For a water column which still includes an unstable part, modification for α_k^{n+1} is done using β_k^n by the following. After the procedure [2], any instability will be found only at the bottom of the part that is neutral as a result of prior mixing. In that case, the neutral part must be treated as an unstable part, that is, $\alpha_k^{n+1} = 1$. On the other hand, no more procedure is needed if the upper and lower end is stable, giving $\alpha_k^{n+1} = 0$. This is done by a recursive formula going down and up in the following.

$$
\begin{aligned}
&\gamma_1 = \alpha_1^{(n+1)}, \quad \gamma_k = \alpha_k^{(n+1)} + (1 - \alpha_k^{(n+1)})\beta_k^{(n)}\gamma_{k-1} \\
&\alpha_{KM-1}^{(n+1)} = \gamma_{KM-1}, \quad \alpha_k^{(n+1)} = \gamma_k + (1 - \gamma_k)\beta_k^{(n)}\alpha_{k+1}^{(n+1)},
\end{aligned} \tag{12.13}
$$

where γ is a work variable, but may be treated as α itself in a FORTRAN program. Then, the procedure goes back to [2].

Table 12.4 shows an example of the case with six levels. Static instability is removed after the three-time iteration. The second column of α in the table is the result of the corrected α_k^{n+1} using β_k^n based on equation (12.13), as described in [6]. Note that $\beta_k^0 = 0$, though there is no description in the table.

Table12.4 Example of the convective adjustment procedure

n	k	α		VU	VL	VU+VL	TU+TL	β
1	1	1	1	$\mathbf{V}_{\frac{1}{2}}$	$\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{T}_{\frac{1}{2}}\mathbf{V}_{\frac{1}{2}}+\mathbf{T}_{1\frac{1}{2}}\mathbf{V}_{1\frac{1}{2}}+\mathbf{T}_{2\frac{1}{2}}\mathbf{V}_{2\frac{1}{2}}$	1
	2	1	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}$	$\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{T}_{\frac{1}{2}}\mathbf{V}_{\frac{1}{2}}+\mathbf{T}_{1\frac{1}{2}}\mathbf{V}_{1\frac{1}{2}}+\mathbf{T}_{2\frac{1}{2}}\mathbf{V}_{2\frac{1}{2}}$	1
	3	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
	5	1	1	$\mathbf{V}_{4\frac{1}{2}}$	$\mathbf{V}_{5\frac{1}{2}}$	$\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\mathbf{T}_{4\frac{1}{2}}\mathbf{V}_{4\frac{1}{2}}+\mathbf{T}_{5\frac{1}{2}}\mathbf{V}_{5\frac{1}{2}}$	1
2	1	0	1	$\mathbf{V}_{\frac{1}{2}}$	$\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\sum_{k=0}^{3}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	2	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}$	$\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\sum_{k=0}^{3}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	3	1	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\sum_{k=0}^{3}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	4	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	1
3	1	0	1	$\mathbf{V}_{\frac{1}{2}}$	$\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	2	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}$	$\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	3	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	4	1	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	5	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}$	$\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1

Chapter 13

Tracer Package Structure and Usage

MRI.COM handles a wide variety of tracers, including physical variables for temperature and salinity, ecosystem model variables, and passive tracers. The user can individually specify initial values, advection schemes, body and surface forcings, etc. for each tracer. This chapter provides an overview of the program as well as common methods for their specification.

13.1 Program package structure

Program packages relevant to tracers are listed as follows.

13.1.1 Tracer equation

Core/tracer_ctl.F90:	Controller of this package
Core/tracer.F90:	Main program of this package
Core/tracer_vars.F90:	Setting of tracer attributes
Core/upc_adv.F90:	Upcurrent advection scheme
Core/quick_adv.F90:	QUICK advection scheme (`QUICKADVEC`)
Core/utzq_adv.F90:	Combination of UTOPIA and QUICKEST advection scheme (`UTZQADVEC`)
Core/som_adv.F90:	Second order moment advection scheme (`SOMADVEC`)
Core/mpdata_adv.F90:	MPDATA advection scheme (`MPDATAADVEC`)
Core/ppm_adv.F90:	PPM advection scheme (`PPMADVEC`)
+Vvdimp/trcimp.F90:	Solver of the vertical diffusion part using the implicit method (`VVDIMP`)
+Isopycnal/ipcoef.F90:	Calculation of tensor components of neutral physics parameterization (`ISOPYCNAL`)
+Isopycnal/ipycmix.F90:	Calculation of tendency due to neutral physics parameterization (`ISOPYCNAL`)

13.1.2 Vertical mixing coefficients

Core/vmixcoef_ctl.F90:	Controller of the vertical mixing package
Core/vmixcoef.F90:	Main program of the vertical mixing package
Core/vmixcoef_vmbg.F90:	Estimation of background vertical diffusion coefficient
Core/vmixcoef_vars.F90:	Declaration of variables
+Runoff/vmixcoef_rivermouth.F90:	Estimation of vertical mixing coefficient around the river mouth

13.1.3 Stratification and convective adjustment

Core/strat_adjust_ctl.F90:	Controller of stratification and adjustment package
Core/stratification.F90:	Main program of calculation of stratification
Core/cnvajs.F90:	Main program of convective adjustment
Core/strat_adjust_vars.F90:	Declaration of variables

13.1.4 Reference state and restoring coefficient

Core/restore_cond_ctl.F90:	Controller of reference state and restoring coefficient
Core/restore_cond.F90:	Main program of reference state and restoring coefficient

Core/force_data.F90:	Service package that handles external forcing data

13.1.5 Passive tracers

Core/ptrc_ctl.F90:	Controller of passive tracer evolution
Core/ptrc.F90:	Main program of passive tracer evolution that mainly treats surface sources and sinks
Core/ptrc_subp.F90:	Sub package of passive tracer evolution that describes internal sources and sinks
+Ptrc/cfc.F90, sf6.F90, etc.	Sub package of passive tracer evolution that describes specialized processes of particular tracers

13.2 Handling the initial state

How to determine the initial state for temperature and salinity is specified in namelist `nml_tracer_run` and `nml_restart`. Parameters are listed on Table 13.1.

Table13.1 Namelist `nml_tracer_run`

variable name	units	description	usage
`l_rst_tracer_in`	logical	`.true.` : Read restart files specified by `nml_restart` for the initial condition. `.false.`: Start condition depends on the `l_rst_uni_strati`	Default is the same as `l_rst_in` of `nml_run_ini_state`.
`l_rst_uni_strati`	logical	`.false.`: Start from 3D-distribution at the starting time of reference data following `nml_tracer_data`. `.true.` : Start from uniform stratification created by the reference data following `nml_tracer_data`. Time average is conducted based on `start_rec_uni_strati` and `end_rec_uni_strati`.	if `l_rst_tracer_in` = `.false.`
`start_rec_uni_strati` `end_rec_uni_strati`	integer	uniform stratification is created by the average from `start_rec_uni_strati` data record to `end_rec_uni_strati` record.	if `l_rst_uni_strati` = `.true.`

13.3 Configuration of tracers

The attributes of each tracer such as name, advection scheme, restoring condition, reference data, and restoring coefficients, are specified using a structural type (`type_tracer_data` defined in `tracer_vars.F90`). The contents of this structural type are specified by namelist `nml_tracer_data`, which should be repeatedly defined as many times as the number of tracers that should be calculated. Tables 13.2 through 13.8 list the variables.

13.3.1 Name

List of effective names is found in subroutine `tracer_vars__set_num_and_name` of `tracer_vars.F90`

Table13.2 Namelist `nml_tracer_data`

variable name	units	description	usage
`name`	character	Name of tracer. Two tracers are necessary: "Potential Temperature" and "Salinity."	Case sensitive. For example, "potential temperature" is not correct.

13.3.2 Advection scheme

Following can be specified as the name of the advection scheme (`adv_scheme%name`).

- "upc" : weighted UP-Current advection scheme (always available)
- "quick" : QUICK advection scheme (`QUICKADVEC`)

- "utzq" : UTOPIA + ZQUICKEST schemes with ultimate limiter (UTZQADVEC)
- "som" : Second-Order Moment advection scheme (SOMADVEC)
- "ppm" : Picewise Parabolic advection scheme (PPMADVEC)
- "mpdata" : MPDATA advection scheme (MPDATAADVEC)

Table13.3 Namelist nml_tracer_data related to advection scheme

variable name	units	description	usage
adv_scheme%name	character	Name of the advection scheme used for the corrector phase.	Different advection schemes can be set for individual tracers.
adv_scheme%limiter_som_org	logical	Use flux limiter for SOM by Prather (1986)	SOMADVEC
adv_scheme%limiter_som_Merryfield03	logical	Use flux limiter for SOM by Merryfield and Holloway (2003)	SOMADVEC
adv_scheme%lrstin_som	logical	The SOM initial state of moments is read from file	SOMADVEC
adv_scheme%lrstout_som	logical	The SOM final state of moments is written to file	SOMADVEC
adv_scheme%limiter_ppm_org	logical	Use flux limiter for PPM by Colella and Woodward (1984)	PPMADVEC
adv_scheme%limiter_ppm_lin	logical	Use flux limiter for PPM by Lin et al. (1994)	PPMADVEC
adv_scheme%limiter_mpdata_nonoscillatory	logical	Apply flux limiter for MPDATA	MPDATAADVEC
adv_scheme%eps_lim_mpdata	same as tracer unit	Very small value to avoid zero division	MPDATAADVEC
adv_scheme%min_value_mpdata	same as tracer unit	Minimum value for tracer	MPDATAADVEC
adv_scheme_predictor%name	character	Name of the advection scheme used for the predictor phase. Only active tracers (θ&S) need to be configured.	Since accuracy is not so demanding, QUICKADVEC is recommended due to its small computational cost.

13.3.3 Restoring condition

The following are variables related to the restoring condition for a tracer.

Table13.4 Namelist nml_tracer_data related to restoring condition

variable name	units	description	usage
restore_conf%l_surf_restore	logical	restore condition at the surface is applied or not	default = .false.
restore_conf%l_body_restore	logical	restore condition in the interior is applied or not	default = .false.

13.3.4 Reference data

a. Three dimensional reference state for restoring

When the field of a tracer is restored to a reference state, the attributes of the reference state should be given by the variables listed on Table 13.5. This reference state is also used to produce an initial state for that tracer when its restart file is not available (See Table 13.1).

Table13.5 Namelist `nml_tracer_data` related to reference values for body forcing and initial condition

variable name	units	description	usage
`trcref_conf%file_data`	character	a file name that contains reference values for body forcing and initial condition.	
`trcref_conf%file_data_grid`	character	a file name of grid	needed if `linterp` = `.true`.
`trcref_conf%imfrc`	integer	grid size of data in x direction	
`trcref_conf%jmfrc`	integer	grid size of data in y direction	
`trcref_conf%kmfrc`	integer	grid size of data in z direction	
`trcref_conf%interval`	integer	regular time interval of data	positive value : unit is sec −1 : monthly −999 : steady forcing
`trcref_conf%num_data_max`	integer	the number of record contained in the file	
`trcref_conf%ifstart`	integer, dimension (6)	[ymdhms] of the first record of the input file	1999,1,1,0,0,0 when the first record is the average value of Jan 1999 and its data interval is monthly.
`trcref_conf%lrepeat`	logical	climatological data is repeatedly used	default = `.false`.
`trcref_conf%linterp`	logical	interpolate horizontally or not	default = `.false`.
`trcref_conf%linterp_v`	logical	interpolate vertically or not	default = `.false`.
`trcref_conf%ilinear`	integer	interpolation method	1 : linear, 2 : spline
`trcref_conf%luniform`	logical	data is horizontally uniform or not	default = `.false`.
`trcref_conf%luniform_v`	logical	data is vertically uniform or not	default = `.false`.
`trcref_conf%ldouble`	logical	input data is double or not	default = `.false`.
`trcref_conf%iverbose`	integer	standard output of progress	1 : extensive, 0 : minimum
`trcref_conf%ldefined`	logical	the input data is defined or not	default = `.false`.

Format of tracer reference / restoring data is shown in the following.

Format of tracer reference / restoring data (`trcref(_surf)_conf%file_data`)

```
integer(4), parameter :: imn = 12, nu = 99
integer(4) :: imfrc, jmftc, kmfrc          ! data size
character(128) :: file_data, fname_grid
real(4) :: ttlev(imfrc,jmfrc,kmfrc,imn)
real(8) :: alonf(imfrc), alatf(jmfrc), dpf(kmfrc)
logical :: linterp, linterp_v

! main data
open (unit=nu,file=file_data,access=direct,recl=4*imfrc*jmfrc*kmfrc)
do m = 1, imn
  write(unit=nu,rec=m) ttlev(:,:,:,m)
end do
close(nu)

! longitude/latitude of main data
if (linterp) then  ! If input data is horizontally interpolated in the model.
  open (unit=nu,file=file_grid)
  write(nu) alonf, alatf
  if (linterp_v) then ! If input data is vertically interpolated in the model.
    write(nu) dpf
  end if
  close(nu)
end if
```

b. Two dimensional reference state for surface restoring

When the surface field of a tracer is intended to be restored to a reference state, the attributes of the surface reference state should be given by the variables listed on Table 13.6.

Table13.6 Namelist `nml_tracer_data` related to reference values for surface restoring forcing.

variable name	units	description	usage
`trcref_surf_conf%file_data`	character	a file name that contains reference values for surface restoring forcing	
`trcref_surf_conf%file_data_grid`	character	a file name of grid	needed if `linterp` = `.true.`
`trcref_surf_conf%imfrc`	integer	grid size of data in x direction	
`trcref_surf_conf%jmfrc`	integer	grid size of data in y direction	
`trcref_surf_conf%interval`	integer	regular time interval of data	positive value : unit is sec −1 : monthly −999 : steady forcing
`trcref_surf_conf%num_data_max`	integer	the number of record contained in the file	
`trcref_surf_conf%ifstart`	integer, dimension (6)	[ymdhms] of the first record of the input file	1999,1,1,0,0,0 when the first record is the average value of Jan 1999 and its data interval is monthly.
`trcref_surf_conf%lrepeat`	logical	climatological data is repeatedly used	default = `.false.`
`trcref_surf_conf%linterp`	logical	interpolate horizontally or not	default = `.false.`
`trcref_surf_conf%ilinear`	integer	interpolation method	1 : linear, 2 : spline
`trcref_surf_conf%luniform`	logical	data is horizontally uniform or not	default = `.false.`
`trcref_surf_conf%ldouble`	logical	input data is double or not	default = `.false.`
`trcref_surf_conf%iverbose`	integer	standard output of progress	1 : extensive, 0 : minimum
`trcref_surf_conf%ldefined`	logical	the input data is defined or not	default = `.false.`

13.3.5 Restoring coefficient

a. Coefficient for three dimensional restoring

When the field of a tracer is restored to a reference state, the attributes of the file that contains restoring coefficients should be given by the variables listed on Table 13.7. Units of restoring coefficient is sec^{-1}.

Table13.7 Namelist `nml_tracer_data` related to restoring coefficient for body forcing.

variable name	units	description	usage
`rstcoef_conf%file_data`	character	a file name that contains restoring coefficient for body forcing.	
`rstcoef_conf%file_data_grid`	character	a file name of grid	needed if `linterp` = `.true.`
`rstcoef_conf%imfrc`	integer	grid size of data in x direction	
`rstcoef_conf%jmfrc`	integer	grid size of data in y direction	
`rstcoef_conf%kmfrc`	integer	grid size of data in z direction	
`rstcoef_conf%interval`	integer	regular time interval of data	positive value : unit is sec

Continued on next page

Table 13.7 – continued from previous page

variable name	units	description	usage
			-1 : monthly -999 : steady forcing
`rstcoef_conf%num_data_max`	integer	the number of record contained in the file	
`rstcoef_conf%ifstart`	integer, dimension (6)	[ymdhms] of the first record of the input file	1999,1,1,0,0,0 when the first record is the average value of Jan 1999 and its data interval is monthly.
`rstcoef_conf%lrepeat`	logical	climatological data is repeatedly used	default = `.false.`
`rstcoef_conf%linterp`	logical	interpolate horizontally or not	default = `.false.`
`rstcoef_conf%linterp_v`	logical	interpolate vertically or not	default = `.false.`
`rstcoef_conf%ilinear`	integer	interpolation method	1 : linear, 2 : spline
`rstcoef_conf%luniform`	logical	data is horizontally uniform or not	default = `.true.`
`rstcoef_conf%luniform_v`	logical	data is vertically uniform or not	default = `.false.`
`rstcoef_conf%ldouble`	logical	input data is double or not	default = `.true.`
`rstcoef_conf%iverbose`	logical	standard output of progress	1 : extensive 0 : minimum
`rstcoef_conf%ldefined`	logical	the input data is defined or not	default = `.false.`

Note that the default settings for `luniform` and `ldouble` are differ from those of the other attributes.

b. Coefficient for surface restoring

When the surface field of a tracer is intended to be restored to a reference state, the attributes of the file that contains surface restoring coefficients should be given by the variables listed on Table 13.6. Units of the surface restoring coefficient is sec^{-1}.

Table13.8 Namelist `nml_tracer_data` related to restoring coefficient for surface restoring forcing.

variable name	units	description	usage
`rstcoef_surf_conf%file_data`	character	a file name that contains restoring coefficient for for surface restoring forcing.	
`rstcoef_surf_conf%file_data_grid`	character	a file name of grid	needed if `linterp` = `.true.`
`rstcoef_surf_conf%imfrc`	integer	grid size of data in x direction	
`rstcoef_surf_conf%jmfrc`	integer	grid size of data in y direction	
`rstcoef_surf_conf%interval`	integer	regular time interval of data	positive value : unit is sec -1 : monthly -999 : steady forcing
`rstcoef_surf_conf%num_data_max`		the number of record contained in the file	
`rstcoef_surf_conf%ifstart`	integer, dimension (6)	[ymdhms] of the first record of the input file	1999,1,1,0,0,0 when the first record is the average value of Jan 1999 and its data interval is monthly.
`rstcoef_surf_conf%lrepeat`	logical	climatological data is repeatedly used	default = `.false.`
`rstcoef_surf_conf%linterp`	logical	interpolate horizontally or not	default = `.false.`
`rstcoef_surf_conf%ilinear`	integer	interpolation method	1 : linear, 2 : spline
`rstcoef_surf_conf%luniform`	logical	data is horizontally uniform or not	default = `.false.`
`rstcoef_surf_conf%ldouble`	logical	input data is double or not	default = `.false.`
`rstcoef_surf_conf%iverbose`	logical	standard output of progress	1 : extensive, 0 : minimum
`rstcoef_surf_conf%ldefined`	logical	the input data is defined or not	default = `.false.`

13.3.6 Example

Following is an example of namelist `nml_tracer_data` for Salinity. Some systems may not allow blank lines or comment lines in a namelist. In this case, you should delete them.

```
&nml_tracer_data
 name="Salinity",

! advection scheme
 adv_scheme%name="som",
 adv_scheme%limiter_som_org=.false.,
 adv_scheme%limiter_som_Merryfield03=.true.,
 adv_scheme%lrstin_som=.false.,
 adv_scheme%lrstout_som=.true.,
 adv_scheme_predictor%name="quick",

! restore_condition
 restore_conf%l_surf_restore=.true.
 restore_conf%l_body_restore=.false.

! trcref
 trcref_conf%file_data='../data/file_sclim.grd',
 trcref_conf%file_data_grid='dummy.d',
 trcref_conf%imfrc=184,
 trcref_conf%jmfrc=152,
 trcref_conf%kmfrc=51,
 trcref_conf%interval=-1,
 trcref_conf%ifstart=1947,12,1,0,0,0,
 trcref_conf%num_data_max=14,
 trcref_conf%lrepeat=.false.,
 trcref_conf%linterp=.false.,
 trcref_conf%ilinear=1,
 trcref_conf%iverbose=1,

! rstcoef
 rstcoef_conf%ldefined=.false.,

! trcref_surf
 trcref_surf_conf%file_data='../data/file_ssurf.grd',
 trcref_surf_conf%file_data_grid='dummy.d',
 trcref_surf_conf%imfrc=184,
 trcref_surf_conf%jmfrc=152,
 trcref_surf_conf%interval=-1,
 trcref_surf_conf%ifstart=1947,12,1,0,0,0,
 trcref_surf_conf%num_data_max=14,
 trcref_surf_conf%lrepeat=.false.,
 trcref_surf_conf%linterp=.false.,
 trcref_surf_conf%ilinear=1,
 trcref_surf_conf%iverbose=1

! rstcoef_surf
 rstcoef_surf_conf%file_data='../data/rstcoef_surf_s.grd',
 rstcoef_surf_conf%file_data_grid='dummy.d',
 rstcoef_surf_conf%imfrc=1,
 rstcoef_surf_conf%jmfrc=1,
 rstcoef_surf_conf%interval=-999,
 rstcoef_surf_conf%num_data_max=1,
 rstcoef_surf_conf%iverbose=1,
 rstcoef_surf_conf%luniform=.true.,
 rstcoef_surf_conf%ldouble=.true.,
/
```

Part V

Boundary Processes

Chapter 14

Sea surface fluxes

The governing equations introduced in Chapter 2 need boundary conditions. At the sea surface, they take the form of the flux boundary condition. Momentum fluxes for the equations of motion, heat and material fluxes for the temperature and other tracer equations, and fresh water fluxes for the continuity equation. These surface fluxes are the main driving force of oceanic motions. This chapter explains how they are treated in MRI.COM.

Ideally, the realistic momentum, heat and fresh water fluxes based on observation should be used when a realistic simulation is intended. However, because existing sea surface fluxes derived from observations include large errors and the ocean models have their own errors, it is not common for an ocean-sea ice model in stand-alone mode to be driven by fluxes without a feedback process. It is a common practice for an ocean-sea ice model to compute fluxes using a bulk formula with sea surface meteorological elements derived from atmospheric reanalysis data and the surface temperature and velocity of its own.

Section 14.1 describes momentum flux. How heat fluxes are treated is described in Sections 14.2 through 14.5. Treatment of fresh water fluxes is described in Section 14.6, including the computations of the equivalent fluxes under invariable first layer volume condition (Section 14.6.2). Sections 14.7 and 14.8 explain how heat and fresh water fluxes are used in temperature and salinity equations. Several options are available for the choice of the bulk formula to calculate momentum, latent and sensible heat, and evaporative fluxes. They are detailed in Section 14.9 along with a general formulation for bulk transfer coefficients. How to give the attributes of external forcing data to the model is explained in Section 14.10. Finally, some technical details are presented in Sections 14.11 and 14.12.

Here is a cautionary note. Although recent satellite observations enable us to obtain sea surface fluxes with high resolution in space and time, even higher accuracy is necessary for practical uses. For example, a bias of several W m^{-2} in heat flux may greatly affect the thickness of sea ice, meaning that accuracy on the order of several W m^{-2} is necessary to clarify climatic change (WGASF, 2000). Efforts in enhancing observations and evaluating sea surface fluxes based on various methods have been expanded globally. In the future, a high-resolution ocean model and a new scheme for advection and diffusion may be developed to improve the simulation capability. It is noted, however, that increased observation frequency does not necessarily guarantee improved accuracy of the fluxes (for example, it is unlikely that the accuracy of a bulk coefficient would be improved). Hence, it should be kept in mind that sea surface fluxes presently involve large uncertainties.

14.1 Momentum flux (surface stress)

14.1.1 Formulation

Surface forcing to the momentum equation, or surface momentum flux into the ocean, is in the form of wind stress (or stress from sea ice), and is treated as a body force to the first level velocity in the model algorithm

$$\frac{\partial}{\partial t}(u_1, v_1) = \ldots + \frac{(\tau_x, \tau_y)}{\rho_0 \Delta z_{\frac{1}{2}}}, \tag{14.1}$$

where (u_1, v_1) is horizontal velocity at the first level, $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer, ρ_0 is the reference density of sea water (Table 2.1), and τ_x and τ_y are zonal and meridional stresses.

Surface stress is an area weighted average of stresses due to surface wind and sea ice:

$$(\tau_x, \tau_y) = (1 - A)(\tau_{AO_x}, \tau_{AO_y}) + A(\tau_{IO_x}, \tau_{IO_y}), \tag{14.2}$$

where A is the area fraction of sea ice. $\overrightarrow{\tau_{AO}}$ and $\overrightarrow{\tau_{IO}}$ are air-ocean and ice-ocean stresses, respectively.

Wind stress (air-sea momentum flux) vector, $\overrightarrow{\tau_{AO}}$ is computed by using the wind vector of the surface air, $\overrightarrow{U_a}$, the first level velocity (surface current), $\overrightarrow{U_s}$, and a bulk transfer coefficient C_D (see Section 14.9 for computation) as follows:

$$\overrightarrow{\tau_{AO}} = \rho_a C_D |\overrightarrow{U_a} - \alpha \overrightarrow{U_s}|(\overrightarrow{U_a} - \alpha \overrightarrow{U_s}), \tag{14.3}$$

where ρ_a is the density of air and α is the contribution of the surface current to the calculation of relative surface wind. Usually, α is unity, but it can be tuned to reduce damping effect of the wind stress on the oceanic current (see Table 14.3).

See Chapter 17 for the computation of ice-ocean stress ($\overrightarrow{\tau_{IO}}$).

14.1.2 Numerical implementation

On implementing Eq. (14.3), the time level of the surface velocity must be the starting time level of the integration. Because the corrector substep is the main body of the LFAM3 time integration scheme (Section 4.3), we use the n-th step surface velocity at the starting time level of the corrector substep.

$$\overrightarrow{\tau_{AO}}^n = \rho_a C_D |\overrightarrow{U_a}^n - \alpha \overrightarrow{U_s}^n|(\overrightarrow{U_a}^n - \alpha \overrightarrow{U_s}^n). \tag{14.4}$$

The same treatment should be applied to the computation of ice-ocean stress. Note that surface fluxes including the momentum flux are calculated in the predictor substep and the same fluxes are reused in the corrector substep.

In the bulk formulae implemented in MRI.COM, bulk transfer coefficients are computed on the tracer point. Those on the velocity point are obtained as area-weighted averages from the surrounding tracer points.

14.1.3 Usage

a. Model option

By default, wind stress vector is given as external data and each component should be read from separate files. (Wind stress data is always required, and thus files filled with zeros are necessary when wind stress is not applied.) The input data should be in units of [dyn cm^{-2}]. When `TAUBULK` option is specified, the wind stress is calculated by using a bulk formula. In this option, components of wind vector at 10 m in units of [cm s^{-1}] should be prepared as external data, instead of the default wind stress data.

b. Input data

Time series of wind stress or surface wind vector data on a structured lattice should be prepared in the single precision, direct access format with a constant time interval. The data may be interpolated in time and space in the model. Attributes of the data are specified by `nml_force_data`. The name of the data and the units assumed by the model are listed on Table 14.1. The name of the data should be "U-wind" and "V-wind" regardless of wind stress or wind vector. See Section 14.10 for details on how to specify external forcing files.

Table14.1 Wind data to be read from namelist `nml_force_data`.

	name	units	usage
X-ward wind stress	U-wind	dyn cm^{-2}	if not `TAUBULK`
Y-ward wind stress	V-wind	dyn cm^{-2}	if not `TAUBULK`
X-ward wind speed	U-wind	cm s^{-1}	if `TAUBULK`
Y-ward wind speed	V-wind	cm s^{-1}	if `TAUBULK`

c. Namelist

At the initial stage of the model integration, wind stress or wind may be gradually set up from zero at the start and unity after a specified period. The set up period is specified in namelist `nml_force_windsetup` (Table 14.2).

Table14.2 Namelist `nml_force_windsetup`.

variable name	units	description	usage
`idays_wind_setup`	day	wind set up period in days	default : no tapering on wind

Contribution of the surface current to the calculation of relative surface wind in (14.3) may be tuned by namelist `nml_bulk_wind` (Table 14.3). This is because the use of full surface current may lead to too strong damping on swift surface current in some model settings.

Table14.3 Namelist `nml_bulk_wind`.

variable name	units	description	usage
`current_contri_factor`	1	Contribution of the surface current to the calculation of relative surface wind	default = 1.0

14.2 Heat flux

14.2.1 Formulation

There are four main components in surface heat flux: shortwave radiation (Q_{SH}), longwave radiation (Q_{LO}), latent (Q_{LA}), and sensible (Q_{SN}) heat flux. Specific expression for these fluxes is shown in the following sections. In addition to those fluxes, the heat exchanges with the sea ice (Q_{ice}) and the heat transport due to surface fresh water fluxes (F^{θ}_{WF}) contribute to surface heat flux. See Section 14.6 for fresh water fluxes and Chapter 17 for details of exchange with sea ice.

The total sea surface heat flux Q_{NET} is given by

$$Q_{NET} = Q_{SH} + Q_{LO} + Q_{LA} + Q_{SN} + Q_{\mathrm{ice}} + \rho_0 c_p F^{\theta}_{WF}, \tag{14.5}$$

where ρ_0 and c_p are density and specific heat of sea water. Note that downward heat flux is defined as positive.

14.2.2 Numerical implementation

The components of net heat flux (14.5) are broken into three terms,

$$Q_{NET} = Q_{SH} + Q_{OTHER} + \rho_0 c_p F^{\theta}_{WF}. \tag{14.6}$$

Shortwave radiation (Q_{SH}) penetrates into the ocean interior. Other terms (Q_{OTHER}) are absorbed in the first layer. Fresh water transport term ($\rho_0 c_p F^{\theta}_{WF}$) needs separate treatment because it should be consistent with the continuity equation and included in the solution procedures of the tracer advection term.

14.2.3 Usage

a. Model options

The `HFLUX` option must be selected to treat surface heat fluxes explicitly in the model.

b. Input data

Time series of shortwave and longwave radiation, surface air temperature, surface specific humidity, surface wind speed, and sea-level pressure on a structured lattice should be prepared. The data may be interpolated in time and space in the model. Attributes of the data are specified by `nml_force_data`. The name of the data and the units assumed by the model are listed on Table 14.4. See Section 14.10 for details on how to specify external forcing files at run time.

Table14.4 Heat flux data to be read from namelist `nml_force_data` when `HFLUX` option is selected.

elements	name	units	usage
Downward shortwave radiation	ShortWave	$\mathrm{erg\,s^{-1}\,cm^{-2}} = \times 10^{-3}\,\mathrm{W\,m^{-2}}$	
Downward longwave radiation	LongWave	$\mathrm{erg\,s^{-1}\,cm^{-2}}$	
Surface air temperature	TempAir	°C	
Surface air specific humidity	SphAir	1	
Scalar wind speed	ScalarWind	$\mathrm{cm\,s^{-1}}$	unnecessary if `TAUBULK`
Sea level pressure	SeaLevelPressure	hPa	also used for SLP

14.3 Shortwave radiation flux

14.3.1 Formulation

By default, the downward shortwave radiation (Q_{SH}^{down}) is read as external forcing data. A part of the downward shortwave radiation ($\alpha_o Q_{SH}^{\mathrm{down}}$) is reflected at the sea surface, and the remainder penetrates into the ocean interior as the net shortwave radiation flux ($Q_{SH}(z=\eta) = (1-\alpha_o)Q_{SH}^{\mathrm{down}}$), where α_o is the albedo at the sea surface and η is the position of sea surface in the fixed vertical coordinate. More than 50% of the radiation (in the near infra-red band) is absorbed within a depth of 1 m below the sea surface, but the remainder (in the visible and ultraviolet bands) penetrates further into the ocean interior and affects the subsurface temperature structure.

The penetrating shortwave radiation is expressed as a function of depth (z) in a fixed frame $Q_{SH}(z)$ and its vertical convergence $\partial Q_{SH}(z)/\partial z$ warms the sea water.

$$\frac{\partial \theta}{\partial t} = \ldots + \frac{1}{\rho_0 c_p}\frac{\partial Q_{SH}(z)}{\partial z}, \tag{14.7}$$

where ρ_0 is the density of sea water and c_p is the specific heat of sea water. There are three options regarding how the vertical profile of shortwave penetration is specified.

a. Standard scheme of penetration

According to Paulson and Simpson (1977), the shortwave radiation flux penetrating into the ocean interior is given by using the fraction of the near infra-red band (R) and the attenuation distances for the near infra-red band (ζ_1) and the visible and ultraviolet band (ζ_2) as,

$$Q_{SH}(z) = Q_{SH}(0)[R\exp\{(-\eta+z)/\zeta_1\} + (1-R)\exp\{(-\eta+z)/\zeta_2\}], \tag{14.8}$$

where we set $R = 0.58$, $\zeta_1 = 35\,\mathrm{cm}$, and $\zeta_2 = 2300\,\mathrm{cm}$, using the optical properties of Water Type I based on the classification by Jerlov (1976). $Q_{SH}(0)$ is the net solar radiation flux at the sea surface.

b. A penetration scheme that takes into consideration of the insolation angle

The SOLARANGLE option enables us to include the effect of the insolation angle on the shortwave radiation flux in the oceanic interior. This scheme is based on Ishizaki and Yamanaka (2010). In this case, if the depth (z) on the r.h.s. of Equation (14.8) is replaced by the penetrating distance from the sea surface, the shortwave radiation is given by

$$Q_{SH}(z) = Q_{SH}(0)[R\exp\{(-\eta+z)/(\zeta_1\sin\theta_w)\} + (1-R)\exp\{(-\eta+z)/(\zeta_2\sin\theta_w)\}], \tag{14.9}$$

where θ_w is the penetrating angle in the ocean interior. When SOLARANGLE option is specified in coupled models, Equation (14.9) is replaced by

$$Q_{SH}(z) = Q_{SH_b}(z) + Q_{SH_d}(z), \tag{14.10}$$

where $Q_{SH_b}(z)$ and $Q_{SH_d}(z)$ are the shortwave radiation fluxes due to direct (beam) and diffuse solar radiation. Those fluxes are expressed as follows:

$$Q_{SH_b}(z) = Q_{SH_b}(0)[R_b\exp\{(-\eta+z)/(\zeta_1\sin\theta_w)\} + (1-R_b)\exp\{(-\eta+z)/(\zeta_2\sin\theta_w)\}], \tag{14.11}$$

$$Q_{SH_d}(z) = Q_{SH_d}(0)[R_d\exp\{(-\eta+z)/\zeta_1\} + (1-R_d)\exp\{(-\eta+z)/\zeta_2\}], \tag{14.12}$$

where R_b and R_d are the ratios of near infra-red to total radiation in direct (beam) and diffuse solar radiation, respectively. $Q_{SH_b}(0)$ and $Q_{SH_d}(0)$ are the net direct (beam) and diffuse solar radiation fluxes at the sea surface.

c. A penetration scheme that takes into consideration of chlorophyll concentration

Recent studies indicate that solar radiation absorption and local heating within the upper ocean are strongly influenced by the chlorophyll concentration. By specifying CHLMA94 and NPZD options, we can use a shortwave penetration model with the chlorophyll concentration (Morel and Antoine, 1994). In this scheme, the shortwave radiation flux penetrating into the ocean interior is given by

$$Q_{SH}(z) = Q_{SH}(0)[R\exp\{(-\eta+z)/\zeta_0\} + (1-R)(V_1\exp\{(-\eta+z)/\zeta_1\} + V_2\exp\{(-\eta+z)/\zeta_2\})]. \tag{14.13}$$

The first exponential is for the infra-red waveband (> 750 nm), which is not influenced by biological materials and attenuates with a distance of $\zeta_0 = 26.7\,\mathrm{cm}$. The second and third exponentials are for the visible and ultraviolet bands (< 750 nm). V_1, V_2, ζ_1, and ζ_2 are calculated from an empirical relationship as a function of chlorophyll concentration (chl [$\mathrm{mg \cdot m^{-3}}$]) as follows:

$$V_1 = 0.321 + 0.008C + 0.132C^2 + 0.038C^3 - 0.017C^4 - 0.007C^5, \tag{14.14}$$

$$V_2 = 0.679 - 0.008C - 0.132C^2 - 0.038C^3 + 0.017C^4 + 0.007C^5, \tag{14.15}$$

$$\zeta_1 = 1.540 - 0.197C + 0.166C^2 - 0.252C^3 - 0.055C^4 + 0.042C^5, \tag{14.16}$$

$$\zeta_2 = 7.925 - 6.644C + 3.662C^2 - 1.815C^3 - 0.218C^4 + 0.502C^5, \tag{14.17}$$

where $C = \log_{10}(chl)$. It is noted that $V_1 + V_2 = 1$.

When `SOLARANGLE` option is added to the options mentioned above, the shortwave radiation is slightly modified by

$$\begin{aligned} Q_{SH}(z) = &Q_{SH}(0)[R\exp\{(-\eta+z)/(\zeta_0\sin\theta_w)\} \\ &+ (1-R)(V_1\exp\{(-\eta+z)/(\zeta_1\sin\theta_w)\} + V_2\exp\{(-\eta+z)/(\zeta_2\sin\theta_w)\})]. \end{aligned} \tag{14.18}$$

In coupled models, Equation (14.18) is replaced by

$$Q_{SH}(z) = Q_{SH_b}(z) + Q_{SH_d}(z), \tag{14.19}$$

where $Q_{SH_b}(z)$ and $Q_{SH_d}(z)$ are the shortwave radiation fluxes due to direct (beam) and diffuse solar radiation. These fluxes are expressed as follows:

$$\begin{aligned} Q_{SH_b}(z) = &Q_{SH_b}(0)[R_b\exp\{(-\eta+z)/(\zeta_0\sin\theta_w)\} \\ &+ (1-R_b)(V_1\exp\{(-\eta+z)/(\zeta_1\sin\theta_w)\} + V_2\exp\{(-\eta+z)/(\zeta_2\sin\theta_w)\})], \end{aligned} \tag{14.20}$$

$$\begin{aligned} Q_{SH_d}(z) = &Q_{SH_d}(0)[R_d\exp\{(-\eta+z)/\zeta_0\} \\ &+ (1-R_d)(V_1\exp\{(-\eta+z)/\zeta_1\} + V_2\exp\{(-\eta+z)/\zeta_2\})], \end{aligned} \tag{14.21}$$

where R_b and R_d are the ratios of near infra-red to total radiation for direct (beam) and diffuse solar radiation, respectively.

d. Albedo schemes

There are five options for the sea surface albedo. The first option is a constant value, which should be specified as `alb` in namelist `nml_albedo_ocean`.

The second option is based on Large and Yeager (2009) and is given by

$$\alpha_o = 0.069 - 0.011\cos(2\phi), \tag{14.22}$$

where ϕ is latitude.

The third option is based on Barker and Li (1995) and is given by

$$\alpha_o = 0.06 + 0.0421x^2 + 0.128x^3 - 0.04x^4 + \left(\frac{3.12}{5.68+U} + \frac{0.074x}{1.0+3.0U}\right)x^5, \tag{14.23}$$

where $x = 1 - \sin\theta_a$ (θ_a is a height angle of the Sun), and U is the surface wind speed [$\mathrm{m\,s^{-1}}$].

The fourth option is based on Briegleb et al. (1986) and is given by

$$\alpha_o = \frac{0.026}{(\sin\theta_a)^{1.7} + 0.065} + 0.15\,(\sin\theta_a - 0.1)\,(\sin\theta_a - 0.5)\,(\sin\theta_a - 1.0)\,. \tag{14.24}$$

The fifth option is based on Taylor et al. (1996) and is given by

$$\alpha_0 = \frac{0.037}{1.1\,(\sin\theta_a)^{1.4} + 0.15}. \tag{14.25}$$

14.3.2 Numerical implementation

Shortwave flux is computed on T-cells, which means that variations of horizontal cross section due to topography (S) must be taken into account. The vertical profile of the shortwave penetration can be expressed as a function of the vertical depth from the sea surface. Discrete form of the absorption of shortwave heat for a grid cell $(i, j, k-\frac{1}{2})$ is

$$(\theta V)^{n+1}_{i,j,k-\frac{1}{2}} = (\theta V)^{n}_{i,j,k-\frac{1}{2}} + \ldots + \Delta t\Big(Q^n_{SHi,j,k-1}S_{i,j,k-\frac{1}{2}} - Q^n_{SHi,j,k}S_{i,j,k+\frac{1}{2}}\Big). \tag{14.26}$$

14.3.3 Usage

a. Model option

- SOLARANGLE must be selected to take the solar penetration angle into consideration
- CHLMA94 must be selected to take the effect of chlorophyll concentration into consideration. The chlorophyll concentration will be taken from either the biogeochemical model with NPZD option or climatology (name = ChlorophyllClimatology) with CHLCLIM option.

When CHLCLIM option is selected, time series of 2-D chlorophyll concentration should be prepared. The data may be interpolated in time and space in the model. See Section 14.10 for how to specify external forcing files.

b. Namelist

Namelists related to shortwave radiation are listed on Tables 14.5 through 14.7.

Table14.5 Namelist nml_albedo_ocean.

variable name	units	description	usage
alb	no (real8)	constant albedo	
albedo_scheme_name	no (character)	name of albedo scheme	'constant' or 'const' : constant given by alb (default) 'CORE' or 'Large and Yeager 2009' : Large and Yeager (2009) 'MRI-ESM1', 'Barker and Li 1995' or 'Barker95' : Barker and Li (1995) 'Briegleb et al 1986' or 'Briegleb86' : Briegleb et al. (1986) 'Taylor et al 1996' or 'Taylor96' : Taylor et al. (1996)
albedo_choice	no (integer)	choice of albedo scheme (obsolete)	1 : constant given by alb 2 : Large and Yeager (2009) 3 : Barker and Li (1995)

Table14.6 Namelist nml_solarangle.

variable name	units	description	usage
interval_shortwave_sec	sec	time interval of shortwave data	required
l_variable_zenith_ang	logical	use zenith angle with daily variation	default = .true.
l_const_zenith_ang_mean	logical	use average zenith angle from sunrise to sunset	valid if l_variable_zenith_ang = .false.
l_const_zenith_ang_noon	logical	use zenith angle at noon	valid if l_variable_zenith_ang = .false.
l_diurnal_var_shortwave	logical	daily variational amplitude of solar flux is calculated in this module	default = .false., If this is true, interval_shortwave_sec = 86400
l_penetr_ang_considered	logical	introduction of transmitted angle based on Snell's law	default = .true.

Table14.7 Namelist nml_frac_nearir.

variable name	units	description	usage
qsratio_nearIR	no (real8)	ratio of near infra-red band to the total shortwave flux (R)	default = 0.58

14.4 Longwave radiation flux

14.4.1 Formulation

Net longwave radiation into the ocean Q_{LO}^{net} is the residual between the downward longwave radiation Q_{LO}^{down} (external data given by user) and the upward black body radiation of sea water with the sea surface temperature (θ_{surf}),

$$Q_{LO}^{\text{net}} = (1 - r)Q_{LO}^{\text{down}} - e_m\sigma(\theta_{\text{surf}} + 273.15)^4. \tag{14.27}$$

Here, e_m is the emissivity for sea water, which should be specified by user, r is the reflectance of longwave radiation, and $\sigma = 5.67 \times 10^{-5}$ erg $\cdot$ s^{-1} $\cdot$ cm^{-2} $\cdot$ K^{-4} is the Stefan-Boltzmann constant. The longwave reflectance is calculated as $r = 1 - a - t$, where a and t are the absorbance and the transmittance, respectively. Assuming that $a = e_m$ from the Kirchhoff's law and $t = 0$, we give the reflectance r by $1 - e_m$.

14.4.2 Numerical implementation

There is no special issue on the numerical implementation. The net longwave flux is absorbed in the first layer of the model. Before used to update the first level temperature, it is summed into an array (`qothr`) which gathers all heat flux components except for shortwave. Note that the sea surface (first model level) temperature at the starting time level is used to evaluate upward longwave flux in any time integration scheme.

14.4.3 Usage

a. Namelist

Emissivity (e_m) of black body radiation from the sea surface may be specified by user at run time with namelist `nml_emissivity_sea_water` (Table 14.8).

Table14.8 Namelist `nml_emissivity_sea_water`.

variable name	units	description	usage
`emissivity`	no (real8)	emissivity of upward longwave radiation from sea water	default = 1.0

14.5 Latent and sensible heat fluxes

14.5.1 Formulation

The bulk method is used to calculate latent and sensible heat fluxes. In the bulk method, latent heat flux (Q_{LA}) and sensible heat flux (Q_{SN}) are calculated using bulk transfer coefficients C_E and C_H (see Section 14.9),

$$Q_{LA} = -\rho_a L C_E |\Delta\vec{U}|(q_s - q_a), \tag{14.28}$$

$$Q_{SN} = -\rho_a c_{pa} C_H |\Delta\vec{U}|(T_s - \theta_a). \tag{14.29}$$

Here, ρ_a is the air density, L is the latent heat of evaporation, q_a is the specific humidity of the surface air, q_s is the saturated specific humidity of the sea surface temperature, θ_a is the surface air potential temperature (K), T_s is the absolute sea surface water temperature (K), and $\Delta\vec{U} = \vec{U}_a - \alpha\vec{U}_s$ is the vector difference between the surface wind and the first level velocity (surface current). As explained in relation to the wind stress (Section 14.1), α is the contribution of the surface current to the calculation of the relative wind vector. The quantity c_{pa} is the specific heat of air.

14.5.2 Numerical implementation

There is no special issue on the numerical implementation. The latent and sensible heat fluxes are absorbed in the first layer of the model. Before used to update the first level temperature, they are summed into an array (`qothr`) which gathers all heat flux components except for shortwave. Note that the sea surface (first model level) temperature at the starting time level is used to evaluate fluxes in any time integration scheme.

14.5.3 Usage

Model option

MRI.COM can provide two calculation methods for bulk transfer coefficients. One of them must be specified when `HFLUX` option is chosen.

- `BULKNCAR` for Large and Yeager (2004; 2009)
- `BULKECMWF` for ECMWF (2016b)

See Sections 14.9.2 and 14.9.3 for details.

14.6 Freshwater flux

14.6.1 Formulation

Freshwater flux through the sea surface (F^W) is caused mainly by precipitation (P), evaporation (E), river discharge (R), and formation-melting of sea ice (I):

$$F^W = P - E + R + I, \tag{14.30}$$

Precipitation and river discharge are given as input data. MRI.COM conventionally considers precipitation on the sea as rainfall and precipitation on sea ice as snowfall. It is possible to separately specify rainfall data and snowfall data ($P = P_{\mathrm{rain}} + P_{\mathrm{snow}}$; Table 14.9). Note that rainfall on the sea ice in this case passes through the sea ice and falls directly on the sea. Fresh water flux due to sea ice formation-melting is given by the sea ice model. Evaporation is computed by a bulk formula in the ocean model and it should be consistent with the latent heat flux:

$$E = \rho_a C_E |\Delta\vec{U}|(q_s - q_a) = -Q_{LA}/L. \tag{14.31}$$

Note that evaporation (E) is defined as positive upward.

Fresh water flux is treated as the surface boundary condition for the surface height equation:

$$\frac{\partial \eta}{\partial t} = \ldots + F^W. \tag{14.32}$$

Fresh water is added to/subtracted from the surface grid cell. Thickness of grid cells in the whole vertical column should be modified to adapt to this change in the surface height. Concentration of materials (tracer species) in a grid cell should be updated by taking into account of these volume changes. Knowledge about the contents of materials in the fresh water flux is required for the computation, which is explained below.

14.6.2 Material fluxes due to fresh water flux

Rainfall, evaporation, and river discharge ($P_{\mathrm{rain}} - E + R$) are assumed to have the first level temperature (θ_{surf}) and zero salinity. We assume that the snow temperature is 0°C and its salinity is 0. Melting heat of snowfall is supplied from the top layer of the ocean model. In addition, fresh water flux is simply added to the first layer volume. Thus, the downward temperature and salinity transports to the first layer due to the fresh water flux are given as follows:

$$F^{\theta}_{WF} = (P_{\mathrm{rain}} - E + R) \cdot \theta_{\mathrm{surf}} + F^{\theta}_{I} - F^{\theta}_{\mathrm{SnowMelt}}, \tag{14.33}$$

$$F^{S}_{WF} = F^{S}_{I}. \tag{14.34}$$

where F^{θ}_{I} and F^{S}_{I} are temperature and salinity transport due to the water exchange between the ocean and sea ice, respectively. Those fluxes are calculated in the sea ice model (Eqs. (17.252) and (17.255)). The salinity transport is induced only by the water exchange between the ocean and sea ice, and other water inflows are assumed to have zero salinity. These downward transports of temperature and salinity are treated as downward vertical advection terms at the top of the first ocean model layer (Eqs. (14.36) and (14.39)).

14.6.3 Numerical implementation

a. Suppressing long-term drift of sea water volume

Because the global mean of F^W should not necessarily be zero and there is no feedback mechanism in an ocean-sea ice system to balance the fresh water budget, the globally averaged volume of the ocean may continue to increase or decrease.

To avoid this, the integral of the globally averaged freshwater flux is set to zero by selecting WADJ option,

$$F^W_{ADJ} = F^W - \overline{F^W}, \tag{14.35}$$

where $\overline{F^W}$ is the global mean of F^W $(= P - E + R)$, where E includes sublimation over sea ice. Note that exchange with sea ice is not included in the computation of $\overline{F^W}$.

14.6.4 Usage

a. Model options

- WFLUX should be specified to explicitly treat fresh water fluxes in the model. In this case, precipitation data must be prepared as input data.
- RUNOFF enables river discharge to be treated. River run-off data should be given as external data.
- WADJ imposes conservation of sea water volume of the ocean - sea ice system.

b. Input data

When WFLUX and RUNOFF options are specified, time series of precipitation and river run-off data must be prepared, respectively. Users can use namelist nml_force_snowfall (Table 14.9) to control whether snowfall data is read individually or not. The freshwater flux read from "Precipitation" data is regarded as rainfall if the users explicitly specify the snowfall data. Data on a structured lattice should be prepared in the single precision direct access format with a constant time interval. Attributes of the data are specified by nml_force_data. Name and units assumed by the model are summarized in Table 14.10.

The data may be interpolated in time and space in the model. See Section 14.10 for how to specify external forcing files.

Table14.9 Namelist nml_force_snowfall.

variable name	units	description	usage
l_read_snowfall	logical	Read snowfall data in addition to precipitation (rainfall) data	default = .false.

Table14.10 Water flux data to be read from namelist nml_force_data.

element	name	units	usage
precipitation	Precipitation	$\mathrm{g\,s^{-1}\,cm^{-2}}$	WFLUX
river discharge	RiverDischargeRate	$\mathrm{g\,s^{-1}\,cm^{-2}}$	RUNOFF
snowfall	Snowfall	$\mathrm{g\,s^{-1}\,cm^{-2}}$	WFLUX; required when l_read_snowfall = .true. (Table 14.9)

c. Namelist

The river run-off data around Antarctica may be treated as the discharge of iceberg if sea ice is present in the grid. Namelist nml_mkflux specifies this behavior (Table 14.11).

Table14.11 Namelist nml_mkflux. See also Table 14.13.

variable name	units	description	usage
l_glacier	logical	if a rivermouth grid has sea ice then river discharge is converted into iceberg input from glacier	default = .false.

Continued on next page

Table 14.11 – continued from previous page

variable name	units	description	usage
`l_after_rivermouth_spread`	logical	river discharge is converted into iceberg after a smoothing operation of river discharge data	default = `.false.`

When `WADJ` option is selected, the global summation of surface fresh water fluxes, which is used for determining the offsetting flux, can be made MPI parallel independent by choosing `l_strict_waterflux_adjust` = `.true.` in namelist `nml_waterflux_adjust` (Table 14.12).

Table14.12 Namelist `nml_waterflux_adjust`.

variable name	units	description	usage
`l_strict_waterflux_adjust`	logical	water flux adjustment is made MPI parallel independent	default = `.false.`

14.7 Treatment in the temperature equation

14.7.1 Formulation

The contributions of surface forcing (F_z^θ), shortwave radiation flux (Q_{SH}) and the heat transport due to fresh water flux (14.33, F_{WF}^θ) to the first layer temperature are expressed as

$$\left.\frac{\partial\theta}{\partial t}\right|_{k=\frac{1}{2}} = \ldots + \frac{F_z^\theta \Delta A}{\Delta V} + \frac{1}{\rho_0 c_p}\frac{\partial Q_{SH}}{\partial z} + \frac{F_{WF}^\theta \Delta A}{\Delta V}, \tag{14.36}$$

where ΔA and ΔV are the horizontal area and the volume of the first layer of a T-cell, respectively. The term (F_z^θ) consists of the net sea surface heat flux except for the shortwave radiation flux (Q_{OTHER}), restoring of SST (θ_{surf}) to a specified value (θ^*), and heating associated with freezing of supercooled water $\theta_t^{\text{supercool}}$:

$$\frac{F_z^\theta \Delta A}{\Delta V} = \frac{Q_{OTHER}\Delta A}{\rho_0 c_p \Delta V} - \frac{\overline{\Delta z}_{\frac{1}{2}}}{\gamma_\theta}\frac{(\theta_{\text{surf}} - \theta^*)\Delta A}{\Delta V} + \theta_t^{\text{supercool}}. \tag{14.37}$$

The heat flux (Q_{OTHER}) has been converted to temperature transport using the reference density (ρ_0) and specific heat (c_p) of sea water. The second term on the r.h.s. is a restoring term to a specified SST. The parameter γ_θ is a restoring time (in units of seconds) and should be read from input file (see Chapter 13). The restoring term is converted to temperature transport by multiplying the (fixed) thickness of the first layer of a T-cell ($\overline{\Delta z}_{\frac{1}{2}}$).

By default, surface temperature forcing consists only of the restoring term. By specifying `HFLUX` option, sea surface heat fluxes are treated explicitly. The heat transport due to fresh water flux are included when `WFLUX` option is specified.

14.7.2 Numerical Implementation

The heat transport due to fresh water flux (F_{WF}^θ in 14.36) is incorporated in the advection schemes. Shortwave absorption is treated separately from other heat flux terms. Heat flux components except for the shortwave radiation are gathered in an array (`qothr`) and added to the tendency of the first layer temperature.

Heating of supercooled water ($\theta_t^{\text{supercool}}$) is calculated as follows:

$$\theta_t^{\text{supercool}} = -\frac{1}{\Delta t}(\theta_{\text{surf}} - \theta_{\text{freeze}}) \qquad \text{if} \quad \theta_{\text{surf}} < \theta_{\text{freeze}}. \tag{14.38}$$

When the temperature of the first layer is below the freezing point, the temperature is set to the freezing point and this heat comes from the latent heat release of the new sea ice formation. This is done in the sea ice model. Because the sea ice part uses the forward scheme to proceed in time, the sea surface temperature at the "current" time level must be given to the sea ice part from the ocean part. The result is that the temperature below the freezing point at the "current" time level is modified and adjusted to the freezing point by the sea ice part. This treatment is not compatible with the leap-frog time stepping employed by the ocean part. Specifically, treating this temperature adjustment process in a form of heat flux exchange may result in a numerical instability. Though the modification of sea surface temperature in the sea ice part is inconvenient for some purposes, we keep this treatment until a revision of time-stepping scheme is considered. (That is, the sea water temperature is directly corrected in the sea ice model, and its heating tendency is calculated in the ocean model afterward.) See Sec. 17.2.3 for sea ice formation by freezing of supercooled water in the sea ice model.

14.8 Treatment in the salinity equation

The fresh water flux (F_W) modifies the volume but not the salt content of the surface layer, changing the salinity of the surface layer (see also Chapter 7). Since fresh water flux is not related directly to the sea surface salinity, the model sea surface salinity might be far from the observed value. Hence, an adjustment is sometimes needed to restore the model sea surface salinity to the observed one (see the last term on the r.h.s. of the next equation). Generally, the model salinity is restored to the observed climatological sea surface salinity since no reliable data set of historical sea surface salinity is available at present.

14.8.1 Formulation

The contributions of explicit surface forcing (F_z^S) and the salt transport due to formation and melting of sea ice (14.34, F_{WF}^S) to the first layer salinity are expressed as

$$\left.\frac{\partial S}{\partial t}\right|_{k=\frac{1}{2}} = \ldots + \frac{F_z^S \Delta A}{\Delta V} + \frac{F_{WF}^S \Delta A}{\Delta V}, \tag{14.39}$$

where $\Delta V, \Delta A$ are the same as those in (14.36).

The explicit surface forcing (F_z^S) consists of restoring of SSS (S_{surf}) to a specified value (S^*) and salinity correction by the sea ice model S_t^{ice}:

$$\frac{F_z^S \Delta A}{\Delta V} = -\frac{\overline{\Delta z}_{\frac{1}{2}}}{\gamma_s} \frac{(S_{\mathrm{surf}} - S^*)\Delta A}{\Delta V} + S_t^{\mathrm{ice}}. \tag{14.40}$$

The parameter $1/\gamma_s$ is a reciprocal of a restoring time (in units of seconds) and should be read from input file (see Chapter 13). The restoring term is converted to salinity transport by multiplying $\overline{\Delta z}_{\frac{1}{2}}$, as in (14.37). Users have a variety of options for SSS restoring (see Table 14.14).

The salinity correction by the sea ice model S_t^{ice} is calculated as follows:

$$S_t^{\mathrm{ice}} = \frac{1}{\Delta t}(S_{\mathrm{ice}} - S_{\mathrm{surf}}) - \frac{F_{IO}^S \Delta A}{\Delta V}, \tag{14.41}$$

where S_{surf} is SSS at the current time step, S_{ice} is the modified SSS in the sea ice model, and F_{IO}^S is a salinity flux between ocean and sea ice without the water exchange [cm psu s^{-1}] (positive upward) (Eq. 17.95). $S_t^{\mathrm{ice}} = 0$ in an ocean only model without a sea ice model.

14.8.2 Numerical Implementation

The salinity transport due to fresh water flux is incorporated in the advection schemes. The first layer salinity is calculated by taking into consideration of the volume change of the grid cell due to the continuity equation.

14.8.3 Usage

a. Model option

- `WFLUX` treats fresh water fluxes explicitly in the model. Salinity will be affected by fresh water fluxes.
- `STABLERUN` restores sea surface salinity to climatology when salinity is lower than 5 pss

b. Namelist

Namelists related to salinity flux at the sea surface are listed on Tables 14.13 and 14.14.

Table14.13 Namelist `nml_mkflux`. See also Table 14.11.

variable name	units	description	usage
`rs_local`		how salinity for iceberg and snow is extracted	1 : taken from the local grid 0 : taken from global sea surface

Table14.14 Namelist `nml_sss_restore`.

variable name	units	description	usage
`sss_rst_diff_max`	pss	the maximum of the difference between model and reference sea surface salinity	default = 100 pss
`l_nosss_rst_seaice`	logical	no surface salinity restoring below sea ice	default = `.false.`
`l_nosss_rst_cstice`	logical	no surface salinity restoring at the coastal grid with sea ice	default = `.false.`
`l_sss_rst_cnsv`	logical	set globally integrated surface salinity restoring to zero	default = `.false.`
`l_strict_sss_rst_cnsv`	logical	global summation of surface salinity restoring flux is done serially so that the result becomes MPI parallel independent	default = `.false.`

14.9 Bulk transfer coefficient

This section briefly describes how to calculate sea surface fluxes using a bulk formula. For details, readers are referred to Kantha and Clayson (2000), Large and Yeager (2004) and ECMWF (2016b). Bulk transfer coefficient for air-ice interface is given in Section 17.2.1.

14.9.1 Formulation of bulk formula

Transfer processes through atmosphere and ocean boundaries are governed by complicated turbulent processes. Traditionally, these turbulent fluxes are parameterized as a bulk transfer law. This method does not explicitly solve the interaction between the atmosphere and the ocean, and attributes all unknown processes to bulk transfer coefficients.

Momentum ($|\overrightarrow{\tau_{AO}}|$), sensible heat ($Q_{SN}$), latent heat ($Q_{LA}$), and water vapor ($E$) fluxes are written in terms of turbulent components as follows:

$$|\overrightarrow{\tau_{AO}}| = -\rho_a \overline{wu} = \rho_a u_*^2, \tag{14.42}$$

$$Q_{SN} = -\rho_a c_{pa} \overline{w\theta} = \rho_a c_{pa} u_* \theta_*, \tag{14.43}$$

$$Q_{LA} = -\rho_a L_E \overline{wq} = \rho_a L_E u_* q_*, \tag{14.44}$$

$$E = -\rho_a \overline{wq} = \rho_a u_* q_* = Q_{LA}/L_E, \tag{14.45}$$

where a bar over two variables denotes a covariance between the turbulent component of vertical velocity (w) and that of each physical property, wind speed (u), air potential temperature (θ), and specific humidity (q). ρ_a is air density, u_* is friction velocity, $\theta_* \equiv Q_{SN}/(\rho_a c_{pa} u_*)$ is the temperature scale in the boundary layer, $q_* \equiv Q_{LA}/(\rho_a L_E u_*) = E/(\rho_a u_*)$ is the scale of specific humidity in the boundary layer, c_{pa} is the specific heat of air, and L_E is the latent heat of water vapor evaporation.

Sea surface fluxes are also represented using bulk formulae as follows:

$$\tau_{AO_x} = \rho_a C_{\mathrm{Dh}} |\vec{U}_a - \vec{U}_s| (U_a - U_s), \tag{14.46}$$

$$\tau_{AO_y} = \rho_a C_{\mathrm{Dh}} |\vec{U}_a - \vec{U}_s| (V_a - V_s), \tag{14.47}$$

$$Q_{SN} = -\rho_a c_{pa} |\vec{U}_a - \vec{U}_s| C_{\mathrm{Hh}} (T_s - \theta_a), \tag{14.48}$$

$$Q_{LA} = -\rho_a L_E |\vec{U}_a - \vec{U}_s| C_{\mathrm{Eh}} (q_s - q_a), \tag{14.49}$$

$$E = \rho_a |\vec{U}_a - \vec{U}_s| C_{\mathrm{Eh}} (q_s - q_a), \tag{14.50}$$

where $\vec{U}_a = (U_a, V_a)$ is the wind vector at a height of z_{h}, $\vec{U}_s = (U_s, V_s)$ is the current vector at the sea surface, τ_{AO_x} and τ_{AO_y} are the zonal and meridional component of wind stress $\overrightarrow{\tau_{AO}}$, respectively, and T_s is the absolute sea surface water temperature (K). The subscript "a" means the value at a height of $z = z_{\mathrm{h}}$, and the subscript "s" means the value at the sea surface. Specific humidity at sea surface q_s is calculated by Eq. (14.122). The contribution factor (α) of the surface current to the calculation of the relative wind vector explained in Section 14.1 is omitted here for brevity.

Parameter C_{Dh} is called a drag coefficient. C_{Hh} and C_{Eh} are transfer coefficients for heat and water vapor and are called the Stanton coefficient and the Dalton coefficient, respectively. These can be estimated by observed atmospheric elements (wind velocity, air temperature, and humidity) at a height in the boundary layer, not by observed turbulent fluxes, using the following Equations (14.51), (14.52), and (14.53) and the similarity law of Monin-Obukhov mentioned below,

$$C_{\mathrm{Dh}} = \frac{u_*^2}{|\vec{U}_a - \vec{U}_s|^2}, \tag{14.51}$$

$$C_{\mathrm{Hh}} = -\frac{Q_{SN}/\rho_a c_{pa}}{|\vec{U}_a - \vec{U}_s|(T_s - \theta_a)} = \frac{u_* \theta_*}{|\vec{U}_a - \vec{U}_s|(T_s - \theta_a)}, \tag{14.52}$$

$$C_{\mathrm{Eh}} = \frac{E/\rho_a}{|\vec{U}_a - \vec{U}_s|(q_s - q_a)} = \frac{u_* q_*}{|\vec{U}_a - \vec{U}_s|(q_s - q_a)}. \tag{14.53}$$

The similarity law of Monin-Obukhov assumes that physical properties in the atmosphere-ocean boundary layer (a layer with a thickness of several tens of meters located below the lower mixed layer of the atmosphere) have similar vertical profiles when they are scaled with the stability parameter and sea surface fluxes. Vertical profiles of wind speed (U), air temperature (θ), and humidity (q) can be written as follows:

$$\frac{\kappa z}{u_*}\frac{\mathrm{d}U(z)}{\mathrm{d}z} = \phi_M\left(\frac{z}{\mathcal{L}}\right), \tag{14.54}$$

$$\frac{\kappa z}{\theta_*}\frac{\mathrm{d}\theta(z)}{\mathrm{d}z} = \phi_H\left(\frac{z}{\mathcal{L}}\right), \tag{14.55}$$

$$\frac{\kappa z}{q_*}\frac{\mathrm{d}q(z)}{\mathrm{d}z} = \phi_E\left(\frac{z}{\mathcal{L}}\right), \tag{14.56}$$

where $\kappa = 0.4$ is the von Karman constant, and $\mathcal{L}$ is the Monin-Obukhov length scale

$$\mathcal{L} = \frac{u_*^3}{\kappa Q_b} = -\frac{u_*^3 \theta_v}{\kappa\, g\, \overline{w\theta_v}}. \tag{14.57}$$

In (14.57), $Q_b = -g\,\overline{w\theta_v}/\theta_v$ is the buoyancy flux, θ_v is the virtual temperature ($\theta_v = \theta\,(1 + c_{vt}\, q)$; see also (14.134)), and $\overline{w\theta_v}$ is the net heat flux including the water vapor flux:

$$\overline{w\theta_v} = \overline{w\theta}\,(1 + c_{vt}\, q) + c_{vt}\,\theta\,\overline{wq}. \tag{14.58}$$

In (14.54) to (14.56), $\zeta = z/\mathcal{L}$ is the Monin-Obukhov similarity variable, and $\phi_{M,H,E}$ are nondimensional functions for wind velocity, air temperature, and specific humidity. The nondimensional functions are assumed to be mathematically simple functions.

Now we integrate equations (14.54), (14.55), and (14.56) in the vertical direction and gains the following equations:

$$U(z) - U(z_0) = \frac{u_*}{\kappa}\int_{z_0}^{z}\frac{1}{z'}\phi_M\left(\frac{z'}{\mathcal{L}}\right)dz', \tag{14.59}$$

$$\theta(z) - \theta(z_{0\mathrm{T}}) = \frac{\theta_*}{\kappa}\int_{z_{0\mathrm{T}}}^{z}\frac{1}{z'}\phi_H\left(\frac{z'}{\mathcal{L}}\right)dz', \tag{14.60}$$

$$q(z) - q(z_{0\mathrm{E}}) = \frac{\theta_*}{\kappa}\int_{z_{0\mathrm{E}}}^{z}\frac{1}{z'}\phi_E\left(\frac{z'}{\mathcal{L}}\right)dz', \tag{14.61}$$

where z_0, $z_{0\mathrm{T}}$, and $z_{0\mathrm{E}}$ are roughness lengths for each physical property. These are the aerodynamic roughness lengths, which are the heights at which each physical characteristic value coincides with the value at the sea level boundary when extrapolated downward based on the Monin-Obukhov similarity law. In short, $U(z_0) = U_s$, $\theta(z_{0\mathrm{T}}) = T_s$, and $q(z_{0\mathrm{E}}) = q_s$. Eqs. (14.59), (14.60), and (14.61) are rewritten as follows:

$$U(z) - U_s = \frac{u_*}{\kappa}\left[\ln\frac{z}{z_0} - \Psi_M(\zeta, \zeta_M)\right], \tag{14.62}$$

$$\theta(z) - T_s = \frac{\theta_*}{\kappa}\left[\ln\frac{z}{z_{0\mathrm{T}}} - \Psi_H(\zeta, \zeta_H)\right], \tag{14.63}$$

$$q(z) - q_s = \frac{q_*}{\kappa}\left[\ln\frac{z}{z_{0\mathrm{E}}} - \Psi_E(\zeta, \zeta_E)\right], \tag{14.64}$$

where

$$\Psi_{M,H,E}(\zeta,\zeta_{M,H,E}) = \int_{\zeta_{M,H,E}}^{\zeta} \left[1-\phi_{M,H,E}(\zeta')\right]\frac{d\zeta'}{\zeta'}. \tag{14.65}$$

The lower ends of the integral in (14.65), ζ_M, ζ_H and ζ_E are calculated as $z_0/\mathcal{L}$, $z_{0\mathrm{T}}/\mathcal{L}$ and $z_{0\mathrm{E}}/\mathcal{L}$, respectively. When the stability of the boundary layer (ζ) is already known, bulk transfer coefficients can be estimated from the equations (14.51, 14.52, 14.53) using these roughness lengths and the relationships $U_a = U(z_h), \theta_a = \theta(z_h), q_a = q(z_h)$,

$$C_{\mathrm{Dh}} = \frac{\kappa^2}{\left[\ln\dfrac{z_{\mathrm{h}}}{z_0} - \Psi_M(\zeta_h,\zeta_M)\right]^2}, \tag{14.66}$$

$$C_{\mathrm{Hh,Eh}} = \frac{\kappa^2}{\left[\ln\dfrac{z_{\mathrm{h}}}{z_0} - \Psi_M(\zeta_h,\zeta_M)\right]\left[\ln\dfrac{z_{\mathrm{h}}}{z_{0\mathrm{T},0\mathrm{E}}} - \Psi_{H,E}(\zeta_h,\zeta_{H,E})\right]}. \tag{14.67}$$

14.9.2 Large and Yeager (2004; 2009): BULKNCAR

In neutral stability, $\phi_{M,H,E} = 1$ gives a good approximation for Eqs. (14.54), (14.55) and (14.56). Then, the equation (14.65) gives $\Psi_{M,H,E} = 0$. The bulk transfer coefficient is a function of the roughness length (z_0,$z_{0\mathrm{T}}$, and $z_{0\mathrm{E}}$ for each) only and is expressed as follows:

$$C_{\mathrm{DNh}} = \frac{\kappa^2}{\left[\ln\dfrac{z_{\mathrm{h}}}{z_0}\right]^2}, \tag{14.68}$$

$$C_{\mathrm{HNh,ENh}} = \frac{\kappa^2}{\left[\ln\dfrac{z_{\mathrm{h}}}{z_0}\right]\left[\ln\dfrac{z_{\mathrm{h}}}{z_{0\mathrm{T},0\mathrm{E}}}\right]} \tag{14.69}$$

$$= \frac{C_{\mathrm{DNh}}}{\left[1-\dfrac{1}{\kappa}C_{\mathrm{DNh}}^{1/2}\ln\dfrac{z_{0\mathrm{T},0\mathrm{E}}}{z_0}\right]} \tag{14.70}$$

$$= \frac{\kappa C_{\mathrm{DNh}}^{1/2}}{\left[\ln\dfrac{z_{\mathrm{h}}}{z_{0\mathrm{T},0\mathrm{E}}}\right]}. \tag{14.71}$$

Normally, these neutral bulk transfer coefficients are estimated at a height of 10 m. Non-neutral bulk transfer coefficients at an arbitrary height (z_{h}) are connected with the neutral bulk transfer coefficients at a height of 10 m as follows:

$$C_{\mathrm{Dh}} = \frac{C_{\mathrm{DN10}}}{\left[1+\dfrac{1}{\kappa}C_{\mathrm{DN10}}C_{\mathrm{DN10}}^{-1/2}\left(\ln\dfrac{z_{\mathrm{h}}}{z_{10}} - \Psi_M(\zeta_h)\right)\right]^2}, \tag{14.72}$$

$$C_{\mathrm{Hh,Eh}} = \frac{C_{\mathrm{HN10,EN10}}\left(\dfrac{C_{\mathrm{Dh}}}{C_{\mathrm{DN10}}}\right)^{1/2}}{\left[1+\dfrac{1}{\kappa}C_{\mathrm{HN10,EN10}}C_{\mathrm{DN10}}^{-1/2}\left(\ln\dfrac{z_{\mathrm{h}}}{z_{10}} - \Psi_{H,E}(\zeta_h)\right)\right]}, \tag{14.73}$$

where z_{10} means $z = 10$ m. If we approximate $\zeta_{M,H,E}$ by zero in Eq. (14.65), the equations above are independent from the roughness length.

The neutral bulk transfer coefficients at a height of 10 m (C_{DN10}, C_{HN10}, and C_{EN10}) are often estimated, according to the stability, as a function of velocity at 10 m. In the bulk formula based on Large and Yeager (2004; 2009), the bulk transfer coefficient at 10 m in the neutral stability is given as follows:

$$10^3 C_{\mathrm{DN10}} = \begin{cases} \frac{2.70}{U_{10\mathrm{N}}} + 0.142 + \frac{U_{10\mathrm{N}}}{13.09} - 3.14807\times 10^{-10}U_{10\mathrm{N}}^6, & U_{10\mathrm{N}} < 33\,\mathrm{m\,s^{-1}} \\ 2.34 & U_{10\mathrm{N}} \geq 33\,\mathrm{m\,s^{-1}} \end{cases} \tag{14.74}$$

$$10^3 C_{\mathrm{EN10}} = 34.6\sqrt{C_{\mathrm{DN10}}}, \tag{14.75}$$

$$10^3 C_{\mathrm{HN10}} = \begin{cases} 18.0\sqrt{C_{\mathrm{DN10}}}, & \text{stable } \zeta > 0, \\ 32.7\sqrt{C_{\mathrm{DN10}}}, & \text{unstable } \zeta \le 0, \end{cases} \tag{14.76}$$

Each turbulent flux is estimated at a height where wind speed is observed by transforming air temperature and humidity from the height where they are observed (z_θ and z_q, respectively) to the height where wind speed is observed. The observed heights are specified in namelist `nml_srfflx_dbase` (see Table 14.20).

Each physical property is estimated at a height where wind speed is observed in the following operations. First, calculate the virtual temperature θ_v according to (14.134) with air temperature at z_θ and specific humidity at z_q as follows:

$$\theta_v = \theta(z_\theta)(1 + c_{vt}\, q(z_q)) \tag{14.77}$$

Next, calculate the bulk transfer coefficient at 10 m in the neutral stability assuming that the first guess for the 10 m wind speed in the neutral stability is $U_{\mathrm{10N}} = |\Delta\vec{U}(z_u)| = |\vec{U_a}(z_u) - \vec{U_s}|$ (Equations (14.74) to (14.76)). The first guesses of the scales for the friction velocity, air temperature, and specific humidity are estimated assuming that these bulk coefficients are at the observed height and stability,

$$u_* = \sqrt{\frac{|\overrightarrow{\tau_{AO}}|}{\rho_a}} = \sqrt{C_{\mathrm{DN10}}}\,|\Delta\vec{U}(z_u)|, \tag{14.78}$$

$$\theta_* = \frac{Q_{SN}}{\rho_a c_p u_*} = \frac{C_{\mathrm{HN10}}}{\sqrt{C_{\mathrm{DN10}}}}\,(\theta(z_\theta) - T_s), \tag{14.79}$$

$$q_* = \frac{E}{\rho_a u_*} = \frac{C_{\mathrm{EN10}}}{\sqrt{C_{\mathrm{DN10}}}}\,(q(z_q) - q_s), \tag{14.80}$$

where q_s is the saturated specific humidity at sea surface temperature T_s.

Next, perform the iteration using the three Monin-Obukhov similarity variables, $\zeta_u = z_u/\mathcal{L}$, $\zeta_\theta = z_\theta/\mathcal{L}$, and $\zeta_q = z_q/\mathcal{L}$, and an integral of the nondimensional profile function for the vertical gradient of each physical property, $\Psi_M(\zeta)$ for momentum and $\Psi_H(\zeta)$ for scalars.

The Monin-Obukhov similarity variables are calculated as follows:

$$\zeta = \frac{\kappa g z}{u_*^2}\left[\frac{\theta_*}{\theta_v} + \frac{q_*}{(q(z_q) + c_{vt}^{-1})}\right]. \tag{14.81}$$

The integral of the non-dimensional profile function is expressed as

$$\Psi_M(\zeta) = \Psi_H(\zeta) = -5\zeta, \tag{14.82}$$

if it is stable ($\zeta \ge 0$), and

$$\Psi_M(\zeta) = 2\ln\left(\frac{1+X}{2}\right) + \ln\left(\frac{1+X^2}{2}\right) - 2\tan^{-1}(X) + \frac{\pi}{2}, \tag{14.83}$$

$$\Psi_H(\zeta) = 2\ln\left(\frac{1+X^2}{2}\right), \tag{14.84}$$

if it is unstable ($\zeta < 0$). In the above,

$$X = (1 - 16\zeta)^{1/4}. \tag{14.85}$$

Using these values, convert the wind speed to that at 10 m in the neutral stability, and convert the temperature and specific humidity to those at a height where the wind speed is observed,

$$U_{\mathrm{10N}} = |\Delta\vec{U}(z_u)|\left(1 + \frac{\sqrt{C_{\mathrm{DN10}}}}{\kappa}\left[\ln\frac{z_u}{z_{10}} - \Psi_M(\zeta_u)\right]\right)^{-1}, \tag{14.86}$$

$$\theta(z_u) = \theta(z_\theta) - \frac{\theta_*}{\kappa}\left[\ln\frac{z_\theta}{z_u} + \Psi_H(\zeta_u) - \Psi_H(\zeta_\theta)\right], \tag{14.87}$$

$$q(z_u) = q(z_q) - \frac{q_*}{\kappa}\left[\ln\frac{z_q}{z_u} + \Psi_H(\zeta_u) - \Psi_H(\zeta_q)\right], \tag{14.88}$$

where z_{10} means $z = 10\,\mathrm{m}$. Estimate the bulk coefficient at 10 m in the neutral stability using $U_{10\mathrm{N}}$, and then obtain the bulk coefficient at a height (z_u) where wind speed is observed,

$$C_{\mathrm{Du}} = \frac{C_{\mathrm{DN10}}}{\left[1 + \frac{1}{\kappa} C_{\mathrm{DN10}} C_{\mathrm{DN10}}^{-1/2} \left(\ln \frac{z_u}{z_{10}} - \Psi_M(\zeta_u)\right)\right]^2}, \tag{14.89}$$

$$C_{\mathrm{Hu}} = \frac{C_{\mathrm{HN10}} \left(\frac{C_{\mathrm{Du}}}{C_{\mathrm{DN10}}}\right)^{1/2}}{\left[1 + \frac{1}{\kappa} C_{\mathrm{HN10}} C_{\mathrm{DN10}}^{-1/2} \left(\ln \frac{z_u}{z_{10}} - \Psi_H(\zeta_u)\right)\right]}, \tag{14.90}$$

$$C_{\mathrm{Eu}} = \frac{C_{\mathrm{EN10}} \left(\frac{C_{\mathrm{Du}}}{C_{\mathrm{DN10}}}\right)^{1/2}}{\left[1 + \frac{1}{\kappa} C_{\mathrm{EN10}} C_{\mathrm{DN10}}^{-1/2} \left(\ln \frac{z_u}{z_{10}} - \Psi_H(\zeta_u)\right)\right]}. \tag{14.91}$$

Repeat the procedures to calculate the bulk coefficients using these bulk coefficients with temperature and specific humidity at $z = z_u$, and recalculate virtual temperature (14.134) and the scales for friction velocity, temperature, and specific humidity (Equations of (14.78), (14.79), and (14.80)). Note that Large and Yeager (2004) recommend to use a bulk water temperature a meter of more beneath the sea surface as sea surface temperature T_s in Eq. (14.48) because their formulation is based on fitting to observation data using a bulk water temperature not a skin temperature.

14.9.3 ECMWF (2016): BULKECMWF

Bulk formulae of Integrated Forecasting System (IFS) developed in European Centre for Medium-Range Weather Forecasts (ECMWF) (ECMWF, 2016b) are also available in MRI.COM. Bulk transfer coefficients are calculated by Eqs. (14.66) and (14.67). Turbulent surface fluxes are calculated as follows:

$$\tau_{AO_x} = \rho_a C_{\mathrm{Dh}} |U_{\mathrm{rel}}| (U_a - U_s), \tag{14.92}$$
$$\tau_{AO_y} = \rho_a C_{\mathrm{Dh}} |U_{\mathrm{rel}}| (V_a - V_s), \tag{14.93}$$
$$Q_{SN} = -\rho_a c_{pa} |U_{\mathrm{rel}}| C_{\mathrm{Hh}} (\theta_{vs} - \theta_v), \tag{14.94}$$
$$Q_{LA} = -\rho_a L_E |U_{\mathrm{rel}}| C_{\mathrm{Eh}} (q_s - q_a), \tag{14.95}$$
$$E = \rho_a |U_{\mathrm{rel}}| C_{\mathrm{Eh}} (q_s - q_a), \tag{14.96}$$

where $|U_{\mathrm{rel}}|$ denotes wind speed in the atmosphere bottom layer. This wind speed is calculated as follows:

$$|U_{\mathrm{rel}}| = \left(|\vec{U}_a - \vec{U}_s|^2 + w_\dagger^2\right)^{\frac{1}{2}}, \tag{14.97}$$

where $w_\dagger$ is a vertical velocity scale which parameterizes vertical motion associate with the free convection due to heating. The sensible flux Q_{SN} (14.94) is calculated from the virtual temperature θ_v and θ_{vs}. Those are the virtual temperature in the atmosphere bottom layer and that at the sea surface calculated with skin temperature T_{skin}, respectively. Note that the skin temperature usually differs from sea water temperature in the top layer of the ocean model. Details about calculation of $w_\dagger$, the virtual temperature and T_{skin} are described afterward.

The calculation procedures are summarized as follows.

1. Calculate the skin temperature (14.9.3a)
2. Read the free convection velocity scale ($w_\dagger$) at the previous time step
3. Calculate the wind speed in the atmosphere bottom layer ($|U_{\mathrm{rel}}|$)
4. Calculate bulk Richardson number (14.9.3b)
5. Read friction velocity (u_*) at the previous time step
6. Initialize Monin-Obukhov length
7. Calculate the bulk transfer coefficients by iteration method (repeat the following procedures `itr_z0_num` times)
   (a) Calculate roughness length $z_{0,0T,0E}$ from the friction velocity (14.9.3c)
   (b) Calculate the Monin-Obukhov length (14.9.3d)
   (c) Calculate the bulk transfer coefficients (14.66 and 14.67)
   (d) Calculate the friction velocity u_* from the bulk transfer coefficient of momentum C_{Dh} (14.9.3e)

8. Calculate the surface turbulent fluxes using bulk transfer coefficients
9. Calculate the free convection velocity scale $w_{\dagger}$ from the sensible and latent heat fluxes for the next time step (14.9.3f)

The free convection velocity scale $w_{\dagger}$ is assumed to be zero when its value at the previous step is unknown in the procedure (2). When the friction velocity u_* at the previous step is not available in the procedure (5), it is calculated by Eq. (14.117) described afterward with $C_{\mathrm{Dh}} = 0.001$. The Monin-Obukhov length $\mathcal{L}$ also requires initialization in the procedure (6). Its initial value is calculated as follows:

$$\frac{1}{\mathcal{L}} = \begin{cases} \exp\left(Ri_{bulk}/3 + 1\right) - 1.5 & (Ri_{bulk} < Ri_{lim}), \\ \zeta_{lim}/z_h & (otherwise). \end{cases} \tag{14.98}$$

This bulk formulae assume that the marine meteorological properties are observed at the same height. The observed height is specified by `alt_wind_meter` in namelist `nml_srfflx_dbase`. The other parameters `alt_temp_meter` and `alt_sphm_meter` are neglected and not used in the current formulae. Users may specify coefficients of the hybrid vertical coordinate in the bottom layer of the IFS atmosphere model. Then, the observed height is diagnosed from sea level pressure and the virtual temperature in the atmosphere bottom layer according to the equation below (ECMWF, 2016a):

$$z_h = \frac{R_{dry}\theta_v}{g}\left[1 - \frac{p_t}{p_s - p_t}\ln\left(\frac{p_s}{p_t}\right)\right], \tag{14.99}$$

where R_{dry} is the gas constants for dry air and p_s and p_t are the sea level pressure and pressure at the top interface of the atmosphere bottom layer, respectively. The top-interface pressure p_t is calculated as follows:

$$p_t = A + Bp_s. \tag{14.100}$$

The coefficients A and B are specified in namelist `nml_bulkecmwf`.

a. Calculation of skin temperature

The current bulk formulae use the skin temperature T_{skin} and it usually differs from the top-layer temperature of the ocean model. MRI.COM adopts a simple skin layer scheme implemented in the Japan Meteorological Agency's global atmosphere model. The skin temperature T_{skin} is calculated as follows:

$$T_{\mathrm{skin}} = T_o + \Delta T, \tag{14.101}$$

$$\Delta T = \max\left[0, \frac{Q_{SH}(z=\eta) + Q_{LO}^{\mathrm{net}}}{500} \cdot f_{wind}\right], \tag{14.102}$$

$$f_{wind} = \max\left[0.2, \frac{10 - |\vec{U_a} - \vec{U_s}|}{10}\right], \tag{14.103}$$

where T_o denotes the ocean top-layer potential temperature, $Q_{SH}(z=\eta)$ (14.3) and $Q_{LO^{\mathrm{net}}}$ (14.27) are net downward shortwave and net downward longwave at the sea surface in units of $\mathrm{W/m^2}$, respectively, and $|\vec{U_a} - \vec{U_s}|$ is relative wind speed in units of m/s.

b. Calculation of bulk Richardson number

The following equation gives the bulk Richardson number:

$$Ri_{bulk} = \left(\frac{g}{\overline{\theta_v}}\right)\frac{z_h(\theta_v - \theta_{vs})}{|U_{\mathrm{rel}}|^2}, \tag{14.104}$$

where θ_v and θ_{vs} denote the virtual temperature at the height of $z = z_h$ and the sea level (14.134). They are calculated as follows:

$$\theta_v = \theta_a(1 + c_{vt}q_a), \tag{14.105}$$

$$\theta_{vs} = T_{\mathrm{skin}}(1 + c_{vt}q_s), \tag{14.106}$$

where $\overline{\theta_v}$ is mean virtual temperature within the atmosphere bottom layer, which is given by $\overline{\theta_v} = 0.5(\theta_v + \theta_{vs})$.

c. Calculation of roughness length

The roughness lengths over the sea are calculated by the following equation with the friction velocity u_*:

$$z_0 = \alpha_M \frac{\nu}{u_*} + \alpha_{Ch} \frac{u_*^2}{g}, \tag{14.107}$$

$$z_{0T,0E} = \alpha_{H,E} \frac{\nu}{u_*}, \tag{14.108}$$

where $\alpha_M = 0.11, \alpha_H = 0.40, \alpha_E = 0.62$, ν is kinematic viscosity coefficient ($\nu = 1.5 \times 10^{-5} m^2 s^{-1}$), and α_{Ch} is the Charnock coefficient ($\alpha_{Ch} = 0.018$).

d. Calculation of the Monin-Obukhov length

The Monin-Obukhov length is calculated from the bulk Richardson number Ri_{bulk}. A stability parameter $\zeta = z/\mathcal{L}$ is associated with the bulk Richardson number as follows:

$$Ri_{bulk} = \zeta \cdot \frac{\left[\ln\left(\frac{z_h}{z_{0H}}\right) - \Psi_H\left(\zeta_h, \zeta_H\right)\right]}{\left[\ln\left(\frac{z_h}{z_{0M}}\right) - \Psi_M\left(\zeta_h, \zeta_M\right)\right]^2}. \tag{14.109}$$

When the bulk Richardson number calculated by Eq. (14.104) goes below the upper limit $Ri_{lim} = 6$, the Monin-Obukhov length is calculated by solving Eq. (14.109) with iteration method. The inverse of the Monin-Obukhov length is iteratively recalculated by the following equation:

$$\frac{1}{\mathcal{L}} = \frac{Ri_{bulk}\left[\ln\left(\frac{z_h}{z_{0M}}\right) - \Psi_M\left(\zeta_h, \zeta_M\right)\right]^2}{z_h\left[\ln\left(\frac{z_h}{z_{0H}}\right) - \Psi_H\left(\zeta_h, \zeta_H\right)\right]}. \tag{14.110}$$

The iteration stops if the bulk Richardson number calculated by Eq. (14.109) matches the value calculated by Eq. (14.104) within a predetermined error range or the iteration count exceeds the maximum number.

When the bulk Richardson number calculated by Eq. (14.104) is larger than Ri_{lim} or the Monin-Obukhov length calculated above satisfies $\zeta_h = z_h/\mathcal{L} > \zeta_{lim} = 0.5$, the Monin-Obukhov length is given by the following equation:

$$\frac{1}{\mathcal{L}} = \frac{\zeta_{lim}}{z_h}. \tag{14.111}$$

Stability functions $\Psi_{M,H,E}$ are calculated as follows:

$$\Psi_{M,H,E}(\zeta_h, \zeta_{M,H,E}) = \psi_{M,H,E}(\zeta_h) - \psi_{M,H,E}(\zeta_{M,H,E}). \tag{14.112}$$

Under unstable conditions ($\zeta < 0$),

$$\psi_M(\zeta) = \frac{\pi}{2} - 2\,\mathrm{atan}(X) + \ln\frac{(1+X)^2(1+X^2)}{8}, \tag{14.113}$$

$$\psi_{H,E}(\zeta) = 2\ln\left(\frac{1+X^2}{2}\right), \tag{14.114}$$

where $X = (1-16\zeta)^{1/4}$. Under stable conditions ($\zeta > 0$),

$$\psi_M(\zeta) = -b\left(\zeta - \frac{c}{d}\right)\exp(-d\zeta) - a\zeta - \frac{bc}{d}, \tag{14.115}$$

$$\psi_{H,E}(\zeta) = -b\left(\zeta - \frac{c}{d}\right)\exp(-d\zeta) - \left(1 + \frac{2}{3}a\zeta\right)^{1.5} - \frac{bc}{d} + 1, \tag{14.116}$$

where $a = 1, b = 2/3, c = 5$, and $d = 0.35$.

e. Calculation of friction velocity

The following equation gives the friction velocity with the bulk transfer coefficient of momentum C_{Dh}:

$$u_* = C_{\mathrm{Dh}}^{\frac{1}{2}}|U_{\mathrm{rel}}|. \tag{14.117}$$

f. Calculation of free-convection velocity scale $w_\dagger$

As noted above, the current bulk formulae take the free-convection velocity $w_\dagger$ into account for the wind speed of the atmosphere bottom layer. This velocity scale is calculated as follows:

$$w_\dagger = \left(z_i \frac{g}{\theta_a} Q_{ov} \right)^{\frac{1}{3}}, \tag{14.118}$$

$$Q_{ov} = \frac{Q_{SN}}{\rho_a c_{pa}} + \frac{c_{vt} \theta_a E}{\rho_a}, \tag{14.119}$$

where $c_{vt} = m_\mathrm{a}/m_\mathrm{w} - 1 = R_{vap}/R_{dry} - 1$ (see (14.135)), $R_{vap}(m_\mathrm{w})$ and $R_{dry}(m_\mathrm{a})$ are the gas constants (mean molecular weights) for water vapor and dry air, respectively, $z_i = 1000$ m, Q_{SN} denotes the sensible heat flux calculated by Eq. (14.94), and E is the water vapor flux calculated by Eq. (14.96). We use the surface fluxes Q_{SN} and E calculated at the previous step there.

14.9.4 Usage

a. Model option

- BULKNCAR for Large and Yeager (2004; 2009)
- BULKECMWF and SKINGSAM for ECMWF (2016b)

b. Namelist

Tables 14.15 and 14.16 show namelists available when BULKECMWF is specified.

Table14.15 Namelist nml_bulkecmwf)

variable name	units	description	usage
itr_z0_num	1	The number of iteration of roughness length calculation	default = 5
l_bulkecmwf_calc_altitude	logical	Calculate the observed height of marine meteorological properties	default = .true.
a_sfc_layer	Pa	Coefficient A in Eq. (14.100)	default = 0.00316
b_sfc_layer	1	Coefficient B in Eq. (14.100)	default = 0.99763
l_GSAM_compatible	logical	Use a parameter set used in the Japan Meteorological Agency's global atmosphere model GSM1705	default = .false.

Table14.16 Namelist nml_bulkecmwf_ocn_run)

variable name	units	description	usage
l_rst_bulkecmwf_ocn_in	logical	read initial restart for the friction velocity and the free convection velocity scale or not	default = l_rst_in

14.10 Specification of input data

The attributes of external forcing data should be specified by namelist nml_force_data. Table 14.17 explains variables that are contained in this namelist block.

Table14.17 Namelist nml_force_data related to surface forcing data.

variable name	units	description	usage
name		name of data object	
txyu		grid point on which data should be placed	
l2d	logical	data is 2D or not	.true. for surface data, default = .false.)
file_data		file that contains reference vales for surface restoring forcing	
file_data_grid		data file name of grid	needed if linterp = .true.
imfrc		grid number of data in x direction	
jmfrc		grid number of data in y direction	
num_elm		the number of elements contained in file	
interval	sec	regular time interval of data	−1 : monthly −999 : steady forcing
num_data_max		the number of record contained in the file	
ifstart	ifstart(6)	[ymdhms] of the first record of the input file	
lrepeat	logical	repeat use of climatological data or not	default = .false.
linterp	logical	interpolate horizontally or not	default = .false.
ilinear		interpolation method	1 : linear, 2 : spline
luniform	logical	data is horizontally uniform or not	default = .false.
loffset	logical	data is offset or not	default = .false.
offset		offsetting factor	
lfactor	logical	data is multiplied or not	default = .false.
factor		multiplicative factor	
ldouble	logical	input data is double precision or not	default = .false.
iverbose		standard output of progress	1 : extensive, 0 : minimum
ldefined	logical	data object with this name is defined or not	default = .true.

Format of data files that should be prepared is shown in the following. Note that one data file should contain only one element.

Format of surface forcing data (trcref(_surf)_conf%file_data)

```
integer(4), parameter :: num_data_max = 365*4, nu = 99
          ! "num_data_max" is  the number of data
integer(4) :: imfrc, jmfrc    ! data size
character(128) :: file_data, file_data_grid
real(4) :: precip(imfrc,jmfrc,num_data_max)
real(8) :: alonf(imfrc), alatf(jmfrc)
logical :: linterp

! main data

open (unit=nu,file=file_data,access=direct,recl=4*imfrc*jmfrc)
do n = 1, num_data_max
  write(unit=nu,rec=n) precip(:,:,n)
end do
close(nu)

! longitude/latitude of the main data

if (linterp) then  ! If input data is horizontally interpolated in the model.
  open (unit=nu,file=file_grid)
  write(nu) alonf, alatf
  close(nu)
end if
```

14.11 Surface data objects

External data either read from file or received from AGCM are stored in the data set object type type_surf_dbase (Table 14.18). This object contains a sub object meteor_alt of type type_surf_meteor_alt (Table 14.19) that stores the measurement height of marine meteorological variables. The name (ReadData for Reanalysis or FromCGCM for AGCM) of the object and the measurement height of marine meteorological variables must be specified by namelist nml_srfflx_dbase (Table 14.20).

Air-sea fluxes computed on the basis of external data are stored in the data set object type type_surf_product_ao (Table 14.21). This object contains sub objects givn_ao and calc_ao of the object type type_surf_bulk_ao (Table 14.22). The object calc_ao stores elements related to the computation of fluxes based on the bulk formula and the object givn_ao stores the corresponding elements received by other components, that is, atmospheric model. Similarly, air-ice fluxes are stored in type_surf_product_ai (Table 14.23), which contains sub objects givn_ai and calc_ai of the object type type_surf_bulk_ai (Table 14.24).

Table14.18 Contents of the object type_surf_dbase related to external surface forcing data.

variable name	units	description
name		name of data object
meteor_alt	type_surf_meteor_alt	Observation altitude of this data set
x_wind	cm sec^{-1} or dyn cm^{-2}	X-ward wind or wind stress
y_wind	cm sec^{-1} or dyn cm^{-2}	Y-ward wind or wind stress
temp_air	°C	surface air temperature
tdew_air	°C	dew point temperature
sphm_air	1	specific humidity
s_wind	cm sec^{-1}	scalar surface wind
slpress	hPa	sea level pressure
sw_flx	erg s^{-1} cm^{-2}	upward or downward shortwave, summed
sw_flx_visb	erg s^{-1} cm^{-2}	upward or downward shortwave, visible, beam
sw_flx_visd	erg s^{-1} cm^{-2}	upward or downward shortwave, visible, diffuse
sw_flx_nirb	erg s^{-1} cm^{-2}	upward or downward shortwave, near Infra-red, beam
sw_flx_nird	erg s^{-1} cm^{-2}	upward or downward shortwave, near Infra-red, diffuse
lw_flx	erg s^{-1} cm^{-2}	upward or downward longwave
sst	°C	sea surface temperature
precip	cm s^{-1}	precipitation water flux
river_w	cm s^{-1}	river discharge water flux
river_h	erg s^{-1} cm^{-2}	river discharge heat flux
snowfall	cm s^{-1}	snow fall water flux
iceberg_w	cm s^{-1}	iceberg discharge water flux
iceberg_h	erg s^{-1} cm^{-2}	iceberg discharge heat flux
frac_ice	1	sea ice fraction

Table14.19 Contents of the object type_surf_meteor_alt that stores measurement height of meteorological elements of external data.

variable name	units	description
alt_wind	m	altitude of wind
alt_temp	m	altitude of temperature
alt_sphm	m	altitude of specific humidity

Table14.20 Namelist nml_srfflx_dbase.

variable name	units	description	usage
name		name of the external dataset	ReadData (Reanalysis) or FromCGCM (AGCM) (required)

Continued on next page

Table 14.20 – continued from previous page

variable name	units	description	usage
l_marine_meteor	logical	read marine meteorological data or not	required
alt_wind_meter	m	observed height of wind	required if l_marine_meteor = .true.
alt_temp_meter	m	observed height of temperature	required if l_marine_meteor = .true.
alt_sphm_meter	m	observed height of specific humidity	required if l_marine_meteor = .true.

Table14.21 Contents of the object type_surf_product_ao that stores air-sea fluxes based on type_surf_dbase.

variable name	units	description
name		name of data object
l_calc_wind_stress	logical	calculate wind stress for this data set
l_calc_bulk_fluxes	logical	calculate fluxes based on a bulk formula for this data set
sw_nt_ao	erg s^{-1} cm^{-2}	net shortwave, summed
sw_nt_visb_ao	erg s^{-1} cm^{-2}	net shortwave, visible, beam
sw_nt_visd_ao	erg s^{-1} cm^{-2}	net shortwave, visible, diffuse
sw_nt_nirb_ao	erg s^{-1} cm^{-2}	net shortwave, near Infra-red, beam
sw_nt_nird_ao	erg s^{-1} cm^{-2}	net shortwave, near Infra-red, diffuse
lw_nt_ao	erg s^{-1} cm^{-2}	net longwave, air-ocean
tsfc_ocn	°C	sea surface (skin) temperature
wind_speed	cm sec^{-1}	scalar wind speed
givn_ao	type_surf_bulk_ao	given from other model or forcing data file
calc_ao	type_surf_bulk_ao	calculated using bulk formulae

Table14.22 Contents of the object `type_surf_bulk_ao` that stores elements related to computation of air-sea fluxes based on bulk formula.

variable name	units	description
`name`		name of data object
`x_rel_wind`	$\mathrm{m\,sec^{-1}}$	surface wind relative to surface current
`y_rel_wind`	$\mathrm{m\,sec^{-1}}$	surface wind relative to surface current
`x_stress`	$\mathrm{dyn\,cm^{-2}}$	X-ward surface stress
`y_stress`	$\mathrm{dyn\,cm^{-2}}$	Y-ward surface stress
`frcvl`	$\mathrm{m\,sec^{-1}}$	friction velocity
`rel_wind`	$\mathrm{m\,sec^{-1}}$	surface wind relative to surface current
`latent`	$\mathrm{erg\,s^{-1}\,cm^{-2}}$	latent heat flux, air-ocean
`sensible`	$\mathrm{erg\,s^{-1}\,cm^{-2}}$	sensible heat flux, air-ocean
`evapo`	$\mathrm{cm\,sec^{-1}}$	evaporative water flux, air-ocean

Table14.23 Contents of the object `type_surf_product_ai` that stores air-ice fluxes based on `type_surf_dbase`.

variable name	units	description
`name`		name of data object
`l_calc_wind_stress`	logical	calculate wind stress for this data set
`l_calc_bulk_fluxes`	logical	calculate fluxes based on a bulk formula for this data set
`sw_nt_ai`	$\mathrm{erg\,s^{-1}\,cm^{-2}}$	net shortwave, summed, absorbed at the ice surface
`sw_nt_in`	$\mathrm{erg\,s^{-1}\,cm^{-2}}$	net shortwave, summed, penetrate into the ice interior
`sw_nt_visb_ai`	$\mathrm{erg\,s^{-1}\,cm^{-2}}$	net shortwave, visible, beam
`sw_nt_visd_ai`	$\mathrm{erg\,s^{-1}\,cm^{-2}}$	net shortwave, visible, diffuse
`sw_nt_nirb_ai`	$\mathrm{erg\,s^{-1}\,cm^{-2}}$	net shortwave, near Infra-red, beam
`sw_nt_nird_ai`	$\mathrm{erg\,s^{-1}\,cm^{-2}}$	net shortwave, near Infra-red, diffuse
`lw_nt_ai`	$\mathrm{erg\,s^{-1}\,cm^{-2}}$	net longwave, air-ice
`tsfc_ice`	°C	ice surface temperature
`wind_speed`	$\mathrm{m\,sec^{-1}}$	scalar wind speed
`givn_ai`	`type_surf_bulk_ai`	given from other model or forcing data file
`calc_ai`	`type_surf_bulk_ai`	calculated using bulk formulae

Table14.24 Contents of the object `type_surf_bulk_ai` that stores elements related to computation of air-ice fluxes based on bulk formula.

variable name	units	description
`name`		name of data object
`x_rel_wind`	$\mathrm{m\,sec^{-1}}$	surface wind relative to ice drift
`y_rel_wind`	$\mathrm{m\,sec^{-1}}$	surface wind relative to ice drift
`x_stress`	$\mathrm{dyn\,cm^{-2}}$	X-ward surface stress
`y_stress`	$\mathrm{dyn\,cm^{-2}}$	Y-ward surface stress
`frcvl`	$\mathrm{m\,sec^{-1}}$	friction velocity
`rel_wind`	$\mathrm{m\,sec^{-1}}$	surface wind relative to ice drift
`latent`	$\mathrm{W\,m^{-2}}$	latent heat flux, air-ice
`sensible`	$\mathrm{W\,m^{-2}}$	sensible heat flux, air-ice
`sublim`	$\mathrm{m\,sec^{-1}}$ of sea water	sublimation water flux, air-ice
`ctntmp`	$\mathrm{m\,sec^{-1}}$	bulk transfer coefficient × scalar wind
`csntmp`	$\mathrm{m\,sec^{-1}}$	bulk transfer coefficient × scalar wind

14.12 Technical details

14.12.1 Recalculating and replacing fluxes in the coupled mode

The ocean model usually receives the surface fluxes calculated in the atmosphere model in the coupled mode (Chapter 21). In coupled mode (`SCUPCGCM` option) or in child mode of nesting (`SUB` option) in which fluxes are taken from the parent model (`NGETFLUX` option which is often used when the parent model is coupled with the atmosphere), this received fluxes may be replaced by the ones calculated on its own by using the surface atmospheric states taken from the parent component, its surface oceanic states, and its bulk formula. User may also specify the grid points where fluxes are replaced. This option is mainly intended to be applied to coastal region of a child model of nesting in which the fluxes received from the parent component may not fully reflect the fine scale surface state due to complex topography. Without an appropriate feedback mechanism at the sea surface, a child model may sometimes get unstable. This runtime option is controlled by namelist `nml_cgcm_recalc` and `nml_cgcm_recalc_replace`, which is explained in Tables 14.25 and 14.26.

Table14.25 Namelist `nml_cgcm_recalc`.

variable name	units	description	usage
`l_recalc_flux`	logical	Recalculating of fluxes	default = `.false.`

Table14.26 Namelist `nml_cgcm_recalc_replace`.

variable name	units	description	usage
`l_recalc_replace_all`	logical	Replacing fluxes for all grid points	default = `.false.`
`file_recalc_replace`	character	File that contains mask that specify grid points where fluxes are replaced.	real(8) 2D array (1/0) written with direct access

14.12.2 Properties of moist air

In MRI.COM, a set of formulae given by Gill (1982) is used to compute properties of moist air. Physical constants of sea water is listed on Table 2.1 of Chapter 2.

a. Saturation specific humidity

We consider computing saturation specific humidity in an environment with the sea surface temperature t [°C] and the pressure p [hPa].

The relation between the vapor pressure e [hPa] and specific humidity q is give by

$$e/p = q/(\epsilon + (1-\epsilon)q), \tag{14.120}$$

where ϵ is the molecular weight ratio between water vapor and air:

$$\epsilon = m_\mathrm{w}/m_\mathrm{a} = 18.016/28.966 = 0.62197. \tag{14.121}$$

This is solved for the specific humidity as

$$q = \epsilon e/(p - (1-\epsilon)e). \tag{14.122}$$

The relative humidity γ is the ratio of the mixing ratio of the mass of vapor to the mass of dry air to the saturated one. The mixing ratio (r) of the mass of vapor to the mass of dry air is given by

$$r = \frac{q}{1-q}. \tag{14.123}$$

Thus the relative humidity is computed using the specific humidity q and the saturation specific humidity q_s as

$$\gamma = \frac{q(1-q_s)}{q_s(1-q)}. \tag{14.124}$$

The saturation vapor pressure e_{sw} [hPa] of pure water vapor over a plane water surface is given by

$$\log_{10} e_{sw}(t) = (0.7859 + 0.03477\,t)/(1 + 0.00412\,t), \tag{14.125}$$

where t is temperature between ±40 °C.

In air, the partial pressure e'_{sw} of water vapor at saturation is not exactly e_{sw} but is given by

$$e'_{sw} = f_w e_{sw}. \tag{14.126}$$

The value of f_w is given by

$$f_w = 1 + 10^{-6} p(4.5 + 0.0006\,t^2), \tag{14.127}$$

where p is the pressure (in units [hPa]).

The saturation vapor pressure over a salt solution is less than over fresh water. For sea water, the reduction is about 2 % (a factor of 0.98 should be applied to e'_{sw}).

The saturation vapor pressure e_{si} of pure water vapor over ice is given by

$$\log_{10} e_{si}(t) = \log_{10} e_{sw}(t) + 0.00422\,t. \tag{14.128}$$

The saturation partial pressure e'_{si} in moist air is f_i times e_{si}. Values of f_i are given correct to 1 part in 10^4 by (14.127). Thus

$$e'_{si} = f_w e_{si}. \tag{14.129}$$

b. Specific heat of air

$$\begin{aligned} c_{pa} &= \frac{7}{2}R\left(1 - q + \frac{8q}{7\epsilon}\right) \\ &= 1004.6 \times (1 + 0.8735q)\,\mathrm{J\,kg^{-1}\,K^{-1}} \end{aligned} \tag{14.130}$$

c. Latent heat

The latent heat of vaporization is given by

$$L_v = 2.5008 \times 10^6 - 2.3 \times 10^3 t\,\mathrm{J\,kg^{-1}}. \tag{14.131}$$

The latent heat of sublimation is given by

$$L_s = 2.839 \times 10^6 - 3.6(t + 35)^2\,\mathrm{J\,kg^{-1}}. \tag{14.132}$$

d. Air density (ρ_a)

The equation of state of moist air of temperature T [K] under the pressure P [Pa] is given by

$$\rho_a = \frac{P}{RT(1 - q + q/\epsilon)} \equiv P/RT_v, \tag{14.133}$$

where

$$T_v \equiv T(1 - q + q/\epsilon) = T(1 + c_{vt}q) \tag{14.134}$$

is called the virtual temperature,

$$c_{vt} = \frac{1}{\epsilon} - 1 = \frac{m_\mathrm{a}}{m_\mathrm{w}} - 1 \sim 0.6078, \tag{14.135}$$

and

$$R = 287.04\,\mathrm{J\,kg^{-1}\,K^{-1}} \tag{14.136}$$

is the gas constant of dry air.

Chapter 15

Turbulence Closure Models

The surface boundary layer is made turbulent by wind injecting momentum, so vertical mixing may be induced even if the stratification is stable. The stratification near the surface may be made gravitationally unstable by the surface buoyancy loss, so the vertical convection may be induced. Though these phenomena occur in small scales, they are important for the large scale oceanic processes. However, these are neither expressed by the fundamental equation of the general circulation models*, nor resolved by large scale models. Therefore, general circulation models express the effects of small scale turbulence by making some closure assumptions called turbulence closure in which mixing is expressed by synoptic states (velocity and temperature (salinity) solution of the model).

There are mainly two approaches to the turbulence closure problem: statistical closure models and empirical approaches. MRI.COM provides several options of statistical closure models, which are explained in this chapter. Note that an empirical approach of K-profile parameterization proposed by Large et al. (1994) is most popular in the ocean-climate modeling community. However, the current version of MRI.COM does not take this approach. In a future version, an interface to the Community Vertical Mixing Project (CVMix; Griffies et al., 2015) will be implemented to test the empirical approaches.

Three statistical turbulence closure models are supported: So called Mellor-Yamada model (Mellor and Yamada 1982, Mellor and Blumberg 2004) presented in Section 15.2, Noh and Kim (1999) model presented in Section 15.3, and a two-equation turbulence closure model using generic length-scale (GLS) equation by Umlauf and Burchard (2003) presented in Section 15.4. Table 15.1 summarizes features of turbulence closure models.

Table15.1 Features and equation numbers of turbulence closure models explained in this chapter.

	Mellor and Yamada	Noh and Kim	Umlauf and Burchard
model option	`MELYAM`	`NOHKIM`	`GLS`
prognosticated variables	$q^2/2$ (15.42)	E(15.57)	E (15.83), Ψ (15.84)
surface boundary condition for TKE	specified (15.45)	flux (15.61)	flux (15.93)
formula for the length scale	(15.49), (15.48)	(15.75)	(15.78)
stability function	Solution of (15.39)	(15.72), (15.73)	(15.99), (15.100)

15.1 Statistical closure model

In the statistical closure models, the physical properties in the basic equations of motion for a Boussinesq fluid are separated into averaged components and perturbed components (Reynolds decomposition), and then the equations are time averaged. The expressions for averaged velocity $\boldsymbol{U}$, averaged pressure P, and averaged potential temperature Θ are

$$\frac{\partial U_i}{\partial x_i} = 0, \tag{15.1}$$

$$\frac{DU_j}{Dt} + \epsilon_{jkl} f_k U_l = \frac{\partial}{\partial x_k}(-\langle u_k u_j \rangle) - \frac{1}{\rho_0}\frac{\partial P}{\partial x_j} - g_j \frac{\rho}{\rho_0} + \frac{\partial}{\partial x_k}\left[2\nu \frac{1}{2}\left(\frac{\partial U_k}{\partial x_j} + \frac{\partial U_j}{\partial x_k}\right)\right], \tag{15.2}$$

$$\frac{D\Theta}{Dt} = \frac{\partial}{\partial x_k}(-\langle u_k \theta \rangle) + \frac{\partial}{\partial x_k}\left(\kappa \frac{\partial \Theta}{\partial x_k}\right), \tag{15.3}$$

where $D(\ )/Dt \equiv U_k \partial(\ )/\partial x_k + \partial(\ )/\partial t$, g_j is the gravity vector, f_k is the Coriolis vector, ϵ_{ijk} is the alternating tensor, ν is viscosity, and κ is diffusivity. Averaged quantities (resolved by the general circulation model) are represented by capital letters, and turbulent components (unresolved by the general circulation model) are represented by lower-case

[*] It could be expressed by relaxing the hydrostatic approximation but this is for future work.

letters. The statistical averages of the turbulent components are represented by $\langle\ \rangle$. The equation for salinity is similar to that for temperature (15.3). If density is calculated from temperature and salinity, the whole expression would become complicated. Therefore, density is assumed to be a function only of temperature here (we set $\rho = \beta\theta$, where β is the coefficient of thermal expansion). Though these equations basically correspond to the MRI.COM governing equations (2.16)-(2.20), they now have contributions from subgrid-scale processes ($\langle u_k u_j\rangle$ and $\langle u_k\theta\rangle$), which is the focus of this chapter.

The equations (15.1)-(15.3) show that the statistical averages of the turbulent components ($\langle u_k u_j\rangle$ and $\langle u_k\theta\rangle$), which are called as the *second order moments*, are necessary to calculate the evolution of large scale quantities. However, they are unknown and their evolution must be solved explicitly if we do not use a parameterization to presume them on the basis of the large scale quantities. Following Kantha and Clayson (2000), the equations for the second order moments are expressed as follows:

$$\begin{aligned}
&\frac{D\langle u_iu_j\rangle}{Dt} + \frac{\partial}{\partial x_k}\left[\langle u_ku_iu_j\rangle - \nu\frac{\partial\langle u_iu_j\rangle}{\partial x_k}\right] + \frac{1}{\rho_0}\left[\frac{\partial\langle pu_i\rangle}{\partial x_j} + \frac{\partial\langle pu_j\rangle}{\partial x_i}\right] + f_k(\epsilon_{jkl}\langle u_lu_i\rangle + \epsilon_{ikl}\langle u_lu_j\rangle)\\
&\quad = \left[-\langle u_ku_i\rangle\frac{\partial U_j}{\partial x_k} - \langle u_ku_j\rangle\frac{\partial U_i}{\partial x_k}\right] - \frac{\beta}{\rho_0}(g_j\langle u_i\theta\rangle + g_i\langle u_j\theta\rangle) + \left\langle\frac{p}{\rho_0}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right\rangle - 2\nu\left\langle\frac{\partial u_i}{\partial x_k}\frac{\partial u_j}{\partial x_k}\right\rangle,
\end{aligned}\tag{15.4}$$

$$\begin{aligned}
&\frac{D\langle u_j\theta\rangle}{Dt} + \frac{\partial}{\partial x_k}\left[\langle u_ku_j\theta\rangle - \kappa\left\langle u_j\frac{\partial\theta}{\partial x_k}\right\rangle - \nu\left\langle\theta\frac{\partial u_j}{\partial x_k}\right\rangle\right] + \frac{1}{\rho_0}\left\langle\theta\frac{\partial p}{\partial x_j}\right\rangle + \epsilon_{jkl}f_k\langle u_l\theta\rangle\\
&\quad = \left[-\langle u_ju_k\rangle\frac{\partial\Theta}{\partial x_k} - \langle u_k\theta\rangle\frac{\partial U_j}{\partial x_k}\right] - \frac{\beta}{\rho_0}g_j\langle\theta^2\rangle - (\kappa+\nu)\left\langle\frac{\partial u_j}{\partial x_k}\frac{\partial\theta}{\partial x_k}\right\rangle,
\end{aligned}\tag{15.5}$$

$$\frac{D\langle\theta^2\rangle}{Dt} + \frac{\partial}{\partial x_k}\left[\langle u_k\theta^2\rangle - \kappa\left\langle\frac{\partial\theta^2}{\partial x_k}\right\rangle\right] = -2\langle u_k\theta\rangle\frac{\partial\Theta}{\partial x_k} - 2\kappa\left\langle\frac{\partial\theta}{\partial x_k}\frac{\partial\theta}{\partial x_k}\right\rangle. \tag{15.6}$$

Thus, to solve the evolution of the second order moments, information about higher moments such as $\langle uuu\rangle$, $\langle uu\theta\rangle$, $\langle u\theta^2\rangle$ is necessary. The system of equations is not closed, and assumption must be made at some level to close the problem.

The expression closed in the second order of the turbulent components is named second moment closure and is frequently used in modeling turbulence. Many models have been proposed to parameterize the higher order correlation terms in (15.4) through (15.6) in order to close the problem at the second moment. Those models differ mainly in the manner how the pressure-strain correlators are parameterized (Burchard and Bolding, 2001). After applying independent parameterizations for the higher order terms, the majority of models adopt several similar simplifications such as algebraization and boundary layer approximation that result in the "stability functions" of their own. Stability functions are determined by local shear and stability and relate the vertical gradient of large scale fields with vertical turbulent fluxes. Procedures adopted by major statistical closure models are summarized and compared by Burchard and Bolding (2001).

15.2 Mellor and Yamada model

Mellor and Yamada (1982) model is a classic model as well as a milestone in the efforts of finding practical solutions for marine and atmospheric boundary layers. We show some details of the simplifying procedures made in this model as a typical example. This is also intended to be an introduction to the problem of turbulence closure. However, it should be noted that this is neither unique nor final solution.

15.2.1 Fundamental closure assumptions

Mellor and Yamada (1982) closed the system (15.1)-(15.6) by reducing the higher order terms as follows. Based on Rotta's (1951a,1951b) hypothesis of energy redistribution, the covariances of pressure and velocity gradients are assumed to be linear functions of Reynolds stress:

$$\left\langle\frac{p}{\rho_0}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right\rangle = -\frac{q}{3l_1}\left(\langle u_iu_j\rangle - \frac{\delta_{ij}}{3}q^2\right) + C_1q^2\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right), \tag{15.7}$$

where $q^2 \equiv \langle u_i^2\rangle$, l_1 is the length scale, C_1 is a non-dimensional constant, and δ_{ij} is Kronecker's delta, which is unity for $i = j$ and zero for $i \neq j$.

Using Kolmogorov's hypothesis of local isotropy in small eddies, the energy dissipation is modeled as follows:

$$2\nu\left\langle\frac{\partial u_i}{\partial x_k}\frac{\partial u_j}{\partial x_k}\right\rangle = -\frac{2}{3}\frac{q^3}{\Lambda_1}\delta_{ij}, \tag{15.8}$$

where Λ_1 is the length scale.

The redistribution of temperature and the dissipation of heat are modeled in the same form as above:

$$\Big\langle \frac{p}{\rho_0}\frac{\partial \theta}{\partial x_j}\Big\rangle = -\frac{q}{3l_2}\langle u_j\theta\rangle \tag{15.9}$$

$$(\kappa+\nu)\Big\langle \frac{\partial u_j}{\partial x_k}\frac{\partial \theta}{\partial x_k}\Big\rangle = 0, \tag{15.10}$$

where l_2 is the length scale. The dissipation of temperature variance is

$$2\kappa\Big\langle \frac{\partial \theta}{\partial x_k}\frac{\partial \theta}{\partial x_k}\Big\rangle = -2\frac{q}{\Lambda_2}\langle\theta^2\rangle, \tag{15.11}$$

where Λ_2 is the length scale.

In order to avoid the higher order problems, the turbulent velocity diffusion term and the other higher order terms are modeled as follows:

$$\langle u_k u_i u_j\rangle = \frac{3}{5}lqS_q\Big(\frac{\partial\langle u_i u_j\rangle}{\partial x_k}+\frac{\partial\langle u_i u_k\rangle}{\partial x_j}+\frac{\partial\langle u_j u_k\rangle}{\partial x_i}\Big), \tag{15.12}$$

$$\langle u_k u_j\theta\rangle = -\,lqS_{u\theta}\Big(\frac{\partial\langle u_k\theta\rangle}{\partial x_j}+\frac{\partial\langle u_j\theta\rangle}{\partial x_k}\Big), \tag{15.13}$$

$$\langle u_k\theta^2\rangle = -\,lqS_\theta\frac{\partial\langle\theta^2\rangle}{\partial x_k}, \tag{15.14}$$

where S_q, $S_{u\theta}$, and S_θ are non-dimensional numbers and can be set as constants or functions of certain parameters. Other relations are $\langle p\theta\rangle = 0$ and $\langle pu_i\rangle = 0$.

The essence of the Mellor-Yamada mixed layer model is that the above length scales are related linearly to each other:

$$(l_1,\Lambda_1,l_2,\Lambda_2) = (A_1,B_1,A_2,B_2)l, \tag{15.15}$$

where l is the vertical scale of turbulence (also called the master length scale), and A_1, B_1, A_2, B_2, and C_1 are empirical constants and are determined from experiment data. Mellor and Yamada (1982) employ $(A_1,B_1,A_2,B_2,C_1) = (0.92, 16.6, 0.74, 10.1, 0.08)$.

15.2.2 Algebraization, boundary layer approximation, and stability functions for the level 2.5 model

The turbulence model that solves the evolution of the statistically averaged values of the second-order turbulent components based on the simplification described in the previous subsection is called the level-4 model.

The level-3 model solves the evolution of the turbulent kinetic energy ($q^2/2$) and the variance of potential temperature ($\langle\theta^2\rangle$) (in some cases, the covariance of potential temperature and salinity ($\langle\theta s\rangle$) and the variance of salinity ($\langle s^2\rangle$)). The other statistically averaged values are solved diagnostically through algebraic equations assuming them to be in the steady state.

In the level-2.5 model, the variance of the potential temperature is also assumed to be in a statistically steady state (see expression (15.33) that appears later).

In the level-2 model, the turbulent kinetic energy is also assumed to be in a statistically steady state.

The level-2.5 model, which is employed as a surface boundary layer model by MRI.COM, is further simplified by applying the following boundary layer approximations.

- Neglect the Coriolis term in the equations of motion for the turbulent components.
- Neglect the molecular viscosity and diffusivity.
- Use the hydrostatic assumption in the vertical component of the equation of motion.
- Consider only vertical differentiation (direction perpendicular to the boundary) in the spatial differentiation for the term involving turbulent velocity.

The equations for the large scale quantities, (15.2) and (15.3), become

$$\frac{DU}{Dt} + \frac{\partial}{\partial z}\langle uw \rangle = -\frac{1}{\rho_0}\frac{\partial P}{\partial x} + fV, \tag{15.16}$$

$$\frac{DV}{Dt} + \frac{\partial}{\partial z}\langle vw \rangle = -\frac{1}{\rho_0}\frac{\partial P}{\partial y} - fU, \tag{15.17}$$

$$0 = -\frac{1}{\rho_0}\frac{\partial P}{\partial z} - g\frac{\rho}{\rho_0}, \tag{15.18}$$

$$\frac{D\Theta}{Dt} + \frac{\partial}{\partial z}(\langle w\theta \rangle) = \frac{1}{\rho_0 c_p}\frac{\partial R}{\partial z}. \tag{15.19}$$

In (15.19), R is the heat flux due to penetration of shortwave radiation. The level-2.5 system consists of the time evolution equation for turbulent kinetic energy and algebraic equations for other second-moment turbulent quantities.

The time evolution equation for the turbulent kinetic energy is

$$\frac{D}{Dt}\left(\frac{q^2}{2}\right) - \frac{\partial}{\partial z}\left[lqS_q\frac{\partial}{\partial z}\left(\frac{q^2}{2}\right)\right] = P_s + P_b - \epsilon, \tag{15.20}$$

where

$$P_s = -\langle wu \rangle\frac{\partial U}{\partial z} - \langle wv \rangle\frac{\partial V}{\partial z} \tag{15.21}$$

is the term for energy produced by the vertical shear of the averaged flow,

$$P_b = -g\langle w\rho \rangle/\rho_0 \tag{15.22}$$

is the term for energy produced by buoyancy, and

$$\epsilon = q^3/\Lambda_1 \tag{15.23}$$

is the energy dissipation term. The constant S_q is given in the next subsection.

The algebraic equations for the statistically averaged values, which are expressed by other second-moment turbulent quantities, are given below.

$$\langle u^2 \rangle = \frac{q^2}{3} + \frac{l_1}{q}\left[-4\langle wu \rangle\frac{\partial U}{\partial z} + 2\langle wv \rangle\frac{\partial V}{\partial z} - 2P_b\right], \tag{15.24}$$

$$\langle v^2 \rangle = \frac{q^2}{3} + \frac{l_1}{q}\left[2\langle wu \rangle\frac{\partial U}{\partial z} - 4\langle wv \rangle\frac{\partial V}{\partial z} - 2P_b\right], \tag{15.25}$$

$$\langle w^2 \rangle = \frac{q^2}{3} + \frac{l_1}{q}\left[2\langle wu \rangle\frac{\partial U}{\partial z} + 2\langle wv \rangle\frac{\partial V}{\partial z} + 4P_b\right], \tag{15.26}$$

$$\langle uv \rangle = \frac{3l_1}{q}\left[-\langle uw \rangle\frac{\partial V}{\partial z} - \langle vw \rangle\frac{\partial U}{\partial z}\right], \tag{15.27}$$

$$\langle wu \rangle = \frac{3l_1}{q}\left[-(\langle w^2 \rangle - C_1q^2)\frac{\partial U}{\partial z} - g\langle u\rho \rangle\right], \tag{15.28}$$

$$\langle vw \rangle = \frac{3l_1}{q}\left[-(\langle w^2 \rangle - C_1q^2)\frac{\partial V}{\partial z} - g\langle v\rho \rangle\right], \tag{15.29}$$

$$\langle u\theta \rangle = \frac{3l_2}{q}\left[-\langle uw \rangle\frac{\partial \Theta}{\partial z} - \langle w\theta \rangle\frac{\partial U}{\partial z}\right], \tag{15.30}$$

$$\langle v\theta \rangle = \frac{3l_2}{q}\left[-\langle vw \rangle\frac{\partial \Theta}{\partial z} - \langle w\theta \rangle\frac{\partial V}{\partial z}\right], \tag{15.31}$$

$$\langle w\theta \rangle = \frac{3l_2}{q}\left[-\langle w^2 \rangle\frac{\partial \Theta}{\partial z} - g\langle \theta\rho \rangle\right], \tag{15.32}$$

$$\langle \theta^2 \rangle = -\frac{\Lambda_2}{q}\langle w\theta \rangle\frac{\partial \Theta}{\partial z}. \tag{15.33}$$

Some of the terms in these equations, which are used in (15.16)-(15.19), can be further deformed by substitution as follows:

$$-\langle uw \rangle = K_M \frac{\partial U}{\partial z}, \tag{15.34}$$

$$-\langle vw \rangle = K_M \frac{\partial V}{\partial z}, \tag{15.35}$$

$$-\langle \theta w \rangle = K_H \frac{\partial \Theta}{\partial z}, \tag{15.36}$$

$$K_M = lqS_M, \tag{15.37}$$

$$K_H = lqS_H. \tag{15.38}$$

This simplification means that the vertical turbulent fluxes are proportional to the gradient of the large scale fields. The ultimate purpose of solving the mixed layer model is to determine the coefficients of momentum and heat fluxes, K_M and K_H, using (15.37) and (15.38).

Assuming that the potential density is linearly related to the potential temperature (and salinity), the simultaneous equations for S_M and S_H are derived as follows:

$$\begin{aligned} S_M[6A_1A_2g_M] + S_H[1 - 3A_2B_2g_H - 12A_1A_2g_H] &= A_2, \\ S_M[1 + 6A_1^2 g_M - 9A_1A_2g_H] - S_H[12A_1^2 g_H + 9A_1A_2g_H] &= A_1(1 - 3C_1), \end{aligned} \tag{15.39}$$

where

$$g_M \equiv \frac{l^2}{q^2}\left[\left(\frac{\partial U}{\partial z}\right)^2 + \left(\frac{\partial V}{\partial z}\right)^2\right] = \frac{l^2}{q^2}S^2, \tag{15.40}$$

$$g_H \equiv -\frac{l^2}{q^2}\frac{g}{\rho_0}\frac{\partial \tilde{\rho}}{\partial z} = \frac{l^2}{q^2}N^2, \tag{15.41}$$

and $\partial\tilde{\rho}/\partial z$ is the vertical gradient of potential density. Note that S_M and S_H are stability functions of this closure model.

Using S_M and S_H, K_M and K_H are then obtained from (15.37) and (15.38) by determining q and l.

15.2.3 Solving for velocity scale (q) and length scale (l)

The turbulent velocity q is obtained by solving the following expression that is modified from (15.20) using the above results:

$$\frac{\partial}{\partial t}\left(\frac{q^2}{2}\right) - \frac{\partial}{\partial z}\left[K_E \frac{\partial}{\partial z}\left(\frac{q^2}{2}\right)\right] = K_M\left[\left(\frac{\partial U}{\partial z}\right)^2 + \left(\frac{\partial V}{\partial z}\right)^2\right] + \frac{g}{\rho_0}K_H\frac{\partial \tilde{\rho}}{\partial z} - \epsilon, \tag{15.42}$$

where

$$K_E = lqS_q \tag{15.43}$$

and advection terms are neglected. In MRI.COM, S_q is set proportional to S_M ($S_q \propto S_M$). We adopt the form $S_q = S_{qc}S_M/S_{Mn}$, where $S_{qc} = 0.2$ and $S_{Mn} = 0.3927$. With this choice, $S_q = 0.2$ when the stratification is neutral ($g_H = 0$).

The sea surface boundary condition for the turbulent kinetic energy follows Mellor and Blumberg (2004):

$$K_q \frac{\partial q^2}{\partial z} = 2\alpha_{\mathrm{CB}} u_\tau^3, \tag{15.44}$$

where $\alpha_{\mathrm{CB}} = 100$ and u_τ is the frictional velocity defined as $u_\tau \equiv (\tau_s/\rho_s)^{1/2}$ by using the surface stress (τ_s) and the sea surface density (ρ_s). Mellor and Blumberg (2004) showed that this flux boundary condition is analytically converted to the condition for q at the sea surface:

$$q^2 = (15.8\alpha_{\mathrm{CB}})^{2/3}u_\tau^2. \tag{15.45}$$

On the other hand, the bottom boundary condition is

$$q = 0. \tag{15.46}$$

The vertical scale of the turbulence (master length scale, l) is estimated by many formulae such as a time evolution equation (which is usually empirical and is not completely based on physics) and a diagnosis. MRI.COM uses different diagnosis equations for the surface layer influenced directly by surface wind wave effects and for the internal region. The surface layer is defined by a depth of $|z| < z_\mathrm{w}$, where the roughness parameter z_w due to surface wind waves is given by Mellor and Blumberg (2004) as follows:

$$z_\mathrm{w} = \beta_\mathrm{w}\frac{u_\tau^2}{g}, \qquad \beta_\mathrm{w} = 2.0 \times 10^5. \tag{15.47}$$

In this layer, the formula for l is

$$l = \kappa z_\mathrm{w}, \tag{15.48}$$

where κ is the von Karman constant ($\kappa = 0.4$). In the internal region ($|z| > z_\mathrm{w}$), the formula for l is

$$l = \gamma \int_{z_b}^{0} |z'| q dz' \Big/ \int_{z_b}^{0} q dz', \tag{15.49}$$

where $\gamma = 0.2$, and z_b is the water depth. This is recognized as the averaged depth with the weight of the kinetic energy, which is sufficient for the ocean boundary layer according to Mellor and Yamada (1982).

15.2.4 Implementation

This section briefly describes the solution procedure.

The mixed layer model (subroutine name `mysl25` in `my25.F90`) is called as the last procedure of each time step that proceeds from n to n+1. After the master length scale (l) for the present time step (n) is determined using (15.48) and (15.49), where the latter is actually computed in the previous time step, the turbulent kinetic energy ($q^2/2$) is solved using (15.42) with (15.23) under the boundary conditions of (15.45) and (15.46), where the forward finite difference (n → n+1) is used in time. The implicit method is used for the vertical diffusion of the turbulent kinetic energy and energy dissipation term, since these terms could become significantly large (see Section 23.5). The vertical viscosity and diffusivity for the time step n+1 are estimated using q, l, and (15.37) to (15.39). These are used in the governing equations for the synoptic scale field. The vertical scale of the turbulence (master length scale) based on the turbulence kinetic energy (15.49) is calculated to prepare for the next time step.

The turbulent kinetic energy and the master length scale are defined at the bottom of the tracer cell (i, j, k). The specific expression for the discretized form of the turbulent kinetic energy ($E = q^2/2$) equation, (15.42), is as follows:

$$\begin{aligned}\frac{E_k^{n+1} - E_k^n}{\Delta t} = &\frac{1}{\Delta z_k}\left[\frac{K_{E\,k-\frac{1}{2}}(E_{k-1}^{n+1} - E_k^{n+1})}{\Delta z_{k-\frac{1}{2}}} - \frac{K_{E\,k+\frac{1}{2}}(E_k^{n+1} - E_{k+1}^{n+1})}{\Delta z_{k+\frac{1}{2}}}\right]\\ &+ K_{M\,k}\frac{(U_{k-\frac{1}{2}}^{n+1} - U_{k+\frac{1}{2}}^{n+1})(\tilde{U}_{k-\frac{1}{2}} - \tilde{U}_{k+\frac{1}{2}})}{\Delta z_k^2} + K_{M\,k}\frac{(V_{k-\frac{1}{2}}^{n+1} - V_{k+\frac{1}{2}}^{n+1})(\tilde{V}_{k-\frac{1}{2}} - \tilde{V}_{k+\frac{1}{2}})}{\Delta z_k^2}\\ &- K_{H\,k}\frac{B_{k-\frac{1}{2}}^{n+1} - B_{k+\frac{1}{2}}^{n+1}}{\Delta z_k} - 2E_k^{n+1}q_k^n/B_1 l_k^n,\end{aligned} \tag{15.50}$$

where $\tilde{U} = (U^{n+1} + U^n)/2$, $\tilde{V} = (V^{n+1} + V^n)/2$, and B is buoyancy ($= -\frac{g\rho}{\rho_0}$). The discrete expression for shear production (the second and third terms on the r.h.s.) and the buoyancy sink (the fourth term on the r.h.s.) follows Burchard (2002), which is consistent with the conservation law of the sum of mean and turbulent energy.

To summarize, the numerical operations proceed in the following order:

1. Calculate the master length scale for the present time step using (15.47) and (15.48) as well as (15.49) from the previous time step.
2. Update the turbulent kinetic energy using (15.50), (15.45) and (15.46).
3. Solve the algebraic equation for S_M and S_H using (15.39).
4. Calculate the vertical viscosity and diffusivity for the next time step using (15.37), (15.38), and (15.43).
5. Calculate the master length scale using (15.49) in preparation for the next time step.

15.2.5 Usage

Mellor and Yamada model is invoked by `MELYAM` option on compilation. There is no tuning parameter at run time. Initial state for vertical diffusion of turbulence kinetic energy (K_E), turbulence kinetic energy (q), and length scale (l) is read from restart file. Otherwise it is set as a state without turbulence. The state without turbulence is zero turbulence kinetic energy ($q = 0$), the mixing length of 1 meter ($l = 1$ m) and the vertical diffusion of turbulence kinetic energy is the constant background vertical viscosity given by visc_vert_bg_cm2ps in namelist nml_visc_vert_bg. How model is initialized is specified by namelist nml_melyam_run. Parameters are listed on Table 15.2.

Table15.2 namelist nml_melyam_run

variable name	units	description	usage
l_rst_melyam_in	logical	.true. : Read restart files specified by nmlrs_my_avq, nmlrs_my_q, and nmlrs_my_alo for the initial condition. .false.: Start condition is that of no turbulence. See text for details.	default = l_rst_in of nml_run_ini_state

15.3 Noh and Kim (1999) model

The mixed layer model proposed by Mellor and Yamada was originally developed for the atmospheric boundary layer, and its surface boundary is treated as a solid wall. When they applied this model to the ocean, they regarded that the turbulent kinetic energy is injected into the ocean by the wind stress at the solid-wall sea surface. The model by Mellor and Yamada could therefore be considered to insufficiently represent the oceanic turbulent mixed layer.

Noh and Kim (1999) presented a model that can resolve this insufficiency. The model is basically the same as Mellor and Yamada level 2.5 scheme in that turbulent kinetic energy is prognosticated and the length scale is diagnostically determined. It differs in the stability function and the treatment of surface boundary condition.

15.3.1 Fundamental equation

The equations for the zonal and meridional components of the velocity, U, V, buoyancy $B = -g\Delta\rho/\rho_0$, and turbulent energy E in the large scale fields are

$$\frac{DU}{Dt} = -\frac{\partial}{\partial z}\langle uw\rangle - \frac{1}{\rho_0}\frac{\partial P}{\partial x} + fV, \tag{15.51}$$

$$\frac{DV}{Dt} = -\frac{\partial}{\partial z}\langle vw\rangle - \frac{1}{\rho_0}\frac{\partial P}{\partial y} - fU, \tag{15.52}$$

$$\frac{DB}{Dt} = -\frac{\partial}{\partial z}\langle bw\rangle + \frac{\partial R}{\partial z}, \tag{15.53}$$

$$\frac{DE}{Dt} = -\frac{\partial}{\partial z}\left\langle w\left(\frac{p}{\rho_0} + uu + vv + ww\right)\right\rangle - \langle uw\rangle\frac{\partial U}{\partial z} - \langle vw\rangle\frac{\partial V}{\partial z} + \langle bw\rangle - \epsilon, \tag{15.54}$$

where R is the downward shortwave radiation and $\partial R/\partial z$ is its convergence.

The turbulent flux is expressed by using the large scale fields (in capital letters) as follows:

$$\frac{DU}{Dt} = \frac{\partial}{\partial z}\left(K_M\frac{\partial U}{\partial z}\right) - \frac{1}{\rho_0}\frac{\partial P}{\partial x} + fV, \tag{15.55}$$

$$\frac{DV}{Dt} = \frac{\partial}{\partial z}\left(K_M\frac{\partial V}{\partial z}\right) - \frac{1}{\rho_0}\frac{\partial P}{\partial y} - fU, \tag{15.56}$$

$$\frac{DB}{Dt} = \frac{\partial}{\partial z}\left(K_H\frac{\partial B}{\partial z}\right) - \frac{\partial R}{\partial z}, \tag{15.57}$$

$$\frac{DE}{Dt} = \frac{\partial}{\partial z}\left(K_E\frac{\partial E}{\partial z}\right) + K_M\frac{\partial U}{\partial z}\frac{\partial U}{\partial z} + K_M\frac{\partial V}{\partial z}\frac{\partial V}{\partial z} - \left(K_H\frac{\partial B}{\partial z}\right) - \epsilon. \tag{15.58}$$

The surface boundary conditions are as follows:

$$K_M\frac{\partial U}{\partial z} = \frac{\tau}{\rho_0}, \tag{15.59}$$

$$K_H\frac{\partial B}{\partial z} = Q_0, \tag{15.60}$$

$$K_E\frac{\partial E}{\partial z} = mu_\tau^3, \tag{15.61}$$

where m is a tuning parameter and $m = 100$ is recommended by Noh and Kim (1999). Unlike Mellor and Yamada model, a flux boundary condition is used for turbulence kinetic energy. This is intended to express the direct input of turbulence kinetic energy due to wind waves. As such, in the presence of sea ice, the direct input of wave energy may be more appropriate only through open water. The boundary condition may be optionally modified so that

$$K_E \frac{\partial E}{\partial z} = m u_\tau^3 (1 - a), \tag{15.62}$$

where a is area fraction of sea ice. The no-flux condition is used for the bottom:

$$K_E \frac{\partial E}{\partial z} = 0. \tag{15.63}$$

15.3.2 Stability function

The central problem is how to determine the viscosity, diffusivity (K_M, K_H, K_E), and turbulent energy dissipation rate (ϵ). In the Noh and Kim (1999) scheme, by using the typical velocity scale $(q = (2E)^{1/2})$ and the vertical length scale (l) of the turbulence, they are obtained by the following formula:

$$K_M = Sql, \tag{15.64}$$

$$K_H = S_B ql, \tag{15.65}$$

$$K_E = S_E ql, \tag{15.66}$$

$$\epsilon = Cq^3 l^{-1}. \tag{15.67}$$

The constants (S, S_B, S_E, C) are obtained from experiments. For neutral stratification, it is assumed that $S = 0.39$, $Pr \equiv S/S_B = 0.8$, $\sigma \equiv S/S_E = 1.95$, and $C = 0.06$. Hereafter, these values for neutral stratification are denoted as S_0, Pr_0, σ_0, and C_0, respectively.

Regarding the influence of the stratification, that is, stability function, we assume that the vertical scale of turbulence is limited by the vertical scale of buoyancy $l_b = q/N$ $(N^2 = \partial B/\partial z)$. That is,

$$K \sim q l_b \sim q l \mathrm{Ri}_t^{-1/2}, \tag{15.68}$$

where Ri_t is the turbulent Richardson number

$$\mathrm{Ri}_t = (Nl/q)^2. \tag{15.69}$$

Ri_t corresponds to g_H of Mellor and Yamada model. See (15.41).

This means that when the stratification is strong (N is large, Ri_t is large, and K is small), the turbulent energy is not transported downwards. It could also be considered that the local turbulent energy dissipation becomes large.

The following equation is used for S so that it satisfies (15.68) when Ri_t is large:

$$S/S_0 = (1 + \alpha \mathrm{Ri}_t)^{-1/2}, \tag{15.70}$$

where α is a tuning parameter. Noh and Kim (1999) recommend $\alpha \sim 120.0$, but $\alpha \sim 5.0$ is the default value of MRI.COM.

The effect of stratification on the Prandtl number (Pr) is set following Noh et al. (2005):

$$Pr/Pr_0 = (1 + \beta \mathrm{Ri}_t)^{1/2}, \tag{15.71}$$

where β is a tuning parameter and 0.5 is used following Noh et al. (2005).

To summarize, the stability function for this model is expressed as follows:

$$S = \frac{S_0}{(1 + \alpha \mathrm{Ri}_t)^{1/2}} \tag{15.72}$$

$$S_B = \frac{S}{Pr} = \frac{S_0}{Pr_0 (1 + \alpha \mathrm{Ri}_t)^{1/2} (1 + \beta \mathrm{Ri}_t)^{1/2}}. \tag{15.73}$$

The effect of stratification on the energy dissipation (C) is set as follows:

$$C/C_0 = (1 + \alpha \mathrm{Ri}_t)^{1/2}. \tag{15.74}$$

Note that this is independent of stratification for Mellor and Yamada and Umlauf and Burchard (2003) model.

In the case of unstable stratification ($N^2 < 0$), $K_M = K_H = 1.0\,\mathrm{m^2\,s^{-1}}$ and K_E is estimated from the turbulent velocity scale and the vertical length scale in the model. This treatment is due to the difference between the time scales of the vertical convection and the development of turbulence.

Finally, the vertical scale of turbulence is diagnosed as

$$l = \frac{\kappa(|z| + z_0)}{(1 + \kappa(|z| + z_0)/h)}, \tag{15.75}$$

where z_0 is the sea surface roughness ($z_0 = 1$ m), z is the depth, and h is the mixed layer depth. The mixed layer depth (h) is determined as the depth at which the buoyancy frequency becomes the maximum in the vertical column. The vertical scale becomes longer as the mixed layer becomes deeper.

15.3.3 Implementation

Equation (15.58) is solved for the prognostic variable E in the subroutine `nkoblm` in `nkoblm.F90` as the last procedure of each time step. The forward finite difference is used in the time evolution. The implicit method is used for the vertical diffusion of the turbulent kinetic energy and energy dissipation term, since these terms could become significantly large (see Section 23.5). The new E is used to determine the coefficients of viscosity and diffusivity for the next time step.

The turbulent kinetic energy and the master length scale are defined at the center of the tracer cell $(i, j, k - \frac{1}{2})$. The specific expression for the discretized form of the turbulent kinetic energy (E) equation is as follows:

$$\begin{aligned}
\frac{E^{n+1}_{k-\frac{1}{2}} - E^{n}_{k-\frac{1}{2}}}{\Delta t} =& \frac{1}{\Delta z_{k-\frac{1}{2}}} \left[\frac{K_{E\,k-1}(E^{n+1}_{k-\frac{3}{2}} - E^{n+1}_{k-\frac{1}{2}})}{\Delta z_{k-1}} - \frac{K_{E\,k}(E^{n+1}_{k-\frac{1}{2}} - E^{n+1}_{k+\frac{1}{2}})}{\Delta z_{k}} \right] \\
&+ \frac{1}{2} K_{M\,k-1} \frac{(u^{n+1}_{k-\frac{3}{2}} - u^{n+1}_{k-\frac{1}{2}})(u^{n+1}_{k-\frac{3}{2}} - u^{n}_{k-\frac{1}{2}})}{\Delta z_{k-1} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{M\,k} \frac{(u^{n+1}_{k-\frac{1}{2}} - u^{n+1}_{k+\frac{1}{2}})(u^{n}_{k-\frac{1}{2}} - u^{n+1}_{k+\frac{1}{2}})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} \\
&+ \frac{1}{2} K_{M\,k-1} \frac{(v^{n+1}_{k-\frac{3}{2}} - v^{n+1}_{k-\frac{1}{2}})(v^{n+1}_{k-\frac{3}{2}} - v^{n}_{k-\frac{1}{2}})}{\Delta z_{k-1} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{M\,k} \frac{(v^{n+1}_{k-\frac{1}{2}} - v^{n+1}_{k+\frac{1}{2}})(v^{n}_{k-\frac{1}{2}} - v^{n+1}_{k+\frac{1}{2}})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} \\
&- \frac{1}{2} K_{H\,k-1} \frac{B^{n+1}_{k-\frac{3}{2}} - B^{n+1}_{k-\frac{1}{2}}}{\Delta z_{k-1}} - \frac{1}{2} K_{H\,k} \frac{B^{n+1}_{k-\frac{1}{2}} - B^{n+1}_{k+\frac{1}{2}}}{\Delta z_{k}} - 2CE^{n+1}_{k-\frac{1}{2}} q^{n}_{k-\frac{1}{2}} / l^{n}_{k-\frac{1}{2}}.
\end{aligned} \tag{15.76}$$

The discrete expression for the shear production (the second through fifth terms on the r.h.s.) and the buoyancy sink (the sixth term on the r.h.s.) follows Burchard (2002), which is consistent with the conservation law of the sum of mean and turbulent energy.

To summarize, the numerical operations proceed in the following order:

1. Update the master length scale using (15.75),
2. Update the turbulent kinetic energy using (15.76) under the boundary conditions (15.61) and (15.63)
3. Calculate the vertical viscosity and diffusivity for the next time step (15.64) and (15.65).

15.3.4 Usage

Noh and Kim (1999) model is invoked by `NOHKIM` option on compilation.

Initial state for vertical diffusion of turbulence kinetic energy (K_E), turbulence kinetic energy (E) is either read from restart file or set as a state without turbulence. The state without turbulence means that E is very small as $\frac{1}{2}{q_{\mathrm{min}}}^2$, where $q_{\mathrm{min}} = 1 \times 10^{-5}\,\mathrm{m\,s^{-1}}$ is the minimum of turbulence velocity scale, and that K_E is set the constant background vertical viscosity given by `visc_vert_bg_cm2ps` in namelist `nml_visc_vert_bg`. How model is initialized is specified by namelist `nml_nohkim_run`. Parameters are listed on Table 15.4.

Tuning parameters may be specified at run time by namelist `nml_nkoblm`. Parameters are listed on Table 15.3.

Table15.3 namelist `nml_nkoblm`

variable name	units	description	usage
`mfnk`	1	m of Eq. (15.61)	default = 100.0
`alpnk`	1	α of Eq. (15.70)	default = 5.0
`betank`	1	β of Eq. (15.71)	default = 0.5
`l_wave_damp_seaice`	logical	Eq. (15.62) is used instead of (15.61)	default = `.false.`

Table15.4 namelist `nml_nohkim_run`

variable name	units	description	usage
`l_rst_nohkim_in`	logical	`.true.` : Read restart files specified by `nmlrs_nk_avq` and `nmlrs_nk_eb` for the initial condition. `.false.`: Start condition is that of no turbulence. See text for details.	default = `l_rst_in` of `nml_run_ini_state`

15.4 Generic length scale model by Umlauf and Burchard (2003)†

The turbulent closure models of Mellor and Yamada level 2.5 (section 15.2) and Noh and Kim (section 15.3) as incorporated in MRI.COM have only one prognostic variable of turbulence (turbulent kinetic energy). These may be classed as "one-equation" turbulent closure models. In these models, some formulae for length scale are given (e.g., (15.49) and (15.75)). Therefore the turbulent length scale l is not a predicted but a diagnosed quantity.

Generic length scale model by Umlauf and Burchard (2003) is a generalized form of two-equation turbulence models. It introduces a generic length-scale variable ψ as the second variable, expressed as

$$\psi = (c_\mu^0)^p E^m l^n. \tag{15.77}$$

With E and ψ given, the turbulent length scale can be computed as

$$l = (c_\mu^0)^{-\frac{p}{n}} E^{-\frac{m}{n}} \psi^{\frac{1}{n}}, \tag{15.78}$$

and the dissipation rate ϵ can be computed from the following relation,

$$\epsilon = (c_\mu^0)^3 \frac{E^{3/2}}{l}. \tag{15.79}$$

With appropriate values of the exponents p, m, n in (15.77), a number of well-known two-equation models (e.g., k-ϵ model and k-ω model) can be identified as special case of the generic model.

15.4.1 Fundamental equation

The fundamental equations for U, V, B and E are basically common to those of Noh and Kim model. The equation for generic length scale variable ψ is added. Expressing the turbulent flux using the large scale fields with the assumption of horizontally homogeneous flow, the fundamental equations of the model is expressed as follows:

$$\frac{DU}{Dt} = \frac{\partial}{\partial z}\left(K_M \frac{\partial U}{\partial z}\right) - \frac{1}{\rho_0}\frac{\partial P}{\partial x} + fV, \tag{15.80}$$

$$\frac{DV}{Dt} = \frac{\partial}{\partial z}\left(K_M \frac{\partial V}{\partial z}\right) - \frac{1}{\rho_0}\frac{\partial P}{\partial y} - fU, \tag{15.81}$$

$$\frac{DB}{Dt} = \frac{\partial}{\partial z}\left(K_H \frac{\partial B}{\partial z}\right) - \frac{\partial R}{\partial z}, \tag{15.82}$$

$$\frac{DE}{Dt} = \frac{\partial}{\partial z}\left(K_E \frac{\partial E}{\partial z}\right) + P_\mathrm{s} + P_\mathrm{b} - \epsilon, \tag{15.83}$$

$$\frac{D\psi}{Dt} = \frac{\partial}{\partial z}\left(K_\psi \frac{\partial \psi}{\partial z}\right) + \frac{\psi}{E}(c_{\psi 1} P_\mathrm{s} + c_{\psi 3} P_\mathrm{b} - c_{\psi 2}\epsilon), \tag{15.84}$$

† This model is implemented by Shiro Nishikawa as he was a visiting researcher at MRI.

where $P_{\rm s}$ and $P_{\rm b}$ are the shear production and the buoyancy production expressed according to

$$P_{\rm s} = -\langle uw \rangle \frac{\partial u}{\partial z} - \langle vw \rangle \frac{\partial v}{\partial z} = K_M M^2, \qquad P_{\rm b} = \langle bw \rangle = -K_H N^2, \tag{15.85}$$

and

$$M^2 = \left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2, \qquad N^2 = \frac{\partial B}{\partial z} \tag{15.86}$$

are the shear frequency and the buoyancy frequency, respectively. The vertical turbulent diffusivities in horizontally homogeneous flows are expressed as

$$K_M = c_\mu E^{1/2} l, \tag{15.87}$$

$$K_H = c'_\mu E^{1/2} l, \tag{15.88}$$

$$K_E = c_\mu E^{1/2} l / \sigma_k^\psi, \tag{15.89}$$

$$K_\psi = c_\mu E^{1/2} l / \sigma_\psi, \tag{15.90}$$

where c_μ and c'_μ are stability functions and σ_k^ψ and σ_ψ are the constant Schmidt numbers. Here, c_μ and c'_μ corresponds to S_M, and S_H of the Mellor and Yamada model with a relation

$$c_\mu = \sqrt{2} S_M, \quad c'_\mu = \sqrt{2} S_H. \tag{15.91}$$

In the case of unstable stratification ($N^2 < 0$), $K_M = K_H = 1.0\,{\rm m^2\,s^{-1}}$ is used in MRI.COM. This treatment is common to Noh and Kim model (section 15.3).

15.4.2 Boundary conditions

At the surface, the shear-free boundary layers with injection of turbulent kinetic energy are assumed, in which

$$E = K(-z + z_0)^\alpha, \quad l = L(-z + z_0), \tag{15.92}$$

are supposed, where K is the scale of turbulent kinetic energy, L and α are empirical constants given in Table 15.5, and z_0 is the sea surface roughness ($z_0 = 1$ m). K is determined by the injection of turbulent kinetic energy which is given by the surface flux. We use flux forms for the surface boundary conditions ($z = 0$) of E and ψ as follows:

$$F_E = K_E \frac{\partial E}{\partial z} = \eta u_\tau^3, \tag{15.93}$$

$$F_\psi = K_\psi \frac{\partial \psi}{\partial z} = -\frac{c_\mu (c_\mu^0)^p}{\sigma_\psi} (m\alpha + n) K^{m+\frac{1}{2}} L^{n+1} (z_0)^{(m+\frac{1}{2})\alpha + n}. \tag{15.94}$$

As explained in the previous section for Noh and Kim (1999) model, injection of turbulence kinetic energy may be made only through the open water in the presence of sea ice.

$$F_E = K_E \frac{\partial E}{\partial z} = \eta u_\tau^3 (1 - a), \tag{15.95}$$

where a is sea ice area fraction.

The parameter K can be determined from

$$K = \left(-\frac{\sigma_k^\psi}{c_\mu \alpha L} F_E\right)^{\frac{2}{3}} \frac{1}{z_0^\alpha}, \tag{15.96}$$

where (15.89), (15.92) and (15.93) are used.

At the bottom, the logarithmic boundary layers are assumed, in which $l = \kappa(z + H + z_{0\rm b})$ is supposed. The bottom boundary conditions ($z = -H$) for E and ψ are given by flux forms as follows:

$$F_E = K_E \frac{\partial E}{\partial z} = 0, \tag{15.97}$$

$$F_\psi = K_\psi \frac{\partial \psi}{\partial z} = \frac{n (c_\mu^0)^{p+1} \kappa^{n+1}}{\sigma_\psi} E^{m+\frac{1}{2}} (z_{0\rm b})^n, \tag{15.98}$$

where $z_{0\rm b}$ is the bottom roughness, which is set to be $z_{0\rm b} = 1$ m. Implementation of the bottom boundary condition is realistic as compared to the Mellor and Yamada scheme and the Noh and Kim scheme.

15.4.3 Model parameters

Umlauf and Burchard (2003) investigated the properties of the generic model in some fundamental flows and derived the constraints on the model parameters (c^0_μ, $c_{\psi 1}$, $c_{\psi 2}$, $c_{\psi 3}$, σ^ψ_k, σ_ψ, m and n). Table 15.5 is a recommended parameter set for the generic model by Umlauf and Burchard (2003), based on their calibration. MRI.COM uses this as the default parameters. Note that the parameter p (the factor $(c^0_\mu)^p$ in (15.77)) is mathematically irrelevant, since (15.84) can be multiplied by any constant without changing the solution ($p = 2.0$ is used as the default value).

Table15.5 A recommended parameter set for the generic model by Umlauf and Burchard (2003), where $\kappa = 0.4$, $(c^0_\mu)^2 = 0.3$, and $c_{\psi 1} = m$ are assumed. The value of the parameter $c^-_{\psi 3}$ is based on the Algebraic Reynolds-Stress Model (ASM) of Canuto et al. (2001). In unstable situations ($P_\mathrm{b} > 0$), a different value of the parameter $c_{\psi 3}$ needs to be used. In MRI.COM, $c^+_{\psi 3} = 1.0$ is adopted.

α	L	m	n	σ^ψ_k	σ_ψ	$c_{\psi 1}$	$c_{\psi 2}$	$c^-_{\psi 3}$
−2.0	0.20	1.00	−0.67	0.80	1.07	1.00	1.22	0.05

15.4.4 Stability functions

The version A of the Canuto et al. (2001) stability functions is expressed as follows:

$$\hat{c}_\mu = c_\mu (c^0_\mu)^3 = \frac{0.1070 + 0.01741\alpha_N - 0.00012\alpha_M}{1 + 0.2555\alpha_N + 0.02872\alpha_M + 0.008677\alpha_N^2 + 0.005222\alpha_N\alpha_M - 0.0000337\alpha_M^2}, \tag{15.99}$$

$$\hat{c}'_\mu = c'_\mu (c^0_\mu)^3 = \frac{0.1120 + 0.004519\alpha_N + 0.00088\alpha_M}{1 + 0.2555\alpha_N + 0.02872\alpha_M + 0.008677\alpha_N^2 + 0.005222\alpha_N\alpha_M - 0.0000337\alpha_M^2}, \tag{15.100}$$

where

$$\alpha_M = \frac{E^2}{\epsilon^2} M^2, \qquad \alpha_N = \frac{E^2}{\epsilon^2} N^2 \tag{15.101}$$

are nondimensional parameters referred to as the shear number and the buoyancy number. Using (15.79), they are related to the nondimensional parameter used by the Mellor and Yamada model (15.40) and (15.41) as

$$g_M = \frac{(c^0_\mu)^6}{2}\alpha_M, \quad g_H = \frac{(c^0_\mu)^6}{2}\alpha_N. \tag{15.102}$$

These are presented in Burchard and Bolding (2001)[‡]. MRI.COM adopts these as the stability functions for the Umlauf and Burchard (2003) model.

15.4.5 Implementation

Equations (15.83) and (15.84) are solved for the prognostic variable E and ψ in the subroutine `gls_main` in `gls.F90` as the last procedure of each time step. The forward finite difference is used in the time evolution. The implicit method is used for the vertical diffusion of the turbulent kinetic energy or generic variable and energy dissipation term, since these terms could become significantly large (see Section 23.5). The new E and ψ are used to determine the coefficients of viscosity and diffusivity for the next time step.

The turbulent kinetic energy and the master length scale are defined at the center of the tracer cell $(i, j, k - \frac{1}{2})$. The specific expression for the discretized form of the turbulent kinetic energy (E) equation is as follows:

$$\begin{aligned} \frac{E^{n+1}_{k-\frac12} - E^n_{k-\frac12}}{\Delta t} = & \frac{1}{\Delta z_{k-\frac12}} \left[\frac{K_{E\,k-1}(E^{n+1}_{k-\frac32} - E^{n+1}_{k-\frac12})}{\Delta z_{k-1}} - \frac{K_{E\,k}(E^{n+1}_{k-\frac12} - E^{n+1}_{k+\frac12})}{\Delta z_k} \right] \\ & + P_\mathrm{s} + P_\mathrm{b} - (c^0_\mu)^3 E^{n+1}_{k-\frac12} (E^n_{k-\frac12})^{\frac12} / l^n_{k-\frac12}. \end{aligned} \tag{15.103}$$

[‡] Note that definition of c_μ and c'_μ in Burchard and Bolding (2001) is different than that of this chapter. $\hat{c}_\mu$ and $\hat{c}'_\mu$ here correspond to c_μ and c'_μ of Burchard and Bolding (2001).

The specific expression for the discretized form of the generic variable (ψ) equation is as follows:

$$\begin{aligned}\frac{\psi^{n+1}_{k-\frac{1}{2}} - \psi^{n}_{k-\frac{1}{2}}}{\Delta t} =& \frac{1}{\Delta z_{k-\frac{1}{2}}}\left[\frac{K_{\psi\,k-1}(\psi^{n+1}_{k-\frac{3}{2}} - \psi^{n+1}_{k-\frac{1}{2}})}{\Delta z_{k-1}} - \frac{K_{\psi\,k}(\psi^{n+1}_{k-\frac{1}{2}} - \psi^{n+1}_{k+\frac{1}{2}})}{\Delta z_{k}}\right] \\ &+ \frac{\psi^{n}_{k-\frac{1}{2}}}{E^{n}_{k-\frac{1}{2}}}(c_{\psi 1}P_{\mathrm{s}} + c_{\psi 3}P_{\mathrm{b}}) - c_{\psi 2}(c^{0}_{\mu})^{3}\psi^{n+1}_{k-\frac{1}{2}}(E^{n}_{k-\frac{1}{2}})^{\frac{1}{2}}/l^{n}_{k-\frac{1}{2}}.\end{aligned} \tag{15.104}$$

The shear and buoyancy production terms P_{s} and P_{b} are discretized as follows:

$$\begin{aligned}P_{\mathrm{s}} =& \frac{1}{2}K_{M\,k-1}\frac{(u^{n+1}_{k-\frac{3}{2}} - u^{n+1}_{k-\frac{1}{2}})(u^{n+1}_{k-\frac{3}{2}} - u^{n}_{k-\frac{1}{2}})}{\Delta z_{k-1}\Delta z_{k-\frac{1}{2}}} + \frac{1}{2}K_{M\,k}\frac{(u^{n+1}_{k-\frac{1}{2}} - u^{n+1}_{k+\frac{1}{2}})(u^{n}_{k-\frac{1}{2}} - u^{n+1}_{k+\frac{1}{2}})}{\Delta z_{k}\Delta z_{k-\frac{1}{2}}} \\ &+ \frac{1}{2}K_{M\,k-1}\frac{(v^{n+1}_{k-\frac{3}{2}} - v^{n+1}_{k-\frac{1}{2}})(v^{n+1}_{k-\frac{3}{2}} - v^{n}_{k-\frac{1}{2}})}{\Delta z_{k-1}\Delta z_{k-\frac{1}{2}}} + \frac{1}{2}K_{M\,k}\frac{(v^{n+1}_{k-\frac{1}{2}} - v^{n+1}_{k+\frac{1}{2}})(v^{n}_{k-\frac{1}{2}} - v^{n+1}_{k+\frac{1}{2}})}{\Delta z_{k}\Delta z_{k-\frac{1}{2}}},\end{aligned} \tag{15.105}$$

$$P_{\mathrm{b}} = -\frac{1}{2}K_{H\,k-1}\frac{B^{n+1}_{k-\frac{3}{2}} - B^{n+1}_{k-\frac{1}{2}}}{\Delta z_{k-1}} - \frac{1}{2}K_{H\,k}\frac{B^{n+1}_{k-\frac{1}{2}} - B^{n+1}_{k+\frac{1}{2}}}{\Delta z_{k}}. \tag{15.106}$$

The discrete expression for the shear production and the buoyancy production follows Burchard (2002), which is consistent with the conservation law of the sum of mean and turbulent energy.

To summarize, the numerical operations proceed in the following order:

1. Update the turbulent kinetic energy and generic variable using (15.103), (15.104), (15.105) and (15.106) under the boundary conditions, (15.93), (15.94), (15.97) and (15.98).
2. Diagnose the turbulent length scale using (15.78),
3. Calculate the vertical viscosity and diffusivity using (15.88) and (15.88) for the next time step.

15.4.6 Limiters

Some kind of limiters for the turbulent variables may be needed to ensure numerical stability. MRI.COM supposes the following conditions.

- Lower limiters of the turbulent kinetic energy and generic variable, E_{min}, ψ_{min}, are introduced as basic limiters.
- If one of E and ψ becomes less than the limiter, turbulence is considered as stagnant. That is,

$$E = E_{\mathrm{min}}, \quad \psi = \psi_{\mathrm{min}}, \quad l = l_{\mathrm{stag}}, \qquad \text{if either} \;\; E \le E_{\mathrm{min}} \;\; \text{or} \;\; \psi \le \psi_{\mathrm{min}} \tag{15.107}$$

- When turbulence is stagnant, the diagnosed viscosity and diffusivity (K_M, K_H) should not exceed the background values ($K_{M\,\mathrm{bg}}$, $K_{H\,\mathrm{bg}} = 1.0 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$):

$$K_M = c_{\mu}(2E_{\mathrm{min}})^{1/2}l_{\mathrm{stag}} \le K_{M\,\mathrm{bg}}, \qquad K_H = c'_{\mu}(2E_{\mathrm{min}})^{1/2}l_{\mathrm{stag}} \le K_{H\,\mathrm{bg}} \tag{15.108}$$

- Stagnant value of the turbulent length-scale (l_{stag}) is supposed to be the local vertical grid size ($\Delta z_{\mathrm{local}}$) for numerical stability.

Table15.6 Default limiters for GLS model in MRI.COM.

E_{min}	ψ_{min}	l_{stag}
$\frac{1}{2}\left(\frac{10^{-6}}{0.1 l_{\mathrm{stag}}}\right)^2$	$(c^{0}_{\mu})^2 \frac{E_{\mathrm{min}}}{l^{2/3}_{\mathrm{stag}}}$	$\Delta z_{\mathrm{local}}$

15.4.7 Usage

Umlauf and Burchard (2003) model is invoked by GLS option on compilation.

Initial state for vertical diffusion of turbulence kinetic energy (K_E) and generic variable K_{Ψ}, turbulence kinetic energy (E), generic variable (Ψ), and length scale (l) is either read from restart file or set as the stagnant state defined in the previous subsection. In the stagnant state, both of K_E and K_Ψ are set as $1 \times 10^{-6}\,\mathrm{m}^2\,\mathrm{s}$. The initial state is specified by namelist `nml_gls_run`. Parameters are listed on Table 15.7.

Tuning parameters may be specified at run time by namelist `nml_gls`. Parameters are listed on Table 15.8.

Table15.7 namelist `nml_gls_run`

variable name	units	description	usage
`l_rst_gls_in`	logical	`.true.` : Read restart files specified by `nmlrs_gls_(avk, avq, eke, psi, alo)` for the initial condition. `.false.`: Start condition is that of no turbulence. See text for details.	default = `l_rst_in` of `nml_run_ini_state`

Table15.8 namelist `nml_gls`

variable name	units	description	usage
`mfnk`	1	η of Eq. (15.93)	default = 100.0
`l_wave_damp_seaice`	logical	Eq. (15.93) is used instead of (15.95)	default = `.false.`

Chapter 16

Bottom Boundary Layer (BBL)

MRI.COM has an option of adopting a simple bottom boundary layer model (`BBL` option). This chapter describes the formulation and usage of this model.

16.1 General description

In general, neither dense overflows from the Nordic-Scotland-Greenland ridges to the Atlantic Ocean nor the dense downslope flows from the continental slope around Antarctica can be properly reproduced by z-coordinate models. The former becomes the core water of the North Atlantic Deep Water, and the latter becomes that of the Antarctic Bottom Water. As a result, the abyssal waters in the modeled ocean tend to exhibit a warming bias. To mitigate these deficiencies, MRI.COM incorporates a simple bottom boundary layer (BBL) model as introduced by Nakano and Suginohara (2002). In this model, BBL is attached immediately below the bottom of the interior oceanic domain and conceptually treated as a plane along the sea floor at the model's lowest vertical level (Figure 16.1). The following processes relevant to the overflow/downslope flows are (partly) incorporated into this BBL model:

- The advection along the bottom topography.
- The pressure gradient terms when the dense water lies on slope.
- The eddy activity to create the flow crossing f/h contours.

16.2 Grid arrangement and exchanges

In z-coordinate models, the flow along the bottom topography is expressed as a sequence of horizontal and vertical movements along a staircase-like topography. When the number of vertical grid points representing the bottom topography is roughly the same as that of the horizontal grid points, the downslope flow can be reasonably represented even in z-coordinate models (Winton et al., 1998). In general, however, the number of the vertical grids used in ocean general circulation models is not large enough. Even if the number is large enough, the concentrations of tracers might be significantly diffused during their movement owing to the horizontal mixing with the surrounding waters. The incorporation of BBL remedies these problems.

The BBL grid cells in MRI.COM are arranged as in Figure 16.1. General interior grid cells in MRI.COM are composed of U-cells, but it is intuitively easier to understand the grid arrangement of BBL as if they are located at T-cells (represented by "•" points of Figure 16.1a and b). The T-cells in BBL are located just below the bottom of interior T-cells even if an interior T-cell consists of only one quarter of a U-cell (Figure 16.1b). The U-cells in BBL are regarded as slanting cells connecting neighboring T-points in BBL (Figure 16.1d).

For the purpose of calculating horizontal exchanges between BBL cells due to advection and subgrid-scale mixing, the *primary* BBL is attached below the normal ocean grid (Figures 16.1a and c). Assuming that the number of the vertical grids, `km`, is 50 before BBL option is applied, the number of vertical grids, `km`, becomes 51 (50 + 1) when BBL option is applied. In addition, to easily express the vertical exchange between the BBL cells and the inner ocean cells, we place a *dummy* BBL cell just below the bottom grid of each inner column and copy the temperature, salinity, and velocities from the primary BBL cell (Figures 16.1b and d).

Regarding vertical advection, primary vertical mass transport at the upper surface of BBL is defined at T-cell as in the interior (there is not vertical transport at the bottom of BBL). Even when the horizontal area of the bottom of the interior T-cell is smaller than that of BBL, all vertical mass transport at the top of BBL flows into or out of the bottom of the interior cell (Figure 16.1b and d). Tracers immediately above and below are carried by this vertical transport. In the same manner as explained in Chapter 6, the continuity equation for the U-cell is defined as an average of those of surrounding

T-cells (6.20). At each of the T-points connecting the bottom of the interior and the BBL, velocity (momentum) at all the U-cells in BBL that surround the T-point participates in the vertical momentum advection.

Regarding vertical mixing, tracer in BBL is exchanged with the one at the bottom of the interior as in vertical advection (the blue arrow in Figure 16.1b). On the other hand, velocity in BBL is exchanged with the one at the bottom of the water column of the same U-point (the blue arrow in Figure 16.1d), whose quarter cells may not necessarily constitute bottom T-cells that communicate with T-cells of the BBL.

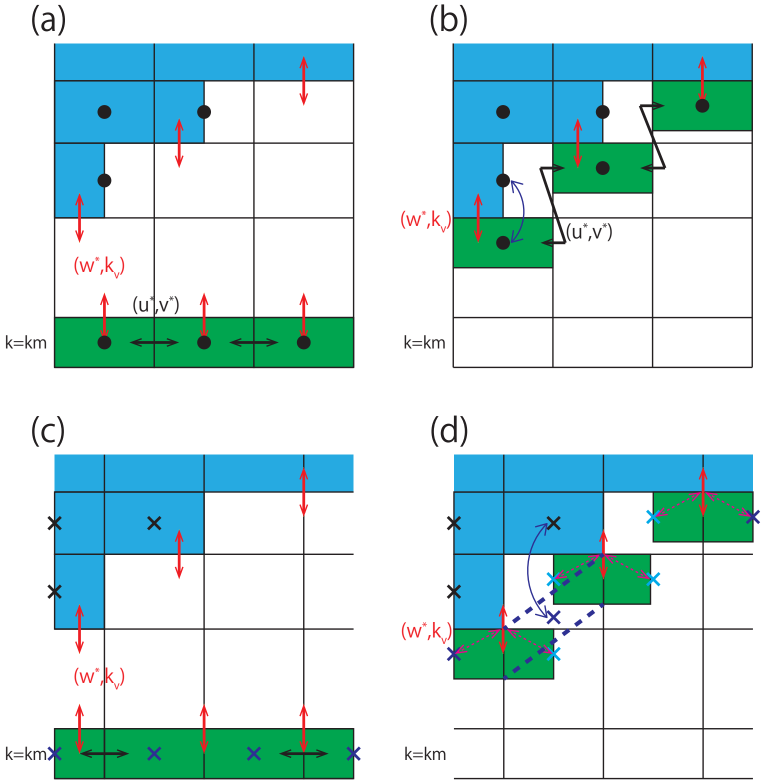

Figure16.1 Grid arrangement and schematic of horizontal and vertical exchange of tracers. (a, b) Arrangement of T-cell represented by "•" symbols. (c, d) Arrangement of U-cell represented by "×" symbols. The primary BBL cells lies at the bottom of the main (interior) oceanic domain (a and c). Horizontal advection and subgrid-scale mixing is computed there (solid black arrows). In relation to the interior, the T-cells in BBL are located just below the bottom of interior T-cells even when an interior T-cell consists of only one quarter of a U-cell (Figure 16.1b). The U-cells in BBL are regarded as slanting cells connecting neighboring T-points in BBL (dashed blue line in d). Vertical transport into and out of BBL occurs according to the primary vertical velocity defined at T-points (solid red arrows). Diagonally upward / downward advection of momentum may occur along the dashed red arrows in (d). Vertical mixing of tracers and momentum occurs between the cells indicated blue arrow in (b) and (d), respectively.

16.3 Pressure gradient terms

a. Formulation

BBL of MRI.COM is attached at the bottom of interior T-cells. U-cells of BBL are regarded as slanting and connecting the surrounding T-points. (Figure 16.1d). In other words, sea floor is considered to have gradient along U-cells.

The pressure perturbation term in eq. (2.54) is

$$\begin{aligned}\frac{1}{\rho_0}\nabla_s p' &= \frac{1}{\rho_0}\nabla p'\big|_{s=-H(x,y)} - \frac{1}{\rho_0}\partial_s p' \nabla H(x,y)\\ &= \frac{1}{\rho_0}\nabla p'\big|_{s=-H(x,y)} + \frac{g\rho'}{\rho_0} z_s \nabla H(x,y),\end{aligned} \tag{16.1}$$

where s means the depth coordinate along the sea floor. The geopotential term is

$$\frac{g\rho'}{\rho_0}\nabla_s z = \frac{g\rho'}{\rho_0}\nabla z\big|_{s=-H(x,y)} - \frac{g\rho'}{\rho_0} z_s \nabla H(x,y). \tag{16.2}$$

Then, the pressure gradient for a U-cell in BBL is expressed as

$$\frac{1}{\rho_0}\nabla_s p' + \frac{g\rho'}{\rho_0}\nabla_s z = \frac{1}{\rho_0}\nabla p'\Big|_{s=-H_{\mathrm{BBL}}(x,y)} + \frac{g\rho'}{\rho_0}\nabla z\Big|_{s=-H_{\mathrm{BBL}}(x,y)}. \tag{16.3}$$

b. Discretization

Horizontal gradient in (16.3) is evaluated along the bottom slope, not along the constant s-surface.

The depth of U-point in BBL ($z = z_{\mathrm{BBL}}$) that appears in the geopotential term is separated into time-independent ($-H_{\mathrm{BBL}}$) and time-dependent (z') part, so that

$$\frac{g\rho'}{\rho_0}\nabla z\Big|_{s=-H_{\mathrm{BBL}}(x,y)} = -\frac{g\rho'}{\rho_0}\nabla H_{\mathrm{BBL}} + \frac{g\rho'}{\rho_0}\nabla z'\Big|_{s=-H_{\mathrm{BBL}}(x,y)}. \tag{16.4}$$

The pressure perturbation at T-point in BBL, whose vertical grid index is $k = ktbtm$, is computed by adding an increment due the downward vertical integration along the T-point from the corner of the upper face of the U-cell at the bottom of interior, whose vertical grid index is $k = kbtm - 1$, to the T-cell in BBL ($k = ktbtm$). The finite difference form is given as follows:

$$\Delta p'_{i,j,ktbtm} = \sum_{l=kbtm-1}^{ktbtm-1} (\rho')_{i,j,l+\frac{1}{2}}(\mathtt{dzt})_{i,j,l+\frac{1}{2}} + g(\rho')_{i,j,ktbtm+\frac{1}{2}}\frac{(\mathtt{dzt})_{i,j,ktbtm+\frac{1}{2}}}{2}. \tag{16.5}$$

The gradient of pressure perturbation is computed as

$$\begin{aligned}(\nabla p_{\mathrm{BBL}}{}')_{i+\frac{1}{2},j+\frac{1}{2}} =& (\nabla_r p')_{i+\frac{1}{2},j+\frac{1}{2},kbtm-1} + \hat{\mathbf{x}}\frac{\frac{\Delta p'_{i+1,j+1,ktbtm}+\Delta p'_{i+1,j,ktbtm}}{2} - \frac{\Delta p'_{i,j+1,ktbtm}+\Delta p'_{i,j,ktbtm}}{2}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\\ &+ \hat{\mathbf{y}}\frac{\frac{\Delta p'_{i+1,j+1,ktbtm}+\Delta p'_{i,j+1,ktbtm}}{2} - \frac{\Delta p'_{i+1,j,ktbtm}+\Delta p'_{i,j,ktbtm}}{2}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}.\end{aligned} \tag{16.6}$$

The geopotential term is computed as

$$\begin{aligned}&\left[\frac{g\rho'}{\rho_0}\nabla z_{\mathrm{BBL}}\right]_{i+\frac{1}{2},j+\frac{1}{2}}\\ &= -\frac{g}{\rho_0}\left\{\left[\frac{\frac{\rho'_{i+1,j}+\rho'_{i,j}}{2}\frac{H_{\mathrm{BBL}i+1,j}-H_{\mathrm{BBL}i,j}}{\Delta x_{i+\frac{1}{2},j}} + \frac{\rho'_{i+1,j+1}+\rho'_{i,j+1}}{2}\frac{H_{\mathrm{BBL}i+1,j+1}-H_{\mathrm{BBL}i,j+1}}{\Delta x_{i+\frac{1}{2},j+1}}}{2}\hat{\mathbf{x}}\right]\right.\\ &\qquad\left. + \left[\frac{\frac{\rho'_{i,j+1}+\rho'_{i,j}}{2}\frac{H_{\mathrm{BBL}i,j+1}-H_{\mathrm{BBL}i,j}}{\Delta y_{i,j+\frac{1}{2}}} + \frac{\rho'_{i+1,j+1}+\rho'_{i+1,j}}{2}\frac{H_{\mathrm{BBL}i+1,j+1}-H_{\mathrm{BBL}i+1,j}}{\Delta y_{i+1,j+\frac{1}{2}}}}{2}\hat{\mathbf{y}}\right]\right\}\\ &\quad + \frac{g}{\rho_0}\left\{\left[\frac{\frac{\rho'_{i+1,j}+\rho'_{i,j}}{2}\frac{z'_{i+1,j}-z'_{i,j}}{\Delta x_{i+\frac{1}{2},j}} + \frac{\rho'_{i+1,j+1}+\rho'_{i,j+1}}{2}\frac{z'_{i+1,j+1}-z'_{i,j+1}}{\Delta x_{i+\frac{1}{2},j+1}}}{2}\hat{\mathbf{x}}\right] + \left[\frac{\frac{\rho'_{i,j+1}+\rho'_{i,j}}{2}\frac{z'_{i,j+1}-z'_{i,j}}{\Delta y_{i,j+\frac{1}{2}}} + \frac{\rho'_{i+1,j+1}+\rho'_{i+1,j}}{2}\frac{z'_{i+1,j+1}-z'_{i+1,j}}{\Delta y_{i+1,j+\frac{1}{2}}}}{2}\hat{\mathbf{y}}\right]\right\}.\end{aligned} \tag{16.7}$$

In the source code, the depth of T-point `hot` is used for H_{BBL}. To be exact, (`hot` – `dzt_cnst`/2) should be used. However, because the initial thickness of the BBL (`dzt_cnst`) is spatially uniform in MRI.COM, this omission would not affect the result.

16.4 Eddy effects

In the rotational frame, the dense water along the continental slope flows along f/h without the eddy effects. In this case, introducing the BBL model does not lead to a better representation of the overflow/downslope-flow in the Nordic Seas or on the continental shelf around Antarctica.

In the real world, eddy effects create the cross f/h flow, resulting in the overflow/downslope-flows. Jiang and Garwood (1996) investigated the three-dimensional features of downslope flows using an eddy-resolving model and demonstrated that the dense water descends roughly 45° left to the geostrophic contour (in the northern hemisphere) with vigorous eddy activity. The observation of significant eddy activity south of the Denmark Strait is consistent with this result. In order to incorporate this effect into the non-eddy resolving models, we apply Rayleigh drag whose coefficient is nearly equal to the Coriolis parameter, $\alpha \simeq f$. In this case, the geostrophic balance is modified and written as follows:

$$-fv = -\frac{1}{\rho_0}\frac{\partial p}{\partial x} - \alpha u, \tag{16.8}$$

$$fu = -\frac{1}{\rho_0}\frac{\partial p}{\partial y} - \alpha v, \tag{16.9}$$

After some algebra we obtain

$$-f\left(v - \frac{\alpha}{f}u\right) = -fv' = -\frac{1}{\rho_0}\frac{\partial p}{\partial x}, \tag{16.10}$$

$$f\left(u + \frac{\alpha}{f}v\right) = fu' = -\frac{1}{\rho_0}\frac{\partial p}{\partial y}, \tag{16.11}$$

where $v' = v - (\alpha/f)u$ and $u' = u + (\alpha/f)v$ correspond to the geostrophic velocity for the pressure gradient. If we put $\alpha = f$, the direction of the flow is 45° to the right of the pressure gradient. In general, the horizontal pressure field is parallel to the topographic contour near the deep water formation area. Thus, incorporating the Rayleigh drag coefficient, $\alpha = f$, leads to the dense water descending 45° to the left of the geostrophic contour.

Because this Rayleigh drag is thought to be caused by the eddy activity and is observed where the dense water descends to the deep layer, the coefficient of the Rayleigh drag should be parameterized as a function of the local velocity and topography. However it is very difficult to appropriately determine this function in coarse-resolution models. Accordingly, in the default setting of BBL option in MRI.COM, the depth range where $\alpha = f$ is arbitrarily set above 2500 m in the northern Atlantic and above 4000 m around Antarctica to represent the observed dense overflow/downslope-flow. Below those depths, α is set to zero. This setting may be explicitly specified by using namelist `nml_bblrayfrc` (Table 16.1).

16.5 Usage

In this section, we show how to use the BBL model in the MRI.COM.

- Add `BBL` option in the file `configure.in`.
- Set the lowest layer of the array `dz(km)` as the thickness of BBL. Fifty to one hundred meters is recommended for the thickness of BBL, which roughly corresponds to the observed thickness of the BBL layer near the Denmark Strait.
- Append `ho4bbl`, `exnnbbl` to `file_topo` after `ho4`, `exnn` (Section 25.2.2), following `docs/README_Options.md`.
- Set the region where non-zero Rayleigh drag coefficient is applied (`nml_bblrayfrc`; Table 16.1). The default setting is above 2500 m in the North Atlantic and above 4000 m around the Southern Ocean.

If model integration starts from the 3-D distribution of temperature and salinity, the temperature and salinity in the BBL cells are set to those in the lowermost inner cells.

Example of `configure.in` with `BBL` option

```
OPTIONS="BBL"
IMUT=184
JMUT=171
KM=48
```

The file that contains topographic information is read by the model at run time as follows (see also Section 25.2.2).

How topography data (`file_topo`) with BBL is read by model

```
      integer(4) :: ho4(imut,jmut),exnn(imut,jmut)
      integer(4) :: ho4bbl(imut,jmut),exnnbbl(imut,jmut)
      integer(4),parameter :: lun = 10
      open(lun, file=file_topo, form=unformatted)
      read(lun) ho4, exnn
      read(lun) ho4bbl, exnnbbl ! This is for BBL.
```

Table16.1 Namelist `nml_bblrayfrc` for setting the lower limit above which Rayleigh damping is applied in BBL

variable	units	description	usage
`deprfcn`	cm	subjective lower limit of depth (for northern hemisphere) above which Rayleigh damping is applied	default = 2500 m = 2500×10^2 cm
`deprfcs`	cm	subjective lower limit of depth (for southern hemisphere) above which Rayleigh damping is applied	default = 4000 m = 4000×10^2 cm

16.6 Usage notes

16.6.1 Limit of the region where BBL model should be applied

Ideally, the BBL model should be applied universally and its behavior should change according to the local oceanic state. In the real configuration, the BBL model is only effective near the abyssal water formation areas. When used in other areas, such as the near the equator, the BBL model does not improve the tracer and velocity fields. Furthermore, because the BBL model connects the model cells along the topography, it inevitably induces a water mass exchange between the cold abyssal water and the warm shallow water. Thus, unphysical diapycnal diffusion could occur with the BBL model. This effect is not severe in high latitudes where the difference in temperature between the shallow continental shelves and the deep layer is expected to be small, but it is extremely problematic for the cells in low latitudes. This problem is similar to the problem for typical σ-layer models. To prevent this, it is recommended that the BBL model should be applied only to the regions where the processes reproduced by the BBL model are thought to be important.

16.6.2 Limits of the BBL

Linear interpolation of the temperature and salinity along an extremely steep slope may cause problems. For the default setting of MRI.COM, the BBL model is not applied in such places and isolated grids. This setting may be modified by using namelist `nml_bblgeo_limit` (Table 16.2).

Table16.2 Namelist `nml_bblgeo_limit` for limiting the range where BBL scheme is applied

variable	units	description	usage
`hbblmin`	cm	for sea floor depth shallower than `hbbldmin`, BBL is not applied	default = 0 m
`bbldmax`	cm	if the sea floor is deeper than that of the neighboring point by `bbldmax`, the BBL is not applied	default = 10 000 m $10\,000 \times 10^2$ cm

16.6.3 Notes for program coding and analyses

In each vertical column, the BBL exists both at the bottom cell ($k = \texttt{km}$) and at the lowest ocean cell ($k = \texttt{exn}(\texttt{i}, \texttt{j})$). In general, the lowest ocean cell of the T-point and U-point for the same horizontal indices might differ. Thus, the treatment of topographic mask array `atexl` and `aexl` is always very confusing. At the bottom cells ($k = \texttt{km}$) that corresponds the primary BBL cells, we set `atexl` = 1 and `aexl` = 1 while we set `atexl` = 1 and `aexl` = 0 at the lowest ocean cells that corresponds the dummy BBL cells. In taking global average, the treatment of `atexl` = 1 in the BBL model needs special care to avoid double counting.

Part VI

Coupling with other models

Chapter 17

Sea ice

This chapter describes the sea ice model in MRI.COM. It treats formation, accretion, melting, and transfer of sea ice and snow. Heat, fresh water, salt, and momentum fluxes are exchanged with the ocean. Sea ice is categorized by its thickness, but it has a single layer. Snow does not have heat capacity (so-called zero-layer). Thus, it might be regarded as an intermediate complexity ice model.

The ice model can be only used as a part of the ocean model. In the program, a few main subroutines of the sea ice model are called from surface flux routines of the ocean model. The sea ice model is only called in the predictor step, not in the corrector step. The sea ice model uses the forward scheme in time integration. The sea ice part of an ocean-sea ice model gives sea surface boundary conditions to the oceanic part. Heat, fresh water, salt, and momentum are exchanged at their interfaces.

This chapter is organized as follows. Section 17.1 outlines the model. The following sections describe details of the solution procedure. According to the order of solving the equations, we deal with thermodynamics in Section 17.2, remapping among thickness categories in Section 17.3, dynamics (rheology) in Section 17.4, advection in Section 17.5, and ridging in Section 17.6, change from snow to sea ice in Section 17.7. Adjustment on the sea ice distribution is explained in Section 17.8. Section 17.9 summarizes fluxes from sea ice to ocean. The option of variable sea ice salinity (v5.1 or later) is explained in Section 17.10. Discretization issues are described in Section 17.13, coupling with an AGCM in Section 17.11, and nesting of the sea ice model in Section 17.12. Finally, usage notes are presented in Section 17.14.

State variables and fluxes in the sea ice model are summarized in Table 17.1, and constants and parameters in Section 17.1.6. For the thermal energy conservation in the sea ice model, see memorandum ("On the thermodynamics processes and the heat balance of the MRI.COM sea ice model (in Japanese)"[*]).

17.1 Outline

The ice model of MRI.COM is based on the ice-ocean coupled model of Mellor and Kantha (1989). For processes that are not explicitly discussed or included there, such as categorizing by thickness, ridging, and rheology, we adopt those of the Los Alamos sea ice model (CICE) version 3.14 (Hunke and Lipscomb, 2006).

17.1.1 The fundamental equation

The fundamental property that defines the state of sea ice is the fractional area as a function of location (μ, ψ) and thickness (h). The equation for this distribution function $(g(\mu, \psi, h))$ is

$$\frac{\partial g}{\partial t} = -\frac{\partial}{\partial h}(fg) - \frac{1}{h_\mu h_\psi}\left(\frac{\partial (g h_\psi u_I)}{\partial \mu} + \frac{\partial (g h_\mu v_I)}{\partial \psi}\right) + \chi_I^{ridge} + \chi_I^{slush} \tag{17.1}$$

where f is the thermodynamic growth rate of thickness, (u_I, v_I) is the velocity vector of ice pack, χ_I^{ridge} is the rate of change of distribution function caused by ridging (Hunke and Lipscomb, 2006, Section 3), and χ_I^{slush} is the transformation of drafted snow into sea ice (snow ice formation). MRI.COM does not deal with this continuity equation directly, but integrates it for each thickness category and predicts the integrated amount.

[*] https://mri-ocean.github.io/mricom/mri.com-user_doc.html

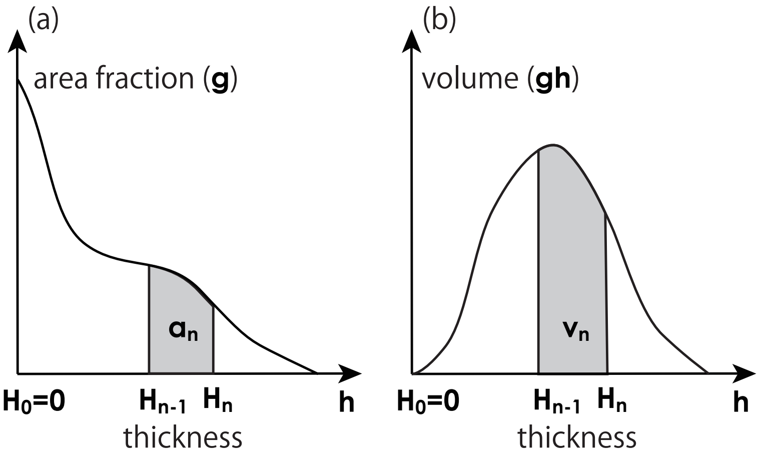

Figure17.1 Schematic views of categorization of sea ice thickness in terms of (a) fractional area a_n and (b) ice volume v_n.

17.1.2 Categorization of sea ice thickness

We discretize the thickness in several categories. If an ice pack is divided into N_c categories separated at $H_0 = 0\,\mathrm{m}$, H_1, H_2, ... and H_{N_c}, the fractional area of category n, a_n, is defined as follows:

$$a_n = \int_{H_{n-1}}^{H_n} g(h)dh. \tag{17.2}$$

As shown in Fig. 17.1, when g is considered as a continuous probability distribution function, a_n is a relative frequency of class n. If sea ice of category n does not exist on the grid, then $a_n = 0$.

Similar to the area fraction a_n, the sea ice volume of category n per unit area, v_n, is given below:

$$v_n = \int_{H_{n-1}}^{H_n} g(h)hdh. \tag{17.3}$$

Other prognostic variables, such as snow volume, v_s, internal energy of ice, e_n, surface temperature, T_3, are also defined for each category. Velocity is defined for an ice pack, the total ice mass in a grid cell. In the vertical direction, both sea ice and snow have one layer. Sea ice has heat capacity, but snow does not. The heat capacity for sea ice is due to brine and is represented by the temperature at the center of the sea ice. Figure 17.2 and Table 17.1 summarize symbols used in this chapter and their variable names in the source code of MRI.COM. Note that the symbol T is used to represent the temperature of any substance treated in this chapter. The unit is °C, though the use of an upper case letter (T) may recall absolute temperature.

Table17.1 State variables and fluxes used in the sea ice part (cf. Figure 17.2) and their variable names in the source code. Note that all water fluxes ($\mathrm{m\,s^{-1}}$) are expressed in terms of sea water with density ρ_o.

	Meaning	Units	Array name for each thickness category	Array name for average or sum over thickness categories

Continued on next page

Table 17.1 – continued from previous page

	Meaning	Units	Array name for each thickness category	Array name for average or sum over thickness categories
a_n, A	area fraction (compactness) (17.2)	1	`aicen`	`a0iceo apr`
v_n	volume per unit area (grid cell average of ice thickness) (17.3)	m	`hin`	`hi`
e_n	volume-integrated enthalpy ($\leq C_{po} m S_I v_n$)	J kg^{-1} m	`eicen`	-
v_s	snow volume per unit area (grid cell average of snow thickness)	m	`hsn`	`hsnw`
h_I	mean ice thickness ($= v_n / a_n$) (17.15)	m	`hicen`	`hiceo`
h_s	mean snow thickness ($= v_s / a_n$)	m	`hsnwn`	`hsnwo`
q_n	enthalpy of ice per unit volume ($= e_n / v_n$)	J kg^{-1}	`qicen`	-
T_3	skin temperature of upper surface	°C	`tsfcin`	`tsfci`
T_2	temperature at snow-ice interface	°C	`t2icen`	-
T_1	temperature of ice ($\leq m S_I$)	°C	`t1icen`	-
$T_{0_I} (= T_0)$	skin temperature of lower surface	°C	`t0icen(1 :)`	`t0iceo`
T_{0_L}	skin temperature of sea surface at open leads	°C	`t0icen(0)`	`t0icel`
$S_{0_I} (= S_0)$	skin salinity of lower surface	pss	`s0n(1 :)`	-
S_{0_L}	skin salinity of sea surface at open leads	pss	`s0n(0)`	-
Q_{I2}	heat flux in the upper half of ice ($= Q_S$)	W m^{-2}	`fheatu`	-
Q_{IO}	heat flux on the ice side of the ice bottom	W m^{-2}	`fheatn`	-
Q_{AO}	heat flux on the air side at open leads	W m^{-2}	`fheat`	-
Q_{LO}	ice-ocean latent heat flux due to melting of snow	W m^{-2}	-	`latent_forced_melt`
Q_{SO}	ice-ocean sensible heat flux due to melting of ice	W m^{-2}	-	`sens_io_all_melt`
F_{T_I}	heat flux on the ocean side of the ice bottom	W m^{-2}	`ftio`	-
F_{T_L}	heat flux on the ocean side at open leads	W m^{-2}	`ftao`	-
F_{S_I}	virtual salinity flux below sea ice driving the top layer of the ocean model	pss m s^{-1}	-	-
F_{S_L}	virtual salinity flux in open water driving the top layer of the ocean model	pss m s^{-1}	-	-
W	fresh water flux driving the top layer of the ocean model ($= A W_{IO} + (1 - A) W_{AO} + W_{RO} + W_{FR}$)	m s^{-1}	-	`wfluxi`
W_{AI}	fresh water flux due to snow fall at the upper surface of ice	m s^{-1}	-	`snowfall`
W_{AI}	fresh water flux due to sublimation at the upper surface of ice	m s^{-1}	`sublim`	-
W_{IO}	fresh water flux due to freezing and melting at the bottom of ice	m s^{-1}	`wio`	-
W_{AO}	fresh water flux due to freezing at open leads	m s^{-1}	`wao`	-
W_{ROice}	fresh water flux due to melting of sea ice at the upper surface of ice	m s^{-1}	-	`wrsi`

Continued on next page

Table 17.1 – continued from previous page

	Meaning	Units	Array name for each thickness category	Array name for average or sum over thickness categories
W_{ROsnow}	fresh water flux due to melting of snow at the upper surface of ice	$\mathrm{m\,s^{-1}}$	-	`wrss`
W_{FR}	fresh water flux due to formation of frazil ice	$\mathrm{m\,s^{-1}}$	-	`wrso`
u_I	zonal component of ice pack velocity	$\mathrm{m\,s^{-1}}$	-	`uice`
v_I	meridional component of ice pack velocity	$\mathrm{m\,s^{-1}}$	-	`vice`

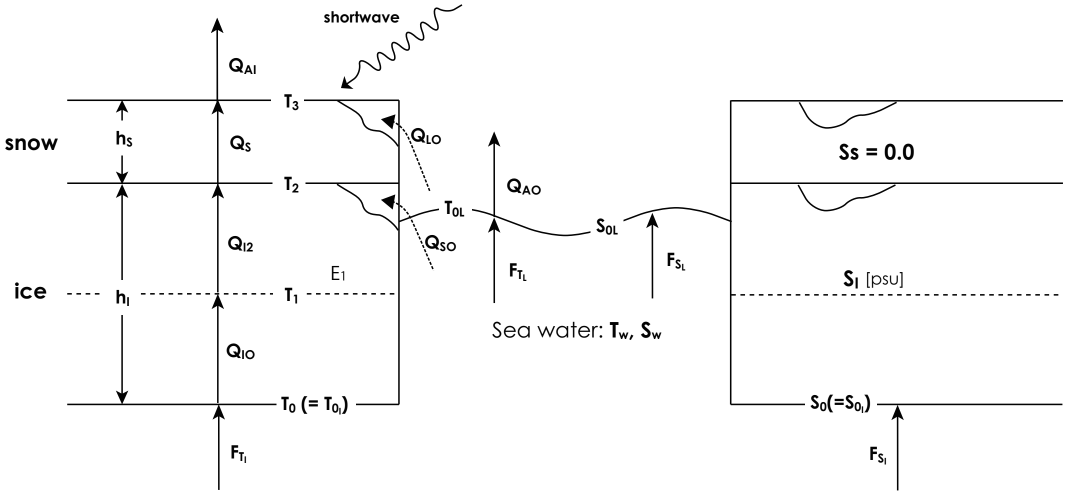

Figure17.2 Schematic diagram of the layer structure of the sea ice model, and related temperature and salinity. The left side is related to temperature and heat flux, and the right side is related to salinity. The central part without sea ice shows the modeling in open lead. See Sec. 17.2 and Table 17.1 for the layer structure and symbols.

17.1.3 The thickness-category governing equations

The sea ice model solves time evolution of the prognostic variables per unit area (sea ice volume, snow volume and sea ice enthalpy) integrated for each category, as defined in the previous section. Based on Eq. (17.1), the equation for the sea ice volume, v_n, is

$$\frac{\partial v_n}{\partial t} = f_I + \mathcal{A}(v_n) + \mathcal{D}(v_n) + \chi_I^{ridge} + \chi_I^{slush}, \tag{17.4}$$

where f_I indicates increase and decrease of sea ice due to thermodynamic processes, $\mathcal{A}(v_n)$ horizontal advection, and $\mathcal{D}(v_n)$ horizontal diffusion. Similarly, the equation for the snow volume, v_s, is

$$\frac{\partial v_s}{\partial t} = f_s + \mathcal{A}(v_s) + \mathcal{D}(v_s) + \chi_s^{ridge} + \chi_s^{slush}. \tag{17.5}$$

The equation for the sea ice enthalpy, e_n, is

$$\frac{\partial e_n}{\partial t} = f_e + \mathcal{A}(e_n) + \mathcal{D}(e_n) + \chi_e^{ridge} + \chi_e^{slush}. \tag{17.6}$$

It should be noted that the physical quantities divided by the density are treated as the conserved quantity, instead of mass conservation and energy conservation.

The growths of ice and snow, f_I and f_s, and the change of enthalpy, f_e, are computed by solving thermodynamic processes (Section 17.2). Using this result, thickness categories are remapped and the fractional area a_n is updated (Section 17.3). To compute the velocity of the ice pack (u_I, v_I), we have to solve the momentum equation (Section 17.4). Then, based on velocity (u_I, v_I), the advection terms, $\mathcal{A}$, and the diffusion terms, $\mathcal{D}$, are calculated (Section 17.5). Using the transported ice distribution function, the ridging process (χ^{ridge}) is solved (Section 17.6). Then, conversion of snow to sea ice due to slush formation, χ^{slush}, is obtained based on the predicted snow thickness and sea ice thickness (Section 17.7).

The outline of the thermodynamic processes is shown in Fig. 17.3. The amounts of change due to each process are calculated from the state variables and fluxes shown in Fig. 17.2 so as to satisfy the conservation laws. The formulation and solving procedure of each process are presented in later sections.

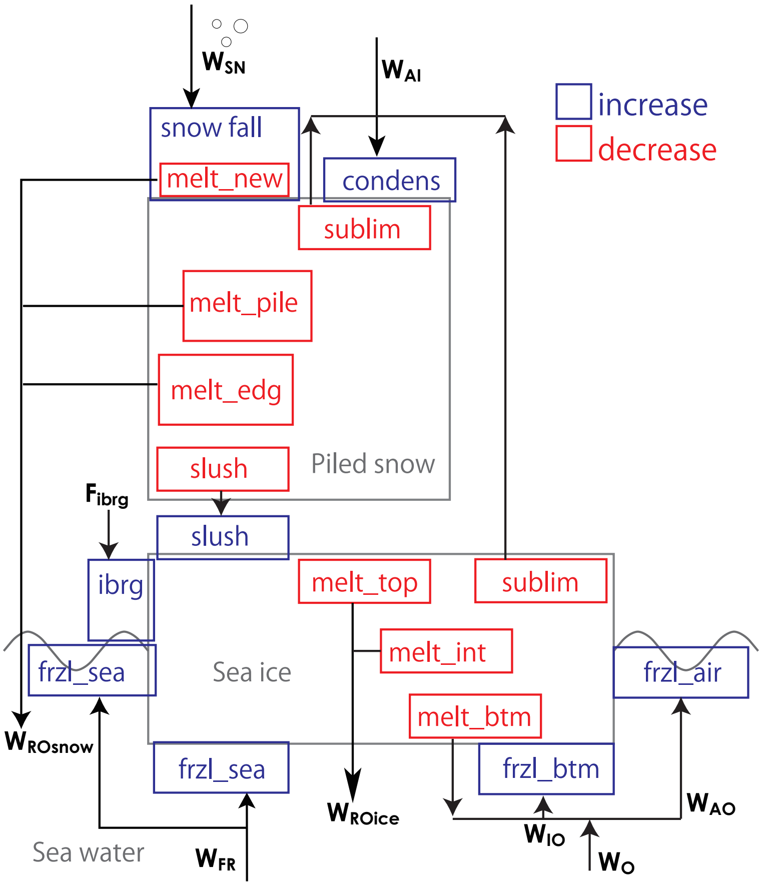

Figure17.3 Schematic diagram of changes of sea ice volume and snow volume due to thermodynamic processes and slash formation. Blue boxes indicate increase and red boxes indicate decrease. Arrows indicate water fluxes between sea ice and ocean or atmosphere.

17.1.4 Thermal energy of sea ice

In considering thermodynamics, thermal energy of sea ice should be defined. The basis of energy (i.e., zero energy) is defined here as that of sea water at 0 °C. The thermal energy (enthalpy; $E(T,r)$) of sea ice that has temperature $T(<0)$ [°C] and brine (salt water) fraction r is the negative of the energy needed to raise the temperature to 0 °C and melt all of it:

$$E(T,r) = r(C_{po}T) + (1-r)(-L_F + C_{pi}T), \tag{17.7}$$

where C_{po} and C_{pi} are the specific heats of sea water and sea ice, and L_F is the latent heat of melting/freezing. Thus defined, the energy of sea ice is negative definite.

Assuming that salinity in the brine is S_b and salinity in the other sea ice is 0, the average salinity S_I in the sea ice is given by

$$S_I = rS_b. \tag{17.8}$$

On the other hand, assuming that the change rate of the freezing temperature with respect to salinity is a constant negative value m, the freezing point in the brine is mS_b. In addition, the temperature of the entire sea ice, T_1, is assumed to be uniformly the freezing point, mS_b:

$$T_1 = mS_b. \tag{17.9}$$

Based on the above two equations, the brine fraction of sea ice, r, can be determined from S_I, T_1,

$$r = mS_I/T_1. \tag{17.10}$$

Since $r \leq 1$, a constraint of $T_1 \leq mS_I = T_{frz}$ is obtained, where T_{frz} is the freezing point of sea water with a salinity of S_I. As in Mellor and Kantha (1989), the specific heat of snow is not considered.

In the sea ice model, enthalpy E (more accurately, enthalpy integrated for each category, e_n), since it is a conserved quantity, is easier to be handled than temperature. Thus, T_1 is determined diagnostically based on E by solving the quadratic equation with (17.10),

$$E(T_1, r) = r(C_{po}T_1) + \big(1 - r\big)(-L_F + C_{pi}T_1), \tag{17.11}$$

which gives

$$T_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \tag{17.12}$$

where

$$a = C_{pi}, \quad b = -L_F - E + mS_I(C_{po} - C_{pi}), \quad c = mS_I L_F. \tag{17.13}$$

Figure 17.4 shows that the other root is unphysical. [†]

In the sea ice model, enthalpy is often calculated from T_1 and S_I by (17.11) on the premise of Eq. (17.10). Hereafter, sea ice enthalpy calculated from S_I instead of r is referred to as E_S.

$$E_S(T_1, S_I) = mS_I\left(C_{po} - C_{pi} + \frac{L_F}{T_1}\right) + C_{pi}T_1 - L_F, \tag{17.14}$$

The domain of definition for S_I is $S_I \geq 0$, while that for T_1 is $T_1 \leq mS_I$.

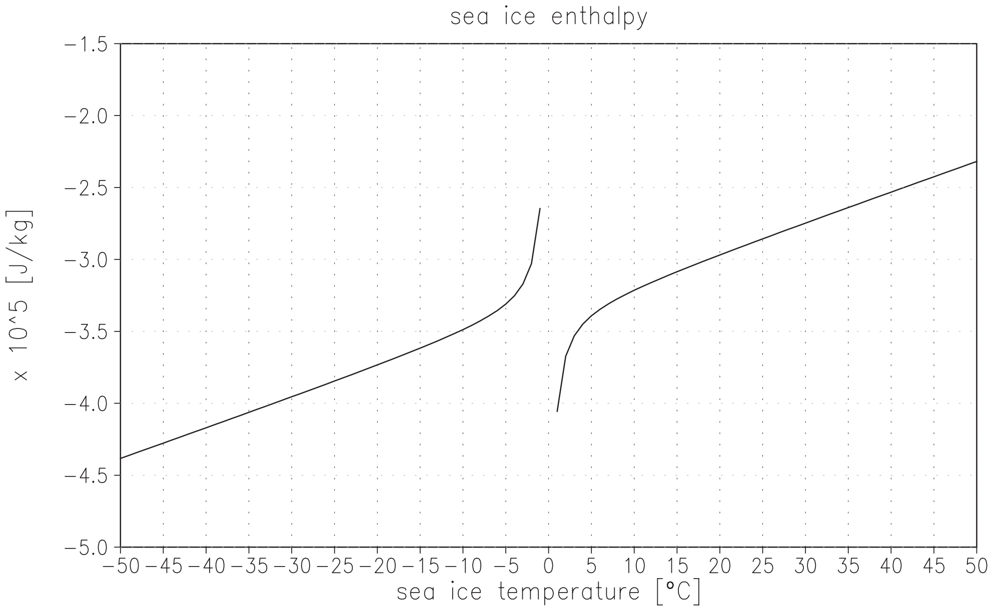

Figure17.4 Sea ice enthalpy E (17.11) as a function of sea ice temperature (T_1). Units in J kg^{-1}.

17.1.5 Diagnostic variables

Secondary diagnostic variables used in the sea ice model are summarized here.

The average sea ice thickness h_I for category n is determined as follows, if a_n is not zero,

$$h_I = v_n / a_n. \tag{17.15}$$

Similarly, the average snow thickness h_s is

$$h_s = v_s / a_n. \tag{17.16}$$

[†] It does not meet the assumption of Eq. (17.9.) When $r = 1$, that is, $T_1 = T_{frz} = mS_I$, enthalpy E takes the maximum value $C_{po}mS_I$ (sea water at freezing temperature).

The total sea ice area fraction for all categories is expressed as A.

$$A = \sum_{n=1}^{N_C} a_n. \tag{17.17}$$

The fraction of open lead is a_0,

$$a_0 = 1 - A. \tag{17.18}$$

If area fraction deviates from the domain, from zero to one, due to numerical error, it is corrected as follows:

- If $a_n < 0$, a_n is set as 0.
- If $A > 1$, a_n is replaced by a_n/A.

These two modifications guarantee $0 \leq A \leq 1$. Since area fraction itself is not a conserved quantity, such modification does not affect consistency of the model.

In the sea ice model, areas where the ocean is exposed to the atmosphere, that is, where there is no sea ice, may be treated differently depending on whether or not there is sea ice on the same grid. In this chapter, the former ($0 < a_0 < 1$) is called "open lead" and the latter ($a_0 = 1$) is called "open water".

17.1.6 Physical constant, parameters

Since the ice part is coded in SI units, constants and parameters are written in SI units.

a. Thermodynamics

Table17.2 Physical constants and parameters relevant to sea ice thermodynamics

parameter	value	variable name in MRI.COM
Thermal ice conductivity	$k_I = 2.04\,\mathrm{J\,m^{-1}\,s^{-1}\,K^{-1}}$	`cki`
Thermal snow conductivity	$k_s = 0.31\,\mathrm{J\,m^{-1}\,s^{-1}\,K^{-1}}$	`cks`
Specific heat of sea water	$C_{po} = 3990\,\mathrm{J\,kg^{-1}\,K^{-1}}$	`cp0`
Specific heat of air	Equation (14.130)	`cpair`
Specific heat of ice	$C_{pi} = 2093\,\mathrm{J\,kg^{-1}\,K^{-1}}$	`cpi`
Specific heat of snow	$C_{ps} = 0.0\,\mathrm{J\,kg^{-1}\,K^{-1}}$	—
Stefan Boltzmann constant	$\sigma = 5.67 \times 10^{-8}\,\mathrm{W\,m^{-2}\,K^{-4}}$	`stbl`
Albedo of open ocean surface	$\alpha_o = 0.1$ (default)	`albw`
Albedo of ice	$\alpha_I = 0.6$ (default)	`albi`
Albedo of snow	$\alpha_s = 0.75$ (default)	`albs`
Emissivity of ocean surface	$\epsilon_o = 1.0$	`eew`
Emissivity of ice surface	$\epsilon_I = 1.0$ (default)	`emissivity_sea_ice`
Emissivity of snow surface	$\epsilon_s = 1.0$ (default)	`emissivity_snow`
neutral bulk transfer coefficient for sensible heat	$C_{HAI} = 1.5 \times 10^{-3}$ (default)	`c_transfer_sensible_heat_ai`
neutral bulk transfer coefficient for latent heat	$C_{EAI} = 1.5 \times 10^{-3}$ (default)	`c_transfer_latent_heat_ai`
Latent heat of fusion	$L_F = 3.347 \times 10^5\,\mathrm{J\,kg^{-1}}$	`alf`
Latent heat of sublimation	Equation (14.132)	`rlth`
Constants for fusion phase equation: $T_f = mS + nz$	$m = -0.0543$ °C/pss $n = -0.000\,759\,\mathrm{°C\,m^{-1}}$	`xmxm` `xnxn`
Ice roughness parameter	$z_{0_I} = 0.05 h_I/3$	`z0`

Continued on next page

Table 17.2 – continued from previous page

parameter	value	variable name in MRI.COM
Salinity of sea ice	$S_I = 4.0\,\mathrm{pss}$	si
von Karman's constant	$k = 0.4$	xk
Thickness/compactness diffusion of ice	$\kappa_H = 1.0 \times 10^3\,\mathrm{m^2\,s^{-1}}$	akh
Seawater kinematic viscosity	$\nu = 1.8 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$	anu
Seawater heat diffusivity	$\alpha_t = 1.39 \times 10^{-7}\,\mathrm{m^2\,s^{-1}}$	at
Seawater salinity diffusivity	$\alpha_b = 6.8 \times 10^{-10}\,\mathrm{m^2\,s^{-1}}$	as
Turbulent Prandtl number	$P_{rt} = 0.85$	prt
b in eqs (17.118),(17.120)	$b = 3.14$	ab
sea ice enthalpy (17.14)	$E_S(T_1, S_I)\,\mathrm{J\,kg^{-1}}$	ts2qice_jpkg

b. Dynamics

Table17.3 Physical constants and parameters relevant to sea ice dynamics

parameter	value	variable name in MRI.COM
Density of sea water	$\rho_o = 1036\,\mathrm{kg\,m^{-3}}$	ro0
Density of air	ρ_a, Equation (14.133)	roair
Density of sea ice	$\rho_I = 900\,\mathrm{kg\,m^{-3}}$	rice
Density of snow	$\rho_s = 330\,\mathrm{kg\,m^{-3}}$	
Reference snow density (ratio between snow and water)	ρ_s/ρ_o	rdsw
e-folding constant for ice pressure	$c^* = 20.0$ (default)	e_fold_const_ice_prs
pressure scaling factor	$P^* = 2.75 \times 10^4\,\mathrm{N\,m^{-2}}$ (default)	prs_scale_factor
drag coefficient (air-ice)	$C_{DAI} = 3.0 \times 10^{-3}$ (default)	c_drag_ai
drag coefficient (ice-ocean)	$C_{DIO} = 5.5 \times 10^{-3}$ (default)	c_drag_io
yield curve axis ratio	$e = 2.0$	elips
scaling factor for Young's modulus	$E_o = 0.25$	eyoung
water turning angle	$\theta_o = \pm cz$ (positive/negative in the northern/southern hemisphere, c: $1°\mathrm{m}^{-1}$, z: the first layer depth [m], default)	turning_angle_drag_io
air turning angle	$\theta_a = 0°$ (not considered)	–

17.2 Thermodynamic processes

Vertical one-dimensional thermodynamic processes are solved in each category based on Mellor and Kantha (1989). The sea ice model has a layered structure of sea ice (thickness: h_I) and piled snow (thickness: h_s), as shown in Fig. 17.2. The former is further divided into upper and lower halves. Thus a column of sea ice has three vertical layers. The temperatures at the layer boundaries are T_0, T_1, T_2, and T_3 from the bottom. Heat fluxes within each layer are Q_{IO}, Q_{I2}, and Q_S from the bottom. The heat flux at the air-ice interface is Q_{AI} and that at the ice-ocean interface is F_{T_I}. Sea ice is in fact categorized by thickness, and each symbol should have a suffix (n) of the category number. At open leads or open water, symbols have the suffix L. Sea surface temperature of the ocean is indicated by T_w, and sea surface salinity by S_w.

Although the surface fluxes are positive downward (positive toward the ocean) in the ocean model, the sea ice part is coded such that fluxes are positive upward. In this section, we assume that the fluxes are positive upward.

17.2.1 Fluxes at air-ice interface

To solve the thermodynamic processes, we first evaluate the air-ice heat flux, Q_{AI}, and the freshwater flux, W_{AI} . The heat flux, Q_{AI}, is expressed as follows:

$$Q_{AI} = -(1 - \alpha_I)SW - LW + Q_{SI} + Q_{LI} + \epsilon_I \sigma (T_3 + 273.15)^4, \tag{17.19}$$

where SW is downward shortwave radiation flux, LW is downward longwave radiation flux, Q_{SI} is sensible heat flux, Q_{LI} is latent heat flux, α_I is albedo, ϵ_I is emissivity, and σ is the Stefan-Boltzmann constant. Except for the last term of Eq.

(17.19) (thermal radiation), each of these terms is calculated as follows. The freshwater flux (does not include snow fall), W_{AI}, is caused only by sublimation and condensation on the top of the sea ice.

■ i. Shortwave: The downward shortwave radiation is represented by SW. The albedo of sea ice is α_I, which is 0.82 for cold ($T_3 < -1\,°\mathrm{C}$) snow, 0.73 for melting snow, and 0.64 for bare ice (while melting).

One might use a more sophisticated albedo scheme included in the Los Alamos sea ice model (CICE; Hunke and Lipscomb, 2006) by choosing `CALALBSI` option. We briefly describe how the incoming shortwave radiation is treated by this scheme. The downward shortwave radiation is treated at each interface as follows: First, among the net absorbed shortwave flux ($= (1 - \alpha_I)SW$), some fraction (i_0) penetrates into ice and the rest is absorbed at the surface and used to warm the upper interface. See Table 17.4 for the specific value of i_0. Second, the part penetrating into the ice $SW_{surface}$,

$$SW_{surface} = (1 - \alpha_I)i_0 SW, \tag{17.20}$$

is attenuated according to Beer's Law with the bulk extinction coefficient $\kappa_i = 1.4\,\mathrm{m}^{-1}$. The attenuated part ΔSW_{ice} is used to warm the ice interior (Sec. 17.2.3)

$$SW_{bottom} = (1 - \alpha_I)i_0 SW \times \exp(-\kappa_i h_I), \tag{17.21}$$

$$\Delta SW_{ice} = SW_{surface} - SW_{bottom}. \tag{17.22}$$

Last, the rest SW_{bottom} enters the ocean (Sec. 14.3).

The albedos and penetration coefficients of CICE are listed on Table 17.4. The property f_{snow} is the snow fraction of the upper surface of the ice, which is expressed as follows:

$$f_{\mathrm{snow}} = \frac{h_s}{h_s + h_{snowpatch}}, \tag{17.23}$$

where h_s is the snow thickness and $h_{snowpatch} = 0.02\,\mathrm{m}$.

If the ice thickness (h_I) is less than $h_{\mathrm{ref}} = 0.5\,\mathrm{m}$, the albedo of thin ice is computed as

$$\alpha_{thinice} = \alpha_o + \beta(\alpha_{coldice} - \alpha_o), \tag{17.24}$$

where

$$\beta = \frac{\arctan(a_r h_I)}{\arctan(a_r h_{\mathrm{ref}})}, \quad a_r = 4.0, \tag{17.25}$$

and α_o is the albedo of the ocean.

If the top surface temperature T_3 becomes $-1 < T_3 < 0\,°\mathrm{C}$, the albedo of melting ice and snow is computed as

$$\alpha_{meltice} = \alpha_{thinice} - \gamma_i(T_3 + 1.0), \tag{17.26}$$

$$\alpha_{meltsnow} = \alpha_{coldsnow} - \gamma_s(T_3 + 1.0), \tag{17.27}$$

where the condition $\alpha_{meltice} > \alpha_o$ is imposed. Using the snow fraction on the surface of the ice f_{snow}, the total albedo is computed as

$$\alpha_i = \alpha_{meltice}(1 - f_{\mathrm{snow}}) + \alpha_{meltsnow} f_{\mathrm{snow}}. \tag{17.28}$$

The albedos for visible and near infra-red wave lengths are computed separately. If the shortwave flux is given as the sum of all four components (direct and diffuse for visible and near infra-red wave lengths), a constant ratio (visible) : (near infra-red) = 0.575 : 0.425 is assumed, and the total albedo is computed as the weighted average.

Table17.4 Albedo and surface transparency of the albedo scheme of CICE.

	near infra-red (> 700 nm)	visible (< 700 nm)
albedo for cold snow $\alpha_{coldsnow}$ ($T_3 < -1\,°\mathrm{C}$)	0.70	0.98
albedo for cold ice $\alpha_{coldice}$ ($T_3 < -1\,°\mathrm{C}$, $h_i > 0.5\,\mathrm{m}$)	0.36	0.78
reduction rate of albedo for melting ice γ_i ($-1\,°\mathrm{C} < T_3 < 0$, $h_i < 0.5\,\mathrm{m}$)	−0.075 /°C	−0.075 /°C
reduction rate of albedo for melting snow γ_s ($-1\,°\mathrm{C} < T_3 < 0$, $h_i < 0.5\,\mathrm{m}$)	−0.15 /°C	−0.10 /°C
fraction of transparent shortwave flux through the ice surface (i_0)	0.0	$0.7 \times (1.0 - f_{\mathrm{snow}})$

■ ii. Longwave: The downward longwave radiation from the atmosphere is represented by LW in (17.19). The black body radiation from the ice surface is $\epsilon_I \sigma (T_3 + 273.15)^4$, where ϵ_I is emissivity, and σ is the Stefan-Boltzmann constant. Hereinafter, we use

$$LW_I(T_3) = LW - \epsilon_I \sigma (T_3 + 273.15)^4 \tag{17.29}$$

as the net longwave radiation.

■ iii. Sensible heat flux: The sensible heat flux (Q_{SI} in (17.19)) is computed using a bulk formula:

$$Q_{SI}(T_3) = \rho_a C_{pa} C_{HAI} U_{10} (T_3 - T_A), \tag{17.30}$$

where ρ_a is the density of air, C_{pa} is the specific heat of air, C_{HAI} is the bulk transfer coefficient for heat, U_{10} is the scalar wind speed at 10 m, T_A is the surface air temperature. The derivative, $\partial Q_{SI}/\partial T_3$, is also used in later calculations.

$$\frac{\partial Q_{SI}}{\partial T_3} = \rho_a C_{pa} C_{HAI} U_{10}. \tag{17.31}$$

■ iv. Latent heat flux: The latent heat flux (Q_{LI} in (17.19)) is computed using a bulk formula:

$$Q_{LI}(T_3) = \rho_a L_s C_{EAI} U_{10} (q_i - q_A), \tag{17.32}$$

where L_s is the latent heat of sublimation, C_{EAI} is the bulk transfer coefficient for moisture, q_i is the saturation humidity at T_3, and q_A is the specific humidity of air. Section 14.12.2 details a computing method for q_i. The derivative, $\partial Q_{LI}/\partial T_3$ is also used later, as with sensible heat.

$$\frac{\partial Q_{LI}}{\partial T_3} = \rho_a L_s C_{EAI} U_{10} \frac{\partial q_i}{\partial T_3}. \tag{17.33}$$

■ v. Bulk coefficient over sea ice: Three schemes are available for bulk coefficients on sea ice, depending on the options `BULKNCAR`, `BULKECMWF`.

If neither `BULKNCAR`, `BULKECMWF` is specified, the following bulk coefficients are used according to Mellor and Kantha (1989).

$$C_{\mathrm{DAI}} = 3.0 \times 10^{-3} \tag{17.34}$$

$$C_{\mathrm{HAI}} = C_{\mathrm{EAI}} = 1.5 \times 10^{-3}. \tag{17.35}$$

These coefficients are given by namelist `nml_air_ice` at run time. See Table 17.11. The wind speed, air temperature, and specific humidity are used without altitude correction.

If `BULKNCAR` is specified, the constant value specified by `nml_air_ice` is usually used as neutral bulk coefficients at 10 m. To use the same scheme as Large and Yeager (2004), the following values must be specified at runtime.

$$C_{\mathrm{DAI10}} = C_{\mathrm{HAI10}} = C_{\mathrm{EAI10}} = 1.63 \times 10^{-3}. \tag{17.36}$$

The procedures except for neutral bulk coefficients is the same as Section 14.9.2. The bulk coefficients, air temperature and specific humidity are corrected for the altitude at which wind speed is defined, and then used for calculation of turbulent fluxes.

If `BULKECMWF` is specified, roughness length is calculated by the following equation (ECMWF, 2016b):

$$z_0 = \max\left(10^{-3}, 0.93 \times 10^{-3}(1 - a_n) + 0.05 \times 10^{-3} \exp\left[-17(a_n - 0.5)^2\right]\right), \tag{17.37}$$

$$z_{0\mathrm{T}} = 10^{-3}, \tag{17.38}$$

$$z_{0\mathrm{E}} = 10^{-3}. \tag{17.39}$$

Except for this, the procedure is the same as Section 14.9.3. Since the roughness length is uniquely determined by the sea ice fraction a_n, iteration for calculating the roughness length is unnecessary.

Regardless of which scheme you use, the wind stress over sea ice can be calculated with `TAUBULK` option.

■ vi. Fresh water flux: Fresh water loss due to sublimation is computed as

$$W_{AI} = \rho_a C_{EAI} U_{10}(q_i - q_A)/\rho_o, \tag{17.40}$$

where W_{AI} is converted to the volume of sea water by dividing by ρ_o. As is clear from comparison with Eq. (17.32), W_{AI} is determined by the latent heat flux at the air-ice boundary, Q_{LI}.

Finally, in preparation for the next section, we define the following heat flux at the ice-air boundary, Q_{AI}^{old} [W m $^{-2}$] (positive upward).

$$Q_{AI}^{old} = -(1-\alpha_I)(1-i_0)SW - LW_I(T_3^{old}) + Q_{SI}(T_3^{old}) + Q_{LI}(T_3^{old}) + \epsilon_I \sigma (T_3^{old} + 273.15)^4. \tag{17.41}$$

Permeation to the inner region is excluded, and T_3^{old} is the top surface temperature at the previous time step.

17.2.2 Interface between air and snow/sea ice

In response to the air-ice fluxes obtained in the previous section, the existing sea ice and snow are changed. Specifically, the following three steps are performed.

condensation and sublimation

Condensation and sublimation of the freshwater flux, W_{AI} [m / s], causes water mass change per unit area, Δm [kg],

$$\Delta m = -W_{AI} a_n \rho_o \Delta t, \tag{17.42}$$

during a time step interval of Δt for category n. Positive values indicate increase of sea ice or snow mass (condensation), and negative values indicate decrease (sublimation). (It should be noted that this effect works only on sea ice, not in the open water, open lead, and sea ice-free categories.) On the other hand, the mass of snow, m_s, and the mass of sea ice, m_I, existing at the current step for category n are

$$m_s = v_s \rho_{snow}, \qquad m_I = v_n \rho_I. \tag{17.43}$$

Using these masses, the mass change of snow, Δm_s, and that of sea ice, Δm_I, due to condensation and sublimation are calculated according to the following procedure. If condensation occurs, only snow is increased. If sublimation occurs, snow is reduced, and then, sea ice is reduced. When expressed with conditional branch,

$$\begin{cases} \Delta m_s = \Delta m, & \Delta m_I = 0 & \text{if} \quad -m_s < \Delta m \\ \Delta m_s = -m_s, & \Delta m_I = \Delta m + m_s & \text{if} \quad -m_s - m_I < \Delta m < -m_s \\ \Delta m_s = -m_s, & \Delta m_I = -m_I & \text{if} \quad \Delta m < -m_s - m_I \end{cases} \tag{17.44}$$

Note that the first condition includes two situations where condensation occurs ($0 < \Delta m$) and only snow sublimation occurs ($-m_s < \Delta m < 0$).

These conditions can be summarized as follows to express volume change of snow, Δv_s [m], and volume change of sea ice, Δv_n [m].

$$\begin{cases} \Delta v_s = -W_{AI} a_n \frac{\rho_o}{\rho_s} \Delta t, & \Delta v_n = 0 & \text{if} \quad -m_s < \Delta m \\ \Delta v_s = -v_s, & \Delta v_n = -W_{AI} a_n \frac{\rho_o}{\rho_I} \Delta t + v_s \frac{\rho_s}{\rho_I} & \text{if} \quad -m_s - m_I < \Delta m < -m_s \\ \Delta v_s = -v_s, & \Delta v_n = -v_n & \text{if} \quad \Delta m < -m_s - m_I \end{cases} \tag{17.45}$$

Since sea ice salinity is set constant as S_I, salinity of the first layer of the ocean is changed so as to compensate for the decrease of sea ice salinity.

$$S'_w = S_w - \Sigma_n \frac{S_I \Delta v_n \rho_I}{\Delta z \rho_o}. \tag{17.46}$$

The change of sea ice enthalpy, Δe_n, is proportional to the change of sea ice volume.

$$\Delta e_n = q_n \Delta v_n, \tag{17.47}$$

where q_n is sea ice enthalpy per unit volume at the previous step.

When the variable sea ice salinity option is used (v5.1), the change of integrated sea ice salinity, ΔS_v is always zero, because salinity included in sea ice does not increase or decrease during sublimation.

Determination of sea ice surface temperature, T_3

Sea ice surface temperature at the next time step, T_3^{new}, is determined so that the heat flux equilibrates on the sea ice surface. [‡] The basis of this calculation is the relationship between heat fluxes and temperature at the air-ice boundary, in the snow layer, and in the sea ice upper layer, as shown in Fig. 17.2. First, the heat flux at the air-ice boundary, Q_{AI}, is given from (17.19) except for permeation into sea ice:

$$Q_{AI} = -(1-\alpha_I)(1-i_0)SW - LW + Q_{SI}(T_3^{\mathrm{new}}) + Q_{LI}(T_3^{\mathrm{new}}) + \epsilon_I \sigma (T_3^{\mathrm{new}} + 273.15)^4, \tag{17.48}$$

where SW and LW have already been obtained, but Q_{LI}, Q_{SI} and thermal radiation depend on T_3^{new}. Next, the heat flux in the snow, Q_S, is constant within the snow layer (we neglect the heat capacity of snow,) and is computed as

$$Q_S = \frac{k_s}{h_s}(T_2^{\mathrm{new}} - T_3^{\mathrm{new}}), \tag{17.49}$$

where h_s is the thickness of the snow layer given by (17.16), and k_s is the thermal conductivity of snow. In the upper half of the ice layer, the heat flux Q_{I2} is computed as follows:

$$Q_{I2} = \frac{k_I}{h_I/2}(T_1^{\mathrm{old}} - T_2^{\mathrm{new}}), \tag{17.50}$$

where k_I is the thermal conductivity of sea ice. Since we neglect heat capacity of snow, the following equation is obtained:

$$Q_S = Q_{I2}. \tag{17.51}$$

We adopt the semi-implicit method described below to decide T_3^{new}. [§] The balanced surface temperature is computed by assuming that the fluxes on both sides are the same. That is, using the value at the previous time step, T_3^{old}, as the initial estimate,

$$T_3^{new} = T_3^{old} + \delta T_3. \tag{17.52}$$

the following equation is assumed:

$$Q_S = Q_{AI}. \tag{17.53}$$

Thus, based on T_3^{old} and T_1^{old}, the unknown variables, $\delta T_3, T_2^{new}, Q_{AI}, Q_S$ and Q_{I2} are found.

The equation for δT_3 can be derived using Eqs. (17.48), (17.49), (17.50), (17.51), (17.52) and (17.53).

$$\frac{k_I}{h_I/2}\{T_1^{old} - \frac{k_s(h_I/2)(T_3^{old}+\delta T_3) + k_I h_s T_1^{old}}{k_s(h_I/2)+k_I h_s}\} = Q_{LI}(T_3^{old}+\delta T_3) + Q_{SI}(T_3^{old}+\delta T_3) - (1-\alpha_I)(1-i_0)SW - LW + \epsilon_I\sigma\{(T_3^{old}+\delta T_3)+273.15\}^4. \tag{17.54}$$

By expanding the specific heat, latent heat, and thermal radiation in a Taylor series, we have,

$$\frac{k_I}{h_I/2}\{T_1^{old} - \frac{k_s(h_I/2)(T_3^{old}+\delta T_3) + k_I h_s T_1^{old}}{k_s(h_I/2)+k_I h_s}\} = Q_{LI}(T_3^{old}) + Q_{SI}(T_3^{old}) + \frac{\partial Q_{LI}}{\partial T_3}\delta T_3 + \frac{\partial Q_{SI}}{\partial T_3}\delta T_3 - (1-\alpha_I)(1-i_0)SW - LW_I(T_3^{old}) + 4\epsilon_I\sigma(T_3^{old}+273.15)^3\delta T_3. \tag{17.55}$$

Using this, we compute δT_3:

$$\delta T_3 = \frac{-Q_{SI}(T_3^{old}) - Q_{LI}(T_3^{old}) + (1-\alpha_I)(1-i_0)SW + LW_I(T_3^{old}) + c(T_1^{old}-T_3^{old})}{\frac{\partial Q_{LI}}{\partial T_3} + \frac{\partial Q_{SI}}{\partial T_3} + 4\epsilon_I\sigma(T_3^{old}+273.15)^3 + c}, \tag{17.56}$$

$$= \frac{-Q_{AI}^{old} + c(T_1^{old}-T_3^{old})}{\frac{\partial Q_{LI}}{\partial T_3} + \frac{\partial Q_{SI}}{\partial T_3} + 4\epsilon_I\sigma(T_3^{old}+273.15)^3 + c}, \tag{17.57}$$

$$c = \frac{k_s k_I}{k_s(h_I/2)+k_I h_s}, \tag{17.58}$$

[‡] Thus, T_3 is a diagnostic amount. However, it is treated as a prognostic state variable in the program, since the value of the previous time step, T_3^{old}, is used as the initial estimate for T_3^{new}.

[§] To be exact, the interface fluxes should be computed iteratively by adjusting surface temperature T_3^{new} until a balance is achieved, but this is not implemented.

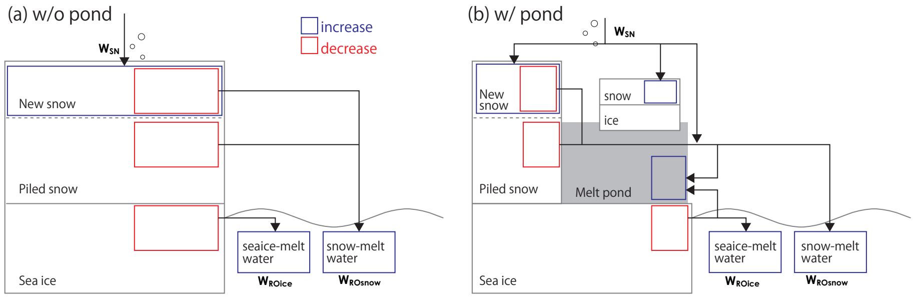

Figure17.5 Diagram of the modeled processes at the sea ice surface. Melt Pond is (a) not considered, and (b) considered.

where $Q_{AI}^{old}, \partial Q_{LI}/\partial T_3, \partial Q_{SI}/\partial T_3$ in the right hand side have been already obtained in Sec. 17.2.1, Eqs. (17.31), (17.33) and (17.41) . Since the solution procedures are basically the same with or without snow, a situation without snow ($h_s = 0, T_3 = T_2$) is included in the above equation. Note that the dependency of L_s on temperature (Section 14.12.2) is not considered in the partial differentiation with respect to temperature. The specific form for the partial derivative of specific humidity ($\partial q_i/\partial T_3$) is presented in Section 17.15.1.

Next, T_3^{new} is given by Eq. (17.52), though an upper limit is set depending on the presence or absence of snow, v_s, as

$$T_3^{new} = \begin{cases} 0 & \text{if} \quad v_s > 0 \quad \text{and} \quad T_3^{old} + \delta T_3 > 0 \\ T_{frz} & \text{if} \quad v_s = 0 \quad \text{and} \quad T_3^{old} + \delta T_3 > T_{frz} \\ T_3^{old} + \delta T_3 & \text{otherwise,} \end{cases} \tag{17.59}$$

where $T_{frz} = mS_I$ is the freezing point of sea ice. In the three cases, the top two conditions cause melting of sea ice or snow, which will be explained in the next section. Then, the interface temperature is computed as

$$T_2^{new} = \frac{k_s(h_I/2)T_3^{new} + k_I h_s T_1^{old}}{k_s(h_I/2) + k_I h_s}, \tag{17.60}$$

and $Q_{I2}(= Q_S)$ is given by Eq. (17.50).

Snowing and melting at the sea ice surface

Here, we solve the snowing and melting processes that occur on the sea ice surface, as shown in Fig. 17.5a (Melt pond is not considered). First, fresh snow is formed on the surface due to snowfall. Next, if the new surface temperature is above the freezing temperature (0.0 °C for snow and mS_I [°C] for sea ice), melting occurs as noted about Eq. (17.59). The whole procedure is as follows:

1. Fresh snow is formed by snowfall. (Δv_s^{new})
2. If surface melting occurs
   (a) All fresh snow melts. ($\Delta v_s^{new\_melt}$)
   (b) Piled snow and sea ice melt. ($\Delta v_s^{pile\_melt}, \Delta v_n$)
   (c) Melted water Inflows to the ocean. (W_{ROice}, W_{ROsnow})

The changes in sea ice and snow at each procedure are shown below.

The fresh snow volume, Δv_s^{new}, due to snowfall, W_{SN}, is given by the following equation:

$$\Delta v_s^{new} = \rho_o W_{SN} a_n \Delta t/\rho_s. \tag{17.61}$$

If T_3^{new} is below the freezing point, only this process occurs in this section.

Below this is the case where surface melting occurs. First, all the fresh snow melts, and the snow volume change due to this melting, $\Delta v_s^{new\_melt}$, is given by

$$\Delta v_s^{new\_melt} = -\Delta v_s^{new}, \tag{17.62}$$

where the negative sign means decrease.

Next, the amounts of melting of piled snow and sea ice are calculated as follows. The heat energy used for melting, ΔE, is given from the balance of heat flux by

$$\Delta E = (Q_{I2} - Q_{AI}^{old})a_n \Delta t, \tag{17.63}$$

where Q_{I2} is given by Eq.(17.50), and Q_{AI}^{old} is given by Eq. (17.41). On the other hand, the energy required to melt all the piled snow, E_s, and that to melt all the sea ice, E_I, are as follows:

$$E_s = v_s \rho_s L_F, \qquad E_I = v_n \rho_I L_3, \tag{17.64}$$

where L_3 is enthalpy change per unit mass due to melting of sea ice

$$L_3 = E_S(T_{frz}, S_I) - q_n = C_{p0} T_{frz} - q_n, \tag{17.65}$$

where q_n is sea ice enthalpy per unit mass. The piled snow is assumed to have zero heat capacity $E(0,0) = -L_F$. In addition, the lower limit, h_{min}, is set for the sea ice thickness, and the corresponding energy amount, E_{min}, is

$$E_{min} = a_n h_{min} \rho_I L_3. \tag{17.66}$$

From these energies, the snow volume change, $\Delta v_s^{pile\_melt}$, and the sea ice volume change, Δv_n, are calculated. When $\Delta E \leq E_s$, only snow is melted as

$$\Delta v_s^{pile\_melt} = -\Delta E/(\rho_s L_f), \qquad \Delta v_n = 0. \tag{17.67}$$

When $E_s < \Delta E \leq E_s + E_I - E_{min}$, snow and sea ice are melted

$$\Delta v_s^{pile\_melt} = -v_s, \qquad \Delta v_n = -(\Delta E - E_s)/(\rho_I L_3). \tag{17.68}$$

When $E_s + E_I - E_{min} < \Delta E$, all the piled snow and sea ice are melted away

$$\Delta v_s^{pile\_melt} = -v_s, \qquad \Delta v_n = -v_n, \tag{17.69}$$

and then excess heat flows into the ocean. That is, the sensible heat flux at the ice-ocean interface, Q_{SO}, is given by

$$Q_{SO} = Q_{AI}^{old} + (E_s + E_I)/(a_n \Delta t). \tag{17.70}$$

Regardless of the conditions, the sea ice enthalpy changes as the sea ice volume decreases, as in Eq. (17.47).

$$\Delta e_n = q_n \Delta v_n. \tag{17.71}$$

The freshwater flux between the ocean and sea ice models due to the melting of sea ice, W_{ROice}, is calculated as follows in terms of seawater volume:

$$W_{ROice} = \frac{\Delta v_n}{\Delta t} \frac{\rho_I}{\rho_o}. \tag{17.72}$$

The freshwater flux due to the melting of snow, W_{ROsnow}, is calculated as follows:

$$W_{ROsnow} = \frac{\Delta v_s^{new\_melt} + \Delta v_s^{pile\_melt}}{\Delta t} \frac{\rho_s}{\rho_o}. \tag{17.73}$$

Melted water is assumed to run off to the ocean immediately, that is, there is no explicit melt pond. However, effect of melt pond is implicitly parameterized by the albedo scheme. Assuming that seawater with a freezing temperature of T_{frz} pours into the ocean, the temperature transport between the ocean and sea ice models is given by

$$F_{top}^{\theta} = T_{frz} \sum_n \Delta v_n. \tag{17.74}$$

The temperature of the melted snow is assumed as 0° C.

Snowing and melting at the sea ice surface with melt pond (v5.1)

Modeling of melt pond is currently under development, some of which are explained here. When melt pond is considered, snowing and melting at the sea ice surface are processed as follows as a whole (Fig 17.5b):

1. Fresh snow is formed by snowfall (modified). (Δv_s^{new})
2. If surface melting occurs
   (a) All fresh snow melts. ($\Delta v_s^{new\_melt}$)
   (b) Piled snow and sea ice melt. ($\Delta v_s^{pile\_melt}, \Delta v_n$)
   (c) Melted water inflows to the ocean and melt pond (modified). (W_{ROice}, W_{ROsnow})

Snowfall causes increase of melt pond volume, as well as fresh snow formation on sea ice. Fresh snow volume, Δv_s^{new}, is

$$\Delta v_s^{new} = \frac{\rho_o W_{SN} \Delta t}{\rho_s}(a_n - a_p), \tag{17.75}$$

where a_p is the area fraction of ice pond. If there is ice on the pond, the volume of fresh snow on pond ice, Δv_{ps}^{new}, and the volume of snow that melts immediately after entering the pond, $\Delta v_s^{melt\_on\_pond}$, are

$$\Delta v_{ps}^{new} = \frac{\rho_o W_{SN} \Delta t}{\rho_s} a_p \tag{17.76}$$

$$\Delta v_s^{melt\_on\_pond} = 0. \tag{17.77}$$

If there is not ice, they are

$$\Delta v_{ps}^{new} = 0 \tag{17.78}$$

$$\Delta v_s^{melt\_on\_pond} = \frac{\rho_o W_{SN} \Delta t}{\rho_s} a_p. \tag{17.79}$$

17.2.3 Heat balance in the ice interior

Next, the time change of enthalpy inside the sea ice is solved from the heat fluxes obtained in the previous section. The thermal energy of the ice is affected by vertical heat fluxes and horizontal heat transport due to advection. The equation for the thermal energy (enthalpy) is written as follows:

$$\rho_I h_I \left[\frac{\partial}{\partial t} E(T_1, r_1) + u_{Ii} \frac{\partial}{\partial x_i} E(T_1, r_1) \right] = Q_{IO} - Q_{I2} + \Delta SW_{ice}, \tag{17.80}$$

where the heat flux Q_{IO} in the lower half of the ice layer is computed as

$$Q_{IO} = -\frac{k_I}{h_I/2}(T_1^{old} - T_0^{old}) = \frac{k_I}{h_I/2}(T_0^{old} - T_1^{old}). \tag{17.81}$$

The heat flux of Q_{I2} has been obtained by Eq. (17.50), and absorption of short waves inside sea ice, ΔSW_{ice}, by Eq. (17.22). When (17.80) is rewritten in term of volume-integrated values for each category,

$$\frac{\partial(\rho_I e_n)}{\partial t} + u_{Ii} \frac{\partial(\rho_I e_n)}{\partial x_i} = (Q_{IO} - Q_{I2} + \Delta SW_{ice}) a_n, \tag{17.82}$$

where integrated enthalpy, e_n, is one of the main prognostic variables of the model. The above equation can be solved explicitly without causing serious problems when the time step is not too long.

In the sea ice model, first, the vertical one-dimensional process of the above equation is solved by the forward finite difference. The enthalpy change due to the vertical heat flux, Δe^{est} [J m / kg], is estimated as follows.

$$\Delta e^{est} = (Q_{IO} - Q_{I2} + \Delta SW_{ice}) \frac{a_n \Delta t}{\rho_I}. \tag{17.83}$$

The upper limit of the sea ice enthalpy with a volume of v_n, e_{max} [J m / kg], depends on the freezing point $T_{frz} = mS_I$,

$$e_{max} = v_n C_{po} T_{frz}. \tag{17.84}$$

When $e_n + \Delta e^{est} \leq e_{max}$, only the enthalpy changes as

$$\Delta v_n = 0, \qquad \Delta v_s = 0, \qquad \Delta e_n = \Delta e^{est}. \tag{17.85}$$

When $e_n + \Delta e^{est} > e_{max}$, all sea ice and snow are melted,

$$\Delta v_n = -v_n, \qquad \Delta v_s = -v_s, \qquad \Delta e_n = -e_n. \tag{17.86}$$

(The integrated enthalpy at the next step is also set zero.) In this case, the heat flux across the sea ice bottom, Q_{IO}, is adjusted so that the enthalpy reaches the upper limit value at the end of the time step (Δt),

$$Q_{SO} = \frac{\rho_I (e_{max} - e^n)}{\Delta t} + (Q_{I2} - \Delta SW_{ice}) a_n, \tag{17.87}$$

Q_{SO} is used instead of $a_n Q_{IO}$ as the sensible heat flux at the ice-ocean interface (17.256).

According to the above decreases of v_n and v_s, fluxes between the sea ice and ocean are increased as follows.

$$\Delta W_{ROsnow} = \Delta v_s \rho_s / (\rho_o \Delta t) \tag{17.88}$$

$$\Delta W_{ROice} = \Delta v_n \rho_I / (\rho_o \Delta t) \tag{17.89}$$

$$\Delta Q_{LO} = \Delta v_s \rho_s L_F / \Delta t, \tag{17.90}$$

where ΔQ_{LO} means ocean heat used to melt snow. Assuming that seawater with a freezing temperature of T_{frz} has been formed, the temperature transport to the ocean is given by

$$F^{\theta}_{intr} = T_{frz} \sum_n \Delta v_n. \tag{17.91}$$

17.2.4 Formation of new sea ice

In addition to the vertical one-dimensional process of heat, sea ice is also formed by iceberg runoff and supercooling of seawater. This results in new ice formation in open water and open leads, which will be distributed to a category of sea ice (Sec. 17.3.2), as well as increase of sea ice in existing categories.

a. Input of iceberg

Input of iceberg (F_{ibrg}) is given as a sea-water volume flux per unit area (m^3 m^{-2} s^{-1}, negative values indicate input), and distributed in proportion to a_n. The increase of sea ice for existing category n is

$$\Delta v_n = -\frac{\rho_o}{\rho_I} F_{ibrg} a_n \Delta t, \qquad \Delta e_n = \Delta v_n q_n. \tag{17.92}$$

The new ice formation in open water and open leads is based on a_0 of (17.18),

$$\Delta v_0 = -\frac{\rho_o}{\rho_I} F_{ibrg} a_0 \Delta t, \qquad \Delta e_0 = \Delta v_0 E_S(T_{frz}, S_I^{ibrg}), \tag{17.93}$$

where $T_{frz} = m S_w$ is the freezing point of ocean (Sec. 14.7.2).

If the variable sea ice salinity option is used (v5.1), the change of salinity is

$$\Delta S_v = \Delta v_0 S_I^{ibrg} \quad \text{or} \quad \Delta v_n S_I^{ibrg}. \tag{17.94}$$

Considering that the salinity in the formed sea ice is supplied from the ocean, an upward salinity flux of F_{IO}^S [m psu s^{-1}] is given to the surface of the ocean model (that is, salinity reduction in ocean):

$$F_{IO}^S = \frac{\rho_I}{\Delta t \rho_o} \sum_{n=0}^{N_c} \Delta S_v(n). \tag{17.95}$$

If the salinity of the formed sea ice is assumed constant at S_I^{ibrg} (v5.0),

$$F_{IO}^S = -F_{ibrg} S_I^{ibrg}. \tag{17.96}$$

b. From sea water with temperature below the freezing point

Sea ice is formed when sea surface temperature is below the freezing point. In this model, this process is considered as follows. If the temperature of the first layer of the ocean model, T_w, is below the freezing point as a function of salinitym, T_{frz}, the temperature is set to the freezing point, and the heat needed to raise the temperature is regarded as the release of latent heat and is used to form new ice. Therefore, the thickness of the new ice (Δh_I) can be computed by assuming that the total thermal energy of the first layer of the ocean model (which has the layer thickness of $\Delta z_{\frac{1}{2}}$) is conserved before and after the sea ice formation:

$$\rho_o \Delta z_{\frac{1}{2}} E(T_w, 1) = \rho_o \Delta z_{\frac{1}{2}} [C_{po} T_w] = \rho_I \Delta h_I E_S(T_{\text{ice}}, S_I) + \rho_o (\Delta z_{\frac{1}{2}} - \rho_I \Delta h_I / \rho_o)(C_{po} T_{frz}) \tag{17.97}$$

where E is sea ice enthalpy per unit mass (17.7), and T_{ice} is the temperature of the newly formed sea ice. For T_{ice}, $T_1(n)$ is used if sea ice exists for category n, while T_{0_L} in open leads and T_{frz} in open water. (In v5.1, T_{frz} is used in both of open leads and open water.) Using the above equation, we compute the thickness of the new ice:

$$\Delta h_I = \frac{\rho_o \Delta z_{\frac{1}{2}}}{\rho_I} \frac{C_{po}(T_{frz} - T_w)}{(T_{frz}C_{po} - E_S(T_{\mathrm{ice}}, S_I))}. \tag{17.98}$$

(The values at the m time step are used for all the state variables.)

Based on the sea ice growth, increase of sea ice volume during one time step is evaluated for each of the open water and leads and the sea ice bottom. First, in open water and leads,

$$\Delta v_0 = \Delta h_I a_0, \qquad \Delta e_0 = \Delta v_0 E_S(T_{frz}, S_I^{frzl}), \tag{17.99}$$

where S_I^{frzl} is salinity of the newly formed sea ice (a constant value). Sea ice increase for the existing category-n sea ice is

$$\Delta v_n = \Delta h_I a_n, \qquad \Delta e_n = \Delta v_n q_n. \tag{17.100}$$

If the variable sea ice salinity option is used (v5.1), the change of salinity is

$$\Delta S_v = \Delta v_0 S_I^{frzl} \quad \text{or} \quad \Delta v_n S_I^{frzl}. \tag{17.101}$$

The salinity of the formed sea ice, S_I^{frzl}, is determined by the growth rate of the sea ice (Cox and Week, 1978).

The freshwater flux between the ocean and sea ice due to the sea ice formation is given by

$$W_{FR} = \frac{\rho_I}{\rho_o}\left(\Delta v_0 + \sum_{n=1}^{N_c} \Delta v_n\right)/\Delta t. \tag{17.102}$$

The water transported from the ocean to the sea ice is assumed to have the sea surface (first level) temperature, T_w, and then

$$F_{frzl}^{\theta} = T_w \sum_{n=0}^{N_c} \Delta v_n. \tag{17.103}$$

Note that sea surface temperature, T_w, and sea surface salinity, S_w, are rewritten in the process. (In v5.1, instead of modifying T_w and S_w, temperature and salinity fluxes to the ocean are used.)

Note that this operation practically eliminates super-cooling in the ocean interior. Hence, the formation of frazil ice is not considered in this model.

17.2.5 Ice-ocean interface

Melting and freezing at the ice-ocean interface is computed using heat fluxes at the interface as depicted in Figure 17.2. The solution method slightly differs from that of Mellor and Kantha (1989). In this section, we first list main relational equations between state variables and fluxes. Then, the bulk coefficients, skin layer salinity and temperature, and heat fluxes are derived, and, based on them, sea ice change and fluxes to the ocean are evaluated. The ice-covered area and the open leads are treated separately, while this process does not work in open water.

Basic relations

In the ice-covered area, the heat flux on the ice side of the ice-ocean interface (Q_{IO}) has been already computed according to (17.81). In open leads, the heat flux on the air side of the air-ocean interface (Q_{AO}) has also been computed as in Chapter 14,

$$Q_{AO}^{all} = Q_{SO}(T_w) + Q_{LO}(T_w) - (1-\alpha_o)SW - LW + \epsilon_o\sigma(T_w + 273.15)^4. \tag{17.104}$$

All the heat and fresh water fluxes are evaluated using the temperature and salinity at the first level of the ocean model (T_w and S_w). By doing so, the equation to compute melting and freezing rates becomes linear. Here, shortwave radiation is assumed to pass through the skin layer without absorption and is excluded from the evaluation of the freezing rate in open water:

$$Q_{AO} = Q_{SO} + Q_{LO} - LW + \epsilon_o\sigma(T_w + 273.15)^4. \tag{17.105}$$

This operation causes the shortwave radiation to be absorbed in the ocean interior. In reality, the heat stored in the skin layer in open water is used to melt ice laterally (edge melting), and its details are described in a later part of this section.

Melting and freezing represented by W_{IO} and W_{AO} occur owing to the imbalance between fluxes above (Q_{IO}, Q_{AO}) and below (F_{T_I}, F_{T_L}) the interface:

$$F_{T_I} = Q_{IO} - W_{IO}\rho_o L_{o_I}(n), \tag{17.106}$$

$$F_{T_L} = Q_{AO} - W_{AO}\rho_o L_{o_L}, \tag{17.107}$$

where $L_{o_I}(n)$ and L_{o_L} are amounts of energy [J / kg] required to form unit mass of sea ice,

$$L_{o_I}(n) \equiv [E(T_w, 1) - q_n(n)], \tag{17.108}$$

$$L_{o_L} \equiv [E(T_w, 1) - E_S(\widetilde{T}_{0_L}, S_I^{frzl})], \tag{17.109}$$

where S_I^{frzl} is salinity of sea ice formed in open leads, and $\widetilde{T}_{0_L}$ is the ice skin temperature calculated for open leads at the previous time step. (In v5.1, the ocean freezing temperature, $T_{\text{frz}} = mS_w$, is used instead of the somewhat virtual value $\widetilde{T}_{0_L}$.)

Following the formulation adopted by Mellor and Kantha (1989), we introduce skin layer temperature and salinity (T_{0_I}, T_{0_L}, S_{0_I}, and S_{0_L}) to solve for W_{IO} and W_{AO}, and thus F_{T_I} and F_{T_L}. To incorporate skin layer salinity in the system of equations, we consider the problem in terms of virtual salt flux, in which the effect of freshwater flux on salinity is considered in terms of salt flux by keeping the water volume. The flux balance for salt below the interface is written as follows:

$$F_{S_I} = W_{IO}(S_I - S_w), \tag{17.110}$$

$$F_{S_L} = W_{AO}(S_{\text{frzl}} - S_w). \tag{17.111}$$

Here, unlike Mellor and Kantha (1989), S_w is used instead of the salinity at the skin layer (S_{0_I}, S_{0_L}). By doing so, the equations to solve for S_{0_I} and S_{0_L} become linear as shown below. It could also be said that it is natural to use the first level salinity itself in evaluating the salt flux that drives the first level of the ocean model. Note that only fresh water fluxes that are relevant to freezing and melting at the ice-ocean interface are included in the above equations. The restoration to climatological salinity and fresh water fluxes caused by surface melting, precipitation, and evaporation are excluded in the above balance.

Fluxes on the oceanic side of the interface (F_{T_I}, F_{T_L}, F_{S_I}, and F_{S_L}) can also be obtained as the boundary conditions ($z \longrightarrow 0$) for the molecular boundary layer:

$$F_{T_I}/(\rho_o C_{po}) = -C_{T_z}(T_{0_I} - T_w), \tag{17.112}$$

$$F_{T_L}/(\rho_o C_{po}) = -C_{T_z}(T_{0_L} - T_w), \tag{17.113}$$

and

$$F_{S_I} = -C_{S_z}(S_{0_I} - S_w), \tag{17.114}$$

$$F_{S_L} = -C_{S_z}(S_{0_L} - S_w). \tag{17.115}$$

The coefficients, C_{T_z} and C_{S_z}, will be explained in detail in the next section.

The above equations are solved simultaneously to obtain melting and freezing at the upper surface of the ocean under the following constraints:

$$T_{0_I} = mS_{0_I} \quad \text{and} \quad T_{0_L} = mS_{0_L}, \tag{17.116}$$

where m defines the freezing line as a function of salinity.

Bulk coefficients in the ice bottom boundary layer

Before solving the above system of equations, the bulk coefficients, C_{T_z} and C_{S_z}, must be determined. The temperature coefficient, C_{T_z}, is

$$C_{T_z} = \frac{u_\tau}{(P_{rt}k^{-1}\ln(-z/z_0) + B_T)}, \tag{17.117}$$

$u_\tau \equiv (\tau_{IO_x}^2 + \tau_{IO_y}^2)^{1/4}\rho_o^{-1/2}$ is the friction velocity, $k = 0.4$ is von Karman's constant, z_0 is the roughness parameter, $(\tau_{IO_x}, \tau_{IO_y})$ is the stress vector at the ocean-ice interface, and

$$B_T = b\left(\frac{z_0 u_\tau}{\nu}\right)^{1/2} Pr^{2/3}, \tag{17.118}$$

with $Pr \equiv \nu/\alpha_t = 12.9$. The specific values for other parameters are given in Section 17.1.6.

Parameters related to salinity are given by

$$C_{S_z} = \frac{u_\tau}{(P_{rt}k^{-1}\ln(-z/z_0) + B_S)}, \tag{17.119}$$

and

$$B_S = b\left(\frac{z_0 u_\tau}{\nu}\right)^{1/2} Sc^{2/3}, \tag{17.120}$$

where $Sc \equiv \nu/\alpha_b = 2432$.

Following Mellor and Kantha (1989), roughness parameter z_0 is computed as follows:

$$\ln z_0 = A \ln z_{0_I} + (1 - A)\ln z_{0_L}, \tag{17.121}$$

where

$$z_{0_I} = 0.05\frac{h_I}{h_{I\mathrm{lim}}}, \quad h_{I\mathrm{lim}} = 3.0\,\mathrm{m}, \tag{17.122}$$

and

$$z_{0_L} = 0.016\,\frac{\rho_o}{\rho_a}\,\frac{u_\tau^2}{g}. \tag{17.123}$$

However, the roughness parameter below ice (17.122) is also used for open water (z_{0_L}) in MRI.COM.

Ice-bottom boundary flux based on skin salinity and temperature

Using the obtained bulk coefficients, the five unknown variables, $W_{IO}, F_{T_I}, F_{S_I}, T_{0_I}$ and S_{0_I} are solved under the five equations, (17.106), (17.110), (17.112), (17.114) and (17.116), for the ice-covered area. Also for open leads, the equation system is solved in the same way. Specifically, we first solve the skin layer salinity S_{0_I} (ice-covered area) and S_{0_L} (open leads) using (17.106), (17.110), (17.112), (17.114), (17.107), (17.111), (17.113), and (17.115):

$$S_{0_I} = \frac{C_{S_z}S_w + (\rho_o C_{po} C_{T_z} T_w - Q_{IO})(S_I - S_w)/\rho_o L_o}{C_{S_z} + \rho_0 C_{po} C_{T_z} m(S_I - S_w)/\rho_o L_o}, \tag{17.124}$$

$$S_{0_L} = \frac{C_{S_z}S_w + (\rho_o C_{po} C_{T_z} T_w - Q_{AO})(S_{\mathrm{frzl}} - S)/\rho_o L_o}{C_{S_z} + \rho_o C_{po} C_{T_z} m(S_{\mathrm{frzl}} - S_w)/\rho_o L_o}. \tag{17.125}$$

However, the upper and lower limits are set as $S_I \le S_{0_I} \le S_{\max}$ and $S_{\mathrm{frzl}} \le S_{0_L} \le S_{\max}$, in order to avoid deviation from the domain due to rounding error etc. (In v5.1, since S_{0_L} is used only here, the upper and lower limits are not set according to the original paper.) Using S_{0_I} and S_{0_L}, T_{0_I} and T_{0_L} are computed from (17.116), and then, F_{T_I} and F_{T_L} are computed from (17.112) and (17.113).

Here, the meaning of the skin salinity, S_{0_I} or S_{0_L}, is explained in detail. In this scheme, the salinity flux between ocean and sea ice is treated as a "virtual salinity flux". In other words, decrease in ocean salinity due to sea ice melting is treated as an upward salinity outflow across the sea surface boundary for the ocean model, and increase in ocean salinity due to freezing (brine discharge) is treated as a downward inflow. To represent them in (17.114) and (17.115), the skin salinity, S_{0_I} or S_{0_L}, are virtually set. (Therefore, impossible values such as negative values are acceptable.) That is, the skin salinity is a "virtual boundary condition" for obtaining the salinity flux in the ocean model, and has no substance. Similarly, the skin temperature is also a virtual boundary value.

Sea ice change

In the ice-covered area, the difference between the heat flux inside the sea ice, Q_{IO}, and that in the ice-bottom boundary layer, F_{T_I}, is the amount of enthalpy used to melt sea ice or freeze sea water, Δe.

$$\Delta e = \frac{1}{\rho_I}(F_{T_I} - Q_{IO})a_n\Delta t. \tag{17.126}$$

Meanwhile, the amount of enthalpy required to melt all sea ice and transform it into sea water with a temperature of T_w, Δe_{max}, is given by

$$\Delta e_{max} = v_n(E(T_w, 1) - q_n(n)) = C_{po}T_w v_n - e_n, \tag{17.127}$$

where subscripts n designate the thickness category. Using these two amounts, the sea ice change during one time step such as $\Delta v_n, \Delta v_s, \Delta e_n$ and Δa_n can be determined.

If $\Delta e < 0$, sea water freezes, and

$$\Delta v_n = -\frac{\Delta e}{L_{O_I}} = -\frac{\Delta e}{C_{po}T_w - q_n(n)}, \qquad \Delta v_s = 0, \qquad \Delta e_n = \Delta v_n q_n(n) = \Delta e \frac{q_n(n)}{q_n(n) - C_{po}T_w}, \qquad \Delta a_n = 0, \quad (17.128)$$

If $0 < \Delta e < \Delta e_{max}$, part of sea ice melts. Assuming that it melts into sea water with a temperature of T_w, Δv_n and Δe_n are

$$\Delta v_n = -\frac{\Delta e}{C_{po}T_w - q_n(n)}, \qquad \Delta e_n = \Delta v_n q_n(n) = \Delta e \frac{q_n(n)}{q_n(n) - C_{po}T_w}. \quad (17.129)$$

In this case, reduction of the ice fraction is allowed by "edge melting" as follows. When decrease of sea ice thickness is written as Δh, a part of it, $\Psi\Delta h$, is assumed to be a loss caused by decrease in ice fraction, Δa_n. When the original sea ice thickness is written as h, the following relationship holds

$$\Psi\Delta h a_n = (h - (1 - \Psi)\Delta h)\Delta a_n. \quad (17.130)$$

This becomes

$$c = \Psi \frac{\Delta v_n}{v_n - (1 - \Psi)\Delta v_n} \quad (17.131)$$

$$\Delta a_n = c a_n. \quad (17.132)$$

As snow on the reduced ice fraction melts,

$$\Delta v_s = \frac{\Delta a_n}{a_n} v_s = c v_s. \quad (17.133)$$

There seems to be no widely accepted parameterization scheme for edge melting. According to Steele (1992), bottom and top melting are dominant processes, and thus $\Psi \sim 0.0 - 0.1$ is usually used in MRI.COM.

Finally, if $\Delta e > \Delta e_{max}$, all sea ice and snow melt:

$$\Delta v_n = -v_n, \qquad \Delta v_s = -v_s, \qquad \Delta e_n = -e_n, \qquad \Delta a_n = -a_n. \quad (17.134)$$

In this case, the remaining heat of Δe is added to the heat flux to the ocean. Thus, $a_n F_{T_I}$ is re-evaluated and given to the ocean.

$$a_n F_{T_I} = a_n Q_{IO} + \frac{\Delta e_{max} \rho_I}{\Delta t}. \quad (17.135)$$

In open leads, the amount of enthalpy used for freezing on the sea surface, Δe, is

$$\Delta e = \frac{1}{\rho_I}(F_{T_L} - Q_{AO}) a_0 \Delta t. \quad (17.136)$$

If $\Delta e < 0$, frazil ice is formed in open leads. The volume of the newly-created ice, $\Delta v_{\text{frazil\_air}}$, and its enthalpy, $\Delta e_{\text{frazil\_air}}$, are

$$\Delta v_{\text{frazil\_air}} = -\frac{\Delta e}{L_{O_L}}, \qquad \Delta e_{\text{frazil\_air}} = \Delta v_{\text{frazil\_air}} E_S(T_{frz}, S_I^{frzl}) = \Delta e \frac{E_S(T_{frz}, S_I^{frzl})}{E_S(T_{frz}, S_I^{frzl}) - C_{po}T_w}, \quad (17.137)$$

where L_{O_L} is given by (17.109), assuming that sea ice with a temperature of T_{0_L} (T_{frz} in v5.1) and with a salinity of S_I^{frzl} is formed. Note that $\Delta v_{\text{frazil\_air}}$ is in the range of $[0., H_{N_c}]$, and that $\Delta e_{\text{frazil\_air}}$ is negative. If $\Delta e \geq 0$, frazil ice is not formed, and the heat flux at the top of the sea surface is used for the heat flux to the ocean instead of (17.113),

$$F_{T_L} = Q_{AO}. \quad (17.138)$$

Fluxes to ocean

At the end of this chapter, we summarize the water and heat fluxes to the ocean due to the sea ice changes. The water flux associated with the freezing and melting, W_O, is

$$W_O = a_0 W_{AO} + \sum_n a_n W_{IO} = \frac{\rho_I}{\rho_o \Delta t}\left(\Delta v_{\text{frazil\_air}} + \sum_n \Delta v_n\right), \quad (17.139)$$

using the volume of seawater as a unit. The freshwater flux due to melting of piled snow is

$$W_{ROsnow} = \frac{\rho_s}{\rho_o \Delta t}\left(\sum_n \Delta v_s\right). \tag{17.140}$$

where W_{ROsnow} is added to (17.73) calculated in Sec. 17.2.1. For the temperature transport (positive upward), $F^{\theta}_{btm\_frz}$ associated with freezing and $F^{\theta}_{btm\_melt}$ associated with thawing are calculated separately, though they will be added up later. Assuming seawater with a temperature of SST, T_w, for increase and decrease, (Even in the case of melting, sea ice is transformed into seawater with a temperature of T_w as in (17.129))

$$F^{\theta}_{btm\_frz} = T_w \sum_{n=0} \Delta v_n \qquad (\Delta v_n > 0) \tag{17.141}$$

$$F^{\theta}_{btm\_melt} = T_w \sum_{n=0} \Delta v_n \qquad (\Delta v_n < 0) \tag{17.142}$$

The heat flux that drives the first level of the ocean model, F_T, is given by combining those in open leads and ice-covered areas:

$$F_T = a_0 F_{T_L} + \sum_n (a_n F_{T_I}). \tag{17.143}$$

Since ocean's heat is used to melt snow, the (upward) latent heat flux from the ocean, Q_{LO}, is

$$Q_{LO} = -\frac{\rho_s L_F}{\Delta t}\sum_n (\Delta v_s). \tag{17.144}$$

17.3 Remapping in thickness space

17.3.1 Re-categorization

After the thermodynamic processes are solved, the resultant ice thickness in some thickness categories might not be within the specified bound. Following the method adopted by CICE (Sec.3.2), we assume that there is a thickness distribution function in each category and use it to redistribute the new thickness distribution into original categories.

This procedure corresponds to the first term on the r.h.s. of (17.1):

$$\frac{\partial g}{\partial t} = -\frac{\partial}{\partial h}(fg). \tag{17.145}$$

In practice, a thickness category is regarded as a Lagrange particle, and the category boundaries are displaced as a result of thermodynamics. A linear thickness distribution function is assumed within each displaced category, and ice is remapped into the original categories using these functions.

First, boundaries of thickness categories are displaced. If the ice thickness in category n ($h_{I,n}$) changes from $h^m_{I,n}$ to $h^{m+1}_{I,n}$ (m is the time step index), the growth rate (f_n) at thickness $h^m_{I,n}$ is represented as

$$f_n = (h^{m+1}_{I,n} - h^m_{I,n})/\Delta t \qquad \text{for} \quad n = 1, 2, \cdots, n_c \tag{17.146}$$

$$f_0 = (\Delta v_{\text{frazil\_sea}} + \Delta v_{\text{frazil\_air}})/(a_0 \Delta t). \tag{17.147}$$

($f_0 \geq 0$ by definition) Using this, the growth rate (F_n) at the upper category boundary H_n is obtained by linear interpolation:

$$F_n = f_n + \frac{H_n - h^m_{I,n}}{h^m_{I,n+1} - h^m_{I,n}}(f_{n+1} - f_n) \qquad \text{for } n = 1, 2, \cdots, n_c - 1. \tag{17.148}$$

Note that, if the fractional area is zero in either category n or $n+1$, F_n is set to the growth rate at the non-zero category, f_n or f_{n+1}. When the fractional area is zero on both categories, $F_n = 0$. The lowest boundary, F_0, is given by

$$F_0 = \begin{cases} f_0 & \text{if } f_0 > 0 \\ f_1 & \text{if } f_0 = 0 \text{ and } f_1 < 0 \\ 0 & \text{if } f_0 = 0 \text{ and } f_1 \geq 0. \end{cases} \tag{17.149}$$

The new category boundary after thermodynamics is obtained as

$$H_n^* = H_n + F_n \Delta t \qquad \text{for } n = 1, 2, \cdots, n_c - 1, \tag{17.150}$$

though H_n^* is assumed to be in the range from $h_{I,n}^{m+1}$ to $h_{I,n+1}^{m+1}$. The boundary for the maximum category n_c, $H_{n_c}^*$, is set by linear extrapolation as

$$H_{n_c}^* = h_{I,n_c}^{m+1} + 2(h_{I,n_c}^{m+1} - H_{n_c-1}) = 3h_{I,n_c}^{m+1} - 2H_{n_c-1}. \tag{17.151}$$

The lowest boundary is

$$H_0^* = \begin{cases} H_0 + F_0 \Delta t & \text{if } F_0 > 0 \text{ i.e. } f_0 > 0 \\ 0 & \text{if } F_0 \leq 0 \text{ i.e. } f_0 = 0. \end{cases} \tag{17.152}$$

Next, the thickness distribution function g within the displaced category $[H_{n-1}^*, H_n^*]$ is determined. First, tentatively, the lower boundary height at the $m+1$ step, H_L, is given by H_{n-1}^*, while the higher boundary height, H_R, is given by H_n^*. Function $g(h)$ should satisfy the following definitions for fractional area (a_n, 17.2) and volume (v_n, 17.3):

$$\int_{H_L}^{H_R} g(h) dh = a_n, \tag{17.153}$$

$$\int_{H_L}^{H_R} h g(h) dh = v_n, \tag{17.154}$$

where h is a continuous variable as explained in Sec. 17.1.2, and a_n and v_n in the right hand side are prognostic variables at the $m+1$ time step.

We adopt a linear function of thickness for g. The thickness space is transformed to $\eta = h - H_L$, and the thickness distribution function is written as $g = g_1 \eta + g_0$. These are substituted into (17.153) and (17.154) to yield

$$g_1 \frac{\eta_R^2}{2} + g_0 \eta_R = a_n, \tag{17.155}$$

$$g_1 \frac{\eta_R^3}{3} + g_0 \frac{\eta_R^2}{2} = a_n \eta_n, \tag{17.156}$$

where $\eta_R = H_R - H_L$ and $\eta_n = h_{I,n}^{m+1} - H_L$, and $v_n = h_{I,n} a_n$ is used. These are algebraically solved for g_0 and g_1 as

$$g_0 = \frac{6a_n}{\eta_R^2} \left(\frac{2\eta_R}{3} - \eta_n \right) = \frac{a_n}{\eta_R^2} \Big(4\eta_R - 6\eta_n \Big), \tag{17.157}$$

$$g_1 = \frac{12a_n}{\eta_R^3} \left(\eta_n - \frac{\eta_R}{2} \right) = \frac{a_n}{\eta_R^3} \Big(12\eta_n - 6\eta_R \Big). \tag{17.158}$$

The distribution of g at the $m+1$ time step can be determined by the above equation, but the following modifications are required due to the constraint of $g \geq 0$. The values of the thickness distribution function at category boundaries are given as follows:

$$g(\eta = 0) = \frac{6a_n}{\eta_R^2} \left(\frac{2\eta_R}{3} - \eta_n \right), \tag{17.159}$$

$$g(\eta = \eta_R) = \frac{6a_n}{\eta_R^2} \left(\eta_n - \frac{\eta_R}{3} \right). \tag{17.160}$$

Equation (17.159) gives $g(0) < 0$ when the thickness is in the right third of the thickness range or $\eta_n > 2\eta_R/3$. Equation (17.160) gives $g(\eta_R) < 0$ when the thickness is in the left third of the thickness range or $\eta_n < \eta_R/3$. Since a negative g is physically impossible, we redefine the range of the thickness distribution function. Specifically, H_R and H_L are modified as

$$\begin{cases} H_L = H_{n-1}^*, \quad H_R = 3h_{I,n}^{m+1} - 2H_{n-1}^* & \text{if } h_{I,n}^{m+1} < H_{n-1}^* + \eta_R/3 \\ H_L = H_{n-1}^*, \quad H_R = H_n^* & \text{if } H_{n-1}^* + \eta_R/3 \leq h_{I,n}^{m+1} \leq H_{n-1}^* + 2\eta_R/3 \\ H_L = 3h_{I,n}^{m+1} - 2H_n^*, \quad H_R = H_n^* & \text{if } H_{n-1}^* + 2\eta_R/3 < h_{I,n}^{m+1}. \end{cases} \tag{17.161}$$

The values of η_R and η_n are calculated based on these H_L and H_R, then the distribution of $g(h)$ can be obtained through (17.157) and (17.158). The conditions of $H_{n-1}^* \leq H_L$ and $H_R \leq H_n^*$ are satisfied.

Finally, we remap ice into the original categories using the above thickness distribution function. If $H_R > H_n$, ice is transferred from category n to $n+1$. The transferred area $\Delta a_{n,+}$ and volume $\Delta v_{n,+}$ are

$$\begin{aligned} H'_n =& \max(H_L, H_n) \\ \Delta a_{n,+} =& \int_{H'_n}^{H_R} g(h)dh = \int_{H'_n}^{H_R} (g_1 h - g_1 H_L + g_0)dh \\ =& (H_R - H'_n)\left\{\frac{g_1}{2}\left(H_R + H'_n - 2H_L\right) + g_0\right\} \end{aligned} \tag{17.162}$$

and

$$\begin{aligned} \Delta v_{n,+} =& \int_{H'_n}^{H_R} hg(h)dh = \int_{H'_n}^{H_R} h(g_1 h - g_1 H_L + g_0)dh \\ =& \left(H_R - H'_n\right)\left\{\frac{g_1}{6}\left(2H_R^2 + 2H_R H'_n + 2{H'_n}^2 - 3H_R H_L - 3H'_n H_L\right) + \frac{g_0}{2}\left(H_R + H'_n\right)\right\}. \end{aligned} \tag{17.163}$$

If $H_L < H_{n-1}$, ice is transferred from category n to $n-1$. The transferred area $\Delta a_{n,-}$ and volume $\Delta v_{n,-}$ are

$$\begin{aligned} H'_{n-1} =& \min(H_R, H_{n-1}) \\ \Delta a_{n,-} =& \int_{H_L}^{H'_{n-1}} g(h)dh = \int_{H_L}^{H'_{n-1}} (g_1 h - g_1 H_L + g_0)dh \\ =& \left(H'_{n-1} - H_L\right)\left\{\frac{g_1}{2}\left(H'_{n-1} - H_L\right) + g_0\right\} \end{aligned} \tag{17.164}$$

and

$$\begin{aligned} \Delta v_{n,-} =& \int_{H_L}^{H'_{n-1}} hg(h)dh = \int_{H_L}^{H'_{n-1}} h(g_1 h - g_1 H_L + g_0)dh \\ =& \left(H'_{n-1} - H_L\right)\left\{\frac{g_1}{6}\left(2{H'_{n-1}}^2 - H'_{n-1}H_L - H_L^2\right) + \frac{g_0}{2}\left(H'_{n-1} + H_L\right)\right\}. \end{aligned} \tag{17.165}$$

However, no redistribution among categories occurs unless Δa_n and Δv_n exceed thresholds, whether frozen or thawed.

Snow and thermal energy are also transferred in proportion to the transferred volume. For example, $\Delta v_{sn} = v_{sn}(\Delta v_{in}/v_{in})$ for snow and $\Delta e_{in} = e_{in}(\Delta v_{in}/v_{in})$ for thermal energy.

As a special situation, consider the case where category 1 sea ice is melted ($f_1 < 0$) and ice is not created in open water ($f_0 = 0$). Then, the lower boundary of category 1, H_0^*, remains zero from (17.152), while its growth rate, F_0, is f_1 from (17.149). This means that the fractional area of category 1 thinner than $\Delta h_0 = -F_0 \Delta t = -f_1 \Delta t$ is added to open water area. In this operation, volume and energy are invariant. The area to be added to open water is [¶]

$$\begin{aligned} \Delta a_0 =& \int_0^{\Delta h_0} g(h)dh = \int_0^{\Delta h_0} (g_1 h - g_1 H_L + g_0)dh \\ =& \Delta h_0 \left\{g_1\left(\frac{\Delta h_0}{2} - H_L\right) + g_0\right\}. \end{aligned} \tag{17.166}$$

The right boundary of the thickest category n_c (H_{n_c}) is a function of its mean thickness h_{n_c}. When h_{n_c} is given, H_{n_c} is computed as $H_{n_c} = 3h_{n_c} - 2H_{n_c-1}$. It is guaranteed that $g(h) > 0$ for $H_{n_c-1} < h < H_{n_c}$ and $g(h) = 0$ for $H_{n_c} < h$.

17.3.2 Categorization of sea ice newly formed in open water/lead

If the sum of $\Delta v_{\text{frazil\_sea}}$ (Δv_0 in (17.93) and (17.99)) and $\Delta v_{\text{frazil\_air}}$ in (17.137), Δv, is bigger than zero, that is, if ice is created in open water and leads, the new sea ice is distributed to the category to which the average height h_{new} applies.

$$\Delta v = \Delta v_{\text{frazil\_sea}} + \Delta v_{\text{frazil\_air}} \tag{17.167}$$

$$h_{\text{new}} = \Delta v / a_0. \tag{17.168}$$

The sea ice volume Δv is added to v_n, and the following Δa is moved from a_0 to a_n.

$$\Delta a = \min(a_0, \Delta v / h_{\min}), \tag{17.169}$$

[¶] Strictly speaking, it may be better to integrate $g(h)$ in the range of $[H_L, \Delta h_0]$, instead of $[0, \Delta h_0]$.

where $h_{\min} = 0.1[\mathrm{m}]$ is the minimum thickness of the newly-formed sea ice, and the value at the previous time is used for a_0. The thermal energy of $\Delta e_{\mathrm{frazil\_sea}}$ (Δe_0 in (17.93) and (17.99)) and $\Delta e_{\mathrm{frazil\_air}}$ in (17.137) is also added to the corresponding category,

$$\Delta e = \Delta e_{\mathrm{frazil\_sea}} + \Delta e_{\mathrm{frazil\_air}}. \tag{17.170}$$

On the other hand, the snow volume in each category does not change. ‖

17.4 Dynamics

17.4.1 Momentum equation for ice pack

The momentum equation for an ice pack with mass $\rho_I A h_I$ is

$$\rho_I \frac{\partial}{\partial t}(A h_I u_I) - \rho_I A h_I f v_I = -\rho_I A h_I g \frac{1}{h_\mu} \frac{\partial h}{\partial \mu} + F_\mu(\sigma) + A(\tau_{AI_x} + \tau_{IO_x}), \tag{17.171}$$

$$\rho_I \frac{\partial}{\partial t}(A h_I v_I) + \rho_I A h_I f u_I = -\rho_I A h_I g \frac{1}{h_\psi} \frac{\partial h}{\partial \psi} + F_\psi(\sigma) + A(\tau_{AI_y} + \tau_{IO_y}), \tag{17.172}$$

where (u_I, v_I) is the velocity vector, h is the sea surface height, (F_μ, F_ψ) is the ice's internal stress (which is a function of internal stress tensor (σ)), and $\overrightarrow{\tau_{AI}}$ and $\overrightarrow{\tau_{IO}}$ are stresses exerted by the atmosphere and ocean.

17.4.2 Stresses at top and bottom

The stress at the top is wind stress:

$$\overrightarrow{\tau_{AI}} = C_{DAI} \rho_a |\mathbf{U}_a - \mathbf{u}_I| [(\mathbf{U}_a - \mathbf{u}_I) \cos\theta_a + \mathbf{k} \times (\mathbf{U}_a - \mathbf{u}_I) \sin\theta_a], \tag{17.173}$$

where $\mathbf{U}_a$ is the surface wind vector, C_{DAI} is the bulk transfer coefficient between air and ice, ρ_a is the density of air, and θ_a is the angle between the wind vector and the ice drift vector, which is set to zero in MRI.COM, since 10-m wind is generally similar in direction to the surface stress (Lepparanta, 2011).

Stress at the bottom is ocean stress:

$$\overrightarrow{\tau_{IO}} = C_{DIO} \rho_o |\mathbf{U}_w - \mathbf{u}_I| [(\mathbf{U}_w - \mathbf{u}_I) \cos\theta_o + \mathbf{k} \times (\mathbf{U}_w - \mathbf{u}_I) \sin\theta_o], \tag{17.174}$$

where $\mathbf{U}_w$ is the velocity of the first level of the ocean model, C_{DIO} is the bulk transfer coefficient between the ice and ocean, ρ_o is the density of sea water, and θ_o is the angle between the ice drift vector and the surface velocity of the ocean. The angle θ_o is set to cz in the northern and $-cz$ in the southern hemisphere, where z is the depth of the first layer (velocity point) and c is a constant of $1^\circ\ \mathrm{m}^{-1}$.

17.4.3 Internal stress

In a highly concentrated icepack, the effect of the internal stress is as large as the Coriolis effect and the surface stresses. The expression of the internal stress is derived by regarding the ice as continuous media. The elastic-plastic-viscous (EVP) model by Hunke and Dukowicz (1997, 2002) is adopted for the constitutive law (the relation between stress and strain rate). The EVP model is a computationally efficient modification of the viscous-plastic (VP) model (Hibler, 1979). In the VP model, the internal stress could be very large when the concentration is high and strain rate is near zero, which makes the explicit integration infeasible. An alternative, the implicit method, is usually adopted, but it is not suitable for parallel computing. The EVP model treats the ice as an elastic medium and a large local force is released by elastic waves, which would be damped within the time scale of the wind forcing.

The constitutive law of the EVP model is

$$\frac{1}{E} \frac{\partial \sigma_{ij}}{\partial t} + \frac{1}{2\eta} \sigma_{ij} + \frac{\eta - \zeta}{4\eta\zeta} \sigma_{kk} \delta_{ij} + \frac{P}{4\zeta} \delta_{ij} = \dot{\epsilon}_{ij}, \quad i, j = 1, 2, \tag{17.175}$$

where ζ and η are viscous parameters, P represents ice strength, and E is an elastic parameter (mimics Young's modulus). In the VP model, tendency terms are zero.

‖ The average snow depth can change due to change in area.

The r.h.s. ($\dot{\epsilon}_{ij}$) is the strain rate tensor, expressed in Cartesian coordinates as:

$$\dot{\epsilon}_{ij} = \frac{1}{2}\Big(\frac{\partial u_{Ii}}{\partial x_j} + \frac{\partial u_{Ij}}{\partial x_i}\Big). \tag{17.176}$$

The divergence, tension, and shear of the strain rate are defined as follows:

$$D_D = \dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \;\; D_T = \dot{\epsilon}_{11} - \dot{\epsilon}_{22}, \;\; D_S = 2\dot{\epsilon}_{12}. \tag{17.177}$$

The equation for the stress tensor for $\sigma_1 = \sigma_{11} + \sigma_{22}$ and $\sigma_2 = \sigma_{11} - \sigma_{22}$ is given by

$$\frac{1}{E}\frac{\partial \sigma_1}{\partial t} + \frac{\sigma_1}{2\zeta} + \frac{P}{2\zeta} = D_D, \tag{17.178}$$

$$\frac{1}{E}\frac{\partial \sigma_2}{\partial t} + \frac{\sigma_2}{2\eta} = D_T, \tag{17.179}$$

$$\frac{1}{E}\frac{\partial \sigma_{12}}{\partial t} + \frac{\sigma_{12}}{2\eta} = \frac{1}{2}D_S. \tag{17.180}$$

In generalized orthogonal coordinates, divergence, tension, and shear of the strain rate are expressed by

$$D_D = \frac{1}{h_\mu h_\psi}\Big[\frac{\partial(h_\psi u_I)}{\partial \mu} + \frac{\partial(h_\mu v_I)}{\partial \psi}\Big], \tag{17.181}$$

$$D_T = \frac{h_\psi}{h_\mu}\frac{\partial}{\partial \mu}\Big(\frac{u_I}{h_\psi}\Big) - \frac{h_\mu}{h_\psi}\frac{\partial}{\partial \psi}\Big(\frac{v_I}{h_\mu}\Big), \tag{17.182}$$

$$D_S = \frac{h_\mu}{h_\psi}\frac{\partial}{\partial \psi}\Big(\frac{u_I}{h_\mu}\Big) + \frac{h_\psi}{h_\mu}\frac{\partial}{\partial \mu}\Big(\frac{v_I}{h_\psi}\Big). \tag{17.183}$$

The internal stress is obtained as the divergence of the internal stress tensor,

$$F_\mu = \frac{1}{2}\Big[\frac{1}{h_\mu}\frac{\partial \sigma_1}{\partial \mu} + \frac{1}{h_\mu h_\psi^2}\frac{\partial(h_\psi^2 \sigma_2)}{\partial \mu} + \frac{2}{h_\mu^2 h_\psi}\frac{\partial}{\partial \psi}(h_\mu^2 \sigma_{12})\Big], \tag{17.184}$$

$$F_\psi = \frac{1}{2}\Big[\frac{1}{h_\psi}\frac{\partial \sigma_1}{\partial \psi} - \frac{1}{h_\mu^2 h_\psi}\frac{\partial(h_\mu^2 \sigma_2)}{\partial \psi} + \frac{2}{h_\mu h_\psi^2}\frac{\partial}{\partial \mu}(h_\psi^2 \sigma_{12})\Big]. \tag{17.185}$$

The viscous parameters are obtained from the ice strength and velocity as follows:

$$\zeta = \frac{P}{2\Delta}, \tag{17.186}$$

$$\eta = \frac{P}{2e^2\Delta}, \tag{17.187}$$

$$\Delta = \Big[D_D^2 + \frac{1}{e^2}(D_T^2 + D_S^2)\Big]^{1/2}. \tag{17.188}$$

The pressure (strength) of the ice is a function of ice concentration and thickness:

$$P = P^* A h_I e \exp[-c^*(1 - A)], \tag{17.189}$$

where P^* is the scaling factor for pressure, c^* is a parameter that defines the dependency on concentration, and e is the axis ratio of the elliptic yield curve ($e = 2$).

The elastic parameter E is given by

$$E = \frac{2E_o \rho_I A h_I}{\Delta t_e^2}\min(\Delta x^2, \Delta y^2), \tag{17.190}$$

where E_o is a tuning factor that satisfies $0 < E_o < 1$, Δt_e is the time step for ice dynamics, and Δx and Δy are the zonal and meridional grid widths.

17.4.4 Boundary conditions

Surface stresses on the ice are exerted for the fractional area of the ice within a grid cell. The ice concentration is multiplied by the wind and ocean stresses.

For the stress on the ocean, the ice-ocean stress is exerted for the ice-covered area, and the wind stress is exerted for the open water area:

$$\nu_V \left(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z} \right) \Big|_{k=0} = -\frac{A}{\rho_o} (\tau_{IO_x}, \tau_{IO_y}) + \frac{(1-A)}{\rho_o} (\tau_{AO_x}, \tau_{AO_y}). \tag{17.191}$$

Note that $(\tau_{IO_x}, \tau_{IO_y})$ is reversed in sign because it is defined by (17.174) as the stress on the ice.

17.4.5 Solution procedure

Given the surface wind vector and the surface velocity of the ocean needed to compute surface stresses, the momentum equations ((17.171) and (17.172)) and the equations for stress tensors ((17.178), (17.179), and (17.180)) are solved.

First, the stress tensor is computed using the equations for stress tensors, the momentum equation is then solved using the stress tensor. Basically, the implicit method is used for prognostic variables for each equation. For example, stress tensor σ_1 is solved for σ_1^{m+1} as follows:

$$\frac{1}{E} \frac{\sigma_1^{m+1} - \sigma_1^m}{\Delta t} + \frac{\sigma_1^{m+1}}{2\zeta^m} + \frac{P}{2\zeta^m} = D_D^m. \tag{17.192}$$

Note that strain rate tensors and viscous parameters are updated every time step using a new velocity.

The momentum equations are solved using σ^{m+1} above:

$$\begin{aligned} \rho_I A h_I \frac{u_I^{m+1} - u_I^m}{\Delta t} &= \rho_I A h_I f v_I^{m+1} - \rho_I A h_I g \frac{1}{h_\mu} \frac{\partial h}{\partial \mu} + F_\mu(\sigma^{m+1}) + A \tau_{AIx} \\ &\quad + A C_{DAI} \rho_o |\mathbf{U}_w - \mathbf{u}_I^m| [(U_w - u_I^{m+1}) \cos\theta_o - (V_w - v_I^{m+1}) \sin\theta_o], \end{aligned} \tag{17.193}$$

$$\begin{aligned} \rho_I A h_I \frac{v_I^{m+1} - v_I^m}{\Delta t} &= -\rho_I A h_I f u_I^{m+1} - \rho_I A h_I g \frac{1}{h_\psi} \frac{\partial h}{\partial \psi} + F_\psi(\sigma^{m+1}) + A \tau_{AIy} \\ &\quad + A C_{DAI} \rho_o |\mathbf{U}_w - \mathbf{u}_I^m| [(V_w - v_I^{m+1}) \cos\theta_o + (U_w - u_I^{m+1}) \sin\theta_o]. \end{aligned} \tag{17.194}$$

Note that the surface velocity of the ocean (U_w, V_w) is constant during the integration. The starting time level of the ocean model is used, $n-1$ for the leap-frog time step, and n for the Matsuno scheme. For the leap-frog time step of the ocean model,

$$\overrightarrow{\tau_{IO}} = C_{DAI} \rho_o |\mathbf{U}_w^{n-1} - \mathbf{u}_I^m| [(\mathbf{U}_w^{n-1} - \mathbf{u}_I^{m+1}) \cos\theta_o + \mathbf{k} \times (\mathbf{U}_w^{n-1} - \mathbf{u}_I^{m+1}) \sin\theta_o]. \tag{17.195}$$

The time step of the ice dynamics is limited by the phase speed of the elastic wave. To damp the elastic waves during the sub-cycle, the ice dynamics is sub-cycled several tens of steps during one ocean model time step.

17.5 Advection and Diffusion

Fractional area (for both sea ice and open ocean), snow volume, ice volume, ice energy, and ice surface temperature (optional; set `flg_advec_tskin = .true.` in `mod_seaice_cat.F90`) of each category are advected. A multidimensional positive definite advection transport algorithm (MPDATA; Smolarkiewicz, 1984) is used. For all quantities which are advected, the harmonic-type diffusion can be applied to remove noises. In MRI.COM, advection and diffusion are solved serially as follows.

The advection equation for a property (α) is given by

$$\frac{\partial \alpha}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (h_\psi u \alpha)}{\partial \mu} + \frac{\partial (h_\mu v \alpha)}{\partial \psi} \right\} = 0. \tag{17.196}$$

The specific representation for α is a_0 for open water fraction, a_n for fractional area, v_n for ice volume per grid area, v_s for snow volume per grid area, and e_n for ice energy.

In MPDATA, (17.196) is first solved to obtain a temporary value (α^*) using the upstream scheme with a mid-point velocity between time levels n and $n+1$. Using this temporary value, an anti-diffusive velocity is computed as

$$\begin{aligned}
\tilde{u}_{i+\frac{1}{2},j} &= \frac{1}{\alpha^*}\left[\frac{1}{2}\left(|u^{n+\frac{1}{2}}|\Delta x - \Delta t(u^{n+\frac{1}{2}})^2\right)\frac{\partial \alpha^*}{\partial x} - \frac{1}{2}\Delta t u^{n+\frac{1}{2}} v^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial y}\right] \\
&\quad - \frac{1}{2}\Delta t u^{n+\frac{1}{2}}\left(\frac{\partial u}{\partial x}^{n+\frac{1}{2}} + \frac{\partial v}{\partial y}^{n+\frac{1}{2}}\right), \qquad (17.197)
\end{aligned}$$

$$\begin{aligned}
\tilde{v}_{i,j+\frac{1}{2}} &= \frac{1}{\alpha^*}\left[\frac{1}{2}\left(|v^{n+\frac{1}{2}}|\Delta y - \Delta t(v^{n+\frac{1}{2}})^2\right)\frac{\partial \alpha^*}{\partial y} - \frac{1}{2}\Delta t u^{n+\frac{1}{2}} v^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial x}\right] \\
&\quad - \frac{1}{2}\Delta t v^{n+\frac{1}{2}}\left(\frac{\partial u}{\partial x}^{n+\frac{1}{2}} + \frac{\partial v}{\partial y}^{n+\frac{1}{2}}\right). \qquad (17.198)
\end{aligned}$$

This velocity is used to compute a new value using the upstream scheme. See Section 17.13.1 for the discretized form.

Since MPDATA is positive definite, the new area and thickness should be positive. If the sum of the fractional area exceeds one, the ridging scheme will adjust the fractional area. Since energy is negative definite, the sign is reversed just before advection and returned to a negative value after the advection.

Next, the diffusion equation is solved for the distribution modified by advection. The specific form of the harmonic-type diffusion is represented as follows:

$$\begin{aligned}
\frac{\partial \alpha}{\partial t} &= \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial \mu}\left(\frac{h_\psi \kappa_H}{h_\mu}\frac{\partial \alpha}{\partial \mu}\right) + \frac{\partial}{\partial \psi}\left(\frac{h_\mu \kappa_H}{h_\psi}\frac{\partial \alpha}{\partial \psi}\right)\right\} \\
&= -\frac{1}{h_\mu h_\psi}\left\{\frac{\partial F_\mu^\alpha}{\partial \mu} + \frac{\partial F_\psi^\alpha}{\partial \psi}\right\} \qquad (17.199)
\end{aligned}$$

$$F_\mu^\alpha = -\frac{h_\psi \kappa_H}{h_\mu}\frac{\partial \alpha}{\partial \mu} \qquad (17.200)$$

$$F_\psi^\alpha = -\frac{h_\mu \kappa_H}{h_\psi}\frac{\partial \alpha}{\partial \psi}, \qquad (17.201)$$

where κ_H is the horizontal diffusion coefficients (required parameter, `diff_seaice_m2ps`). The diffusion term is advanced in time using the flux form (17.199), which are calculated for α modified by advection. If the diffusion CFL condition is met, the diffusion acts to smooth the input distribution, so that the output value does not deviate from the domain of the input value. See Section 17.13.2 for the discretized form. (Though MPDATA is positive difinite, it may take a negative value due to a numerical error. If a_n becomes negative, state variables are modified considering that there is no sea ice in the category ($a_n = 0$).

17.6 Ridging

As a result of advection, the sum of the fractional area might exceed one, especially where the velocity field is convergent. In such a case, it is assumed that ridging occurs among ice to yield a sum equal to or less than one. Even if the sum is less than one, ridging or rafting might occur where the concentration is high.

The ridging scheme of MRI.COM follows that of CICE (originally based on Thorndike et al. (1975)), which is briefly summarized in this section. In this scheme, the change rate in ice area fraction due to ridging (total of the categories) is determined by the strain calculated in the dynamics scheme. This scheme decides how to distribute the amount of change to each category, and redistributes the main variables of each category (i.e. a_0, a_n, v_n, v_s and e_n) based on it.

First, in preparation for the scheme, we define the cumulative function $G(h)$ of ice area fraction:

$$G(h) = \int_0^h g(h')dh', \qquad (17.202)$$

and $G(h)$ indicates the area of ice thinner than h including open water fraction (a_0). The boundary value for the discrete category, G_n, is given by

$$G_n = \begin{cases} G(H_n) & \text{for } n \geq 1 \\ a_0 & \text{for } n = 0 \end{cases} \qquad (17.203)$$

The property G_n is the area fraction summed from category 0 to n. As a result of advection, $A + a_0 = \sum_{n=0}^{N_c} a_n = 1$ is not guaranteed, so MRI.COM adjusts it so that $G_{N_c} = 1$.

First, we determine a fractional area that undergoes ridging: $a_P(h) = b(h)g(h)$. The weighting function $b(h)$ is chosen such that the ridging occurs for thin ice:

$$b(h) = \begin{cases} \frac{2}{G^*}(1 - \frac{G(h)}{G^*}) & \text{if } G(h) < G^* \\ 0 & \text{otherwise} \end{cases} \tag{17.204}$$

where $G(h)$ is the area fraction of ice thinner than h and G^* is an empirical constant with $G^* = 0.15$. The participation function for category n (a_{P_n}) is obtained by integrating $a_P(h)$ for a range $[H_{n-1},\ H_n]$ as

$$a_{P_n} = \int_{H_{n-1}}^{H_n} b(h)g(h)dh = \frac{2}{G^*}(G_n - G_{n-1})\left(1 - \frac{G_{n-1} + G_n}{2G^*}\right), \tag{17.205}$$

where $dG = gdh$ is used. The property a_{P_n} is the fractional contribution from the category n among the total area of ice subject to ridging. This equation is used for the category that satisfies $G_n < G^*$. If $G_{n-1} < G^* < G_n$, then G^* is replaced by G_n. If $G_{n-1} > G^*$, then $a_{P_n} = 0$. If the fractional area of open leads exceeds G^* ($a_0 > G^*$), then ridging does not occur. In (17.205), a_{P_0} is open leads' allocation of the ridging effect, and means closing of open leads due to flow convergence,

$$a_{P_0} = \frac{2}{G^*}G_0\left(1 - \frac{G_0}{2G^*}\right). \tag{17.206}$$

Opening due to ridging will be discussed later.

Ridging occurs such that the total area is reduced while conserving ice volume and energy. It is assumed that ice of thickness h_n is homogeneously distributed between $H_{\min} = 2h_n$ and $H_{\max} = 2\sqrt{H^* h_n}$ after ridging, where $H^* = 30\,\text{m}$ (as a default). The thickness ratio before and after the ridging is $k_n = (H_{\min} + H_{\max})/(2h_n)$. Therefore, when an area of category n is reduced by ridging at a rate r_n, the area of thicker categories is increased by r_n/k_n.

Among the new ridges, the fractional area that is distributed in category m is:

$$f_m^{\text{area}} = \frac{H_R - H_L}{H_{\max} - H_{\min}}, \tag{17.207}$$

where $H_L = \max(H_{m-1}, H_{\min})$ and $H_R = \min(H_m, H_{\max})$. The fractional volume that is distributed in category m is:

$$f_m^{\text{vol}} = \frac{(H_R)^2 - (H_L)^2}{(H_{\max})^2 - (H_{\min})^2}. \tag{17.208}$$

The snow volume and ice energy are distributed by the same ratio as the ice volume.

The net area lost by ridging and open water closing is assumed to be a function of the strain rates. The net rate of area loss of the ice pack (R_{net}) [unit: 1/s] is given by

$$R_{\text{net}} = \frac{C_s}{2}(\Delta - |D_D|) - \min(D_D, 0), \tag{17.209}$$

where C_s is the fraction of shear dissipation energy that contributes to ridge building (0.5 is used in MRI.COM), D_D is the divergence, and $\Delta = \left[D_D^2 + \frac{1}{e^2}(D_T^2 + D_S^2)\right]^{1/2}$. These strain rates are computed by the dynamics scheme.

The total rate of area loss due to ridging, $R_{\text{tot}} = \sum_{n=0}^{N_c} r_n$, is related to the net rate as follows:

$$R_{\text{net}} = r_0 + \sum_{n=1}^{N}\left(r_n - \frac{r_n}{k_n}\right) = \left[a_{P_0} + \sum_{n=1}^{N} a_{P_n}\left(1 - \frac{1}{k_n}\right)\right]R_{\text{tot}}, \tag{17.210}$$

where, r_0 indicates convergence or divergence of the open leads fraction, a_0. Since R_{net} is computed from (17.209), R_{tot} is computed from (17.210).

Based on the above discussion, the amounts of change due to ridging during the time step interval, Δt, is calculated. The decrease of area fraction for category n, a_r^n, is given by

$$a_r^n = r_n \Delta t = a_{P_n} R_{\text{tot}} \Delta t \qquad (n = 0 \sim N_c). \tag{17.211}$$

Taking into account the area reduction due to ridging ($1/k_n$), the area fraction transfer from category n to m, $\Delta a_{n,m}$, is

$$\Delta a_{n,m} = f_m^{\text{area}} \frac{a_r^n}{k_n} \qquad (m = 1 \sim N_c, n = 1 \sim m). \tag{17.212}$$

Note that transfers between the same category ($a_{m,m}$) can also exist. Similarly, the sea ice volume transfer, $\Delta v_{n,m}$, the snow volume transfer, $\Delta v_{s,n,m}$, and the enthalpy transfer, $\Delta e_{n,n}$, are

$$c_n = a_r^n / a_n \tag{17.213}$$

$$\Delta v_{n,m} = f_m^{\text{vol}} a_r^n h_I = f_m^{\text{vol}} c_n v_n \tag{17.214}$$

$$\Delta v_{s,n,m} = f_m^{\text{vol}} a_r^n h_s = f_m^{\text{vol}} c_n v_s \tag{17.215}$$

$$\Delta e_{n,m} = f_m^{\text{vol}} a_r^n (e_n / a_n) = f_m^{\text{vol}} c_n e_n . \tag{17.216}$$

Redistribution by ridging is calculated by the combination of the above transfer amounts. The area fraction decrease for category $n(= 1 \sim N_c)$, Δa_-^n, and the increase for category $m(= 1 \sim N_c)$, Δa_+^m, are

$$\Delta a_-^n = a_r^n \tag{17.217}$$

$$\Delta a_+^m = \sum_{n=1}^{m} \Delta a_{n,m} . \tag{17.218}$$

The area fraction decrease of open leads, Δa_-^0, and the increase, Δa_+^0, are

$$\Delta a_-^0 = a_r^0 \tag{17.219}$$

$$\Delta a_+^0 = R_{\text{net}} \Delta t . \tag{17.220}$$

In response to area fraction change, the decreases of v_n, v_s and e_n for category n, Δv_-^n, Δv_{s-}^n and Δe_-^n are

$$\Delta v_-^n = \begin{cases} \displaystyle\sum_{m=n}^{N_c} \Delta v_{n,m} & \text{for } n \leq N_c - 1 \\ 0 & \text{for } n = N_c \end{cases} \tag{17.221}$$

$$\Delta v_{s-}^n = \begin{cases} \displaystyle\sum_{m=n}^{N_c} \Delta v_{s,n,m} & \text{for } n \leq N_c - 1 \\ 0 & \text{for } n = N_c \end{cases} \tag{17.222}$$

$$\Delta e_-^n = \begin{cases} \displaystyle\sum_{m=n}^{N_c} \Delta e_{n,m} & \text{for } n \leq N_c - 1 \\ 0 & \text{for } n = N_c \end{cases} \tag{17.223}$$

Note that treatment at $n = N_c$ is different from a_n. Similarly, the increases of v_n, v_s and e_n for category m, Δv_+^m, Δv_{s+}^m and Δe_+^m are

$$\Delta v_+^m = \begin{cases} \displaystyle\sum_{n=1}^{m} \Delta v_{n,m} & \text{for } m \leq N_c - 1 \\ \displaystyle\sum_{n=1}^{N_c - 1} \Delta v_{n,m} & \text{for } m = N_c \end{cases} \tag{17.224}$$

$$\Delta v_{s+}^m = \begin{cases} \displaystyle\sum_{n=1}^{m} \Delta v_{s,n,m} & \text{for } m \leq N_c - 1 \\ \displaystyle\sum_{n=1}^{N_c - 1} \Delta v_{s,n,m} & \text{for } m = N_c \end{cases} \tag{17.225}$$

$$\Delta v_+^m = \begin{cases} \displaystyle\sum_{n=1}^{m} \Delta e_{n,m} & \text{for } m \leq N_c - 1 \\ \displaystyle\sum_{n=1}^{N_c - 1} \Delta e_{n,m} & \text{for } m = N_c \end{cases} \tag{17.226}$$

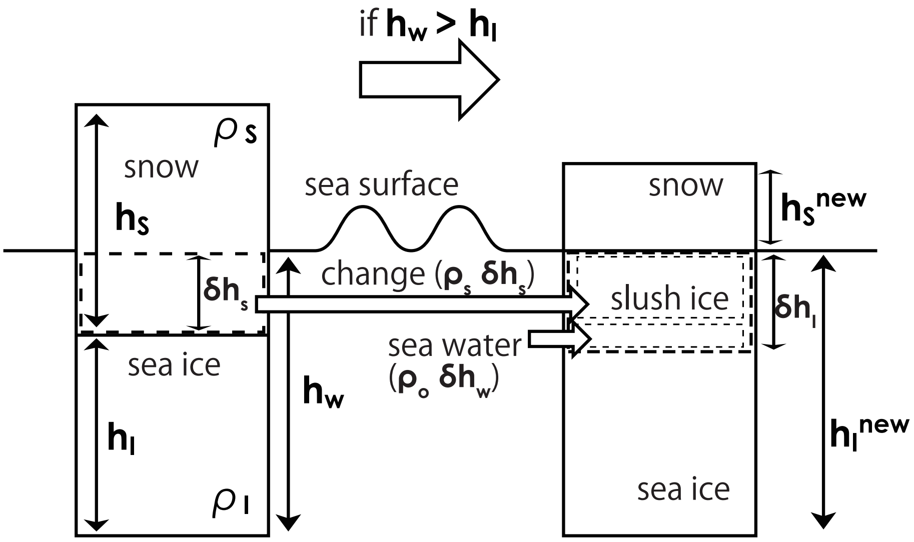

Figure17.6 A diagram of the modeled process in which the drafted part of the piled snow changes to sea ice (slush).

The notation of the loop is different for addition and subtraction, but it can be integrated in the same loop. Using these, the state variables after redistribution by ridging, a_n^{new}, v_n^{new}, v_s^{new} and e_n^{new} are

$$a_n^{\text{new}} = a_n - \Delta a_-^n + \Delta a_+^n \qquad (n = 0 \sim N_c) \tag{17.227}$$

$$v_n^{\text{new}} = v_n - \Delta v_-^n + \Delta v_+^n \qquad (n = 1 \sim N_c) \tag{17.228}$$

$$v_s^{\text{new}} = v_s - \Delta v_{s-}^n + \Delta v_{s+}^n \qquad (n = 1 \sim N_c) \tag{17.229}$$

$$e_n^{\text{new}} = e_n - \Delta e_-^n + \Delta e_+^n \qquad (n = 1 \sim N_c) \tag{17.230}$$

(Note that the category notation for incease is changed from m to n.)

In practice, we require that $a_r^n \leq a_n$ to avoid negative area fraction. If $A + a_0 > 1$ after ridging, R_{net} is adjusted to yield $A + a_0 = 1$, and the ridging procedure is repeated.

17.7 Transformation from snow to sea ice (slush ice formation)

When piled snow on sea ice is drafted, seawater permeates the drafted part and slush ice is formed. In the model, this process is expressed as shown in Fig. 17.6, and slush ice formation is treated as increase in sea ice (an extension of Hunke and Lipscomb (2006)). For the model stability, this process is calculated for each category after sea ice ridging. In this section, the case of variable sea ice salinity is described first, and then the case of fixed salinity is explained.

First, whether or not piled snow has drafted is determined as follows. The draft depth of sea ice, h_w, satisfies the following relation from the Archimedes' principle, assuming an equilibrium state,

$$\rho_I h_I + \rho_s h_s = \rho_o h_w, \tag{17.231}$$

thus

$$h_w = \frac{\rho_s}{\rho_o} h_s + \frac{\rho_I}{\rho_o} h_I. \tag{17.232}$$

Using the values at the next time step, which have been calculated, h_w and h_I are compared, and if $h_w > h_I$, that is,

$$h_s > \frac{\rho_o - \rho_I}{\rho_s} h_I, \tag{17.233}$$

it is judged that snow has drafted.

Next, as the thickness of the piled snow that changes to slush ice is written as δh_s and the thickness originated to sea water as δh_w, the masses of the snow and sea water in the slush ice are $\rho_s \delta h_s$ and $\rho_o \delta h_w$, respectively. Then, the mass ratio of sea water in the slush is

$$r_w = \rho_o \delta h_w / (\rho_o \delta h_w + \rho_s \delta h_s). \tag{17.234}$$

In the model, r_w is assumed as a constant value of 0.15 (Sturn and Masson, 2010). Since the sum of the decreased snow mass and sea water mass is equal to the mass of the newly formed sea ice,

$$\rho_I \delta h_I = \rho_s \delta h_s + \rho_o \delta h_w, \tag{17.235}$$

where the increased sea ice thickness is written by δh_I. Further, assuming that the buoyancy equilibrium is established even after the transformation, the following relationship is obtained.

$$\rho_o(h_I + \delta h_I) = \rho_I(h_I + \delta h_I) + \rho_s(h_s - \delta h_s). \tag{17.236}$$

Using the above three equations, $\delta h_I, \delta h_s$ and δh_w can be obtained as

$$\delta h_I = \frac{\rho_s h_s - (\rho_o - \rho_I) h_I}{\rho_o - r_w \rho_I}, \tag{17.237}$$

$$\delta h_s = (1 - r_w)\frac{\rho_I}{\rho_s}\delta h_I, \tag{17.238}$$

$$\delta h_w = r_w \frac{\rho_I}{\rho_o}\delta h_I. \tag{17.239}$$

In the model, the equations for the sea ice and snow volumes are derived by using the relations, $v_s = a_n h_s$ and $v_n = a_n h_I$, based on the above equations about one-dimensional heights. First, the draft judgment is

$$v_s > \frac{\rho_o - \rho_I}{\rho_s} v_n. \tag{17.240}$$

After some deformation, the increase of sea ice volume, Δv_n, the decrease of piled snow volume, Δv_s, and the increase of integrated sea ice salinity (referred by (17.259)), ΔF^S, for each category are obtained

$$\Delta v_n = a_n \delta h_I = \frac{\rho_s v_s - (\rho_o - \rho_I) v_n}{\rho_o - r_w \rho_I}, \tag{17.241}$$

$$\Delta v_s = a_n \delta h_s = (1 - r_w)\frac{\rho_I}{\rho_s}\Delta v_n, \tag{17.242}$$

$$\Delta F^S_{slush} = a_n \frac{\rho_o}{\rho_I} S_w \delta h_w = r_w \Delta v_n S_w, \tag{17.243}$$

where S_w is sea surface salinity. Thus, salinity in the newly formed slush ice, S_I, is

$$S_I = \Delta F^S_{slush} / \Delta v_n = r_w S_w. \tag{17.244}$$

Therefore, assuming that temperature of the slush ice is the freezing point, mS_I, the increase of sea ice enthalpy, Δe_n, can be given by

$$\Delta e_n = a_n E_S(mS_I, S_I)\delta h_I = E_S(mS_I, S_I)\Delta v_n, \tag{17.245}$$

where E_S is given by (17.14). In slush ice formation, latent heat cooling is required for sea water with a temperature of T_w to freeze, but the latent heat can be considered to be released to the atmosphere rather than exchange with the ocean. This is because the formation of slush ice under snow cover releases latent heat to the atmosphere through voids in the snow cover.

The upper and lower limits of the snow volume change, Δv_s, are shown here. From (17.240), (17.241), (17.242), $v_n > 0$ and $\rho_o > \rho_I$,

$$0 \le \Delta v_s < v_s. \tag{17.246}$$

Therefore, the modified snow volume, $v_s - \Delta v_s$, must not be negative. Since $\Delta v_n \ge 0$, sea ice must not decrease by this process.

This process causes water transport between the ocean and sea ice models. The water volume transport, W_{slush}, is

$$W_{slush} = \sum_n a_n \delta h_w / \Delta t = \frac{\rho_I}{\rho_o}\frac{r_w}{\Delta t}\sum_n \Delta v_n, \tag{17.247}$$

using the volume of seawater as a unit (positive upward). The temperature transport during a time step interval, F^θ_{slush} [° C m], is

$$F^\theta_{slush} = \frac{\rho_o}{\rho_I}\sum_n (T_w a_n \delta h_w) = T_w r_w \sum_n \Delta v_n, \tag{17.248}$$

where the sea ice volume is used as a unit to be consistent with other temperature transports. See (17.274) for the salinity transport flux.

At the end of this section, the case where sea ice salinity is fixed as $S_I = 4.0$ psu is explained (v5.0). First, $r_w = 0$ is used to consider only the change from piled snow to sea ice. The change amounts, (17.241), (17.242) and (17.245), can be used without modification. However, salinity is fixed so that

$$\Delta F^S_{slush} = S_I \Delta v_n. \tag{17.249}$$

Salinity in slush ice must be compensated from the first layer of the ocean model:

$$S^{\text{new}} = S^{\text{old}} - S_I \frac{\Delta v_n (\rho_I / \rho_o)}{\Delta z_1}. \tag{17.250}$$

17.8 Adjustment

As the final procedure in each time step, the following adjustment is applied to the sea ice distribution.

- If the thickness of sea ice of some category is out of its category bounds, the sea ice in that category is moved to the appropriate category.
- If the thickness of sea ice in category 1 is less than 0.1 meter, the thickness is set to 0.1 meter and the area fraction is adjusted accordingly.
- If the fractional area that sea ice occupies is too small, the sea ice of that category is forcibly melted (See Table 17.10).
- If sea ice temperature is above the freezing point, it melts.
- If ice area fraction is outside the definition range, the fraction is corrected to recalculate diagnostic variables (Section 17.1.5)

17.9 Fluxes from sea ice to ocean

The processes of exchanging physical quantities between the sea ice and ocean models can be roughly classified into two types, except for momentum. One is exchanges accompanied with water transport (mainly, melting and freezing of sea ice), by which water volume (I), temperature (F^θ_I) and salinity (F^S_I) are transported (Section 14.6.2). The other is heat flux (F^T) and salinity flux (F^S_{IO}) without water transport. Fluxes generated by various processes in the sea ice model are merged to the above five fluxes at the end of the sea ice model procedure, and they are handed over to the ocean model.

The fresh water transport, I, is

$$I = W_{FR} + W_O + W_{ROice} + W_{adjust} + W_{slush} + W_{ROsnow}, \tag{17.251}$$

where W_{FR} is given by (17.102), W_O by (17.139), W_{ROice} by (17.72) and (17.89), W_{ROsnow} by (17.73) and (17.140), and W_{slush} by (17.247) (see Fig. 17.3). The water flux W_{adjust} is due to the adjustment process of Sec. 17.8, which melts small sea ice below threshold.

The temperature transport, F^θ_I, is expressed by the sum of two components, the upward transport from ocean to sea ice, F^θ_{Iup}, and the downward transport from sea ice to ocean, F^θ_{Idown}, as follows

$$F^\theta_I = F^\theta_{Iup} + F^\theta_{Idown}, \tag{17.252}$$

$$F^\theta_{Iup} = F^\theta_{frzl} + F^\theta_{btm\_frz} + F^\theta_{slush} \tag{17.253}$$

$$F^\theta_{Idown} = F^\theta_{top} + F^\theta_{intr} + F^\theta_{btm\_melt} + F^\theta_{adjust}, \tag{17.254}$$

where F^θ_{top} is given by (17.74), F^θ_{intr} by (17.91), F^θ_{frzl} by (17.103), $F^\theta_{btm\_frz}$ by (17.141), $F^\theta_{btm\_melt}$ by (17.142), and F^θ_{slush} by (17.248). Note that the right hand side is all calculated by the sea ice volume. When passed to the ocean model, it is converted to sea water volume by multiplying by ρ_I/ρ_o.

The salinity transport, F^S_I, is

$$F^S_I = (W_{FR} + W_O + W_{ROice} + W_{adjust}) S_I. \tag{17.255}$$

In v5.0, the sea water exchanged with ice is assumed to have a constant salinity ($S_I = 4.0$ pss). Note that the temperature and salinity transports by snow melting, W_{ROsnow}, are always zero because snow is assumed to be 0° C and 0 psu.

The heat flux that is not involved in water exchange is

$$F_T = Q_{SO} + Q_{LO} + (1 - A)F_{T_L} + \sum_n a_n F_{T_I} + F_{adjust}, \tag{17.256}$$

where Q_{SO} is the latent heat flux due to melting sea ice or snow (17.87). The flux F_T is handed over to the ocean model, and added to the sea surface heat flux, Q_{OTHER} in (14.37).

The salinity flux without water transport is given by (17.95). This is added to the sea surface salinity flux of the ocean model, (14.41).

Other impacts on the ocean model.

1. There is a salinity flux from the ocean to the sea ice in transformation from snow to sea ice, (17.250), but the model treats it specially.
2. The rise in sea water temperature due to freezing of supercooled sea water (Sec. 17.2.3) is also treated specially, and converted to the heat flux in the ocean model (Sec. 14.7.2).

In v5.1, they are treated in a unified manner as the above five fluxes between the sea ice and ocean models.

17.10 Variable salinity of sea ice (v5.1)

Sea ice salinity has been treated as a constant value of 4.0 psu, but an option to make it variable was introduced in v5.1. This section describes the governing equation of sea ice salinity for this option.

17.10.1 Prognostic variable

The prognostic variable to be solved is sea ice salinity integrated for each category n per unit area, $S_v(n)$ [psu m^3 m^{-2}]. Similar to the expression (17.3), it is defined by

$$S_v(n) = \int_{H_{n-1}}^{H_n} Sg(h)hdh. \tag{17.257}$$

In the following, we will not go into the details of salinity S changing spatially in sea ice, but directly consider the time evolution of $S_v(n)$ as a total amount. When there is no sea ice for category n, i.e. $a_n = 0$, $S_v(n)$ is set zero, and $S_v(n) \geq 0$ is alway satisfied. Salinity in snow is always zero.

17.10.2 The governing equation

The integrated salinity is governed by the following equation

$$\frac{dS_v(n)}{dt} = F_S(n) + F_{drain}(n) + \mathcal{R}_{\mathrm{remap}}(n) + \mathcal{A}(n) + \mathcal{R}_{\mathrm{ridge}}(n) + F^S_{slush}(n) + F_{\mathrm{adjust}}(n), \tag{17.258}$$

where F_S is change due to sea ice melting and freezing, F_{drain} is decrease due to desalination, $\mathcal{R}_{\mathrm{remap}}$ is transfer among categories, $\mathcal{A}$ is advection and diffusion, and $\mathcal{R}_{\mathrm{ridge}}$ is category redistribution by ridging. In addition, F^S_{slush} is increase due to slush ice formation (transformation of piled snow to sea ice), and F_{adjust} is increase or decrease due to other adjustments (Sec. 17.8).

The forward difference is used for the time difference, then

$$S^{m+1}_v(n) = S^m_v(n) + \Delta F_S(n) + \Delta F^S_{drain}(n) + \Delta\mathcal{R}_{\mathrm{remap}}(n) + \Delta\mathcal{A}(n) + \Delta\mathcal{R}_{\mathrm{ridge}}(n) + \Delta F^S_{slush}(n) + \Delta F_{\mathrm{adjust}}(n), \tag{17.259}$$

where the terms with Δ are amounts of change during the time step from m to $m+1$ (tendencies). In the following, how to evaluate each term on the right hand side and solve the time evolution will be explained according to the seven calculation steps.

First, $\Delta F_S(n)$ is evaluated by decomposing it into the contributions of each process as follows.

$$\Delta F_S(n) = \Delta F_{\mathrm{melt}}(n) + \Delta F_{\mathrm{frzl\_btm}}(n) + \Delta F_{\mathrm{frzl\_sea}} + \Delta F_{\mathrm{conduct}}(n) + \Delta F_{\mathrm{frzl\_air}}. \tag{17.260}$$

The term $\Delta F_{\mathrm{melt}}(n)$ is decrease due to melting at the sea ice surface (Sec. 17.2.1) and in the sea ice (Sec. 17.2.3),

$$\Delta F_{\mathrm{melt}}(n) = \frac{\rho_0}{\rho_I} W_{RO_{\mathrm{ice}}}(n) S^m(n) a_n \Delta t \qquad \leq 0, \tag{17.261}$$

where $S^m(n)$ indicates average salinity ($S_v^m(n)/v_n^m$) for category n at the time step m. The factor ρ_0/ρ_I is necessary to convert the volume of seawater into the volume of sea ice. The term $\Delta F_{frzl\_btm}(n)$ is increase due to frazil ice formation at the ice bottom (sea ice growth by ocean supercooling, i.e. sea water with a temperature below the freezing point, Sec. 17.2.3), and $\Delta F_{\mathrm{frzl\_sea}}$ is increase due to that in open water and open leads. They are given by Eqs. (17.94) and (17.101). The term $\Delta F_{\mathrm{conduct}}(n)$ indicates increase due to freezing or decrease due to melting at the ice bottom through heat conduction,

$$\Delta F_{\mathrm{conduct}}(n) = \begin{cases} \dfrac{\rho_0}{\rho_I} W_{IO}(n) S_{0_I} a_n \Delta t & \text{if } W_{IO}(n) > 0 \\ \dfrac{\rho_0}{\rho_I} W_{IO}(n) S^m(n) a_n \Delta t & \text{if } W_{IO}(n) < 0, \end{cases} \tag{17.262}$$

where S_{0_I} is the ice bottom skin salinity by the scheme of Mellor and Kantha (1989). The term $\Delta F_{\mathrm{frzl\_air}}$ is increase due to frazil ice formation in open water and open leads,

$$\Delta F_{\mathrm{frzl\_air}} = \Delta v_{\mathrm{frazil\_air}} S_{0_I}, \tag{17.263}$$

where $\Delta v_{\mathrm{frazil\_air}}$ is given by (17.137).

Second, the desalination, $\Delta F^S_{drain}(n)$, is decided (under development).

Thrid, increase or decrease due to the sea ice remapping, $\Delta\mathcal{R}_{\mathrm{remap}}$, is evaluated. Since the sea ice volume transfer among categories is given by (17.163) and (17.165),

$$\Delta\mathcal{R}_{\mathrm{remap}}(n) = \Delta v_{n-1,+} S'(n-1) - \Delta v_{n,+} S'(n) + \Delta v_{n+1,-} S'(n+1) - \Delta v_{n,-} S'(n), \tag{17.264}$$

where average salinity for category n, $S'(n)$, includes the effects of ΔF_S and $\Delta F^S_{drain}(n)$ at this time step, that is,

$$S'(n) = \left(S_v^m(n) + \Delta F_S + \Delta F^S_{drain}(n) \right) / v_n, \tag{17.265}$$

(The volume v_n has already been modified to reflect the thermodynamic process.) In addition, the sum of $\Delta F_{\mathrm{frzl\_air}}$ and $\Delta F_{\mathrm{frzl\_sea}}$ is added to $S_v(n)$ of the corresponding category after Sec. 17.3.2.

Fourth, advection and diffusion are solved. We use the MPDATA scheme like any other sea ice variable, and update S_v itself as written in Sec. 17.13.1 instead of finding the advection-diffusion term, $\mathcal{A}$. In the program, a provisional value is obtained by

$$S_v''(n) = S_v^m(n) + \Delta F_S(n) + \Delta\mathcal{R}_{\mathrm{remap}}(n), \tag{17.266}$$

and then

$$S_v'''(n) = S_v''(n) + \Delta\mathcal{A} \tag{17.267}$$

is derived directly from the scheme.

Fifth, increase or decrease due to sea ice ridging, $\Delta\mathcal{R}_{\mathrm{ridge}}$, is evaluated. As with any other variable, such as (17.215), the transfer from category n to m, $\Delta S_{v,n,m}$, is

$$\Delta S_{v,n,m} = f_m^{\mathrm{vol}} c_n S_v'''(n). \tag{17.268}$$

Using this,

$$\Delta S^n_{v,-} = \begin{cases} \displaystyle\sum_{m=n}^{N_c} \Delta S_{v,n,m} & \text{for } n \le N_c - 1 \\ 0 & \text{for } n = N_c \end{cases} \tag{17.269}$$

$$\Delta S^n_{v,+} = \begin{cases} \displaystyle\sum_{m=1}^{n} \Delta S_{v,m,n} & \text{for } n \le N_c - 1 \\ \displaystyle\sum_{m=1}^{N_c-1} \Delta S_{v,m,n} & \text{for } n = N_c \end{cases} \tag{17.270}$$

$$\Delta\mathcal{R}_{\mathrm{ridge}}(n) = -\Delta S^n_{v,-} + \Delta S^n_{v,+} \qquad (n = 1 \sim N_c). \tag{17.271}$$

Sixth, salinity increase due to slush ice formation, $\Delta F^S_{slush}(n)$, is given by (17.243).

Finally, regarding the values obtained from the above,

$$S_v''''(n) = S_v'''(n) + \Delta\mathcal{R}_{\mathrm{ridge}}(n) + \Delta F^S_{slush}(n) \tag{17.272}$$

the adjustment term, $\Delta F_{\mathrm{adjust}}(n)$, is calculated following sea ice incease or decrease in Sec. 17.8. Then, salinity at the next step, $m + 1$, is determined by

$$S_v^{m+1}(n) = S_v''''(n) + \Delta F_{\mathrm{adjust}}(n). \tag{17.273}$$

17.10.3 Salinity transport between sea ice and ocean

In v5.0, where sea ice salinity is fixed, the temperature transport associated with water transport between sea ice and ocean, F_I^θ in (14.33), and the salinity transport, F_I^S in (14.34), are obtained directly from components of the water transport, I in (17.251), such as (17.255). In v5.1 or later, the transport calculation has been modified to allow variable sea ice salinity. Sea ice has different salinity and freezing point at each grid and each category, so that salinity and temperature fo the transported water due to ice forming or melting vary at each grid and each category. Therefore, the model has been changed to calculate F_I^S and F_I^θ for each process, as described below.

The salinity transport between ocean and sea ice during one time step, F^S [psu m^3 m^{-2}], is

$$F^S = \Sigma_{n=1}^{N_c} \left\{\Delta F_S(n) + \Delta F^S_{drain}(n) + \Delta F^S_{slush}(n) + \Delta F_{\mathrm{adjust}}(n)\right\}, \tag{17.274}$$

using the volume of sea ice as a unit. The positive values indicate transfer from ocean to sea ice (upward), while the negative values the reverse direction (downward). Based on this, the salinity flux for the ocean, F_I^S [psu cm^3 cm^{-2} s^{-1}], is given by

$$F_I^S = 10^2(\rho_I/\rho_o)F^S/\Delta t, \tag{17.275}$$

where the factor ρ_I/ρ_o is a unit conversion from sea ice volume to sea water volume, and 10^2 is a unit conversion from [m] to [cm].

The temperature transport, F_I^θ, is basically same as (17.252), though transport due to discharge of the brine water, $F^\theta_{\mathrm{drain}}$, is added

$$F_I^\theta = F^\theta_{Iup} + F^\theta_{Idown} + F^\theta_{drain}. \tag{17.276}$$

17.10.4 Usage

See `docs/README_namelist.md` for use of this scheme.

17.11 Coupling with an atmospheric model

17.11.1 General features

In coupled mode (`SCUPCGCM`), the boundary between the atmospheric component and the ocean-ice component is at the air-ice(snow) boundary. The fluxes above the air-ice boundary are computed by the atmospheric component and passed to the ocean-ice component via the coupler Scup (Yoshimura and Yukimoto, 2008). All the information needed by the oceanic component is received by calling `cgcm_scup__get_a2o` (`cgcm_scup.F90`) from `ogcm__run` (`ogcm.F90`) at the beginning of a coupling cycle. The information needed by the ice part is extracted by calling `get_fluxi_a2o` (`get_fluxs.F90`) from `iaflux` (`ice_flux.F90`). The main part of the ice is solved using surface fluxes and ice surface temperature from the atmospheric component.

In the atmospheric component, temperature in the atmospheric boundary layer and at the ice surface (T_3(`tsfcin`)) are computed along with heat flux in snow layer ($Q_S = Q_{I2}$ (`fheatu`)) using ice temperature (T_1(`t1icen`)), snow thickness (h_s(`hsnwn`)), and ice thickness (h_I(`hicen`)) given by the oceanic component (see Figure 17.2 and Table 17.1). The properties needed by the atmospheric component are sent via `cgcm_scup__put_o2a` (`cgcm_scup.F90`) from `ogcm__run` (`ogcm.F90`) at the end of a coupling cycle.

To conserve heat and water in the coupled atmosphere-ocean system, globally integrated heat and fresh water flux must match between the atmospheric and oceanic components. To achieve this, surface fluxes are adjusted in several steps, which are explained in the following.

17.11.2 Correct errors due to interpolation

First, to resolve errors that arise during the transformation process, a globally constant adjustment factor for heat flux $q_{\mathrm{adj}}^{\mathrm{trans}}$ is introduced so that

$$\int \Big(\sum_{n=0}^{ncat} a_n^{\mathrm{ini}} q_n + q_{\mathrm{adj}}^{\mathrm{trans}} \Big) dS = Q_{\mathrm{ocean}}^{\mathrm{atm}}, \tag{17.277}$$

where a_n^{ini} is the area fraction of sea ice for category n at the time of exchange, q_n is the net heat flux for category n, and $Q_{\mathrm{ocean}}^{\mathrm{atm}}$ is the net heat flux of the atmospheric model integrated over the ocean. The adjustment factor is given by

$$q_{\mathrm{adj}}^{\mathrm{trans}} = \big[Q_{\mathrm{ocean}}^{\mathrm{atm}} - \int \sum_{n=0}^{ncat} a_n^{\mathrm{ini}} q_n dS \big] / S_{\mathrm{ocean}}, \tag{17.278}$$

where S_{ocean} is the global ocean area of the ocean model. $q_{\mathrm{adj}}^{\mathrm{trans}}$ is added to the net longwave radiation flux.

For fresh water, $f_{\mathrm{adj}}^{\mathrm{trans}}$ is introduced so that

$$\int \Big(\sum_{n=0}^{ncat} a_n^{\mathrm{ini}} f_n^{\mathrm{evap}} + f^{\mathrm{prcp}} + f^{\mathrm{rof}} + f_{\mathrm{adj}}^{\mathrm{trans}} \Big) dS = F_{\mathrm{ocean}}^{\mathrm{atm}}, \tag{17.279}$$

where a_n^{ini} is the area fraction of sea ice for category n at the time of exchange, f_n^{evap} is the evaporation or sublimation for category n, f^{prcp} is the precipitation, and f^{rof} is the continental run off, and $F_{\mathrm{ocean}}^{\mathrm{atm}}$ is the net fresh water of the atmospheric model integrated over the ocean. The adjustment factor is given by

$$f_{\mathrm{adj}}^{\mathrm{trans}} = \big[F_{\mathrm{ocean}}^{\mathrm{atm}} - \int \Big(\sum_{n=0}^{ncat} a_n^{\mathrm{ini}} f_n^{\mathrm{evap}} + f^{\mathrm{prcp}} + f^{\mathrm{rof}} \Big) dS \big] / S_{\mathrm{ocean}}, \tag{17.280}$$

where S_{ocean} is the global ocean area of the ocean model. $f_{\mathrm{adj}}^{\mathrm{trans}}$ is added to either precipitation or evaporation according to its sign.

This operation is done at subroutine `adjust_fluxes_global` of `get_fluxs.F90`.

17.11.3 Taking into account of the evolution of sea ice state

Second, as the ocean-sea ice system evolves during a coupling cycle, the area fraction of sea ice changes from the one at the time of data exchange. Therefore, the global integral would be different from the initial state. To resolve this difference, another adjustment factor $q_{\mathrm{adj}}^{\mathrm{evo}}$ is introduced so that

$$\int \Big(\sum_{n=0}^{ncat} a_n^{\mathrm{evo}} q_n + q_{\mathrm{adj}}^{\mathrm{evo}} \Big) dS = \int \sum_{n=0}^{ncat} a_n^{\mathrm{ini}} q_n dS, \tag{17.281}$$

where a_n^{evo} is the evolved area fraction of sea ice for category n. The adjustment factor is given as follows:

$$q_{\mathrm{adj}}^{\mathrm{evo}} = \Big[\int \sum_{n=0}^{ncat} a_n^{\mathrm{ini}} q_n dS - \int \sum_{n=0}^{ncat} a_n^{\mathrm{evo}} q_n dS \Big] / S_{\mathrm{ocean}}. \tag{17.282}$$

In total, $q_{\mathrm{adj}}^{\mathrm{trans}} + q_{\mathrm{adj}}^{\mathrm{evo}}$ is added to the longwave radiation flux.

$${q_{\mathrm{adj}}^{LW}}_n = q_{\mathrm{atm}n}^{LW} + q_{\mathrm{adj}}^{\mathrm{trans}} + q_{\mathrm{adj}}^{\mathrm{evo}} \tag{17.283}$$

For fresh water, because evaporation or sublimation may be affected by the evolution of sea ice, $f_{\mathrm{adj}}^{\mathrm{evo}}$ is introduced so that

$$\int \Big(\sum_{n=0}^{ncat} a_n^{\mathrm{evo}} f_n^{\mathrm{evap}} + f_{\mathrm{adj}}^{\mathrm{evo}} \Big) dS = \int \sum_{n=0}^{ncat} a_n^{\mathrm{ini}} f_n^{\mathrm{evap}} dS, \tag{17.284}$$

where a_n^{evo} is the evolved area fraction of sea ice for category n. The adjustment factor is given as follows:

$$f_{\mathrm{adj}}^{\mathrm{evo}} = \Big[\int \sum_{n=0}^{ncat} a_n^{\mathrm{ini}} f_n^{\mathrm{evap}} dS - \int \sum_{n=0}^{ncat} a_n^{\mathrm{evo}} f_n^{\mathrm{evap}} dS \Big] / S_{\mathrm{ocean}}. \tag{17.285}$$

$f_{\text{adj}}^{\text{evo}}$ is added to either evaporation or precipitation according to its sign.

This operation is done at subroutine `adjust_fluxes_local` of `get_fluxs.F90`.

17.11.4 Correct errors due to solution method

Third, discrepancies due to solution methods adopted at the air-ice interface (Section 17.2.1) are corrected. To ensure a stable model integration, the ice surface temperature is set as an average of those of the atmospheric component and the ocean-ice component. This new temperature T_3^o is used to replace the upward longwave flux of the adjusted heat flux above the air-ice interface (q_{adj}^{AI}) that contains the net longwave flux (q_{adj}^{LW}). Because the longwave flux q_{adj}^{LW} contains the upward flux due to T_3^a of the atmospheric model, q_{adj}^{LW} is replaced with $q_{\text{adj}}^{LW'}$ in the following manner,

$$q_{\text{adj}}^{LW'} = q_{\text{adj}}^{LW} - \epsilon\sigma(T_3^a)^4 + \epsilon\sigma(T_3^o)^4. \tag{17.286}$$

For the surface where snow or ice is melting, the amount of melting is calculated using $q_{\text{adj}}^{AI'}$ that contains $q_{\text{adj}}^{LW'}$, the discrepancy of the longwave radiation over melting surface should be stored for flux correction.

For the surface where melting does not occur, the interior flux (Q_S or Q_{I2}, they are equal), is obtained by equating $q_{\text{adj}}^{AI'}$ and Q_S. Q_S is different than q_{adj}^{AI} owing to the adjustment for the longwave radiation explained above. Furthermore, $q_{\text{adj}}^{AI'}$ and Q_S are not completely the same owing to the approximations used in the solution method explained in Section 17.2.1. This is not a serious problem for an ocean-only simulation. But this should be taken into account for the coupled model where exact surface heat flux conservation is required. Thus the difference between the adjusted air-ice flux and the interior flux, $q_{\text{adj}}^{AI} - Q_S$, should be stored.

To summarize, discrepancy of fluxes are stored in q_{diff} in the following manner,

$$q_{\text{diff}} = q_{\text{adj}}^{AI} - q_{\text{adj}}^{AI'} = q_{\text{adj}}^{LW}(T_3^a) - q_{\text{adj}}^{LW}(T_3^o), \qquad \text{if surface is melting} \tag{17.287}$$

$$= q_{\text{adj}}^{AI} - Q_S, \qquad \text{if surface is not melting.} \tag{17.288}$$

This is integrated over sea ice and divided by the oceanic area of the ocean model to give an offset factor $q_{\text{adj}}^{\text{stable}}$, which is added to the oceanic surface heat flux globally,

$$q_{\text{adj}}^{\text{stable}} = \int \sum_{n=1}^{ncat} a_n (q_{\text{diff}})_n dS / S_{\text{ocean}}. \tag{17.289}$$

Sampling is done at subroutine `iaflux` of ice_flux.F90 and flux adjustment is done at subroutine `si_exit` of `mod_seaice_cat.F90`.

17.12 Nesting

17.12.1 One-way nesting

In one-way nesting, it is recommended to set the boundary between coarse and fine resolution models (parent and child respectively) along the line connecting tracer points. Setting velocity points as the boundary is awkward for several reasons. For example, the velocity of the child model is zero at the boundary if sea ice of the parent model does not exist around boundary. However, sea ice may exist inside the boundary in the child model in one-way nesting. Lacking the transport removing sea ice from the child model through the boundary, sea ice may accumulate at tracer cells adjacent to the boundary. A serious discontinuity may result. If tracer points are set as the boundary, this kind of accumulation will not occur. However, conservation does not hold.

17.12.2 Two-way nesting

In two-way nesting, it is necessary to set velocity points as the boundary to impose global conservation including sea ice. By reflecting the child model state to the parent model, the problem seen in one-way nesting, an inconsistent distribution of sea ice around the boundary, may be avoided. However, owing to the application of the flux adjustment for conservation, the positive definiteness of area fraction and volume of sea ice and snow, or negative definiteness of sea ice enthalpy are not guaranteed after the advection equation is integrated.

This problem is avoided by the following treatment on the transport fluxes. First, the child model flux is adjusted to match the parent model flux (Chapter 22). Second, if the original flux calculated by the child model is going out of child model's main region, the flux received from the parent model is replaced by this outgoing flux. Third, the difference between the outgoing flux and the parent model flux is sent to the parent model. Fourth, if the value just inside the boundary of the child model is predicted to be negative, the boundary flux is adjusted so that the value does not get negative (a certain amount of quantity is taken from the parent). The adjustment factor is sent to the parent model. Fifth, in the parent model, the flux corrections received from the child model are summed over the corresponding parent grid and added to the original flux of the parent model.

Another problem arises for the interior temperature of sea ice. Owing to the discrepancy between the thermal energy and the volume of the ice pack after advection, the interior temperature of sea ice layer may get erroneous. Therefore, on both sides of the parent-child boundary, interior temperature of sea ice is calculated as the average of the top and bottom surfaces of the sea ice,

$$T_1^{\text{adj}} = \frac{T_2 + T_0}{2}. \tag{17.290}$$

The difference between the predicted and adjusted ice energy is integrated along the parent-child boundary (Γ) as

$$Q_{\text{diff}} = \int_\Gamma \sum_{n=1}^{ncat} \rho_I a_n h_n \left[E(T_1^{\text{adj}}, r_1^{\text{adj}}) - E(T_1, r_1)\right] dS. \tag{17.291}$$

This is reflected as a correction factor for the ocean surface heat flux of each model, which is computed as

$$q_{\text{adj}}^{thermal} = Q_{\text{diff}} / S_{\text{ocean}}^{\text{core}}, \tag{17.292}$$

where $S_{\text{ocean}}^{\text{core}}$ is the area of the main region of the parent or child model.

17.13 Discretization

17.13.1 Advection (MPDATA)

In MPDATA, tracer (α in Figure 17.7) is updated following a three-step procedure.

a. A temporary value is computed using an upstream scheme.

The tracer fluxes at the side boundaries of a T-cell are:

$$\begin{aligned}
F_x\left(\alpha_{i,j}^n, \alpha_{i+1,j}^n, u_{i+\frac{1}{2},j}^n\right) &= \left[\alpha_{i,j}^n(u_{i+\frac{1}{2},j}^n + |u_{i+\frac{1}{2},j}^n|) + \alpha_{i+1,j}^n(u_{i+\frac{1}{2},j}^n - |u_{i+\frac{1}{2},j}^n|)\right] \frac{\Delta y}{2}, \\
F_y\left(\alpha_{i,j}^n, \alpha_{i,j+1}^n, v_{i,j+\frac{1}{2}}^n\right) &= \left[\alpha_{i,j}^n(v_{i,j+\frac{1}{2}}^n + |v_{i,j+\frac{1}{2}}^n|) + \alpha_{i,j+1}^n(v_{i,j+\frac{1}{2}}^n - |v_{i,j+\frac{1}{2}}^n|)\right] \frac{\Delta x}{2},
\end{aligned} \tag{17.293}$$

where

$$u_{i+\frac{1}{2},j} = \frac{1}{2}\left(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}}\right), \tag{17.294}$$

$$v_{i,j+\frac{1}{2}} = \frac{1}{2}\left(v_{i+\frac{1}{2},j+\frac{1}{2}} + v_{i-\frac{1}{2},j+\frac{1}{2}}\right). \tag{17.295}$$

The zonal flux is defined at the closed circle and the meridional flux is defined at the closed square in Figure 17.7. Using this, the temporary value (α^*) is computed using an upstream scheme:

$$\begin{aligned}
\frac{(\alpha_{i,j}^* - \alpha_{i,j}^n)\Delta S_{i,j}}{\Delta t} =& F_x\left(\alpha_{i-1,j}^n, \alpha_{i,j}^n, u_{i-\frac{1}{2},j}^{n+\frac{1}{2}}\right) - F_x\left(\alpha_{i,j}^n, \alpha_{i+1,j}^n, u_{i+\frac{1}{2},j}^{n+\frac{1}{2}}\right) \\
&+ F_y\left(\alpha_{i,j-1}^n, \alpha_{i,j}^n, v_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}\right) - F_y\left(\alpha_{i,j}^n, \alpha_{i,j+1}^n, v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}\right).
\end{aligned} \tag{17.296}$$

b. Compute the anti-diffusive transport velocity.

Using the temporary value computed in the first step, the anti-diffusive transport velocity is computed as follows:

$$\begin{aligned}\tilde{u}_{i+\frac{1}{2},j} =& \frac{1}{2}\left[\frac{1}{\alpha^{*(2)}_{i+\frac{1}{2},j}}\left(|u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}|\Delta x - \Delta t(u^{n+\frac{1}{2}}_{i+\frac{1}{2},j})^2\right)\left(\frac{\partial \alpha^*}{\partial x}\right)_{i+\frac{1}{2},j} - \frac{1}{\alpha^{*(6)}_{i+\frac{1}{2},j}}\Delta t u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\left(v^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial y}\right)_{i+\frac{1}{2},j}\right] \\ &- \frac{1}{2}\Delta t u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i+\frac{1}{2},j},\end{aligned} \tag{17.297}$$

$$\begin{aligned}\tilde{v}_{i,j+\frac{1}{2}} =& \frac{1}{2}\left[\frac{1}{\alpha^{*(2)}_{i,j+\frac{1}{2}}}\left(|v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}|\Delta y - \Delta t(v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}})^2\right)\left(\frac{\partial \alpha^*}{\partial y}\right)_{i,j+\frac{1}{2}} - \frac{1}{\alpha^{*(6)}_{i,j+\frac{1}{2}}}\Delta t v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(u^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial x}\right)_{i,j+\frac{1}{2}}\right] \\ &- \frac{1}{2}\Delta t v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j+\frac{1}{2}},\end{aligned} \tag{17.298}$$

where

$$\alpha^{*(2)}_{i+\frac{1}{2},j} = \frac{1}{2}(\alpha^*_{i,j}+\alpha^*_{i+1,j}), \tag{17.299}$$

$$\alpha^{*(2)}_{i,j+\frac{1}{2}} = \frac{1}{2}(\alpha^*_{i,j}+\alpha^*_{i,j+1}), \tag{17.300}$$

$$\alpha^{*(6)}_{i+\frac{1}{2},j} = \frac{1}{8}(\alpha^*_{i,j-1}+\alpha^*_{i+1,j-1}+2\alpha^*_{i,j}+2\alpha^*_{i+1,j}+\alpha^*_{i,j+1}+\alpha^*_{i+1,j+1}), \tag{17.301}$$

$$\alpha^{*(6)}_{i,j+\frac{1}{2}} = \frac{1}{8}(\alpha^*_{i-1,j}+\alpha^*_{i-1,j+1}+2\alpha^*_{i,j}+2\alpha^*_{i,j+1}+\alpha^*_{i+1,j}+\alpha^*_{i+1,j+1}), \tag{17.302}$$

$$\left(\frac{\partial \alpha^*}{\partial x}\right)_{i+\frac{1}{2},j} = \frac{\alpha^*_{i+1,j}-\alpha^*_{i,j}}{\Delta x}, \tag{17.303}$$

$$\left(\frac{\partial \alpha^*}{\partial y}\right)_{i,j+\frac{1}{2}} = \frac{\alpha^*_{i,j+1}-\alpha^*_{i,j}}{\Delta y}, \tag{17.304}$$

$$\begin{aligned}\left(v^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial y}\right)_{i+\frac{1}{2},j} =& \frac{1}{4}\Bigg[v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(\frac{\partial \alpha^*}{\partial y}\right)_{i,j+\frac{1}{2}} + v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\left(\frac{\partial \alpha^*}{\partial y}\right)_{i,j-\frac{1}{2}} \\ &+ v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(\frac{\partial \alpha^*}{\partial y}\right)_{i+1,j+\frac{1}{2}} + v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\left(\frac{\partial \alpha^*}{\partial y}\right)_{i+1,j-\frac{1}{2}}\Bigg],\end{aligned} \tag{17.305}$$

$$\begin{aligned}\left(u^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial x}\right)_{i,j+\frac{1}{2}} =& \frac{1}{4}\Bigg[u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\left(\frac{\partial \alpha^*}{\partial x}\right)_{i+\frac{1}{2},j} + u^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\left(\frac{\partial \alpha^*}{\partial x}\right)_{i-\frac{1}{2},j} \\ &+ u^{n+\frac{1}{2}}_{i+\frac{1}{2},j+1}\left(\frac{\partial \alpha^*}{\partial x}\right)_{i+\frac{1}{2},j+1} + u^{n+\frac{1}{2}}_{i-\frac{1}{2},j+1}\left(\frac{\partial \alpha^*}{\partial x}\right)_{i-\frac{1}{2},j+1}\Bigg],\end{aligned} \tag{17.306}$$

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = \frac{1}{2}\left[\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j} + \left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i+1,j}\right], \tag{17.307}$$

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} = \frac{1}{2}\left[\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j} + \left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j+1}\right], \tag{17.308}$$

and

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n}_{i,j} = \frac{(u^n_{i+\frac{1}{2},j}\Delta y_{i+\frac{1}{2},j} - u^n_{i-\frac{1}{2},j}\Delta y_{i-\frac{1}{2},j} + v^n_{i,j+\frac{1}{2}}\Delta x_{i,j+\frac{1}{2}} - v^n_{i,j-\frac{1}{2}}\Delta x_{i,j-\frac{1}{2}})}{\Delta S_{i,j}}. \tag{17.309}$$

c. Update tracer using the anti-diffusive velocity starting from the temporary value.

$$\begin{aligned}\frac{(\alpha^{n+1}_{i,j}-\alpha^{*}_{i,j})\Delta S_{i,j}}{\Delta t} =& F_x\left(\alpha^{*}_{i-1,j},\alpha^{*}_{i,j},\tilde{u}_{i-\frac{1}{2},j}\right)-F_x\left(\alpha^{*}_{i,j},\alpha^{*}_{i+1,j},\tilde{u}_{i+\frac{1}{2},j}\right)\\ &+F_y\left(\alpha^{*}_{i,j-1},\alpha^{*}_{i,j},\tilde{v}_{i,j-\frac{1}{2}}\right)-F_y\left(\alpha^{*}_{i,j},\alpha^{*}_{i,j+1},\tilde{v}_{i,j+\frac{1}{2}}\right).\end{aligned} \tag{17.310}$$

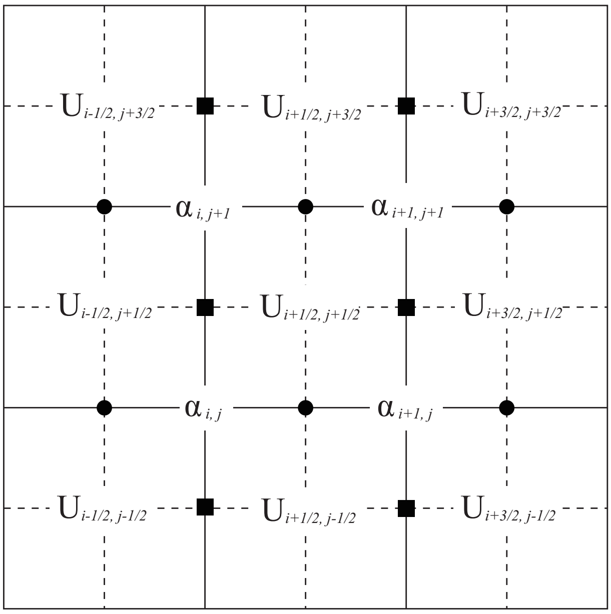

Figure17.7 Position for tracer (α) and velocity (U). Area and thickness are defined at tracer points. Zonal fluxes are computed at closed circles and meridional fluxes are computed at closed squares. The budget is computed for a unit cell for α.

17.13.2 Diffusion (Laplace operator)

The discrete equation of the diffusion equation (17.199) is

$$\alpha^{\text{new}}_{i,j} = \alpha_{i,j}+\frac{\Delta t}{\Delta S_{i,j}}\left(F^{x}_{i-\frac{1}{2},j}-F^{x}_{i+\frac{1}{2},j}+F^{y}_{i,j-\frac{1}{2}}-F^{y}_{i,j+\frac{1}{2}}\right) \tag{17.311}$$

$$F^{x}_{i+\frac{1}{2},j} = -\kappa_H \Delta y_{i+\frac{1}{2},j}\frac{1}{\Delta x_{i+\frac{1}{2},j}}(\alpha_{i+1,j}-\alpha_{i,j}) \tag{17.312}$$

$$F^{y}_{i,j+\frac{1}{2}} = -\kappa_H \Delta x_{i,j+\frac{1}{2}}\frac{1}{\Delta y_{i,j+\frac{1}{2}}}(\alpha_{i,j+1}-\alpha_{i,j}), \tag{17.313}$$

as in Section 11.2.1.

The grid widths (shown in Fig. 3.5) are given by

$$\Delta x_{i+\frac{1}{2},j} = (\texttt{dx\_br})_{i,j}+(\texttt{dx\_bl})_{i,j} \tag{17.314}$$

$$\Delta y_{i+\frac{1}{2},j} = (\texttt{dy\_br})_{i,j}+(\texttt{dy\_tr})_{i,j-1} \tag{17.315}$$

$$\Delta x_{i,j+\frac{1}{2}} = (\texttt{dx\_tr})_{i-1,j}+(\texttt{dx\_tl})_{i,j} \tag{17.316}$$

$$\Delta y_{i,j+\frac{1}{2}} = (\texttt{dy\_bl})_{i,j}+(\texttt{dy\_tl})_{i,j}. \tag{17.317}$$

17.13.3 Momentum equation

Specific forms of discretization for properties related to internal stress are given here.

The strain rate tensor ($\dot{\epsilon}$) and stress tensor (σ) are defined at tracer points (Figure 17.8).

Components (divergence, tension, and shear) of the strain rate tensor are expressed in a discretized form as follows:

$$
\begin{aligned}
(D_D)_{i,j} = {} & \frac{1}{\Delta x_{i,j}\Delta y_{i,j}}\Big(\frac{\Delta y_{i+\frac{1}{2},j}}{2}(u_{I\,i+\frac{1}{2},j+\frac{1}{2}} + u_{I\,i+\frac{1}{2},j-\frac{1}{2}}) - \frac{\Delta y_{i-\frac{1}{2},j}}{2}(u_{I\,i-\frac{1}{2},j+\frac{1}{2}} + u_{I\,i-\frac{1}{2},j-\frac{1}{2}}) \\
& + \frac{\Delta x_{i,j+\frac{1}{2}}}{2}(v_{I\,i+\frac{1}{2},j+\frac{1}{2}} + v_{I\,i-\frac{1}{2},j+\frac{1}{2}}) - \frac{\Delta x_{i,j-\frac{1}{2}}}{2}(v_{I\,i+\frac{1}{2},j-\frac{1}{2}} + v_{I\,i-\frac{1}{2},j-\frac{1}{2}})\Big),
\end{aligned}
$$

$$
\begin{aligned}
(D_T)_{i,j} = {} & \frac{1}{2}\Big[\frac{\Delta y_{i,j+\frac{1}{2}}}{\Delta x_{i,j+\frac{1}{2}}}\Big(\frac{u_{I\,i+\frac{1}{2},j+\frac{1}{2}}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{u_{I\,i-\frac{1}{2},j+\frac{1}{2}}}{\Delta y_{i-\frac{1}{2},j+\frac{1}{2}}}\Big) + \frac{\Delta y_{i,j-\frac{1}{2}}}{\Delta x_{i,j-\frac{1}{2}}}\Big(\frac{u_{I\,i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y_{i+\frac{1}{2},j-\frac{1}{2}}} - \frac{u_{I\,i-\frac{1}{2},j-\frac{1}{2}}}{\Delta y_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big] \\
& - \frac{1}{2}\Big[\frac{\Delta x_{i+\frac{1}{2},j}}{\Delta y_{i+\frac{1}{2},j}}\Big(\frac{v_{I\,i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{v_{I\,i+\frac{1}{2},j-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j-\frac{1}{2}}}\Big) + \frac{\Delta x_{i,j-\frac{1}{2}}}{\Delta y_{i,j-\frac{1}{2}}}\Big(\frac{v_{I\,i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{i-\frac{1}{2},j+\frac{1}{2}}} - \frac{v_{I\,i-\frac{1}{2},j-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big],
\end{aligned}
$$

$$
\begin{aligned}
(D_S)_{i,j} = {} & \frac{1}{2}\Big[\frac{\Delta y_{i,j+\frac{1}{2}}}{\Delta x_{i,j+\frac{1}{2}}}\Big(\frac{v_{I\,i+\frac{1}{2},j+\frac{1}{2}}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{v_{I\,i-\frac{1}{2},j+\frac{1}{2}}}{\Delta y_{i-\frac{1}{2},j+\frac{1}{2}}}\Big) + \frac{\Delta y_{i,j-\frac{1}{2}}}{\Delta x_{i,j-\frac{1}{2}}}\Big(\frac{v_{I\,i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y_{i+\frac{1}{2},j-\frac{1}{2}}} - \frac{v_{I\,i-\frac{1}{2},j-\frac{1}{2}}}{\Delta y_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big] \\
& + \frac{1}{2}\Big[\frac{\Delta x_{i+\frac{1}{2},j}}{\Delta y_{i+\frac{1}{2},j}}\Big(\frac{u_{I\,i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{u_{I\,i+\frac{1}{2},j-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j-\frac{1}{2}}}\Big) + \frac{\Delta x_{i,j-\frac{1}{2}}}{\Delta y_{i,j-\frac{1}{2}}}\Big(\frac{u_{I\,i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{i-\frac{1}{2},j+\frac{1}{2}}} - \frac{u_{I\,i-\frac{1}{2},j-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big].
\end{aligned}
$$

The internal stress is defined at velocity points and computed from stress tensor as follows:

$$
\begin{aligned}
(F_\mu)_{i+\frac{1}{2},j+\frac{1}{2}} = {} & \frac{1}{2}\Big[\frac{1}{2}\Big(\frac{(\sigma_1)_{i+1,j+1} + (\sigma_1)_{i+1,j} - (\sigma_1)_{i,j+1} - (\sigma_1)_{i,j}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\Big) \\
& + \frac{1}{2}\Big(\frac{\Delta y^2_{i+1,j+\frac{1}{2}}[(\sigma_2)_{i+1,j+1} + (\sigma_2)_{i+1,j}] - \Delta y^2_{i,j+\frac{1}{2}}[(\sigma_2)_{i,j+1} + (\sigma_2)_{i,j}]}{\Delta y^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\Big) \\
& + \Big(\frac{\Delta x^2_{i+\frac{1}{2},j+1}[(\sigma_{12})_{i+1,j+1} + (\sigma_{12})_{i,j+1}] - \Delta x^2_{i+\frac{1}{2},j}[(\sigma_{12})_{i+1,j} + (\sigma_{12})_{i,j}]}{\Delta x^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}\Big),
\end{aligned}
\tag{17.318}
$$

$$
\begin{aligned}
(F_\psi)_{i+\frac{1}{2},j+\frac{1}{2}} = {} & \frac{1}{2}\Big[\frac{1}{2}\Big(\frac{(\sigma_1)_{i+1,j+1} + (\sigma_1)_{i,j+1} - (\sigma_1)_{i+1,j} - (\sigma_1)_{i,j}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}\Big) \\
& - \frac{1}{2}\Big(\frac{\Delta x^2_{i+\frac{1}{2},j+1}[(\sigma_2)_{i+1,j+1} + (\sigma_2)_{i,j+1}] - \Delta x^2_{i+\frac{1}{2},j}[(\sigma_2)_{i+1,j} + (\sigma_2)_{i,j}]}{\Delta x^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}\Big) \\
& + \Big(\frac{\Delta y^2_{i+1,j+\frac{1}{2}}[(\sigma_{12})_{i+1,j+1} + (\sigma_{12})_{i+1,j}] - \Delta y^2_{i,j+\frac{1}{2}}[(\sigma_{12})_{i,j+1} + (\sigma_{12})_{i,j}]}{\Delta y^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\Big).
\end{aligned}
\tag{17.319}
$$

17.14 Usage

17.14.1 Compilation

The information needed to compile the sea ice model with the OGCM should be given to `configure.in`. At least, `ICE` option should be specified in the `OPTIONS` line and the number of categories of the sea ice thickness should be specified as `NUM_ICECAT`, for example, a line `NUM_ICECAT = 5` should be inserted somewhere in `configure.in`. The model options related to the sea ice model are listed on Table 17.5.

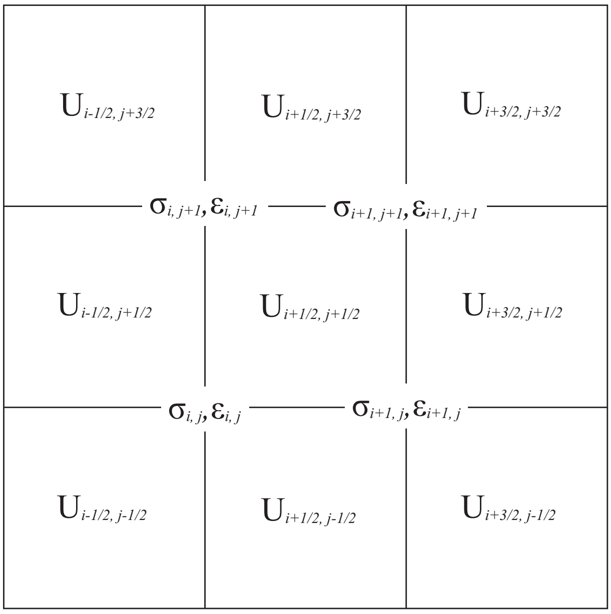

Figure17.8 Position of variables used by dynamics scheme.

Table17.5 Model options related to the sea ice model

option name	description	usage
ICE	Sea ice model is included to the OGCM	
SIDYN	Dynamics of sea ice is solved	otherwise, sea ice drifts with a third of the surface ocean velocity.
CALALBSI	An albedo scheme of CICE is used	otherwise constant
HISTICECAT	Monitor time evolution of basic state variables	Use NAMELIST.*name_model*.MONITOR to specify the output group and interval. *name_model* is the name of the model specified by configure.in
ICEFULLMONIT	Activate extensive monitor	Use NAMELIST.*name_model*.MONITOR to specify the output group and interval

17.14.2 Job parameters (namelist)

The runtime job parameters (namelist) that should be given by NAMELIST.*name_model* are listed on Tables 17.6 to 17.15. Job parameters related to monitoring are specified in the next subsection.

a. Thickness category

The bounds of thickness categories, whose number should be specified by NUM_ICECAT in configure.in before compilation, should be given at run time to namelist nml_seaice_thickness_category, which is explained in Table 17.6.

Table17.6 Namelist of thickness category of the sea ice model (nml_seaice_thickness_category)

variable name	dimension	description	usage (default value)
hbound(0 : ncat)	m	category boundary	required

Continued on next page

Table 17.6 – continued from previous page

variable name	dimension	description	usage (default value)
lsicat_volchk	logical	flag for checking mass conservation	default = .false.

b. Time

The parameters related to time should be given by nml_seaice_time_conf and nml_seaice_time_calendar, which are explained in Table 17.7. Note that the specification of nml_seaice_time_conf is not mandatory, necessary information will be taken from the ocean model.

Table17.7 Namelist related to time of the sea ice model

variable name	group	description	usage (default value)
nstep_seaice	nml_seaice_time_conf	time steps to be proceed	same as OGCM
ibyri	nml_seaice_time_conf	start year of this run	same as OGCM
ibmni	nml_seaice_time_conf	start month of this run	same as OGCM
ibdyi	nml_seaice_time_conf	start day of this run	same as OGCM
ibhri	nml_seaice_time_conf	start hour of this run	same as OGCM
ibmii	nml_seaice_time_conf	start minute of this run	same as OGCM
ibsci	nml_seaice_time_conf	start second of this run	same as OGCM
l_force_leap	nml_seaice_calendar	forcibly specify leap year at run time with l_leap	.false.
l_leap	nml_seaice_calendar	specification of whether this year is leap or not (valid when l_force_leap = .true.)	.false.

c. Dynamics

The parameters related to dynamics of sea ice (SIDYN option) should be given to namelist nml_seaice_dyn, which is listed on Table 17.8

Table17.8 Namelist related to dynamics of the sea icel (nml_seaice_dyn)

variable name	dimension	description	usage (default value)
dt_seaice_dyn_sec	sec	time step interval for dynamics (in seconds)	about a tenth of the baroclinic time step of the ocean model
damp_vice	m sec^{-1}	Rayleigh damping coefficient for rapid velocity	0.0
damp_vmax	m sec^{-1}	minimum velocity for damping	1.0
prs_scale_factor	N m^{-2}	pressure scaling factor	2.75×10^4
e_fold_const_ice_prs	1	e-folding constant for ice pressure	20

d. Diffusion

Coefficient for the harmonic-type horizontal diffusion should be given to namelist nml_seaice_diff, which is explained in Table 17.9.

Table17.9 Namelist of diffusion of the sea ice model (nml_seaice_diff)

variable name	dimension	description	usage (default value)
diff_seaice_m2ps	$\mathrm{m}^2\ \mathrm{sec}^{-1}$	horizontal diffusion coefficient	required

e. Minimum fractional area

Minimum of fractional area applied to all thickness categories may be given at run time to namelist nml_seaice_model, which is explained in Table 17.10.

Table17.10 Namelist of minimum fractional area of the sea ice model (nml_seaice_model)

variable name	dimension	description	usage (default value)
concentration_min	1	minimum fractional area	1×10^{-4}

f. Air-ice transfer

The bulk transfer coefficient at the air - sea ice interface should be given to nml_air_ice, which is listed on Table 17.11.

Table17.11 Namelist of air-ice bulk transfer coefficient (nml_air_ice)

variable name	units	description	usage (default value)
c_drag_ai	1	air-ice exchange coefficient for wind	3.0×10^{-3}
c_transfer_sensible_heat_ai	1	air-ice exchange coefficient for sensible heat	1.5×10^{-3}
c_transfer_latent_heat_ai	1	air-ice exchange coefficient for evaporation	1.5×10^{-3}

g. Ice-ocean transfer

The bulk transfer coefficient at the ice - ocean interface should be given to nml_ice_ocean, which is listed on Table 17.12. If BULKECMWF is used, user must specify nml_bulkecmwf_seaice_run to control restart input and output for vertical velocity due to convection, $w_{\dagger}$, which is used to calculate the wind velocity in the lowest layer of the atmosphere model.

Table17.12 Namelist of ice-ocean bulk transfer coefficient (nml_ice_ocean)

variable name	dimension	description	usage (default value)
c_drag_io	1	ice-ocean exchange coefficient for wind	5.5×10^{-3}
turning_angle_drag_io	degree	turning angle for ice-ocean drag	-999 (set to $\pm cz$, where c: $1°\mathrm{m}^{-1}$, z: the first layer depth [m]; positive/negative in the northern/southern hemisphere)

Table17.13 Namelist nml_bulkecmwf_seaice_run)

variable name	units	description	usage
l_rst_bulkecmwf_wstar_in	logical	read initial restart for the free convection velocity scale over sea ice or not	default = l_rst_seaice_in

h. Restart

How to handle restart files is specified by namelists nml_seaice_run (Table 17.14), and nml_restart with name = ”Sea Ice”.

Table17.14 Namelist related to restart of the sea ice model

variable name	group	usage (default value)
l_rst_seaice_in	nml_seaice_run	.true. : the initial state is read from restart file .false.(default) : start from the state without sea ice

i. Albedo

When CALALBSI option is selected, an albedo scheme of CICE is used. The parameters of this scheme should be given by namelist nml_albedo_seaice, which is listed on Table 17.15.

Table17.15 Namelist for the albedo of sea ice used by CALALBSI option (nml_albedo_seaice)

variable name	dimension	description	usage (default value)
alb_ice_visible_t0		visible ice albedo for thicker ice	0.78
alb_ice_nearIR_t0		near infrared ice albedo for thicker ice	0.36
alb_snw_visible_t0		visible, cold snow albedo	0.98
alb_snw_nearIR_t0		near infrared, cold snow albedo	0.70
alb_ice_visible_dec_ratio	$(°C)^{-1}$	visible ice albedo declination rate	0.075
alb_ice_nearIR_dec_ratio	$(°C)^{-1}$	near infrared ice albedo declination rate	0.075
alb_snw_visible_dec_ratio	$(°C)^{-1}$	visible snow albedo declination rate	0.10
alb_snw_nearIR_dec_ratio	$(°C)^{-1}$	near infrared ice albedo declination rate	0.15
hi_ref	m	the maximum ice thickness to which connection function is used	0.50
atan_ref		the base value of the tangent hyperbolic connection function	4.0
tsfci_t0	°C	the temperature at which ice albedo is equated to that of ocean	0.0
tsfci_t1	°C	the temperature at which ice albedo is started to decline to that of the ocean	−1.0
fsnow_patch	meter	thickness of snow patch on melting bare ice	0.02

j. Emissivity

The emissivity of sea ice surface and snow surface should be given to nml_seaice_emissivity, which is listed on Table 17.16.

Table17.16 Namelist of emissivity of sea ice surface and snow surface (nml_seaice_emissivity)

variable name	dimension	description	usage (default value)
emissivity_sea_ice	1	Emissivity of sea ice surface	1
emissivity_snow	1	Emissivity of snow surface	1

17.14.3 Monitoring

To monitor or sample the state of sea ice, HISTICECAT option must be specified. Extensive monitoring is activated by further specifying ICEFULLMONIT option.

The general behavior should be specified by the nml_seaice_budget and nml_seaice_hst in NAMELIST.*name_model*, which is explained in Table 17.17.

Table17.17 Namelist of monitor of the sea ice model

variable name	group	description	usage
nstep_si_budget_interval	nml_seaice_budget	the interval of time step by which water budget is written	0 : none, −1 : monthly
undef_value	nml_seaice_hst	undefined value to fill ice free grid point	real(8)
undef_for_land	nml_seaice_hst	undefined value to fill land grid points	real(8)

Sampling variables are grouped according to the processes they are involved, the grid points they are defined, and the priorities given by Sea Ice Model Intercomparison Project (SIMIP) (Notz et al., 2016).

The groups are listed on Table 17.18. Users specify nml_history blocks in file NAMELIST.*name_model*.MONITOR in the following manner.

Specification of NAMELIST.*name_model*.MONITORforseaicemonitoring

```
&nml_history
  name          = 'sea ice basic state vars at t-point',
  file_base     = 'result/hs_ice_t_sta1',
  interval_step = 12,
  suffix        = 'minute',
&end
```

Table17.18 List of groups for sea ice monitoring

group name	usage note
sea ice basic state vars at t-point	"mean" is forced
sea ice state vars at t-point	"mean" is forced
tendencies of sea-ice mass and area fraction at t-point	"mean" is forced
heat and freshwater fluxes over sea ice fraction at t-point	"mean" is forced
sea ice state vars of each category at t-point	"mean" is forced
sea ice basic dynamics at u-point	"mean" is forced
sea ice dynamics at u-point	"mean" is forced
sea ice dynamics at t-point	"mean" is forced
sea ice vars at x-point	
sea ice vars at y-point	
sea ice integrated measures	
sea ice dynamics at u-point priority 3	valid if ICEFULLMONIT, "mean" is forced
sea ice dynamics at t-point priority 3	valid if ICEFULLMONIT, "mean" is forced
sea ice fluxes of each category at t-point	valid if ICEFULLMONIT
heat and freshwater fluxes over open leads at t-point	valid if ICEFULLMONIT

17.15 Appendix

17.15.1 Partial derivative of specific humidity with respect to temperature

In MRI.COM, surface temperature of ice (T_3) is computed using semi-implicit method, where an expression for the partial derivative of the saturation specific humidity with respect to temperature (T) is needed.

Properties of moist air used by MRI.COM is based on Gill (1982) and replicated in Section 14.12.2. Using (14.122), the saturation specific humidity (q_i) is given as

$$q_i = \epsilon e'_{si}/(p_s - (1-\epsilon)e'_{si}). \tag{17.320}$$

where saturation partial pressure of moist air over ice (e'_{si}) is given by (14.128) and (14.129). Using (17.320),

$$\frac{\partial q_i}{\partial T} = \frac{\epsilon p_s}{\{p_s - (1-\epsilon)e'_{si}\}^2}\frac{\partial e'_{si}}{\partial T}, \tag{17.321}$$

where $\partial e'_{si}/\partial T$ is expressed by setting $f_w = 1$ as

$$\frac{\partial e'_{si}}{\partial T} = \ln 10 \cdot 10^{g(T)} \cdot g'(T), \tag{17.322}$$

where $g(T) = (0.7859 + 0.03477T)/(1 + 0.00412T) + 0.00422T$ and $g'(T) = \partial g(T)/\partial T$, where T is temperature at the upper surface of sea ice in °C.

Chapter 18

Tide model

This chapter explains introduction of a tidal model. MRI.COM uses the scheme of Sakamoto et al. (2013), which runs a barotropic tide model with the same grid and topography as the barotropic model (Chapter 7) in parallel, and incorporates tidal effects into OGCM at each baroclinic time step. We describe derivation of the governing equations of the tide model (Sec. 18.1), incorporation of tidal effects into OGCM (Sec. 18.2), calculation of the tidal forcing potential (Sec. 18.3), execution of nesting experiments (Sec. 18.4), and usage (Sec. 18.5).

18.1 Governing equations

Adding the tide-related terms directly to the equation of motion of the barotropic model, Eqs. (7.1) and (7.2), gives: (e.g., Schiller, 2004)

$$\frac{\partial \mathbf{U}}{\partial t} + f\mathbf{k}\times\mathbf{U} = -g(\eta + H)\nabla(\eta - \beta\eta_{\mathrm{eq}} - \eta_{\mathrm{SAL}}) + \mathbf{F}^{\mathrm{horz}} + \frac{\tau^{\mathrm{btm}}}{\rho_0} + \mathbf{X}', \tag{18.1}$$

where η_{eq} is equilibrium tide, or astronomical tide-generating potential (18.3), β defines effect of tide-generating potential and correction due to earth tides, and η_{SAL} represents the self-attraction and loading of the ocean tides. (In MRI.COM, we usually set $\beta = 0.7$, though in general $\beta = 1 + k - h$ with k and h being Love numbers.) The term $\mathbf{F}^{\mathrm{horz}}$ is the vertically integrated horizontal viscosity parameterization terms, and τ^{btm} is the bottom friction terms that may be specialized for tidal currents. They are separated from $\mathbf{X}$ defined by (7.3) and (7.4) and $\mathbf{X}' \equiv \mathbf{X} - \mathbf{F}^{\mathrm{horz}} - \tau^{\mathrm{btm}}/\rho_0$. Note also that atmospheric pressure term is dropped for brevity. The continuity equation is same as Eq. (7.5):

$$\frac{\partial \eta}{\partial t} + \nabla\cdot\mathbf{U} = F_w, \tag{18.2}$$

where $F_w \equiv P - E + R + I$ is surface fresh water forcing that will be enter the system through the basic fields.

As Eq. (18.1) indicates, the terms specialized for tides affect the basic fields unintentionally. For example, the self-attraction and loading term will modify the relationships between the sea surface height gradient and the barotropic currents. Horizontal viscosity and bottom friction will also change the results, if they are modified to be suitable for tidal modeling. As a result, the model solution without tidal potential (i.e. $\eta_{\mathrm{eq}} = 0$) is no longer the same as that of the OGCM barotropic model. To resolve this problem, our tide scheme calculates the tidal fields separately from the basic fields within the barotropic submodel as explained by Sakamoto et al. (2013). The essence of the solution method is explained below.

Sakamoto et al. (2013) derived the governing equations of the barotropic tidal model, focusing on smooth coupling with an OGCM. First, the variables are decomposed into the basic and linear tidal components,

$$\mathbf{U} = \mathbf{U}_b + \mathbf{U}_{lt} \tag{18.3}$$

$$\eta = \eta_b + \eta_{lt} \tag{18.4}$$

$$\mathbf{F}^{\mathrm{horz}} = \mathbf{F}_b^{\mathrm{horz}} + \mathbf{F}_{lt}^{\mathrm{horz}} \tag{18.5}$$

$$\tau^{\mathrm{btm}} = \tau_b^{\mathrm{btm}} + \tau_{lt}^{\mathrm{btm}}, \tag{18.6}$$

where "b" and "lt" refer to the basic and linear tidal components, respectively. Now we decompose (18.1) and (18.2) to obtain governing equations for each of the two components.

For linear tidal component:

$$\frac{\partial \mathbf{U}_{lt}}{\partial t} + f\mathbf{k}\times\mathbf{U}_{lt} = -g(\eta + H)\nabla(\eta_{lt} - \beta\eta_{\mathrm{eq}} - \eta_{\mathrm{SAL}}) + \mathbf{F}_{lt}^{\mathrm{horz}} + \frac{\tau_{lt}^{\mathrm{btm}}}{\rho_0}, \tag{18.7}$$

$$\frac{\partial \eta_{lt}}{\partial t} + \nabla\cdot\mathbf{U}_{lt} = 0, \tag{18.8}$$

where we assume that self-attraction and loading acts only to linear tidal component in such a way as $\eta_{\rm SAL} = (1-\alpha)\eta_{lt}$.

For the basic component:

$$\frac{\partial \mathbf{U}_b}{\partial t} + f\mathbf{k}\times\mathbf{U}_b = -g(\eta+H)\nabla\eta_b + \mathbf{F}_b^{\rm horz} + \frac{\tau_b^{\rm btm}}{\rho_0} + \mathbf{X}', \tag{18.9}$$

$$\frac{\partial \eta_b}{\partial t} + \nabla\cdot\mathbf{U}_b = F_w. \tag{18.10}$$

This equation is basically same as the barotropic model written in Chapter 7.

In the above decomposition, the linear tidal component represents only the linear response to tidal forcing, $\eta_{\rm eq}$, and the secondary effects of tide, such as tidal advection and internal tides, are represented by the basic component. The linear terms in the barotropic equations, such as the Coriolis force and the Laplacian horizontal viscosity with a constant viscosity coefficient, may be split into the basic and linear tidal components naturally.

The bottom friction term ($\tau^{\rm btm}$) is non-linear and should be treated carefully. The bottom friction term (see Section 8.3.6 for details) is expressed as

$$\tau^{\rm btm} = -\rho_0 C_D|\mathbf{u}|\mathbf{T}_\theta\mathbf{u}, \tag{18.11}$$

where C_D is a drag coefficient and $\mathbf{T}_\theta$ is a matrix representing horizontal veering with an angle of θ,

$$\mathbf{T}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{18.12}$$

Sum of the two components are used for the scalar part ($|\mathbf{u}|$) and only the vector part is decomposed:

$$\tau_b^{\rm btm} = -\rho_0 C_D|\mathbf{u}_b+\mathbf{u}_{lt}|\mathbf{T}_\theta\mathbf{u}_b, \tag{18.13}$$

$$\tau_{lt}^{\rm btm} = -\rho_0 C_D|\mathbf{u}_b+\mathbf{u}_{lt}|\mathbf{T}_\theta\mathbf{u}_{lt}. \tag{18.14}$$

Similarly, when the horizontal viscosity parameterization is non-linear as in the Smagorinsky scheme, the coefficient part (e.g., ν_H in the equations below) is calculated by using the full velocity, while the decomposed fields are applied to the viscosity operator. In MRI.COM, the horizontal tension D_T and shear D_S of the velocity field,

$$D_T = h_\psi\frac{\partial}{h_\mu\partial\mu}\left(\frac{u}{h_\psi}\right) - h_\mu\frac{\partial}{h_\psi\partial\psi}\left(\frac{v}{h_\mu}\right), \tag{18.15}$$

$$D_S = h_\mu\frac{\partial}{h_\psi\partial\psi}\left(\frac{u}{h_\mu}\right) + h_\psi\frac{\partial}{h_\mu\partial\mu}\left(\frac{v}{h_\psi}\right), \tag{18.16}$$

are used to calculate horizontal viscosity as

$$F_u^{\rm horz} = \frac{1}{h_\psi^2}\frac{\partial}{h_\mu\partial\mu}\left(h_\psi^2\nu_H D_T\right) + \frac{1}{h_\mu^2}\frac{\partial}{h_\psi\partial\psi}\left(h_\mu^2\nu_H D_S\right), \tag{18.17}$$

$$F_v^{\rm horz} = \frac{1}{h_\psi^2}\frac{\partial}{h_\mu\partial\mu}\left(h_\psi^2\nu_H D_S\right) - \frac{1}{h_\mu^2}\frac{\partial}{h_\psi\partial\psi}\left(h_\mu^2\nu_H D_T\right). \tag{18.18}$$

Both of horizontal viscosity of the basic component, $\mathbf{F}_b^{\rm horz}$, and that of the linear tidal component, $\mathbf{F}_{lt}^{\rm horz}$, are calculated by Eqs. (18.17) and (18.18) using common ν_H. However, D_T and D_S are calculated for $\mathbf{u}_b$ and $\mathbf{u}_{lt}$, respectively.

By the above decomposition, the effect of the tidal model can be taken in while using the original barotropic model. In fact, if $\eta_{\rm eq}\equiv 0$, then $\eta_{\rm lt}\equiv 0, \mathbf{U}_{lt}\equiv \mathbf{0}$, and the above equations return to the original ones.

By formulation, $\mathbf{F}^{\rm horz}$, $\tau^{\rm btm}$, and $\mathbf{X}'$ are computed at the baroclinic time level of n and kept fixed during the integration of barotropic submodel. However, when the barotropic field varies significantly during the integration, which is expected when tidal forcing is included, $\mathbf{F}^{\rm horz}$ and $\tau^{\rm btm}$ should be allowed to vary with the evolution of barotropic velocity field. In this case, $\mathbf{F}_{lt}^{\rm horz}$ and $\tau_{lt}^{\rm btm}$ in (18.7) may be replaced by the viscosity parameterization suitable to the tidal velocity fields (represented by $\mathbf{F}_{tide}^{\rm horz}$ and $\tau_{tide}^{\rm btm}$, respectively). Then, (18.7) becomes,

$$\frac{\partial \mathbf{U}_{lt}}{\partial t} + f\mathbf{k}\times\mathbf{U}_{lt} = -g(\eta+H)\nabla(\alpha\eta_{lt}-\beta\eta_{\rm eq}) + \mathbf{F}_{tide}^{\rm horz} + \frac{\tau_{tide}^{\rm btm}}{\rho_0}. \tag{18.19}$$

There is no strong restriction on the form of the viscosity parameterization applied to the tidal velocity fields. However, by using a common form for both baroclinic and barotropic fields, depth integrated viscosity terms would cancel when barotropic and baroclinic modes are synchronous (see (Eq. 18.23)). In MRI.COM, the bottom friction for the linear tidal component is calculated as follows:

$$\tau^{\mathrm{btm}}_{tide} = -\rho_0 C_D \left|\frac{\mathbf{U}_b + \mathbf{U}_{lt}}{\eta + H}\right| \mathbf{T}_\theta \frac{\mathbf{U}_{lt}}{\eta + H}. \tag{18.20}$$

Horizontal viscosity parameterization is the Laplacian operator with the viscosity coefficient fixed in time.

The space-time finite difference of the tidal model is omitted, since it is the same as that of the barotropic model. A sequence of operations explained in Section 7.4.1 is applied to both basic and tidal components. But now we use surface height at time level $t = t_n$ as the height of column applied to the pressure gradient term during the barotropic subcycle.

18.2 Incorporation of the tidal effects into MRI.COM

The solution method to predict the two components of the barotropic mode is schematically illustrated by Figure 18.1. On starting the barotropic submodel, the total barotropic mode is split into the basic and the linear tidal components (U_b, η_b, U_{lt}, and η_{lt}) and the time evolution is calculated separately. A simple sum of the two modes is returned to the main part of OGCM. For the next step, the total and the linear tidal components are retained. The basic component is calculated by subtracting the tidal part from the total.

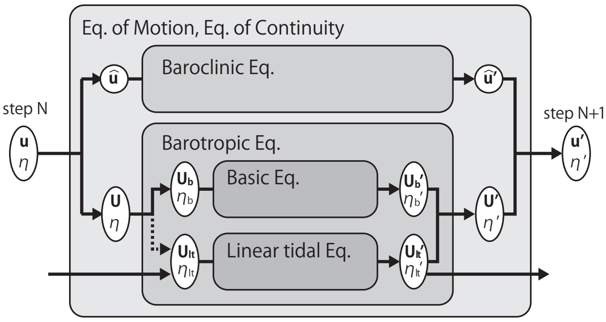

Figure18.1 Schematic figure of the separating basic and linear tidal component of the barotropic mode

The weighted average of the barotropic fields is written as follows:

$$\begin{aligned}\langle \mathbf{U}_{lt}\rangle^{n+1} =& \mathbf{U}^0_{lt} - \Delta t_{\mathrm{cl}} f\mathbf{k}\times\langle\langle \mathbf{U}_{lt}\rangle\rangle^{n+\frac{1}{2}} - \Delta t_{\mathrm{cl}} g(\eta^n + H)\Big\langle\Big\langle \nabla(\alpha\eta_{lt} - \beta\eta_{\mathrm{eq}})\Big\rangle\Big\rangle^{n+\frac{1}{2}} \\ &+ \Delta t_{\mathrm{cl}}\langle\langle \mathbf{F}^{\mathrm{horz}}_{tide}\rangle\rangle^{n+\frac{1}{2}} + \frac{1}{\rho_0}\Delta t_{\mathrm{cl}}\langle\langle \tau^{\mathrm{btm}}_{tide}\rangle\rangle^{n+\frac{1}{2}},\end{aligned} \tag{18.21}$$

$$\langle \mathbf{U}_b\rangle^{n+1} = \mathbf{U}^0_b - \Delta t_{\mathrm{cl}} f\mathbf{k}\times\langle\langle \mathbf{U}_b\rangle\rangle^{n+\frac{1}{2}} - \Delta t_{\mathrm{cl}} g(\eta^n + H)\Big\langle\Big\langle \nabla\eta_b\Big\rangle\Big\rangle^{n+\frac{1}{2}} + \Delta t_{\mathrm{cl}}\Big(\mathbf{X} - \mathbf{F}^{\mathrm{horz}}_{lt} - \frac{\tau^{\mathrm{btm}}_{lt}}{\rho_0}\Big)^n. \tag{18.22}$$

These are summed and then combined with the baroclinic part (7.49) and (7.50) to give

$$\begin{aligned}\frac{(\langle \mathbf{u}\rangle^{n+1} + \mathbf{u}'_{k-\frac{1}{2}} - \overline{\mathbf{u}'}^z)\Delta z^{n+1}_{k-\frac{1}{2}} - \mathbf{u}^n_{k-\frac{1}{2}}\Delta z^n_{k-\frac{1}{2}}}{\Delta t_{\mathrm{cl}}} =& - f\mathbf{k}\times\langle\langle \mathbf{u}\rangle\rangle^{n+\frac{1}{2}}\Delta z^{n+1}_{k-\frac{1}{2}} - f\mathbf{k}\times[\mathbf{u}_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + f\mathbf{k}\times\overline{[\mathbf{u}]^{n+\frac{1}{2}}}^z\Delta z^{n+1}_{k-\frac{1}{2}} \\ &- g\Delta z^{n+1}_{k-\frac{1}{2}}\frac{\eta^n + H}{\eta^{n+1} + H}\Big\langle\Big\langle \nabla(\eta_b + \alpha\eta_{lt} - \beta\eta_{\mathrm{eq}})\Big\rangle\Big\rangle^{n+\frac{1}{2}} \\ &+ \Delta z^{n+1}_{k-\frac{1}{2}}\frac{\langle\langle \mathbf{F}^{\mathrm{horz}}_{tide}\rangle\rangle^{n+\frac{1}{2}} - (\mathbf{F}^{\mathrm{horz}}_{lt})^n}{\eta^{n+1} + H} + \Delta z^{n+1}_{k-\frac{1}{2}}\frac{\langle\langle \tau^{\mathrm{btm}}_{tide}\rangle\rangle^{n+\frac{1}{2}} - (\tau^{\mathrm{btm}}_{lt})^n}{\rho_0(\eta^{n+1} + H)} + \Delta z^{n+1}_{k-\frac{1}{2}}\mathbf{F}^n.\end{aligned} \tag{18.23}$$

18.3 Tidal potential

The tide producing potential η_{eq} of MRI.COM includes the eight primary constituents that consist of the four largest equilibrium tides of the diurnal and semi-diurnal species (K_1, O_1. P_1, Q_1, M_2, S_2, N_2, K_2). Following Schwiderski

Table18.1 Constants of major tidal modes

n	Tidal Mode	K, m	σ, 10^{-4} /sec	χ, deg
1	K_1, declination luni-solar	0.141565	0.72921	$h_0 + 90$
2	Q_1, principal lunar	0.100514	0.67598	$h_0 - 2s_0 + 90$
3	P_1, principal solar	0.046843	0.72523	$-h_0 - 90$
4	Q_1, elliptical lunar	0.019256	0.64959	$h_0 - 3s_0 + p_0 - 90$
5	M_2, principal lunar	0.242334	1.40519	$2h_0 - 2s_0$
6	S_2, principal solar	0.112841	1.45444	0
7	N_2, elliptical lunar	0.046398	1.37880	$2h_0 - 3s_0 + p_0$
8	K_2, declination luni-solar	0.030704	1.45842	$2h_0$

Here h_0, s_0, and p_0 are the mean longitudes of the sun and moon and the lunar perigee at Greenwich midnight: $h_0 = 279.69668 + 36000.768930485T + 3.03 \times 10^4 T^2$, $s_0 = 270.434358 + 481267.88314137T - 0.001133T^2 + 1.9 \times 10^{-6} T^3$, $p_0 = 334.329653 + 4069.0340329575T - 0.010325T^2 - 1.2 \times 10^{-5} T^3$, where $T = (27392.500528 + 1.0000000356D)/36525$, $D = d + 365(y - 1975) + \mathrm{Int}[(y - 1973)/4]$, d is the day numbe of the year ($d = 1$ for January 1), $y \geq 1975$ is the year number, and Int$[x]$ is the integral part of x.

(1980), we write the n-th ($n = [1, 4]$) diurnal equilibrium tide as

$$\begin{aligned}\eta_{\mathrm{eq},n} &= K_n \cos^2\phi \cos(\sigma_n t + \chi_n + 2\lambda) \\ &= K_n \cos^2\phi [\cos(\sigma_n t + \chi_n) \cos 2\lambda - \sin(\sigma_n t + \chi_n) \sin 2\lambda],\end{aligned} \tag{18.24}$$

and the n-th semi-diurnal component ($n = [5, 8]$) as

$$\begin{aligned}\eta_{\mathrm{eq},n} &= K_n \sin 2\phi \cos(\sigma_n t + \chi_n + \lambda) \\ &= K_n \sin 2\phi [\cos(\sigma_n t + \chi_n) \cos \lambda - \sin(\sigma_n t + \chi_n) \sin \lambda].\end{aligned} \tag{18.25}$$

The meaning of the mathematical symbols is as follows:

t	universal standard time in seconds
λ	east longitude
ϕ	latitude
K	amplitude of partial equilibrium tide in meters
σ	frequency of partial equilibrium tide in sec^{-1}
χ	astronomical argument of partial equilibrium tide relative to Greenwich midnight

And constants are listed on Table 18.1.

18.4 Nesting experiment with tide

The tide scheme can work together with the nesting schemes in MRI.COM. This subsection describes briefly how the tide scheme works with on-line nesting and off-line one-way nesting methods. See Chapter 22 for the general explanation of the nesting schemes.

For on-line nesting, the tide models communicate each other between the parent low-resolution model and the child high-resolution model in order to predict tides at the next step in the same manner as the barotropic model. The lateral boundaries of the child model receive $\mathbf{U}_{lt}$ and η_{lt} from the parent model every steps of the barotropic mode in the both cases of one-way and two-way nesting. On the other hand, in the case of two-way nesting with the "replace" configuration, the internal region of the parent model receives them from the child model.

For off-line one-way nesting, the communication process of the tide model is also the same as that of the barotropic model. The parent model saves $\mathbf{U}_{lt}$ and η_{lt} on bands corresponding to the lateral boundaries of the child model. The child model reads the files and interpolates them spatially and temporally to the lateral boundaries every barotropic steps. It should be noted that other lateral boundary data output by the parent model, such as three-dimensional velocity, sea surface height and x and y transports, also includes the linear tidal component. That is, the parent model saves $\mathbf{u}$, $\mathbf{U}_b + \mathbf{U}_{lt}$ and $\eta_b + \eta_{lt}$.

As a special function for off-line one-way nesting, the child model can be executed under tidal forcing even when the parent model does not include tides. That is, users can run the parent model without `TIDE`, while the child model with it. In this case, in advance, users have to create files for the child model boundaries of $\mathbf{U}_{lt}$ and η_{lt} in the same format as the parent model output. The MXE (see Section 25.6) package offers a tool for it based on the Matsumoto et al. (2000)'s dataset in the directory prep/nest/offline/. In addition, boundary files output by the parent model do not include the tidal component, which is different from the case that the parent model includes tides. Users have to specify namelist to handle this difference (l_parent_include_tide = .false. in nml_submodel_tide).

18.5 Usage

When TIDE is added to OPTIONS, the tide model is coupled. Specify parameters of the tide model by namelist nml_tide_model, the initial condition by nml_tide_run, and forced tidal constituents by nml_tide_forcing. If a specific state is used for the initial condition, three restart files must be prepared.

Chapter 19

Biogeochemical model

There are several options for Biogeochemical models in MRI.COM. These biogeochemical models have been developed for both ocean-only and coupled ocean-atmosphere-vegetation carbon cycle studies. They feature an explicit representation of a marine ecosystem, which is assumed to be limited by light, temperature, and nutrients availability. This chapter describes the details of the biogeochemical models.

19.1 Inorganic carbon cycle and biological model

Biogeochemical models are composed of inorganic carbon-cycle and ecosystem component models. In the inorganic carbon-cycle component, the partial pressure of CO_2 at the sea surface ($p\mathrm{CO}_2$) is diagnosed from the values of dissolved inorganic carbon (DIC) and Alkalinity (Alk) at the sea surface, which should be affected by the ecosystem component. The difference in $p\mathrm{CO}_2$ between the atmosphere and ocean determines ocean CO_2 uptake from or release to the atmosphere and is essential for simulating the CO_2 concentration in the atmosphere. Inorganic carbonate chemistry and partial pressure physics are well understood and can be reproduced with fair accuracy. The ecosystem component deals with various biological activities, and gives sources and sinks of the nutrients, DIC, Alk, and dissolved oxygen through these activities. Our knowledge of these activities is far from complete, and they are difficult to estimate even in state-of-the-art models.

There are many biological models and methods for calculating the ecosystem components. One of the simplest biological models has only one nutrient component (such as PO_4) as a prognostic variable and calculates neither phytoplankton nor zooplankton explicitly. In this case, the export of biologically generated soft tissue (organic matter) and hard tissue (carbonate) to the deep ocean, collectively known as the biological pump, is parameterized in terms of temperature, salinity, shortwave radiation, and nutrients.

A Nutrient-Phytoplankton-Zooplankton-Detritus (NPZD) model is more complex than the above model, but still a simple biological model. The NPZD model has four prognostic variables (nutrient, phytoplankton, zooplankton, and detritus). Though parameterized in a simple form, basic biological activities, such as photosynthesis, excretion, grazing, and mortality are explicitly calculated.

More complex models classify phytoplankton and zooplankton into several groups, and deal with many complex interactions between them. In general, it is expected that the more complex the biological model becomes the more realistic pattern the model can simulate. However, because of our incomplete knowledge about the biological activities, the complex models do not always yield better results, even though they require more computer resources.

To simulate the carbon cycle in the ocean, some biological processes should be calculated in the ecosystem component to obtain DIC at the sea surface. However, the carbon cycle component is not always necessary when our interests are to simulate the ecosystem itself. The Ocean Carbon-Cycle Model Intercomparison Project (OCMIP) protocols and studies of Yamanaka and Tajika (1996) and Obata and Kitamura (2003) focus on the former carbon cycle in the ocean, and the ecosystem components in these studies are quite simple. Biogeochemical models adopted in MRI.COM are classified in this category. The latter studies usually use complex biological models such as NEMURO (Kishi et al., 2001). Of course, this type of model could be adopted as an ecosystem component of the biogeochemical model in the former studies in hopes of better simulation of carbon cycle.

Originally, the carbon cycle component followed the OCMIP protocols (Orr et al., 1999) whose authority is recognized in the community. Recently, biogeochemical protocols for CMIP6 require that the former OCMIP code should be replaced by *mocsy* routines (Orr and Epitalon, 2015) to use the equilibrium constants recommended for best practices (Orr et al., 2017). MRI.COM can choose either option by setting a namelist. Here we describe the procedure using the *mocsy* routines.

MRI.COM has several options for the ecosystem component. At present, MRI.COM can incorporate the Obata and Kitamura model (Obata and Kitamura, 2003) or an NPZD model based on Oschlies (2001). The biogeochemical model of MRI.COM is largely based on Schmittner et al. (2008) when an NPZD model is adopted as an ecosystem component.

Basic units in MRI.COM are cgs, but in these biogeochemical subroutines, we use MKS units for the sake of future development. We use $\mathrm{mol/m^3}$ for the units of nutrients. When the coefficients of their model are applied, they should be converted to the corresponding units.

19.2 Governing equations

Here we describe the biogeochemical models of MRI.COM. When an NPZD model is incorporated as the ecosystem component, the governing equations are as follows. When Obata and Kitamura model is used instead of the NPZD model, the first four biogeochemical compartments (DIC, Alk, PO_4, and O_2) are used.

$$\frac{\partial \mathrm{DIC}}{\partial t} = -\mathcal{A}(\mathrm{DIC}) + \mathcal{D}(\mathrm{DIC}) + S_b(\mathrm{DIC}) + J_v(\mathrm{DIC}) + J_g(\mathrm{DIC}), \tag{19.1}$$

$$\frac{\partial \mathrm{Alk}}{\partial t} = -\mathcal{A}(\mathrm{Alk}) + \mathcal{D}(\mathrm{Alk}) + S_b(\mathrm{Alk}) + J_v(\mathrm{Alk}), \tag{19.2}$$

$$\frac{\partial [\mathrm{PO_4}]}{\partial t} = -\mathcal{A}([\mathrm{PO_4}]) + \mathcal{D}([\mathrm{PO_4}]) + S_b([\mathrm{PO_4}]), \tag{19.3}$$

$$\frac{\partial [\mathrm{O_2}]}{\partial t} = -\mathcal{A}([\mathrm{O_2}]) + \mathcal{D}([\mathrm{O_2}]) + S_b([\mathrm{O_2}]) + J_g([\mathrm{O_2}]), \tag{19.4}$$

$$\frac{\partial [\mathrm{NO_3}]}{\partial t} = -\mathcal{A}([\mathrm{NO_3}]) + \mathcal{D}([\mathrm{NO_3}]) + S_b([\mathrm{NO_3}]), \tag{19.5}$$

$$\frac{\partial [\mathrm{PhyPl}]}{\partial t} = -\mathcal{A}([\mathrm{PhyPl}]) + \mathcal{D}([\mathrm{PhyPl}]) + S_b([\mathrm{PhyPl}]), \tag{19.6}$$

$$\frac{\partial [\mathrm{ZooPl}]}{\partial t} = -\mathcal{A}([\mathrm{ZooPl}]) + \mathcal{D}([\mathrm{ZooPl}]) + S_b([\mathrm{ZooPl}]), \tag{19.7}$$

$$\frac{\partial [\mathrm{Detri}]}{\partial t} = -\mathcal{A}([\mathrm{Detri}]) + \mathcal{D}([\mathrm{Detri}]) + S_b([\mathrm{Detri}]), \tag{19.8}$$

where $\mathcal{A}()$ and $\mathcal{D}()$ represent advection and diffusion operator, respectively, and $S_b()$ is source minus sink due to the biogeochemical activities. The square brackets mean dissolved concentration in $\mathrm{mol/m^3}$ of the substance within them. The terms represented by $J_g()$ and $J_v()$ are the air-sea gas fluxes (Section 19.3.1) and the dilution and concentration effects of evaporation and precipitation on DIC and Alk (Section 19.3.2), respectively, which appear only at the sea surface. The term $J_g()$ is calculated based on the OMIP protocol by using the air-sea gas transfer velocity and concentration in the seawater. The term $J_v()$ appears only when the salinity flux is given virtually instead of the increase or decrease of the volume at the surface layers due to evaporation and precipitation.

19.3 Carbon cycle component

To estimate J_g and J_v, we follow the protocol of OMIP, which is described in detail in Orr et al. (2017). The program to calculate them is based on the *mocsy* subroutines. We have modified this subroutine so that it can be used in the calculation of the MRI.COM code.

19.3.1 Air-sea gas exchange fluxes at the sea surface (J_g)

The air-sea gas transfer must be calculated for DIC and $[O_2]$. The terms $J_g(\mathrm{DIC})$ and $J_g([O_2])$ appear only in the uppermost layer. When these fluxes are expressed as $F_g(\mathrm{DIC})$ and $F_g([O_2])$, $J_g(\mathrm{DIC})$ and $J_g([O_2])$ are given as follows:

$$J_g(\mathrm{DIC}) = \frac{F_g(\mathrm{DIC})}{\Delta z_{\frac{1}{2}}}, \tag{19.9}$$

$$J_g([\mathrm{O_2}]) = \frac{F_g([\mathrm{O_2}])}{\Delta z_{\frac{1}{2}}}, \tag{19.10}$$

where

$$F_g(\mathrm{DIC}) = K_w^{\mathrm{CO_2}} * ([\mathrm{CO_2}]_{\mathrm{sat}} - [\mathrm{CO_2}]_{\mathrm{surf}}), \tag{19.11}$$

$$F_g([\mathrm{O_2}]) = K_w^{\mathrm{O_2}} * ([\mathrm{O_2}]_{\mathrm{sat}} - [\mathrm{O_2}]_{\mathrm{surf}}), \tag{19.12}$$

and $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer of the model. Here a standard gas transfer formulation is adopted. $K_w^{\mathrm{CO_2}}$ and $K_w^{\mathrm{O_2}}$ are their gas transfer velocity, $[\mathrm{CO_2}]_{\mathrm{sat}}$ and $[\mathrm{O_2}]_{\mathrm{sat}}$ are their saturation concentrations with respect to the atmosphere, and $[\mathrm{CO_2}]_{\mathrm{surf}}$, $[\mathrm{O_2}]_{\mathrm{surf}}$, are their surface concentration.

a. Gas transfer velocity

The coefficients $K_w^{\mathrm{CO}_2}$ and $K_w^{\mathrm{O}_2}$ are the air-sea gas transfer (piston) velocity and are diagnosed as follows:

$$K_w^{\mathrm{CO}_2} = a\left(\frac{Sc^{\mathrm{CO}_2}}{660}\right)^{-1/2} U_{10}^2(1-f_i), \tag{19.13}$$

$$K_w^{\mathrm{O}_2} = a\left(\frac{Sc^{\mathrm{O}_2}}{660}\right)^{-1/2} U_{10}^2(1-f_i), \tag{19.14}$$

where

- f_i is the fraction of the sea surface covered with ice,
- U_{10} is 10 m scalar wind speed,
- a is the coefficient of 0.251 cm hr^{-1}/(m s^{-1})^2, which will give K_w in units of cm hr^{-1} when winds speeds (U_{10}) are in m s^{-1}. This is specified in the OMIP protocol,
- Sc^{CO_2} and Sc^{O_2} are the Schmidt numbers for CO_2 and O_2.

b. Saturation concentration with respect to the atmosphere

The surface water gas concentration in equilibrium with the atmosphere (saturation concentrations) for gas A has the following relationship.

$$[A]_{\mathrm{sat}} = K_0 f_A \tag{19.15}$$

$$= K_0 C_f\, p_A \tag{19.16}$$

$$= K_0 C_f\,(P_a - p\mathrm{H_2O})x_A \tag{19.17}$$

$$= \phi_A x_A \tag{19.18}$$

$$\simeq \frac{P_a}{P_a^\circ}\phi_A^0 x_A \qquad \text{(with errors less than 0.1\%)} \tag{19.19}$$

$$= \frac{P_a}{P_a^\circ}[A]_{\mathrm{sat}}^0, \tag{19.20}$$

where K_0 is its solubility, f_A is its atmospheric fugacity, C_f is its fugacity coefficient, p_A is its atmospheric partial pressure, x_A is its mole fraction in dry air, P_a is total atmospheric pressure, P_a° is the standard atmosphere (=1013.25 hPa), $p\mathrm{H_2O}$ is water vapor pressure at saturation, ϕ_A is its solubility function, ϕ_A^0 and $[A]_{\mathrm{sat}}^0$ are its solubility function and saturation concentration at the reference pressure.

Orr et al. (2017) explain two approaches to calculate the saturation concentration.

The first approach uses equations (19.19) or (19.20), computing solubility function ϕ_A^0 or saturation concentration $[A]_{\mathrm{sat}}^0$ at the reference atmospheric pressure first, and then converting them to those at the atmospheric pressure P_a. For gases that are often used as tracers in oceanography such as CFCs, ϕ_A^0 can be expressed as a function of in-situ temperature and salinity by using empirical fit. For oxygen, an empirical fit for $[\mathrm{O_2}]_{\mathrm{sat}}^0$ is available. This is a rather conventional approach, and is used in MRI.COM except for carbon.

The other approach uses equations (19.15)-(19.17), computing its partial pressure $p_A \equiv (P_a - p\mathrm{H_2O})x_A$, then multiplying by $K' \equiv K_0 C_f$, or multiplying by C_f and then multiplying by K_0 if atmospheric fugacity (f_A) is needed. For typical gases, K', K_0, and C_f are available as a function of temperature and salinity. The *mocsy* routine adopts this approach, and MRI.COM also follows it for calculating carbon flux at the surface.

c. Computing $\mathrm{CO_2}$ concentrations at the surface

In the *mocsy* routine, $[\mathrm{CO_2}]_{\mathrm{surf}}$ is diagnosed every step from DIC, Alk, temperature, salinity, $[\mathrm{PO_4}]$, and silicate concentration at the surface. After $[\mathrm{CO_2}]_{\mathrm{surf}}$ is computed, fugacity ($f_{\mathrm{CO_2}}$) and partial pressure ($p\mathrm{CO_2}$) of the ocean surface are computed as

$$f_{\mathrm{CO_2}} = [\mathrm{CO_2}]_{\mathrm{surf}}/K_0, \tag{19.21}$$

$$p\mathrm{CO_2} = f_{\mathrm{CO_2}}/C_f\,, \tag{19.22}$$

where K_0 and C_f are estimated by using temperature and salinity.

Diagnosis of $[CO_2]_{surf}$ is the most complex of the above calculations and has the largest computational cost. To be precise, diagnosis of $[CO_2]$ actually means diagnosing $[CO_2] + [H_2CO_3]$, which are difficult to distinguish analytically. These two species are usually combined and the sum is expressed as the concentration of a hypothetical species, $[CO_2^*]$ or $[H_2CO_3^*]$. Here, the former notation is used. The relationship between this $[CO_2^*]$ and DIC is as follows:

$$\mathrm{DIC} = [\mathrm{CO_2}] + [\mathrm{H_2CO_3}] + [\mathrm{HCO_3^-}] + [\mathrm{CO_3^{2-}}] \tag{19.23}$$

$$= [\mathrm{CO_2^*}] \qquad\qquad + [\mathrm{HCO_3^-}] + [\mathrm{CO_3^{2-}}]. \tag{19.24}$$

In the OMIP protocol, the following equations are solved to obtain $[CO_2^*]$.

The equilibrium constants for dissociation reactions are:

$$K_1 = \frac{[\mathrm{H^+}][\mathrm{HCO_3^-}]}{[\mathrm{CO_2^*}]} \qquad K_2 = \frac{[\mathrm{H^+}][\mathrm{CO_3^{2-}}]}{[\mathrm{HCO_3^-}]}, \tag{19.25}$$

$$K_\mathrm{B} = \frac{[\mathrm{H^+}][\mathrm{B(OH)_4^-}]}{[\mathrm{B(OH)_3}]}, \tag{19.26}$$

$$K_\mathrm{1P} = \frac{[\mathrm{H^+}][\mathrm{H_2PO_4^-}]}{[\mathrm{H_3PO_4}]} \qquad K_\mathrm{2P} = \frac{[\mathrm{H^+}][\mathrm{HPO_4^{2-}}]}{[\mathrm{H_2PO_4^-}]} \qquad K_\mathrm{3P} = \frac{[\mathrm{H^+}][\mathrm{PO_4^{3-}}]}{[\mathrm{HPO_4^{2-}}]}, \tag{19.27}$$

$$K_\mathrm{Si} = \frac{[\mathrm{H^+}][\mathrm{SiO(OH)_3^-}]}{[\mathrm{Si(OH)_4}]}, \tag{19.28}$$

$$K_W = [\mathrm{H^+}][\mathrm{OH^-}], \tag{19.29}$$

$$K_\mathrm{S} = \frac{[\mathrm{H^+}]_F[\mathrm{SO_4^{2-}}]}{[\mathrm{H\,SO_4^-}]}, \tag{19.30}$$

and

$$K_\mathrm{F} = \frac{[\mathrm{H^+}]_F[\mathrm{F^-}]}{[\mathrm{HF}]}, \tag{19.31}$$

where $[H^+]$ is the hydrogen ion concentration in sea water and $[H^+]_F$ is the free hydrogen ion concentration. There is another scale for the hydrogen ion concentration, the total hydrogen ion concentration $[H^+]_T$. The subscript T means "total" and F means "free." These three hydrogen ion concentrations are related as follows:

$$[\mathrm{H^+}] = [\mathrm{H^+}]_F\left(1 + \frac{S_T}{K_\mathrm{S}} + \frac{F_T}{K_\mathrm{F}}\right), \tag{19.32}$$

$$\text{and} \qquad [\mathrm{H^+}]_T = [\mathrm{H^+}]_F\left(1 + \frac{S_T}{K_\mathrm{S}}\right). \tag{19.33}$$

There are three pH scales corresponding to these three hydrogen ion concentrations.

The equilibrium constants K_x are given as a function of temperature, salinity, and pH. Note that the equilibrium constants are given in terms of concentrations, and that all constants are referenced to the seawater pH scale, except for K_S, which is referenced to the free pH scale.

The total dissolved inorganic carbon, boron, phosphate, silicate, sulfate, and fluoride are expressed as follows:

$$\mathrm{DIC} = [\mathrm{CO_2^*}] + [\mathrm{HCO_3^-}] + [\mathrm{CO_3^{2-}}], \tag{19.34}$$

$$B_\mathrm{T} = [\mathrm{B(OH)_3}] + [\mathrm{B(OH)_4^-}], \tag{19.35}$$

$$P_\mathrm{T} = [\mathrm{H_3PO_4}] + [\mathrm{H_2PO_4^-}] + [\mathrm{HPO_4^{2-}}] + [\mathrm{PO_4^{3-}}], \tag{19.36}$$

$$Si_\mathrm{T} = [\mathrm{Si(OH)_4}] + [\mathrm{Si(OH)_3^-}], \tag{19.37}$$

$$S_{\mathrm{T}} = [\mathrm{HSO}_4^-] + [\mathrm{SO}_4^{2-}], \tag{19.38}$$

and

$$F_{\mathrm{T}} = [\mathrm{HF}] + [\mathrm{F}^-]. \tag{19.39}$$

Alkalinity used in this calculation is defined as follows:

$$\begin{aligned}\mathrm{Alk} = {} & [\mathrm{HCO}_3^-] + 2[\mathrm{CO}_3^{2-}] + [\mathrm{B(OH)}_4^-] + [\mathrm{OH}^-] + [\mathrm{HPO}_4^{2-}] + 2[\mathrm{PO}_4^{3-}] + [\mathrm{SiO(OH)}_3^-] \\ & - [\mathrm{H}^+]_F - [\mathrm{HSO}_4^-] - [\mathrm{HF}] - [\mathrm{H_3PO_4}].\end{aligned} \tag{19.40}$$

These expressions exclude the contribution of $\mathrm{NH_3}$, HS^-, and S^{2-}.

If we assume that DIC, Alk, P_T, and Si_T are known, this system contains 18 equations with 18 unknowns, so they can be solved. Detailed solver method has been described in Orr and Epitalon (2015). The concentration $[Si_T]$ is not predicted in the biogeochemical model adopted in MRI.COM but is rather specified by the annual mean value from WOA2013.

19.3.2 Dilution and concentration effects of evaporation and precipitation on DIC and Alk

The dilution and concentration effects of evaporation and precipitation significantly impact the concentrations of some chemical species in seawater. This is particularly true for DIC and Alk, which have large background concentrations compared with their spatial variability. MRI.COM uses a free surface, so the impact of evaporation and precipitation is straightforward to model as far as `WFLUX` is used. When `WFLUX` (or this option) is not used, salinity flux is diagnosed and applied instead of the freshwater flux. In this case, the dilution and concentration effect of evaporation (E) and precipitation (P) should be taken into account. Here, they are parameterized as virtual DIC and Alk fluxes, similar to the virtual salt flux used in physical ocean GCMs.

In MRI.COM, the tendency of salinity due to the virtual salt flux is given by

$$\mathtt{sflux}(i,j) = -(P - E) * S(i,j,1)/\Delta z, \tag{19.41}$$

where $S(i,j,1)$ and Δz are the salinity and thickness of the uppermost layer. Note that the variable $\mathtt{sflux}(i,j)$ is *not* the salinity flux but the time change rate of salinity due to the flux even though the spelling brings up the image of the flux.

In MRI.COM, DIC and Alk are modified by the virtual salt flux as follows:

$$J_v(\mathrm{DIC}(i,j,1)) = \mathtt{sflux}(i,j)/S(i,j,1) * \mathrm{DIC}(i,j,1), \tag{19.42}$$

$$J_v(\mathrm{Alk}(i,j,1)) = \mathtt{sflux}(i,j)/S(i,j,1) * \mathrm{Alk}(i,j,1). \tag{19.43}$$

Strictly speaking, air-sea fluxes of fresh water impact other species. However, these modifications are not usually applied because their spatial variabilities are significantly greater than those of DIC and Alk.

In the OMIP protocol, as well as the previous OCMIP protocol, the global averaged salinity S_g is used instead of $S(i,j,1)$ in equations(19.42,19.43). In addition, globally integrated $J_v(\mathrm{DIC})$ and $J_v(\mathrm{Alk})$ are set to 0. In MRI.COM, these modifications are not the default considering the use in regional ocean models.

19.4 Obata and Kitamura model

This section was contributed by A. Obata.

The Obata and Kitamura model used in MRI.COM simply represents the source and sink terms of DIC, Alk, $[\mathrm{PO_4}]$, and $[\mathrm{O_2}]$ due to the biogeochemical activities: new production driven by insolation and phosphate concentration in the surface ocean, its export to depth, and remineralization in the deep ocean. According to the Michaelis-Menten kinetics (Dugdale, 1967), phosphorus in the new production exported to depth (ExprodP) is parameterized as $rL[\mathrm{PO_4}]^2/([\mathrm{PO_4}] + k)$, where r is a proportional factor ($r = 0.9\,\mathrm{yr}^{-1}$), L is the insolation normalized by the annual mean insolation on the equator, and k is the half-saturation constant ($k = 0.377\,\mathrm{mmol/m}^{-3}$). The values of r and k are adjusted to reproduce the optimum atmospheric $\mathrm{CO_2}$ concentration and ocean biogeochemical distribution for the preindustrial state of the model. The relationship between the changes in the chemical composition of seawater and the composition of particulate organic matter (POM) is assumed to follow the Redfield ratio $\mathrm{P} : \mathrm{N} : \mathrm{C} : \mathrm{O_2} = 1 : 16 : 106 : -138 \equiv 1 : R_{np} : R_{cp} : -R_{op}$, where $R_{ab} = A/B$ and "o" represents $\mathrm{O_2}$ (Redfield et al., 1963). The rain ratio of calcite to particulate organic carbon (POC) is 0.09, which is in the range proposed by Yamanaka and Tajika (1996). The surface thickness where the export production occurs is fixed at 60 m. The vertical distribution of POM and calcite vertical flux below a depth of 100 m is proportional to $(z/100\mathrm{m})^{-0.9}$ and $\exp(-z/3500\mathrm{m})$ (z is the depth in meters), respectively, following the work of Yamanaka and Tajika (1996). The remineralization of POM (RemiP for phosphorus) and the dissolution of calcite (SolnCa) at depth are parameterized by these fluxes. Oxygen saturation is prescribed at the sea surface. The solubility of oxygen is computed from the formula of Weiss (1970). Source and sink terms of $S_b()$ representing the above processes are as follows:

$$
\begin{aligned}
S_b(\mathrm{DIC}) &= R_{cp} * \mathrm{RemiP} + \mathrm{SolnCa} - R_{cp} * \mathrm{ExprodP} && (19.44)\\
S_b(\mathrm{Alk}) &= 2 * \mathrm{SolnCa} + R_{np} * \mathrm{ExprodP} - R_{np} * \mathrm{RemiP} && (19.45)\\
S_b([\mathrm{PO_4}]) &= \mathrm{RemiP} - \mathrm{ExprodP} && (19.46)\\
S_b([\mathrm{O_2}]) &= -R_{op} * S_b([\mathrm{PO_4}]) && (19.47)
\end{aligned}
$$

19.5 NPZD model

The NPZD model used in MRI.COM is constructed on the assumptions that the biological elemental composition ratio is nearly constant (Redfield ratio) and that the concentration of organisms can be estimated by nitrogen or phosphorus. The prognostic variables of nitrate ($\mathrm{NO_3}$), phytoplankton (PhyPl), zooplankton (ZooPl), and detritus (Detri) are normalized in terms of nitrogen $1\,\mathrm{mol/m^3}$. For example, [PhyPl] represents the concentration of phytoplankton estimated in terms of nitrogen in one cubic meter ($\mathrm{N\ mol/m^3}$). The increase and decrease of carbon can be diagnosed by multiplying by R_{cn}.

Source and sink terms $S_b()$ calculated in the NPZD model are as follows. Those for DIC and Alk, $S_b(\mathrm{DIC})$ and $S_b(\mathrm{Alk})$, used for calculating the carbon cycle, are described later in this section.

$$
\begin{aligned}
S_b([\mathrm{PhyPl}]) &= \mathrm{Priprod} - \mathrm{MortP1} - \mathrm{MortP2} - \mathrm{GrP2Z} && (19.48)\\
S_b([\mathrm{ZooPl}]) &= \mathit{assim} * \mathrm{GrP2Z} - \mathrm{Excrtn} - \mathrm{MortZ} && (19.49)\\
S_b([\mathrm{Detri}]) &= [(1-\mathit{assim}) * \mathrm{GrP2Z} + \mathrm{MortP2} + \mathrm{MortZ}] - \mathrm{RemiD} - w_{detri}\frac{\partial \mathrm{Detri}}{\partial z} && (19.50)\\
S_b([\mathrm{NO_3}]) &= \mathrm{MortP1} + \mathrm{Excrtn} + \mathrm{RmeiD} - \mathrm{PriProd} && (19.51)\\
S_b([\mathrm{PO_4}]) &= R_{pn} * S_b([\mathrm{NO_3}]) && (19.52)\\
S_b([\mathrm{O_2}]) &= -R_{on} * R_{np} * S_b([\mathrm{PO_4}]) && (19.53)
\end{aligned}
$$

There is no input from the atmosphere such as nitrogen fixation in the above equations, so the sum of these five equations becomes zero at each grid point except for the term for detritus sinking ($-w_{detri}\frac{\partial \mathrm{Detri}}{\partial z}$). The term for detritus sinking expresses the biological pump, whose role is to remove nutrients from the upper layers and transport them into the deep ocean where the plankton cannot use the nutrients. When vertically integrated, the sum of each grid is 0 even though this sinking term is included. The nutrients are transported horizontally through physical processes such as advection and diffusion.

In general, the nitrate limitation is more severe than the phosphate limitation so it is not always necessary to calculate phosphate. However, in the simpler model of Obata and Kitamura (2003), phosphate is used as a prognostic variable. So, to be consistent, phosphate is calculated in the ecosystem component of MRI.COM. Next, we elaborate on the above equations.

19.5.1 Description of each term

- Priprod = $J(I,N,P) * [\mathrm{PhyPl}]$
  Primary production expresses photosynthesis (described in detail in the next subsection).
- MortP1 = $\phi_P * [\mathrm{PhyPl}]$
  The conversion of mortality phytoplankton directly into nutrients. This term was introduced by Oschlies (2001) to increase the primary production of subtropical gyre, where the nutrient limit is severe.
- MortP2 = $\phi_{PP} * [\mathrm{PhyPl}]^2$
  The conversion from phytoplankton to detritus (normal mortality of phytoplankton).
- GrP2Z = $G(P) * [\mathrm{ZooPl}]$
  The grazing of zooplankton. There are a number of parameterizations of grazing. In this formulation, this is given as $G(P) = g * \epsilon * [\mathrm{PhyPl}]^2/(g + \epsilon * [\mathrm{PhyPl}]^2)$. Among the grazing, the ratio *assim* is used for the growth of zooplankton, and the remainder $(1 - \mathit{assim})$ is converted to detritus.
- Excrtn = $d * [\mathrm{ZooPl}]$
  Excretion of zooplankton. The excretion is dissolute and directly returned to nutrients ($\mathrm{NO_3}$).

- MortZ = $\phi_Z * [\text{ZooPl}]^2$
  The conversion from the zooplankton to detritus (mortality of zooplankton).
- RemiD = $\phi_D * [\text{Detri}]$
  Remineralization of detritus. This is converted to nutrients through the activity of bacteria.

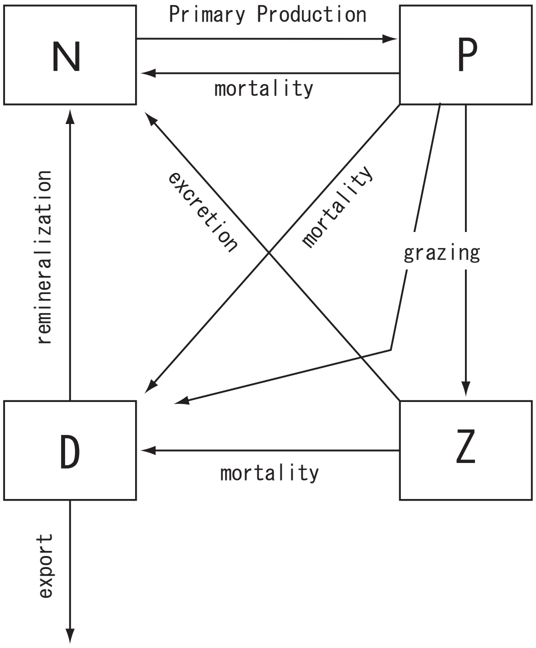

Figure19.1 Schematic of NPZD model

19.5.2 Primary Production

The growth rate of phytoplankton is limited by the irradiance (I) and nutrients. This limitation is expressed in several ways. Here we adopt an expression with a minimum function:

$$J(I, \mathrm{N}, \mathrm{P}) = \min\left(J_I, J_N, J_P\right), \tag{19.54}$$

where J_I denotes the purely light-limited growth rate, and J_N and J_P are nutrient-limited growth rates that are functions of nitrate or phosphate.

The light-limited growth is calculated as follows:

$$J_I = \frac{J_{max}\alpha I}{\left[J_{max}^2 + (\alpha I)^2\right]^{1/2}}. \tag{19.55}$$

Here, J_{max} is the light-saturated growth, which depends on temperature based on Eppley (1972) as

$$J_{max} = a \cdot b^{c\,\theta}, \tag{19.56}$$

where $a = 0.6\,\mathrm{day}^{-1}$, $b = 1.066$, and $c = 1\,(°\mathrm{C})^{-1}$. Note that the default values in MRI.COM are based on Schmittner et al. (2008) and differ from these values (see Table 19.1). Equation (19.55) is called Smith-type growth. The coefficient α in the equation is "the initial slope of photosynthesis versus irradiance (P-I) curve," that is,

$$\alpha = \lim_{I \to 0} \frac{\partial J_I}{\partial I}. \tag{19.57}$$

Thus, it represents how sensitive J_I is to the irradiance when the light is weak.

Irradiance (I) depends on the angle of incidence and the refraction and absorption in the seawater.

$$I = I_{z=0}\ \mathrm{PAR}\ \exp\left(-k_w \tilde{z} - k_e \int_0^{\tilde{z}} P dz\right), \tag{19.58}$$

where $I_{z=0}$ denotes the downward shortwave radiation at the sea surface, PAR is the photosynthetically active radiation ratio (0.43) and $\tilde{z} = z/\cos\theta = z/\sqrt{1-\sin^2\theta/1.33^2}$ is the effective vertical coordinate (positive downward) with 1.33 as the refraction index according to Snell's law relating the zenith angle of incidence in air (θ) to the angle of incidence in water. The angle of incidence θ is a function of the latitude ϕ and declination δ.

For the nutrient-limited growth rate (J_N and J_P), we adopt the Optimal Uptake (OU) equation instead of the classic Michaelis-Menten (MM) equation. For the classic MM equation, the nitrate-limited growth rate is expressed as

$$J_N = J_{MM} = \frac{J_{max}N}{K_N + N}, \tag{19.59}$$

where K_N is a half-saturation constant for NO_3 uptake. In contrast, the Optimal Uptake (OU) equation for a nitrate is expressed as follows:

$$J_N = J_{OU} = \frac{V_0 N}{N + 2\sqrt{\frac{V_0}{A_0}N} + \frac{V_0}{A_0}}, \tag{19.60}$$

where A_0 and V_0 are the potential maximum values of affinity and uptake rate, respectively (see Smith et al. (2009) for details). Optimal Uptake (OU) kinetics assumes a physiological trade-off between the efficiency of nutrient encounter at the cell surface and the maximum rate at which a nutrient can be assimilated (Smith et al., 2009). The key idea is that phytoplankton alters the number of its surface uptake sites (or ion channels), which determines the encounter timescale, versus internal enzymes, which assimilate the nutrients once encountered.

We set parameters V_0 and A_0 so that the rates of uptake, J_{MM} and J_{OU}, are equal at $N = K_N$. In addition, we fix the ratio $V_0/A_0 = \alpha_{OU}$, where α_{OU} is determined from fitting the data. This requires

$$V_0 = 0.5\left(1 + \sqrt{\frac{\alpha_{OU}}{K_N}}\right)^2 J_{max}. \tag{19.61}$$

Finally, we obtain

$$J_{OU} = \frac{V_0 N}{N + 2\sqrt{\alpha_{OU}N} + \alpha_{OU}}. \tag{19.62}$$

We use $\alpha_{OU} = 0.19$, which is determined from the fitting of $\log K_N$ vs $\log N$ in the wide range of N by Smith et al. (2009).

19.5.3 Variation of DIC and Alk due to biological activity

Production of DIC and Alk is controlled by changes in inorganic nutrients and calcium carbonate ($CaCO_3$), in molar numbers according to

$$S_b(\mathrm{DIC}) = S_b([\mathrm{PO_4}])R_{cp} - S_b([\mathrm{CaCO_3}]), \tag{19.63}$$

$$S_b(\mathrm{Alk}) = -S_b([\mathrm{NO_3}]) - 2 \cdot S_b([\mathrm{CaCO_3}]). \tag{19.64}$$

Thus, only these source and sink terms of DIC and Alk are estimated. Since $[PO_4]$ and $[NO_3]$ are prognostic variables, their source and sink are explicitly calculated by the biological model. In contrast, the downward movement of $CaCO_3$ is much faster than the modeled downward velocity of water mass, so $[CaCO_3]$ is not a prognostic variable, and its source (Pr) and sink (Di) are diagnosed by the following equation,

$$S_b([\mathrm{CaCO_3}]) = Pr([\mathrm{CaCO_3}]) - Di([\mathrm{CaCO_3}]). \tag{19.65}$$

Following Schmittner et al. (2008), the source term ($Pr([\mathrm{CaCO_3}])$) of calcium carbonate is determined by the production of detritus as follows:

$$Pr([\mathrm{CaCO_3}]) = [(1 - assim) * [\mathrm{GrP2Z}] + [\mathrm{MortP2}] + [\mathrm{MortZ}]]\, R_{\mathrm{CaCO_3/POC}} R_{C:N}, \tag{19.66}$$

where $assim$, GrP2Z, MortP2, and MortZ are as described above. The sink term ($Di([\mathrm{CaCO_3}])$) of calcium carbonate is parameterized as

$$Di([\mathrm{CaCO_3}]) = \int Pr([\mathrm{CaCO_3}])dz \cdot \frac{d}{dz}\left(\exp\left(-z/D_{\mathrm{CaCO_3}}\right)\right), \tag{19.67}$$

which expresses an instantaneous sinking with an e-holding depth of $D_{\mathrm{CaCO_3}} = 3500\,\mathrm{m}$. In this equation, z is positive downward. This depth of 3500 m was estimated by Yamanaka and Tajika (1996) to reproduce the observed nutrient profile.

This value is standard and also is used in the simple biological model in the protocol of OCMIP. The vertical integral of the source minus sink should be zero. Thus, when the sea bottom appears before the sum becomes zero, the remaining calcium carbonate is assumed to be dissolved in the lowermost layer. By using the ratio $R_{\mathrm{CaCO_3/POC}} = 0.035$ used by Schmittner et al. (2008), the resultant global mean Rain ratio should be roughly consistent with the recently estimated range (0.07 to 0.11) based on various observations.

19.6 Usage

In MRI.COM, carbon, dissolved oxygen, and ecosystem model as well as other passive tracers like CFCs can be calculated together or separately. Options described in `configure.in` determines what components should be used in MRI.COM. The combination of the options also determines the number of passive tracers that is necessary for calculation. See detail for the program `tracer_vars.F90`.

- When `CARBON` is used, carbon component is calculated in the model. For the ocean only model, the `CBNHSTRUN` option should be used to set the atmospheric CO2. Its internal source or sink is calculated in an ecosystem model.
- When `CARBON` is used, dissolved oxygen component is calculated.
- When `CBNHSTRUN` is used, the ‘AtmosphericxCO2’ in ppm should be applied as additional atmospheric forcing. This is the standard configuration for ocean only run. See Chapter 14 for the detailed setting.
- When `O2` is used, dissolved oxygen component is calculated. Its internal source or sink is calculated in an ecosystem model.

For ecosystem model, MRI.COM can use `NPZD` or `OBT` options.

- When `NPZD` is used, an NPZD model is used as the ecosystem component. In addition, when `CHLMA94` option is used, the chlorophyll concentration is considered to calculate the shortwave penetration following Morel and Antoine (1994).
  The parameters of NPZD are set in namelist `nml_bioNPZD`. The default values are based on Schmittner et al. (2008) and listed on Table 19.1. If the parameters of Oschlies (2001) are used, the high nutrient-low chlorophyll (HNLC) region in the North Pacific is not appropriately expressed. This may be because the parameters of Oschlies (2001) are calibrated for the North Atlantic biological model. The commonly used unit of time in biological models is [day]. Thus, in the namelist, the time unit of the biological parameter is specified by using the unit [day]. In the model, the time unit is converted to seconds, [sec].
- When `OBT` is used, a simple biological model of Obata and Kitamura (2003) is applied as the ecosystem component.

Restart files for the passive tracers should be specified by `nml_restart`. Following is an example when `CARBON`, `O2`, and `NPZD` options are chosen. In this case, `numtrc_p` = 8 should be specified in `configure.in`. See `docs/README_Restart.md` for more details.

An example specification of restart files when `CARBON`, `O2`, and `NPZD` options are chosen.

```
&nml_restart name='DIC', file_base='result/rs_dic'/
&nml_restart name='Total Alkalinity', file_base='result/rs_alk'/
&nml_restart name='Dissolved Oxygen', file_base='result/rs_o2'  /
&nml_restart name='Phosphate', file_base='result/rs_po4' /
&nml_restart name='Nitrate', file_base='result/rs_no3' /
&nml_restart name='Phytoplankton', file_base='result/rs_PhyPl'/
&nml_restart name='Zooplankton', file_base='result/rs_ZooPl'/
&nml_restart name='Detritus', file_base='result/rs_Detri'/
```

Table19.1 Parameters used for the NPZD ecosystem component (`NPZD`).

variable name	description	units	default value
`alphabio`	Initial slope of P-I curve	$(\mathrm{W\,m^{-2}})^{-1}\,\mathrm{day}^{-1}$	0.1d0
`abio`	Maximum growth rate parameter	day^{-1}	0.2d0
`bbio`	Maximum growth rate = a * b ** (c * T)		1.066d0
`cbio`			1.d0
`PARbio`	Photosynthetically active radiation		0.43d0
`dkcbio`	Light attenuation due to phytoplankton	$\mathrm{m}^{-1}\,(\mathrm{mol\,m^{-3}})^{-1}$	0.03d3
`dkwbio`	Light attenuation in the water	m^{-1}	0.04d0
`rk1bioNO3`	Half-saturation constant for NO_3 uptake	$\mathrm{mol\,m^{-3}}$	0.7d-3
`rk1bioPO4`	Half-saturation constant for PO_4 uptake	$\mathrm{mol\,m^{-3}}$	0.0d0
`alpha_ou`	Fitting constant for Optical Uptake kinetics		0.19d0
`gbio`	Maximum grazing rate	day^{-1}	1.575d0
`epsbio`	Prey capture rate	$(\mathrm{mol\,m^{-3}})^{-2}\,\mathrm{day}^{-1}$	1.6d6
`phiphy`	Specific mortality/recycling rate	s^{-1}	0.014d0
`phiphyq`	Quadratic mortality rate	$(\mathrm{mol\,m^{-3}})^{-1}\,\mathrm{day}^{-1}$	0.05d3
`a_npz`	Assimilation efficiency		0.925d0
`phizoo`	Quadratic mortality of zooplankton	$(\mathrm{mol\,m^{-3}})^{-1}\,\mathrm{day}^{-1}$	0.34d3
`d_npz`	Excretion	day^{-1}	0.01d0
`remina`	Remineralization rate	day^{-1}	0.048d0
`w_detr`	Sinking velocity	$\mathrm{m\,day}^{-1}$	2.0d0
`fac_wdetr`	Arbitrary parameter for numerical stability. When the concentration of detritus in the n+1 st level is higher than `fac_wdetr` times that in the n th level, `w_detr` is set to 0 between n and n+1 level.		3.d0
`c_mrtn`	Dimensionless scaling factor for Martin et al. (1987) Phi(z) = Phi(zo) * (z/`dp_mrtn`)**(-`c_mrtn`)		0.858d0
`Rcn`	Molar elemental ratio (C/N)		7.d0
`Ron`	Molar elemental ratio (O_2/N)		10.d0
`Rnp`	Molar elemental ratio (N/P)		16.d0
`dp_euph`	Maximum depth of euphotic zone	m	150.d0
`dp_mrtn`	Characteristic depth of martin curve	m	100.d0
`dp_eprdc`	The depth where the bio-export is diagnosed.	m	100.d0
	This value should be less than `dp_mrtn`.		
`Rcaco3poc`	$CaCO_3$ over nonphotosynthetic POC production ratio		0.03d0
`Dcaco3`	$CaCO_3$ remineralization e-folding depth	m	3500.d0
`shwv_intv_sec`	Interval for calculating the irradiance and light-limited growth rate. This must be a divisor of the time step for tracer.	min	600.d0

19.7 Program structure

```
ogcm__ini
  |
  +-- ptrc_ctl__ini
  |     |
  |     +-- ptrc__ini
  |           |
  |           +-- cbn__ini
  |           |
  |           +-- cbn__ini_history
  |
  +-- tracer__ctl__ini
        |
        +-- tracer__ini
              |
              +-- bio__ini
              |
              +-- bio__ini_history

ogcm__run
  |
  +-- part_1
  |     |
  |     +-- ptrc_ctl__main
  |     |     |
  |     |     +-- ptrc__surfflux
  |     |     |    |
  |     |     |    +-- cbn__set_xco2a
  |     |     |    |
  |     |     |    +-- cbn__calc_virtual_flux
  |     |     |    |
  |     |     |    +-- cbn__calc_co2_flux
  |     |     |          |
  |     |     |          +-- co2flux_mocsy
  |     |     |
  |     |     +-- ptrc__predict_surf
  |     |          |
  |     |          +-- cbn__predict_surf
  |     |
  |     +-- tracer_ctl__predict
  |          |
  |          +-- tracer__internal_source
  |                |
  |                +-- bio__predict
  |
  +-- hist_ctl
        |
        +-- hist_ctl__write
             |
             +-- cbn__write_history
```

Chapter 20

Inert tracers

This chapter explains inert tracers implemented in MRI.COM. In general, to include an inert tracer in a model integration, you need to specify the model options and tracer attributes corresponding to that tracer. General explanation on how to specify attributes of a tracer is given at Section 13.3.

20.1 Ideal age tracer

Ideal age of a water mass is the time in year since it last contacted with the sea surface. This tracer is introduced by England (1995).

20.1.1 Source / Sink term

Source and sink of an age tracer a [year], is expressed as follows. In the oceanic interior,

$$\frac{\partial a}{\partial t} = -\mathcal{A}(a) + \mathcal{D}(a) + \frac{1}{\text{the total number of seconds of this year}} \quad \text{year/sec}, \tag{20.1}$$

where $\mathcal{A}()$ and $\mathcal{D}()$ represent advection and diffusion operator, respectively. Note that the r. h. s. depends on whether it is leap year or not. The tracer ages slowly in leap years than in normal years.

At the sea surface, the ideal age is set to zero:

$$a(k = 1) = 0. \tag{20.2}$$

20.1.2 Usage

Required Model option `IDEALAGE`

Required Add one to `numtrc_p`

Required Restart file for passive tracer (`nmlrs_ptrc`) (See Section 19.6)

Required Name for `nml_tracer_data` is 'Ideal Age Tracer' (See Section 13.3)

Optional Namelist `nml_tracer_idealage_start` to specify the base date and time [year, month, day, hour, minute, second] for the age of water.

20.2 CFCs

MRI.COM can treat CFCs (CFC11 and CFC12) following the protocols of OMIP (Orr et al., 2017). There is no source and sink for CFCs in the interior.

20.2.1 Surface boundary condition

The CFCs have air-sea gas fluxes at the sea surface as the source. Their dissolved concentrations $[A]$ [mol m^{-3}] evolve according to the advection ($\mathcal{A}()$) and diffusion ($\mathcal{D}()$) in the oceanic interior,

$$\frac{\partial [A]}{\partial t} = -\mathcal{A}([A]) + \mathcal{D}([A]) + J_g([A]). \tag{20.3}$$

$J_g()$ represents source/sink due to the air-sea gas fluxes $F([A])$ at the sea surface,

$$J_g([A]) = \frac{F([A])}{\Delta z_{\frac{1}{2}}}, \tag{20.4}$$

with

$$F(A) = k_w([A]_{sat} - [A]), \tag{20.5}$$

where k_w is its gas transfer velocity, $[A]_{sat}$ is the surface gas concentration in equilibrium with the atmosphere at an atmospheric pressure at the surface (P_a). k_w is a function of wind velocity, surface temperature, and atmospheric pressure at the sea surface. $[A]_{sat}$ is given by

$$[A]_{sat} = \phi_A^0 x_A, \tag{20.6}$$

where x_A is its mole fraction in dry air. The combined solubility term ϕ_A^0 is computed using the empirical fit of temperature and salinity. See Orr et al. (2017) for further details.

20.2.2 Usage

Required Model option `CFC`
Required Add two to `numtrc_p`
Required Restart file for passive tracer (`nmlrs_ptrc`) (See Section 19.6)
Required Names for `nml_tracer_data` are 'CFC11' and 'CFC12' (See Section 13.3)
Required Namelist `nml_force_data` to specify partial gas pressure (See Section 14.10)
  file_data File that contains mole fraction in dry air of CFC11 (and CFC12) [ppt]
  name CFC11 (and 12)
  txyu 't'

20.3 SF$_6$

The calculation of SF$_6$ also follows the protocols of OMIP (Orr et al., 2017). However, the formula for k_w in SF6 is not given in (Orr et al., 2017), so we use (Wanninkhof, 2014) as the latest formula to replace it.

The formulation of SF$_6$ is nearly the same as CFCs. Only the coefficients of the empirical fit differ.

20.3.1 Surface boundary condition

See Section 20.2.1, where CFCs should read SF$_6$.

20.3.2 Usage

Required Model option `SF6`
Required Add one to `numtrc_p`
Required Restart file (`nmlrs_ptrc`) (See Section 19.6)
Required Names for `nml_tracer_data` is 'SF6' (See Section 13.3)
Required Namelist `nml_force_data` to specify partial gas pressure (See Section 14.10)
  file_data File that contains atmospheric SF$_6$ [ppt]
  name SF6
  txyu 't'

Chapter 21

Atmospheric model

21.1 Coupling with an atmospheric model

MRI.COM can be coupled with an atmospheric model (e.g., Yukimoto et al., 2019). MRI.COM exchanges information necessary to drive this coupled model with the atmospheric model at specified time intervals via Scup (Yoshimura and Yukimoto, 2008) in the coupled mode. The ocean model receives all kinds of surface fluxes from the atmospheric model and the surface oceanic state such as sea surface temperature, surface oceanic current, and sea ice state, are transferred to the atmospheric model. The coupling interval is usually longer than the time step of the ocean model and the fluxes are kept fixed in the ocean model during the coupling cycle. See Section 17.11 for details of flux adjustment for conserving heat and fresh water fluxes by correcting errors caused by interpolation and sea ice evolution during the coupling loop.

Work flow is summarized as follows:

1. Receive atmosphere-ocean and atmosphere-sea ice fluxes for the current coupling cycle from the atmospheric model.
2. Adjust the fluxes to compensate for errors associated with the interpolation between the numerical grids of the atmospheric model and the coupled sea ice-ocean model.
   (a) Adjust the fluxes to compensate for errors associated with the sea-ice time integration within the current coupling cycle.
   (b) Time integration of the coupled sea ice-ocean model.
   (c) Go to the next step 3 if the current coupling cycle has finished. Otherwise, go back to the step 2a.
3. Send the surface ocean and sea ice states to the atmospheric model and go back to the step 1.

21.1.1 Serial and parallel executions

The surface fluxes at air-sea and air-ice interfaces are calculated in the atmosphere model and sent to the ocean model via coupler in the standard way for the coupled atmosphere-ocean model in the Meteorological Research Institute. The atmosphere model uses an initial snapshot of each coupling cycle of ocean state variables such as SST and sea ice concentration required to calculate surface fluxes. In order to conserve the heat and freshwater of the atmosphere-ocean system, the sea surface heat and freshwater flux imposed on the ocean model must be the time mean of an coupling cycle. We need to serially integrate the atmosphere model and then the ocean model if we want to impose the surface fluxes in the same coupling cycle on the ocean model.

Such a serial execution tends to waste computational resources because one model must wait for the other model to finish its time integration. Therefore, the surface fluxes averaged over the previous coupling cycle are usually imposed on the ocean model. By doing so, the information required for time integration of both atmosphere and ocean models is known at the beginning of each coupling cycle, enabling efficient parallel computing. Strict heat and freshwater conservations are not available because of the delay of surface fluxes imposed on the ocean model, but we can minimize its undesirable effects by using the adequately small coupling interval. Users can specify either the serial execution or parallel execution in the Scup runtime options. Refer to Yoshimura and Yukimoto (2008) for more details of Scup.

21.1.2 Time integration from an initial state

When a coupled model is integrated from an initial state, the parallel execution noted in the previous section is not available because the previous coupling cycle does not exist there. We recommend that users serially run the atmosphere and ocean models for the first coupling cycle and run the models in parallel for the subsequent cycles in this case. The users need to specify an Scup run-time option for this function and also need to set `flg_send_initial_to_agcm` in namelist `nml_cgcm_scup` `.true.` (Table 21.1). The second and subsequent runs can be executed in the parallel mode from the first coupling cycle because Scup restart files contain the temporal mean of surface fluxes in the last coupling cycle of each preceding run. Note that `flg_send_initial_to_agcm` must be `.false.` in these restart runs.

21.2 Usage

21.2.1 Compilation

You must specify model options `SCUP` and `SCUPCGCM` in `configure.in`.

21.2.2 Job parameters (namelist)

MRI.COM parameters associated with the atmosphere-ocean coupling are specified by namelist `nml_cgcm_scup`. The coupled atmosphere-ocean model also requires a namelist file `NAMELIST_SCUP` that specifies configurations of variables exchanged between the atmosphere model and MRI.COM.

Table21.1 Namelist `nml_cgcm_scup`.

variable name	units	description	usage
`intkt_loop`	hour (integer)	Time interval of a coupling cycle	default = 1
`flg_get_l2o`	logical	Receive heat and freshwater fluxes from river and glacier runoff models or not	default = `.true.`
`flg_carbon`	logical	Couple carbon cycle	default = `.true.`
`flg_send_initial_to_agcm`	logical	Serial execution in the first coupling cycle	default = `.false.`

Chapter 22

Nesting

In MRI.COM, embedding of a fine-resolution regional model within a coarse-resolution model can be realized by nesting method. The available methods range from a serial execution of a coarse-resolution (parent) model writing a boundary data and a fine-resolution (child) model reading it (off-line one-way nesting) to a parallel execution of both models exchanging data in both directions (on-line two-way nesting), as summarized in Table 22.1.

This chapter explains facilities available in MRI.COM for nesting, and how to construct and run a set of coarse- and fine-resolution models. Section 22.1 explains outline of the nesting function, Sec. 22.2 how to create sub-model grid and topography, and Sec. 22.3 communication between the models. Section 22.4 describes optional features for conservation of variables in nesting experiments, and Sec. 22.5 optional features for improving stability near the side boundaries. Finally, Sec. 22.6 explains how to set configuration options for users. Usage of nesting in the sea ice model and the tidal scheme is described in Section 17.12 and 18.4, respectively.

22.1 Outline

In a set of nested grid models, a fine-resolution (child) model is embedded in a coarse-resolution (parent) model. In one-way nesting, values at the side boundary of the child model are given by the parent model. The side boundary data may be transferred both off-line and on-line. In off-line mode, the data needed to calculate side boundary values are written to files by first running the parent model, and the child model is executed reading these data and calculating the side boundary values by interpolation. In on-line mode, parent and child models are run in parallel. Simple coupler (Scup), which is originally developed by Yoshimura and Yukimoto (2008) for the communication among components of the MRI Earth System Model, is used to transfer data.

In two-way nesting, in addition to the transfer of the side boundary data from parent to child model, the result of the child model is reflected to the parent model in the embedded region. Data are exchanged on-line and Scup is used for data transfer.

The boundary between parent and child models are formed by connecting either the tracer points or the velocity points of the Arakawa B-grid arrangement (Figure 22.1). Table 22.1 shows the side boundaries suited for the available nesting methods. From experiences, we recommend setting the tracer points as the side boundary for one-way nesting (imposing Dirichlet boundary condition for the sea surface height and tracer equations) and the velocity points as the side boundary for two-way nesting (imposing Neumann boundary condition for the sea surface height and tracer equations). In one-way nesting, the child model is quite stable when the side boundary is set to the tracer points. In two-way nesting, imposing the conservation on water volume and tracer content for the set of parent and child models is more straightforward when the side boundary is set to the velocity (flux) points.

To be able to treat both side boundary conditions with a single topographic configuration, the domain of a child model is constructed by taking tracer points as the boundary of the main region (Figure 22.1a). Inside the boundary, the same topography as the parent model should be given for at least one parent grid cell, as written in Sec.22.2. By doing so, we may use the velocity points located inside the boundary by a half parent grid width as the parent-child boundary for two-way nesting (Figure 22.1b). The model domain needs additional two rows or columns of velocity cells outside the

Table22.1 Nesting methods available in MRI.COM and the recommended side boundary of a child model in each method. It should be noted that the region and the topography of a child model must be made as if tracer-points are used as the side boundary.

model run	on-line (coupler)	off-line (file I/O)
one-way nest	tracer point	tracer point
two-way nest	velocity point	unavailable

main region along each side boundary (hereinafter referred to as halo; the region shaded by dark green in Figure 22.1a).

Whether the side boundary of the child model is the tracer points or the velocity points, three columns or rows of the parent model are sufficient for a child model to fill both halo and boundary regions. Here, the boundary region refers to a region inside the boundary of the main region by one grid of the parent model. For example, $T_1 - T_3$ and $U_1 - U_3$ in Figure 22.1 are sufficient for filling the southern boundary (except for one-to-one nesting). In off-line one-way nesting, the parent model outputs the three columns or rows to files, and the child model reads them. In on-line nesting, the mapping table between parent and child grids should be prepared for the Scup library. The Scup library exchanges only necessary data among MPI processes of different models.

In this way, prognostic variables at the side boundary of the child model are directly replaced by those from the parent model. This is usually called the clamped method (Cailleau et al., 2008). This method does not guarantee conservation of tracers in contrast to the method where fluxes are imposed at the boundary, but the integration is quite stable.

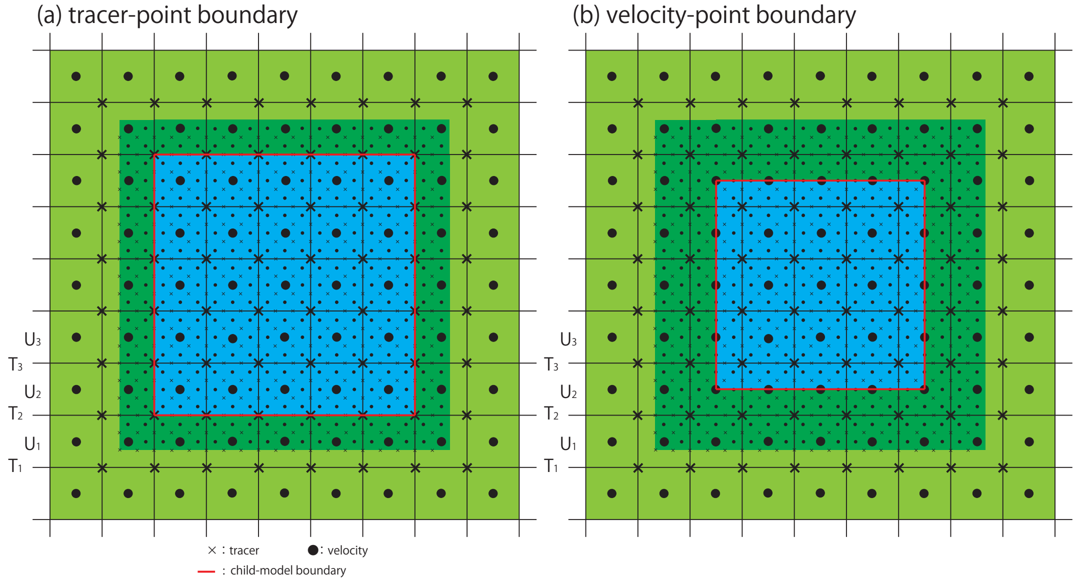

Figure22.1 Relation between parent (large symbols) and child (small symbols) grid points. The region shaded with blue (green) is the main region of the child (parent) model. The boundary of the main region is indicated by the red lines. In (a), the tracer points, and in (b), the velocity points, are set as the boundary between the two models. The domain of the child model should be constructed to include the dark green area, which is expanded outward by two velocity cells from the tracer point boundary (red line in (a)). This expanded region is used as halos for the main region. Variables in the halo region are obtained by using values of the parent model. Thus, for the southern boundary, three rows (T_1 - T_3 and U_1 - U_3) are sent to the child model from the parent model in both case (a) and (b). In two-way nesting, variables in the main region of the child model shown in (b) are sent to the parent model, except on the boundary line (red line).

Multistage nesting is available in MRI.COM, and on-line nesting and off-line nesting can be used together. This flexibility makes various kinds of experiments possible, e.g., a double on-line nesting experiment using global, North-Pacific, and Japanese coastal models, or an off-line one-way nesting experiment of a Japanese coastal model reading the side boundary values from an on-line nesting experiment for a set of global and North-Pacific models.

22.2 Grid and Topography

22.2.1 Parent (coarse-resolution) model

There is no particular issue in constructing the parent model. Because the grid size ratio between the child and parent models must be odd, the grid size of the child model may be taken into account in determining the grid size of the parent model. A grid size ratio of 1:3 or 1:5 is recommended.

22.2.2 Child (fine-resolution) model

a. Horizontal grid size

The domain of a child model is constructed by taking the tracer point as the boundary of the main region (the red line in Figure 22.1a) and adding the halo region (the dark green region), even if the velocity-point boundary will be used. Determining the grid arrangement of the child model and the corresponding parent model region (T_1 , U_1 in Figure 22.1a) is complicated, depending on presence or absence of the side boundaries and the ratio of grid division, as seen below. Users are advised to use the MXE package tool (`prep/nest/topo`, `prep/offnestsub`) to determine the grid size and other settings. With this tool, users can not only divide the parent model grid by a constant ratio, but also change the division ratio depending on the location.

Details of the horizontal grid division are as follows.

- the main internal region
  The distance between the velocity points of the parent model (i.e., T-cell) is divided equally in the main region of the child model (the blue region in Figure 22.1a). Thus, the T-cell arrangement is constructed, and then the velocity cell, which represents the model topography, is created. (Only when the parent model grid width is constant and the division ratio is constant, it can be considered that the velocity cell of the parent model is simply divided into equal parts.)
- the western and southern halo region
  There should be always two marginal velocity cells outside the main region, as shown by the dark green region in Figure 22.1a. The grid range of the parent model used to interpolate the values there depends on the division ratio.
- the eastern and northern halo region
  There should be two marginal velocity cells. However, if the boundary is filled by land and does not receive boundary data, one marginal velocity cell will be enough. The corresponding grid range of the parent model depends on the division ratio.

Our nesting scheme does not support division of the vertical layers. The same vertical grid intervals as the parent model must be used. The number of the model layers can be reduced, if the deepest topography of the child model is shallower than that of the parent model. (If a user prepares boundary data interpolated vertically, it is possible to run the child model with different vertical grid intervals by using the off-line nesting option.)

b. Topography of the child model

To achieve flux conservation, the child model should have the same topography as the parent model around the side boundary of the child model. It is recommended that the topography of the child model should have the same topography as the parent model for two parent velocity cells inward and one parent velocity cell outward relative to the boundary of the main region constructed by connecting tracer points (Figure 22.2).

c. Weighting ratio between parent and child models

For smoothness, prognostic values of the child model around the side boundary may be given as a weighted average of the solutions of the parent and child models. For variable ϕ at a grid point with index (i, j),

$$\phi_{ij}^{child} = \alpha_{ij}\phi_{ij}^{p} + (1 - \alpha_{ij})\phi_{ij}^{c}, \tag{22.1}$$

where ϕ^{child} is the value for the child model, ϕ^{p} is the parent model solution interpolated on the child grid, ϕ^{c} is the child model solution, and α is a spatially dependent weight between parent and child solutions.

Although this weight is introduced for a smooth transition from parent to child model solution, we recommend that the weight for the parent model to be unity in the halo and at the boundary of the main region and zero elsewhere. Instead, this weight is used to specify the boundary between the main regions of parent and child models. When the boundary is at the tracer points, the weight for the parent solution should be unity in the dark green region of Figure 22.1a. When the boundary is at the velocity points, the weight for the parent solution should be unity in the dark green region of Figure 22.1b. Weight must be prepared separately for the tracer and velocity points. (Incidentally, a radiation boundary condition may be imposed by elaborating weight values.)

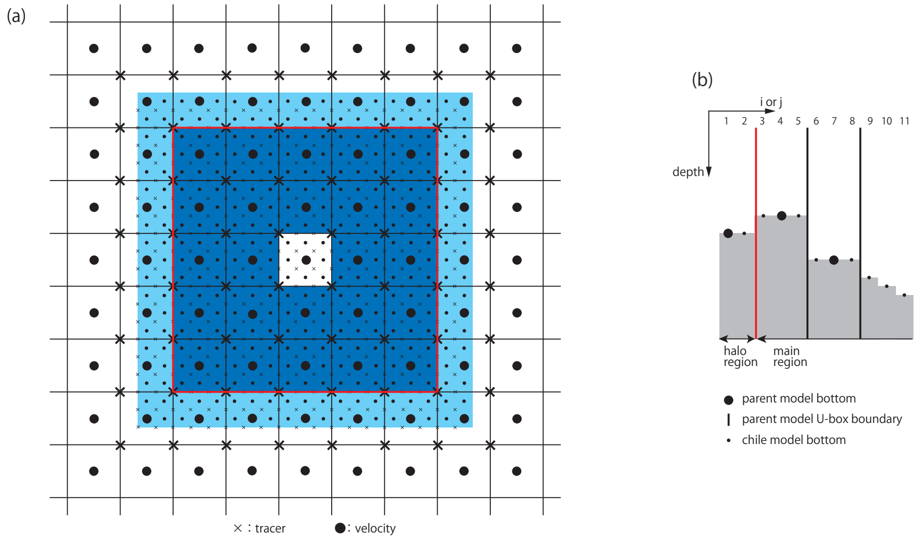

Figure22.2 (a) Condition on the topography of the child model around the side boundary. The topography of the child model should have the same topography as the parent model for two parent velocity cells inward (dark blue) and one parent velocity cell outward (light blue) relative to the boundary of the main region constructed by connecting tracer points (red rectangle). (b) Bathymetry near the boundary of the child model, when the grids of the parent model are divided by a factor of 3.

22.3 Intermodel transfer

Communication between models is both downscaling and upscaling in two-way nesting, whereas only downscaling in one-way nesting.

22.3.1 Downscaling

In the downscaling from parent to child model, the method of data transfer and interpolation is different between tracer point and velocity point.

a. Tracer points

It is straightforward to do a linear interpolation for variables on tracer points. Figure 22.3 shows the relative positions of tracer points of parent and child models. Because topography is common between parent and child models around the side boundary, all child tracer points in an ocean velocity cell of the parent model (bounded by the red rectangle) are ocean points. Values on these child tracer points are computed by using values at the four corner tracer points of the ocean velocity cell of the parent model (◎ in Figure 22.3). All of the four corner points are always available for bilinear interpolation to compute the tracer values of the child model. Thus, assuming that parent and child grid points coincide at (i_p, j_p) and (i_c, j_c), the tracer value at $(i_c + m, j_c + n)$, where $(0 \le m \le M)$ and $(0 \le n \le N)$, and M and N are the zonal and meridional nesting ratios, is computed as follows:

$$\phi_{i_c+m,j_c+n} = (1-w_x)(1-w_y)\phi^p_{i_p,j_p} + w_x(1-w_y)\phi^p_{i_p+1,j_p} + (1-w_x)w_y\phi^p_{i_p,j_p+1} + w_x w_y\phi^p_{i_p+1,j_p+1}, \tag{22.2}$$

where $w_x = (x^t_{i_c+m} - x^t_{i_p})/(x^t_{i_p+1} - x^t_{i_p})$ and $w_y = (y^t_{j_c+n} - y^t_{j_p})/(y^t_{j_p+1} - x^t_{j_p})$, x^t and y^t represent the tracer point in zonal and meridional coordinates, respectively.

b. Velocity points

Figure 22.4 shows the relative positions of velocity points of parent and child models. Unlike the tracer points, all of the four parent velocity points surrounding a child velocity point are not necessarily ocean points. When all parent grid points are ocean, bi-linear interpolation can be applied.

A problem arises for velocity points of the child model between an ocean point and a land point of the parent model (shaded by blue in Figure 22.5a), because only data at a single point is available for interpolation. The same problem also

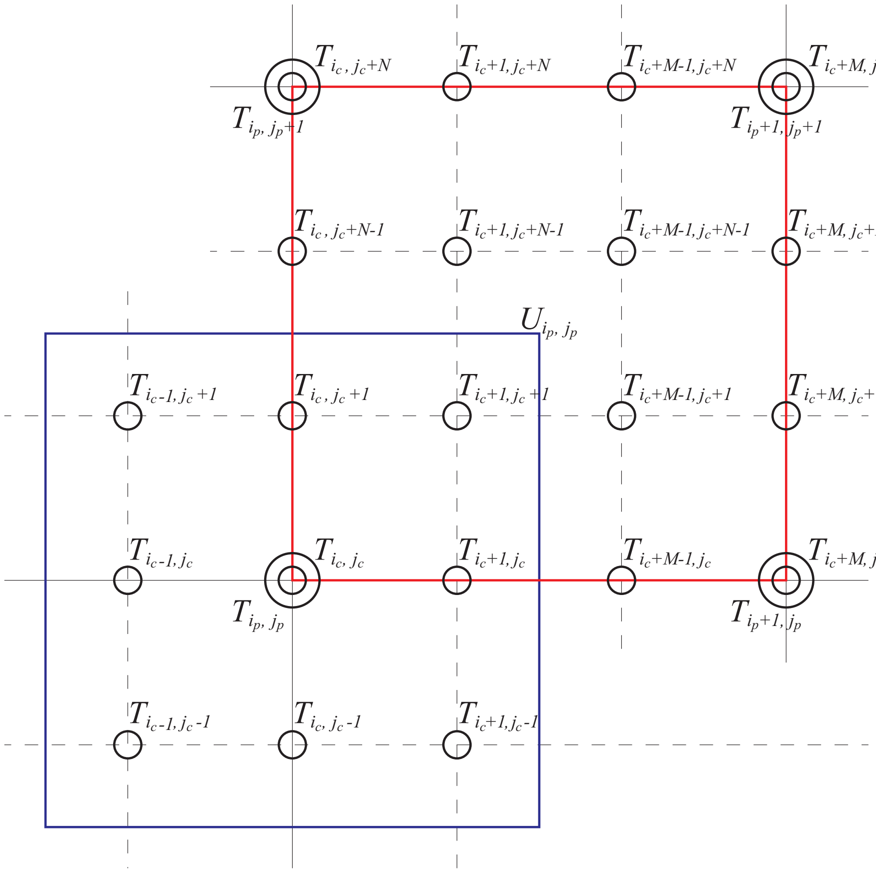
T_{i_c, j_c+N}
T_{i_c+1, j_c+N}
$T_{i_c+M\text{-}1, j_c+N}$
T_{i_c+M, j_c+N}
T_{i_p, j_p+1}
T_{i_p+1, j_p+1}
$T_{i_c, j_c+N\text{-}1}$
$T_{i_c+1, j_c+N\text{-}1}$
$T_{i_c+M\text{-}1, j_c+N\text{-}1}$
$T_{i_c+M, j_c+N\text{-}1}$
U_{i_p, j_p}
$T_{i_c\text{-}1, j_c+1}$
T_{i_c, j_c+1}
T_{i_c+1, j_c+1}
$T_{i_c+M\text{-}1, j_c+1}$
T_{i_c+M, j_c+1}
$T_{i_c\text{-}1, j_c}$
T_{i_c, j_c}
T_{i_c+1, j_c}
$T_{i_c+M\text{-}1, j_c}$
T_{i_c+M, j_c}
T_{i_p, j_p}
T_{i_p+1, j_p}
$T_{i_c\text{-}1, j_c\text{-}1}$
$T_{i_c, j_c\text{-}1}$
$T_{i_c+1, j_c\text{-}1}$

Figure22.3 Relative positions of tracer points of parent and child models (◎ and ○, respectively), coinciding at (i_p, j_p) and (i_c, j_c). The red and blue frames indicate U- and T-boxes in the parent model, respectively. M and N are the zonal and meridional nesting ratios.

arises for a velocity cell that has a partial velocity cell at its neighbor. The velocity for the partially blocked part (shaded by purple in Figure 22.5a) should be treated separately from the upper unblocked part.

In this case, the child model velocity perpendicular to the coast line (u^c) is assumed to be zero at the coast and linearly interpolated from the nearest ocean grid point of the parent model (Figure 22.5b). The component tangent to the coast line (v^c) is set to the same value at the nearest ocean grid point of the parent model (Figure 22.5c). By doing so, the interpolated velocity field yields nearly the same transport along the coast line as the parent model. To be specific, when the parent model has a partially shaved bottom cell at $(i_p - 1, j_p, k)$, velocities of the child model between $x^u(i_p - 1) \leq x^u(i) \leq x^u(i_p)$ are computed as follows:

$$u^c(i) = w_x u^p_{i_p} + (1 - w_x) u^p_{i_p-1}, \tag{22.3}$$

$$v^c(i) = w_x v^p_{i_p} + (1 - w_x) v^p_{i_p-1}, \qquad \text{if} \quad x^u(i) < x^t(i_p) \tag{22.4}$$

$$u^c(i) = \frac{\Delta z_{i_p-1,k}}{\Delta z_{i_p,k}} (w_x u^p_{i_p} + (1 - w_x) u^p_{i_p-1}) + \frac{\Delta z_{i_p,k} - \Delta z_{i_p-1,k}}{\Delta z_{i_p,k}} w u^p_{i_p}, \tag{22.5}$$

$$v^c(i) = \frac{\Delta z_{i_p-1,k}}{\Delta z_{i_p,k}} (w_x v^p_{i_p} + (1 - w_x) v^p_{i_p-1}) + \frac{\Delta z_{i_p,k} - \Delta z_{i_p-1,k}}{\Delta z_{i_p,k}} w v^p_{i_p}, \qquad \text{if} \quad x^t(i_p) < x^u(i) \tag{22.6}$$

where $w_x = (x^u(i) - x^u(i_p - 1))/(x^u(i_p) - x^u(i_p - 1))$ and $w = (x^u(i) - x^t(i_p))/(x^u(i_p) - x^t(i_p))$.

To apply this method to the barotropic transport, the barotropic transport is first divided by the height of the water column to obtain the vertically uniform barotropic velocity. This velocity is given to all the velocity points in the water column and then the same method as used for the 3-D velocity field is applied.

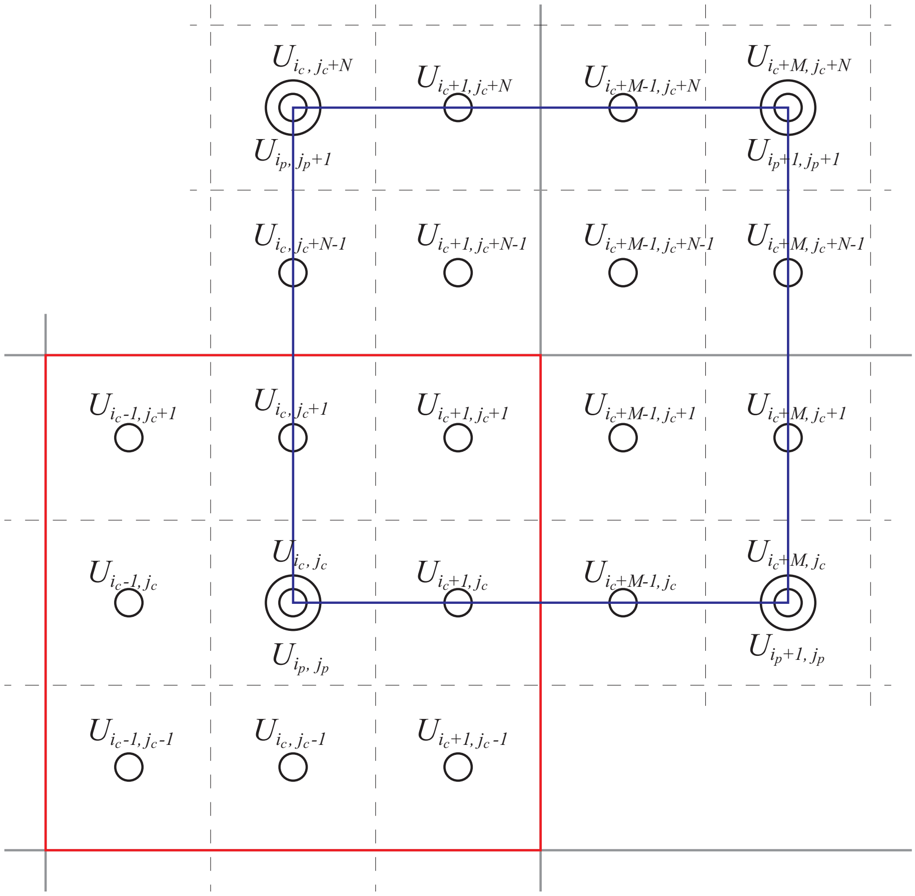

Figure22.4 Same as Figure 22.3 except for velocity point.

22.3.2 Upscaling

In two-way nesting, values of the parent model in the child model's main region are filled with those based on the child model. The region to receive feedback in the parent model is the main region of the child model excluding the boundary line (inside the red line of Figure 22.1b). MRI.COM has two options for upscaling, replacement and nudging.

a. Replacement

One option is to simply replace variables of the parent model by those of the child model. Ideally, fine scale features that cannot be resolved by the parent model should be removed in upscaling. In MRI.COM, simple averaging within a parent grid is used (averaging over blue rectangle in Figures 22.3 and 22.4), though more sophisticated filters have been proposed (e.g., Debreu et al., 2012). Because applying a full averaging for the entire embedded region is costly in terms of data transfer between models, a full averaging is only applied to the 2 to 3 grid points inside the parent-child boundary,

$$\phi^p_{i_p,j_p} = \frac{1}{M \times N} \sum_{i,j} \phi^c_{i,j}, \quad i_c - i_m \le i \le i_c + i_m \quad \text{and} \quad j_c - j_n \le j \le j_c + j_n, \tag{22.7}$$

where $i_m = (M-1)/2$ and $j_n = (N-1)/2$. For the region further inside the boundary, values on the child grid points that coincide with the parent grid points may be used without filtering,

$$\phi^p_{i_p,j_p} = \phi^c_{i_c,j_c}, \quad x^t(i_p) = x^t(i_c) \quad \text{and} \quad y^t(j_p) = y^t(j_c). \tag{22.8}$$

This treatment does not cause a problem because this inner region away from the boundary does not affect the temporal evolution of the physical field in the main region of the parent model. (There is an option to omit communication in this inner region for speeding up, `flg_feedback_repl_bnd`.)

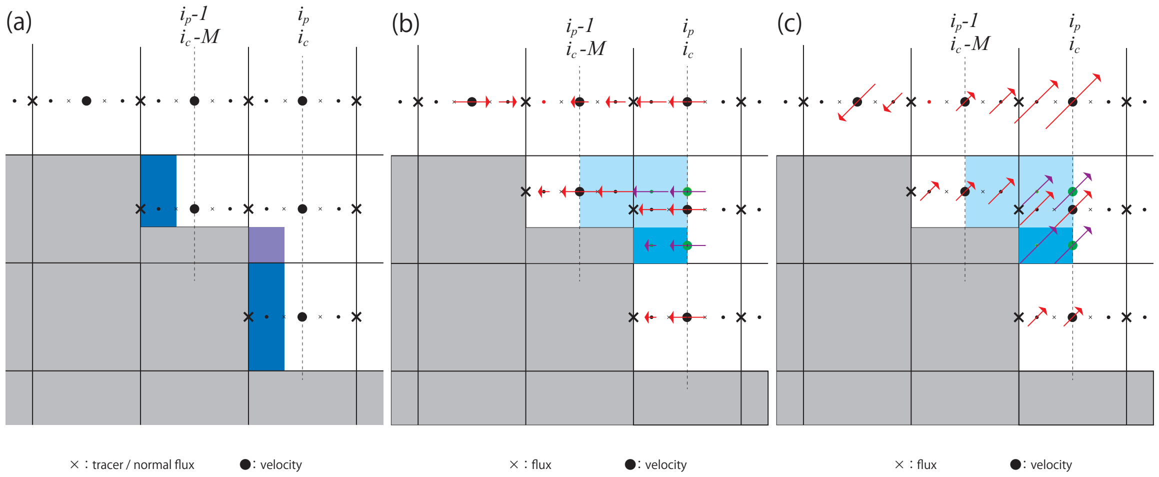

Figure22.5 Interpolation of the parent model velocity to the child grid with bottom topography. (a) For velocity points of the child model between an ocean point and a land point of the parent model (shaded with blue or purple), only one data is available for interpolation. Examples of (b) normal and (c) tangential component of velocity relative to the coast line (red arrows). For an ocean velocity cell next to a partial cell, the cell is divided into a part blocked by a wall and a part unblocked (green point). Linear interpolation is used for an unblocked part and special treatment is applied to the blocked part (purple arrows). The velocity is obtained as a weighted average of the two parts.

b. Nudging

Nudging is another option to reflect the child result to the parent model. In MRI.COM, tracer values of the child model may be used as reference values (ϕ^p_C) to which parent tracer fields (ϕ^p) are nudged,

$$\frac{\partial \phi^p}{\partial t} = -\frac{1}{\gamma}(\phi^p - \phi^p_C), \tag{22.9}$$

where γ is a restoring time scale usually taken to be very small, about 0.1 day. Only tracer fields are corrected in the parent model. Neither velocity nor sea surface height is modified. Our assumption lying behind this treatment is that reflecting only slowly evolving quasi-geostrophic baroclinic fields is enough for a parent model in many applications.

22.4 Conservation

Conservation of sea water volume and scalar quantities are generally preferable. This is especially so for the case of climate applications with long-term integration. To achieve this in nesting, MRI.COM has options to adjust side boundary and surface fluxes in the child model, or source / sink due to nudging in the parent model.

22.4.1 Side boundaries

Adjustment of the flux at the side boundary is performed as follows. Consider an evolution equation for a quantity ϕ in a simple flux form:

$$\frac{\partial \phi}{\partial t} = -\nabla \cdot \mathbf{f}, \tag{22.10}$$

where $\mathbf{f} = (f_x, f_y)$, which are due to either advective or diffusive fluxes. Integration (22.10) over each model's main region (Figure 22.6) gives

$$\frac{\partial \Phi^p_{\Omega-\omega}}{\partial t} \equiv \int_{\Omega-\omega} \frac{\partial \phi}{\partial t} dS = -\int_{\partial\Omega} \mathbf{f}^p \cdot \mathbf{n} dl \tag{22.11}$$

for parent model's main region and

$$\frac{\partial \Phi^c_{\omega}}{\partial t} \equiv \int_{\omega} \frac{\partial \phi}{\partial t} dS = -\int_{\partial\omega} \mathbf{f}^c \cdot \mathbf{n} dl \tag{22.12}$$

for child model's main region. If integral values of the flux normal to the boundary (that is, r.h.s.) cancel between parent and child models, the sum of integral values of the tendency of ϕ in each model's main region is conserved,

$$\frac{\partial \Phi^p_{\Omega-\omega}}{\partial t} + \frac{\partial \Phi^c_{\omega}}{\partial t} = 0. \tag{22.13}$$

However, this conservation does not always hold owing to discretization. Some adjustment is needed.

In MRI.COM, conservation of water volume and tracers is achieved by adjusting boundary fluxes computed in each model. Different approaches are taken for advective and diffusive fluxes.

For advective fluxes, fluxes of the child model are adjusted to match those of the parent model. Fluxes for tracer cells of the child model (f^c_y) are area-integrated along the side boundary section over the width of a tracer cell of the parent model (Figure 22.7a),

$$F^c_y(i_p) = \sum_i f^c_y(i) \times \Delta s^c(i), \quad i_c - i_m \le i \le i_c + i_m, \tag{22.14}$$

where Δs^c is the area of the side boundary of a tracer cell of the child model and $i_m = (M-1)/2$. This is compared with the flux for the tracer cell of the parent model (f^p_y) (Figure 22.7b),

$$F^p_y(i_p) = f^p_y(i_p) \times \Delta s^p(i_p), \tag{22.15}$$

where Δs^p is the area of the side boundary of a tracer cell of the parent model. On the basis of this comparison, fluxes of the child model are adjusted using a quantity $f^c_{y\ \mathrm{adj}}(i_p)$ given by

$$f^c_{y\ \mathrm{adj}}(i_p) = (F^p_y(i_p) - F^c_y(i_p))/\Delta S^c(i_p), \tag{22.16}$$

where

$$\Delta S^c(i_p) = \sum_i \Delta s^c(i), \quad i_c - i_m \le i \le i_c + i_m. \tag{22.17}$$

The adjusted flux $f^c_{y\ \mathrm{mod}}$ is given by

$$f^c_{y\ \mathrm{mod}}(i) = f^c_y(i) + f^c_{y\ \mathrm{adj}}(i_p), \quad i_c - i_m \le i \le i_c + i_m, \tag{22.18}$$

so that the flux integral in the child model equals the parent model flux for parent's tracer cell.

For diffusive fluxes, the sum of fluxes in the child model (f^c_x) may be given to the parent model instead of adjusting the child model flux to the parent model flux as in the advective fluxes (Figure 22.8),

$$F^p_{x\ \mathrm{mod}}(j_p) = f^p_{x\ \mathrm{mod}}(j_p)\Delta s^p(j_p) = \sum_j f^c_x(j) \times \Delta s^c(j), \quad j_c - j_n \le j \le j_c + j_n. \tag{22.19}$$

$F^p_{x\ \mathrm{mod}}(j_p)$ replaces the flux calculated on the parent grid $F^p_x(j_p)$ at the boundary.

The flux adjustment should be applied to any horizontal flux of volume and tracers. Currently, this is implemented in the following fluxes:

- Volume flux in the 3-D continuity equation
- Advective volume flux in the 2-D continuity equation (both barotropic and baroclinic time step)
- Diffusive volume flux in the 2-D continuity equation (both barotropic and baroclinic time step)
- Tracer advection flux by Second Order Moment and MPDATA schemes
- Tracer diffusion flux by horizontal diffusion
- Tracer flux by isoneutral transport scheme (isoneutral diffusion plus Gent-McWilliams parameterization)
- Sea ice advection plus diffusion flux (area fraction, volume, thermal energy) with some special treatment. See Chapter 17 for details.

22.4.2 Surface fluxes

In order to impose conservation on some properties in a set of nested models, surface fluxes must be also taken into account.

In MRI.COM, you may impose conservation condition for the following cases by adjusting the net surface flux.

- Volume of sea water in an ocean-only mode (`WADJ` option).

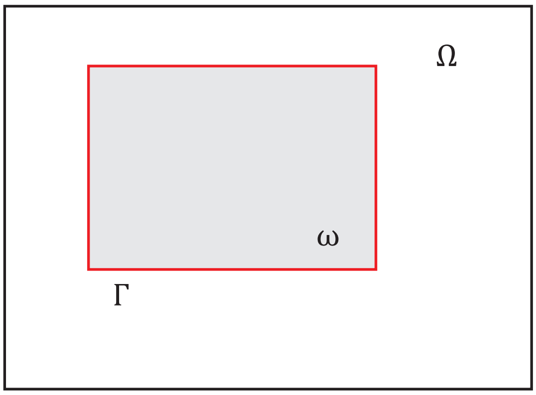

Figure22.6 Schematic for considering conservation in a system of coarse and fine grid models. The coarse grid (parent) model region is denoted by Ω and the fine grid (child) model region is denoted by ω. The boundary between models is denoted by Γ. These notations follow Figure 1 of Debreu et al. (2012).

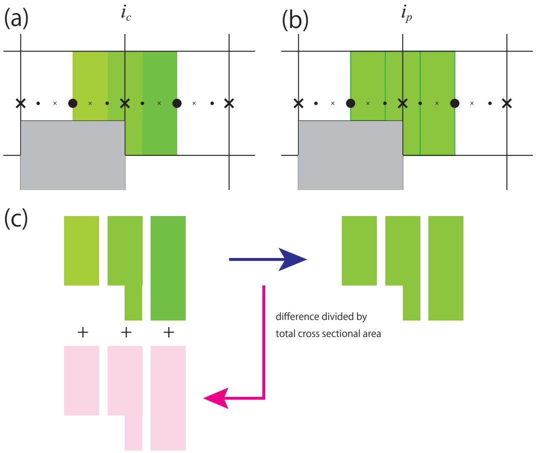

Figure22.7 Schematic explaining flux adjustment for a child model. In this zonal-vertical section, tracer points coincide at i_p and i_c. (a) Meridional fluxes are computed in the child model using the values interpolated from the parent model. Flux is integrated over the corresponding parent grid and this is compared with that of the parent model (b). (c) Difference between the integrated fluxes is divided by the integration area (unit parent grid area) to give flux correction for the child model (pink).

- Salt content in the ocean - sea ice system in any mode (`l_sss_rst_cnsv = .true.` in namelist `nml_sss_restore`)
- Heat content and volume of sea water in a coupled mode (`SCUPCGCM` and `NPUTFLUX` options for parent and `NGETFLUX` option for child)

For example, to conserve sea water volume in the system (`WADJ` option), a globally constant adjustment factor $f_{w\,\mathrm{adj}}$ is used so that

$$\int_{\Omega-\omega}(f_w^p + f_{w\,\mathrm{adj}})dS + \int_{\omega}(f_w^c + f_{w\,\mathrm{adj}})dS = 0, \tag{22.20}$$

where f_w^p and f_w^c represent total surface water fluxes of parent and child models, respectively. The adjustment factor is given by

$$f_{w\,\mathrm{adj}} = -(\int_{\Omega-\omega} f_w^p dS + \int_{\omega} f_w^c dS)/(S_{\Omega-\omega} + S_{\omega}), \tag{22.21}$$

where $S_{\Omega-\omega} = \int_{\Omega-\omega} dS$ and $S_{\omega} = \int_{\omega} dS$.

To avoid a drift of total salt content of the system (`l_sss_rst_cnsv = .true.` in namelist `nml_sss_restore`) in the case of surface salinity restoring, a globally constant adjustment factor $f_{s\,\mathrm{adj}}$ is used so that

$$\int_{\Omega-\omega}(f_s^p + f_{s\,\mathrm{adj}})dS + \int_{\omega}(f_s^c + f_{s\,\mathrm{adj}})dS = 0, \tag{22.22}$$

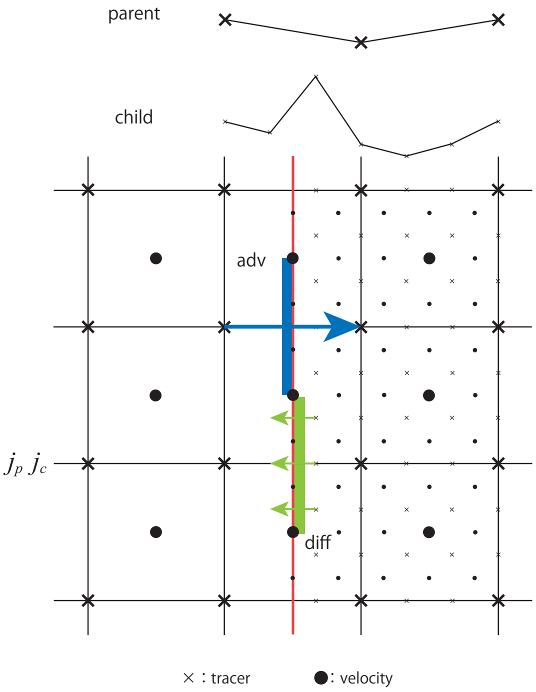

Figure22.8 Computation of parent model's zonal diffusive flux across the boundary, coinciding at j_p and j_c in terms of the tracer point. In this case, the sum of diffusive fluxes of the child model (green) is used as the diffusive flux for the parent model at j_p. This is contrasted with the advective flux (blue) where the parent model flux is used to adjust the child model flux.

where f_s^p and f_s^c represent surface salinity flux due to restoring in parent and child models, respectively. The adjustment factor is given by

$$f_{s\,\mathrm{adj}} = -\Big(\int_{\Omega-\omega} f_s^p\, dS + \int_{\omega} f_s^c\, dS\Big)/(S_{\Omega-\omega} + S_{\omega}). \tag{22.23}$$

When a set of nested grid models is coupled with an atmospheric model (SCUPCGCM and NPUTFLUX options for parent and NGETFLUX option for child), the sum of sea surface heat flux of the ocean model must equal that of the atmospheric model. To achieve this, a globally constant adjustment factor $f_{q\,\mathrm{adj}}$ is used so that

$$\int_{\Omega-\omega} (f_q^p + f_{q\,\mathrm{adj}})dS + \int_{\omega} (f_q^c + f_{q\,\mathrm{adj}})dS = \int_{\Omega} f_q^a\, dS, \tag{22.24}$$

where f_q^p, f_q^c, and f_q^a represent net surface heat flux in parent oceanic, child oceanic, and atmospheric models, respectively. The adjustment factor is given by

$$f_{q\,\mathrm{adj}} = \Big[\int_{\Omega} f_q^a\, dS - \Big(\int_{\Omega-\omega} f_q^p\, dS + \int_{\omega} f_q^c\, dS\Big)\Big]/(S_{\Omega-\omega} + S_{\omega}). \tag{22.25}$$

$f_{q\,\mathrm{adj}}$ is used to offset the net longwave flux in the ocean models.

These adjustments are applicable only to on-line two-way nesting by using pre-communicator of Scup.

22.4.3 Nudging

To conserve the total amount of tracer of the parent model when a tracer of the parent model is nudged toward that of the child model, an additional flux adjustment is needed. A globally constant adjustment factor S_{adj}^{ϕ} is added to the right hand side of (22.9) so that

$$\int_{\omega} [-\frac{1}{\gamma}(\phi^p - \phi_C^p) + S_{\mathrm{adj}}^{\phi}]dV = 0. \tag{22.26}$$

The adjustment factor is given by

$$S_{\mathrm{adj}}^{\phi} = \int_{\omega} \frac{1}{\gamma}(\phi^p - \phi_C^p)dV/V_{\omega}, \tag{22.27}$$

where $V_\omega = \int_\omega dV$. By doing so, nudging only changes the distribution of the tracer in the parent model without affecting its total amount. MRI.COM offers this adjustment as an option (`flg_feedback_cnsv_ndg` = `.true.` in namelist `nml_nest_par`).

22.5 Stabilization of parent-child boundaries

Appearance of noises and discontinuity around the parent-child boundary is almost unavoidable. It is a common practice to set a "sponge" region around the boundary to reduce them. (The word "sponge" is used because it works to relieve shock.) Several methods are available for MRI.COM.

22.5.1 Giving 2-D distribution of diffusion and viscosity coefficient

A simple choice would be to give 2-D distribution of time-invariant coefficient for diffusion and viscosity operators that have been selected for each model. For example, when a common operator is selected for both parent and child models, the coefficients of the child model should be the same as those of the parent model around the side boundary to obtain similar results there. The coefficients in the boundary region should be connected smoothly with the small coefficients in the interior.

Among the mixing operators available in MRI.COM, the additional harmonic viscosity is most effective for getting a smooth solution around the parent/child boundary. However, this scheme should be used with caution because the harmonic viscosity may even damp resolved processes of the parent model. These additional diffusivity and viscosity are available, whether nesting is one-way or two-way.

22.5.2 Scheme specialized for two-way nesting

MRI.COM has some schemes specialized for two-way nesting in order to make the model stable.

a. Parent model

A parent model can use an additional Laplacian operator for both velocity and tracer. This operator may be applied to the velocity points on the boundary (red line in Figure 22.9) and to the tracer point (blue line) just outside the feedback region where parent-model values are upscaled (replaced) by child-model values (light blue region). Laplacian is calculated in the direction perpendicular to the boundary.

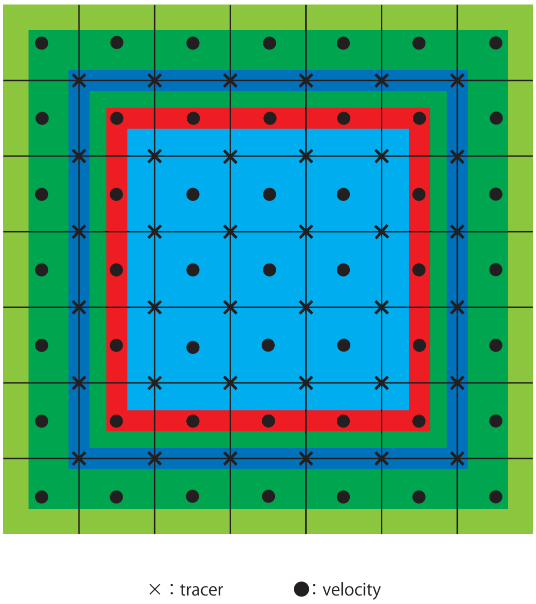

Figure22.9 Same as Figure 22.1b except for indicating of sponge regions for a parent model. Laplacian diffusion and viscosity operators are applied in the blue and red bands, respectively.

b. Child model

For a child model, Laplacian operator is applied to the difference between the child value and the parent value mapped on the child grid in the sponge region following Section 2.5 of Debreu et al. (2012),

$$\frac{\partial \phi^c}{\partial t} = \nabla \cdot \kappa \nabla (\phi^c - \phi^c_P), \tag{22.28}$$

where ϕ^c is the child model value, ϕ^c_P is the parent value mapped on the child grid, and κ is viscosity/diffusion coefficient. The operator is prepared for velocity and tracers. The width of the sponge region is two parent velocity cells from the boundary constructed by connecting tracer cells (region shaded with purple in Figure 22.10).

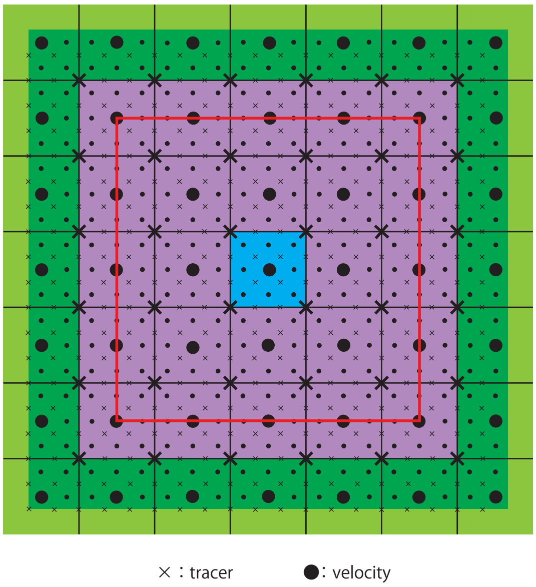

Figure22.10 Same as Figure 22.1b except for indicating of sponge region for child model. Laplacian diffusion/viscosity operator is applied to the difference between child and parent values in the region shaded with purple.

22.6 Usage

22.6.1 Offline one-way nesting (`OFFNESTPAR`, `OFFNESTSUB` option)

a. Compilation

The model option for the parent model is `OFFNESTPAR` (OFFline-NESTing PARent), and that for the child model is `OFFNESTSUB`. These model options and the model name should be specified in `configure.in`.

b. Runtime specification for the parent model

At run time, namelists `nml_parentmodel_grid` (Table 22.2) and `nml_parentmodel_package` (Table 22.3) must be specified. They are read by `Parent/parent_common.F90` on execution. Users should set the namelist based on the MXE tool (`prep/offnestsub`). See `docs/README_Namelist.md` for details.

Following files are output from the parent model to be read by the child model.

- Three rows or columns of data for each side boundary of specified packages. For package (`package_name`), 'uv', 'barotropic', 'active_tracer', 'passive_tracer', 'vmixcoef', 'nohkim', 'my25', 'gls', 'ice_thermodynamics', 'ice_dynamics', and 'tide' are available.
- Land-sea index of the parent model (`file_parent_sea_index_out`).
- Latitude and longitude of the parent model (`file_parent_grid_out`).

The daily output of side boundary data will work in running the child model.

Table22.2 Namelist nml_parentmodel_grid for the off-line parent model

variable name	description	usage
sub_region_ifirst	the western end point in the parent model to be used to interpolate data (T-point)	
sub_region_ilast	the eastern end point in the parent model to be used to interpolate data (T-point)	
sub_region_jfirst	the southern end point in the parent model to be used to interpolate data (T-point)	
sub_region_jlast	the northern end point in the parent model to be used to interpolate data (T-point)	
dx_ratio_w	refinement ratio at west boundary	odd number, zero for no nesting along this side
dx_ratio_e	refinement ratio at east boundary	odd number, zero for no nesting along this side
dy_ratio_s	refinement ratio at south boundary	odd number, zero for no nesting along this side
dy_ratio_n	refinement ratio at north boundary	odd number, zero for no nesting along this side
file_parent_sea_index_out	land-sea index of the parent model	
file_parent_grid_out	latitude-longitude of the parent model	

Table22.3 Namelist nml_parentmodel_package for the off-line parent model

variable name	description	usage
package_name		
file_root_name	root name of the output boundary data files	
bndr_dt_sec	time interval [sec] of output	must be multiple of model time step
nrec_first	the first record number of the output files	= 1 : create new file > 1 : append to old file
l_write_initial_state		.true. : output initial state .false. : do not output initial state

c. Runtime specification for the child model

At run time, namelists nml_submodel_grid (Table 22.4), nml_submodel_bnd_cnd (Table 22.5), and nml_submodel_package (Table 22.6) must be specified. They are read by Sub/sub_common.F90 on execution. See docs/README_Namelist.md for details.

To run the child model, prepare the following two data files:

- file_sub_wgt_t: The contribution ratio between parent and child models around the boundary (T-point),
- file_sub_wgt_u: The contribution ratio between parent and child models around the boundary (U-point).

In addition, produce the following files by running the parent model:

- file_root_name in nml_submodel_package: Three rows or columns of the necessary elements at each side boundary,
- file_parent_sea_index: Land-sea index of the parent model,
- file_parent_grid: Latitude and longitude of the parent model.

Note the following when preparing barotropic boundary data. Since time filtering is used to feedback the result of the barotropic equations to the baroclinic mode, the barotropic equations are integrated over the end of the baroclinic time. Thus, the child model needs barotropic data longer than the original output from the parent model. To fulfill this need, prepare the barotropic data file to include one additional time record. For example, in a historical (yearly) run, the second record of the following year of the parent model should be appended to the present year data. If the additional record is not prepared, MRI.COM integrates the barotropic equation by repeating the final record just as needed.

Table22.4 Namelist nml_submodel_grid for the off-line child model

variable name	description	usage
parent_grid_im	Total number of grid points in X direction of the parent model	
parent_grid_jm	Total number of grid points in Y direction of the parent model	
sub_region_ifirst	the western end point in the parent model to be used to interpolate data (T-point)	
sub_region_ilast	the eastern end point in the parent model to be used to interpolate data (T-point)	
sub_region_jfirst	the southern end point in the parent model to be used to interpolate data (T-point)	
sub_region_jlast	the northern end point in the parent model to be used to interpolate data (T-point)	
dx_ratio_w	refinement ratio at west boundary	odd number, zero for no nesting along this side
dx_ratio_e	refinement ratio at east boundary	odd number, zero for no nesting along this side
dy_ratio_s	refinement ratio at south boundary	odd number, zero for no nesting along this side
dy_ratio_n	refinement ratio at north boundary	odd number, zero for no nesting along this side
file_parent_grid	latitude-longitude of the parent model	
file_parent_sea_index	land-sea index of the parent model	

Table22.5 Namelist nml_submodel_bnd_cnd for the off-line child model

variable name	description	usage
file_sub_wgt_t	filename of weighing factor for parent/child model around the boundary (T-points)	
file_sub_wgt_u	filename of weighing factor for parent/child model around the boundary (U-points)	

Table22.6 Namelist nml_submodel_package for the off-line child model

variable name	description	usage
package_name		
file_root_name	root name of the output boundary data files	
bndr_dt_sec	time interval [sec] of the input data	
bndr_first_date	integer array(6) indicating date and time (YMDMHS) of the first record	
num_bndr_record	the last record number of the input files	

22.6.2 On-line nesting (SCUPNEST option)

a. Compilation

For on-line mode, SCUPNEST option should be specified. Also, the name of the model should be explicitly specified such as NAME_MODEL = *modelname*. The model option for the parent model is PARENT, and that for the child model is SUB.

In ocean-only mode, surface fluxes are calculated in each model, preferably on the basis of common atmospheric state.

When the parent model is coupled with an atmospheric model (SCUPCGCM), the parent model receives surface fluxes from the atmospheric model. The surface flux may be sent from parent to child model. In the parent (child) model, NPUTFLUX (NGETFLUX) option must be selected.

These model options and the model name should be specified in configure.in.

b. Runtime specification: NAMELIST_SCUP

In on-line mode, parent and child models are run at the same time and Scup (simple coupler) by Yoshimura and Yukimoto (2008) is used to exchange data. User should tell the coupler how data are exchanged between parent and child models via a namelist file NAMELIST_SCUP. A template of NAMELIST_SCUP is available from MRI.COM execution environment (MXE). Mapping tables for data transfer used by Scup can be created in MXE. User may comment out unnecessary data exchange according to the model and runtime options chosen for a specific experiment. An example of NAMELIST_SCUP is listed in the following. In this example, the model name of the parent model is GLOBAL and that of the child model is NP01.

An example of NAMELIST_SCUP

```
# pre-communicator
#
# (parent -> child)
 &nam_scup_pre model_put=’GLOBAL’, model_get=’NP01’, type=’REAL8’ /
 &nam_scup_pre var_put=’ALONTC’,  var_get=’ALONTC’,  dst_get=’ALL’ /
 &nam_scup_pre var_put=’ALATTC’,  var_get=’ALATTC’,  dst_get=’ALL’ /
 &nam_scup_pre var_put=’ALONUC’,  var_get=’ALONUC’,  dst_get=’ALL’ /
 &nam_scup_pre var_put=’ALATUC’,  var_get=’ALATUC’,  dst_get=’ALL’ /
# activate if l_sss_rst_cnsv = .true.
 &nam_scup_pre var_put=’SFLXADJ’, var_get=’SFLXADJ’, dst_get=’ALL’ /
#
# (child -> parent)
 &nam_scup_pre model_put=’NP01’, model_get=’GLOBAL’, type=’REAL8’ /
# activate if l_sss_rst_cnsv = .true. and flg_trcflux_cnsv=.true.,
 &nam_scup_pre var_put=’SFLXSUB’, var_get=’SFLXSUB’, dst_get=’ALL’ /
#
# main-communicator
#
# (parent -> child) side boundary prognosticated variables (ALWAYS activate)
 &nam_scup model_put=’GLOBAL’    , model_get=’NP01’   ,
    comp_put =’BAROCLIP_GLOBAL’ , comp_get =’BAROCLIS_NP01’   ,
    grid_put =’BCLI3DP_GLOBAL_T’, grid_get =’BCLI3DS_NP01_T’  ,
    fl_remap=’../data-np/rmp_tw_p2ct_3d.d’ /
 &nam_scup
    var_put=’TRC_SIDE01’, var_get=’TRC_SIDE01’, intvl=-1, lag=0, flag=’SNP’/
 &nam_scup
    var_put=’TRC_SIDE02’, var_get=’TRC_SIDE02’, intvl=-1, lag=0, flag=’SNP’/
#
# (child -> parent) for replace (activate IF flg_feedback_replace=.true.)
 &nam_scup model_put=’NP01’    , model_get=’GLOBAL’   ,
    comp_put =’BAROCLIS_NP01’   , comp_get =’BAROCLIP_GLOBAL’,
    grid_put =’BC3DS_NP01_TC’   , grid_get =’BC3DP_GLOBAL_TA’,
    fl_remap=’../data-np/rmp_c2pt_merge_3d.d’ /
 &nam_scup
   var_put=’TRC_REPL01’, var_get=’TRC_REPL01’, intvl=-1, lag=0, flag=’SNP’/
 &nam_scup
   var_put=’TRC_REPL02’, var_get=’TRC_REPL02’, intvl=-1, lag=0, flag=’SNP’/
```

c. Runtime specification for parent model

At run time, namelist nml_nest_par (Table 22.7) must be specified. They are read by nest_scup_par.F90 on execution. See docs/README_Namelist.md for details. If sponge layer is used for the parent model (flg_sponge = .true.), the two-dimensional distribution of the diffusion and viscosity coefficient in the sponge layer is read from a file whose name is given by namelist nml_par_diff (Table 22.8) and nml_par_visc (Table 22.9).

Table22.7 Namelist `nml_nest_par` for the on-line parent model

variable name	units	description	usage
`flg_carbon`	logical	Send surface CO2 flux or not	
`flg_verbose`	logical	Output details of the processing	
`flg_sponge`	logical	Sponge region is set just outside the boundary	
`flg_sponge_sub`	logical	Send data to child model's sponge region	
`flg_feedback_nudge`	logical	Use nudging of tracers as two-way feedback	default = `.false.`
`flg_feedback_replace`	logical	Use replacing of all fields as two-way feedback	default = `.false.`
`flg_feedback_repl_bnd`	logical	Replace boundary region only in two-way feedback	default = `.false.`
`flg_volflux_cnsv`	logical	Impose conservation of volume flux at u(v)star for child model	default = `.false.`
`flg_iceflux_cnsv`	logical	Impose conservation of sea ice flux at u(v)star for child model	default = `.false.`
`flg_send_sshhdif_c2p`	logical	Send SSH diffusion flux from child to parent	default = `.false.`
`flg_trcflux_cnsv`	logical	Impose conservation of tracer flux at u(v)star for child model	default = `.false.`
`flg_send_trchdif_p2c`	logical	Send tracer lateral diffusion flux from parent to child	default = `.false.`, choose either p2c or c2p or neither
`flg_send_trchdif_c2p`	logical	Send tracer lateral diffusion flux from child to parent	default = `.false.`, choose either p2c or c2p or neither
`flg_send_trcidif_p2c`	logical	Send tracer isopycnal diffusion flux from parent to parent	default = `.false.`, choose either p2c or c2p or neither
`flg_send_trcidif_c2p`	logical	Send tracer isopycnal diffusion flux from child to parent	default = `.false.`, choose either p2c or c2p or neither
`chfbc_sub_day`	day	restoring time of received tracer in day	
`scup_norecv`	1	undefined value of received tracer given by scup	
`nest_depth`	integer	nesting is operated from surface to `nest_depth` level	
`num_sub_models`	integer	the number of child (sub) models	default = 1
`file_upscale`	character	Name of the file that contains mask of the upscale region	

Table22.8 Namelist `nml_par_diff` for the on-line parent model

variable name	description	usage
`file_sponge_par_diff`	2D distribution of horizontal diffusivity just outside the boundary	

Table22.9 Namelist `nml_par_visc` for the on-line parent model

variable name	description	usage
`file_sponge_par_visc`	2D distribution of horizontal viscocity just outside the boundary	

d. Runtime specification for child model

At run time, namelist `nml_submodel_grid` (Table 22.10) and `nml_nest_sub` (Table 22.11) must be specified. They are read by `nest_scup_sub.F90` on execution. See `docs/README_Namelist.md` for details.

The two dimensional distribution of the diffusion and viscosity coefficient in the sponge layer is optionally read from a file whose name is given by namelist `nml_sub_diff` (Table 22.12), `nml_sub_visc` (Table 22.13) (if `flg_sponge_sub` = `.true.`), `nml_sub_isopyc` (Table 22.14), and `nml_sub_ssh_diff` (Table 22.15).

Table22.10 Namelist `nml_submodel_grid` for the on-line child model

variable name	description	usage
`parent_grid_im`	Total number of grid points in X direction of the parent model	
`parent_grid_jm`	Total number of grid points in Y direction of the parent model	
`sub_region_ifirst`	the western end point in the parent model to be used to interpolate data (T-point)	
`sub_region_ilast`	the eastern end point in the parent model to be used to interpolate data (T-point)	
`sub_region_jfirst`	the southern end point in the parent model to be used to interpolate data (T-point)	
`sub_region_jlast`	the northern end point in the parent model to be used to interpolate data (T-point)	
`dx_ratio_w`	refinement ratio at west boundary	odd number, zero for no nesting along this side
`dx_ratio_e`	refinement ratio at east boundary	odd number, zero for no nesting along this side
`dy_ratio_s`	refinement ratio at south boundary	odd number, zero for no nesting along this side
`dy_ratio_n`	refinement ratio at north boundary	odd number, zero for no nesting along this side
`l_parent_tripolar`		`.true.` : The parent model uses `TRIPOLAR` grid

Table22.11 Namelist `nml_nest_sub` for the on-line child model

variable name	units	description	usage
`file_sub_wgt_t`	character	filename of weighing factor for parent/child model around the boundary (T-points)	
`file_sub_wgt_u`	character	filename of weighing factor for parent/child model around the boundary (U-points)	
`file_sub_wgt_x`	character	filename of weighing factor for parent/child model around the boundary (Ustar-points)	
`file_sub_wgt_y`	character	filename of weighing factor for parent/child model around the boundary (Vstar-points)	
`flg_carbon`	logical	Send surface CO2 flux or not	
`flg_verbose`	logical	Output details of the processing	
`flg_sponge_sub`	logical	Sponge region is set just inside the boundary	
`flg_feedback_nudge`	logical	Use nudging of tracers as two-way feedback	default = `.false.`
`flg_feedback_replace`	logical	Use replacing of all fields as two-way feedback	default = `.false.`
`flg_volflux_cnsv`	logical	Impose conservation of volume flux at u(v)star for the child model	default = `.false.`
`flg_iceflux_cnsv`	logical	Impose conservation of sea ice flux at u(v)star for the child model	default = `.false.`
`flg_send_sshhdif_c2p`	logical	Send SSH diffusion flux from child to parent	default = `.false.`
`flg_trcflux_cnsv`	logical	Impose conservation of tracer flux at u(v)star for the child model	default = `.false.`

Continued on next page

Table 22.11 – continued from previous page

variable name	units	description	usage
flg_send_trchdif_p2c	logical	Send tracer lateral diffusion flux from parent to child	default = .false., choose either p2c or c2p or neither
flg_send_trchdif_c2p	logical	Send tracer lateral diffusion flux from child to parent	default = .false., choose either p2c or c2p or neither
flg_send_trcidif_p2c	logical	Send tracer isopycnal diffusion flux from parent to parent	default = .false., choose either p2c or c2p or neither
flg_send_trcidif_c2p	logical	Send tracer isopycnal diffusion flux from child to parent	default = .false., choose either p2c or c2p or neither
nest_depth	integer	Nesting is operated from surface to nest_depth level	

Table22.12 Namelist nml_sub_diff for the on-line child model (tracer_nest.F90)

variable name	description	usage
file_sponge_sub_diff	3D distribution of horizontal diffusivity just inside the boundary	valid if flg_sponge_sub = .true.

Table22.13 Namelist nml_sub_visc for the on-line child model (clinic_nest.F90)

variable name	description	usage
file_sponge_sub_visc	3D distribution of horizontal diffusivity just inside the boundary	valid if flg_sponge_sub = .true.

Table22.14 Namelist nml_sub_isopyc for the on-line child model (ipcoef.F90)

variable name	description	usage
l_isopyc_mask_file	.true./.false 2D distribution of isopycnal diffusion mask is read from file	
file_isopyc_mask_sub	2D mask of isopycnal and GM diffusion	diffusion coefficients are determined as in the normal case

Table22.15 Namelist nml_sub_ssh_diff for the on-line child model (surface_integ.F90)

variable name	description	usage
l_ssh_diff_file	.true./.false 2D-distribution of SSH diffusion is read from file	
file_ssh_diff_sub	2D distribution of horizontal diffusivity of SSH	

Part VII

Miscellaneous

Chapter 23

Basics of the finite difference method

This chapter describes the basics of finite difference methods for solving differential equations. The general principles of the finite differencing methods are introduced using the diffusion equation as an example in Section 23.1. Sections 23.2 and 23.3 describe applying finite difference methods of basic time and space derivatives in differential equations. A more sophisticated time-integration method will be explained later in 23.6. Considerations in finite-difference methods for advection-diffusion equations are discussed in Section 23.4. An implicit method for solving the diffusion equation is described in Section 23.5. In 23.6, we mainly discuss the errors of various time-integration methods for the wave equation. A comprehensive description of these differencing methods in geophysical fluid dynamics is given in detail in Durran (2010).

23.1 Diffusion equation

As an example, consider an initial-boundary value problem expressed by a one-dimensional diffusion equation (heat conductive equation),

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2}. \tag{23.1}$$

Given $T(x, 0) = f(x)$ as the initial distribution and $T(0, t) = T(L, t) = 0$ as the boundary condition, the analytical solution is

$$T(x, t) = \sum_{m=0}^{\infty} f_m e^{-\kappa k_m^2 t} \sin(k_m x), \tag{23.2}$$

where

$$f_m = \frac{2}{L} \int_0^L f(x) \sin(k_m x) dx, \quad k_m = \frac{m\pi}{L}. \tag{23.3}$$

Next, consider the finite difference method to get the solution numerically. In the finite difference method, grids are set with a finite increment in space and time, and each term in the equation is evaluated at each grid using $T_j^n = T(x_j, t_n)$. For example,

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \kappa \frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{\Delta x^2}, \tag{23.4}$$

where $\Delta t = t_{n+1} - t_n$ and $\Delta x = x_{j+1} - x_j$.

Distribution at the new time level T^{n+1} can then be calculated if T^n is known. This finite difference equation is identical to the original differential equation (23.1) in the limit $\Delta t \to 0, \Delta x \to 0$ (**consistency**).

If the initial distribution is assumed to be $f(x) = T_0 \sin k_1 x$, the solution of the finite difference equation (23.4) for $t = t_n$ is

$$T_j^n = \lambda^n T_0 \sin k_1 x_j, \tag{23.5}$$

where

$$\lambda = 1 - \frac{2\kappa \Delta t}{\Delta x^2}(1 - \cos k_1 \Delta x). \tag{23.6}$$

In order to suppress oscillation and divergence of the solution (**stability**), $0 < \lambda < 1$ is necessary and Δx and Δt must be set to satisfy this condition. This solution is identical to the analytical solution in the limit of $\Delta t \to 0, \Delta x \to 0$ (**convergence**).

To summarize, the finite difference method that satisfies consistency, stability, and convergence is the necessary condition for an accurate solution.

23.2 Finite difference expressions for time derivatives

The following are the four basic finite difference expressions. These had been used in the MRI.COM before ver. 4.0.

$$\text{forward}: \frac{T^{n+1}-T^n}{\Delta t} = F(T^n) \tag{23.7}$$

$$\text{backward}: \frac{T^{n+1}-T^n}{\Delta t} = F(T^{n+1}) \tag{23.8}$$

$$\text{Matsuno}: \frac{T^{*n+1}-T^n}{\Delta t} = F(T^n), \quad \frac{T^{n+1}-T^n}{\Delta t} = F(T^{*n+1}) \tag{23.9}$$

$$\text{leap-frog}: \frac{T^{n+1}-T^{n-1}}{2\Delta t} = F(T^n). \tag{23.10}$$

The scheme used in the previous section is the forward scheme. The forward, backward, and Matsuno schemes use the values at two time levels and are accurate to $O(\Delta t)$, while the leap-frog scheme uses three time levels and is accurate to $O(\Delta t)^2$. Basically, the leap-frog scheme is employed in MRI.COM because of its higher order accuracy.

However, the leap-frog scheme cannot be applied to the diffusion equation. A solution by the finite difference method using the leap-frog scheme is given by

$$T_j^n = (T_a\lambda_a^n + T_b\lambda_b^n)\sin k_1 x_j, \tag{23.11}$$

where

$$\lambda_a = -\frac{-\alpha+\sqrt{\alpha^2+4}}{2}, \quad \lambda_b = -\frac{-\alpha-\sqrt{\alpha^2+4}}{2} \quad \left(\alpha \equiv \frac{4\kappa\Delta t}{\Delta x^2}(1-\cos k_1\Delta x)\right). \tag{23.12}$$

Because $\lambda_b < -1$ for arbitrary values of α, the divergent mode with oscillation is always included (**computational mode**). In order to avoid this computational mode, the forward scheme is employed for diffusion and viscosity terms in MRI.COM.

When the diffusion and viscosity coefficients are very large as in the surface mixed layer, the time step has to be unusually small for the stability of the forward scheme according to (23.6). In such a case, the backward scheme is used for vertical diffusion and viscosity (implicit method; see Section 23.5). Though the time integration at each point can proceed without referring to the result of other points by the forward, leap-frog, and Matsuno schemes, it must be done by solving combined linear equations in the backward scheme (see Section 23.5).

The Matsuno scheme is useful for suppressing the computational mode in the leap-frog scheme. By defaults, the Matsuno scheme is used once per twelve steps of the leap-frog scheme in MRI.COM. In the old version, this interval can be changed at run time using a namelist parameter (`nstep_matsuno_interval`) of namelist `nml_time_step` (Table 25.6). It should be noted that the Matsuno scheme needs twice as many numerical operations as the forward and leap-frog schemes.

The simplest forward method is used in components such as the sea ice model, mixed layer model, and ecosystem model. The 3rd-order leap-frog Adams-Moulton scheme (LFAM3) used in the current version of the MRI.COM dynamical frame is discussed in the next section, mainly considering the errors of various time integration methods for the wave equation.

23.3 Finite difference expression for space derivatives

Let us consider a one-dimensional advection equation,

$$\frac{\partial T}{\partial t} = -u\frac{\partial T}{\partial x}, \tag{23.13}$$

where u is a constant velocity. The solution is

$$T(x,t) = T(x-ut,0). \tag{23.14}$$

Using the leap-frog scheme for time differencing, the finite difference equation can be written as follows:

$$\frac{T_j^{n+1}-T_j^{n-1}}{2\Delta t} = -u\frac{T_{j+\frac{1}{2}}^n - T_{j-\frac{1}{2}}^n}{\Delta x}, \tag{23.15}$$

where $T_{j-\frac{1}{2}}^n$ and $T_{j+\frac{1}{2}}^n$ are the values at the left and right faces of the grid cell for x_j, i.e., values at $x_{j-\frac{1}{2}}$ and $x_{j+\frac{1}{2}}$. The point $x_{j-\frac{1}{2}}$ is defined as the central point between x_j and x_{j-1}. Because the transport of T at the boundary that enters a grid cell is identical to that leaving the adjacent grid cell, the total T in the whole system is conserved in this finite difference equation.

There are several methods to decide $T^n_{j-\frac{1}{2}}$ using a value at a single or multiple grid points. The following are two simple and frequently used formulations,

$$\text{upstream finite difference}: T^n_{j-\frac{1}{2}} = T^n_{j-1}\ (u > 0),\quad T^n_{j-\frac{1}{2}} = T^n_j\ (u < 0), \tag{23.16}$$

$$\text{central finite difference}: T^n_{j-\frac{1}{2}} = \frac{T^n_{j-1} + T^n_j}{2}. \tag{23.17}$$

The former is accurate to $O(\Delta x)$, and the latter is accurate to $O(\Delta x^2)$.

In central finite differencing, the expression for (23.15) is

$$\frac{T^{n+1}_j - T^{n-1}_j}{2\Delta t} = -u\frac{T^n_{j+1} - T^n_{j-1}}{2\Delta x}. \tag{23.18}$$

Assuming the solution to be $T(x,t) = \tau(t)e^{-ikx}$,

$$\tau^{n+1} = \tau^{n-1} + 2i\alpha\tau^n, \quad \text{where } \alpha \equiv \frac{u\Delta t}{\Delta x}\sin k\Delta x. \tag{23.19}$$

It is stable (neutral) if $|\alpha| \leq 1$. To be stable for any wave number,

$$\left|\frac{u\Delta t}{\Delta x}\right| \leq 1 \tag{23.20}$$

must be satisfied (**CFL condition**). However, if $\tau^n = \tau^0 e^{-in\Delta\theta}$,

$$\Delta\theta = -\sin^{-1}[\mu\sin k\Delta x](\text{where } \mu \equiv \frac{u\Delta t}{\Delta x}). \tag{23.21}$$

Expanding the r.h.s. by a Taylor expansion we obtain

$$\begin{aligned}\Delta\theta &\simeq -\mu\sin k\Delta x - \frac{1}{6}(\mu\sin k\Delta x)^3 \\ &\simeq -\mu k\Delta x + \frac{\mu(k\Delta x)^3}{6} - \frac{\mu^3(k\Delta x)^3}{6} \\ &= -\mu k\Delta x\left\{1 - \frac{(k\Delta x)^2}{6}(1-\mu^2)\right\}.\end{aligned} \tag{23.22}$$

This means that the phase of the solution from this finite difference scheme is delayed relative to that of analytical solution, depending on its wavenumber (**numerical dispersion**). Therefore, a distribution with maxima and minima that do not exist in the initial distribution arises. However, this method is popularly used since the kinetic energy is conserved by employing the central difference in the advection term in the equation of motion. Moreover, the "Arakawa method," which can nearly conserve the enstrophy (squared vorticity) for horizontally non-divergent flows, is adopted in MRI.COM by using the central difference. This topic is treated in Chapter 8.

Using the upstream finite difference, the finite difference equation (23.15) is

$$\frac{T^{n+1}_j - T^{n-1}_j}{2\Delta t} = -u\frac{T^n_j - T^n_{j-1}}{\Delta x}. \tag{23.23}$$

Expanding the r.h.s. by a Taylor expansion we obtain

$$-u\frac{\partial T}{\partial x} + \frac{u\Delta x}{2}\frac{\partial^2 T}{\partial x^2} + O(\Delta x^2). \tag{23.24}$$

The second term has the diffusion (heat conductive) form (which disappears in the central finite differencing). Actually, the initial distribution diffuses when the advection equation is solved by the upstream finite difference (**numerical diffusion**).

The third order schemes (QUICK, QUICKEST, and UTOPIA) can be used in MRI.COM to suppress the numerical dispersion and diffusion somewhat in the advection calculation for tracers, but not completely. The grid boundary value is set in QUICK as

$$T^n_{j-\frac{1}{2}} = \frac{-T^n_{j-2} + 6T^n_{j-1} + 3T^n_j}{8}(u > 0),\quad T^n_{j-\frac{1}{2}} = \frac{3T^n_{j-1} + 6T^n_j - T^n_{j+1}}{8}(u < 0). \tag{23.25}$$

The QUICKEST method uses the time averaged value at the grid boundary as the tracer value to be transported, considering the change of the value there by advection during one time step. UTOPIA is a multi-dimensional extension of QUICKEST. The details of these schemes are described in Chapter 10.

23.4 Finite differencing of advection-diffusion equation

According to the above restriction, when the advection-diffusion equation (Eqs. (10.1) and (11.1)) is expressed in finite difference form using the leap-frog scheme, it is necessary to use present (previous) time level for advection (diffusion) term. The following finite difference equation is then employed:

$$\frac{T^{n+1}-T^{n-1}}{2\Delta t} = -\mathcal{A}(T^n) + \mathcal{D}(T^{n-1}), \tag{23.26}$$

where $\mathcal{A}$ and $\mathcal{D}$ are advection and diffusion operators, respectively.

When the vertical diffusion term is very large, $\mathcal{D}(T^{n+1})$ is used instead of $\mathcal{D}(T^{n-1})$. This formula is an implicit scheme and is described in the next section.

23.5 Implicit method for vertical diffusion equation

Turbulent mixing is parameterized by using high vertical diffusivity and viscosity determined by boundary layer models, which was treated in Chapter 15. The time step must be set very small to keep the calculation stable when viscosity and diffusivity are very high, since the time tendency becomes very large due to the rapid mixing. To avoid this problem, the implicit method uses the advanced (mixed) state for evaluating viscosity and diffusivity, unlike the normal explicit method where previous or present values are used.

Expressing the present time step as n and the time steps before and after as $n \pm 1$, the finite-difference method is applied to the advection-diffusion equation using the leap-frog scheme. The diffusion term is written separately using $(n-1)$ step for the horizontal direction and $(n+1)$ step for the vertical direction,

$$\frac{(T^{n+1}-T^{n-1})}{2\Delta t} = -\mathcal{A}(T^n) + \mathcal{D}_H(T^{n-1}) + \mathcal{D}_V(T^{n+1}). \tag{23.27}$$

Putting all the terms involving T^{n+1} on the l.h.s.,

$$T^{n+1} - 2\Delta t \mathcal{D}_V(T^{n+1}) = T^{n-1} + 2\Delta t(-\mathcal{A}(T^n) + \mathcal{D}_H(T^{n-1})) \tag{23.28}$$

is obtained, which is an algebraic equation for T^{n+1}. The finite difference form is rewritten specifically as

$$\begin{aligned} T_k^{n+1} - 2\Delta t \frac{1}{\Delta z_k}\left(\kappa_{k-\frac{1}{2}}(T_{k-1}^{n+1} - T_k^{n+1})/\Delta z_{k-\frac{1}{2}} - \kappa_{k+\frac{1}{2}}(T_k^{n+1} - T_{k+1}^{n+1})/\Delta z_{k+\frac{1}{2}}\right) \\ = T_k^{n-1} + 2\Delta t(-\mathcal{A}(T_k^n) + \mathcal{D}_H(T_k^{n-1})). \end{aligned} \tag{23.29}$$

By putting

$$a = \frac{2\Delta t \kappa_{k-\frac{1}{2}}}{\Delta z_k \Delta z_{k-\frac{1}{2}}}, \quad b = 1 + a + c, \quad c = \frac{2\Delta t \kappa_{k+\frac{1}{2}}}{\Delta z_k \Delta z_{k+\frac{1}{2}}}, \tag{23.30}$$

we get,

$$-aT_{k-1}^{n+1} + bT_k^{n+1} - cT_{k+1}^{n+1} = T_k^{n-1} + 2\Delta t(-\mathcal{A}(T_k^n) + \mathcal{D}_H(T_k^{n-1})). \tag{23.31}$$

Setting $\mathcal{F}_k \equiv -\mathcal{A}(T_k^n) + \mathcal{D}_H(T_k^{n-1})$, this is expressed in the matrix form as

$$\begin{pmatrix} b & -c & & & & & \\ -a & b & -c & & & & \\ & -a & b & -c & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -a & b & -c & \\ & & & & -a & b & -c \\ & & & & & -a & b \end{pmatrix} \begin{pmatrix} T_1^{n+1} \\ T_2^{n+1} \\ T_3^{n+1} \\ \vdots \\ T_{KM-2}^{n+1} \\ T_{KM-1}^{n+1} \\ T_{KM}^{n+1} \end{pmatrix} = \begin{pmatrix} T_1^{n-1} + 2\Delta t \mathcal{F}_1 \\ T_2^{n-1} + 2\Delta t \mathcal{F}_2 \\ T_3^{n-1} + 2\Delta t \mathcal{F}_3 \\ \vdots \\ T_{KM-2}^{n-1} + 2\Delta t \mathcal{F}_{KM-2} \\ T_{KM-1}^{n-1} + 2\Delta t \mathcal{F}_{KM-1} \\ T_{KM}^{n-1} + 2\Delta t \mathcal{F}_{KM} \end{pmatrix}. \tag{23.32}$$

The l.h.s. has the form of the tri-diagonal matrix.

23.5.1 A solution of tri-diagonal matrix

In general, simultaneous linear equations for n variables with tri-diagonal matrix coefficients

$$\begin{pmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & B_{n-1} & C_{n-1} \\ & & & A_n & B_n \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n-1} \\ X_n \end{pmatrix} = \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_{n-1} \\ D_n \end{pmatrix} \tag{23.33}$$

are solved using the Thomas method, which is modified from LU decomposition,

$$P_1 = C_1/B_1 \tag{23.34}$$

$$Q_1 = D_1/B_1 \tag{23.35}$$

$$P_k = \frac{C_k}{B_k - A_k P_{k-1}} \qquad (2 \le k \le n-1) \tag{23.36}$$

$$Q_k = \frac{D_k - A_k Q_{k-1}}{B_k - A_k P_{k-1}} \qquad (2 \le k \le n) \tag{23.37}$$

$$X_n = Q_n \tag{23.38}$$

$$X_k = Q_k - P_k X_{k+1} \qquad (1 \le k \le n-1). \tag{23.39}$$

23.6 Evaluation of time-integration methods for wave equations

In the ocean models, the CFL condition is mainly determined by the speed of baroclinic gravity waves. In evaluating time integration in such wave propagation, it is not sufficient to evaluate only one variable as we have done in the previous sections, but it is necessary to consider the interaction of two variables. Here, we evaluate time integration methods for the simplest one-dimensional wave equation.

Consider the following one-dimensional wave equation,

$$\frac{\partial p}{\partial t} = -c\frac{\partial u}{\partial x}, \tag{23.40}$$

$$\frac{\partial u}{\partial t} = -c\frac{\partial p}{\partial x}. \tag{23.41}$$

We will discuss the accuracy of the time integration method based on Shchepetkin and McWilliams (2005) (hereafter, SM2005). For the parts not explained in SM2005, see Durran (2010), Section 2.2 for more details.

Applying the Fourier transformation with respect to space by placing $p(t,x) = \hat{p}_k e^{ikx}, u(t,x) = \hat{u}_k u^{ikx}$ and $\omega_k = kc$ yields,

$$\frac{\partial \hat{p}_k}{\partial t} = -i\omega_k \hat{u}_k, \tag{23.42}$$

$$\frac{\partial \hat{u}_k}{\partial t} = -i\omega_k \hat{p}_k. \tag{23.43}$$

For simplicity, we omit ˆ and the subscript k, and denote the variable ζ^n at the time $t = n\Delta t$ of the variable ζ. To further simplify the equation, we introduce the dimensionless quantity $\alpha = \omega_k \Delta t = ck\Delta t$ (Durran (2010) uses $\omega\Delta t$ throughout his text though it is a bit redundant). Since the maximum k (k_{max}) that can be resolved in the model is $1/\Delta x$, the maximum α (α_{max}) is $c\Delta t/\Delta x$, i.e. Courant number.

In studying the stability and accuracy of the time integration scheme for the wave equation, we assume that a variable ζ is multiplied by the complex number λ between n and $n+1$ steps. (Durran (2010) uses A instead of λ).

$$\zeta^{n+1} = \lambda\zeta^n. \tag{23.44}$$

During the time step, the amplification rate is $|\lambda| = \left(\Re(\lambda)^2 + \Im(\lambda)^2\right)^{1/2}$, and phase change is $\theta = \arctan(\Im(\lambda)/\Re(\lambda))$. ($\Re(), \Im()$ are functions to extract real and complex numbers, respectively). The relative phase error is estimated as $R = \theta/\alpha$ using the positive phase α. If $R > 1$, the scheme is accelerating; if $R < 1$, the scheme is decelerating. If the wave is completely represented, then $\lambda = e^{i\alpha}$. In SM2005, to visually represent the error, $\lambda(\alpha)$ is drawn on the complex plane by varying α and compared with the complete solution $e^{i\alpha}$ (i.e. the unit circle on the complex plane).

23.6.1 LFAM3 method

Here, we consider solving the wave equation with the third-order leap-frog Adams-Moulton method (LFAM3) as the time-integration method used in the current version of MRI.COM. In the LFAM3 method, we first obtain a tentative value of $p^{n+1,*}, u^{n+1,*}$ in the *predictor step*,

$$p^{n+1,*} = p^{n-1} - 2i\alpha u^n, \tag{23.45}$$

$$u^{n+1,*} = u^{n-1} - 2i\alpha p^n. \tag{23.46}$$

In the *corrector step*, we conduct the final time integration,

$$p^{n+1} = p^n - i\alpha\left\{\frac{5}{12}u^{n+1,*} + \frac{2}{3}u^n - \frac{1}{12}u^{n-1}\right\}, \tag{23.47}$$

$$u^{n+1} = u^n - i\alpha\left\{\frac{5}{12}p^{n+1,*} + \frac{2}{3}p^n - \frac{1}{12}p^{n-1}\right\}. \tag{23.48}$$

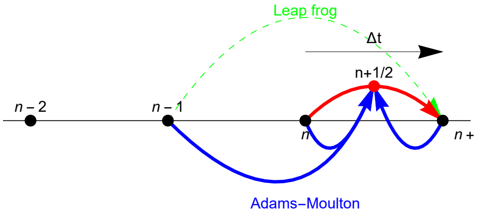

Figure23.1 Schematics of the LFAM3 time integration. The 3rd-order Adams-Moulton method uses values at $n-1$, n, and, $n+1$ steps to find the value at $n+\frac{1}{2}$ step. The LFAM3 method used the leap-frog method to create the value at $n+1$ step. This is why it is named LFAM3. For reference the third-order Adams-Bashforth methods uses values at $n-2$, $n-1$, and n steps to find the value at $n+\frac{1}{2}$.

The LFAM3 used for OGCM is as follows, which corresponds to SM2005 equations (2.38)-(2.41) (or SM2009 equations (4.1)-(4.2)). This is composed of the predictor step,

$$p^{n+1/2} = \left(\frac{1}{2} - 2\gamma\right)p^{n-1} + \left(\frac{1}{2} + 2\gamma\right)p^n - i\alpha(1-2\gamma)u^n \tag{23.49}$$

$$u^{n+1/2} = \left(\frac{1}{2} - 2\gamma\right)u^{n-1} + \left(\frac{1}{2} + 2\gamma\right)u^n - i\alpha(1-2\gamma)\left[p^n + \beta\frac{2p^{n+1/2} - 3p^n + p^{n-1}}{1-2\gamma}\right] \tag{23.50}$$

and corrector step,

$$p^{n+1} = p^n - i\alpha u^{n+1/2} \tag{23.51}$$

$$u^{n+1} = u^n - i\alpha\left\{(1-\epsilon)p^{n+1/2} + \epsilon\left[\left(\frac{1}{2} - \gamma\right)p^{n+1} + \left(\frac{1}{2} + 2\gamma\right)p^n - \gamma p^{n-1}\right]\right\}. \tag{23.52}$$

In the case of $\gamma = 1/12, \beta = 0, \epsilon = 0$, (23.45–23.48) and (23.49–23.52) are mathematically equivalent. The procedure for (23.49–23.52) is described in detail in Section 3.

Now let us return to (23.45–23.48) to evaluate their errors. (23.45–23.48) is combined into two equations.

$$p^{n+1} = \left(1 - \frac{5}{6}\alpha^2\right)p^n - i\alpha\left\{\frac{2}{3}u^n + \frac{1}{3}u^{n-1}\right\}, \tag{23.53}$$

$$u^{n+1} = \left(1 - \frac{5}{6}\alpha^2\right)u^n - i\alpha\left\{\frac{2}{3}p^n + \frac{1}{3}p^{n-1}\right\}. \tag{23.54}$$

The above equations can be written in matrix form.

$$\begin{pmatrix} p^{n+1} \\ u^{n+1} \end{pmatrix} = \begin{pmatrix} 1-\frac{5}{6}\alpha^2 & -i\frac{2}{3}\alpha \\ -i\frac{2}{3}\alpha & 1-\frac{5}{6}\alpha^2 \end{pmatrix}\begin{pmatrix} p^n \\ u^n \end{pmatrix} + \begin{pmatrix} 0 & -i\frac{1}{3}\alpha \\ -i\frac{1}{3}\alpha & 0 \end{pmatrix}\begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} \tag{23.55}$$

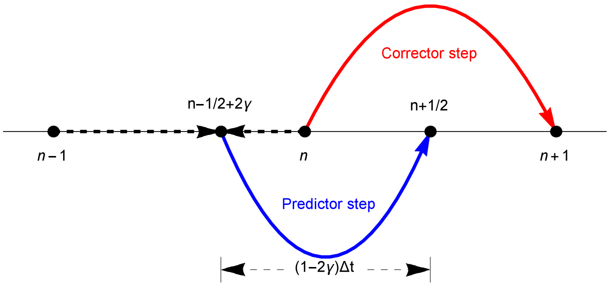

Figure23.2 Schematics of LFAM3 method used in the model. Mathematically this is equivalent to Fig. 23.1. The starting point of the predictor step is $n - 1/2 + 2\gamma$ step, which is a linear combination of $n - 1$ and n steps. The values at $n + \frac{1}{2}$ step is obtained by integrating over the interval $(1 - 2\gamma)\Delta t$. In the corrector step, the values at $n + 1$ step is obtained from n by integrating over the domain n-$n + 1$. The values at $n + \frac{1}{2}$ step are discarded after the corrector step.

The characteristic equation of this system is obtained by substituting $p^{n+1} = \lambda p^n = \lambda^2 p^{n-1}$, $u^{n+1} = \lambda u^n = \lambda^2 u^{n-1}$.

$$\lambda^2 \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} = \lambda \begin{pmatrix} 1 - \frac{5}{6}\alpha^2 & -i\frac{2}{3}\alpha \\ -i\frac{2}{3}\alpha & 1 - \frac{5}{6}\alpha^2 \end{pmatrix} \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} + \begin{pmatrix} 0 & -i\frac{1}{3}\alpha \\ -i\frac{1}{3}\alpha & 0 \end{pmatrix} \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} \tag{23.56}$$

Rearranging it yields the following equation.

$$\begin{pmatrix} \lambda^2 - \lambda(1 - \frac{5}{6}\alpha^2) & i\alpha(\frac{2}{3}\lambda + \frac{1}{3}) \\ i\alpha(\frac{2}{3}\lambda + \frac{1}{3}) & \lambda^2 - \lambda(1 + \frac{5}{6}\alpha^2) \end{pmatrix} \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{23.57}$$

From the above determinant = 0, we obtain the following solution

$$\lambda^2 - \left(1 - \frac{5}{6}\alpha^2\right)\lambda = \pm i\alpha \left(\frac{2}{3}\lambda + \frac{1}{3}\right) \tag{23.58}$$

The plus and minus in the equation correspond to the propagation of the waves in both directions, and placing $\alpha = -\alpha$ does not change the equation. Also, since the form is the same for p and u, the properties of the wave equation in LFAM are the same for that in LFAM3 for one variable.

Adopting the plus of equation (23.58), we obtain the following equation,

$$\lambda^{\pm} = \frac{1}{12}\left(g_1 \pm \sqrt{(-g_1)^2 + 48i\alpha}\right) \tag{23.59}$$

$$g_1 = 6 + 4i\alpha - 5\alpha^2 \tag{23.60}$$

Physical mode and Computational mode are λ^+ and λ^- for the range $0 < \alpha < 1.55$, respectively. In the range $1.55 < \alpha < 1.59$ they are λ^- and λ^+ respectively.

Figure 23.3 shows the $\lambda(\alpha)$ of LFAM3 drawn in the complex plane by varying α and compared to the exact solution $e^{i\alpha}$ (i.e., the unit circle in the complex plane). Stability is determined by the amplitude of the physical mode exceeding 1, $\alpha = 1.587$ at this time. Though computational mode decreases faster than physical mode, there is a drawback that the amplitudes of the computational mode and the physical mode approach each other when α exceeds 1.

23.6.2 Generalized Leap-Frog Adams-Mouton method (G-LFAM3)

SM2005 has shown that the accuracy and stability of the time integration can be increased for the wave equation by making minor modifications over the original LFAM3. The characteristic equation of Generalized LFAM3 derived by SM2005 is as follows (SM2005 Equation 2.31),

$$\begin{pmatrix} p^{n+1} \\ u^{n+1} \end{pmatrix} = \begin{pmatrix} A & -iB \\ -iC & D \end{pmatrix} \begin{pmatrix} p^n \\ u^n \end{pmatrix} + \begin{pmatrix} E & -iF \\ -iG & H \end{pmatrix} \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} \tag{23.61}$$

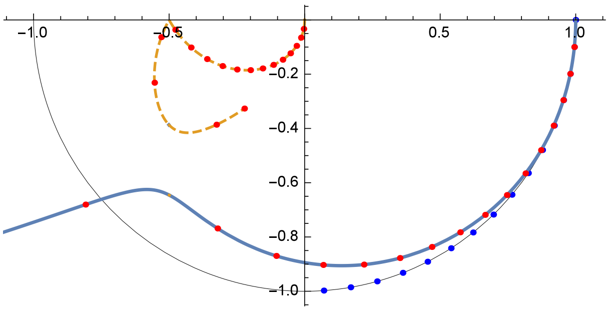

Figure23.3 Characteristic solution of LFAM3 ($\beta = \epsilon = 0, \gamma = 1/12$) on the complex plane. The solid line from (1,0) is the physical mode, and the dashed line from (0,0) is the computational mode. Dots are marked every time α changes by 0.1. The point where the amplitude in physical mode exceeds 1 is at $\alpha = 1.587$.

Here, *A*–*H* are

$$A = 1 - 2\alpha^2 \left(\frac{1}{2} - \gamma\right)(1 - 2\beta), \qquad B = \alpha \left\{\frac{1}{2} + 2\gamma - 4\alpha^2 \left(\frac{1}{2} - \gamma\right)\beta\right\}, \tag{23.62}$$

$$C = \alpha \left\{\frac{1}{2} + 2\gamma + \epsilon \left(\frac{1}{2} - \gamma\right)\left[1 - 2\alpha^2 \left(\frac{1}{2} - \gamma\right)(1 - 2\beta)\right]\right\}, \tag{23.63}$$

$$D = 1 - 2\alpha^2 \left(\frac{1}{2} - \gamma\right)\left\{1 - \epsilon \left[\frac{3}{4} - \gamma + 2\alpha^2 \left(\frac{1}{2} - \gamma\right)\beta\right]\right\}, \tag{23.64}$$

$$E = -4\alpha^2 \left(\frac{1}{2} - \gamma\right)\beta, \qquad F = \alpha \left(\frac{1}{2} - 2\gamma\right), \tag{23.65}$$

$$H = -\alpha^2 \left(\frac{1}{2} - \gamma\right)\left(\frac{1}{2} - 2\gamma\right)\epsilon, \quad G = \alpha \left\{\frac{1}{2} - 2\gamma - \epsilon \left(\frac{1}{2} - \gamma\right)\left[1 + 4\alpha^2 \left(\frac{1}{2} - \gamma\right)\beta\right]\right\} \tag{23.66}$$

SM2005 discusses how to move γ, ϵ, β to reduce error and increase stability. According to this, the case $\gamma = 1/12, \epsilon = 11/20, \beta = 17/120$ has the smallest error: its accuracy for the wave equation is $O(\alpha^4)$. In this case the solution is very close to the ideal solution as shown in Fig. 23.4. Stability is determined by the amplitude of computational mode exceeding 1, where $\alpha = 1.851640$. This is the default setting of MRI.COM. The most stable combination is $\gamma = 1/12, \beta = 0.126, \epsilon = 0.83$ (Fig. 23.5). In this case the place where the amplitude of computational mode exceeds 1 is $\alpha = 1.958537$, but its accuracy for the wave equation is $O(\alpha^3)$.

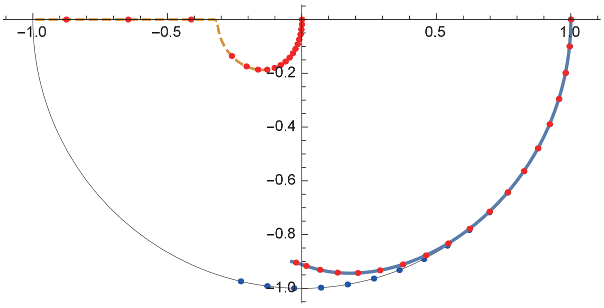

Figure23.4 Characteristic solutions of G-LFAM3 in the complex plane for the most accurate case $\gamma = 1/12, \beta = 17/120, \epsilon = 11/20$. Its accuracy for the wave equation is $O(\alpha^4)$. The solid line from (1,0) is the physical mode, and the dashed line from (0,0) is the computational mode. The point is marked every time α changes by 0.1, and the place where the amplitude of the computational mode exceeds 1 is $\alpha = 1.851640$.

23.6.3 Evaluation of errors in leap-frog method for wave equation

Here, we estimate the error in solving the wave equation with leap frog method previously used in MRI.COM.

$$p^{n+1} = -2i\alpha u^n + p^{n-1} \tag{23.67}$$

$$u^{n+1} = -2i\alpha p^n + u^{n-1} \tag{23.68}$$

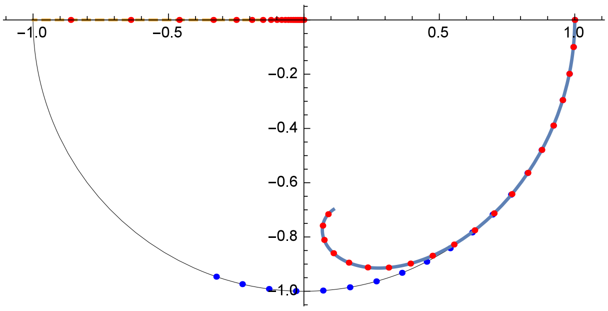

Figure23.5 Characteristic solutions of G-LFAM3 in the complex plane for the most stable case $\gamma = 1/12, \beta = 0.126, \epsilon = 0.83$. Its accuracy for the wave equation is $O(\alpha^3)$. The solid line following from (1,0) is the physical mode, and the dashed line following from (0,0) is the computational mode. Dots are marked every time α changes by 0.1. The point where the amplitude of the computational mode exceeds 1 is $\alpha = 1.958537$.

The following equation is obtained when written using matrices.

$$\begin{pmatrix} p^{n+1} \\ u^{n+1} \end{pmatrix} = \begin{pmatrix} 0 & -2i\alpha \\ -2i\alpha & 0 \end{pmatrix} \begin{pmatrix} p^{n} \\ u^{n} \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} \tag{23.69}$$

The characteristic equations of this system are obtained by substituting $p^{n+1} = \lambda p^n = \lambda^2 p^{n-1}$, $u^{n+1} = \lambda u^n = \lambda^2 u^{n-1}$.

$$\lambda^2 \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} = \lambda \begin{pmatrix} 0 & -2i\alpha \\ -2i\alpha & 0 \end{pmatrix} \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} \tag{23.70}$$

This is rewritten as

$$\begin{pmatrix} \lambda^2 - 1 & 2i\alpha\lambda \\ 2i\alpha\lambda & \lambda^2 - 1 \end{pmatrix} \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{23.71}$$

From the determinant = 0, we obtain the following equation,

$$(\lambda^2 - 1)^2 + 4\alpha^2\lambda^2 = 0. \tag{23.72}$$

$$(\lambda^2 - 1) = \pm 2i\alpha\lambda \tag{23.73}$$

From the negative sign (the result is the same for the positive sign), we obtain the characteristic equation,

$$\lambda^2 + 2i\alpha\lambda - 1 = 0. \tag{23.74}$$

The characteristic equation above is exactly the same as the characteristic equation derived from the following one-variable leap-frog equation.

$$p^{n+1} = p^{n-1} - 2i\alpha p^n \tag{23.75}$$

$$\lambda^2 + 2i\alpha\lambda - 1 = 0 \tag{23.76}$$

The quadratic formula yields

$$\lambda = -i\alpha \pm \sqrt{1 - \alpha^2} \tag{23.77}$$

However, the above expression is correct only for $0 < \alpha < 1$. Of these, $-i\alpha + \sqrt{1-\alpha^2}$ is called the physical mode and $-i\alpha - \sqrt{1-\alpha^2}$ is called the computational mode.

The amplification factor $|\lambda|$ is always 1 in the range $0 < \alpha < 1$. The relative phase shift is written as *

$$R = \frac{1}{\alpha} \arctan\left(\frac{\alpha}{\sqrt{1-\alpha^2}}\right) \simeq 1 + \frac{\alpha^2}{6} + \frac{3\alpha^4}{40} + O(\alpha^5) \tag{23.78}$$

From Fig. 23.6, we can see that the phase is accelerated in the leap-frog time integration scheme. Since the amplitude of not only the physical mode but also the computational mode is 1, it is necessary to introduce some mechanism to attenuate

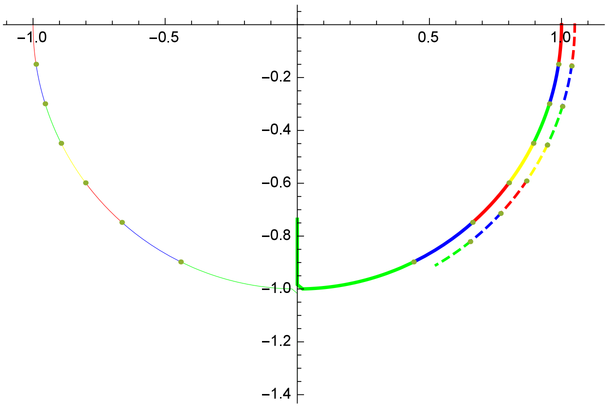

Figure23.6 Characteristic solutions in the complex plane of LF. The line starting from (1,0) is the physical mode, and the line starting from (-1,0) is the computational mode. The outer dotted line is the exact solution ($e^{i\alpha}$). The dotted line outside is the exact solution ($e^{i\alpha}$). Originally, the two lines overlap, but the exact solution is shifted outward to make the difference in phase easier to understand. The color is changed every time α changes $\pi/18$.

the computational mode. In the old version of MRI.COM, this was handled by inserting the Matsuno scheme about once every 10 times.

23.6.4 Leap frog with Shuman's averaging (LF-SA)

The method discussed in this subsection is not employed by MRI.COM, but is used by NEMO and others and is explained for comparison. This method adopts Shuman's averaging (LF-SA), which uses a weighted average of $n+1$, $n-1$, and n steps to calculate the pressure gradient. In this case, it is known that the time step can be set to be twice as long as that of a normal leap-frog method for wave equation. Here, we confirm this by actually solving the wave equations with the Shuman's averaging method (LF-SA), which are written as

$$p^{n+1} = p^{n-1} - 2i\alpha u^n, \tag{23.79}$$

$$u^{n+1} = u^{n-1} - 2i\alpha \left(\frac{p^{n-1} + 2p^n + p^{n+1}}{4} \right). \tag{23.80}$$

The characteristic equation in the matrix form is

$$\lambda^2 \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} = \begin{pmatrix} 1 & -2i\alpha\lambda \\ -2i\alpha \left(\frac{1+2\lambda+\lambda^2}{4} \right) & 1 \end{pmatrix} \begin{pmatrix} p^{n-1} \\ u^{n-1} \end{pmatrix} \tag{23.81}$$

From the above determinant = 0, the eigenvalues are solutions of the following characteristic equations.

$$(\lambda^2 - 1)^2 + \alpha^2 \lambda (\lambda + 1)^2 = 0 \tag{23.82}$$

This equation has multiple solutions as $\lambda = -1$, which is the computational mode. The physical mode is

$$\lambda = \frac{2 - \alpha^2 \pm i\alpha\sqrt{4 - \alpha^2}}{2}. \tag{23.83}$$

The amplification factor $|\lambda|$ is always 1 in the range $0 < \alpha < 2$, and its relative phase error is

$$R = \frac{1}{\alpha} \arctan \left(\frac{\alpha\sqrt{4-\alpha^2}}{2-\alpha^2} \right) \simeq 1 + \frac{\alpha^2}{24} + \frac{3\alpha^4}{640} + O(\alpha^5). \tag{23.84}$$

This is practically the same as replacing $\alpha' = \alpha/2$ in the LF method.

[*] The derivation of the last Taylor expansion is complicated. Here, it was derived using mathematica. The relative phase accuracy is $O(\alpha)$ and the absolute phase accuracy is $O(\alpha^2)$.

23.6.5 Time-staggered method

The method discussed in this subsection is not employed by MRI.COM, but is used by MITgcm, MOM, and others and is explained for comparison. The properties of this solution are the same as the physical mode of LF-SA according to Lemariè et al. (2015). Here, we confirm this by actually solving the wave equations with the time-staggered method, which are written as

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} - i\alpha u^n, \tag{23.85}$$

$$u^{n+1} = u^{n-1} - i\alpha p^{n+\frac{1}{2}}. \tag{23.86}$$

In matrix form, it is rewritten as

$$\lambda \begin{pmatrix} p^{n-1/2} \\ u^n \end{pmatrix} = \begin{pmatrix} 1 & -i\alpha \\ -i\alpha\lambda & 1 \end{pmatrix} \begin{pmatrix} p^{n-\frac{1}{2}} \\ u^n \end{pmatrix} \tag{23.87}$$

Its characteristic equation is

$$(\lambda - 1)^2 + \alpha^2 \lambda = 0. \tag{23.88}$$

This solution has no computational mode. Its physical mode is

$$\lambda = \frac{2 - \alpha^2 \pm i\alpha\sqrt{4 - a^2}}{2}. \tag{23.89}$$

This solution is indeed identical to the physical mode of LF-SA shown in the previous subsection.

23.6.6 The 3rd-order Adams-Bashforth method (AB3)

The third-order Adams-Bashforth method (AB3), which may be used in the time extrapolation of the external force for the barotropic component of the momentum equation, is also evaluated here with a single variable.

$$q^{n+\frac{1}{2}} = \frac{23}{12}q^n - \frac{4}{3}q^{n-1} + \frac{5}{12}q^{n-2}, \tag{23.90}$$

$$q^{n+1} = q^n - i\alpha q^{n+\frac{1}{2}}. \tag{23.91}$$

The characteristic equation is

$$\lambda^3 = \left(1 - i\alpha\frac{23}{12}\right)\lambda^2 + i\alpha\frac{4}{3}\lambda - i\alpha\frac{5}{12}. \tag{23.92}$$

Analytical solutions exist, but they are very long and have been omitted. As shown in Fig. 23.7, there are physical and computational modes. The instability occurs when the amplitude of the computational mode exceeds 1 at $\alpha = 0.72$. It has $O(\alpha^3)$ accuracy.

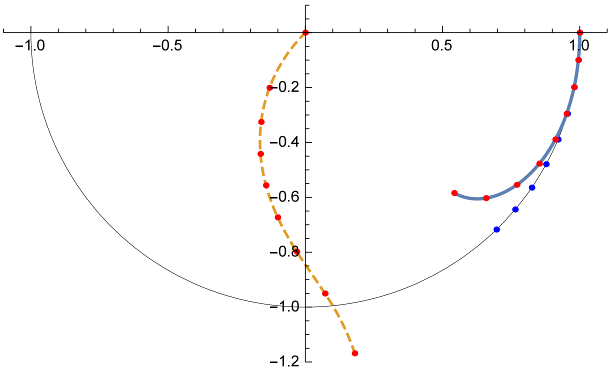

Figure23.7 The characteristic solution of the AB3 method in the complex plane. The solid line starting from (1,0) is the physical mode, and the dashed line starting from (0,0) is the computational mode. The points are marked every time α changes by 0.1. The place where the amplitude of the computational mode exceeds 1 is $\alpha = 0.72$.

23.6.7 Quasi-second-order Adams-Bashforth method (AB2ϵ)

The quasi-second-order Adams-Bashforth method (AB2ϵ), which may be used in the time extrapolation of external forces for the forward pressure equation, is also evaluated for a single variable. Quasi-second order means that the original AB2 is second-order accurate but unstable, so by introducing ϵ, stability is obtained instead of giving up perfect $O(\Delta t^2)$ accuracy.

$$q^{n+\frac{1}{2}} = \left(\frac{3}{2} + \epsilon\right) q^n - \left(\frac{1}{2} + \epsilon\right) q^{n-1} \tag{23.93}$$

$$q^{n+1} = q^n - i\alpha q^{n+\frac{1}{2}} \tag{23.94}$$

The characteristic equation and its solution are

$$\lambda^2 = \left\{1 - i\alpha\left(\frac{3}{2} + \epsilon\right)\right\}\lambda + i\alpha\left(\frac{1}{2} + \epsilon\right), \tag{23.95}$$

$$\lambda^{\pm} = \frac{1}{4}\left(g \pm \sqrt{g^2 + 8i\alpha(1 + 2\epsilon))}\right) \tag{23.96}$$

$$g = -i\alpha(3 + 2\epsilon) + 2 \tag{23.97}$$

The λ^+, λ^- are the physical mode and computational mode, respectively. As shown in Fig. 23.8, the physical mode is distributed almost on the unit circle and slightly outside the unit circle around $\alpha = 0.50$. The amplitude of the computational mode is kept small enough. When ϵ=0, it is outside the unit circle at the starting point of (1,0).

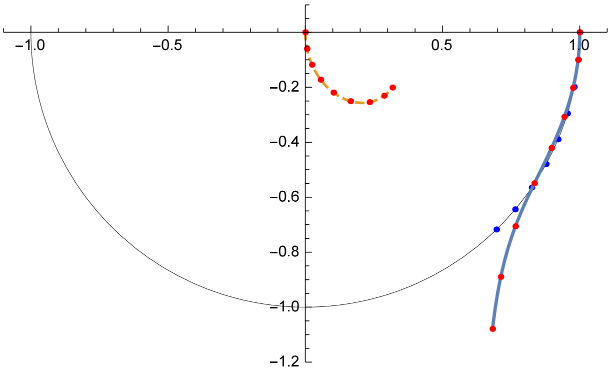

Figure23.8 The characteristic solution of the AB2ϵ method in the complex plane. The solid line starting from (1,0) is the physical mode, and the dashed line starting from (0,0) is the computational mode. The points are marked every time α changes by 0.1. The place where the amplitude of the computational mode exceeds 1 is α = 0.50254.

23.6.8 Matsuno scheme

Here, we evaluate the Matsuno scheme, which was previously used with the LF method on MRI.COM.

$$q^{n+1*} = q^n - i\alpha q^n \tag{23.98}$$

$$q^{n+1} = q^n - i\alpha q^{n+1*} = \left(1 - i\alpha - \alpha^2\right) q^n \tag{23.99}$$

Its characteristic solution is

$$\lambda = 1 - i\alpha - \alpha^2 \tag{23.100}$$

Though it is stable for $0 < \alpha < 1$, it has large amplitude and phase errors (Fig. 23.9).

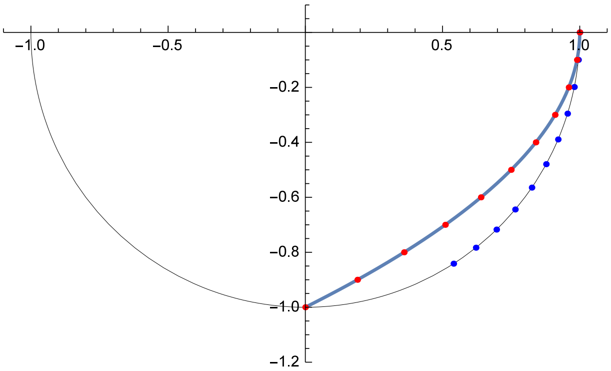

Figure23.9 The characteristic solution of the Matsuno method in the complex plane. The points are marked every time α changes by 0.1. The place where its amplitude exceeds 1 is $\alpha = 1$.

23.7 Evaluation of the time-integration method for advection-diffusion equation

Until now, we have concentrated on stability analysis for waves (internal gravity waves) according to SM2005, so we have evaluated the time-integration method by considering only ω_i in the following equation,

$$\frac{\partial \hat{q}_k}{\partial t} = -\omega \hat{q}_k, \tag{23.101}$$

$$\omega = \omega_r + i\omega_i, \tag{23.102}$$

and moving $\alpha \equiv \omega_i \Delta t$ in the range of real numbers. Then the method has been evaluated by comparing the behavior of the eigenvalues (λ) of the characteristic equation with the analytical solution ($e^{i\alpha}$).

This is equivalent to placing $M = 0$ in the following one-dimensional advection-diffusion equation.

$$\frac{\partial q}{\partial t} + c\frac{\partial q}{\partial x} = M\frac{\partial^2 q}{\partial x^2} \tag{23.103}$$

On the other hand, Lemariè et al. (2015) considers that ω_r can be used to evaluate the difference between upstream differential (numerical diffusion) and central differential advection in stability analysis. To do so, he investigated the behavior of the eigenvalues (λ) of the characteristic equation by moving ω on the complex plane. Following this method, in this section we examine stability analysis for various time-difference schemes, i.e., the range of amplitudes $|\lambda| \leq 1$ of the characteristic equations. This is equivalent to the analysis called Absolute Stability (A-Stability) in Durran (2010). Here we perform A-Stability for the LFAM3 method in the one-variable case and show that the imaginary part of this solution is equivalent to the analysis of SM2005. The time integral of LFAM3 can be expressed for the Fourier series $\hat{q}$ using $\mu \equiv \omega \Delta t$ as follows

$$\hat{q}^{n+1,*} = \hat{q}^{n-1} + 2\mu\hat{q}^{n}, \qquad (LF) \tag{23.104}$$

$$\hat{q}^{n+1/2} = \frac{5}{12}\hat{q}^{n+1,*} + \frac{2}{3}\hat{q}^{n} - \frac{1}{12}\hat{q}^{n-1} \qquad \text{(AM3)} \tag{23.105}$$

$$\hat{q}^{n+1} = \hat{q}^{n} + \mu\hat{q}^{n+1/2} \qquad \text{(corrector)}, \tag{23.106}$$

$$= \left(1 + \frac{2\mu}{3} + \frac{5\mu^2}{6}\right)\hat{q}^{n} + \frac{\mu}{3}\hat{q}^{n-1}. \tag{23.107}$$

The characteristic equation for (23.107) is

$$\lambda^2 = \left(1 + \frac{2\mu}{3} + \frac{5\mu^2}{6}\right)\lambda + \frac{\mu}{3}. \tag{23.108}$$

Its solution is

$$\lambda^{\pm} = \frac{1}{12}\left(g_1 \pm \sqrt{(-g_1)^2 + 48\mu}\right) \tag{23.109}$$

$$g_1 = 6 + 4\mu + 5\mu^2 \tag{23.110}$$

The region of A-Stability is obtained by finding the region of $|\lambda| \leq 1$ on the complex plane of μ, which corresponds to the interior of the contour in Fig. 23.10. On the other hand, if we set $\mu_r = 0$ in λ and vary λ on the complex plane with $\alpha \equiv \mu_i$ as a parameter, we obtain Fig. 23.3. In fact, substituting $\mu = -i\alpha$ in equation (23.108), we obtain equation (23.60). The place where the contour in Fig. 23.10 intersects the imaginary axis is $\mu_i = 1.59$, which is equal to the place where the physical mode intersects the unit circle in Fig. 23.3. Incidentally, the region of stability of LF is on the imaginary axis in the range from $(0, -i)$ to $(0, i)$, and is absolutely unstable with respect to diffusion, etc.

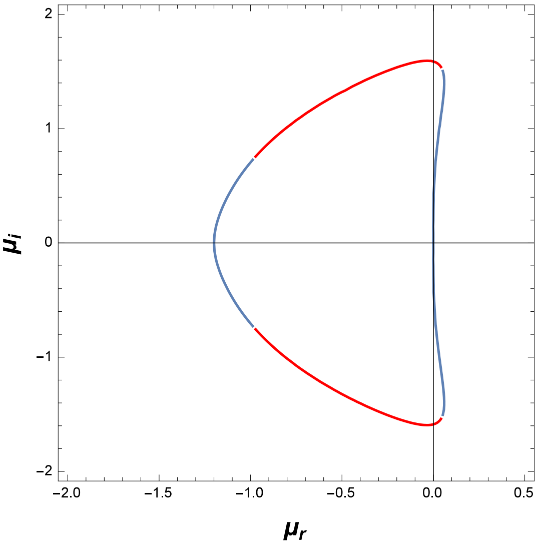

Figure23.10 Contour of $|\lambda| = 1$ on the complex plane of μ. This is equivalent to the curve of LFAM3 in Fig.1(a) in Lemariè et al. (2015). The blue line corresponds to λ^+ and the red lines correspond to λ^-. The region enclosed by the curve is the stable region.

When we draw the region of stability for AB2ϵ and AB3, we get Fig. 23.11. The intersection of the imaginary axes is 0.502 for AB2ϵ versus 0.72 for AB3, which means that AB3 is more stable in the pure wave equation. Since AB2ϵ is more extended toward the real axis, the stability for diffusion is more stable for AB2ϵ. For other schemes, see Fig. 1 of Lemariè et al. (2015).

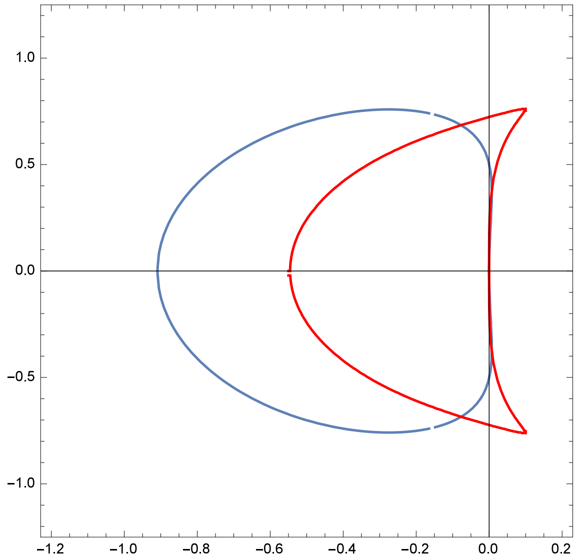

Figure23.11 Contours of $|\lambda| = 1$ on the complex plane of μ, which are equivalent to the curve of AB2ϵ and AB3 in Fig.1(a) in Lemariè et al. (2015). The blue line corresponds to the AB2ϵ method and the red line corresponds to the AB3λ^- method. The regions enclosed by the curves are the stable regions.

Chapter 24

Generalized orthogonal curvilinear coordinate grids

This chapter introduces generalized orthogonal curvilinear coordinates and presents related calculus.

24.1 Outline

A finite volume ocean model on geographic coordinates does not have any problem concerning the South Pole because it does not calculate around the South Pole. However, serious problems arise around the North Pole where the meridian concentrates to one point in the ocean. First, it is necessary to calculate the temporal evolution of the physical quantity in a special way only there, because the relations between U-cells that surround the North Pole and the northernmost T-cell are topologically peculiar. Next, even if a cell doesn't touch the North Pole, its zonal lattice interval is extremely small near the North Pole. Therefore, a short time step for integration is required owing to the limitation of the CFL condition. This limitation is reflected directly in the increased calculation time required. Moreover, when the zonal grid intervals in low latitudes and the Arctic region are extremely different, the arguments about accuracies of numerical schemes and the parameters for diffusion and viscosity operators generally cannot be applied uniformly to a model domain.

The following can be considered to avoid such problems concerning the North Pole. 1) Creating a huge island including the North Pole. The finite-difference calculation in the island is abandoned, and the lateral boundary values are restored to the climatology. 2) Shifting the singular points of the model to a continent or a huge island by changing the model's horizontal grid system. The MRI.COM scheme adopts the latter approach, which is outlined in this section.

Because the MRI.COM code is based on generalized orthogonal coordinates, the geographical latitude (ϕ) and longitude (λ) are not of a great concern for the calculus in the model. However, it is necessary to know the land and sea distribution, sea depth, scale factor, and the Coriolis parameter given as a function of λ and ϕ at every grid point of the model prior to the calculation. We describe the method of generating the orthogonal coordinate grid system in Section 24.2. Using a conformal transformation in the general sense, the functions that describe the relation between model coordinates (μ, ψ) and geographic coordinates (λ, ϕ), $\lambda(\mu, \psi)$, $\phi(\mu, \psi)$, $\mu(\lambda, \phi)$, and $\psi(\lambda, \phi)$, are obtained.

Because an atmospheric boundary condition is given in many cases at grid points in geographic coordinates, it is necessary to prepare tables for converting the surface atmospheric temperature, the wind stress, and so on. To convert a vector quantity, we must remember that the direction of the μ contour differs from that of the λ contour (meridian). The difference is described in Section 24.3. We can use the functions, $(\lambda, \phi) \Longleftrightarrow (\mu, \psi)$, to convert a scalar quantity as well as sea depth and the Coriolis parameter. The total flux that the ocean receives from the atmosphere should be equal to the total flux that the atmosphere gives to the ocean. The method for conserving the total flux is explored in Section 24.4. The vector operation in generalized orthogonal coordinates is concisely described in Section 24.5.

24.2 Generation of orthogonal coordinate system using conformal mapping

We designate the plane that touches the sphere at the North Pole as $\mathbf{S}_\mathrm{N}$. A polar stereographic projection is a conformal transformation in the general sense, so that an orthogonal coordinate system on the sphere is mapped onto an orthogonal coordinate system on $\mathbf{S}_\mathrm{N}$ and the orthogonality is preserved on the reverse transformation (Figure 24.1). Moreover, if $\mathbf{S}_\mathrm{N}$ is assumed to be a complex plane, various conformal transformations can be defined on it. Therefore, applying (i) the polar stereographic projection, (ii) a conformal transformation on the complex plane $\mathbf{S}_\mathrm{N}$, and (iii) the reverse polar stereographic projection to a geographic coordinate grid point (λ, ϕ) on the sphere, an orthogonal coordinate grid point on the sphere can be obtained (Bentzen et al., 1999).

The functions $\mu(\lambda, \phi)$ and $\psi(\lambda, \phi)$ are obtained by the following procedure:

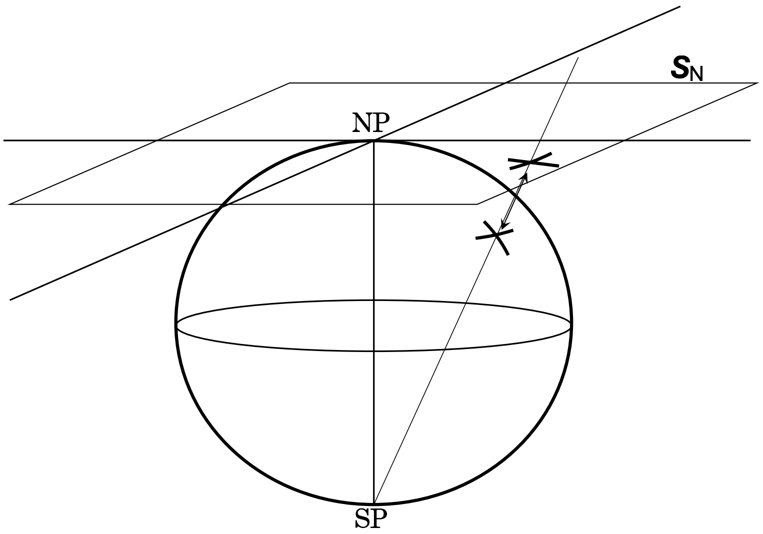

Figure24.1 Schematic illustration of a Polar stereographic projection (a conformal transformation in the general sense between the sphere and $\mathbf{S}_{\mathrm{N}}$).

1. From a point (λ, ϕ) on the sphere to a point z on $\mathbf{S}_{\mathrm{N}}$ (polar stereographic projection). Defining colatitude $\phi' = \pi/2 - \phi$,

$$z = \tan\left(\frac{\phi'}{2}\right) e^{i\lambda}, \tag{24.1}$$

where the origin of $\mathbf{S}_{\mathrm{N}}$ corresponds to $\phi' = 0$ ($\phi = \pi/2$), and the positive part of the real axis corresponds to $\lambda = 0$.

2. Conformal transformation M_{C} on $\mathbf{S}_{\mathrm{N}}$:

$$\zeta = M_{\mathrm{C}}(z). \tag{24.2}$$

3. From a point ζ on $\mathbf{S}_{\mathrm{N}}$ to a point (μ, ψ) on the sphere (reverse polar stereographic projection).

$$\mu = \arg(\zeta), \tag{24.3}$$
$$\psi' = 2 \arctan |\zeta|, \tag{24.4}$$
$$\psi = \pi/2 - \psi'. \tag{24.5}$$

Functions $\lambda(\mu, \psi)$ and $\phi(\mu, \psi)$ are obtained by reversing the above procedure.
Defining $\psi' = \pi/2 - \psi$,

$$\zeta = \tan\left(\frac{\psi'}{2}\right) e^{i\mu}, \tag{24.6}$$

$$z = M_{\mathrm{C}}^{-1}(\zeta), \tag{24.7}$$

and

$$\lambda = \arg(z), \tag{24.8}$$
$$\phi' = 2 \arctan |z|, \tag{24.9}$$
$$\phi = \pi/2 - \phi'. \tag{24.10}$$

Thus, when a model coordinate grid point, $(\mu_0 + \Delta\mu \times (i-1),\ \psi_0 + \Delta\psi \times (j-1))$ is given, we know the geographic position of the point, Coriolis parameter, etc., at once.

Bentzen et al. (1999) used the linear fraction conversion as a conformal transformation on $\mathbf{S}_{\mathrm{N}}$. That is,

$$\zeta = M_{\mathrm{C}}(z) = \frac{(z-a)(b-c)}{(c-a)(b-z)}, \tag{24.11}$$

where the three complex numbers a, b, and c expressed by

$$a = \tan\left(\frac{\phi'_a}{2}\right) e^{i\lambda_a}, \quad b = \tan\left(\frac{\phi'_b}{2}\right) e^{i\lambda_b}, \quad c = \tan\left(\frac{\phi'_c}{2}\right) e^{i\lambda_c}, \tag{24.12}$$

correspond to the three geographic coordinate grid points (λ_a, ϕ_a), (λ_b, ϕ_b), and (λ_c, ϕ_c), which are mapped to the model coordinate grid points, $(\mu, \psi) = (0, \pi/2)$, $(0, -\pi/2)$, $(0, 0)$, respectively. Therefore, the singular point $(\mu, \psi) = (0, \pi/2)$ in the model calculation can be put on Greenland, by setting (λ_a, ϕ_a) at 75°N and 40°W. If `TRIPOLAR` or `JOT` option is specified instead of `SPHERICAL`, then two singular points: $(\mu, \psi) = (0, \pi/2)$ and $(0, -\pi/2)$ can be put on arbitrary land locations by suitably setting (λ_a, ϕ_a) and (λ_b, ϕ_b), which are model parameters (`north_pole_lon`, `north_pole_lat`, `south_pole_lon`, and `south_pole_lat` in degree) that should be specified in namelist `nml_poles` (Table 3.3).

When `TRIPOLAR` option is specified, the parameters are set to $\phi_a = \phi_b =$ 64°N, $\lambda_a =$ 80°E, and $\lambda_b =$ 100°W. The transformed grids are used for the region north of 64°N, and geographic coordinates are used for the region south of 64°N. This tripolar coordinate system can express the Arctic Sea with a higher resolution than the Southern Ocean. The adoption of geographic coordinates south of 64°N enables us to do the assimilation and analysis with relative ease (Figure 24.2).

When `JOT` option is specified, the Joukowski conversion is used as a conformal transformation on $\mathbf{S}_\mathrm{N}$. That is,

$$z = M_C^{-1}(\zeta) = \left(\zeta + \frac{\psi_0'^2}{\zeta}\right) e^{i\mu_0}. \tag{24.13}$$

This Joukowski conversion maps the area outside the circle with a radius of ψ'_0 centered at the origin of the ζ-plane to the whole domain of the z-plane, and rotates it by μ_0. The left panel of Figure 24.2 presents an example where the coordinate system is created by setting ψ'_0 to 20° and μ_0 to 80°. Because there is no discontinuity of grid spacing in this coordinate system, the singular points can be put at various positions. For instance, the singular point on the North American side can be put on the Labrador Peninsula or in Greenland.

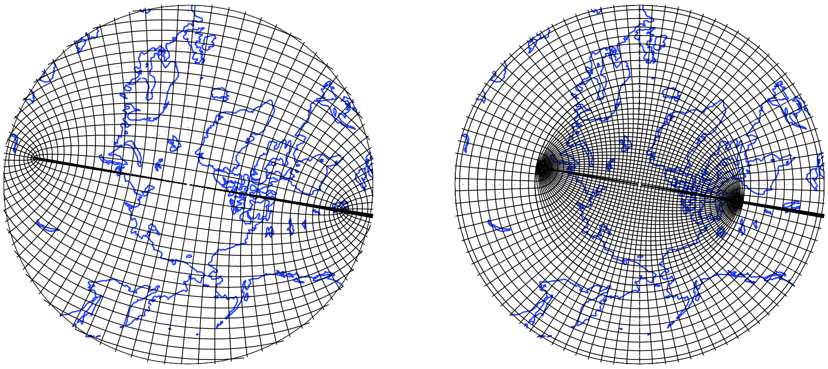

Figure24.2 Model coordinate grid arrangement in the Arctic sea. Left: Grid system made through the Joukowski conversion (`JOT`). Right: Combination of the coordinate systems made through the linear fraction conversion and conventional geographic coordinates (`TRIPOLAR`).

Functions $\lambda(\mu, \psi)$ and $\phi(\mu, \psi)$ are defined as subroutine `mp2lp`, and functions $\mu(\lambda, \phi)$ and $\psi(\lambda, \phi)$ are defined as subroutine `lp2mp`. Module programs `trnsfrm.{spherical, moebius, tripolar, jot}.F90` contain these internal subroutines. These functions, especially `mp2lp`, are frequently used when the topography and the surface boundary condition are made before starting the main integration of model.

24.3 Rotation of vector

A vector expressed in geographic coordinates (λ, ϕ) should be rotated when observed from model coordinates (μ, ψ). Taking advantage of the local orthogonality of coordinate axes, the angle (α) is obtained as an angle at which the meridian

($\lambda = \lambda_0$) of geographic coordinates intersects that of the model coordinates ($\mu = \mu_0$) at a certain point. First, we set

$$z = f(\zeta), \quad z = x + iy, \quad \zeta = u + iv, \tag{24.14}$$

$$f'(\zeta) \quad = \quad \frac{\partial x}{\partial u} + i\frac{\partial y}{\partial u} \quad = \quad \frac{\partial y}{\partial v} - i\frac{\partial x}{\partial v}. \tag{24.15}$$

At a certain point $z_0 = f(\zeta_0)$ on geographic coordinates, the angle θ at which a curve $v = v_0$ meets a straight line $y = y_0$ is given by

$$\tan\theta = \left[\frac{\partial y}{\partial x}\right]_{v_0} = \left[\frac{\partial y}{\partial u}\Big/\frac{\partial x}{\partial u}\right]_{v_0},$$

then (see Figure 24.3),

$$\theta = \arg(f'(\zeta_0)). \tag{24.16}$$

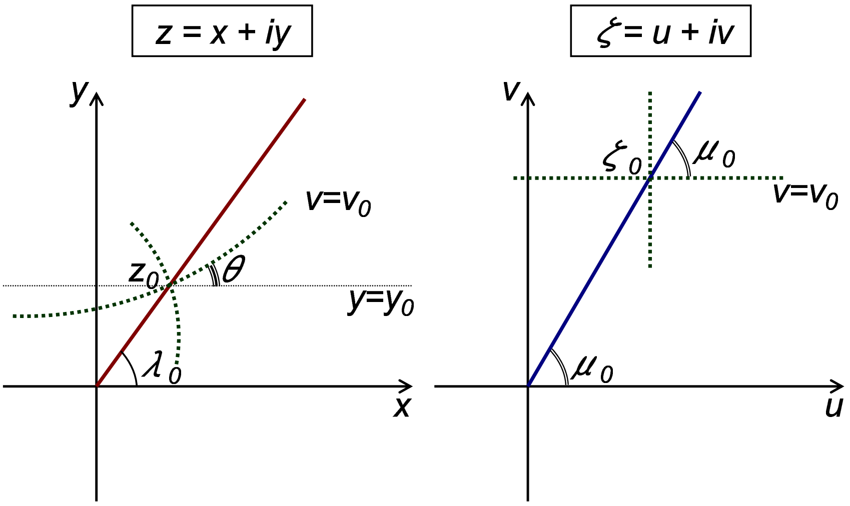

Figure24.3 A meridian (red) in geographic coordinates (λ, ϕ) (left) and a meridian (blue) in model coordinates (μ, ψ) (right).

Assuming $\lambda = \arg(z)$ and $\mu = \arg(\zeta)$, at point ζ_0 the straight line ($v = v_0$) meets the straight line ($\mu = \mu_0$) at the angle $-\mu_0\,(\mu_0 \rightarrow v_0)$, and at point z_0 the meridian ($\lambda = \lambda_0$) meets the curve ($v = v_0$) at the angle $\lambda_0 - \theta\,(v_0 \rightarrow \lambda_0)$. The meridian ($\lambda = \lambda_0$) in λ-ϕ coordinates meets the line ($\mu = \mu_0$) in μ-ψ coordinates at angle α given as follows:

$$\begin{aligned}\alpha &= -\mu_0 + \lambda_0 - \theta \\ &= \lambda_0 - \mu_0 - \arg(f'(\zeta_0)).\end{aligned} \tag{24.17}$$

Subroutine `rot_mp2lp` defined in `trnsfrm.{spherical,moebius,tripolar,jot}.F90` returns $(\cos\alpha, \sin\alpha)$ at a specified grid point of the model. A wind stress vector (τ_x, τ_y) in geographic coordinates should appear in the model ocean described in the μ-ψ coordinate system as $(\tau_x\cos\alpha - \tau_y\sin\alpha, \tau_x\sin\alpha + \tau_y\cos\alpha)$.

24.4 Mapping a quantity from geographic coordinates to transformed coordinates

We consider a method to receive a quantity $G_{I,J}$ given at the geographic coordinate grids (I, J) as the quantity $H_{M,N}$ at the model coordinate grids (M, N) (Figure 24.4). The quantities are wind stress components after the vector rotation, precipitation per unit area, sea surface atmospheric temperature, and so on. In addition, the average depth at a model grid point can also be calculated by the following method because bottom topography (depths of sea floor) is usually given in geographic coordinates.

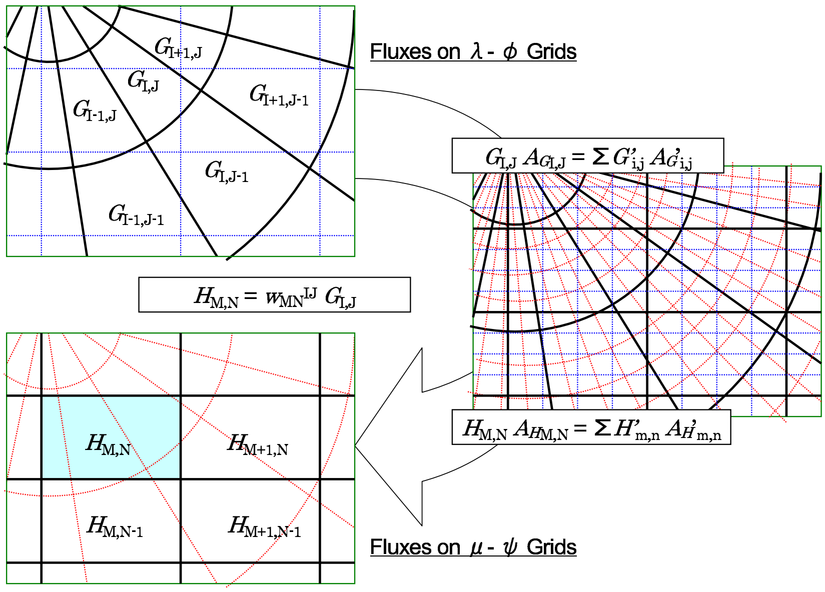
$G_{I+1,J}$
$G_{I,J}$
$G_{I+1,J-1}$
$G_{I-1,J}$
$G_{I,J-1}$
$G_{I-1,J-1}$
Fluxes on λ - ϕ Grids
$G_{I,J}\ A_{G I,J} = \Sigma\, G'_{i,j}\ A_{G' i,j}$
$H_{M,N} = w_{MN}^{IJ}\ G_{I,J}$
$H_{M,N}$
$H_{M+1,N}$
$H_{M,N-1}$
$H_{M+1,N-1}$
$H_{M,N}\ A_{H M,N} = \Sigma\, H'_{m,n}\ A_{H' m,n}$
Fluxes on μ - ψ Grids

Figure24.4 Grids (I, J) and (M, N) subdivided into finer grids (i, j) and (m, n).

Grids (I, J) and (M, N) are suitably subdivided into finer grids (i, j) and (m, n). We call these filter grids. A quantity $G'_{i,j}$ is assumed to be homogeneously distributed in the geographic filter grids (i, j) covered by grid (I, J),

$$G'_{i,j} = G_{I,J}.$$

Assume the quantity at a model filter grid $H'_{m,n}$ is equal to that at the nearest geographic filter grid,

$$H'_{m,n} = G'_{i(m,n),j(m,n)}.$$

The quantity at model grid (M, N) is obtained as the area-weighted average:

$$H_{M,N} = \frac{1}{A_{H\,M,N}} \sum_{m,n} A_{H'\,m,n} H'_{m,n}, \tag{24.18}$$

where $A_{H\,M,N}$ is the area of model grid and $A_{H'\,m,n}$ is the area of model filter grid.

When the grid intervals of geographic filter grid (i, j) and model filter grid (m, n) are extremely small, the total quantity (flux) received on the model grids (M, N) is equal to the total quantity (flux) given by the geographic grids (I, J). The relation between the quantity in the geographic grids and that in the model grids is defined by weight w,

$$H_{M,N} = \sum_{I,J} w(M, N, I, J) G_{I,J}. \tag{24.19}$$

How is the quantity converted in an actual calculation in the model?

1. When the strict conservation of quantity (flux) is necessary:
   Fresh water is not permitted to be generated or vanish at the surface boundary in a run using an atmosphere-ocean coupled model, for instance. In this case, $w(M, N, I, J)$ is prepared beforehand, and the flux is passed from the atmosphere through equation (24.19) to the ocean. The resolution of the filter grid need not be extremely fine, provided that every geographic filter grid is linked to more-than-zero model filter grids and

$$\sum_{I,J} A_{G\,I,J} = \sum_{i,j} A'_{G'\,i,j} = \sum_{m,n} A'_{H'\,m,n} = \sum_{M,N} A_{H\,M,N}.$$

2. When conservation need not be guaranteed:
   When the ocean model is driven by the surface boundary condition based on atmospheric re-analysis data, the amount of fresh water entering the sea as precipitation and river discharge is not equal to that drawn from the ocean

through evaporation and sublimation. Therefore, the global sea surface height rises or descends during years of integration. It is not very important to pursue complete conservation of fresh-water under this condition. In such a case, the flux at a model grid point can be prepared beforehand using equation (24.19), to avoid the time-consuming flux conversions in the model calculation.

24.5 Vector operation and differentiation in generalized orthogonal coordinates

To formulate the model equations, we have to know the vector operation and differentiation in generalized orthogonal coordinates. Some basic formulae used in formulating primitive equations are presented here.

The line element vector $\delta\mathbf{x}$ at a certain point (μ, ψ, r) in an arbitrary general orthogonal coordinate system is expressed as

$$\delta\mathbf{x} = h_\mu \delta\mu \mathbf{e}_\mu + h_\psi \delta\psi \mathbf{e}_\psi + h_r \delta r \mathbf{e}_r, \tag{24.20}$$

where basis vectors $\mathbf{e}_\mu$, $\mathbf{e}_\psi$, and $\mathbf{e}_r$ are mutually orthogonal unit vectors, and h_μ, h_ψ, and h_r are scale factors.

Defining

$$\nabla = \frac{\mathbf{e}_\mu}{h_\mu}\frac{\partial}{\partial\mu} + \frac{\mathbf{e}_\psi}{h_\psi}\frac{\partial}{\partial\psi} + \frac{\mathbf{e}_r}{h_r}\frac{\partial}{\partial r}, \tag{24.21}$$

the gradient of scalar $A(\mu, \psi, r)$ is

$$\nabla A = \frac{\mathbf{e}_\mu}{h_\mu}\frac{\partial A}{\partial\mu} + \frac{\mathbf{e}_\psi}{h_\psi}\frac{\partial A}{\partial\psi} + \frac{\mathbf{e}_r}{h_r}\frac{\partial A}{\partial r}, \tag{24.22}$$

and the divergence of vector $\mathbf{A} = A_\mu \mathbf{e}_\mu + A_\psi \mathbf{e}_\psi + A_r \mathbf{e}_r$ is

$$\nabla \cdot \mathbf{A} = \frac{1}{h_\mu h_\psi h_r}\left[\frac{\partial(h_\psi h_r A_\mu)}{\partial\mu} + \frac{\partial(h_r h_\mu A_\psi)}{\partial\psi} + \frac{\partial(h_\mu h_\psi A_r)}{\partial r}\right]. \tag{24.23}$$

The r component of curl$\mathbf{A}$ is

$$\frac{1}{h_\mu h_\psi}\left[\frac{\partial(h_\psi A_\psi)}{\partial\mu} - \frac{\partial(h_\mu A_\mu)}{\partial\psi}\right]. \tag{24.24}$$

The calculation of velocity advection includes $(\mathbf{a}\cdot\nabla)\mathbf{A}$, where $\mathbf{a}$ is an arbitrary vector ($\mathbf{a} = a_\mu \mathbf{e}_\mu + a_\psi \mathbf{e}_\psi + a_r \mathbf{e}_r$).

The μ component of $(\mathbf{a}\cdot\nabla)\mathbf{A}$ is

$$\mathbf{a}\cdot\nabla A_\mu + \frac{A_\psi}{h_\mu h_\psi}\left(a_\mu \frac{\partial h_\mu}{\partial\psi} - a_\psi \frac{\partial h_\psi}{\partial\mu}\right) + \frac{A_r}{h_r h_\mu}\left(a_\mu \frac{\partial h_\mu}{\partial r} - a_r \frac{\partial h_r}{\partial\mu}\right). \tag{24.25}$$

The second and third terms are so-called "metric" terms in the equation of motion in spherical coordinates.

These expressions in spherical coordinates (λ, ϕ, r) are shown next. Defining longitude λ, latitude ϕ, and radius of the earth r, scale factors are $h_\lambda = r\cos\phi$, $h_\phi = r$, and $h_r = 1$.

Velocity vector $\mathbf{v}$ is

$$\mathbf{v} = u\mathbf{e}_\lambda + v\mathbf{e}_\phi + w\mathbf{e}_r, \tag{24.26}$$

where $\mathbf{e}_\lambda$, $\mathbf{e}_\phi$, and $\mathbf{e}_r$ are the eastward, northward, and upward unit vectors, respectively, and $(u, v, w) = (r\cos\phi\dot{\lambda}, r\dot{\phi}, \dot{r})$.

The gradient of scalar function $A(\lambda, \phi, r)$ is,

$$\nabla A = \frac{\mathbf{e}_\lambda}{r\cos\phi}\frac{\partial A}{\partial\lambda} + \frac{\mathbf{e}_\phi}{r}\frac{\partial A}{\partial\phi} + \mathbf{e}_r\frac{\partial A}{\partial r}, \tag{24.27}$$

where

$$\nabla = \frac{\mathbf{e}_\lambda}{r\cos\phi}\frac{\partial}{\partial\lambda} + \frac{\mathbf{e}_\phi}{r}\frac{\partial}{\partial\phi} + \mathbf{e}_r\frac{\partial}{\partial r}. \tag{24.28}$$

For vector $\mathbf{A} = A_\lambda \mathbf{e}_\lambda + A_\phi \mathbf{e}_\phi + A_r \mathbf{e}_r$, the divergence is

$$\nabla \cdot \mathbf{A} = \frac{1}{r\cos\phi}\left[\frac{\partial A_\lambda}{\partial\lambda} + \frac{\partial(\cos\phi A_\phi)}{\partial\phi}\right] + \frac{\partial(r^2 A_r)}{r^2\partial r}. \tag{24.29}$$

and the r component of curl$\mathbf{A}$ is

$$[\text{curl}\mathbf{A}]_r = \frac{1}{r\cos\phi}\left[\frac{\partial A_\phi}{\partial\lambda} - \frac{\partial(\cos\phi A_\lambda)}{\partial\phi}\right]. \tag{24.30}$$

The λ component of $(\mathbf{a}\cdot\nabla)\mathbf{A}$ is

$$[(\mathbf{a}\cdot\nabla)\mathbf{A}]_\lambda = \mathbf{a}\cdot\nabla A_\lambda - \frac{A_\phi a_\lambda \tan\phi}{r} + \frac{A_r a_\lambda}{r}. \tag{24.31}$$

The Coriolis force in generalized orthogonal coordinates (μ, ψ, r) is given as

$$2\Omega\times\mathbf{v} = (2\Omega_\psi w - 2\Omega_r v)\mathbf{e}_\mu + (2\Omega_r u - 2\Omega_\mu w)\mathbf{e}_\psi + (2\Omega_\mu v - 2\Omega_\psi u)\mathbf{e}_r, \tag{24.32}$$

where $\Omega = \Omega_\mu \mathbf{e}_\mu + \Omega_\psi \mathbf{e}_\psi + \Omega_r \mathbf{e}_r$ is the rotation vector of the Earth, and $\mathbf{v} = u\mathbf{e}_\mu + v\mathbf{e}_\psi + w\mathbf{e}_r$ is the velocity vector. We designate $f_\mu = 2\Omega_\mu$, $f_\psi = 2\Omega_\psi$, and $f = f_r = 2\Omega_r$ in Chapter 2. The rotation vector of the Earth is $(\Omega_\lambda, \Omega_\phi, \Omega_r) = (0, \Omega\cos\phi, \Omega\sin\phi)$ in geographic coordinates (λ, ϕ, r).

Chapter 25

User's Guide

This chapter briefly explains the procedures needed to run MRI.COM. The description in this chapter is based on MRI.COM version 5.0 and some contents presented in this chapter may not be used for the latest version. It is recommended that users refer to `README_First.md`, `README_Options.md`, `README_Namelist.md`, `README_Monitor.md`, and `README_Restart.md` in the `docs/` directory when setting up a model.

The minimal information to prepare, run, and post-process is presented in this chapter in the following order:

- **Model setup:** User defined parameter files and compilation (Section 25.1 and Table 25.1).
- **Input data:** Grid spacing, topography, and surface forcing etc., to be read at run time (Section 25.2 and Table 25.2).
- **Restart file:** Explanation of restart files (Section 25.3).
- **Execution:** Explanation of the runtime parameters that control time-integration (Section 25.4).
- **Post process:** A description of monitor files (Section 25.5).

Note that cgs units are employed to express physical values in the model.

We are developing a comprehensive package of tools "MRI.COM eXecution Environment (MXE)" that aggregates programs for preprocessing, execution, postprocessing, and analysis of MRI.COM experiments. This is briefly explained in Section 25.6. Appendix (Sec. 25.7) contains website information and a list of model compilation options.

25.1 Model setup

25.1.1 Model configuration file (configure.in)

This section describes the procedures necessary for setting up the model and compiling its programs. First, prepare `configure.in` that contains the information about model options and grid size. This is needed for compilation. Additional parameters are required for some particular model options. Those are listed in Table 25.1 (see also `README_Options.md`). Following is an example of `configure.in`.

An example `configure.in` for Global tripolar 1° × 0.5° grid model

```
DEFAULT_OPTIONS="IDEALAGE ICE SIDYN CALALBSI SMAGOR VIS9P DIFAJS GLS VVDIMP
 SOMADVEC ISOPYCNAL HFLUX TAUBULK WFLUX RUNOFF SFLUXR BULKNCAR
 CYCLIC BBL TRIPOLAR PARALLEL"
NAME_MODEL='GLOBAL'
IMUT=364
JMUT=368
KM=51
NPARTX=8
NPARTY=4
NUM_ICECAT=5
NUMTRC_P=1
```

Table25.1 Model parameters to be set in configure.in

option name	variable name	description
always required	IMUT, JMUT, KM	zonal/meridional/vertical grid number
	NAME_MODEL	name of the model (default = "tmOGCM")
	NSFMRGN	the number of side-boundary ghost cells to reduce the communication cost in parallel computation (see Ishizaki and Ishikawa, 2006)
PARALLEL	NPARTX, NPARTY	the number of zonally/meridionally partitioned region for a computation using parallel processors: the number of parallel processes should be NPARTX × NPARTY
passive tracers	NUMTRC_P	only when any passive tracer is calculated (NUMTRC_P $\geq$ 1)
ICE	NUM_ICECAT	the number of thickness categories of sea ice

25.1.2 Compilation of the model

A standard compiling script is prepared as compile.sh in the src/ directory. The part depending on the system (OS, Fortran compiler, and compiler option) are found in Machines/ directory.

To compile the programs, execute compile.sh. The script compile.sh creates param.F90, Icecat/ice_param.F90, and Makefile from configure.in by running configure, and then executes the command make to create the executable file ogcm. The environment variables for compilation are set in configure using the options prescribed in configure.in. If configure.in is newer than param.F90, a new param.F90 is created based on the parameter values defined in configure.in.

The program files that should be compiled are automatically selected according to the descriptions of the relationships in Makefile, but users should be careful since it might not be perfect. The compilation should be carried out after executing make clean, when any compile option in configure.in is changed.

25.2 Preparation of input data files for execution

Data files listed on Table 25.2 must be prepared, according to user's specification of compile options and runtime parameters. See Section 25.3 for how to handle restart files.

Table25.2 Main input data files and their related program files. Here, *name_model* represents the specific name given as NAME_MODEL in configure.in.

subject	file name specified in (NAMELIST.*name_model*)
runtime parameters	NAMELIST.*name_model*
specification about monitoring	NAMELIST.*name_model*.MONITOR
variable horizontal grid spacing	file_dxdy_tbox_deg (nml_horz_grid)
variable vertical grid spacing	file_dz_cm (nml_vert_grid)
grid cell area and line elements	file_scale (nml_grid_scale)
topography	file_topo (nml_topo)
reference data for tracers	trcref(_surf)_conf%file_data
restoring coefficient for tracers	rstcoef(_surf)_conf%file_data
surface forcing	file_data (nml_force_data)
surface forcing grid	file_data_grid (nml_force_data)
restart files	file_base (nml_restart)

25.2.1 Grid spacing and cell area

Details on how to specify grid information and how to prepare the necessary data is given in Section 3.6. The grid spacing data file should be prepared for each of the horizontal (file_dxdy_tbox_deg) or vertical (file_dz_cm) directions when variable grid spacing is used for that direction. The units are in degrees for the horizontal and in cm for the vertical.

When the model grid points are defined on the basis of general orthogonal coordinates, the quarter cell area and line elements should be prepared. The units are in cgs. It is read from the file file_scale.

When spherical coordinates are used (SPHERICAL option), e.g., the grids are defined on geographical latitude and longitude, the grid information is analytically calculated in the model, and the file `file_scale` is not necessary.

25.2.2 Topography

Land-sea distribution and sea-floor topography should be given by the topography data file `file_topo`. The topographic data consist of the 4-byte integer array `HO4(imut, jmut)` that contains the sea floor depths of the velocity grid points (in cm) and the 4-byte integer array `EXNN(imut, jmut)` that contains its corresponding vertical level. They should be written unformatted and sequentially as follows:

Format of topographic data (`file_topo`)

```
    integer(4) :: ho4(imut,jmut),exnn(imut,jmut)
    open(unit=nu,file=file_topo)
    write(unit=nu) ho4, exnn
```

An example of the topography for global 1° × 0.5° model is shown in Figure 25.1. In creating a model topography, especially for a low-resolution model, the user should be careful that the important gateways for the ocean circulation be kept open and that the land blocking the ocean circulation be kept closed.

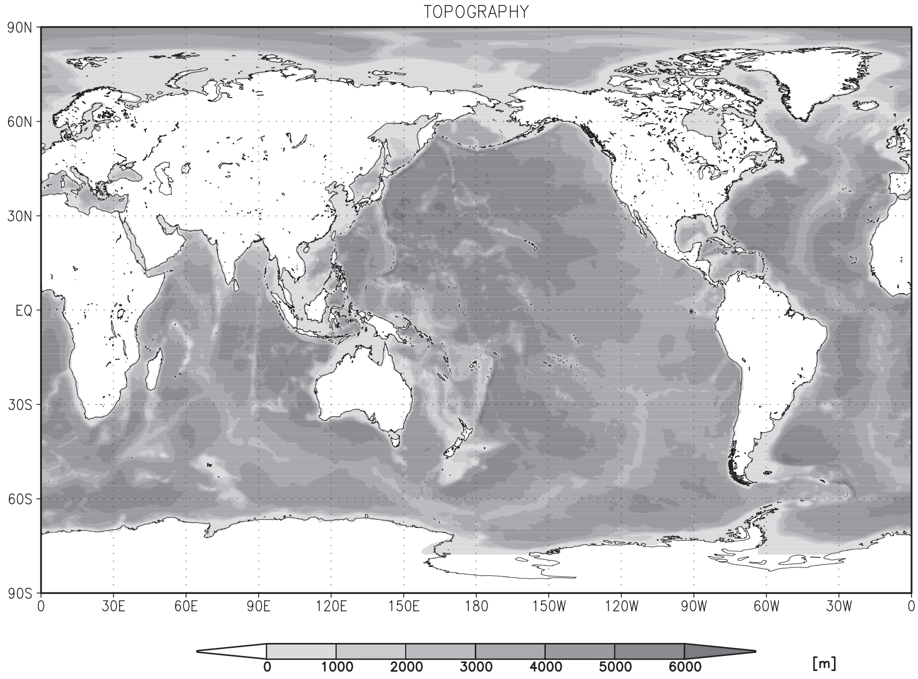

Figure25.1 Example of ocean model topography (global 1° × 0.5° grid model).

25.2.3 Reference data and restoring coefficient for tracers

For each tracer, the integration may be started from an initial state based on the reference data and the tracer values may be restored to the reference data during the integration. To do this, tracer reference values and restoring coefficients should be prepared. See Chapter 13 for how to prepare these data.

25.2.4 Surface forcing data

See Chapter 14 for how to prepare surface forcing data. The surface forcing data are read at a uniform time interval. A leap year is set according to an optional namelist, `nml_calendar` (Table 25.10). Climatological data may be used repeatedly.

The following data files should be prepared according to the chosen model options. Each file is opened only once at the beginning of run time and thus should contain all the data needed for that run.

Table25.3 Surface data to be read from namelist nml_force_data.

	name	units	usage
X-ward wind stress	U-wind	dyn cm^{-2}	if not TAUBULK
Y-ward wind stress	V-wind	dyn cm^{-2}	if not TAUBULK
X-ward wind speed	U-wind	cm s^{-1}	if TAUBULK
Y-ward wind speed	V-wind	cm s^{-1}	if TAUBULK
Downward shortwave radiation	ShortWave	erg s^{-1} cm^{-2} = 10^{-3} W m^{-2}	
Downward longwave radiation	LongWave	erg s^{-1} cm^{-2}	
Surface air temperature	TempAir	°C	
Surface air specific humidity	SphAir	1	
Scalar wind speed	ScalarWind	cm s^{-1}	unnecessary if TAUBULK
Sea level pressure	SeaLevelPressure	hPa	also available for SLP
precipitation	Precipitation	g s^{-1} cm^{-2}	WFLUX
river discharge	RiverDischargeRate	g s^{-1} cm^{-2}	RUNOFF
Sea ice area fraction	IceConcentrationClimatology	1	ICECLIM

25.3 Restart file

A set of restart files provides an instantaneous state necessary to resume a model integration. This section explains how to handle restart files.

25.3.1 Restart for the ocean model

Restart files for the ocean model are specified using a common namelist block, nml_restart, in the same manner as history outputs. The namelists must be written in NAMELIST.*name_model* (default) as follows:

An example namelist for temperature restart files

```
&nml_restart
   name      = 'temperature',
   file_base = 'result/rs_t',
   (interval_step = 0,)
/
```

where name is a variable name to be input/output (listed in Table 25.4), and file_base specifies the basename of the restart files. This information is shared by both input and output files. In the above example, the name of a restart file for temperature is result/rs_t.YYYYMMDDHHMMSS. A suffix indicating the date and time of data is always added to the basename. (Namlist block nmlrs_ in MRI.COMv4 has been removed.)

Table 25.4 lists available names in nml_restart (name is not case sensitive). Required elements depend on model options. See docs/README_Restart.md for more information. An example for a set of biogeochemical tracers is also shown in Section 19.6.

Table25.4 Restart variables and their namelist block that specifies their restart file attributes.

variable name	model options	namelist in MRI.COMv4
Potential temperature		nmlrs_t
Salinity		nmlrs_s
X Velocity		nmlrs_u
Y Velocity		nmlrs_v
		Continued on next page

Table 25.4 – continued from previous page

variable name	model options	namelist in MRI.COMv4
Sea Surface Height		nmlrs_ssh
X Transport		nmlrs_uml
Y Transport		nmlrs_vml
X diffusion flux for SSH		nmlrs_ssh_dflx_x
Y diffusion flux for SSH		nmlrs_ssh_dflx_y
Global Mean Density and Pressure	required if CALPP	nmlrs_density
Vertical Tracer Diffusivity	VVDIMP	nmlrs_vmix_avd
Vertical Momentum Diffusivity	VVDIMP	nmlrs_vmix_avm
passive tracers (see Section 19.6)	NUMTRC_P > 0	nmlrs_ptrc
Tidal X Transport	TIDE	nmlrs_tide_um
Tidal Y Transport	TIDE	nmlrs_tide_vm
Tidal Sea Surface Height	TIDE	nmlrs_tide_ssh
M-Y eddy kinetic energy diffusion	MELYAM	nmlrs_my_avq
M-Y turbulent velocity scale	MELYAM	nmlrs_my_q
M-Y length scale	MELYAM	nmlrs_my_alo
Noh-Kim eddy kinetic energy diffusion	NOHKIM	nmlrs_nk_avq
Noh-Kim eddy kinetic energy	NOHKIM	nmlrs_nk_eb
GLS eddy kinetic energy diffusion	GLS	nmlrs_gls_avk
GLS generic variable diffusion	GLS	nmlrs_gls_avp
GLS eddy kinetic energy	GLS	nmlrs_gls_eke
GLS generic variable	GLS	nmlrs_gls_psi
GLS length scale	GLS	nmlrs_gls_alo
(see Section 25.3.2)	SOMADVEC and adv_scheme%name = ”som” in nml_tracer_data for any tracer	nml_somadv
Sea ice	ICE	nml_seaice_rst_in

Restart files are saved in a Fortran direct-access format, which can be read by a following program.

Program to read a restart file of the ocean model

```
character(14)        :: date = '20010101000000'  !- for 0:00z1JAN2001
real(8)              :: d(imut,jmut,km)
integer(4),parameter :: nu = 10 ! device number

open(nu, file='result/rs_t.'//date, form='unformatted', &
      & access='direct',recl=imut*jmut*km*8)
read(nu,rec=1) d
close(nu)
```

The file endian is that of the computer where the model runs. There are support tools to visualize restart files in the directory, tools/MK_GRADS. Please follow the instructions given by tools/MK_GRADS/00README.txt. The MXE package (Section 25.6) also has tools for visualization in the directory, postp/.

25.3.2 Restart for the SOM advection scheme

When the SOM tracer advection scheme is used (SOMADVEC), the model may read and write additional 9 restart files of the 2nd order moments for each tracer. Their names are BASENAME_SOM_SS.YYYYMMDDHHMMSS, where BASENAME indicates a character string specified by file_base in namelist nml_restart, SS elements of the moments (9 types), and the suffix for the date and time. The format of each file is the same as the tracer restart files. See Section 10.5 for the SOM scheme, and nml_tracer_data in docs/README_Namelist.md to control the SOM restart input/output. (Naming of the SOM restart files was changed in MRI.COMv5. Use tools/ver46to47/SOMRESTART/ to rename SOM restart files.)

25.3.3 Restart for the sea ice model

To handle restart files of the sea ice model, a namelist block, `nml_restart` with `name` = "Sea ice", must be specified. (`nml_seaice_rst_in` and `nml_seaice_rst_out` are obsolete.) The restart file contains a number of variables needed to restart the sea ice model, and can be read by the following program.

Program to read restart for the sea ice model

```
integer(4), parameter :: ncat = 5 ! number of thickness categories
! ice concentration
real(8) :: aicen (1:imut,1:jmut,0:ncat), a0iceo(imut,jmut)
! ice thickness
real(8) :: hicen (1:imut,1:jmut,0:ncat), hiceo (imut,jmut)
! averaged sea-ice thickness
real(8) :: hin   (1:imut,1:jmut,0:ncat), hi    (imut,jmut)
! snow depth
real(8) :: hsnwn (1:imut,1:jmut,0:ncat), hsnwo (imut,jmut)
! averaged snow thickness
real(8) :: hsn   (1:imut,1:jmut,0:ncat), hsnw  (imut,jmut)
! ice surface temperature
real(8) :: tsfcin(1:imut,1:jmut,0:ncat), tsfci (imut,jmut)
! ice temperature
real(8) :: t1icen(1:imut,1:jmut,0:ncat)
! sea surface skin temperature
real(8) :: t0icen(1:imut,1:jmut,0:ncat)
! sea surface skin salinity
real(8) :: s0n   (1:imut,1:jmut,0:ncat)
! skin temperature beneath the sea ice
real(8) :: t0iceo(imut,jmut)
! skin temperature in the open leads
real(8) :: t0icel(imut,jmut)
! stress tensor
real(8) :: sigma1(imut,jmut), sigma2(imut,jmut), sigma3(imut,jmut)

integer(4),parameter :: nu = 10 ! device number
integer(4)          :: i = 0, m

open(nu, file='result/rs_ice', form='unformatted', &
  & access='direct', recl=imut*jmut*8)

do m = 0, ncat ; i = i + 1 ; read(nu,rec=i) aicen(1:imut,1:jmut,m) ; enddo
do m = 0, ncat ; i = i + 1 ; read(nu,rec=i) hin  (1:imut,1:jmut,m) ; enddo
do m = 0, ncat ; i = i + 1 ; read(nu,rec=i) hsn  (1:imut,1:jmut,m) ; enddo
do m = 0, ncat ; i = i + 1 ; read(nu,rec=i) hicen(1:imut,1:jmut,m) ; enddo
do m = 0, ncat ; i = i + 1 ; read(nu,rec=i) hsnwn(1:imut,1:jmut,m) ; enddo
do m = 0, ncat ; i = i + 1 ; read(nu,rec=i) tsfcin(1:imut,1:jmut,m) ; enddo
do m = 0, ncat ; i = i + 1 ; read(nu,rec=i) t1icen(1:imut,1:jmut,m) ; enddo
do m = 0, ncat ; i = i + 1 ; read(nu,rec=i) t0icen(1:imut,1:jmut,m) ; enddo
do m = 0, ncat ; i = i + 1 ; read(nu,rec=i) s0n(1:imut,1:jmut,m) ; end do

i = i + 1 ; read(nu,rec=i) a0iceo(1:imut,1:jmut)
i = i + 1 ; read(nu,rec=i) hi(1:imut,1:jmut)
i = i + 1 ; read(nu,rec=i) hsnw(1:imut,1:jmut)
i = i + 1 ; read(nu,rec=i) hiceo(1:imut,1:jmut)
i = i + 1 ; read(nu,rec=i) hsnwo(1:imut,1:jmut)
i = i + 1 ; read(nu,rec=i) uice(1:imut,1:jmut)
i = i + 1 ; read(nu,rec=i) vice(1:imut,1:jmut)
i = i + 1 ; read(nu,rec=i) sigma1(1:imut,1:jmut)
i = i + 1 ; read(nu,rec=i) sigma2(1:imut,1:jmut)
i = i + 1 ; read(nu,rec=i) sigma3(1:imut,1:jmut)

close(nu)
```

25.3.4 Control of input and output at run time

a. Input

Input of restart files at run time is controlled by namelist listed on Table 25.5. By setting `l_rst_in = .true.` in `nml_run_ini_state`, all necessary elements are read from restart files. However, the specification by `l_rst_in` may be overridden for some elements by specifying namelists dedicated to those elements. Note that model tries to read restart files for all passive tracers, thus namelist `nmlrs_ptrc` must be always specified appropriately.

Table25.5 Namelist blocks that control input of restart files.

namelist block	variable name	Usage
`nml_run_ini_state`	`l_rst_in`	`.true.` : read restart files and resume integration `.false.` (default): initial state is set internally No motion (u,v,ssh=0). Initial condition for temperature and salinity are deteremined by `nml_tracer_run`
`nml_barotropic_run`	`l_rst_barotropic_in`	default = `l_rst_in`
`nml_baroclinic_run`	`l_rst_baroclinic_in`	default = `l_rst_in`
`nml_tracer_run`	`l_rst_tracer_in`	default = `l_rst_in` applicable to only temperature and salinity
`nml_vmix_run`	`l_rst_vmix_in`	default = `l_rst_in`
`nml_tide_run`	`l_rst_tide_in`	default = `l_rst_in`
`nml_melyam_run`	`l_rst_melyam_in`	default = `l_rst_in`
`nml_nohkim_run`	`l_rst_nohkim_in`	default = `l_rst_in`
`nml_gls_run`	`l_rst_gls_in`	default = `l_rst_in`
`nml_density_run`	`l_rst_density_in`	default = `l_rst_in`
`nml_seaice_run`	`l_rst_seaice_in`	default = `.false.`
`nml_tracer_data`	`adv_scheme%lrstin_som`	= `.true.` to read restart of SOM scheme
`nml_tracer_data`	`adv_scheme%lrstout_som`	= `.true.` to write restart of SOM scheme

b. Output

Model always try to write restart files of the oceanic part according to the specification by the namelists listed on Table 25.4. Users have to specify those namelists appropriately to obtain restart files and terminate the model normally. Output from the SOM scheme may be suppressed if `adv_scheme%lrstout_som` = `.false.` in `nml_tracer_data`.

25.4 Execution

To run a model, a shell script that handles input/output files, executes the compiled binary `ogcm`, and post-processes is usually prepared. The following two namelist files have to be given.

- `NAMELIST.`*name_model* gives runtime parameters. See `docs/README_Namelist.md` for details.
- `NAMELIST.`*name_model*`.MONITOR` specifies sampling. See `docs/README_Monitor.md` for details.

Runtime parameters relevant to time-integration are listed on Tables 25.6 through 25.11. Other parameters are explained in corresponding chapters.

Table25.6 Namelist `nml_time_step`.

variable name	units	description	usage
`dt_sec`	sec, real(8)	unit time interval for equation of motion and tracer	required
`alpha_bryan_1984`	real(8)	acceleration parameter α of Bryan (1984). See Section 2.3.1.	default = 1
`l_monitor_time`	logical	time step monitor is written to standard output ()	default = `.true.`

Table25.7 Namelist `nml_run_period`.

variable name	units	description	usage
`nstep_total`	integer	total number of time step of this run	required

Table25.8 Namelist `nml_exp_start` that specifies start time of the whole experiment.

variable name	units	description	usage
`year`	integer(4)	year	default = −999
`month`	integer(4)	month	default = 1
`day`	integer(4)	day	default = 1
`hour`	integer(4)	hour	default = 0
`minute`	integer(4)	minute	default = 0
`second`	integer(4)	second	default = 0

Table25.9 Namelist `nml_run_ini` that specifies start time of this run.

variable name	units	description	usage
`year`	integer(4)	year	default = 1
`month`	integer(4)	month	default = 1
`day`	integer(4)	day	default = 1
`hour`	integer(4)	hour	default = 0
`minute`	integer(4)	minute	default = 0
`second`	integer(4)	second	default = 0

Table25.10 Namelist `nml_calendar` that specifies treatment of leap year.

variable name	units	description	usage
`l_force_leap`	logical	Use `l_leap` to decide whether the current year is leap or not.	default = `.false.`, the realistic calendar is followed
`l_leap`	logical	the current year is a leap year or not.	default = `.false.`

Table25.11 Namelist `nml_stdout` that specifies standard output (program log).

variable name	units	description	usage
`l_stdout2file`	logical	Standard output is written to separate files for MPI processes	default = `.false.`
`file_base_stdout`		file name is *name_model*-`file_base_stdout`-stdout.XXXX (XXXX: mpi process number)	
`l_debug`	logical	debug mode	default = `.false.`

25.5 Monitoring

This section summarizes outputs of states in experiments (so called history data) used for monitoring and analyses.

25.5.1 History of the ocean and ice models

Specification of history outputs is common in the ocean and sea ice models. Users prepare a namelist file named `NAMELIST`.*name_model*.`MONITOR` dedicated to history outputs (*name_model* can be specified by `NAME_MODEL` in `configure.in`, default=`OGCM`), and write the following namelist blocks, `nml_history`, in it as needed,

An example namelist for mean temperature output

```
&nml_history
  name          = 'temperature'
  file_base     = 'result/hs_t'
  interval_step = 48
  [suffix       = 'day']
/
```

where `name` specifies the variable name to be output、 `file_base` the basename of the history files, `interval_step` the output interval in terms of the integration time step. An option item of `suffix` specifies the depth of calendar date and time used in the file suffix. In the above example, the time-mean temperature is written to the file, `result/hs_t.YYYYMMDD`, at every 48-th time step.

History files are saved in a Fortran direct-access format, which can be read by a following program.

Program to read ocean model history data

```
character(8)         :: date = '20010101'  !- for 1JAN2001
real(4)              :: d(imut,jmut,km)
integer(4),parameter :: nu = 10 ! device number

open(nu, file='result/hs_t.'//date, form='unformatted', &
  & access='direct', recl=imut*jmut*km*4)
read(nu,rec=1) d
close(nu)
```

A GrADS control file to visualize data is also made by default (`result/hs_t.ctl` in the above example).

The state variables that can be monitored by namelist `nml_history` are listed in `docs/README_Monitor.md`. There are many options in `nml_history`, such as netCDF output, snapshot output, averaging in the model region, output in a specified sub region, addition of an offset value, multiplication by a factor, and so on. See `docs/README_Monitor.md` for the available options.

History outputs of the sea ice model are also specified by `nml_history` in the same manner as the ocean model. State variables that can be monitored are listed in the sea ice section of `docs/README_Monitor.md`. The format of output files is also common, except that a missing value for grids of ocean without sea ice should be different from the one for land grids. By default, the value of 0.e0 is used for the former, while -9.99e33 for the latter. (These values can be changed by `nml_seaice_hst` written in `NAMELIST`.*name_model*.)

25.6 MRI.COM eXecution Environment (MXE)

We have been developing a package "MRI.COM eXecution Environment (MXE)" that aggregates programs for preprocessing, execution, postprocessing, and analysis of MRI.COM experiments. By using prepared tools, users can relatively easily perform complex work of model building and various analyses. In fact, this package is also used as a common basis for developing various ocean models at the Meteorological Research Institute.

The MXE package includes the following tools.

- Preprocessing tools for creating experiment setup files (directory `prep/`)
- Directory template for running MRI.COM (`exp/`)
- Postprocessing tools for visualizing output files (`postp/`)
- Fortran analysis tools (`anl/`)
- Python analysis tools (`anlpy/`)
- A Fortran library that provides basic subroutines such as grid and topography information (`lib/`)

The main programs are written in Fortran and sample shell scripts are also included for execution. Several tools in the package have been confirmed by unit testing. This package is managed independently from MRI.COM, but it is provided to users as an accompanying material. Since it is often updated like MRI.COM, it is recommended to use the latest version. For details on how to use MXE, see the file README-MXE.md in the top directory (in Japanese).

25.7 Appendix

25.7.1 Web site

See the web page https://mri-ocean.github.io/ for getting MRI.COM.

25.7.2 Model options

The model options are listed on Table 25.12. Only major options are listed here. Description about those related to bio-geochemical models can be found in Chapter 19. The description of all options for the latest version can be found in src/README.Options. Though an expression like OGCM_PARALLEL is used in the source program, OGCM_ is omitted when users specify options in configure.in.

Table25.12 Description of Model Options

Model option	Description
BBL	uses the bottom boundary layer model
BIHARMONIC	uses biharmonic operator for both horizontal viscosity and diffusion (*) If ISOPYCNAL is also selected, the biharmonic form is used only for viscosity and not for diffusion.
BULKNCAR	Large and Yeager (2004; 2009) is used for the surface flux bulk formula. This option corresponds to the COREs. (*) BULKNCAR is available only for HFLUX case.
CALALBSI	Sea-ice albedo is calculated using sea-ice conditions according to Los-Alamos model instead of using a constant value
CALPP	considers the time variation of pressure for the equation of state
CARBON	bio-geochemical process is included (*) numtrc_p= 4 for Obata-Kitamura model; numtrc_p= 8 for NPZD model
CBNHSTRUN	atmospheric pCO_2 is given from file (*) use with CARBON
CHLMA94	shortwave penetration scheme with chlorophyll concentration by Morel and Antoine (1994) (*) use with NPZD and SOLARANGLE
CYCLIC	uses zonally cyclic condition
DIFAJS	sets large vertical diffusion coefficient ($1.0\,\mathrm{m^2\,s^{-1}}$) between unstable points instead of convective adjustment
F2003	Program uses Fortran 2003 features
FSVISC	calculates viscosity explicitly in the barotropic momentum equation
GMANISOTROP	Anisotropic horizontal variation of thickness diffusion is used (*) use with ISOPYCNAL
GMTAPER	Taper GM vector stream function near the sea surface (*) cannot be used with SLIMIT, GMANISOTROP, AFC
GMVAR	Horizontal thickness diffusion is allowed to vary in horizontal
GLS	Generic length scale model of Umlauf and Burchard (2003)
HFLUX	calculates sea surface heat flux using bulk formula
ICE	sea ice is included

Continued on next page

Table 25.12 – continued from previous page

Model option	Description
ICECLIM	reading climatological sea-ice fractional area from file
ISOPYCNAL	uses isopycnal diffusion and Gent-McWilliams' parameterization for eddy induced tracer transport velocity (thickness diffusion)
ISOTAPER	Taper isopycnal diffusion coefficient (ISOPYCNAL)
MELYAM	uses Mellor and Yamada Level 2.5 for mixed layer model
MPDATAADVEC	uses MPDATA for tracer advection
NOHKIM	uses Noh's mixed layer model
NPZD	NPZD process is included (*) use with CARBON
OFFNESTPAR	Used as the lower-resolution part of an off-line one-way nesting calculation
OFFNESTSUB	Used as the higher-resolution part of an off-line one-way nesting calculation
PARALLEL	parallel calculation using MPI. The number of zonally and meridionally partitioned regions should be specified as NPARTX and NPARTY, respectively.
PARENT	executed as low resolution model of the on-line nesting calculation
RUNOFF	uses river runoff data (*) available only for WFLUX
SIDYN	sea-ice dynamics model with EVP rheology (*) available only for ICE
SLIMIT	Tapers thickness diffusion near the sea surface
SLP	Sea surface is elevated/depressed according to surface atmospheric pressure
SMAGHD	uses the Smagorinsky viscosity coefficient multiplied by a constant ratio as the horizontal diffusion coefficient (*) available only for SMAGOR
SMAGOR	uses the Smagorinsky parameterization for horizontal viscosity
SOLARANGLE	solar insolation angle is considered in calculating shortwave penetration
SOMADVEC	uses second order moment advection by Prather (1986)
SPHERICAL	calculates scale factor semi-analytically for spherical coordinates
SUB	executed as a high resolution model of the on-line nesting calculation
TAUBULK	calculates the wind stress using bulk formulae by reading wind speed over the ocean
TDEW	reads dew-point temperature and converts to specific humidity (*) available only for HFLUX
TIDE	Tide producing forcing is activated
TRCBIHARM	uses biharmonic operator for horizontal diffusion (*) Should not be used with ISOPYCNAL
TRIPOLAR	Tripolar system is used to construct model grids of a global model (*) Cannot be used with SPHERICAL
UTZQADVEC	"UTOPIA" and "QUICKEST" scheme can be used for horizontal and vertical tracer advection with ultimate limiter
VISANISO	Anisotropic viscosity coefficients are used (*) use with VIS9P
VIS9P	calculates the viscosity using adjacent 9 grid points
VISBIHARM	uses biharmonic operator for both horizontal viscosity
VMBG3D	reads 3-D vertical viscosity and diffusion coefficients from a file
VVDIMP	calculates the vertical diffusion/viscosity by implicit method

Continued on next page

Table 25.12 – continued from previous page

Model option	Description
	(*) it is automatically loaded if any mixed layer model is used or ISOPYCNAL option is selected
WADJ	adjusts sea surface freshwater flux every time step to keep its global sum to be zero (*) available only for WFLUX
WFLUX	uses the sea surface freshwater flux to force the model
MOVE	used as ocean module for data assimilation (MOVE) system
SCUP	use simple coupler (SCUP) library
SCUPCGCM	used as an ocean module for a coupled model using scup for communication
SCUPNEST	on-line nesting (*) use with SCUP

Acknowledgments

The authors of this manual made new contributions to the source code development of MRI.COM after publication of the previous version in 2017. They would like to express their sincere gratitude to the following persons who were involved in the development of the previous versions on which MRI.COM version 5 has been built: The development of MRI.COM was first proposed by Yoshiteru Kitamura. The early stage of its development was led by Ichiro Ishikawa, whose contribution was indispensable for the existence of MRI.COM. The thermodynamic process of sea ice is based on what developed by Tatsuo Motoi. During the course of the development, significant contributions were also made by Tamaki Yasuda, Atsushi Obata, Seiji Yukimoto, Hiroshi Ishizaki, and Mikitoshi Hirabara.

The authors owe special thanks to Hiroyasu Hasumi of the Atmosphere and Ocean Research Institute (AORI), the University of Tokyo for his valuable suggestions. Application of the EVP rheology to the sea ice model, of σ-coordinate to the free surface formulation, and of high-accuracy advection schemes to the tracer equations would not have been successful without his support. The continuous collaborative relation with the development group of COCO, which is also maintained by Hasumi-san, has been of great help to keep MRI.COM at a high level in international standards.

We would like to emphasize that the feedback from operational system development groups of the Japan Meteorological Agency also contributed significantly to the improvement of the model. In particular, the introduction of various methods for speeding up and thorough debugging could not be done by the authors alone, which is indispensable not only for JMA's operation but also for establishing the foundation of scientific research. Among them, Takafumi Kanehama, Hiroyuki Sugimoto, Masakazu Higaki, Hiroaki Asai, and Yutaro Kubo directly improved the source code. We would like to express my heartfelt gratitude here.

The experiences of coupling with other component models, such as atmospheric models and data assimilation schemes, greatly helped improve this model. The authors are grateful to the developers of those models. Among them, various comments made by Yosuke Fujii during his development of an adjoint code of this model were indispensable for refining the model codes.

Thanks are extended to Shunji Konaga, Tatsushi Tokioka, Masahiro Endoh, Yoshihiro Kimura, Noriya Yoshioka for their numerous contributions to the original models that MRI.COM is based on.

Introduction of state-of-the-art technologies of typesetting using LATEX was contributed by Shigeru Hayashi. Most figures of this manual were drawn by Yukiko Suda.

The people listed above and all other members and former members of the departments involved in the ocean of MRI contributed to the development and refinement of this model. Specifically, earlier versions of this manual (for version 5) were proofread by some members of MRI, which was of great help for the improvement of those earlier versions.

Since the first version of this manual was published in Japanese, some authors had been requested or encouraged to publish an English version on many occasions to introduce this model at international scientific conferences and workshops. This fact strongly motivated us to write the later versions in English. Continuous encouragement and numerous suggestions by Stephen Griffies at Geophysical Fluid Dynamics Lab., NOAA, USA and Gurvan Madec at LODYC, Institute Pierre Simon Laplace, France, during panel meetings of the working group for ocean model development (WGOMD), now the ocean model developing panel (OMDP), of CLIVAR are gratefully acknowledged.

Bibliography

Adcroft, A., and J.-M. Campin, 2004: Rescaled height coordinates for accurate representation of free-surface flows in ocean circulation models. *Ocean Modell.*, **7**, 269–284, doi:10.1016/j.ocemod.2003.09.003.

Arakawa, A., 1972: Design of the ucla general circulation model. Tech. Rep. 7, Dep. of Meteorology, University of California, Los Angeles, 116 pp. https://ntrs.nasa.gov/search.jsp?R=19730012781.

Barker, H. W., and Z. Li, 1995: Improved simulation of clear-sky shortwave radiative transfer in the ccc-gcm. *J. Clim.*, **8**, 2213–2223, doi:10.1175/1520-0442(1995)008<2213:ISOCSS>2.0.CO;2.

Bentzen, M., G. Evensen, H. Drange, and A. D. Jenkins, 1999: Coordinate transformation on a sphere using conformal mapping. *Mon. Wea. Rev.*, **127**, 2733–2740, doi:10.1175/1520-0493(1999)127<2733:CTOASU>2.0.CO;2.

Bleck, R., and D. B. Boudra, 1986: Wind-driven spin up in eddy-resolving ocean models formulated in isopycnic and isobaric coordinates. *J. Geophys. Res.*, **91**, 7611–7621, doi:10.1029/JC091iC06p07611.

Bleck, R., G. Halliwell, A. Wallcraft, S. Carroll, K. Kelly, and K. Rushing, 2002: *Hybrid Coordinate Ocean Model (HYCOM) User's Manual*. http://hycom.rsmas.miami.edu/hycom-model/documentation.html.

Briegleb, B. P., P. Minnis, V. Ramanathan, and E. Harrison, 1986: Comparison of regional clear-sky albedos inferred from satellite observations and model computations. *J. Climate Appl. Meteor.*, **25**, 214–226, doi:10.1175/1520-0450(1986)025<0214:CORCSA>2.0.CO;2.

Bryan, K., 1969: A numerical method for the study of the circulation of the world ocean. *J. Comput. Phys.*, **4**, 347–376, doi:10.1016/0021-9991(69)90004-7.

Bryan, K., 1984: Accelerating the convergence to equilibrium of ocean-climate models. *J. Phys. Oceanogr.*, **14**, 666–673, doi:10.1175/1520-0485(1984)014<0666:ATCTEO>2.0.CO;2.

Burchard, H., 2002: Energy-conserving discretization of turbulent shear and buoyancy production. *Ocean Modell.*, **4**, 347–361, doi:10.1016/S1463-5003(02)00009-4.

Burchard, H., and K. Bolding, 2001: Comparative analysis of four second-moment turbulence closure models for the oceanic mixed layer. *J. Phys. Oceanogr.*, **31**, 1943–1968, doi:10.1175/1520-0485(2001)031<1943:CAOFSM>2.0.CO;2.

Cailleau, S., V. Fedorenko, B. Barnier, E. Blayo, and L. Debreu, 2008: Comparison of different numerical methods used to handle the open boundary of a regional ocean circulation model of the bay of biscay. *Ocean Modell.*, **25**, 1–16, doi:10.1016/j.ocemod.2008.05.009.

Campin, J. M., J. Marshall, and D. Ferreira, 2008: Sea ice-ocean coupling using a rescaled vertical coordinate z^*. *Ocean Modell.*, **24**, 1–14, doi:10.1016/j.ocemod.2008.05.005.

Canuto, A., A. Howard, Y. Cheng, and M. S. Dubovikov, 2001: Ocean turbulence. part i: One-point closure model momentum and heat vertical diffusivities. *J. Phys. Oceanogr.*, **31**, 1413–1426, doi:10.1175/1520-0485(2001)031<1413:OTPIOP>2.0.CO;2.

Chen, C., and Coauthors, 2011: *An unstructured grid, finite-volume community ocean model FVCOM user manual*. http://fvcom.smast.umassd.edu/wp-content/uploads/2013/11/MITSG_12-25.pdf.

Colella, P., and P. R. Woodward, 1984: The piecewise parabolic method (ppm) for gas-dynamical simulations. *J. Comp. Phys.*, **54**, 174–201, doi:10.1016/0021-9991(84)90143-8.

Cox, M. D., 1987: Isopycnal diffusion in a z-coordinate ocean model, (Unpublished manuscript).

Danabasoglu, G., R. Ferrari, and J. C. McWilliams, 2008: Sensitivity of an ocean general circulation model to a parameterization of near-surface eddy fluxes. *J. Clim.*, 1192–1208, doi:10.1175/2007JCLI1508.1.

Danabasoglu, G., and J. Marshall, 2007: Effects of vertical variations of thickness diffusivity in an ocean general circulation model. *Ocean Modell.*, **18**, 122–141, doi:10.1016/j.ocemod.2007.03.006.

Danabasoglu, G., and J. C. McWilliams, 1995: Sensitivity of the global ocean circulation to parameterizations of mesoscale tracer transports. *J. Clim.*, **8**, 2967–2987, doi:10.1175/1520-0442(1995)008<2967:SOTGOC>2.0.CO;2.

Danilov, S., 2013: Ocean modeling on unstructured meshes. *Ocean Modell.*, **69**, 195–210, doi:10.1016/j.ocemod.2013.05.005.

Debreu, L., P. Marchesiello, P. Penven, and G. Cambon, 2012: Two-way nesting in split-explicit ocean models: Algorithms, implementation and validation. *Ocean Modell.*, **49–50**, 1–21, doi:10.1016/j.ocemod.2012.03.003.

Delhez, E. J. M., and E. Deleersnijder, 2007: Overshootings and spurious oscillations caused by biharmonic mixing. *Ocean Modell.*, **17**, 183–198, doi:10.1016/j.ocemod.2007.01.002.

Dugdale, R. C., 1967: Nutrient limitation in the sea: dynamics, identification, and significance. *Limnol. Oceanogr.*, **12**, 685–695, doi:10.4319/lo.1967.12.4.0685.

Durran, D. R., 2010: *Numerical methods for fluid dynamics: with applications to geophysics*. Springer-Verlag, 532 pp., doi:10.1007/978-1-4419-6412-0.

ECMWF, 2016a: *IFS Documentation CY43R1 - Part III: Dynamics and Numerical Procedures*. No. 3, IFS Documentation, ECMWF, doi:10.21957/m1u2yxwrl, URL https://www.ecmwf.int/node/17116.

ECMWF, 2016b: *IFS Documentation CY43R1 - Part IV: Physical Processes*. No. 4, IFS Documentation, ECMWF, doi:10.21957/sqvo5yxja, URL https://www.ecmwf.int/node/17117.

Eden, C., 2007: Eddy length scales in the north atlantic. *J. Geophys. Res.*, doi:10.1029/2006JC003901.

Eden, C., and R. J. Greatbatch, 2008: Towards a mesoscale eddy closure. *Ocean Modell.*, 223–239, doi:10.1016/j.ocemod.2007.09.002.

Eden, C., M. Jochum, and G. Danabasoglu, 2009: Effects of different closures for thickness diffusivity. *Ocean Modell.*, 47–59, doi:10.1016/j.ocemod.2008.08.004.

England, M. H., 1995: The age of water and ventilation timescales in a global ocean model. *J. Phys. Oceanogr.*, 2756–2777, doi:10.1175/1520-0485(1995)025<2756:TAOWAV>2.0.CO;2.

Eppley, R., 1972: Temperature and phytoplankton growth in the sea. *Fish. Bull.*, **70**, 1063–1085, http://fishbull.noaa.gov/70-4/eppley.pdf.

Ferrari, R., J. C. McWilliams, V. M. Canuto, and M. Dubovikov, 2008: Parameterization of eddy fluxes near oceanic boundaries. *J. Clim.*, 2770–2789, doi:10.1175/2007JCLI1510.1.

Ferreira, D., and J. Marshall, 2006: Formulation and implementation of a "residual-mean" ocean circulation model. *Ocean Modell.*, **13**, 86–107, doi:10.1016/j.ocemod.2005.12.001.

Ferreira, D., J. Marshall, and P. Heimbach, 2005: Estimating eddy stresses by fitting dynamics to observations using a residual-mean ocean circulation model and its adjoint. *J. Phys. Oceanogr.*, **35**, 1891–1910, doi:10.1175/JPO2785.1.

Gent, P. R., and J. C. McWilliams, 1990: Isopycnal mixing in ocean circulation models. *J. Phys. Oceanogr.*, 150–155, doi:10.1175/1520-0485(1990)020<0150:IMIOCM>2.0.CO;2.

Gent, P. R., J. Willebrand, T. J. McDougall, and J. C. McWilliams, 1995: Parameterizing eddy-induced tracer transports in ocean circulation models. *J. Phys. Oceanogr.*, **25**, 463–474, doi:10.1175/1520-0485(1995)025<0463:PEITTI>2.0.CO;2.

Gill, A. E., 1982: *Atmosphere-Ocean Dynamics*. Academic Press, 662 pp., doi:10.1002/qj.49711046322.

Griffies, S. M., 1998: The gent-mcwilliams skew flux. *J. Phys. Oceanogr.*, **28**, 831–841, doi:10.1175/1520-0485(1998)028<0831:TGMSF>2.0.CO;2.

Griffies, S. M., 2004: *Fundamentals of ocean climate models*. Princeton University Press, 518 pp., http://press.princeton.edu/titles/7797.html.

Griffies, S. M., 2012: Elements of the modular ocean model (mom) (2012 release with updates). GFDL Ocean Group Technical Report 7, NOAA/Geophysical Fluid Dynamics Laboratory, 632 + xiii pages pp. http://mom-ocean.science/web/docs/project/MOM5_manual.pdf.

Griffies, S. M., and A. J. Adcroft, 2008: Formulating the equations of ocean models. *Ocean modeling in an eddying regime*, M. Hecht and H. Hasumi, 281–317, doi:10.1029/177GM18.

Griffies, S. M., A. Gnanadesikan, R. C. Pacanowski, V. D. Larichev, J. K. Dukowicz, and R. D. Smith, 1998: Isoneutral diffusion in a z-coordinate ocean model. *J. Phys. Oceanogr.*, 805–830, doi:10.1175/1520-0485(1998)028<0805:IDIAZC>2.0.CO;2.

Griffies, S. M., and R. Hallberg, 2000: Biharmonic friction with a smagorinsky-like viscosity for use in large-scale eddy-permitting ocean models. *Mon. Wea. Rev.*, **128**, 2935–2946, doi:10.1175/1520-0493(2000)128<2935:BFWASL>2.0.CO;2.

Griffies, S. M., M. Levy, A. J. Adcroft, G. Danabasoglu, R. W. Hallberg, D. Jacobsen, W. Large, and T. Ringler, 2015: Theory and numerics of the community ocean vertical mixing (cvmix) project. Technical report, CVMix project, 98 pp. https://github.com/CVMix/CVMix-description/blob/master/cvmix.pdf.

Hasumi, H., 2006: Ccsr ocean component model (coco) version 4.0. Tech. Rep. 25, CCSR, 103 pp. http://ccsr.aori.u-tokyo.ac.jp/~hasumi/COCO/coco4.pdf.

Hibler, I., W. D., 1979: A dynamic thermodynamic sea ice model. *J. Phys. Oceanogr.*, **9**, 815–846, doi:10.1175/1520-0485(1979)009<0815:ADTSIM>2.0.CO;2.

Hirabara, M., H. Ishizaki, G. Yamanaka, H. Tsujino, and I. Ishikawa, 2010: Additional flux arising from unresolved scales in eddying ocean models. *J. Oceanogr.*, **66**, 633–647, doi:10.1007/s10872-010-0052-6.

Holland, W. R., J. C. Chow, and F. O. Bryan, 1998: Application of a third-order upwind scheme in the NCAR ocean model. *J. Clim.*, **11**, 1487–1493, doi:10.1175/1520-0442(1998)011<1487:AOATOU>2.0.CO;2.

Hunke, E. C., and J. K. Dukowicz, 1997: An elastic-viscous-plastic model for sea ice dynamics. *J. Phys. Oceanogr.*, **94**, 1849–1867, doi:10.1175/1520-0485(1997)027<1849:AEVPMF>2.0.CO;2.

Hunke, E. C., and J. K. Dukowicz, 2002: The elastic-viscous-plastic sea ice dynamics model in general orthogonal curvilinear coordinates on a sphere – incorporation of metric terms. *Mon. Wea. Rev.*, **130**, 1848–1865, doi:10.1175/1520-0493(2002)130<1848:TEVPSI>2.0.CO;2.

Hunke, E. C., and W. H. Lipscomb, 2006: *CICE: the Los Alamos Sea Ice Model Documentation and Software User's Manual*. http://oceans11.lanl.gov/trac/CICE.

IOC, SCOR, and IAPSO, 2010: The international thermodynamic equation of seawater - 2010: Calculation and use of thermodynamic properties. Manuals and Guides 56, Intergovernmental Oceanographic Commission, 196 pp. http://www.teos-10.org/pubs/TEOS-10_Manual.pdf.

Ishikawa, I., H. Tsujino, M. Hirabara, H. Nakano, T. Yasuda, and H. Ishizaki, 2005: Meteorological research institute community ocean model (mri.com) manual. Technical Reports of the Meteorological Research Institute 47, the Meteorological Research Institute, 189 pp. doi:10.11483/mritechrepo.47.

Ishizaki, H., 1994: A simulation of the abyssal circulation in the north pacific ocean. part ii: Theoretical rationale. *J. Phys. Oceanogr.*, **24**, 1941–1954, doi:10.1175/1520-0485(1994)024<1941:ASOTAC>2.0.CO;2.

Ishizaki, H., 1997: A massive treatment scheme of complete convection for ocean models, unpublished manuscript.

Ishizaki, H., and I. Ishikawa, 2006: High parallelization efficiency in barotropic-mode computation of ocean models based on multi-grid boundary ghost area. *Ocean Modell.*, **13**, 238–254, doi:10.1016/j.ocemod.2006.02.001.

Ishizaki, H., and T. Motoi, 1999: Reevaluation of the takano-oonishi scheme for momentum advection on bottom relief in ocean models. *J. Atmos. Oceanic Technol.*, **16**, 1994–2010, doi:10.1175/1520-0426(1999)016<1994:ROTTOS>2.0.CO;2.

Ishizaki, H., and G. Yamanaka, 2010: Impact of explicit sun altitude in solar radiation on an ocean model simulation. *Ocean Modell.*, 52–69, doi:10.1016/j.ocemod.2009.12.002.

Jerlov, N. G., 1976: *Marine Optics*. Elsevier, 231 pp.

Jiang, L., and R. W. Garwood, Jr., 1996: Three-dimensional simulations of overflows on continental slopes. *J. Phys. Oceanogr.*, **26**, 1214–1233, doi:10.1175/1520-0485(1996)026<1214:TDSOOO>2.0.CO;2.

Kantha, L. H., and C. A. Clayson, 2000: *Small Scale Processes in Geophysical Fluid Flows*. Academic Press, 888 pp., doi:10.1063/1.1420560.

Killworth, P. D., D. Stainforth, D. J. Webb, and S. M. Paterson, 1991: The development of a free-surface bryan-cox-semtner ocean model. *J. Phys. Oceanogr.*, **31**, 1333–1348, doi:10.1175/1520-0485(1991)021<1333:TDOAFS>2.0.CO;2.

Kishi, M. J., H. Motono, M. Kashiwai, and A. Tsuda, 2001: An ecological-physical coupled model with ontogenetic vertical migration of zooplankton in the northwestern pacific. *J. Oceanogr.*, **57**, 499–507, doi:10.1023/A:1021517129545.

Large, W. G., G. Danabasoglu, S. C. Doney, and J. C. McWilliams, 1997: Sensitivity to surface forcing and boundary layer mixing in a global ocean model: annual-mean climatology. *J. Phys. Oceanogr.*, 2418–2447, doi:10.1175/1520-0485(1997)027<2418:STSFAB>2.0.CO;2.

Large, W. G., J. C. McWilliams, and S. C. Doney, 1994: Oceanic vertical mixing: a review and a model with a nonlocal boundary layer parameterization. *Rev. Geophys.*, **32**, 363–403, doi:10.1029/94RG01872.

Large, W. G., and S. Yeager, 2004: Diurnal to decadal global forcing for ocean and sea-ice models:the data sets and flux climatologies. Technical Note NCAR/TN-460+STR., CGD Division of the National Center for Atmospheric Research. doi:10.5065/D6KK98Q6.

Large, W. G., and S. Yeager, 2009: The global climatology of an interannually varying air-sea flux data set. *Clim. Dyn.*, **33**, 341–364, doi:10.1007/s00382-008-0441-3.

Leclair, M., and G. Madec, 2009: A conservative leapfrog time stepping method. *Ocean Modell.*, **30**, 88–94, doi:10.1016/j.ocemod.2009.06.006.

Lemariè, F., L. Debreu, G. Madec, J. Demange, J. M. Molines, and H. M., 2015: Stability constraints for oceanic numerical models: implications for the formulation of time and space discretizations. *Ocean Modell.*, **92**, 124–148, doi:10.1016/j.ocemod.2015.06.006.

Leonard, B. P., 1979: A stable and accurate convective modeling procedure based upon quadratic upstream interpolation. *J. Comput. Methods Appl. Mech. Eng.*, **19**, 59–98, doi:10.1016/0045-7825(79)90034-3.

Leonard, B. P., M. K. MacVean, and A. P. Lock, 1993: Positivity-preserving numerical schemes for multidimensional advection. Tech. Memo 106055, ICOMP-93-05, NASA, 62 pp. https://ntrs.nasa.gov/search.jsp?R=19930017902.

Leonard, B. P., M. K. MacVean, and A. P. Lock, 1994: The flux integral method for multidimensional convection and diffusion. NASA Tech. Memo. 106679, ICOMP-94-13, NASA, 27 pp. doi:10.1016/0307-904X(95)00017-E.

Lepparanta, M., 2011: *The Drift of Sea Ice*. Springer-Verlag, 347 pp., doi:10.1007/978-3-642-04683-4.

Lin, S.-J., W. C. Chao, Y. C. Sud, and G. K. Walter, 1994: A class of the van Leer-type transport schemes and its application to the moisture transport in a general circulation model. *Mon. Wea. Rev.*, **122**, 1575–1593, doi:10.1175/1520-0493(1994)122<1575:ACOTVL>2.0.CO;2.

Madec, G., and the NEMO team, 2008: Nemo ocean engine. Note du Pôle de modélisation 27, Institut Pierre-Simon Laplace (IPSL), France. http://www.nemo-ocean.eu/About-NEMO/Reference-manuals/NEMO_book_3.6_STABLE.

Marshall, J., C. Hill, L. Perelman, and A. Adcroft, 1997: Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling. *J. Geophys. Res.*, **102**, 5733–5752, doi:10.1029/96JC02776.

Martin, J. H., G. A. Knauer, D. M. Karl, and W. W. Broenkow, 1987: Vertex: Carbon, cycling in the northeast pacific. *Deep-Sea Res.*, **34**, 267–285, doi:10.1016/0198-0149(87)90086-0.

Matsumoto, K., T. Takanezawa, and N. Ooe, 2000: Ocean tide models developed by assimilating topex/poseidon altimeter data into hydrodynamical model: A global model and a regional model around japan. *J. Oceanogr.*, **56**, 567–581, doi:10.1023/A:1011157212596.

McDougall, T., D. Jackett, D. Wright, and R. Feistel, 2003: Accurate and computationally efficient algorithms for potential temperature and density of sea water. *J. Atmos. Oceanic Technol.*, **20**, 730–741, doi:10.1175/1520-0426(2003)20<730:AACEAF>2.0.CO;2.

McWilliams, J. C., 2016: Submesoscale currents in the ocean. *Proc. R. Soc. A.*, **472**, doi:10.1098/rspa.2016.0117.

Mellor, G. L., 2004: *Users guide for a three-dimensional, primitive equation, numerical ocean model*. Prog. in Atmos. and Ocean. Sci, Princeton University, 53 pp.

Mellor, G. L., and A. Blumberg, 2004: Wave breaking and ocean surface layer thermal response. *J. Phys. Oceanogr.*, 693–698, doi:10.1175/2517.1.

Mellor, G. L., and L. Kantha, 1989: An ice-ocean coupled model. *J. Geophys. Res.*, **94**, 10 937–10 954, doi:10.1029/JC094iC08p10937.

Mellor, G. L., and T. Yamada, 1982: Development of a turbulence closure model for geophysical fluid problems. *Rev. Geophys. Space Phys.*, **20**, 851–875, doi:10.1029/RG020i004p00851.

Merryfield, W. J., and G. Holloway, 2003: Application of an accurate advection algorithm to sea-ice modelling. *Ocean Modell.*, **5**, 1–15, doi:10.1016/S1463-5003(02)00011-2.

Mesinger, F., and A. Arakawa, 1976: Numerical methods used in atmospheric models. 65 pp.

Morales Maqueda, M. A., and G. Holloway, 2006: Second-order moment advection scheme applied to arctic ocean simulation. *Ocean Modell.*, **14**, 197–221, doi:10.1016/j.ocemod.2006.05.003.

Morel, A., and D. Antoine, 1994: Heating rate within the upper ocean in relation to its bio-optical state. *J. Phys. Oceanogr.*, **24**, 1652–1665, doi:10.1175/1520-0485(1994)024<1652:HRWTUO>2.0.CO;2.

Nakano, H., and N. Suginohara, 2002: Effects of Bottom Boundary Layer parameterization on reproducing deep and bottom waters in a world ocean model. *J. Phys. Oceanogr.*, **32**, 1209–1227, doi:10.1175/1520-0485(2002)032<1209:EOBBLP>2.0.CO;2.

Noh, Y., Y. J. Kang, T. Matsuura, and S. Iizuka, 2005: Effect of the prandtl number in the parameterization of vertical mixing in an ogcm of the tropical pacific. *Geophys. Res. Lett.*, L23609, doi:10.1029/2005GL024540.

Noh, Y., and H. J. Kim, 1999: Simulations of temperature and turbulence structure of the oceanic boundary layer with the improved near-surface process. *J. Geophys. Res.*, **104**, 15 621–15 634, doi:10.1029/1999JC900068.

Notz, D., A. Jahn, M. Holland, E. Hunke, F. Massonnet, J. Stroeve, B. Tremblay, and M. Vancoppenolle, 2016: Sea ice model intercomparison project (simip): Understanding sea ice through climate-model simulations. *Geosci. Model Dev.*, **9**, 3427–3446, doi:10.5194/gmd-9-3427-2016.

Obata, A., and Y. Kitamura, 2003: Interannual variability of the sea-air exchange of co$_2$ from 1961 to 1998 simulated with a global ocean circulation-biogeochemistry model. *J. Geophys. Res.*, **108**, 3377, doi:10.1029/2001JC001088.

Orr, J., R. Najjar, C. Sabine, and F. Joos, 1999: Abiotic-howto. Internal OCMIP Report LSCE/CEA Saclay, OCMIP, Gif-sur-Yvette, France, 25 pp. http://ocmip5.ipsl.jussieu.fr/OCMIP/phase2/simulations/Abiotic/HOWTO-Abiotic.html.

Orr, J., and Coauthors, 2017: Biogeochemical protocols and diagnostics for the cmip6 ocean model intercomparison project (omip). *Geosci. Model Dev.*, 2169–2199, doi:10.5194/gmd-10-2169-2017.

Orr, J. C., and J. M. Epitalon, 2015: Improved routines to model the ocean carbonate system:mocsy 2.0. *Geosci. Model Dev.*, **8**, 485–499, doi:10.5194/gmd-8-485-2015.

Oschlies, A., 2001: Model-derived estimates of new production: New results point toward lower values. *Deep-Sea Res. II*, **48**, 2173–2197, doi:10.1016/S0967-0645(00)00184-3.

Paulson, C. A., and J. J. Simpson, 1977: Irradiance measurements in the upper ocean. *J. Phys. Oceanogr.*, **7**, 952–956, doi:10.1175/1520-0485(1977)007<0952:IMITUO>2.0.CO;2.

Phillips, N., 1966: The equation of motion for a shallow rotating atmosphere and the "traditional approximation". *J. Atmos. Sci.*, **23**, 626–628, doi:10.1175/1520-0469(1966)023<0626:TEOMFA>2.0.CO;2.

Prather, M. J., 1986: Numerical advection by conservation of second-order moments. *J. Geophys. Res.*, **91**, 6671–6681, doi:10.1029/JD091iD06p06671.

Redfield, A. C., B. H. Ketchum, and F. A. Richards, 1963: The influence of organisms on the composition of sea water. *The sea*, M. N. Hill, Ed., Vol. vol. 2, Wiley-Intersci. New York., 26–77.

Redi, M. H., 1982: Oceanic isopycnal mixing by coordinate rotation. *J. Phys. Oceanogr.*, 1154–1158, doi:10.1175/1520-0485(1982)012<1154:OIMBCR>2.0.CO;2.

Rotta, J. C., 1951a: Statistische theorie nichthomogener turbulenz. *Z. Phys.*, **129**, 547–572, doi:10.1007/BF01330059.

Rotta, J. C., 1951b: Statistische theorie nichthomogener turbulenz. *Z. Phys.*, **131**, 51–77, doi:10.1007/BF01329645.

Sakamoto, K., H. Tsujino, H. Nakano, M. Hirabara, and G. Yamanaka, 2013: A practical scheme to introduce explicit tidal forcing into an ogcm. *Ocean Sci.*, **9**, 1089–1108, doi:10.5194/os-9-1089-2013.

Sakamoto, K., H. Tsujino, H. Nakano, S. Urakawa, and G. Yamanaka, 2018: Development management of meteorological research institute community ocean model (mri.com) using git and redmine. *Oceanography in Japan*, **27**, 175–188, doi:10.5928/kaiyou.27.5_175, (in Japanese).

Saunders, P., 1990: The international temperature scale of 1990, its-90. *WOCE Neweletter*, **10**, 10, https://www.nodc.noaa.gov/woce/wdiu/wocedocs/newsltr/news10/news10.pdf.

Schiller, A., 2004: Effects of explicit tidal forcing in an ogcm on the water-mass structure and circulation in the indonesian throughflow region. *Ocean Modell.*, **6**, 31–49, doi:10.1016/S1463-5003(02)00057-4.

Schmittner, A., A. Oschlies, H. Matthews, and E. Galbraith, 2008: Future changes in climate, ocean circulation, ecosystems, and biogeochemical cycling simulated for a business-as-usual co$_2$ emission scenario until year 4000 ad. *Global Biogeochem. Cycles*, **22**, GB1013, doi:10.1029/2007GB002953.

Schwiderski, E. W., 1980: On charting global ocean tides. *Rev. Geophys. Space Phys.*, 243–268, doi:10.1029/RG018i001p00243.

Shchepetkin, A. F., and J. C. McWilliams, 2003: A method for computing horizontal pressure-gradient force in an oceanic model with a non-aligned vertical coordinate. *J. Geophys. Res.*, **108**, 3090–3124, doi:10.1029/2001JC001047.

Shchepetkin, A. F., and J. C. McWilliams, 2005: The regional ocean modeling system: A split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modell.*, **9**, 347–404, doi:10.1016/j.ocemod.2004.08.002.

Shchepetkin, A. F., and J. C. McWilliams, 2009: Computational kernel algorithms for fine-scale, multi-process, long-term oceanic simulations. *Handbook of Numerical Analysis, Computational Methods for the Ocean and the Atmosphere*, J. T. P.G. Ciarlet, R. Temam, Ed., Vol. XIV, Elsevier Science, 119–182, doi:10.1016/S1570-8659(08)01202-0.

Skamarock, W. C., 2006: Positive-definite and monotonic limiters for unrestricted-time-step transport schemes. *Mon. Wea. Rev.*, **134**, 2241–2250, doi:10.1175/MWR3170.1.

Smagorinsky, J., 1963: General circulation experiments with the primitive equations: I. the basic experiment. *Mon. Wea. Rev.*, **91**, 99–164, doi:10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2.

Smith, L., Y. Yamanaka, M. Pahlow, and A. Oschlies, 2009: Optimal uptake kinetics: physiological acclimation explains the pattern of nitrate uptake by phytoplankton in the ocean. *Mar., Ecol. Prog. Ser.*, **384**, 1–12, doi:10.3354/meps08022.

Smith, R. D., and P. R. Gent, 2004: Anisotropic gent-mcwilliams parameterization for ocean models. *J. Phys. Oceanogr.*, 2541–2564, doi:10.1175/JPO2613.1.

Smith, R. D., and J. C. McWilliams, 2003: Anisotropic horizontal viscosity for ocean models. *Ocean Modell.*, **5**, 129–156, doi:10.1016/S1463-5003(02)00016-1.

Smolarkiewicz, P. K., 1984: A fully multidimensional positive definite advection transport algorism with small implicit diffusion. *J. Comput. Phys.*, **54**, 325–362, doi:10.1016/0021-9991(84)90121-9.

Smolarkiewicz, P. K., and W. W. Grabowski, 1990: The multidimensional positive definite advection transport algorithm: nonoscillatory option. *J. Comp. Phys.*, **86**, 335–375, doi:10.1016/0021-9991(90)90105-A.

Smolarkiewicz, P. K., and L. G. Margolin, 1998: Mpdata: A finite-difference solver for geophysical flows. *J. Comp. Phys.*, **140**, 459–480, doi:10.1006/jcph.1998.5901.

St. Laurent, L. C., H. L. Simmons, and S. R. Jayne, 2002: Estimating tidally driven mixing in the deep ocean. *Geophys. Res. Lett.*, **29**, 2106–2110, doi:10.1029/2002GL015633.

Steele, M., 1992: Sea ice melting and floe geometry in a simple ice-ocean model. *J. Geophys. Res.*, **97**, 17,729–17,738, doi:10.1029/92JC01755.

Sueyoshi, M., and T. Yasuda, 2009: Reproducibility and future projection of the ocean first baroclinic rossby radius based on the cmip3 multi-model dataset. *J. Meteorol. Soc. Japan*, 821–827, doi:10.2151/jmsj.87.821.

Suginohara, N., and S. Aoki, 1991: Buoyancy-driven circulation as horizontal convection on β-plane. *J. Marine Res.*, **49**, 295–320, doi:10.1357/002224091784995891.

Taylor, J. P., J. M. Edwards, M. D. Glew, P. Hignett, and A. Slingo, 1996: Studies with a flexible new radiation code. II: Comparisons with aircraft short-wave observations. *Q. J. R. Meteorol. Soc.*, **122**, 839–861, doi:10.1256/smsqj.53203.

Theiss, J., 2004: Equatorward energy cascade, critical latitude, and the predominance of cyclonic vortices in geostrophic turbulence. *J. Phys. Oceanogr.*, **34**, 1663–1678, doi:10.1175/1520-0485(2004)034<1663:EECCLA>2.0.CO;2.

Thorndike, A. S., D. A. Rothrock, M. G. A., and R. Colony, 1975: The thickness distribution of sea ice. *J. Geophys. Res.*, **80**, 4,501–4,513, doi:10.1029/JC080i033p04501.

Tsujino, H., T. Motoi, I. Ishikawa, M. Hirabara, H. Nakano, G. Yamanaka, T. Yasuda, and H. Ishizaki, 2010: Reference manual for the meteorological research institute community ocean model(mri.com) version 3. Technical Reports of the Meteorological Research Institute 59, Meteorological Research Institute, 241 pp. doi:10.11483/mritechrepo.59.

Tsujino, H., H. Nakano, K. Sakamoto, S. Urakawa, M. Hirabara, H. Ishizaki, and G. Yamanaka, 2017: Reference manual for the meteorological research institute community ocean model version 4 (mri.comv4). Technical Reports of the Meteorological Research Institute 80, Meteorological Research Institute, 284 pp. doi:10.11483/mritechrepo.80.

Umlauf, L., and H. Burchard, 2003: A generic length-scale equation for geophysical turbulence models. *J. Marine Res.*, **61**, 235–265, doi:10.1357/002224003322005087.

UNESCO, 1981: Tenth report of the Joint Panel on Oceanographic Tables and Standards, Sidney, B. C., September 1980. Unesco Technical papers in marine science 36, UNESCO, 25 pp. http://unesdoc.unesco.org/images/0004/000461/046148eb.pdf.

UNESCO, 1983: Algorithms for computation of fundamental properties of seawater. Unesco Technical papers in marine science 44, UNESCO, 53 pp. http://unesdoc.unesco.org/images/0005/000598/059832eb.pdf.

Visbeck, M., J. Marshall, T. Haine, and M. Spall, 1997: Specification of eddy transfer coefficients in coarse-resolution ocean circulation models. *J. Phys. Oceanogr.*, **27**, 381–402, doi:10.1175/1520-0485(1997)027<0381:SOETCI>2.0.CO;2.

von Storch, H., and F. Zwiers, 2001: *Statistical Analysis in climate research*. Cambridge university press, 484 pp., doi:10.1017/CBO9780511612336.

Wanninkhof, R., 2014: Relationship between wind speed and gas exchange over the ocean revisited. *Limnol. oceanogr. Methods*, 351–362, doi:10.4319/lom.2014.12.351.

Weatherly, G. L., 1972: A study of the bottom boundary layer of the florida current. *J. Phys. Oceanogr.*, **2**, 54–72.

Weatherly, G. L., S. L. Blumsack, and A. A. Bird, 1980: On the effect of diurnal tidal currents in determining the thickness of the turbulent ekman bottom boundary layer. *J. Phys. Oceanogr.*, **10**, 297–300, doi:10.1175/1520-0485(1980)010<0297:OTEODT>2.0.CO;2.

Webb, D. J., 1995: The vertical advection of momentum in bryan-cox-semtner ocean general circulation models. *J. Phys. Oceanogr.*, 3186–3195, doi:10.1175/1520-0485(1995)025<3186:TVAOMI>2.0.CO;2.

Weiss, R. F., 1970: The solubility of nitrogen, oxygen and argon in water and seawater. *Deep-Sea Res.*, **17**, 721–735, doi:10.1016/0011-7471(70)90037-9.

WGASF, 2000: Intercomparison and validation of ocean-atmosphere energy flux fields. Final report of the joint wcrp/scor working group on air-sea fluxes (wgasf), WGASF, 306 pp. http://eprints.soton.ac.uk/69522/.

Winton, M., R. Hallberg, and A. Gnanadesikan, 1998: Simulation of density-driven frictional downslope flow in z-coordinate ocean models. *J. Phys. Oceanogr.*, **28**, 2163–2174, doi:10.1175/1520-0485(1998)028<2163:SODDFD>2.0.CO;2.

Yamanaka, Y., R. Furue, H. Hasumi, and N. Suginohara, 2000: Comparison of two classical advection schemes in a general circulation model. *J. Phys. Oceanogr.*, **30**, 2439–2451, doi:10.1175/1520-0485(2000)030<2439:COTCAS>2.0.CO;2.

Yamanaka, Y., and E. Tajika, 1996: The role of the vertical fluxes of particulate organic matter and calcite in the oceanic carbon cycle: Studies using an ocean biogeochemical general circulation model. *Global Biogeochem. Cycles*, **10**, 361–382, doi:10.1029/96GB00634.

Yoshimura, H., and S. Yukimoto, 2008: Development of simple coupler (scup) for earth system modeling. *Pap. Meteor. Geophys.*, **59**, 19–29, doi:10.2467/mripapers.59.19.

Yukimoto, S., and Coauthors, 2011: Meteorological research institute-earth system model v1 (mri-esm1) - model description -. Technical Reports of the Meteorological Research Institute 64, Meteorological Research Institute, 88 pp. doi:10.11483/mritechrepo.64.

Yukimoto, S., and Coauthors, 2019: The Meteorological Research Institute Earth System Model Version 2.0, MRI-ESM2.0: Description and Basic Evaluation of the Physical Component. *Journal of the Meteorological Society of Japan. Ser. II*, **97** (**5**), 931–965, doi:10.2151/jmsj.2019-051.

Index

気象研究所技術報告一覧表

第1号 バックグラウンド大気汚染の測定法の開発（地球規模大気汚染特別研究班，1978）
Development of Monitoring Techniques for Global Background Air Pollution. (MRI Special Research Group on Global Atmospheric Pollution, 1978)

第2号 主要活火山の地殻変動並びに地熱状態の調査研究（地震火山研究部，1979）
Investigation of Ground Movement and Geothermal State of Main Active Volcanoes in Japan. (Seismology and Volcanology Research Division, 1979)

第3号 筑波研究学園都市に新設された気象観測用鉄塔施設（花房龍男，藤谷徳之助，伴野登，魚津博，1979）
On the Meteorological Tower and Its Observational System at Tsukuba Science City. (T. Hanafusa, T. Fujitani, N. Banno, and H. Uozu, 1979)

第4号 海底地震常時観測システムの開発（地震火山研究部，1980）
Permanent Ocean－Bottom Seismograph Observation System. (Seismology and Volcanology Research Division, 1980）

第5号 本州南方海域水温図－400m（又は500m）深と1,000m深－（1934－1943年及び1954－1980年）（海洋研究部，1981）
Horizontal Distribution of Temperature in 400m (or 500m) and 1,000m Depth in Sea South of Honshu, Japan and Western－North Pacific Ocean from 1934 to 1943 and from 1954 to 1980. (Oceanographical Research Division, 1981)

第6号 成層圏オゾンの破壊につながる大気成分及び紫外日射の観測（高層物理研究部，1982）
Observations of the Atmospheric Constituents Related to the Stratospheric ozon Depletion and the Ultraviolet Radiation. (Upper Atmosphere Physics Research Division, 1982)

第7号 83型強震計の開発（地震火山研究部，1983）
Strong－Motion Seismograph Model 83 for the Japan Meteorological Agency Network. (Seismology and Volcanology Research Division, 1983)

第8号 大気中における雪片の融解現象に関する研究（物理気象研究部，1984）
The Study of Melting of Snowflakes in the Atmosphere. (Physical Meteorology Research Division, 1984)

第9号 御前崎南方沖における海底水圧観測（地震火山研究部・海洋研究部，1984）
Bottom Pressure Observation South off Omaezaki, Central Honsyu. (Seismology and Volcanology Research Division and Oceanographical Research Division, 1984)

第10号 日本付近の低気圧の統計（予報研究部，1984）
Statistics on Cyclones around Japan. (Forecast Research Division, 1984)

第11号 局地風と大気汚染質の輸送に関する研究（応用気象研究部, 1984）
Observations and Numerical Experiments on Local Circulation and Medium－Range Transport of Air Pollutions. (Applied Meteorology Research Division, 1984)

第12号 火山活動監視手法に関する研究（地震火山研究部，1984）
Investigation on the Techniques for Volcanic Activity Surveillance. (Seismology and Volcanology Research Division, 1984)

第13号 気象研究所大気大循環モデル－Ⅰ（MRI・GCM－Ⅰ）（予報研究部，1984）
A Description of the MRI Atmospheric General Circulation Model (The MRI・GCM－Ⅰ). (Forecast Research Division, 1984)

第14号 台風の構造の変化と移動に関する研究－台風7916の一生－（台風研究部，1985）
A Study on the Changes of the Three - Dimensional Structure and the Movement Speed of the Typhoon through its Life Time. (Typhoon Research Division, 1985)

第15号 波浪推算モデルMRIとMRI－Ⅱの相互比較研究－計算結果図集－（海洋気象研究部，1985）
An Intercomparison Study between the Wave Models MRI and MRI－Ⅱ －A Compilation of Results－ (Oceanographical Research Division, 1985)

第16号 地震予知に関する実験的及び理論的研究（地震火山研究部，1985）
Study on Earthquake Prediction by Geophysical Method. (Seismology and Volcanology Research Division, 1985)

第17号 北半球地上月平均気温偏差図（予報研究部，1986）
Maps of Monthly Mean Surface Temperature Anomalies over the Northern Hemisphere for 1891－1981. (Forecast Research Division, 1986)

第18号 中層大気の研究（高層物理研究部，気象衛星研究部，予報研究部，地磁気観測所，1986）
Studies of the Middle Atmosphere. (Upper Atmosphere Physics Research Division, Meteorological Satellite Research Division, Forecast Research Division, MRI and the Magnetic Observatory, 1986)

第19号 ドップラーレーダによる気象・海象の研究（気象衛星研究部・台風研究部・予報研究部・応用気象研究部・海洋研究部，1986）
Studies on Meteorological and Sea Surface Phenomena by Doppler Radar. (Meteorological Satellite Research Division, Typhoon Research Division, Forecast Research Division, Applied Meteorology Research Division, and Oceanographical Research Division, 1986)

第20号 気象研究所対流圏大気大循環モデル（MRI・GCM－Ⅰ）による12年間分の積分（予報研究部，1986）
Mean Statistics of the Tropospheric MRI・GCM－Ⅰ based on 12－year Integration. (Forecast Research Division, 1986)

第21号 宇宙線中間子強度1983－1986（高層物理研究部，1987）
Multi－Directional Cosmic Ray Meson Intensity 1983－1986. (Upper Atmosphere Physics Research Division, 1987)

第22号 静止気象衛星「ひまわり」画像の噴火噴煙データに基づく噴火活動の解析に関する研究（地震火山研究部，1987）

Study on Analysis of Volcanic Eruptions based on Eruption Cloud Image Data obtained by the Geostationary Meteorological satellite (GMS). (Seismology and Volcanology Research Division, 1987)

第 23 号 オホーツク海海洋気候図（篠原吉雄，四竃信行，1988）
Marine Climatological Atlas of the sea of Okhotsk. (Y. Shinohara and N. Shikama, 1988)

第 24 号 海洋大循環モデルを用いた風の応力異常に対する太平洋の応答実験（海洋研究部，1989）
Response Experiment of Pacific Ocean to Anomalous Wind Stress with Ocean General Circulation Model. (Oceanographical Research Division, 1989)

第 25 号 太平洋における海洋諸要素の季節平均分布（海洋研究部，1989）
Seasonal Mean Distribution of Sea Properties in the Pacific. (Oceanographical Research Division, 1989)

第 26 号 地震前兆現象のデータベース（地震火山研究部, 1990）
Database of Earthquake Precursors. (Seismology and Volcanology Research Division, 1990)

第 27 号 沖縄地方における梅雨期の降水システムの特性（台風研究部，1991）
Characteristics of Precipitation Systems During the Baiu Season in the Okinawa Area. (Typhoon Research Division, 1991)

第 28 号 気象研究所・予報研究部で開発された非静水圧モデル（猪川元興・斉藤和雄，1991）
Description of a Nonhydrostatic Model Developed at the Forecast Research Department of the MRI. (M. Ikawa and K. Saito, 1991)

第 29 号 雲の放射過程に関する総合的研究（気候研究部・物理気象研究部・応用気象研究部・気象衛星・観測システム研究部・台風研究部，1992）
A Synthetic Study on Cloud－Radiation Processes. (Climate Research Department, Physical Meteorology Research Department, Applied Meteorology Research Department, Meteorological Satellite and Observation System Research Department, and Typhoon Research Department, 1992)

第 30 号 大気と海洋・地表とのエネルギー交換過程に関する研究（三上正男・遠藤昌宏・新野　宏・山崎孝治，1992）
Studies of Energy Exchange Processes between the Ocean－Ground Surface and Atmosphere. (M. Mikami, M. Endoh, H. Niino, and K. Yamazaki, 1992)

第 31 号 降水日の出現頻度からみた日本の季節推移－30 年間の日降水量資料に基づく統計－（秋山孝子，1993）
Seasonal Transition in Japan, as Revealed by Appearance Frequency of Precipitating-Days. －Statistics of Daily Precipitation Data During 30 Years－(T. Akiyama, 1993)

第 32 号 直下型地震予知に関する観測的研究（地震火山研究部，1994）
Observational Study on the Prediction of Disastrous Intraplate Earthquakes. (Seismology and Volcanology Research Department, 1994)

第 33 号 各種気象観測機器による比較観測（気象衛星・観測システム研究部，1994）
Intercomparisons of Meteorological Observation Instruments. (Meteorological Satellite and Observation System Research Department, 1994)

第 34 号 硫黄酸化物の長距離輸送モデルと東アジア地域への適用（応用気象研究部，1995）
The Long－Range Transport Model of Sulfur Oxides and Its Application to the East Asian Region. (Applied Meteorology Research Department, 1995)

第 35 号 ウインドプロファイラーによる気象の観測法の研究（気象衛星・観測システム研究部，1995）
Studies on Wind Profiler Techniques for the Measurements of Winds. (Meteorological Satellite and Observation System Research Department, 1995)

第 36 号 降水・落下塵中の人工放射性核種の分析法及びその地球化学的研究（地球化学研究部，1996）
Geochemical Studies and Analytical Methods of Anthropogenic Radionuclides in Fallout Samples. (Geochemical Research Department, 1996)

第 37 号 大気と海洋の地球化学的研究（1995 年及び 1996 年）（地球化学研究部，1998）
Geochemical Study of the Atmosphere and Ocean in 1995 and 1996. (Geochemical Research Department, 1998)

第 38 号 鉛直２次元非線形問題（金久博忠，1999）
Vertically 2-dmensional Nonlinear Problem (H. Kanehisa, 1999)

第 39 号 客観的予報技術の研究（予報研究部，2000）
Study on the Objective Forecasting Techniques（Forecast Research Department, 2000）

第 40 号 南関東地域における応力場と地震活動予測に関する研究（地震火山研究部，2000）
Study on Stress Field and Forecast of Seismic Activity in the Kanto Region（Seismology and Volcanology Research Department, 2000）

第 41 号 電量滴定法による海水中の全炭酸濃度の高精度分析および大気中の二酸化炭素と海水中の全炭酸の放射性炭素同位体比の測定（石井雅男・吉川久幸・松枝秀和，2000）
Coulometric Precise Analysis of Total Inorganic Carbon in Seawater and Measurements of Radiocarbon for the Carbon Dioxide in the Atmosphere and for the Total Inorganic Carbon in Seawater（I.Masao, H.Y.Inoue and H.Matsueda, 2000）

第 42 号 気象研究所／数値予報課統一非静力学モデル（斉藤和雄・加藤輝之・永戸久喜・室井ちあし，2001）
Documentation of the Meteorological Research Institute / Numerical Prediction Division Unified Nonhydrostatic Model (Kazuo Saito, Teruyuki Kato, Hisaki Eito and Chiashi Muroi，2001)

第 43 号 大気および海水中のクロロフルオロカーボン類の精密測定と気象研究所クロロフルオロカーボン類標準ガスの確立（時枝隆之・井上(吉川)久幸，2004）
Precise measurements of atmospheric and oceanic chlorofluorocarbons and MRI chlorofluorocarbons calibration scale (Takayuki Tokieda and Hisayuki Y. Inoue，2004)

第 44 号 PostScript コードを生成する描画ツール"PLOTPS"マニュアル（加藤輝之，2004）

Documentation of "PLOTPS": Outputting Tools for PostScript Code (Teruyuki Kato，2004)

第 45 号 気象庁及び気象研究所における二酸化炭素の長期観測に使用された標準ガスのスケールとその安定性の再評価に関する調査・研究（松枝秀和・須田一人・西岡佐喜子・平野礼朗・澤　庸介・坪井一寛・堤　之智・神谷ひとみ・根本和宏・長井秀樹・吉田雅司・岩野園城・山本　治・森下秀昭・鎌田匡俊・和田　晃，2004）

Re-evaluation for scale and stability of CO_2 standard gases used as long-term observations at the Japan Meteorological Agency and the Meteorological Research Institute (Hidekazu Matsueda, Kazuto Suda, Sakiko Nishioka, Toshirou Hirano, Yousuke, Sawa, Kazuhiro Tuboi, Tsutumi, Hitomi Kamiya, Kazuhiro Nemoto, Hideki Nagai, Masashi Yoshida, Sonoki Iwano, Osamu Yamamoto, Hideaki Morishita, Kamata, Akira Wada，2004)

第 46 号 地震発生過程の詳細なモデリングによる東海地震発生の推定精度向上に関する研究（地震火山研究部，2005）

A Study to Improve Accuracy of Forecasting the Tokai Earthquake by Modeling the Generation Processes (Seismology and Volcanology Research Department, 2005)

第 47 号 気象研究所共用海洋モデル（MRI.COM）解説（海洋研究部，2005）

Meteorological Research Institute Community Ocean Model (MRI.COM) Manual (Oceanographical Research Department, 2005)

第 48 号 日本海降雪雲の降水機構と人工調節の可能性に関する研究（物理気象研究部・予報研究部，2005）

Study of Precipitation Mechanisms in Snow Clouds over the Sea of Japan and Feasibility of Their Modification by Seeding (Physical Meteorology Research Department, Forecast Research Department, 2005)

第 49 号 2004 年日本上陸台風の概要と環境場（台風研究部, 2006）

Summary of Landfalling Typhoons in Japan, 2004 (Typhoon Research Department, 2006)

第 50 号 栄養塩測定用海水組成標準の 2003 年国際共同実験報告（青山道夫，2006）

2003 Intercomparison Exercise for Reference Material for Nutrients in Seawater in a Seawater Matrix (Michio Aoyama, 2006)

第 51 号 大気および海水中の超微量六フッ化硫黄(SF_6)の測定手法の高度化と SF_6 標準ガスの長期安定性の評価（時枝隆之、石井雅男、斉藤 秀、緑川 貴, 2007）

Highly developed precise analysis of atmospheric and oceanic sulfur hexafluoride (SF_6) and evaluation of SF_6 standard gas stability (Takayuki Tokieda, Masao Ishii, Shu Saito and Takashi Midorikawa, 2007)

第 52 号 地球温暖化による東北地方の気候変化に関する研究（仙台管区気象台, 環境・応用気象研究部, 2008）

Study of Climate Change over Tohoku District due to Global Warming (Sendai District Meteorological Observatory, Atmospheric Environment and Applied Meteorology Research Department, 2008)

第 53 号 火山活動評価手法の開発研究（地震火山研究部, 2008）

Studies on Evaluation Method of Volcanic Activity (Seismology and Volcanology Research Department, 2008)

第 54 号 日本における活性炭冷却捕集およびガスクロ分離による気体計数システムによる ^{85}Kr の測定システムの構築および 1995 年から 2006 年の測定結果（青山道夫, 藤井憲治, 廣瀬勝己, 五十嵐康人, 磯貝啓介, 新田 済, Hartmut Sartorius, Clemens Schlosser, Wolfgang Weiss, 2008）

Establishment of a cold charcoal trap-gas chromatography-gas counting system for ^{85}Kr measurements in Japan and results from 1995 to 2006 (Michio Aoyama, Kenji Fujii, Katsumi Hirose, Yasuhito Igarashi, Keisuke Isogai, Wataru Nitta, Hartmut Sartorius, Clemens Schlosser, Wolfgang Weiss, 2008)

第 55 号 長期係留による 4 種類の流速計観測結果の比較（中野俊也, 石崎 廣, 四竈信行, 2008）

Comparison of Data from Four Current Meters Obtained by Long-Term Deep-Sea Moorings (Toshiya Nakano, Hiroshi Ishizaki and Nobuyuki Shikama, 2008)

第 56 号 CMIP3 マルチモデルアンサンブル平均を利用した将来の海面水温・海氷分布の推定（水田 亮, 足立恭将, 行本誠史, 楠 昌司, 2008）

Estimation of the Future Distribution of Sea Surface Temperature and Sea Ice Using the CMIP3 Multi-model Ensemble Mean (Ryo Mizuta, Yukimasa Adachi, Seiji Yukimoto and Shoji Kusunoki, 2008)

第 57 号 閉流路中のフローセルを用いた分光光度法自動分析装置による海水の高精度 pH_T 測定（斉藤 秀, 石井雅男, 緑川 貴, 井上（吉川）久幸, 2008）

Precise Spectrophotometric Measurement of Seawater pH_T with an Automated Apparatus using a Flow Cell in a Closed Circuit（Shu Saito, Masao Ishii, Takashi Midorikawa and Hisayuki Y. Inoue, 2008）

第 58 号 栄養塩測定用海水組成標準の 2006 年国際共同実験報告（青山道夫,J. Barwell-Clarke, S. Becker, M. Blum, Braga E.S., S. C. Coverly, E. Czobik, I. Dahllöf, M. Dai, G. O Donnell, C. Engelke, Gwo-Ching Gong, Gi-Hoon Hong, D. J. Hydes, Ming-Ming Jin, 葛西広海, R. Kerouel, 清本容子, M. Knockaert, N. Kress, K. A. Krogslund, 熊谷正光, S. Leterme, Yarong Li, 増田真次, 宮尾 孝, T. Moutin, 村田昌彦, 永井直樹, G. Nausch, A. Nybakk, M. K. Ngirchechol, 小川浩史, J. van Ooijen, 太田秀和, J. Pan, C. Payne, O. Pierre-Duplessix, M. Pujo-Pay, T. Raabe, 齊藤一浩, 佐藤憲一郎, C. Schmidt, M. Schuett, T. M. Shammon, J. Sun, T. Tanhua, L. White, E.M.S. Woodward, P. Worsfold, P. Yeats, 芳村 毅, A. Youénou, Jia-Zhong Zhang, 2008）

2006 Inter-laboratory Comparison Study for Reference Material for Nutrients in Seawater（M. Aoyama, J. Barwell-Clarke, S. Becker, M. Blum, Braga E. S., S. C. Coverly, E. Czobik, I. Dahllöf, M. H. Dai, G. O. Donnell, C. Engelke, G. C. Gong, Gi-Hoon Hong, D. J. Hydes, M. M. Jin, H. Kasai, R. Kerouel, Y. Kiyomono, M. Knockaert, N. Kress, K. A. Krogslund, M. Kumagai, S. Leterme, Yarong Li, S. Masuda, T. Miyao, T. Moutin, A. Murata, N. Nagai, G. Nausch, M. K. Ngirchechol, A. Nybakk, H. Ogawa, J. van Ooijen, H. Ota, J. M. Pan, C. Payne, O. Pierre-Duplessix, M. Pujo-Pay, T. Raabe, K. Saito, K. Sato, C. Schmidt, M. Schuett, T. M. Shammon, J. Sun, T. Tanhua, L. White, E.M.S. Woodward, P. Worsfold, P. Yeats, T. Yoshimura, A. Youénou, J. Z. Zhang, 2008）

第 59 号 気象研究所共用海洋モデル(MRI.COM)第 3 版解説（辻野博之, 本井達夫, 石川一郎, 平原幹俊, 中野英之, 山中

吾郎, 安田珠幾, 石崎廣（気象研究所海洋研究部）, 2010）

Reference manual for the Meteorological Research Institute Community Ocean Model (MRI.COM) Version 3（Hiroyuki Tsujino, Tatsuo Motoi, Ichiro Ishikawa, Mikitoshi Hirabara, Hideyuki Nakano, Goro Yamanaka, Tamaki Yasuda, and Hiroshi Ishizaki (Oceanographic Research Department), 2010）

第60号 栄養塩測定用海水組成標準の2008年国際共同実験報告（青山道夫, Carol Anstey, Janet Barwell-Clarke, François Baurand, Susan Becker, Marguerite Blum, Stephen C. Coverly, Edward Czobik, Florence D’amico, Ingela Dahllöf, Minhan Dai, Judy Dobson, Magali Duval, Clemens Engelke, Gwo-Ching Gong, Olivier Grosso, 平山篤史, 井上博敬, 石田雄三, David J. Hydes, 葛西広海, Roger Kerouel, Marc Knockaert, Nurit Kress, Katherine A. Krogslund, 熊谷正光, Sophie C. Leterme, Claire Mahaffey, 光田均, Pascal Morin, Thierry Moutin, Dominique Munaron, 村田昌彦, Günther Nausch, 小川浩史, Jan van Ooijen, Jianming Pan, Georges Paradis, Chris Payne, Olivier Pierre-Duplessix, Gary Prove, Patrick Raimbault, Malcolm Rose, 齊藤一浩, 斉藤宏明, 佐藤憲一郎, Cristopher Schmidt, Monika Schütt, Theresa M. Shammon, Solveig Olafsdottir, Jun Sun, Toste Tanhua, Sieglinde Weigelt-Krenz, Linda White, E. Malcolm. S. Woodward, Paul Worsfold, 芳村毅, Agnès Youénou, Jia-Zhong Zhang, 2010）

2008 Inter-laboratory Comparison Study of a Reference Material for Nutrients in Seawater（青山道夫, Carol Anstey, Janet Barwell-Clarke, François Baurand, Susan Becker, Marguerite Blum, Stephen C. Coverly, Edward Czobik, Florence D’amico, Ingela Dahllöf, Minhan Dai, Judy Dobson, Magali Duval, Clemens Engelke, Gwo-Ching Gong, Olivier Grosso, 平山篤史, 井上博敬, 石田雄三, David J. Hydes, 葛西広海, Roger Kerouel, Marc Knockaert, Nurit Kress, Katherine A. Krogslund, 熊谷正光, Sophie C. Leterme, Claire Mahaffey, 光田均, Pascal Morin, Thierry Moutin, Dominique Munaron, 村田昌彦, Günther Nausch, 小川浩史, Jan van Ooijen, Jianming Pan, Georges Paradis, Chris Payne, Olivier Pierre-Duplessix, Gary Prove, Patrick Raimbault, Malcolm Rose, 齊藤一浩, 斉藤宏明, 佐藤憲一郎, Cristopher Schmidt, Monika Schütt, Theresa M. Shammon, Solveig Olafsdottir, Jun Sun, Toste Tanhua, Sieglinde Weigelt-Krenz, Linda White, E. Malcolm. S. Woodward, Paul Worsfold, 芳村毅, Agnès Youénou, Jia-Zhong Zhang, 2010）

第61号 強雨をもたらす線状降水帯の形成機構等の解明及び降水強度・移動速度の予測に関する研究（大阪管区気象台・彦根地方気象台・京都地方気象台・奈良地方気象台・和歌山地方気象台・神戸海洋気象台・松江地方気象台・鳥取地方気象台・舞鶴海洋気象台・広島地方気象台・徳島地方気象台・予報研究部, 2010）

Studies on formation process of line-shaped rainfall systems and predictability of rainfall intensity and moving speed（Osaka District Meteorological Observatory, Hikone Local Meteorological Observatory, Kyoto Local Meteorological Observatory, Nara Local Meteorological Observatory, Wakayama Local Meteorological Observatory, Kobe Marine Observatory, Matsue Local Meteorological Observatory, Tottori Local Meteorological Observatory, Maizuru Marine Observatory, Hiroshima Local Meteorological Observatory, Tokushima Local Meteorological Observatory AND Forecast Research Department, 2010）

第62号 WWRP北京オリンピック2008予報実証/研究開発プロジェクト（齊藤和雄, 國井勝, 原昌弘, 瀬古弘, 原旅人, 山口宗彦, 三好建正, 黃偉健, 2010）

WWRP Beijing Olympics 2008 Forecast Demonstration/Research and Development Project (B08FDP/RDP)（Kazuo Saito, Masaru Kunii, Masahiro Hara, Hiromu Seko, Tabito Hara, Munehiko Yamaguchi, Takemasa Miyoshi and Wai-kin Wong, 2010）

第63号 東海地震の予測精度向上及び東南海・南海地震の発生準備過程の研究（地震火山研究部, 2011）

Improvement in prediction accuracy for the Tokai earthquake and research of the preparation process of the Tonankai and the Nankai earthquakes (Seismology and Volcanology Research Department, 2011)

第64号 気象研究所地球システムモデル第1版（MRI-ESM1）—モデルの記述—（行本誠史, 吉村裕正, 保坂征宏, 坂見智法, 辻野博之, 平原幹俊, 田中泰宙, 出牛真, 小畑淳, 中野英之, 足立恭将, 新藤永樹, 籔将吉, 尾瀬智昭, 鬼頭昭雄, 2011）

Meteorological Research Institute-Earth System Model Version 1 (MRI-ESM1) — Model Description — (Seiji Yukimoto, Hiromasa Yoshimura, Masahiro Hosaka, Tomonori Sakami, Hiroyuki Tsujino, Mikitoshi Hirabara, Taichu Y. Tanaka, Makoto Deushi, Atsushi Obata, Hideyuki Nakano, Yukimasa Adachi, Eiki Shindo, Shoukichi Yabu, Tomoaki Ose and Akio Kitoh, 2011)

第65号 東南アジア地域の気象災害軽減国際共同研究（斉藤和雄, 黒田徹, 林修吾, 瀬古弘, 國井勝, 小司禎教, 上野充, 川畑拓矢, 余田成男, 大塚成徳, Nurjanna Joko Trilaksono, 許智揚, 古関俊也, Le Duc, Kieu Thi Xin, 黄偉健, Krushna Chandra Gouda, 2011）

International Research for Prevention and Mitigation of Meteorological Disasters in Southeast Asia (Kazuo Saito, Tohru Kuroda, Syugo Hayashi, Hiromu Seko, Masaru Kunii, Yoshinori Shoji, Mitsuru Ueno, Takuya Kawabata, Shigeo Yoden, Shigenori Otsuka, Nurjanna Joko Trilaksono, Tieh-Yong Koh, Syunya Koseki, Le Duc, Kieu Thi Xin, Wai-Kin Wong and Krushna Chandra Gouda, 2011)

第66号 太平洋における大気－海洋間二酸化炭素フラックス推定手法（杉本裕之, 平石直孝, 石井雅男, 緑川貴, 2012）

A method for estimating the sea-air CO2 flux in the Pacific Ocean (Hiroyuki Sugimoto, Naotaka Hiraishi, Masao Ishii and Takashi Midorikawa, 2012)

第67号 C-130H輸送機を利用した温室効果ガス観測のためのフラスコ採取とその高精度測定システムの開発（坪井一寛, 松枝秀和, 澤庸介, 丹羽洋介, 中村雅道, 久保池大輔, 岩坪昇平, 齊藤和幸, 花宮義和, 辻健太郎, 大森英裕, 西秀紘, 2012）

Development of a flask sampling and its high-precision measuring system for greenhouse gases observations using a cargo aircraft C-130H (Kazuhiro Tsuboi, Hidekazu Matsueda, Yousuke Sawa, Yosuke Niwa Masamichi Nakamura, Daisuke Kuboike, Shohei Iwatsubo, Kazuyuki Saito Yoshikazu Hanamiya, Kentaro Tsuji, Hidehiro Ohmori, Hidehiro Nishi, 2012)

第 68 号　国際シンポジウム 電子顕微鏡を用いたエアロゾル研究（五十嵐康人, Weijun Li, Peter.R.Buseck, 岡田菊雄, 張代洲, 足立光司, 藤谷雄二, 嶋寺光, 五藤大輔, 三井千珠, 野島雅, 大島長, 松井仁志, 石元裕史, 松木篤, Pradeep Khatri, 中山智喜, 向井将平, 大石乾詞, 間山憲仁, 坂本哲夫, 直江寛明, 財前祐二, 塩流水洋樹, 田中泰宙, 梶野瑞王, 2013）
International Symposium on Aerosol Studies Explored by Electron Microscopy (Yasuhito Igarashi, Weijun Li, Peter. R. Buseck, Kikuo Okada, Daizhou Zhang, Kouji Adachi, Yuji Fujitani, Hikari Shimadera, Daisuke Goto, Chizu Mitsui, Masashi Nojima, Naga Oshima, Hitoshi Matsui, Hiroshi Ishimoto, Atsushi Matsuki, Pradeep Khatri, Tomoki Nakayama, Shohei Mukai, Kenji Ohishi, Norihito Mayama, Tetsuo Sakamoto, Hiroaki Naoe, Yuji Zaizen, Hiroki Shiozuru, Taichu Y. Tanaka and Mizuo Kajino, 2013)

第 69 号　マグマ活動の定量的把握技術の開発とそれに基づく火山活動度判定の高度化に関する研究（地震火山研究部, 2013）
Development of Quantitative Detection Techniques of Magma Activity and Improvement of Evaluation of Volcanic Activity Level (Seismology and Volcanology Research Department, MRI, 2013)

第 70 号　平成 23 年（2011 年）東北地方太平洋沖地震による津波高の現地調査報告（林豊，前田憲二，対馬弘晃，岡田正實，木村一洋，岩切一宏, 2013）
Reports on Field Surveys of Tsunami Heights from the 2011 off the Pacific Coast of Tohoku Earthquake (Yutaka Hayashi, Kenji Maeda, Hiroaki Tsushima, Masami Okada, Kazuhiro Kimura and Kazuhiro Iwakiri, 2013)

第 71 号　気候研究のための気象研究所アンサンブル予測システムの概要とその応用（藪将吉，水田亮，吉村裕正，黒田友二，向川均, 2014）
Meteorological Research Institute Ensemble Prediction System (MRI-EPS) for climate research - Outline and its applications – (Shoukichi Yabu, Ryo Mizuta, Hiromasa Yoshimura, Yuhji Kuroda, and Hitoshi Mukougawa, 2014)

第 72 号　日本各地域の繰り返し相似地震発生状況に関する研究（地震火山研究部，地震火山部，気象大学校，札幌管区気象台，仙台管区気象台，大阪管区気象台，福岡管区気象台，沖縄気象台, 2014）
Survey of moderate repeating earthquakes in Japan (Seismology and Volcanology Research Department of MRI, Seismology and Volcanology Department, Meteorological College, Sapporo Regional Headquarters, Sendai Regional Headquarters, Osaka Regional Headquarters, Fukuoka Regional Headquarters, and Okinawa Regional Headquarters, 2014)

第 73 号　気象研究所非静力学地域気候モデルによる日本付近の将来気候変化予測について（佐々木秀孝，村田昭彦，川瀬宏明，花房瑞樹，野坂真也，大泉三津夫，水田亮，青栁曉典，志藤文武，石原幸司, 2015）
Projection of Future Climate Change around Japan by using MRI Non-hydrostatic Regional Climate Model (Hidetaka Sasaki, Akihiko Murata, Hiroaki Kawase, Mizuki Hanafusa, Masaya Nosaka, Mitsuo Oh'izumi, Ryou Mizuta, Toshinori Aoyagi, Fumitake Shido, and Koji Ishihara, 2015)

第 74 号　新型自己浮上式海底水圧計の開発（平田賢治, 山崎明, 対馬弘晃, 2015）
Development of a new pop-up ocean-bottom pressure gauge (Kenji Hirata, Akira Yamazaki, and Hiroaki Tsushima, 2015)

第 75 号　2012 年・2013 年に日本に接近・上陸した台風の概要と特性（北畠尚子, 小山亮, 嶋田宇大, 櫻木智明, 沢田雅洋, 2015）
Summary and Characteristics of Approaching and Landfalling Tropical Cyclones in Japan in 2012 and 2013 (Naoko Kitabatake, Ryo Oyama, Udai Shimada, Tomoaki Sakuragi and Masahiro Sawada, 2015)

第 76 号　ＷＭＯ福島第一原発事故に関する気象解析技術タスクチーム活動と気象研究所の大気拡散モデリング（斉藤和雄, 新堀敏基, R. Draxler, 原旅人, 豊田英司, 本田有機, 永田和彦, 藤田司, 坂本雅巳, 加藤輝之, 梶野瑞王, 関山剛, 田中泰宙, 眞木貴史, 寺田宏明, 茅野政道, 岩崎俊樹, M.C. Hort, S.J. Leadbetter, G. Wotawa, D. Arnold, C. Maurer, A. Malo, R. Servranckx, P. Chen, 2015）
Contribution of JMA to the WMO Technical Task Team on Meteorological Analyses for Fukushima Daiichi Nuclear Power Plant Accident and Relevant Atmospheric Transport Modelling at MRI(K. Saito, T. Shimbori, R. Draxler, T. Hara, E. Toyoda, Y. Honda, K. Nagata, T. Fujita, M. Sakamoto, T. Kato, M. Kajino, T.T. Sekiyama, T.Y. Tanaka, T. Maki, H. Terada, M. Chino, T. Iwasaki, M.C. Hort, S.J. Leadbetter, G. Wotawa, D. Arnold, C. Maurer, A. Malo, R. Servranckx and P. Chen, 2015)

第 77 号　海溝沿い巨大地震の地震像の即時的把握に関する研究（地震津波研究部, 2017）
Research on rapid estimation of the parameters for large earthquakes along trenches (Seismology and Tsunami Research Department, 2017)

第 78 号　2013-2015 年西之島噴火のモニタリングに関する研究（高木朗充, 長岡優, 福井敬一, 安藤忍, 木村一洋, 土山博昭，2017）
Studies on Monitoring of the 2013 – 2015 Nishinoshima Eruption (Akimichi Takagi, Yutaka Nagaoka, Keiichi Fukui, Shinobu Ando, Kazuhiro Kimura, and Hiroaki Tsuchiyama, 2017)

第 79 号　2012 年から 2016 年に実施された温室効果ガス観測に関する相互比較実験（iceGGO）（坪井一寛、中澤高清、松枝秀和、町田敏暢、青木周司、森本真司、後藤大輔、下坂琢哉、加藤健次、青木伸行、渡邉卓朗、向井人史、遠嶋康徳、勝又啓一、村山昌平、石戸谷重之、藤谷徳之助、小出寛、髙橋正臣、川﨑照夫、滝沢厚詩、澤庸介, 2017）
InterComparison Experiments for Greenhouse Gases Observation (iceGGO) in 2012–2016 (A K. Tsuboi, T. Nakazawa, H. Matsueda, T. Machida, S. Aoki, S. Morimoto, D. Goto, T. Shimosaka, K. Kato, N. Aoki, T. Watanabe, H. Mukai, Y. Tohjima, K. Katsumata, S. Murayama, S. Ishidoya, T. Fujitani, H. Koide, M. Takahashi, T. Kawasaki, A. Takizawa and Y. Sawa, 2017)

第 80 号　気象研究所共用海洋モデル第４版(MRI.COMv4)解説（辻野博之、中野英之、坂本圭、浦川昇吾、平原幹俊、石崎廣、山中吾郎（気象研究所海洋・地球化学研究部）, 2017）
Reference Manual for the Meteorological Research Institute Community Ocean Model version 4 (MRI.COMv4) (Hiroyuki Tsujino, Hideyuki Nakano, Kei Sakamoto, Shogo Urakawa, Mikitoshi Hirabara, Hiroshi Ishizaki, and Goro Yamanaka, (Oceanography and Geochemistry Research Department), 2017)

第 81 号　集中豪雨・大雨発生の必要条件の抽出・妥当性の確認と十分条件の抽出（津口裕茂（予報研究部）・大阪管区気象台・彦根地方気象台・京都地方気象台・神戸地方気象台・奈良地方気象台・和歌山地方気象台・広島地方気象台・岡山地方気象台・松江地方気象台・鳥取地方気象台・高松地方気象台・徳島地方気象台・松山地方気象台・高知地方気象台, 2018）

Extraction and Validation of Necessary Conditions and Analysis of Sufficient Conditions for Causing Heavy Rainfall (Hiroshige TSUGUTI (Forecast Research Department), Osaka District Meteorological Observatory, Hikone Local Meteorological Observatory, Kyoto Local Meteorologica Observatory, Kobe Local Meteorological Observatory, Nara Local Meteorological Observatory, Wakayama Local Meteorological Observatory, Hiroshima Local Meteorological Observatory, Okayama Local Meteorological Observatory, Matsue Local Meteorological Observatory, Tottori Local Meteorological Observatory, Takamatsu Local Meteorological Observatory, Tokushima Local Meteorological Observatory, Matsuyama Local Meteorological Observatory and Kochi Local Meteorological Observatory), 2018)

第 82 号　台風予報・解析技術高度化プロジェクトチームによる 5 日先台風強度予報ガイダンスの開発（山口宗彦、嶋田宇大、沢田雅洋、入口武史(台風研究部)、大和田浩美（気象庁））, 2019）

Development of 5-day Typhoon Intensity Forecast Guidance by the Project Team for Improvement in Operational Typhoon Forecasts and Analysis (Munehiko Yamaguchi, Udai Shimada, Masahiro Sawada, Takeshi Iriguchi(Typhoon Research Department, Meteorological Research Institute), and Hiromi Owada(Japan Meteorological Agency), 2019)

第 83 号　日本沿岸海況監視予測システム 10 年再解析値（JPN Atlas 2020）（広瀬成章、坂本圭、碓氷典久、山中吾郎、高野洋雄(全球大気海洋研究部)）, 2020）

The 10-year reanalysis dataset of an operational system for monitoring and forecasting coastal and open-ocean status around Japan (JPN Atlas 2020) (Nariaki Hirose, Kei Sakamoto, Norihisa Usui, Goro Yamanaka, and Nadao Kohno (Department of Atmosphere, Ocean and Earth System Modeling Research, Meteorological Research Institute), 2020)

第 84 号　気象庁移流拡散モデル設計書（新堀敏基、石井憲介(火山研究部）,2021）

Design of the Japan Meteorological Agency Atmospheric Transport Model(Toshiki Shimbori and Kensuke Ishii (Department of Volcanology Research, Meteorological Research Institute),2021))

第 85 号　旅客機搭載型の自動大気採取装置（ASE）の開発—経緯と技術的要件—（松枝秀和(気候・環境研究部), 近藤直人(ジャムコ社), 工藤明宏(日本アンス社), 坪井一寛(気候・環境研究部)）, 2021）

Development of Automatic air Sampling Equipment (ASE) for passenger aircraft - Backgrounds and Techniques –(Hidekazu Matsueda (Climate and Geochemistry Research, Meteorological Research Institute), Naoto Kondo (JAMCO), Akihiro Kudo (JANS), Kazuhiro Tsuboi (Climate and Geochemistry Research Department, Meteorological Research Institute), 2019)

第 86 号　遠地津波の観測データに基づく経験的な減衰予測手法（山本 剛靖(地震津波研究部)）, 2022）

Empirical Forecast Method for the Decay Process of Far-Field Tsunamis (Takeyasu Yamamoto (Seismology and Tsunami Research Department, Meteorological Research Institute), 2022)

気象研究所技術報告は、1978（昭和53）年の初刊以来、気象研究所が必要の都度発行する刊行物であり、気象研究所の研究計画に基づき実施した研究に関する手法、データ、結果等についてのまとめ、または、すでに公表した研究論文類をとりまとめ総合的報告としたものを掲載する。

気象研究所技術報告　ISSN 2189-8871
第 87 号

令和５年２月　発行

編集兼
発行者　気象研究所

〒305-0052　茨城県つくば市長峰１－１
TEL(029)853-8535