TECHNICAL REPORTS OF THE METEOROLOGICAL RESEARCH INSTITUTE No.80

Reference Manual for the Meteorological Research Institute

Community Ocean Model version 4

(MRI.COMv4)

BY

Hiroyuki Tsujino, Hideyuki Nakano, Kei Sakamoto, Shogo Urakawa Mikitoshi Hirabara, Hiroshi Ishizaki, and Goro Yamanaka, (Oceanography and Geochemistry Research Department)

気象研究所技術報告 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)



気象研究所

METEOROLOGICAL RESEARCH INSTITUTE, JAPAN March 2017

METEOROLOGICAL RESEARCH INSTITUTE

Established in 1946

Director-General: Dr. Kiyoharu Takano Senior Director for Research Affairs: Dr. Kazuo Saito Senior Director for Research Coordination: Mr. Yoshiaki Takeuchi

Forecast Research Department	Director: Mr. Ko Koizumi
Climate Research Department	Director: Dr. Tomoaki Ose
Typhoon Research Department	Director: Mr. Isao Takano
Atmospheric Environment and Applied Meteorology Research Department	Director: Dr. Izuru Takayabu
Meteorological Satellite and Observation System Research Department	Director: Mr. Osamu Suzuki
Seismology and Tsunami Research Department	Director: Dr. Kenji Maeda
Volcanology Research Department	Director: Dr. Hitoshi Yamasato
Oceanography and Geochemistry Research Department	Director: Dr. Tsurane Kuragano

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

TECHNICAL REPORTS OF THE METEOROLOGICAL RESEARCH INSTITUTE

Editor-in-chief: Tomoaki Ose

Editors:

Wataru Mashiko Yuji Kitamura Shinya Onizawa Managing Editors: Rai Okabe, Miyuki Kawamata

Masayoshi Ishii Hanako Inoue Norihisa Usui

Masahiro Sawada Hiroaki Tsushima

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.

©2017 by the Meteorological Research Institute.

The copyright of reports in this journal belongs to the Meteorological Research Institute (MRI). Permission is granted to use figures, tables and short quotes from reports in this journal, provided that the source is acknowledged. Republication, reproduction, translation, and other uses of any extent of reports in this journal require written permission from the MRI.

In exception of this requirement, personal uses for research, study or educational purposes do not require permission from the MRI, provided that the source is acknowledged.

Reference Manual for the Meteorological Research Institute Community Ocean Model version 4 (MRI.COMv4)

気象研究所共用海洋モデル第4版 (MRI.COMv4)

解説

Hiroyuki Tsujino, Hideyuki Nakano, Kei Sakamoto, Shogo Urakawa Mikitoshi Hirabara, Hiroshi Ishizaki, and Goro Yamanaka, (Oceanography and Geochemistry Research Department)

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)

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 Oceanography and Geochemistry Research Department of 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), and "Research on advanced ocean models" (starting from fiscal year 2014). The MRI.COM was revised as its fourth version. The present publication is the new MRI.COM manual corresponding to the MRI.COM (version 4). The manual has been thoroughly revised from the previous version published in 2010.

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 (at the time) and the Climate Research Department, 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.

Since 2010 in which the first English version of the MRI.COM manual was published, 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 4) is the introduction of a new vertical coordinate system (z^* coordinate system) which better represents the shallow area in coastal modeling. The introduction of a new two-way nesting scheme which accurately conserves water and heat has the benefit in climate modeling. This makes it possible for an ocean model to seamlessly handle climate change and 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 stand-alone experiments. MRI.COM has already been in operation at JMA for 9 years, contributing to providing reliable information such as seasonal prediction and ocean prediction. Based on our experiences,

we believe MRI.COM is one of the best ocean models in the world. I express special thanks to the present and past participants in the model development for their great deal of efforts and help. 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.

Tsurane Kuragano Director Oceanography and Geochemistry Research Department

前文

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。

気象研究所海洋・地球化学研究部では、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研究は、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年度~平成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一フィージビリティ・スタディー」(平 成20年度)、「海洋環境の予測技術の開発」(平成21年度~平成25年度)、「海洋モデルの高度化に関する研究」 (平成26年度~平成30年度)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更に汎用性を高くした MRI.COM(第4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(2010年出版)の内容を大幅に改訂して出版するものである。

海洋・地球化学研究部におけるモデリング研究は、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(当時)と気候 研究部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、海洋研究部では、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。

前回の英語版解説書が出版された 2010 年以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続けられた。MRI.COM(第4版)について特筆すべきは、沿岸モデリングにおける浅海域の再現性向上のために、新しい鉛直座標系(z*座標系)を導入したことである。また、気候モデリングに資するため保存性を向上させたネスティングスキームを導入する等の改良が行なわれ、気候変動から沿岸防災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9年間現業運用され、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。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。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。

海洋・地球化学研究部長 倉賀野 連

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.

The current version of MRI.COM is version 4. Version 1 (developed around 2000) was intended to present a prototype. Efforts were devoted to combining the two ocean models used until then in MRI. For this reason, users at that time tended to be restricted to MRI research scientists who were committed to the development. Thus, users were deeply knowledgeable about the model.

Version 2 (early 2000s) was intended for use in the operational forecasting system in JMA. Since the number of users without direct experience in developing models was expected to increase, the developers decided to write a detailed manual for that version. The Japanese 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). One of the reasons for creating a new version was that the definition of vertical grid arrangement was modified during the development. MRI participated in the Coupled Model Intercomparison Project Phase 5 (CMIP5) using MRI-ESM1, and its results on the future projection were expected to be used by a wide range of communities, so a detailed description of its oceanic component was prepared in English (Tsujino et al., 2010).

Version 4 is 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 includes the following improvements.

- Tide producing potential and inverse barometer effect of surface atmospheric pressure are introduced into the depth-integrated (barotropic) equations, considering an application to coastal processes. The linear response of the system to the tide producing potential is calculated separately from the other processes, so that self-attraction and loading affect only tidal processes.
- A vertically rescaled height coordinate (z^* coordinate; Adcroft and Campin, 2004) has been introduced.
- Nesting function has been expanded. Off-line (parent and child models are executed serially) one-way, on-line (parent and child models are executed in parallel) one-way and two-way nesting become possible. In on-line two-way nesting, sea water volume and scalar quantities can be conserved for a system of parent and child models.
- Program files are classified into several packages to ease the maintenance. Each package is made as independent as possible from other ones and is controlled by a driver program, which is named as *package_ctl.F90*.
- The user interface is entirely revised. Specifically, monitoring and sampling functions have been expanded, several input and output files use a format different than the previous versions so that their contents can be checked easily, and the block names and the contents of namelists are renamed so that they are more self-explanatory. See READMEs for details.

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.

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 is explained in Chapter 3.

Part II describes the solution procedures of diagnostic equations. Chapter 4 presents definition of the equation of state of sea water. Chapter 5 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 6 and 7, respectively.

Part IV describes the method of solving the advection-diffusion equation for tracers. Chapter 8 presents advection schemes. Subgrid-scale parameterizations for horizontal and vertical mixing are explained in Chapters 9 and 10, respectively. Biogeochemical models are presented in Chapter 11 and inert tracers are explained in Chapter 12. Chapter 13 explains usage of the tracer package.

Part V describes boundary processes. Surface fluxes are presented in Chapter 14. Surface mixed-layer models are presented in Chapter 15. Bottom boundary layer parameterization is explained in Chapter 16. Sea ice part is explained in Chapter 17. How to construct and run a pair of nested-grid models is presented in Chapter 18.

Part VI contains miscellaneous topics. Basics of the finite difference method are presented in Chapter 19, general orthogonal curvilinear coordinates and related calculus are introduced in Chapter 20, and finally user's guide to construct and run a model is presented in Chapter 21.

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.

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 の最新バージョンは4である。バージョン1(2000年頃)は基本型を作成することを目的としたもので あった。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精力が注がれた。それ故、 当時の利用者は開発に直接携わる気象研究所の研究者に限定されていた。言い換えれば、利用者が本モデルに関する 深い知識を有していた。

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ものであった。モデル開発に直接 関与しない利用者数の増加が見込まれたため、当時の開発者らはこのバージョンに対する詳細な説明書の執筆を決め、 日本語による解説書が 2005 年に出版された (Ishikawa et al., 2005)。この日本語版は、その英語版である本解説書の 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。バージョン更新の理由のひとつは、開発段階で、鉛直格子点の定義位置を変更したことであっ た。気象研究所は MRI-ESM1 を用いて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。同モ デルによる将来予測結果等は、幅広い分野の人々に使用されることが見込まれたため、著者らは、その海洋部分につ いての詳細な説明を英語で執筆することとした (Tsujino et al., 2010)。

バージョン4(本バージョン)は気象庁における次期現業海洋予測システム、並びに気象研究所が参画する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。本バージョンでは以下の改良がおこなわれている。

- ・起潮力ポテンシャルおよび海面気圧による押し下げ・吸い上げ効果が順圧モードの方程式に導入された。これ らは沿岸海洋予測への適用を考慮したものである。起潮力ポテンシャルに対する系の線形応答は他のプロセス から独立して解かれる。そうすることによって、self-attraction 及び loading 効果は潮汐過程にのみ影響を及ぼ すようになっている。
- 海面水位変動を海面から海底までスケーリングする新しい鉛直座標 (z* coordinate; Adcroft and Campin, 2004)
 を導入した。
- ネスティング機能が拡充された。オフライン(親と子モデルが逐次実行される)単方向、オンライン(親と子 モデルが並列同時実行される)単方向、双方向ネスティングが可能となった。オンライン双方向ネスティング では、海水体積とトレーサー積算値が親モデルと子モデルのシステムで保存させることが可能となっている。
- プログラムを全面的に書き換え、各プロセスをパッケージ化し、保守を容易にした。各パッケージは可能な限 り他のパッケージに依存しないようにし、package_ctl.F90という名前のファイルに含まれる、ドライバープ ログラムのみにより制御される。
- ユーザーインターフェイスが全面的に改訂された。具体的には、モデルの挙動監視機能とサンプリング機能が 拡張された。いくつかの入出力ファイルでは以前のバージョンとは異なるフォーマットが使用され、中身が容 易に確認できるようになった。NAMELIST ブロック名および変数名を改め、より自明となるようにした。詳 しくは各種 README ファイルを参照されたい。

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。

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。

第Ⅲ部では、診断式について説明を行う。第4章は海水の状態方程式の定義を示す。第5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6章と第7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。トレーサー移流スキームについて第 8 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 9 章および第 10 章でそれぞれ解説する。幾つかの生物地球化学モデルを第 11 章で解説する。パッシブトレーサーの幾つかを第 12 章で解説する。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、幾つかの海面混合層モデルを第 15 章で、海底境界層モデルを第 16 章で、海氷モデルを第 17 章で解説する。入れ子モデルの作成と使用法について第 18 章で解説する。

第 VI 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 19 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 20 章で、最後にモデルの作成と実行方法の解説を第 21 章で行う。

各章はほぼ他の章から独立しており、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。

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	1
1.1	What do OGCMs cover?	1
1.2	Classification of OGCMs	2
1.3	About MRI.COM	3
1.4	Future of OGCMs and MRI.COM	3
Part I Co	nfiguration	5
Chapter 2	Governing Equations	7
2.1	Formulation	7
2.2	Numerical Methods	14
2.3	Appendix	19
Chapter 3	Spatial grid arrangement	21
3.1	Horizontal grid arrangement	21
3.2	Vertical grid arrangement	22
3.3	Indices and symbols	23
3.4	Calculation of horizontal grid cell area and width	23
3.5	Calculation of vertical cell thickness	26
3.6	Usage	28
Part II Dia	agnostic Equations	33
Chapter 4	Equation of State	35
4.1	Basics of the equation of state	35
4.2	An equation of state used by MRI.COM	36
Chapter 5	Continuity equation	41
5.1	Introduction	41
5.2	Finite difference expression for the T-cell continuity equation	41
5.3	Finite difference expression for the U-cell continuity equation	44
5.4	Summary	45
Part III Ed	quations of Motion	47
Chapter 6	Equations of motion (barotropic component)	49
6.1	Governing equations	49
6.2	Time integration on barotropic time levels	50
6.3	Prognostication of state variables at the baroclinic time level	53
6.4	Horizontal diffusivity of sea surface height	57
6.5	Inclusion of tidal effect	57
6.6	Usage	61
Chapter 7	Equations of motion (baroclinic component)	63
7.1	Advection terms	63
7.2	Pressure gradient term	73
7.3	Viscosity	75

7.4	Coriolis term	80
7.5	Usage	80

Part IV	Evolution of tracers	83
Chapter 8	Tracer advection schemes	85
. 8.1	Finite volume or flux form method	85
8.2	Weighted upcurrent scheme	86
8.3	QUICK scheme	88
8.4	QUICKEST for vertical advection	89
8.5	UTOPIA for horizontal advection	91
8.6	Second Order Moment (SOM) scheme	97
8.7	MPDATA scheme	103
Chapter 9	SGS parameterization of lateral mixing of tracers	107
9.1	Introduction and Formulation	107
9.2	Horizontal diffusion	108
9.3	Isopycnal diffusion	109
9.4	Gent and McWilliams parameterization for eddy-induced transport	112
Chapter 10	SGS parameterization of vertical mixing of tracers	119
10.1	Vertical diffusion	119
10.2	Convective adjustment	120
Chapter 11	Biogeochemical model	125
11.1	Inorganic carbon cycle and biological model	125
11.2	Governing equations	126
11.3	Carbon cycle component	126
11.4	Obata and Kitamura model	129
11.5	NPZD model	130
11.6	Usage	133
11.7	Program structure	134
Chapter 12	Inert tracers	137
12.1	Ideal age tracer	137
12.2	CFCs	137
12.3	SF_6	138
Chapter 13	Package Structure and Usage	139
13.1	Package structure	139
13.2	Handling the initial state	140
13.3	Configuration of tracers	140

Part V Boundary Processes

Chapter 14	Sea surface fluxes	149
14.1	Momentum flux (surface stress)	149
14.2	Heat flux	151
14.3	Shortwave radiation flux	152
14.4	Longwave radiation flux	155
14.5	Latent and sensible heat fluxes	155
14.6	Freshwater flux	156
14.7	Treatment in the temperature equation	159
14.8	Treatment in the salinity equation	159
14.9	Bulk transfer coefficient	161
14.10	Specification of input data	167

14.11 14.12	Surface data objects	. 168 . 170
Chapter 15 15.1	Turbulence Closure Models Statistical closure model	175 . 175
15.2	Mellor and Yamada model	. 176
15.3	Noh and Kim (1999) model	. 181
15.4	Generic length scale model by Umlauf and Burchard (2003)	. 184
Chapter 16	Bottom Boundary Layer (BBL)	189
16.1	General description	. 189
16.2	Grid arrangement and exchanges	. 189
16.3	Pressure gradient terms	. 190
16.4	Eddy effects	. 191
16.5	Usage	. 192
16.6	Usage notes	. 193
Chapter 17	Sea ice	195
17.1	Outline	. 195
17.2	Thermodynamic processes	. 197
17.3	Remapping in thickness space	. 205
17.4	Dynamics	. 207
17.5	Advection and Diffusion	. 210
17.6	Ridging	. 210
17.7	Adjustment	. 211
17.8	Discretization	. 211
17.9	Technical Notes	. 214
17.10	Usage notes	. 219
17.11	Appendix	. 223
Chapter 18	Nesting	225
18.1	Feature	. 225
18.2	Grid and Topography	. 226
18.3	Intermodel transfer	. 228
18.4	Conservation	. 231
18.5	Sponge region	. 235
18.6	Usage	. 236
18.7	Program structure	. 242
Part VI Mi	iscellaneous	245
Chapter 19	Basics of the finite difference method	247

Unapler 13	Dasics of the linke difference method	24/
19.1	Diffusion equation	. 247
19.2	Finite difference expressions for time derivatives	. 248
19.3	Finite difference expression for space derivatives	. 248
19.4	Finite differencing of advection-diffusion equation	. 250
19.5	Implicit method for vertical diffusion equation	. 250
19.6	Summary of schemes used by MRI.COM	. 251
Chapter 20	Generalized orthogonal curvilinear coordinate grids	253
20.1	Outline	. 253
20.2	Generation of orthogonal coordinate system using conformal mapping	. 253
20.3	Rotation of vector	. 255
20.4	Mapping a quantity from geographic coordinates to transformed coordinates	. 256
20.5	Vector operation and differentiation in generalized orthogonal coordinates	. 258
Chapter 21	User's Guide	261

21.1	Model setup	261
21.2	Preparation of input data files for execution	262
21.3	Restart file	264
21.4	Execution	268
21.5	Monitoring	270
21.6	MRI.COM eXecution Environment (MXE)	270
21.7	Appendix	271
Bibliography		277

Index

Chapter 1

OGCMs and MRI.COM

This chapter outlines the ocean general circulation models (OGCM) and the status of MRI.COM.

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.



Figure 1.1 Various oceanic phenomena in terms of 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.

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.

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 program structure of MRI.COM is presented along with a typical computational cost of each process in Figure 1.2.

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.

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. To rewrite the current vector-friendly program codes to adapt to this shift is immediately needed for some OGCMs. 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

1.4 Future of OGCMs and MRI.COM



Figure 1.2 Structure and typical computational cost of components of MRI.COM. Note that the sum of computational costs does not become 100% owing to round-off at each component.

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 20 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

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

where Ω is the rotation vector of the Earth, Φ is the geopotential, Φ_T is the tide producing potential, τ is the frictional stress tensor. In the rest of this chapter, the tide producing potential is neglected for brevity. Implementation of tide producing potential is thoroughly discussed in Chapter 6.

Mass conservation equation is

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

Equations for potential temperature and salinity are

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

and

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

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 10.2), 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 4.

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 (ρ **v**) that appears in the momentum equation (2.1) is replaced by a reference density (ρ_0). The momentum equation becomes

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

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

Chapter 2 Governing Equations

and

$$\frac{\partial S}{\partial t} + \mathbf{v} \cdot \nabla S = -\nabla \cdot \mathbf{F}^S + Q^S.$$
(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}_{μ} , \mathbf{e}_{ψ} , 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

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

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

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

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.. **F**_{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

$$2\Omega \times \mathbf{v} = (2\Omega_{\psi}w - 2\Omega_{r}v)\mathbf{e}_{\mu} + (2\Omega_{r}u - 2\Omega_{\mu}w)\mathbf{e}_{\psi} + (2\Omega_{\mu}v - 2\Omega_{\psi}u)\mathbf{e}_{r},$$
(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 v_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:

$$\frac{\partial u}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial (h_{\psi}uu)}{\partial \mu} + \frac{\partial (h_{\mu}vu)}{\partial \psi} \right\} + \frac{\partial (wu)}{\partial z} + \frac{v}{h_{\mu}h_{\psi}} \left(\frac{\partial h_{\mu}}{\partial \psi} u - \frac{\partial h_{\psi}}{\partial \mu} v \right) - fv$$

$$= -\frac{1}{\rho_{0}h_{\mu}} \frac{\partial p}{\partial \mu} + \frac{1}{\rho_{0}} (\nabla \cdot \tau_{\text{horizontal strain}})_{u} + \frac{\partial}{\partial z} \left(v_{V} \frac{\partial u}{\partial z} \right), \qquad (2.16)$$

$$\frac{\partial v}{\partial z} = 1 - \left(\frac{\partial (h_{\psi}uv)}{\partial \mu} - \frac{\partial (h_{\mu}vv)}{\partial \mu} \right) - \frac{\partial (wv)}{\partial \mu} - \frac{u}{\partial z} \left(\frac{\partial h_{\psi}}{\partial z} - \frac{\partial h_{\psi}}{\partial \mu} \right) + \frac{\partial (h_{\mu}vv)}{\partial z} \right) = \frac{\partial (wv)}{\partial z} - \frac{u}{\partial z} \left(\frac{\partial h_{\psi}}{\partial z} - \frac{\partial h_{\psi}}{\partial \mu} \right) + \frac{\partial (h_{\mu}vv)}{\partial z} + \frac{\partial (h_{\mu}vv)}{\partial$$

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

Continuity equation is written as:

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

The equations for potential temperature and salinity are written as:

$$\frac{\partial\theta}{\partial t} = -\frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial(h_{\psi}u\theta)}{\partial\mu} + \frac{\partial(h_{\mu}v\theta)}{\partial\psi} \right\} - \frac{\partial(w\theta)}{\partial z} - \nabla \cdot \mathbf{F}^{\theta} + Q^{\theta},$$
(2.19)
$$\frac{\partial S}{\partial t} = -\frac{1}{h_{\nu}h_{\psi}} \left\{ \frac{\partial(h_{\psi}uS)}{\partial\mu} + \frac{\partial(h_{\mu}vS)}{\partial\psi} \right\} - \frac{\partial(wS)}{\partial z} - \nabla \cdot \mathbf{F}^{S} + Q^{S}.$$
(2.20)

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

$$v_V \frac{\partial(u,v)}{\partial z}\Big|_{z=\eta} = \frac{(\tau^{\mu},\tau^{\psi})}{\rho_0}.$$
(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,

$$(F_{\rm surf}^{\mu}, F_{\rm surf}^{\psi}) = \frac{(\tau^{\mu}, \tau^{\psi})}{\rho_0 \Delta z_{\frac{1}{2}}},$$
(2.22)

where $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer, and τ^{μ} and τ^{ψ} are zonal and meridional stress, respectively.

Surface fresh water flux is assumed to have zero velocity.

Bottom (z = -H): Bottom friction $(\tau_h^{\mu} \text{ in zonal and } \tau_h^{\psi} \text{ in meridional direction})$ exists at the sea floor $(z = -H(\mu, \psi))$.

$$v_V \frac{\partial(u,v)}{\partial z}\Big|_{z=-H(\mu,\psi)} = -\frac{(\tau_b^{\mu},\tau_b^{\psi})}{\rho_0}.$$
(2.23)

In the model algorithm, this is treated as a body force to the bottom level (k = kbtm) velocity,

$$(F_{\text{bottom}}^{\mu}, F_{\text{bottom}}^{\psi}) = \frac{(\tau_b^{\mu}, \tau_b^{\psi})}{\rho_0 \Delta z_{\text{kbtm}} - \frac{1}{2}},$$
(2.24)

where $\Delta z_{\text{kbtm}-\frac{1}{2}}$ is the thickness of the bottom layer.

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:

$$\kappa_V \frac{\partial \theta}{\partial z}\Big|_{z=\eta} = F_{\text{surf}}^{\theta}, \tag{2.25}$$

$$\kappa_V \frac{\partial S}{\partial z}\Big|_{z=\eta} = F_{\text{surf}}^S,\tag{2.26}$$

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

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

$$\kappa_V \frac{\partial \theta}{\partial z}\Big|_{z=-H(\mu,\psi)} = F^{\theta}_{\text{bottom}}, \quad \kappa_V \frac{\partial S}{\partial z}\Big|_{z=-H(\mu,\psi)} = F^S_{\text{bottom}}, \tag{2.27}$$

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:

$$\frac{\partial \theta}{\partial n} = 0, \quad \frac{\partial S}{\partial n} = 0,$$
 (2.28)

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.

$$w = \frac{d\eta}{dt} - (P - E + R + I) = \frac{\partial\eta}{\partial t} + u\frac{1}{h_{\mu}}\frac{\partial\eta}{\partial\mu} + v\frac{1}{h_{\psi}}\frac{\partial\eta}{\partial\psi} - (P - E + R + I),$$
(2.29)

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:

$$w = -\left(u\frac{1}{h_{\mu}}\frac{\partial H}{\partial \mu} + v\frac{1}{h_{\psi}}\frac{\partial H}{\partial \psi}\right).$$
(2.30)

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),

$$\frac{\partial \eta}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial (h_{\psi}(H+\eta)\langle u \rangle)}{\partial \mu} + \frac{\partial (h_{\mu}(H+\eta)\langle v \rangle)}{\partial \psi} \right\} = P - E + R + I,$$
(2.31)

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:

$$z_s \equiv \frac{\partial z}{\partial s}\Big|_{\mu\psi t}.$$
(2.32)

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 in generalized vertical coordinates as follows:

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

$$= -z_s \frac{1}{\rho_0 h_\mu} \frac{\partial p}{\partial \mu} - z_s \frac{\rho}{\rho_0 h_\mu} \frac{\partial(g z)}{\partial \mu} + z_s \frac{1}{\rho_0} (\nabla \cdot \tau_{\text{horizontal strain}})_u + z_s \frac{1}{z_s} \frac{\partial}{\partial s} \left(\frac{v_V}{z_s} \frac{\partial u}{\partial s} \right), \quad (2.33)$$

$$\frac{\partial(z_{s}v)}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial(z_{s}h_{\psi}uv)}{\partial\mu} + \frac{\partial(z_{s}h_{\mu}vv)}{\partial\psi} \right\} + \frac{\partial(z_{s}\dot{s}v)}{\partial s} + z_{s}\frac{u}{h_{\mu}h_{\psi}} \left(\frac{\partial h_{\psi}}{\partial\mu}v - \frac{\partial h_{\mu}}{\partial\psi}u \right) + z_{s}fu$$
$$= -z_{s}\frac{1}{\rho_{0}h_{\psi}}\frac{\partial p}{\partial\psi} - z_{s}\frac{\rho}{\rho_{0}h_{\psi}}\frac{\partial(gz)}{\partial\psi} + z_{s}\frac{1}{\rho_{0}}(\nabla \cdot \tau_{\text{horizontal strain}})_{v} + z_{s}\frac{1}{z_{s}}\frac{\partial}{\partial s}\left(\frac{v_{v}}{z_{s}}\frac{\partial v}{\partial s}\right), \tag{2.34}$$

$$\rho \frac{\partial (gz)}{\partial s} + \frac{\partial p}{\partial s} = 0, \tag{2.35}$$

$$\frac{\partial z_s}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (z_s h_\psi u)}{\partial \mu} + \frac{\partial (z_s h_\mu v)}{\partial \psi} \right\} + \frac{\partial (z_s \dot{s})}{\partial s} = 0,$$
(2.36)

$$\frac{\partial(z_s\theta)}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial(z_s h_\psi u\theta)}{\partial \mu} + \frac{\partial(z_s h_\mu v\theta)}{\partial \psi} \right\} + \frac{\partial(z_s \dot{s}\theta)}{\partial s} = -z_s \nabla \cdot \mathbf{F}_\theta + z_s Q^\theta, \tag{2.37}$$

$$\frac{\partial(z_sS)}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial(z_s h_\psi uS)}{\partial \mu} + \frac{\partial(z_s h_\mu vS)}{\partial \psi} \right\} + \frac{\partial(z_s \dot{s}S)}{\partial s} = -z_s \nabla \cdot \mathbf{F}_S + z_s Q^S, \tag{2.38}$$

and

$$\rho = \rho(\theta, S, p). \tag{2.39}$$

Evolution equation for sea surface height is obtained by vertically integrating the continuity equation (2.36) and considering the boundary condition. It has the same form as in the depth coordinate system:

$$\frac{\partial(\eta+H)}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left[\frac{\partial(h_{\psi}U)}{\partial\mu} + \frac{\partial(h_{\mu}V)}{\partial\psi} \right] = P - E + R + I, \qquad (2.40)$$

where

$$(U,V) = \int_{s(z=-H)}^{s(z=\eta)} [z_s(u,v)] ds.$$
 (2.41)

2.1.6 z^* coordinate

a. Definition and boundary condition

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

$$z^{*} = \sigma(\mu, \psi, z, t) H(\mu, \psi) = \frac{z - \eta(\mu, \psi, t)}{H(\mu, \psi) + \eta(\mu, \psi, t)} H(\mu, \psi),$$
(2.42)

Chapter 2 Governing Equations

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

$$\frac{\partial z}{\partial z^*} = \frac{H + \eta}{H}.$$
(2.43)

The vertical velocity in this coordinate system is expressed as w^* . This has the following relation with the vertical velocity *w* of the depth coordinate *z*,

$$w^* \equiv D_t z^* = \frac{H}{H+\eta} \Big(w - \Big(1 + \frac{z^*}{H} \Big) D_t \eta + \frac{z^* \eta}{H^2} \mathbf{v}_h \cdot \nabla H \Big),$$
(2.44)

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

$$D_t \equiv \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla = \frac{\partial}{\partial t} + \frac{u}{h_\mu} \frac{\partial}{\partial \mu} + \frac{v}{h_\psi} \frac{\partial}{\partial \psi} + \frac{w}{z_s} \frac{\partial}{\partial s}.$$
(2.45)

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.

$$z = -H(\mu, \psi) \implies z^* = -H(\mu, \psi). \tag{2.46}$$

Thus the kinematic boundary condition at sloping bottom is

$$w_{z=-H} = -\mathbf{v}_h \cdot \nabla H \implies w_{z^*=-H}^* = -\mathbf{v}_h \cdot \nabla H, \tag{2.47}$$

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

$$z = \eta(x, y, t) \implies z^* = 0. \tag{2.48}$$

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

$$w_{z=\eta} = D_t \eta - (P - E + R + I) \implies w_{z^*=0}^* = -\frac{H}{H + \eta} (P - E + R + I).$$
(2.49)

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

The governing equations in z^* vertical coordinate are in the same form as equations (2.33) through (2.40), with z_s replaced by $z_{z^*} = \frac{H+\eta}{H}$.

b. Pressure gradient term

Horizontal momentum equations in generalized vertical coordinates (2.33), (2.34) 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

$$\rho = \rho_0 + \rho'(x, y, z^*, t). \tag{2.50}$$

Pressure is also separated in the same manner $(p = p_0 + p')$ and the hydrostatic relation (2.35) is separated into two equations

$$\partial_{z^*} p_0 = -g \Big(\frac{H+\eta}{H} \Big) \rho_0 \text{ and } p_0(z^* = 0) = 0,$$
 (2.51)

$$\partial_{z^*} p' = -g \Big(\frac{H+\eta}{H} \Big) \rho' \text{ and } p'(z^* = 0) = p_a,$$
 (2.52)

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

where p_a is atmospheric pressure[¶]. Specific expression for p_0 is obtained by integrating (2.51)

$$p_0(z^*) = p_0(x, y, z, t) = -g\rho_0 \frac{H+\eta}{H} z^* = g\rho_0(\eta - z) = g\rho_0\eta - \rho_0\Phi.$$
(2.53)

Rewriting pressure gradient term by using this, we have

$$\frac{1}{\rho_0} \nabla_{z^*} (p_0 + p') + \frac{\rho}{\rho_0} \nabla_{z^*} \Phi = g \nabla \eta - \nabla_{z^*} \Phi + \frac{1}{\rho_0} \nabla_{z^*} p' + \frac{\rho_0 + \rho'}{\rho_0} \nabla_{z^*} \Phi$$
(2.54)

$$= g \nabla \eta + \frac{1}{\rho_0} \nabla_{z^*} p' + \frac{\rho'}{\rho_0} \nabla_{z^*} \Phi.$$
 (2.55)

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

$$\partial_t \mathbf{v}_h + \mathbf{v}_h \cdot \nabla_{z^*} \mathbf{v}_h + w^* \partial_{z^*} \mathbf{v}_h + f \hat{\mathbf{z}} \times \mathbf{v}_h + g \nabla \eta + \frac{1}{\rho_0} \nabla_{z^*} p' + \frac{g \rho'}{\rho_0} \nabla_{z^*} z = \mathbf{F}$$
(2.56)

Perturbation pressure is obtained by integrating (2.52) as

$$p'(z^*) = p_a + g \int_{z^*}^0 \rho' z_{z^*} dz^* = p_a + g \int_{z(z^*)}^\eta \rho' dz.$$
(2.57)

Then (2.56) becomes

$$\partial_{t} \mathbf{v}_{h} + \mathbf{v}_{h} \cdot \nabla_{z^{*}} \mathbf{v}_{h} + w^{*} \partial_{z^{*}} \mathbf{v}_{h} + f \hat{\mathbf{z}} \times \mathbf{v}_{h}$$

$$+ \underbrace{\frac{1}{\rho_{0}} \nabla_{z^{*}} (p_{a} + g\rho_{0}\eta)}_{\text{fast}} + \underbrace{\frac{1}{\rho_{0}} \nabla_{z^{*}} \left(g \int_{z(z^{*})}^{\eta} \rho' dz\right) + \frac{g\rho'}{\rho_{0}} \nabla_{z^{*}} z}_{\text{slow}} = \mathbf{F}.$$
(2.58)

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,

$$\frac{\partial(z_s\alpha)}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial(h_\psi z_s u\alpha)}{\partial \mu} + \frac{\partial(h_\mu z_s v\alpha)}{\partial \psi} \right\} + \frac{\partial(z_s \dot{s}\alpha)}{\partial s}$$
(2.59)

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

Chapter 2 Governing Equations

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

$$\int_{s_{k}}^{s_{k-1}} \frac{\partial(z_{s}\alpha)}{\partial t} ds + \int_{s_{k}}^{s_{k-1}} \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial(h_{\psi}z_{s}u\alpha)}{\partial\mu} + \frac{\partial(h_{\mu}z_{s}v\alpha)}{\partial\psi} \right\} ds + \int_{s_{k}}^{s_{k-1}} \frac{\partial(z_{s}\dot{s}\alpha)}{\partial s} ds$$

$$= \frac{\partial}{\partial t} \left(\int_{s_{k}}^{s_{k-1}} (z_{s}\alpha) ds \right) + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial}{\partial\mu} \left(\int_{s_{k}}^{s_{k-1}} h_{\psi}z_{s}u\alpha ds \right) + \frac{\partial}{\partial\psi} \left(\int_{s_{k}}^{s_{k-1}} h_{\mu}z_{s}v\alpha ds \right) \right\}$$

$$- \left(z_{s} \frac{\partial s_{k-1}}{\partial t} + \frac{z_{s}u(s_{k-1})}{h_{\mu}} \frac{\partial s_{k-1}}{\partial\mu} + \frac{z_{s}v(s_{k-1})}{h_{\psi}} \frac{\partial s_{k-1}}{\partial\psi} - z_{s}\dot{s}(s_{k-1}) \right) \alpha(s_{k-1})$$

$$+ \left(z_{s} \frac{\partial s_{k}}{\partial t} + \frac{z_{s}u(s_{k})}{h_{\mu}} \frac{\partial s_{k}}{\partial\mu} + \frac{z_{s}v(s_{k})}{h_{\psi}} \frac{\partial s_{k}}{\partial\psi} - z_{s}\dot{s}(s_{k}) \right) \alpha(s_{k}). \tag{2.60}$$

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

$$\frac{\partial}{\partial t} \left(\Delta z \alpha \right)_{k-\frac{1}{2}} + \frac{1}{h_{\mu} h_{\psi}} \left\{ \frac{\partial}{\partial \mu} \left(h_{\psi} \Delta z u \alpha \right)_{k-\frac{1}{2}} + \frac{\partial}{\partial \psi} \left(h_{\mu} \Delta z v \alpha \right)_{k-\frac{1}{2}} \right\},$$
(2.61)

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

Using $\dot{s} \equiv D_t s$ and (2.45), the last two lines are reduced to the difference between vertical advective fluxes:

the last two lines of $(2.60) = z_s \dot{s}(s_{k-1})\alpha(s_{k-1}) - z_s \dot{s}(s_k)\alpha(s_k).$ (2.62)

For the sea surface $(k = 1; s_0 = 0)$ and the bottom $(k = kbtm; s_{kbtm} = -H)$, kinematic conditions (2.49) and (2.47) are used to give

the last two lines of
$$(2.60) = -(P - E + R + I)\alpha(0) - z_s \dot{s}(s_1)\alpha(s_1)$$
 (2.63)

at the surface and

the last two lines of
$$(2.60) = z_s \dot{s}(s_{kbtm-1})\alpha(s_{kbtm-1}) - 0$$
 (2.64)

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

$$\int_{s_{k}}^{s_{k-1}} \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial(h_{\psi}z_{s}F_{\mu})}{\partial\mu} + \frac{\partial(h_{\mu}z_{s}F_{\psi})}{\partial\psi} \right\} ds + \int_{s_{k}}^{s_{k-1}} \frac{\partial F_{s}}{\partial s} ds$$
$$= \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial}{\partial\mu} \left(h_{\psi}\Delta zF_{\mu} \right)_{k-\frac{1}{2}} + \frac{\partial}{\partial\psi} \left(h_{\mu}\Delta zF_{\psi} \right)_{k-\frac{1}{2}} \right\}$$
$$- \left(\frac{F_{\mu}(s)}{h_{\mu}} \frac{\partial s}{\partial\mu} + \frac{F_{\psi}(s)}{h_{\psi}} \frac{\partial s}{\partial\psi} - F_{s}(s) \right)_{k-1} + \left(\frac{F_{\mu}(s)}{h_{\mu}} \frac{\partial s}{\partial\mu} + \frac{F_{\psi}(s)}{h_{\psi}} \frac{\partial s}{\partial\psi} - F_{s}(s) \right)_{k}.$$
(2.65)

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.,

$$\int_{s_{k}}^{s_{k-1}} \frac{\partial(z_{s}\alpha)}{\partial t} ds + \int_{s_{k}}^{s_{k-1}} \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial(h_{\psi}z_{s}u\alpha)}{\partial\mu} + \frac{\partial(h_{\mu}z_{s}v\alpha)}{\partial\psi} \right\} ds + \int_{s_{k}}^{s_{k-1}} \frac{\partial(z_{s}\dot{s}\alpha)}{\partial s} ds \\
+ \int_{s_{k}}^{s_{k-1}} \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial(h_{\psi}z_{s}F_{\mu})}{\partial\mu} + \frac{\partial(h_{\mu}z_{s}F_{\psi})}{\partial\psi} \right\} ds + \int_{s_{k}}^{s_{k-1}} \frac{\partial F_{s}}{\partial s} ds \\
= \frac{\partial}{\partial t} \left(\Delta z\alpha \right)_{k-\frac{1}{2}} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial}{\partial\mu} \left(h_{\psi}\Delta zu\alpha \right)_{k-\frac{1}{2}} + \frac{\partial}{\partial\psi} \left(h_{\mu}\Delta zv\alpha \right)_{k-\frac{1}{2}} \right\} + z_{s}\dot{s}(s_{k-1})\alpha(s_{k-1}) - z_{s}\dot{s}(s_{k})\alpha(s_{k}) \\
+ \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial}{\partial\mu} \left(h_{\psi}\Delta zF_{\mu} \right)_{k-\frac{1}{2}} + \frac{\partial}{\partial\psi} \left(h_{\mu}\Delta zF_{\psi} \right)_{k-\frac{1}{2}} \right\} + F_{s}(s_{k-1}) - F_{s}(s_{k}).$$
(2.66)

The quantity

$$F_{\text{surf}}^{\alpha} = (P - E + R + I)\alpha(0) + \left(\frac{F_{\mu}(s_0)}{h_{\mu}}\frac{\partial s_0}{\partial \mu} + \frac{F_{\psi}(s_0)}{h_{\psi}}\frac{\partial s_0}{\partial \psi} - F_s(s_0)\right) = (P - E + R + I)\alpha(0) - F_s(0)$$
(2.67)

taken from (2.63) and (2.65) 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 (2.67) 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 Momentum equation

a. Mode splitting and explicit solution method for the barotropic mode

Here we consider to solve the momentum equation with hydrostatic and Boussinesq approximation. Equations are (2.33), (2.34), and (2.35). 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 (2.40).

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 6 for details.

b. Barotropic mode

If we put

$$U = \int_{-H}^{0} z_s u ds = \sum_{k=1}^{N} u_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}, \quad V = \int_{-H}^{0} z_s v ds = \sum_{k=1}^{N} v_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}, \quad (2.68)$$

and separate fast and slow modes of the pressure gradient as in (2.58), then the vertically summed semi-discrete momentum equations are

$$\frac{\partial U}{\partial t} - fV = -\frac{(\eta + H)}{\rho_0 h_u} \frac{\partial (p_a + g\rho_0 \eta)}{\partial u} + X,$$
(2.69)

$$\frac{\partial V}{\partial t} + fU = -\frac{(\eta + H)}{\rho_0 h_{\psi}} \frac{\partial (p_a + g\rho_0 \eta)}{\partial \psi} + Y, \qquad (2.70)$$

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

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

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

$$(2.72)$$

Chapter 2 Governing Equations

The vertically integrated continuity equation is given by

$$\frac{\partial \eta}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left(\frac{\partial (h_{\psi}U)}{\partial \mu} + \frac{\partial (h_{\mu}V)}{\partial \psi} \right) = (P - E + R + I).$$
(2.73)

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:

$$\langle U \rangle^{n+1} = U^n + \Delta t_{\rm cl} f \langle \langle V \rangle \rangle^{n+\frac{1}{2}} - \Delta t_{\rm cl} \Big\langle \Big\langle \frac{(\eta+H)}{\rho_0 h_\mu} \frac{\partial(p_a+\rho_0 g\eta)}{\partial \mu} \Big\rangle \Big\rangle^{n+\frac{1}{2}} + \Delta t_{\rm cl} X^n, \tag{2.74}$$

$$\langle V \rangle^{n+1} = V^n - \Delta t_{\rm cl} f \langle \langle U \rangle \rangle^{n+\frac{1}{2}} - \Delta t_{\rm cl} \left\langle \left\langle \frac{(\eta+H)}{\rho_0 h_{\psi}} \frac{\partial (p_a+\rho_0 g\eta)}{\partial \psi} \right\rangle \right\rangle^{n+\frac{1}{2}} + \Delta t_{\rm cl} Y^n, \tag{2.75}$$

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 6.

c. Baroclinic mode

To solve the baroclinic mode, we can omit to obtain absolute pressure by using the method described below.

v

Velocity is decomposed into a barotropic component and a baroclinic component as follows:

$$u = \hat{u} + \bar{u}, \tag{2.76}$$

$$=\hat{v}+\bar{v},\tag{2.77}$$

where \bar{u} and \bar{v} are barotropic components and \hat{u} and \hat{v} are baroclinic components.

We consider updating a new velocity (u', v') using a momentum equation where the fast pressure gradient term is dropped:

$$\frac{u_{k-\frac{1}{2}}^{'}\Delta z_{k-\frac{1}{2}}^{n+1} - u_{k-\frac{1}{2}}^{n}\Delta z_{k-\frac{1}{2}}^{n}}{\Delta t_{\rm cl}} = f[v_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + F_{\mu}^{n},$$
(2.78)

$$\frac{v_{k-\frac{1}{2}}^{'}\Delta z_{k-\frac{1}{2}}^{n+1} - v_{k-\frac{1}{2}}^{n}\Delta z_{k-\frac{1}{2}}^{n}}{\Delta t_{\rm cl}} = -f[u_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + F_{\psi}^{n},$$
(2.79)

where

$$F_{\mu}^{n} = -\nabla_{H} \cdot \left(\Delta z(u, v)u\right)_{k-\frac{1}{2}} - \left[\frac{v}{h_{\mu}h_{\psi}}\left(\frac{\partial h_{\mu}}{\partial \psi}u - \frac{\partial h_{\psi}}{\partial \mu}v\right)\right]_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}} - \left[\frac{1}{\rho_{0}}\frac{1}{h_{\mu}}\int_{s_{k-\frac{1}{2}}}^{0} z_{s}g\rho_{\mu}ds'\right]\Delta z_{k-\frac{1}{2}} - \frac{g}{\rho_{0}h_{\mu}}\sum_{k=1}^{N}[\rho'z_{\mu}]\Delta z_{k-\frac{1}{2}} + (\Delta zF_{\text{horz}}^{\mu})_{k-\frac{1}{2}} - F_{\text{vert}k-1}^{\mu} + F_{\text{vert}k}^{\mu},$$
(2.80)

$$F_{\psi}^{n} = -\nabla_{H} \cdot \left(\Delta z(u, v)v\right)_{k-\frac{1}{2}} + \left[\frac{u}{h_{\mu}h_{\psi}}\left(\frac{\partial h_{\mu}}{\partial \psi}u - \frac{\partial h_{\psi}}{\partial \mu}v\right)\right]_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}} - \left[\frac{1}{\rho_{0}}\frac{1}{h_{\psi}}\int_{s_{k-\frac{1}{2}}}^{0} z_{s}g\rho_{\psi}ds'\right]\Delta z_{k-\frac{1}{2}} - \frac{g}{\rho_{0}h_{\psi}}\sum_{k=1}^{N} [\rho' z_{\psi}]\Delta z_{k-\frac{1}{2}} + (\Delta z F_{\text{horz}}^{\psi})_{k-\frac{1}{2}} - F_{\text{vert}k-1}^{\psi} + F_{\text{vert}k}^{\psi}.$$
 (2.81)

Summing over the whole water column gives

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

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

 X^n and Y^n are removed by using the equations from the barotropic mode (2.74), (2.75) to give

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

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

$$\frac{\left(u_{k-\frac{1}{2}}^{'}-\overline{u^{'}}^{z}+\langle u\rangle^{n+1}\right)\Delta z_{k-\frac{1}{2}}^{n+1}-u_{k-\frac{1}{2}}^{n}\Delta z_{k-\frac{1}{2}}^{n}}{\Delta t_{cl}} = f\left[v_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}\right]^{n+\frac{1}{2}}-f\overline{[v]^{n+\frac{1}{2}}}^{z}\Delta z_{k-\frac{1}{2}}^{n+1}+f\langle\langle v\rangle\rangle^{n+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}^{n+1} -\frac{\Delta z_{k-\frac{1}{2}}^{n+1}}{\eta^{n+1}+H}\left\langle\left\langle\frac{(\eta+H)}{\rho_{0}h_{\mu}}\frac{\partial(p_{a}+\rho_{0}g\eta)}{\partial\mu}\right\rangle\right\rangle^{n+\frac{1}{2}}+F_{\mu}^{n} \qquad (2.86)$$

$$\frac{\left(v_{k-\frac{1}{2}}^{'}-\overline{v^{'}}^{z}+\langle v\rangle^{n+1}\right)\Delta z_{k-\frac{1}{2}}^{n+1}-v_{k-\frac{1}{2}}^{n}\Delta z_{k-\frac{1}{2}}^{n}}{\Delta t_{cl}} = -f\left[u_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}\right]^{n+\frac{1}{2}}+f\overline{[u]^{n+\frac{1}{2}}}^{z}\Delta z_{k-\frac{1}{2}}^{n+1}-f\langle\langle u\rangle\rangle^{n+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}^{n+1} -\frac{\Delta z_{k-\frac{1}{2}}^{n+1}}{\eta^{n+1}+H}\langle\langle\frac{(\eta+H)}{\rho_{0}h_{\psi}}\frac{\partial(p_{a}+\rho_{0}g\eta)}{\partial\psi}\rangle\rangle^{n+\frac{1}{2}}+F_{\psi}^{n},$$
(2.87)

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

Since $f[v]^{n+\frac{1}{2}^{z}} \Delta z_{k-\frac{1}{2}}^{n+1} \sim f\langle\langle v \rangle\rangle^{n+\frac{1}{2}} \Delta z_{k-\frac{1}{2}}^{n+1}$, $f[u]^{n+\frac{1}{2}^{z}} \Delta z_{k-\frac{1}{2}}^{n+1} \sim f\langle\langle u \rangle\rangle^{n+\frac{1}{2}} \Delta z_{k-\frac{1}{2}}^{n+1}$, 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 $u = \hat{u} + \overline{u}$ and $v = \hat{v} + \overline{v}$.

2.2.3 Continuity equation

The vertical component of velocity is obtained by vertically integrating the continuity equation (2.36) from top to bottom. By using a flux form (setting $\alpha = 1$ in the r.h.s. of (2.66)), the surface boundary condition (2.49) may be naturally included. The vertical integration for the *k*-th vertical level is performed as follows:

$$(z_{s}\dot{s})_{k} = (z_{s}\dot{s})_{k-1} + \left[\Delta s_{k-\frac{1}{2}}(\partial_{t}z_{s})_{k-\frac{1}{2}}\right] + \frac{1}{h_{\mu}h_{\psi}} \left\{\frac{\partial(h_{\psi}\Delta z_{k-\frac{1}{2}}u_{k-\frac{1}{2}})}{\partial\mu} + \frac{\partial(h_{\mu}\Delta z_{k-\frac{1}{2}}v_{k-\frac{1}{2}})}{\partial\psi}\right\},$$
(2.88)

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.

2.2.4 Temperature and salinity equation

We solve for potential temperature instead of *in situ* temperature, because the potential temperature is conserved through vertical movement.

a. A semi-discrete expression

The equation for potential temperature and salinity is an advection-diffusion equation (2.37) and (2.38). Its semi-discrete expression is as follows:

$$\frac{\partial}{\partial t}(\theta_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}) = -\nabla_H \cdot \left(\Delta z h_{\psi} u\theta, \ \Delta z h_{\mu} v\theta\right)_{k-\frac{1}{2}} - (z_s \dot{s}\theta)_{k-1} + (z_s \dot{s}\theta)_k - \nabla_H \cdot \left(\Delta z h_{\psi} F_{\mu}^{\theta}, \ \Delta z h_{\mu} F_{\psi}^{\theta}\right)_{k-\frac{1}{2}} - F_{s\ k-1}^{\theta} + F_{s\ k}^{\theta} + Q^{\theta} \Delta z_{k-\frac{1}{2}},$$
(2.89)

$$\frac{\partial}{\partial t} (S_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}) = -\nabla_H \cdot \left(\Delta z h_{\psi} u S, \ \Delta z h_{\mu} v S \right)_{k-\frac{1}{2}} - (z_s \dot{s} S)_{k-1} + (z_s \dot{s} S)_k - \nabla_H \cdot \left(\Delta z h_{\psi} F^S_{\mu}, \ \Delta z h_{\mu} F^S_{\psi} \right)_{k-\frac{1}{2}} - F^S_{s \ k-1} + F^S_{s \ k} + Q^S \Delta z_{k-\frac{1}{2}}.$$
(2.90)

Several options for discretizing each term on the r.h.s. are detailed in Chapters 8 through 10.

b. Treating the unstably stratified layer

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 10.2.

One might also choose to mix tracers by setting the local vertical diffusion coefficient to a large value such as $10\,000\,\text{cm}^2\,\text{s}^{-1}$ where stratification is unstable. In this case, the tracer equation should be solved using the partial implicit method, which is described in Section 19.5.

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 ($\sim 200 \text{ m/s}$) for the barotropic mode and that of internal gravity waves ($\sim a \text{ few m/s}$) 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

$$\alpha \frac{\partial u}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial (h_{\psi}uu)}{\partial \mu} + \frac{\partial (h_{\mu}vu)}{\partial \psi} \right\} + \frac{\partial (wu)}{\partial z} + \frac{v}{h_{\mu}h_{\psi}} \left(\frac{\partial h_{\mu}}{\partial \psi} u - \frac{\partial h_{\psi}}{\partial \mu} v \right) - fv$$
$$= -\frac{1}{\rho_0 h_{\mu}} \frac{\partial p}{\partial \mu} + \frac{1}{\rho_0} (\nabla \cdot \tau_{\text{horizontal strain}})_u + \frac{\partial}{\partial z} \left(v_V \frac{\partial u}{\partial z} \right), \tag{2.91}$$

$$\alpha \frac{\partial v}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial (h_{\psi}uv)}{\partial \mu} + \frac{\partial (h_{\mu}vv)}{\partial \psi} \right\} + \frac{\partial (wv)}{\partial z} + \frac{u}{h_{\mu}h_{\psi}} \left(\frac{\partial h_{\psi}}{\partial \mu}v - \frac{\partial h_{\mu}}{\partial \psi}u \right) + fu$$
$$= -\frac{1}{\rho_0 h_{\psi}} \frac{\partial p}{\partial \psi} + \frac{1}{\rho_0} (\nabla \cdot \tau_{\text{horizontal strain}})_v + \frac{\partial}{\partial z} \left(v_V \frac{\partial v}{\partial z} \right).$$
(2.92)

The modified temperature and salinity equations are given by

$$\gamma \frac{\partial \theta}{\partial t} = -\frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial (h_{\psi}u\theta)}{\partial \mu} + \frac{\partial (h_{\mu}v\theta)}{\partial \psi} \right\} - \frac{\partial (w\theta)}{\partial z} - \nabla \cdot \mathbf{F}^{\theta} + Q^{\theta}, \tag{2.93}$$

$$\gamma \frac{\partial S}{\partial t} = -\frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial (h_{\psi}uS)}{\partial \mu} + \frac{\partial (h_{\mu}vS)}{\partial \psi} \right\} - \frac{\partial (wS)}{\partial z} - \nabla \cdot \mathbf{F}^{S} + Q^{S}.$$
(2.94)

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:

$$\omega^{2} = \frac{f^{2}}{\alpha^{2}} + \left(\frac{gH_{n}}{\alpha}\right)(k^{2} + l^{2}).$$
(2.95)

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:

$$\omega = -\beta k \left[\alpha (k^2 + l^2) + \frac{f^2}{gH_n} \right]^{-1}.$$
(2.96)

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.

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

Table2.1 Physical constants used in the model
Chapter 3

Spatial grid arrangement

The model ocean domain is defined as a three-dimensional aggregate of rectangular grid cells limited by surfaces of constant values on model's logical coordinate system. Just above the bottom, vertical thickness of the cell can be locally changed (so called partial cell). The horizontal and vertical lengths of the cells are regarded as the horizontal and vertical grid sizes, respectively. In MRI.COM, the grid size can be varied spatially in each direction (variable grid size). Note that the introduction of z^* coordinate results in the temporally variable actual vertical grid sizes, though the logical grid size is fixed in time.

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 (\times), and tracers such as temperature and salinity, density, and sea-surface height (SSH) are defined at the four corners of the cell (\circ). 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.



Figure 3.1 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



Figure 3.2 Horizontal grid lattices in relation to topography. (a) Tracer lattice. (b) Velocity lattice. Land distribution (shade) is common for (a) and (b).

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 (\circ) and velocity (\times) are defined at just the mid-depth level of the cell, and the vertical mass fluxes $W(\Delta, \Box)$ are defined at the boundary of the cell. There are two kinds of W, one for the T-cell (W^T ; Δ) and another for the U-cell (W^U ; \Box). 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.



Figure 3.3 Vertical grid arrangement. (a) Placement and numbering of vertical grid. (b) Smooth bottom slope with partial bottom cells. (c) Stair-like bottom slope.

3.3 Indices and symbols

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 actual meridional distance is defined similarly:

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

The vertical distance is expressed by Δz . For a discretized grid cell, the horizontal area is expressed by either ΔS or 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 $k - \frac{1}{2}$ (Figure 3.3a). In some cases, the T-point and U-point levels may be numbered as k, with a notice.

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 and analytically for geographic coordinate grids.

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) ,

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

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_b1; Figure 3.4a) is approximated numerically as follows taking $\mu_1 = \mu(i), \mu_2 = \mu(i + \frac{1}{2}), \text{ and } \psi_1 = \psi(j)$:

$$\sum_{m=1}^{M} L \Big[\lambda \Big(\mu_1 + (m-1)\delta\mu, \psi_1 \Big), \phi \Big(\mu_1 + (m-1)\delta\mu, \psi_1 \Big), \lambda \Big(\mu_1 + m\delta\mu, \psi_1 \Big), \phi \Big(\mu_1 + m\delta\mu, \psi_1 \Big) \Big].$$
(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_b1; Figure 3.4b) surrounded by four points $(\mu(i), \psi(j))$, $(\mu(i + \frac{1}{2}), \psi(j))$, $(\mu(i + \frac{1}{2}), \psi(j))$, $(\mu(i + \frac{1}{2}), \psi(j + \frac{1}{2}))$, 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 ψ_2), calculated as:

$$\sum_{n=1}^{N} \sum_{m=1}^{M} L \Big[\lambda \Big(\mu_{1} + (m-1)\delta\mu, \psi_{1} + (n-\frac{1}{2})\delta\psi \Big), \qquad \phi \Big(\mu_{1} + (m-1)\delta\mu, \psi_{1} + (n-\frac{1}{2})\delta\psi \Big), \\ \lambda \Big(\mu_{1} + m\delta\mu, \psi_{1} + (n-\frac{1}{2})\delta\psi \Big), \qquad \phi \Big(\mu_{1} + m\delta\mu, \psi_{1} + (n-\frac{1}{2})\delta\psi \Big) \Big] \\ \times \\ L \Big[\lambda \Big(\mu_{1} + (m-\frac{1}{2})\delta\mu, \psi_{1} + (n-1)\delta\psi \Big), \qquad \phi \Big(\mu_{1} + (m-\frac{1}{2})\delta\mu, \psi_{1} + (n-1)\delta\psi \Big), \\ \lambda \Big(\mu_{1} + (m-\frac{1}{2})\delta\mu, \psi_{1} + n\delta\psi \Big), \qquad \phi \Big(\mu_{1} + (m-\frac{1}{2})\delta\mu, \psi_{1} + n\delta\psi \Big) \Big].$$
(3.4)

As depicted in Figure 3.4, $(a_bl)_{i,j}$ is the area of the lower left quarter of the central U-point. Those for the lower right $(a_br)_{i,j}$, upper left $(a_tl)_{i,j}$, and upper right $(a_tr)_{i,j}$ quarters are obtained similarly.

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

$$(areau)_{i,j} = (a_bl)_{i,j} + (a_br)_{i,j} + (a_tl)_{i,j} + (a_tr)_{i,j},$$
 (3.5)

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

$$(\operatorname{areat})_{i,j} = (a_bl)_{i,j} + (a_br)_{i-1,j} + (a_tl)_{i,j-1} + (a_tr)_{i-1,j-1}.$$
 (3.6)

Following the conventions for indexing introduced in Section 3.3, the above equations are expressed in later chapters as follows:

$$(\operatorname{areau})_{i+\frac{1}{2},j+\frac{1}{2}} = (\operatorname{a_bl})_{i+\frac{1}{2},j+\frac{1}{2}} + (\operatorname{a_br})_{i+\frac{1}{2},j+\frac{1}{2}} + (\operatorname{a_tl})_{i+\frac{1}{2},j+\frac{1}{2}} + (\operatorname{a_tr})_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{3.7}$$

$$(\operatorname{areat})_{i,j} = (a_b l)_{i+\frac{1}{2},j+\frac{1}{2}} + (a_b r)_{i-\frac{1}{2},j+\frac{1}{2}} + (a_t l)_{i+\frac{1}{2},j-\frac{1}{2}} + (a_t r)_{i-\frac{1}{2},j-\frac{1}{2}}.$$
(3.8)

3.4.2 Geographic coordinates

For grids in the geographic coordinate system, we use more precise and computationally lighter analytical solutions. Let us examine the situation of T-cell divided in quarters (Figure 3.4). 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 3.4b, 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.

$$(\operatorname{anhft})_{i,j} = \int_{\phi}^{\phi + \frac{\Delta\phi}{2}} \Delta sad\phi = \frac{a^{2}\Delta\lambda}{2} \int_{\phi}^{\phi + \frac{\Delta\phi}{2}} \cos\phi d\phi = \frac{a^{2}\Delta\lambda}{2} \left\{ \sin\left(\phi + \frac{\Delta\phi}{2}\right) - \sin\phi \right\}$$
$$= a^{2}\Delta\lambda\cos\left(\phi + \frac{\Delta\phi}{4}\right)\sin\frac{\Delta\phi}{4}$$
$$= a^{2}\Delta\lambda\left(\cos\phi\cos\frac{\Delta\phi}{4} - \sin\phi\sin\frac{\Delta\phi}{4}\right)\sin\frac{\Delta\phi}{4}$$
$$= a^{2}\Delta\lambda\cos\phi\cos\frac{\Delta\phi}{4}\sin\frac{\Delta\phi}{4}\left(1 - \tan\phi\tan\frac{\Delta\phi}{4}\right)$$
$$= \frac{a^{2}}{2}\Delta\lambda\cos\phi\sin\frac{\Delta\phi}{2}\left(1 - \tan\phi\tan\frac{\Delta\phi}{4}\right).$$
(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:

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

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

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

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

(a)



Figure 3.4 Variables that define a grid unit. (a) Distance. (b) Area. Grid indices (i, j) follow the convention described in Section 3.3.

At the north pole:

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

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

At the south pole:

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

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

In our model

$$(a_bl)_{i,j} = (anhft)_{i,j},$$
 $(a_br)_{i,j} = (anhft)_{i+1,j},$
 $(a_tl)_{i,j} = (ashft)_{i,j+1},$ $(a_tr)_{i,j} = (ashft)_{i+1,j+1},$

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

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

If the sea level variation $\eta_{i,j}$ is known at a T-point, the thickness of four divided cells that comprise a T-cell is determined as follows:

$$(dzu_bl)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = (\eta_{i,j} + H_{i+\frac{1}{2},j+\frac{1}{2}}) \frac{(dzu_cnst)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}}{H_{i+\frac{1}{2},j+\frac{1}{2}}}$$
(3.18)

$$(\mathsf{dzu_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} = (\eta_{i,j} + H_{i+\frac{1}{2},j-\frac{1}{2}}) \frac{(\mathsf{dzu_cnst})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{H_{i+\frac{1}{2},j-\frac{1}{2}}}$$
(3.19)

$$(dzu_br)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = (\eta_{i,j} + H_{i-\frac{1}{2},j+\frac{1}{2}}) \frac{(dzu_cnst)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}}{H_{i-\frac{1}{2},i+\frac{1}{2}}}$$
(3.20)

$$(dzu_tr)_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} = (\eta_{i,j} + H_{i-\frac{1}{2},j-\frac{1}{2}}) \frac{(dzu_cnst)_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{H_{i-\frac{1}{2},j-\frac{1}{2}}},$$
(3.21)

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

$$(volu_bl)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = (a_bl)_{i+\frac{1}{2},j+\frac{1}{2}} (dzu_bl)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}$$
(3.22)

$$(volu_tl)_{i+\frac{1}{2}, j-\frac{1}{2}, k-\frac{1}{2}} = (a_tl)_{i+\frac{1}{2}, j-\frac{1}{2}} (dzu_tl)_{i+\frac{1}{2}, j-\frac{1}{2}, k-\frac{1}{2}}$$
(3.23)

$$(volu_br)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = (a_br)_{i-\frac{1}{2},j+\frac{1}{2}} (dzu_br)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}$$
(3.24)

$$(volu_tr)_{i-\frac{1}{2}, j-\frac{1}{2}, k-\frac{1}{2}} = (a_tr)_{i-\frac{1}{2}, j-\frac{1}{2}} (dzu_tr)_{i-\frac{1}{2}, j-\frac{1}{2}, k-\frac{1}{2}}.$$
(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 3.5 illustrates the procedure.



Figure 3.5 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.

Chapter 3 Spatial grid arrangement

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,

$$(\text{volu})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = (\text{volu_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\text{volu_br})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\text{volu_tr})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\text{volu_tr})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}.$$
(3.26)

Then the thickness is computed by

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

$$(volt)_{i,j,k-\frac{1}{2}} = (volu_bl)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (volu_tl)_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + (volu_br)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (volu_tr)_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}.$$
(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,

$$(dzt_cnst)_{i,j,k-\frac{1}{2}} = max((dzu_cnst)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}, (dzu_cnst)_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}, (dzu_cnst)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}, (dzu_cnst)_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}).$$
(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,

$$(dpu_bl)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = -\frac{H_{i+\frac{1}{2},j+\frac{1}{2}} - (dpu_cnst)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}}{H_{i+\frac{1}{2},j+\frac{1}{2}}}\eta_{i,j}$$
(3.30)

$$(dpu_tl)_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} = -\frac{H_{i+\frac{1}{2},j-\frac{1}{2}} - (dpu_cnst)_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{H_{i+\frac{1}{2},j-\frac{1}{2}}}\eta_{i,j}$$
(3.31)

$$(dpu_br)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = -\frac{H_{i-\frac{1}{2},j+\frac{1}{2}} - (dpu_cnst)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}}{H_{i-\frac{1}{2},j+\frac{1}{2}}}\eta_{i,j}$$
(3.32)

$$(dpu_tr)_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} = -\frac{H_{i-\frac{1}{2},j-\frac{1}{2}} - (dpu_trst)_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{H_{i-\frac{1}{2},j-\frac{1}{2}}}\eta_{i,j}$$
(3.33)

and for the bottom,

$$(depu_bl)_{i+\frac{1}{2},j+\frac{1}{2},k} = -\frac{H_{i+\frac{1}{2},j+\frac{1}{2}} - (depu_cnst)_{i+\frac{1}{2},j+\frac{1}{2},k}}{H_{i+\frac{1}{2},j+\frac{1}{2}}}\eta_{i,j}$$
(3.34)

$$(depu_tl)_{i+\frac{1}{2},j-\frac{1}{2},k} = -\frac{H_{i+\frac{1}{2},j-\frac{1}{2}} - (depu_cnst)_{i+\frac{1}{2},j-\frac{1}{2},k}}{H_{i+\frac{1}{2},j-\frac{1}{2}}}\eta_{i,j}$$
(3.35)

$$(\text{depu_br})_{i-\frac{1}{2},j+\frac{1}{2},k} = -\frac{H_{i-\frac{1}{2},j+\frac{1}{2}} - (\text{depu_cnst})_{i-\frac{1}{2},j+\frac{1}{2},k}}{H_{i-\frac{1}{2},j+\frac{1}{2}}}\eta_{i,j}$$
(3.36)

$$(depu_tr)_{i-\frac{1}{2},j-\frac{1}{2},k} = -\frac{H_{i-\frac{1}{2},j-\frac{1}{2}} - (depu_tcnst)_{i-\frac{1}{2},j-\frac{1}{2},k}}{H_{i-\frac{1}{2},j-\frac{1}{2}}}\eta_{i,j},$$
(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,

$$(dpt)_{i,j,k-\frac{1}{2}} = \left\{ (a_bl)_{i+\frac{1}{2},j+\frac{1}{2}} (dpu_bl)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (a_tl)_{i+\frac{1}{2},j-\frac{1}{2}} (dpu_tl)_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + (a_br)_{i-\frac{1}{2},j+\frac{1}{2}} (dpu_br)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (a_tr)_{i-\frac{1}{2},j-\frac{1}{2}} (dpu_tr)_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \right\} / (areat)_{i,j,k+\frac{1}{2}}$$
(3.38)

and for the bottom,

$$\begin{aligned} (dept)_{i+\frac{1}{2},j+\frac{1}{2},k} &= \left\{ (a_bl)_{i+\frac{1}{2},j+\frac{1}{2}} (depu_bl)_{i+\frac{1}{2},j+\frac{1}{2},k} + (a_tl)_{i+\frac{1}{2},j-\frac{1}{2}} (depu_tl)_{i+\frac{1}{2},j-\frac{1}{2},k} \\ &+ (a_br)_{i-\frac{1}{2},j+\frac{1}{2}} (depu_br)_{i-\frac{1}{2},j+\frac{1}{2},k} + (a_tr)_{i-\frac{1}{2},j-\frac{1}{2}} (depu_tr)_{i-\frac{1}{2},j-\frac{1}{2},k} \right\} / (areat)_{i,j,k+\frac{1}{2}}. \end{aligned}$$
(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 $(areat)_{i,j,k+\frac{1}{2}}$ instead of $(areat)_{i,j,k-\frac{1}{2}}$.

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

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

3.6 Usage

3.6.1 Choice of 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 3.1).

Table3.1 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 3.2). See Chapter 20 for details.

Chapter 3 Spatial grid arrangement

Table3.2 Namelist nml_poles for TRIPOLAR or JOT option. See also Figure 3.6b 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	specification is required
		singular points	
north_pole_lat	degree	geographical latitude of one of the two new	specification is required
		singular points	
south_pole_lon	degree	geographical longitude of the other of the two	specification is required
		new singular points	
south_pole_lat	degree	geographical latitude of the other of the two	specification is required
		new singular points	

3.6.2 Definition of model domain

To determine the model domain, the region of interest must be first filled with U-cells, where sea floor depths are given. This is the lightly shaded region of Figure 3.6, which is called the core region. For a closed basin model, this core region must be surrounded by land cells (Figure 3.6a). For a model domain with cyclic boundary condition (Figure 3.6b), two halo U-cells must be added to both western and eastern sides. When a zonally cyclic condition is used, CYCLIC option must be specified in configure.in. For a tri-pole model with TRIPOLAR or JOT option, three U-cells must be added as halos to the northern end of the core region. The total number of grid points (imut and jmut) must include halo cells.

The total numbers of grid points in the three directions (imut, jmut, and km) must be given to configure.in. 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 3.3). 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 3.6c). Note that they can vary only in the direction of their own axis.

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 3.4) 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.

variable name	units	description	usage
lon west end of core	model longitude	the longitude of the western end	specification is required
	in degree	of model's main region	
lat_south_end_of_core	model latitude in	the latitude of the southern end	specification is required
	degree	of model's main region	
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	If either X or Y axis grid spacing
		given by this file	is not uniform, prepare this file.

Table3.3 Namelist nml_horz_grid

- Format of vertical grid spacing data (file_dz_cm). —

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

Table3.4 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

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 3.5).

Table3.5	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 3.4 for positions) is read by the model at run time as follows.



Figure 3.6 Illustration to explain how to determine the model domain and how to give information to the model. Schematic of a model with (a) closed domain, (b) 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 halos. (c) Definition of the most 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 (a) and (b).

3.6 Usage

```
- 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)
```

Part II

Diagnostic Equations

Chapter 4 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.

4.1 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):

$$\rho_w(t) = 999.842594 + 6.793952 \times 10^{-2}t - 9.095290 \times 10^{-3}t^2$$

$$+ 1.001685 \times 10^{-4}t^3 - 1.120083 \times 10^{-6}t^4 + 6.536332 \times 10^{-9}t^5.$$
(4.1)

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

$$\rho_{\text{surf}} = \rho_w$$

$$+ (0.824493 - 4.0899 \times 10^{-3}t + 7.6438 \times 10^{-5}t^2 - 8.2467 \times 10^{-7}t^3 + 5.3875 \times 10^{-9}t^4)S$$

$$+ (-5.72466 \times 10^{-3} + 1.0227 \times 10^{-4}t - 1.6546 \times 10^{-6}t^2)S^{\frac{3}{2}}$$

$$+ 4.8314 \times 10^{-4}S^2.$$

$$(4.2)$$

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

$$\rho = \rho_{\text{surf}} / (1 - P/K), \tag{4.3}$$

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

$$K_w = 19652.21 + 1.484206 \times 10^2 t - 2.327105t^2 + 1.360477 \times 10^{-2}t^3 - 5.155288 \times 10^{-5}t^4.$$
(4.4)

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

$$K_0 = K_w + (54.6746 - 0.603459t + 1.09987 \times 10^{-2}t^2 - 6.1670 \times 10^{-5}t^3)S$$

$$+ (7.944 \times 10^{-2} + 1.6483 \times 10^{-2}t - 5.3009 \times 10^{-4}t^2)S^{\frac{3}{2}},$$
(4.5)

^{*} 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.

a_0	$+3.5803 \times 10^{-5}$	b_0	$+1.8932 \times 10^{-6}$	<i>c</i> 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}	<i>c</i> ₁	-6.7795×10^{-10}	d_1	$+2.7759 \times 10^{-12}$
a_2	-6.8360×10^{-8}			<i>c</i> ₂	$+8.7330 \times 10^{-12}$	e_0	-4.6206×10^{-13}
<i>a</i> ₃	$+6.6228 \times 10^{-10}$			<i>c</i> ₃	-5.4481×10^{-14}	e_1	$+1.8676 \times 10^{-14}$
						e_2	-2.1687×10^{-16}

Table4.1 Coefficients for the lapse rate of sea water (Eq. 4.8).

and the value at pressure P is given by

 $K = K_0$ $+ P (3.239908 + 1.43713 \times 10^{-3}t + 1.16092 \times 10^{-4}t^2 - 5.77905 \times 10^{-7}t^3)$ $+ P (2.2838 \times 10^{-3} - 1.0981 \times 10^{-5}t - 1.6078 \times 10^{-6}t^2) S$ $+ P (1.91075 \times 10^{-4}) S^{\frac{3}{2}}$ $+ P^2 (8.50935 \times 10^{-5} - 6.12293 \times 10^{-6}t + 5.2787 \times 10^{-8}t^2)$ $+ P^2 (-9.9348 \times 10^{-7} + 2.0816 \times 10^{-8}t + 9.1697 \times 10^{-10}t^2) S.$ (4.6)

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 $\Gamma(t, S, P)$:

$$t(\theta_0, S, P) = \theta_0 + \int_{P_0}^{P} \Gamma(t, S, P') dP'.$$
(4.7)

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

$$\Gamma(t, S, P) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$+ (b_0 + b_1 t)(S - 35)$$

$$+ \{c_0 + c_1 t + c_2 t^2 + c_3 t^3 + (d_0 + d_1 t)(S - 35)\}P$$

$$+ (e_0 + e_1 t + e_2 t^2)P^2.$$
(4.8)

Coefficients are given on Table 4.1.

4.2 An equation of state used by MRI.COM

4.2.1 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 ($-2 \le \theta \le 40$ °C, $-2 \le S \le 42$ pss, and $0 \le P \le 1000$ bar) than the previous version.

First, density is calculated at the sea surface (potential density or σ_{θ}) using equations (4.1) and (4.2). 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

$$K(\theta, S, P) = e_1(P) + e_2(P)\theta + e_3(P)\theta^2 + e_4(P)\theta^3 + e_5(P)\theta^4$$

$$+ S(f_1(P) + f_2(P)\theta + f_3(P)\theta^2 + f_4(P)\theta^3)$$

$$+ (|S|)^{\frac{3}{2}}(f_5(P) + f_6(P)\theta + f_7(P)\theta^2),$$
(4.9)

Chapter 4 Equation of State

where

$$e_{1}(P) = ec_{1} + (gc_{1} + hc_{1}P)P, \quad f_{1}(P) = fc_{1} + (gc_{5} + hc_{4}P)P,$$
(4.10)

$$e_{2}(P) = ec_{2} + (gc_{2} + hc_{2}P)P, \quad f_{2}(P) = fc_{2} + (gc_{6} + hc_{5}P)P,$$
(4.10)

$$e_{3}(P) = ec_{3} + (gc_{3} + hc_{3}P)P, \quad f_{3}(P) = fc_{3} + (gc_{7} + hc_{6}P)P,$$
(4.10)

$$e_{4}(P) = ec_{4} + gc_{4}P, \quad f_{4}(P) = fc_{4},$$
(5.10)

$$e_{5}(P) = ec_{5}, \quad f_{5}(P) = fc_{5} + gc_{8}P,$$
(6.10)

$$f_{6}(P) = fc_{6},$$
(7.10)

$$f_{7}(P) = fc_{7}.$$

The set of coefficients in the above equation is computed using a least square fit and listed on Table 4.2. 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 $1.0 \,^{\circ}$ C and $1.0 \,^{\circ}$ C and $0.1 \,^{\circ}$ S for $-2 \le \theta \le 10 \,^{\circ}$ C and $-2 \le S \le 10 \,^{\circ}$ S, respectively. Using $151 \times 151 \times 101$ combinations of the above range of potential temperature, salinity, and pressure, *in situ* temperature is first computed using (4.7). 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 4.2.

Table 4.2 Coefficients for the equation of state of sea water. See equation (4.10).

ec_1	19659.35	fc_1	52.85624	gc_1	3.185918	hc_1	2.111102×10^{-4}
ec_2	144.5863	fc_2	-3.128126×10^{-1}	gc_2	2.189412×10^{-2}	hc_2	-1.196438×10^{-5}
ec ₃	-1.722523	fc_3	6.456036×10^{-3}	gc ₃	-2.823685×10^{-4}	hc_3	1.364330×10^{-7}
ec_4	1.019238×10^{-2}	fc_4	-5.370396×10^{-5}	gc_4	1.715739×10^{-6}	hc_4	-2.048755×10^{-6}
ec5	-4.768276×10^{-5}	fc_5	3.884013×10^{-1}	gc_5	6.703377×10^{-3}	hc_5	6.375979×10^{-8}
		fc_6	9.116446×10^{-3}	gc ₆	-1.839953×10^{-4}	hc_6	5.240967×10^{-10}
		fc_7	-4.628163×10^{-4}	gc ₇	1.912264×10^{-7}		
				gc_8	1.477291×10^{-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 4.1). This maximum error is less than that $(3 \times 10^{-3}$ 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 \approx 40 pss or S \approx 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.

4.2.2 Implementation

In MRI.COM, *in situ* density is computed by (4.3) using (4.2) for ρ_{surf} and (4.9) 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:

$$10^6 \overline{P_{k-\frac{1}{2}}} = \rho_0 g z_{k-\frac{1}{2}},\tag{4.11}$$

$$10^6 \overline{P_k} = \rho_0 g z_k, \tag{4.12}$$

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 10⁶ 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:



Figure 4.1 Deviation from the UNESCO equation for the old (version 3 and earlier) and new (version 4) set of coefficients. Unit is 10^{-3} 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.

$$10^{6}\overline{P_{\frac{1}{2}}} = \frac{1}{2}g\Delta z_{\frac{1}{2}}\overline{\rho_{\frac{1}{2}}}^{xy},$$
(4.13)

$$10^{6}\overline{P_{k-\frac{1}{2}}} = 10^{6}P_{\frac{1}{2}} + g\sum_{l=2}^{k}\Delta z_{l-1}\frac{\overline{\rho_{l-\frac{3}{2}}}^{xy} + \overline{\rho_{l-\frac{1}{2}}}^{xy}}{2} \quad (k \ge 2),$$
(4.14)

$$10^{6}\overline{P_{k}} = g \sum_{k=1}^{k} \Delta z_{k-\frac{1}{2}} \overline{\rho_{k-\frac{1}{2}}}^{xy}.$$
(4.15)

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 4.3) and should be stored in a restart file to be used in the next run (Table 4.4). 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 $(\overline{\rho})$, 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:

```
Chapter 4 Equation of State
```

```
- 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
```

Table4.3 namelist nml_calpp

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

Table4.4 namelist nml_density_run

variable name	units	description	usage
l_rst_density_in	logical	.true. : Read restart files specified by	default = 1_rst_in of
		nmlrs_density for the initial condition.	nml_run_ini_state,
		.false. : Start condition is calculated by the 3-D	see Table 21.5.
		density field of the initial state.	

Chapter 5

Continuity equation

5.1 Introduction

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 (2.88) differs for the T-cell and U-cell (Figure 5.1). In MRI.COM, the mass continuity for the T-cell is primary and that for the U-cell is derived from the former by an averaging operation. 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).

5.2 Finite difference expression for the T-cell continuity equation

Finite difference expressions of the free surface equation (2.40),

$$\frac{\partial(\eta+H)}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left[\frac{\partial(h_{\psi}U)}{\partial\mu} + \frac{\partial(h_{\mu}V)}{\partial\psi} \right] = P - E + R + I,$$
(5.1)

the continuity equation (2.36),

$$\frac{\partial z_s}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (z_s h_\psi u)}{\partial \mu} + \frac{\partial (z_s h_\mu v)}{\partial \psi} \right\} + \frac{\partial (z_s \dot{s})}{\partial s} = 0, \tag{5.2}$$

and the tracer equations (2.37, 2.38)

$$\frac{\partial(z_s\theta)}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial(z_s h_\psi u\theta)}{\partial \mu} + \frac{\partial(z_s h_\mu v\theta)}{\partial \psi} \right\} + \frac{\partial(z_s \dot{s}\theta)}{\partial s} = -z_s \nabla \cdot \mathbf{F}_\theta + z_s Q_\theta, \tag{5.3}$$

$$\frac{\partial(z_sS)}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial(z_sh_{\psi}uS)}{\partial \mu} + \frac{\partial(z_sh_{\mu}vS)}{\partial \psi} \right\} + \frac{\partial(z_s\dot{s}S)}{\partial s} = -z_s\nabla\cdot\mathbf{F}_S + z_sQ_S, \tag{5.4}$$

must be mutually consistent in order to keep sign-definiteness of tracers. Among them, the continuity equation is fundamental. The continuity equation is used to diagnostically obtain vertical velocity \dot{s} . Actually, in MRI.COM, vertical velocity is not computed but vertical transport (wlwl) is extensively used. Vertical transport is mathematically expressed as $W^T \equiv z_s \dot{s} \Delta A^T(s)$, where $\Delta A^T(s)$ is the horizontal area of a T-cell^{*}. To obtain W^T , we horizontally integrate the semi discrete expression for the continuity equation (2.88) using the surface boundary condition for w^* .

Using the surface boundary condition for the vertical velocity in z^* coordinate (2.49), we have

$$W_{z^*=0}^T = z_s \dot{s}_{s=0} \Delta A_{s=0}^T = \frac{H+\eta}{H} w_{z^*=0}^* \Delta A_{z^*=0}^T = -(P-E+R+I) \Delta A_{z^*=0}^T.$$
(5.5)

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.

We now integrate the semi-discrete expression for the continuity equation (2.88) over the oceanic part of the vertically *k*-th T-cell. The integral over a T-cell of the tendency term of the scaling factor (2nd term on r.h.s.) is expressed as a

^{*} 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

5.2 Finite difference expression for the T-cell continuity equation

multiple of the average cross sectional area $(\Delta A_{i,j,k-\frac{1}{2}}^T)$ and the average thickness for the *k*-th vertical cell in z^* coordinate $(\Delta s_{i,j,k-\frac{1}{2}})$:

$$W_{i,j,k}^{T} = W_{i,j,k-1}^{T} + \left[\Delta A_{i,j,k-\frac{1}{2}}^{T} \Delta s_{i,j,k-\frac{1}{2}}(\partial_{t} z_{s})_{i,j} \right] \\ + \frac{(dzu \cdot u)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (dzu \cdot u)_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{2} \Delta y_{i+\frac{1}{2},j} - \frac{(dzu \cdot u)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (dzu \cdot u)_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{2} \Delta y_{i-\frac{1}{2},j} \\ + \frac{(dzu \cdot v)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (dzu \cdot v)_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}}{2} \Delta x_{i,j+\frac{1}{2}} - \frac{(dzu \cdot v)_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + (dzu \cdot v)_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}{2} \Delta x_{i,j-\frac{1}{2}}, (5.6)$$

where it is assumed that the actual thickness of a U-cell (dzu) also works as a land-sea mask. The second term on the right hand side is regarded as an actual temporal variation of the volume of a T-cell, that is,

$$\Delta A_{i,j,k-\frac{1}{2}}^{T} \Delta s_{i,j,k-\frac{1}{2}} (\partial_t z_s)_{i,j} = \partial_t (\operatorname{volt})_{i,j,k-\frac{1}{2}},$$
(5.7)

where equation (3.28) is used. Conversely, by using W^T calculated from (5.6), calculation of the volume flux divergence for a T-cell results in the volume change rate of the T-cell.

To check consistency with the free surface equation, we insert (5.7) into (5.6) and then sum them up from the surface to the bottom to have

$$W_{i,j,ktbtm}^{T} = 0 = -(P - E + R + I)\Delta A_{i,j,\frac{1}{2}}^{T} + \frac{\partial (V_{column}^{T})_{i,j}}{\partial t} + \frac{U_{i+\frac{1}{2},j+\frac{1}{2}} + U_{i+\frac{1}{2},j-\frac{1}{2}}}{2}\Delta y_{i+\frac{1}{2},j} - \frac{U_{i-\frac{1}{2},j+\frac{1}{2}} + U_{i-\frac{1}{2},j-\frac{1}{2}}}{2}\Delta y_{i-\frac{1}{2},j} + \frac{V_{i+\frac{1}{2},j+\frac{1}{2}} + V_{i-\frac{1}{2},j+\frac{1}{2}}}{2}\Delta x_{i,j+\frac{1}{2}} - \frac{V_{i+\frac{1}{2},j-\frac{1}{2}} + V_{i-\frac{1}{2},j-\frac{1}{2}}}{2}\Delta x_{i,j-\frac{1}{2}}.$$
 (5.8)

Because the tendency of the volume of a T-point water column V_{column}^T is determined by the undulation of sea level, we have,

$$\frac{\partial (V_{\text{column}}^{I})_{i,j}}{\partial t} = (\operatorname{areat})_{i,j,\frac{1}{2}} \frac{\partial \eta_{i,j}}{\partial t} = (ws)_{i,j} + (\operatorname{transport}_w flux)_{i,j},$$
(5.9)

where

$$(\mathbf{ws})_{i,j} = -\left(\frac{U_{i+\frac{1}{2},j+\frac{1}{2}} + U_{i+\frac{1}{2},j-\frac{1}{2}}}{2}\Delta y_{i+\frac{1}{2},j} - \frac{U_{i-\frac{1}{2},j+\frac{1}{2}} + U_{i-\frac{1}{2},j-\frac{1}{2}}}{2}\Delta y_{i-\frac{1}{2},j} + \frac{V_{i+\frac{1}{2},j+\frac{1}{2}} + V_{i-\frac{1}{2},j+\frac{1}{2}}}{2}\Delta x_{i,j+\frac{1}{2}} - \frac{V_{i+\frac{1}{2},j-\frac{1}{2}} + V_{i-\frac{1}{2},j-\frac{1}{2}}}{2}\Delta x_{i,j-\frac{1}{2}}}\right)$$
(5.10)

and

$$(transport_wflux)_{i,j} = (P - E + R + I)_{i,j} \Delta A_{i,j,\frac{1}{2}}^T.$$
 (5.11)

It is confirmed that (5.8) has an integrated form of the free surface equation (2.40).

As a preparation for explaining the U-cell continuity equation, the finite difference expression of the continuity equation for the vertically k-th T-cell (5.6) is rewritten in a concise form as follows, by defining the mass fluxes passing through each side of the grid cell (Figure 5.1):

$$MC_{i,j,k-\frac{1}{2}}^{T} \equiv U_{i-\frac{1}{2},j}^{T} - U_{i+\frac{1}{2},j}^{T} + V_{i,j-\frac{1}{2}}^{T} - V_{i,j+\frac{1}{2}}^{T} + W_{i,j,k}^{T} - W_{i,j,k-1}^{T}$$

= $\partial_{t}(\text{volt})_{i,j,k-\frac{1}{2}},$ (5.12)

where

$$U_{i+\frac{1}{2},j}^{T} = u_{i+\frac{1}{2},j}^{*} \Delta y_{i+\frac{1}{2},j} \Delta z, \quad V_{i,j+\frac{1}{2}}^{T} = v_{i,j+\frac{1}{2}}^{*} \Delta x_{i,j+\frac{1}{2}} \Delta z,$$
(5.13)

$$u_{i+\frac{1}{2},j}^{*} = \frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}}), \quad v_{i,j+\frac{1}{2}}^{*} = \frac{1}{2}(v_{i+\frac{1}{2},j+\frac{1}{2}} + v_{i-\frac{1}{2},j+\frac{1}{2}}).$$
(5.14)

The finite difference analog of the continuity for the partial T-cell along the coastline (Figure 5.1b) is defined as follows:

$$V_{i,j+\frac{1}{2}}^{T} = \frac{1}{2} v_{i,j+\frac{1}{2}}^{*} \Delta x_{i,j+\frac{1}{2}} \Delta z, \quad V_{i,j-\frac{1}{2}}^{T} = \frac{1}{2} v_{i,j-\frac{1}{2}}^{*} \Delta x_{i,j-\frac{1}{2}} \Delta z, \quad (5.15)$$



Figure 5.1 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.

$$v_{i,j+\frac{1}{2}}^* = v_{i+\frac{1}{2},j+\frac{1}{2}}, \quad v_{i,j-\frac{1}{2}}^* = v_{i+\frac{1}{2},j-\frac{1}{2}}, \quad u_{i-\frac{1}{2},j}^* = 0.$$
 (5.16)

For the corner part of land as shown in Figure 5.1(c), it is given as follows:

$$U_{i-\frac{1}{2},j}^{T} = \frac{1}{2}u_{i-\frac{1}{2},j}^{*}\Delta y_{i-\frac{1}{2},j}\Delta z, \quad V_{i,j-\frac{1}{2}}^{T} = \frac{1}{2}v_{i,j-\frac{1}{2}}^{*}\Delta x_{i,j-\frac{1}{2}}\Delta z, \tag{5.17}$$

$$u_{i-\frac{1}{2},j}^{*} = u_{i-\frac{1}{2},j+\frac{1}{2}}, \quad v_{i,j-\frac{1}{2}}^{*} = v_{i+\frac{1}{2},j-\frac{1}{2}}.$$
(5.18)

The boundary condition for $W_{i,j}^T$ is as follows:

$$W_{i,j,0}^{T} = (\Delta A^{T} \partial_{t} \eta)_{i,j} = \left(U_{i-\frac{1}{2},j} \Delta y_{i-\frac{1}{2},j} - U_{i+\frac{1}{2},j} \Delta y_{i+\frac{1}{2},j} + V_{i,j-\frac{1}{2}} \Delta x_{i,j-\frac{1}{2}} - V_{i,j+\frac{1}{2}} \Delta x_{i,j+\frac{1}{2}} \right) - (P - E + R + I)_{i,j} \Delta A_{i,j,\frac{1}{2}}^{T}$$
(5.19)
at the surface (where $U = \sum_{k=1}^{N} u_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}$ and $V = \sum_{k=1}^{N} v_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}$), and

$$W_{i,j,ktbtm}^T = 0 \tag{5.20}$$

at the bottom (k = ktbtm).

5.3 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 as follows (Figure 5.1(a, b, and c)):

$$MC_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}^{U} \equiv \frac{MC_{i,j,k-\frac{1}{2}}^{T}}{N_{i,j,k-\frac{1}{2}}} + \frac{MC_{i+1,j,k-\frac{1}{2}}^{T}}{N_{i+1,j,k-\frac{1}{2}}} + \frac{MC_{i,j+1,k-\frac{1}{2}}^{T}}{N_{i,j+1,k-\frac{1}{2}}} + \frac{MC_{i+1,j+1,k-\frac{1}{2}}^{T}}{N_{i+1,j+1,k-\frac{1}{2}}} \\ = \frac{\partial_{t}(\Delta V^{T})_{i,j,k-\frac{1}{2}}}{N_{i,j,k-\frac{1}{2}}} + \frac{\partial_{t}(\Delta V^{T})_{i+1,j,k-\frac{1}{2}}}{N_{i+1,j,k-\frac{1}{2}}} + \frac{\partial_{t}(\Delta V^{T})_{i,j+1,k-\frac{1}{2}}}{N_{i,j+1,k-\frac{1}{2}}} + \frac{\partial_{t}(\Delta V^{T})_{i+1,j+1,k-\frac{1}{2}}}{N_{i+1,j+1,k-\frac{1}{2}}}, \quad (5.21)$$

where $N_{i,j,k-\frac{1}{2}}$ 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 5.1a), but N < 4 for the partial T-cells along coast lines (Figure 5.1b,c). This equation means that the mass convergence in a U-cell consists of the sum of the contributions from four surrounding T-cells.

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:

$$MC^{U}_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \equiv \frac{1}{4} (MC^{T}_{i,j,k-\frac{1}{2}} + MC^{T}_{i+1,j,k-\frac{1}{2}} + MC^{T}_{i,j+1,k-\frac{1}{2}} + MC^{T}_{i+1,j+1,k-\frac{1}{2}})$$

$$= \frac{1}{4} [\partial_{t} (\Delta V^{T})_{i,j,k-\frac{1}{2}} + \partial_{t} (\Delta V^{T})_{i+1,j,k-\frac{1}{2}} + \partial_{t} (\Delta V^{T})_{i,j+1,k-\frac{1}{2}} + \partial_{t} (\Delta V^{T})_{i+1,j+1,k-\frac{1}{2}}].$$
(5.22)

This can be rewritten as

$$\frac{1}{2}(U_{i,j}^{U}+U_{i,j+1}^{U}) - \frac{1}{2}(U_{i+1,j}^{U}+U_{i+1,j+1}^{U}) + \frac{1}{2}(V_{i,j}^{U}+V_{i+1,j}^{U}) - \frac{1}{2}(V_{i,j+1}^{U}+V_{i+1,j+1}^{U}) - W_{i+\frac{1}{2},j+\frac{1}{2},k-1}^{U} + W_{i+\frac{1}{2},j+\frac{1}{2},k}^{U} \approx \partial_{t}(\Delta V^{U})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}.$$
(5.23)

Note that the l.h.s. 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 (5.21) for defining the control volume of a U-cell. This comes from the need for a simple algebraic expression. Terms $U_{i,j}^U$, $V_{i,j}^U$, and $W_{i+\frac{1}{2},j+\frac{1}{2},k}^U$ are defined as follows:

$$U_{i,j}^{U} = u_{mi,j} \Delta y_{i,j} \Delta z, \quad V_{i,j}^{U} = v_{mi,j} \Delta x_{i,j} \Delta z, \tag{5.24}$$

$$u_{mi,j} = \frac{1}{2} (u_{i+\frac{1}{2},j}^* + u_{i-\frac{1}{2},j}^*), \quad v_{mi,j} = \frac{1}{2} (v_{i,j+\frac{1}{2}}^* + v_{i,j-\frac{1}{2}}^*), \quad (5.25)$$

$$W_{i+\frac{1}{2},j+\frac{1}{2},k}^{U} = \frac{1}{4} (W_{i,j,k}^{T} + W_{i+1,j,k}^{T} + W_{i,j+1,k}^{T} + W_{i+1,j+1,k}^{T}).$$
(5.26)

Note that W^U is obtained by an averaging operation on W^T and also that the following equations are derived from (5.13), (5.24), and (5.25) for the standard form of the U-cell continuity:

$$U_{i,j}^{U} = \frac{1}{2} (U_{i+\frac{1}{2},j}^{T} + U_{i-\frac{1}{2},j}^{T}),$$

$$V_{i,j}^{U} = \frac{1}{2} (V_{i,j+\frac{1}{2}}^{T} + V_{i,j-\frac{1}{2}}^{T}).$$
(5.27)

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).

The l.h.s. of the standard form of the continuity equation (5.23) 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 l.h.s. of (5.23) 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 (5.23):

$$\frac{1}{2}(U_{i,j}^{U}+V_{i,j}^{U}) - \frac{1}{2}(U_{i+1,j+1}^{U}+V_{i+1,j+1}^{U}) + \frac{1}{2}(U_{i,j+1}^{U}-V_{i,j+1}^{U}) - \frac{1}{2}(U_{i+1,j}^{U}-V_{i+1,j}^{U}) - W_{i+\frac{1}{2},j+\frac{1}{2},k-1}^{U} + W_{i+\frac{1}{2},j+\frac{1}{2},k}^{U} \approx \partial_{t}(\Delta V^{U})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}$$
(5.28)

Chapter 5 Continuity equation

Let us explain the meaning taking the first term on the l.h.s. of (5.28),

$$U_{i,j}^{U} + V_{i,j}^{U} = \frac{1}{2} \{ (U_{i-\frac{1}{2},j}^{T} + V_{i,j-\frac{1}{2}}^{T}) + (U_{i+\frac{1}{2},j}^{T} + V_{i,j+\frac{1}{2}}^{T}) \},$$
(5.29)

as an example, where (5.27) is used. If the flow is horizontally nondivergent, the horizontal mass fluxes $U_{i-\frac{1}{2},j}^{T}$ and $V_{i,j-\frac{1}{2}}^{T}$ 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 $(i - \frac{1}{2}, j - \frac{1}{2})$, 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 $(i + \frac{1}{2}, j - \frac{1}{2})$, 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 5.1d). 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 l.h.s. of (5.28) means the horizontal mass convergence in the diagonal square defined by four U-points $(i - \frac{1}{2}, j + \frac{1}{2}), (i + \frac{1}{2}, j - \frac{1}{2}), (i + \frac{3}{2}, j + \frac{1}{2}), and <math>(i + \frac{1}{2}, j + \frac{3}{2})$. Multiplying by a factor of $\frac{1}{2}$, the l.h.s. of (5.28) 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 l.h.s. of (5.23) as $A_{i+\frac{1}{2},j+\frac{1}{2}}$ and those of (5.28) 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:

$$\alpha A_{i+\frac{1}{2},j+\frac{1}{2}} + \beta B_{i+\frac{1}{2},j+\frac{1}{2}} - W^U_{i+\frac{1}{2},j+\frac{1}{2},k-1} + W^U_{i+\frac{1}{2},j+\frac{1}{2},k} \simeq \partial_t (\Delta V^U)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}},$$
(5.30)

where

$$\alpha + \beta = 1. \tag{5.31}$$

As shown later in Chapter 7, $(\alpha, \beta) = (2/3, 1/3)$ 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 $(\alpha, \beta) = (1, 0)$.

5.4 Summary

To summarize the solution procedure, MRI.COM uses (5.8) to obtain W^T under the boundary conditions (5.19) and (5.20). Then, using W^T , W^U is obtained by (5.26).

Part III

Equations of Motion

Chapter 6

Equations of motion (barotropic component)

Historically, iterative methods have been used to solve the barotropic part of the momentum equations by applying the rigid-lid approximation. The number of iterations to get convergence of a solution is of order N, where N is the larger number of grid points of the two horizontal directions. Thus the number of iterations increases as the number of grid points increases. This means that the iterative process could occupy a large part of the total CPU time. This is a severe burden and should be remedied.

An alternative is to replace the rigid lid with a free surface. The number of short barotropic time steps within a long baroclinic time step is roughly decided by the ratio between external and internal gravity wave speeds, around 70 to 100, which becomes smaller than N when the number of grid points of the model is large.* In addition, this method is more suitable for parallel computation than the iterative methods. Thus, for fine-resolution models, the free-surface formulation has numerous advantages over the iterative methods. The free-surface formulation was adopted in the Bryan-Cox-Semtner numerical ocean general circulation models by Killworth et al. (1991).

However, the free-surface formulation has a problem when it is used with a mixed-layer model. To appropriately resolve the surface mixed layer, the uppermost layer should be less than a few meters thick. This leads to a serious problem, since a free-surface model does not work when the sea surface is below the bottom of the uppermost layer and the thickness of this layer vanishes. This occurs in world ocean models, because the difference between the maximum and minimum of the sea surface height becomes several meters.^{\dagger}

To remedy this problem, σ -coordinate has been adopted near the sea surface until MRI.COM version 3. This method is called σ -*z* formulation and is introduced by Hasumi (2006). The thicknesses of the several upper layers ($z \le -H_B$) vary as the sea surface height does whereas the vertical position of the layers below ($z = -H_B$) is kept fixed (Figure 6.1a).

There were two major shortcomings in the implementation of the σ -*z* formulation in MRI.COM. First, because we set H_B as a constant, the sea-floor cannot be shallower than H_B . We typically take H_B as 20 - 30 meters, thus the representation of circulation around coastal areas with shallow sea-floor may not be accurate. Second, because a hybrid of vertical coordinates is used, some sort of dynamical analyses may be awkward.

To resolve these problems, a vertically rescaled height coordinate (z^*) introduced by Adcroft and Campin (2004), where undulation of the sea surface is distributed to all vertical grid cells in a water column, is adopted in MRI.COM version 4. In this chapter, we explain the free-surface formulation of MRI.COM with z^* vertical coordinate.

6.1 Governing equations

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

The momentum equations (2.69) and (2.70) are re-written:

$$\frac{\partial U}{\partial t} - fV = -\frac{(\eta + H)}{\rho_0 h_\mu} \frac{\partial (p_a + \rho_0 g \eta)}{\partial \mu} + X,$$
(6.1)

$$\frac{\partial V}{\partial t} + fU = -\frac{(\eta + H)}{\rho_0 h_{\psi}} \frac{\partial (p_a + \rho_0 g\eta)}{\partial \psi} + Y,$$
(6.2)

^{*} A North Pacific model with $1/4^{\circ} \times 1/6^{\circ}$ resolution, which was the first meso-scale eddy permitting model developed by the ocean modeling group of MRI about 20 years ago, had 742 grid points in the zonal direction. This figure prompted us to make a transition from the rigid-lid to free surface formulation.

[†] For example, the maximum and minimum sea-surface heights in a $1^{\circ} \times 1^{\circ}$ world ocean model are about 1 m (subtropical gyres) and -2 m (the Ross Sea), respectively.

6.2 Time integration on barotropic time levels



Figure 6.1 Schematic of (a) the near surface σ -coordinate layers (σ -z coordinate) and (b) z^* vertical coordinate.

where

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 (2.73) is re-written:

$$\frac{\partial \eta}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial (h_{\psi}U)}{\partial \mu} + \frac{\partial (h_{\mu}V)}{\partial \psi} \right\} = (P - E + R + I), \tag{6.5}$$

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).

Figure 6.2 illustrates the grid arrangement of the free-surface formulation. 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. (6.1) and (6.2) are calculated in the subroutine for the baroclinic component and defined at U-points.

6.2 Time integration on barotropic time levels

Figure 6.3 presents schematics of the time integration of the barotropic mode in the free-surface formulation. When the time integration of the baroclinic mode is performed from step n ($t = t_n$) to step n + 1 ($t = t_{n+1}, \Delta t = t_{n+1} - t_n$), 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 values (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 at $t = t_{n+1}$.

The "Euler forward-backward" scheme is a stable and economical numerical scheme for linear gravity wave equations without advection terms (Mesinger and Arakawa, 1976), and this is adopted for the governing equations in the free-surface formulation of MRI.COM. This scheme is more stable than the leap-frog scheme. The time step can be doubled for

Chapter 6 Equations of motion (barotropic component)



Figure 6.2 Grid arrangement of the free-surface formulation

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; the calculated surface elevation is then used to calculate the pressure gradient terms of the momentum equations.

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.

The finite-difference expression of the continuity equation (Eq. 6.5) is

$$\frac{(\eta_{i,j}^{'} - \eta_{i,j})}{\Delta t_{\rm tr}} + \frac{1}{(h_{\mu}h_{\psi})_{i,j}} \Big[(\delta_{\mu}\overline{h_{\psi}U}^{\psi})_{i,j} + (\delta_{\psi}\overline{h_{\mu}V}^{\mu})_{i,j} \Big] = (P - E + R + I)_{i,j}, \tag{6.6}$$

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:

$$\delta_{\mu}A_{i} \equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta\mu_{i}}, \qquad \qquad \delta_{\mu}A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_{i}}{\Delta\mu_{i+\frac{1}{2}}}, \\ \overline{A_{i}}^{\mu} \equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2}, \qquad \qquad \overline{A_{i+\frac{1}{2}}}^{\mu} \equiv \frac{A_{i+1} + A_{i}}{2}.$$
(6.7)

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

$$\begin{aligned} &(\eta'_{i,j} - \eta_{i,j}) \cdot \Delta S_{T_{i,j}} \\ &= \Delta t_{\text{tr}} \cdot \left\{ (P - E + R + I)_{i,j} \cdot \Delta S_{T_{i,j}} - \left(\Delta y_{i+\frac{1}{2},j} \overline{U}^{\psi}_{i+\frac{1}{2},j} - \Delta y_{i-\frac{1}{2},j} \overline{U}^{\psi}_{i-\frac{1}{2},j} \right) - \left(\Delta x_{i,j+\frac{1}{2}} \overline{V}^{\mu}_{i,j+\frac{1}{2}} - \Delta x_{i,j-\frac{1}{2}} \overline{V}^{\mu}_{i,j-\frac{1}{2}} \right) \right\}. \end{aligned}$$

$$(6.8)$$

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 $\eta'_{i,i}$, the momentum equations, Eqs. (6.1) and (6.2), are solved. A longer time step can be used when

6.2 Time integration on barotropic time levels

the semi-implicit scheme is applied for the Coriolis terms in the momentum equations. Their finite-difference forms are

$$\frac{(U'_{i+\frac{1}{2},j+\frac{1}{2}} - U_{i+\frac{1}{2},j+\frac{1}{2}})}{\Delta t_{\rm tr}} - \frac{f(V'_{i+\frac{1}{2},j+\frac{1}{2}} + V_{i+\frac{1}{2},j+\frac{1}{2}})}{2} = -\frac{g(H_{i+\frac{1}{2},j+\frac{1}{2}} + \eta'_{i+\frac{1}{2},j+\frac{1}{2}})}{(h_{\mu})_{i+\frac{1}{2},j+\frac{1}{2}}} \delta_{\mu} \overline{\eta'}_{i+\frac{1}{2},j+\frac{1}{2}} + X_{i+\frac{1}{2},j+\frac{1}{2}}$$
(6.9)

$$\frac{(V_{i+\frac{1}{2},j+\frac{1}{2}}^{'}-V_{i+\frac{1}{2},j+\frac{1}{2}})}{\Delta t_{\rm tr}} + \frac{f(U_{i+\frac{1}{2},j+\frac{1}{2}}^{'}+U_{i+\frac{1}{2},j+\frac{1}{2}})}{2} = -\frac{g(H_{i+\frac{1}{2},j+\frac{1}{2}}+\overline{\eta'}_{i+\frac{1}{2},j+\frac{1}{2}})}{(h_{\psi})_{i+\frac{1}{2},j+\frac{1}{2}}}\delta_{\psi}\overline{\eta'}_{i+\frac{1}{2},j+\frac{1}{2}}^{\mu,\psi}$$
(6.10)

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

$$(U_{i+\frac{1}{2},j+\frac{1}{2}}^{'}-U_{i+\frac{1}{2},j+\frac{1}{2}}) - \frac{f\Delta t_{\rm tr}}{2}(V_{i+\frac{1}{2},j+\frac{1}{2}}^{'}+V_{i+\frac{1}{2},j+\frac{1}{2}}) = \Delta t_{\rm tr}GX_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{6.11}$$

$$(V_{i+\frac{1}{2},j+\frac{1}{2}}^{'}-V_{i+\frac{1}{2},j+\frac{1}{2}})+\frac{f\Delta t_{\rm tr}}{2}(U_{i+\frac{1}{2},j+\frac{1}{2}}^{'}+U_{i+\frac{1}{2},j+\frac{1}{2}})=\Delta t_{\rm tr}GY_{i+\frac{1}{2},j+\frac{1}{2}},$$
(6.12)

leading to

$$U_{i+\frac{1}{2},j+\frac{1}{2}}^{'} - \frac{f\Delta t_{\rm tr}}{2}V_{i+\frac{1}{2},j+\frac{1}{2}}^{'} = U_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{f\Delta t_{\rm tr}}{2}V_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta t_{\rm tr}GX_{i+\frac{1}{2},j+\frac{1}{2}},$$
(6.13)

$$V_{i+\frac{1}{2},j+\frac{1}{2}}^{'} + \frac{f\Delta t_{\rm tr}}{2}U_{i+\frac{1}{2},j+\frac{1}{2}}^{'} = V_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{f\Delta t_{\rm tr}}{2}U_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta t_{\rm tr}GY_{i+\frac{1}{2},j+\frac{1}{2}}.$$
(6.14)

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:

$$U_{i+\frac{1}{2},j+\frac{1}{2}}' = \left\{ RX_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{f\Delta t_{\rm tr}}{2}RY_{i+\frac{1}{2},j+\frac{1}{2}} \right\} / \left\{ 1 + \left(\frac{f\Delta t_{\rm tr}}{2}\right)^2 \right\},\tag{6.15}$$

$$V_{i+\frac{1}{2},j+\frac{1}{2}}' = \left\{ RY_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{f\Delta t_{\rm tr}}{2} RX_{i+\frac{1}{2},j+\frac{1}{2}} \right\} / \left\{ 1 + \left(\frac{f\Delta t_{\rm tr}}{2}\right)^2 \right\}.$$
(6.16)



Figure 6.3 Schematic figure of the time integration of the barotropic mode and its time-filtering procedure. a_m and b_m are weights for t = n + 2 and t = n + 1, respectively.

6.3 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.

6.3.1 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 $t = t_n$ to $t = t_{n+1}$ is to obtain the barotropic state variables η , U, and V at the baroclinic time level of $t = t_{n+1}$. To achieve this, starting from the baroclinic time level of $t = t_n$, the governing equations of the barotropic submodel are integrated slightly beyond $t = t_{n+2}$ to compute the barotropic state variables as a weighted average of those at barotropic time levels.

We follow the method adopted by the Regional Oceanic Modeling System (ROMS; Shchepetkin and McWilliams, 2005) to determine a weighting function for averaging. The method of ROMS is actually designed to compute the state variables at $t = t_{n+2}$ as well as a flow field at $t = t_{n+1}$ which is consistent with surface height at $t = t_{n+2}$ under the discretization of (6.5) by the leap-frog scheme,

$$\frac{\eta^{n+2} - \eta^n}{2\Delta t_{\rm cl}} = -\nabla \cdot \boldsymbol{U}^{n+1} + (P - E + R + I).$$
(6.17)

In the framework of the time integration of MRI.COM, only flow field at $t = t_{n+1}$, (U^{n+1}, V^{n+1}) , is used among the results of the barotropic submodel. The surface height at baroclinic time levels are computed by (6.17). This is to make the surface height equation be consistent with the continuity equations in a vertical column (Leclair and Madec, 2009). (By setting the runtime parameter l_global_local_cnsv_ssh = .false., an averaged state from the barotropic submodel may be used as η^{n+1} instead of (6.17), although this should be used with care.)

In the old versions of MRI.COM, simple averaging of $(2 \times \Delta t_{cl}/\Delta t_{tr} + 1)$ barotropic time steps between the two baroclinic time intervals $[t_n, t_{n+2}]$ is used to obtain state variables at t_{n+1} . From version 4, we use a weighted average as explained by Shchepetkin and McWilliams (2005). Derivation of the weighting function for (U^{n+1}, V^{n+1}) , b_m , is briefly summarized below.

First, a weighting shape function to calculate variables at $t = t_{n+2}$ is denoted with $\{a_m\}$ (Note that the target is at $t = t_{n+2}$, not $t = t_{n+1}$). This must satisfy discrete normalization and centroid conditions,

$$\sum_{m=1}^{M^*} a_m \equiv 1, \quad \sum_{m=1}^{M^*} m a_m \equiv 2M, \tag{6.18}$$

where *M* is the ratio between barotropic-baroclinic time step $(M = \Delta t_{cl}/\Delta t_{tr})$, and M^* is the last index at which $a_m > 0$, where $2M \le M^*$. 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+2} is computed as

$$\eta^{n+2} = \sum_{m=1}^{M^*} a_m \eta^m, \quad U^{n+2} = \sum_{m=1}^{M^*} a_m U^m, \quad V^{n+2} = \sum_{m=1}^{M^*} a_m V^m, \tag{6.19}$$

where η^m , U^m , and V^m are instantaneous state variables at the barotropic time level $t = t_m$.

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

$$\frac{\eta^{n+2} - \eta^n}{2\Delta t_{\rm cl}} = -\nabla \cdot \mathbf{U}^{n+1}.$$
(6.20)

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

$$\frac{\eta^{m+1} - \eta^m}{\Delta t_{\rm tr}} = -\nabla \cdot \mathbf{U}^{m+\frac{1}{2}},\tag{6.21}$$

we may obtain the expression for η^m as

$$\eta^{m} = \eta^{0} - \Delta t_{\rm tr} \sum_{m'=0}^{m-1} \nabla \cdot \mathbf{U}^{m'+\frac{1}{2}}.$$
(6.22)

Inserting this into (6.19) and after some manipulations, which is detailed in Shchepetkin and McWilliams (2005), we have

$$\eta^{n+2} = \eta^0 - 2\Delta t_{\rm cl} \nabla \cdot \sum_{m'=1}^{M^*} b_{m'} \mathbf{U}^{m'-\frac{1}{2}},\tag{6.23}$$

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

$$b_{m'} = \frac{1}{2M} \sum_{m=m'}^{M^*} a_m.$$
(6.24)

By comparing (6.20) and (6.23), vertically integrated flow field at the baroclinic time level $t = t_{n+1}$ is obtained as

$$U^{n+1} = \sum_{m=1}^{M^*} b_m U^{m-\frac{1}{2}}, \quad V^{n+1} = \sum_{m=1}^{M^*} b_m V^{m-\frac{1}{2}}.$$
(6.25)

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

$$\sum_{m=1}^{M^*} b_m \equiv 1, \quad \sum_{m=1}^{M^*} m b_m \equiv M, \tag{6.26}$$

as well as (6.18).

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

$$A(\tau) = A_0 \left\{ \left(\frac{\tau}{\tau_0}\right)^p \left[1 - \left(\frac{\tau}{\tau_0}\right)^q \right] - r\frac{\tau}{\tau_0} \right\},\tag{6.27}$$

where p = 2, q = 2, r = 0.2346283, and A_0 and τ_0 are chosen to satisfy normalization conditions (6.18) and (6.26) iteratively. The initial guess for A_0 and τ_0 is given as follows:

$$A_0 = 1, \quad \tau_0 = \frac{(p+2)(p+q+2)}{(p+1)(p+q+1)}.$$
(6.28)

Approximate shape of $\{a_m\}$ and $\{b_m\}$ are shown by red and blue lines of Figure 6.3.

In the following, 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 (6.25) 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:

$$U^{m+1} = U^m + \Delta t_{\rm tr} f V^{m+\frac{1}{2}} - \Delta t_{\rm tr} \Big[\frac{(\eta + H)}{\rho_0 h_\mu} \frac{\partial (p_a + \rho_0 g \eta)}{\partial \mu} \Big]^{m+\frac{1}{2}} + \Delta t_{\rm tr} X, \tag{6.29}$$

$$V^{m+1} = V^m - \Delta t_{\rm tr} f U^{m+\frac{1}{2}} - \Delta t_{\rm tr} \Big[\frac{(\eta + H)}{\rho_0 h_{\psi}} \frac{\partial (p_a + \rho_0 g \eta)}{\partial \psi} \Big]^{m+\frac{1}{2}} + \Delta t_{\rm tr} Y, \tag{6.30}$$

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 (6.29) and (6.30) over m' = [0, m-1] yields

$$U^{m} = U^{0} + \Delta t_{\text{tr}} \sum_{m'=0}^{m-1} f V^{m'+\frac{1}{2}} - \Delta t_{\text{tr}} \sum_{m'=0}^{m-1} \left[\frac{(\eta+H)}{\rho_{0}h_{\mu}} \frac{\partial(p_{a}+\rho_{0}g\eta)}{\partial\mu} \right]^{m'+\frac{1}{2}} + m\Delta t_{\text{tr}}X,$$
(6.31)

$$V^{m} = V^{0} - \Delta t_{\rm tr} \sum_{m'=0}^{m-1} f U^{m'+\frac{1}{2}} - \Delta t_{\rm tr} \sum_{m'=0}^{m-1} \left[\frac{(\eta+H)}{\rho_{0}h_{\psi}} \frac{\partial(p_{a}+\rho_{0}g\eta)}{\partial\psi} \right]^{m'+\frac{1}{2}} + m\Delta t_{\rm tr}Y, \tag{6.32}$$

Chapter 6 Equations of motion (barotropic component)

Applying the time averaging procedure of (6.25) yields

$$\langle U \rangle^{n+1} \equiv \sum_{m=1}^{M^*} b_m U^m \tag{6.33}$$

$$= U^{0} + \Delta t_{\text{tr}} \sum_{m=1}^{M^{*}} \left[b_{m} \sum_{m'=1}^{m} f V^{m'-\frac{1}{2}} \right] - \Delta t_{\text{tr}} \sum_{m=1}^{M^{*}} \left\{ b_{m} \sum_{m'=1}^{m} \left[\frac{(\eta + H)}{\rho_{0}h_{\mu}} \frac{\partial(p_{a} + \rho_{0}g\eta)}{\partial\mu} \right]^{m'-\frac{1}{2}} \right\} + \sum_{m=1}^{M^{*}} m b_{m} \Delta t_{\text{tr}} X, \quad (6.34)$$

$$\langle V \rangle^{n+1} \equiv \sum_{m=1}^{\infty} b_m V^m \tag{6.35}$$

$$=V^{0} - \Delta t_{\rm tr} \sum_{m=1}^{M^{*}} \left[b_m \sum_{m'=1}^{m} f U^{m'-\frac{1}{2}} \right] - \Delta t_{\rm tr} \sum_{m=1}^{M^{*}} \left\{ b_m \sum_{m'=1}^{m} \left[\frac{(\eta+H)}{\rho_0 h_{\psi}} \frac{\partial(p_a + \rho_0 g \eta)}{\partial \psi} \right]^{m'-\frac{1}{2}} \right\} + \sum_{m=1}^{M^{*}} m b_m \Delta t_{\rm tr} Y, \quad (6.36)$$

which are rearranged to have a form

$$\langle U \rangle^{n+1} = U^0 + \Delta t_{\rm cl} f \langle \langle V \rangle \rangle^{n+\frac{1}{2}} - \Delta t_{\rm cl} \left\langle \left\langle \frac{(\eta+H)}{\rho_0 h_\mu} \frac{\partial (p_a+\rho_0 g\eta)}{\partial \mu} \right\rangle \right\rangle^{n+\frac{1}{2}} + \Delta t_{\rm cl} X^n, \tag{6.37}$$

$$\langle V \rangle^{n+1} = V^0 - \Delta t_{\rm cl} f \langle \langle U \rangle \rangle^{n+\frac{1}{2}} - \Delta t_{\rm cl} \left\langle \left\langle \frac{(\eta+H)}{\rho_0 h_{\psi}} \frac{\partial (p_a+\rho_0 g\eta)}{\partial \psi} \right\rangle \right\rangle^{n+\frac{1}{2}} + \Delta t_{\rm cl} Y^n, \tag{6.38}$$

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

$$\frac{u_{k-\frac{1}{2}}^{'}\Delta z_{k-\frac{1}{2}}^{n+1} - u_{k-\frac{1}{2}}^{n}\Delta z_{k-\frac{1}{2}}^{n}}{\Delta t_{\rm cl}} = f[v_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + F_{\mu}^{n},$$
(6.39)

$$\frac{v_{k-\frac{1}{2}}^{\prime}\Delta z_{k-\frac{1}{2}}^{n+1} - v_{k-\frac{1}{2}}^{n}\Delta z_{k-\frac{1}{2}}^{n}}{\Delta t_{\rm cl}} = -f[u_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + F_{\psi}^{n}.$$
(6.40)

This is vertically summed up to give

$$\frac{\sum_{k=1}^{N} (u_{k-\frac{1}{2}}^{'} \Delta z_{k-\frac{1}{2}}^{n+1}) - \sum_{k=1}^{N} (u_{k-\frac{1}{2}}^{n} \Delta z_{k-\frac{1}{2}}^{n})}{\Delta t_{\rm cl}} = f \sum_{k=1}^{N} [v_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + X^{n}, \tag{6.41}$$

$$\frac{\sum_{k=1}^{N} (v_{k-\frac{1}{2}}^{'} \Delta z_{k-\frac{1}{2}}^{n+1}) - \sum_{k=1}^{N} (v_{k-\frac{1}{2}}^{n} \Delta z_{k-\frac{1}{2}}^{n})}{\Delta t_{\rm cl}} = -f \sum_{k=1}^{N} [u_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + Y^{n}.$$
(6.42)

 X^n and Y^n are removed from (6.37), (6.38), (6.41), and (6.42) to give

$$\frac{\sum_{k=1}^{N} (\langle u \rangle^{n+1} \Delta z_{k-\frac{1}{2}}^{n+1}) - \sum_{k=1}^{N} (u_{k-\frac{1}{2}}^{'} \Delta z_{k-\frac{1}{2}}^{n+1})}{\Delta t_{\rm cl}} = f \langle \langle V \rangle \rangle^{n+\frac{1}{2}} - f \sum_{k=1}^{N} [v_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} - \left\langle \left\langle \frac{(\eta+H)}{\rho_0 h_{\mu}} \frac{\partial(p_a+\rho_0 g\eta)}{\partial \mu} \right\rangle \right\rangle^{n+\frac{1}{2}}$$
(6.43)

$$\frac{\sum_{k=1}^{N} (\langle v \rangle^{n+1} \Delta z_{k-\frac{1}{2}}^{n+1}) - \sum_{k=1}^{N} (v_{k-\frac{1}{2}}^{'} \Delta z_{k-\frac{1}{2}}^{n+1})}{\Delta t_{\rm cl}} = -f \langle \langle U \rangle \rangle^{n+\frac{1}{2}} + f \sum_{k=1}^{N} [u_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} - \left\langle \left\langle \frac{(\eta+H)}{\rho_0 h_{\psi}} \frac{\partial (p_a+\rho_0 g\eta)}{\partial \psi} \right\rangle \right\rangle^{n+\frac{1}{2}}.$$
(6.44)

6.3 Prognostication of state variables at the baroclinic time level

This is combined with (6.39) and (6.40) to give

$$\frac{\left(u_{k-\frac{1}{2}}^{'}-\overline{u'}^{z}+\langle u\rangle^{n+1}\right)\Delta z_{k-\frac{1}{2}}^{n+1}-u_{k-\frac{1}{2}}^{n}\Delta z_{k-\frac{1}{2}}^{n}}{\Delta t_{cl}}=f\left[v_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}\right]^{n+\frac{1}{2}}-f\overline{[v]^{n+\frac{1}{2}}}^{z}\Delta z_{k-\frac{1}{2}}^{n+1}+f\langle\langle v\rangle\rangle^{n+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}^{n+1}-\frac{\Delta z_{k-\frac{1}{2}}^{n+1}}{\eta^{n+1}+H}\left\langle\left\langle\frac{(\eta+H)}{\rho_{0}h_{\mu}}\frac{\partial(p_{a}+\rho_{0}g\eta)}{\partial\mu}\right\rangle\right\rangle^{n+\frac{1}{2}}+F_{\mu}^{n}\right)$$
(6.45)

$$\frac{\left(v_{k-\frac{1}{2}}^{'}-\overline{v^{'}}^{z}+\langle v\rangle^{n+1}\right)\Delta z_{k-\frac{1}{2}}^{n+1}-v_{k-\frac{1}{2}}^{n}\Delta z_{k-\frac{1}{2}}^{n}}{\Delta t_{cl}} = -f\left[u_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}\right]^{n+\frac{1}{2}}+f\overline{[u]^{n+\frac{1}{2}}}^{z}\Delta z_{k-\frac{1}{2}}^{n+1}-f\langle\langle u\rangle\rangle^{n+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}^{n+1} -\frac{\Delta z_{k-\frac{1}{2}}^{n+1}}{\eta^{n+1}+H}\left\langle\left\langle\frac{(\eta+H)}{\rho_{0}h_{\psi}}\frac{\partial(p_{a}+\rho_{0}g\eta)}{\partial\psi}\right\rangle\right\rangle^{n+\frac{1}{2}}+F_{\psi}^{n},$$
(6.46)

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 (6.39) and (6.40), 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 $\langle \langle \mathbf{u} \rangle \rangle^{n+\frac{1}{2}}$.

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

Vertically integrated transport at time $t = t_{n+1}$ is obtained by a weighted averaging of those at barotropic time levels as explained in the previous section. Surface elevation at time $t = t_{n+1}$ is computed by using vertically integrated transport at time t_n , which has been calculated in the previous time step and stored, and updated with the leap frog scheme:

$$\frac{\eta^{n+1} - \eta^{n-1}}{2\Delta t} + \nabla \cdot U^n = P - E + R + I,$$
(6.47)

as explained in Section 6.3.1.

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

These operations correspond to the expression for the velocity at the (n + 1)th time step in the tendency term of (6.45), (6.46), for example, $u'_{k-1/2} - \overline{u'}^z + \langle u \rangle^{n+1}$.

Because (6.47) 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

$$\left(\frac{\partial u}{\partial t}, \frac{\partial v}{\partial t}\right)\Big|_{k=\frac{1}{2}} = \dots + \frac{1}{\rho_0} \frac{(\tau_\mu, \tau_\psi)}{\Delta z_{\frac{1}{2}}},\tag{6.48}$$

where $(\tau_{\mu}, \tau_{\psi})$ 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

$$F_{z}^{\theta} = -\frac{1}{\gamma_{\theta}}(\theta - \theta^{*})\Delta z_{\frac{1}{2}}, \qquad \left. \frac{\partial \theta}{\partial t} \right|_{k = \frac{1}{2}} = \dots + \frac{F_{z}^{\theta}}{\Delta z_{\frac{1}{2}}}, \tag{6.49}$$

$$F_z^S = -\frac{1}{\gamma_s}(S - S^*)\Delta z_{\frac{1}{2}}, \qquad \left. \frac{\partial S}{\partial t} \right|_{k = \frac{1}{2}} = \dots + \frac{F_z^S}{\Delta z_{\frac{1}{2}}}.$$
(6.50)

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

6.4 Horizontal diffusivity of sea surface height

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,

$$\frac{\partial \eta}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial (h_{\psi}U)}{\partial \mu} + \frac{\partial (h_{\mu}V)}{\partial \psi} \right\} = \mathcal{D}(\eta) + (P - E + R + I)$$
(6.51)

where 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:

$$\mathcal{D}(\eta) = \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial}{\partial\mu} \left(\frac{h_{\psi}\kappa_H}{h_{\mu}} \frac{\partial\eta}{\partial\mu} \right) + \frac{\partial}{\partial\psi} \left(\frac{h_{\mu}\kappa_H}{h_{\psi}} \frac{\partial\eta}{\partial\psi} \right) \right\}$$
(6.52)

$$=\frac{1}{h_{\mu}h_{\psi}}\left\{\frac{\partial h_{\psi}F_{\mu}^{\eta}}{\partial\mu}+\frac{\partial h_{\mu}F_{\psi}^{\eta}}{\partial\psi}\right\}$$
(6.53)

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

$$\mathbf{F}^{\eta} = -\left(\frac{\kappa_H}{h_{\mu}}\frac{\partial\eta}{\partial\mu}, \frac{\kappa_H}{h_{\psi}}\frac{\partial\eta}{\partial\psi}\right). \tag{6.54}$$

Therefore, continuity equation may be rewritten as follows:

$$\frac{\partial \eta}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial [h_{\psi}(U+F_{\mu}^{\eta})]}{\partial \mu} + \frac{\partial [h_{\mu}(V+F_{\psi}^{\eta})]}{\partial \psi} \right\} = (P-E+R+I).$$
(6.55)

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_sh_dflx_y for this purpose.

6.5 Inclusion of tidal effect

This section explains the tide option (TIDE) in MRI.COM.

6.5.1 Tide producing term in the momentum equation

Tidal forcing can be introduced into the barotropic part of the momentum equations (6.1) and (6.2) as follows (e.g., Schiller, 2004):

$$\frac{\partial U}{\partial t} - fV = -\frac{(\eta + H)}{\rho_0 h_\mu} \frac{\partial (p_a + \rho_0 g\eta)}{\partial \mu} - \frac{g(\eta + H)}{h_\mu} \frac{\partial}{\partial \mu} [(1 - \alpha)\eta - \beta \eta_{\text{eq}}] + X, \tag{6.56}$$

$$\frac{\partial V}{\partial t} + fU = -\frac{(\eta + H)}{\rho_0 h_{\psi}} \frac{\partial (p_a + \rho_0 g\eta)}{\partial \psi} - \frac{g(\eta + H)}{h_{\psi}} \frac{\partial}{\partial \psi} \left[(1 - \alpha)\eta - \beta \eta_{\text{eq}} \right] + Y, \tag{6.57}$$

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	$2h_0$

Table6.1 Constants of major tidal modes

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) + Int[(y - 1973)/4], *d* is the day numbe of the year (*d* = 1 for January 1), $y \ge 1975$ is the year number, and Int[x] is the integral part of x.

where α defines self-attraction and loading of the ocean tides, $\beta = 1 + k - h$ defines effect of tide-generating potential and correction due to earth tides with *k* and *h* being Love numbers, and η_{eq} is equilibrium tide, or astronomical tide-generating potential. In MRI.COM, we usually set $\alpha = 0.88$ and $\beta = 0.7$.

The eight primary constituents that consist of the four largest equilibrium tides of the diurnal and semi-diurnal species $(K_1, O_1, P_1, Q_1, M_2, S_2, N_2, K_2)$ may be included in the tide producing potential η_{eq} of MRI.COM. Following Schwiderski (1980), we write the *n*-th (n = [1, 4]) diurnal equilibrium tide as

$$\eta_{\text{eq},n} = K_n \cos^2 \phi \cos(\sigma_n t + \chi_n + 2\lambda)$$

= $K_n \cos^2 \phi [\cos(\sigma_n t + \chi_n) \cos 2\lambda - \sin(\sigma_n t + \chi_n) \sin 2\lambda],$ (6.58)

and the *n*-th semi-diurnal component (n = [5, 8]) as

$$\eta_{\text{eq},n} = K_n \sin 2\phi \cos(\sigma_n t + \chi_n + \lambda)$$

= $K_n \sin 2\phi [\cos(\sigma_n t + \chi_n) \cos \lambda - \sin(\sigma_n t + \chi_n) \sin \lambda].$ (6.59)

The meaning of the mathematical symbols is as follows:

- 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 6.1.

t

6.5.2 Separation of linear responses to tidal forcing

By including tidal forcing, the balance in the basic fields solved without tides will be modified through non-linear terms of the equations. In addition to the non-linear terms, self-attraction and loading terms in (6.56) and (6.57) will modify the relationships between the sea surface height gradient and the barotropic currents. That is, the solution without astronomical tide-generating potential ($\eta_{eq} = 0$) will be different than that solved with standard equations without self-attraction and loading terms.

To resolve the possible problem that the terms specialized for tides affect the basic fields unintentionally, 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. First, we rewrite the basic equation of the barotropic submodel in the following form that highlights the terms that may be specialized for tides:

$$\frac{\partial \mathbf{U}}{\partial t} + f\mathbf{k} \times \mathbf{U} = -g(\eta + H)\nabla(\eta - \beta\eta_{\text{eq}} - \eta_{\text{SAL}}) + \mathbf{F}^{\text{horz}} + \frac{\tau^{\text{btm}}}{\rho_0} + \mathbf{X}',$$
(6.60)

where \mathbf{F}^{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 **X** defined by (6.3) and (6.4) and $\mathbf{X}' \equiv \mathbf{X} - \mathbf{F}^{\text{horz}} - \tau^{\text{btm}}/\rho_0$.

Chapter 6 Equations of motion (barotropic component)

Note also that atmospheric pressure term is dropped for brevity. Continuity equation is

$$\frac{\partial \eta}{\partial t} + \nabla \cdot \mathbf{U} = F_w, \tag{6.61}$$

where $F_w \equiv P - E + R + I$ is surface fresh water forcing that will be enter the system through the basic fields. Next, the variables are decomposed into the linear tidal component and the basic component,

$$\mathbf{U} = \mathbf{U}_h + \mathbf{U}_{lt} \tag{6.62}$$

$$\eta = \eta_b + \eta_{lt} \tag{6.63}$$

$$\mathbf{F}^{\text{horz}} = \mathbf{F}_{b}^{\text{horz}} + \mathbf{F}_{lt}^{\text{horz}} \tag{6.64}$$

$$\tau^{\rm btm} = \tau_b^{\rm btm} + \tau_{lt}^{\rm btm},\tag{6.65}$$

where "lt" refers to linear tidal component and "b" refers to basic fields. Now we decompose (6.60) and (6.61) to obtain governing equations for each of the two components.

For linear tidal component:

$$\frac{\partial \mathbf{U}_{lt}}{\partial t} + f\mathbf{k} \times \mathbf{U}_{lt} = -g(\eta + H)\nabla(\eta_{lt} - \beta\eta_{eq} - \eta_{SAL}) + \mathbf{F}_{lt}^{horz} + \frac{\tau_{lt}^{btm}}{\rho_0},$$
(6.66)

$$\frac{\partial \eta_{lt}}{\partial t} + \nabla \cdot \mathbf{U}_{lt} = 0, \tag{6.67}$$

where we assume that self-attraction and loading acts only to linear tidal component in such a way as $\eta_{SAL} = (1 - \alpha)\eta_{lt}$. For the basic component:

$$\frac{\partial \mathbf{U}_b}{\partial t} + f\mathbf{k} \times \mathbf{U}_b = -g(\eta + H)\nabla\eta_b + \mathbf{F}_b^{\text{horz}} + \frac{\tau_b^{\text{btm}}}{\rho_0} + \mathbf{X}', \tag{6.68}$$

$$\frac{\partial \eta_b}{\partial t} + \nabla \cdot \mathbf{U}_b = F_w. \tag{6.69}$$

In the above decomposition, the linear tidal component represents only the linear response to tidal forcing, 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 (τ^{btm}) is non-linear and should be treated carefully. The bottom friction term (see Section 7.3.6 for details) is expressed as

$$\tau^{\rm btm} = -\rho_0 C_D |\mathbf{u}| \mathbf{T}_\theta \mathbf{u},\tag{6.70}$$

where C_D is a drag coefficient and \mathbf{T}_{θ} is a matrix representing horizontal veering with an angle of θ ,

$$\mathbf{T}_{\theta} = \begin{pmatrix} \cos\theta & -\sin\theta\\ \sin\theta & \cos\theta \end{pmatrix}.$$
(6.71)

Sum of the two components are used for the scalar part $(|\mathbf{u}|)$ and only the vector part is decomposed:

$$\tau_b^{\text{btm}} = -\rho_0 C_D |\mathbf{u}_b + \mathbf{u}_{ll} | \mathbf{T}_\theta \mathbf{u}_b, \tag{6.72}$$

$$\tau_{lt}^{\text{btm}} = -\rho_0 C_D |\mathbf{u}_b + \mathbf{u}_{lt}| \mathbf{T}_{\theta} \mathbf{u}_{lt}.$$
(6.73)

Similarly, when the horizontal viscosity parameterization is non-linear as in the Smagorinsky scheme, the coefficient part (e.g., v_H in the equations below) is calculated by using the full velocity and the decomposed fields are applied to the viscosity operator. When the horizontal tension D_T and shear D_S of the velocity field,

$$D_T = h_{\psi} \frac{\partial}{h_{\mu} \partial \mu} \left(\frac{u}{h_{\psi}} \right) - h_{\mu} \frac{\partial}{h_{\psi} \partial \psi} \left(\frac{v}{h_{\mu}} \right), \tag{6.74}$$

$$D_{S} = h_{\mu} \frac{\partial}{h_{\psi} \partial \psi} \left(\frac{u}{h_{\mu}} \right) + h_{\psi} \frac{\partial}{h_{\mu} \partial \mu} \left(\frac{v}{h_{\psi}} \right), \tag{6.75}$$

is used to calculate horizontal viscosity as

$$F_{u}^{\text{horz}} = \frac{1}{h_{\psi}^{2}} \frac{\partial}{h_{\mu}\partial\mu} \left(h_{\psi}^{2} \nu_{H} D_{T}\right) + \frac{1}{h_{\mu}^{2}} \frac{\partial}{h_{\psi}\partial\psi} \left(h_{\mu}^{2} \nu_{H} D_{S}\right), \tag{6.76}$$

$$F_{\nu}^{\text{horz}} = \frac{1}{h_{\psi}^2} \frac{\partial}{h_{\mu} \partial \mu} \left(h_{\psi}^2 \nu_H D_S \right) - \frac{1}{h_{\mu}^2} \frac{\partial}{h_{\psi} \partial \psi} \left(h_{\mu}^2 \nu_H D_T \right), \tag{6.77}$$

where v_H is the horizontal viscosity coefficient, \mathbf{u}_b and \mathbf{u}_{lt} is used for D_T and D_S to calculate $\mathbf{F}_b^{\text{horz}}$ and $\mathbf{F}_{lt}^{\text{horz}}$ by using (6.76) and (6.77).

By formulation, \mathbf{F}^{horz} , τ^{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}^{horz} and τ^{btm} should be allowed to vary with the evolution of barotropic velocity field. In this case, $\mathbf{F}_{lt}^{\text{horz}}$ and τ_{lt}^{btm} in (6.66) may be replaced by the viscosity parameterization suitable to the tidal velocity fields (represented by $\mathbf{F}_{tide}^{\text{horz}}$ and τ_{tide}^{btm} , respectively). Then, (6.66) and (6.68) becomes,

$$\frac{\partial \mathbf{U}_{lt}}{\partial t} + f\mathbf{k} \times \mathbf{U}_{lt} = -g(\eta + H)\nabla(\alpha\eta_{lt} - \beta\eta_{eq}) + \mathbf{F}_{tide}^{\text{horz}} + \frac{\tau_{tide}^{\text{btm}}}{\rho_0}, \tag{6.78}$$

$$\frac{\partial \mathbf{U}_b}{\partial t} + f\mathbf{k} \times \mathbf{U}_b = -g(\eta + H)\nabla\eta_b + \mathbf{X} - \mathbf{F}_{lt}^{\text{horz}} - \frac{\tau_{lt}^{\text{btm}}}{\rho_0}.$$
(6.79)

A sequence of operations explained in Section 6.3.1 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. The weighted average of the barotropic fields is written as follows:

$$\langle \mathbf{U}_{lt} \rangle^{n+1} = \mathbf{U}_{lt}^{0} - \Delta t_{cl} f \mathbf{k} \times \langle \langle \mathbf{U}_{lt} \rangle \rangle^{n+\frac{1}{2}} - \Delta t_{cl} g (\eta^{n} + H) \left\langle \left\langle \nabla (\alpha \eta_{lt} - \beta \eta_{eq}) \right\rangle \right\rangle^{n+\frac{1}{2}} + \Delta t_{cl} \langle \langle \mathbf{F}_{tide}^{\text{horz}} \rangle \rangle^{n+\frac{1}{2}} + \frac{1}{\rho_{0}} \Delta t_{cl} \langle \langle \tau_{tide}^{\text{btm}} \rangle \rangle^{n+\frac{1}{2}},$$
(6.80)

$$\langle \mathbf{U}_b \rangle^{n+1} = \mathbf{U}_b^0 - \Delta t_{\rm cl} f \mathbf{k} \times \langle \langle \mathbf{U}_b \rangle \rangle^{n+\frac{1}{2}} - \Delta t_{\rm cl} g(\eta^n + H) \left\langle \left\langle \nabla \eta_b \right\rangle \right\rangle^{n+\frac{1}{2}} + \Delta t_{\rm cl} \left(\mathbf{X} - \mathbf{F}_{lt}^{\rm horz} - \frac{\tau_{lt}^{\rm out}}{\rho_0} \right)^n.$$
(6.81)

These are summed and then combined with the baroclinic part (6.39) and (6.40) to give

$$\frac{(\langle \mathbf{u} \rangle^{n+1} + \mathbf{u}_{k-\frac{1}{2}}^{'} - \overline{\mathbf{u}^{\prime}}^{c}) \Delta z_{k-\frac{1}{2}}^{n+1} - \mathbf{u}_{k-\frac{1}{2}}^{n} \Delta z_{k-\frac{1}{2}}^{n}}{\Delta t_{cl}} = -f\mathbf{k} \times \langle \langle \mathbf{u} \rangle \rangle^{n+\frac{1}{2}} \Delta z_{k-\frac{1}{2}}^{n+1} - 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}}}^{c} \Delta z_{k-\frac{1}{2}}^{n+1}}{-g\Delta z_{k-\frac{1}{2}}^{n+1} \frac{\eta^{n} + H}{\eta^{n+1} + H} \langle \langle \nabla(\eta_{b} + \alpha \eta_{lt} - \beta \eta_{eq}) \rangle \rangle^{n+\frac{1}{2}}}{+\Delta z_{k-\frac{1}{2}}^{n+1} \frac{\langle \langle \mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lt}^{horz})^{n}}{\eta^{n+1} + H} + \Delta z_{k-\frac{1}{2}}^{n+1} \frac{\langle \langle \tau_{lide}^{bm} \rangle \rangle^{n+\frac{1}{2}} - (\tau_{lt}^{btm})^{n}}{\rho_{0}(\eta^{n+1} + H)} + \Delta z_{k-\frac{1}{2}}^{n+1} \mathbf{F}_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lt}^{horz})^{n}}{\rho_{0}(\eta^{n+1} + H)} + \Delta z_{k-\frac{1}{2}}^{n+1} \mathbf{F}_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lt}^{horz})^{n}}{\rho_{0}(\eta^{n+1} + H)} + \Delta z_{k-\frac{1}{2}}^{n+1} \mathbf{F}_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lt}^{horz})^{n}}{\rho_{0}(\eta^{n+1} + H)} + \Delta z_{k-\frac{1}{2}}^{n+1} \mathbf{F}_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lt}^{horz})^{n}}{\rho_{0}(\eta^{n+1} + H)} + \Delta z_{k-\frac{1}{2}}^{n+1} \mathbf{F}_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lt}^{horz})^{n}}{\rho_{0}(\eta^{n+1} + H)} + \Delta z_{k-\frac{1}{2}}^{n+1} \mathbf{F}_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lide}^{horz})^{n}}{\rho_{0}(\eta^{n+1} + H)} + \Delta z_{k-\frac{1}{2}}^{n+1} \mathbf{F}_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lide}^{horz})^{n}}{\rho_{0}(\eta^{n+1} + H)} + \Delta z_{k-\frac{1}{2}}^{n+1} \mathbf{F}_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lide}^{horz})^{n}}{\rho_{0}(\eta^{n+1} + H)}} + \Delta z_{k-\frac{1}{2}}^{n+1} \mathbf{F}_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lide}^{horz})^{n+\frac{1}{2}} - (\mathbf{F}_{lide}^{horz})^{n}}{\rho_{0}(\eta^{n+1} + H)} + \Delta z_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lide}^{horz})^{n}}{\rho_{0}(\eta^{n+1} + H)}} + \Delta z_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide}^{horz} \rangle^{n}}{\rho_{0}(\eta^{n+1} + H)}} + \Delta z_{k-\frac{1}{2}}^{n} \frac{\langle (\mathbf{F}_{lide$$

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 in (6.82) would cancel when barotropic and baroclinic mode are synchronous. In MRI.COM, the bottom friction for the linear tidal component is calculated as follows:

$$\tau_{tide}^{\text{btm}} = -\rho_0 C_D \Big| \frac{\mathbf{U}_b + \mathbf{U}_{lt}}{\eta + H} \Big| \mathbf{T}_\theta \frac{\mathbf{U}_{lt}}{\eta + H}.$$
(6.83)

Horizontal viscosity parameterization is the Laplacian operator with the viscosity coefficient fixed in time.

The solution method to predict the two components of the barotropic mode is schematically illustrated by Figure 6.4. On starting the barotropic submodel, the total barotropic mode is split into the basic and the linear tidal components $(U_b, \eta_b, U_{lt}, \text{ and } \eta_{lt})$ 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.



Figure 6.4 Schematic figure of the separating basic and linear tidal component of the barotropic mode

6.5.3 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 18 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 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 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 U_{lt} and η_{lt} in the same format as the parent model output. The MXE (see Section 21.6) 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).

6.6 Usage

Behavior of the barotropic model at run time is specified by the three namelist blocks shown on Tables 6.2 through 6.4. 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 (unnecessary if $1_global_local_cnsv_ssh = .false.$).

In addition, following model options are available.

TIDE: Tide producing forcing is activated

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. See Section 6.5 for details.

SLP: Sea surface is elevated/depressed according to surface atmospheric pressure Give atmospheric sea-level pressure, p_a in (6.1) and (6.2), 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 6.2 for the time discretization method.

See docs/README.Namelist for namelist details.

$Table 6.2 Namelist \verb"nml_barotropic_model" for the specifying basic features of the barotropic model (see Section$	<mark>6.6</mark>)
--	--------------------

variable	units	description	usage
dt_barotropic_sec	sec	time step interval	required
l_global_local_cnsv_ssh	logical	how to predict sea surface height	default = .true.
		(Section 6.3.1)	

 Table6.3
 Namelist nml_barotropic_run for starting the barotropic model (see Section 6.6)

variable	units	description	usage
l_rst_barotropic_in	logical	read initial restart for sea surface	default = 1_rst_in
		and transport or not	
l_rst_barotropic_dflx_in	logical	read initial restart for diffusive	<pre>default = 1_rst_barotropic_in,</pre>
		flux of SSH or not (Section 6.4)	valid only when
			l_global_local_cnsv_ssh = .true.

Table0.4 Namensi funt_barotropre_arri for specifying 5511 anasion (see Section 0.0)

variable	units	description	usage
ssh_diff_cm2ps	$\mathrm{cm}^2\mathrm{s}^{-1}$	diffusivity for SSH (Section 6.4)	default = zero

Chapter 7

Equations of motion (baroclinic component)

This chapter explains the advection terms (Section 7.1), the pressure gradient terms (Section 7.2), and the viscosity terms (Section 7.3) of the hydrostatic momentum equation. The basic equations (2.33) and (2.34) in Chapter 2 are re-written:

$$\frac{\partial(z_{s}u)}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial(z_{s}h_{\psi}uu)}{\partial\mu} + \frac{\partial(z_{s}h_{\mu}vu)}{\partial\psi} \right\} + \frac{\partial(z_{s}\dot{s}u)}{\partial s} + z_{s}\frac{v}{h_{\mu}h_{\psi}} \left(\frac{\partial h_{\mu}}{\partial\psi}u - \frac{\partial h_{\psi}}{\partial\mu}v \right) - z_{s}fv$$

$$= -z_{s}\frac{1}{\rho_{0}h_{\mu}}\frac{\partial(p_{a} + g\rho_{0}\eta)}{\partial\mu} - z_{s}\frac{1}{\rho_{0}h_{\mu}}\frac{\partial}{\partial\mu}\left(g\int_{z(z^{*})}^{\eta}\rho'dz\right) - z_{s}\frac{g\rho'}{\rho_{0}h_{\mu}}\frac{\partial z}{\partial\mu}$$

$$+ z_{s}\frac{1}{\rho_{0}}\left(\nabla \cdot \tau_{\text{horizontal strain}}\right)u + z_{s}\frac{1}{z_{s}}\frac{\partial}{\partial s}\left(\frac{v_{V}}{z_{s}}\frac{\partial u}{\partial s}\right), \qquad (7.1)$$

$$\frac{\partial(z_{s}v)}{\partial t} + \frac{1}{h_{\mu}h_{\psi}}\left\{\frac{\partial(z_{s}h_{\psi}uv)}{\partial\mu} + \frac{\partial(z_{s}h_{\mu}vv)}{\partial\psi}\right\} + \frac{\partial(z_{s}\dot{s}v)}{\partial s} + z_{s}\frac{u}{h_{\mu}h_{\psi}}\left(\frac{\partial h_{\psi}}{\partial\mu}v - \frac{\partial h_{\psi}}{\partial\psi}u\right) + z_{s}fu$$

$$= -z_{s}\frac{1}{\rho_{0}h_{\psi}}\frac{\partial(p_{a} + g\rho_{0}\eta)}{\partial\psi} - z_{s}\frac{1}{\rho_{0}h_{\psi}}\frac{\partial}{\partial\psi}\left(g\int_{z(z^{*})}^{\eta}\rho'dz\right) - z_{s}\frac{g\rho'}{\rho_{0}h_{\psi}}\frac{\partial z}{\partial\psi}$$

$$+ z_{s}\frac{1}{\rho_{0}}(\nabla \cdot \tau_{\text{horizontal strain}})_{v} + z_{s}\frac{1}{z_{s}}\frac{\partial}{\partial s}\left(\frac{v_{V}}{z_{s}}\frac{\partial v}{\partial s}\right). \qquad (7.2)$$

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 7.1 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.

7.1 Advection terms

Chapter 5 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 (5.21) of those for the T-cell mass continuity (5.12) to (5.18). 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)).

7.1.1 Vertical mass fluxes and its momentum advection

According to the definition (5.12) and (5.21) in Chapter 5, the vertical mass flux at the upper surface, level k, of the U-cell $(i + \frac{1}{2}, j + \frac{1}{2}, k + \frac{1}{2}), \overline{W}_{i+\frac{1}{2},j+\frac{1}{2},k}^{U}$, is defined by surrounding W^{T} as

$$\overline{W}_{i+\frac{1}{2},j+\frac{1}{2},k}^{U} = \frac{W_{i,j,k}^{T}}{N_{i,j,k+\frac{1}{2}}} + \frac{W_{i+1,j,k}^{T}}{N_{i+1,j,k+\frac{1}{2}}} + \frac{W_{i,j+1,k}^{T}}{N_{i,j+1,k+\frac{1}{2}}} + \frac{W_{i+1,j+1,k}^{T}}{N_{i+1,j+1,k+\frac{1}{2}}},$$
(7.3)

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 $(i + \frac{1}{2}, j + \frac{1}{2}, k - \frac{1}{2})$, $W_{i+\frac{1}{2}, j+\frac{1}{2}, k}^U$ is defined as

$$\mathcal{W}_{i+\frac{1}{2},j+\frac{1}{2},k}^{U} = \frac{W_{i,j,k}^{T}}{N_{i,j,k-\frac{1}{2}}} + \frac{W_{i+1,j,k}^{T}}{N_{i+1,j,k-\frac{1}{2}}} + \frac{W_{i,j+1,k}^{T}}{N_{i,j+1,k-\frac{1}{2}}} + \frac{W_{i+1,j+1,k}^{T}}{N_{i+1,j+1,k-\frac{1}{2}}}.$$
(7.4)

Though W^T are continuous at the boundary of vertically adjacent T-cells, $\overline{W}_{i+\frac{1}{2},j+\frac{1}{2},k}^U$ and $\mathcal{W}_{i+\frac{1}{2},j+\frac{1}{2},k}^U$ 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 7.1(a) indicates the mass continuity for T-cells to conserve barotropy in shallow regions (the U-points are just intermediate between T-points). Figure 7.1(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.

(a)

(b)



Figure 7.1 (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 7.2). 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 \mathbf{a} , \mathbf{b} , \mathbf{c} , and \mathbf{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.



Figure 7.2 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 7.3). 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 7.4). 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 ($1 \le N_l < N_u \le 4$). Each cell in the lower layer carries W^T/N_l , and W^T/N_u of it goes vertically upward. The remaining

$$W^{T}/N_{l} - W^{T}/N_{u} = W^{T}(N_{u} - N_{l})/(N_{l}N_{u})$$
(7.5)

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

$$W^{T}(N_{u} - N_{l})/(N_{l}N_{u}) \times 1/(N_{u} - N_{l}) = W^{T}/(N_{l}N_{u}).$$
(7.6)

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

$$W^T / (N_l N_u) \times N_l = W^T / N_u. \tag{7.7}$$



Figure 7.3 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.



Figure 7.4 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.

Based on these discussions we understand the difference between (7.3) and (7.4).

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

$$\mathbf{a}_l W^T / (N_l N_u)$$
 and $\mathbf{a}_l W^T (u_{\mathbf{a}_l} + u_{\mathbf{d}_u}) / (2N_l N_u),$ (7.8)

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

$$\mathbf{a}_l W^T / N_u \quad \text{and} \quad \mathbf{a}_l W^T (u_{\mathbf{a}_l} + u_{\mathbf{a}_u}) / (2N_u), \tag{7.9}$$

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 (7.9) 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{split} 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_{i,j,k-\frac{1}{2}}^{T}}{N_{i,j,k-\frac{1}{2}}} + \frac{W_{i,j+1,k}^{T}}{N_{i,j+1,k-\frac{1}{2}}} + \frac{W_{i+1,j+1,k}^{T}}{N_{i+1,j+1,k-\frac{1}{2}}} \right] \end{split} \tag{7.10} \\ &+ \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 \\ &\left[\left\{ 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) \right\} \times \frac{W_{i,j,k}^{T}}{N_{i,j,k+\frac{1}{2}}N_{i,j,k-\frac{1}{2}}} \\ &+ \left\{ 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) \right\} \times \frac{W_{i,j,k}^{T}}{N_{i,j,k+\frac{1}{2}}N_{i,j,k-\frac{1}{2}}} \\ &+ \left\{ 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) \\ &+ \left\{ 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{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) \\ &+ \left\{ 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{3}{2},k+\frac{1}{2}} \right) \\ &+ \left\{ 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{3}{2},k+\frac{1}{2}} \right) \right\} \times \frac{W_{i,j+1,k+\frac{1}{2}}^{T}}{N_{i,j+1,k+\frac{1}{2}}N_{i,j+1,k-\frac{1}{2}}} \\ &+ \left\{ 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{3}{2},k+\frac{1}{2}} \right) \right\} \times \frac{W_{i,j+1,k+\frac{1}{2}}}^{T}}{N_{i,j+1,k+\frac{1}{2}}N_{i,j+1,k-\frac{1}{2}}} \\ &+ \left\{ 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{3}{2},k+\frac{1}{2}} \right) \right\} \times \frac{W_{i,j+1,k+\frac{1}{2}}}^{T}}{N_{i,j+1,k+\frac{1}{2}}} \left\{ u_{$$

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.

7.1.2 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 (5.12)-(5.18). 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 $U_{i+\frac{1}{2},j}^T$ and $V_{i,j+\frac{1}{2}}^T$ are given as

$$U_{i+\frac{1}{2},j}^{T} = \frac{1}{2} (e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u_{i+\frac{1}{2},j}^{*} \Delta y_{i+\frac{1}{2},j} \Delta z$$

and
$$V_{i,j+\frac{1}{2}}^{T} = \frac{1}{2} (e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) v_{i,j+\frac{1}{2}}^{*} \Delta x_{i,j+\frac{1}{2}} \Delta z,$$
(7.11)

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) as computed by (5.14), (5.16), and (5.18). Here we neglect the vertical subscript $(k - \frac{1}{2})$.

Substituting these formulae into the T-cell mass continuity (5.12), the X (zonal) component of the mass continuity for

U-cell $(i + \frac{1}{2}, j + \frac{1}{2})$ (5.21), 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

$$\begin{aligned} \operatorname{XMC}_{i+\frac{1}{2},j+\frac{1}{2}}^{U} &= e_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \Delta z \\ &\times \Big\{ \frac{1}{N_{i,j}} \Big[(e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}}) u_{i-\frac{1}{2},j}^{*} - (e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u_{i+\frac{1}{2},j}^{*} \Big] \\ &+ \frac{1}{N_{i+1,j}} \Big[(e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u_{i+\frac{1}{2},j}^{*} - (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}}) u_{i+\frac{1}{2},j}^{*} \Big] \\ &+ \frac{1}{N_{i,j+1}} \Big[(e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}}) u_{i+\frac{1}{2},j+1}^{*} - (e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) u_{i+\frac{1}{2},j+1}^{*} \Big] \\ &+ \frac{1}{N_{i+1,j+1}} \Big[(e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) u_{i+\frac{1}{2},j+1}^{*} - (e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}}) u_{i+\frac{3}{2},j+1}^{*} \Big] \Big\} \\ &= e_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \Delta z \times \Big\{ \Big[\frac{1}{N_{i,j}} (e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}}) u_{i+\frac{1}{2},j}^{*} \\ &+ \Big(-\frac{1}{N_{i,j}} + \frac{1}{N_{i+1,j}} \Big) (e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u_{i+\frac{1}{2},j}^{*} \\ &+ \Big(-\frac{1}{N_{i+1,j}} (e_{i-\frac{3}{2},j-\frac{1}{2}} + e_{i-\frac{3}{2},j+\frac{1}{2}}) u_{i+\frac{3}{2},j}^{*} \Big] \Big] \\ &+ \Big[\frac{1}{N_{i,j+1}} (e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}}) u_{i+\frac{3}{2},j+1}^{*} \\ &+ \Big(-\frac{1}{N_{i,j+1}} + \frac{1}{N_{i+1,j+1}} \Big) (e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) u_{i+\frac{3}{2},j+1}^{*} \Big] \Big\}. \tag{7.12}$$

Here, recalling

$$N_{i,j} = e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}},$$
(7.13)

we have,

$$\left(-\frac{1}{N_{i,j}} + \frac{1}{N_{i+1,j}} \right) (e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}})$$

$$= \frac{1}{N_{i,j}} (e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}}) - \frac{1}{N_{i+1,j}} (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}})$$

$$(7.14)$$

and

$$\left(-\frac{1}{N_{i,j+1}} + \frac{1}{N_{i+1,j+1}}\right)\left(e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}\right) = \frac{1}{N_{i,j+1}}\left(e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}}\right) - \frac{1}{N_{i+1,j+1}}\left(e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}}\right).$$
(7.15)

Chapter 7 Equations of motion (baroclinic component)

Thus, based on (5.24) and (5.25),

$$\begin{aligned} \text{XMC}_{i+\frac{1}{2},j+\frac{1}{2}}^{U} &= e_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \Delta z \Big\{ \Big[\frac{1}{N_{i,j}} (e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}}) (u_{i-\frac{1}{2},j}^{*} + u_{i+\frac{1}{2},j}^{*}) \\ &\quad - \frac{1}{N_{i+1,j}} (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}}) (u_{i+\frac{1}{2},j}^{*} + u_{i+\frac{3}{2},j}^{*}) \Big] \\ &\quad + \Big[\frac{1}{N_{i,j+1}} (e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}}) (u_{i-\frac{1}{2},j+1}^{*} + u_{i+\frac{1}{2},j+1}^{*}) \\ &\quad - \frac{1}{N_{i+1,j+1}} (e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}}) (u_{i+\frac{1}{2},j+1}^{*} + u_{i+\frac{3}{2},j+1}^{*}) \Big] \Big\} \\ &\quad = e_{i+\frac{1}{2},j+\frac{1}{2}} \Big[e_{i-\frac{1}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i,j}} U_{i,j}^{U} + \frac{1}{N_{i,j+1}} U_{i,j+1}^{U} \Big) \\ &\quad - e_{i+\frac{3}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i+1,j}} U_{i+1,j}^{U} + \frac{1}{N_{i+1,j+1}} U_{i+1,j+1}^{U} \Big) + e_{i-\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} U_{i,j}^{U} \\ &\quad - e_{i+\frac{3}{2},j+\frac{3}{2}} \frac{1}{N_{i+1,j+1}} U_{i+1,j+1}^{U} + e_{i-\frac{1}{2},j+\frac{3}{2}} \frac{1}{N_{i,j+1}} U_{i,j+1}^{U} - e_{i+\frac{3}{2},j-\frac{1}{2}} \frac{1}{N_{i+1,j}} U_{i+1,j}^{U} \Big]. \end{aligned}$$

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:

$$\begin{split} \mathrm{HMC}_{i+\frac{1}{2},j+\frac{1}{2}}^{U} &= \mathrm{XMC}_{i+\frac{1}{2},j+\frac{1}{2}}^{U} + \mathrm{YMC}_{i+\frac{1}{2},j+\frac{1}{2}}^{U} \\ &= e_{i+\frac{1}{2},j+\frac{1}{2}} \\ &\times \left[e_{i-\frac{1}{2},j+\frac{1}{2}} \left(\frac{1}{N_{i,j}} U_{i,j}^{U} + \frac{1}{N_{i,j+1}} U_{i,j+1}^{U} \right) - e_{i+\frac{3}{2},j+\frac{1}{2}} \left(\frac{1}{N_{i+1,j}} U_{i+1,j}^{U} + \frac{1}{N_{i+1,j+1}} U_{i+1,j+1}^{U} \right) \\ &+ e_{i+\frac{1}{2},j-\frac{1}{2}} \left(\frac{1}{N_{i,j}} V_{i,j}^{U} + \frac{1}{N_{i+1,j}} V_{i+1,j}^{U} \right) - e_{i+\frac{1}{2},j+\frac{3}{2}} \left(\frac{1}{N_{i,j+1}} V_{i,j+1}^{U} + \frac{1}{N_{i+1,j+1}} V_{i+1,j+1}^{U} \right) \\ &+ e_{i-\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} (U_{i,j}^{U} + V_{i,j}^{U}) - e_{i+\frac{3}{2},j+\frac{3}{2}} \frac{1}{N_{i+1,j+1}} (U_{i+1,j+1}^{U} + V_{i+1,j+1}^{U}) \\ &+ e_{i-\frac{1}{2},j+\frac{3}{2}} \frac{1}{N_{i,j+1}} (U_{i,j+1}^{U} - V_{i,j+1}^{U}) - e_{i+\frac{3}{2},j-\frac{1}{2}} \frac{1}{N_{i+1,j}} (U_{i+1,j}^{U} - V_{i+1,j}^{U}) \right]. \end{split}$$
(7.17)

Assuming mass fluxes $M_{\rm E}$, $M_{\rm N}$, $M_{\rm NE}$, $M_{\rm SE}$ as follows:

$$M_{\mathbf{E}_{i,j+\frac{1}{2}}} = e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j+\frac{1}{2}} \left(\frac{1}{N_{i,j}} U_{i,j}^{U} + \frac{1}{N_{i,j+1}} U_{i,j+1}^{U} \right),$$

$$M_{\mathbf{N}_{i+\frac{1}{2},j}} = e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i+\frac{1}{2},j-\frac{1}{2}} \left(\frac{1}{N_{i,j}} V_{i,j}^{U} + \frac{1}{N_{i+1,j}} V_{i+1,j}^{U} \right),$$

$$M_{\mathbf{N}\mathbf{E}_{i,j}} = e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} (U_{i,j}^{U} + V_{i,j}^{U}),$$

$$M_{\mathbf{S}\mathbf{E}_{i,j}} = e_{i-\frac{1}{2},j+\frac{1}{2}} e_{i+\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} (U_{i,j}^{U} - V_{i,j}^{U}),$$
(7.18)

then,

$$HMC_{i+\frac{1}{2},j+\frac{1}{2}}^{U} = M_{\mathbf{E}_{i,j+\frac{1}{2}}} - M_{\mathbf{E}_{i+1,j+\frac{1}{2}}} + M_{\mathbf{N}_{i+\frac{1}{2},j}} - M_{\mathbf{N}_{i+\frac{1}{2},j+1}} + M_{\mathbf{N}\mathbf{E}_{i,j}} - M_{\mathbf{N}\mathbf{E}_{i+1,j+1}} + M_{\mathbf{S}\mathbf{E}_{i,j+1}} - M_{\mathbf{S}\mathbf{E}_{i+1,j}}.$$
(7.19)

Here, $M_{\rm E}$ and $M_{\rm N}$ are axis-parallel mass fluxes, and $M_{\rm NE}$ and $M_{\rm SE}$ are horizontally diagonal ones (Figure 7.5).



Figure 7.5 Distribution of generalized mass fluxes for U-cell (i + 1/2, j + 1/2)

If we derive the formula for the standard case from (7.17) (all of *N* are 4),

$$\begin{aligned} \text{HMC}_{i+\frac{1}{2},j+\frac{1}{2}}^{U} &= \frac{1}{2} \left[\frac{1}{2} (U_{i,j}^{U} + U_{i,j+1}^{U}) - \frac{1}{2} (U_{i+1,j}^{U} + U_{i+1,j+1}^{U}) \right. \\ &\quad + \frac{1}{2} (V_{i,j}^{U} + V_{i,j+1}^{U}) - \frac{1}{2} (V_{i+1,j}^{U} + V_{i+1,j+1}^{U}) \right] \\ &\quad + \frac{1}{2} \left[\frac{1}{2} (U_{i,j}^{U} + V_{i,j}^{U}) - \frac{1}{2} (U_{i+1,j+1}^{U} + V_{i+1,j+1}^{U}) \right. \\ &\quad + \frac{1}{2} (U_{i,j+1}^{U} - V_{i,j+1}^{U}) - \frac{1}{2} (U_{i+1,j}^{U} - V_{i+1,j}^{U}) \right]. \end{aligned}$$
(7.20)

This expression means that the horizontal mass flux convergence is a mean of those of the axis-parallel mass fluxes (5.23) and of the diagonal ones (5.28). 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 (7.20). 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

Chapter 7 Equations of motion (baroclinic component)

Table7.1 Definition of a land-sea index, the index identifying each case (column **A**), the coefficient of $U_{i,j}^U$ in the axis-parallel mass flux (column **B**), and the coefficient of $U_{i,j}^U$ in the horizontally diagonal mass flux (column **C**), for eight combinations of indices **a**, **b**, and **c** in Figure 7.2b (see the main text). Cell **d** is assumed to be a sea cell, and the momentum advection by means of $U_{i,j}^U$ into and from cell **d** is generalized. Note that $U_{i,j}^U$ is identically zero for cases 5 and 8 (**b** = **c** = 0).

			В	С
			Coefficient of $U_{i,i}^U$	Coefficient of $U_{i,i}^U$
CASE	Land-sea index		(axis-parallel)	(horizontally-diagonal)
n	a b c	Α	+	×
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:

$$\begin{aligned} \operatorname{CAD}_{i+\frac{1}{2},j+\frac{1}{2}}(u) &= \frac{2}{3} \left[\frac{1}{2} (u_{i-\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2}}) \frac{1}{2} (U_{i,j}^{U} + U_{i,j+1}^{U}) \\ &\quad - \frac{1}{2} (u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{3}{2},j+\frac{1}{2}}) \frac{1}{2} (U_{i+1,j}^{U} + U_{i+1,j+1}^{U}) \\ &\quad + \frac{1}{2} (u_{i+\frac{1}{2},j-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2}}) \frac{1}{2} (V_{i,j}^{U} + V_{i+1,j}^{U}) \\ &\quad - \frac{1}{2} (u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{3}{2}}) \frac{1}{2} (V_{i,j+1}^{U} + V_{i+1,j+1}^{U}) \\ &\quad + \frac{1}{3} \left[\frac{1}{2} (u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i-\frac{1}{2},j-\frac{1}{2}}) \frac{1}{2} (U_{i,j}^{U} + V_{i,j}^{U}) \\ &\quad - \frac{1}{2} (u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{3}{2},j+\frac{3}{2}}) \frac{1}{2} (U_{i+1,j+1}^{U} + V_{i+1,j+1}^{U}) \\ &\quad + \frac{1}{2} (u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i-\frac{1}{2},j+\frac{3}{2}}) \frac{1}{2} (U_{i,j+1}^{U} - V_{i,j+1}^{U}) \\ &\quad - \frac{1}{2} (u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{3}{2},j-\frac{1}{2}}) \frac{1}{2} (U_{i+1,j}^{U} - V_{i+1,j}^{U}) \\ &\quad - \frac{1}{2} (u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{3}{2},j-\frac{1}{2}}) \frac{1}{2} (U_{i+1,j}^{U} - V_{i+1,j}^{U}) \end{aligned}$$
(7.21)

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 7.5 and related momentum flux, let us examine the axis-parallel and horizontally diagonal mass flux associated with $U_{i,j}^U$, taking topography into account. Look at Figure 7.2b, 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 $U_{i,j}^U$ under different combinations of **a**, **b**, and **c** (eight cases), as indicated in the first column in Table 7.1. 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 $U_{i,j}^U$ in the axis-parallel mass flux of the U-cell mass continuity (7.17). Column (C) indicates the coefficient of $U_{i,j}^U$ in the horizontally diagonal mass flux of (7.17).

the coefficient of $U_{i,j}^U$ in the horizontally diagonal mass flux of (7.17). The generalized coefficient of $U_{i,j}^U$ 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,

$$c_{1} = \sum_{n=1}^{8} \mathbf{A}_{n} \mathbf{B}_{n} = \frac{1}{6} \mathbf{c} (\mathbf{a}\mathbf{b} - \mathbf{a} - \mathbf{b} + 3)$$

and
$$c_{2} = \sum_{n=1}^{8} \mathbf{A}_{n} \mathbf{C}_{n} = \frac{1}{6} \mathbf{b} (3 - \mathbf{a} - \mathbf{c}).$$
 (7.22)

Then, the axis-parallel and the horizontally diagonal flux of zonal momentum (*u*) related with $U_{i,j}^U$, multiplied by the land-sea index **d**, are

$$\frac{\mathbf{d}}{2}(u_{\mathbf{c}} + u_{\mathbf{d}})c_{1}U_{i,j}^{U} = \frac{1}{2}(u_{\mathbf{c}} + u_{\mathbf{d}})\frac{1}{6}\mathbf{cd}(\mathbf{ab} - \mathbf{a} - \mathbf{b} + 3)U_{i,j}^{U},$$

and
$$\frac{\mathbf{d}}{2}(u_{\mathbf{b}} + u_{\mathbf{d}})c_{2}U_{i,j}^{U} = \frac{1}{2}(u_{\mathbf{b}} + u_{\mathbf{d}})\frac{1}{6}\mathbf{bd}(3 - \mathbf{a} - \mathbf{c})U_{i,j}^{U},$$
(7.23)

respectively.

The resultant momentum fluxes are as follows:

$$F_{\mathbf{E}_{i,j+\frac{1}{2}}}(u) = \frac{1}{2} (u_{i-\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2}}) M_{\mathbf{E}_{i,j+\frac{1}{2}}},$$

$$F_{\mathbf{N}_{i+\frac{1}{2},j}}(u) = \frac{1}{2} (u_{i+\frac{1}{2},j-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2}}) M_{\mathbf{N}_{i+\frac{1}{2},j}},$$

$$F_{\mathbf{NE}_{i,j}}(u) = \frac{1}{2} (u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i-\frac{1}{2},j-\frac{1}{2}}) M_{\mathbf{NE}_{i,j}},$$

$$F_{\mathbf{SE}_{i,j}}(u) = \frac{1}{2} (u_{i-\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}}) M_{\mathbf{SE}_{i,j}},$$
(7.24)

where

$$M_{\mathbf{E}_{i,j+\frac{1}{2}}} = \frac{1}{6} (C_{\mathbf{XN}_{i,j}} U_{i,j}^{U} + C_{\mathbf{XS}_{i,j+1}} U_{i,j+1}^{U}),$$

$$M_{\mathbf{N}_{i+\frac{1}{2},j}} = \frac{1}{6} (C_{\mathbf{YE}_{i,j}} V_{i,j}^{U} + C_{\mathbf{YW}_{i+1,j}} V_{i+1,j}^{U}),$$

$$M_{\mathbf{NE}_{i,j}} = \frac{1}{6} C_{\mathbf{NE}_{i,j}} (U_{i,j}^{U} + V_{i,j}^{U}),$$

$$M_{\mathbf{SE}_{i,j}} = \frac{1}{6} C_{\mathbf{SE}_{i,j}} (U_{i,j}^{U} - V_{i,j}^{U}),$$
(7.25)

and

$$C_{\mathbf{XN}_{i,j}} = e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j+\frac{1}{2}} (e_{i+\frac{1}{2},j-\frac{1}{2}} e_{i-\frac{1}{2},j-\frac{1}{2}} - e_{i+\frac{1}{2},j-\frac{1}{2}} - e_{i-\frac{1}{2},j-\frac{1}{2}} + 3),$$

$$C_{\mathbf{XS}_{i,j}} = e_{i+\frac{1}{2},j-\frac{1}{2}} e_{i-\frac{1}{2},j-\frac{1}{2}} (e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j+\frac{1}{2}} - e_{i-\frac{1}{2},j+\frac{1}{2}} - e_{i-\frac{1}{2},j+\frac{1}{2}} + 3),$$

$$C_{\mathbf{YE}_{i,j}} = e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i+\frac{1}{2},j-\frac{1}{2}} (e_{i-\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j-\frac{1}{2}} - e_{i-\frac{1}{2},j+\frac{1}{2}} - e_{i-\frac{1}{2},j-\frac{1}{2}} + 3),$$

$$C_{\mathbf{YW}_{i,j}} = e_{i-\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j-\frac{1}{2}} (e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i+\frac{1}{2},j-\frac{1}{2}} - e_{i+\frac{1}{2},j+\frac{1}{2}} - e_{i+\frac{1}{2},j-\frac{1}{2}} + 3),$$

$$C_{\mathbf{NE}_{i,j}} = e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j-\frac{1}{2}} (3 - e_{i-\frac{1}{2},j+\frac{1}{2}} - e_{i+\frac{1}{2},j-\frac{1}{2}}),$$

$$C_{\mathbf{SE}_{i,j}} = e_{i-\frac{1}{2},j+\frac{1}{2}} e_{i+\frac{1}{2},j-\frac{1}{2}} (3 - e_{i+\frac{1}{2},j+\frac{1}{2}} - e_{i-\frac{1}{2},j-\frac{1}{2}}).$$

$$(7.26)$$

Finally, convergence of the horizontal momentum fluxes is written as

$$CAD_{i+\frac{1}{2},j+\frac{1}{2}}(u) = F_{\mathbf{E}_{i,j+\frac{1}{2}}}(u) - F_{\mathbf{E}_{i+1,j+\frac{1}{2}}}(u) + F_{\mathbf{N}_{i+\frac{1}{2},j}}(u) - F_{\mathbf{N}_{i+\frac{1}{2},j+1}}(u) + F_{\mathbf{N}\mathbf{E}_{i,j}}(u) - F_{\mathbf{N}\mathbf{E}_{i+1,j+1}}(u) + F_{\mathbf{S}\mathbf{E}_{i,j+1}}(u) - F_{\mathbf{S}\mathbf{E}_{i+1,j}}(u).$$
(7.27)

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).

Chapter 7 Equations of motion (baroclinic component)

7.2 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. 2.58). Its form is

$$\underbrace{\frac{1}{\rho_0}\nabla_s(p_a + g\rho_0\eta)}_{\text{fast}} + \underbrace{\frac{1}{\rho_0}\nabla_s\left[g\int_{z(s)}^{\eta}\rho'dz\right]}_{\text{slow}} + \underbrace{\frac{g\rho'}{\rho_0}\nabla_sz}_{\text{slow}}.$$
(7.28)

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 (7.1) and (7.2). 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 (7.28), 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.



Figure 7.6 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 (Δ), 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 7.6 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. (7.28) is expressed as

$$\frac{1}{\rho_0} \nabla_s \left[g \int_{z(s)}^{\eta} \rho' dz \right] = \frac{1}{\rho_0} \nabla_s p', \tag{7.29}$$

where
$$p' = g \int_{z(s)}^{\eta} \rho' dz.$$
 (7.30)

A simple discrete expression would be given as follows $(1/\rho_0 \text{ is omitted})$:

$$(\nabla_{s}p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = \hat{\mathbf{x}} \frac{\frac{p'_{i+1,j+1,k-\frac{1}{2}} + p'_{i+1,j,k-\frac{1}{2}}}{2} - \frac{p'_{i,j+1,k-\frac{1}{2}} + p'_{i,j,k-\frac{1}{2}}}{2}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} + \hat{\mathbf{y}} \frac{\frac{p'_{i+1,j+1,k-\frac{1}{2}} + p'_{i,j+1,k-\frac{1}{2}}}{2} - \frac{p'_{i+1,j,k-\frac{1}{2}} + p'_{i,j,k-\frac{1}{2}}}{2}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}, \quad (7.31)$$

where
$$p'_{i,j,k-\frac{1}{2}} = g \sum_{l=1}^{k-1} (\rho')_{i,j,l-\frac{1}{2}} (dzt)_{i,j,l-\frac{1}{2}} + g(\rho')_{i,j,k-\frac{1}{2}} \frac{(dzt)_{i,j,k-\frac{1}{2}}}{2}.$$
 (7.32)

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:

$$\begin{aligned} (\nabla_{s}p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} &= (\nabla_{s}p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{3}{2}} + \left\{ (\nabla_{s}p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} - (\nabla_{s}p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{3}{2}} \right\} \\ &= (\nabla_{s}p')_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{3}{2}} + \hat{\mathbf{x}} \frac{\frac{\Delta p'_{i+1,j+1,k-1} + \Delta p'_{i+1,j,k-1}}{2} - \frac{\Delta p'_{i,j+1,k-1} + \Delta p'_{i,j,k-1}}{2}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} \\ &+ \hat{\mathbf{y}} \frac{\frac{\Delta p'_{i+1,j+1,k-1} + \Delta p'_{i,j+1,k-1}}{2} - \frac{\Delta p'_{i+1,j,k-1} + \Delta p'_{i,j,k-1}}{2}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}, \end{aligned}$$
(7.33)

where

$$\Delta p'_{i,j,k-1} \equiv p'_{i,j,k-\frac{1}{2}} - p'_{i,j,k-\frac{3}{2}} = g(\rho')_{i,j,k-\frac{3}{2}} \frac{(\mathsf{dzt})_{i,j,k-\frac{1}{2}}}{2} + g(\rho')_{i,j,k-\frac{1}{2}} \frac{(\mathsf{dzt})_{i,j,k-\frac{1}{2}}}{2}.$$
(7.34)

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,

$$\frac{g\rho'}{\rho_0}\nabla_s z = \frac{g\rho'}{\rho_0}\nabla_s z'(s).$$
(7.35)

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

$$\left[\frac{g\rho'}{\rho_{0}}\nabla_{s}z'\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\ = \frac{g}{\rho_{0}}\left\{\left[\frac{\frac{\rho'_{i+1,j}+\rho'_{i,j}}{2}\frac{z'_{i+1,j}-z'_{i,j}}{\Delta x_{i+\frac{1}{2},j}} + \frac{\rho'_{i+1,j+1}+\rho'_{i,j+1}}{2}\frac{z'_{i+1,j+1}-z'_{i,j+1}}{\Delta x_{i+\frac{1}{2},j+1}}}{2}\hat{\mathbf{x}}\right] + \left[\frac{\frac{\rho'_{i,j+1}+\rho'_{i,j}}{2}\frac{z'_{i,j+1}-z'_{i,j}}{\Delta y_{i,j+\frac{1}{2}}} + \frac{\rho'_{i+1,j+1}+\rho'_{i+1,j}}{2}\frac{z'_{i+1,j+1}-z'_{i+1,j}}{\Delta y_{i+1,j+\frac{1}{2}}}}{2}\hat{\mathbf{y}}\right]\right\}.$$

$$(7.36)$$

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 (7.28) 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

Chapter 7 Equations of motion (baroclinic component)

is defined on z^* coordinate (the central depth of the bottom U-cell). These are the points with triangle (\triangle) symbols in Figure 3.5.

To calculate the pressure gradient, the actual depth of the vertical level where velocity is defined for the bottom U-cell $(s = s_{k=kbtm}^{U})$ 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} :

$$z_{kbtm}^{U} = z_{kbtm-1}^{T} - \Delta z_{kbtm}^{T} \frac{\Delta s_{kbtm}^{U}}{2\Delta s_{kbtm}^{T}}.$$
(7.37)

The depth anomaly used for computing geopotential gradient anomaly may be obtained in the same way:

$$z'_{kbtm}^{U} = z'_{kbtm-1}^{T} - \Delta z'_{kbtm}^{T} \frac{\Delta s_{kbtm}^{U}}{2\Delta s_{kbtm}^{T}}.$$
(7.38)

The finite difference expression for horizontal gradient is formally the same as (7.33) and (7.36). 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.

7.3 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 (7.1) and (7.2). 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.

7.3.1 Horizontal viscosity

The specific form for harmonic viscosity is shown here. Horizontal tension D_T and shear D_S are defined as follows:

$$D_T = h_{\psi} \frac{\partial}{h_{\mu} \partial \mu} \left(\frac{u}{h_{\psi}} \right) - h_{\mu} \frac{\partial}{h_{\psi} \partial \psi} \left(\frac{v}{h_{\mu}} \right), \tag{7.39}$$

$$D_{S} = h_{\psi} \frac{\partial}{h_{\mu} \partial \mu} \left(\frac{v}{h_{\psi}} \right) + h_{\mu} \frac{\partial}{h_{\psi} \partial \psi} \left(\frac{u}{h_{\mu}} \right).$$
(7.40)

The viscosity terms are

$$\mathcal{V}_{u} = \frac{1}{h_{\psi}^{2}} \frac{\partial}{h_{\mu} \partial \mu} \left(h_{\psi}^{2} \sigma_{T} \right) + \frac{1}{h_{\mu}^{2}} \frac{\partial}{h_{\psi} \partial \psi} \left(h_{\mu}^{2} \sigma_{S} \right), \tag{7.41}$$

$$\mathcal{V}_{\nu} = \frac{1}{h_{\psi}^{2}} \frac{\partial}{h_{\mu} \partial \mu} \left(h_{\psi}^{2} \sigma_{S} \right) - \frac{1}{h_{\mu}^{2}} \frac{\partial}{h_{\psi} \partial \psi} \left(h_{\mu}^{2} \sigma_{T} \right), \tag{7.42}$$

where $\sigma_T = v_H D_T$ and $\sigma_S = v_H D_S$, and v_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

$$D_T = \frac{1}{a\cos\phi}\frac{\partial u}{\partial\lambda} - \frac{1}{a}\frac{\partial v}{\partial\phi} - \frac{v}{a}\tan\phi,$$
(7.43)

$$D_S = \frac{1}{a\cos\phi}\frac{\partial v}{\partial\lambda} + \frac{1}{a}\frac{\partial u}{\partial\phi} + \frac{u}{a}\tan\phi.$$
(7.44)

The viscosity terms in this case are

$$\mathcal{V}_{u} = \frac{1}{a\cos\phi}\frac{\partial}{\partial\lambda}\sigma_{T} + \frac{1}{a}\frac{\partial}{\partial\phi}\sigma_{S} - \sigma_{S}\frac{2\tan\phi}{a},\tag{7.45}$$

$$\mathcal{V}_{\nu} = \frac{1}{a\cos\phi} \frac{\partial}{\partial\lambda} \sigma_{S} - \frac{1}{a} \frac{\partial}{\partial\phi} \sigma_{T} + \sigma_{T} \frac{2\tan\phi}{a}, \tag{7.46}$$

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 v_{BH} . The terms \mathcal{V}_u and \mathcal{V}_v given by (7.41) and (7.42) are sign-reversed and substituted as u and v in equations (7.39) and (7.40). 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.

7.3.2 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 (7.41) and (7.42) to

$$\begin{pmatrix} \sigma_T \\ \sigma_S \end{pmatrix} = \begin{bmatrix} \begin{pmatrix} \frac{1}{2}(\nu_0 + \nu_1) & 0 \\ 0 & \nu_1 \end{pmatrix} + (\nu_0 - \nu_1)n_\mu n_\psi \begin{pmatrix} -2n_\mu n_\psi & n_\mu^2 - n_\psi^2 \\ n_\mu^2 - n_\psi^2 & 2n_\mu n_\psi \end{bmatrix} \begin{bmatrix} D_T \\ D_S \end{pmatrix},$$
(7.47)

where $\hat{\mathbf{n}} = (n_{\mu}, n_{\psi})$ is a unit vector in an arbitrary direction and $v_0(v_1)$ 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 ($v_1 = 0$), 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 7.2). The ratio v_1/v_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 v_1/v_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.

variable	units	description	usage
cc0	1	the factor of anisotropy in viscosity; perpendicular / par-	default = 0.2
		allel (v_1/v_0) to the flow	
cc1	1	the factor (v_1/v_0) at lateral boundary	default = 0.5
figvisequator	degree	the factor is tapered toward the Equator; when LAT <	default = -1 (no
	latitude	flgvisequator [deg]	tapering)
vis_factor_equator	1	the factor at Equator when flgvisequator > 0	default = 0.0

Table 7.2 Namelist nml_visaniso anisotropic horizontal viscosity

7.3.3 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|:

$$|D| = \sqrt{D_T^2 + D_S^2},\tag{7.48}$$

the viscosity coefficients are set as follows:

$$v_H = \left(\frac{C\Delta_{\min}}{\pi}\right)^2 |D|,\tag{7.49}$$

$$\nu_{BH} = \frac{\Delta_{\min}^2}{8} \nu_H,\tag{7.50}$$

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:

$$v_H > U \frac{\Delta_{\min}}{2},\tag{7.51}$$

• Restriction on the width of the lateral boundary layer:

$$\nu_H > \beta \Delta_{\min}^3, \tag{7.52}$$

• CFL condition:

$$v_H < \frac{\Delta_{\min}^2}{2\Delta t},\tag{7.53}$$

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 (7.51) (Griffies and Hallberg, 2000).

Behavior of this scheme at run time is specified by namelist nml_smagor, whose components are listed on Table 7.3.

Table7.3 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	default = -1.0
l_smagor_nonzero_bg	logical	Use background viscosity as given by	default = .false.
		<pre>nml_baroclinic_visc_horz (Table 7.4)</pre>	

7.3.4 Discretization of the horizontal viscosity term

Using the notations

$$\begin{split} \delta_{\mu}A_{i,j} &\equiv \frac{A_{i+\frac{1}{2},j} - A_{i-\frac{1}{2},j}}{\Delta\mu}, \quad \delta_{\psi}A_{i,j} &\equiv \frac{A_{i,j+\frac{1}{2}} - A_{i,j-\frac{1}{2}}}{\Delta\psi}, \\ \delta_{i}A_{i,j} &\equiv A_{i+\frac{1}{2},j} - A_{i-\frac{1}{2},j}, \quad \delta_{j}A_{i,j} &\equiv A_{i,j+\frac{1}{2}} - A_{i,j-\frac{1}{2}}, \end{split}$$

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:

$$D_{T\,i,j} = \frac{h_{\psi i,j}}{h_{\mu i,j}} \overline{\delta_{\mu} \left(\frac{u}{h_{\psi}}\right)_{i,j}}^{\psi} - \frac{h_{\mu i,j}}{h_{\psi i,j}} \overline{\delta_{\psi} \left(\frac{v}{h_{\mu}}\right)_{i,j}}^{\mu},$$

$$D_{S\,i,j} = \frac{h_{\psi i,j}}{h_{\mu i,j}} \overline{\delta_{\mu} \left(\frac{v}{h_{\psi}}\right)_{i,j}}^{\psi} + \frac{h_{\mu i,j}}{h_{\psi i,j}} \overline{\delta_{\psi} \left(\frac{u}{h_{\mu}}\right)_{i,j}}^{\mu}.$$
(7.54)

Horizontal viscosity forces of (7.41) and (7.42) are discretized as follows:

$$F_{x_{i+\frac{1}{2},j+\frac{1}{2}}} = \frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}}} \\ \times \left[\frac{1}{h_{\psi i+\frac{1}{2},j+\frac{1}{2}}^{2}} \delta_{i} \left(\Delta y \Delta z h_{\psi}^{2} \overline{\nu_{H} D_{T}}^{\psi} \right)_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{1}{h_{\mu_{i+\frac{1}{2},j+\frac{1}{2}}}^{2}} \delta_{j} \left(\Delta x \Delta z h_{\mu}^{2} \overline{\nu_{H} D_{S}}^{\mu} \right)_{i+\frac{1}{2},j+\frac{1}{2}} \right],$$

$$F_{y_{i+\frac{1}{2},j+\frac{1}{2}}} = \frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}}} \\ \times \left[\frac{1}{h_{\psi i+\frac{1}{2},j+\frac{1}{2}}^{2}} \delta_{i} \left(\Delta y \Delta z h_{\psi}^{2} \overline{\nu_{H} D_{S}}^{\psi} \right)_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{1}{h_{\mu_{i+\frac{1}{2},j+\frac{1}{2}}}^{2}} \delta_{j} \left(\Delta x \Delta z h_{\mu}^{2} \overline{\nu_{H} D_{T}}^{\mu} \right)_{i+\frac{1}{2},j+\frac{1}{2}} \right].$$

$$(7.55)$$

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 7.7a), the velocity gradients at the wall are calculated as follows:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j+\frac{1}{2}} = \frac{u_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{ij}^{-}},$$

$$\left(\frac{\partial v}{\partial x}\right)_{i,j+\frac{1}{2}} = \frac{v_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{ij}^{-}},$$

$$(7.56)$$

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:

$$F_{x}^{W}{}_{i+\frac{1}{2},j+\frac{1}{2}} = -\frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}}h\psi_{i+\frac{1}{2},j+\frac{1}{2}}^{2}}\Delta y_{i,j+\frac{1}{2}}\Delta \tilde{z}_{i,j+\frac{1}{2}}h\psi_{i-\frac{1}{2},j}^{2}\nu_{Hi-\frac{1}{2},j}\frac{u_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}^{2}},$$

$$F_{y}^{W}{}_{i+\frac{1}{2},j+\frac{1}{2}} = -\frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}}h\psi_{i+\frac{1}{2},j+\frac{1}{2}}^{2}}\Delta y_{i,j+\frac{1}{2}}\Delta \tilde{z}_{i,j+\frac{1}{2}}h\psi_{i-\frac{1}{2},j}^{2}\nu_{Hi-\frac{1}{2},j}\frac{v_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}^{2}},$$
(7.57)

where $\Delta \tilde{z}_{i,j+\frac{1}{2}}$ is the wall height.

7.3.5 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:

$$-\left(\nu_{\nu}\frac{\partial u}{\partial z}\right)_{k-1} = -\nu_{\nu k-1}\frac{u_{k-\frac{3}{2}}-u_{k-\frac{1}{2}}}{\Delta z_{k-1}}$$

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 v_v is the vertical viscosity coefficient. Similarly, the momentum flux in the lower part of the U-cell is calculated as follows:

$$-\left(\nu_{\nu}\frac{\partial u}{\partial z}\right)_{k}=-\nu_{\nu k}\frac{u_{k-\frac{1}{2}}-u_{k+\frac{1}{2}}}{\Delta z_{k}},$$

where v_{vk} 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.



Figure 7.7 (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

$$\frac{\partial}{\partial z} \left(\nu_{\nu} \frac{\partial u}{\partial z} \right)_{k-\frac{1}{2}} = \frac{\left(\nu_{\nu} \frac{\partial u}{\partial z} \right)_{k-1} - \left(\nu_{\nu} \frac{\partial u}{\partial z} \right)_{k}}{\widehat{\Delta z}_{k-\frac{1}{2}}} = \frac{\nu_{\nu k-1} \left(u_{k-\frac{3}{2}} - u_{k-\frac{1}{2}} \right)}{\Delta z_{k-1} \widehat{\Delta z}_{k-\frac{1}{2}}} - \frac{\nu_{\nu k} \left(u_{k-\frac{1}{2}} - u_{k+\frac{1}{2}} \right)}{\Delta z_{k} \widehat{\Delta z}_{k-\frac{1}{2}}}$$
(7.58)

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 7.7(b)). Note that the first term on the r.h.s. of equation (7.58) 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 v_v , MRI.COM uses the larger of the value predicted by a turbulent closure scheme and a background one.

7.3.6 Bottom friction

When a U-cell in the $(k - \frac{1}{2})$ th layer contains solid earth (Figure 7.7(b) right), the stress from the lower boundary (τ_x^b, τ_y^b) is calculated following Weatherly et al. (1980). The specific expression is as follows:

$$\begin{pmatrix} \tau_x^b \\ \tau_y^b \end{pmatrix} = -\rho_0 C_{\text{btm}} \sqrt{u_{k-\frac{1}{2}}^2 + v_{k-\frac{1}{2}}^2} \begin{pmatrix} \cos\theta_0 & -\sin\theta_0 \\ \sin\theta_0 & \cos\theta_0 \end{pmatrix} \begin{pmatrix} u_{k-\frac{1}{2}} \\ v_{k-\frac{1}{2}} \end{pmatrix},$$

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,

$$C_{\text{btm}} = 1.225 \times 10^{-3}$$

 $\theta_0 = \pm \pi/18 \text{ rad} \quad (\equiv 10^\circ),$

where θ_0 is positive (negative) in the northern (southern) hemisphere. The variables are designated in the model as $C_{\text{btm}} = \text{abtm}, \cos \theta_0 = \text{bcs}, \text{ and } \sin(\pm \theta_0) = \text{isgn} * \text{bsn}$, where isgn = 1 in the northern hemisphere and isgn = -1 in the southern hemisphere.

7.4 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.

7.5 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 7.4 through 7.8. 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 [cm⁴sec⁻¹] instead of visc_horz_cm2ps [cm²sec⁻¹] in namelist nml_baroclinic_visc_horz. See Section 7.3.1 for detail.

VISANISO: Anisotropic viscosity coefficients are used Specify the coefficients by nml_visaniso (Table 7.2). See Section 7.3.2 for detail.

SMAGOR or SMAGHD: Smagorinsky parameterization is used for horizontal viscosity Specify nml_smagor (Table 7.3) for factors of the Smagorinsky parameterization. See Section 7.3.3 for detail.

VIS9P: Nine points are used in computing viscosity Five points are used by default.

See docs/README.Namelist for namelist details.

Table7.4	Namelist nml	baroclinic	visc hor	z (required)	for the	horizontal	viscosity	(see Section 7	(.5)
	-								

variable	units	description		usage
visc_horz_cm2ps	$cm^2 sec^{-1}$	horizontal viscosity (v_H in Section 7.3.1)	required	
		·	•	Continued on next page

Chapter 7 Equations of motion (baroclinic component)

Tuete /// Continue from previous puge					
variable	units	description	usage		
file_visc_horz_2d	$cm^2 sec^{-1}$	2D distribution of viscosity	optional (This overwrites		
			<pre>visc_horz_cm2ps.)</pre>		
slip_factor	factor	0.d0 means no-slip condition, while	default = $0.d0$ (only valid with		
		1.d0 slip (Section 7.3.1)	VIS9P option)		

Table 7.4 – continued from previous page

Table7.5 Namelist nml_visc_vert_bg (optional) for vertical viscosity (see Section 7.5)

variable	units	description	usage
visc_vert_bg_cm2ps	$\rm cm^2sec^{-1}$	background vertical viscosity (v_v in Section 7.3.5)	default = 1.d0

Table7.6 Namelist nml_bottom_friction (optional) for the bottom friction (see Section 7.5)

variable	units	description	usage
btmfrc_scale	1	$C_{\rm btm}$ in Section 7.3.6	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
			<pre>btmfrc_scale)</pre>

Table7.7 Namelist nml_hvisc_add (optional) for additional horizonta viscosity (see Section 7.5)

variable	units	description	usage
l_hvisc_add_harmonic	logical	use additional harmonic viscosity or	default = .false.
		not	
file_hvisc_harmonic	$cm^2 sec^{-1}$	2D distribution of viscosity	file name

 Table7.8
 Namelist nml_baroclinic_run (optional) for starting the baroclinic mode (see Section 7.5)

variable	units	description	usage
l_rst_baroclinic_in	logical	read initial restart for 3D velocity or	default = l_rst_in
		not	

Part IV

Evolution of tracers

Chapter 8

Tracer advection schemes

This chapter describes tracer advection schemes available in MRI.COM. The tracer advection schemes available in MRI.COM are: the weighted upcurrent scheme (Section 8.2), the Quadratic Upstream Interpolation for Convective Kinematics (QUICK; Leonard, 1979) as described in Section 8.3 (QUICKADVEC option), a combination of the QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979) for vertical advection (Section 8.4) and the Uniformly Third-Order Polynomial Interpolation Algorithm (UTOPIA; Leonard et al., 1993), which is a two-dimensional generalization of the QUICKEST scheme, for horizontal advection (Section 8.5) with UTZQADVEC option. For QUICKEST and UTOPIA, a flux limiter that prevents unrealistic extrema is applied (Leonard et al., 1994).

The above schemes seek to improve the accuracy by refining the finite-difference expression at the cell faces. There are other approaches. The Second Order Moment (SOM; Prather, 1986) scheme seeks to improve the accuracy by considering the distribution within the cell and is available in MRI.COM through SOMADVEC option (Section 8.6). The Multidimensional Positive Definite Advection Transport Algorithm (MPDATA; Smolarkiewicz and Margolin, 1998) scheme seeks to cancel the numerical diffusion of the lower order scheme by using anti-diffusive velocities and is available in MRI.COM through MPDATAADVEC option (Section 8.7).

Different advection schemes can be used for individual tracers. They should be chosen from among compiled schemes at run time (Chapter 13).

8.1 Finite volume or flux form method

The equation governing the time change of tracer T is

$$\frac{\partial(z_sT)}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial(z_s h_\psi uT)}{\partial \mu} + \frac{\partial(z_s h_\mu vT)}{\partial \psi} \right\} + \frac{\partial(z_s \dot{s}T)}{\partial s} = \frac{\partial(z_sT)}{\partial t} + \mathcal{A}(T) = -z_s \nabla \cdot \mathbf{F}_T + z_s Q^T, \quad (8.1)$$

where $\mathcal{A}(T)$ represents the advection operator for *T*. The finite difference form of the l.h.s. of (8.1) is given by first considering a control cell volume and then calculating fluxes through each cell face, and setting their divergence and convergence to be the time change rate at the grid cell (Figure 8.1). In finite difference form, this is expressed as follows:

$$T_{i,j,k-\frac{1}{2}}^{n+1} \Delta V_{i,j,k-\frac{1}{2}}^{n+1} = T_{i,j,k-\frac{1}{2}}^{n-1} \Delta V_{i,j,k-\frac{1}{2}}^{n-1} + 2 \Delta t \{FXA_{i-\frac{1}{2},j,k-\frac{1}{2}} - FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} + FYA_{i,j-\frac{1}{2},k-\frac{1}{2}} - FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} + FZA_{i,j,k} - FZA_{i,j,k-1}\}, \quad (8.2)$$

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),

$$\Delta V_{i,j,k-\frac{1}{2}} \equiv (\text{volt})_{i,j,k-\frac{1}{2}} = (\text{volu_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\text{volu_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + (\text{volu_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + (\text{volu_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}.$$
(8.3)

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 $(volt_tr)_{i-\frac{1}{2},j-\frac{1}{2}}$. Similarly, $(volt_tl)_{i+\frac{1}{2},j-\frac{1}{2}}$ is the right-lower (southeastern), $(volt_br)_{i-\frac{1}{2},j+\frac{1}{2}}$ is the left-upper (northwestern), and $(volt_bl)_{i+\frac{1}{2},j+\frac{1}{2}}$ is the right-upper (northeastern) quarter of a T-cell at (i, j).

Fluxes due to advection are given as follows:

$$FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} = U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}T_{i+\frac{1}{2},j,k-\frac{1}{2}},$$
(8.4)

$$FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} = V_{i,j+\frac{1}{2},k-\frac{1}{2}}^T T_{i,j+\frac{1}{2},k-\frac{1}{2}},$$
(8.5)

$$FZA_{i,j,k} = W_{i,j,k}^T T_{i,j,k},$$
(8.6)

where horizontal volume transport U^T and V^T are defined as follows, using (5.13) to (5.18):

$$U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T} = \frac{\Delta y_{i+\frac{1}{2},j}}{2} \left(u_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \Delta z_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \right),$$
(8.7)

$$V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{T} = \frac{\Delta x_{i,j+\frac{1}{2}}}{2} \left(v_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \Delta z_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + v_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} \right).$$
(8.8)

Vertical volume transport W^T is then obtained by diagnostically solving (5.12). Moreover, the vertical velocity w, which is necessary for using QUICKEST, is calculated as follows (w is not needed except for QUICKEST):

$$W_{i,j,k}^T = w_{i,j,k} \times (\operatorname{areat})_{i,j,k-\frac{1}{2}},$$
 (8.9)

where

$$(\operatorname{areat})_{i,j,k-\frac{1}{2}} = (\operatorname{a_tr})_{i-\frac{1}{2},j-\frac{1}{2}} \times (\operatorname{aexl})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + (\operatorname{a_tl})_{i+\frac{1}{2},j-\frac{1}{2}} \times (\operatorname{aexl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + (\operatorname{a_br})_{i-\frac{1}{2},j+\frac{1}{2}} \times (\operatorname{aexl})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\operatorname{a_bl})_{i+\frac{1}{2},j+\frac{1}{2}} \times (\operatorname{aexl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}.$$
(8.10)

An array aexl is set to be unity if the corresponding U-cell is a sea cell and set to be zero otherwise.

This formulation does not depend on the choice of the advection scheme. The difference arises from the way of determining cell boundary values of tracer, $T_{i+\frac{1}{2},j,k-\frac{1}{2}}$, $T_{i,j+\frac{1}{2},k-\frac{1}{2}}$, and $T_{i,j,k}$. There are several choices for tracer advection schemes, which are explained in the following sections.

- Weighted upcurrent scheme (adv_scheme%name = "upc"). This scheme is always available for use. Specify the weighting ratio listed on Table 8.1.
- Quadratic Upstream Interpolation for Convective Kinematics (QUICK; Leonard, 1979) for both horizontal and vertical direction (QUICKADVEC option; adv_scheme%name = "quick").
- The combination of the Uniformly Third-Order Polynomial Interpolation Algorithm (UTOPIA; Leonard et al., 1993) for horizontal direction and QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979) for the vertical direction (UTZQADVEC option; adv_scheme%name = "utzqadv").
- Second Order Moment (SOM; Prather, 1986) schemes for both horizontal and vertical direction (SOMADVEC option; adv_scheme%name = "som").
- Multidimensional Positive Definite Advection Transport Algorithm (MPDATA; Smolarkiewicz and Margolin, 1998) for all directions (MPDATAADVEC option; adv_scheme%name = "mpdata").

8.2 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:

$$T_{i+\frac{1}{2},j,k-\frac{1}{2}}^{upcurrent} = \begin{cases} T_{i,j,k-\frac{1}{2}}, & \text{if } U_{i+\frac{1}{2},j,k-\frac{1}{2}} > 0, \\ T_{i+1,j,k-\frac{1}{2}}, & \text{if } U_{i+\frac{1}{2},j,k-\frac{1}{2}} < 0. \end{cases}$$
(8.11)

The centered finite difference scheme uses the average between the two neighboring points of tracer as the cell boundary value:

$$T_{i+\frac{1}{2},j,k-\frac{1}{2}}^{center} = \frac{I_{i+1,j,k-\frac{1}{2}} + I_{i,j,k-\frac{1}{2}}}{2}.$$
(8.12)



Figure 8.1 Grid arrangement around TS-Box (Upper: Views from the upper, Below: views from the horizontal). Fluxes represented by an arrow are calculated.

Table8.1 Namelist nml_tracer_adv for weighted upcurrent scheme

variable name	units	description	usage
weight_upcurrent_horz	1	upstream-side weighting ratio for the	1.0: up-current, 0.5: centered differ-
		horizontal advection	ence
<pre>weight_upcurrent_vert</pre>	1	upstream-side weighting ratio for the	1.0: up-current, 0.5: centered differ-
		vertical advection	ence

Taking the ratio 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:

$$FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} = \alpha \left\{ \frac{U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T} + |U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}|}{2} T_{i} + \frac{U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T} - |U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}|}{2} T_{i+1} \right\} + (1-\alpha)U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T} \frac{T_{i+1,j,k-\frac{1}{2}} + T_{i,j,k-\frac{1}{2}}}{2}$$

$$(8.13)$$

$$=U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}\left\{\frac{1}{2}\left(1+\alpha\frac{U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}}{|U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}|}\right)T_{i}+\frac{1}{2}\left(1-\alpha\frac{U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}}{|U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}|}\right)T_{i+1}\right\}.$$
(8.14)

Mainly for the sake of computational efficiency, we give a parameter $\beta = \frac{1}{2}(\alpha + 1)$ instead of α at run time. Different parameters for horizontal and vertical directions may be given, which are listed on Table 8.1.

8.3 QUICK scheme

This section explains the QUICK scheme. 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 8.2).



Figure 8.2 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} .

Originally, cell boundary values in the QUICK scheme are given as follows:

$$T_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta x_i T_{i+1,j,k-\frac{1}{2}} + \Delta x_{i+1} T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+1} + \Delta x_i} - \frac{\Delta x_{i+1} \Delta x_i}{4} c_{i+\frac{1}{2},j,k-\frac{1}{2}},$$
(8.15)

$$T_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{\Delta y_j T_{i,j+1,k-\frac{1}{2}} + \Delta y_{j+1} T_{i,j,k-\frac{1}{2}}}{\Delta y_{i+1} + \Delta y_i} - \frac{\Delta y_{j+1} \Delta y_j}{4} d_{i,j+\frac{1}{2},k-\frac{1}{2}},$$
(8.16)

$$T_{i,j,k} = \frac{\Delta z_{k-\frac{1}{2}} T_{i,j+1,k+\frac{1}{2}} + \Delta z_{k+\frac{1}{2}} T_{i,j,k-\frac{1}{2}}}{\Delta z_{k+\frac{1}{2}} + \Delta z_{k-\frac{1}{2}}} - \frac{\Delta z_{k-\frac{1}{2}} \Delta z_{k+\frac{1}{2}}}{4} e_{i,j,k},$$
(8.17)

where c, d, and e are defined depending on the direction of the mass flux as follows:

$$\begin{aligned} c_{i+\frac{1}{2},j,k-\frac{1}{2}} &= \frac{\Delta x_i \delta_x \delta_x T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta x_i}^x} (\equiv c_p), & \text{if } U_{i+\frac{1}{2},j,k-\frac{1}{2}}^T > 0, \\ &= \frac{\Delta x_{i+1} \delta_x \delta_x T_{i+1,j,k-\frac{1}{2}}}{2\overline{\Delta x_i}^{T}} (\equiv c_m), & \text{if } U_{i+\frac{1}{2},j,k-\frac{1}{2}}^T < 0, \\ d_{i,j+\frac{1}{2},k-\frac{1}{2}} &= \frac{\Delta y_j \delta_y \delta_y T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta y_j}^y} (\equiv d_p), & \text{if } V_{i,j+\frac{1}{2},k-\frac{1}{2}}^T > 0, \\ &= \frac{\Delta y_{j+1} \delta_y \delta_y T_{i,j+1,k-\frac{1}{2}}}{2\overline{\Delta y_{j+1}}^y} (\equiv d_m), & \text{if } V_{i,j+\frac{1}{2},k-\frac{1}{2}}^T < 0, \\ e_{i,j,k} &= \frac{\Delta z_{k+\frac{1}{2}} \delta_z \delta_z T_{i,j,k+\frac{1}{2}}}{2\overline{\Delta z_{i,j,k+\frac{1}{2}}}} (\equiv e_p), & \text{if } W_{i,j,k}^T > 0, \\ &= \frac{\Delta z_{k-\frac{1}{2}} \delta_z \delta_z T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta z_{i,j,k+\frac{1}{2}}}} (\equiv e_m), & \text{if } W_{i,j,k}^T < 0. \end{aligned}$$

Chapter 8 Tracer advection schemes

The finite difference operators are defined as follows (definitions in y and z directions are the same):

$$\delta_{x}A_{i} \equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta x_{i}}, \quad \delta_{x}A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_{i}}{\Delta x_{i+\frac{1}{2}}},$$
$$\overline{A_{i}}^{x} \equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2}, \quad \overline{A_{i+\frac{1}{2}}}^{x} \equiv \frac{A_{i+1} + A_{i}}{2}.$$
(8.19)

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

$$c_a = c_m + c_p \tag{8.20}$$

$$c_d = c_m - c_p \tag{8.21}$$

$$d_a = d_m + d_p \tag{8.22}$$

$$d_a = d_m - d_m \tag{8.23}$$

$$a_d = a_m - a_p \tag{8.23}$$

$$e_a = e_m + e_p \tag{8.24}$$

$$e_d = e_m - e_p, \tag{8.25}$$

we obtain

$$FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} = U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T} \left[\frac{\Delta x_{i,j}T_{i+1,j,k-\frac{1}{2}} + \Delta x_{i+1,j}T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+1,j} + \Delta x_{i,j}} - \frac{\Delta x_{i+1,j}\Delta x_{i,j}}{8}c_{a_{i+\frac{1}{2},j,k-\frac{1}{2}}} \right] + |U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}| \frac{\Delta x_{i+1,j}\Delta x_{i,j}}{8}c_{d_{i+\frac{1}{2},j,k-\frac{1}{2}}},$$

$$(8.26)$$

$$FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} = V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{T} \left[\frac{\Delta y_{i,j}T_{i,j+1,k-\frac{1}{2}} + \Delta y_{i,j+1}T_{i,j,k-\frac{1}{2}}}{\Delta y_{i,j+1} + \Delta y_{i,j}} - \frac{\Delta y_{i,j+1}\Delta y_{i,j}}{8} d_{a_{i,j+\frac{1}{2},k-\frac{1}{2}}} \right]$$

$$+ |V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{T}|\frac{\Delta y_{i,j+1}\Delta y_{i,j}}{8}d_{d_{i,j+\frac{1}{2},k-\frac{1}{2}}},$$

$$FZA_{i,j,k} = W_{i,j,k}^{T}\left[\frac{\Delta z_{i,j,k-\frac{1}{2}}T_{i,j,k+\frac{1}{2}} + \Delta z_{i,j,k+\frac{1}{2}}T_{i,j,k-\frac{1}{2}}}{\Delta z_{i,j,k+\frac{1}{2}} + \Delta z_{i,j,k-\frac{1}{2}}} - \frac{\Delta z_{i,j,k+\frac{1}{2}}\Delta z_{i,j,k-\frac{1}{2}}}{8}e_{a_{i,j,k}}\right]$$

$$(8.27)$$

$$+|W_{i,j,k-\frac{1}{2}}^{T}|\frac{\Delta z_{i,j,k+\frac{1}{2}}\Delta z_{i,j,k-\frac{1}{2}}}{8}e_{d_{i,j,k}}.$$
(8.28)

Equation (8.26) can be rewritten as

$$FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} \simeq U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}\tilde{T}_{i+\frac{1}{2},j,k-\frac{1}{2}} + A_{Q}\frac{\partial^{3}T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\partial x^{3}},$$
(8.29)

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

$$A_Q = |U_{i+\frac{1}{2},j,k-\frac{1}{2}}^T| \frac{\Delta x_{i+1} \Delta x_{i+\frac{1}{2}} \Delta x_i}{8}.$$
(8.30)

Although the time integration for advection is done by the leap-frog scheme, the second term on the r.h.s. of (8.29) 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 8.1).

8.4 QUICKEST for vertical advection

This section describes the specific expression and the accuracy of the QUICK with Estimated Streaming Terms (QUICK-EST; Leonard, 1979) for vertical advection.

Consider a one-dimensional equation of advection for incompressible fluid

$$\frac{\partial T}{\partial t} + \frac{\partial}{\partial z}(wT) = 0, \tag{8.31}$$

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:

$$\Delta z_k = \frac{\Delta z_{k+\frac{1}{2}} + \Delta z_{k-\frac{1}{2}}}{2}.$$
(8.32)

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

$$T_{k-\frac{3}{2}} = c_0 + c_1 \left(\frac{\Delta z_{k-\frac{3}{2}}}{2} + \Delta z_{k-\frac{1}{2}}\right) + c_2 \left(\frac{\Delta z_{k-\frac{3}{2}}}{2} + \Delta z_{k-\frac{1}{2}}\right)^2 + O(\Delta z^3),$$
(8.33)

$$T_{k-\frac{1}{2}} = c_0 + c_1 \frac{\Delta z_{k-\frac{1}{2}}}{2} + c_2 \frac{\Delta z_{k-\frac{1}{2}}^2}{4} + O(\Delta z^3),$$
(8.34)

$$T_{k+\frac{1}{2}} = c_0 - c_1 \frac{\Delta z_{k+\frac{1}{2}}}{2} + c_2 \frac{\Delta z_{k+\frac{1}{2}}^2}{4} + O(\Delta z^3),$$
(8.35)

$$T_{k+\frac{3}{2}} = c_0 - c_1 \left(\frac{\Delta z_{k+\frac{3}{2}}}{2} + \Delta z_{k+\frac{1}{2}}\right) + c_2 \left(\frac{\Delta z_{k+\frac{3}{2}}}{2} + \Delta z_{k+\frac{1}{2}}\right)^2 + O(\Delta z^3).$$
(8.36)

Coefficients c_0 , c_1 , and c_2 can be solved using three of the four equations (8.33), (8.34), (8.35), and (8.36). The three upstream-side equations are chosen. When w > 0 (w < 0), equations (8.34), (8.35), and (8.36) ((8.33), (8.34), and (8.35)) are used. The solution is as follows:

$$c_{0} = \frac{T_{k-\frac{1}{2}}\Delta z_{k+\frac{1}{2}} + T_{k+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}}{2\Delta z_{k}} - \frac{\Delta z_{k+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}}{4}c_{2},$$
(8.37)

$$c_{1} = \frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_{k}} - \frac{\Delta z_{k-\frac{1}{2}} - \Delta z_{k+\frac{1}{2}}}{2}c_{2},$$
(8.38)

$$c_{2} = \begin{cases} \frac{1}{\Delta z_{k} + \Delta z_{k+1}} \left(\frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_{k}} - \frac{T_{k+\frac{1}{2}} - T_{k+\frac{3}{2}}}{\Delta z_{k+1}} \right) & (w > 0), \\ \frac{1}{\Delta z_{k-1} + \Delta z_{k}} \left(\frac{T_{k-\frac{3}{2}} - T_{k-\frac{1}{2}}}{\Delta z_{k-1}} - \frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_{k}} \right) & (w < 0). \end{cases}$$
(8.39)

Next, equation (8.31) is integrated over one time step and one grid cell.

$$\int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz \frac{\partial T}{\partial t} = -\int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz \frac{\partial}{\partial z} (wT).$$
(8.40)

The r.h.s. of (8.40) can be written as

$$-\int_{t^n}^{t^{n+1}} dt (w_u T_u - w_l T_l), \tag{8.41}$$

where subscript u(l) denotes $z = z_{k-1}$ ($z = z_k$). Assuming that w does not depend on time,

$$\int_{t^n}^{t^{n+1}} dt T_l = \int_{-w_l \Delta t}^0 [c_0^n + c_1^n \xi + c_2^n \xi^2 + O(\Delta z^3)] \frac{d\xi}{w_l}.$$
(8.42)

Thus expression (8.41) becomes

$$-\Delta t (w_u \overline{T_u^n} - w_l \overline{T_l^n}) + O(\Delta z^3 w \Delta t), \tag{8.43}$$

Chapter 8 Tracer advection schemes

where

$$\overline{T_l^n} = \frac{1}{w_l \Delta t} \int_{-w_l \Delta t}^0 (c_0^n + c_1^n \xi + c_2^n \xi^2) d\xi$$

= $c_0^n - \frac{c_1^n}{2} w_l \Delta t + \frac{c_2^n}{3} w_l^2 \Delta t^2.$ (8.44)

Using up to the second order terms of a Taylor expansion, the l.h.s. of (8.40) can be written as follows:

$$\int_{t^{n}}^{t^{n+1}} dt \int_{z_{k}}^{z_{k-1}} dz \frac{\partial T}{\partial t} = \Delta z_{k-\frac{1}{2}} \left[T_{k-\frac{1}{2}}^{n+1} - T_{k-\frac{1}{2}}^{n} + \frac{\Delta z_{k-\frac{1}{2}}^{2}}{24} (T_{zz_{k-\frac{1}{2}}}^{n+1} - T_{zz_{k-\frac{1}{2}}}^{n}) + O(\Delta z^{3}) \right],$$
(8.45)

where

$$T_{zz_{k-\frac{1}{2}}}^{n+1} - T_{zz_{k-\frac{1}{2}}}^{n} = \Delta t \left. \frac{\partial T_{zz}}{\partial t} \right|_{k-\frac{1}{2}}^{n} + O(\Delta t^{2})$$

$$= -\Delta t \frac{\partial^{2}}{\partial z^{2}} \left[\frac{\partial}{\partial z} (wT) \right]_{k-\frac{1}{2}}^{n} + O(\Delta t^{2})$$

$$= -\Delta t \frac{\partial}{\partial z} (wT_{zz})_{k-\frac{1}{2}}^{n} + O(\Delta t^{2})$$

$$= -\Delta t \left\{ \frac{w_{u}T_{zzu} - w_{l}T_{zzl}}{\Delta z_{k-\frac{1}{2}}} \right\} + O(w\Delta t\Delta z) + O(\Delta t^{2}).$$
(8.46)

The expression for the r.h.s. of (8.45) becomes

$$\Delta z_{k-\frac{1}{2}} \left[T_{k-\frac{1}{2}}^{n+1} - T_{k-\frac{1}{2}}^{n} - \frac{\Delta z_{k-\frac{1}{2}}^{2}}{24} \Delta t \frac{w_{u} T_{zz_{u}}^{n} - w_{l} T_{zz_{l}}^{n}}{\Delta z_{k-\frac{1}{2}}} + O(\Delta z^{3}) \right] + O(w \Delta t \Delta z^{4}) + O(\Delta z^{3} \Delta t^{2}).$$
(8.47)

Based on (8.43) and (8.47), the discretized forecasting equation is expressed as follows:

$$T_{k-\frac{1}{2}}^{n+1} = T_{k-\frac{1}{2}}^{n} - \frac{\Delta t}{\Delta z_{k-\frac{1}{2}}} \left[w_{u} \overline{T_{u}^{n}} - w_{l} \overline{T_{l}^{n}} - \frac{\Delta z_{k-\frac{1}{2}}^{2}}{24} (w_{u} T_{zzu}^{n} - w_{l} T_{zzl}^{n}) \right] + O(\alpha \Delta z^{4}) + O(\Delta z^{2} \Delta t^{2}),$$
(8.48)

where

$$\alpha \equiv \frac{w\Delta t}{\Delta z} < 1, \tag{8.49}$$

$$T_{zz_{l}}^{n} = 2c_{2} + O(\Delta z).$$
(8.50)

The accuracy of equation (8.48) is $\max(O(\Delta z^4), O(\Delta z^2 \Delta t^2))$.

8.5 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:

$$\frac{\partial T}{\partial t} + \frac{1}{h_{\mu}h_{\psi}}\frac{\partial}{\partial\mu}(h_{\psi}uT) + \frac{1}{h_{\mu}h_{\psi}}\frac{\partial}{\partial\psi}(h_{\mu}vT) = 0.$$
(8.51)

Integrated over a tracer cell and for one time step,

$$\int_{\psi_L - \Delta \psi_L/2}^{\psi_L + \Delta \psi_L/2} d\psi \int_{\mu_L - \Delta \mu_L/2}^{\mu_L + \Delta \mu_L/2} d\mu (\chi^{n+1} - \chi^n) = -\Delta t (u_r^n \overline{T_r^n} \Delta y_r - u_l^n \overline{T_l^n} \Delta y_l + v_u^n \overline{T_u^n} \Delta x_u - v_d^n \overline{T_d^n} \Delta x_d),$$
(8.52)

where $\chi \equiv h_{\mu}h_{\psi}T$ and $\overline{T_r^n}$ etc. on the r.h.s. are the face values described later. On the l.h.s. of (8.52), 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 8.3 for the label of the point):

$$\chi = \chi_L + a_{10}(\mu - \mu_L) + a_{20}(\mu - \mu_L)^2 + a_{01}(\psi - \psi_L) + a_{02}(\psi - \psi_L)^2 + a_{11}(\mu - \mu_L)(\psi - \psi_L).$$
(8.53)

Then values at points E, W, N, and S are



Figure 8.3 Labels of tracer grid points (upper case characters) and faces (lower case characters).

$$\chi_E = \chi_L + a_{10} \Delta \mu_r + a_{20} \Delta \mu_r^2, \tag{8.54}$$

$$\chi_W = \chi_L - a_{10} \Delta \mu_l + a_{20} \Delta \mu_l^2, \tag{8.55}$$

$$\chi_N = \chi_L + a_{01} \Delta \psi_u + a_{02} \Delta \psi_u^2,$$
(8.56)

$$\chi_S = \chi_L - a_{01} \Delta \psi_d + a_{02} \Delta \psi_d^2, \tag{8.57}$$

where

$$\Delta\psi_u \equiv \frac{\Delta\psi_L + \Delta\psi_N}{2},\tag{8.58}$$

$$\Delta \psi_d \equiv \frac{\Delta \psi_L + \Delta \psi_S}{2},\tag{8.59}$$

$$\Delta\mu_r \equiv \frac{\Delta\mu_L + \Delta\mu_E}{2},\tag{8.60}$$

$$\Delta \mu_l \equiv \frac{\Delta \mu_L + \Delta \mu_W}{2}.$$
(8.61)

Using these known values, the following parameters are obtained,

$$a_{10} = \frac{\Delta \mu_l \frac{\chi_E - \chi_L}{\Delta \mu_r} + \Delta \mu_r \frac{\chi_L - \chi_W}{\Delta \mu_l}}{\Delta \mu_r},$$
(8.62)

$$a_{20} = \frac{\frac{\chi_E - \chi_L}{\Delta \mu_r} - \frac{\chi_L - \chi_W}{\Delta \mu_l}}{\Lambda \mu_r + \Lambda \mu_l},$$
(8.63)

$$a_{01} = \frac{\Delta\psi_d \frac{\chi_N - \chi_L}{\Delta\psi_u} + \Delta\psi_u \frac{\chi_L - \chi_S}{\Delta\psi_d}}{(8.64)}$$

$$a_{01} = \frac{\Delta \psi_u + \Delta \psi_d}{\Delta \psi_u + \Delta \psi_d}, \qquad (8.64)$$

$$a_{02} = \frac{\Delta \psi_u \quad \Delta \psi_d}{\Delta \psi_u + \Delta \psi_d}.$$
(8.65)
Chapter 8 Tracer advection schemes

Substituting (8.53) into the l.h.s. of (8.52) yields

$$\Delta \mu_L \Delta \psi_L \left[\chi_L^{n+1} - \chi_L^n + \frac{\Delta \mu_L^2}{12} (a_{20}^{n+1} - a_{20}^n) + \frac{\Delta \psi_L^2}{12} (a_{02}^{n+1} - a_{02}^n) \right].$$
(8.66)

Using equation (8.51), the following approximation is allowed:

$$a_{20}^{n+1} - a_{20}^{n} = -\Delta t \left[\frac{h_{\psi r} u_{r}^{n} T_{\mu\mu r}^{n} - h_{\psi l} u_{l}^{n} T_{\mu\mu l}^{n}}{\Delta \mu_{L}} + \frac{h_{\mu u} v_{u}^{n} T_{\mu\mu u}^{n} - h_{\mu d} v_{d}^{n} T_{\mu\mu d}^{n}}{\Delta \psi_{L}} \right],$$
(8.67)

$$a_{02}^{n+1} - a_{02}^{n} = -\Delta t \left[\frac{h_{\psi r} u_{r}^{n} T_{\psi \psi r}^{n} - h_{\psi l} u_{l}^{n} T_{\psi \psi l}^{n}}{\Delta \mu_{L}} + \frac{h_{\mu u} v_{u}^{n} T_{\psi \psi u}^{n} - h_{\mu d} v_{d}^{n} T_{\psi \psi d}^{n}}{\Delta \psi_{L}} \right],$$
(8.68)

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,

$$T_L^{n+1} = T_L^n - \frac{\Delta t}{\Delta S_L} (u_r^n \tilde{T}_r^n \Delta y_r - u_l^n \tilde{T}_l^n \Delta y_l + v_u^n \tilde{T}_u^n \Delta x_u - v_d^n \tilde{T}_d^n \Delta x_d),$$
(8.69)

where

$$\tilde{T_l^n} = \overline{T_l^n} - \frac{\Delta \mu_L^2}{24} T_{\mu\mu l}^n - \frac{\Delta \psi_L^2}{24} T_{\psi\psi l}^n, \qquad (8.70)$$

$$\tilde{T_{d}^{n}} = \overline{T_{d}^{n}} - \frac{\Delta \mu_{L}^{2}}{24} T_{\mu\mu d}^{n} - \frac{\Delta \psi_{L}^{2}}{24} T_{\psi\psi d}^{n}.$$
(8.71)

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 8.4, and the values of T^n are given as the second order interpolation about l of Figure 8.3. Similar operations will be used to obtain the expression for $\overline{T_d^n}$.



Figure 8.4 Area used to average tracer values for the face l

First, Taylor expansions of T^n about **l** and **d** are written as follows:

$$T^{n}|_{\mathbf{l}} = c_{00} + c_{10}(\mu - \mu_{l}) + c_{20}(\mu - \mu_{l})^{2} + c_{01}(\psi - \psi_{L}) + c_{02}(\psi - \psi_{L})^{2} + c_{11}(\mu - \mu_{l})(\psi - \psi_{L}),$$
(8.72)

$$T^{n}|_{\mathbf{d}} = d_{00} + d_{10}(\mu - \mu_{l}) + d_{20}(\mu - \mu_{l})^{2} + d_{01}(\psi - \psi_{L}) + d_{02}(\psi - \psi_{L})^{2} + d_{11}(\mu - \mu_{l})(\psi - \psi_{L}).$$
(8.73)

The T^n values at eight points around **l** are,

$$T_L^n = c_{00} + c_{10} \frac{\Delta \mu_L}{2} + c_{20} \frac{\Delta \mu_L^2}{4}, \tag{8.74}$$

$$T_W^n = c_{00} - c_{10} \frac{\Delta \mu_W}{2} + c_{20} \frac{\Delta \mu_W^2}{4},$$
(8.75)

$$T_E^n = c_{00} + c_{10} \left(\Delta \mu_L + \frac{\Delta \mu_E}{2} \right) + c_{20} \left(\Delta \mu_L + \frac{\Delta \mu_E}{2} \right)^2,$$
(8.76)

$$T_{WW}^{n} = c_{00} - c_{10} \left(\Delta \mu_{W} + \frac{\Delta \mu_{WW}}{2} \right) + c_{20} \left(\Delta \mu_{W} + \frac{\Delta \mu_{WW}}{2} \right)^{2},$$
(8.77)

$$T_N^n = T_L^n + c_{01} \Delta \psi_u + c_{02} \Delta \psi_u^2 + c_{11} \frac{\Delta \mu_L}{2} \Delta \psi_u,$$
(8.78)

$$T_{S}^{n} = T_{L}^{n} - c_{01}\Delta\psi_{d} + c_{02}\Delta\psi_{d}^{2} - c_{11}\frac{\Delta\mu_{L}}{2}\Delta\psi_{d},$$
(8.79)

$$T_{NW}^{n} = T_{W}^{n} + c_{01}\Delta\psi_{u} + c_{02}\Delta\psi_{u}^{2} - c_{11}\frac{\Delta\mu_{W}}{2}\Delta\psi_{u},$$
(8.80)

$$T_{SW}^{n} = T_{W}^{n} - c_{01}\Delta\psi_{d} + c_{02}\Delta\psi_{d}^{2} + c_{11}\frac{\Delta\mu_{W}}{2}\Delta\psi_{d}.$$
(8.81)

To obtain all six coefficients, six of these equations (points) are used. The equations are chosen according to the following flow direction.

$$u_l^n > 0, \quad v_l^n > 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{WW}, \mathbf{S}, \mathbf{NW}, \mathbf{SW}$$

$$(8.82)$$

$$u_l^n < 0, \quad v_l^n > 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{E}, \mathbf{N}, \mathbf{S}, \mathbf{SW}$$
(8.83)

$$u_l^n > 0, \quad v_l^n < 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{WW}, \mathbf{N}, \mathbf{NW}, \mathbf{SW}$$

$$(8.84)$$

$$u_l^n < 0, \quad v_l^n < 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{E}, \mathbf{N}, \mathbf{S}, \mathbf{NW}$$

$$(8.85)$$

From equations (8.74) and (8.75),

$$c_{00} = \frac{\Delta\mu_W T_L^n + \Delta\mu_L T_W^n}{2\Delta\mu_l} - c_{20} \frac{\Delta\mu_L \Delta\mu_W}{4},$$
(8.86)

$$c_{10} = \frac{T_L^n - T_W^n}{\Delta \mu_l} - c_{20} \frac{\Delta \mu_L - \Delta \mu_W}{2}.$$
(8.87)

When $u_l^n > 0$, from (8.74) and (8.77),

$$c_{20} = \frac{\frac{T_L^n - T_W^n}{\Delta \mu_l} - \frac{T_W^n - T_W^n}{\Delta \mu_{ll}}}{\Delta \mu_l + \Delta \mu_{ll}},$$
(8.88)
where $\Delta \mu_{ll} \equiv \frac{\Delta \mu_W + \Delta \mu_{WW}}{2}.$

Using equations (8.80) and (8.81),

$$c_{02} = \frac{\frac{T_{NW}^n - T_W^n}{\Delta \psi_u} - \frac{T_W^n - T_{SW}^n}{\Delta \psi_d}}{\Delta \psi_u + \Delta \psi_d}.$$
(8.89)

When $u_l^n < 0$, from (8.75) and (8.76),

$$c_{20} = \frac{\frac{T_E^n - T_L^n}{\Delta \mu_r} - \frac{T_L^n - T_W^n}{\Delta \mu_l}}{\Delta \mu_r + \Delta \mu_l}.$$
(8.90)

Using equations (8.78) and (8.79),

$$c_{02} = \frac{\frac{T_N^n - T_L^n}{\Delta \psi_u} - \frac{T_L^n - T_S^n}{\Delta \psi_d}}{\Delta \psi_u + \Delta \psi_d}.$$
(8.91)

Chapter 8 Tracer advection schemes

When $v_l^n > 0$, from (8.79) and (8.81),

$$c_{01} = \frac{\Delta\mu_W (T_L^n - T_S^n) + \Delta\mu_L (T_W^n - T_{SW}^n)}{2\Delta\mu_l \Delta\psi_d} + c_{02}\Delta\psi_d,$$
(8.92)

$$c_{11} = \frac{T_{SW}^n - T_W^n - T_S^n + T_L^n}{\Delta \mu_l \Delta \psi_d}.$$
(8.93)

When $v_l^n < 0$, from (8.78) and (8.80),

$$c_{01} = \frac{\Delta\mu_W (T_N^n - T_L^n) + \Delta\mu_L (T_{NW}^n - T_W^n)}{2\Delta\mu_L \Delta\psi_u} - c_{02}\Delta\psi_u,$$
(8.94)

$$c_{11} = \frac{T_N^n - T_L^n - T_{NW}^n + T_W^n}{\Delta \mu_l \Delta \psi_u}.$$
(8.95)

Next, using equation (8.73), the T^n values at eight points around **d** are

$$T_L^n = d_{00} + d_{01} \frac{\Delta \psi_L}{2} + d_{02} \frac{\Delta \psi_L^2}{4},$$
(8.96)

$$T_{S}^{n} = d_{00} - d_{01} \frac{\Delta \psi_{S}}{2} + d_{02} \frac{\Delta \psi_{S}^{2}}{4},$$
(8.97)

$$T_{N}^{n} = d_{00} + d_{01} \left(\Delta \psi_{L} + \frac{\Delta \psi_{N}}{2} \right) + d_{02} \left(\Delta \psi_{L} + \frac{\Delta \psi_{N}}{2} \right)^{2},$$
(8.98)

$$T_{SS}^{n} = d_{00} - d_{01} \left(\Delta \psi_{S} + \frac{\Delta \psi_{SS}}{2} \right) + d_{02} \left(\Delta \psi_{S} + \frac{\Delta \psi_{SS}}{2} \right)^{2},$$
(8.99)

$$T_E^n = T_L^n + d_{10}\Delta\mu_r + d_{20}\Delta\mu_r^2 + d_{11}\frac{\Delta\psi_L}{2}\Delta\mu_r,$$
(8.100)

$$T_W^n = T_L^n - d_{10}\Delta\mu_l + d_{20}\Delta\mu_l^2 - d_{11}\frac{\Delta\psi_L}{2}\Delta\mu_l,$$
(8.101)

$$T_{SE}^{n} = T_{S}^{n} + d_{10}\Delta\mu_{r} + d_{20}\Delta\mu_{r}^{2} - d_{11}\frac{\Delta\psi_{S}}{2}\Delta\mu_{r},$$
(8.102)

$$T_{SW}^{n} = T_{S}^{n} - d_{10}\Delta\mu_{l} + d_{20}\Delta\mu_{l}^{2} + d_{11}\frac{\Delta\psi_{S}}{2}\Delta\mu_{l}.$$
(8.103)

From equations (8.96) and (8.97),

$$d_{00} = \frac{\Delta\psi_S T_L^n + \Delta\psi_L T_S^n}{2\Delta\psi_d} - d_{02} \frac{\Delta\psi_L \Delta\psi_S}{4}, \qquad (8.104)$$

$$d_{01} = \frac{T_L^n - T_S^n}{\Delta \psi_d} - d_{02} \frac{\Delta \psi_L - \Delta \psi_S}{2}.$$
 (8.105)

When $v_d^n > 0$, from (8.96) and (8.99),

$$d_{02} = \frac{\frac{T_L^n - T_S^n}{\Delta \psi_d} - \frac{T_S^n - T_{SS}^n}{\Delta \psi_{dd}}}{\Delta \psi_d + \Delta \psi_{dd}},$$
(8.106)
where $\Delta \psi_{dd} \equiv \frac{\Delta \psi_S + \Delta \psi_{SS}}{2}.$

From (8.102) and (8.103),

$$d_{20} = \frac{\frac{T_{SE}^{n} - T_{S}^{n}}{\Delta \mu_{r}} - \frac{T_{S}^{n} - T_{SW}^{n}}{\Delta \mu_{l}}}{\Delta \mu_{r} + \Delta \mu_{l}}.$$
(8.107)

When $v_d^n < 0$, from (8.97) and (8.98),

$$d_{02} = \frac{\frac{T_N^n - T_L}{\Delta \psi_u} - \frac{T_L^n - T_S^n}{\Delta \psi_d}}{\Delta \psi_u + \Delta \psi_d}.$$
(8.108)

-n

-m

From (8.100) and (8.101),

$$d_{20} = \frac{\frac{T_E^n - T_L^n}{\Delta \mu_r} - \frac{T_L^n - T_W^n}{\Delta \mu_l}}{\Delta \mu_r + \Delta \mu_l}.$$
(8.109)

When $u_d^n > 0$, from (8.101) and (8.103),

$$d_{10} = \frac{\Delta \psi_S (T_L^n - T_W^n) + \Delta \psi_L (T_S^n - T_{SW}^n)}{2\Delta \psi_d \Delta \mu_l} + d_{20} \Delta \mu_l,$$
(8.110)

$$d_{11} = \frac{T_L^n - T_S^n - T_W^n + T_{SW}^n}{\Delta \psi_d \Delta \mu_l}.$$
(8.111)

When $u_d^n < 0$, from (8.100) and (8.102),

$$d_{10} = \frac{\Delta \psi_S (T_E^n - T_L^n) + \Delta \psi_L (T_{SE}^n - T_S^n)}{2\Delta \psi_d \Delta \mu_r} - d_{20} \Delta \mu_r,$$
(8.112)

$$d_{11} = \frac{T_E^n - T_{SE}^n - T_L^n + T_S^n}{\Delta \psi_d \Delta \mu_r}.$$
(8.113)

The value of $\overline{T_l^n}$ is the average of T^n over the hatched area of Figure 8.4. Defining

$$\xi_l^n = \frac{u_l^n}{h_{\mu l}}, \quad \eta_l^n = \frac{v_l^n}{h_{\psi l}},$$
 (8.114)

we have

$$\overline{T_{l}^{n}} = \frac{1}{\xi_{l}^{n} \Delta t \Delta \psi_{L}} \left[\int_{\psi_{L} - \Delta \psi_{L}/2}^{\psi_{L} + \Delta \psi_{L}/2} \int_{\mu_{l} - \xi_{l}^{n} \Delta t}^{\mu_{l}} T^{n} d\mu d\psi + \int_{\psi_{L} - \Delta \psi_{L}/2 - \eta_{l}^{n} \Delta t}^{\psi_{L} - \Delta \psi_{L}/2} \int_{\mu_{l} - \xi_{l}^{n} \Delta t}^{\mu_{l} + \frac{\xi_{l}^{n}}{\eta_{l}^{n}} (\psi - \psi_{L} + \Delta \psi_{L}/2)} \{T^{n}(\psi) - T^{n}(\psi + \Delta \psi_{L})\} d\mu d\psi \right]$$
$$= c_{00} - \frac{1}{2} \eta_{l}^{n} \Delta t c_{01} + \left[\frac{1}{12} \Delta \psi_{L}^{2} + \frac{1}{3}^{*} (\eta_{l}^{n} \Delta t)^{2} \right] c_{02} - \frac{1}{2} \xi_{l}^{n} \Delta t c_{10} + \frac{1}{3} (\xi_{l}^{n} \Delta t)^{2} c_{20} + \frac{1}{3}^{*} \xi_{l}^{n} \eta_{l}^{n} \Delta t^{2} c_{11}.$$
(8.115)

This is the result for $u_l^n > 0$ and $v_l^n > 0$. The result is the same independent of the sign of u_l^n and v_l^n . Similarly,

$$\overline{T_d^n} = d_{00} - \frac{1}{2} \eta_d^n \Delta t \, d_{01} + \frac{1}{3} (\eta_d^n \Delta t)^2 \, d_{02} - \frac{1}{2} \xi_d^n \Delta t \, d_{10} + \left[\frac{1}{12} \Delta \mu_L^2 + \frac{1}{3} {}^{\ddagger} (\xi_d^n \Delta t)^2 \right] d_{20} + \frac{1}{3} {}^{\$} \xi_d^n \eta_d^n \Delta t^2 d_{11},$$
(8.116)

where

$$\xi_d^n = \frac{u_d^n}{h_{\mu d}}, \quad \eta_d^n = \frac{v_d^n}{h_{\psi d}}.$$
 (8.117)

 $[\]frac{1}{2}$ in COCO

 $[\]uparrow \frac{1}{4} \text{ in COCO}$ $\uparrow \frac{1}{4} \text{ in COCO}$ $\ddagger \frac{1}{2} \text{ in COCO}$ $\$ \frac{1}{4} \text{ in COCO}$

Chapter 8 Tracer advection schemes

Therefore,

$$\begin{split} \tilde{T}_{l}^{\tilde{n}} &= c_{00} - \frac{1}{2} \xi_{l}^{n} \Delta t \, c_{10} + \left[\frac{1}{3} (\xi_{l}^{n} \Delta t)^{2} - \frac{\Delta \mu_{L}^{2}}{12} \right] c_{20} \\ &- \frac{1}{2} \eta_{l}^{n} \Delta t \, c_{01} + \frac{1}{3} (\eta_{l}^{n} \Delta t)^{2} c_{02} + \frac{1}{3} \xi_{l}^{n} \eta_{l}^{n} \Delta t^{2} \, c_{11}, \end{split}$$

$$\tilde{T}_{d}^{\tilde{n}} &= d_{00} - \frac{1}{2} \eta_{d}^{n} \Delta t \, d_{01} + \left[\frac{1}{3} (\eta_{d}^{n} \Delta t)^{2} - \frac{\Delta \psi_{L}^{2}}{12} \right] d_{02} \end{split}$$

$$(8.118)$$

$$-\frac{1}{2}\xi_{d}^{n}\Delta t \, d_{10} + \frac{1}{3}(\xi_{d}^{n}\Delta t)^{2}d_{20} + \frac{1}{3}\xi_{d}^{n}\eta_{d}^{n}\Delta t^{2} \, d_{11}.$$
(8.119)

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

$$(T_N^n - T_L^n)\Delta\psi_d^2 = (T_S^n - T_L^n)\Delta\psi_u^2.$$
(8.120)

When **NW** or **SW** is a land grid, the following should be assumed.

$$(T_{NW}^{n} - T_{W}^{n})\Delta\psi_{d}^{2} = (T_{SW}^{n} - T_{W}^{n})\Delta\psi_{u}^{2}$$
(8.121)

When WW is a land grid,

$$(T_{WW}^n - T_W^n)\Delta\mu_l^2 = (T_L^n - T_W^n)\Delta\mu_{ll}^2.$$
(8.122)

When **E** is a land grid,

$$(T_E^n - T_L^n)\Delta\mu_l^2 = (T_W^n - T_L^n)\Delta\mu_r^2.$$
(8.123)

Similar boundary conditions are specified for face d.

8.6 Second Order Moment (SOM) scheme

8.6.1 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 $(0 \le x \le X, 0 \le y \le Y, 0 \le z \le Z;$ volume V = XYZ can be represented using second order functions as follows:

$$f(x, y, z) = a_0 + a_x x + a_{xx} x^2 + a_y y + a_{yy} y^2 + a_z z + a_{zz} z^2 + a_{xy} xy + a_{yz} yz + a_{zx} zx.$$
(8.124)

Prather (1986) expressed the above as a sum of orthogonal functions $K_i(x, y, z)$;

$$f(x, y, z) = m_0 K_0 + m_x K_x + m_{xx} K_{xx} + m_y K_y + m_{yy} K_{yy} + m_z K_z + m_{zz} K_{zz} + m_{xy} K_{xy} + m_{yz} K_{yz} + m_{zx} K_{zx}, \quad (8.125)$$

(8.127)

where the orthogonal functions are given as follows:

$$K_{0} = 1,$$

$$K_{x}(x) = x - X/2,$$

$$K_{xx}(x) = x^{2} - Xx + X^{2}/6,$$

$$K_{y}(y) = y - Y/2,$$

$$K_{yy}(y) = y^{2} - Yy + Y^{2}/6,$$

$$K_{z}(z) = z - Z/2,$$

$$K_{zz}(z) = z^{2} - Zz + Z^{2}/6,$$

$$K_{xy}(x, y) = (x - X/2)(y - Y/2),$$

$$K_{yz}(y, z) = (y - Y/2)(z - Z/2),$$

$$K_{zx}(z, x) = (z - Z/2)(x - X/2),$$
(8.126)

and

The constants for normalization are decided using

$$\int K_x^2 dV = VX^2/12, \quad \int K_{xx}^2 dV = VX^4/180,$$
$$\int K_y^2 dV = VY^2/12, \quad \int K_{yy}^2 dV = VY^4/180,$$
$$\int K_z^2 dV = VZ^2/12, \quad \int K_{zz}^2 dV = VZ^4/180,$$
$$\int K_{xy}^2 dV = VX^2Y^2/144, \quad \int K_{yz}^2 dV = VY^2Z^2/144, \quad \int K_{zx}^2 dV = VZ^2X^2/144.$$

 $\int K_i K_j dV = 0 \quad (i \neq j).$

The moments are set by the following expressions:

$$S_{0} = \int f(x, y, z)K_{0}dV = m_{0}V,$$

$$S_{x} = (6/X) \int f(x, y, z)K_{x}(x)dV = m_{x}VX/2,$$

$$S_{xx} = (30/X^{2}) \int f(x, y, z)K_{xx}(x)dV = m_{xx}VX^{2}/6,$$

$$S_{y} = (6/Y) \int f(x, y, z)K_{y}(y)dV = m_{y}VY/2,$$

$$S_{yy} = (30/Y^{2}) \int f(x, y, z)K_{z}(z)dV = m_{z}VZ/2,$$

$$S_{z} = (6/Z) \int f(x, y, z)K_{z}(z)dV = m_{zz}VZ^{2}/6,$$

$$S_{zz} = (30/Z^{2}) \int f(x, y, z)K_{xy}(x, y)dV = m_{xy}VXY/4,$$

$$S_{yz} = (36/XY) \int f(x, y, z)K_{yz}(y, z)dV = m_{yz}VYZ/4,$$

$$S_{zx} = (36/ZX) \int f(x, y, z)K_{zx}(z, x)dV = m_{zx}VZX/4.$$
(8.128)

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 procedures use the results of the last procedure. For simplicity, we describe the change

Chapter 8 Tracer advection schemes

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).

When velocity c in the x direction is positive, the right part of the grid cell,

$$X - ct \le x \le X, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{8.129}$$

is removed from the cell and added to the adjacent cell on the right during time interval t. This part is expressed using superscript R. The remaining part,

$$0 \le x \le X - ct, \quad 0 \le y \le Y, \quad 0 \le z \le Z,$$
 (8.130)

is expressed by superscript *L*. New orthogonal functions $K_i^R(K_i^L)$ are calculated in the part *R* (*L*) with the volume $V^R = ctYZ$ ($V^L = (X - ct)YZ$). The orthogonal functions are given as follows:

$$K_{0}^{L} = K_{0}^{R} = 1,$$

$$K_{x}^{L} = x - (X - ct)/2, \quad K_{x}^{R} = x - (2X - ct)/2,$$

$$K_{xx}^{L} = x^{2} - (X - ct)x + (X - ct)^{2}/6,$$

$$K_{xx}^{R} = x^{2} - (2X - ct)x + (X - ct)X + (ct)^{2}/6,$$

$$K_{yy}^{L} = K_{y}^{R} = y - Y/2,$$

$$K_{yy}^{L} = K_{yy}^{R} = y^{2} - Yy + Y^{2}/6,$$

$$K_{xy}^{L} = [x - (X - ct)/2](y - Y/2),$$

$$K_{xy}^{R} = [x - (2X - ct)/2](y - Y/2).$$
(8.131)

The basic quantities for calculating the moments are

$$m_{0}^{R} = m_{0} + \bar{K}_{x}^{R}m_{x} + \bar{K}_{xx}^{R}m_{xx},$$

$$m_{x}^{R} = m_{x} + 2\bar{K}_{x}^{R}m_{xx},$$

$$m_{xx}^{R} = m_{xx},$$

$$m_{y}^{R} = m_{y} + 2\bar{K}_{x}^{R}m_{xy},$$

$$m_{yy}^{R} = m_{yy},$$

$$m_{xy}^{R} = m_{xy},$$
(8.132)

where \bar{K} is the average of the new orthogonal function:

$$\bar{K}_x^L = -ct/2, \quad \bar{K}_x^R = (X - ct)/2,$$

 $\bar{K}_{xx}^L = ct(2ct - X)/6, \quad \bar{K}_{xx}^R = (X - ct)(X - 2ct)/6.$

The moments in the right part to be removed are expressed as follows:

$$S_{0}^{R} = \alpha [S_{0} + (1 - \alpha)S_{x} + (1 - \alpha)(1 - 2\alpha)S_{xx}],$$

$$S_{x}^{R} = \alpha^{2} [S_{x} + 3(1 - \alpha)S_{xx}],$$

$$S_{xx}^{R} = \alpha^{3}S_{xx},$$

$$S_{y}^{R} = \alpha [S_{y} + (1 - \alpha)S_{xy}],$$

$$S_{yy}^{R} = \alpha S_{yy},$$

$$S_{xy}^{R} = \alpha^{2}S_{xy},$$
(8.133)

where $\alpha = \alpha^R = ct/X = V^R/V$. The moments in the remaining part are expressed as follows:

$$S_{0}^{L} = (1 - \alpha)[S_{0} - \alpha S_{x} - \alpha(1 - 2\alpha)S_{xx}],$$

$$S_{x}^{L} = (1 - \alpha)^{2}(S_{x} - 3\alpha S_{xx}),$$

$$S_{xx}^{L} = (1 - \alpha)^{3}S_{xx},$$

$$S_{y}^{L} = (1 - \alpha)(S_{y} - \alpha S_{xy}),$$

$$S_{yy}^{L} = (1 - \alpha)S_{yy},$$

$$S_{xy}^{L} = (1 - \alpha)^{2}S_{xy}.$$
(8.134)

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. The calculation is terribly complex, and only the results are presented: $S_0 = S_0^R + S_0^L$

$$S_{0} = S_{0}^{R} + S_{0}^{R},$$

$$S_{x} = \alpha S_{x}^{R} + (1 - \alpha) S_{x}^{L} + 3[(1 - \alpha)S_{0}^{R} - \alpha S_{0}^{L}],$$

$$S_{xx} = \alpha^{2} S_{xx}^{R} + (1 - \alpha)^{2} S_{xx}^{L} + 5\{\alpha(1 - \alpha)(S_{x}^{R} - S_{x}^{L}) + (1 - 2\alpha)[(1 - \alpha)S_{0}^{R} - \alpha S_{0}^{L}]\},$$

$$S_{y} = S_{y}^{R} + S_{y}^{L},$$

$$S_{yy} = S_{yy}^{R} + S_{yy}^{L},$$

$$S_{xy} = \alpha S_{xy}^{R} + (1 - \alpha)S_{xy}^{L} + 3[(1 - \alpha)S_{y}^{R} - \alpha S_{y}^{L}],$$

$$\alpha = \alpha^{R} = V^{R}/(V^{R} + V^{L}).$$
(8.135)

where

The globally integrated tracer is conserved through these operations.

8.6.2 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,

$$S_{0} \ge 0,$$

$$S'_{x} = \min[+1.5S_{0}, \max(-1.5S_{0}, S_{x})],$$

$$S'_{xx} = \min[2S_{0} - |S'_{x}|/3, \max(|S'_{x}| - S_{0}, S_{xx})],$$

$$S'_{xy} = \min[+S_{0}, \max(-S_{0}, S_{xy})].$$

(8.136)

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).

$$\bar{\tau}(x) = \frac{1}{YZ} \int_0^Y \int_0^Z dy \, dz \, \tau(x, y, z) = \frac{1}{V} \left[S_0 - S_x + S_{xx} + 2(S_x - 3S_{xx})\frac{x}{X} + 6S_{xx} \left(\frac{x}{X}\right)^2 \right],$$
(8.137)

Chapter 8 Tracer advection schemes

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 $0 \le x \le X$ (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:

$$V\min\bar{\tau}(x) = \begin{cases} S_0 - |S_x| + S_{xx}, & \text{if } S_{xx} \le 0 & (\text{region I}), \\ S_0 - |S_x| + S_{xx}, & \text{if } |S_x| \ge 3S_{xx} \ge 0 & (\text{region II}), \\ S_0 - \frac{S_x^2 + 3S_{xx}^2}{6S_{xx}}, & \text{if } 3S_{xx} \ge |S_x| & (\text{region III}). \end{cases}$$
(8.138)

The line of $V \min \bar{\tau}(x) = 0$ is plotted in the $(|S_x|, S_{xx})$ space in Figure 8.5. 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.



Figure 8.5 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,

$$S'_{x} = \min \left[3^{1/2} S_{0}, \max \left[-3^{1/2} S_{0}, S_{x} \right] \right],$$

$$S'_{xx} = \begin{cases} \min \left[S_{0} + \left(S_{0}^{2} - \frac{S_{x}^{\prime 2}}{3} \right)^{1/2}, \max \left[|S'_{x}| - S_{0}, S_{xx} \right] \right], & \text{if } |S'_{x}| < \frac{3}{2} S_{0}, \\ \sin \left[S_{0} + \left(S_{0}^{2} - \frac{S_{x}^{\prime 2}}{3} \right)^{1/2}, \max \left[S_{0} - \left(S_{0}^{2} - \frac{S_{x}^{\prime 2}}{3} \right)^{1/2}, S_{xx} \right] \right], & \text{if } |S'_{x}| \ge \frac{3}{2} S_{0}, \\ S'_{xy} = \min \left[S_{0}, \max[-S_{0}, S_{xy}] \right], \\ S'_{xz} = \min \left[S_{0}, \max[-S_{0}, S_{xz}] \right]. \end{cases}$$

$$(8.139)$$

Here, $S'_{xx}, S'_{xx}, S'_{xy}$ are modified moments and it is assumed that $S_0 \ge 0$. This is slightly different from PR86 because an approximation $\sqrt{3} \sim 1.5$ (see 8.136) was used in PR86.

This modification is explained using Figure 8.5. 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.

8.6 Second Order Moment (SOM) scheme

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 ($\tau_a \le \overline{\tau}(x) \le \tau_b$). 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 $\overline{\tau}(x)$ must be less than τ_b " as "the minimum of $-\overline{\tau}(x)$ must be more than $-\tau_b$." In other words, the condition is "the minimum of $\tau_b - \overline{\tau}(x)$ must be positive."

Using (8.137), $\tau_b - \bar{\tau}(x)$ may be expressed as follows:

$$\tau_{b} - \bar{\tau}(x) = \frac{1}{V} \left[V \tau_{b} - S_{0} + S_{x} - S_{xx} - 2(S_{x} - 3S_{xx})\frac{x}{X} - 6S_{xx}\left(\frac{x}{X}\right)^{2} \right]$$
$$= \frac{1}{V} \left[S_{0}^{*} + S_{x} - S_{xx} - 2(S_{x} - 3S_{xx})\frac{x}{X} - 6S_{xx}\left(\frac{x}{X}\right)^{2} \right],$$
(8.140)

where $S_0^* = V \tau_b - S_0$. Its minimum is obtained as follows:

$$V \min[\tau_b - \bar{\tau}(x)] = \begin{cases} S_0^* - |S_x| - S_{xx}, & \text{if } S_{xx} \ge 0 & (\text{region I}), \\ S_0^* - |S_x| - S_{xx}, & \text{if } |S_x| \ge -3S_{xx} \ge 0 & (\text{region II}), \\ S_0^* + \frac{S_x^2 + 3S_{xx}^2}{6S_{xx}}, & \text{if } -3S_{xx} \ge |S_x| & (\text{region III}). \end{cases}$$
(8.141)

Following Figure 8.5, this is visualized as Figure 8.6.



Figure 8.6 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 (8.139) is expressed as follows:

$$S'_{x} = \min \left[3^{1/2} S_{0}^{*}, \max \left[-3^{1/2} S_{0}^{*}, S_{x} \right] \right],$$

$$S'_{xx} = \begin{cases} \min \left[S_{0}^{*} - |S'_{x}|, \max \left[-S_{0}^{*} - \left(S_{0}^{*2} - \frac{S'_{x}^{2}}{3} \right)^{1/2}, S_{xx} \right] \right], & \text{if } |S'_{x}| < \frac{3}{2} S_{0}^{*},$$

$$\left[-S_{0}^{*} + \left(S_{0}^{*2} - \frac{S'_{x}^{2}}{3} \right)^{1/2}, \max \left[-S_{0}^{*} - \left(S_{0}^{*2} - \frac{S'_{x}^{2}}{3} \right)^{1/2}, S_{xx} \right] \right], & \text{if } |S'_{x}| \ge \frac{3}{2} S_{0}^{*},$$

$$S'_{xy} = \min \left[S_{0}^{*}, \max(-S_{0}^{*}, S_{xy}) \right],$$

$$S'_{xz} = \min \left[S_{0}^{*}, \max(-S_{0}^{*}, S_{xz}) \right].$$
(8.142)

MH06 proposed three modification methods (Method A - C). Following COCO, 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. (8.139) and (8.142). This method does not guarantee exact monotonicity, but improves monotonicity of the zeroth moment.

Calculating SOM advection in MRI.COM 8.6.3

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 (USTARL, VSTARL, and WLWL), 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 8.2) 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.

name	explanation	string or value
file_som_in	base name of the restart file to be input	{file_som_in}_i_n *
file_som_out	base name of the restart file to be output	{file_som_out}_i_n *
<pre>nstep_som_restart_interval</pre>	interval in time steps with which restart	default is 0 (at the end of this job)
	files are output	
lsomstrang	The order of calling x and y direction is	.true./.false.
	reversed every time step	
lsommonitor	flag to monitor the conservation of the	.true./.false.
	moments	
* i - (x x x y y x 7 77 x y y 7 7x) n - (0)	1.02 numtre le tracer number	

Table8.2 Variables defined in namelist nml_somadv

 $i = \{x, xx, y, yy, z, zz, xy, yz, zx\}, n = \{01, 02, ..., numtrc\}; tracer number$

MPDATA scheme 8.7

8.7.1 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.

Original dimensionless form of the anti-diffusive velocity given by Smolarkiewicz and Margolin (1998) is written as

follows (their equation 16):

$$\mathcal{U}^{(1)} \equiv \frac{u^{(1)}\delta t}{\delta x} = |\mathcal{U}|(1-\mathcal{U})A^{(1)} - \mathcal{U}\mathcal{V}B^{(1)} - \mathcal{U}\mathcal{W}C^{(1)},$$
(8.143)

$$\mathcal{V}^{(1)} \equiv \frac{v^{(1)}\delta t}{\delta y} = |\mathcal{V}|(1-\mathcal{V})B^{(1)} - \mathcal{V}\mathcal{W}C^{(1)} - \mathcal{V}\mathcal{U}A^{(1)},\tag{8.144}$$

$$\mathcal{W}^{(1)} \equiv \frac{w^{(1)}\delta t}{\delta z} = |\mathcal{W}|(1-\mathcal{W})C^{(1)} - \mathcal{W}\mathcal{U}A^{(1)} - \mathcal{W}\mathcal{V}B^{(1)}, \tag{8.145}$$

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

$$A^{(1)} \equiv \left[\frac{\delta x}{2T}\frac{\partial T}{\partial x}\right]^{(1)},\tag{8.146}$$

$$B^{(1)} \equiv \left[\frac{\delta y}{2T}\frac{\partial T}{\partial y}\right]^{(1)},\tag{8.147}$$

$$C^{(1)} \equiv \left[\frac{\delta z}{2T}\frac{\partial T}{\partial z}\right]^{(1)}.$$
(8.148)

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 (8.7) to (8.9), the dimensionless form of volume transports can be written as

$$\mathcal{U}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{2U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T}\Delta t}{\Delta x_{i+\frac{1}{2}j}\Delta y_{i+\frac{1}{2},j} \left(\Delta z_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right),$$
(8.149)

$$\mathcal{V}_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{2V_{i,j+\frac{1}{2},k-\frac{1}{2}}\Delta t}{\Delta x_{i,j+\frac{1}{2}}\Delta y_{i,j+\frac{1}{2}} \left(\Delta z_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right),$$
(8.150)

$$\mathcal{W}_{i,j,k} = \frac{W_{i,j,k}^T \Delta t}{\Delta z_{i,j,k} (\operatorname{areat})_{i,j,k+\frac{1}{2}}}.$$
(8.151)

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

$$A_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} = \frac{\Delta x_{i+\frac{1}{2},j} \delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)}}{2\overline{T_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)}} + \epsilon}$$
(8.152)

$$B_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} = \frac{\Delta y_{i+\frac{1}{2},j} \overline{\delta_y T_{i+\frac{1}{2},j}^{(1)}}}{\frac{1}{2T_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)}} + \epsilon}$$
(8.153)

$$C_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} = \frac{\Delta z_{i+\frac{1}{2},j} \overline{\delta_z T_{i+\frac{1}{2},j}^{(1)}}^{x^z}}{2\overline{\overline{T_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)}}}^{x^z} + \epsilon}$$
(8.154)

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):

$$\delta_{x}A_{i} \equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta x_{i}}, \quad \delta_{x}A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_{i}}{\Delta x_{i+\frac{1}{2}}},$$
$$\overline{A_{i}}^{x} \equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2}, \quad \overline{A_{i+\frac{1}{2}}}^{x} \equiv \frac{A_{i+1} + A_{i}}{2}.$$
(8.155)

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

$$\begin{aligned} U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} &= \left| U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T} \right| (1 - \mathcal{U}_{i+\frac{1}{2},j,k-\frac{1}{2}}) A_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} - U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T} \overline{\mathcal{V}_{i+\frac{1}{2},j,k-\frac{1}{2}}^{x^{y}}} B_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} - U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{T} \overline{\mathcal{W}_{i+\frac{1}{2},j,k-\frac{1}{2}}^{x^{z}}} C_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)}, \\ (8.156) \\ V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{(1)} &= \left| V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{T} \right| (1 - \mathcal{V}_{i,j+\frac{1}{2},k-\frac{1}{2}}) B_{i,j+\frac{1}{2},k-\frac{1}{2}}^{(1)} - V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{T} \overline{\mathcal{V}_{i,j+\frac{1}{2},k-\frac{1}{2}}^{x^{y}}} A_{i,j+\frac{1}{2},k-\frac{1}{2}}^{(1)} - V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{T} \overline{\mathcal{V}_{i,j+\frac{1}{2},k-\frac{1}{2}}^{x^{y}}} C_{i,j+\frac{1}{2},k-\frac{1}{2}}^{(1)}, \\ W_{i,j+\frac{1}{2},k-\frac{1}{2}}^{(1)} &= \left| W_{i,j,k}^{T} \right| (1 - \mathcal{W}_{i,j,k}) C_{i,j,k}^{(1)} - W_{i,j,k}^{T} \overline{\mathcal{U}_{i+\frac{1}{2},j,k}^{x^{y}}} A_{i,j,k}^{(1)} - W_{i,j,k}^{T} \overline{\mathcal{V}_{i+\frac{1}{2},j,k}^{x^{y}}} B_{i,j,k}^{(1)}. \end{aligned}$$

$$(8.158)$$

8.7.2 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:

$$\begin{bmatrix} U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} \end{bmatrix}^{mon} = \begin{cases} \min(1,\beta_{i,j,k-\frac{1}{2}}^{out},\beta_{i+1,j,k-\frac{1}{2}}^{in})U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} & \text{if } U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} > 0, \\ \min(1,\beta_{i+1,j,k-\frac{1}{2}}^{out},\beta_{i,j,k-\frac{1}{2}}^{in})U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} & \text{if } U_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} < 0, \end{cases}$$
(8.159)

$$\begin{bmatrix} V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{(1)} \end{bmatrix}^{mon} = \begin{cases} \min(1,\beta_{i,j,k-\frac{1}{2}}^{out},\beta_{i,j+1,k-\frac{1}{2}}^{in})V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{(1)} & \text{if } V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{(1)} > 0, \\ \min(1,\beta_{i,j+1,k-\frac{1}{2}}^{out},\beta_{i,j,k-\frac{1}{2}}^{in})V_{i+\frac{1}{2},j,k-\frac{1}{2}}^{(1)} & \text{if } V_{i,j+\frac{1}{2},k-\frac{1}{2}}^{(1)} < 0, \end{cases}$$
(8.160)

$$\left[W_{i,j,k}^{(1)} \right]^{mon} = \begin{cases} \min(1, \beta_{i,j,k+\frac{1}{2}}^{out}, \beta_{i,j,k-\frac{1}{2}}^{in}) W_{i,j,k}^{(1)} & \text{if } W_{i,j,k}^{(1)} > 0, \\ \min(1, \beta_{i,j,k-\frac{1}{2}}^{out}, \beta_{i,j,k+\frac{1}{2}}^{in}) W_{i,j,k}^{(1)} & \text{if } W_{i,j,k}^{(1)} < 0, \end{cases}$$

$$(8.161)$$

where

$$\beta_{i,j,k-\frac{1}{2}}^{in} \equiv \frac{T_{i,j,k-\frac{1}{2}}^{max} - T_{i,j,k-\frac{1}{2}}^{(1)}}{A_{i,j,k-\frac{1}{2}}^{in} + \epsilon},$$
(8.162)

$$\beta_{i,j,k-\frac{1}{2}}^{out} \equiv \frac{T_{i,j,k-\frac{1}{2}}^{(1)} - T_{i,j,k-\frac{1}{2}}^{min}}{A_{i,j,k-\frac{1}{2}}^{out} + \epsilon},$$
(8.163)

$$T_{i,j,k-\frac{1}{2}}^{max} = \max\left(\begin{array}{cccc} T_{i,j,k-\frac{1}{2}}, & T_{i+1,j,k-\frac{1}{2}}, & T_{i-1,j,k-\frac{1}{2}}, & T_{i,j+1,k-\frac{1}{2}}, & T_{i,j-1,k-\frac{1}{2}}, & T_{i,j,k+\frac{1}{2}}, & T_{i,j,k-\frac{3}{2}}, \\ T_{i,j,k-\frac{1}{2}}^{(1)}, & T_{i+1,j,k-\frac{1}{2}}^{(1)}, & T_{i-1,j,k-\frac{1}{2}}^{(1)}, & T_{i,j+1,k-\frac{1}{2}}^{(1)}, & T_{i,j,k+\frac{1}{2}}^{(1)}, & T_{i,j,k-\frac{3}{2}}^{(1)}\right),$$
(8.164)

$$T_{i,j,k-\frac{1}{2}}^{min} = \min\left(\begin{array}{ccc} T_{i,j,k-\frac{1}{2}}, & T_{i+1,j,k-\frac{1}{2}}, & T_{i-1,j,k-\frac{1}{2}}, & T_{i,j+1,k-\frac{1}{2}}, & T_{i,j-1,k-\frac{1}{2}}, & T_{i,j,k+\frac{1}{2}}, & T_{i,j,k-\frac{3}{2}}, \\ T_{i,j,k-\frac{1}{2}}^{(1)}, & T_{i+1,j,k-\frac{1}{2}}^{(1)}, & T_{i,j+1,k-\frac{1}{2}}^{(1)}, & T_{i,j-1,k-\frac{1}{2}}^{(1)}, & T_{i,j,k+\frac{1}{2}}^{(1)}, & T_{i,j,k-\frac{3}{2}}^{(1)}, \end{array}\right),$$
(8.165)

and Aⁱⁿ and A^{out} are the absolute values of the total incoming and outgoing advection flux at the T-box.

8.7.3 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 9

SGS parameterization of lateral mixing of tracers

This chapter explains subgrid-scale parameterizations for horizontal mixing of tracers.

9.1 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 (20.22) and (20.23), the harmonic-type diffusivity is represented for any tracer (T) as follows:

$$-z_{s}\nabla \cdot \mathbf{F}_{T} = \mathcal{D}(T) = \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial}{\partial\mu} \left(\frac{h_{\psi}z_{s}\kappa_{H}}{h_{\mu}} \frac{\partial T}{\partial\mu} \right) + \frac{\partial}{\partial\psi} \left(\frac{h_{\mu}z_{s}\kappa_{H}}{h_{\psi}} \frac{\partial T}{\partial\psi} \right) \right\} + \frac{\partial}{\partial s} \left(\frac{\kappa_{V}}{z_{s}} \frac{\partial T}{\partial s} \right)$$
$$= -\frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial(h_{\psi}z_{s}F_{\mu}^{T})}{\partial\mu} + \frac{\partial(h_{\mu}z_{s}F_{\psi}^{T})}{\partial\psi} \right\} - \frac{\partial F_{z}^{T}}{\partial s}, \tag{9.1}$$

where κ_H and κ_V are the horizontal and vertical diffusion coefficients and the diffusive fluxes are represented by *F*. $\mathcal{D}(T)$ represents the diffusion operator for *T*. 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):

$$\frac{DT}{Dt} + \nabla_H \cdot \left[T \frac{\partial}{\partial z} (\kappa_T \nabla_H \rho / \rho_z) \right] + \frac{\partial}{\partial z} \left[T \nabla_H \cdot (-\kappa_T \nabla_H \rho / \rho_z) \right] = \mathcal{D}(T) + Q^T,$$
(9.2)

where the first term on the r.h.s. is the isopycnal diffusion, whose form is given by

$$\mathcal{D}(T) = \nabla \cdot (\kappa_I \mathbf{K} \nabla T), \tag{9.3}$$

where

$$\mathbf{K} = \frac{1}{\rho_x^2 + \rho_y^2 + \rho_z^2} \begin{pmatrix} \rho_y^2 + \rho_z^2 & -\rho_x \rho_y & -\rho_x \rho_z \\ -\rho_x \rho_y & \rho_x^2 + \rho_z^2 & -\rho_y \rho_z \\ -\rho_x \rho_z & -\rho_y \rho_z & \rho_x^2 + \rho_y^2 \end{pmatrix}$$
(9.4)

$$= \frac{1}{1 + (\rho_x/\rho_z)^2 + (\rho_y/\rho_z)^2} \begin{pmatrix} 1 + (\rho_y/\rho_z)^2 & -(\rho_x/\rho_z)(\rho_y/\rho_z) & -\rho_x/\rho_z \\ -(\rho_x/\rho_z)(\rho_y/\rho_z) & 1 + (\rho_x/\rho_z)^2 & -\rho_y/\rho_z \\ -\rho_x/\rho_z & -\rho_y/\rho_z & (\rho_x/\rho_z)^2 + (\rho_y/\rho_z)^2 \end{pmatrix},$$
(9.5)

(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 (9.2) have the form of advection terms with a transport velocity vector (u_T, v_T, w_T) :

$$u_T \equiv \frac{\partial}{\partial z} \Big(\kappa_T \frac{1}{h_\mu} \frac{\partial \rho}{\partial \mu} / \frac{\partial \rho}{\partial z} \Big), \tag{9.6}$$

$$v_T \equiv \frac{\partial}{\partial z} \Big(\kappa_T \frac{1}{h_{\psi}} \frac{\partial \rho}{\partial \psi} / \frac{\partial \rho}{\partial z} \Big), \tag{9.7}$$

$$w_T \equiv -\frac{1}{h_\mu h_\psi} \Big\{ \frac{\partial}{\partial \mu} \Big(\kappa_T \frac{h_\psi}{h_\mu} \frac{\partial \rho}{\partial \mu} / \frac{\partial \rho}{\partial z} \Big) + \frac{\partial}{\partial \psi} \Big(\kappa_T \frac{h_\mu}{h_\psi} \frac{\partial \rho}{\partial \psi} / \frac{\partial \rho}{\partial z} \Big) \Big\}, \tag{9.8}$$

(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

$$\mathcal{G}(T) = \nabla \cdot (\kappa_T \mathbf{A} \nabla T) \tag{9.9}$$

with

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & -\rho_x/\rho_z \\ 0 & 0 & -\rho_y/\rho_z \\ \rho_x/\rho_z & \rho_y/\rho_z & 0 \end{pmatrix}.$$
 (9.10)

Comparing with (9.5), we notice that the isopycnal diffusion and the thickness diffusion terms are combined to yield a concise form (Griffies, 1998) and (9.2) can be rewritten as:

$$\frac{DT}{Dt} = \nabla \cdot \{ (\kappa_I \mathbf{K} - \kappa_T \mathbf{A}) \nabla T \} + Q^T.$$
(9.11)

Three types of horizontal 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 9.4 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 (9.11) is given by taking the finite volume approach as follows:

$$T_{i,j,k-\frac{1}{2}}^{n+1} \Delta V_{i,j,k-\frac{1}{2}}^{n+1} = T_{i,j,k-\frac{1}{2}}^{n-1} \Delta V_{i,j,k-\frac{1}{2}}^{n-1} + 2 \Delta t \{FXD_{i-\frac{1}{2},j,k-\frac{1}{2}} - FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} + FYD_{i,j-\frac{1}{2},k-\frac{1}{2}} - FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} + FZD_{i,j,k} - FZD_{i,j,k-1}\} + (other terms).$$

$$(9.12)$$

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 19.

9.2 Horizontal diffusion

9.2.1 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:

$$FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} = -\kappa_H \Delta y_{i+\frac{1}{2},j} \overline{\Delta z_{i+\frac{1}{2},j,k-\frac{1}{2}}}^y \delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}},$$
(9.13)

$$FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} = -\kappa_H \Delta x_{i,j+\frac{1}{2}} \overline{\Delta z_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{\lambda} \delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}},$$
(9.14)

where

$$\delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}} \equiv \frac{T_{i+1,j,k-\frac{1}{2}} - T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j}},\tag{9.15}$$

$$\delta_{y}T_{i,j+\frac{1}{2},k-\frac{1}{2}} \equiv \frac{T_{i,j+1,k-\frac{1}{2}} - T_{i,j,k-\frac{1}{2}}}{\Delta y_{i,j+\frac{1}{2}}}.$$
(9.16)

9.2.2 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:

$$FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} = \sqrt{\kappa_b} \,\Delta y_{i+\frac{1}{2},j} \overline{\Delta z_{i+\frac{1}{2},j,k-\frac{1}{2}}}^y \delta_x [Lap(T)]_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{9.17}$$

$$FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} = \sqrt{\kappa_b} \,\Delta x_{i,j+\frac{1}{2}} \overline{\Delta z_{i,j+\frac{1}{2},k-\frac{1}{2}}}^x \delta_y [\text{Lap}(T)]_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{9.18}$$

where κ_b is diffusivity coefficient and

$$Lap(T)_{i,j,k-\frac{1}{2}} = \frac{\Delta x_{i,j} \Delta y_{i,j}}{\Delta V_{i,j,k-\frac{1}{2}}} (\delta_x \sqrt{\kappa_b} \,\overline{\Delta z_{i,j,k-\frac{1}{2}}}^y \delta_x T_{i,j,k-\frac{1}{2}} + \delta_y \sqrt{\kappa_b} \,\overline{\Delta z_{i,j,k-\frac{1}{2}}}^x \delta_y T_{i,j,k-\frac{1}{2}}). \tag{9.19}$$

9.2.3 Specification of coefficient

The diffusion coefficient of horizontal diffusion is specified using the namelists listed on Tables 9.1 and 9.2. This must be zero if ISOPYCNAL option is selected, unless the horizontal diffusion is applied intentionally.

Table9.1	Namelist nml_	_tracer_	_diff_	_horz
----------	---------------	----------	--------	-------

variable name	units	description	usage
diff_horz_cm2ps	$\mathrm{cm}^2\mathrm{s}^{-1}$	Laplacian diffusion coefficient (κ_H)	if not TRCBIHARM
diff_horz_cm4ps	$\mathrm{cm}^4\mathrm{s}^{-1}$	Biharmonic diffusion coefficient (κ_b)	if TRCBIHARM
file_diff_horz_2d	character	2D distribution of diffusion coefficient is read from	
		this file	

Table9.2 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.	

9.3 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 **K**, each flux component is written as $F^m(T) = -K^{mn}\partial_n T$, and then

$$\mathbf{K} = \frac{\kappa_I}{1+S^2} \begin{pmatrix} 1+S_y^2 + \epsilon S_x^2 & (\epsilon-1)S_x S_y & (1-\epsilon)S_x \\ (\epsilon-1)S_x S_y & 1+S_x^2 + \epsilon S_y^2 & (1-\epsilon)S_y \\ (1-\epsilon)S_x & (1-\epsilon)S_y & \epsilon+S^2 \end{pmatrix},$$
(9.20)

where

$$\mathbf{S} = (S_x, S_y, 0) = \begin{pmatrix} \frac{\partial \rho}{\partial x}, \frac{\partial \rho}{\partial y}, \\ \frac{\partial \rho}{\partial z}, -\frac{\partial \rho}{\partial z}, 0 \\ \frac{\partial \rho}{\partial z}, -\frac{\partial \rho}{\partial z} \end{pmatrix},$$
(9.21)

$$S = |\mathbf{S}|, \tag{9.22}$$

$$\epsilon = \frac{\kappa_D}{\kappa_I} \tag{9.23}$$

(Redi, 1982).

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 9.1):

(

$$(\delta_x T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}},\tag{9.24}$$

$$(\delta_{y}T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \overline{\delta_{y}T_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{xy},$$
(9.25)

$$(\delta_z T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \overline{\delta_z T_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{XZ}.$$
(9.26)

Similarly, the north-south component $FYD_{i,j+\frac{1}{2},k-\frac{1}{2}}$ is given by

$$(\delta_x T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \overline{\delta_x T_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{xy}, \tag{9.27}$$

$$(\delta_y T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}},\tag{9.28}$$

$$(\delta_z T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \overline{\delta_z T_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{y_z},\tag{9.29}$$

and the vertical component $FZD_{i,j,k}$ is given by

$$\delta_x T)_{i,j,k} = \overline{\delta_x T_{i,j,k}}^{x_z},\tag{9.30}$$

$$\delta_{\mathbf{v}}T)_{i,i,k} = \overline{\delta_{\mathbf{v}}T_{i,i,k}}^{\mathbf{v}z},\tag{9.31}$$

$$(\delta_z T)_{i,j,k} = \delta_z T_{i,j,k}. \tag{9.32}$$



Figure 9.1 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.

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 9.4) is set because a nearly vertical isopycnal slope and the resultant low horizontal diffusivity could cause numerical instability. If $|\mathbf{S}| > S_{\text{max}}$, $\partial_z \rho$ in all flux components is replaced so as to satisfy $|\mathbf{S}| = S_{\text{max}}$.

Chapter 9 SGS parameterization of lateral mixing of tracers

The vertical flux due to the third diagonal component of the diffusion tensor (9.20) is

$$FZD_{i,j,k} = (\operatorname{areat})_{i,j,k} \frac{\kappa_I(\epsilon + S^2)}{1 + S^2} \delta_z T_{i,j,k}.$$
(9.33)

Thus the effective vertical diffusivity κ_{eff} is

$$\kappa_{\rm eff} = \frac{\kappa_D + \kappa_I S^2}{1 + S^2}.$$
(9.34)

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 $\kappa_D \sim 10^{-1} \text{ cm}^2 \text{ s}^{-1}$,

$$\kappa_{\rm eff} \sim 10^3 \,{\rm cm}^2 \,{\rm s}^{-1}.$$
 (9.35)

This is a fairly large value which warrants use of an implicit scheme (Section 19.5). In MRI.COM, this term is separated from other terms and solved with other vertical diffusion terms using an implicit method.

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.

9.3.1 Tapering around steep isopycnal slopes

As noted above, 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. Griffies (2004) shows that such slope clipping could lead to an unrealistically large tracer flux. Danabasoglu and McWilliams (1995) propose another method that uses a hyperbolic tangent to exponentially taper isopycnal diffusion in steep slope regions. The following factor is multiplied to the isopycnal diffusion coefficient there:

$$f_{\text{steep}} = \frac{1}{2} \left\{ 1 + \tanh\left(\frac{S_{\text{center}} - |\mathbf{S}|}{S_{\text{width}}}\right) \right\}.$$
(9.36)

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 9.5). This tapering method is valid when ISOTAPER option is chosen and used with the other factor $f_{surface}$ introduced in the subsequent section.

9.3.2 Surface tapering

The vertical displacement of a water parcel due to mesoscale eddy stirring (D) is approximately calculated as follows:

$$D = R|\mathbf{S}|,\tag{9.37}$$

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 ISOTAPER option enables the use of the following tapering scheme proposed by Large et al. (1997):

$$f_{\text{surface}} = \frac{1}{2} \left\{ 1 + \sin \pi \left(r - \frac{1}{2} \right) \right\},\tag{9.38}$$

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 - 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 9.5).

By default, the horizontal diagonal terms of the isopycnal diffusion tensor (9.20) are not multiplied by the two factors noted above. This means that the isopycnal diffusion is rendered the horizontal diffusion within the boundary layer and in the steep slope region. One may apply these tapering methods to the horizontal diagonal terms by setting 1_apply_hdiag = .true. in namelist nml_tracer_diff_isopy_taper (Table 9.5).

9.3.3 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 9.3. We can use different slope maximal limits for isopycnal diffusion and GM parameterization that are specified using the namelist listed on Table 9.4. Configurations of surface tapering for the isopycnal diffusion scheme with ISOTAPER option are specified using the namelist listed on Table 9.5.

variable name	units	description	usage		
diff_isopy_cm2ps	$\rm cm^2sec^{-1}$	isopycnal diffusion coefficient	if ISOPYCNAL		
diff_diapy_cm2ps	$\rm cm^2sec^{-1}$	diapycnal diffusion coefficient	if ISOPYCNAL		
diff_thick_cm2ps	$\rm cm^2sec^{-1}$	coefficient of GM parameterization	if ISOPYCNAL		

Table9.3 Namelist nml_tracer_diff_isopy

variable name	units	description	usage
<pre>slope_clip_iso</pre>	1	maximum slope of isopycnal surface for isopy-	if ISOPYCNAL
		cnal diffusion	
<pre>slope_clip_gm</pre>	1	maximum slope of isopycnal surface for GM parameterization	if ISOPYCNAL

Table9.5 Namelist nml_tracer_diff_isopy_taper

variable name	units	description	usage
l_apply_hdiag	logical	apply tapering factors to horizontal diago-	if ISOTAPER
		nal terms (default = .false.)	
center_transition	1	center of the transition region of tapering	if ISOTAPER
		with hyperbolic tangent (default = 0.005)	
width_transition	1	width of the transition region of tapering	if ISOTAPER
		with hyperbolic tangent (default = 0.001)	
ai_min	$\rm cm^2sec^{-1}$	lower limit of horizontal isopycnal diffu-	if ISOTAPER
		sion coefficient when l_apply_hdiag =	
		.true. (default = 0.0)	
upper_level_isotaper_m	m	BLD for the sine taper (default = mixed	if ISOTAPER
		layer depth (-1.0))	
l_explicit_vdif	logical	handle the vertical diffusion term explicitly	if ISOTAPER
		(default = .false.)	

9.4 Gent and McWilliams parameterization for eddy-induced transport

9.4.1 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:

$$R(T) = \partial_m (J^{mn} \partial_n T) \tag{9.39}$$

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

Chapter 9 SGS parameterization of lateral mixing of tracers

skew flux caused by the anti-symmetric component $F_{\text{skew}}^m = -A^{mn}\partial_n T$ is as follows:

$$R_A(T) = \partial_m (A^{mn} \partial_n T)$$

= $\partial_m (A^{mn}) \partial_n T$
= $\partial_n (\partial_m A^{mn} T),$ (9.40)

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 $\mathbf{F}_{adv} = \mathbf{u}_* T$ 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

$$\mathbf{u}_* = -\frac{\partial}{\partial\rho} (\kappa_{\rm GM} \nabla_\rho h) \left| \frac{\partial h}{\partial\rho}, \right.$$
(9.41)

where h is the depth of the neutral surface ($\rho = \text{const}$). This velocity is expressed in the depth coordinate as

$$\mathbf{u}_* = \left(-\partial_z(\kappa_{\rm GM}S_x), \ -\partial_z(\kappa_{\rm GM}S_y), \ \nabla_h \cdot (\kappa_{\rm GM}\mathbf{S})\right)$$
(9.42)

where

$$\mathbf{S} = (S_x, S_y, 0) = \left(-\rho_x / \rho_z, -\rho_y / \rho_z, 0\right).$$
(9.43)

(Gent et al., 1995).

Griffies (1998) showed that the tendency of a tracer due to this parameterization might be expressed using an antisymmetric diffusion tensor A

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & -\kappa_{\rm GM} S_x \\ 0 & 0 & -\kappa_{\rm GM} S_y \\ \kappa_{\rm GM} S_x & \kappa_{\rm GM} S_y & 0 \end{pmatrix},$$
(9.44)

so that

$$\frac{\partial T}{\partial t} = \cdots + \nabla \cdot (\mathbf{A} \nabla T). \tag{9.45}$$

The flux due to advection can be expressed using a vector streamfunction,

$$\Psi = \kappa_{\rm GM} \hat{\mathbf{z}} \times \mathbf{S} = (-\kappa_{\rm GM} S_{\rm y}, \ \kappa_{\rm GM} S_{\rm x}, \ 0), \tag{9.46}$$

which produces \mathbf{u}_* in (9.42):

$$\mathbf{F}_{\mathrm{adv}} = \mathbf{u}_* T = T(\nabla \times \Psi)$$

The skew diffusive expression for the flux using tensor A in (9.44) 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).

9.4.2 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, $[\kappa_{GM}]_{ref}$, is given by diff_thick_cm2ps in namelist nml_tracer_diff_isopy (Table 9.3). 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 9.8).

a. Simple scheme

If l_grid_size_change_mix_coefs = .true. in namelist nml_grid_size_change_mix_coefs (Table 9.7), the coefficient may be dependent on the horizontal grid size according to the following formula

$$\kappa_{\rm GM} = [\kappa_{\rm GM}]_{\rm ref} \times \min(\Delta x, \Delta y) / (100 \,\rm km), \tag{9.47}$$

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), l_visbeck = .true. in namelist nml_gmvar_select. Visbeck et al. (1997) proposed to give the GM coefficient κ_{GM} as

$$\kappa_{\rm GM} = \alpha \frac{M^2}{N} l^2, \tag{9.48}$$

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,

$$M^{2} = \frac{g}{\rho_{0}D} \left[\left(\int_{-H_{1}}^{-H_{0}} \frac{\partial \rho}{\partial x} dz \right)^{2} + \left(\int_{-H_{1}}^{-H_{0}} \frac{\partial \rho}{\partial y} dz \right)^{2} \right]^{\frac{1}{2}}, \tag{9.49}$$

and

$$N^{2} = \frac{g \left[\sigma_{0}(H_{1}) - \sigma_{0}(H_{0})\right]}{\rho_{0} D},$$
(9.50)

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 $N^2 \ge 10^{-9} \text{ s}^{-1}$.

Using the following formula for the phase speed of the 1st baroclinic mode gravity wave (Sueyoshi and Yasuda, 2009)

$$c_1 = \frac{1}{\pi} \int_{-H_B}^0 \left(-\frac{g}{\rho_0} \frac{\partial \sigma_0}{\partial z} \right)^{\frac{1}{2}} dz, \tag{9.51}$$

where H_B is the depth of sea floor, deformation radius λ_1 is calculated as follows:

$$\lambda_1 = \min\left(\left|\frac{c_1}{f}\right|, 4 \times 10^4\right) \quad \text{m.}$$
(9.52)

Using a factor r = 7, GM coefficient is determined as follows:

$$\kappa_{\rm GM} = \alpha \frac{M^2}{N} (r\lambda_1)^2 \,. \tag{9.53}$$

Lower and upper limits for the coefficient are set as follows:

$$300 \le \kappa_{\rm GM} \le 1500 \quad {\rm m}^2 \,{\rm s}^{-1}.$$
 (9.54)

c. Eden and Greatbatch (2008)

To use the method proposed by Eden and Greatbatch (2008), 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

$$\kappa_{\rm GM} = c L^2 \sigma \tag{9.55}$$

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 recent theoretical considerations (Theiss, 2004). L_{Rhi} is estimated from variables of the coarse resolution model as

$$L_{Rhi} = \frac{\sigma}{\beta} \tag{9.56}$$

(Eden and Greatbatch, 2008), while L_r is given by

$$L_r = \min\left[\frac{c_1}{|f|}, \sqrt{\frac{c_1}{2\beta}}\right],\tag{9.57}$$

Chapter 9 SGS parameterization of lateral mixing of tracers

where c_1 denotes the 1st baroclinic gravity wave speed calculated approximately as eq. (9.51). Considering the thermal wind relation in mid-latitudes, Eden and Greatbatch (2008) proposed that the inverse eddy time scale σ is given by

$$\sigma = f(\operatorname{Ri} + \gamma)^{-\frac{1}{2}}.$$
(9.58)

Here, $\operatorname{Ri} = N^2 |\frac{\partial}{\partial z} u_h|^{-2}$ denotes the local Richardson number. $\gamma(>0)$ is introduced to prevent the singularity as $N \to 0$, which acts effectively as an upper limit for σ and consequently for κ_{GM} . The default values of γ and *c* in eq. (9.55), 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

$$\kappa_{\rm GM} = \Big[\frac{N^2}{N_{\rm ref}^2}\Big][\kappa_{\rm GM}]_{\rm ref},\tag{9.59}$$

where N is the local buoyancy frequency and $[\kappa_{GM}]_{ref}$ 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_{ref}^2 is set to 1 for all shallower depths. Between the depth at which $N^2 = N_{ref}^2$ and the ocean bottom, we also ensure that

$$N_{\min} \le \frac{N^2}{N_{\text{ref}}^2} \le 1.0,$$
 (9.60)

where N_{\min} is the lower limit specified by the user (ratio_bvf_min in namelist nml_gmvar_danabasoglu).

9.4.3 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,

$$\kappa_{\rm GM}(z) = \kappa_{\rm GM}(z = -\rm{MLD}) \times (-z/\rm{MLD}) \quad \text{for} \quad -\rm{MLD} \le z \le 0, \tag{9.61}$$

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 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.

9.4 Gent and McWilliams parameterization for eddy-induced transport

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 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 (|**S**|):

$$D = R|\mathbf{S}|. \tag{9.62}$$

Thus we calculate D at each grid point and the DLD is obtained as follows:

$$DLD = BLD + D. \tag{9.63}$$

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:

$$\Psi_{\rm BL} = \frac{\eta - z}{\eta + BLD} \Psi_0 \quad \text{for} \quad -BLD \le z \le \eta \tag{9.64}$$

and

$$\Psi_{\rm TL} = \left(\frac{z + BLD}{TLT}\right)^2 \Phi + \left(\frac{\eta - z}{\eta + BLD}\right) \Psi_0 \quad \text{for} \quad -DLD \le z < -BLD \tag{9.65}$$

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

$$\Psi_0 = \frac{\eta + \text{BLD}}{2(\eta + \text{BLD}) + \text{TLT}} (2\Psi_I + \text{TLT}\partial_z \Psi_I)$$
(9.66)

and

$$\Phi = -\frac{\text{TLT}}{2(\eta + \text{BLD}) + \text{TLT}}(\Psi_I + (\eta + \text{DLD})\partial_z \Psi_I), \qquad (9.67)$$

where Ψ_I is the interior eddy-induced streamfunction at the base of the transition layer given by the Gent-McWilliams parameterization,

$$\Psi_I = -\kappa_{\rm GM} \frac{\mathbf{z} \times \nabla_H \rho}{\partial_z \rho} \quad \text{at} \quad z = -\text{DLD}.$$
(9.68)

In the implementation, to evaluate both Ψ_I and $\partial_z \Psi_I$ at z = -DLD, Ψ_I are evaluated at the vertical grid points that straddle z = -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.

9.4.4 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:

$$\mathbf{K}_{2} = \begin{pmatrix} L & M \\ M & N \end{pmatrix} = \begin{pmatrix} \kappa_{A}n_{x}^{2} + \kappa_{B}n_{y}^{2} & \kappa_{B}n_{x}n_{y} \\ \kappa_{B}n_{x}n_{y} & \kappa_{B}n_{x}^{2} + \kappa_{A}n_{y}^{2} \end{pmatrix},$$
(9.69)

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 antisymmetric tensor in the Gent-McWilliams scheme, and the following expression is obtained (Smith and Gent, 2004),

$$\mathbf{A}' = \begin{pmatrix} 0 & 0 & -LS_x - MS_y \\ 0 & 0 & -MS_x - NS_y \\ LS_x + MS_y & MS_x + NS_y & 0 \end{pmatrix}.$$
 (9.70)

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.

9.4.5 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 9.6

Table9.6	List of model	options related	d to GM parameterizatio	n.
----------	---------------	-----------------	-------------------------	----

option name	description	usage
GMVAR	Coefficient of GM parameterization is allowed to vary	specify nml_gmvar
SLIMIT	Linearly reduce the coefficient of GM parameterization	cannot be used with GMTAPER
	from the bottom of the mixed layer to the sea surface	
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	cannot be used with TRCBIHARM
	density and velocity (Hirabara et al., 2010)	

b. Spatial dependency

The diffusion coefficient of GM parameterization may be grid size dependent by using the namelist listed on Table 9.7.

Table0.7	Namelist nm]	arid	ci 70	change	miv	coofc
Table9.7	Namenst nm1	_gria_	_size_	_cnange_	_mix_	_coers

variable name	units	description	usage
l_grid_size_change_mix_coefs	logical	the given coefficient is multiplied by the	default = .false.
		fraction of the local grid size to 100 km.	

Overall behavior of GM parameterization with GMVAR option should be specified by using the namelists listed on Tables 9.8 through 9.12.

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

Table9.9 Namelist nml_gmvar_vi	sbeck 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	$cm^2 sec^{-1}$	upper limit of thickness diffusivity	l_visbeck = .true.
lower_limit	$cm^2 sec^{-1}$	lower limit of thickness diffusivity	l_visbeck = .true.
lcalc_defrad	logical	flag whether deformation radius is calculated or	l_visbeck = .true.
		not	
defrad_const	cm	upper limit of deformation radius when	l_visbeck = .true.
		<pre>lcalc_defrad = .true.</pre>	
		constant horizontal length scale when	
		<pre>lcalc_defrad = .false.</pre>	
length_factor	1	[horizontal length scale] = [deformation radius] ×	l_visbeck = .true.
		length_factor	

9.4 Gent and McWilliams parameterization for eddy-induced transport

variable name	units	description	usage
c_EG	1	In this parameterization, Thickness diffusivity is	l_eden = .true.
		parameterized as $c_EG \times L^2 \times sigma_EG$.	
gamma_EG	1	$sigma_EG = f/(Ri + gamma_EG)$	l_eden = .true.
upper_limit	$\rm cm^2sec^{-1}$	upper limit of thickness diffusivity	l_eden = .true.
lower_limit	$\rm cm^2sec^{-1}$	lower limit of thickness diffusivity	l_eden = .true.

Table9.10 Namelist nml_gmvar_eden for GMVAR

Table9.11 Namelist nml_gmvar_danabasoglu for GMVAR

variable name	units	description	usage
ratio_bvf_min	1	Lower bound for the squared buoyancy frequency	l_danabasoglu = .true.
		relative to the reference value (default = 0.1)	
ratio_bvf_max	1	Upper bound for the squared buoyancy frequency	l_danabasoglu = .true.
		relative to the reference value (default = 1.0)	

Table9.12 Namelist nml_gm_transition

variable name	units	description	usage		
l_transition_layer	logical	include transition layer into diabatic	effective when l_danabasoglu =		
		layer (default = .false.)	.true.		

c. Anisotropic scheme

Behavior of GMANISOTROP option should be specified using the namelist listed on Table 9.13.

Table9.13 Namelist nml_gm_anisotrop

variable name	units	description	usage	
cscl_isotrop	1	factor for anisotropy in GM diffusivity	if GMANISOTROP	

Chapter 10

SGS parameterization of vertical mixing of tracers

This chapter explains subgrid-scale parameterizations of vertical mixing of tracers.

10.1 Vertical diffusion

Vertical diffusion term takes the form of Laplacian and the vertical diffusion flux is proportional to the vertical gradient of tracer. The finite difference form is as follows:

$$FZD_{i,j,k} = -\kappa_z \left(\operatorname{areat}\right)_{i,i,k+\frac{1}{2}} \delta_z T_{i,j,k},\tag{10.1}$$

where the use of $(areat)_{i,j,k+\frac{1}{2}}$ implies that the flux occurs only through the oceanic part of the grid interface and

$$\delta_z T_{i,j,k} \equiv \frac{T_{i,j,k-\frac{1}{2}} - T_{i,j,k+\frac{1}{2}}}{\Delta z_k}.$$
(10.2)

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 19.5) because high diffusivity is expected owing to the choice of parameterizations needed for realistic simulations. Otherwise, a forward scheme is used.

10.1.1 Specification of coefficient

Background vertical diffusivity, which is horizontally uniform, a function of depth, and fixed in time, should be always given. Additionally, a three dimensional distribution can be set by selecting VMBG3D option to incorporate locally enhanced mixing processes 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 10.1 and 10.2 summarizes how to give background vertical diffusivity.

Table10.1 Namelist nml_diff_vert_bg. Specify only one of the two variables

variable name	units	description	usage
diff_vert_bg_cm2ps	$cm^2 s^{-1}$	vertically uniform value of back-	Usable only if a vertically uniform
		ground vertical diffusivity	value is intended
file_diff_vert_1d_cm2ps		file having vertical 1D distribution	cannot be specified with the above

variable name	units	description	usage
file_vmix_3d		file having 3D distribution	
imvm		east-west data size	
jmvm		north-south data size	

Table10.2 Namelist nml_vmbg3d. Specify when VMBG3D is selected

Continued on next page

variable name	units	description	usage
l_vmintpol	logical	interpolate input data to model grid points	
slatvm		latitude of the western end	only if l_vmintpol = .true.
slonvm		longitude of the southern end	only if 1_vmintpol = .true.
dlatvm		uniform grid spacing in the meridional direction	only if 1_vmintpol = .true.
dlonvm		uniform grid spacing in the zonal direction	only if 1_vmintpol = .true.

Table 10.2 – continued from previous page

In addition to the static background profiles, the following processes give varying vertical diffusivity coefficients at every model time step.

- Surface mixed layer models (TURBULENCE option).
- Vertical component of isopycnal diffusion (ISOPYCNAL option).
- Enhanced diffusivity (= $1.0 \text{ m}^2 \text{ s}^{-1} = 10^4 \text{ cm}^2 \text{ s}^{-1}$) where the stratification is unstable (DIFAJS option).
- Enhanced diffusivity around rivermouths to avoid too low salinity if river run-off is received by the model (RUNOFF option). This scheme is especially needed when positive definiteness is not guaranteed by a tracer advection algorithm. See Table 10.3 for how to specify the mixing.

The vertical diffusion for "this" time step is taken as the largest of the above estimations.

variable name	units	description	usage
l_enhance_vmix_rivmouth	logical	diffusivity is enlarged around the	default = .false.
		rivermouth	
diff_max_vmixriv_cm2ps	$\rm cm^2sec^{-1}$	maximum value of the enlarged dif-	default = $1 \times 10^4 \text{ cm}^2 \text{ sec}^{-1}$
		fusivity (= κ_{rivmax})	
<pre>depth_max_vmixriv_cm</pre>	cm	vertical diffusion is enlarged from	default = 30×10^2 cm
		surface to this depth, this is also used	
		by subroutine salinity_limit	
para_vmixriv_1	1	parameter for the enlarged vertical	default = 10
		diffusion formula (= a). See below.	
para_vmixriv_2	1	parameter for the enlarged verti-	default = 7
		cal diffusion formula (= b). En-	
		larged diffusion is calculated as	
		$\kappa_{\rm riv} = \min(a^{\log_{10} W_{\rm river}+b}, \kappa_{\rm rivmax}),$	
		where W_{river} is river discharge rate	
		in cm sec ^{-1}	

10.2 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 smooths the distribution.

In general, there are three numerical schemes for convective adjustment. 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. 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 $10\,000\,\mathrm{cm}^2\,\mathrm{s}^{-1}$.

Chapter 10 SGS parameterization of vertical mixing of tracers

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).

10.2.1 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 10.1). 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.

$$VU_{1} = \alpha_{1}V_{\frac{1}{2}},$$

$$VU_{2} = \alpha_{2}(V_{1+\frac{1}{2}} + \alpha_{1}V_{\frac{1}{2}}) = \alpha_{2}(V_{1+\frac{1}{2}} + VU_{1}),$$

$$\cdots,$$

$$VU_{k} = \alpha_{k}(V_{k-\frac{1}{2}} + VU_{k-1}),$$

$$\cdots,$$

$$VU_{KM-1} = \alpha_{KM-1}(V_{KM-1-\frac{1}{2}} + VU_{KM-2}),$$
(10.3)

and

$$VL_{KM-1} = \alpha_{KM-1}V_{KM-\frac{1}{2}},$$

$$VL_{KM-2} = \alpha_{KM-2}(V_{KM-1-\frac{1}{2}} + \alpha_{KM-1}V_{KM-\frac{1}{2}}) = \alpha_{KM-2}(V_{KM-1-\frac{1}{2}} + VL_{KM-1}),$$

$$\cdots,$$

$$VL_{k} = \alpha_{k}(V_{k+\frac{1}{2}} + VL_{k+1}),$$

$$\cdots,$$

$$VL_{1} = \alpha_{1}(V_{1+\frac{1}{2}} + VL_{2}),$$
(10.4)

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:

$$TU_{1} = \alpha_{1}T_{\frac{1}{2}}V_{\frac{1}{2}}, \quad TU_{k} = \alpha_{k}(T_{k-\frac{1}{2}}V_{k-\frac{1}{2}} + TU_{k-1}),$$

$$SU_{1} = \alpha_{1}S_{\frac{1}{2}}V_{\frac{1}{2}}, \quad SU_{k} = \alpha_{k}(S_{k-\frac{1}{2}}V_{k-\frac{1}{2}} + SU_{k-1}),$$

$$TL_{KM-1} = \alpha_{KM-1}T_{KM-\frac{1}{2}}V_{KM-\frac{1}{2}}, \quad TL_{k} = \alpha_{k}(T_{k+\frac{1}{2}}V_{k+\frac{1}{2}} + TL_{k+1}),$$

$$SL_{KM-1} = \alpha_{KM-1}S_{KM-\frac{1}{2}}V_{KM-\frac{1}{2}}, \quad SL_{k} = \alpha_{k}(S_{k+\frac{1}{2}}V_{k+\frac{1}{2}} + SL_{k+1}),$$
(10.5)

where $T_{k+\frac{1}{2}}$ and $S_{k+\frac{1}{2}}$ are temperature and salinity at the level $k + \frac{1}{2}$.

According to this definition, if $\alpha_k = 1$ and elsewhere 0, we get

$$VU_{k} + VL_{k} = V_{k-\frac{1}{2}} + V_{k+\frac{1}{2}},$$

$$TU_{k} + TL_{k} = T_{k-\frac{1}{2}}V_{k-\frac{1}{2}} + T_{k+\frac{1}{2}}V_{k+\frac{1}{2}}$$

$$SU_{k} + SL_{k} = S_{k-\frac{1}{2}}V_{k-\frac{1}{2}} + S_{k+\frac{1}{2}}V_{k+\frac{1}{2}}$$



Figure10.1 Reference vertical grid points in Section 10.2

indicating a volume and accumulated temperature and salinity in an unstable part and

$$TM_{k-\frac{1}{2},k+\frac{1}{2}} = \frac{TU_k + TL_k}{VU_k + VL_k},$$

$$SM_{k-\frac{1}{2},k+\frac{1}{2}} = \frac{SU_k + SL_k}{VU_k + VL_k},$$
(10.6)

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 $\alpha_{k-2} = \alpha_{k+1} = 0$,

$$VU_{k-1} + VL_{k-1} = VU_k + VL_k$$

$$= V_{k-1-\frac{1}{2}} + V_{k-\frac{1}{2}} + V_{k+\frac{1}{2}},$$

$$TU_{k-1} + TL_{k-1} = TU_k + TL_k$$

$$= T_{k-1-\frac{1}{2}}V_{k-1-\frac{1}{2}} + T_{k-\frac{1}{2}}V_{k-\frac{1}{2}} + T_{k+\frac{1}{2}}V_{k+\frac{1}{2}},$$

$$SU_{k-1} + SL_{k-1} = SU_k + SL_k$$

$$= S_{k-1-\frac{1}{2}}V_{k-1-\frac{1}{2}} + S_{k-\frac{1}{2}}V_{k-\frac{1}{2}} + S_{k+\frac{1}{2}}V_{k+\frac{1}{2}},$$

(10.7)

and

$$TM_{k-1-\frac{1}{2},k+\frac{1}{2}} = \frac{TU_{k-1} + TL_{k-1}}{VU_{k-1} + VL_{k-1}} = \frac{TU_k + TL_k}{VU_k + VL_k},$$

$$SM_{k-1-\frac{1}{2},k+\frac{1}{2}} = \frac{SU_{k-1} + SL_{k-1}}{VU_{k-1} + VL_{k-1}} = \frac{SU_k + SL_k}{VU_k + VL_k}.$$
(10.8)

These are averages of the three layer, $k - 1 - \frac{1}{2}$, $k - \frac{1}{2}$, and $k + \frac{1}{2}$.

10.2.2 Numerical procedure

In summary, numerical procedures are summarized as follows.

[1] Density is calculated at the intermediate depth between adjacent levels using a pair of temperature and salinity and is judged to be statically stable or unstable. If an instability occurs, $\alpha(\alpha^1)$ 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 (10.3) to (10.5), 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 (10.6) 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:

$$\beta_k^n = \alpha_k^n + \beta_k^{n-1} (1 - \alpha_k^n).$$
(10.9)

[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.

$$\gamma_{1} = \alpha_{1}^{(n+1)}, \quad \gamma_{k} = \alpha_{k}^{(n+1)} + (1 - \alpha_{k}^{(n+1)})\beta_{k}^{(n)}\gamma_{k-1}$$
$$\alpha_{KM-1}^{(n+1)} = \gamma_{KM-1}, \quad \alpha_{k}^{(n+1)} = \gamma_{k} + (1 - \gamma_{k})\beta_{k}^{(n)}\alpha_{k+1}^{(n+1)}, \quad (10.10)$$

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 10.4 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 (10.10), as described in [6]. Note that $\beta_k^0 = 0$, though there is no description in the table.

radie 10.4 Example of the convective adjustment procedure								
n	k	α		VU	VL	VU+VL	TU+TL	β
1	1 2 3 4 5	1 1 0 0 1	1 1 0 0 1		$\begin{array}{c} \mathbf{V}_{1\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} \\ \mathbf{V}_{2\frac{1}{2}} \\ 0 \\ 0 \\ \mathbf{V}_{5\frac{1}{2}} \end{array}$	$ \begin{array}{c} \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}} \\ 0 \\ 0 \\ \mathbf{V}_{4\frac{1}{2}} + \mathbf{V}_{5\frac{1}{2}} \end{array} $	$ \begin{array}{c} \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}} \\ \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}} \\ 0 \\ 0 \\ \mathbf{T}_{4\frac{1}{2}}\mathbf{V}_{4\frac{1}{2}} + \mathbf{T}_{5\frac{1}{2}}\mathbf{V}_{5\frac{1}{2}} \end{array} $	1 1 0 0 1
2	1 2 3 4 5	0 0 1 0 0	1 1 1 0 0		$+ \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} \mathbf{V}_{\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} \mathbf{V}_{\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} \mathbf{V}_{\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} \mathbf{V}_{3\frac{1}{2}} \mathbf{V}_{\frac{1}{2}} + 0 \\ 0 0 0 0 $	$\begin{aligned} \mathbf{V}_{1\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} \\ \mathbf{V}_{1\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} \\ \mathbf{V}_{1\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} \\ 0 \\ 0 \end{aligned}$	$\sum_{k=0}^{3} \mathbf{T}_{k+\frac{1}{2}} \mathbf{V}_{k+\frac{1}{2}}$ $\sum_{k=0}^{3} \mathbf{T}_{k+\frac{1}{2}} \mathbf{V}_{k+\frac{1}{2}}$ $\sum_{k=0}^{3} \mathbf{T}_{k+\frac{1}{2}} \mathbf{V}_{k+\frac{1}{2}}$ 0 0	1 1 1 0 1
3	1 2 3 4 5	0 0 0 1 0	1 1 1 1		$+ \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} + \mathbf{V}_{4\frac{1}{2}} + \mathbf{V}_{4\frac{1}{2}} + \mathbf{V}_{4\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} + \mathbf{V}_{4\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} + \mathbf{V}_{4\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} - \mathbf{V}_{4\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} + \mathbf{V}_{4\frac{1}{2}} + \mathbf{V}_{2\frac{1}{2}} + \mathbf{V}_{3\frac{1}{2}} + \mathbf{V}_{4\frac{1}{2}} + \mathbf{V}_{4\frac$	+ $\mathbf{V}_{5\frac{1}{2}}$ $\sum_{k=0}^{5} \mathbf{V}_{k+\frac{1}{2}}$ + $\mathbf{V}_{5\frac{1}{2}}$ $\sum_{k=0}^{5} \mathbf{V}_{k+\frac{1}{2}}$ + $\mathbf{V}_{5\frac{1}{2}}$ $\sum_{k=0}^{5} \mathbf{V}_{k+\frac{1}{2}}$ + $\mathbf{V}_{5\frac{1}{2}}$ $\sum_{k=0}^{5} \mathbf{V}_{k+\frac{1}{2}}$ $\mathbf{V}_{5\frac{1}{2}}$ $\sum_{k=0}^{5} \mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5} \mathbf{T}_{k+\frac{1}{2}} \mathbf{V}_{k+\frac{1}{2}}$	1 1 1 1

 Table10.4
 Example of the convective adjustment procedure

Chapter 11

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.

11.1 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 (pCO_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 pCO_2 between the atmosphere and ocean determines uptake or release of CO_2 from the ocean 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 these cases, 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., 2016). 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 mol/m^3 for the units of nutrients. When the coefficients of their model are applied, they should be converted to the corresponding units.

11.2 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, PO₄, and O₂) are used.

$$\frac{\partial \text{DIC}}{\partial t} = -\mathcal{A}(\text{DIC}) + \mathcal{D}(\text{DIC}) + S_b(\text{DIC}) + J_v(\text{DIC}) + J_g(\text{DIC}), \tag{11.1}$$

$$\frac{\partial Alk}{\partial t} = -\mathcal{A}(Alk) + \mathcal{D}(Alk) + S_b(Alk) + J_v(Alk), \qquad (11.2)$$

$$\frac{\partial[\mathrm{PO}_4]}{\partial t} = -\mathcal{A}([\mathrm{PO}_4]) + \mathcal{D}([\mathrm{PO}_4]) + S_b([\mathrm{PO}_4]), \tag{11.3}$$

$$\frac{\partial[O_2]}{\partial t} = -\mathcal{A}([O_2]) + \mathcal{D}([O_2]) + S_b([O_2]) + J_g([O_2]), \tag{11.4}$$

$$\frac{\partial[\mathrm{NO}_3]}{\partial t} = -\mathcal{A}([\mathrm{NO}_3]) + \mathcal{D}([\mathrm{NO}_3]) + S_b([\mathrm{NO}_3]), \tag{11.5}$$

$$\frac{\partial [\text{PhyPl}]}{\partial t} = -\mathcal{A}([\text{PhyPl}]) + \mathcal{D}([\text{PhyPl}]) + S_b([\text{PhyPl}]), \qquad (11.6)$$

$$\frac{\partial [\text{ZooPl}]}{\partial t} = -\mathcal{A}([\text{ZooPl}]) + \mathcal{D}([\text{ZooPl}]) + S_b([\text{ZooPl}]),$$
(11.7)

$$\frac{\partial [\text{Detri}]}{\partial t} = -\mathcal{A}([\text{Detri}]) + \mathcal{D}([\text{Detri}]) + S_b([\text{Detri}]), \qquad (11.8)$$

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 mol/m³ of the substance within them. The terms represented by $J_g()$ and $J_v()$ are the air-sea gas fluxes (Section 11.3.1) and the dilution and concentration effects of evaporation and precipitation on DIC and Alk (Section 11.3.2), 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.

11.3 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. (2016). 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.

11.3.1 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 $J_g(DIC)$ and $J_g([O_2])$ appear only in the uppermost layer. When these fluxes are expressed as $F_g(DIC)$ and $F_g([O_2])$, $J_g(DIC)$ and $J_g([O_2])$ are given as follows:

$$J_g(\text{DIC}) = \frac{F_g(\text{DIC})}{\Delta z_{\frac{1}{2}}},$$
(11.9)

$$J_g([O_2]) = \frac{F_g([O_2])}{\Delta z_{\frac{1}{2}}},$$
(11.10)

where

$$F_g(\text{DIC}) = K_w^{\text{CO}_2} * ([\text{CO}_2]_{\text{sat}} - [\text{CO}_2]_{\text{surf}}),$$
(11.11)

$$F_g([O_2]) = K_w^{O_2} * ([O_2]_{\text{sat}} - [O_2]_{\text{surf}}),$$
(11.12)

and $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer of the model. Here a standard gas transfer formulation is adopted. $K_w^{CO_2}$ and $K_w^{O_2}$ are their gas transfer velocity, $[CO_2]_{sat}$ and $[O_2]_{sat}$ are their saturation concentrations with respect to the atmosphere, and $[CO_2]_{surf}$, $[O_2]_{surf}$, are their surface concentration.

a. Gas transfer velocity

The coefficients $K_w^{CO_2}$ and $K_w^{O_2}$ are the air-sea gas transfer (piston) velocity and are diagnosed as follows:

$$K_w^{\rm CO_2} = a \left(\frac{Sc^{\rm CO_2}}{660}\right)^{-1/2} U_{10}^2 (1 - f_i), \tag{11.13}$$

$$K_w^{O_2} = a \left(\frac{Sc^{O_2}}{660}\right)^{-1/2} U_{10}^2 (1 - f_i), \qquad (11.14)$$

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 CO₂ and O₂.

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.

$$[A]_{\text{sat}} = K_0 f_A \tag{11.15}$$

$$=K_0 C_f p_A \tag{11.16}$$

$$= K_0 C_f (P_a - p H_2 O) x_A \tag{11.17}$$

$$=\phi_A x_A \tag{11.18}$$

$$\simeq \frac{r_a}{P_a^{\circ}} \phi_A^0 x_A \qquad \text{(with errors less than 0.1\%)} \tag{11.19}$$

$$= \frac{P_a}{P_a^{\circ}} [A]_{\rm sat}^0, \tag{11.20}$$

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), pH_2O is water vapor pressure at saturation, ϕ_A is its solubility function, ϕ_A^0 and $[A]_{sat}^0$ are its solubility function and saturation concentration at the reference pressure.

Orr et al. (2016) explain two approaches to calculate the saturation concentration.

The first approach uses equations (11.19) or (11.20), computing solubility function ϕ_A^0 or saturation concentration $[A]_{sat}^0$ 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 $[O_2]_{sat}^0$ is available. This is a rather conventional approach, and is used in MRI.COM except for carbon.

The other approach uses equations (11.15)-(11.17), computing its partial pressure $p_A \equiv (P_a - pH_2O)x_A$, then multiplying by $K' \equiv K_0C_f$, 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 CO₂ concentrations at the surface

In the *mocsy* routine, $[CO_2]_{surf}$ is diagnosed every step from DIC, Alk, temperature, salinity, $[PO_4]$, and silicate concentration at the surface. After $[CO_2]_{surf}$ is computed, fugacity (f_{CO_2}) and partial pressure (pCO_2) of the ocean surface are computed as

$$f_{\rm CO_2} = [\rm CO_2]_{\rm surf}/K_0,$$
 (11.21)

$$pCO_2 = f_{CO_2}/C_f,$$
 (11.22)

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:

$$DIC = [CO_2] + [H_2CO_3] + [HCO_3^-] + [CO_3^{2-}]$$
(11.23)

$$= [CO_2^*] + [HCO_3^-] + [CO_3^{2-}].$$
(11.24)

In the OMIP protocol, the following equations are solved to obtain $[CO_2^*]$. The equilibrium constants for dissociation reactions are:

$$K_1 = \frac{[\mathrm{H}^+][\mathrm{HCO}_3^-]}{[\mathrm{CO}_2^*]} \qquad K_2 = \frac{[\mathrm{H}^+][\mathrm{CO}_3^{2-}]}{[\mathrm{HCO}_3^-]}, \tag{11.25}$$

$$K_{\rm B} = \frac{[{\rm H}^+][{\rm B}({\rm OH})_4^-]}{[{\rm B}({\rm OH})_3]},$$
(11.26)

$$K_{1P} = \frac{[H^+][H_2PO_4^-]}{[H_3PO_4]} \qquad K_{2P} = \frac{[H^+][HPO_4^{2-}]}{[H_2PO_4^-]} \qquad K_{3P} = \frac{[H^+][PO_4^{3-}]}{[HPO_4^{2-}]},$$
(11.27)

$$K_{\rm Si} = \frac{[\rm H^+][\rm SiO(OH)_3^-]}{[\rm Si(OH)_4]},$$
(11.28)

$$K_W = [H^+][OH^-], (11.29)$$

$$K_{\rm S} = \frac{[{\rm H}^+]_F[{\rm SO}_4^{2-}]}{[{\rm H}\,{\rm SO}_4^{-}]},\tag{11.30}$$

and

$$K_{\rm F} = \frac{[{\rm H}^+]_F [{\rm F}^-]}{[{\rm H}{\rm F}]},\tag{11.31}$$

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:

$$[\mathrm{H}^{+}] = [\mathrm{H}^{+}]_{F} \left(1 + \frac{S_{T}}{K_{\mathrm{S}}} + \frac{F_{T}}{K_{\mathrm{F}}} \right), \tag{11.32}$$

and
$$[\mathrm{H}^+]_T = [\mathrm{H}^+]_F \left(1 + \frac{S_T}{K_S}\right).$$
 (11.33)

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:

$$DIC = [CO_2^*] + [HCO_3^-] + [CO_3^{2-}],$$
(11.34)

$$B_{\rm T} = [{\rm B}({\rm OH})_3] + [{\rm B}({\rm OH})_4^-], \qquad (11.35)$$

$$P_{\rm T} = [{\rm H}_3{\rm PO}_4] + [{\rm H}_2{\rm PO}_4^-] + [{\rm HPO}_4^{2-}] + [{\rm PO}_4^{3-}], \qquad (11.36)$$

$$Si_{\rm T} = [Si(OH)_4] + [Si(OH)_3^-],$$
 (11.37)
Chapter 11 **Biogeochemical model**

$$S_{\rm T} = [{\rm HSO}_4^-] + [{\rm SO}_4^{2-}], \tag{11.38}$$

and

$$F_{\rm T} = [{\rm HF}] + [{\rm F}^-].$$
 (11.39)

Alkalinity used in this calculation is defined as follows:

$$Alk = [HCO_3^-] + 2[CO_3^{2-}] + [B(OH)_4^-] + [OH^-] + [HPO_4^{2-}] + 2[PO_4^{3-}] + [SiO(OH)_3^-]$$
(11.40)
- [H⁺]_F - [HSO_4^-] - [HF] - [H₃PO₄].

These expressions exclude the contribution of 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.

11.3.2 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 unless SFLUXW or SFLUXR option is used. In these options, 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

$$sflux(i, j) = -(P - E) * S(i, j, 1)/\Delta z,$$
 (11.41)

where S(i, j, 1) and Δz are the salinity and thickness of the uppermost layer. Note that the variable 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:

$$J_{v}(\text{DIC}(i, j, 1)) = \text{sflux}(i, j) / S(i, j, 1) * \text{DIC}(i, j, 1),$$
(11.42)

$$J_{\nu}(\text{Alk}(i, j, 1)) = \text{sflux}(i, j) / S(i, j, 1) * \text{Alk}(i, j, 1).$$
(11.43)

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(11.42,11.43). In addition, globally integrated J_{ν} (DIC) and J_{ν} (Alk) are set to 0. In MRI.COM, these modifications are not the default considering the use in regional ocean models.

Obata and Kitamura model 11.4

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, [PO₄], and $[O_2]$ 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 $rL[PO_4]^2/([PO_4] + k)$, where r is a proportional factor ($r = 0.9 \text{ yr}^{-1}$), L is the insolation normalized by the annual mean insolation on the equator, and k is the half-saturation constant ($k = 0.377 \text{ mmol/m}^{-3}$). The values of r and k are adjusted to reproduce the optimum atmospheric CO₂ 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 $P: N: C: O_2 = 1: 16: 106: -138 \equiv 1: R_{np}: R_{cp}: -R_{op}$ where $R_{ab} = A/B$ and "o" represents O₂ (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 $(z/100m)^{-0.9}$ and $\exp(-z/3500m)$ (*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:

$$S_b(\text{DIC}) = R_{cp} * \text{RemiP} + \text{SolnCa} - R_{cp} * \text{ExprodP}$$
 (11.44)

$$S_b(Alk) = 2 * SolnCa + R_{np} * ExprodP - R_{np} * RemiP$$
(11.45)

$$S_b([PO_4]) = \text{RemiP} - \text{ExprodP}$$
(11.46)

$$S_b([O_2]) = -R_{op} * S_b([PO_4])$$
(11.47)

11.5 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 (NO₃), phytoplankton (PhyPl), zooplankton (ZooPl), and detritus (Detri) are normalized in terms of nitrogen 1 mol/m³. For example, [PhyPl] represents the concentration of phytoplankton estimated in terms of nitrogen in one cubic meter (N mol/m³). 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, $S_b(DIC)$ and $S_b(Alk)$, used for calculating the carbon cycle, are described later in this section.

$$S_b([PhyPl]) = Priprod - MortP1 - MortP2 - GrP2Z$$
(11.48)

$$S_b([\text{ZooPl}]) = assim * \text{GrP2Z} - \text{Excrtn} - \text{MortZ}$$
(11.49)

$$S_b([\text{Detri}]) = [(1 - assim) * \text{GrP2Z} + \text{MortP2} + \text{MortZ}] - \text{RemiD} - w_{detri} \frac{\partial \text{Detri}}{\partial z}$$
(11.50)

$$S_b([NO_3]) = MortP1 + Excrtn + RmeiD - PriProd$$
(11.51)

$$S_b([PO_4]) = R_{pn} * S_b([NO_3])$$
(11.52)

$$S_b([O_2]) = -R_{on} * R_{np} * S_b([PO_4])$$
(11.53)

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 $(-w_{detri}\frac{\partial \text{Detri}}{\partial z})$. 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.

11.5.1 Description of each term

Priprod = J(I, N, P) * [PhyPl] Primary production expresses photosynthesis (described in detail in the next subsection).
MortP1 = φ_P * [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 = $\phi_{PP} * [PhyP1]^2$ The conversion from phytoplankton to detritus (normal mortality of phytoplankton).

• GrP2Z = $G(P) * [\text{ZooPl}]^2$ The grazing of zooplankton. There are a number of parameterizations of grazing. In this formulation, this is given as $G(P) = g * \epsilon * [\text{PhyPl}]^2/(g + \epsilon * [\text{PhyPl}]^2)$. Among the grazing, the ratio *assim* is used for the growth of zooplankton, and the remainder (1 - assim) is converted to detritus.

• Excrtn = d * [ZooPl]Excretion of zooplankton. The excretion is dissolute and directly returned to nutrients (NO₃).

Chapter 11 Biogeochemical model

• Mort $Z = \phi_Z * [ZooPl]$

The conversion from the zooplankton to detritus (mortality of zooplankton).

• RemiD = $\phi_D * [Detri]$

Remineralization of detritus. This is converted to nutrients through the activity of bacteria.



Figure11.1 Schematic of NPZD model

11.5.2 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:

$$J(I, N, P) = \min(J_I, J_N, J_P),$$
(11.54)

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:

$$J_{I} = \frac{J_{max} \alpha I}{\left[J_{max}^{2} + (\alpha I)^{2}\right]^{1/2}}.$$
(11.55)

Here, J_{max} is the light-saturated growth, which depends on temperature based on Eppley (1972) as

$$J_{max} = a \cdot b^{c\theta},\tag{11.56}$$

where $a = 0.6 \text{ day}^{-1}$, b = 1.066, and $c = 1 (^{\circ}\text{C})^{-1}$. Note that the default values in MRI.COM are based on Schmittner et al. (2008) and differ from these values (see Table 11.1). Equation (11.55) is called Smith-type growth. The coefficient α in the equation is "the initial slope of photosynthesis versus irradiance (P-I) curve," that is,

$$\alpha = \lim_{I \to 0} \frac{\partial J_I}{\partial I}.$$
(11.57)

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.

$$I = I_{z=0} \operatorname{PAR} \exp\left(-k_{w}\tilde{z} - k_{e} \int_{0}^{\tilde{z}} P dz\right),$$
(11.58)

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

$$J_N = J_{MM} = \frac{J_{max}N}{K_N + N},$$
(11.59)

where K_N is a half-saturation constant for NO₃ uptake. In contrast, the Optimal Uptake (OU) equation for a nitrate is expressed as follows:

$$J_N = J_{OU} = \frac{V_0 N}{N + 2\sqrt{\frac{V_0}{A_0}N} + \frac{V_0}{A_0}},$$
(11.60)

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

$$V_0 = 0.5 \left(1 + \sqrt{\frac{\alpha_{OU}}{K_N}} \right)^2 J_{max}.$$
 (11.61)

Finally, we obtain

$$J_{OU} = \frac{V_0 N}{N + 2\sqrt{\alpha_{OU}N} + \alpha_{OU}}.$$
 (11.62)

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).

11.5.3 Variation of DIC and Alk due to biological activity

Production of DIC and Alk is controlled by changes in inorganic nutrients and calcium carbonate (CaCO₃), in molar numbers according to

$$S_b(\text{DIC}) = S_b([\text{PO}_4])R_{cp} - S_b([\text{CaCO}_3]),$$
 (11.63)

$$S_b(Alk) = -S_b([NO_3]) - 2 \cdot S_b([CaCO_3]).$$
 (11.64)

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 CaCO₃ 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,

$$S_b([CaCO_3]) = Pr([CaCO_3]) - Di([CaCO_3]).$$
 (11.65)

Following Schmittner et al. (2008), the source term ($Pr([CaCO_3])$) of calcium carbonate is determined by the production of detritus as follows:

$$Pr([CaCO_3]) = [(1 - assim) * [GrP2Z] + [MortP2] + [MortZ]] R_{CaCO_3/POC} R_{C:N},$$
(11.66)

where *assim*, GrP2Z, MortP2, and MortZ are as described above. The sink term $(Di([CaCO_3]))$ of calcium carbonate is parameterized as

$$Di([CaCO_3]) = \int Pr([CaCO_3])dz \cdot \frac{d}{dz} \left(\exp\left(-z/D_{CaCO_3}\right)\right), \qquad (11.67)$$

which expresses an instantaneous sinking with an *e*-holding depth of $D_{CaCO_3} = 3500$ 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.

Chapter 11 Biogeochemical model

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 $R_{CaCO_3/POC} = 0.035$ 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.

11.6 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 02 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 11.1. 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 repeatedly writing nmlrs_ptrc. The order of passive tracers is determined in tracer_vars.F90. Following is an example when CARBON, O2, and NPZD options are chosen. In this case, numtrc_p = 8 should be specified in configure.in.

- An example specification of restart files when CARBON, 02, and NPZD options are chosen. -

```
&nmlrs_ptrc fname='result/rs_dic'/
&nmlrs_ptrc fname='result/rs_alk'/
&nmlrs_ptrc fname='result/rs_o2' /
&nmlrs_ptrc fname='result/rs_po4' /
&nmlrs_ptrc fname='result/rs_no3' /
&nmlrs_ptrc fname='result/rs_PhyPl'/
&nmlrs_ptrc fname='result/rs_ZooPl'/
&nmlrs_ptrc fname='result/rs_Detri'/
```

variable name	description	units	default value
alphabio	Initial slope of P-I curve	$(W m^{-2})^{-1} day^{-1}$	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	$m^{-1} (mol m^{-3})^{-1}$	0.03d3
dkwbio	Light attenuation in the water	m^{-1}	0.04d0
rk1bioNO3	Half-saturation constant for NO ₃ uptake	$mol m^{-3}$	0.7d-3
rk1bioP04	Half-saturation constant for PO ₄ uptake	$mol m^{-3}$	0.0d0
alpha_ou	Fitting constant for Optical Uptake kinetics		0.19d0
gbio	Maximum grazing rate	day ⁻¹	1.575d0
epsbio	Prey capture rate	$(mol m^{-3})^{-2} day^{-1}$	1.6d6
phiphy	Specific mortality/recycling rate	s^{-1}	0.014d0
phiphyq	Quadratic mortality rate	$(mol m^{-3})^{-1} day^{-1}$	0.05d3
a_npz	Assimilation efficiency		0.925d0
phizoo	Quadratic mortality of zooplankton	$(mol m^{-3})^{-1} day^{-1}$	0.34d3
d_npz	Excretion	day ⁻¹	0.01d0
remina	Remineralization rate	day ⁻¹	0.048d0
w_detr	Sinking velocity	m day ⁻¹	2.0d0
fac_wdetr	Arbitrary parameter for numerical stability.		3.d0
	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.		
c_mrtn	Dimensionless scaling factor for Martin et al. (1987)		0.858d0
	$Phi(z) = Phi(zo) * (z/dp_mrtn)**(-c_mrtn)$		
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	400.d0
dp_eprdc	The depth where the bio-export is diagnosed.	m	126.d0
	This value should be less than dp_mrtn.		
Rcaco3poc	CaCO ₃ over nonphotosynthetic POC production ratio		0.05d0
Dcaco3	CaCO ₃ remineralization e-folding depth	m	3500.d0
shwv_intv	Interval for calculating the irradiance and light-limited growth	min	10.d0
	rate. This must be a divisor of the time step for tracer.		

Table11.1 Parameters used for the NPZD ecosystem component (NPZD).

11.7 Program structure

```
ogcm__ini
 +-- ptrc_ctl__ini
 +-- ptrc__ini
 1
 L
          |
           +-- cbn__ini
 L
          |
  L
          +-- cbn__ini_history
  Ι
  Τ
 +-- tracer__ctl__ini
      +-- tracer__ini
           +-- bio__ini
```

```
+-- bio__ini_history
ogcm__run
 +-- part_1
 | +-- ptrc_ctl__main
| | |
        +-- ptrc____
| |
| +-- cbn__set_xco2a
| |
| +-- cbn__calc_virtu
| |
| +-- cbn__calc_co2_
 | |
          +-- ptrc__surfflux
    |
 Ι
 |
    +-- ptrc__predict_surf
 1
 |
    +-- cbn__predict_surf
 |
    |
     +-- tracer_ctl__predict
 |
         Ι
          +-- tracer__internal_source
 1
              +-- bio__predict
 1
 +-- hist_ctl
     +-- hist_ctl__write
         +-- cbn__write_history
```

Chapter 12

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.

12.1 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).

12.1.1 Source / Sink term

Source and sink of an age tracer a [year], is expressed as follows. In the oceanic interior,

$$\frac{\partial a}{\partial t} = -\mathcal{A}(a) + \mathcal{D}(a) + \frac{1}{\text{the total number of seconds of this year}} \quad \text{year/sec,}$$
(12.1)

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:

$$a(k=1) = 0. (12.2)$$

12.1.2 Usage

Required Model option IDEALAGE

Required Add one to numtrc_p

Required Restart file for passive tracer (nmlrs_ptrc) (See Section 11.6)

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.

12.2 CFCs

MRI.COM can treat CFCs (CFC11 and CFC12) following the protocols of OMIP (Orr et al., 2016). There is no source and sink for CFCs in the interior.

12.2.1 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,

$$\frac{\partial[A]}{\partial t} = -\mathcal{A}([A]) + \mathcal{D}([A]) + J_g([A]).$$
(12.3)

 $J_g()$ represents source/sink due to the air-sea gas fluxes F([A]) at the sea surface,

$$J_g([A]) = \frac{F([A])}{\Delta z_{\frac{1}{2}}},$$
(12.4)

with

$$F(A) = k_w([A]_{sat} - [A]),$$
(12.5)

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

$$[A]_{sat} = \phi_A^0 x_A, \tag{12.6}$$

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. (2016) for further details.

12.2.2 Usage

```
Required Model option CFC
Required Add two to numtrc_p
Required Restart file for passive tracer (nmlrs_ptrc) (See Section 11.6)
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'
```

12.3 SF₆

The calculation of SF_6 also follows the protocols of OMIP (Orr et al., 2016). The formulation of SF_6 is nearly the same as CFCs. Only the coefficients of the empirical fit differ.

12.3.1 Surface boundary condition

See Section 12.2.1, where CFCs should read SF_6 .

12.3.2 Usage

Required Model option SF6
Required Add one to numtrc_p
Required Restart file (nmlrs_ptrc) (See Section 11.6)
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 SF₆ [ppt]
 name SF6
 txyu 't'

Chapter 13

Package Structure and Usage

13.1 Package structure

Packages relevant to tracers are listed as follows.

13.1.1 Tracer equation

tracer_ctl.F90:	Controller of this package
tracer.F90:	Main program of this package
upc_adv.F90:	Upcurrent advection scheme
quick_adv.F90:	QUICK advection scheme (QUICKADVEC)
utzq_adv.F90:	Combination of UTOPIA and QUICKEST advection scheme (UTZQADVEC)
som_adv.F90:	Second order moment advection scheme (SOMADVEC)
mpdata_adv.F90:	MPDATA advection scheme (MPDATAADVEC)
tracer_vars.F90:	Setting of tracer attributes
+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

vmixcoef_ctl.F90:	Controller of the vertical mixing package
vmixcoef.F90:	Main program of the vertical mixing package
vmixcoef_vmbg.F90:	Estimation of background vertical diffusion coefficient
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

strat_adjust_ctl.F90:	Controller of stratification and adjustment package
stratification.F90:	Main program of calculation of stratification
cnvajs.F90:	Main program of convective adjustment
strat_adjust_vars.F90:	Declaration of variables

13.1.4 Reference state and restoring coefficient

restore_cond_ctl.F90:	Controller of reference state and restoring coefficient
restore_cond.F90:	Main program of reference state and restoring coefficient
force_data.F90:	Service package that handles external forcing data

13.1.5 Passive tracers

ptrc_ctl.F90:	Controller of passive tracer evolution
ptrc.F90:	Main program of passive tracer evolution that mainly treats surface sources and sinks

ptrc_subp.F90:	Sub package of passive tracer evolution that describes internal sources and sinks
	(called from tracer.F90)
+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. Parameters are listed on Table 13.1.

variable name	units	description	usage
l_rst_tracer_in	logical	.true. : Read restart files specified by	Default is the same as 1_rst_in
		nmlrs_t and nmlrs_s for the initial con-	ofnml_run_ini_state.
		dition.	
		.false.: Start condition depends on the	
		l_rst_uni_strati	
l_rst_uni_strati	logical	.false.: Start from 3D-distribution at the	if l_rst_tracer_in = .false.
		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 con-	
		ducted based on start_rec_uni_strati	
		and end_rec_uni_strati.	
<pre>start_rec_uni_strati</pre>	integer	uniform stratification is created by the av-	if l_rst_uni_strati = .true.
end_rec_uni_strati		erage from start_rec_uni_strati data	
		record to end_rec_uni_strati record.	

13.3 Configuration of tracers

The attributes of each tracer such as name, advection scheme, restoring condition, reference data, and restoring coefficients, are stored in the 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_data.F90

Table13.2 Namelist nml_tracer_data

variable name	units	description	115age		
variable name	unito	description	usuge		
name	character	Name of tracer. Two tracers are necessary:	Case sensitive. For example, "potential		
		"Potential Temperature" and "Salinity."	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)
- "mpdata" : MPDATA advection scheme (MPDATAADVEC)

variable name	units	description	usage
adv_scheme%name	character	Name of the advection	Different advection
		scheme.	schemes can be set
			for individual tracers.
<pre>adv_scheme%limiter_som_org</pre>	logical	Use flux limiter for SOM	SOMADVEC
		by Prather (1986)	
<pre>adv_scheme%limiter_som_Merryfield03</pre>	logical	Use flux limiter for som by	SOMADVEC
		Merryfield and Holloway	
		(2003)	
adv_scheme%lrstin_som	logical	The SOM initial state of	SOMADVEC
		moments is read from file	
adv_scheme%lrstout_som	logical	The SOM final state of mo-	SOMADVEC
		ments is written to file	
<pre>adv_scheme%limiter_mpdata_nonoscillatory</pre>	logical	Apply flux limiter for MP-	MPDATAADVEC
		DATA	
adv_scheme%eps_lim_mpdata	that of	Very small value to avoid	MPDATAADVEC
	tracer	zero division	
adv_scheme%min_value_mpdata	that of	Minimum value for tracer	MPDATAADVEC
	tracer		

Table13.3 Namelist nml_tracer_data related to advection scheme

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.
<pre>restore_conf%l_body_restore</pre>	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	l to reference	values for	body	forcing	and initial	condition

units	description	usage
character	a file name that contains reference	
	values for body forcing and initial	
	condition.	
character	a file name of grid	needed if linterp = .true.
integer	grid size of data in x direction	
integer	grid size of data in y direction	
integer	grid size of data in z direction	
integer	regular time interval of data	positive value : unit is sec
		-1: monthly
		-999 : steady forcing
		Continued on next page
	units character character integer integer integer integer	unitsdescriptioncharactera file name that contains reference values for body forcing and initial condition.charactera file name of gridintegergrid size of data in x directionintegergrid size of data in y directionintegergrid size of data in z directionintegergrid size of data in z directionintegergrid size of data in z directionintegerregular time interval of data

variable name	units	description	usage
<pre>trcref_conf%num_data_max</pre>	integer	the number of record contained in the	
		file	
trcref_conf%ifstart	ifstart(6)	[ymdhms] of the first record of the	1999,1,1,0,0,0 when the first
		input file	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
<pre>trcref_conf%luniform</pre>	integer	data is horizontally uniform or not	default = .false.
<pre>trcref_conf%luniform_v</pre>	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	integer	the input data is defined or not	default = .false.

Table 13.5 – continued from previous page

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.

variable name	units	description	usage
<pre>trcref_surf_conf%file_data</pre>	character	a file name that contains refer-	
		ence values for surface restor-	
		ing forcing	
<pre>trcref_surf_conf%file_data_grid</pre>	character	a file name of grid	needed if linterp = .true.
<pre>trcref_surf_conf%imfrc</pre>	integer	grid size of data in x direction	
<pre>trcref_surf_conf%jmfrc</pre>	integer	grid size of data in y direction	
<pre>trcref_surf_conf%interval</pre>	integer	regular time interval of data	positive value : unit is sec
			-1: monthly
			-999 : steady forcing
<pre>trcref_surf_conf%num_data_max</pre>	integer	the number of record contained	
		in the file	
<pre>trcref_surf_conf%ifstart</pre>	ifstart(6)	[ymdhms] of the first record of	1999,1,1,0,0,0 when the first
		the input file	record is the average value of
			Jan 1999 and its data interval
			is monthly.
trcref_surf_conf%lrepeat	logical	climatological data is repeat-	default = .false.
		edly used	
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	default = .false.
		not	
<pre>trcref_surf_conf%ldouble</pre>	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.

Table13.6	Namelist nml	tracer	data related	to reference	values	for surface	restoring	forcing.

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 for	orcing
140101010.1	"indificition in the state of t	orenng.

variable name	units	description	usage
<pre>rstcoef_conf%file_data</pre>	character	a file name that contains restoring	
		coefficient for body forcing.	
<pre>rstcoef_conf%file_data_grid</pre>	character	a file name of grid	needed if linterp = .true.
<pre>rstcoef_conf%imfrc</pre>	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

variable name	units	description	usage
			-1: monthly
			-999 : steady forcing
<pre>rstcoef_conf%num_data_max</pre>	integer	the number of record contained in	
		the file	
<pre>rstcoef_conf%ifstart</pre>	ifstart(6)	[ymdhms] of the first record of the	1999,1,1,0,0,0 when the first
		input file	record is the average value of
			Jan 1999 and its data interval
			is monthly.
<pre>rstcoef_conf%lrepeat</pre>	logical	climatological data is repeatedly	default = .false.
		used	
<pre>rstcoef_conf%linterp</pre>	logical	interpolate horizontally or not	default = .false.
<pre>rstcoef_conf%linterp_v</pre>	logical	interpolate vertically or not	default = .false.
<pre>rstcoef_conf%ilinear</pre>	integer	interpolation method	1 : linear, 2 : spline
<pre>rstcoef_conf%luniform</pre>	logical	data is horizontally uniform or not	default = .true.
<pre>rstcoef_conf%luniform_v</pre>	logical	data is vertically uniform or not	default = .false.
<pre>rstcoef_conf%ldouble</pre>	logical	input data is double or not	default = .true.
rstcoef_conf%iverbose	logical	standard output of progress	1 : extensive 0 : minimum
<pre>rstcoef_conf%ldefined</pre>	logical	the input data is defined or not	default = .false.

Table 13.7 – continued from previous page

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} .

	1	
units	description	usage
character	a file name that contains	
	restoring coefficient for for	
	surface restoring forcing.	
character	a file name of grid	needed if linterp = .true.
integer	grid size of data in x direction	
integer	grid size of data in y direction	
integer	regular time interval of data	positive value : unit is sec
		-1: monthly
		-999 : steady forcing
	the number of record con-	
	tained in the file	
ifstart(6)	[ymdhms] of the first record	1999,1,1,0,0,0 when the first
	of the input file	record is the average value of
	-	Jan 1999 and its data interval
		is monthly.
logical	climatological data is repeat-	default = .false.
C C	edly used	
logical	interpolate horizontally or	default = .false.
-	not	
integer	interpolation method	1 : linear, 2 : spline
logical	data is horizontally uniform	default = .false.
	or not	
logical	input data is double or not	default = .false.
logical	standard output of progress	1 : extensive, 0 : minimum
logical	the input data is defined or not	default = .false.
	units character integer integer integer integer integer logical logical logical logical logical logical	unitsdescriptioncharactera file name that contains restoring coefficient for for surface restoring forcing.charactera file name of gridintegergrid size of data in x directionintegergrid size of data in y directionintegerregular time interval of dataintegerthe number of record con- tained in the fileifstart(6)[ymdhms] of the first record of the input filelogicalclimatological data is repeat- edly usedlogicalinterpolate horizontally or notintegerinterpolation methodlogicalinput data is double or notlogicalinput data is double or notlogicalthe input data is defined or not

Table 13.8 Namelist nml_tracer_data related to restoring coefficient for surface restoring for	Table13.8
--	-----------

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.,
! 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.

For a simulation using a coupled atmosphere-ocean model, all surface fluxes may be given by the atmospheric component that computes fluxes using the surface states of the oceanic component. Users are referred to Yukimoto et al. (2011) for the treatment of fluxes in MRI-ESM.

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

$$\frac{\partial}{\partial t}(u_1, v_1) = \dots + \frac{(\tau_x, \tau_y)}{\rho_0 \Delta z_{\frac{1}{2}}},\tag{14.1}$$

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:

$$(\tau_x, \tau_y) = (1 - A)(\tau_{AO_x}, \tau_{AO_y}) + A(\tau_{IO_x}, \tau_{IO_y}),$$
(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. For the time level *n* of the leap-frog scheme,

$$\overrightarrow{\tau_{AO}}^{n} = \rho_a C_D |\overrightarrow{U_a}^{n} - \alpha \overrightarrow{U_s}^{n-1}| (\overrightarrow{U_a}^{n} - \alpha \overrightarrow{U_s}^{n-1}),$$
(14.4)

and for the first part of the Matsuno scheme,

$$\overrightarrow{\tau_{AO}}^n = \rho_a C_D |\overrightarrow{U_a}^n - \alpha \overrightarrow{U_s}^n| (\overrightarrow{U_a}^n - \alpha \overrightarrow{U_s}^n).$$
(14.5)

The same treatment should be applied to the computation of ice-ocean stress. The surface velocity at the previous time level should be passed to the sea ice part for the leap-frog scheme.

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.

	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	$\mathrm{cm}\mathrm{s}^{-1}$	if TAUBULK
Y-ward wind speed	V-wind	$\mathrm{cm}\mathrm{s}^{-1}$	if TAUBULK

Table14.1 Wind data to be read from namelist nml_force_data.

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).

variable name	units	description	usage
idays_wind_setup	day	wind set up period in days	default : no tapering on wind

Table14.2	Namelist nml	force	windsetup.

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 b	oulk	wind.

variable name	units	description	usage
current_contri_factor	1	Contribution of the surface current to the	default = 1.0
		calculation of relative surface wind	

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_{WF}^{θ}) 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

$$Q_{NET} = Q_{SH} + Q_{LO} + Q_{LA} + Q_{SN} + Q_{ice} + \rho_0 c_p F_{WF}^{\theta},$$
(14.6)

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 (14.6) are broken into three terms,

$$Q_{NET} = Q_{SH} + Q_{OTHER} + \rho_0 c_p F_{WF}^{\theta}.$$
(14.7)

Shortwave radiation (Q_{SH}) penetrates into the ocean interior. Other terms (Q_{OTHER}) are absorbed in the first layer. Fresh water transport term ($\rho_0 c_p F_{WF}^{\theta}$) 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, SST (unnecessary for LWDOWN option), 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.

elements	name	units	usage
Downward shortwave radiation	ShortWave	$erg s^{-1} cm^{-2} = \times 10^{-3} W m^{-2}$	
Downward longwave radiation	LongWave	$\mathrm{erg}\mathrm{s}^{-1}\mathrm{cm}^{-2}$	Net longwave by default
			(SST is required).
			Downward if LWDOWN
Sea surface temperature	SST	°C	unnecessary if LWDOWN
Surface air temperature	TempAir	°C	
Surface air specific humidity	SphAir	1	
Scalar wind speed	ScalarWind	$\mathrm{cm}\mathrm{s}^{-1}$	unnecessary if TAUBULK
Sea level pressure	SeaLevelPressure	hPa	also available for SLP

Table14.4 Heat flux data to be read from namelist nml_force_data when HFLUX option is selected.

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 $(\alpha_o Q_{SH}^{\text{down}})$ is reflected at the sea surface, and the remainder penetrates into the ocean interior as the net shortwave radiation flux $(Q_{SH}(z = \eta) = (1 - \alpha_o)Q_{SH}^{\text{down}})$, 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.

$$\frac{\partial \theta}{\partial t} = \dots + \frac{1}{\rho_0 c_p} \frac{\partial Q_{SH}(z)}{\partial z},\tag{14.8}$$

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,

$$Q_{SH}(z) = Q_{SH}(0)[R\exp\{(-\eta + z)/\zeta_1\} + (1 - R)\exp\{(-\eta + z)/\zeta_2\}],$$
(14.9)

where we set R = 0.58, $\zeta_1 = 35$ cm, and $\zeta_2 = 2300$ 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 (14.9) is replaced by the penetrating distance from the sea surface, the shortwave radiation is given by

$$Q_{SH}(z) = Q_{SH}(0) [R \exp\{(-\eta + z)/(\zeta_1 \sin \theta_w)\} + (1 - R) \exp\{(-\eta + z)/(\zeta_2 \sin \theta_w)\}],$$
(14.10)

where θ_w is the penetrating angle in the ocean interior. When SOLARANGLE option is specified in coupled models, Equation (14.10) is replaced by

$$Q_{SH}(z) = Q_{SH_b}(z) + Q_{SH_d}(z), \tag{14.11}$$

where $Q_{SH_b}(z)$ and $Q_{SH_d}(z)$ are the shortwave radiation fluxes due to direct and diffuse solar radiation. Those fluxes are expressed as follows:

$$Q_{SH_b}(z) = Q_{SH_b}(0) [R_b \exp\{(-\eta + z)/(\zeta_1 \sin \theta_w)\} + (1 - R_b) \exp\{(-\eta + z)/(\zeta_2 \sin \theta_w)\}],$$
(14.12)

$$Q_{SH_d}(z) = Q_{SH_d}(0) [R_d \exp\{(-\eta + z)/\zeta_1\} + (1 - R_d) \exp\{(-\eta + z)/\zeta_2\}],$$
(14.13)

where R_b and R_d are the ratios of near infra-red to total radiation in direct and diffuse solar radiation, respectively. $Q_{SH_b}(0)$ and $Q_{SH_d}(0)$ are the net direct 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

$$Q_{SH}(z) = Q_{SH}(0)[R\exp\{(-\eta + z)/\zeta_0\} + (1 - R)(V_1\exp\{(-\eta + z)/\zeta_1\} + V_2\exp\{(-\eta + z)/\zeta_2\})].$$
(14.14)

The first exponential is for the infra-red waveband (> 750 nm), which is not influenced by biological materials and attenuates with a distance of $\zeta_0 = 26.7$ 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* [mg · m⁻³]) as follows:

$$V_1 = 0.321 + 0.008C + 0.132C^2 + 0.038C^3 - 0.017C^4 - 0.007C^5,$$
(14.15)

$$V_2 = 0.679 - 0.008C - 0.132C^2 - 0.038C^3 + 0.017C^4 + 0.007C^5,$$
(14.16)

$$\zeta_1 = 1.540 - 0.197C + 0.166C^2 - 0.252C^3 - 0.055C^4 + 0.042C^5, \tag{14.17}$$

$$\zeta_2 = 7.925 - 6.644C + 3.662C^2 - 1.815C^3 - 0.218C^4 + 0.502C^5,$$
(14.18)

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

$$Q_{SH}(z) = Q_{SH}(0)[R \exp\{(-\eta + z)/(\zeta_0 \sin \theta_w)\} + (1 - R)(V_1 \exp\{(-\eta + z)/(\zeta_1 \sin \theta_w)\} + V_2 \exp\{(-\eta + z)/(\zeta_2 \sin \theta_w)\})].$$
(14.19)

In coupled models, Equation (14.19) is replaced by

$$Q_{SH}(z) = Q_{SH_b}(z) + Q_{SH_d}(z),$$
(14.20)

where $Q_{SH_b}(z)$ and $Q_{SH_d}(z)$ are the shortwave radiation fluxes due to direct and diffuse solar radiation. These fluxes are expressed as follows:

$$Q_{SH_b}(z) = Q_{SH_b}(0)[R_b \exp\{(-\eta + z)/(\zeta_0 \sin \theta_w)\} + (1 - R_b)(V_1 \exp\{(-\eta + z)/(\zeta_1 \sin \theta_w)\} + V_2 \exp\{(-\eta + z)/(\zeta_2 \sin \theta_w)\})], \quad (14.21)$$

$$Q_{SH_d}(z) = Q_{SH_d}(0)[R_d \exp\{(-\eta + z)/\zeta_0\} + (1 - R_d)(V_1 \exp\{(-\eta + z)/\zeta_1\} + V_2 \exp\{(-\eta + z)/\zeta_2\})], \quad (14.22)$$

where
$$R_b$$
 and R_d are the ratios of near infra-red to total radiation for direct and diffuse solar radiation, respectively.

d. Albedo schemes

There are three options for the sea surface albedo. The first option (albedo_choice = 1 in namelist nml_albedo_ocean) is a constant value, which should be specified as alb in namelist nml_albedo_ocean. The second option (albedo_choice = 2) is based on Large and Yeager (2009) and is given by

$$\alpha_o = 0.069 - 0.011 \cos(2\phi), \tag{14.23}$$

where ϕ is latitude. The third option (albedo_choice = 3) is based on Baker and Li (1995) and is given by

$$\alpha_o = 0.06 + 0.0421x^2 + 0.128x^3 - 0.04x^4 + \left(\frac{3.12}{5.68 + U} + \frac{0.074x}{1.0 + 3.0U}\right)x^5,$$
(14.24)

where $x = 1 - \sin \theta_a$ (θ_a is a height angle of the Sun), and U is the surface wind speed [m s⁻¹].

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

$$(\theta V)_{i,j,k-\frac{1}{2}}^{n+1} = (\theta V)_{i,j,k-\frac{1}{2}}^{n-1} + \dots + 2\Delta t \Big(Q_{SH_{i,j,k-1}}^n S_{i,j,k-\frac{1}{2}} - Q_{SH_{i,j,k}}^n S_{i,j,k+\frac{1}{2}} \Big),$$
(14.25)

for the leap-frog scheme.

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	
		for albedo_choice = 1	
albedo_choice	no (integer)	choice of albedo scheme	1 : constant given by alb
			2 : Large and Yeager (2009)
			3 : MRI-ESM1

Table14.6 Namelist nml_solarangle.

variable name	units	description	usage
interval_shortwave_sec	sec	time interval of short-	required
		wave data	
l_variable_zenith_ang	logical	use zenith angle with	default = .true.
		daily variation	
l_const_zenith_ang_mean	logical	use average zenith angle	<pre>valid if l_variable_zenith_ang = .false.</pre>
		from sunrise to sunset	
l_const_zenith_ang_noon	logical	use zenith angle at noon	<pre>valid if l_variable_zenith_ang = .false.</pre>
l_diurnal_var_shortwave	logical	daily variational ampli-	default = .false., If this is true,
		tude of solar flux is cal-	<pre>interval_shortwave_sec = 86400</pre>
		culated in this module	
l_penetr_ang_considered	logical	introduction of trans-	default = .true.
		mitted angle based on	
		Snell's law	

Table14.7 Namelist nml_frac_nearir.

variable name	units	description	usage
qsratio_nearIR	real(8)	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 is the residual between the downward longwave radiation Q_{LO}^{down} (external data) and the upward black body radiation of sea water with the sea surface temperature (θ_{surf}),

$$Q_{LO}^{\text{net}} = Q_{LO}^{\text{down}} - e_m \sigma (\theta_{\text{surf}} + 273.15)^4.$$
(14.26)

Here, e_m is the emissivity for sea water, which should be specified by user, and $\sigma = 5.67 \times 10^{-5} \text{ erg} \cdot \text{s}^{-1} \cdot \text{cm}^{-2} \cdot \text{K}^{-4}$ is the Stefan-Boltzmann constant.

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. Model option

By default, the net longwave radiation is read as external forcing data. By choosing LWDOWN option, the downward longwave radiation is read. LWDOWN option will become a standard (default) choice in the near future.

b. 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	default = 1.0
		from sea water	

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),

$$Q_{LA} = -\rho_a L C_E |\Delta \vec{U}| (q_s - q_a), \qquad (14.27)$$

$$Q_{SN} = -\rho_a c_{pa} C_H |\Delta \vec{U}| (T_s - \theta_a).$$
(14.28)

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)
- BULKKONDO2 for Kondo (1975)

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):

$$F^W = P - E + R + I, (14.29)$$

Precipitation and river discharge are given as input data. 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:

$$E = \rho_a C_E |\Delta U| (q_s - q_a) = -Q_{LA}/L.$$
(14.30)

Note that evaporation (E) is defined as positive upward.

Fresh water flux is treated as the surface boundary condition for the surface height equation:

$$\frac{\partial \eta}{\partial t} = \dots + F^W. \tag{14.31}$$

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

Precipitation, evaporation, and river discharge ($W_{AO} = P - E + R$) are assumed to have the first level temperature (θ_{surf}) and zero salinity. The water transported through the ice bottom (W_{Ibot}) is also assumed to have the first level temperature. The water transported from the ice surface ($W_{Isurf} > 0$) is assumed to have the freezing point temperature (θ_{freeze}). The water exchanged with ice is assumed to have low salinity ($S_I = 4.0 \text{ pss}$). Note that the freezing point temperature (θ_{freeze}) is given by mS_I , where m = -0.0543 °C/pss.

a. When fresh water fluxes are explicitly added to the system

If fresh water flux is simply added to the first layer volume, surface temperature and salinity fluxes due to the fresh water flux are given as follows:

$$F_{WF}^{\theta} = (W_{AO} + W_{Ibot}) \cdot \theta_{\text{surf}} + W_{Isurf} \cdot \theta_{\text{freeze}}, \qquad (14.32)$$

$$F_{WF}^{S} = (W_{Ibot} + W_{Isurf}) \cdot S_{I}.$$
(14.33)

b. When fresh water fluxes are not explicitly added to the system

If the surface fresh water flux is not added to the first layer to avoid a long-term drift of sea level (SFLUXW option), equivalent temperature and salinity fluxes should be imposed to incorporate the effects of surface fresh water flux. To obtain a formula, the conservation equations for the first layer heat and salinity are considered. Temperature and salinity are assumed to evolve from the old values (θ_{old} , S_{old}) to the new values (θ_{new} , S_{new}) due to fresh water flux.

Chapter 14 Sea surface fluxes

If a volume change is allowed,

$$V_{\text{new}}\theta_{\text{new}} = V_{\text{old}}\theta_{\text{old}} + (W_{AO}\theta_{\text{old}} + W_{Ibot}\theta_{\text{old}} + W_{Isurf}\theta_{\text{freeze}}) \cdot \Delta A \cdot \Delta t, \qquad (14.34)$$

$$V_{\text{new}}S_{\text{new}} = V_{\text{old}}S_{\text{old}} + (W_{Ibot}S_I + W_{Isurf}S_I) \cdot \Delta A \cdot \Delta t, \qquad (14.35)$$

$$V_{\text{new}} = V_{\text{old}} + (W_{AO} + W_{Ibot} + W_{Isurf}) \cdot \Delta A \cdot \Delta t, \qquad (14.36)$$

where ΔA is the horizontal area.

If a volume change is *not* allowed,

$$V_{\text{old}}\theta_{\text{new}} = V_{\text{old}}\theta_{\text{old}} + F_{WF}^{\theta} \cdot \Delta A \cdot \Delta t, \qquad (14.37)$$

$$V_{\text{old}}S_{\text{new}} = V_{\text{old}}S_{\text{old}} + F_{WF}^S \cdot \Delta A \cdot \Delta t, \qquad (14.38)$$

where F_{WF}^{θ} and F_{WF}^{S} are temperature and salinity flux for the constant volume case. Removing θ_{new} , S_{new} , and V_{new} from the above equations, we have,

$$V_{\text{old}}\{V_{\text{old}}\theta_{\text{old}} + (W_{AO}\theta_{\text{old}} + W_{Ibot}\theta_{\text{old}} + W_{Isurf}\theta_{\text{freeze}}) \cdot \Delta A \cdot \Delta t\}$$

= { $V_{\text{old}} + (W_{AO} + W_{Ibot} + W_{Isurf}) \cdot \Delta A \cdot \Delta t$ }($V_{\text{old}}\theta_{\text{old}} + F_{WF}^{\theta} \cdot \Delta A \cdot \Delta t$), (14.39)

$$V_{\text{old}}\{V_{\text{old}}S_{\text{old}} + (W_{Ibot}S_I + W_{Isurf}S_I) \cdot \Delta A \cdot \Delta t\}$$

$$= \{V_{\text{old}} + (W_{AO} + W_{Ibot} + W_{Isurf}) \cdot \Delta A \cdot \Delta t\}(V_{\text{old}}S_{\text{old}} + F_{WF}^S \cdot \Delta A \cdot \Delta t).$$
(14.40)

The fluxes for the constant volume case are given by

$$F_{WF}^{\theta} = -\frac{V_{\text{old}}}{V_{\text{old}} + W \cdot \Delta A \cdot \Delta t} W_{Isurf} \cdot (\theta_{\text{old}} - \theta_{\text{freeze}}),$$
(14.41)

$$F_{WF}^{S} = -\frac{V_{\text{old}}}{V_{\text{old}} + W \cdot \Delta A \cdot \Delta t} \{ W_{AO} \cdot S_{\text{old}} + (W_{Ibot} + W_{Isurf}) \cdot (S_{\text{old}} - S_I) \},$$
(14.42)

where $W = W_{AO} + W_{Ibot} + W_{Isurf}$.

If $|V_{\text{old}}| \gg |W \cdot \Delta A \cdot \Delta t|$, we have,

$$F_{WF}^{\theta} = -W_{Isurf} \cdot (\theta_{\text{old}} - \theta_{\text{freeze}}), \tag{14.43}$$

$$F_{WF}^{S} = -W_{AO} \cdot S_{old} - (W_{Ibot} + W_{Isurf}) \cdot (S_{old} - S_{I}).$$
(14.44)

14.6.3 Numerical implementation

a. Adding/subtracting new sea water

Fresh water flux is applied to the sea level prediction in the baroclinic time interval:

$$\eta^{n+1} = \eta^{n-1} + \dots + 2\Delta t \frac{(F^W)^n + (F^W)^{n-1}}{2},$$
(14.45)

where $(F^W)^{n-1}$ and $(F^W)^n$ are fresh water fluxes evaluated at time levels n-1 and n, respectively.

In the same manner, fresh water transport is reflected in the equations for temperature

$$(\theta V)_{\frac{1}{2}}^{n+1} = (\theta V)_{\frac{1}{2}}^{n-1} + \dots + \Delta t \left[\{ (W_{AO})^{n-1} + (W_{Ibot})^{n-1} \} \cdot (\theta_{surf})^{n-1} + (W_{Isurf})^{n-1} \cdot (\theta_{freeze})^{n-1} \right] + \Delta t \left[\{ (W_{AO})^n + (W_{Ibot})^n \} \cdot (\theta_{surf})^n + (W_{Isurf})^n \cdot (\theta_{freeze})^n \right]$$
(14.46)

and salinity

$$(SV)_{\frac{1}{2}}^{n+1} = (SV)_{\frac{1}{2}}^{n-1} + \dots + \Delta t \{ (W_{Ibot})^{n-1} + (W_{Isurf})^{n-1} \} \cdot S_I + \Delta t \{ (W_{Ibot})^n + (W_{Isurf})^n \} \cdot S_I.$$
(14.47)

b. 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,

$$F_{ADJ}^{W} = F^{W} - \overline{F^{W}},\tag{14.48}$$

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 $\overline{F^W}$.

Another option for suppressing the long term drift of sea level is to transform all freshwater fluxes to salinity flux, which is activated by specifying SFLUXW option. Formulae for the temperature and salinity fluxes with this option are given by (14.43) and (14.44).

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.
- SFLUXW computes equivalent temperature and salinity fluxes due to fresh water flux

b. Input data

When WFLUX and RUNOFF options are specified, time series of precipitation and river run-off data must be prepared, respectively. 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 14.9.

The data may be interpolated in time and space in the model. See Section 14.10 for how to specify external forcing files.

element	name	units	usage
precipitation	Precipitation	$g s^{-1} cm^{-2}$	WFLUX
river discharge	RiverDischargeRate	${\rm g}{\rm s}^{-1}{\rm cm}^{-2}$	RUNOFF

Table14.9 Water flux data to be read from namelist nml_force_data.

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 14.10).

variable name	units	description	usage
l_glacier	logical	if a rivermouth grid has sea ice then river	default = .false.
		discharge is converted into iceberg input	
		from glacier	
l_after_rivermouth_spread	logical	river discharge is converted into iceberg af-	default = .false.
		ter a smoothing operation of river discharge	
		data	

Table14.10 Namelist nml_mkflux. See also Table 14.12.

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 14.11).

variable name	units	description	usage
l_strict_waterflux_adjust	logical	water flux adjustment is made MPI parallel	default = .false.
		independent	

Table14.11 Namelist nml_waterflux_adjust.

14.7 Treatment in the temperature equation

14.7.1 Formulation

The contribution of surface forcing (F_z^{θ}) and shortwave radiation flux (Q_{SH}) to the first layer temperature is expressed as

$$\frac{\partial\theta}{\partial t}\Big|_{k=\frac{1}{2}} = \dots + \frac{F_z^{\theta}}{\Delta z_{\frac{1}{2}}} + \frac{1}{\rho_0 c_p} \frac{\partial Q_{SH}}{\partial z},\tag{14.49}$$

where $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer of a T-cell. The term (F_z^{θ}) consists of the net sea surface heat flux except for the shortwave radiation flux (Q_{OTHER}) , heat transport due to fresh water flux (F_{WF}^{θ}) , and restoring of SST (θ_{surf}) to a specified value (θ^*) :

$$F_z^{\theta} = \frac{Q_{OTHER}}{\rho_0 c_p} + F_{WF}^{\theta} - \frac{1}{\gamma_{\theta}} (\theta_{\text{surf}} - \theta^*) \Delta z_{\frac{1}{2}}.$$
(14.50)

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 third term on the r.h.s. is a restoring term to a specified SST (θ^*). 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 thickness of the first layer of a T-cell (Δz_{\pm}).

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 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.

Treatment of supercooled water needs some explanation. 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.

14.8 Treatment in the salinity equation

14.8.1 Formulation

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 6). 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.

The explicit surface forcing (F_z^S) consists of salt transport due to formation and melting of sea ice and restoring of SSS (S_{surf}) to a specified value (S^*) :

$$F_{z}^{S} = (W_{Ibot} + W_{Isurf}) \cdot S_{I} - \frac{1}{\gamma_{s}} (S_{surf} - S^{*}) \Delta z_{\frac{1}{2}}.$$
 (14.51)

The water exchanged with ice $(W_{Ibot} + W_{Isurf})$ is assumed to have low salinity $(S_I = 4.0 \text{ pss})$. The parameter γ_s is 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 the thickness of the first layer of a T-cell (Δz_{\perp}) .

By default, surface salinity forcing consists only of the restoring term. When WFLUX option is specified, the effect of the fresh water flux on salinity may be included.

With WFLUX option, by default, the restoring term is converted to the fresh water flux. The expression for F^W becomes

$$F^{W} = P - E + R + I + \frac{1}{\gamma_{s}} \frac{S_{\text{surf}} - S^{*}}{S_{\text{surf}}} \Delta z_{\frac{1}{2}}.$$
(14.52)

In this case, the salinity flux should be

$$F_z^S = (W_{Ibot} + W_{Isurf}) \cdot S_I. \tag{14.53}$$

However, the restoring term is not converted to the freshwater flux if SFLUXR option is selected.

If SFLUXW option is specified, the fresh water flux is not added to the ocean and its effect on salinity is now explicitly expressed as the salinity flux:

$$F_{z}^{S} = -W_{AO} \cdot S_{\text{surf}} - (W_{Ibot} + W_{Isurf}) \cdot (S_{\text{surf}} - S_{I}) - \frac{1}{\gamma_{s}}(S_{\text{surf}} - S^{*})\Delta z_{\frac{1}{2}},$$
(14.54)

where S_{surf} is the first level salinity.

The contribution of surface forcing (F_z^S) to the first layer salinity is expressed as

$$\frac{\partial S}{\partial t}\Big|_{k=\frac{1}{2}} = \dots + \frac{F_z^S}{\Delta z_{\frac{1}{2}}},$$
(14.55)

where $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer of a T-cell.

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.
- SFLUXW converts fresh water flux to surface salinity flux
- SFLUXR suppresses conversion of salinity restoring to water flux
- 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 14.12 and 14.13.

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

variable name	units	description	usage
<pre>sss_rst_diff_max</pre>	pss	the maximum of the difference between model	default = 100 pss
		and reference sea surface salinity	
			Continued on next page

Table14.13 Namelist nml_sss_restore.

variable name	units	description	usage	
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	default = .false.	
		with sea ice		
l_sss_rst_cnsv	logical	set globally integrated surface salinity restoring	default = .false.	
		to zero		
l_strict_sss_rst_cnsv	logical	global summation of surface salinity restoring	default = .false.	
		flux is done serially so that the result becomes		
		MPI parallel independent		

Table 14.13 - continued from previous page

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) and Large and Yeager (2004). Bulk transfer coefficient for air-ice interface is given in Section 17.2.3.

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 (τ), sensible heat (H_S), latent heat (H_L), and water vapor (E) fluxes are written in terms of turbulent components as follows:

$$\tau = -\rho_a \overline{wu} = \rho_a u_*^2, \tag{14.56}$$

$$H_S = -\rho_a c_{pa} w \theta = \rho_a c_{pa} u_* \theta_*, \tag{14.57}$$

$$H_L = -\rho_a L_E wq = \rho_a L_E u_* q_*, \tag{14.58}$$

$$E = -\rho_a wq = \rho_a u_* q_* = H_L / L_E, \tag{14.59}$$

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, $\theta_* \equiv H_S/(\rho_a c_{pa} u_*)$ is the temperature scale in the boundary layer, $q_* \equiv H_L/(\rho_a L_E u_*) = E/(\rho_a u_*)$ 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:

$$\tau = \rho_a C_{\rm Dh} |\vec{U}_a - \vec{U}_s|^2, \tag{14.60}$$

$$H_{S} = -\rho_{a}c_{pa}|\vec{U}_{a} - \vec{U}_{s}|C_{\text{Hh}}(T_{s} - \theta_{a}), \qquad (14.61)$$

$$H_L = -\rho_a L_E |\vec{U_a} - \vec{U_s}| C_{\rm Eh} (q_s - q_a), \tag{14.62}$$

$$E = \rho_a |\vec{U_a} - \vec{U_s}| C_{\rm Eh} (q_s - q_a), \tag{14.63}$$

where $\vec{U_a}$ is the wind vector at a height of z_h , $\vec{U_s}$ is the current vector at the sea surface, and T_s is the absolute sea surface water temperature (K). The subscript "*a*" means the value at a height of $z = z_h$, and the subscript "*s*" means the value at the sea surface. 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 (14.64), (14.65), and (14.66) and the similarity law of Monin-Obukhov mentioned below,

$$C_{\rm Dh} = \frac{u_*^2}{|\vec{U}_a - \vec{U}_s|^2},\tag{14.64}$$

$$C_{\rm Hh} = -\frac{H_S/\rho_a c_{pa}}{|\vec{U_a} - \vec{U_s}|(T_s - \theta_a)} = \frac{u_* \theta_*}{|\vec{U_a} - \vec{U_s}|(T_s - \theta_a)},$$
(14.65)

$$C_{\rm Eh} = \frac{E/\rho_a}{|\vec{U_a} - \vec{U_s}|(q_s - q_a)} = \frac{u_* q_*}{|\vec{U_a} - \vec{U_s}|(q_s - q_a)}.$$
(14.66)

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:

$$\frac{\kappa z}{u_*} \frac{\partial U}{\partial z} = \phi_M \left(\frac{z}{L}\right),\tag{14.67}$$

$$\frac{\kappa z}{\theta_*} \frac{\partial \theta}{\partial z} = \phi_H \left(\frac{z}{L}\right),\tag{14.68}$$

$$\frac{\kappa z}{q_*} \frac{\partial q}{\partial z} = \phi_E \left(\frac{z}{L}\right),\tag{14.69}$$

where $\kappa = 0.4$ is the von Karman constant, and L is the Monin-Obukhov length scale

$$L = \frac{u_*^3}{\kappa Q_b} = -\frac{u_*^3 \theta_v}{\kappa g \,\overline{w \theta_v}}.$$
(14.70)

In (14.70), $Q_b = -g \overline{w\theta_v}/\theta_v$ is the buoyancy flux, θ_v is the virtual temperature ($\theta_v = \theta (1 + 0.6078 q)$), and $\overline{w\theta_v}$ is the net heat flux including the water vapor flux:

$$\overline{w\theta_v} = \overline{w\theta} \left(1 + 0.6078 \, q\right) + 0.6078 \, \theta \, \overline{wq}. \tag{14.71}$$

In (14.67) to (14.69), $\zeta = z/L$ is the Monin-Obukhov similarity variable, and $\phi_{M,H,E}$ is a nondimensional function for wind velocity, air temperature, and specific humidity. The nondimensional function is assumed to be a mathematically simple function.

Integrating Equations (14.67), (14.68), and (14.69) vertically, we have the following.

$$U(z) - U_s = \frac{u_*}{\kappa} \left[\ln \frac{z}{z_0} - \Psi_M(\zeta) \right],$$
(14.72)

$$\theta(z) - T_s = \frac{\theta_*}{\kappa} \left[\ln \frac{z}{z_{0T}} - \Psi_H(\zeta) \right], \tag{14.73}$$

$$q(z) - q_s = \frac{q_*}{\kappa} \left[\ln \frac{z}{z_{0\rm E}} - \Psi_E(\zeta) \right],$$
(14.74)

where

$$\Psi_{M,H,E}(\zeta) = \int_0^{\zeta} \left[1 - \phi_{M,H,E}(\zeta') \right] \frac{d\zeta'}{\zeta'}.$$
(14.75)

In (14.72) to (14.74), z_0 , z_{0T} , and z_{0E} are roughness lengths for each physical property. When the stability of the boundary layer is already known, bulk transfer coefficients can be estimated using these roughness lengths,

$$C_{\rm Dh} = \frac{\kappa^2}{\left[\ln\frac{z_{\rm h}}{z_0} - \Psi_M(\zeta_h)\right]^2},$$
(14.76)

$$C_{\text{Hh,Eh}} = \frac{\kappa^2}{\left[\ln\frac{z_{\text{h}}}{z_0} - \Psi_M(\zeta_h)\right] \left[\ln\frac{z_{\text{h}}}{z_{0\text{T,OE}}} - \Psi_{H,E}(\zeta_h)\right]}.$$
(14.77)

Chapter 14 Sea surface fluxes

In neutral stability ($\zeta = 0$), the bulk transfer coefficient is a function of the roughness length (z_{0N}, z_{0TN} , and z_{0EN} for each) only and is expressed as follows:

$$C_{\rm DNh} = \frac{\kappa^2}{\left[\ln\frac{z_{\rm h}}{z_{\rm 0N}}\right]^2},\tag{14.78}$$

$$C_{\text{HNh,ENh}} = \frac{\kappa^2}{\left[\ln\frac{z_{\text{h}}}{z_{\text{ON}}}\right] \left[\ln\frac{z_{\text{h}}}{z_{\text{OTN,OEN}}}\right]}$$
(14.79)

$$= \frac{C_{\rm DNh}}{\left[1 - \frac{1}{\kappa} C_{\rm DNh}^{1/2} \ln \frac{z_{\rm 0TN,0EN}}{z_{\rm 0N}}\right]}$$
(14.80)

$$=\frac{\kappa C_{\rm DNh}^{1/2}}{\left[\ln\frac{z_{\rm h}}{z_{\rm OTN,OEN}}\right]}.$$
(14.81)

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 by eliminating the roughness length as follows:

$$C_{\rm Dh} = \frac{C_{\rm DN10}}{\left[1 + \frac{1}{\kappa}C_{\rm DN10}C_{\rm DN10}^{-1/2}\left(\ln\frac{z_{\rm h}}{z_{10}} - \Psi_M(\zeta_h)\right)\right]^2},$$
(14.82)

$$C_{\rm Hh,Eh} = \frac{C_{\rm HN10,EN10}\left(\frac{C_{\rm Dh}}{C_{\rm DN10}}\right)^{1/2}}{\left[1 + \frac{1}{\kappa}C_{\rm HN10,EN10}C_{\rm DN10}^{-1/2}\left(\ln\frac{z_{\rm h}}{z_{10}} - \Psi_{H,E}(\zeta_h)\right)\right]},$$
(14.83)

where z_{10} means z = 10 m.

The neutral bulk transfer coefficients at a height of $10 \text{ m} (C_{\text{DN10}}, C_{\text{HN10}}, \text{and } C_{\text{EN10}})$ are often estimated, according to the stability, as a function of velocity at 10 m. Various formulations can be used (see the following subsections). To be more realistic, the bulk transfer coefficients should be regarded as a function of wave age, but its general formulation has not been achieved yet.

In principle, if we have a complete set of atmospheric elements in the boundary layer, we can get the bulk coefficients and fluxes because height can be adjusted using the similarity law. This requires an iteration of the calculation as follows (you can select an appropriate way for your data):

- Convert wind speed, air temperature, and specific humidity into those at 10 m in the neutral stability and use a bulk coefficient at 10 m in the neutral stability.
- Convert a bulk coefficient at 10 m into one with a height and a stability at which atmospheric elements are observed.
- Convert air temperature and specific humidity into those at a height where wind speeds are observed, and convert the bulk coefficient at 10 m in the neutral stability into one with a height and a stability at which wind speeds are observed.

14.9.2 Large and Yeager (2004; 2009): BULKNCAR

The bulk formula based on Large and Yeager (2004; 2009) closely follows the formulation explained in the previous subsection. Each 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. To do this iteration in the program bulk.F90, specify BULKITER option in configure.in. The observed heights are specified in namelist nml_srfflx_dbase (see Table 14.18).

The bulk transfer coefficient at 10 m in the neutral stability is given as follows:

$$10^{3}C_{\text{DN10}} = \begin{cases} \frac{2.70}{U_{10N}} + 0.142 + \frac{U_{10N}}{13.09} - 3.14807 \times 10^{-10} U_{10N}^{6}, & U_{10N} < 33 \,\text{m s}^{-1} \\ 2.34 & U_{10N} \ge 33 \,\text{m s}^{-1} \end{cases}$$
(14.84)

$$10^{3}C_{\text{HN10,EN10}} = 18.0\sqrt{C_{\text{DN10}}}, \text{ stable } \zeta > 0,$$
 (14.85)

$$10^{3}C_{\text{HN10,EN10}} = 32.7\sqrt{C_{\text{DN10}}}, \text{ unstable } \zeta \le 0,$$
 (14.86)

Each physical property is estimated at a height where wind speed is observed in the following operations. First, calculate the virtual temperature θ_v as follows:

$$\theta_{\nu} = \theta(z_{\theta})(1 + 0.6078 \, q(z_q)) \tag{14.87}$$

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 $U_{10N} = |\Delta \vec{U}(z_u)| = |\vec{U}_a(z_u) - \vec{U}_s|$ (Equations (14.84) to (14.86)). 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,

$$u_* = \sqrt{\frac{\tau}{\rho_a}} = \sqrt{C_{\text{DN10}}} \left| \Delta \vec{U}(z_u) \right|,\tag{14.88}$$

$$\theta_* = \frac{H_S}{\rho_a c_p u_*} = \frac{C_{\rm HN10}}{\sqrt{C_{\rm DN10}}} \,(\theta(z_\theta) - T_s),\tag{14.89}$$

$$q_* = \frac{E}{\rho_a u_*} = \frac{C_{\text{EN10}}}{\sqrt{C_{\text{DN10}}}} \left(q(z_q) - q_{\text{sat}}(T_s) \right), \tag{14.90}$$

where $q_{sat}(T_s)$ is the saturated specific humidity at sea surface temperature T_s .

Next, perform the iteration using the three Monin-Obukhov similarity variables, $\zeta_u = z_u/L$, $\zeta_\theta = z_\theta/L$, and $\zeta_q = z_q/L$, 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:

$$\zeta = \frac{\kappa g z}{u_*^2} \left[\frac{\theta_*}{\theta_v} + \frac{q_*}{(q(z_q) + 0.6078^{-1})} \right].$$
(14.91)

The integral of the non-dimensional profile function is expressed as

$$\Psi_M(\zeta) = \Psi_H(\zeta) = -5\zeta, \tag{14.92}$$

if it is stable ($\zeta \ge 0$), and

$$\Psi_M(\zeta) = 2\ln\left(\frac{1+X}{2}\right) + \ln\left(\frac{1+X^2}{2}\right) - 2\tan^{-1}(X) + \frac{\pi}{2},\tag{14.93}$$

$$\Psi_H(\zeta) = 2\ln\left(\frac{1+X^2}{2}\right),$$
(14.94)

if it is unstable ($\zeta < 0$). In the above,

$$X = (1 - 16\zeta)^{1/4}.$$
(14.95)

1

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,

$$U_{10N} = |\Delta \vec{U}(z_u)| \left(1 + \frac{\sqrt{C_{DN10}}}{\kappa} \left[\ln \frac{z_u}{z_{10}} - \Psi_M(\zeta_u) \right] \right)^{-1},$$
(14.96)

$$\theta(z_u) = \theta(z_\theta) - \frac{\theta_*}{\kappa} \left[\ln \frac{z_\theta}{z_u} + \Psi_H(\zeta_u) - \Psi_H(\zeta_\theta) \right], \tag{14.97}$$

$$q(z_u) = q(z_q) - \frac{q_*}{\kappa} \left[\ln \frac{z_q}{z_u} + \Psi_H(\zeta_u) - \Psi_H(\zeta_q) \right],$$
(14.98)
Chapter 14 Sea surface fluxes

where z_{10} means z = 10 m. Estimate the bulk coefficient at 10 m in the neutral stability using U_{10N} , and then obtain the bulk coefficient at a height (z_u) where wind speed is observed,

 $i \sim 1/2$

$$C_{\rm Du} = \frac{C_{\rm DN10}}{\left[1 + \frac{1}{\kappa} C_{\rm DN10} C_{\rm DN10}^{-1/2} \left(\ln \frac{z_u}{z_{10}} - \Psi_M(\zeta_u)\right)\right]^2},\tag{14.99}$$

$$C_{\rm Hu} = \frac{C_{\rm HN10} \left(\frac{C_{\rm Du}}{C_{\rm DN10}}\right)^{1/2}}{\left[1 + \frac{1}{\kappa} C_{\rm HN10} C_{\rm DN10}^{-1/2} \left(\ln \frac{z_u}{z_{10}} - \Psi_H(\zeta_u)\right)\right]},$$
(14.100)
$$C_{\rm Eu} = \frac{C_{\rm EN10} \left(\frac{C_{\rm Du}}{C_{\rm DN10}}\right)^{1/2}}{\left[1 + \frac{1}{\kappa} C_{\rm EN10} C_{\rm DN10}^{-1/2} \left(\ln \frac{z_u}{z_{10}} - \Psi_H(\zeta_u)\right)\right]}.$$
(14.101)

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 (14.87) and the scales for friction velocity, temperature, and specific humidity (Equations of (14.88), (14.89), and (14.90)).

14.9.3 Appendix of Kondo (1975): BULKKONDO2

In Appendix 1-3 of Kondo (1975), approximate formulae of the bulk transfer coefficients are given in support of practical applications. The BULKKONDO2 option realizes this in MRI.COM. In this scheme, providing that atmospheric elements are observed at various heights, latent and sensible heat fluxes are estimated at a height where air temperature and humidity are observed. Momentum fluxes are also estimated at this height. Wind speed and bulk transfer coefficients should be corrected to ones at the height where the air temperature and humidity are observed. In the program of bulk.F90, users may specify BULKITER option in configure.in to do this iteration. The observed heights are specified in namelist nml_srfflx_dbase (see Table 14.18).

The following procedure is executed starting from the observed wind speed relative to the surface current $|\Delta \vec{U}(z_u)|$ in units of [m/s] at a height of z_u [m]:

1. Estimate the neutral bulk transfer coefficient at 10 m assuming that the observed wind speed is at 10 m in the neutral stability $(U_{10} = |\Delta \vec{U}(z_u)|)$. When the 10 m wind speed is already known, the neutral bulk transfer coefficient is given as follows:

$$C_{\rm DN10} = 10^{-3} \{ a_d + b_d U_{10}^{p_d} \}, \tag{14.102}$$

where a_d , b_d , and p_d are nondimensional numbers depending on the wind speed (Table 14.14).

2. Recalculate the roughness length using this neutral bulk transfer coefficient at 10 m and correct the height for wind speed once again:

$$z_0 = \exp\left\{\ln z_{10} - \kappa C_{\rm DN10}^{-1/2}\right\},\tag{14.103}$$

$$U_{10} = |\Delta U(z_u)| \ln(10/z_0) / \ln(z_u/z_0), \qquad (14.104)$$

where z_{10} means z = 10 m.

- 3. Re-estimate the neutral bulk transfer coefficient at 10 m using (14.102).
- 4. Operations 1 through 3 above are repeated twice or three times to determine C_{DN10} and U_{10} .
- 5. Convert the bulk transfer coefficient to that in the neutral stability at the observed height of temperature and humidity (z_h) :

$$C_{\rm DNh} = \frac{\kappa^2}{\left[\kappa C_{\rm DN10}^{-1/2} - \ln(z_{10}/z_h)\right]^2}$$
(14.105)

$$C_{\rm HNh, ENh} = \frac{\kappa C_{\rm DNh}^{1/2}}{\kappa C_{\rm DN10}^{1/2} / C_{\rm HN10, EN10} + \ln(z_h/z_{10})}.$$
(14.106)

The neutral bulk transfer coefficients at 10 m for evaporative flux and sensible heat flux (C_{EN10} and C_{HN10} respectively) are given as follows:

$$C_{\text{EN10}} = 10^{-3} \{ a_e + b_e U_{10}^{p_e} + c_e (U_{10} - 8)^2 \},$$
(14.107)

$$C_{\rm HN10} = 10^{-3} \{ a_h + b_h U_{10}^{p_h} + c_h (U_{10} - 8)^2 \},$$
(14.108)

where $a_{e,h}$, $b_{e,h}$, $c_{e,h}$, and $p_{e,h}$ are nondimensional numbers depending on wind speed (Table 14.14).

6. Estimate the bulk transfer coefficient in which stability is considered using the neutral bulk transfer coefficient obtained above.

Table14.14 Non-dimensional parameters used in calculating the bulk transfer coefficient based on Kondo (1975)

U ₁₀	a _d	a_e	a_h	b_d	b_e	b_h	Ce	Ch	<i>Pd</i>	p_e	p_h
$(\mathbf{m} \cdot \mathbf{s}^{-1})$											
0.3-2.2	0	0	0	1.08	1.23	1.185	0	0	-0.15	-0.16	-0.157
2.2-5	0.771	0.969	0.927	0.0858	0.0521	0.0546	0	0	1	1	1
5-8	0.867	1.18	1.15	0.0667	0.01	0.01	0	0	1	1	1
8-25	1.2	1.196	1.17	0.025	0.008	0.0075	-0.0004	-0.00045	1	1	1
25-50	0	1.68	1.652	0.073	-0.016	-0.017	0	0	1	1	1

The bulk transfer coefficient depending on the stability is estimated in the following operations. First, the stability of the atmospheric boundary layer is defined as follows:

$$s = s_0 \frac{|s_0|}{|s_0| + 0.01},\tag{14.109}$$

where

$$s_0 = \frac{T_s - \theta_a}{|\Delta \vec{U}(z_h)|^2 [1.0 + \log_{10}(10/z_h)]^2)}.$$
(14.110)

and $|\Delta \vec{U}(z_h)|$ is approximated by

$$|\Delta \vec{U}(z_h)| = |\Delta \vec{U}(z_u)| \ln(z_h/z_0) / \ln(z_u/z_0).$$
(14.111)

The atmospheric boundary layer is unstable if s > 0 ($T_s - \theta_a > 0$) and stable if s < 0 ($T_s - \theta_a < 0$). Finally, the bulk transfer coefficient depending on the stability is given as follows:

$$C_D = \begin{cases} 0 & s < -3.3\\ C_{\text{DNh}}\{0.1 + 0.03s + 0.9 \exp(4.8s)\} & -3.3 \le s < 0\\ C_{\text{DNh}}(1.0 + 0.47\sqrt{s}) & 0 \le s, \end{cases}$$
(14.112)

$$C_E = \begin{cases} 0 & s < -3.3\\ C_{\text{ENh}}\{0.1 + 0.03s + 0.9 \exp(4.8s)\} & -3.3 \le s < 0\\ C_{\text{ENh}}(1.0 + 0.63\sqrt{s}) & 0 \le s, \end{cases}$$
(14.113)

$$C_{H} = \begin{cases} 0 & s < -3.3\\ C_{\text{HNh}}\{0.1 + 0.03s + 0.9 \exp(4.8s)\} & -3.3 \le s < 0\\ C_{\text{HNh}}(1.0 + 0.63\sqrt{s}) & 0 \le s. \end{cases}$$
(14.114)

14.9.4 Numerical implementation

Because the iteration loop of the bulk formulae is not vectorized, the count of iteration is set twice and the repeating operation is explicitly written in the source programs.

14.9.5 Usage

- BULKKONDO2 for Kondo (1975)
- BULKNCAR for Large and Yeager (2004; 2009)
- BULKITER for computing bulk-transfer coefficient using the iterative method.

14.10 Specification of input data

The attributes of external forcing data should be specified by namelist nml_force_data. Table 14.15 explains variables that are contained in this namelist block.

variable name	units	description	usage
name		name of data object	
txyu		grid point on which data should be placed	
12d	logical	data is 2D or not	.true. for surface data, de-
			fault = .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 size of data in x direction	
jmfrc		grid size 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.

Table14.15 Namelist nml_force_data related to surface forcing data.

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 14.16). This object contains a sub object meteor_alt of type type_surf_meteor_alt (Table 14.17) 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 14.18).

Air-sea fluxes computed on the basis of external data are stored in the data set object type type_surf_product_ao (Table 14.19). This object contains sub objects givn_ao and calc_ao of the object type type_surf_bulk_ao (Table 14.20). 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 14.21), which contains sub objects givn_ai and calc_ai of the object type type_surf_bulk_ai (Table 14.22).

variable name	units	description
name		name of data object
<pre>meteor_alt</pre>	<pre>type_surf_meteor_alt</pre>	Observation altitude of this data set
x_wind	$cm sec^{-1}$ or $dyn cm^{-2}$	X-ward wind or wind stress
y_wind	$\mathrm{cm}\mathrm{sec}^{-1}$ or $\mathrm{dyn}\mathrm{cm}^{-2}$	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
		Continued on next page

Table14.16 Contents of the object type_surf_dbase related to external surface forcing data.

Chapter 14 Sea surface fluxes

variable name	units	description
slpress	hPa	sea level pressure
sw_flx	$\mathrm{erg}\mathrm{s}^{-1}\mathrm{cm}^{-2}$	upward or downward shortwave, summed
sw_flx_visb	$\mathrm{erg}\mathrm{s}^{-1}\mathrm{cm}^{-2}$	upward or downward shortwave, visible, beam
sw_flx_visd	$\mathrm{erg}\mathrm{s}^{-1}\mathrm{cm}^{-2}$	upward or downward shortwave, visible, diffuse
sw_flx_nirb	$\mathrm{erg}\mathrm{s}^{-1}\mathrm{cm}^{-2}$	upward or downward shortwave, near Infra-red, beam
sw_flx_nird	$\mathrm{erg}\mathrm{s}^{-1}\mathrm{cm}^{-2}$	upward or downward shortwave, near Infra-red, diffuse
lw_flx	$\mathrm{erg}\mathrm{s}^{-1}\mathrm{cm}^{-2}$	upward or downward longwave
sst	°C	sea surface temperature
precip	$\mathrm{cm}\mathrm{s}^{-1}$	precipitation water flux
river_w	$\mathrm{cm}\mathrm{s}^{-1}$	river discharge water flux
river_h	$\mathrm{erg}\mathrm{s}^{-1}\mathrm{cm}^{-2}$	river discharge heat flux
snowfall	$\mathrm{cm}\mathrm{s}^{-1}$	snow fall water flux
iceberg_w	$\mathrm{cm}\mathrm{s}^{-1}$	iceberg discharge water flux
iceberg_h	$\mathrm{erg}\mathrm{s}^{-1}\mathrm{cm}^{-2}$	iceberg discharge heat flux
frac_ice	1	sea ice fraction

Table 14.16 – continued	from	previous	page
-------------------------	------	----------	------

Table14.17 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

Table14.18 Namelist nml_srfflx_dbase.

variable name	units	description	usage
name		name of the external dataset	ReadData (Reanalysis) or FromCGCM (AGCM)
			(required)
l_marine_meteor	logical	read marine meteorological data or not	required
alt_wind_meter	m	observed height of wind	required if 1_marine_meteor = .true.
alt_temp_meter	m	observed height of temperature	required if 1_marine_meteor = .true.
alt_sphm_meter	m	observed height of specific humidity	required if 1_marine_meteor = .true.

Table14.19 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
<pre>sw_nt_visd_ao</pre>	$erg s^{-1} cm^{-2}$	net shortwave, visible, diffuse
sw_nt_nirb_ao	$erg s^{-1} cm^{-2}$	net shortwave, near Infra-red, beam
<pre>sw_nt_nird_ao</pre>	$erg s^{-1} cm^{-2}$	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	<pre>type_surf_bulk_ao</pre>	calculated using bulk formulae

Table14.20	Contents of the object type	_surf_bulk_	_ao that stores	elements related	to computation of	of air-sea fluxes
based on bul	lk formula.					

variable name	units	description
name		name of data object
x_rel_wind	m sec ⁻¹	surface wind relative to surface current
y_rel_wind	m sec ⁻¹	surface wind relative to surface current
x_stress	dyn cm ⁻²	X-ward surface stress
y_stress	dyn cm ⁻²	Y-ward surface stress
frcvl	m sec ⁻¹	friction velocity
rel_wind	m sec ⁻¹	surface wind relative to surface current
latent	$erg s^{-1} cm^{-2}$	latent heat flux, air-ocean
sensible	$erg s^{-1} cm^{-2}$	sensible heat flux, air-ocean
evapo	$\rm cmsec^{-1}$	evaporative water flux, air-ocean

Table14.21 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	erg s cm ^{−2}	net shortwave, summed, absorbed at the ice surface
sw_nt_in	erg s cm ^{−2}	net shortwave, summed, penetrate into the ice interior
sw_nt_visb_ai	erg s cm ^{−2}	net shortwave, visible, beam
sw_nt_visd_ai	erg s cm ^{−2}	net shortwave, visible, diffuse
sw_nt_nirb_ai	erg s cm ^{−2}	net shortwave, near Infra-red, beam
sw_nt_nird_ai	erg s cm ⁻²	net shortwave, near Infra-red, diffuse
lw_nt_ai	erg s cm ^{−2}	net longwave, air-ice
tsfc_ice	°C	ice surface temperature
wind_speed	$m sec^{-1}$	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

Table14.22 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	$m sec^{-1}$	surface wind relative to ice drift
y_rel_wind	$m sec^{-1}$	surface wind relative to ice drift
x_stress	dyn cm ⁻²	X-ward surface stress
y_stress	dyn cm ⁻²	Y-ward surface stress
frcvl	$m sec^{-1}$	friction velocity
rel_wind	m sec ⁻¹	surface wind relative to ice drift
latent	$W m^{-2}$	latent heat flux, air-ice
sensible	$ m Wm^{-2}$	sensible heat flux, air-ice
sublim	m sec ⁻¹ of sea water	sublimation water flux, air-ice
ctntmp	m sec ⁻¹	bulk transfer coefficient \times scalar wind
csntmp	$m sec^{-1}$	bulk transfer coefficient \times scalar wind

14.12 Technical details

14.12.1 Work flow in the main program

Work flow in the main program is as follows. Note that the sea ice model is part of the surface flux module (surfflux). In the surface flux module, air-sea fluxes are calculated first, then they are partly modified by the presence of sea ice.

```
ogcm__run (ogcm.F90)
  + LOOP_CGCM_SCUP
  + cgcm_scup_ctl__cpl_loop_start
  T
     (Update surface fluxes 'FromCGCM')
  L
     + cgcm_scup__get_a2o
  L
     T
 + cgcm_scup__get_atmos_a2o
                                 (Update surface atmospheric state 'FromCGCM')
  L
 + LOOP_TIME_STEP
  T
     T
     + part_1
  + surfflux_ctl__set_surf_state
         + force__everytime
                                      (Update external forcing 'ReadData')
         Т
         (Compute solar insolation angle)
             + solarangle__main
         + surfflux_ctl__main
              T
              + surfflux__air_ocean
                                      (Determine Air-Sea flux,
                                      either 'ReadData' or 'FromCGCM')
              + surfflux__seaice
                                      (Flux modification due to the presence of Sea Ice)
              + surfflux__oceanflux
                                      (Determine fluxes at the sea surface)
              I
              + solarangle__shrtwv_flux (Determine veritcal distribution
                                        of solar insolation flux)
   end LOOP_TIME_STEP
  +
  I
   cgcm_scup_ctl__cpl_loop_end
  +
  (Transfer surface oceanic state to coupler)
     + cgcm_scup__put_o2a
  I
 + end LOOP_CGCM_SCUP
```

14.12.2 Coupling with an atmospheric model

a. Standard case

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 17.9.2 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. Getting fluxes from the atmospheric model at the beginning of a coupling cycle in cgcm_scup_get_a2o (cgcm_scup.F90).
- 2. Getting fluxes for this step from get_flux_a2o (get_fluxes.F90).
- 3. Sea surface fluxes are determined after solving the sea ice model
- 4. Solve time evolution of the oceanic state
- 5. Send oceanic state to the atmospheric model

b. Recalculating and replacing fluxes in the ocean model

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), the 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_neplace, which is explained in Tables 14.23 and 14.24.

Table14.23	Namelist nml	cacm	recalc.
140101	1 (control and a second		

variable name	units	description	usage
l_recalc_flux	logical	Recalculating of fluxes	default = .false.

Table14.24	Namelist nml	cqcm	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	real(8) 2d array (1/0) written with di-
		grid points where fluxes are replaced.	rect access

14.12.3 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

$$e/p = q/(\epsilon + (1 - \epsilon)q), \tag{14.115}$$

where ϵ is the molecular weight ratio between water vapor and air:

$$\epsilon = m_{\rm w}/m_{\rm a} = 18.016/28.966 = 0.62197.$$
 (14.116)

This is solved for the specific humidity as

$$q = \epsilon e/(p - (1 - \epsilon)e). \tag{14.117}$$

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

$$r = \frac{q}{1 - q}.$$
(14.118)

Thus the relative humidity is computed using the specific humidity q and the saturation specific humidity q_s as

$$\gamma = \frac{q(1-q_s)}{q_s(1-q)}.$$
(14.119)

The saturation vapor pressure e_{sw} [hPa] of pure water vapor over a plane water surface is given by

$$\log_{10} e_{sw}(t) = (0.7859 + 0.03477 t)/(1 + 0.00412 t), \tag{14.120}$$

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

$$e_{sw}' = f_w e_{sw}.$$
 (14.121)

The value of f_w is given by

$$f_w = 1 + 10^{-6} p (4.5 + 0.0006 t^2), \tag{14.122}$$

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

$$\log_{10} e_{si}(t) = \log_{10} e_{sw}(t) + 0.00422t.$$
(14.123)

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 10⁴ by (14.122). Thus

$$e_{si}' = f_w e_{si}.$$
 (14.124)

b. Specific heat of air

$$c_{pa} = \frac{7}{2} R \left(1 - q + \frac{8q}{7\epsilon} \right)$$

= 1004.6 × (1 + 0.8735q) J kg⁻¹ K⁻¹ (14.125)

c. Latent heat

The latent heat of vaporization is given by

$$L_{\nu} = 2.5008 \times 10^{6} - 2.3 \times 10^{3} t \,\mathrm{J \, kg^{-1}}.$$
(14.126)

The latent heat of sublimation is given by

$$L_s = 2.839 \times 10^6 - 3.6(t+35)^2 \,\mathrm{J\,kg^{-1}}.$$
(14.127)

d. Air density (ρ_a)

The equation of state of moist air of temperature T [K] under the pressure P [Pa] is given by

$$\rho_a = \frac{P}{RT(1 - q + q/\epsilon)} \equiv P/RT_\nu, \tag{14.128}$$

where

$$T_{\nu} \equiv T(1 - q + q/\epsilon) = T(1 + 0.6078q)$$
(14.129)

is called the virtual temperature and

$$R = 287.04 \,\mathrm{J}\,\mathrm{kg}^{-1}\,\mathrm{K}^{-1} \tag{14.130}$$

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 twoequation 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.

	Mellor and Yamada	Noh and Kim	Umlauf and Burchard
model option	MELYAM	NOHKIM	GLS
prognosticated variables	$q^2/2$ (15.42)	<i>E</i> (15.57)	<i>E</i> (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)

Table15.1 Features and equation numbers of turbulence closure models explained in this chapter.

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 U, averaged pressure P, and averaged potential temperature Θ are

$$\frac{\partial U_i}{\partial x_i} = 0,\tag{15.1}$$

$$\frac{DU_j}{Dt} + \epsilon_{jkl} f_k U_l = \frac{\partial}{\partial x_k} (-\langle u_k u_j \rangle) - \frac{1}{\rho_0} \frac{\partial P}{\partial x_j} - g_j \frac{\rho}{\rho_0} + \frac{\partial}{\partial x_k} \Big[2\nu \frac{1}{2} \Big(\frac{\partial U_k}{\partial x_j} + \frac{\partial U_j}{\partial x_k} \Big) \Big], \tag{15.2}$$

$$\frac{D\Theta}{Dt} = \frac{\partial}{\partial x_k} (-\langle u_k \theta \rangle) + \frac{\partial}{\partial x_k} \Big(\kappa \frac{\partial \Theta}{\partial x_k} \Big), \tag{15.3}$$

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:

$$\frac{D\langle u_{i}u_{j}\rangle}{Dt} + \frac{\partial}{\partial x_{k}} \Big[\langle u_{k}u_{i}u_{j}\rangle - \nu \frac{\partial \langle u_{i}u_{j}\rangle}{\partial x_{k}} \Big] + \frac{1}{\rho_{0}} \Big[\frac{\partial \langle pu_{i}\rangle}{\partial x_{j}} + \frac{\partial \langle pu_{j}\rangle}{\partial x_{i}} \Big] + f_{k}(\epsilon_{jkl}\langle u_{l}u_{i}\rangle + \epsilon_{ikl}\langle u_{l}u_{j}\rangle)
= \Big[-\langle u_{k}u_{i}\rangle \frac{\partial U_{j}}{\partial x_{k}} - \langle u_{k}u_{j}\rangle \frac{\partial U_{i}}{\partial x_{k}} \Big] - \frac{\beta}{\rho_{0}} (g_{j}\langle u_{i}\theta\rangle + g_{i}\langle u_{j}\theta\rangle) + \Big\langle \frac{p}{\rho_{0}} \Big(\frac{\partial u_{i}}{\partial x_{j}} + \frac{\partial u_{j}}{\partial x_{i}} \Big) \Big\rangle - 2\nu \Big\langle \frac{\partial u_{i}}{\partial x_{k}} \frac{\partial u_{j}}{\partial x_{k}} \Big\rangle, \quad (15.4)$$

$$\frac{D\langle u_{j}\theta\rangle}{Dt} + \frac{\partial}{\partial x_{k}} \Big[\langle u_{k}u_{j}\theta\rangle - \kappa \langle u_{j}\frac{\partial\theta}{\partial x_{k}}\rangle - \nu \langle \theta\frac{\partial u_{j}}{\partial x_{k}}\rangle \Big] + \frac{1}{\rho_{0}} \langle \theta\frac{\partial p}{\partial x_{j}}\rangle + \epsilon_{jkl}f_{k}\langle u_{l}\theta\rangle$$

$$= \Big[-\langle u_{i}u_{k}\rangle \frac{\partial\Theta}{\partial x_{k}} - \langle u_{k}\theta\rangle \frac{\partial U_{j}}{\partial x_{k}} \Big] - \frac{\beta}{\rho_{g}}g_{i}\langle\theta^{2}\rangle - (\kappa + \nu) \langle \frac{\partial u_{j}}{\partial x_{k}}\frac{\partial\theta}{\partial x_{k}} \rangle, \qquad (15.5)$$

$$\frac{\partial \langle \theta^2 \rangle}{\partial t} + \frac{\partial}{\partial x_k} \left[\langle u_k \theta^2 \rangle - \kappa \langle \frac{\partial \theta^2}{\partial x_k} \rangle \right] = -2 \langle u_k \theta \rangle \frac{\partial \Theta}{\partial x_k} - 2\kappa \left(\frac{\partial \theta}{\partial x_k} \frac{\partial \theta}{\partial x_k} \right).$$
(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:

$$\left\langle \frac{p}{\rho_0} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right\rangle = -\frac{q}{3l_1} \left(\langle u_i u_j \rangle - \frac{\delta_{ij}}{3} q^2 \right) + C_1 q^2 \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right), \tag{15.7}$$

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:

$$\left\langle \frac{p}{\rho_0} \frac{\partial \theta}{\partial x_j} \right\rangle = -\frac{q}{3l_2} \langle u_j \theta \rangle \tag{15.9}$$

$$(\kappa + \nu) \Big\langle \frac{\partial u_j}{\partial x_k} \frac{\partial \theta}{\partial x_k} \Big\rangle = 0, \tag{15.10}$$

where l_2 is the length scale. The dissipation of temperature variance is

$$2\kappa \left\langle \frac{\partial \theta}{\partial x_k} \frac{\partial \theta}{\partial x_k} \right\rangle = -2 \frac{q}{\Lambda_2} \langle \theta^2 \rangle, \tag{15.11}$$

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:

$$\langle u_k u_i u_j \rangle = \frac{3}{5} lq S_q \Big(\frac{\partial \langle u_i u_j \rangle}{\partial x_k} + \frac{\partial \langle u_i u_k \rangle}{\partial x_j} + \frac{\partial \langle u_j u_k \rangle}{\partial x_i} \Big), \tag{15.12}$$

$$\langle u_k u_j \theta \rangle = -lq S_{u\theta} \Big(\frac{\partial \langle u_k \theta \rangle}{\partial x_j} + \frac{\partial \langle u_j \theta \rangle}{\partial x_k} \Big), \tag{15.13}$$

$$\langle u_k \theta^2 \rangle = -lq S_\theta \frac{\partial \langle \theta^2 \rangle}{\partial x_k},\tag{15.14}$$

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:

$$(l_1, \Lambda_1, l_2, \Lambda_2) = (A_1, B_1, A_2, B_2)l,$$
(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

$$\frac{DU}{Dt} + \frac{\partial}{\partial z} \langle uw \rangle = -\frac{1}{\rho_0} \frac{\partial P}{\partial x} + fV, \qquad (15.16)$$

$$\frac{DV}{Dt} + \frac{\partial}{\partial z} \langle vw \rangle = -\frac{1}{\rho_0} \frac{\partial P}{\partial y} - fU, \qquad (15.17)$$

$$0 = -\frac{1}{\rho_0}\frac{\partial P}{\partial z} - g\frac{\rho}{\rho_0},\tag{15.18}$$

$$\frac{D\Theta}{Dt} + \frac{\partial}{\partial z} \langle \langle w\theta \rangle \rangle = \frac{1}{\rho_0 c_p} \frac{\partial R}{\partial z}.$$
(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

$$P_s = -\langle wu \rangle \frac{\partial U}{\partial z} - \langle wv \rangle \frac{\partial V}{\partial z}$$
(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.

$$\langle u^2 \rangle = \frac{q^2}{3} + \frac{l_1}{q} \Big[-4 \langle wu \rangle \frac{\partial U}{\partial z} + 2 \langle wv \rangle \frac{\partial V}{\partial z} - 2P_b \Big], \tag{15.24}$$

$$\langle v^2 \rangle = \frac{q^2}{3} + \frac{l_1}{q} \Big[2 \langle wu \rangle \frac{\partial U}{\partial z} - 4 \langle wv \rangle \frac{\partial V}{\partial z} - 2P_b \Big], \tag{15.25}$$

$$\langle w^2 \rangle = \frac{q^2}{3} + \frac{l_1}{q} \Big[2 \langle wu \rangle \frac{\partial U}{\partial z} + 2 \langle wv \rangle \frac{\partial V}{\partial z} + 4P_b \Big], \tag{15.26}$$

$$\langle uv \rangle = \frac{3l_1}{q} \Big[-\langle uw \rangle \frac{\partial V}{\partial z} - \langle vw \rangle \frac{\partial U}{\partial z} \Big], \tag{15.27}$$

$$\langle wu \rangle = \frac{3l_1}{q} \Big[-(\langle w^2 \rangle - C_1 q^2) \frac{\partial U}{\partial z} - g \langle u\rho \rangle \Big], \tag{15.28}$$

$$\langle vw \rangle = \frac{3l_1}{q} \Big[-(\langle w^2 \rangle - C_1 q^2) \frac{\partial V}{\partial z} - g \langle v\rho \rangle \Big], \tag{15.29}$$

$$\langle u\theta \rangle = \frac{3l_2}{q} \Big[-\langle uw \rangle \frac{\partial \Theta}{\partial z} - \langle w\theta \rangle \frac{\partial U}{\partial z} \Big], \tag{15.30}$$

$$\langle v\theta \rangle = \frac{3l_2}{q} \Big[-\langle vw \rangle \frac{\partial \Theta}{\partial z} - \langle w\theta \rangle \frac{\partial V}{\partial z} \Big], \tag{15.31}$$

$$\langle w\theta \rangle = \frac{3l_2}{q} \Big[-\langle w^2 \rangle \frac{\partial \Theta}{\partial z} - g \langle \theta \rho \rangle \Big], \tag{15.32}$$

$$\langle \theta^2 \rangle = -\frac{\Lambda_2}{q} \langle w \theta \rangle \frac{\partial \Theta}{\partial z}.$$
(15.33)

Some of the terms in these equations, which are used in (15.16)-(15.19), can be further deformed by substitution as

Chapter 15 Turbulence Closure Models

follows:

$$-\langle uw\rangle = K_M \frac{\partial U}{\partial z},\tag{15.34}$$

$$-\langle vw\rangle = K_M \frac{\partial V}{\partial z},\tag{15.35}$$

$$-\langle \theta w \rangle = K_H \frac{\partial \Theta}{\partial z},\tag{15.36}$$

$$K_M = lqS_M,\tag{15.37}$$

$$K_H = lqS_H. (15.38)$$

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:

$$S_{M}[6A_{1}A_{2}g_{M}] + S_{H}[1 - 3A_{2}B_{2}g_{H} - 12A_{1}A_{2}g_{H}] = A_{2},$$

$$S_{M}[1 + 6A_{1}^{2}g_{M} - 9A_{1}A_{2}g_{H}] - S_{H}[12A_{1}^{2}g_{H} + 9A_{1}A_{2}g_{H}] = A_{1}(1 - 3C_{1}),$$
(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):

$$K_q \frac{\partial q^2}{\partial z} = 2\alpha_{\rm CB} u_\tau^3, \tag{15.44}$$

where $\alpha_{CB} = 100$ 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:

$$q^2 = (15.8\alpha_{\rm CB})^{2/3} u_{\tau}^2. \tag{15.45}$$

On the other hand, the bottom boundary condition is

$$q = 0.$$
 (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 $|z| < z_w$, where the roughness parameter z_w due to surface wind waves is given by Mellor and Blumberg (2004) as follows:

$$z_{\rm w} = \beta_{\rm w} \frac{u_{\tau}^2}{g}, \quad \beta_{\rm w} = 2.0 \times 10^5.$$
 (15.47)

In this layer, the formula for *l* is

$$l = \kappa z_{\rm w},\tag{15.48}$$

where κ is the von Karman constant ($\kappa = 0.4$). In the internal region ($|z| > z_w$), the formula for l is

$$l = \gamma \int_{z_b}^{0} |z'| q dz' / \int_{z_b}^{0} q dz', \qquad (15.49)$$

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 \rightarrow 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 19.5). 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:

$$\frac{E_{k}^{n+1} - E_{k}^{n}}{\Delta t} = \frac{1}{\Delta z_{k}} \left[\frac{K_{E_{k-\frac{1}{2}}}(E_{k-1}^{n+1} - E_{k}^{n+1})}{\Delta z_{k-\frac{1}{2}}} - \frac{K_{E_{k+\frac{1}{2}}}(E_{k}^{n+1} - E_{k+1}^{n+1})}{\Delta z_{k+\frac{1}{2}}} \right]
+ K_{M_{k}} \frac{(U_{k-\frac{1}{2}}^{n+1} - U_{k+\frac{1}{2}}^{n+1})(\tilde{U}_{k-\frac{1}{2}} - \tilde{U}_{k+\frac{1}{2}})}{\Delta z_{k}^{2}} + K_{M_{k}} \frac{(V_{k-\frac{1}{2}}^{n+1} - V_{k+\frac{1}{2}}^{n+1})(\tilde{V}_{k-\frac{1}{2}} - \tilde{V}_{k+\frac{1}{2}})}{\Delta z_{k}^{2}}
- K_{H_{k}} \frac{B_{k-\frac{1}{2}}^{n+1} - B_{k+\frac{1}{2}}^{n+1}}{\Delta z_{k}} - 2E_{k}^{n+1}q_{k}^{n}/B_{1}l_{k}^{n},$$
(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

Chapter 15 Turbulence Closure Models

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.

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 condi- tion. .false.: Start condition is that of no turbulence. See text for details.	<pre>default = l_rst_in of nml_run_ini_state</pre>

Table15.2 namelist nml_melyam_run

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

$$\frac{DU}{Dt} = -\frac{\partial}{\partial z} \langle uw \rangle - \frac{1}{\rho_0} \frac{\partial P}{\partial x} + fV, \qquad (15.51)$$

$$\frac{DV}{Dt} = -\frac{\partial}{\partial z} \langle vw \rangle - \frac{1}{\rho_0} \frac{\partial P}{\partial y} - fU, \qquad (15.52)$$

$$\frac{DB}{Dt} = -\frac{\partial}{\partial z} \langle bw \rangle + \frac{\partial R}{\partial z},$$
(15.53)

$$\frac{DE}{Dt} = -\frac{\partial}{\partial z} \langle w \Big(\frac{p}{\rho_0} + uu + vv + ww \Big) \rangle - \langle uw \rangle \frac{\partial U}{\partial z} - \langle vw \rangle \frac{\partial V}{\partial z} + \langle bw \rangle - \epsilon,$$
(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:

$$\frac{DU}{Dt} = \frac{\partial}{\partial z} \left(K_M \frac{\partial U}{\partial z} \right) - \frac{1}{\rho_0} \frac{\partial P}{\partial x} + fV, \tag{15.55}$$

$$\frac{DV}{Dt} = \frac{\partial}{\partial z} \left(K_M \frac{\partial V}{\partial z} \right) - \frac{1}{\rho_0} \frac{\partial P}{\partial y} - fU, \tag{15.56}$$

$$\frac{DB}{Dt} = \frac{\partial}{\partial z} \left(K_H \frac{\partial B}{\partial z} \right) - \frac{\partial R}{\partial z},\tag{15.57}$$

$$\frac{DE}{Dt} = \frac{\partial}{\partial z} \left(K_E \frac{\partial E}{\partial z} \right) + K_M \frac{\partial U}{\partial z} \frac{\partial U}{\partial z} + K_M \frac{\partial V}{\partial z} \frac{\partial V}{\partial z} - \left(K_H \frac{\partial B}{\partial z} \right) - \epsilon.$$
(15.58)

The surface boundary conditions are as follows:

$$K_M \frac{\partial U}{\partial z} = \frac{\tau}{\rho_0},\tag{15.59}$$

$$K_H \frac{\partial B}{\partial z} = Q_0, \tag{15.60}$$

$$K_E \frac{\partial E}{\partial z} = m u_\tau^3, \tag{15.61}$$

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:

$$K_M = Sql, \tag{15.64}$$

$$K_H = S_B q l, \tag{15.65}$$

$$K_E = S_E q l, \tag{15.66}$$

$$\epsilon = Cq^3 l^{-1}.\tag{15.67}$$

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

$$Ri_t = (Nl/q)^2.$$
 (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:

$$S/S_0 = (1 + \alpha \text{Ri}_t)^{-1/2},$$
(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):

$$Pr/Pr_0 = (1 + \beta Ri_t)^{1/2}, \tag{15.71}$$

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:

$$S = \frac{S_0}{(1 + \alpha \text{Ri}_t)^{1/2}}$$
(15.72)

$$S_B = \frac{S}{Pr} = \frac{S_0}{Pr_0(1 + \alpha \text{Ri}_t)^{1/2}(1 + \beta \text{Ri}_t)^{1/2}}.$$
(15.73)

The effect of stratification on the energy dissipation (C) is set as follows:

$$C/C_0 = (1 + \alpha \text{Ri}_t)^{1/2}.$$
 (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 ($N^2 < 0$), $K_M = K_H = 1.0 \text{ m}^2 \text{ s}^{-1}$ 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

$$l = \frac{\kappa(|z| + z_0)}{(1 + \kappa(|z| + z_0)/h)},$$
(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 19.5). 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:

$$\frac{E_{k-\frac{1}{2}}^{n+1} - E_{k-\frac{1}{2}}^{n}}{\Delta t} = \frac{1}{\Delta z_{k-\frac{1}{2}}} \left[\frac{K_{Ek-1}(E_{k-\frac{3}{2}}^{n+1} - E_{k-\frac{1}{2}}^{n+1})}{\Delta z_{k-1}} - \frac{K_{Ek}(E_{k-\frac{1}{2}}^{n+1} - E_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k}} \right] \\
+ \frac{1}{2} K_{Mk-1} \frac{(u_{k-\frac{3}{2}}^{n+1} - u_{k-\frac{1}{2}}^{n+1})(u_{k-\frac{3}{2}}^{n+1} - u_{k-\frac{1}{2}}^{n})}{\Delta z_{k-1}\Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k}\Delta z_{k-\frac{1}{2}}} \\
+ \frac{1}{2} K_{Mk-1} \frac{(v_{k-\frac{3}{2}}^{n+1} - v_{k-\frac{1}{2}}^{n+1})(v_{k-\frac{3}{2}}^{n+1} - v_{k-\frac{1}{2}}^{n})}{\Delta z_{k-1}\Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(v_{k-\frac{1}{2}}^{n+1} - v_{k+\frac{1}{2}}^{n+1})(v_{k-\frac{1}{2}}^{n} - v_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k}\Delta z_{k-\frac{1}{2}}} \\
- \frac{1}{2} K_{Hk-1} \frac{B_{k-\frac{3}{2}}^{n+1} - B_{k-\frac{1}{2}}^{n+1}}{\Delta z_{k-1}} - \frac{1}{2} K_{Hk} \frac{B_{k-\frac{1}{2}}^{n+1} - B_{k+\frac{1}{2}}^{n+1}}{\Delta z_{k}} - 2CE_{k-\frac{1}{2}}^{n+1} q_{k-\frac{1}{2}}^{n} / l_{k-\frac{1}{2}}^{n}. \tag{15.76}$$

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 $\frac{1}{2}q_{\min}^2$, where $q_{\min} = 1 \times 10^{-5} \text{ m s}^{-1}$ 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.

variable name	units	description	usage
mfnk	1	<i>m</i> 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.

Table15.3 namelist nml_nkoblm

Table15.4 namelist nml_nohkim_ru

variable name	units	description	usage
l_rst_nohkim_in	logical	.true. : Read restart files specified by nmlrs_nk_avq	default = 1_rst_in of
		and nmlrs_nk_eb for the initial condition.	nml_run_ini_state
		.false.: Start condition is that of no turbulence. See text	
		for details.	

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

$$\psi = (c_u^0)^p E^m l^n. \tag{15.77}$$

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,

$$\epsilon = (c_{\mu}^{0})^{3} \frac{E^{3/2}}{l}.$$
(15.79)

With appropriate values of the exponents p, m, n in (15.77), a number of well-known two-equation models (e.g., $k - \epsilon$ model and $k - \omega$ 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:

$$\frac{DU}{Dt} = \frac{\partial}{\partial z} \left(K_M \frac{\partial U}{\partial z} \right) - \frac{1}{\rho_0} \frac{\partial P}{\partial x} + fV, \qquad (15.80)$$

$$\frac{DV}{Dt} = \frac{\partial}{\partial z} \left(K_M \frac{\partial V}{\partial z} \right) - \frac{1}{\rho_0} \frac{\partial P}{\partial y} - fU, \tag{15.81}$$

$$\frac{DB}{Dt} = \frac{\partial}{\partial z} \left(K_H \frac{\partial B}{\partial z} \right) - \frac{\partial R}{\partial z},\tag{15.82}$$

$$\frac{DE}{Dt} = \frac{\partial}{\partial z} \left(K_E \frac{\partial E}{\partial z} \right) + P_{\rm s} + P_{\rm b} - \epsilon, \tag{15.83}$$

$$\frac{D\psi}{Dt} = \frac{\partial}{\partial z} \left(K_{\psi} \frac{\partial \psi}{\partial z} \right) + \frac{\psi}{E} (c_{\psi 1} P_{\rm s} + c_{\psi 3} P_{\rm b} - c_{\psi 2} \epsilon), \tag{15.84}$$

[†] 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

$$P_{\rm s} = -\langle uw \rangle \frac{\partial u}{\partial z} - \langle vw \rangle \frac{\partial v}{\partial z} = K_M M^2, \qquad P_{\rm b} = \langle bw \rangle = -K_H N^2, \qquad (15.85)$$

and

$$M^{2} = \left(\frac{\partial u}{\partial z}\right)^{2} + \left(\frac{\partial v}{\partial z}\right)^{2}, \qquad N^{2} = \frac{\partial B}{\partial z}$$
(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^{ψ}_{k} 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

$$c_{\mu} = \sqrt{2}S_M, \quad c'_{\mu} = \sqrt{2}S_H.$$
 (15.91)

In the case of unstable stratification ($N^2 < 0$), $K_M = K_H = 1.0 \text{ m}^2 \text{ s}^{-1}$ 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:

$$F_E = K_E \frac{\partial E}{\partial z} = \eta u_\tau^3, \tag{15.93}$$

$$F_{\psi} = K_{\psi} \frac{\partial \psi}{\partial z} = -\frac{c_{\mu}(c_{\mu}^{0})^{p}}{\sigma_{\psi}} (m\alpha + n) K^{m + \frac{1}{2}} L^{n+1}(z_{0})^{(m + \frac{1}{2})\alpha + n}.$$
(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.

$$F_E = K_E \frac{\partial E}{\partial z} = \eta u_\tau^3 (1 - a), \qquad (15.95)$$

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 $l = \kappa (z + H + z_{0b})$ is supposed. The bottom boundary conditions (z = -H) for *E* and ψ are given by flux forms as follows:

$$F_E = K_E \frac{\partial E}{\partial z} = 0, \tag{15.97}$$

$$F_{\psi} = K_{\psi} \frac{\partial \psi}{\partial z} = \frac{n(c_{\mu}^{0})^{p+1} \kappa^{n+1}}{\sigma_{\psi}} E^{m+\frac{1}{2}} (z_{0b})^{n},$$
(15.98)

where z_{0b} is the bottom roughness, which is set to be $z_{0b} = 1$ 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_{μ})^{*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 $\kappa = 0.4$, $(c_{\mu}^0)^2 = 0.3$, 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 ($P_b > 0$), 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	т	п	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:

$$\hat{c}_{\mu} = c_{\mu} (c_{\mu}^{0})^{3} = \frac{0.1070 + 0.01741\alpha_{N} - 0.00012\alpha_{M}}{1 + 0.2555\alpha_{N} + 0.02872\alpha_{M} + 0.008677\alpha_{N}^{2} + 0.005222\alpha_{N}\alpha_{M} - 0.0000337\alpha_{M}^{2}},$$
(15.99)
$$\hat{c}' = c' (c_{\mu}^{0})^{3} = \frac{0.1120 + 0.004519\alpha_{N} + 0.00088\alpha_{M}}{0.1120 + 0.004519\alpha_{N} + 0.00088\alpha_{M}}$$
(15.100)

$$\hat{c}'_{\mu} = c'_{\mu} (c^{0}_{\mu})^{3} = \frac{\alpha}{1 + 0.2555\alpha_{N} + 0.02872\alpha_{M} + 0.008677\alpha_{N}^{2} + 0.005222\alpha_{N}\alpha_{M} - 0.0000337\alpha_{M}^{2}},$$
(15.100)

where

$$\alpha_M = \frac{E^2}{\epsilon^2} M^2, \qquad \alpha_N = \frac{E^2}{\epsilon^2} N^2$$
(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

$$g_M = \frac{(c_\mu^0)^6}{2} \alpha_M, \quad g_H = \frac{(c_\mu^0)^6}{2} \alpha_N.$$
 (15.102)

These are presented in Burchard and Bolding $(2001)^{\ddagger}$. 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 19.5). 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:

$$\frac{E_{k-\frac{1}{2}}^{n+1} - E_{k-\frac{1}{2}}^{n}}{\Delta t} = \frac{1}{\Delta z_{k-\frac{1}{2}}} \Big[\frac{K_{Ek-1}(E_{k-\frac{3}{2}}^{n+1} - E_{k-\frac{1}{2}}^{n+1})}{\Delta z_{k-1}} - \frac{K_{Ek}(E_{k-\frac{1}{2}}^{n+1} - E_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k}} \Big] + P_{s} + P_{b} - (c_{\mu}^{0})^{3} E_{k-\frac{1}{2}}^{n+1}(E_{k-\frac{1}{2}}^{n})^{\frac{1}{2}} / l_{k-\frac{1}{2}}^{n}.$$
(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}_{μ} and \hat{c}'_{μ} 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:

$$\frac{\psi_{k-\frac{1}{2}}^{n+1} - \psi_{k-\frac{1}{2}}^{n}}{\Delta t} = \frac{1}{\Delta z_{k-\frac{1}{2}}} \Big[\frac{K_{\psi_{k-1}}(\psi_{k-\frac{3}{2}}^{n+1} - \psi_{k-\frac{1}{2}}^{n+1})}{\Delta z_{k-1}} - \frac{K_{\psi_{k}}(\psi_{k-\frac{1}{2}}^{n+1} - \psi_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k}} \Big] \\ + \frac{\psi_{k-\frac{1}{2}}^{n}}{\frac{E_{k-\frac{1}{2}}^{n}}{E_{k-\frac{1}{2}}^{n}}} (c_{\psi 1}P_{s} + c_{\psi 3}P_{b}) - c_{\psi 2}(c_{\mu}^{0})^{3}\psi_{k-\frac{1}{2}}^{n+1}(E_{k-\frac{1}{2}}^{n})^{\frac{1}{2}}/l_{k-\frac{1}{2}}^{n}.$$
(15.104)

The shear and buoyancy production terms P_s and P_b are discretized as follows:

$$P_{s} = \frac{1}{2} K_{Mk-1} \frac{(u_{k-\frac{3}{2}}^{n+1} - u_{k-\frac{1}{2}}^{n+1})(u_{k-\frac{3}{2}}^{n+1} - u_{k-\frac{1}{2}}^{n})}{\Delta z_{k-1} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n+1} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n+1})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n})}{\Delta z_{k} \Delta z_{k-\frac{1}{2}}} + \frac{1}{2} K_{Mk} \frac{(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n})(u_{k-\frac{1}{2}}^{n} - u_{k+\frac{1}{2}}^{n})}{\Delta z_{$$

$$P_{\rm b} = -\frac{1}{2} K_{Hk-1} \frac{B_{k-\frac{3}{2}}^{n+1} - B_{k-\frac{1}{2}}^{n+1}}{\Delta z_{k-1}} - \frac{1}{2} K_{Hk} \frac{B_{k-\frac{1}{2}}^{n+1} - B_{k+\frac{1}{2}}^{n+1}}{\Delta z_{k}}.$$
(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,

$$E = E_{\min}, \quad \psi = \psi_{\min}, \quad l = l_{stag}, \quad \text{if either } E \le E_{\min} \quad \text{or } \psi \le \psi_{\min}$$
 (15.107)

• When turbulence is stagnant, the diagnosed viscosity and diffusivity (K_M , K_H) should not exceed the background values ($K_{M \text{ bg}}$, $K_{H \text{ bg}} = 1.0 \times 10^{-6} \text{ m}^2 \text{ s}^{-1}$):

$$K_M = c_\mu (2E_{\min})^{1/2} l_{\text{stag}} \le K_{M\,\text{bg}}, \qquad K_H = c'_\mu (2E_{\min})^{1/2} l_{\text{stag}} \le K_{H\,\text{bg}}$$
 (15.108)

• Stagnant value of the turbulent length-scale (l_{stag}) is supposed to be the local vertical grid size (Δz_{local}) for numerical stability.

Table15.6 De	efault limiters for	GLS model i	in MRI.COM.
--------------	---------------------	-------------	-------------

E_{\min}	$\psi_{ m min}$	lstag
$\frac{1}{2} \left(\frac{10^{-6}}{0.1 l_{\text{stag}}} \right)^2$	$(c_{\mu}^{0})^{2}rac{E_{ m min}}{l_{ m stag}^{2/3}}$	Δz_{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 $1 \times 10^{-6} \text{ m}^2 \text{ s}$. 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.

variable name	units	description	usage
l_rst_gls_in	logical	.true. : Read restart files specified by	default = 1_rst_in of
		nmlrs_gls_(avk, avq, eke, psi, alo) for the initial condi-	nml_run_ini_state
		tion.	
		.false.: Start condition is that of no turbulence. See text for	
		details.	

Table15.7 namelist nml_gls_run

Table15.8 nam	nelist 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 5, the continuity equation for the U-cell is defined as an average of those of surrounding

T-cells (5.21). 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. (2.56) is

$$\frac{1}{\rho_0} \nabla_s p' = \frac{1}{\rho_0} \nabla p' |_{s=-H(x,y)} - \frac{1}{\rho_0} \partial_s p' \nabla H(x,y) = \frac{1}{\rho_0} \nabla p' |_{s=-H(x,y)} + \frac{g\rho'}{\rho_0} z_s \nabla H(x,y),$$
(16.1)

and the geopotential term is

$$\frac{g\rho'}{\rho_0} \nabla_s z = \frac{g\rho'}{\rho_0} \nabla z |_{s=-H(x,y)} - \frac{g\rho'}{\rho_0} z_s \nabla H(x,y).$$
(16.2)

Chapter 16 Bottom Boundary Layer (BBL)

Then, the pressure gradient for a U-cell in BBL is expressed as

$$\frac{1}{\rho_0} \nabla_s p' + \frac{g\rho'}{\rho_0} \nabla_s z = \frac{1}{\rho_0} \nabla p'|_{s=-H_{\text{BBL}}(x,y)} + \frac{g\rho'}{\rho_0} \nabla z|_{s=-H_{\text{BBL}}(x,y)}.$$
(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 ($z = z_{BBL}$) that appears in the geopotential term is separated into time-independent ($-H_{BBL}$) and time-dependent (z') part, so that

$$\frac{g\rho'}{\rho_0} \nabla z|_{s=-H_{\text{BBL}}(x,y)} = -\frac{g\rho'}{\rho_0} \nabla H_{\text{BBL}} + \frac{g\rho'}{\rho_0} \nabla z'|_{s=-H_{\text{BBL}}(x,y)}.$$
(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:

$$\Delta p'_{i,j,ktbtm} = \sum_{l=kbtm-1}^{ktbtm-1} (\rho')_{i,j,l+\frac{1}{2}} (dzt)_{i,j,l+\frac{1}{2}} + g(\rho')_{i,j,ktbtm+\frac{1}{2}} \frac{(dzt)_{i,j,ktbtm+\frac{1}{2}}}{2}.$$
 (16.5)

The gradient of pressure perturbation is computed as

$$(\nabla p_{\text{BBL}}')_{i+\frac{1}{2},j+\frac{1}{2}} = (\nabla_r p')_{i+\frac{1}{2},j+\frac{1}{2},kbtm-1} + \hat{\mathbf{x}} \frac{\frac{\Delta p'_{i+1,j+1,ktbtm} + \Delta p'_{i+1,j,ktbtm}}{2} - \frac{\Delta p'_{i+1,j+1,ktbtm} + \Delta p'_{i,j+1,ktbtm}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} + \hat{\mathbf{y}} \frac{\frac{\Delta p'_{i+1,j+1,ktbtm} + \Delta p'_{i,j+1,ktbtm}}{2} - \frac{\Delta p'_{i+1,j,ktbtm} + \Delta p'_{i,j,ktbtm}}{2}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}.$$
(16.6)

The geopotential term is computed as

$$\begin{split} \left[\frac{g\rho'}{\rho_{0}}\nabla z_{BBL}\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\ &= -\frac{g}{\rho_{0}}\left\{\left[\frac{\frac{\rho'_{i+1,j}+\rho'_{i,j}}{2}\frac{H_{BBL_{i+1,j}}-H_{BBL_{i,j}}}{\Delta x_{i+\frac{1}{2},j}} + \frac{\rho'_{i+1,j+1}+\rho'_{i,j+1}}{2}\frac{H_{BBL_{i+1,j+1}}-H_{BBL_{i,j+1}}}{\Delta x_{i+\frac{1}{2},j+1}}\hat{\mathbf{x}}\right] \\ &+ \left[\frac{\frac{\rho'_{i,j+1}+\rho'_{i,j}}{2}\frac{H_{BBL_{i,j+1}}-H_{BBL_{i,j}}}{\Delta y_{i,j+\frac{1}{2}}} + \frac{\rho'_{i+1,j+1}+\rho'_{i+1,j}}{2}\frac{H_{BBL_{i+1,j+1}}-H_{BBL_{i+1,j}}}{\Delta y_{i+1,j+\frac{1}{2}}}\hat{\mathbf{y}}\right]\right\} \\ &+ \frac{g}{\rho_{0}}\left\{\left[\frac{\frac{\rho'_{i+1,j}+\rho'_{i,j}}{2}\frac{z'_{i+1,j}-z'_{i,j}}{\Delta x_{i+\frac{1}{2},j}} + \frac{\rho'_{i+1,j+1}+\rho'_{i,j+1}}{2}\frac{z'_{i+1,j+1}-z'_{i,j+1}}{\Delta x_{i+\frac{1}{2},j+1}}}\hat{\mathbf{x}}\right] + \left[\frac{\frac{\rho'_{i,j+1}+\rho'_{i,j}}{2}\frac{z'_{i,j+1}-z'_{i,j}}{\Delta y_{i,j+\frac{1}{2}}} + \frac{\rho'_{i+1,j+1}-z'_{i+1,j}}{2}}{2}\hat{\mathbf{y}}\right]\right\}. \end{split}$$
(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 kbbl = 1 in the file configure.in. (kbbl is the number of BBL layers. Presently, only one layer (kbbl = 1) is supported).
- 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 21.2.2). These two variables have the same format as ho4, exnn. Variable ho4bbl corresponds to the thickness of the BBL, and variable exnnbbl corresponds to the number of BBL layers; only one layer is presently supported. Thus exnnbbl should be equal to or less than one.
- 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 KSGM=48 KBBL=1

The file that contains topographic information is read by the model at run time as follows (see also Section 21.2.2).

Chapter 16 Bottom Boundary Layer (BBL)

```
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 hemi-	$default = 2500 \mathrm{m}$
		sphere) above which Rayleigh damping is applied	$= 2500 \times 10^2 \mathrm{cm}$
deprfcs	cm	subjective lower limit of depth (for southern hemi-	$default = 4000 \mathrm{m}$
		sphere) above which Rayleigh damping is applied	$=4000 \times 10^2 \mathrm{cm}$

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).

variable	units	description	usage
hbblmin	cm	for sea floor depth shallower than hbbldmin, BBL	default = 0 m
		is not applied	
bbldmax	cm	if the sea floor is deeper than that of the neighbor-	$default = 10000 \mathrm{m}$
		ing point by bbldmax, the BBL is not applied	$10000 \times 10^2\mathrm{cm}$

Table16.2 Namelist nml_bblgeo_limit for limiting the range where BBL scheme is applied

16.6.3 Notes for program coding and analyses

In each vertical column, the BBL exists both at the bottom cell (k = km) and at the lowest ocean cell (k = exn(i, 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 = 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.

Chapter 17

Sea ice

This chapter describes the sea ice part. The sea ice part of MRI.COM treats formation, accretion, melting, and transfer of sea ice and snow. Heat, 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.

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 in Section 17.4, advection in Section 17.5, and ridging in Section 17.6. Adjustment on the sea ice distribution is explained in Section 17.7. Discretization issues are described in Section 17.8 and some technical issues are presented in Section 17.9. Finally, usage notes are presented in Section 17.10.

17.1 Outline

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. The sea ice part solves fractional area, heat content, thickness, and their transport of ice categorized by its thickness and dynamics of the grid-cell averaged ice pack.

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).

The fundamental property that defines the state of sea ice is the fractional area as a function of location (μ, ψ) and thickness (h_I) . The equation for this distribution function $(g(\mu, \psi, h_I))$ is

$$\frac{\partial g}{\partial t} = -\frac{\partial}{\partial h_I} (fg) - \frac{1}{h_\mu h_\psi} \Big(\frac{\partial (gh_\psi u_I)}{\partial \mu} + \frac{\partial (gh_\mu v_I)}{\partial \psi} \Big) + \chi, \tag{17.1}$$

where f is the thermodynamic growth rate of thickness, (u_I, v_I) is the velocity vector of ice pack, and χ is the rate of change of distribution function caused by ridging. The growth rate of ice thickness is computed by solving thermodynamic processes. Using this rate (f), thickness categories are remapped according to the first term on the r.h.s. of (17.1). To compute the velocity of the ice pack (u_I, v_I) , we have to solve the momentum equation. On transporting the ice distribution (second and third terms on the r.h.s.), other conservative properties such as volume and energy are also transported. Using the transported ice distribution function, the ridging process $(\chi$; fourth term on the r.h.s.) is solved. The formulation and solving procedure of each process are presented in later sections.

We discretize the thickness in several categories. If an ice pack is divided into *n* categories separated at H_n with $H_0 = 0$ m, the fractional area of each category (a_n) is defined as follows:

$$a_n = \int_{H_{n-1}}^{H_n} gdh.$$
(17.2)

Other major variables, ice and snow thickness, surface temperature, bottom temperature and salinity, and internal energy of ice, are 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 17.1 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.

	Meaning	Units	Array name for	Array name for
			each thickness	average or sum
			category	over thickness
				categories
h _I	ice thickness	m	hicen	hiceo
hs	snow thickness	m	hsnwn	hsnwo
A	area fraction (compactness)	1	aicen	a0iceo
Ah _I	grid cell average of ice thickness (volume per unit area)	m	hin	hi
Ah _s	grid cell average of snow thickness (volume per unit area)	m	hsn	hsnw
E_1	enthalpy of ice	J kg ⁻¹	qicen	-
$Ah_I E_1$	grid cell average of enthalpy	$J kg^{-1} m$	eicen	-
$\overline{T_3}$	skin temperature of upper surface	°C	tsfcin	tsfci
$\overline{T_2}$	temperature at snow-ice interface	°C	t2icen	-
T_1	temperature of ice	°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:)	-
$\overline{S_{0_I}}$	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	-
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 (= $AW_{IO} + (1 - A)W_{AO} + W_{RO} + W_{FR}$)	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
	1	ı	Continu	ed on next page

Table 17.1 Physical quantities used in the sea ice part (cf. Figure 17.1) and their variable names in the source code.

	Meaning	Units	Array name for each thickness category	Array name for average or sum over thickness
		1		categories
W _{ROsnow}	fresh water flux due to melting of snow at the upper surface of ice	$m s^{-1}$	-	Wrss
W _{FR}	fresh water flux due to formation of frazil ice	${ m ms^{-1}}$	-	wrso
u _I	zonal component of ice pack velocity	$m s^{-1}$	-	uice
v _I	meridional component of ice pack velocity	${ m ms^{-1}}$	-	vice

Table 17.1 - continued from previous page



Figure 17.1 Meaning of symbols and their locations. The left side is related to heat flux, and the right side is related to fresh water flux. A column of ice is separated into the part that originated from sea water (thickness: h_I and salinity: $S_I = 4.0 \text{ pss}$) and the part that originated from snow (thickness: h_s). The former is further divided into upper and lower halves. Thus a column of 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. For the definitions of fresh water fluxes, see Table 17.1.

17.2 Thermodynamic processes

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 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 $T(< 0)[^{\circ}C]$ 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:

$$E(T,r) = r(C_{po}T) + (1-r)(-L_F + C_{pi}T),$$
(17.3)

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. The positive brine fraction represents the fact that the salt with salinity (S_I) in sea ice exists in a liquid state. If the ice temperature is T_1 , the brine is assumed to have the same temperature, and its salinity is $S = T_1/m$, where *m* defines the freezing temperature as a function of salinity. Hence, the brine fraction of sea ice is $r = S_I/S = mS_I/T_1$. As in Mellor and Kantha (1989), the specific heat of snow is not considered.

Once sea ice enthalpy E is known, sea ice temperature T_1 is determined by solving the quadratic equation,

$$E = \frac{mS_I}{T_1}(C_{po}T_1) + (1 - \frac{mS_I}{T_1})(-L_F + C_{pi}T_1),$$
(17.4)

which gives

$$T_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a},\tag{17.5}$$

where

$$a = C_{pi}, \quad b = -L_F - E + mS_I(C_{po} - C_{pi}), \quad c = mS_I L_F.$$
(17.6)

Figure 17.2 shows that the other root is unphysical.



Figure 17.2 Sea ice enthalpy E(17.4) as a function of sea ice temperature (T_1) . Units in J kg⁻¹.

17.2.2 Formation of new sea ice

a. From sea water with temperature below the freezing point

Sea ice is formed when sea surface temperature is below the freezing point. If the temperature of the first layer of the ocean model is below the freezing point as a function of salinity, the temperature is set to the freezing point (T_{freeze}), and the heat needed to raise the temperature is regarded as the release of latent heat and is used to form new ice.

The thickness of the new ice (h_I) is computed by assuming that the total thermal energy of the first layer of the ocean model (whose layer thickness is $\Delta z_{\frac{1}{2}}$) is conserved before and after the sea ice formation:

$$\rho_o \Delta z_{\frac{1}{2}}[C_{po}T] = \rho_I h_I \{ r(C_{po}T_{\text{freeze}}) + (1-r)(-L_F + C_{pi}T_{\text{freeze}}) \} + \rho_o (\Delta z_{\frac{1}{2}} - \rho_I h_I / \rho_o)(C_{po}T_{\text{freeze}}),$$
(17.7)

where $r = mS_I/T_{\text{freeze}}$. Using the above equation, we compute the thickness of the new ice:

$$h_{I} = \frac{\rho_{o}\Delta z_{\frac{1}{2}}}{\rho_{I}(1-r)} \frac{C_{po}(T_{\text{freeze}} - T)}{L_{F} + T_{\text{freeze}}(C_{po} - C_{pi})}.$$
(17.8)

With the ice thickness known, the fractional area is determined by the following procedure:

- For a grid cell without sea ice, the first thickness is set to $h_I^{\text{new}} = 10 \text{ cm}$ and the fractional area is computed as $A = h_I / h_I^{\text{new}}$. If A > 1 (i.e., $h_I > 10$ cm), $h_I^{\text{new}} = h_I$ and A = 1. • For a grid cell where sea ice already exists, h_I is added to each category and open water.

Chapter 17 Sea ice

Note that this operation practically eliminates super-cooling in the ocean interior. Hence, the formation of frazil ice is not considered in this model.

The medium that contacts the sea ice differs at the upper (air) and lower (sea water) surfaces. Even for a grid cell with sea ice, not all the surface is covered by sea ice, i.e., there may be open water. We treat processes at each interface separately. Heat flux and thermal energy are computed by solving the balance equation at each interface and each layer in Figure 17.1.

b. Input of iceberg

Input of iceberg ($F_{iceberg}$) is given as a water mass flux per unit area (kg m⁻² s⁻¹). The thickness of the new ice (h_I) is given by

$$h_I = F_{\text{iceberg}} \Delta t / \rho_I, \tag{17.9}$$

where Δt is the unit time step. With the ice thickness known, the fractional area is determined by the same procedure used for ice created from supercooled water as explained above.

17.2.3 Air-ice interface

a. Heat flux at the upper surface of ice (Q_{AI})

The surface heat flux (Q_{AI}) is expressed as follows:

$$Q_{AI} = Q_{SI} + Q_{LI} - (1 - \alpha_I)SW - LW + \epsilon_I \sigma (T_3 + 273.15)^4,$$
(17.10)

where Q_{SI} is sensible heat flux, Q_{LI} is latent heat flux, SW is downward shortwave radiation flux, LW is downward longwave radiation flux, α_I is albedo, ϵ_I is emissivity, and σ is the Stefan-Boltzmann constant. We will now examine each component.

■ i. Shortwave: The downward shortwave radiation is represented by SW. The albedo of sea ice is α_I , which is 0.82 for cold ($T_3 < -1$ °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:

- 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 17.2 for the specific value of i_0 .
- The part penetrating into the ice (= $(1 \alpha_I)i_0SW$) is attenuated according to Beer's Law with the bulk extinction coefficient $\kappa_i = 1.4 \text{ m}^{-1}$. The attenuated part is used to warm the ice interior.
- The rest enters the ocean.

The albedos and penetration coefficients of CICE are listed on Table 17.2. The property f_{snow} is the snow fraction of the upper surface of the ice, which is expressed as follows:

$$f_{\rm snow} = \frac{h_s}{h_s + h_{snowpatch}},\tag{17.11}$$

where h_s is the snow thickness and $h_{snowpatch} = 0.02$ m.

If the ice thickness (h_I) is less than $h_{ref} = 0.5$ m, the albedo of thin ice is computed as

$$\alpha_{thinice} = \alpha_o + \beta(\alpha_{coldice} - \alpha_o), \tag{17.12}$$

where

$$\beta = \frac{\arctan(a_r h_I)}{\arctan(a_r h_{\text{ref}})}, \quad a_r = 4.0, \tag{17.13}$$

and α_o is the albedo of the ocean.

If the surface temperature T_3 becomes $-1 < T_3 < 0$ °C, the albedo of melting ice and snow is computed as

$$\alpha_{meltice} = \alpha_{thinice} - \gamma_i (T_3 + 1.0), \tag{17.14}$$

$$\alpha_{meltsnow} = \alpha_{coldsnow} - \gamma_s (T_3 + 1.0), \tag{17.15}$$

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

$$\alpha_i = \alpha_{meltice}(1 - f_{\text{snow}}) + \alpha_{meltsnow} f_{\text{snow}}.$$
(17.16)

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.

	near infra-red	visible
	(> 700 nm)	(< 700 nm)
albedo for cold snow $\alpha_{coldsnow}$ ($T_3 < -1$ °C)	0.70	0.98
albedo for cold ice $\alpha_{coldice}$ ($T_3 < -1$ °C, $h_i > 0.5$ m)	0.36	0.78
reduction rate of albedo for melting ice γ_i (-1 °C < T_3 < 0, h_i < 0.5 m)	−0.075 /°C	−0.075 /°C
reduction rate of albedo for melting snow γ_s (-1 °C < T_3 < 0, h_i < 0.5 m)	−0.15 /°C	−0.10 /°C
fraction of transparent shortwave flux through the ice surface (i_0)	0.0	$0.7 \times (1.0 - f_{\rm snow})$

Table17.2 Albedo and surface transparency of the albedo scheme of CICE.

■ ii. Longwave: The downward longwave radiation from the atmosphere is represented by *LW* in (17.10). 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 $LW_I = LW - \epsilon_I \sigma (T_3 + 273.15)^4$ as the net longwave radiation.

■ iii. Sensible heat flux: The sensible heat flux (Q_{SI} in (17.10)) is computed using a bulk formula:

$$Q_{SI} = \rho_a C_{pa} C_{HAI} U_{10} (T_3 - T_A), \qquad (17.17)$$

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, and T_3 is the ice surface temperature (Figure 17.1).

iv. Latent heat flux: The latent heat flux $(Q_{LI} \text{ in } (17.10))$ is computed using a bulk formula:

$$Q_{LI} = \rho_a L_s C_{EAI} U_{10} (q_i - q_A), \tag{17.18}$$

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 the ice surface temperature T_3 , and q_A is the specific humidity of air. Section 14.12.3 details a computing method for q_i .

Fresh water loss due to sublimation is computed as

$$W_{AI} = \rho_a C_{EAI} U_{10} (q_i - q_A) / \rho_o. \tag{17.19}$$

■ v. Bulk coefficient over sea ice: Over sea ice, it is common to set constant neutral bulk coefficients at 10 m, as described in Large and Yeager (2004). Then, bulk coefficients and wind speed at each height are calculated on the basis of stability. Large and Yeager (2004) uses

$$C_{\text{DAI10}} = C_{\text{HAI10}} = C_{\text{EAI10}} = 1.63 \times 10^{-3},$$
 (17.20)

but the default in MRI.COM uses, following Mellor and Kantha (1989),

$$C_{\text{DAI10}} = 3.0 \times 10^{-3} \tag{17.21}$$

$$C_{\text{HAI10}} = C_{\text{EAI10}} = 1.5 \times 10^{-3}.$$
 (17.22)

These coefficients are given by namelist nml_air_ice at run time. See Table 17.9.

The wind stress over sea ice can be calculated with TAUBULK option.

b. Heat flux in the snow (Q_S)

If we neglect the heat capacity of snow, the heat flux is constant within the snow layer and is computed as

$$Q_S = \frac{k_s}{h_s} (T_2 - T_3), \tag{17.23}$$

where h_s is the thickness of the snow layer and k_s is the thermal conductivity of snow.
c. Heat flux in the ice interior (Q_{I2}, Q_{IO})

In the upper half of the ice layer, the heat flux is computed as follows:

$$Q_{I2} = \frac{k_I}{h_I/2} (T_1 - T_2), \tag{17.24}$$

where k_I is the thermal conductivity of sea ice. If we neglect heat capacity of snow, $Q_S = Q_{I2}$. Using this relation, the interface temperature is computed as

$$T_2 = \frac{\frac{k_s}{h_s}T_3 + \frac{k_I}{h_I/2}T_1}{\frac{k_s}{h_s} + \frac{k_I}{h_I/2}}.$$
(17.25)

In the lower half of the ice layer, the heat flux is computed as follows:

$$Q_{IO} = \frac{k_I}{h_I/2} (T_0 - T_1). \tag{17.26}$$

d. Melting at the upper surface (W_{RO})

The melting rate at the upper surface is computed as follows: First, the surface temperature (T_3) is computed by equating the fluxes at the upper surface $(Q_{AI} = Q_S)$. If T_3 is lower than the freezing temperature $(0.0 \,^{\circ}\text{C}$ for snow and $mS_I \,[^{\circ}\text{C}]$ for sea ice), melting does not occur. If T_3 is higher than the freezing temperature, melting occurs. In this case, T_3 is set to the freezing temperature, and the interior heat flux just below the ice surface (Q_S) is recalculated. The imbalance between Q_{AI} and Q_S is used to melt snow or ice. The fresh water flux due to melting (by sign convention in this Chapter, this is always negative) is computed as

$$W_{RO} = (Q_{AI} - Q_S) / (\rho_o L_3) \tag{17.27}$$

where $L_3^{\text{snow}} = L_F$ for snow melt and

$$L_3^{\text{ice}} \equiv [E(T_3, 1) - E(T_1, r_1)] \tag{17.28}$$

for ice melt. The brine fraction for sea ice is $r_1 = mS_I/T_1$. The temperatures of the melted water are mS_I [°C] for sea ice and 0 °C for snow.

If all pre-existing snow and ice is going to be melted, that is, $(Q_S - Q_{AI})\Delta t > \rho_s h_s L_3^{\text{snow}} + \rho_I h_I L_3^{\text{ice}}$, a new interior flux (Q'_S) is computed as follows:

$$Q'_{S} = Q_{AI} + (\rho_{s}h_{s}L_{3}^{\text{snow}} + \rho_{I}h_{I}L_{3}^{\text{ice}})/\Delta t.$$
(17.29)

In this case, Q'_{S} is equated with Q_{IO} and treated as the sensible heat flux at the ice-ocean interface.

e. Notes

- All precipitation on sea ice is assumed to be snow.
- Melted water is assumed to run off to the ocean immediately, that is, there is no melt pond.

f. Solution procedure

The solution procedures are basically the same with or without snow. To be exact, the interface fluxes should be computed iteratively by adjusting surface temperature T_3 until a balance is achieved. We adopt the semi-implicit method described below. A situation without snow ($h_s = 0, T_3 = T_2$) is considered. First, the surface temperature (T_3) is computed by assuming that the fluxes on both sides are the same. Inserting (17.10) and (17.24) into $Q_{AI} = Q_{I2}$ with $T_3 \rightarrow T_3 + \delta T_3$,

$$\frac{k_I}{h_I/2} \{T_1 - (T_3 + \delta T_3)\} = Q_{LI}(T_3 + \delta T_3) + Q_{SI}(T_3 + \delta T_3) - (1 - \alpha_I)(1 - i_0)SW - LW + \epsilon_I \sigma\{(T_3 + \delta T_3) + 273.15\}^4.$$
(17.30)

By expanding the specific heat, latent heat, and black body radiation in a Taylor series, we have,

$$\frac{k_I}{h_I/2}(T_1 - (T_3 + \delta T_3)) = Q_{LI} + Q_{SI} + \frac{\partial Q_{LI}}{\partial T_3}\delta T_3 + \frac{\partial Q_{SI}}{\partial T_3}\delta T_3 - (1 - \alpha_I)(1 - i_0)SW - LW_I + 4\epsilon_I\sigma(T_3 + 273.15)^3\delta T_3.$$
(17.31)

Using this, we compute δT_3 and add it to T_3 to obtain a new temperature:

$$\delta T_3 = \frac{-Q_{SI} - Q_{LI} + (1 - \alpha_I)(1 - i_0)SW + LW_I + \frac{k_I}{h_I/2}(T_1 - T_3)}{\frac{\partial Q_{II}}{\partial T_3} + \frac{\partial Q_{SI}}{\partial T_3} + 4\epsilon_I\sigma(T_3 + 273.15)^3 + \frac{k_I}{h_I/2}},$$
(17.32)

and

$$T_3^{new} = T_3^{old} + \delta T_3, \tag{17.33}$$

where

$$\frac{\partial Q_{SI}}{\partial T_3} = \rho_a C_{pa} C_{HAI} U_{10}, \tag{17.34}$$

$$\frac{\partial Q_{LI}}{\partial T_3} = \rho_a L_s C_{EAI} U_{10} \frac{\partial q_i}{\partial T_3}.$$
(17.35)

Note that the dependency of L_s on temperature (Section 14.12.3) 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 17.11.1.

If the new surface temperature (T_3^{new}) is below the freezing point, melting does not occur. If not, T_3^{new} is set to the freezing temperature (= mS_I), and the heat flux in the ice interior is re-evaluated. The amount of melting (W_{RO}) is obtained using the imbalance:

$$Q_{AI} = -Q_{SI} - Q_{LI} + (1 - \alpha_I)(1 - i_0)SW + LW_I,$$

$$Q_{I2} = \frac{k_I}{h_I/2}(T_1 - T_3^{new}),$$

$$L_3 = [E(T_3^{new}, 1) - E(T_1, r_1)],$$

$$\Delta h_I = \frac{(Q_{AI} - Q_{I2})\Delta t}{\rho_I L_3} = W_{RO}\Delta t.$$
(17.36)

If there is snow, the above procedure is performed for the snow surface $(T_3^{new} = 0 \,^{\circ}\text{C}, L_3 = L_F)$. If all the snow melts away, the residual heat (E_{res}) is used to melt the ice:

$$E_{res} = (Q_{I2} - Q_{AI})\Delta t - h_s \rho_s L_F,$$

$$L_3 = [E(mS_I, 1) - E(T_1, r_1)],$$

$$\Delta h_I = -E_{res} / \rho_I L_3 = W_{RO}\Delta t.$$
(17.37)

17.2.4 Heat balance in the ice interior

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:

$$\rho_I h_I \Big[\frac{\partial}{\partial t} E(T_1, r_1) + u_{Ii} \frac{\partial}{\partial x_i} E(T_1, r_1) \Big] = Q_{IO} - Q_{I2} + [SW_{surface} - SW_{bottom}], \tag{17.38}$$

where

$$SW_{surface} = (1 - \alpha_I)i_0 SW, \tag{17.39}$$

$$SW_{bottom} = (1 - \alpha_I)i_0 SW \times \exp(-\kappa_i h_I).$$
(17.40)

The above equation can be solved explicitly without causing serious problems when the time step is not too long.

If the enthalpy exceeds the upper limit, $E(mS_I, 1)$, Q_{IO} is adjusted so that the enthalpy gets the upper limit value at the end of the time step (Δt),

$$Q_{IO}^{'} = \frac{\rho_I h_I [E(mS_I, 1) - E^n]}{\Delta t} - [SW_{surface} - SW_{bottom}] + Q_{I2}, \qquad (17.41)$$

where E^n is the enthalpy at the time level *n*. Q'_{IO} is used instead of Q_{IO} as the sensible heat flux at the ice-ocean interface.

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 17.1. The ice-covered area and the open water are treated separately. The solution method slightly differs from that of Mellor and Kantha (1989).

In the ice-covered area, the heat flux on the ice side of the ice-ocean interface (Q_{IO}) is computed according to (17.26). In open water, the heat flux on the air side of the air-ocean interface (Q_{AO}) is computed as in Chapter 14,

$$Q_{AO}^{all} = Q_{SO}(T) + Q_{LO}(T) - (1 - \alpha_o)SW - LW + \epsilon_o \sigma (T + 273.15)^4.$$
(17.42)

The temperature at the first level of the ocean model (T) is used. All the heat and fresh water fluxes are evaluated using the temperature and salinity at the first level of the ocean model. 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:

$$Q_{AO} = Q_{SO} + Q_{LO} - LW + \epsilon_o \sigma (T + 273.15)^4.$$
(17.43)

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). To include this effect, a certain fraction (Ψ) of the bottom melting (W_{IO}) may be used for the edge melting. The details are described in the last part of this section.

Melting and freezing occur owing to the imbalance between fluxes above (Q_{IO}, Q_{AO}) and below (F_{T_I}, F_{T_L}) the interface:

$$F_{T_I} = Q_{IO} - W_{IO}\rho_o L_o, \tag{17.44}$$

$$F_{T_L} = Q_{AO} - W_{AO} \rho_o L_o, \tag{17.45}$$

where

$$L_o \equiv [E(T,1) - E(T_1,r_1)]. \tag{17.46}$$

T is the temperature at the first level of the ocean model. The heat flux that drives the first level of the ocean model is given by

$$F_T = (AQ_{IO} + (1 - A)Q_{AO}) - W_O \rho_o L_o, \tag{17.47}$$

where

$$W_O \equiv AW_{IO} + (1 - A)W_{AO}.$$
 (17.48)

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:

$$F_{S_I} = W_{IO}(S_I - S), \tag{17.49}$$

$$F_{S_{I}} = W_{AO}(S_{I} - S). \tag{17.50}$$

Here, unlike Mellor and Kantha (1989), the salinity at the first level of the ocean model (*S*) 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.

The virtual salinity flux caused by melting and freezing at the ocean surface that drives the first level of the ocean model is given by

$$F_S = (AW_{IO} + (1 - A)W_{AO})(S_I - S).$$
(17.51)

Fluxes on the oceanic side of the interface $(F_{T_I}, F_{T_L}, F_{S_I})$ can also be obtained as the boundary conditions $(z \rightarrow 0)$ for the molecular boundary layer:

$$F_{T_I}/(\rho_o C_{po}) = -C_{T_z}(T_{0_I} - T), \qquad (17.52)$$

$$F_{T_L}/(\rho_o C_{po}) = -C_{T_z} (T_{0_L} - T), \qquad (17.53)$$

and

$$F_{S_I} = -C_{S_T}(S_{0_I} - S), \tag{17.54}$$

$$F_{S_L} = -C_{S_z}(S_{0_L} - S), (17.55)$$

where

$$C_{T_z} = \frac{u_\tau}{(P_{rt}k^{-1}\ln(-z/z_0) + B_T)},$$
(17.56)

 $u_{\tau} \equiv (\tau_{IO_x}^2 + \tau_{IO_y}^2)^{1/4} \rho_o^{-1/2}$ 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

$$B_T = b \left(\frac{z_0 u_\tau}{v}\right)^{1/2} P r^{2/3},\tag{17.57}$$

with $Pr \equiv v/\alpha_t = 12.9$. The specific values for other parameters are given in Section 17.11.2.

Parameters related to salinity are given by

$$C_{S_z} = \frac{u_\tau}{(P_{rt}k^{-1}\ln(-z/z_0) + B_S)},$$
(17.58)

and

$$B_S = b \left(\frac{z_0 u_\tau}{\nu}\right)^{1/2} S c^{2/3},\tag{17.59}$$

where $Sc \equiv v/\alpha_b = 2432$.

Following Mellor and Kantha (1989), roughness parameter z_0 is computed as follows:

$$\ln z_0 = A \ln z_{0_I} + (1 - A) \ln z_{0_L}, \tag{17.60}$$

where

$$z_{0_I} = 0.05 \frac{h_I}{h_{I \text{lim}}}, \quad h_{I \text{lim}} = 3.0 \text{ m},$$
 (17.61)

and

$$z_{0_L} = 0.016 \,\frac{\rho_o}{\rho_a} \,\frac{u_\tau^2}{g}.\tag{17.62}$$

However, the roughness parameter below ice (17.61) is also used for open water (z_{0_1}) in MRI.COM.

The above equations are solved simultaneously to obtain melting and freezing at the upper surface of the ocean under the following constraints:

$$W_O = 0, \quad \text{if} \quad A = 0, \tag{17.63}$$

and

$$T_{0_I} = mS_{0_I}$$
 and $T_{0_L} = mS_{0_L}$, if $A > 0$, (17.64)

where m defines the freezing line as a function of salinity.

Solution procedure

We first solve for S_{0_I} and S_{0_L} using (17.44), (17.49), (17.52), (17.54), (17.45), (17.50), (17.53), and (17.55):

$$S_{0I} = \frac{C_{S_z}S + (\rho_o C_{po} C_{T_z}T - Q_{IO})(S_I - S)/\rho_o L_o}{C_{S_z} + \rho_0 C_{po} C_{T_z} m(S_I - S)/\rho_o L_o},$$
(17.65)

$$S_{0_L} = \frac{C_{S_z} S + (\rho_o C_{po} C_{T_z} T - Q_{AO})(S_I - S)/\rho_o L_o}{C_{S_z} + \rho_o C_{po} C_{T_z} m(S_I - S)/\rho_o L_o}.$$
(17.66)

Using S_{0_L} and S_{0_L} , T_{0_I} and T_{0_L} are computed from (17.64), F_{T_I} and F_{T_L} are computed from (17.52) and (17.53), F_{S_I} and F_{S_L} are computed from (17.54) and (17.55), and finally W_{IO} and W_{AO} are computed from (17.44) and (17.45). Since W_{AO} should be positive by definition (only freezing is allowed in open water), W_{AO} is set to zero when $W_{AO} < 0$. In this case, $S_{0_L} = S$ from (17.50) and (17.55). Furthermore, $F_{T_L} = Q_{AO}$ in (17.45) and $T_{0_L} = T$ assuming that there is no ice effect in open water if freezing does not occur. Finally, the surface boundary conditions for the ocean model, the surface heat and virtual salt fluxes, are computed from (17.47) and (17.51). If we allow variation of volume of the first layer of the ocean

Chapter 17 Sea ice

model, virtual salt flux is not necessary. The fresh water flux (17.48) is used as a boundary condition, assuming that W_O has salinity of S_I and temperature of the first layer of the ocean model.

If $-\rho_o W_{IO}\Delta t/\rho_I > h_I$, all ice would melt away. In this case, the amount of heat needed to melt all the ice is consumed, and the residual is returned to the ocean. That is, the heat flux necessary to melt all the ice is removed from the ocean. Specifically, W'_{IO} is recomputed by

$$h_I = -\frac{\rho_o}{\rho_I} W'_{IO} \Delta t \tag{17.67}$$

and F_{T_I} is obtained as

$$F_{T_I} = Q_{IO} - W'_{IO}\rho_o L_o. (17.68)$$

If snow remains above the completely melted sea ice, the snow is also melted using ocean's heat, which is treated as the latent heat loss from the ocean (Q_{LO}) ,

$$Q_{LO} = \rho_s h_s L_F. \tag{17.69}$$

When $-\rho_o W_{IO}\Delta t/\rho_I < h_I$, reduction of the fractional area is allowed by edge melting. The procedure is as follows: The fraction used for edge melting is defined as Ψ , and $-(1 - \Psi)W_{IO}\rho_o\Delta t/\rho_I$ is used to reduce the thickness holding the fractional area. The new thickness is h'_I and $\Psi\rho_o W_{IO}\Delta t/\rho_I$ is used to reduce compactness holding the thickness:

$$A' = A(1 + \Psi \rho_o W_{IO} \Delta t / \rho_I h_I').$$
(17.70)

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.

17.2.6 Archimedes' Principle

From Archimedes' principle, the part of snow that is below freeboard absorbs sea water to become sea ice. The following equality will be achieved at equilibrium:

$$\rho_I h_I + \rho_s h_s = \rho_o h_I. \tag{17.71}$$

At the end of the time step, the above equality is checked, and if

$$\frac{\rho_I h_I + \rho_s h_s}{\rho_0} > h_I,\tag{17.72}$$

then the new ice thickness is set as:

$$h_I^{\text{new}} = \frac{\rho_I h_I + \rho_s h_s}{\rho_o}.$$
(17.73)

Since the change in snow thickness is

$$\delta h_s = -\frac{(h_I^{\text{new}} - h_I)\rho_I}{\rho_s},\tag{17.74}$$

the new snow thickness is obtained as follows:

$$h_s^{\text{new}} = h_s + \delta h_s. \tag{17.75}$$

Since the salinity of sea ice is $S_I = 4.0$ pss, salt is removed from the first layer of the ocean model:

$$S^{\text{new}} = S^{\text{old}} - S_I \frac{\delta h_s \rho_s}{\Delta z_1 \rho_o}.$$
 (17.76)

17.3 Remapping in thickness space

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, 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):

$$\frac{\partial g}{\partial t} = -\frac{\partial}{\partial h_I} (fg). \tag{17.77}$$

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 $n(h_n)$ changes from h_n^m to h_n^{m+1} (*m* is the time step index), the growth rate (f_n) at thickness h_n is represented as $f_n = (h_n^{m+1} - h_n^m)/\Delta t$. Using this, the growth rate (F_n) at the upper category boundary H_n is obtained by linear interpolation:

$$F_n = f_n + \frac{f_{n+1} - f_n}{h_{n+1} - h_n} (H_n - h_n).$$
(17.78)

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. When the fractional area is zero on both categories, $F_n = 0$. The new category boundary after thermodynamics is obtained as

$$H_n^* = H_n + F_n \Delta t. \tag{17.79}$$

Next, the thickness distribution function g within the displaced category $[H_{n-1}^*, H_n^*]$ is determined. For simplicity, we write $H_L = H_{n-1}^*$ and $H_R = H_n^*$. Function g should satisfy the following equality for fractional area (a_n) and volume (v_n) :

$$\int_{H_L}^{H_R} gdh = a_n,\tag{17.80}$$

$$\int_{H_L}^{H_R} hgdh = v_n. \tag{17.81}$$

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 (17.80) and (17.81) to yield

$$g_1 \frac{\eta_R^2}{2} + g_0 \eta_R = a_n, \tag{17.82}$$

$$g_1 \frac{\eta_R^3}{3} + g_0 \frac{\eta_R^2}{2} = a_n \eta_n, \tag{17.83}$$

where $\eta_R = H_R - H_L$ and $\eta_n = h_n - H_L$. These are algebraically solved for g_0 and g_1 as

$$g_0 = \frac{6a_n}{\eta_R^2} \Big(\frac{2\eta_R}{3} - \eta_n\Big),$$
 (17.84)

$$g_1 = \frac{12a_n}{\eta_R^3} \Big(\eta_n - \frac{\eta_R}{2} \Big).$$
(17.85)

The values of the thickness distribution function at category boundaries are given as follows:

$$g(0) = \frac{6a_n}{\eta_R^2} \left(\frac{2\eta_R}{3} - \eta_n\right),$$
(17.86)

$$g(\eta_R) = \frac{6a_n}{\eta_R^2} \Big(\eta_n - \frac{\eta_R}{3} \Big).$$
(17.87)

Equation (17.86) gives g(0) < 0 when the thickness is in the right third of the thickness range or $\eta_n > 2\eta_R/3$. Equation (17.87) 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, when the thickness is within the left third of the thickness range, a new right boundary is set at $H_C = 3h_n - 2H_L$ and g is set to zero for $[H_C, H_R]$. In this case, $\eta_R = H_C - H_L$ in (17.84) and (17.85). When the thickness is within the right third of the thickness range, a new left boundary is set at $H_C = 3h_n - 2H_R$ and g is set to zero for $[H_L, H_C]$. In this case, $\eta_R = H_R - H_C$ and $\eta_n = h_n - H_C$ in (17.84) and (17.85).

Finally, we remap ice into the original categories using the above thickness distribution function. If $H_n^* > H_n$, ice is transferred from category *n* to *n* + 1. The transferred area Δa_n and volume Δv_n are

$$\Delta a_n = \int_{H_n}^{H_n^*} g dh \tag{17.88}$$

Chapter 17 Sea ice

and

$$\Delta v_n = \int_{H_n}^{H_n^*} hg dh. \tag{17.89}$$

If $H_n^* < H_n$, ice is transferred from category n + 1 to n. The transferred area Δa_n and volume Δv_n are

$$\Delta a_n = \int_{H_n^*}^{H_n} g dh \tag{17.90}$$

and

$$\Delta v_n = \int_{H_n^*}^{H_n} hgdh. \tag{17.91}$$

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.

If ice is created in open water, the left boundary of category 1 (H_0) is moved to $F_0\Delta t$, where F_0 is the growth rate in open water. After area and volume are remapped in higher categories, ice area, volume, and energy are added to category 1.

If ice is not created in open water, H_0 remains zero, but the growth rate at the left boundary of category 1 is set to $F_0 = f_1$. If $F_0 < 0$, the fractional area of category 1 thinner than $\Delta h_0 = -F_0\Delta t$ is added to open water area. In this operation, volume and energy are invariant. The area to be added to open water is

$$\Delta a_0 = \int_0^{\Delta h_0} g dh. \tag{17.92}$$

The right boundary of the thickest category $N(H_N)$ is a function of its mean thickness h_N . When h_N is given, H_N is computed as $H_N = 3h_N - 2H_{N-1}$. It is guaranteed that g(h) > 0 for $H_{N-1} < h < H_N$ and g(h) = 0 for $H_N < h$.

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

$$\rho_I \frac{\partial}{\partial t} (Ah_I u_I) - \rho_I Ah_I f v_I = -\rho_I Ah_I g \frac{1}{h_\mu} \frac{\partial h}{\partial \mu} + F_\mu(\sigma) + A(\tau_{AI_x} + \tau_{IO_x}), \qquad (17.93)$$

$$\rho_I \frac{\partial}{\partial t} (Ah_I v_I) + \rho_I Ah_I f u_I = -\rho_I Ah_I g \frac{1}{h_{\psi}} \frac{\partial h}{\partial \psi} + F_{\psi}(\sigma) + A(\tau_{AI_y} + \tau_{IO_y}), \qquad (17.94)$$

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:

$$\overrightarrow{\tau_{AI}} = C_{DAI}\rho_a |\mathbf{U}_a - \mathbf{u}_I| [(\mathbf{U}_a - \mathbf{u}_I)\cos\theta_a + \mathbf{k} \times (\mathbf{U}_a - \mathbf{u}_I)\sin\theta_a],$$
(17.95)

where 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.

Stress at the bottom is ocean stress:

$$\overrightarrow{\tau_{IO}} = C_{DIO}\rho_o |\mathbf{U}_w - \mathbf{u}_I| [(\mathbf{U}_w - \mathbf{u}_I)\cos\theta_o + \mathbf{k} \times (\mathbf{U}_w - \mathbf{u}_I)\sin\theta_o],$$
(17.96)

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, which is set to 25° in the northern and -25° in the southern hemisphere following Hunke and Dukowicz (2002).

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

$$\frac{1}{E}\frac{\partial\sigma_{ij}}{\partial t} + \frac{1}{2\eta}\sigma_{ij} + \frac{\eta - \zeta}{4\eta\zeta}\sigma_{kk}\delta_{ij} + \frac{P}{4\zeta}\delta_{ij} = \dot{\epsilon}_{ij}, \quad i, j = 1, 2,$$
(17.97)

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 r.h.s. $(\dot{\epsilon}_{ii})$ is the strain rate tensor, expressed in Cartesian coordinates as:

$$\dot{\epsilon}_{ij} = \frac{1}{2} \Big(\frac{\partial u_{Ii}}{\partial x_j} + \frac{\partial u_{Ij}}{\partial x_i} \Big).$$
(17.98)

The divergence, tension, and shear of the strain rate are defined as follows:

$$D_D = \dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \quad D_T = \dot{\epsilon}_{11} - \dot{\epsilon}_{22}, \quad D_S = 2\dot{\epsilon}_{12}. \tag{17.99}$$

The equation for the stress tensor for $\sigma_1 = \sigma_{11} + \sigma_{22}$ and $\sigma_2 = \sigma_{11} - \sigma_{22}$ is given by

$$\frac{1}{E}\frac{\partial\sigma_1}{\partial t} + \frac{\sigma_1}{2\zeta} + \frac{P}{2\zeta} = D_D,$$
(17.100)

$$\frac{1}{E}\frac{\partial\sigma_2}{\partial t} + \frac{\sigma_2}{2\eta} = D_T,\tag{17.101}$$

$$\frac{1}{E}\frac{\partial\sigma_{12}}{\partial t} + \frac{\sigma_{12}}{2\eta} = \frac{1}{2}D_S.$$
(17.102)

In generalized orthogonal coordinates, divergence, tension, and shear of the strain rate are expressed by

$$D_D = \frac{1}{h_\mu h_\psi} \Big[\frac{\partial (h_\psi u_I)}{\partial \mu} + \frac{\partial (h_\mu v_I)}{\partial \psi} \Big], \tag{17.103}$$

$$D_T = \frac{h_{\psi}}{h_{\mu}} \frac{\partial}{\partial \mu} \left(\frac{u_I}{h_{\psi}} \right) - \frac{h_{\mu}}{h_{\psi}} \frac{\partial}{\partial \psi} \left(\frac{v_I}{h_{\mu}} \right), \tag{17.104}$$

$$D_{S} = \frac{h_{\mu}}{h_{\psi}} \frac{\partial}{\partial \psi} \left(\frac{u_{I}}{h_{\mu}}\right) + \frac{h_{\psi}}{h_{\mu}} \frac{\partial}{\partial \mu} \left(\frac{v_{I}}{h_{\psi}}\right).$$
(17.105)

The internal stress is obtained as the divergence of the internal stress tensor,

$$F_{\mu} = \frac{1}{2} \left[\frac{1}{h_{\mu}} \frac{\partial \sigma_1}{\partial \mu} + \frac{1}{h_{\mu} h_{\psi}^2} \frac{\partial (h_{\psi}^2 \sigma_2)}{\partial \mu} + \frac{2}{h_{\mu}^2 h_{\psi}} \frac{\partial}{\partial \psi} (h_{\mu}^2 \sigma_{12}) \right], \tag{17.106}$$

$$F_{\psi} = \frac{1}{2} \left[\frac{1}{h_{\psi}} \frac{\partial \sigma_1}{\partial \psi} - \frac{1}{h_{\mu}^2 h_{\psi}} \frac{\partial (h_{\mu}^2 \sigma_2)}{\partial \psi} + \frac{2}{h_{\mu} h_{\psi}^2} \frac{\partial}{\partial \mu} (h_{\psi}^2 \sigma_{12}) \right].$$
(17.107)

The viscous parameters are obtained from the ice strength and velocity as follows:

$$\zeta = \frac{P}{2\Delta},\tag{17.108}$$

$$\eta = \frac{P}{2e^2\Delta},\tag{17.109}$$

$$\Delta = \left[D_D^2 + \frac{1}{e^2} (D_T^2 + D_S^2) \right]^{1/2}.$$
(17.110)

Chapter 17 Sea ice

The pressure (strength) of the ice is a function of ice concentration and thickness:

$$P = P^* A h_I e \exp[-c^*(1 - A)], \qquad (17.111)$$

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

$$E = \frac{2E_o \rho_I A h_I}{\Delta t_e^2} \min(\Delta x^2, \Delta y^2), \qquad (17.112)$$

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:

$$v_V \left(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z}\right)\Big|_{k=0} = -\frac{A}{\rho_o}(\tau_{IO_x}, \tau_{IO_y}) + \frac{(1-A)}{\rho_o}(\tau_{AO_x}, \tau_{AO_y}).$$
(17.113)

Note that $(\tau_{IO_x}, \tau_{IO_y})$ is reversed in sign because it is defined by (17.96) 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 ((17.93) and (17.94)) and the equations for stress tensors ((17.100), (17.101), and (17.102)) 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:

$$\frac{1}{E}\frac{\sigma_1^{m+1} - \sigma_1^m}{\Delta t} + \frac{\sigma_1^{m+1}}{2\zeta^m} + \frac{P}{2\zeta^m} = D_D^m.$$
(17.114)

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:

$$\rho_{I}Ah_{I}\frac{u_{I}^{m+1}-u_{I}^{m}}{\Delta t} = \rho_{I}Ah_{I}fv_{I}^{m+1}-\rho_{I}Ah_{I}g\frac{1}{h_{\mu}}\frac{\partial h}{\partial \mu}+F_{\mu}(\sigma^{m+1})+A\tau_{AIx}$$
(17.115)
+ $Ac_{w}\rho_{o}|\mathbf{U}_{w}-\mathbf{u}_{I}^{m}|[(U_{w}-u_{I}^{m+1})\cos\theta_{o}+(V_{w}-v_{I}^{m+1})\sin\theta_{o}],$

$$\rho_{I}Ah_{I}\frac{v_{I}^{m+1}-v_{I}^{m}}{\Delta t} = -\rho_{I}Ah_{I}fu_{I}^{m+1} - \rho_{I}Ah_{I}g\frac{1}{h_{\psi}}\frac{\partial h}{\partial \psi} + F_{\psi}(\sigma^{m+1}) + A\tau_{AIy}$$
(17.116)
+ $Ac_{w}\rho_{o}|\mathbf{U}_{w}-\mathbf{u}_{I}^{m}|[(V_{w}-v_{I}^{m+1})\cos\theta_{o} - (U_{w}-u_{I}^{m+1})\sin\theta_{o}].$

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,

$$\overrightarrow{\tau_{IO}} = c_w \rho_o |\mathbf{U}_w^{n-1} - \mathbf{u}_I^m| [(\mathbf{U}_w^{n-1} - \mathbf{u}_I^{m+1}) \cos \theta_o + \mathbf{k} \times (\mathbf{U}_w^{n-1} - \mathbf{u}_I^{m+1}) \sin \theta_o].$$
(17.117)

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, 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 may be applied to remove noises.

The advection-diffusion equation for a property (α) is given by

$$\frac{\partial \alpha}{\partial t} + \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial (h_{\psi}u\alpha)}{\partial \mu} + \frac{\partial (h_{\mu}v\alpha)}{\partial \psi} \right\} = \mathcal{D}(\alpha), \tag{17.118}$$

where \mathcal{D} represents diffusion operator. The specific representation for α is A for fractional area, Ah_s for snow volume, Ah_I for ice volume, and $Ah_I E$ for ice energy.

In MPDATA, (17.118) 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

$$\begin{split} \tilde{u}_{i+\frac{1}{2},j} &= \frac{1}{\alpha^*} \left[\frac{1}{2} \left(|u^{n+\frac{1}{2}}| \Delta x - \Delta t (u^{n+\frac{1}{2}})^2 \right) \frac{\partial \alpha^*}{\partial x} - \frac{1}{2} \Delta t u^{n+\frac{1}{2}} v^{n+\frac{1}{2}} \frac{\partial \alpha^*}{\partial y} \right] \\ &\quad - \frac{1}{2} \Delta t u^{n+\frac{1}{2}} \left(\frac{\partial u^{n+\frac{1}{2}}}{\partial x} + \frac{\partial v^{n+\frac{1}{2}}}{\partial y} \right), \end{split}$$
(17.119)
$$\tilde{v}_{i,j+\frac{1}{2}} &= \frac{1}{\alpha^*} \left[\frac{1}{2} \left(|v^{n+\frac{1}{2}}| \Delta y - \Delta t (v^{n+\frac{1}{2}})^2 \right) \frac{\partial \alpha^*}{\partial y} - \frac{1}{2} \Delta t u^{n+\frac{1}{2}} v^{n+\frac{1}{2}} \frac{\partial \alpha^*}{\partial x} \right] \end{split}$$

$$-\frac{1}{2}\Delta t v^{n+\frac{1}{2}} \left(\frac{\partial u}{\partial x}^{n+\frac{1}{2}} + \frac{\partial v}{\partial y}^{n+\frac{1}{2}} \right).$$
(17.120)

This velocity is used to compute a new value using the upstream scheme starting from the above temporary value. 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.

The specific form of the harmonic-type diffusion is represented as follows:

$$\mathcal{D}(\alpha) = \frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial}{\partial\mu} \left(\frac{h_{\psi}\kappa_H}{h_{\mu}} \frac{\partial\alpha}{\partial\mu} \right) + \frac{\partial}{\partial\psi} \left(\frac{h_{\mu}\kappa_H}{h_{\psi}} \frac{\partial\alpha}{\partial\psi} \right) \right\}$$
(17.121)

$$= -\frac{1}{h_{\mu}h_{\psi}} \left\{ \frac{\partial(h_{\psi}F_{\mu}^{\alpha})}{\partial\mu} + \frac{\partial(h_{\mu}F_{\psi}^{\alpha})}{\partial\psi} \right\}$$
(17.122)

where κ_H is the horizontal diffusion coefficients. Using the flux form (17.122), the diffusion term is advanced in time with the advection term (at the "correction" step of the MPDATA scheme).

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, which is briefly summarized in this section.

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:

$$b(h) = \frac{2}{G^*} \left(1 - \frac{G(h)}{G^*}\right) \quad \text{if} \quad G(h) < G^* \tag{17.123}$$

$$= 0 \qquad \text{otherwise,} \qquad (17.124)$$

where G(h) is the area 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

$$a_{P_n} = \frac{2}{G^*} (G_n - G_{n-1}) \Big(1 - \frac{G_{n-1} + G_n}{2G^*} \Big), \tag{17.125}$$

Chapter 17 Sea ice

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. The property G_n is the area summed from category 0 to *n*. 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 water exceeds G^* ($a_0 > G^*$), then ridging does not occur.

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 $H^* = 25$ m. 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:

$$f_m^{\text{area}} = \frac{H_R - H_L}{H_{\text{max}} - H_{\text{min}}},$$
 (17.126)

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:

$$f_m^{\text{vol}} = \frac{(H_R)^2 - (H_L)^2}{(H_{\text{max}})^2 - (H_{\text{min}})^2}.$$
(17.127)

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}) is given by

$$R_{\text{net}} = \frac{C_s}{2} (\Delta - |D_D|) - \min(D_D, 0), \qquad (17.128)$$

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, $R_{\text{tot}} = \sum_{n=0}^{N} r_n$, is related to the net rate as follows:

$$R_{\text{net}} = \left[a_{P_0} + \sum_{n=1}^{N} a_{P_n} \left(1 - \frac{1}{k_n}\right)\right] R_{\text{tot}}.$$
(17.129)

Since R_{net} is computed from (17.128), R_{tot} is computed from (17.129). Thus, the area subjected to ridging from category n is computed as $a_{rn} = r_n \Delta t$ ($r_n = a_{P_n} R_{\text{tot}}$). The area after ridging is a_{rn}/k_n and the volume after ridging is $a_{rn}h_n$. Using these, the ice subjected to ridging is first removed from category n, and the ridged ice is then redistributed into each category.

In practice, we require that $a_{rn} \le a_n$. If A > 1 after ridging, R_{net} is adjusted to yield A = 1, and the ridging procedure is repeated.

17.7 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 17.8).

17.8 Discretization

17.8.1 Advection (MPDATA)

In MPDATA, tracer (α in Figure 17.3) 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:

$$F_{x}\left(\alpha_{i,j}^{n},\alpha_{i+1,j}^{n},u_{i+\frac{1}{2},j}^{n}\right) = \left[\alpha_{i,j}^{n}(u_{i+\frac{1}{2},j}^{n} + |u_{i+\frac{1}{2},j}^{n}|) + \alpha_{i+1,j}^{n}(u_{i+\frac{1}{2},j}^{n} - |u_{i+\frac{1}{2},j}^{n}|)\right]\frac{\Delta y}{2},$$

$$F_{y}\left(\alpha_{i,j}^{n},\alpha_{i,j+1}^{n},v_{i,j+\frac{1}{2}}^{n}\right) = \left[\alpha_{i,j}^{n}(v_{i,j+\frac{1}{2}}^{n} + |v_{i,j+\frac{1}{2}}^{n}|) + \alpha_{i,j+1}^{n}(v_{i,j+\frac{1}{2}}^{n} - |v_{i,j+\frac{1}{2}}^{n}|)\right]\frac{\Delta x}{2},$$
(17.130)

where

$$u_{i+\frac{1}{2},j} = \frac{1}{2} \Big(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}} \Big),$$
(17.131)

$$v_{i,j+\frac{1}{2}} = \frac{1}{2} \left(v_{i+\frac{1}{2},j+\frac{1}{2}} + v_{i-\frac{1}{2},j+\frac{1}{2}} \right).$$
(17.132)

The zonal flux is defined at the closed circle and the meridional flux is defined at the closed square in Figure 17.3. Using this, the temporary value (α^*) is computed using an upstream scheme:

$$\frac{(\alpha_{i,j}^* - \alpha_{i,j}^n)\Delta S_{i,j}}{\Delta t} = F_x \left(\alpha_{i-1,j}^n, \alpha_{i,j}^n, u_{i-\frac{1}{2},j}^{n+\frac{1}{2}} \right) - F_x \left(\alpha_{i,j}^n, \alpha_{i+1,j}^n, u_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \right) + F_y \left(\alpha_{i,j-1}^n, \alpha_{i,j}^n, v_{i,j-\frac{1}{2}}^{n+\frac{1}{2}} \right) - F_y \left(\alpha_{i,j}^n, \alpha_{i,j+1}^n, v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} \right).$$
(17.133)

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:

$$\begin{split} \tilde{u}_{i+\frac{1}{2},j} &= \frac{1}{2} \left[\frac{1}{\alpha_{i+\frac{1}{2},j}^{*(2)}} \left(|u_{i+\frac{1}{2},j}^{n+\frac{1}{2}}| \Delta x - \Delta t (u_{i+\frac{1}{2},j}^{n+\frac{1}{2}})^2 \right) \left(\frac{\partial \alpha^*}{\partial x} \right)_{i+\frac{1}{2},j} - \frac{1}{\alpha_{i+\frac{1}{2},j}^{*(6)}} \Delta t u_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \left(v^{n+\frac{1}{2}} \frac{\partial \alpha^*}{\partial y} \right)_{i+\frac{1}{2},j} \right] \\ &- \frac{1}{2} \Delta t u_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)_{i+\frac{1}{2},j}^{n+\frac{1}{2}}, \end{split}$$
(17.134)
$$\tilde{v}_{i,j+\frac{1}{2}} &= \frac{1}{2} \left[\frac{1}{\alpha_{i,j+\frac{1}{2}}^{*(2)}} \left(|v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}| \Delta y - \Delta t (v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}})^2 \right) \left(\frac{\partial \alpha^*}{\partial y} \right)_{i,j+\frac{1}{2}} - \frac{1}{\alpha_{i,j+\frac{1}{2}}^{*(6)}} \Delta t v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} \left(u^{n+\frac{1}{2}} \frac{\partial \alpha^*}{\partial x} \right)_{i,j+\frac{1}{2}} \right] \\ &- \frac{1}{2} \Delta t v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}, \end{split}$$
(17.135)

where

$$\alpha_{i+\frac{1}{2},j}^{*(2)} = \frac{1}{2} (\alpha_{i,j}^* + \alpha_{i+1,j}^*), \tag{17.136}$$

$$\alpha_{i,j+\frac{1}{2}}^{*(2)} = \frac{1}{2} (\alpha_{i,j}^* + \alpha_{i,j+1}^*), \tag{17.137}$$

$$\alpha_{i+\frac{1}{2},j}^{*(6)} = \frac{1}{8} (\alpha_{i,j-1}^* + \alpha_{i+1,j-1}^* + 2\alpha_{i,j}^* + 2\alpha_{i+1,j}^* + \alpha_{i,j+1}^* + \alpha_{i+1,j+1}^*),$$
(17.138)

$$\alpha_{i,j+\frac{1}{2}}^{*(6)} = \frac{1}{8} (\alpha_{i-1,j}^* + \alpha_{i-1,j+1}^* + 2\alpha_{i,j}^* + 2\alpha_{i,j+1}^* + \alpha_{i+1,j}^* + \alpha_{i+1,j+1}^*),$$
(17.139)

$$\left(\frac{\partial \alpha^*}{\partial x}\right)_{i+\frac{1}{2},j} = \frac{\alpha^*_{i+1,j} - \alpha^*_{i,j}}{\Delta x},\tag{17.140}$$

$$\left(\frac{\partial \alpha^*}{\partial y}\right)_{i,j+\frac{1}{2}} = \frac{\alpha^*_{i,j+1} - \alpha^*_{i,j}}{\Delta y},\tag{17.141}$$

Chapter 17 Sea ice

$$\left(v^{n+\frac{1}{2}} \frac{\partial \alpha^{*}}{\partial y} \right)_{i+\frac{1}{2},j} = \frac{1}{4} \left[v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} \left(\frac{\partial \alpha^{*}}{\partial y} \right)_{i,j+\frac{1}{2}} + v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}} \left(\frac{\partial \alpha^{*}}{\partial y} \right)_{i,j-\frac{1}{2}} \right. \\ \left. + v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} \left(\frac{\partial \alpha^{*}}{\partial y} \right)_{i+1,j+\frac{1}{2}} + v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}} \left(\frac{\partial \alpha^{*}}{\partial y} \right)_{i+1,j-\frac{1}{2}} \right],$$
(17.142)

$$\left(u^{n+\frac{1}{2}} \frac{\partial u}{\partial x} \right)_{i,j+\frac{1}{2}} = \frac{1}{4} \left[u^{n+\frac{1}{2}}_{i+\frac{1}{2},j} \left(\frac{\partial u}{\partial x} \right)_{i+\frac{1}{2},j} + u^{n+\frac{1}{2}}_{i-\frac{1}{2},j} \left(\frac{\partial u}{\partial x} \right)_{i-\frac{1}{2},j} + u^{n+\frac{1}{2}}_{i+\frac{1}{2},j+1} \left(\frac{\partial \alpha^*}{\partial x} \right)_{i+\frac{1}{2},j+1} + u^{n+\frac{1}{2}}_{i-\frac{1}{2},j+1} \left(\frac{\partial \alpha^*}{\partial x} \right)_{i-\frac{1}{2},j+1} \right],$$
(17.143)

$$\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = \frac{1}{2} \left[\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)_{i,j}^{n+\frac{1}{2}} + \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)_{i+1,j}^{n+\frac{1}{2}} \right],$$
(17.144)

$$\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2} \left[\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)_{i,j}^{n+\frac{1}{2}} + \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)_{i,j+1}^{n+\frac{1}{2}} \right],$$
(17.145)

and

$$\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)_{i,j}^{n} = \frac{\left(u_{i+\frac{1}{2},j}^{n} \Delta y_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}^{n} \Delta y_{i-\frac{1}{2},j} + v_{i,j+\frac{1}{2}}^{n} \Delta x_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}^{n} \Delta x_{i,j-\frac{1}{2}}\right)}{\Delta S_{i,j}}.$$
 (17.146)

c. Update tracer using the anti-diffusive velocity starting from the temporary value.

$$\frac{(\alpha_{i,j}^{n+1} - \alpha_{i,j}^{*})\Delta S_{i,j}}{\Delta t} = F_x \Big(\alpha_{i-1,j}^{*}, \alpha_{i,j}^{*}, \tilde{u}_{i-\frac{1}{2},j} \Big) - F_x \Big(\alpha_{i,j}^{*}, \alpha_{i+1,j}^{*}, \tilde{u}_{i+\frac{1}{2},j} \Big) + F_y \Big(\alpha_{i,j-1}^{*}, \alpha_{i,j}^{*}, \tilde{v}_{i,j-\frac{1}{2}} \Big) - F_y \Big(\alpha_{i,j}^{*}, \alpha_{i,j+1}^{*}, \tilde{v}_{i,j+\frac{1}{2}} \Big).$$
(17.147)



Figure 17.3 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 .

17.8.2 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 17.4).

Components (divergence, tension, and shear) of the strain rate tensor are expressed in a discretized form as follows:

$$(D_D)_{i,j} = \frac{1}{\Delta x_{i,j} \Delta y_{i,j}} \Big(\frac{\Delta y_{i+\frac{1}{2},j}}{2} (u_{I_{i+\frac{1}{2},j+\frac{1}{2}}} + u_{I_{i+\frac{1}{2},j-\frac{1}{2}}}) - \frac{\Delta y_{i-\frac{1}{2},j}}{2} (u_{I_{i-\frac{1}{2},j+\frac{1}{2}}} + u_{I_{i-\frac{1}{2},j-\frac{1}{2}}}) \\ + \frac{\Delta x_{i,j+\frac{1}{2}}}{2} (v_{I_{i+\frac{1}{2},j+\frac{1}{2}}} + v_{I_{i-\frac{1}{2},j+\frac{1}{2}}}) - \frac{\Delta x_{i,j-\frac{1}{2}}}{2} (v_{I_{i+\frac{1}{2},j-\frac{1}{2}}} + v_{I_{i-\frac{1}{2},j-\frac{1}{2}}}) \Big),$$

$$(D_T)_{i,j} = \frac{1}{2} \Big[\frac{\Delta y_{i,j+\frac{1}{2}}}{\Delta x_{i,j+\frac{1}{2}}} \Big(\frac{u_{I_{i+\frac{1}{2},j+\frac{1}{2}}}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{u_{I_{i-\frac{1}{2},j+\frac{1}{2}}}}{\Delta y_{i-\frac{1}{2},j+\frac{1}{2}}} \Big) + \frac{\Delta y_{i,j-\frac{1}{2}}}{\Delta x_{i,j-\frac{1}{2}}} \Big(\frac{u_{I_{i+\frac{1}{2},j-\frac{1}{2}}}}{\Delta y_{i+\frac{1}{2},j-\frac{1}{2}}} - \frac{u_{I_{i-\frac{1}{2},j-\frac{1}{2}}}}{\Delta y_{i-\frac{1}{2},j+\frac{1}{2}}} \Big) \Big] \\ - \frac{1}{2} \Big[\frac{\Delta x_{i+\frac{1}{2},j}}{\Delta y_{i+\frac{1}{2},j}} \Big(\frac{v_{I_{i+\frac{1}{2},j+\frac{1}{2}}}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{v_{I_{i+\frac{1}{2},j-\frac{1}{2}}}}{\Delta x_{i+\frac{1}{2},j-\frac{1}{2}}} \Big) + \frac{\Delta x_{i,j-\frac{1}{2}}}{\Delta y_{i,j-\frac{1}{2}}} \Big(\frac{v_{I_{i-\frac{1}{2},j+\frac{1}{2}}}}{\Delta x_{i-\frac{1}{2},j+\frac{1}{2}}} - \frac{v_{I_{i-\frac{1}{2},j-\frac{1}{2}}}}{\Delta x_{i-\frac{1}{2},j-\frac{1}{2}}} \Big) \Big],$$

$$(D_S)_{i,j} = \frac{1}{2} \Big[\frac{\Delta y_{i,j+\frac{1}{2}}}{\Delta x_{i,j+\frac{1}{2}}} \Big(\frac{v_{I_{i+\frac{1}{2},j+\frac{1}{2}}}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{v_{I_{i-\frac{1}{2},j+\frac{1}{2}}}}{\Delta y_{i-\frac{1}{2},j+\frac{1}{2}}} \Big) + \frac{\Delta y_{i,j-\frac{1}{2}}}{\Delta x_{i,j-\frac{1}{2}}} \Big(\frac{v_{I_{i+\frac{1}{2},j-\frac{1}{2}}}}{\Delta y_{i+\frac{1}{2},j-\frac{1}{2}}} - \frac{v_{I_{i-\frac{1}{2},j-\frac{1}{2}}}}{\Delta y_{i-\frac{1}{2},j+\frac{1}{2}}} \Big) \Big] \\ + \frac{1}{2} \Big[\frac{\Delta x_{i+\frac{1}{2},j}}{\Delta y_{i+\frac{1}{2},j}} \Big(\frac{u_{I_{i+\frac{1}{2},j+\frac{1}{2}}}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{u_{I_{i+\frac{1}{2},j-\frac{1}{2}}}}{\Delta x_{i+\frac{1}{2},j-\frac{1}{2}}} \Big) + \frac{\Delta x_{i,j-\frac{1}{2}}}{\Delta y_{i,j-\frac{1}{2}}} \Big(\frac{u_{I_{i-\frac{1}{2},j+\frac{1}{2}}}}{\Delta x_{i-\frac{1}{2},j+\frac{1}{2}}} - \frac{u_{I_{i-\frac{1}{2},j-\frac{1}{2}}}}{\Delta x_{i-\frac{1}{2},j-\frac{1}{2}}} \Big) \Big].$$

The internal stress is defined at velocity points and computed from stress tensor as follows:

$$(F_{\mu})_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2} \Big[\frac{1}{2} \Big(\frac{(\sigma_{1})_{i+1,j+1} + (\sigma_{1})_{i+1,j} - (\sigma_{1})_{i,j+1} - (\sigma_{1})_{i,j}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} \Big)$$

$$+ \frac{1}{2} \Big(\frac{\Delta y_{i+1,j+\frac{1}{2}}^{2} [(\sigma_{2})_{i+1,j+1} + (\sigma_{2})_{i+1,j}] - \Delta y_{i,j+\frac{1}{2}}^{2} [(\sigma_{2})_{i,j+1} + (\sigma_{2})_{i,j}]}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}^{2} \Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} \Big)$$

$$+ \Big(\frac{\Delta x_{i+\frac{1}{2},j+1}^{2} [(\sigma_{12})_{i+1,j+1} + (\sigma_{12})_{i,j+1}] - \Delta x_{i+\frac{1}{2},j}^{2} [(\sigma_{12})_{i+1,j} + (\sigma_{12})_{i,j}]}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}^{2} \Delta y_{i+\frac{1}{2},j+\frac{1}{2}}} \Big),$$

$$(17.148)$$

$$(F_{\psi})_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2} \Big[\frac{1}{2} \Big(\frac{(\sigma_{1})_{i+1,j+1} + (\sigma_{1})_{i,j+1} - (\sigma_{1})_{i+1,j} - (\sigma_{1})_{i,j}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}} \Big)$$

$$- \frac{1}{2} \Big(\frac{\Delta x_{i+\frac{1}{2},j+1}^{2} [(\sigma_{2})_{i+1,j+1} + (\sigma_{2})_{i,j+1}] - \Delta x_{i+\frac{1}{2},j}^{2} [(\sigma_{2})_{i+1,j} + (\sigma_{2})_{i,j}]}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}^{2} \Delta y_{i+\frac{1}{2},j+\frac{1}{2}}} \Big)$$

$$+ \Big(\frac{\Delta y_{i+1,j+\frac{1}{2}}^{2} [(\sigma_{12})_{i+1,j+1} + (\sigma_{12})_{i+1,j}] - \Delta y_{i,j+\frac{1}{2}}^{2} [(\sigma_{12})_{i,j+1} + (\sigma_{12})_{i,j}]}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}^{2} \Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} \Big).$$

$$(17.149)$$

17.9 Technical Notes

17.9.1 Source codes

a. Program files

The sea ice part consists of the following programs.



Figure17.4 Position of variables used by dynamics scheme.

ice_param.F90:	defines basic constants (This file is created by running the script configure on the basis of the template 'ice param F90 in'. The number of thickness categories (NUM_ICECAT) must
	be specified in configure.in before running the script.)
ice_aggr_vars.F90:	defines variables averaged over all categories for use in dynamics and ocean model
ice_cat_com.F90:	defines common variables and arrays
ice_cat_albedo.F90:	computes surface albedo and fraction of shortwave penetrating into ice
ice_cat_bulk.F90:	computes surface fluxes using bulk formula
ice_cat_nest.F90:	contains subroutines relevant to nesting
ice_date.F90:	computes date and time
ice_dyn.F90:	computes ice dynamics
ice_flux.F90:	computes air-ice interface processes
ice_grid.F90:	sets grid cells (substitution from ocean model)
ice_hist.F90:	computes and outputs history
ice_main_cat.F90:	calls subroutines (main part of the ice model)
ice_mpdata.F90:	computes advection term using MPDATA
ice_remapv.F90:	remaps ice into thickness categories after thermodynamics
ice_restart.F90:	reads and writes restart
ice_ridge.F90:	computes ridging
ice_time.F90:	manages calendar
mod_seaice_cat.F90:	computes thermodynamics and various adjustment processes

b. Structure

The ice model can be only used as a part of the ocean model. Among the subroutines included in the above programs, only subroutines in ice_main_cat.F90 are called from surface flux routines of the ocean model. Relation with the OGCM is as follows.

ogcm__prepare (ogcm.F90)

```
+ surfflux_ctl__read_restart (surfflux_ctl.F90)
      + si_read_restart (ice_main_cat.F90)
        ... reads restart and sets initial state
ogcm__run (ogcm.F90)
  I
 + LOOP_TIME_STEP
  I
      I
  L
     + part_1
  I
      + surfflux_ctl__main (surfflux_ctl.F90)
  Ι
      + surfflux_seaice (surfflux.F90)
  I
      T
      + simain (ice_main_cat.F90)
  ... the main program and calls subroutines of the ice model
  L
     T
      + surfflux_ctl__write_restart (surfflux_ctl.F90)
  + si_write_restart (ice_main_cat.F90)
  I
            ... writes restart
  ļ
    surfflux_ctl__run_end (surfflux_ctl.F90)
  +
     + si_run_end (ice_main_cat.F90)
        ... clears clock
```

Other things to be noted are:

- The sea ice model uses the forward scheme in time integration. The sea ice model is not called in the backward part of the Matsuno (Euler backward) scheme.
- History of the basic state (integrated or averaged over all categories) might be monitored by the history routine of the OGCM (specify namelist nmlhs_ice and nmlhs_iceuv).

17.9.2 Coupling with an atmospheric model

a. 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_a20 (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_a20 (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 ($T_1(tlicen)$), snow thickness ($h_s(hsnwn)$), and ice thickness ($h_I(hicen)$) given by the oceanic component (see Figure 17.1 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.

b. Correct errors due to interpolation

First, to resolve errors that arise during the transformation process, a globally constant adjustment factor for heat flux q_{adi}^{trans} is introduced so that

$$\int \Big(\sum_{n=0}^{n\text{cat}} a_n^{\text{ini}} q_n + q_{\text{adj}}^{\text{trans}}\Big) dS = Q_{\text{ocean}}^{\text{atm}},$$
(17.150)

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 $Q_{\text{ocean}}^{\text{atm}}$ is the net heat flux of the atmospheric model integrated over the ocean. The adjustment factor is given by

$$q_{\rm adj}^{\rm trans} = \left[Q_{\rm ocean}^{\rm atm} - \int \sum_{n=0}^{ncat} a_n^{\rm ini} q_n dS\right] / S_{\rm ocean},\tag{17.151}$$

where S_{ocean} is the global ocean area of the ocean model. $q_{\text{adj}}^{\text{trans}}$ is added to the net longwave radiation flux.

For fresh water, f_{adj}^{trans} is introduced so that

$$\int \Big(\sum_{n=0}^{ncat} a_n^{\text{ini}} f_n^{\text{evap}} + f^{\text{prcp}} + f^{\text{rof}} + f_{\text{adj}}^{\text{trans}}\Big) dS = F_{\text{ocean}}^{\text{atm}},$$
(17.152)

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 $F_{\text{occan}}^{\text{atm}}$ is the net fresh water of the atmospheric model integrated over the ocean. The adjustment factor is given by

$$f_{\text{adj}}^{\text{trans}} = \left[F_{\text{ocean}}^{\text{atm}} - \int \left(\sum_{n=0}^{ncat} a_n^{\text{ini}} f_n^{\text{evap}} + f^{\text{prcp}} + f^{\text{rof}}\right) dS\right] / S_{\text{ocean}},$$
(17.153)

where S_{ocean} is the global ocean area of the ocean model. $f_{\text{adj}}^{\text{trans}}$ is added to either precipitation or evaporation according to its sign.

This operation is done at subroutine adjust_fluxes_global of get_fluxs.F90.

c. 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_{adj}^{evo} is introduced so that

$$\int \Big(\sum_{n=0}^{ncat} a_n^{\text{evo}} q_n + q_{\text{adj}}^{\text{evo}}\Big) dS = \int \sum_{n=0}^{ncat} a_n^{\text{ini}} q_n dS,$$
(17.154)

where a_n^{evo} is the evolved area fraction of sea ice for category n. The adjustment factor is given as follows:

$$q_{\text{adj}}^{\text{evo}} = \left[\int \sum_{n=0}^{ncat} a_n^{\text{ini}} q_n dS - \int \sum_{n=0}^{ncat} a_n^{\text{evo}} q_n dS\right] / S_{\text{ocean}}.$$
(17.155)

In total, $q_{adj}^{trans} + q_{adj}^{evo}$ is added to the longwave radiation flux.

$$q_{\text{adj }n}^{LW} = q_{\text{atm}n}^{LW} + q_{\text{adj}}^{\text{trans}} + q_{\text{adj}}^{\text{evo}}$$
(17.156)

For fresh water, because evaporation or sublimation may be affected by the evolution of sea ice, f_{adj}^{evo} is introduced so that

$$\int \left(\sum_{n=0}^{ncat} a_n^{\text{evo}} f_n^{\text{evap}} + f_{\text{adj}}^{\text{evo}}\right) dS = \int \sum_{n=0}^{ncat} a_n^{\text{ini}} f_n^{\text{evap}} dS,$$
(17.157)

where a_n^{evo} is the evolved area fraction of sea ice for category n. The adjustment factor is given as follows:

$$f_{\text{adj}}^{\text{evo}} = \left[\int \sum_{n=0}^{n\text{cat}} a_n^{\text{ini}} f_n^{\text{evap}} dS - \int \sum_{n=0}^{n\text{cat}} a_n^{\text{evo}} f_n^{\text{evap}} dS\right] / S_{\text{ocean}}.$$
(17.158)

 f_{adj}^{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.

d. Correct errors due to solution method

Third, discrepancies due to solution methods adopted at the air-ice interface (Section 17.2.3) 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 $q_{adj}^{LW'}$ in the following manner,

$$q_{\rm adj}^{LW'} = q_{\rm adj}^{LW} - \epsilon \sigma (T_3^a)^4 + \epsilon \sigma (T_3^o)^4.$$
(17.159)

For the surface where snow or ice is melting, the amount of melting is calculated using $q_{adj}^{AI'}$ that contains $q_{adj}^{LW'}$, 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 $q_{adj}^{AI'}$ and Q_S . Q_S is different than q_{adj}^{AI} owing to the adjustment for the longwave radiation explained above. Furthermore, $q_{adj}^{AI'}$ and Q_S are not completely the same owing to the approximations used in the solution method explained in Section 17.2.3. 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, $q_{adj}^{AI} - Q_S$, should be stored.

To summarize, discrepancy of fluxes are stored in q_{diff} in the following manner,

$$q_{\text{diff}} = q_{\text{adj}}^{AI} - q_{\text{adj}}^{AI'} = q_{\text{adj}}^{LW}(T_3^a) - q_{\text{adj}}^{LW}(T_3^o), \qquad \text{if surface is melting} \qquad (17.160)$$
$$= q_{\text{adj}}^{AI} - Q_S, \qquad \text{if surface is not melting.} \qquad (17.161)$$

This is integrated over sea ice and divided by the oceanic area of the ocean model to give an offset factor q_{adj}^{stable} , which is added to the oceanic surface heat flux globally,

$$q_{\rm adj}^{\rm stable} = \int \sum_{n=1}^{ncat} a_n (q_{\rm diff})_n dS / S_{\rm ocean}.$$
(17.162)

Sampling is done at subroutine iaflux of ice_flux.F90 and flux adjustment is done at subroutine si_exit of mod_seaice_cat.F90.

17.9.3 Nesting

a. 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.

b. 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 18). 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,

$$T_1^{\rm adj} = \frac{T_2 + T_0}{2}.$$
 (17.163)

The difference between the predicted and adjusted ice energy is integrated along the parent-child boundary (Γ) as

$$Q_{\text{diff}} = \int_{\Gamma} \sum_{n=1}^{ncat} \rho_I a_n h_n [E(T_1^{\text{adj}}, r_1^{\text{adj}}) - E(T_1, r_1)] dS.$$
(17.164)

output group and interval

This is reflected as a correction factor for the ocean surface heat flux of each model, which is computed as

$$q_{\rm adj}^{thermal} = Q_{\rm diff} / S_{\rm ocean}^{\rm core}, \tag{17.165}$$

where $S_{\text{ocean}}^{\text{core}}$ is the area of the main region of the parent or child model.

17.10 Usage notes

17.10.1 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 17.3.

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 vari-	Use NAMELIST.name_model.MONITOR to specify the
	ables	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

Table17.3 Model options related to the sea ice model

17.10.2 Job parameters (namelist)

The runtime job parameters (namelist) that should be given by NAMELIST.*name_model* are listed on Tables 17.4 to 17.11. 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 17.4.

Table17.4 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
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 17.5. Note that the specification of nml_seaice_time_conf is not mandatory, necessary information will be taken from the ocean model.

variable name	group	description	usage (default value)			
nstep_seaice	<pre>nml_seaice_time_conf</pre>	time steps to be proceed	same as OGCM			
ibyri	nml_seaice_time_conf	start year of this run	same as OGCM			
ibmni	<pre>nml_seaice_time_conf</pre>	start month of this run	same as OGCM			
ibdyi	nml_seaice_time_conf	start day of this run	same as OGCM			
ibhri	<pre>nml_seaice_time_conf</pre>	start hour of this run	same as OGCM			
ibmii	<pre>nml_seaice_time_conf</pre>	start minute of this run	same as OGCM			
ibsci	<pre>nml_seaice_time_conf</pre>	start second of this run	same as OGCM			
l_force_leap	nml_seaice_calendar	forcibly specify leap year at	.false.			
		run time with l_leap				
l_leap	nml_seaice_calendar	specification of whether this	.false.			
		year is leap or not (valid when				
		l_force_leap = .true.)				

Table17.5 Namelist related to time of the sea ice model

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 17.6

Table17.6 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 veloc-	0.0
		ity	
damp_vmax	m sec ⁻¹	minimum velocity for damping	1.0

d. Diffusion

Coefficient for the harmonic-type horizontal diffusion should be given to namelist nml_seaice_diff, which is explained in Table 17.7.

Table17.7	Namelist	of diffusion	of the sea	ice model	(nml	seaice	diff)
140101/1/	1 (411101100	or annaoron	or the set	100 1110 001	(

variable name	dimension	description	usage (default value)
diff_seaice_m2ps	$m^2 sec^{-1}$	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 17.8.

Table17.8	Namelist of minimum	fractional are	ea of the sea ice	model (nml	<pre>seaice model)</pre>

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 17.9.

Table17.9	Namelist	of air-ice	bulk transf	er coefficient	(nml ai	r ice)
1001017.7	1 vaniense v	or an ice	ount transi	er coemerent	(Innit_all	r_rcc)

variable name	dimension	description	usage (default value)
chaiw	1	air-ice exchange coefficient for wind	3.0×10^{-3}
chait	1	air-ice exchange coefficient for sensible heat	1.5×10^{-3}
chaie	1	air-ice exchange coefficient for evaporation	1.5×10^{-3}

g. Restart

How to handle restart files is specified by namelists nml_seaice_run, nml_seaice_rst_in, and nml_seaice_rst_out, which is explained in Table 17.10

variable name	group	description	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
file_icecat_restart_in	nml_seaice_rst_in	restart file name of cate-	required
		gorized ice for input	
nwrt_rst_ic	nml_seaice_rst_out	the interval of time step	required
		for snap shot output	
<pre>file_icecat_restart_out_temp</pre>	nml_seaice_rst_out	core part of the file name	required
		for snap shot	

 Table17.10
 Namelist related to restart of the sea ice model

h. 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 17.11.

Table17.11 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
			Continued on next page

variable name	dimension	description	usage (default value)
hi_ref	m	the maximum ice thickness to which con-	0.50
		nection function is used	
atan_ref		the base value of the tangent hyperbolic con-	4.0
		nection function	
tsfci_t0	°C	the temperature at which ice albedo is	0.0
		equated to that of ocean	
tsfci_t1	°C	the temperature at which ice albedo is	-1.0
		started to decline to that of the ocean	
fsnow_patch	meter	thickness of snow patch on melting bare ice	0.02

Table 17.11 – continued from previous page

17.10.3 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 17.12.

variable name	group	description	usage
<pre>nstep_si_budget_interval</pre>	nml_seaice_budget	the interval of time step by	0: none, -1 : monthly
		which water budget is written	
undef_value	nml_seaice_hst	undefined value to fill ice free	real(8)
		grid point	
undef_for_land	nml_seaice_hst	undefined value to fill land	real(8)
		grid points	

Table17.12 Namelist of monitor of the sea ice model

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 17.13. 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_stal',
    interval_step = 12,
    suffix = 'minute',
    &end
```

Table17.13	List of groups	for sea ic	e monitoring
	0 1		0

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	
	a	

Continued on next page

r r r 8				
group name	usage note			
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			

Table 17.13 – continued from previous page

17.11 Appendix

17.11.1 Partial derivative of specific humidity with respect to temperature

In MRI.COM surface temperature of ice 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 as in (17.35).

Properties of moist air used by MRI.COM is based on Gill (1982) and replicated in Section 14.12.3. Using (14.117), the saturation specific humidity (q_i) is given as

$$q_i = \epsilon e'_{si} / (p_s - (1 - \epsilon) e'_{si}).$$
(17.166)

where saturation partial pressure of moist air over ice (e'_{si}) is given by (14.123) and (14.124). Using (17.166),

$$\frac{\partial q_i}{\partial t} = \frac{\epsilon p_s}{\{p_s - (1 - \epsilon)e'_{si}\}^2} \frac{\partial e'_{si}}{\partial t},\tag{17.167}$$

where $\partial e'_{si}/\partial t$ is expressed by setting $f_w = 1$ as

$$\frac{\partial e'_{si}}{\partial t} = \ln 10 \cdot 10^{g(t)} \cdot g'(t), \tag{17.168}$$

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.

17.11.2 Physical constant, parameters

Since the ice part is coded in SI units, constants and parameters are written in SI units.

a. Thermodynamics

parameter	value	variable name in MRI.COM
Thermal ice conductivity	$k_I = 2.04 \mathrm{J}\mathrm{m}^{-1}\mathrm{s}^{-1}\mathrm{K}^{-1}$	cki
Thermal snow conductivity	$k_s = 0.31 \mathrm{J}\mathrm{m}^{-1}\mathrm{s}^{-1}\mathrm{K}^{-1}$	cks
Specific heat of sea water	$C_{po} = 3990 \mathrm{J kg^{-1} K^{-1}}$	cp0
Specific heat of air	Equation (14.125)	cpair
Specific heat of ice	$C_{pi} = 2093 \mathrm{J kg^{-1} K^{-1}}$	cpi
		Continued on next page

17.11 Appendix

parameter	value	variable name in MRI.COM
Specific heat of snow	$C_{ps} = 0.0 \mathrm{J kg^{-1} K^{-1}}$	
Stefan Boltzmann constant	$\sigma = 5.67 \times 10^{-8} \mathrm{W}\mathrm{m}^{-2}\mathrm{K}^{-4}$	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	$\epsilon_I = 1.0$	eei
Emissivity of snow surface	$\epsilon_s = 1.0$	ees
neutral bulk transfer coefficient for heat	$C_{HAI} = 1.5 \times 10^{-3}$ (default)	chaih
neutral bulk transfer coefficient for moisture	$C_{EAI} = 1.5 \times 10^{-3}$ (default)	chaiw
Latent heat of fusion	$L_F = 3.347 \times 10^5 \mathrm{J kg^{-1}}$	alf
Latent heat of sublimation	Equation (14.127)	rlth
Constants for fusion phase	$m = -0.0543 ^{\circ}\text{C/pss}$	xmxm
equation: $T_f = mS + nz$	$n = -0.000759 ^{\circ}\mathrm{C}\mathrm{m}^{-1}$	xnxn
Ice roughness parameter	$z_{0_I} = 0.05 h_I / 3$	zØ
Salinity of sea ice	$S_I = 4.0 \mathrm{pss}$	si
von Karman's constant	k = 0.4	xk
Thickness/compactness diffusion of ice	$\kappa_H = 1.0 \times 10^3 \mathrm{m}^2 \mathrm{s}^{-1}$	akh
Seawater kinematic viscosity	$\nu = 1.8 \times 10^{-6} \mathrm{m^2 s^{-1}}$	anu
Seawater heat diffusivity	$\alpha_t = 1.39 \times 10^{-7} \mathrm{m}^2 \mathrm{s}^{-1}$	at
Seawater salinity diffusivity	$\alpha_b = 6.8 \times 10^{-10} \mathrm{m}^2 \mathrm{s}^{-1}$	as
Turbulent Prandtl number	$P_{rt} = 0.85$	prt
<i>b</i> in eqs (17.57),(17.59)	<i>b</i> = 3.14	ab

Table 17.14 – continued from previous page

b. Dynamics

Table17.15 Physical constants and parameters relevant to sea ice dynamics

parameter	value	variable name in MRI.COM
Density of sea water	$\rho_o = 1036 \mathrm{kg} \mathrm{m}^{-3}$	ro
Density of air	ρ_a , Equation (14.128)	roair
Density of sea ice	$\rho_I = 900 \mathrm{kg} \mathrm{m}^{-3}$	rice
Reference snow density (ratio between snow and	ρ_s/ρ_o	rdsw
water)	$= 330 \mathrm{kg}\mathrm{m}^{-3}/1036 \mathrm{kg}\mathrm{m}^{-3}$	
e-folding constant for ice pressure	$c^* = 20.0$	cstar
pressure scaling factor	$P^* = 2.75 \times 10^4 \mathrm{N m^{-2}}$	prsref
drag coefficient (air-ice)	$C_{DAI} = 3.0 \times 10^{-3} \text{ (default)}$	chaiw
drag coefficient (ice-ocean)	$C_{DIO} = 5.5 \times 10^{-3}$	cdrgiw
yield curve axis ratio	e = 2.0	elips
scaling factor for Young's modulus	$E_o = 0.25$	eyoung
water turning angle	$\theta_o = \pm 25^\circ$ (positive/negative in the	wiangl
	northern/southern hemisphere)	
air turning angle	$\theta_a = 0^\circ \text{ (not considered)}$	_

Chapter 18

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 18.1.

This chapter explains facilities available in MRI.COM for nesting. How to construct and run a set of coarse- and fineresolution models in nesting are also explained. Usage of nesting in the sea ice model and the tidal scheme is described in Section 17.9.3 and 6.5.3, respectively.

18.1 Feature

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 may be formed by connecting either the tracer points or the velocity points of the Arakawa B-grid arrangement (Figure 18.1). Table 18.1 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 18.1a). Inside the boundary, the same topography as the parent model should be given for at least one parent grid cell. 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 18.1b). The model domain needs additional two rows or columns of velocity cells outside the main region along each side boundary (hereinafter referred to as halo; the region shaded by dark green in Figure 18.1a).

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

Table18.1 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

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 18.1 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.



Figure 18.1 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.

18.2 Grid and Topography

18.2.1 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.

18.2.2 Child (fine-resolution) model

a. Horizontal grid size

The same grid size as the parent model should be used in the vertical direction. The child model is embedded in the horizontal directions. The distance between the velocity points of the parent model (i.e., T-cell) is divided equally in the child model.

Chapter 18 Nesting

The domain of a child model is constructed by taking the tracer point as the boundary of the main region and adding the halo region, even if the velocity-point boundary will be used. Here, the halo region is two rows or columns of velocity cells outside the boundaries as shown in Figure 18.1a. There should be always two marginal velocity cells outside the main region at the western and southern boundaries, even if the western or southern boundary is filled by land and does not receive data from the parent model. However, if the eastern or northern boundary is filled by land and does not receive boundary data, one marginal velocity cell will be enough.

— How to divide parent cells —

"dxt" of the parent model (U-point distance, T-point at its center) should be divided equally in the child model. There should be two marginal U-points in the child model (one cell will do if the eastern or northern boundary is filled by land) ***** parent (coarse-resolution) model ***** "ifst" (variable name is sub_region_ifirst) is the western end grid point number of the parent model to be used for interpolation. boundary (i=ifst) (i=ifst) (i=ifst+1) (i=ifst+1) ----- dxt -----> U <----- dxt -----> U |-----+----|-----|-----+------|-----|----- $T < ----- dxu -----> \dot{T} < ----- dxu -----> T$ (i=ifst) (i=ifst+1) (i=ifst+1) (i=ifst) ***** child (fine-resolution) model ***** (nesting ratio is 1:3) (1) (1) (2) (2) (3) (3)<-dxt->U<-dxt->U<-dxt->U |---+--|---+-- | - - -T < -dxu - >T < -dxu - >T < -dxu - >T(1) (1) (2) (2) (3) (3)

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 18.2).

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),

$$\phi_{ij}^{child} = \alpha_{ij}\phi_{ij}^p + (1 - \alpha_{ij})\phi_{ij}^c, \tag{18.1}$$

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 18.1a. When the boundary is at the velocity points, the weight for the parent solution should be unity in the dark green region of Figure 18.1b. Weight must be prepared separately for the tracer and velocity points. (Incidentally, a radiation boundary condition may be imposed by elaborating weight values.)



Figure 18.2 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).

18.3 Intermodel transfer

Communication between models is both downscaling and upscaling in two-way nesting, whereas only downscaling in one-way nesting.

18.3.1 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 18.3 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 (\odot in Figure 18.3). 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:

$$\phi_{i_c+m,j_c+n} = (1 - w_x)(1 - w_y)\phi_{i_p,j_p}^p + w_x(1 - w_y)\phi_{i_p+1,j_p}^p + (1 - w_x)w_y\phi_{i_p,j_p+1}^p + w_xw_y\phi_{i_p+1,j_p+1}^p,$$
(18.2)

where $w_x = (x_{i_c+m}^t - x_{i_p}^t)/(x_{i_p+1}^t - x_{i_p}^t)$ and $w_y = (y_{j_c+n}^t - y_{j_p}^t)/(y_{j_p+1}^t - x_{j_p}^t)$, x^t and y^t represent the tracer point in zonal and meridional coordinates, respectively.

b. Velocity points

Figure 18.4 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 18.5a), because only data at a single point is available for interpolation. The same problem also



Figure 18.3 Relative positions of tracer points of parent and child models (\odot and \circ , 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 18.5a) 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 18.5b). 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 18.5c). 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 $x^u(i_p - 1) \le x^u \le x^u(i_p)$ are computed as follows:

$$u^{c}(i) = w_{x}u^{p}_{i_{p}} + (1 - w_{x})u^{p}_{i_{p}-1},$$
(18.3)

$$v^{c}(i) = w_{x}v^{p}_{i_{p}} + (1 - w_{x})v^{p}_{i_{p}-1}, \qquad \text{if } x^{u}(i) < x^{t}(i_{p}) \qquad (18.4)$$

$$u^{c}(i) = \frac{\Delta z_{i_{p}-1,k}}{\Delta z_{i_{p},k}} (w_{x} u_{i_{p}}^{p} + (1 - w_{x}) u_{i_{p}-1}^{p}) + \frac{\Delta z_{i_{p},k} - \Delta z_{i_{p}-1,k}}{\Delta z_{i_{p},k}} w u_{i_{p}}^{p},$$
(18.5)

$$v^{c}(i) = \frac{\Delta z_{i_{p}-1,k}}{\Delta z_{i_{p},k}} (w_{x}v_{i_{p}}^{p} + (1 - w_{x})v_{i_{p}-1}^{p}) + \frac{\Delta z_{i_{p},k} - \Delta z_{i_{p}-1,k}}{\Delta z_{i_{p},k}} v_{i_{p}}^{p}, \qquad \text{if } x^{t}(i_{p}) < x^{u}(i) \qquad (18.6)$$

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.



Figure 18.4 Same as Figure 18.3 except for velocity point.

18.3.2 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 18.1b). 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 18.3 and 18.4), 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,

$$\phi_{i_p,j_p}^p = \frac{1}{M \times N} \sum_{i,j} \phi_{i,j}^c, \quad i_c - i_m \le i \le i_c + i_m \quad \text{and} \quad j_c - j_n \le j \le j_c + j_n, \tag{18.7}$$

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,

$$\phi_{i_p,j_p}^p = \phi_{i_c,j_c}^c, \quad x^t(i_p) = x^t(i_c) \quad \text{and} \quad y^t(j_p) = y^t(j_c).$$
 (18.8)

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.



Figure 18.5 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 (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_C^p) to which parent tracer fields (ϕ^p) are nudged,

$$\frac{\partial \phi^p}{\partial t} = -\frac{1}{\gamma} (\phi^p - \phi^p_C), \tag{18.9}$$

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.

18.4 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.

18.4.1 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:

$$\frac{\partial \phi}{\partial t} = -\nabla \cdot \mathbf{f},\tag{18.10}$$

where $\mathbf{f} = (f_x, f_y)$, which are due to either advective or diffusive fluxes. Integration (18.10) over each model's main region (Figure 18.6) gives

$$\frac{\partial \Phi^{p}_{\Omega - \omega}}{\partial t} \equiv \int_{\Omega - \omega} \frac{\partial \phi}{\partial t} dS = -\int_{\partial \Omega} \mathbf{f}^{p} \cdot \mathbf{n} dl$$
(18.11)

for parent model's main region and

$$\frac{\partial \Phi_{\omega}^{c}}{\partial t} \equiv \int_{\omega} \frac{\partial \phi}{\partial t} dS = -\int_{\partial \omega} \mathbf{f}^{c} \cdot \mathbf{n} dl$$
(18.12)

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,

$$\frac{\partial \Phi^{p}_{\Omega-\omega}}{\partial t} + \frac{\partial \Phi^{c}_{\omega}}{\partial t} = 0.$$
(18.13)

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_y^c) are area-integrated along the side boundary section over the width of a tracer cell of the parent model (Figure 18.7a),

$$F_{y}^{c}(i_{p}) = \sum_{i} f_{y}^{c}(i) \times \Delta s^{c}(i), \quad i_{c} - i_{m} \le i \le i_{c} + i_{m},$$
(18.14)

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_y^p) (Figure 18.7b),

$$F_y^p(i_p) = f_y^p(i_p) \times \Delta s^p(i_p), \tag{18.15}$$

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 $f_{y_{adj}}^c(i_p)$ given by

$$f_{y_{\text{adj}}}^{c}(i_{p}) = (F_{y}^{p}(i_{p}) - F_{y}^{c}(i_{p}))/\Delta S^{c}(i_{p}),$$
(18.16)

where

$$\Delta S^{c}(i_{p}) = \sum_{i} \Delta s^{c}(i), \quad i_{c} - i_{m} \le i \le i_{c} + i_{m}.$$
(18.17)

The adjusted flux $f_{y \mod}^c$ is given by

$$f_{y \text{ mod}}^{c}(i) = f_{y}^{c}(i) + f_{y \text{ adj}}^{c}(i_{p}), \quad i_{c} - i_{m} \le i \le i_{c} + i_{m},$$
(18.18)

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_x^c) 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 18.8),

$$F_{x \mod}^{p}(j_{p}) = f_{x \mod}^{p}(j_{p})\Delta s^{p}(j_{p}) = \sum_{j} f_{x}^{c}(j) \times \Delta s^{c}(j), \quad j_{c} - j_{n} \le j \le j_{c} + j_{n}.$$
(18.19)

 $F_{x \mod}^{p}(j_{p})$ replaces the flux calculated on the parent grid $F_{x}^{p}(j_{p})$ 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.

18.4.2 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).



Figure 18.6 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).



Figure 18.7 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 (1_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_{wadj} is used so that

$$\int_{\Omega - \omega} (f_w^p + f_{w \text{ adj}}) dS + \int_{\omega} (f_w^c + f_{w \text{ adj}}) dS = 0,$$
(18.20)

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

$$f_{w \text{ adj}} = -\left(\int_{\Omega - \omega} f_w^p dS + \int_{\omega} f_w^c dS\right) / (S_{\Omega - \omega} + S_{\omega}), \tag{18.21}$$

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 $f_{s_{adj}}$ is used so that

$$\int_{\Omega-\omega} (f_s^p + f_{s_{\text{adj}}})dS + \int_{\omega} (f_s^c + f_{s_{\text{adj}}})dS = 0,$$
(18.22)



Figure 18.8 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

$$f_{sadj} = -\left(\int_{\Omega-\omega} f_s^p dS + \int_{\omega} f_s^c dS\right) / (S_{\Omega-\omega} + S_{\omega}).$$
(18.23)

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 $f_{q_{adj}}$ is used so that

$$\int_{\Omega-\omega} (f_q^p + f_{q_{\text{adj}}})dS + \int_{\omega} (f_q^c + f_{q_{\text{adj}}})dS = \int_{\Omega} f_q^a dS,$$
(18.24)

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

$$f_{q_{\text{adj}}} = \left[\int_{\Omega} f_q^a dS - \left(\int_{\Omega - \omega} f_q^p dS + \int_{\omega} f_q^c dS\right)\right] / (S_{\Omega - \omega} + S_{\omega}).$$
(18.25)

 $f_{q_{adj}}$ 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.

18.4.3 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 (18.9) so that

$$\int_{\omega} \left[-\frac{1}{\gamma} (\phi^p - \phi_C^p) + S_{\rm adj}^{\phi} \right] dV = 0.$$
 (18.26)

The adjustment factor is given by

$$S_{\rm adj}^{\phi} = \int_{\omega} \frac{1}{\gamma} (\phi^p - \phi_C^p) dV / V_{\omega}, \qquad (18.27)$$

Chapter 18 Nesting

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).

18.5 Sponge region

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. Several methods are available for MRI.COM.

18.5.1 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.

18.5.2 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 18.9) 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.



Figure 18.9 Same as Figure 18.1b 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),

$$\frac{\partial \phi^c}{\partial t} = \nabla \cdot \kappa \nabla (\phi^c - \phi_P^c), \qquad (18.28)$$

where ϕ^c is the child model value, ϕ_P^c 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 18.10).



Figure 18.10 Same as Figure 18.1b 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.

18.6 Usage

18.6.1 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 18.2) and nml_parentmodel_package (Table 18.3) must be specified. They are read by parent/parent_common.F90 on execution. See docs/README.Namelist 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.
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)	
<pre>sub_region_jfirst</pre>	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	

Table18.2 Namelist nml_parentmodel_grid for the off-line parent model

Table18.3 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 18.4), nml_submodel_bnd_cnd (Table 18.5), and nml_submodel_package (Table 18.6) must be specified. They are read by sub/sub_common.F90 on execution. See docs/README.Namelist 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.

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)	
<pre>sub_region_jlast</pre>	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	

Table18.4 Namelist nml_submodel_grid for the off-line child model

Table18.5 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)	

Table18.6	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	

18.6.2 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',
&nam_scup_pre var_put='ALATTC', var_get='ALATTC',
                                                                  dst_get='ALL' /
dst_get='ALL' /
dst_get='ALL' /
 &nam_scup_pre var_put='ALATTC',
&nam_scup_pre var_put='ALONUC',
&nam_scup_pre var_put='ALATUC',
                                         var_get='ALATTC',
var_get='ALONUC',
var_get='ALATUC',
                                                                   dst_get='ALL'
                                                                  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)
    comp_put ='BAROCLIP_GLOBAL', comp_get ='PAPCC
grid_put ='RCLIPP_GLOBAL', comp_get ='PAPCC
 &nam_scup model_put='GLOBAL'
    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_get ='BC3DP_GLOBAL_TA',
     grid_put ='BC3DS_NP01_TC'
     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 18.7) must be specified. They are read by nest_scup_par.F90 on execution. See docs/README.Namelist for details. If sponge layer is used for the parent model (flg_sponge = .true.), the twodimensional 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 18.8) and nml_par_visc (Table 18.9).

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_volflux_cnsv	logical	Impose conservation of volume flux at u(v)star	default = .false.
		for child model	
flg_iceflux_cnsv	logical	Impose conservation of sea ice flux at u(v)star	default = .false.
		for child model	
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	default = .false.
		for child model	
flg_send_trchdif_p2c	logical	Send tracer lateral diffusion flux from parent to	default = .false., choose
		child	either p2c or c2p or neither
flg_send_trchdif_c2p	logical	Send tracer lateral diffusion flux from child to	default = .false., choose
		parent	either p2c or c2p or neither
flg_send_trcidif_p2c	logical	Send tracer isopycnal diffusion flux from parent	default = .false., choose
		to parent	either p2c or c2p or neither
flg_send_trcidif_c2p	logical	Send tracer isopycnal diffusion flux from child	default = .false., choose
		to parent	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 up-	
		scale region	

Table18.7	Namelist nml	_nest_	_par fo	r the	on-line	parent	model
-----------	--------------	--------	---------	-------	---------	--------	-------

Table18.8 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	

 Table18.9
 Namelist nml_par_visc for the on-line parent model

variable name	description	usage
file_sponge_par_visc	2D distribution of horizontal viscocity just out-	
	side the boundary	

d. Runtime specification for child model

At run time, namelist nml_submodel_grid (Table 18.10) and nml_nest_sub (Table 18.11) must be specified. They are read by nest_scup_sub.F90 on execution. See docs/README.Namelist 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 18.12), nml_sub_visc (Table 18.13) (if flg_sponge_sub = .true.), nml_sub_isopyc (Table 18.14), and nml_sub_ssh_diff (Table 18.15).

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)	
<pre>sub_region_jfirst</pre>	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

 $Table 18.10 \quad Namelist \verb"nml_submodel_grid" for the on-line child model$

 $Table 18.11 \quad Namelist \verb"nml_nest_sub" for the on-line child model"$

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	default = .false.
	_	for the child model	
flg_iceflux_cnsv	logical	Impose conservation of sea ice flux at u(v)star	default = .false.
	_	for the child model	
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	default = .false.
	-	for the child model	
			Continued on next page

variable name	units	description	usage
flg_send_trchdif_p2c	logical	Send tracer lateral diffusion flux from parent to	default = .false., choose
		child	either p2c or c2p or neither
flg_send_trchdif_c2p	logical	Send tracer lateral diffusion flux from child to	default = .false., choose
		parent	either p2c or c2p or neither
flg_send_trcidif_p2c	logical	Send tracer isopycnal diffusion flux from parent	default = .false., choose
		to parent	either p2c or c2p or neither
flg_send_trcidif_c2p	logical	Send tracer isopycnal diffusion flux from child	default = .false., choose
		to parent	either p2c or c2p or neither
nest_depth	integer	Nesting is operated from surface to nest_depth	
		level	

Table 18.11 - continued from previous page

Table18.12 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	valid if flg_sponge_sub = .true.
	inside the boundary	

Table18.13 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	<pre>valid if flg_sponge_sub = .true.</pre>
	inside the boundary	

Table18.14 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 diffu-	
	sion mask is read from file	
file_isopyc_mask_sub	2D mask of isopycnal and GM diffusion	diffusion coefficients are deter-
		mined as in the normal case

Table18.15 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	

18.7 Program structure

The program structure of an off-line parent model is as follows.

ogcm__prepare
|
+- parent_ctl__ini (initialize the parent model)
|
+- parent_ctl__prepare (output boundary data of the initial state)

```
ogcm__run
|
```

```
+- parent_ctl__write (output boundary data at specified interval)
|
+- parent_ctl__fin_run (close files)
```

The program structure of an off-line child model is as follows.

```
ogcm__ini
 (initialize the child model)
 +- sub_ctl__ini
     +- sub_ctl__prepare (open boundary files, etc.)
ogcm__run
 +- part_1
 +- surface_ctl__predict
 T
          +- surface_predict
  L
               +- surfce_integ__predict
          1
  +- DO LOOP_N
          |
          +- surface_sub__input (replace barotropic boundary data)
          L
  +- surface_sub__input (replace time-filtered data)
  +- surface_bcli_predict_h (l_global_local_cnsv_ssh = .true.)
  Т
               +- surface_sub__input (replace time-filtered data again)
 Т
 +- part_2
     (read/receive baroclinic boundary data
      +- sub_ctl__input_side
                               after "tracer_ctl__implicit_z" and before "strat_adjust_ctl__main" )
      T
      +- sub_ctl__input_side_vd (read/receive viscosity/diffusivity data
                               after mixed layer model)
```

Part VI

Miscellaneous

Chapter 19

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 19.1. Sections 19.2 and 19.3 describe applying finite difference methods of time and space derivatives in differential equations. Considerations in finite-difference methods for advection-diffusion equations are discussed in Section 19.4. An implicit method for solving the diffusion equation is described in Section 19.5. Section 19.6 summarizes discretization schemes used by MRI.COM.

Durran (1999) treats the basics of finite difference methods for solving differential equations of advection and diffusion in geophysical fluid dynamics.

19.1 Diffusion equation

As an example, consider an initial-boundary value problem expressed by a one-dimensional diffusion equation (heat conductive equation),

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2}.$$
(19.1)

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

$$T(x,t) = \sum_{m=0}^{\infty} f_m e^{-\kappa k_m^2 t} \sin(k_m x),$$
(19.2)

where

$$f_m = \frac{2}{L} \int_0^L f(x) \sin(k_m x) dx, \quad k_m = \frac{m\pi}{L}.$$
 (19.3)

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,

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \kappa \frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{\Delta x^2},$$
(19.4)

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 (19.1) in the limit $\Delta t \rightarrow 0$, $\Delta x \rightarrow 0$ (**consistency**).

If the initial distribution is assumed to be $f(x) = T_0 \sin k_1 x$, the solution of the finite difference equation (19.4) for $t = t_n$ is

$$T_j^n = \lambda^n T_0 \sin k_1 x_j, \tag{19.5}$$

where

$$\lambda = 1 - \frac{2\kappa\Delta t}{\Delta x^2} (1 - \cos k_1 \Delta x). \tag{19.6}$$

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 $\Delta t \rightarrow 0$, $\Delta x \rightarrow 0$ (**convergence**).

To summarize, the finite difference method that satisfies consistency, stability, and convergence is the necessary condition for an accurate solution.

19.2 Finite difference expressions for time derivatives

The following four finite difference expressions are employed for the time derivatives in MRI.COM:

forward :
$$\frac{T^{n+1} - T^n}{\Delta t} = F(T^n)$$
(19.7)

backward :
$$\frac{T^{n+1} - T^n}{\Delta t} = F(T^{n+1})$$
 (19.8)

Matsuno :
$$\frac{T^{*n+1} - T^n}{\Delta t} = F(T^n), \quad \frac{T^{n+1} - T^n}{\Delta t} = F(T^{*n+1})$$
 (19.9)

leap-frog :
$$\frac{T^{n+1} - T^{n-1}}{2\Delta t} = F(T^n).$$
 (19.10)

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

$$T_i^n = (T_a \lambda_a^n + T_b \lambda_b^n) \sin k_1 x_j, \tag{19.11}$$

where

$$\lambda_a = -\frac{-\alpha + \sqrt{\alpha^2 + 4}}{2}, \quad \lambda_b = -\frac{-\alpha - \sqrt{\alpha^2 + 4}}{2} \quad (\alpha \equiv \frac{4\kappa\Delta t}{\Delta x^2}(1 - \cos k_1 \Delta x)). \tag{19.12}$$

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 (19.6). In such a case, the backward scheme is used for vertical diffusion and viscosity (implicit method; see Section 19.5). 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 19.5).

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. This interval can be changed at run time using a namelist parameter (nstep_matsuno_interval) of namelist nml_time_step (Table 21.6). It should be noted that the Matsuno scheme needs twice as many numerical operations as the forward and leap-frog schemes.

19.3 Finite difference expression for space derivatives

Let us consider a one-dimensional advection equation,

$$\frac{\partial T}{\partial t} = -u\frac{\partial T}{\partial x},\tag{19.13}$$

where u is a constant velocity. The solution is

$$T(x,t) = T(x - ut, 0).$$
(19.14)

Using the leap-frog scheme for time differencing, the finite difference equation can be written as follows:

$$\frac{T_{j}^{n+1} - T_{j}^{n-1}}{2\Delta t} = -u \frac{T_{j+\frac{1}{2}}^{n} - T_{j-\frac{1}{2}}^{n}}{\Delta x},$$
(19.15)

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 $T_{j-\frac{1}{2}}^n$ using a value at a single or multiple grid points. The following are two simple and frequently used formulations,

upstream finite difference :
$$T_{j-\frac{1}{2}}^{n} = T_{j-1}^{n} (u > 0), \quad T_{j-\frac{1}{2}}^{n} = T_{j}^{n} (u < 0),$$
 (19.16)

central finite difference :
$$T_{j-\frac{1}{2}}^{n} = \frac{T_{j-1}^{n} + T_{j}^{n}}{2}$$
. (19.17)

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 (19.15) is

$$\frac{T_j^{n+1} - T_j^{n-1}}{2\Delta t} = -u \frac{T_{j+1}^n - T_{j-1}^n}{2\Delta x}.$$
(19.18)

Assuming the solution to be $T(x, t) = \tau(t)e^{-ikx}$,

$$\tau^{n+1} = \tau^{n-1} + 2i\alpha\tau^n$$
, where $\alpha \equiv \frac{u\Delta t}{\Delta x}\sin k\Delta x$. (19.19)

It is stable (neutral) if $|\alpha| \le 1$. To be stable for any wave number,

$$\left|\frac{u\Delta t}{\Delta x}\right| \le 1 \tag{19.20}$$

must be satisfied (**CFL condition**). However, if $\tau^n = \tau^0 e^{-in\Delta\theta}$,

$$\Delta \theta = -\sin^{-1}[\mu \sin k \Delta x] \text{(where } \mu \equiv \frac{u \Delta t}{\Delta x}\text{)}.$$
(19.21)

Expanding the r.h.s. by a Taylor expansion we obtain

$$\Delta \theta \simeq -\mu \sin k\Delta x - \frac{1}{6}(\mu \sin k\Delta x)^{3}$$

$$\simeq -\mu k\Delta x + \frac{\mu (k\Delta x)^{3}}{6} - \frac{\mu^{3} (k\Delta x)^{3}}{6}$$

$$= -\mu k\Delta x \left\{ 1 - \frac{(k\Delta x)^{2}}{6}(1 - \mu^{2}) \right\}.$$
 (19.22)

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 7.

Using the upstream finite difference, the finite difference equation (19.15) is

$$\frac{T_j^{n+1} - T_j^{n-1}}{2\Delta t} = -u \frac{T_j^n - T_{j-1}^n}{\Delta x}.$$
(19.23)

Expanding the r.h.s. by a Taylor expansion we obtain

$$-u\frac{\partial T}{\partial x} + \frac{u\Delta x}{2}\frac{\partial^2 T}{\partial x^2} + O(\Delta x^2).$$
(19.24)

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

$$T_{j-\frac{1}{2}}^{n} = \frac{-T_{j-2}^{n} + 6T_{j-1}^{n} + 3T_{j}^{n}}{8}(u > 0), \quad T_{j-\frac{1}{2}}^{n} = \frac{3T_{j-1}^{n} + 6T_{j}^{n} - T_{j+1}^{n}}{8}(u < 0).$$
(19.25)

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 8.

19.4 Finite differencing of advection-diffusion equation

According to the above restriction, when the advection-diffusion equation (Eqs. (8.1) and (9.1)) 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:

$$\frac{T^{n+1} - T^{n-1}}{2\Delta t} = -\mathcal{A}(T^n) + \mathcal{D}(T^{n-1}),$$
(19.26)

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.

19.5 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 step 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,

$$\frac{(T^{n+1} - T^{n-1})}{2\Delta t} = -\mathcal{A}(T^n) + \mathcal{D}_H(T^{n-1}) + \mathcal{D}_V(T^{n+1}).$$
(19.27)

Putting all the terms involving T^{n+1} on the l.h.s.,

$$T^{n+1} - 2\Delta t \mathcal{D}_V(T^{n+1}) = T^{n-1} + 2\Delta t (-\mathcal{A}(T^n) + \mathcal{D}_H(T^{n-1}))$$
(19.28)

is obtained, which is an algebraic equation for T^{n+1} . The finite difference form is rewritten specifically as

$$T_{k}^{n+1} - 2\Delta t \frac{1}{\Delta z_{k}} \Big(\kappa_{k-\frac{1}{2}} (T_{k-1}^{n+1} - T_{k}^{n+1}) / \Delta z_{k-\frac{1}{2}} - \kappa_{k+\frac{1}{2}} (T_{k}^{n+1} - T_{k+1}^{n+1}) / \Delta z_{k+\frac{1}{2}} \Big)$$

$$= T_{k}^{n-1} + 2\Delta t (-\mathcal{A}(T_{k}^{n}) + \mathcal{D}_{H}(T_{k}^{n-1})).$$
(19.29)

By putting

$$a = \frac{2\Delta t \kappa_{k-\frac{1}{2}}}{\Delta z_k \Delta z_{k-\frac{1}{2}}}, \quad b = 1 + a + c, \quad c = \frac{2\Delta t \kappa_{k+\frac{1}{2}}}{\Delta z_k \Delta z_{k+\frac{1}{2}}},$$
(19.30)

we get,

$$-aT_{k-1}^{n+1} + bT_k^{n+1} - cT_{k+1}^{n+1} = T_k^{n-1} + 2\Delta t (-\mathcal{A}(T_k^n) + \mathcal{D}_H(T_k^{n-1})).$$
(19.31)

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

The l.h.s. has the form of the tri-diagonal matrix.

19.5.1 A solution of tri-diagonal matrix

In general, simultaneous linear equations for n variables with tri-diagonal matrix coefficients

$$\begin{pmatrix} B_{1} & C_{1} & & & \\ A_{2} & B_{2} & C_{2} & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & B_{n-1} & C_{n-1} \\ & & & & A_{n} & B_{n} \end{pmatrix} \begin{pmatrix} X_{1} \\ X_{2} \\ \vdots \\ X_{n-1} \\ X_{n} \end{pmatrix} = \begin{pmatrix} D_{1} \\ D_{2} \\ \vdots \\ D_{n-1} \\ D_{n} \end{pmatrix}$$
(19.33)

are solved using the Thomas method, which is modified from LU decomposition,

$$P_1 = C_1 / B_1 \tag{19.34}$$

$$Q_1 = D_1 / B_1 \tag{19.35}$$

$$P_k = \frac{C_k}{B_k - A_k P_{k-1}} \qquad (2 \le k \le n-1)$$
(19.36)

$$Q_{k} = \frac{D_{k} - A_{k}Q_{k-1}}{B_{k} - A_{k}P_{k-1}} \qquad (2 \le k \le n)$$

$$Y_{k} = O \qquad (19.37)$$

$$Y_{k} = O \qquad (19.38)$$

$$A_n = Q_n \tag{19.38}$$

$$X_k = Q_k - P_k X_{k+1} \qquad (1 \le k \le n-1).$$
(19.39)

19.6 Summary of schemes used by MRI.COM

MRI.COM employs centered difference schemes for spatial discretization except for the tracer advection terms.

Figure 19.1 shows the schematic of the relation between components of the governing equations in terms of time integration. The situation depicted here is the advancement in time from n to n + 1, represented by the thick red arrows. The main components employ the leap-frog scheme, inserting Matsuno (Euler-backward) scheme with a specified interval. Sea ice and turbulence closure employs forward scheme.

Colored arrows represent data transfer among the components in advancing in time using the leap-frog time $(n \rightarrow n + 1)$. (a) State at the starting time (n - 1) is used for computing surface fluxes. (b) Surface current at the starting time and SST and SSS of the present time is given to the sea ice component. (c) Flux from the sea ice component is averaged with that of the previous time step for use in the ocean component (d) for conservation. (e) Horizontal velocity field of the present time is used to compute vertical velocity and tracer transport. (f) Diagnosed vertical velocity is used immediately in the advection schemes of momentum and tracers. (g) Temperature and salinity at the present time is used to compute the pressure gradient term. (h) Vertically integrated forcing term computed by the baroclinic component is used by the barotropic component. (i) The barotropic transport is returned to the baroclinic component and used immediately as the vertically averaged velocity for computing the total velocity. For turbulent closure scheme, states at both present (j) and new (k) time level are used to estimate shear production and buoyancy loss of turbulent kinetic energy. (l) Turbulent kinetic energy at the present time is used to compute vertical mixing coefficient. Blue arrows represent the transfer of extant data obtained by the previous time steps. Green arrows represent the immediate use of the new data computed in this time step. These relations constrain the order of operations within a time step. The source side of the green arrows must be calculated earlier than the recipient side.



Figure 19.1 Schematic of the relation between components of the governing equations in terms of time integration. The situation depicted here is the advancement in time from n to n + 1, which is represented by the thick red arrows. Blue arrows represent the transfer of extant data obtained by the previous time steps. Green arrows represent the immediate use of data computed at this time step. These relations constrain the order of operations within a time step. See text for the detailed explanation.

Chapter 20

Generalized orthogonal curvilinear coordinate grids

This chapter introduces generalized orthogonal curvilinear coordinates and presents related calculus.

20.1 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 20.2. 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 20.3. We can use the functions, $(\lambda, \phi) \iff (\mu, \psi)$, 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 20.4. The vector operation in generalized orthogonal coordinates is concisely described in Section 20.5.

20.2 Generation of orthogonal coordinate system using conformal mapping

We designate the plane that touches the sphere at the North Pole as S_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 S_N and the orthogonality is preserved on the reverse transformation (Figure 20.1). Moreover, if S_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 S_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:



Figure 20.1 Schematic illustration of a Polar stereographic projection (a conformal transformation in the general sense between the sphere and \mathbf{S}_N).

1. From a point (λ, ϕ) on the sphere to a point z on S_N (polar stereographic projection). Defining colatitude $\phi' = \pi/2 - \phi$,

$$z = \tan\left(\frac{\phi'}{2}\right)e^{i\lambda},\tag{20.1}$$

where the origin of \mathbf{S}_{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}_{N} :

$$\zeta = M_{\rm C}(z). \tag{20.2}$$

3. From a point ζ on \mathbf{S}_N to a point (μ, ψ) on the sphere (reverse polar stereographic projection).

$$\mu = \arg(\zeta), \tag{20.3}$$

$$\psi' = 2 \arctan |\zeta|, \tag{20.4}$$

$$\psi = \pi/2 - \psi'. \tag{20.5}$$

Functions $\lambda(\mu, \psi)$ and $\phi(\mu, \psi)$ are obtained by reversing the above procedure. Defining $\psi' = \pi/2 - \psi$,

$$\zeta = \tan\left(\frac{\psi'}{2}\right)e^{i\mu},\tag{20.6}$$

$$z = M_{\rm C}^{-1}(\zeta),$$
 (20.7)

and

$$\lambda = \arg(z), \tag{20.8}$$

$$\phi' = 2 \arctan |z|, \tag{20.9}$$

$$\phi = \pi/2 - \phi'. \tag{20.10}$$

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 S_N . That is,

$$\zeta = M_{\rm C}(z) = \frac{(z-a)(b-c)}{(c-a)(b-z)},$$
(20.11)

Chapter 20 Generalized orthogonal curvilinear coordinate grids

where the three complex numbers a, b, and c expressed by

$$a = \tan\left(\frac{\phi'_a}{2}\right)e^{i\lambda_a}, \quad b = \tan\left(\frac{\phi'_b}{2}\right)e^{i\lambda_b}, \quad c = \tan\left(\frac{\phi'_c}{2}\right)e^{i\lambda_c}, \tag{20.12}$$

correspond to the three geographic coordinate grid points $(\lambda_a, \phi_a), (\lambda_b, \phi_b)$, 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_lat in degree) that should be specified in namelist nml_poles (Table 3.2).

When TRIPOLAR option is specified, the parameters are set to $\phi_a = \phi_b = 64^{\circ}$ N, $\lambda_a = 80^{\circ}$ E, and $\lambda_b = 100^{\circ}$ 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 20.2).

When JOT option is specified, the Joukowski conversion is used as a conformal transformation on S_N . That is,

$$z = M_C^{-1}(\zeta) = \left(\zeta + \frac{\psi_0'^2}{\zeta}\right) e^{i\mu_0}.$$
(20.13)

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 20.2 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.



Figure 20.2 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.

20.3 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

$$z = f(\zeta), \quad z = x + iy, \quad \zeta = u + iv,$$
 (20.14)

$$f'(\zeta) = \frac{\partial x}{\partial u} + i\frac{\partial y}{\partial u} = \frac{\partial y}{\partial v} - i\frac{\partial x}{\partial v}.$$
(20.15)

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

$$\tan \theta = \left[\frac{\partial y}{\partial x}\right]_{v_0} = \left[\frac{\partial y}{\partial u} / \frac{\partial x}{\partial u}\right]_{v_0},$$

$$\theta = \arg(f'(\zeta_0)). \tag{20.16}$$

then (see Figure 20.3),



Figure 20.3 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 $-\mu_0 \ (\mu_0 \to v_0)$, and at point z_0 the meridian $(\lambda = \lambda_0)$ meets the curve $(v = v_0)$ at the angle $\lambda_0 - \theta \ (v_0 \to \lambda_0)$. 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:

$$\alpha = -\mu_0 + \lambda_0 - \theta = \lambda_0 - \mu_0 - \arg(f'(\zeta_0)).$$
(20.17)

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 ($\tau_x \cos \alpha - \tau_y \sin \alpha$, $\tau_x \sin \alpha + \tau_y \cos \alpha$).

20.4 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 20.4). 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.

Chapter 20 Generalized orthogonal curvilinear coordinate grids



Figure 20.4 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),

$$G'_{i,i} = G_{I,J}$$

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:

$$H_{M,N} = \frac{1}{A_{HM,N}} \sum_{m,n} A_{H'm,n} H'_{m,n},$$
(20.18)

where $A_{HM,N}$ 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,

$$H_{M,N} = \sum_{I,J} w(M, N, I, J) G_{I,J}.$$
(20.19)

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 (20.19) 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

$$\sum_{I,J} A_{GI,J} = \sum_{i,j} A'_{G'i,j} = \sum_{m,n} A'_{H'm,n} = \sum_{M,N} A_{HM,N}.$$

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 (20.19), to avoid the time-consuming flux conversions in the model calculation.

20.5 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

$$\delta \mathbf{x} = h_{\mu} \delta \mu \mathbf{e}_{\mu} + h_{\psi} \delta \psi \mathbf{e}_{\psi} + h_r \delta r \mathbf{e}_r, \qquad (20.20)$$

where basis vectors \mathbf{e}_{μ} , \mathbf{e}_{ψ} , and \mathbf{e}_{r} are mutually orthogonal unit vectors, and h_{μ} , h_{ψ} , and h_{r} are scale factors. Defining

$$\nabla = \frac{\mathbf{e}_{\mu}}{h_{\mu}}\frac{\partial}{\partial\mu} + \frac{\mathbf{e}_{\psi}}{h_{\psi}}\frac{\partial}{\partial\psi} + \frac{\mathbf{e}_{r}}{h_{r}}\frac{\partial}{\partial r},\tag{20.21}$$

the gradient of scalar $A(\mu, \psi, r)$ is

$$\nabla A = \frac{\mathbf{e}_{\mu}}{h_{\mu}} \frac{\partial A}{\partial \mu} + \frac{\mathbf{e}_{\psi}}{h_{\psi}} \frac{\partial A}{\partial \psi} + \frac{\mathbf{e}_{r}}{h_{r}} \frac{\partial A}{\partial r}, \qquad (20.22)$$

and the divergence of vector $\mathbf{A} = A_{\mu}\mathbf{e}_{\mu} + A_{\psi}\mathbf{e}_{\psi} + A_{r}\mathbf{e}_{r}$ is

$$\nabla \cdot \mathbf{A} = \frac{1}{h_{\mu}h_{\psi}h_{r}} \left[\frac{\partial(h_{\psi}h_{r}A_{\mu})}{\partial\mu} + \frac{\partial(h_{r}h_{\mu}A_{\psi})}{\partial\psi} + \frac{\partial(h_{\mu}h_{\psi}A_{r})}{\partial r} \right].$$
(20.23)

The *r* component of curl**A** is

$$\frac{1}{h_{\mu}h_{\psi}}\left[\frac{\partial(h_{\psi}A_{\psi})}{\partial\mu} - \frac{\partial(h_{\mu}A_{\mu})}{\partial\psi}\right].$$
(20.24)

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

$$\mathbf{a} \cdot \nabla A_{\mu} + \frac{A_{\psi}}{h_{\mu}h_{\psi}} \left(a_{\mu}\frac{\partial h_{\mu}}{\partial \psi} - a_{\psi}\frac{\partial h_{\psi}}{\partial \mu} \right) + \frac{A_{r}}{h_{r}h_{\mu}} \left(a_{\mu}\frac{\partial h_{\mu}}{\partial r} - a_{r}\frac{\partial h_{r}}{\partial \mu} \right).$$
(20.25)

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 v is

$$\mathbf{v} = u\mathbf{e}_{\lambda} + v\mathbf{e}_{\phi} + w\mathbf{e}_{r},\tag{20.26}$$

where \mathbf{e}_{λ} , \mathbf{e}_{ϕ} , 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,

$$\nabla A = \frac{\mathbf{e}_{\lambda}}{r\cos\phi} \frac{\partial A}{\partial\lambda} + \frac{\mathbf{e}_{\phi}}{r} \frac{\partial A}{\partial\phi} + \mathbf{e}_{r} \frac{\partial A}{\partial r}, \qquad (20.27)$$

where

$$\nabla = \frac{\mathbf{e}_{\lambda}}{r\cos\phi}\frac{\partial}{\partial\lambda} + \frac{\mathbf{e}_{\phi}}{r}\frac{\partial}{\partial\phi} + \mathbf{e}_{r}\frac{\partial}{\partial r}.$$
(20.28)

For vector $\mathbf{A} = A_{\lambda} \mathbf{e}_{\lambda} + A_{\phi} \mathbf{e}_{\phi} + A_r \mathbf{e}_r$, the divergence is

$$\nabla \cdot \mathbf{A} = \frac{1}{r \cos \phi} \left[\frac{\partial A_{\lambda}}{\partial \lambda} + \frac{\partial (\cos \phi A_{\phi})}{\partial \phi} \right] + \frac{\partial (r^2 A_r)}{r^2 \partial r}.$$
 (20.29)

and the r component of curlA is

$$[\operatorname{curl}\mathbf{A}]_r = \frac{1}{r\cos\phi} \left[\frac{\partial A_\phi}{\partial\lambda} - \frac{\partial(\cos\phi A_\lambda)}{\partial\phi} \right].$$
(20.30)

Chapter 20 Generalized orthogonal curvilinear coordinate grids

The λ component of $(\mathbf{a} \cdot \nabla)\mathbf{A}$ is

$$[(\mathbf{a} \cdot \nabla)\mathbf{A}]_{\lambda} = \mathbf{a} \cdot \nabla A_{\lambda} - \frac{A_{\phi} a_{\lambda} \tan \phi}{r} + \frac{A_{r} a_{\lambda}}{r}.$$
(20.31)

The Coriolis force in generalized orthogonal coordinates (μ, ψ, r) is given as

$$2\Omega \times \mathbf{v} = (2\Omega_{\psi}w - 2\Omega_{r}v)\mathbf{e}_{\mu} + (2\Omega_{r}u - 2\Omega_{\mu}w)\mathbf{e}_{\psi} + (2\Omega_{\mu}v - 2\Omega_{\psi}u)\mathbf{e}_{r},$$
(20.32)

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 21

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 4.4 (MRICOM-4_4_beta01-20170118) and some contents presented in this chapter may not be used for the latest version. It is recommended that users refer to README.First, README.Options, README.Namelist, README.Monitor, and README.Restart 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 21.1 and Table 21.1).
- **Input data:** Grid spacing, topography, and surface forcing etc., to be read at run time (Section 21.2 and Table 21.2).
- **Restart file:** Explanation of restart files (Section 21.3).
- Execution: Explanation of the runtime parameters that control time-integration (Section 21.4).
- **Post process:** A description of monitor files (Section 21.5).

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 21.6

21.1 Model setup

21.1.1 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. Following is an example of **configure.in**.

- An example configure.in for Global tripolar $1^{\circ} \times 0.5^{\circ}$ grid model -

DEFAULT_OPTIONS="IDEALAGE ICE SIDYN CALALBSI SMAGOR VIS9P DIFAJS GLS VVDIMP SOMADVEC ISOPYCNAL HFLUX TAUBULK WFLUX RUNOFF SFLUXR LWDOWN BULKNCAR BULKITER CYCLIC ZSTAR BBL TRIPOLAR PARALLEL" # NAME_MODEL='GLOBAL' IMUT=364 JMUT=368 KM=51 KSGM=51 KSGM=51 KBBL=1 NPARTX=8 NPARTY=4 NUM_ICECAT=5 NUMTRC_P=1 Additional parameters are required for some particular model options. Those are listed in Table 21.1 (see also README.Options).

option name	variable name	description	
always required	IMUT, JMUT, KM	zonal/meridional/vertical grid number	
	NAME_MODEL	name of the model (default = "tmOGCM")	
	KSGM	The number of variable layers near the sea surface. KSGM must equal KM for	
		ZSTAR option. See Chapter 6.	
	NSFMRGN	the number of side-boundary ghost cells to reduce the communication cost	
		in parallel computation (see Ishizaki and Ishikawa, 2006)	
BBL	KBBL	the number layers of bottom boundary layer model; must be 1	
PARALLEL	NPARTX, NPARTY	the number of zonally/meridionally partitioned region for a computation us-	
		ing parallel processors: the number of parallel processes should be NPARTX	
		× NPARTY	
passive tracers	NUMTRC_P	only when any passive tracer is calculated (NUMTRC_P ≥ 1)	
ICE	NUM_ICECAT	the number of thickness categories of sea ice	

Table21.1 Model parameters to be set in configure.in

21.1.2 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 function of compile.sh is to create param.F90, icecat/ice_param.F90, and Makefile from configure.in, param.F90.in, icecat/ice_param.F90.in, and Makefile.in respectively, by running configure and to execute 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, the parameter values defined in configure.in replace those in param.F90.in to create new param.F90.

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 ./compile.sh clean when any compile option in configure.in is rewritten.

21.2 Preparation of input data files for execution

According to user's specification of model options and job parameters, the data files listed on Table 21.2 should be prepared along with the files that must be always prepared. See Section 21.3 for how to handle restart files.

subject	file name specified in	read from
	(NAMELIST.name_model)	
runtime job parameters	NAMELIST.name_model	each package
specification about monitoring	NAMELIST.name_model.MONITOR	history.F90 etc.
variable horizontal grid spacing	<pre>file_dxdy_tbox_deg(nml_horz_grid)</pre>	gridm.F90
variable vertical grid spacing	<pre>file_dz_cm (nml_vert_grid)</pre>	gridm.F90
grid cell area and line elements	<pre>file_scale (nml_grid_scale)</pre>	gridm.F90
topography	file_topo (nml_topo)	topo.F90
reference data for tracers	<pre>trcref(_surf)_conf%file_data</pre>	restore_cond.F90
restoring coefficient for tracers	<pre>rstcoef(_surf)_conf%file_data</pre>	restore_cond.F90
surface forcing	file_data(nml_force_data)	force.F90
surface forcing grid	<pre>file_data_grid (nml_force_data)</pre>	force.F90
restart files	nmlrs_element, element being elements	history.F90

Table21.2 Main input data files and their related program files. Here, *name_model* represents the specific name given as NAME MODEL in configure.in.

21.2.1 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 gridm.F90, and the file file_scale is not necessary.

21.2.2 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 $1^{\circ} \times 0.5^{\circ}$ model is shown in Figure 21.1. 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.



21.2.3 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.

21.2.4 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 the calendar subroutine. 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.

	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	$\mathrm{cm}\mathrm{s}^{-1}$	if TAUBULK	
Y-ward wind speed	V-wind	$\mathrm{cm}\mathrm{s}^{-1}$	if TAUBULK	
Downward shortwave radiation	ShortWave	$erg s^{-1} cm^{-2} =$		
		$10^{-3} \mathrm{W}\mathrm{m}^{-2}$		
Downward longwave radiation	LongWave	$erg s^{-1} cm^{-2}$	Net longwave by default	
			(SST is required).	
			Downward if LWDOWN	
Sea surface temperature	SST	°C	unnecessary if LWDOWN	
Surface air temperature	TempAir	°C		
Surface air specific humidity	SphAir	1		
Scalar wind speed	ScalarWind	$\mathrm{cms^{-1}}$	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	

Table21.3 Surface data to be read from namelist nml_force_data.

21.3 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. Table 21.4 lists restart variables of MRI.COM as well as their namelist that specifies their restart file attributes.

variable name	namelist block	model options
temperature	nmlrs_t	
salinity	nmlrs_s	
x velocity	nmlrs_u	
y velocity	nmlrs_v	
sea surface height	nmlrs_ssh	
x barotropic transport	nmlrs_uml	
y barotropic transport	nmlrs_vml	
x transport due to SSH diffusion	nmlrs_ssh_dflx_x	
y transport due to SSH diffusion	nmlrs_ssh_dflx_y	
horizontally averaged density and pressure used	nmlrs_density	required if CALPP
for the equation of state		
	·	Continued on next page

Table21.4 Restart variables and their namelist block that specifies their restart file attributes.

variable name	namelist block	model options
vertical diffusivity coefficient	nmlrs_vmix_avd	VVDIMP
vertical viscosity coefficient	nmlrs_vmix_avm	VVDIMP
passive tracers	nmlrs_ptrc	NUMTRC_P > 0
x tidal transport	nmlrs_tide_um	TIDE
y tidal transport	nmlrs_tide_vm	TIDE
tidal sea surface height	nmlrs_tide_ssh	TIDE
vertical diffusion coefficient of turbulence kinetic	nmlrs_my_avq	MELYAM
energy		
turbulence velocity scale	nmlrs_my_q	MELYAM
turbulence length scale	nmlrs_my_alo	MELYAM
vertical diffusion coefficient of turbulence kinetic	nmlrs_nk_avq	NOHKIM
energy		
turbulence kinetic energy	nmlrs_nk_eb	NOHKIM
vertical diffusion coefficient of turbulence kinetic	nmlrs_gls_avk	GLS
energy		
vertical diffusion coefficient of generic variable	nmlrs_gls_avp	GLS
turbulence kinetic energy	nmlrs_gls_eke	GLS
generic variable	nmlrs_gls_psi	GLS
turbulent length scale	nmlrs_gls_alo	GLS
moments of second order moment advection	nml_somadv	SOMADVEC and
scheme		adv_scheme%name = "som"
		in nml_tracer_data
		for any tracer
sea ice initial state	nml_seaice_rst_in	ICE
sea ice final state	nml_seaice_rst_out	ICE

Table 21.4 – continued from previous page

21.3.1 Restart for the ocean model

For restart variables of the main part of ocean model, the attributes of their restart files are given by the namelist blocks that start with "nmlrs_". The element names that follow are t, s, u, v, ssh, etc. as listed on Table 21.4. Required elements depend on model options. See docs/README.Restart for more information. Namelists must be written in NAMELIST.name_model (default) as follows:

—— An example namelist for temperature restart files —

```
&nmlrs_t
    fname = 'result/rs_t',
/
```

where fname 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.

For passive tracers, namelist nmlrs_ptrc should be repeated as many times as the number of passive tracers (=numtrc_p) with a right order. The order is determined at tracer_vars.F90. An example for a set of biogeochemical tracers is shown in Section 11.6.

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 21.6) also has tools for visualization in the directory, postp/.

21.3.2 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_SS_NN.YYYYMMDDHHMMSS, where BASENAME indicates a character string specified separately for input and out files by file_som_in and file_som_out in namelist nml_somadv, SS elements of the moments (9 types), NN a tracer number (01 for temperature, 02 for salinity, numbers after 03 for other tracers), and the suffix for the date and time. The format of each file is the same as the tracer restart files. See Section 8.6 for the SOM scheme, and docs/README.Namelist for other items of nml_somadv.

21.3.3 Restart for the sea ice model

To handle restart files of the sea ice model, dedicated namelist blocks must be specified separately for input and output as follows:

— An example namelist for sea ice restart input —

```
&nml_seaice_rst_in
    file_icecat_restart_in = 'result/rs_ice',
/
```

- An example namelist for sea ice restart output -

```
&nml_seaice_rst_out
   file_icecat_restart_out_temp = 'result/rs_ice',
   nwrt_rst_ic = 480, !- set same as nstep_seaice
/
```

where file_icecat_restart_in and file_icecat_restart_out_temp specify the basenames of input and output files, and nwrt_rst_ic specifies the time step interval to write instantaneous states (usually, set it the same as the time step number of the integration). See docs/README.Namelist for other items. (Before MRI.COM ver.4.2, one must set l_file_name_calendar = .true. in nml_seaice_time_conf to add a suffix of the date and time to the file name in the same manner as the ocean model. After ver.4.4, the suffix is always added.)

The restart file 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) :: tlicen(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) tlicen(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)
```

21.3.4 Control of input and output at run time

a. Input

Input of restart files at run time is controlled by namelist listed on Table 21.5. 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.

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 = 1_rst_in
nml_baroclinic_run	l_rst_baroclinic_in	default = 1_rst_in
nml_tracer_run	l_rst_tracer_in	default = 1_rst_in
		applicable to only temperature and salinity
nml_vmix_run	l_rst_vmix_in	default = 1_rst_in
nml_tide_run	l_rst_tide_in	default = 1_rst_in
nml_melyam_run	l_rst_melyam_in	default = 1_rst_in
nml_nohkim_run	l_rst_nohkim_in	default = 1_rst_in
nml_gls_run	l_rst_gls_in	default = 1_rst_in
nml_density_run	l_rst_density_in	default = 1_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

Table21.5 Namelist blocks that control input of restart files.

b. Output

Model always try to write restart files of the oceanic part according to the specification by the namelists listed on Table 21.4. 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.

21.4 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 namelist parameters that control the time integration and the namelist file that specify sampling have to be given.

- NAMELIST.*name_model* controls the job and gives parameters to sub-packages. See docs/README.Namelist for details.
- NAMELIST.name_model.MONITOR specifies sampling. See docs/README.Monitor for details.

Parameters used by sub-packages are explained in corresponding chapters. Runtime parameters relevant to time-integration are listed on Tables 21.6 through 21.11.

Chapter 21 User's Guide

variable name	units	description	usage
dt_sec	sec	unit time interval for equation of motion	required
		and tracer	
nstep_matsuno_interval	integer	time step interval with which Matsuno	default = 12 steps
		(Euler-backward) time-integration scheme	
		is used	
alpha_bryan_1984	real(8)	acceleration parameter α of Bryan (1984).	default = 1
		See Section 2.3.1.	
l_monitor_time	logical	time step monitor is written to standard out-	default = .true.
		put ()	

Table21.6 Namelist nml_time_step.

Table21.7 Namelist nml_run_period.

variable name	units	description	usage
nstep_total	integer	total number of time step of this run	required

 $Table 21.8 \quad Namelist {\tt nml_exp_start} \ that \ specifies \ start \ time \ of \ the \ whole \ experiment.$

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

Table21.9 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

 $Table 21.10 \quad Namelist {\tt nml_calendar} \ that \ specifies \ treatment \ of \ leap \ year.$

variable name	units	description	usage
l_force_leap	logical	Use 1_leap to decide whether the current year	default = .false., the realistic calendar is
		is leap or not.	followed
l_leap	logical	the current year is a leap year or not.	default = .false.

Table21.11	Namelist nml	_stdout that	specifies standar	d 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 <i>name_model</i> -file_base_stdout-stdout.XXXX	
		(XXXX: mpi process number)	
l_debug	logical	debug mode	default = .false.

21.5 Monitoring

This section summarizes outputs of the mean state (history) data used for monitoring and analyses.

21.5.1 History of the ocean and ice models

In MRI.COM version 4, specification of history outputs has been significantly changed and commonized 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. 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 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. 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.)

21.6 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)
- Various analysis tools (anl)

Chapter 21 User's Guide

• 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).

21.7 Appendix

21.7.1 Model options

The model options are listed on Table 21.12. Only major options are listed here. Description about those related to bio-geochemical models can be found in Chapter 11. 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.

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.		
BULKKONDO2	Kondo (1975) is used for the surface flux bulk formula.		
BULKNCAR	Large and Yeager (2004; 2009) is used for the surface flux bulk formula. This option corresponds to		
	the COREs.		
	(*) BULKKONDO2, BULKNCAR is available only for HFLUX case.		
BULKITER	Bulk transfer coefficient is calculated using iterative method if the observed wind speed is not at		
	10 m.		
	(*) use with BULKKONDO2 and BULKNCAR		
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 $(1.0 \text{ m}^2 \text{ s}^{-1})$ 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		
	Continued on next page		

Table21.12	Description	of Model	Options
------------	-------------	----------	---------

Model option	Description			
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			
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)			
LWDOWN	Longwave radiation data include only downward component instead of default net radiation			
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			
NOMATSUNO	uses forward finite difference instead of Matsuno scheme			
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 nesting calculation			
RUNOFF	uses river runoff data			
	(*) available only for WFLUX			
SFLUXR	SSS is restored to the climatological sea surface salinity as the salinity flux			
SFLUXW	calculates salinity flux converting from the surface freshwater flux			
	(*) 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 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			
	Continued on next page			

Table	21.12 - continu	ued from prev	ious page
Chapter 21 User's Guide

Model option	Description			
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			
	(*) 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			
ZSTAR	zstar (z*) vertical coordinate			
	(*) KSGM = KM			
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			

Table 21.12 – continued from previous page

Acknowledgments

The authors of this manual made new contributions to the source code development of MRI.COM after publication of the previous version in 2010. 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 4 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, and Seiji Yukimoto.

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.

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 Oceanographic Research Department (at the time) and the Oceanography and Geochemistry Research Department of MRI contributed to the development and refinement of this model. Specifically, earlier versions of this manual (for version 4) were proofread by some members of this department, 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.
- Baker, 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.
- 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.
- 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., 1999: Numerical methods for wave equations in geophysical fluid dynamics. Springer-Verlag, 465 pp., doi:10.1007/978-1-4757-3081-4.
- Eden, C., 2007: Eddy length scales in the North Atlantic Ocean. 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:10.1175/2007JCL11510.1.
- 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 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, Eds., 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, University of Tokyo, 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

BIBLIOGRAPHY

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.
- Kondo, J., 1975: Air-sea bulk transfer coefficients in diabatic conditions. *Boundary-Layer Meteorol.*, **9**, 91–112, doi:10.1007/BF00232256.
- 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.
- 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.
- 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, 10937–10954, 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. GARP Publications Series, 17, 65 pp. http://twister.ou.edu/CFD2003/Mesinger_ArakawaGARP.pdf
- 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 CO₂ 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, 2016: Biogeochemical protocols and diagnostics for the CMIP6 ocean model intercomparison project (OMIP). *Geosci. Model Dev. Discuss.*, doi:10.5194/gmd-2016-155.
- 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.
- Saunders, P., 1990: The international temperature scale of 1990, ITS-90. *WOCE Neweletter*, **10**, 10, https://www.nodc. noaa.gov/woce/wdiu/wocedocs/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 CO₂ 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

BIBLIOGRAPHY

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 (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modell.*, **9**, 347–404, doi:10.1016/j.ocemod.2004.08.002.
- 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.
- 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.
- 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.
- 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.
- 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.

Index

Α	_
AFC	l
В	_
BBL	1
BIHARMONIC 80 27	ı.
BULKITER 163 165 166 27	i
BULKKONDO? 156, 165, 166, 27	i
BUILKNCAD 156, 166, 27	с Г
DUERICAR	
<u>C</u>	_
CALALBSI 199, 219, 221, 27	l
CALPP	l
CARBON	2
CBNHSTRUN	L
CFC 138	2
CHI CLIM 154	1
СНЕ МА 94 133 153 154 27	i.
configure in 28, 20, 133, 163, 165, 102, 215, 210, 236, 238, 261, 262, 27(י ה
CVCLIC 27	ì
	1
D	
DIFAIS 120.27	Ē
517435	
<u>F</u>	-
F2003	l –
FSVISC	l.
^	
<u>G</u>	-
GLS 175, 187, 265, 272	2
GMANISOTROP 116, 117, 118, 27	L
GMTAPER	1
GMVAR 113, 117, 118, 272	2
u	
<u>II</u>	-
HFLUX 151, 152, 156, 159, 271, 272	2
HISTICECAT	2
	-
ICE	2
ICECLIM	2
ICEFULLMONIT	3
IDEALAGE	7
IMUT	3
ISOPYCNAL	3
ISOTAPER 111, 112, 272	2
<u>J</u>	_
JMUT	3
JOT	5
<u>K</u>	_
KBBL	2
KM	3
KSGM	3
202, 27.	
L	-
LWDOWN151, 152, 155, 264, 272	2
M	_
MELYAM	2
MOVE	3
MPDATAADVEC 85, 86, 105, 139, 140, 141, 272	2

N	_
NAME_MODEL	0
NAMELIST_SCUP	9
NGETFLUX	8
nml_air_ice	1
nml albedo ocean	4
nml albedo seaice	1
nml baroclinic run	8
nml baroclinic visc horz	0
nml barotropic diff	2
nml barotropic model	2
nml barotropic run 62.26	8
nml barotropic visc horz	1
nml_bhlgeo_limit	3
nml_bblravfrc 197	3
nml_bioNPZD 13	3
nml_bottom_friction8	1
nml_bulk_wind	î
nml_calendar 260	0 0
nml_calpn 38.20	ó
nmi_capp	ז ר
nmi_cgcm_recale_replace	2 2
nmi_cgcmi_recarc_reprace	2 0
mil_defisity_full	0
mil_uni_vert_bg	9 5
nmi_emissivity_sea_water) 0
nmi_exp_start	9
nml_force_data 61, 138, 151, 152, 158, 167, 262, 264	4
nml_force_windsetup 150, 15	1
nml_trac_nearir	4
nml_gls	8
nml_gls_run	8
nml_gm_anisotrop 118	8
nml_gm_transition 116, 118	8
nml_gmanisotrop116, 11	7
nml_gmvar11	7
nml_gmvar_danabasoglu115, 118	8
nml_gmvar_eden 118	8
nml_gmvar_select	7
nml_gmvar_visbeck	7
nml_grid_scale	2
nml_grid_size_change_mix_coefs 109, 113, 117	7
nml_history	0
nml_horz_grid	2
nml_hvisc_add	1
nml_isopy_slope_clip 110, 112, 115	5
nml_melyam_run181, 268	8
nml_mkflux158, 160	0
nml_nest_par235, 239, 240	0
nml_nest_sub	1
nml_nkoblm 183, 184	4
nml_nohkim_run 183, 184, 268	8
nml_par_diff	0
nml_par_visc	0
nml_parentmodel_grid236, 237	7
nml_parentmodel_package	7
nml poles	5
nml run ini	9
nml run ini state	8
nml run period	9
nml seaice budget	2
nml seaice calendar	õ
nml seaice diff	0 0
nml seaice dyn	ő
nml seaice hst.	Ő
	~

1	000 001
nml_searce_model	
nml_seaice_rst_in	
nml seaice rst out	
nml seaice run	221 268
nml_sociae_thickness_actagenv	210, 220
mm_searce_unckness_category	
nml_seaice_time_calendar	
nml_seaice_time_conf	
nml smagor	
nml solarangle	154
	102 265 266
nml_somadv	103, 265, 266
nml_srfflx_dbase	163, 165, 168, 169
nml_sss_restore	
nml stdout	
nml sub diff	240 242
nml_sub_isopyc	
nml_sub_ssh_diff	240, 242
nml_sub_visc	
nml_submodel_bnd_cnd	237 238
nml_submodel_ond_end	227, 228, 240, 241
	237, 236, 240, 241
nml_submodel_package	
nml_submodel_tide	
nml_tide_forcing	61
nml_tide_model	61
nml_tide_mn	61 268
mm_uuc_tun	
nml_time_step	
nml_topo	
nml tracer adv	
nml tracer data 137 138 140 141 143	144 145 150 268
nml_tracer_diff horz	100
nml_tracer_diff_isopy	
nml_tracer_diff_isopy_taper	
nml tracer idealage start	
nml tracer run	140 268
nml_uueer_run	20, 20, 262
nmi_vert_grid	29, 30, 262
nml_visaniso	
nml_visc_vert_bg	81, 181, 183
nml vmbg3d	
nml_vmbg3d	
nml_vmbg3d nml_vmix_river	
nml_vmbg3d nml_vmix_river nml_vmix_run	
nml_vmbg3d nml_vmix_river nml_vmix_run nml_waterflux_adjust	
nml_vmbg3d nml_vmix_river nml_vmix_run nml_waterflux_adjust nmlrs_density	
nml_vmbg3dnml_vmix_rivernml_vmix_runnml_waterflux_adjustnmlrs_densitynmlrs gls alo	
nml_vmbg3dnml_vmix_rivernml_vmix_rivernml_vmix_runnml_waterflux_adjustnmlrs_densitynmlrs_gls_alonmlrs els avk	
nml_vmbg3d nml_vmix_river nml_vmix_run nml_waterflux_adjust nmlrs_density nmlrs_gls_alo nmlrs_gls_avk nmbrs_gls_avk	
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avp .	
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_avq .	
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_avq . nmlrs_gls_eke	
nml_vmbg3d	
nml_vmbg3d	
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_gls_psi . nmlrs_my_alo .	
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_avq . nmlrs_gls_psi . nmlrs_my_alo . nmlrs_my_avq .	
nml_mbg3d nml_mix_river nml_wixerflux_adjust nmlrs_density nmlrs_gls_alo nmlrs_gls_avk nmlrs_gls_avp nmlrs_gls_avq nmlrs_gls_psi nmlrs_my_alo nmlrs_my_aq	
nml_vmis_river nml_vmis_river nml_vmis_river nml_waterflux_adjust nmlrs_density nmlrs_gls_alo nmlrs_gls_avk nmlrs_gls_avp nmlrs_gls_avq nmlrs_gls_psi nmlrs_my_alo nmlrs_my_q nmlrs_nk_avq	
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avq . nmlrs_gls_psi . nmlrs_my_alo . nmlrs_my_aq . nmlrs_nk_avq . nmlrs_nk_eb .	
nml_vmbg3d nml_vmix_river nml_waterflux_adjust nmlrs_gls_alo nmlrs_gls_avk nmlrs_gls_avq nmlrs_gls_psi nmlrs_my_alo nmlrs_nk_avq nmlrs_nk_eb nmlrs_ptrc	
nml_mbg3d nml_mix_river nml_witarflux_adjust nmlrs_density nmlrs_gls_alo nmlrs_gls_avk nmlrs_gls_avk nmlrs_gls_avq nmlrs_gls_eke nmlrs_gls_psi nmlrs_my_alo nmlrs_my_avq nmlrs_my_avq nmlrs_mk_avq nmlrs_nk_eb nmlrs_ptrc nmlrs_s	
nml_mbg3d . nml_mix_river . nml_waterflux_adjust . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_eke . nmlrs_gls_psi . nmlrs_my_alo . nmlrs_my_aq . nmlrs_mk_avq . nmlrs_s . nmlrs_s .	$\begin{array}{c}119\\268\\268\\268\\268\\268\\268\\268\\265\\265\\265\\265\\188, 265\\188, 265\\181, 265\\181, 265\\181, 265\\184, 265\\184, 265\\184, 265\\184, 265\\140, 264\\264$
nml_vmbg3d nml_vmix_river nml_vmix_run nml_waterflux_adjust nmlrs_gls_alo nmlrs_gls_avk nmlrs_gls_avp nmlrs_gls_avq nmlrs_gls_eke nmlrs_my_alo nmlrs_my_q nmlrs_nmy_q nmlrs_nmy_avq nmlrs_ns_s nmlrs_ssh	
nml_vmbg3d nml_vmix_river nml_waterflux_adjust nmlrs_gls_alo nmlrs_gls_avk nmlrs_gls_avq nmlrs_gls_psi nmlrs_my_avq nmlrs_nk_avq nmlrs_ptrc nmlrs_ssh_dflx_x	$\begin{array}{c}119\\268\\268\\268\\268\\268\\268\\268\\265\\265\\188, 265\\188, 265\\181, 265\\181, 265\\181, 265\\184, 265\\184, 265\\184\\ 133, 137, 138, 265\\140, 264\\264\\264\\264\\264\\264\\264\\264\\264\\264\\264\\264\\$
nml_mbg3d . nml_mix_river . nml_wix_river . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_gls_psi . nmlrs_my_alo . nmlrs_my_avq . nmlrs_nk_avq . nmlrs_nk_eb . nmlrs_ptrc . nmlrs_s . nmlrs_sh_dflx_x . nmlrs_sh_dflx_y .	$\begin{array}{c}119\\268\\268\\268\\268\\39, 264\\39, 264\\39, 264\\39, 265\\188, 265\\265\\188, 265\\188, 265\\181, 265\\181, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\184, 265\\$
nml_mbg3d . nml_mix_river . nml_wix_river . nml_wix_river . nml_wix_river . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_eke . nmlrs_gls_psi . nmlrs_my_alo . nmlrs_my_avq . nmlrs_my_q . nmlrs_nk_eb . nmlrs_ssh . nmlrs_ssh_dflx_x . nmlrs_st .	$\begin{array}{c}119\\268\\268\\268\\268\\268\\39, 264\\39, 264\\39, 264\\39, 265\\265\\265\\265\\265\\265\\265\\265\\265\\265\\265\\264\\2$
nml_mbg3d . nml_mix_river . nml_waterflux_adjust . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_gls_psi . nmlrs_my_alo . nmlrs_my_aq . nmlrs_nk_eb . nmlrs_nk_eb . nmlrs_nk_eb . nmlrs_sh . nmlrs_t . nmlr	$\begin{array}{c}$
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_my_alo . nmlrs_my_alo . nmlrs_my_avq . nmlrs_nk_eb . nmlrs_nk_eb . nmlrs_ssh . nmlrs_ssh . nmlrs_ssh . nmlrs_ssh . nmlrs_t . nmlrs_t . mmlrs_tid_um .	$\begin{array}{c} & 119 \\ & 268 \\ & 268 \\ & 268 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 184, 2$
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_gls_psi . nmlrs_my_alo . nmlrs_my_aq . nmlrs_nk_avq . nmlrs_nk_eb . nmlrs_ptrc . nmlrs_ssh . nmlrs_ssh_dflx_x . nmlrs_tt . nmlrs_tide_ssh . nmlrs_tide_um .	$\begin{array}{c} & 119 \\ & 120 \\ & 268 \\ & 158, 159 \\ & 39, 264 \\ & 188, 265 \\ & 265 \\ & 265 \\ & 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 265 $
nml_mbg3d . nml_mix_river . nml_wix_river . nml_wixerflux_adjust . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_gls_psi . nmlrs_my_alo . nmlrs_my_avq . nmlrs_nk_avq . nmlrs_nk_eb . nmlrs_sk . nmlrs_ss . nmlrs_ssh . nmlrs_ssh . nmlrs_t . nmlrs_t . nmlrs_tide_um . nmlrs_u .	$\begin{array}{c} & & & 119 \\ & & & & 120 \\ & & & & 268 \\ & & & & 158, 159 \\ & & & & & 39, 264 \\ & & & & & 188, 265 \\ & & & & & 188, 265 \\ & & & & & & 188, 265 \\ & & & & & & 188, 265 \\ & & & & & & 188, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & & 181, 265 \\ & & & & & & & & 181, 265 \\ & & & & & & & & & 181, 265 \\ & & & & & & & & & & & \\ & & & & & & $
nml_mbg3d . nml_mix_river . nml_wix_river . nml_wix_river . nml_wix_river . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_eke . nmlrs_gls_psi . nmlrs_my_alo . nmlrs_my_avq . nmlrs_my_q . nmlrs_nk_eb . nmlrs_ssh . nmlrs_ssh . nmlrs_ssh . nmlrs_sh_dflx_x . nmlrs_tide_ssh . nmlrs_tide_um . nmlrs_u .	$\begin{array}{c} & 119 \\ & 268 \\ & 268 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 265 \\ & 39, 264 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 183, 137, 138, 265 \\ & 184, 265$
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_my_alo . nmlrs_my_alo . nmlrs_my_q . nmlrs_nk_eb . nmlrs_ptrc . nmlrs_ssh . nmlrs_ssh_dflx_x . nmlrs_t . nmlrs_t . nmlrs_t . nmlrs_t . nmlrs_t . nmlrs_uml .	$\begin{array}{c} & 119 \\ & 268 \\ & 268 \\ & 268 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 265 \\ & 39, 264 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 264 $
nml_vmbg3d nml_vmix_river nml_vmix_run nml_waterflux_adjust nmlrs_gls_alo nmlrs_gls_alo nmlrs_gls_avk nmlrs_gls_avp nmlrs_gls_avq nmlrs_gls_eke nmlrs_gls_eke nmlrs_my_alo nmlrs_my_alo nmlrs_my_q nmlrs_nk_eb nmlrs_nk_eb nmlrs_ssh nmlrs_ssh nmlrs_ssh_dflx_x nmlrs_tide_ssh nmlrs_tide_um nmlrs_um nmlrs_um nmlrs_um nmlrs_vmy_um nmlrs_tide_um nmlrs_vmy_um nmlrs_vmy_um	$\begin{array}{c} & 119 \\ & 120 \\ & 268 \\ & 158, 159 \\ & 39, 264 \\ & 188, 265 \\ & 265 \\ & 265 \\ & 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 265 \\ & 265 \\ & 264 \\ & 264 \\ & 264 \\ & 264 \\ & 264 \\ & 264 \\ & 265 \\ \end{array}$
nml_vmbg3d . nml_vmix_river . nml_vmix_river . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_my_alo . nmlrs_my_avq . nmlrs_nk_avq . nmlrs_nk_eb . nmlrs_ptrc . nmlrs_s . nmlrs_sh_dflx_x . nmlrs_tide_ssh . nmlrs_tide_um . nmlrs_uml . nmlrs_uml . nmlrs_v . nmlrs_v . nmlrs_wix_avd .	$\begin{array}{c} & 119 \\ & 120 \\ & 268 \\ & 158, 159 \\ & 39, 264 \\ & 188, 265 \\ & 265 \\ & 265 \\ & 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 264 \\ & 57, 264 \\ & 57, 264 \\ & 57, 264 \\ & 57, 264 \\ & 57, 264 \\ & 265 \\ & 264 \\ & 264 \\ & 264 \\ & 264 \\ & 264 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 264 \\ & 264 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 264 \\ & 264 \\ & 265 \\ & 26$
nml_mbg3d . nml_mix_river . nml_wix_river . nml_wix_river . nml_wix_river . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_eke . nmlrs_gls_eke . nmlrs_my_alo . nmlrs_my_avq . nmlrs_nk_avq . nmlrs_nk_eb . nmlrs_sh_dflx_x . nmlrs_sh_dflx_v . nmlrs_tide_ssh . nmlrs_tide_um . nmlrs_v .	$\begin{array}{c} & 119 \\ & 268 \\ & 268 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 265 \\ & 39, 264 \\ & 39, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 264 \\ & 265 \\ & 2$
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_eke . nmlrs_gls_eke . nmlrs_my_alo . nmlrs_my_alo . nmlrs_mk_avq . nmlrs_nk_eb . nmlrs_ptrc . nmlrs_ssh . nmlrs_ssh . nmlrs_ssh . nmlrs_t . nmlrs_t . nmlrs_t . nmlrs_t . nmlrs_t . nmlrs_uml . nmlrs_uml . nmlrs_uml . nmlrs_wix_avd . nmlrs_t . nmlt . nmlt . n	$\begin{array}{c} & & & 119 \\ & & & & 120 \\ & & & & 268 \\ & & & & 158, 159 \\ & & & & & 39, 264 \\ & & & & & 39, 264 \\ & & & & & 39, 264 \\ & & & & & & 39, 264 \\ & & & & & & 39, 264 \\ & & & & & & 39, 265 \\ & & & & & & & 188, 265 \\ & & & & & & & 188, 265 \\ & & & & & & & 188, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 184, 265 \\ & & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & 181, 265 \\ & & & & & & & & 181, 265 \\ & & & & & & & & 181, 265 \\ & & & & & & & & 181, 265 \\ & & & & & & & & & & & & & \\ & & & & $
nml_vmbg3d . nml_vmix_river . nml_waterflux_adjust . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_my_alo . nmlrs_my_alo . nmlrs_my_q . nmlrs_nk_eb . nmlrs_nk_eb . nmlrs_ssh . nmlrs_ssh . nmlrs_ssh . nmlrs_ssh_dflx_x . nmlrs_tide_um . nmlrs_tide_um . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_time . nmlrs_time . nmlrs_time . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . NOHKIM .	$\begin{array}{c} & 119 \\ & 120 \\ & 268 \\ & 268 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 188, 265 \\ & 265 \\ & 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 2$
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_gls_eke . nmlrs_my_alo . nmlrs_my_alo . nmlrs_my_q . nmlrs_nk_eb . nmlrs_s . nmlrs_ssh . nmlrs_ssh . nmlrs_ssh . nmlrs_tt . nmlrs_tide_um . nmlrs_uml . nmlrs_v . nmlrs_v . nmlrs_v . nmlrs_v . nmlrs_v . nmlrs_uml . nmlrs_v . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avm . NOHKIM . NOMATSUNO .	$\begin{array}{c} & & & 119 \\ & & & & 120 \\ & & & & 268 \\ & & & & 158, 159 \\ & & & & 39, 264 \\ & & & & & 39, 264 \\ & & & & & 188, 265 \\ & & & & & 188, 265 \\ & & & & & & 188, 265 \\ & & & & & & 188, 265 \\ & & & & & & 188, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 181, 265 \\ & & & & & & 184, 265 \\ & & & & & & & 184, 265 \\ & & & & & & & 184, 265 \\ & & & & & & & 184, 265 \\ & & & & & & & 184, 265 \\ & & & & & & & 184, 265 \\ & & & & & & & 184, 265 \\ & & & & & & & 184, 265 \\ & & & & & & & 184, 265 \\ & & & & & & & 184, 265 \\ & & & & & & & & 184, 265 \\ & & & & & & & & 184, 265 \\ & & & & & & & & 184, 265 \\ & & & & & & & & 184, 265 \\ & & & & & & & & 184, 265 \\ & & & & & & & & & 184, 265 \\ & & & & & & & & & & 184, 265 \\ & & & & & & & & & & 184, 265 \\ & & & & & & & & & & & 184, 265 \\ & & & & & & & & & & & & & & & \\ 133, 137, 138, 265, 2772 \\ & & & & & & & & & & & & & & \\ \end{array}$
nml_mbg3d . nml_mix_river . nml_wix_river . nml_wix_river . nml_wix_river . nml_wix_river . nmlrs_density . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_eke . nmlrs_gls_psi . nmlrs_my_alo . nmlrs_my_avq . nmlrs_mk_avq . nmlrs_nk_eb . nmlrs_nk_eb . nmlrs_sh . nmlrs_ss . nmlrs_ssh . nmlrs_ssh . nmlrs_t . nmlrs_t . nmlrs_tide_ssh . nmlrs_uml . nmlrs_v . NOHKIM . NOMATSUNO . NPARTX	
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_alo . nmlrs_gls_avp . nmlrs_gls_avp . nmlrs_gls_eke . nmlrs_gls_eke . nmlrs_my_alo . nmlrs_my_alo . nmlrs_my_q . nmlrs_nk_eb . nmlrs_nk_eb . nmlrs_sh_dflx_x . nmlrs_ssh_dflx_x . nmlrs_t . nmlrs_tide_ssh . nmlrs_tide_um . nmlrs_uml . nmlrs_v . nmlrs_vmix_avd . nmlrs_tor . nmlrs_t . nmlrs_t . nmlrs_t . nmlrs_t . nmlrs_v . nmlrs_v . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_v . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avm . nmlrs_vmix_avm . NOHKIM . NOMATSUNO . NPARTX . NDAPTY	$\begin{array}{c} & 119 \\ & 268 \\ & 268 \\ & 268 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 264 \\ & 57, 264 \\ & 265 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 265 \\ & 265 \\ & 264 \\ & 262 \\ & 272 \\ & 272 \\ & 262 \\ & 272 \\ & 262 \\ & 272 \\ & 262 \\ & 272 \\ & 2$
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avp . nmlrs_gls_eke . nmlrs_gls_eke . nmlrs_my_alo . nmlrs_my_avq . nmlrs_nk_eb . nmlrs_nk_eb . nmlrs_sh. nmlrs_sh. nmlrs_sh. nmlrs_sh. nmlrs_sh. nmlrs_tide_ssh . nmlrs_tide_um . nmlrs_vmi _avd . nmlrs_uml . nmlrs_vmi _avd . nmlrs_tide_um . nmlrs_time . nmlrs_vmi _avd . nmlrs_vmi . nmlrs_vmi . NOHKIM . NOMATSUNO . NPARTX . NPARTY . NPA	$\begin{array}{c} & 119 \\ & 268 \\ & 268 \\ & 268 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 264 \\ & 39, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 188, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 181, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 184, 265 \\ & 264 \\ & 264 \\ & 264 \\ & 265 \\ & 264 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 264 \\ & 265 \\ & 262 \\ & 272 \\ & 262, 272$
nml_vmbg3d . nml_vmix_river . nml_vmix_run . nml_waterflux_adjust . nmlrs_gls_alo . nmlrs_gls_alo . nmlrs_gls_avk . nmlrs_gls_avk . nmlrs_gls_avq . nmlrs_gls_eke . nmlrs_my_alo . nmlrs_my_alo . nmlrs_my_q . nmlrs_nk_eb . nmlrs_nk_eb . nmlrs_sh . nmlrs_sh . nmlrs_sh . nmlrs_sh . nmlrs_tide_um . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_tide_um . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . nmlrs_vmix_avd . NOHKIM . NOMATSUNO . NPARTX . NPARTY .	

NSFMRGN						262
NUM_ICECAT				215,	219,	262
NUMTRC_P	. 133,	137,	138,	262,	265,	271

ο

02	
OBT	
OFFNESTPAR	
OFFNESTSUB	
OPTIONS	
Р	
PARALLEI	262 272

PARALLEL	262,	272
PARENT	238,	272

Q

R RUNOFF......120, 158, 264, 272

S	
SCUP	
SCUPCGCM	
SCUPNEST	
SF6	
SFLUXR	
SFLUXW	
SIDYN	
SLIMIT	
SLP	50, 61, 152, 264, 272
SMAGHD	
SMAGOR	
SOLARANGLE	
SOMADVEC	85, 86, 103, 139, 140, 141, 265, 266, 272
SPHERICAL	
STABLERUN	
SUB	

т

. 150,	152,	200,	264,	272
				.272
	57	, 61,	265,	272
	108,	109,	117,	272
. 28, 2	.9, 30), 31,	255,	273
				120
	150, 	150, 152, 57 108, .28, 29, 30	150, 152, 200, 	150, 152, 200, 264,

U

V VISANISO 75, 76, 80, 273 VISBIHARM 63, 75, 76, 273

W

WADJ			158,	232,	233,	273
WFLUX	158,	159,	160,	264,	272,	273

Ζ

気象研究所技術報告一覧表

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)
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)
- NUUWI 九子園御川に初取されに、私歌観測用鉄増肥取(11万龍力, 膝骨にとめ, 円灯豆, 黒伴侍, 1979) On the Meteorological Tower and Its Observational System at Tsukuba Science City (T Hanafusa T Fujitani N Banno
and H. Uozu, 1979)
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)
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)
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)
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) 十年中にわける電圧の副報用毎に開去る研究(勝畑信毎研究初 1084)
人 X 中 にわける 当月 の 臨 胜 況 家 に 関 り る 御 九 (初 埋 X 家 卿 九 即, 1964) The Study of Melting of Snowflakes in the Atmosphere (Physical Meteorology Research Division 1984)
御前崎南方沖におけろ海底水圧観測(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)
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)
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)
火山活動監視手法に関する研究(地震火山研究部,1984)
1984)
気象研究所大気大循環モデルーI (MRI・GCM-I) (予報研究部, 1984)
A Description of the MRI Atmospheric General Circulation Model (The MRI \cdot GCM – I). (Forecast Research Division,
1984)
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)
波浪推算モデル MRIと MRI-Ⅱの相互比較研究-計算結果凶集-(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) 地震又加に見たて実験的及び知識的研究(地震しば研究前、1005)
地長了知に関する夫被的及び理論的研先(地長矢山研先前,1985) Study on Farthquake Prediction by Geophysical Method (Seismology and Volcanology Research Division, 1985)
北半球地上日平均气温偏差図(予報研究部 1986)
Maps of Monthly Mean Surface Temperature Anomalies over the Northern Hemisphere for 1891–1981 (Forecast
Research Division, 1986)
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)
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,
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)
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) 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)
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) 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-vear Integration. (Forecast Research Division, 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) 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) 宇宙線中間子強度 1983-1986 (高層物理研究部, 1987)

TECHNICAL REPORTS OF THE METEOROLOGICAL RESEARCH INSTITUTE No.80 2017

- 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

TECHNICAL REPORTS OF THE METEOROLOGICAL RESEARCH INSTITUTE No.80 2017

(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 CO₂ 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(SF₆)の測定手法の高度化と SF₆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 (SF₆) and evaluation of SF₆ standard

Highly developed precise analysis of atmospheric and oceanic sulfur hexafluoride (SF₆) and evaluation of SF₆ 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⁸⁵Kr 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 ⁸⁵Kr 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, 增田真次, 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 (Oseka District Mateorological Observatory, Hikone Local Mateorological Observatory, Kyoto Local Mateorological

(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, 黄偉健, 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,

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 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(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、川崎照夫、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)

気 象 研 究 所

1946 (昭和21) 年 設 立

所 長	:	高	野	清	治
研究総務官	:	齊	藤	和	雄
研究調整官	:	竹	内	義	明

予 報 研 究 部 部長: 小 泉 耕 気 候 研 究 部 部長:理博尾 瀬 智 明 部長: 台 風 研 究 野 功 部 高 部長:理博高 薮 環 境 · 応 用 気 象 研 究 部 出 気象衛星・観測システム研究部 部長: 鈴 木 修 _ 地 震 津 波 研 究 部 部長:理博前 田 憲 部 部長:理博山 里 巫 火 山 研 究 海洋 · 地球 化 学 研 究 部 部長:理博倉賀野 禈

気象研究所技術報告

編集委員長:尾 瀬 智 昭

編集委員	:	益	子	涉	石	井	IE.	好	沢	田	雅	洋
		北	村	祐二	猪	上	華	子	対	馬	弘	晃
		鬼	澤	真 也	碓	氷	典	久				
事務局	:	畄	部	来)	又	幸	Ĩ.				

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。

本紙に掲載された報告の著作権は気象研究所に帰属する。本紙に掲載された報告を引用する場合は、出 所を明示すれば気象研究所の許諾を必要としない。本紙に掲載された報告の全部又は一部を複製、転載、 翻訳、あるいはその他に利用する場合は気象研究所の許諾を得なければならない。個人が研究、学習、教 育に使用する場合は、出所を明示すれば気象研究所の許諾を必要としない。

気	象	研	究	所	技 第	術 80	報 号	告	ISSN 2189-8871
				平反	戈 29	年3	月	発行	2
編 集 発 行	兼 者		気	有刻	Ŕ	研	2/	充	所
〒305-0052 茨城県つくば市長峰1−1 TEL(029)853-8535									