ISSN 0386-4049

気象研究所技術報告 第五十九号

TECHNICAL REPORTS OF THE METEOROLOGICAL RESEARCH INSTITUTE No. 59

REFERENCE MANUAL FOR THE METEORO LOGICAL RESEARCH INSTITUTE COMMUNITY OCEAN MODEL (MRI.COM) VERSION 3

BY

Hiroyuki Tsujino, Tatsuo Motoi, Ichiro Ishikawa,
Mikitoshi Hirabara, Hideyuki Nakano, Goro Yamanaka,
Tamaki Yasuda, and Hiroshi Ishizaki

気象研究所技術報告

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

METEOROLOGICAL RESEARCH INSTITUTE, JAPAN

FEBRUARY 2010

TECHNICAL REPORTS OF THE MRI, NO.59

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.

Reference Manual for the Meteorological Research Institute Community Ocean Model (MRI.COM)

Version 3

気象研究所共用海洋モデル (MRI.COM)

第３版解説

Hiroyuki Tsujino, Tatsuo Motoi, Ichiro Ishikawa, Mikitoshi Hirabara, Hideyuki Nakano, Goro Yamanaka, Tamaki Yasuda, and Hiroshi Ishizaki

(Oceanographic Research Department)

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）

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 climate warming, the long-term variability in the air-sea coupled system, 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 Oceanographic 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), and "Development of ocean environmental forecasting methods" (started from fiscal year 2009). The present publication is the first English version of the MRI.COM manual and has been improved from the Japanese version published in 2005 by including newly developed parts.

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

In the early 1990s, the first coupled ocean-atmosphere model experiment was conducted through cooperation between the Oceanographic Research Department and the Climate Research Department, 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.

The first Japanese version of the MRI.COM manual was published in 2005 and the model has been continuously updated through further improvements in physical processes and addition of new processes. One of the most

pronounced improvements is the introduction of the chemical and biogeochemical processes associated with the oceanic carbon cycle. This is continuing even now and will be finished in a few years.

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 and has already achieved many satisfactory results. Based on our experiences, we believe MRI.COM is one of the best ocean models in the world. We thank 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.

Hiroshi Ishizaki
Director
Oceanographic Research Department

前 文

地球規模の気候温暖化、大気海洋結合系の長期変動、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―フィージビリティ・スタディ―」（平成２０年度）、「海洋環境の予測技術の開発」（平成２１年度～平成２５年度）において継続されている。本技術報告は、2005 年に出版された日本語版の解説書の内容を改訂し、日本語版出版後新たに付加された開発項目に関する解説を加えた、最初の英語による解説書である。

海洋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システム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。

最初の日本語による解説書が 2005 年に出版された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。特筆すべきは、海洋炭素循環に関連して、化学過程および生物地球化学過程が付加されたことである。生物地球化学過程の開発は現在も継続中で今後数年で完了すると思われる。

MRI.COM システム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ある。その経験から、本モデルシステムは世界に幾つかある他の海洋モデルシステム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、本モデルシステム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。

海洋研究部長
石崎 廣

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). As the name suggests, it has been used for studying large scale oceanic phenomena and as the oceanic part of the coupled climate models developed at MRI.

The current version of MRI.COM is version 3. 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., 2010). One of the reasons for creating a new version was that the definition of vertical grid arrangement was modified during the development. MRI plans to participate in the phase five of the Coupled Model Intercomparison Project (CMIP5) using MRI-ESM1, and its results on the future projection are expected to be used by a wide range of communities, so we decided to prepare a detailed description of its oceanic component in English.

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 we believe 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 and definition of continuity equations for unit grid cells are presented in Chapter 3.

Part II describes the dynamical core. The method of solving the barotropic and the baroclinic part of the momentum equation are presented in Chapters 4 and 5, respectively. The method of solving the advection-diffusion equation for tracers (temperature and salinity) is presented in Chapter 6.

Part III describes additional processes. Surface mixed-layer models are presented in Chapter 7, surface fluxes in Chapter 8, sea ice in Chapter 9, bottom boundary layer parameterization in Chapter 10, and biogeochemical models are presented in Chapter 11.

Part IV contains miscellaneous topics. Basics of the finite difference method are presented in Chapter 12, some high-accuracy advection schemes are presented in Chapter 13, general orthogonal curvilinear coordinates and related calculus are introduced in Chapter 14, how to construct a pair of nested-grid models is presented in Chapter 15, and user's guide to construct and run a model is presented in Chapter 16.

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.

References

Griffies, S. M., 2004: Fundamentals of ocean climate models, Princeton University Press, 518pp.

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, No.47, 189pp.

Kantha, L., and C. Clayson, 2000: Numerical models of ocean and oceanic processes, International Geophysics Series, Vol. 66, 940pp.

Yukimoto, S., and coauthors, 2010: Meteorological Research Institute-Earth System Model v1 (MRI-ESM1) - Model Description -, Technical Reports of the Meteorological Research Institute, No.64, in press.

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。海洋大循環モデルの名の通り、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。

MRI.COM の最新バージョンは３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 年台前半）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 （石川他、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（本バージョン）は、気象研地球システムモデル（MRI-ESM1; Yukimoto et al., 2010） の海洋部分への供用を主な目的としたものである。新バージョン作成理由のひとつは、開発段階で、鉛直格子点の定義位置を変更したことである。MRI-ESM1 は第５次結合モデル相互比較プロジェクト（CMIP5）への参加を予定しており、同モデルによる将来予測結果などは、幅広い分野の人々に使用されることが見込まれるため、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。

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。空間格子配置と、単位格子に対する連続方程式の差分式の定義を第 3 章で行う。

第 II 部ではモデルの核心部分の解説を行う。運動方程式の順圧成分と傾圧成分の解法を第 4 章と第 5 章でそれぞれ述べる。トレーサー（水温と塩分）に対する移流拡散方程式の解法を第 6 章に述べる。

第 III 部では付加的物理プロセスの解説を行う。幾つかの海面混合層モデルを第 7 章で、海面フラックスの取り扱いを第 8 章で、海氷モデルを第 9 章で、海底境界層モデルを第 10 章で、幾つかの生物地球化学モデルを第 11 章で解説する。

第 IV 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 12 章で、幾つかの高精度トレーサー移流スキームについて第 13 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 14 章で、入れ子モデルの作成と使用法について第 15 章で解説する。最後にモデルの作成と実行方法の解説を第 16 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が必要である。

References

Griffies, S. M., 2004: Fundamentals of ocean climate models, Princeton University Press, 518pp.

石川一郎・辻野博之・平原幹俊・中野英之・安田珠幾・石崎廣，2005: 気象研究所共用海洋モデル (MRI.COM) 解説，気象研究所技術報告第 47 号，189pp.

Kantha, L., and C. Clayson, 2000: Numerical models of ocean and oceanic processes, International Geophysics Series, Vol. 66, 940pp.

Yukimoto, S., and coauthors, 2010: Meteorological Research Institute-Earth System Model v1 (MRI-ESM1) - Model Description -, Technical Reports of the Meteorological Research Institute, No.64, in press.

Contents

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 are 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 sub-grid scale (SGS) parameterizations. In the future, the coverage of the OGCMs will be extended to directly resolve the smaller space-scale and the shorter time-scale phenomena such as tides and vertical convection, but they are not the primary targets of the current OGCMs. The current OGCMs could cover phenomena from thermohaline circulations to mesoscale eddies. However, it is 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 (e.g., the Earth Simulator), but such a resource is not available to everyone.

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 that attempt to simulate realistic oceanic states adopt the finite difference method to discretize the equations. The spectral approach widely used by an atmospheric model 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 models.

Ocean models 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 adopt depth as the vertical coordinate, σ-coordinate or terrain following models adopt fractional depth between the sea surface and the sea floor as the vertical coordinate, and ρ-coordinate or isopycnal models adopt 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

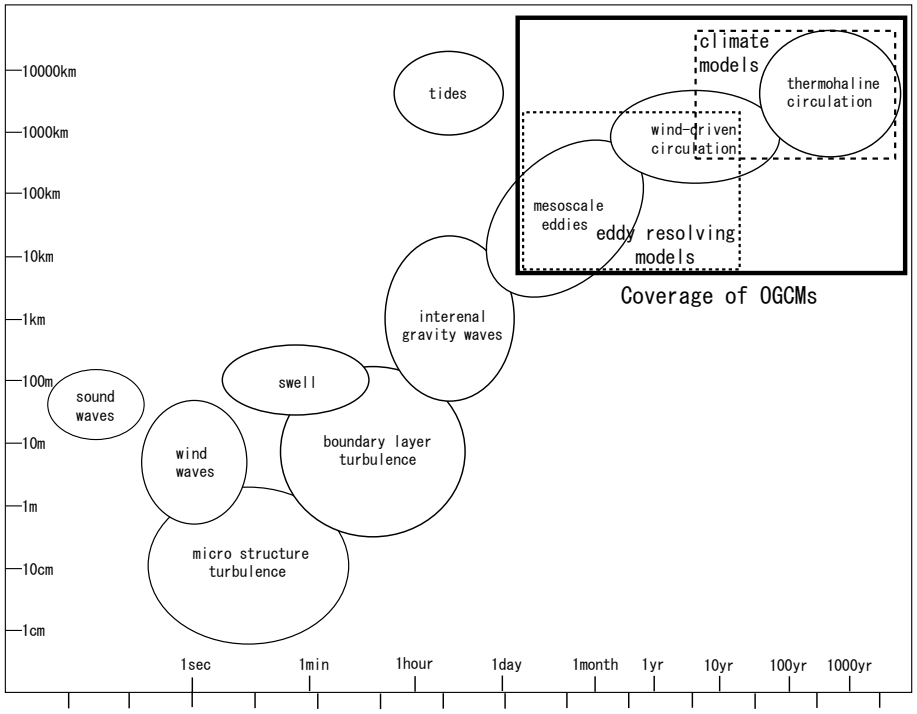

Figure 1.1. Various oceanic phenomena in terms of their space and time scales and coverage of the ocean general circulation model. The figure for oceanic phenomena is adopted from von Storch and Zwiers (2001).

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. 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 implementating 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 et al., 2004) and is the most widely used. Most climate centers participating in climate model intercomparison projects use the z-coordinate models; MRI.COM also adopts z-coordinates. 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 σ-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. 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 points, 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 the typical computational cost of each process in Figure 1.2.

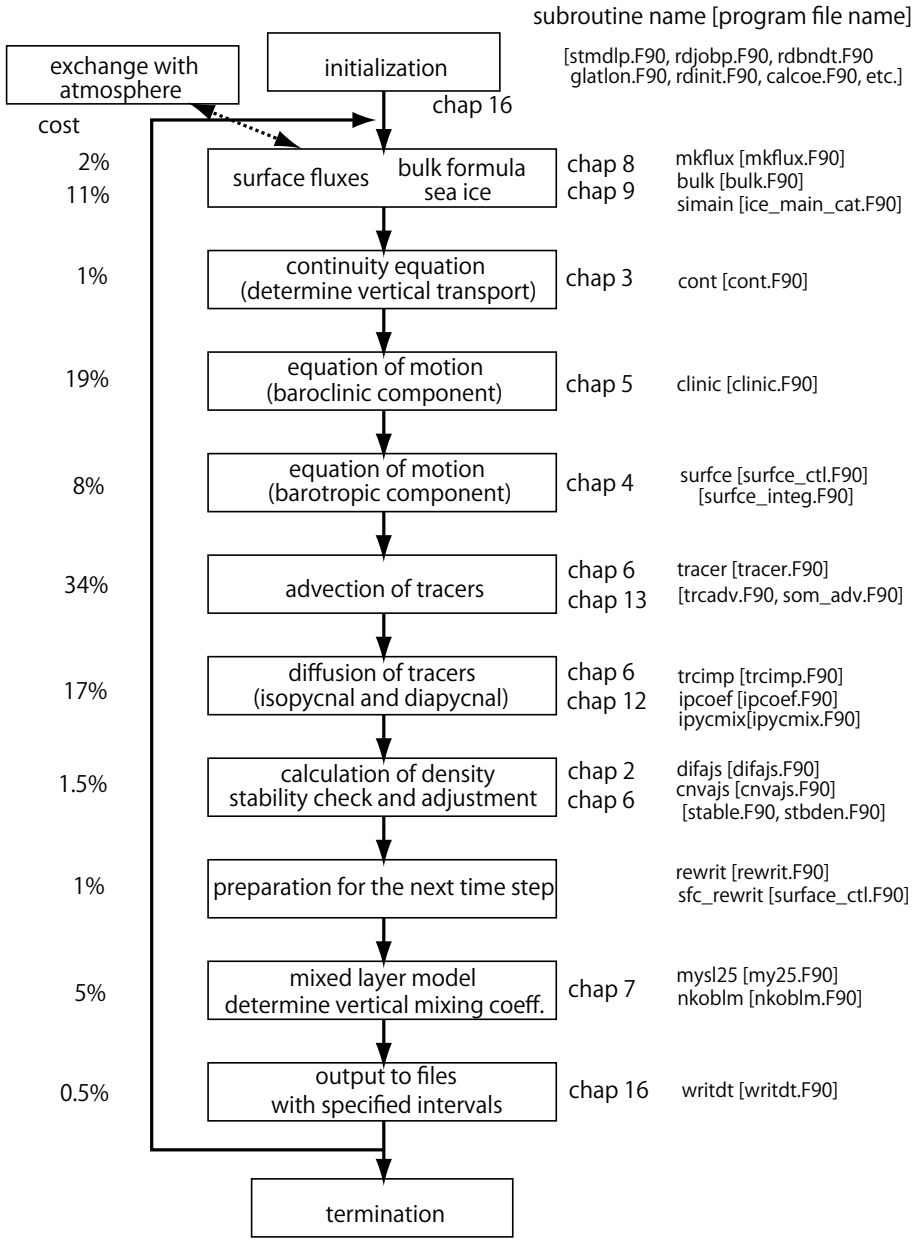

Figure 1.2. Program Structure of MRI.COM

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, and some surface bulk formulae 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 sub-grid scale processes are parameterized and thus sub-grid scale schemes should be selected carefully.

To continue to stand as a multi-purpose model, an OGCM should be easily coupled with other component models and data assimilation schemes. 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.

References

Arakawa, A., 1972: Design of the UCLA general circulation model, *Numerical Simulation Weather and Climate, Tech. Rep. No. 7*, *Dep. of Meteorology, University of California, Los Angeles*, 116 pp.

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.

Bleck, R., G. Halliwell, A. Wallcraft, S. Carroll, K. Kelly, K. Rushing, 2002: Hybrid Coordinate Ocean Model (HYCOM) User's Manual,
available online at http://hycom.rsmas.miami.edu/hycom-model/documentation.html.

Mellor, G. L., 2004: Users guide for a three-dimensional, primitive equation, numerical ocean model, *Prog. in Atmos. and Ocean. Sci, Princeton University*, 53pp.,
available online at http://www.aos.princeton.edu/WWWPUBLIC/htdocs.pom/PubOnLine/POL.html.

Griffies, S. M., 2004: Fundamentals of ocean climate models, Princeton University Press, 518pp.

Griffies, S. M., M. J. Harrison, R. C. Pacanowski, and A. Rosati, 2004: A Technical Guide to MOM4, *GFDL Ocean Group Technical Report No. 5, NOAA/Geophysical Fluid Dynamics Laboratory, Version prepared on March 3, 2004*,
available online at http://www.gfdl.noaa.gov.

Hasumi, H., 2006: CCSR Ocean Component Model (COCO) version 4.0, *CCSR report*, *25*, 103pp.

Shchepetkin, A. F., and J. C. McWilliams, 2003: A method for computing horizontal pressure-gradient force in an oceanic model with a non-aligned vertical coordinate, *J. Geophys. Res.*, *108*, 3090-3124.

Shchepetkin, A. F., and J. C. McWilliams, 2005: The Regional Ocean Modeling System: A split-explicit, free-surface, topography-following-coordinate oceanic model, *Ocean Modell.*, *9*, 347-404.

von Storch, H., and F. Zwiers, 2004: Statistical Analysis in climate research, Cambridge university press, 484pp.

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. The numerical methods to solve these equations are detailed in later chapters.

2.1 Formulation

2.1.1 Coordinate System

The governing equations of an ocean general circulation model need to be formulated on a sphere. Originally, spherical coordinates were adopted, and the equations are discretized on a geographical (latitude-longitude-depth) grid. A problem arises for a global model because the North Pole is singular in spherical coordinates. 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 could hinder long-term integration.

One simple means to remove this North Pole singularity is to shift both poles to land. In this case, one could use the spherical coordinate model without major modification by just adjusting the Coriolis parameter. Unfortunately, there are not many pairs of points on land that are symmetric about the Earth's center.† One would like to choose a pair that provides the largest zonal grid size for the oceanic grid point having the smallest zonal grid size. The land mass around the shifted pole should be as large as possible or the oceanic grid point nearest to the shifted pole should be as far from the pole as possible. However, even the most suitable pair with poles on Greenland and Antarctica (near the Ross Sea) places the nearest oceanic grid point to the shifted pole only five degrees away. 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 may choose the coordinate system most suitable 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 could be arranged on geographical coordinates since the spherical coordinates are one form of the generalized orthogonal coordinates. In our model, the model equations are formulated on generalized orthogonal coordinates. Chapter 14 summarizes the concepts and calculus related to the generalized orthogonal coordinates.

2.1.2 Momentum Equation

Most ocean general circulation models use the Boussinesq approximation, where the density of sea water is replaced by a reference density (ρ_0) for all terms except for the buoyancy term. On a general orthogonal coordinate system ((μ, ψ, r)) whose unit vectors are $\mathbf{e}_\mu$, $\mathbf{e}_\psi$, and $\mathbf{e}_r$, the momentum equation for velocity $\mathbf{v} = u\mathbf{e}_\mu + v\mathbf{e}_\psi + w\mathbf{e}_r$,

*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.

where $u = h_\mu \dot{\mu}$, $v = h_\psi \dot{\psi}$, $w = h_r \dot{r}$, is represented by

$$\frac{\partial u}{\partial t} + \mathbf{v}\cdot\nabla u + f_\psi w - fv = -\frac{1}{\rho_0 h_\mu}\frac{\partial P}{\partial \mu} - \frac{v}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial \psi}u - \frac{\partial h_\psi}{\partial \mu}v\right) - \frac{w}{h_r h_\mu}\left(\frac{\partial h_\mu}{\partial r}u - \frac{\partial h_r}{\partial \mu}w\right) + \mathscr{V}_\mu, \tag{2.1}$$

$$\frac{\partial v}{\partial t} + \mathbf{v}\cdot\nabla v + fu - f_\mu w = -\frac{1}{\rho_0 h_\psi}\frac{\partial P}{\partial \psi} - \frac{w}{h_\psi h_r}\left(\frac{\partial h_\psi}{\partial r}v - \frac{\partial h_r}{\partial \psi}w\right) - \frac{u}{h_\mu h_\psi}\left(\frac{\partial h_\psi}{\partial \mu}v - \frac{\partial h_\mu}{\partial \psi}u\right) + \mathscr{V}_\psi, \tag{2.2}$$

$$\frac{\partial w}{\partial t} + \mathbf{v}\cdot\nabla w + f_\mu v - f_\psi u = -\frac{1}{\rho_0 h_r}\frac{\partial P}{\partial r} - g\frac{\rho}{\rho_0} - \frac{u}{h_r h_\mu}\left(\frac{\partial h_r}{\partial \mu}w - \frac{\partial h_\mu}{\partial r}u\right) - \frac{v}{h_\psi h_r}\left(\frac{\partial h_r}{\partial \psi}w - \frac{\partial h_r}{\partial r}v\right) + \mathscr{V}_r, \tag{2.3}$$

where P is pressure, $\mathscr{V}$ is viscosity, and g is the acceleration due to gravity. The radial distance from the Earth's center is represented by r and the gravity is in the negative direction of r.

The Coriolis force is represented by

$$2\Omega\times\mathbf{v} = (2\Omega_\psi w - 2\Omega_r v)\mathbf{e}_\mu + (2\Omega_r u - 2\Omega_\mu w)\mathbf{e}_\psi + (2\Omega_\mu v - 2\Omega_\psi u)\mathbf{e}_r, \tag{2.4}$$

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 $f = f_r = 2\Omega_r$.

The above expression is the most general form under the Boussinesq approximation. This form is used to formulate a non-hydrostatic model and a quasi-hydrostatic model.

Further approximations are adopted for the standard model calculations. 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 $\partial/\partial z$. To conserve angular momentum under the above approximations, we neglect the metric terms that involve w for the horizontal components, all the metric terms for the vertical components, and the Coriolis terms that do not involve f (Phillips, 1966).

Hydrostatic approximation is adopted for the vertical momentum equation:

$$0 = -\frac{\partial P}{\partial z} - \rho g. \tag{2.5}$$

The resultant momentum equation described below is discretized by the default version of MRI.COM:

$$\frac{\partial u}{\partial t} - fv = -\frac{1}{\rho_0 h_\mu}\frac{\partial P}{\partial \mu} - \mathscr{A}(u) - \frac{v}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial \psi}u - \frac{\partial h_\psi}{\partial \mu}v\right) + \mathscr{V}_u, \tag{2.6}$$

$$\frac{\partial v}{\partial t} + fu = -\frac{1}{\rho_0 h_\psi}\frac{\partial P}{\partial \psi} - \mathscr{A}(v) - \frac{u}{h_\mu h_\psi}\left(\frac{\partial h_\psi}{\partial \mu}v - \frac{\partial h_\mu}{\partial \psi}u\right) + \mathscr{V}_v. \tag{2.7}$$

The term $\mathscr{A}$ in (2.6) and (2.7) represents advection. For a scalar quantity α, $\mathscr{A}$ is defined by

$$\mathscr{A}(\alpha) = \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi u\alpha)}{\partial \mu} + \frac{\partial(h_\mu v\alpha)}{\partial \psi}\right\} + \frac{\partial(w\alpha)}{\partial z}. \tag{2.8}$$

The viscosity term is represented by $\mathscr{V}$, which is calculated separately in the lateral and vertical directions. For lateral viscosity, a harmonic (default) or biharmonic scheme is used. The harmonic scheme is used for vertical viscosity. The specific form for harmonic viscosity is as follows:

$$\mathscr{V}_u = \frac{1}{h_\psi^2}\frac{\partial}{h_\mu \partial \mu}\left(h_\psi^2 \nu_H D_T\right) + \frac{1}{h_\mu^2}\frac{\partial}{h_\psi \partial \psi}\left(h_\mu^2 \nu_H D_S\right) + \frac{\partial}{\partial z}\left(\nu_V \frac{\partial u}{\partial z}\right), \tag{2.9}$$

$$\mathscr{V}_v = \frac{1}{h_\psi^2}\frac{\partial}{h_\mu \partial \mu}\left(h_\psi^2 \nu_H D_S\right) - \frac{1}{h_\mu^2}\frac{\partial}{h_\psi \partial \psi}\left(h_\mu^2 \nu_H D_T\right) + \frac{\partial}{\partial z}\left(\nu_V \frac{\partial v}{\partial z}\right), \tag{2.10}$$

where ν_H is the horizontal viscosity coefficient, ν_V is the vertical viscosity coefficient, D_T is the horizontal tension and D_S is the horizontal shear:

$$D_T = h_\psi \frac{\partial}{h_\mu \partial \mu}\left(\frac{u}{h_\psi}\right) - h_\mu \frac{\partial}{h_\psi \partial \psi}\left(\frac{v}{h_\mu}\right), \tag{2.11}$$

$$D_S = h_\mu \frac{\partial}{h_\psi \partial \psi}\left(\frac{u}{h_\mu}\right) + h_\psi \frac{\partial}{h_\mu \partial \mu}\left(\frac{v}{h_\psi}\right). \tag{2.12}$$

The above representation of the viscous term was derived by Bryan (1969) and is consistent with Smagorinsky (1963). Note that in the latter the coefficients also varies according to D_T and D_S (cf. Section 5.2.1). 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.

When the biharmonic scheme is selected for lateral viscosity, the above harmonic operation is repeated twice. The friction terms $\mathscr{V}_u$ and $\mathscr{V}_v$ obtained by the first procedure are sign-reversed and substituted for u and v in (2.11) and (2.12) in the second procedure.

2.1.3 Continuity equation

Mass conservation is represented by the continuity equation for an incompressible fluid:

$$\mathscr{A}(1) = 0. \tag{2.13}$$

2.1.4 Temperature and salinity equation

The advection-diffusion equation is solved for temperature (potential temperature) and salinity:

$$\frac{\partial T}{\partial t} = -\mathscr{A}(T) + \mathscr{D}(T) + S^T, \tag{2.14}$$

$$\frac{\partial S}{\partial t} = -\mathscr{A}(S) + \mathscr{D}(S) + S^S, \tag{2.15}$$

where $\mathscr{D}$ is the diffusion operator and S^τ represents internal sources and sinks for tracer τ caused by nudging, convective adjustment (Section 6.4), short wave absorption (Section 8.3.1), etc.

There are several forms for the diffusion operator. 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 (14.22) and (14.23), the harmonic-type diffusivity is represented as follows:

$$\mathscr{D}(T) = \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial \mu}\left(\frac{h_\psi \kappa_H}{h_\mu}\frac{\partial T}{\partial \mu}\right) + \frac{\partial}{\partial \psi}\left(\frac{h_\mu \kappa_H}{h_\psi}\frac{\partial T}{\partial \psi}\right)\right\} + \frac{\partial}{\partial z}\left(\kappa_V \frac{\partial T}{\partial z}\right) \tag{2.16}$$

$$= \frac{1}{h_\mu h_\psi}\left\{\frac{\partial h_\psi F_\mu^T}{\partial \mu} + \frac{\partial h_\mu F_\psi^T}{\partial \psi}\right\} + \frac{\partial F_z^T}{\partial z}, \tag{2.17}$$

$$\mathscr{D}(S) = \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial \mu}\left(\frac{h_\psi \kappa_H}{h_\mu}\frac{\partial S}{\partial \mu}\right) + \frac{\partial}{\partial \psi}\left(\frac{h_\mu \kappa_H}{h_\psi}\frac{\partial S}{\partial \psi}\right)\right\} + \frac{\partial}{\partial z}\left(\kappa_V \frac{\partial S}{\partial z}\right) \tag{2.18}$$

$$= \frac{1}{h_\mu h_\psi}\left\{\frac{\partial h_\psi F_\mu^S}{\partial \mu} + \frac{\partial h_\mu F_\psi^S}{\partial \psi}\right\} + \frac{\partial F_z^S}{\partial z}, \tag{2.19}$$

where κ_H and κ_V are the horizontal and vertical diffusion coefficients and the diffusive fluxes are represented by F. When the biharmonic-type is selected for horizontal diffusion, the above Laplacian operation is repeated twice (the diffusion coefficient has a negative value).

In the real ocean, transport and mixing would occur along neutral (isopycnal) surfaces. Thus horizontal mixing along the constant depth surface is generally inappropriate since neutral surfaces are generally slanting relative to the constant depth surface. To represent excursion and mixing of water masses along neutral surfaces, neutral physics schemes are developed and widely used in place of the horizontal diffusion scheme presented above. When the neutral physics schemes are selected, the advection-diffusion equation for any scalar quantity (τ) is expressed as follows (Gent and McWilliams, 1990):

$$\frac{D\tau}{Dt} + \nabla_H \cdot \Big[\tau \frac{\partial}{\partial z}(\kappa_T \nabla_H \rho/\rho_z)\Big] + \frac{\partial}{\partial z}\Big[\tau \nabla_H \cdot (-\kappa_T \nabla_H \rho/\rho_z)\Big] = \mathscr{D}(\tau) + S^\tau, \tag{2.20}$$

where the first term on the r.h.s. is the isopycnal diffusion, whose form is given by

$$\mathscr{D}(\tau) = \nabla \cdot (\kappa_I \mathbf{K} \nabla \tau), \tag{2.21}$$

where

$$\mathbf{K} = \frac{1}{\rho_x^2+\rho_y^2+\rho_z^2}\begin{pmatrix} \rho_y^2+\rho_z^2 & -\rho_x\rho_y & -\rho_x\rho_z \\ -\rho_x\rho_y & \rho_x^2+\rho_z^2 & -\rho_y\rho_z \\ -\rho_x\rho_z & -\rho_y\rho_z & \rho_x^2+\rho_y^2 \end{pmatrix} \tag{2.22}$$

$$= \frac{1}{1+(\rho_x/\rho_z)^2+(\rho_y/\rho_z)^2}\begin{pmatrix} 1+(\rho_y/\rho_z)^2 & -(\rho_x/\rho_z)(\rho_y/\rho_z) & -\rho_x/\rho_z \\ -(\rho_x/\rho_z)(\rho_y/\rho_z) & 1+(\rho_x/\rho_z)^2 & -\rho_y/\rho_z \\ -\rho_x/\rho_z & -\rho_y/\rho_z & (\rho_x/\rho_z)^2+(\rho_y/\rho_z)^2 \end{pmatrix}, \tag{2.23}$$

(Redi, 1982). In the above, the Cartesian notation is used for simplicity. The subscript x represents $\partial/(h_\mu \partial\mu)$ and y represents $\partial/(h_\psi \partial\psi)$. 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 (2.20) have the form of advection terms with a transport velocity vector (u_T, v_T, w_T):

$$u_T \equiv \frac{\partial}{\partial z}\Big(\kappa_T \frac{1}{h_\mu}\frac{\partial\rho}{\partial\mu}\Big/\frac{\partial\rho}{\partial z}\Big), \tag{2.24}$$

$$v_T \equiv \frac{\partial}{\partial z}\Big(\kappa_T \frac{1}{h_\psi}\frac{\partial\rho}{\partial\psi}\Big/\frac{\partial\rho}{\partial z}\Big), \tag{2.25}$$

$$w_T \equiv -\frac{1}{h_\mu h_\psi}\Big\{\frac{\partial}{\partial\mu}\Big(\kappa_T \frac{h_\psi}{h_\mu}\frac{\partial\rho}{\partial\mu}\Big/\frac{\partial\rho}{\partial z}\Big) + \frac{\partial}{\partial\psi}\Big(\kappa_T \frac{h_\mu}{h_\psi}\frac{\partial\rho}{\partial\psi}\Big/\frac{\partial\rho}{\partial z}\Big)\Big\}, \tag{2.26}$$

(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 thickness diffusivity.

Note that these could be rewritten as

$$\mathscr{G}(\tau) = \nabla \cdot (\kappa_T \mathbf{A} \nabla \tau) \tag{2.27}$$

with

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & -\rho_x/\rho_z \\ 0 & 0 & -\rho_y/\rho_z \\ \rho_x/\rho_z & \rho_y/\rho_z & 0 \end{pmatrix}. \tag{2.28}$$

Comparing with (2.23), we notice that the isopycnal diffusion and the thickness diffusion terms are combined to yield a concise form (Griffies, 1998) and (2.20) can be rewritten as:

$$\frac{D\tau}{Dt} = \nabla \cdot \{(\kappa_I \mathbf{K} - \kappa_T \mathbf{A})\nabla\tau\} + S^\tau. \tag{2.29}$$

2.1.5 Equation of state of sea water

The *in situ* density of sea water is a function of (potential) temperature, salinity, and pressure:

$$\rho = \rho(T,S,P). \tag{2.30}$$

The equation of state is usually given as a polynomial fit to the available measurements. A detailed description of this will be presented later (Section 2.2.5).

2.1.6 Boundary conditions

a. Momentum equation

Upper surface ($z = 0$):

At the sea surface, momentum flux enters the ocean in the form of wind stress (or stress from sea ice where sea ice exists):

$$\nu_V \frac{\partial(u,v)}{\partial z}\Big|_{z=0} = \frac{(\tau_x,\tau_y)}{\rho_0}. \tag{2.31}$$

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

In the model algorithm, this is expressed as a body force to the first level velocity,

$$(F^u_{\rm surf}, F^v_{\rm surf}) = \frac{(\tau_x,\tau_y)}{\rho_0 \Delta z_{\frac{1}{2}}}, \tag{2.32}$$

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

Surface fresh water flux is assumed to have zero velocity.

Bottom ($z = -H$):

Bottom friction (τ^b_x in zonal and τ^b_y in meridional direction) exists at the sea floor.

For the bottom level velocity (u,v), the stress caused by bottom friction is proportional to the magnitude of the velocity and has an angle $(\theta_0 + \pi)$ to the velocity vector (Weatherly, 1972):

$$(\tau^b_x, \tau^b_y) = -C_{\rm btm}\rho_0\sqrt{u^2+v^2}(u\cos\theta_0 - v\sin\theta_0, v\cos\theta_0 + u\sin\theta_0), \tag{2.33}$$

where $C_{\rm btm}$ is a dimensionless drag coefficient with a value 1.225×10^{-3} and $\theta_0 = \pm 10^\circ$. The sign of θ_0 is positive in the northern hemisphere and negative in the southern hemisphere.

Side wall:

No slip condition is applied ($u = v = 0$).

b. Temperature and Salinity

Upper surface ($z = 0$):

At the sea surface, heat and fresh water are exchanged with air and sea ice. Salt is also exchanged below the 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:

$$\kappa_V \frac{\partial T}{\partial z}\Big|_{z=0} = F_z^T, \tag{2.34}$$

$$\kappa_V \frac{\partial S}{\partial z}\Big|_{z=0} = F_z^S, \tag{2.35}$$

where flux is defined positive downward (positive into the ocean).

Fresh water flux is explicitly incorporated into the sea level equations ((2.44) and (2.66)). It could also exert a force on temperature and salinity. The water transported through the air-sea interface (W_{AO}) and ice bottom (W_{Ibot}) is assumed to have the first level temperature (T_1). The water transported from the ice surface ($W_{Isurf} > 0$) is assumed to have the freezing point temperature (T_{freeze}). The water exchanged with ice is assumed to have low salinity ($S_I = 4.0$ [psu]). Note that the freezing point temperature (T_{freeze}) is given by mS_I, where $m = -0.0543$ [K / psu].

The expression for the surface fluxes is given by

$$F_z^T = \frac{Q}{\rho_0 C_p} + (W_{AO} + W_{Ibot}) \cdot T_1 + W_{Isurf} \cdot T_{\text{freeze}}, \tag{2.36}$$

$$F_z^S = (W_{Ibot} + W_{Isurf}) \cdot S_I, \tag{2.37}$$

where Q is heat flux, and C_p is the specific heat of sea water.

To avoid an unexpected rising or falling trend of sea level during a long-term integration, fresh water flux might be converted to salt flux. In this case, the surface flux is given by

$$F_z^T = \frac{Q}{\rho_0 C_p} - W_{Isurf} \cdot (T_1 - T_{\text{freeze}}), \tag{2.38}$$

$$F_z^S = -W_{AO} \cdot S_1 - (W_{Ibot} + W_{Isurf}) \cdot (S_1 - S_I), \tag{2.39}$$

where S_1 is the first level salinity (see Chapter 8 for derivation). Note that fresh water flux should not be added to the sea level in this case.

In a long-term run driven by surface flux, temperature and salinity might exhibit unexpected drift. To avoid this, surface temperature and salinity might be restored to observational or climatological values (T^*, S^*):

$$F_z^T = -\frac{1}{\gamma_t}(T - T^*)\Delta z_{\frac{1}{2}}, \tag{2.40}$$

$$F_z^S = -\frac{1}{\gamma_s}(S - S^*)\Delta z_{\frac{1}{2}}, \tag{2.41}$$

where $\Delta z_{\frac{1}{2}}$ is the surface layer thickness, and γ_t and γ_s are restoring time for temperature and salinity in units of seconds.

Bottom ($z = -H$):

At the sea floor, the adiabatic boundary condition is applied:

$$\frac{\partial T}{\partial z} = 0, \quad \frac{\partial S}{\partial z} = 0. \tag{2.42}$$

Side wall:

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

$$\frac{\partial T}{\partial n} = 0, \quad \frac{\partial S}{\partial n} = 0, \tag{2.43}$$

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 ($z = 0$):

At the sea surface, vertical velocity is generated because a water parcel moves following the freely moving sea surface:

$$w = \frac{d\eta}{dt} - (P - E + R) = \frac{\partial \eta}{\partial t} + u\frac{1}{h_\mu}\frac{\partial \eta}{\partial \mu} + v\frac{1}{h_\psi}\frac{\partial \eta}{\partial \psi} - (P - E + R), \tag{2.44}$$

where P is precipitation, E is evaporation, and R is river discharge.

Bottom ($z = -H$):

At the sea floor, vertical velocity is generated because the water parcel moves following the bottom topography:

$$w = -\left(u\frac{1}{h_\mu}\frac{\partial H}{\partial \mu} + v\frac{1}{h_\psi}\frac{\partial H}{\partial \psi}\right). \tag{2.45}$$

d. Mixing at the surface boundary layer

Near the sea surface, strong vertical mixing could 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 model with typical horizontal and vertical resolutions. These processes are parameterized as enhanced vertical viscosity and diffusivity. The user may specify either a high vertical viscosity and diffusivity *a priori* or use a surface boundary layer model. MRI.COM supports three surface boundary layer models: Mellor and Blumberg (2004), Noh and Kim (1999), and Large et al. (1994). In the surface boundary layer models, vertical viscosity and diffusivity change in time and are calculated every time step. See Chapter 7 for details.

2.1.7 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 ($\sim 200\,[\mathrm{m/s}]$). A several-thousand-year integration will not be a workable exercise as long as we are restricted by this criteria 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

$$\alpha\frac{\partial u}{\partial t} - fv = -\frac{1}{\rho_0 h_\mu}\frac{\partial P}{\partial \mu} - \mathscr{A}(u) - \frac{v}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial \psi}u - \frac{\partial h_\psi}{\partial \mu}v\right) + \mathscr{V}_u, \tag{2.46}$$

$$\alpha\frac{\partial v}{\partial t} + fu = -\frac{1}{\rho_0 h_\psi}\frac{\partial P}{\partial \psi} - \mathscr{A}(v) + \frac{u}{h_\mu h_\psi}\left(\frac{\partial h_\mu}{\partial \psi}u - \frac{\partial h_\psi}{\partial \mu}v\right) + \mathscr{V}_v. \tag{2.47}$$

The modified temperature and salinity equations are given by

$$\gamma\frac{\partial T}{\partial t} = -\mathscr{A}(T)+\mathscr{D}(T)+S^T, \tag{2.48}$$

$$\gamma\frac{\partial S}{\partial t} = -\mathscr{A}(S)+\mathscr{D}(S)+S^S. \tag{2.49}$$

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:

$$\omega^2 = \frac{f^2}{\alpha^2}+\Big(\frac{gH_n}{\alpha}\Big)(k^2+l^2). \tag{2.50}$$

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:

$$\omega = -\beta k\Big[\alpha(k^2+l^2)+\frac{f^2}{gH_n}\Big]^{-1}. \tag{2.51}$$

Again, a large α yields a low phase speed.

In standard practice, a value from several tens to a few hundreds 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.

2.2 Numerical Methods

2.2.1 Discretization

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 using solution algorithms.

Since MRI.COM basically adopts z-coordinate, a fixed 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. The finite volume approach realizes a smooth transition from z-coordinates in the interior to σ-coordinates near the sea surface as detailed in Chapter 4.

A vertically integrated expression for the primitive equations is useful for describing the solution procedure. These are called semi-discrete equations. The body force and the metric term will be simply multiplied by the grid width. The material derivative and the sub-grid scale flux need some attention.

The material derivative of any quantity α,

$$\frac{D\alpha}{Dt} = \frac{\partial\alpha}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi u\alpha)}{\partial\mu}+\frac{\partial(h_\mu v\alpha)}{\partial\psi}\right\}+\frac{\partial(w\alpha)}{\partial z} \tag{2.52}$$

[§]But note that the upper several levels are allowed to move with the undulating sea surface like the σ-coordinate models. See Chapter 4 for details.

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

$$\begin{aligned}
&\int_{z_{k+1}}^{z_k} \frac{\partial \alpha}{\partial t} dz + \int_{z_{k+1}}^{z_k} \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (h_\psi u \alpha)}{\partial \mu} + \frac{\partial (h_\mu v \alpha)}{\partial \psi} \right\} dz + \int_{z_{k+1}}^{z_k} \frac{\partial (w\alpha)}{\partial z} dz \\
&\quad = \frac{\partial}{\partial t} \Big(\int_{z_{k+1}}^{z_k} \alpha dz \Big) + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial}{\partial \mu} \Big(\int_{z_{k+1}}^{z_k} h_\psi u \alpha \, dz \Big) + \frac{\partial}{\partial \psi} \Big(\int_{z_{k+1}}^{z_k} h_\mu v \alpha \, dz \Big) \right\} \\
&\quad\quad - \Big(\frac{\partial z_k}{\partial t} + \frac{u(z_k)}{h_\mu} \frac{\partial z_k}{\partial \mu} + \frac{v(z_k)}{h_\psi} \frac{\partial z_k}{\partial \psi} - w(z_k) \Big) \alpha(z_k) \\
&\quad\quad + \Big(\frac{\partial z_{k+1}}{\partial t} + \frac{u(z_{k+1})}{h_\mu} \frac{\partial z_{k+1}}{\partial \mu} + \frac{v(z_{k+1})}{h_\psi} \frac{\partial z_{k+1}}{\partial \psi} - w(z_{k+1}) \Big) \alpha(z_{k+1}).
\end{aligned} \tag{2.53}$$

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

$$\frac{\partial}{\partial t} \Big(\Delta z \alpha \Big)_{k+\frac{1}{2}} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial}{\partial \mu} \Big(h_\psi \Delta z u \alpha \Big)_{k+\frac{1}{2}} + \frac{\partial}{\partial \psi} \Big(h_\mu \Delta z v \alpha \Big)_{k+\frac{1}{2}} \right\}, \tag{2.54}$$

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

For an interior grid cell, the last two lines reduce to the difference between vertical advective fluxes,

$$w(z_k)\alpha(z_k) - w(z_{k+1})\alpha(z_{k+1}). \tag{2.55}$$

For the sea surface ($k = 0; z_0 = \eta$) and the bottom ($k = N; z_N = -H$), kinematic conditions (2.44) and (2.45) are used to give

$$-(P - E + R)\alpha(0) - w(z_1)\alpha(z_1) \tag{2.56}$$

at the surface and

$$w_{z_{N-1}} \alpha(z_{N-1}) \tag{2.57}$$

at the bottom.

Similarly, the vertical integral of the divergence of the sub-grid scale fluxes gives

$$\begin{aligned}
&\int_{z_{k+1}}^{z_k} \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (h_\psi F_\mu)}{\partial \mu} + \frac{\partial (h_\mu F_\psi)}{\partial \psi} \right\} dz + \int_{z_{k+1}}^{z_k} \frac{\partial F_z}{\partial z} dz \\
&\quad = \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial}{\partial \mu} \Big(\Delta z h_\psi F_\mu \Big)_{k+\frac{1}{2}} + \frac{\partial}{\partial \psi} \Big(\Delta z h_\mu F_\psi \Big)_{k+\frac{1}{2}} \right\} \\
&\quad\quad - \Big(\frac{F_\mu(z)}{h_\mu} \frac{\partial z}{\partial \mu} + \frac{F_\psi(z)}{h_\psi} \frac{\partial z}{\partial \psi} - F_z(z) \Big)_k + \Big(\frac{F_\mu(z)}{h_\mu} \frac{\partial z}{\partial \mu} + \frac{F_\psi(z)}{h_\psi} \frac{\partial z}{\partial \psi} - F_z(z) \Big)_{k+1}.
\end{aligned} \tag{2.58}$$

The quantity

$$F^\alpha_{\text{surf}} = (P - E + R)\alpha(0) + \Big(\frac{F_\mu(z_0)}{h_\mu} \frac{\partial z_0}{\partial \mu} + \frac{F_\psi(z_0)}{h_\psi} \frac{\partial z_0}{\partial \psi} - F_z(z_0) \Big) (\equiv -F^\alpha_{z\,0}) \tag{2.59}$$

taken from (2.56) and (2.58) could 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 (2.59) is the tracer transport by the fresh water flux, and the second term is the microstructure flux calculated by sub-grid scale (SGS) parameterizations such as bulk formula. We generalize this flux as a vertical flux (i.e., $F^\alpha_{\text{surf}} = -F^\alpha_{z\,0}$).

2.2.2 Momentum equation

By default, the momentum equation with hydrostatic and Boussinesq approximation is solved. Equations are (2.6) and (2.7). To integrate these equations in time, the instantaneous vector field and pressure should be known.

The vector field of the previous time level is used. The pressure field might be obtained by integrating the hydrostatic equation vertically:

$$P(\mu,\psi,z,t) = P_s(\mu,\psi,t) + g\int_z^0 \rho(\mu,\psi,z')dz'. \tag{2.60}$$

This equation indicates that the surface pressure $P_s(\mu,\psi,t)$ should be known.

There is no problem if the surface height η is known ($P_s(\mu,\psi,t) = \rho_0 g\eta$). To obtain the surface height, we should solve vertically integrated equations of motion. Since the external gravity waves caused by the rise and fall of the sea level have high phase speeds, a short time step is required to satisfy the CFL condition. However, when a target phenomenon has a longer time scale, the external gravity waves are usually not important. One might want to separate or remove this wave since its phase speed is two orders of magnitude greater than that of other waves.

Historically, external gravity waves are removed from the model by prohibiting the vertical movement of the sea surface (rigid-lid approximation). In this case, the vertically integrated equations lead to the vorticity equation in a form of the Poisson equation and are solved iteratively. The surface pressure is diagnostically obtained as pressure to push the lid.

When the sea surface is allowed to move vertically, the problem of external gravity waves can be resolved by separating the barotropic mode from the baroclinic mode. We can achieve a long time step for the baroclinic mode by reflecting a temporally averaged state of the barotropic mode that is sub-cycled with a short time step.

Since the free surface option is more suitable for parallel computation, only the free surface option is now officially supported by MRI.COM.

a. Barotropic mode

If we put

$$U = \int_{-H}^{\eta} u dz = \sum_{k=1}^{N} u_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}, \quad V = \int_{-H}^{\eta} v dz = \sum_{k=1}^{N} v_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}, \tag{2.61}$$

then the vertically summed semi-discrete momentum equations are

$$\frac{\partial U}{\partial t} - fV = -\frac{g(\eta+H)}{h_\mu}\frac{\partial \eta}{\partial \mu} + X, \tag{2.62}$$

$$\frac{\partial V}{\partial t} + fU = -\frac{g(\eta+H)}{h_\psi}\frac{\partial \eta}{\partial \psi} + Y, \tag{2.63}$$

where

$$\begin{aligned} X &= -\nabla_H \cdot \Big(\sum_{k=1}^{N} (\Delta z(u,\, v)u)_{k-\frac{1}{2}} \Big) - \sum_{k=1}^{N} \Big[\frac{v}{h_\mu h_\psi} \Big(\frac{\partial h_\mu}{\partial \psi} u - \frac{\partial h_\psi}{\partial \mu} v \Big) \Big]_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \\ &\quad - \sum_{k=1}^{N} \Big[\frac{1}{\rho_0} \frac{1}{h_\mu} \int_{z_{k-\frac{1}{2}}}^{0} g\rho_\mu dz' \Big] \Delta z_{k-\frac{1}{2}} + \sum_{k=1}^{N} (\Delta z \mathscr{V}_H^u)_{k-\frac{1}{2}} + F^u_{\text{surf}} \Delta z_{\frac{1}{2}} + F^u_{\text{bottom}} \Delta z_{N-\frac{1}{2}} \\ &(\quad \equiv \quad \sum_{k=1}^{N} F_\mu \,), \end{aligned} \tag{2.64}$$

$$\begin{aligned} Y &= -\nabla_H \cdot \Big(\sum_{k=1}^{N} (\Delta z(u,\, v)v)_{k-\frac{1}{2}} \Big) + \sum_{k=1}^{N} \Big[\frac{u}{h_\mu h_\psi} \Big(\frac{\partial h_\mu}{\partial \psi} u - \frac{\partial h_\psi}{\partial \mu} v \Big) \Big]_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \\ &\quad - \sum_{k=1}^{N} \Big[\frac{1}{\rho_0} \frac{1}{h_\psi} \int_{z_{k-\frac{1}{2}}}^{0} g\rho_\psi dz' \Big] \Delta z_{k-\frac{1}{2}} + \sum_{k=1}^{N} (\Delta z \mathscr{V}_H^v)_{k-\frac{1}{2}} + F^v_{\text{surf}} \Delta z_{\frac{1}{2}} + F^v_{\text{bottom}} \Delta z_{N-\frac{1}{2}} \\ &(\quad \equiv \quad \sum_{k=1}^{N} F_\psi \,). \end{aligned} \tag{2.65}$$

The vertically integrated continuity equation is given by

$$\frac{\partial \eta}{\partial t} + \frac{1}{h_\mu h_\psi}\Big(\frac{\partial (h_\psi U)}{\partial \mu} + \frac{\partial (h_\mu V)}{\partial \psi}\Big) = (P - E + R). \tag{2.66}$$

We solve these equations for U, V, and η. See Chapter 4 for details.

b. Baroclinic mode

To solve the baroclinic mode, we could use $P_s(\mu, \psi, t)$ that can be diagnostically obtained using barotropic equations (2.62), (2.63), and (2.66). However, we could avoid the procedure of obtaining $P_s(\mu, \psi, t)$ by using the method described below.

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

$$u = \hat{u} + \bar{u}, \tag{2.67}$$

$$v = \hat{v} + \bar{v}, \tag{2.68}$$

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 surface pressure gradient term is dropped:

$$\frac{u'_{k-\frac{1}{2}} \Delta z^{(\mathrm{new})}_{k-\frac{1}{2}} - u_{k-\frac{1}{2}} \Delta z^{(\mathrm{old})}_{k-\frac{1}{2}}}{\Delta t} = f v \Delta z_{k-\frac{1}{2}} + F_\mu, \tag{2.69}$$

$$\frac{v'_{k-\frac{1}{2}} \Delta z^{(\mathrm{new})}_{k-\frac{1}{2}} - v_{k-\frac{1}{2}} \Delta z^{(\mathrm{old})}_{k-\frac{1}{2}}}{\Delta t} = -f u \Delta z_{k-\frac{1}{2}} + F_\psi, \tag{2.70}$$

where

$$\begin{aligned} F_\mu = &-\nabla_H \cdot \Big(\Delta z(u,\, v)u\Big)_{k-\frac{1}{2}} - \Big[\frac{v}{h_\mu h_\psi}\Big(\frac{\partial h_\mu}{\partial \psi}u - \frac{\partial h_\psi}{\partial \mu}v\Big)\Big]_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \\ &- \Big[\frac{1}{\rho_0}\frac{1}{h_\mu}\int_{z_{k-\frac{1}{2}}}^{0} g\rho_\mu dz'\Big]\Delta z_{k-\frac{1}{2}} + (\Delta z \mathscr{V}^u_H)_{k-\frac{1}{2}} - F^u_{z\,k} + F^u_{z\,k+1}, \end{aligned} \tag{2.71}$$

$$\begin{aligned} F_\psi = &-\nabla_H \cdot \Big(\Delta z(u,\, v)v\Big)_{k-\frac{1}{2}} + \Big[\frac{u}{h_\mu h_\psi}\Big(\frac{\partial h_\mu}{\partial \psi}u - \frac{\partial h_\psi}{\partial \mu}v\Big)\Big]_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \\ &- \Big[\frac{1}{\rho_0}\frac{1}{h_\psi}\int_{z_{k-\frac{1}{2}}}^{0} g\rho_\psi dz'\Big]\Delta z_{k-\frac{1}{2}} + (\Delta z \mathscr{V}^v_H)_{k-\frac{1}{2}} - F^v_{z\,k} + F^v_{z\,k+1}. \end{aligned} \tag{2.72}$$

Summing over the whole water column gives

$$\frac{\sum_{k=1}^{N}(u'_{k-\frac{1}{2}} \Delta z^{(\mathrm{new})}_{k-\frac{1}{2}}) - \sum_{k=1}^{N}(u_{k-\frac{1}{2}} \Delta z^{(\mathrm{old})}_{k-\frac{1}{2}})}{\Delta t} = f \sum_{k=1}^{N} v_{k-\frac{1}{2}} \Delta z^{(\mathrm{old})}_{k-\frac{1}{2}} + \sum_{k=1}^{N} F_\mu, \tag{2.73}$$

$$\frac{\sum_{k=1}^{N}(v'_{k-\frac{1}{2}} \Delta z^{(\mathrm{new})}_{k-\frac{1}{2}}) - \sum_{k=1}^{N}(v_{k-\frac{1}{2}} \Delta z^{(\mathrm{old})}_{k-\frac{1}{2}})}{\Delta t} = -f \sum_{k=1}^{N} u_{k-\frac{1}{2}} \Delta z^{(\mathrm{old})}_{k-\frac{1}{2}} + \sum_{k=1}^{N} F_\psi. \tag{2.74}$$

From (2.62), (2.63), (2.64), (2.65), we have

$$\sum_{k=1}^{N} F_\mu = \frac{\sum_{k=1}^{N}(u^{(\mathrm{new})}_{k-\frac{1}{2}} \Delta z^{(\mathrm{new})}_{k-\frac{1}{2}}) - \sum_{k=1}^{N}(u_{k-\frac{1}{2}} \Delta z^{(\mathrm{old})}_{k-\frac{1}{2}})}{\Delta t} - f \sum_{k=1}^{N} v_{k-\frac{1}{2}} \Delta z^{(\mathrm{old})}_{k-\frac{1}{2}} + \frac{\eta^{(\mathrm{old})} + H}{\rho_0 h_\mu}\frac{\partial P_s}{\partial \mu}, \tag{2.75}$$

$$\sum_{k=1}^{N} F_\psi = \frac{\sum_{k=1}^{N}(v^{(\mathrm{new})}_{k-\frac{1}{2}} \Delta z^{(\mathrm{new})}_{k-\frac{1}{2}}) - \sum_{k=1}^{N}(v_{k-\frac{1}{2}} \Delta z^{(\mathrm{old})}_{k-\frac{1}{2}})}{\Delta t} + f \sum_{k=1}^{N} u_{k-\frac{1}{2}} \Delta z^{(\mathrm{old})}_{k-\frac{1}{2}} + \frac{\eta^{(\mathrm{old})} + H}{\rho_0 h_\psi}\frac{\partial P_s}{\partial \psi}. \tag{2.76}$$

From the above equations, we have

$$\frac{\sum_{k=1}^{N}(u_{k-\frac{1}{2}}^{(\mathrm{new})}-u_{k-\frac{1}{2}}^{'})\Delta z_{k-\frac{1}{2}}^{(\mathrm{new})}}{\Delta t}=\frac{\eta^{(\mathrm{old})}+H}{\rho_0 h_\mu}\frac{\partial P_s}{\partial \mu}, \tag{2.77}$$

$$\frac{\sum_{k=1}^{N}(v_{k-\frac{1}{2}}^{(\mathrm{new})}-v_{k-\frac{1}{2}}^{'})\Delta z_{k-\frac{1}{2}}^{(\mathrm{new})}}{\Delta t}=\frac{\eta^{(\mathrm{old})}+H}{\rho_0 h_\psi}\frac{\partial P_s}{\partial \psi}. \tag{2.78}$$

Thus we obtain

$$\frac{(\overline{u_{k-\frac{1}{2}}^{(\mathrm{new})}}^{z}+u_{k-\frac{1}{2}}^{'}-\overline{u_{k-\frac{1}{2}}^{'}}^{z})\Delta z_{k-\frac{1}{2}}^{(\mathrm{new})}-u_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}^{(\mathrm{old})}}{\Delta t}=fv\Delta z_{k-\frac{1}{2}}-\frac{\eta^{(\mathrm{old})}+H}{\eta^{(\mathrm{new})}+H}\frac{\Delta z_{k-\frac{1}{2}}^{(\mathrm{new})}}{\rho_0 h_\mu}\frac{\partial P_s}{\partial \mu}+F_\mu \tag{2.79}$$

$$\frac{(\overline{v_{k-\frac{1}{2}}^{(\mathrm{new})}}^{z}+v_{k-\frac{1}{2}}^{'}-\overline{v_{k-\frac{1}{2}}^{'}}^{z})\Delta z_{k-\frac{1}{2}}^{(\mathrm{new})}-v_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}^{(\mathrm{old})}}{\Delta t}=-fu\Delta z_{k-\frac{1}{2}}-\frac{\eta^{(\mathrm{old})}+H}{\eta^{(\mathrm{new})}+H}\frac{\Delta z_{k-\frac{1}{2}}^{(\mathrm{new})}}{\rho_0 h_\psi}\frac{\partial P_s}{\partial \psi}+F_\psi, \tag{2.80}$$

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

Since we could regard $\overline{u^{(\mathrm{new})}}^{z}+u'-\bar{u}'^{z}=u$ and $\overline{v^{(\mathrm{new})}}^{z}+v'-\bar{v}'^{z}=v$ as the real updated velocity, the baroclinic component is expressed as $\hat{u}=u'-\bar{u}'$ and $\hat{v}=v'-\bar{v}'$.

To summarize, we first solve for (u',v') using (2.69) and (2.70), and then compute the baroclinic component by $\hat{u}=u'-\bar{u}'$ and $\hat{v}=v'-\bar{v}'$. The absolute velocity is obtained by $u=\hat{u}+\bar{u}$ and $v=\hat{v}+\bar{v}$.

2.2.3 Continuity equation

The vertical component of velocity is obtained by vertically integrating the continuity equation from top to bottom. By using a flux form, the surface boundary condition (2.44) could be naturally included. The vertical integration for the k-th vertical level is performed as follows:

$$w_k=w_{k-1}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi\Delta z_{k-\frac{1}{2}}u_{k-\frac{1}{2}})}{\partial\mu}+\frac{\partial(h_\mu\Delta z_{k-\frac{1}{2}}v_{k-\frac{1}{2}})}{\partial\psi}\right\}, \tag{2.81}$$

where $\Delta z_{k-\frac{1}{2}}$ is the width of the $(k-\frac{1}{2})$-th layer with $\Delta z_{\frac{1}{2}}=\Delta z_{\frac{1}{2}\mathrm{const}}+\eta$ and

$$w_0=\frac{\partial\eta}{\partial t}-(P-E+R)=-\frac{1}{h_\psi h_\mu}\Big(\frac{\partial(h_\psi U)}{\partial\mu}+\frac{\partial(h_\mu V)}{\partial\psi}\Big). \tag{2.82}$$

2.2.4 Temperature and salinity equation

We solve for potential temperature instead of *in situ* temperature, because we want to express vertical mixing as a simple mixing of water masses. If we use *in situ* temperature, the temperature change caused by the change of pressure accompanied by vertical excursion of a water parcel should be calculated before mixing.

a. A semi-discrete expression

The equation for potential temperature and salinity is an advection-diffusion equation (2.14) and (2.15) (or (2.20)). Its semi-discrete expression is as follows:

$$\begin{aligned}\frac{\partial}{\partial t}(T_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}) =& -\nabla_H\cdot\left(\Delta z h_\psi uT,\ \Delta z h_\mu vT\right)_{k-\frac{1}{2}} - (wT)_{k-1} + (wT)_k \\ & -\nabla_H\cdot\left(\Delta z h_\psi F^T_\mu,\ \Delta z h_\mu F^T_\psi\right)_{k-\frac{1}{2}} - {F^T_z}_{k-1} + {F^T_z}_k + S^T\Delta z_{k-\frac{1}{2}}, \qquad (2.83)\\ \frac{\partial}{\partial t}(S_{k-\frac{1}{2}}\Delta z_{k-\frac{1}{2}}) =& -\nabla_H\cdot\left(\Delta z h_\psi uS,\ \Delta z h_\mu vS\right)_{k-\frac{1}{2}} - (wS)_{k-1} + (wS)_k \\ & -\nabla_H\cdot\left(\Delta z h_\psi F^S_\mu,\ \Delta z h_\mu F^S_\psi\right)_{k-\frac{1}{2}} - {F^S_z}_{k-1} + {F^S_z}_k + S^S\Delta z_{k-\frac{1}{2}}. \qquad (2.84)\end{aligned}$$

Several options for discretizing each term on the r.h.s. are detailed in Chapter 6.

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 6.4.

One might also choose to mix tracers by setting the local vertical diffusion coefficient to a large value such as $10000\,[\mathrm{cm^2 s^{-1}}]$ where stratification is unstable. In this case, the tracer equation should be solved using the partial implicit method, which is described in Section 12.5.

2.2.5 Equation of state

The *in situ* density is needed to calculate the pressure gradient term in the momentum equation. As indicated in (2.30), the equation of state is a function of pressure, temperature and salinity. Here we present the specific form of the equation of state.

a. Basics of the equation of state

The standard equation of state provided by UNESCO (1981) is a function of *in situ* temperature, salinity, and pressure. Note that *in situ* temperature is used, not potential temperature. Density (ρ_w) of pure water ($S=0$) under sea level pressure is given as a function of temperature (T):

$$\begin{aligned}\rho_w(T) =& 999.842594 + 6.793952\times10^{-2}T - 9.095290\times10^{-3}T^2 \\ & + 1.001685\times10^{-4}T^3 - 1.120083\times10^{-6}T^4 + 6.536332\times10^{-9}T^5.\end{aligned} \qquad (2.85)$$

Density at the sea surface ($\rho_0 = \rho(T,S,0)$) is expressed using sea surface temperature and salinity:

$$\begin{aligned}\rho_0 =& \rho_w \\ & + (0.824493 - 4.0899\times10^{-3}T + 7.6438\times10^{-5}T^2 - 8.2467\times10^{-7}T^3 + 5.3875\times10^{-9}T^4)S \\ & + (-5.72466\times10^{-3} + 1.0227\times10^{-4}T - 1.6546\times10^{-6}T^2)S^{\frac{3}{2}} \\ & + 4.8314\times10^{-4}S^2.\end{aligned} \qquad (2.86)$$

Density in the interior is calculated using the secant bulk modulus $K(S,T,P)$. The pure water value K_w is given by

$$\begin{aligned} K_w = {} & 19652.21 + 1.484206 \times 10^2 T - 2.327105 T^2 \\ & + 1.360477 \times 10^{-2} T^3 - 5.155288 \times 10^{-5} T^4. \end{aligned} \tag{2.87}$$

The value at the sea surface is given by

$$\begin{aligned} K_0 = {} & K_w + (54.6746 - 0.603459 T + 1.09987 \times 10^{-2} T^2 - 6.1670 \times 10^{-5} T^3) S \\ & + (7.944 \times 10^{-2} + 1.6483 \times 10^{-2} T - 5.3009 \times 10^{-4} T^2) S^{\frac{3}{2}}, \end{aligned} \tag{2.88}$$

and the value at pressure P is given by

$$\begin{aligned} K = {} & K_0 \\ & + P(3.239908 + 1.43713 \times 10^{-3} T + 1.16092 \times 10^{-4} T^2 - 5.77905 \times 10^{-7} T^3) \\ & + P(2.2838 \times 10^{-3} - 1.0981 \times 10^{-5} T - 1.6078 \times 10^{-6} T^2) S \\ & + P(1.91075 \times 10^{-4}) S^{\frac{3}{2}} \\ & + P^2 (8.50935 \times 10^{-5} - 6.12293 \times 10^{-6} T + 5.2787 \times 10^{-8} T^2) \\ & + P^2 (-9.9348 \times 10^{-7} + 2.0816 \times 10^{-8} T + 9.1697 \times 10^{-10} T^2) S. \end{aligned} \tag{2.89}$$

Density is computed using the following equations,

$$\rho = \rho_0 / (1 - P/K) \tag{2.90}$$

and

$$\sigma = \rho - 1000.0. \tag{2.91}$$

Since potential temperature (θ) is the prognostic variable, an equation of state should be given as a function of potential temperature, salinity, and pressure.

To do this, potential temperature should be converted to *in situ* temperature. The conversion equation is obtained as follows using the adiabatic lapse rate $\Gamma(T,S,P)$:

$$T(\theta_0, S, P) = \theta_0 + \int_{P_0}^{P} \Gamma(T,S,P') dP'. \tag{2.92}$$

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

$$\begin{aligned} \Gamma(T,S,P) = {} & a_0 + a_1 T + a_2 T^2 + a_3 T^3 \\ & + (b_0 + b_1 T)(S - 35) \\ & + \{c_0 + c_1 T + c_2 T^2 + c_3 T^3 + (d_0 + d_1 T)(S - 35)\} P \\ & + (e_0 + e_1 T + e_2 T^2) P^2, \end{aligned} \tag{2.93}$$

where

$$
\begin{aligned}
a_0 &= +3.5803\times10^{-5}, & c_2 &= +8.7330\times10^{-12},\\
a_1 &= +8.5258\times10^{-6}, & c_3 &= -5.4481\times10^{-14},\\
a_2 &= -6.8360\times10^{-8}, & d_0 &= -1.1351\times10^{-10},\\
a_3 &= -6.6228\times10^{-10}, & d_1 &= +2.7759\times10^{-12},\\
b_0 &= +1.8932\times10^{-6}, & e_0 &= -4.6206\times10^{-13},\\
b_1 &= -4.2393\times10^{-8}, & e_1 &= +1.8676\times10^{-14},\\
c_0 &= +1.8741\times10^{-8}, & e_2 &= -2.1687\times10^{-16},\\
c_1 &= -6.7795\times10^{-10}. &&
\end{aligned}
\tag{2.94}
$$

With the converted *in situ* temperature, the equation of state is used to calculate density using salinity and pressure.

b. An equation of state used by MRI.COM

An equation of state of MRI.COM is in the same polynomial form as UNESCO, but has a modified set of parameters. The parameters are determined by the least square fit for a realistic range of potential temperature and salinity. We follow the method of Ishizaki (1994), but prescribe a different range ($-2 \le \theta \le 40$[°C], $0 \le S \le 42$ [psu], and $0 \le P \le 1000$ [bar]).

We first calculate density at the sea surface (potential density or σ_θ) using equations (2.85) and (2.86) without any modification.

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

$$
\begin{aligned}
K(\theta,S,P) =& e_1(P)+e_2(P)\theta+e_3(P)\theta^2+e_4(P)\theta^3+e_5(P)\theta^4\\
&+S(f_1(P)+f_2(P)\theta+f_3(P)\theta^2+f_4(P)\theta^3)\\
&+S^{\frac{3}{2}}(f_5(P)+f_6(P)\theta+f_7(P)\theta^2),
\end{aligned}
\tag{2.95}
$$

where

$$
\begin{aligned}
e_1(P) &= ec_1+(gc_1+hc_1P)P, & f_1(P) &= fc_1+(gc_5+hc_4P)P,\\
e_2(P) &= ec_2+(gc_2+hc_2P)P, & f_2(P) &= fc_2+(gc_6+hc_5P)P,\\
e_3(P) &= ec_3+(gc_3+hc_3P)P, & f_3(P) &= fc_3+(gc_7+hc_6P)P,\\
e_4(P) &= ec_4+gc_4P, & f_4(P) &= fc_4,\\
e_5(P) &= ec_5, & f_5(P) &= fc_5+gc_8P,\\
& & f_6(P) &= fc_6,\\
& & f_7(P) &= fc_7.
\end{aligned}
\tag{2.96}
$$

The set of coefficients in the above equation is computed using a least square fit as follows. Given $43\times43\times101$ combinations of the above range of potential temperature, salinity, and pressure, *in situ* temperature is first computed using (2.92). 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 as follows.

ec_1	19659.35	fc_1	52.85624
ec_2	144.5863	fc_2	-3.128126×10^{-1}
ec_3	-1.722523	fc_3	6.456036×10^{-3}
ec_4	1.019238×10^{-2}	fc_4	-5.370396×10^{-5}
ec_5	-4.768276×10^{-5}	fc_5	3.884013×10^{-1}
		fc_6	9.116446×10^{-3}
		fc_7	-4.628163^{-4}

gc_1	3.185918	hc_1	2.111102×10^{-4}
gc_2	2.189412×10^{-2}	hc_2	-1.196438×10^{-5}
gc_3	-2.823685×10^{-4}	hc_3	1.364330×10^{-7}
gc_4	1.715739×10^{-6}	hc_4	-2.048755×10^{-6}
gc_5	6.703377×10^{-3}	hc_5	6.375979×10^{-8}
gc_6	-1.839953×10^{-4}	hc_6	5.240967×10^{-10}
gc_7	1.912264×10^{-7}		
gc_8	1.477291×10^{-4}		

2.3 Appendix

2.3.1 Physical constants

Here we list basic physical constants used for MRI.COM. These are defined in `param.F90`. Physical constants used to calculate surface fluxes and sea ice processes are listed in Chapters 8 and 9, respectively.

	value	variable name in MRI.COM
radius of the Earth	6375.0×10^5 cm	`RADIUS`
acceleration due to gravity	$980.1\,\mathrm{cm^2 \cdot s^{-1}}$	`GRAV`
angular velocity of the Earth's rotation	$\pi/43082.0\,\mathrm{radian \cdot s^{-1}}$	`OMEGA`
the absolute temperature of 0 °C	273.16 K	`TAB`
the average density of sea water	$1.036\,\mathrm{g \cdot cm^{-3}}$	`RO`
the specific heat of sea water	$3.99 \times 10^7\,\mathrm{erg \cdot g^{-1} \cdot K^{-1}}$ ($1.0\,\mathrm{erg \cdot g^{-1} \cdot K^{-1}}$ $= 1.0 \times 10^{-4}\,\mathrm{J \cdot kg^{-1} \cdot K^{-1}}$)	`CP`

References

Bryan, K., 1969: A numerical method for the study of the circulation of the world ocean, *J. Comput. Phys.*, *4*, 347-376.

Bryan, K., 1984: Accelerating the convergence to equilibrium of ocean-climate models, *J. Phys. Oceanogr.*, *14*, 666-673.

Gent, P. R., and J. C. McWilliams, 1990: Isopycnal mixing in ocean circulation models, *J. Phys. Oceanogr.*, *20*, 150-155.

Griffies, S. M., 1998: The Gent-McWilliams skew flux, *J. Phys. Oceanogr.*, *28*, 831-841.

Ishizaki, H., 1994: A Simulation of the abyssal circulation in the North Pacific Ocean. Part II: Theoretical Rationale, *J. Phys. Oceanogr.*, *24*, 1941-1954.

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.

Mellor, G. L., and A. Blumberg, 2004: Wave breaking and ocean surface layer thermal response, *J. Phys. Oceanogr.*, *34*, 693-698.

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.

Phillips, N., 1966: The equation of motion for a shallow rotating atmosphere and the "Traditional approximation", *J. Atmos. Sci.*, *23*, 626–628.

Redi, M. H., 1982: Oceanic isopycnal mixing by coordinate rotation, *J. Phys. Oceanogr.*, *12*, 1154–1158.

Smagorinsky, J., 1963: General circulation experiments with the primitive equations: I. The basic experiment, *Mon. Weather Rev.*, *91*, 99–164.

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*, 25pp.

Weatherly, G. L., 1972: A study of the bottom boundary layer of the Florida current, *J. Phys. Oceanogr.*, *2*, 54–72.

Chapter 3 Spatial grid arrangement and definition of continuity equation

The model ocean domain is defined as a three-dimensional aggregate of rectangular grid cells limited by latitudinal circles, longitudinal lines, and horizontal surfaces with fixed depths. Just above the bottom, vertical thickness of the cell can be locally changed. 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).

3.1 Horizontal grid arrangement

Figure 3.1(a) illustrates the horizontal grid arrangement. 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 3.1(a)); the tracer point as the T-point; and the grid cell centered on the T-point, the T-cell (Figure 3.3). The T-cells are staggered from the U-cells by a half grid size and consists of partial cells along the coast lines (Figure 3.3). 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 (CCSR, U. Tokyo, 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 3.1(b)).

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 3.1(c), 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.

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 3.2(a), tracers ($\circ$) and velocity ($\times$) are defined at just the mid-depth level of the cell, and the vertical mass fluxes W ($\triangle$, $\square$) are defined at the boundary of the cell. There are two kinds of W, one for the T-cell (W^T; $\triangle$) and another for the U-cell (W^U; $\square$), W^U being obtained by an averaging operation on W^T. Their horizontal locations are the T-points and the U-points depicted in Figure 3.1(a).

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, with a limitation that it must exceed 10 percent of the nominal thickness of the layer to avoid violating the vertical CFL condition (Figure 3.2(b)). Otherwise, as presented in Figure 3.2(c), the gentle bottom slope is expressed by wide, flat bottoms and cliffs here and there with height of vertical grid size Δz. In

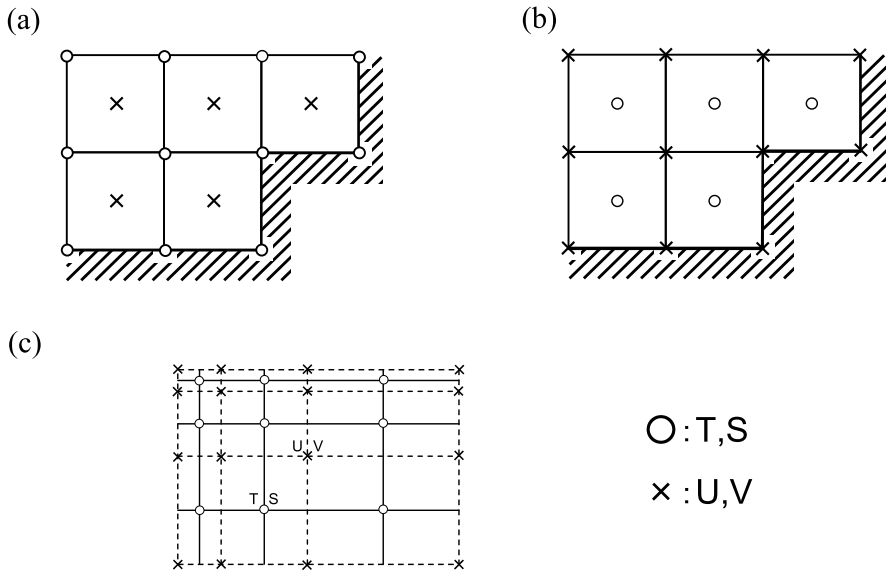

Figure 3.1. Horizontal grid arrangement. (a) MRI.COM ($\circ$:T,S,η, $\times$:u,v,H), (b) MOM and COCO ($\circ$:T,S,η,H, $\times$:u,v), (c) Variable grid size in MRI.COM

these regions, the vertical velocity is concentrated at the cliffs, resulting in relatively strong fictitious horizontal currents there.

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

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

where h_μ is the scaling factor. The 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 ΔS and the volume is expressed by ΔV.

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 $(i+\frac{1}{2}, j+\frac{1}{2})$. 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 3.2(a)). In some cases, the T-point and U-point levels may be numbered as k, with a notice.

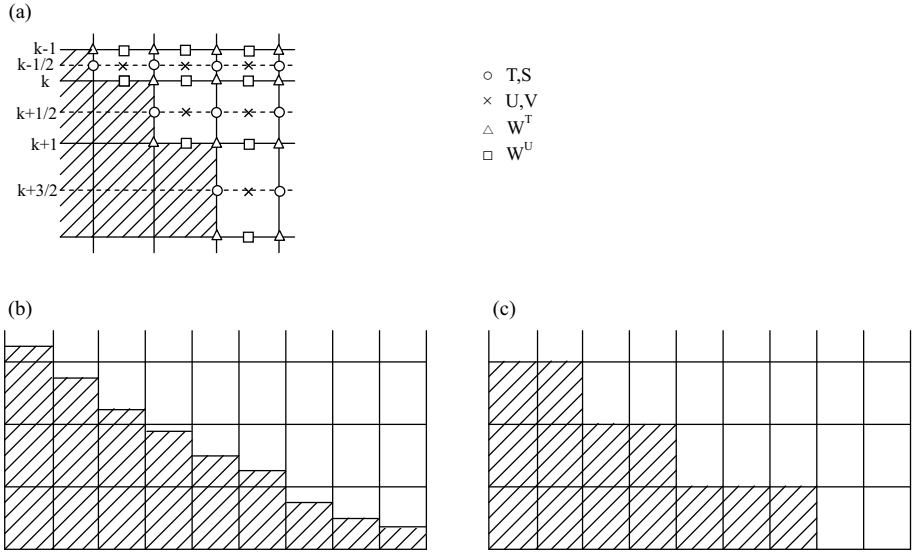

Figure 3.2. Vertical grid arrangement. (a)Vertical grid arrangement. (b) Smooth bottom slope. (c) Stair-like bottom slope.

3.4 Continuity equation

The mass (volume) fluxes, which are fundamental for estimating the advection of momentum and tracers, are calculated based on the finite difference expression of the continuity equation.

The finite difference expression of the continuity equation differs for the T-cell and U-cell (Figure 3.3). In MRI.COM, the mass continuity for the T-cell is fundamental 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).

The finite difference expression of the continuity equation for the T-cell is given as follows, based on the mass fluxes passing through each side of the grid cell (Figure 3.3):

$$\begin{aligned} MC^T_{i,j} &\equiv U^T_{i-\frac{1}{2},j} - U^T_{i+\frac{1}{2},j} + V^T_{i,j-\frac{1}{2}} - V^T_{i,j+\frac{1}{2}} + W^T_{i,j,k+1} - W^T_{i,j,k} \\ &= 0, \end{aligned} \tag{3.3}$$

where

$$U^T_{i+\frac{1}{2},j} = u^*_{i+\frac{1}{2},j}\Delta y_{i+\frac{1}{2},j}\Delta z, \quad V^T_{i,j+\frac{1}{2}} = v^*_{i,j+\frac{1}{2}}\Delta x_{i,j+\frac{1}{2}}\Delta z, \tag{3.4}$$

$$\begin{aligned} u^*_{i+\frac{1}{2},j} &= \frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}}), \\ v^*_{i,j+\frac{1}{2}} &= \frac{1}{2}(v_{i+\frac{1}{2},j+\frac{1}{2}} + v_{i-\frac{1}{2},j+\frac{1}{2}}). \end{aligned} \tag{3.5}$$

The finite difference analog of the continuity for the partial T-cell along the coastline (Figure 3.3(b)) is defined as follows:

$$V^T_{i,j+\frac{1}{2}} = \frac{1}{2}v^*_{i,j+\frac{1}{2}}\Delta x_{i,j+\frac{1}{2}}\Delta z, \quad V^T_{i,j-\frac{1}{2}} = \frac{1}{2}v^*_{i,j-\frac{1}{2}}\Delta x_{i,j-\frac{1}{2}}\Delta z, \tag{3.6}$$

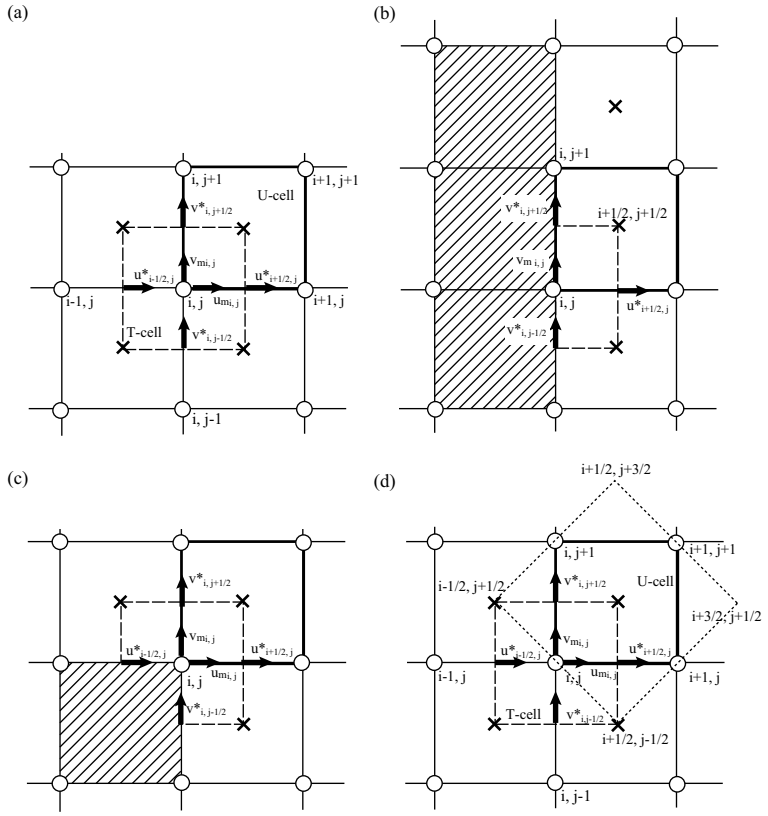

Figure 3.3. Horizontal arrangement of variables for the continuity equation. (a) Relationship between T-cell and U-cell (standard form). (b),(c) Relationship between T-cell and U-cell near the coast. (d) Diagonal square grid cell and mass fluxes.

$$v^*_{i,j+\frac{1}{2}} = v_{i+\frac{1}{2},j+\frac{1}{2}}, \qquad v^*_{i,j-\frac{1}{2}} = v_{i+\frac{1}{2},j-\frac{1}{2}}, \qquad u^*_{i-\frac{1}{2},j} = 0. \tag{3.7}$$

For the corner part of land as shown in Figure 3.3(c), it is given as follows:

$$U^T_{i-\frac{1}{2},j} = \frac{1}{2}u^*_{i-\frac{1}{2},j}\Delta y_{i-\frac{1}{2},j}\Delta z, \quad V^T_{i,j-\frac{1}{2}} = \frac{1}{2}v^*_{i,j-\frac{1}{2}}\Delta x_{i,j-\frac{1}{2}}\Delta z, \tag{3.8}$$

$$u^*_{i-\frac{1}{2},j} = u_{i-\frac{1}{2},j+\frac{1}{2}}, \quad v^*_{i,j-\frac{1}{2}} = v_{i+\frac{1}{2},j-\frac{1}{2}}. \tag{3.9}$$

The boundary condition for $W^T_{i,j}$ is as follows:

$$W^T_{i,j} = \left(U_{i-\frac{1}{2},j}\Delta y_{i-\frac{1}{2},j} - U_{i+\frac{1}{2},j}\Delta y_{i+\frac{1}{2},j} + V_{i,j-\frac{1}{2}}\Delta x_{i,j-\frac{1}{2}} - V_{i,j+\frac{1}{2}}\Delta x_{i,j+\frac{1}{2}}\right)/\Delta S_{i,j}$$

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

$$W^T_{i,j} = 0$$

at the bottom.

On the other hand, the finite difference expression of the continuity equation for the U-cell $(i+\frac{1}{2}, j+\frac{1}{2})$ is defined as follows (Figure 3.3(a, b, and c)):

$$\begin{aligned} MC^U_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} &\equiv \frac{MC^T_{i,j,k+\frac{1}{2}}}{N_{i,j,k+\frac{1}{2}}} + \frac{MC^T_{i+1,j,k+\frac{1}{2}}}{N_{i+1,j,k+\frac{1}{2}}} + \frac{MC^T_{i,j+1,k+\frac{1}{2}}}{N_{i,j+1,k+\frac{1}{2}}} + \frac{MC^T_{i+1,j+1,k+\frac{1}{2}}}{N_{i+1,j+1,k+\frac{1}{2}}} \\ &= 0, \end{aligned} \tag{3.10}$$

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 3.3(a)), but $N<4$ for the partial T-cells along coast lines (Figure 3.3(b,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:

$$\begin{aligned} MC^U_{i+\frac{1}{2},j+\frac{1}{2}} &\equiv \frac{1}{4}(MC^T_{i,j}+MC^T_{i+1,j}+MC^T_{i,j+1}+MC^T_{i+1,j+1}) \\ &= 0. \end{aligned} \tag{3.11}$$

This can be rewritten as

$$\begin{aligned} &\frac{1}{2}(U^U_{i,j}+U^U_{i,j+1})-\frac{1}{2}(U^U_{i+1,j}+U^U_{i+1,j+1})+\frac{1}{2}(V^U_{i,j}+V^U_{i+1,j}) \\ &\quad -\frac{1}{2}(V^U_{i,j+1}+V^U_{i+1,j+1}) = W^U_{i+\frac{1}{2},j+\frac{1}{2},k}-W^U_{i+\frac{1}{2},j+\frac{1}{2},k+1}. \end{aligned} \tag{3.12}$$

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

$$U^U_{i,j} = u_{mi,j}\Delta y_{i,j}\Delta z, \quad V^U_{i,j} = v_{mi,j}\Delta x_{i,j}\Delta z, \tag{3.13}$$

$$\begin{aligned} u_{mi,j} &= \frac{1}{2}(u^*_{i+\frac{1}{2},j}+u^*_{i-\frac{1}{2},j}), \\ v_{mi,j} &= \frac{1}{2}(v^*_{i,j+\frac{1}{2}}+v^*_{i,j-\frac{1}{2}}), \end{aligned} \tag{3.14}$$

$$W^U_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{1}{4}(W^T_{i,j,k}+W^T_{i+1,j,k}+W^T_{i,j+1,k}+W^T_{i+1,j+1,k}). \tag{3.15}$$

Thus, for the standard form of the U-cell continuity, the following equations are derived from (3.4), (3.13), and (3.14):

$$\begin{aligned} U^U_{i,j} &= \frac{1}{2}(U^T_{i+\frac{1}{2},j}+U^T_{i-\frac{1}{2},j}), \\ V^U_{i,j} &= \frac{1}{2}(V^T_{i,j+\frac{1}{2}}+V^T_{i,j-\frac{1}{2}}). \end{aligned} \tag{3.16}$$

All the above relationships hold for the cases with variable grid sizes (Figure 3.1(c)).

The l.h.s. of the standard form of the continuity equation (3.12) 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 (3.12) 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 (3.12):

$$\begin{aligned} &\frac{1}{2}(U^U_{i,j}+V^U_{i,j})-\frac{1}{2}(U^U_{i+1,j+1}+V^U_{i+1,j+1})+\frac{1}{2}(U^U_{i,j+1}-V^U_{i,j+1}) \\ &\quad -\frac{1}{2}(U^U_{i+1,j}-V^U_{i+1,j}) = W^U_{i+\frac{1}{2},j+\frac{1}{2},k}-W^U_{i+\frac{1}{2},j+\frac{1}{2},k+1}. \end{aligned} \tag{3.17}$$

Let us explain the meaning taking the first term on the l.h.s. of (3.17)

$$U^U_{i,j}+V^U_{i,j} = \frac{1}{2}\left\{(U^T_{i-\frac{1}{2},j}+V^T_{i,j-\frac{1}{2}})+(U^T_{i+\frac{1}{2},j}+V^T_{i,j+\frac{1}{2}})\right\} \tag{3.18}$$

as an example, where (3.16) is used. If the flow is horizontally nondivergent, the horizontal mass fluxes $U^T_{i-\frac{1}{2},j}$ and $V^T_{i,j-\frac{1}{2}}$ in the first term on the r.h.s. are expressed by streamfunction at two pairs of U-points, $(i-\frac{1}{2},j+\frac{1}{2})$ and $(i-\frac{1}{2},j-\frac{1}{2})$, and $(i-\frac{1}{2},j-\frac{1}{2})$ and $(i+\frac{1}{2},j-\frac{1}{2})$, respectively. Then, their sum corresponds to the net mass flux crossing the diagonal section connecting the two U-points $(i-\frac{1}{2},j+\frac{1}{2})$ and $(i+\frac{1}{2},j-\frac{1}{2})$ (Figure 3.3(d)). 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 (3.17) means the horizontal mass convergence in the diagonal square defined by four U-points $(i-\frac{1}{2},j+\frac{1}{2})$, $(i+\frac{1}{2},j-\frac{1}{2})$, $(i+\frac{3}{2},j+\frac{1}{2})$, and $(i+\frac{1}{2},j+\frac{3}{2})$. Multiplying by a factor of $\frac{1}{2}$, the l.h.s. of (3.17) 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 l.h.s. of (3.12) as $A_{i+\frac{1}{2},j+\frac{1}{2}}$ and that of (3.17) 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:

$$\alpha\, A_{i+\frac{1}{2},j+\frac{1}{2}} + \beta\, B_{i+\frac{1}{2},j+\frac{1}{2}} = W^U_{i+\frac{1}{2},j+\frac{1}{2},k} - W^U_{i+\frac{1}{2},j+\frac{1}{2},k+1}, \tag{3.19}$$

where

$$\alpha+\beta = 1. \tag{3.20}$$

As shown later in Chapter 5, $(\alpha,\beta) = (2/3, 1/3)$ 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 $(\alpha,\beta) = (1,0)$.

3.5 Calculation of area

3.5.1 General orthogonal coordinates

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. In those situations, it is necessary to know the area and volume of the grid cells. These are numerically calculated as follows.

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

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

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_bl`) is approximated numerically as follows taking $\mu_1 = \mu(i), \mu_2 = \mu(i+\frac{1}{2})$, and $\psi_1 = \psi(j)$:

$$\sum_{m=1}^{M} L\Big[\lambda\Big(\mu_1+(m-1)\delta\mu,\psi_1\Big),\phi\Big(\mu_1+(m-1)\delta\mu,\psi_1\Big),\lambda\Big(\mu_1+m\delta\mu,\psi_1\Big),\phi\Big(\mu_1+m\delta\mu,\psi_1\Big)\Big]. \tag{3.21}$$

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

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

ψ_2), calculated as:

$$\begin{aligned}
\sum_{n=1}^{N}\sum_{m=1}^{M} L\Big[&\lambda\Big(\mu_1+(m-1)\delta\mu,\psi_1+(n-\frac{1}{2})\delta\psi\Big), \qquad \phi\Big(\mu_1+(m-1)\delta\mu,\psi_1+(n-\frac{1}{2})\delta\psi\Big),\\
&\lambda\Big(\mu_1+m\delta\mu,\psi_1+(n-\frac{1}{2})\delta\psi\Big), \qquad \phi\Big(\mu_1+m\delta\mu,\psi_1+(n-\frac{1}{2})\delta\psi\Big)\Big]\\
&\times\\
L\Big[&\lambda\Big(\mu_1+(m-\frac{1}{2})\delta\mu,\psi_1+(n-1)\delta\psi\Big), \qquad \phi\Big(\mu_1+(m-\frac{1}{2})\delta\mu,\psi_1+(n-1)\delta\psi\Big),\\
&\lambda\Big(\mu_1+(m-\frac{1}{2})\delta\mu,\psi_1+n\delta\psi\Big), \qquad \phi\Big(\mu_1+(m-\frac{1}{2})\delta\mu,\psi_1+n\delta\psi\Big)\Big]. \qquad (3.22)
\end{aligned}$$

As depicted in Figure 3.4, $\mathtt{a\_bl}_{i,j}$ is the area of the lower left quarter of the central U-point. Those for the lower right $\mathtt{a\_br}_{i,j}$, upper left $\mathtt{a\_tl}_{i,j}$, and upper right $\mathtt{a\_tr}_{i,j}$ quarters are obtained similarly.

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

$$\mathtt{areau}_{i,j} = \mathtt{a\_bl}_{i,j} + \mathtt{a\_br}_{i,j} + \mathtt{a\_tl}_{i,j} + \mathtt{a\_tr}_{i,j}, \qquad (3.23)$$

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

$$\mathtt{areat}_{i,j} = \mathtt{a\_bl}_{i,j} + \mathtt{a\_br}_{i-1,j} + \mathtt{a\_tl}_{i,j-1} + \mathtt{a\_tr}_{i-1,j-1}. \qquad (3.24)$$

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

$$\mathtt{areau}_{i+\frac{1}{2},j+\frac{1}{2}} = \mathtt{a\_bl}_{i+\frac{1}{2},j+\frac{1}{2}} + \mathtt{a\_br}_{i+\frac{1}{2},j+\frac{1}{2}} + \mathtt{a\_tl}_{i+\frac{1}{2},j+\frac{1}{2}} + \mathtt{a\_tr}_{i+\frac{1}{2},j+\frac{1}{2}}, \qquad (3.25)$$

$$\mathtt{areat}_{i,j} = \mathtt{a\_bl}_{i+\frac{1}{2},j+\frac{1}{2}} + \mathtt{a\_br}_{i-\frac{1}{2},j+\frac{1}{2}} + \mathtt{a\_tl}_{i+\frac{1}{2},j-\frac{1}{2}} + \mathtt{a\_tr}_{i-\frac{1}{2},j-\frac{1}{2}}. \qquad (3.26)$$

3.5.2 Geographic coordinate

Let us examine the situation of T-cell quarterly divided (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 3.4, 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:

$$\Delta s = a\frac{\Delta\lambda}{2}\cos\phi. \qquad (3.27)$$

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

$$\begin{aligned}
\mathtt{anhft}_{i,j} &= \int_{\phi}^{\phi+\frac{\Delta\phi}{2}} \Delta s a d\phi = \frac{a^2\Delta\lambda}{2}\int_{\phi}^{\phi+\frac{\Delta\phi}{2}}\cos\phi d\phi = \frac{a^2\Delta\lambda}{2}\left\{\sin\left(\phi+\frac{\Delta\phi}{2}\right)-\sin\phi\right\}\\
&= a^2\Delta\lambda\cos\left(\phi+\frac{\Delta\phi}{4}\right)\sin\frac{\Delta\phi}{4}\\
&= a^2\Delta\lambda\left(\cos\phi\cos\frac{\Delta\phi}{4}-\sin\phi\sin\frac{\Delta\phi}{4}\right)\sin\frac{\Delta\phi}{4}\\
&= a^2\Delta\lambda\cos\phi\cos\frac{\Delta\phi}{4}\sin\frac{\Delta\phi}{4}\left(1-\tan\phi\tan\frac{\Delta\phi}{4}\right)\\
&= \frac{a^2}{2}\Delta\lambda\cos\phi\sin\frac{\Delta\phi}{2}\left(1-\tan\phi\tan\frac{\Delta\phi}{4}\right). \qquad (3.28)
\end{aligned}$$

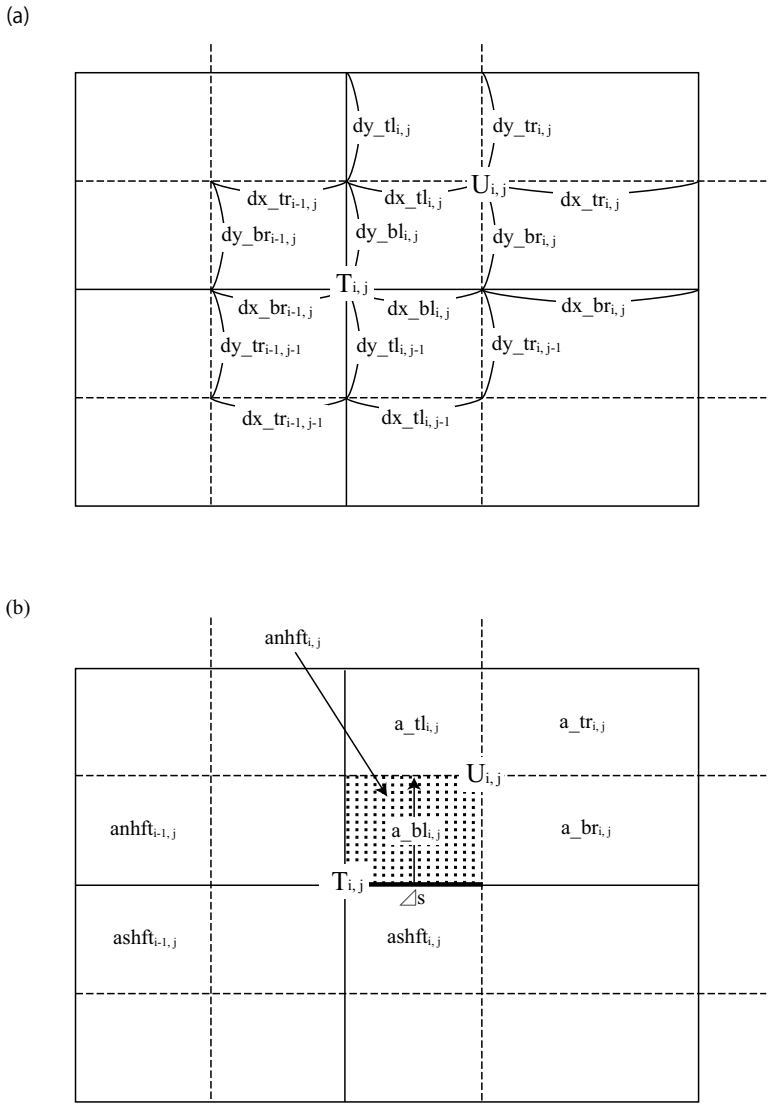

Figure 3.4. Variables that define a grid unit. (a) distance. (b) area. Grid indices (i, j) follow array indices in program codes.

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

$$\texttt{ashft}_{i,j} = \frac{a^2}{2}\Delta\lambda \cos\phi \sin\frac{\Delta\phi}{2}\left(1+\tan\phi\tan\frac{\Delta\phi}{4}\right). \tag{3.29}$$

At the north and the south poles, where $\phi = \pm 90^\circ$, we obtain the following by changing (3.28) and (3.29) to the following forms.

$$\texttt{anhft}_{i,j} = \frac{a^2}{2}\Delta\lambda \sin\frac{\Delta\phi}{2}\left(\cos\phi-\sin\phi\tan\frac{\Delta\phi}{4}\right) \tag{3.30}$$

$$\texttt{ashft}_{i,j} = \frac{a^2}{2}\Delta\lambda \sin\frac{\Delta\phi}{2}\left(\cos\phi+\sin\phi\tan\frac{\Delta\phi}{4}\right). \tag{3.31}$$

At the north pole:

$$\texttt{anhft}_{i,j} = 0 \tag{3.32}$$

$$\texttt{ashft}_{i,j} = \frac{a^2}{2}\Delta\lambda \sin\frac{\Delta\phi}{2}\tan\frac{\Delta\phi}{4}. \tag{3.33}$$

At the south pole:

$$\texttt{anhft}_{i,j} = \frac{a^2}{2}\Delta\lambda \sin\frac{\Delta\phi}{2}\tan\frac{\Delta\phi}{4} \tag{3.34}$$

$$\texttt{ashft}_{i,j} = 0. \tag{3.35}$$

In our model

$$\texttt{a\_bl}_{i,j} = \texttt{anhft}_{i,j}, \quad \texttt{a\_br}_{i,j} = \texttt{anhft}_{i+1,j},$$
$$\texttt{a\_tl}_{i,j} = \texttt{ashft}_{i,j+1}, \quad \texttt{a\_tr}_{i,j} = \texttt{ashft}_{i+1,j+1},$$

and the areas of the grid cells centered on the U-points and T-points are calculated by (3.23) and (3.24).

References

Arakawa, A., 1972: Design of the UCLA general circulation model, *Numerical Simulation Weather and Climate, Tech. Rep. No. 7*, *Dep. of Meteorology, University of California, Los Angeles*, 116 pp.

Webb, D. J., 1995: The vertical advection of momentum in Bryan-Cox-Semtner ocean general circulation models, *J. Phys. Oceanogr.*, *25*, 3186–3195.

Part II

Main Processes

Chapter 4 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 the solution is of order N, where N is the larger number of grid points in the horizontal direction. 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 in each long baroclinic time step is roughly the ratio of linear 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 formulations have numerous advantages over the iterative methods.

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. However, this 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.†

To remedy this problem, the σ-coordinates are introduced for the free-surface formulation near the sea surface. In this method, the thicknesses of the several upper layers ($z \leq -H_B$) vary as the sea surface height does (Figure 4.1). These layers are referred to as the σ-layers. The σ-coordinates are written as

$$\sigma \equiv \frac{z-\eta}{H_B+\eta}. \tag{4.1}$$

This extension of the free-surface formulation is described in Section 4.4.

The free-surface formulation was adopted in the Bryan-Cox-Semtner numerical ocean general circulation models by Killworth et al (1991). The σ-coordinates were adopted in the free-surface formulation by Hasumi (2006). In this chapter, we explain the free-surface formulation of MRI.COM adopting these two methods.

*A North Pacific Model with $1/4^\circ \times 1/6^\circ$ resolution has 742 grid points in the zonal direction.

†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).

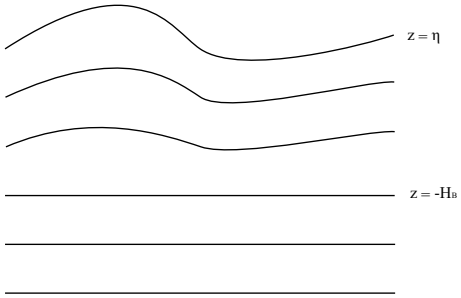

Figure 4.1. Schematic of the near surface σ-layers.

4.1 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 momentum equation is given as

$$\frac{\partial U}{\partial t} - fV = -\frac{g(\eta+H)}{h_\mu}\frac{\partial \eta}{\partial \mu} + X, \tag{4.2}$$

$$\frac{\partial V}{\partial t} + fU = -\frac{g(\eta+H)}{h_\psi}\frac{\partial \eta}{\partial \psi} + Y, \tag{4.3}$$

where

$$\begin{aligned} X = & -\nabla_H \cdot \left\{ \sum_{k=1}^{N} \left(\Delta z(u,\, v)u\right)_{k-\frac{1}{2}} \right\} - \sum_{k=1}^{N} \left[\frac{v}{h_\mu h_\psi} \left(\frac{\partial h_\mu}{\partial \psi} u - \frac{\partial h_\psi}{\partial \mu} v \right) \right]_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \\ & - \sum_{k=1}^{N} \left[\frac{1}{\rho_0} \frac{1}{h_\mu} \int_{z_{k-\frac{1}{2}}}^{0} g\rho_\mu dz' \right] \Delta z_{k-\frac{1}{2}} + \sum_{k=1}^{N} (\Delta z \mathcal{V}_H^u)_{k-\frac{1}{2}} + F^u_{\text{surf}} \Delta z_{\frac{1}{2}} + F^u_{\text{bottom}} \Delta z_{N-\frac{1}{2}} \end{aligned} \tag{4.4}$$

$$\begin{aligned} Y = & -\nabla_H \cdot \left\{ \sum_{k=1}^{N} \left(\Delta z(u,\, v)v\right)_{k-\frac{1}{2}} \right\} + \sum_{k=1}^{N} \left[\frac{u}{h_\mu h_\psi} \left(\frac{\partial h_\mu}{\partial \psi} u - \frac{\partial h_\psi}{\partial \mu} v \right) \right]_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}} \\ & - \sum_{k=1}^{N} \left[\frac{1}{\rho_0} \frac{1}{h_\psi} \int_{z_{k-\frac{1}{2}}}^{0} g\rho_\psi dz' \right] \Delta z_{k-\frac{1}{2}} + \sum_{k=1}^{N} (\Delta z \mathcal{V}_H^v)_{k-\frac{1}{2}} + F^v_{\text{surf}} \Delta z_{\frac{1}{2}} + F^v_{\text{bottom}} \Delta z_{N-\frac{1}{2}}. \end{aligned} \tag{4.5}$$

The continuity equation is

$$\frac{\partial \eta}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (h_\psi U)}{\partial \mu} + \frac{\partial (h_\mu V)}{\partial \psi} \right\} = (P - E + R), \tag{4.6}$$

where P is precipitation (positive downward), E is evaporation (positive upward) and R is the river discharge rate (positive into the ocean).

Figure 4.2 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. (4.2) and (4.3) are calculated in the subroutine for the baroclinic component and defined at U-points.

4.2 Time integration

Figure 4.3 presents schematics of the time integration of the barotropic modes in the free-surface formulation. When the time integration of the baroclinic mode is performed from step n ($t = t_n$) to step $n+1$ ($t = t_{n+1}$, $\Delta t =$

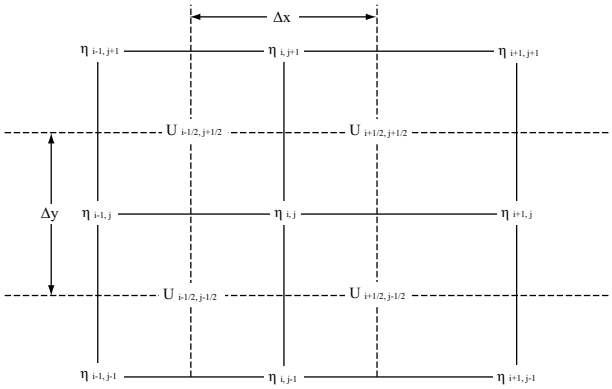

Figure 4.2. Grid arrangement of the free-surface formulation

$t_{n+1} - t_n$), the corresponding time integration of the barotropic mode is carried out from step n to step $n+2$ using the barotropic time interval Δt_{tr} and the vertically integrated values (X, Y) at $t = t_n$ calculated in the program that solves the baroclinic mode. The time-averaged value of the barotropic mode over the two baroclinic time intervals between $t = t_n$ and $t = t_{n+2}$ is used to calculate the total 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 (Meisinger 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 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 this Euler forward-backward scheme because the time filter is applied for the barotropic mode.[‡]

The finite-difference expression of the continuity equation (Eq. 4.6) is

$$\frac{(\eta'_{i,j} - \eta_{i,j})}{\Delta t_{tr}} + \frac{1}{(h_\mu h_\psi)_{i,j}} \left[(\delta_\mu \overline{h_\psi U}^{\psi})_{i,j} + (\delta_\psi \overline{h_\mu V}^{\mu})_{i,j} \right] = (P - E + R)_{i,j}, \tag{4.7}$$

where the subscripts are labeled on the basis of T-grid points. The variables $\eta_{i,j}$ and $w_{i,j}$ are located at the T-grid 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-grid points. (They are located at $(i+\frac{1}{2}, j+\frac{1}{2})$ on the basis of T-grid points; see Figure 3.2).

The finite-difference operator is defined as follows:

$$\begin{aligned} \delta_\mu A_i &\equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta \mu_i} \quad , \quad \delta_\mu A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_i}{\Delta \mu_{i+\frac{1}{2}}}, \\ \overline{A_i}^{\mu} &\equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2} \quad , \quad \overline{A_{i+\frac{1}{2}}}^{\mu} \equiv \frac{A_{i+1} + A_i}{2}. \end{aligned} \tag{4.8}$$

[‡]An option (`FSEB`) in MRI.COM uses the Euler backward scheme for the barotropic equations.

The same applies to ψ. In the program codes, the above variables are multiplied by the area of each model cell at the T-grid point (ΔS_T):

$$(\eta'_{i,j}-\eta_{i,j})\cdot\Delta S_{T\,i,j} \tag{4.9}$$
$$= \Delta t_{tr}\cdot\left\{(P-E+R)_{i,j}\cdot\Delta S_{T\,i,j}-\left(\Delta y_{i+\frac{1}{2},j}\overline{U}^{\psi}_{i+\frac{1}{2},j}-\Delta y_{i-\frac{1}{2},j}\overline{U}^{\psi}_{i-\frac{1}{2},j}\right)-\left(\Delta x_{i,j+\frac{1}{2}}\overline{V}^{\mu}_{i,j+\frac{1}{2}}-\Delta x_{i,j-\frac{1}{2}}\overline{V}^{\mu}_{i,j-\frac{1}{2}}\right)\right\}.$$

Each operator is defined the same way as the previous one. This equation is used to obtain the new surface elevation, $\eta'_{i,j}$.

After obtaining $\eta'_{i,j}$, the momentum equations, Eqs. (4.2) and (4.3), are solved. A longer time step can be used when the semi-implicit scheme is applied for the Coriolis terms in the momentum equations. Their finite-difference forms are

$$\frac{(U'_{i+\frac{1}{2},j+\frac{1}{2}}-U_{i+\frac{1}{2},j+\frac{1}{2}})}{\Delta t_{tr}}-\frac{f(V'_{i+\frac{1}{2},j+\frac{1}{2}}+V_{i+\frac{1}{2},j+\frac{1}{2}})}{2}=-\frac{g(H_{i+\frac{1}{2},j+\frac{1}{2}}+\overline{\overline{\eta'}}^{\mu,\psi}_{i+\frac{1}{2},j+\frac{1}{2}})}{(h_\mu)_{i+\frac{1}{2},j+\frac{1}{2}}}\delta_\mu\overline{\eta'}^{\psi}_{i+\frac{1}{2},j+\frac{1}{2}}+X_{i+\frac{1}{2},j+\frac{1}{2}} \tag{4.10}$$

$$\frac{(V'_{i+\frac{1}{2},j+\frac{1}{2}}-V_{i+\frac{1}{2},j+\frac{1}{2}})}{\Delta t_{tr}}+\frac{f(U'_{i+\frac{1}{2},j+\frac{1}{2}}+U_{i+\frac{1}{2},j+\frac{1}{2}})}{2}=-\frac{g(H_{i+\frac{1}{2},j+\frac{1}{2}}+\overline{\overline{\eta'}}^{\mu,\psi}_{i+\frac{1}{2},j+\frac{1}{2}})}{(h_\psi)_{i+\frac{1}{2},j+\frac{1}{2}}}\delta_\psi\overline{\eta'}^{\mu}_{i+\frac{1}{2},j+\frac{1}{2}}+Y_{i+\frac{1}{2},j+\frac{1}{2}}. \tag{4.11}$$

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

$$(U'_{i+\frac{1}{2},j+\frac{1}{2}}-U_{i+\frac{1}{2},j+\frac{1}{2}})-\frac{f\Delta t_{tr}}{2}(V'_{i+\frac{1}{2},j+\frac{1}{2}}+V_{i+\frac{1}{2},j+\frac{1}{2}}) = \Delta t_{tr}GX_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{4.12}$$
$$(V'_{i+\frac{1}{2},j+\frac{1}{2}}-V_{i+\frac{1}{2},j+\frac{1}{2}})+\frac{f\Delta t_{tr}}{2}(U'_{i+\frac{1}{2},j+\frac{1}{2}}+U_{i+\frac{1}{2},j+\frac{1}{2}}) = \Delta t_{tr}GY_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{4.13}$$

leading to

$$U'_{i+\frac{1}{2},j+\frac{1}{2}}-\frac{f\Delta t_{tr}}{2}V'_{i+\frac{1}{2},j+\frac{1}{2}} = U_{i+\frac{1}{2},j+\frac{1}{2}}+\frac{f\Delta t_{tr}}{2}V_{i+\frac{1}{2},j+\frac{1}{2}}+\Delta t_{tr}GX_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{4.14}$$
$$V'_{i+\frac{1}{2},j+\frac{1}{2}}+\frac{f\Delta t_{tr}}{2}U'_{i+\frac{1}{2},j+\frac{1}{2}} = V_{i+\frac{1}{2},j+\frac{1}{2}}-\frac{f\Delta t_{tr}}{2}U_{i+\frac{1}{2},j+\frac{1}{2}}+\Delta t_{tr}GY_{i+\frac{1}{2},j+\frac{1}{2}}. \tag{4.15}$$

Letting the r.h.s. of the above equations be RX and RY, we could simplify the equations as follows:

$$U'_{i+\frac{1}{2},j+\frac{1}{2}} = \left\{RX_{i+\frac{1}{2},j+\frac{1}{2}}+\frac{f\Delta t_{tr}}{2}RY_{i+\frac{1}{2},j+\frac{1}{2}}\right\}\Big/\left\{1+(\frac{f\Delta t_{tr}}{2})^2\right\}, \tag{4.16}$$
$$V'_{i+\frac{1}{2},j+\frac{1}{2}} = \left\{RY_{i+\frac{1}{2},j+\frac{1}{2}}-\frac{f\Delta t_{tr}}{2}RX_{i+\frac{1}{2},j+\frac{1}{2}}\right\}\Big/\left\{1+(\frac{f\Delta t_{tr}}{2})^2\right\}. \tag{4.17}$$

We obtain η, U, and V at $t=t_{n+1}$ by averaging them between $[t_n,t_{n+2}]$. We can see from Figure 4.3 that there are $2\times\Delta t/\Delta t_{tr}$ barotropic time steps between the two baroclinic time intervals $[t_n,t_{n+2}]$. Thus the value at $t=t_{n+1}$ is the average of $2\times\Delta t/\Delta t_{tr}+1$ barotropic steps centered at $t=t_{n+1}$. It is not always necessary to calculate the governing equations for the barotropic modes until $t=t_{n+2}$ to obtain the value at $t=t_{n+1}$. We can use the value at $t=t_{n+1}$ without any time averaging when they are calculated until $t=t_{n+1}$. This method requires less computational cost than the above method. We apply the time-filter method because we empirically know that this method is more computationally stable.

The degree of the time-filter procedure can be specified by the namelist parameter `ntflt`, which specifies the additional time steps to be calculated past the time $t=t_{n+1}$. When `ntflt`=$\Delta t/\Delta t_{tr}$, the averaged barotropic mode

at $t = t_{n+1}$ is evaluated by calculating the additional $\Delta t/\Delta t_{tr}$ barotropic modes, and then averaging $2 \times \Delta t/\Delta t_{tr} + 1$ barotropic steps. This is the default setting and is applied when `ntflt`=-1 or namelist `ntflt` is not described in the input namelist files. If `ntflt`= 0, time-filtering is not applied. If `ntflt` satisfies $0 <$ `ntflt` $< \Delta t/\Delta t_{tr}$, then the averaged barotropic mode is evaluated by the average of $2 \times$ `ntflt`+1 steps centered at $t = t_{n+1}$. We usually recommend the default setting.

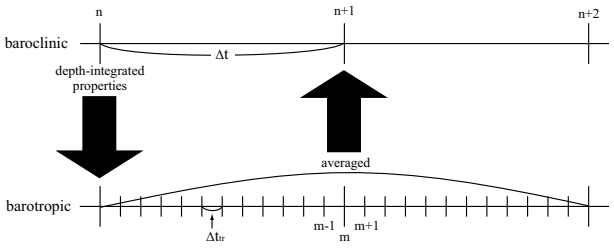

Figure 4.3. Schematic figure of the time integration of the barotropic mode and its time-filtering procedure

Subroutine `SURFCE` diagnoses the volume of the uppermost layer and the mass flux due to the surface elevations using the prognostic variables η, U, and V. The mass flux due to the surface elevations (labeled `ws` in the program codes) is defined at T-points and is obtained by vertically integrating the continuity equation without precipitation and evaporation,

$$\begin{aligned} \mathtt{ws}_{i,j} &= w_{i,j} \cdot \Delta S_{T\,i,j} \\ &= -\Delta t \cdot \left(\Delta y_{i+\frac{1}{2},j} \overline{U}^{\psi}_{i+\frac{1}{2},j} - \Delta y_{i-\frac{1}{2},j} \overline{U}^{\psi}_{i-\frac{1}{2},j} + \Delta x_{i,j+\frac{1}{2}} \overline{V}^{\mu}_{i,j+\frac{1}{2}} - \Delta x_{i,j-\frac{1}{2}} \overline{V}^{\mu}_{i,j-\frac{1}{2}} \right). \end{aligned} \tag{4.18}$$

The relationship between the continuity equation and the vertical velocity at the surface is explained later.

4.3 Prognostics of physical properties in the uppermost layer

4.3.1 Standard scheme

The flux form is used in MRI.COM for calculating the advection and diffusion (viscosity) of momentum, temperature, and salinity. The prognostic variables at the next time step are obtained as the convergence of these fluxes divided by the corresponding volume. In the free-surface formulation, the volume in the uppermost layer changes with time. This requires many careful and special procedures for calculating the prognostic variables in the uppermost layer. In MRI.COM, the prognostic variables themselves and (the prognostic variables) $\times$ (their volume) are saved for the prognostic procedures, which are labeled as `uv1a`, `vv1a`, and `trcv1a` in the program codes. In the prognostic procedures of subroutines `clinic` and `tracer` for momentums and tracers in the uppermost layer, their values times their volume at the new time step are temporarily stored in `uv1a`, `vv1a`, and `trcv1a`. The prognostic value at the new time step is obtained by dividing them by the volume of the new time step, `volt1`, which is calculated at the end of the subroutine `surfce`. In this procedure, the global integrals of the prognostic variables are conserved. However, they are not conserved locally (see the next subsection). The following is the calculation of temperature (T) in the uppermost layer when the leap-frog time is used.

$$\mathtt{trcv1b} = \Delta V_{t\,\mathrm{old}} \cdot T_{\mathrm{old}}$$
$$\downarrow$$

$$\texttt{trcv1a} = \texttt{trcv1b} + \text{Convergence of the fluxes}$$
$$\downarrow$$
$$T_{\text{new}} = \texttt{trcv1a}/\Delta V_{t\,\text{new}}$$
$$\downarrow$$
$$\texttt{trcv1} = \texttt{trcv1a},\ \texttt{trcv1b} = \texttt{trcv1}$$

In MRI.COM, we also use the flux form for body forcing such as wind-forcing and restoration of temperature and salinity to the prescribed values in the uppermost layer.

The body force for the uppermost layer becomes

$$\Big(\frac{\partial u}{\partial t}, \frac{\partial v}{\partial t}\Big)\Big|_{k=\frac{1}{2}} = \ldots + \frac{1}{\rho_0}\frac{(\tau_\mu, \tau_\psi)}{\Delta z_{\frac{1}{2}} + \eta}, \tag{4.19}$$

where (τ_μ, τ_ψ) are the wind stress at the surface (momentum flux), and $\Delta z_{\frac{1}{2}}$ is the standard 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

$$F_z^T = -\frac{1}{\gamma_t}(T - T^*)\Delta z_{\frac{1}{2}}, \qquad \frac{\partial T}{\partial t}\Big|_{k=\frac{1}{2}} = \ldots + \frac{F_z^T}{\Delta z_{\frac{1}{2}} + \eta}, \tag{4.20}$$

$$F_z^S = -\frac{1}{\gamma_s}(S - S^*)\Delta z_{\frac{1}{2}}, \qquad \frac{\partial S}{\partial t}\Big|_{k=\frac{1}{2}} = \ldots + \frac{F_z^S}{\Delta z_{\frac{1}{2}} + \eta}. \tag{4.21}$$

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

4.3.2 Locally conserved scheme (option `FSMOM`)

For the standard free-surface formulations stated above, the relation between the surface elevation, convergence of the barotropic flow, and $P-E+R$ (precipitation minus evaporation) is

$$\frac{\eta^{n+1} - \eta^{n-1}}{2\Delta t} + \nabla \cdot U^n \neq P - E + R. \tag{4.22}$$

Therefore, the mass is not locally conserved even if the r.h.s. $(P-E+R)$ is zero. However, the mass is globally conserved owing to the flux form. Thus, starting from the uniform salinity distribution, we have small deviation from it even if $P-E+R$ is not applied.[§] This inconsistency occurs because η^{n+1} is evaluated as the average of the barotropic steps from t_n to t_{n+2}, while U^n is evaluated as the average of the barotropic steps from t_{n-1} to t_{n+1}.

To remedy this problem, the following procedure can be used in MRI.COM as option `FSMOM`. In this option, $\hat{\eta}^{n+1}$ is calculated in addition to η^{n+1} so that

$$\frac{\hat{\eta}^{n+1} - \eta^{n-1}}{2\Delta t} + \nabla \cdot U^n = P - E + R \tag{4.23}$$

is satisfied. The temperature and salinity in the uppermost layer are obtained based on $\hat{\eta}^{n+1}$, and local conservation is maintained. However, global conservation is not maintained in this case. Thus, we do not usually use this option.

[§]If the times of the barotropic and baroclinic steps are the same, and time-filtering for the free-surface is not applied (`ntflt=0`), then the uniform salinity distribution is maintained. This is not practical for simulations but is useful for checking program codes.

4.4 Introduction of σ-coordinates near the sea surface

This section introduces the σ-coordinates in the upper layers as an extension of the free-surface formulation.[¶] Introducing the σ-coordinates does not require significant modification in the program codes because the barotropic equations are calculated based on the flux form.

In MRI.COM, the σ-coordinates are automatically introduced when option `FREESURFACE` is used. The number of the σ-layer, `ksgm`, must be specified in the model configuration file `configure.in` and should be less than or equal to the total number of vertical grid levels `km`. The bottom topography should not appear for $1 \leq \mathrm{k} \leq$ `ksgm`.

4.4.1 Formulation of σ-layer model

There is an arbitrariness in the definition of the σ-coordinates. We follow the notation of the Princeton Ocean Model (POM; Mellor 2004). Let the surface elevation $z = \eta(\mu, \psi, t)$. A vertical coordinate (σ-coordinate) that is normalized by the thickness between $z = -H(< 0)$ and $z = \eta(\mu, \psi, t)$ is defined as

$$\sigma = \frac{z - \eta}{H + \eta} \qquad (-1 < \sigma < 0). \tag{4.24}$$

See Figure 4.1. Thus, the values of σ range from $\sigma = 0$ at $z = \eta$ to $\sigma = -1$ at $z = -H$. Hereinafter, we define the thickness of the σ-layer as D, i.e., $D \equiv H + \eta$. The quantity $Dd\sigma = (\eta + H)d\sigma$ corresponds to the real vertical grid size `dzu` in the program code of MRI.COM. (Note that `dzu` varies with time.) For the general σ-coordinates, the depth (H) is also a function of horizontal position, but here H is assumed to be constant.

Next we derive the governing equations in the σ-coordinates. The equations expressed in terms of the original Cartesian coordinates $z(\mu^*, \psi^*, z^*, t^*)$ are transformed into sigma coordinates $\sigma(\mu, \psi, \sigma, t)$, where

$$\mu = \mu^*,\ \psi = \psi^*,\ \sigma = \frac{z^* - \eta}{H + \eta},\ t = t^*. \tag{4.25}$$

The partial derivatives transform according to

$$\frac{\partial \alpha^*}{\partial \mu^*} = \frac{\partial \alpha}{\partial \mu} - \frac{1+\sigma}{D}\frac{\partial \eta}{\partial \mu}\frac{\partial \alpha}{\partial \sigma}, \tag{4.26}$$

$$\frac{\partial \alpha^*}{\partial \psi^*} = \frac{\partial \alpha}{\partial \psi} - \frac{1+\sigma}{D}\frac{\partial \eta}{\partial \psi}\frac{\partial \alpha}{\partial \sigma}, \tag{4.27}$$

$$\frac{\partial \alpha^*}{\partial z^*} = \frac{1}{D}\frac{\partial \alpha}{\partial \sigma}, \tag{4.28}$$

$$\frac{\partial \alpha^*}{\partial t^*} = \frac{\partial \alpha}{\partial t} - \frac{1+\sigma}{D}\frac{\partial \alpha}{\partial \sigma}\frac{\partial \eta}{\partial t}. \tag{4.29}$$

The material derivatives transform according to

$$\begin{aligned} \frac{d\alpha^*}{dt^*} &= \frac{\partial \alpha^*}{\partial t^*} + \frac{u^*}{h_\mu}\frac{\partial \alpha^*}{\partial \mu^*} + \frac{v^*}{h_\psi}\frac{\partial \alpha^*}{\partial \psi^*} + w^*\frac{\partial \alpha^*}{\partial z^*} \\ &= \frac{\partial \alpha}{\partial t} + \frac{u}{h_\mu}\frac{\partial \alpha}{\partial \mu} + \frac{v}{h_\psi}\frac{\partial \alpha}{\partial \psi} + \frac{w^*}{D}\frac{\partial \alpha}{\partial \sigma} - \frac{1+\sigma}{D}\frac{\partial \alpha}{\partial \sigma}\left(\frac{u}{h_\mu}\frac{\partial \eta}{\partial \mu} + \frac{v}{h_\psi}\frac{\partial \eta}{\partial \psi} + \frac{\partial \eta}{\partial t}\right). \end{aligned} \tag{4.30}$$

Here, we define ω as

$$\omega \equiv \frac{d\sigma}{dt^*} = \frac{w^*}{D} - \frac{1+\sigma}{D}\left(\frac{u}{h_\mu}\frac{\partial \eta}{\partial \mu} + \frac{v}{h_\psi}\frac{\partial \eta}{\partial \psi} + \frac{\partial \eta}{\partial t}\right). \tag{4.31}$$

[¶]The introduction of the σ-layer, and the expression of equations are based on Hasumi (2006).

The above equation then becomes

$$\frac{d\alpha^*}{dt^*} = \frac{\partial\alpha}{\partial t} + \frac{u}{h_\mu}\frac{\partial\alpha}{\partial\mu} + \frac{v}{h_\psi}\frac{\partial\alpha}{\partial\psi} + \omega\frac{\partial\alpha}{\partial\sigma} \equiv \frac{d\alpha}{dt}. \tag{4.32}$$

Thus, they take nearly the same form as in z-coordinates.

In the program code, we do not directly calculate ω, but diagnose $D\omega$ and $Dd\sigma$ from the equation

$$\omega\frac{\partial\alpha}{\partial\sigma} = D\omega\frac{\partial\alpha}{D\partial\sigma}. \tag{4.33}$$

They are stored in the model code as variables `wlwl` and `dzu`. (Their dimensions are the same as those in the z-coordinates, and their magnitudes are comparable.)

4.4.2 Governing equations in the σ-coordinates

a. Continuity equations

The continuity equation in the σ-layer is

$$\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi u)}{\partial\mu} + \frac{\partial(h_\mu v)}{\partial\psi}\right\} + \frac{\partial\omega}{\partial\sigma} + \frac{1}{D}\left(\frac{u}{h_\mu}\frac{\partial\eta}{\partial\mu} + \frac{v}{h_\psi}\frac{\partial\eta}{\partial\psi} + \frac{\partial\eta}{\partial t}\right) = 0. \tag{4.34}$$

In the flux form, this becomes simpler.

$$\frac{\partial\eta}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi uD)}{\partial\mu} + \frac{\partial(h_\mu vD)}{\partial\psi}\right\} + D\frac{\partial\omega}{\partial\sigma} = 0. \tag{4.35}$$

In the program code, this equation is used to diagnose $D\omega$, which is used as vertical velocity w (`wlwl`) in the σ-layer.

b. Advection terms

Advection terms in the flux form $\mathscr{A}(\alpha D)$ are as follows:

$$\mathscr{A}(\alpha D) = \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi u\alpha D)}{\partial\mu} + \frac{\partial(h_\mu v\alpha D)}{\partial\psi}\right\} + D\frac{\partial(\omega\alpha)}{\partial\sigma}. \tag{4.36}$$

In the program code, the convergence of the advective flux per unit area is calculated as follows:

$$\begin{aligned}\mathscr{A}(\alpha Dd\sigma) = {} & \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi u\alpha Dd\sigma)}{\partial\mu} + \frac{\partial(h_\mu v\alpha Dd\sigma)}{\partial\psi}\right\} \\ & + (\alpha D\omega)_{\text{upper}} - (\alpha D\omega)_{\text{lower}}.\end{aligned} \tag{4.37}$$

c. Pressure gradient term

The hydrostatic equation is

$$\frac{1}{D}\frac{\partial p}{\partial\sigma} = -\rho g. \tag{4.38}$$

When the atmospheric pressure at the surface is neglected, the pressure in the σ-coordinates is

$$p(\sigma) = g\int_\sigma^0 D\rho d\sigma' = gD\int_\sigma^0 \rho d\sigma'. \tag{4.39}$$

After some manipulations, the following expression for the pressure gradient in the σ-coordinates is obtained:

$$\begin{aligned}\frac{\partial p^*}{\partial \mu^*} &= \frac{\partial p}{\partial \mu} - \frac{1+\sigma}{D}\frac{\partial \eta}{\partial \mu}\frac{\partial p}{\partial \sigma} = gD\int_\sigma^0 \Big(\frac{\partial \rho}{\partial \mu} - \frac{\sigma'}{D}\frac{\partial \eta}{\partial \mu}\frac{\partial \rho}{\partial \sigma'}\Big)d\sigma' + \rho g\frac{\partial \eta}{\partial \mu},\\ \frac{\partial p^*}{\partial \psi^*} &= \frac{\partial p}{\partial \psi} - \frac{1+\sigma}{D}\frac{\partial \eta}{\partial \psi}\frac{\partial p}{\partial \sigma} = gD\int_\sigma^0 \Big(\frac{\partial \rho}{\partial \psi} - \frac{\sigma'}{D}\frac{\partial \eta}{\partial \psi}\frac{\partial \rho}{\partial \sigma'}\Big)d\sigma' + \rho g\frac{\partial \eta}{\partial \psi}.\end{aligned} \tag{4.40}$$

The first term on the r.h.s. corresponds to the baroclinic part of the pressure gradient. The second term on the r.h.s. is the barotropic part of the pressure gradient. Density ρ in the last term on the r.h.s. is replaced by the reference density ρ_0.

d. Time integration of velocities and tracers

Horizontal viscosity and diffusion are assumed to be parallel to the σ-layer in the σ-layer model. In MRI.COM, the volume integral of a tracer with variable name `trcl`, `trcv1` ($\equiv$ `trcl` $*$ `tvol1`), is stored to calculate `trcl` in the next step. Introduction of the σ-layer corresponds to the direct extension of `trcv1` in the uppermost layer to all σ-layers. The following equation expresses the tracer equations per unit area:

$$\frac{\partial}{\partial t}[TD] = -\frac{1}{h_\mu h_\psi}\left\{\frac{\partial (h_\psi uTD)}{\partial \mu} + \frac{\partial (h_\mu vTD)}{\partial \psi}\right\} - \frac{\partial}{\partial \sigma}[\omega TD] + D\mathscr{D}(T). \tag{4.41}$$

4.4.3 Redistribution of tracers among the σ-layers

The volume change due to freshwater flux from the atmosphere is distributed among all the σ-layers, so that the ratio of the thicknesses for each σ-layer remains unchanged. However, in reality freshwater flux like rain should affect only the uppermost layer. In addition, the uppermost layer gets excess heat flux because the heat flux due to this freshwater flux is only applied to the uppermost layer. In order to fix this problem, tracers are redistributed among the σ-layers at the beginning of the subroutine `tracer`.

See the schematic diagram in Figure 4.4. Let us consider the k-th layer in the σ-layer ($1 \le k \le k_{\mathrm{sigma}}$). Let surface height, temperature, and salinity before this redistribution be $\eta^{n+1}(\mathrm{old})$, $T^{n+1}(\mathrm{old})$, and $S^{n+1}(\mathrm{old})$. The surface height due to the freshwater flux is

$$\eta^{n+1} = \eta^{n+1}(\mathrm{old}) + \delta\eta, \tag{4.42}$$

where $\delta\eta = 2\Delta t(P-E+R)$ during the leap-frog time integration.

The following thicknesses are redistributed at the lower boundary and the upper boundary of the k-th layer.

$$\delta z_k^T = \left(1 - \sum_{l=1}^{k-1}\Delta\sigma_l\right)\delta\eta \quad (1 < k \le k_{\mathrm{sigma}}), \quad \text{and} \quad \delta z_1^T = 0, \tag{4.43}$$

$$\delta z_k^B = \left(1 - \sum_{l=1}^{k}\Delta\sigma_l\right)\delta\eta \quad (1 \le k < k_{\mathrm{sigma}}), \quad \text{and} \quad \delta z_{k_{\mathrm{sigma}}}^B = 0, \tag{4.44}$$

where z_k^B, and z_k^T are the redistributed thickness at the lower end and the upper end of the k-th layer, respectively ($\delta z_k^T = \delta z_{k-1}^B$).

The redistribution of tracer (S) among the σ-coordinate layer depends on the sign of $\delta\eta$:

When $\delta\eta > 0$ (Figure 4.4a),

$$S_k^{n+1}(\eta^{n+1}+H)\Delta\sigma_k = S_k^{n+1}(\mathrm{old})(\eta^{n+1}(\mathrm{old})+H)\Delta\sigma_k - S_k^{n+1}(\mathrm{old})\delta z_k^B + S_{k-1}^{n+1}(\mathrm{old})\delta z_k^T. \tag{4.45}$$

When $\delta\eta < 0$ (Figure 4.4b),

$$S_k^{n+1}(\eta^{n+1}+H)\Delta\sigma_k = S_k^{n+1}(\text{old})(\eta^{n+1}(\text{old})+H)\Delta\sigma_k - S_{k+1}^{n+1}(\text{old})\delta z_k^B + S_k^{n+1}(\text{old})\delta z_k^T. \tag{4.46}$$

In the program, `trcv1b` is substituted with `trcv1a` at the beginning of the subroutine `tracer`, then `trcv1a` is modified using either (4.45) or (4.46).

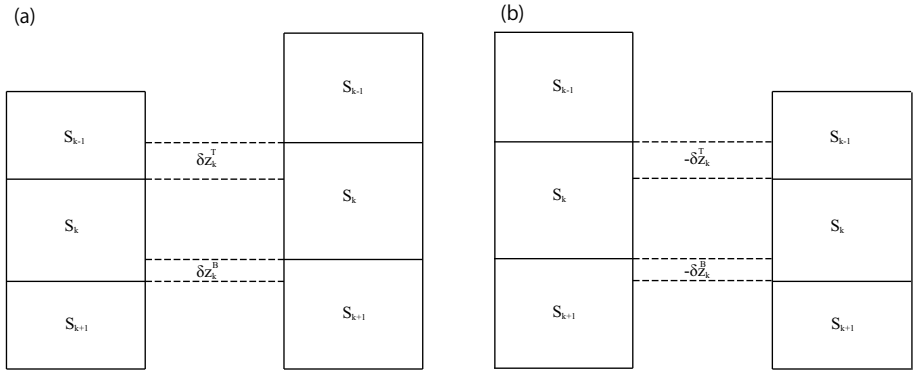

Figure 4.4. Schematic diagram of the redistribution of tracer among the σ-coordinate layers. When fresh water is (a) added, and (b) removed from the sea surface.

References

Hasumi, H., 2006: CCSR Ocean Component Model (COCO) version 4.0, *CCSR report*, *25*, 103pp.

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.

Meisinger, F., and A. Arakawa 1976: Numerical methods used in atmospheric models, *GARP Publications Series*, *17*, 65pp.

Mellor, G. L., 2004: Users guide for a three-dimensional, primitive equation, numerical ocean model, *Prog. in Atmos. and Ocean. Sci, Princeton University*, 53pp.,
available online at http://www.aos.princeton.edu/WWWPUBLIC/htdocs.pom/PubOnLine/POL.html.

Chapter 5 Equations of motion (baroclinic component)

This chapter explains the advection terms (Section 5.1) and the viscosity terms (Section 5.2) of momentum.

One of the unique characteristics of MRI.COM's momentum advection terms is that there are exchanges of momentum between U-cells that share only a corner without a common side boundary. 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 5.1 is based on Ishizaki and Motoi (1999).

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

5.1 Advection terms

Chapter 3 demonstrated that the mass fluxes used for calculating momentum advection are those for the mass continuity of the U-cell and are obtained by an averaging operation (3.10) of those for the T-cell mass continuity (3.3) to (3.9). 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 to be 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 3.2(a)).

5.1.1 Vertical mass fluxes and its momentum advection

According to the definition (3.3) and (3.10) in Chapter 3, the vertical mass flux at the upper surface, level k, of the U-cell $(i+\frac{1}{2}.j+\frac{1}{2},k+\frac{1}{2})$, $\overline{W}^U_{i+\frac{1}{2},j+\frac{1}{2},k}$, is defined by surrounding W^T as

$$\overline{W}^U_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{W^T_{i,j,k}}{N_{i,j,k+\frac{1}{2}}} + \frac{W^T_{i+1,j,k}}{N_{i+1,j,k+\frac{1}{2}}} + \frac{W^T_{i,j+1,k}}{N_{i,j+1,k+\frac{1}{2}}} + \frac{W^T_{i+1,j+1,k}}{N_{i+1,j+1,k+\frac{1}{2}}}, \tag{5.1}$$

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

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

$$\mathscr{W}^U_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{W^T_{i,j,k}}{N_{i,j,k-\frac{1}{2}}} + \frac{W^T_{i+1,j,k}}{N_{i+1,j,k-\frac{1}{2}}} + \frac{W^T_{i,j+1,k}}{N_{i,j+1,k-\frac{1}{2}}} + \frac{W^T_{i+1,j+1,k}}{N_{i+1,j+1,k-\frac{1}{2}}}. \tag{5.2}$$

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

a. One-dimensional variation of bottom relief

We first consider a case in which the bottom depth varies in one direction like a staircase (Figure 5.1(a)). Assume a barotropic current flows over the topography. The U-points are just intermediate between T-points. This indicates the mass continuity for T-cells. The barotropic flow comes from the left and barotropy is conserved in shallow regions. Figure 5.1(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.

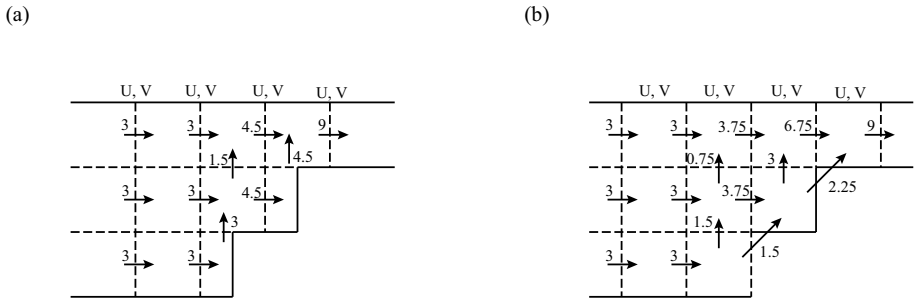

Figure 5.1. (a) Two-dimensional mass fluxes for T-cells on a stair-like topography. (b) Two-dimensional mass fluxes for U-cells on the same topography.

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, 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 5.2). We use suffixes l and u to designate the lower and the upper layer. The central T-point and T-cell are represented by A. The vertical mass flux W^T should be continuous at the interface between cells A_l and A_u, though the area of cell A_l (3/4 measured in grid area units) differs from that of A_u (1 unit). Let us

consider how this T-cell mass flux W^T should be distributed to the mass flux W^U of neighboring cells represented by **a**, **b**, **c**, and **d**. In the lower layer, W_T is shared by three cells, $\mathbf{a}_l$, $\mathbf{b}_l$, and $\mathbf{c}_l$, so the contribution of W^T to each W^U is $W^T/3$, but in the upper layer, it is $W^T/4$ because part of W_T should also be shared by W^U at cell $\mathbf{d}_u$. Here W^U at the bottom of cell $\mathbf{d}_u$ is no longer zero. Therefore, $W^T/4$ of the $W^T/3$ shared by each of the three lower sea grid cells $\mathbf{a}_l$, $\mathbf{b}_l$, and $\mathbf{c}_l$ is purely vertical, and the remaining $W^T/12$ ($= W^T/3 - W^T/4$) flows to cell $\mathbf{d}_u$ through the interface. Gathering these diagonal fluxes from the lower three cells, the total amount entering cell $\mathbf{d}_u$ is certainly $W^T/4$ (=$W^T/12 \times 3$). The advected momentum value should be the mean of those at the starting and ending cells of the flux, if the centered difference scheme is used, which is necessary to conserve the total kinetic energy.

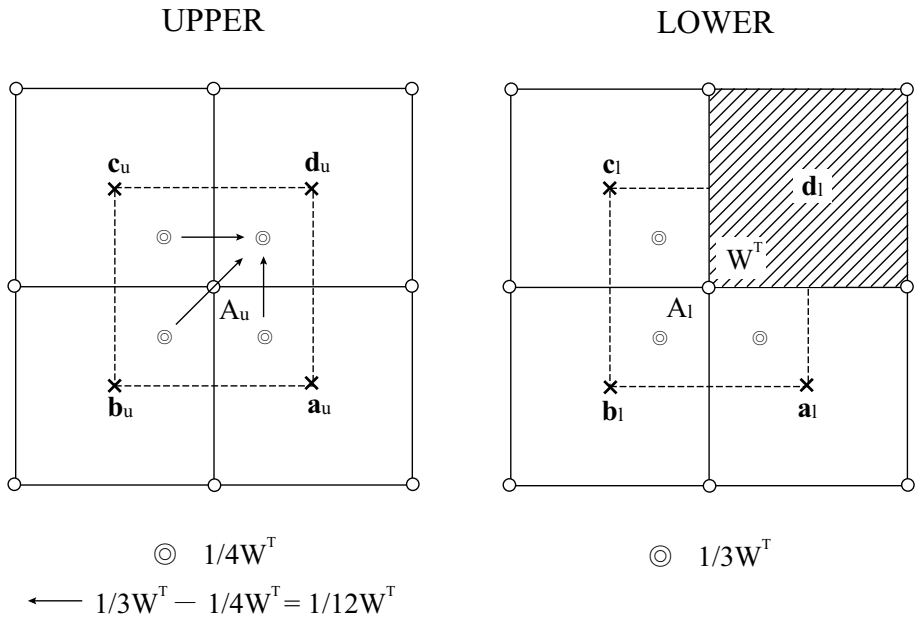

Figure 5.2. First example of land-sea patterns, in which all four upper cells are sea cells, with three sea cells and one land cell in the lower layer.

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 5.3). 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 5.4). 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 $W^T/3$ ($= W^T/6 \times 2$).

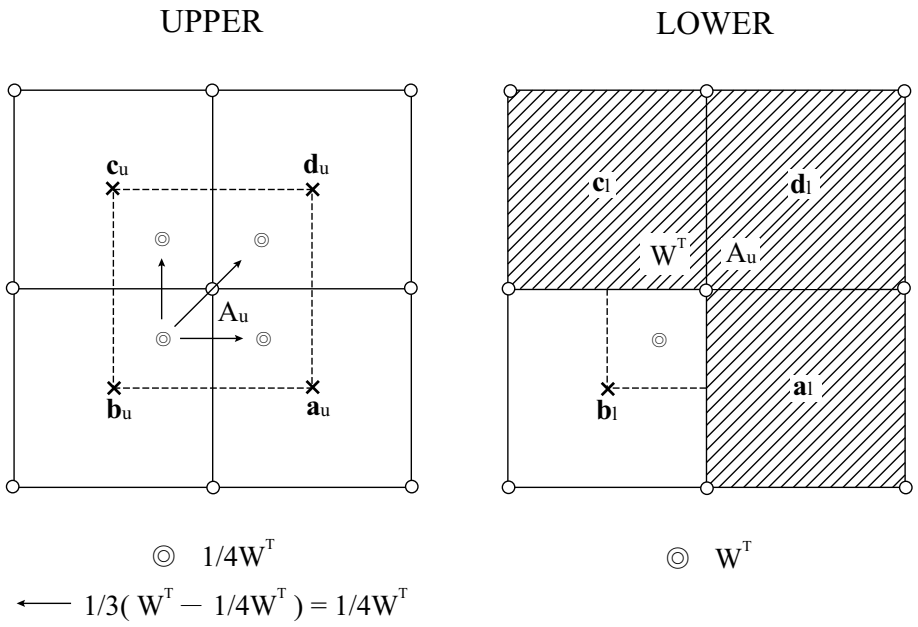

Figure 5.3. Second example of land-sea patterns, in which all four upper cells are sea cells, with one sea cell and three land cells in the lower layer.

c. Generalization

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

$$W^T/N_l - W^T/N_u = W^T(N_u - N_l)/(N_l N_u) \tag{5.3}$$

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

$$W^T(N_u - N_l)/(N_l N_u) \times 1/(N_u - N_l) = W^T/(N_l N_u). \tag{5.4}$$

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

$$W^T/(N_l N_u) \times N_l = W^T/N_u. \tag{5.5}$$

Based on these discussions we understand the difference between (5.1) and (5.2).

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

$$\mathbf{a}_l W^T/(N_l N_u) \quad \text{and} \quad \mathbf{a}_l W^T(u_{\mathrm{a}_l} + u_{\mathrm{d}_u})/(2N_l N_u), \tag{5.6}$$

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

$$\mathbf{a}_l W^T/N_u \quad \text{and} \quad \mathbf{a}_l W^T(u_{\mathrm{a}_l} + u_{\mathrm{a}_u})/(2N_u), \tag{5.7}$$

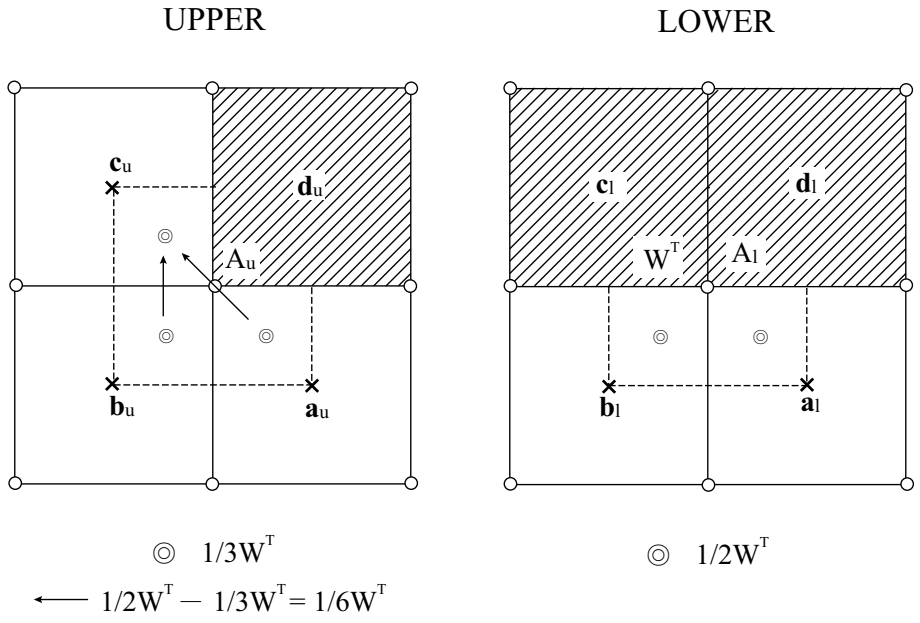

Figure 5.4. Third example of land-sea patterns, in which one of the upper cells is a land cell, with two land and two sea cells in the lower layer.

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 (5.7) apply.

5.1.2 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. To do this, we start with the generalization of the T-cell mass continuity (3.3)-(3.9). Assuming that $e_{i+\frac{1}{2},j+\frac{1}{2}}$ is a land-sea index (unity for sea and zero for land) for U-cell $(i+\frac{1}{2}, j+\frac{1}{2})$, the general formulae for $U^T_{i+\frac{1}{2},j}$ and $V^T_{i,j+\frac{1}{2}}$ are given as

$$U^T_{i+\frac{1}{2},j} = \frac{1}{2}(e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}})u^*_{i+\frac{1}{2},j}\Delta y_{i+\frac{1}{2},j}\Delta z$$

and

$$V^T_{i,j+\frac{1}{2}} = \frac{1}{2}(e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}})v^*_{i,j+\frac{1}{2}}\Delta x_{i,j+\frac{1}{2}}\Delta z \tag{5.8}$$

Here we neglect the vertical subscript $(k+\frac{1}{2})$.

Substituting these formulae into the T-cell mass continuity (3.3), the X (zonal) component of the mass continuity

for U-cell $(i+\frac{1}{2}, j+\frac{1}{2})$ (3.10), $\mathrm{XMC}^{U}_{i+\frac{1}{2},j+\frac{1}{2}}$, multiplied by its own land-sea signature $e_{i+\frac{1}{2},j+\frac{1}{2}}$, is expressed as

$$
\begin{aligned}
\mathrm{XMC}^{U}_{i+\frac{1}{2},j+\frac{1}{2}} &= e_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \Delta z \\
&\times \Big\{ \frac{1}{N_{i,j}} [(e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}}) u^{*}_{i-\frac{1}{2},j} - (e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u^{*}_{i+\frac{1}{2},j}] \\
&+ \frac{1}{N_{i+1,j}} [(e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u^{*}_{i+\frac{1}{2},j} - (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}}) u^{*}_{i+\frac{3}{2},j}] \\
&+ \frac{1}{N_{i,j+1}} [(e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}}) u^{*}_{i-\frac{1}{2},j+1} - (e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) u^{*}_{i+\frac{1}{2},j+1}] \\
&+ \frac{1}{N_{i+1,j+1}} [(e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) u^{*}_{i+\frac{1}{2},j+1} - (e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}}) u^{*}_{i+\frac{3}{2},j+1}] \Big\} \\
&= e_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \Delta z \times \Big\{ \Big[\frac{1}{N_{i,j}} (e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}}) u^{*}_{i-\frac{1}{2},j} \\
&+ \Big(-\frac{1}{N_{i,j}} + \frac{1}{N_{i+1,j}} \Big) (e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) u^{*}_{i+\frac{1}{2},j} \\
&- \frac{1}{N_{i+1,j}} (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}}) u^{*}_{i+\frac{3}{2},j} \Big] \\
&+ \Big[\frac{1}{N_{i,j+1}} (e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}}) u^{*}_{i-\frac{1}{2},j+1} \\
&+ \Big(-\frac{1}{N_{i,j+1}} + \frac{1}{N_{i+1,j+1}} \Big) (e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) u^{*}_{i+\frac{1}{2},j+1} \\
&- \frac{1}{N_{i+1,j+1}} (e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}}) u^{*}_{i+\frac{3}{2},j+1} \Big] \Big\}. \qquad (5.9)
\end{aligned}
$$

Here, recalling

$$
N_{i,j} = e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}, \qquad (5.10)
$$

we have,

$$
\begin{aligned}
&\Big(-\frac{1}{N_{i,j}} + \frac{1}{N_{i+1,j}} \Big) (e_{i+\frac{1}{2},j-\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{1}{2}}) \\
&\qquad = \frac{1}{N_{i,j}} (e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}}) - \frac{1}{N_{i+1,j}} (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}})
\end{aligned} \qquad (5.11)
$$

and

$$
\begin{aligned}
&\Big(-\frac{1}{N_{i,j+1}} + \frac{1}{N_{i+1,j+1}} \Big) (e_{i+\frac{1}{2},j+\frac{1}{2}} + e_{i+\frac{1}{2},j+\frac{3}{2}}) \\
&\qquad = \frac{1}{N_{i,j+1}} (e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}}) - \frac{1}{N_{i+1,j+1}} (e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}}).
\end{aligned} \qquad (5.12)
$$

Thus, based on (3.13) and (3.14),

$$\begin{aligned}
\mathrm{XMC}^U_{i+\frac{1}{2},j+\frac{1}{2}} &= e_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \Delta z \Big\{ \Big[\frac{1}{N_{i,j}} (e_{i-\frac{1}{2},j-\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{1}{2}})(u^*_{i-\frac{1}{2},j} + u^*_{i+\frac{1}{2},j}) \\
&\quad - \frac{1}{N_{i+1,j}} (e_{i+\frac{3}{2},j-\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{1}{2}})(u^*_{i+\frac{1}{2},j} + u^*_{i+\frac{3}{2},j}) \Big] \\
&\quad + \Big[\frac{1}{N_{i,j+1}} (e_{i-\frac{1}{2},j+\frac{1}{2}} + e_{i-\frac{1}{2},j+\frac{3}{2}})(u^*_{i-\frac{1}{2},j+1} + u^*_{i+\frac{1}{2},j+1}) \\
&\quad - \frac{1}{N_{i+1,j+1}} (e_{i+\frac{3}{2},j+\frac{1}{2}} + e_{i+\frac{3}{2},j+\frac{3}{2}})(u^*_{i+\frac{1}{2},j+1} + u^*_{i+\frac{3}{2},j+1}) \Big] \Big\} \\
&= e_{i+\frac{1}{2},j+\frac{1}{2}} \Big[e_{i-\frac{1}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i,j}} U^U_{i,j} + \frac{1}{N_{i,j+1}} U^U_{i,j+1} \Big) \\
&\quad - e_{i+\frac{3}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i+1,j}} U^U_{i+1,j} + \frac{1}{N_{i+1,j+1}} U^U_{i+1,j+1} \Big) + e_{i-\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} U^U_{i,j} \\
&\quad - e_{i+\frac{3}{2},j+\frac{3}{2}} \frac{1}{N_{i+1,j+1}} U^U_{i+1,j+1} + e_{i-\frac{1}{2},j+\frac{3}{2}} \frac{1}{N_{i,j+1}} U^U_{i,j+1} - e_{i+\frac{3}{2},j-\frac{1}{2}} \frac{1}{N_{i+1,j}} U^U_{i+1,j} \Big].
\end{aligned} \tag{5.13}$$

Adding the Y (meridional) component to the above formula, we obtain the horizontal part of the U-cell mass continuity as follows:

$$\begin{aligned}
\mathrm{HMC}^U_{i+\frac{1}{2},j+\frac{1}{2}} &= e_{i+\frac{1}{2},j+\frac{1}{2}} \\
&\quad \times \Big[e_{i-\frac{1}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i,j}} U^U_{i,j} + \frac{1}{N_{i,j+1}} U^U_{i,j+1} \Big) - e_{i+\frac{3}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i+1,j}} U^U_{i+1,j} + \frac{1}{N_{i+1,j+1}} U^U_{i+1,j+1} \Big) \\
&\quad + e_{i+\frac{1}{2},j-\frac{1}{2}} \Big(\frac{1}{N_{i,j}} V^U_{i,j} + \frac{1}{N_{i+1,j}} V^U_{i+1,j} \Big) - e_{i+\frac{1}{2},j+\frac{3}{2}} \Big(\frac{1}{N_{i,j+1}} V^U_{i,j+1} + \frac{1}{N_{i+1,j+1}} V^U_{i+1,j+1} \Big) \\
&\quad + e_{i-\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} (U^U_{i,j} + V^U_{i,j}) - e_{i+\frac{3}{2},j+\frac{3}{2}} \frac{1}{N_{i+1,j+1}} (U^U_{i+1,j+1} + V^U_{i+1,j+1}) \\
&\quad + e_{i-\frac{1}{2},j+\frac{3}{2}} \frac{1}{N_{i,j+1}} (U^U_{i,j+1} - V^U_{i,j+1}) - e_{i+\frac{3}{2},j-\frac{1}{2}} \frac{1}{N_{i+1,j}} (U^U_{i+1,j} - V^U_{i+1,j}) \Big].
\end{aligned} \tag{5.14}$$

Assuming mass fluxes M_E, M_N, M_{NE}, M_{SE} as follows:

$$\begin{aligned}
M_{\mathbf{E}_{i,j+\frac{1}{2}}} &= e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j+\frac{1}{2}} \Big(\frac{1}{N_{i,j}} U^U_{i,j} + \frac{1}{N_{i,j+1}} U^U_{i,j+1} \Big), \\
M_{\mathbf{N}_{i+\frac{1}{2},j}} &= e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i+\frac{1}{2},j-\frac{1}{2}} \Big(\frac{1}{N_{i,j}} V^U_{i,j} + \frac{1}{N_{i+1,j}} V^U_{i+1,j} \Big), \\
M_{\mathbf{NE}_{i,j}} &= e_{i+\frac{1}{2},j+\frac{1}{2}} e_{i-\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} (U^U_{i,j} + V^U_{i,j}), \\
M_{\mathbf{SE}_{i,j}} &= e_{i-\frac{1}{2},j+\frac{1}{2}} e_{i+\frac{1}{2},j-\frac{1}{2}} \frac{1}{N_{i,j}} (U^U_{i,j} - V^U_{i,j}),
\end{aligned} \tag{5.15}$$

then,

$$\begin{aligned}
\mathrm{HMC}^U_{i+\frac{1}{2},j+\frac{1}{2}} &= M_{\mathbf{E}_{i,j+\frac{1}{2}}} - M_{\mathbf{E}_{i+1,j+\frac{1}{2}}} + M_{\mathbf{N}_{i+\frac{1}{2},j}} - M_{\mathbf{N}_{i+\frac{1}{2},j+1}} \\
&\quad + M_{\mathbf{NE}_{i,j}} - M_{\mathbf{NE}_{i+1,j+1}} + M_{\mathbf{SE}_{i,j+1}} - M_{\mathbf{SE}_{i+1,j}}.
\end{aligned} \tag{5.16}$$

Here, M_E and M_N are axis-parallel mass fluxes, and M_{NE} and M_{SE} are horizontally diagonal ones (Figure 5.5).

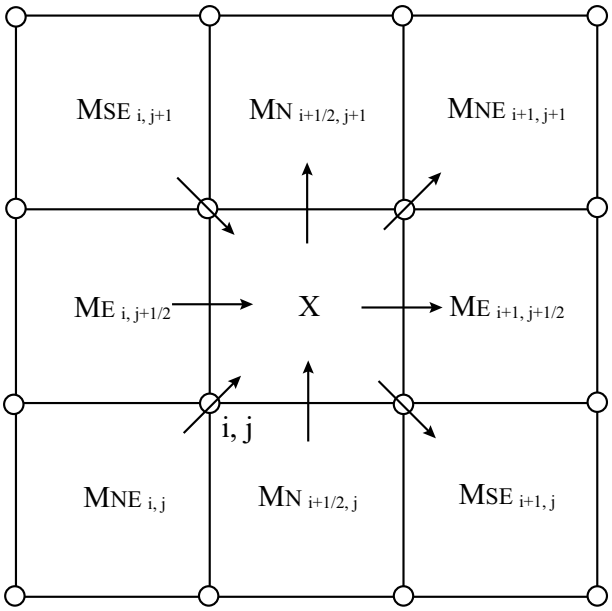

Figure 5.5. Distribution of generalized mass fluxes for U-cell $(i+1/2, j+1/2)$

If we derive the formula for the standard case from (5.14) (all of N are 4),

$$\begin{aligned}\mathrm{HMC}^U_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}\Big[&\frac{1}{2}(U^U_{i,j}+U^U_{i,j+1})-\frac{1}{2}(U^U_{i+1,j}+U^U_{i+1,j+1})\\ &+\frac{1}{2}(V^U_{i,j}+V^U_{i,j+1})-\frac{1}{2}(V^U_{i+1,j}+V^U_{i+1,j+1})\Big]\\ +\frac{1}{2}\Big[&\frac{1}{2}(U^U_{i,j}+V^U_{i,j})-\frac{1}{2}(U^U_{i+1,j+1}+V^U_{i+1,j+1})\\ &+\frac{1}{2}(U^U_{i,j+1}-V^U_{i,j+1})-\frac{1}{2}(U^U_{i+1,j}-V^U_{i+1,j})\Big].\end{aligned} \tag{5.17}$$

This expression means that the horizontal mass flux convergence is a mean of those of the axis-parallel mass fluxes (3.12) and of the diagonal ones (3.17). 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 $(\alpha : \beta)$ to average the convergences of the axis-parallel and the horizontally diagonal mass fluxes, as long as $\alpha+\beta = 1$, as seen in (5.17). 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 is expressed as follows:

$$\begin{aligned}
\mathrm{CAD}_{i+\frac{1}{2},j+\frac{1}{2}}(u) = \frac{2}{3}\Big[&\frac{1}{2}(u_{i-\frac{1}{2},j+\frac{1}{2}}+u_{i+\frac{1}{2},j+\frac{1}{2}})\frac{1}{2}(U^U_{i,j}+U^U_{i,j+1})\\
&-\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}}+u_{i+\frac{3}{2},j+\frac{1}{2}})\frac{1}{2}(U^U_{i+1,j}+U^U_{i+1,j+1})\\
&+\frac{1}{2}(u_{i+\frac{1}{2},j-\frac{1}{2}}+u_{i+\frac{1}{2},j+\frac{1}{2}})\frac{1}{2}(V^U_{i,j}+V^U_{i+1,j})\\
&-\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}}+u_{i+\frac{1}{2},j+\frac{3}{2}})\frac{1}{2}(V^U_{i,j+1}+V^U_{i+1,j+1})\Big]\\
+\frac{1}{3}\Big[&\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}}+u_{i-\frac{1}{2},j-\frac{1}{2}})\frac{1}{2}(U^U_{i,j}+V^U_{i,j})\\
&-\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}}+u_{i+\frac{3}{2},j+\frac{3}{2}})\frac{1}{2}(U^U_{i+1,j+1}+V^U_{i+1,j+1})\\
&+\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}}+u_{i-\frac{1}{2},j+\frac{3}{2}})\frac{1}{2}(U^U_{i,j+1}-V^U_{i,j+1})\\
&-\frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}}+u_{i+\frac{3}{2},j-\frac{1}{2}})\frac{1}{2}(U^U_{i+1,j}-V^U_{i+1,j})\Big]
\end{aligned} \tag{5.18}$$

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 5.5 and related momentum flux, let us examine the axis-parallel and horizontally diagonal mass flux associated with $U^U_{i,j}$. Look at Figure 5.2b, where letters **a**, **b**, **c**, and **d** designate the land-sea index and names of U-cells. Cell **d** is assumed to be a sea cell (**d** = 1). We analyze two kinds of mass fluxes associated with $U^U_{i,j}$ under different combinations of **a**, **b**, and **c** (eight cases), as indicated in the first column in Table 5.1. Column (A) corresponds to an index, which is unity for its own combination and zero for all other combinations. Column (B) lists the coefficient of $U^U_{i,j}$ in the axis-parallel mass flux of the U-cell mass continuity (5.14). Column (C) indicates the coefficient of $U^U_{i,j}$ in the horizontally diagonal mass flux of (5.14).

The generalized coefficient of $U^U_{i,j}$ in the axis-parallel mass flux (c_1) is obtained by summing the product of **A** and **B** over the eight cases. Similarly, the generalized coefficient in the horizontally diagonal mass flux (c_2) is obtained by the summing the product of **A** and **C**. That is,

$$\begin{aligned}
c_1 &= \sum_{n=1}^{8}\mathbf{A}_n\mathbf{B}_n = \frac{1}{6}\mathbf{c}(\mathbf{ab}-\mathbf{a}-\mathbf{b}+3)\\
&\text{and}\\
c_2 &= \sum_{n=1}^{8}\mathbf{A}_n\mathbf{C}_n = \frac{1}{6}\mathbf{b}(3-\mathbf{a}-\mathbf{c}).
\end{aligned} \tag{5.19}$$

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

$$\begin{aligned}
\frac{\mathbf{d}}{2}(u_{\mathbf{c}}+u_{\mathbf{d}})c_1U^U_{i,j} &= \frac{1}{2}(u_{\mathbf{c}}+u_{\mathbf{d}})\frac{1}{6}\mathbf{cd}(\mathbf{ab}-\mathbf{a}-\mathbf{b}+3)U^U_{i,j},\\
&\text{and}\\
\frac{\mathbf{d}}{2}(u_{\mathbf{b}}+u_{\mathbf{d}})c_2U^U_{i,j} &= \frac{1}{2}(u_{\mathbf{b}}+u_{\mathbf{d}})\frac{1}{6}\mathbf{bd}(3-\mathbf{a}-\mathbf{c})U^U_{i,j},
\end{aligned} \tag{5.20}$$

respectively.

Table 5.1. Definition of a land-sea index, the index identifying each case (column **A**), the coefficient of $U^U_{i,j}$ in the axis-parallel mass flux (column **B**), and the coefficient of $U^U_{i,j}$ in the horizontally diagonal mass flux (column **C**), for eight combinations of indices **a**, **b**, and **c** in Figure 5.2b. Cell **d** is assumed to be a sea cell, and the momentum advection by means of $U^U_{i,j}$ into and from cell **d** is generalized.

CASE n	Land-sea index **a** **b** **c**	**A**	**B** Coefficient of $U^U_{i,j}$ (axisi-parallel) +	**C** Coefficient of $U^U_{i,j}$ (horizontally-diagonal) ×
1	1 1 1	**abc**	1/3	1/6
2	1 1 0	**ab**(1−**c**)	0	1/3
3	1 0 1	**ab**(1−**b**)**c**	1/3	0
4	0 1 1	(1−**a**)**bc**	1/3	1/3
5	1 0 0	–	0	0
6	0 1 0	(1−**a**)**b**(1−**c**)	0	1/2
7	0 0 1	(1−**a**)(1−**b**)**c**	1/2	0
8	0 0 0	–	0	0

The resultant momentum fluxes are as follows:

$$\begin{aligned}
F_{E_{i,j+\frac{1}{2}}}(u) &= \frac{1}{2}(u_{i-\frac{1}{2},j+\frac{1}{2}}+u_{i+\frac{1}{2},j+\frac{1}{2}})M_{E_{i,j+\frac{1}{2}}},\\
F_{N_{i+\frac{1}{2},j}}(u) &= \frac{1}{2}(u_{i+\frac{1}{2},j-\frac{1}{2}}+u_{i+\frac{1}{2},j+\frac{1}{2}})M_{N_{i+\frac{1}{2},j}},\\
F_{NE_{i,j}}(u) &= \frac{1}{2}(u_{i+\frac{1}{2},j+\frac{1}{2}}+u_{i-\frac{1}{2},j-\frac{1}{2}})M_{NE_{i,j}},\\
F_{SE_{i,j}}(u) &= \frac{1}{2}(u_{i-\frac{1}{2},j+\frac{1}{2}}+u_{i+\frac{1}{2},j-\frac{1}{2}})M_{SE_{i,j}},
\end{aligned} \tag{5.21}$$

where

$$\begin{aligned}
M_{E_{i,j+\frac{1}{2}}} &= \frac{1}{6}(C_{XN_{i,j}}U^U_{i,j}+C_{XS_{i,j+1}}U^U_{i,j+1}),\\
M_{N_{i+\frac{1}{2},j}} &= \frac{1}{6}(C_{YE_{i,j}}V^U_{i,j}+C_{YW_{i+1,j}}V^U_{i+1,j}),\\
M_{NE_{i,j}} &= \frac{1}{6}C_{NE_{i,j}}(U^U_{i,j}+V^U_{i,j}),\\
M_{SE_{i,j}} &= \frac{1}{6}C_{SE_{i,j}}(U^U_{i,j}-V^U_{i,j}),
\end{aligned} \tag{5.22}$$

and

$$\begin{aligned}
C_{XN_{i,j}} &= e_{i+\frac{1}{2},j+\frac{1}{2}}e_{i-\frac{1}{2},j+\frac{1}{2}}(e_{i+\frac{1}{2},j-\frac{1}{2}}e_{i-\frac{1}{2},j-\frac{1}{2}}-e_{i+\frac{1}{2},j-\frac{1}{2}}-e_{i-\frac{1}{2},j-\frac{1}{2}}+3),\\
C_{XS_{i,j}} &= e_{i+\frac{1}{2},j-\frac{1}{2}}e_{i-\frac{1}{2},j-\frac{1}{2}}(e_{i+\frac{1}{2},j+\frac{1}{2}}e_{i-\frac{1}{2},j+\frac{1}{2}}-e_{i+\frac{1}{2},j+\frac{1}{2}}-e_{i-\frac{1}{2},j+\frac{1}{2}}+3),\\
C_{YE_{i,j}} &= e_{i+\frac{1}{2},j+\frac{1}{2}}e_{i+\frac{1}{2},j-\frac{1}{2}}(e_{i-\frac{1}{2},j+\frac{1}{2}}e_{i-\frac{1}{2},j-\frac{1}{2}}-e_{i-\frac{1}{2},j+\frac{1}{2}}-e_{i-\frac{1}{2},j-\frac{1}{2}}+3),\\
C_{YW_{i,j}} &= e_{i-\frac{1}{2},j+\frac{1}{2}}e_{i-\frac{1}{2},j-\frac{1}{2}}(e_{i+\frac{1}{2},j+\frac{1}{2}}e_{i+\frac{1}{2},j-\frac{1}{2}}-e_{i+\frac{1}{2},j+\frac{1}{2}}-e_{i+\frac{1}{2},j-\frac{1}{2}}+3),\\
C_{NE_{i,j}} &= e_{i+\frac{1}{2},j+\frac{1}{2}}e_{i-\frac{1}{2},j-\frac{1}{2}}(3-e_{i-\frac{1}{2},j+\frac{1}{2}}-e_{i+\frac{1}{2},j-\frac{1}{2}}),\\
C_{SE_{i,j}} &= e_{i-\frac{1}{2},j+\frac{1}{2}}e_{i+\frac{1}{2},j-\frac{1}{2}}(3-e_{i+\frac{1}{2},j+\frac{1}{2}}-e_{i-\frac{1}{2},j-\frac{1}{2}}).
\end{aligned} \tag{5.23}$$

Finally, convergence of the horizontal momentum fluxes is written as

$$\begin{aligned}\mathrm{CAD}_{i+\frac{1}{2},j+\frac{1}{2}}(u) &= F_{E_{i,j+\frac{1}{2}}}(u) - F_{E_{i+1,j+\frac{1}{2}}}(u) + F_{N_{i+\frac{1}{2},j}}(u) - F_{N_{i+\frac{1}{2},j+1}}(u) \\ &\quad + F_{NE_{i,j}}(u) - F_{NE_{i+1,j+1}}(u) + F_{SE_{i,j+1}}(u) - F_{SE_{i+1,j}}(u). \end{aligned} \tag{5.24}$$

5.2 Viscosity

For horizontal viscosity, the harmonic (default) or biharmonic (option `VISBIHARM`) scheme can be selected. Anisotropy of viscosity with respect to the flow direction can be applied (option `VISANISO`) when harmonic viscosity is chosen. The viscosity coefficient is a constant by default but can be a function of local velocity gradients and grid-size (option `SMAGOR`).

For vertical viscosity, the harmonic scheme is used and the local coefficient is the larger of a background constant and that calculated through a turbulence closure scheme. A parameterization of bottom friction (Weatherly, 1972) is adopted at the lowest layer.

5.2.1 Horizontal viscosity

Horizontal tension D_T and horizontal shear D_S are defined as follows.

$$D_T = h_\psi \frac{\partial}{h_\mu \partial \mu}\left(\frac{u}{h_\psi}\right) - h_\mu \frac{\partial}{h_\psi \partial \psi}\left(\frac{v}{h_\mu}\right), \tag{5.25}$$

$$D_S = h_\psi \frac{\partial}{h_\mu \partial \mu}\left(\frac{v}{h_\psi}\right) + h_\mu \frac{\partial}{h_\psi \partial \psi}\left(\frac{u}{h_\mu}\right). \tag{5.26}$$

The friction terms are

$$\mathscr{V}_u = \frac{1}{h_\psi^2}\frac{\partial}{h_\mu \partial \mu}\left(h_\psi^2 \sigma_T\right) + \frac{1}{h_\mu^2}\frac{\partial}{h_\psi \partial \psi}\left(h_\mu^2 \sigma_S\right), \tag{5.27}$$

$$\mathscr{V}_v = \frac{1}{h_\psi^2}\frac{\partial}{h_\mu \partial \mu}\left(h_\psi^2 \sigma_S\right) - \frac{1}{h_\mu^2}\frac{\partial}{h_\psi \partial \psi}\left(h_\mu^2 \sigma_T\right), \tag{5.28}$$

where ν_H is the horizontal viscosity coefficient, $\sigma_T = \nu_H D_T$, and $\sigma_S = \nu_H D_S$.

In the geographical coordinate system, where $(\mu, \psi) = (\lambda, \phi)$, $h_\lambda = a\cos\phi$, and $h_\phi = a$, tension and shear are

$$D_T = \frac{1}{a\cos\phi}\frac{\partial u}{\partial \lambda} - \frac{1}{a}\frac{\partial v}{\partial \phi} - \frac{v}{a}\tan\phi, \tag{5.29}$$

$$D_S = \frac{1}{a\cos\phi}\frac{\partial v}{\partial \lambda} + \frac{1}{a}\frac{\partial u}{\partial \phi} + \frac{u}{a}\tan\phi. \tag{5.30}$$

The friction terms in this case are

$$\mathscr{V}_u = \frac{1}{a\cos\phi}\frac{\partial}{\partial \lambda}\sigma_T + \frac{1}{a}\frac{\partial}{\partial \phi}\sigma_S - \sigma_S\frac{2\tan\phi}{a}, \tag{5.31}$$

$$\mathscr{V}_v = \frac{1}{a\cos\phi}\frac{\partial}{\partial \lambda}\sigma_S - \frac{1}{a}\frac{\partial}{\partial \phi}\sigma_T + \sigma_T\frac{2\tan\phi}{a}, \tag{5.32}$$

where the third term on the r.h.s. is called the metric term.

When the `VISBIHARM` option is selected, the above operation is repeated twice using a viscosity coefficient ν_{BH}. The friction terms $\mathscr{V}_u$ and $\mathscr{V}_v$ given by (5.27) and (5.28) are sign-reversed and substituted as u and v in equations (5.25) and (5.26). 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 meso-scale eddies.

5.2.2 Horizontal anisotropic viscosity

The viscosity in an ocean general circulation model seeks to attenuate numerical noise rather than parameterizing the sub-grid 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 in eddy-less models. A biharmonic viscosity scheme has been widely used to reduce numerical noise while maintaining the eddy structure.

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 (5.27) and (5.28) to

$$\begin{pmatrix} \sigma_T \\ \sigma_S \end{pmatrix} = \left[\begin{pmatrix} \frac{1}{2}(\nu_0+\nu_1) & 0 \\ 0 & \nu_1 \end{pmatrix} + (\nu_0-\nu_1) n_\mu n_\psi \begin{pmatrix} -2n_\mu n_\psi & n_\mu^2 - n_\psi^2 \\ n_\mu^2 - n_\psi^2 & 2n_\mu n_\psi \end{pmatrix} \right] \begin{pmatrix} D_T \\ D_S \end{pmatrix}, \tag{5.33}$$

where $\hat{\mathbf{n}} = (n_\mu, n_\psi)$ is a unit vector in an arbitrary direction and ν_0 (ν_1) is the viscosity coefficient parallel (perpendicular) to $\hat{\mathbf{n}}$. When option `VISANISO` is selected, $\hat{\mathbf{n}}$ is set to the direction of local flow in MRI.COM. Given the harmonic viscosity only in the direction of flow ($\nu_1 = 0$), the numerical noise is erased while the swift currents and eddy structures are maintained.

The following is a note on usage. The ratio ν_1/ν_0 is read from namelist: `nmlvisaniso` (variable name: `cc0`; default value is 0.2). The ratio at the lateral boundary is also read from the namelist (`cc1`; default is 0.5). When the variable `flgvisequator` is set as a positive number in the namelist, the ratio ν_1/ν_0 is tapered linearly from `cc0` at the latitude `flgvisequator` (in degrees) to zero at the equator. The ratio is not tapered when a negative number is set, and the default value of `flgvisequator` is -1.

5.2.3 Smagorinsky parameterization for horizontal viscosity

To give the necessary but minimum viscosity to reduce numerical noise, the viscosity coefficient is made proportional to the local deformation rate (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 $|D|$:

$$T^{-1} = |D| = \sqrt{D_T^2 + D_S^2}, \tag{5.34}$$

the viscosity coefficients are set as follows.

$$\nu_H = \left(\frac{C\Delta_{\min}}{\pi} \right)^2 |D|, \tag{5.35}$$

$$\nu_{BH} = \frac{\Delta_{\min}^2}{8} \nu_H, \tag{5.36}$$

where C (`cscl`) is a dimensionless scaling parameter set by considering numerical stability and $\Delta_{\min}$ is the smaller of the zonal and meridional grid-spacings.

The parameter C should be selected to satisfy the following conditions.

- Restriction of grid Reynolds number:

$$\nu_H > U\frac{\Delta_{\min}}{2}, \tag{5.37}$$

- Restriction on the width of the lateral boundary layer:

$$\nu_H > \beta\Delta_{\min}^3, \tag{5.38}$$

- CFL condition:

$$\nu_H < \frac{\Delta_{\min}^2}{2\Delta t}, \tag{5.39}$$

where $\beta = df/dy$ is the meridional gradient of the Coriolis parameter. Scaling the deformation rate $|D|$ by $U/\Delta_{\min}$ gives the condition for stability: $C > \pi/\sqrt{2} \approx 2.2$ from (5.37) (Griffies and Hallberg, 2000).

5.2.4 Discretization

Using the notations

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

and

$$\begin{aligned}
\overline{A_{i,j}}^{\mu} &\equiv \frac{1}{2}(A_{i-\frac{1}{2},j} + A_{i+\frac{1}{2},j}), \\
\overline{A_{i,j}}^{\psi} &\equiv \frac{1}{2}(A_{i,j-\frac{1}{2}} + A_{i,j+\frac{1}{2}}),
\end{aligned} \tag{5.40}$$

deformation rates are discretized as follows:

$$\begin{aligned}
D_{T\,i,j} &= \frac{h_{\psi i,j}}{h_{\mu i,j}}\overline{\delta_\mu\left(\frac{u}{h_\psi}\right)_{i,j}}^{\psi} - \frac{h_{\mu i,j}}{h_{\psi i,j}}\overline{\delta_\psi\left(\frac{v}{h_\mu}\right)_{i,j}}^{\mu}, \\
D_{S\,i,j} &= \frac{h_{\psi i,j}}{h_{\mu i,j}}\overline{\delta_\mu\left(\frac{v}{h_\psi}\right)_{i,j}}^{\psi} + \frac{h_{\mu i,j}}{h_{\psi i,j}}\overline{\delta_\psi\left(\frac{u}{h_\mu}\right)_{i,j}}^{\mu}.
\end{aligned} \tag{5.41}$$

Viscosity forces are discretized as follows:

$$\begin{aligned}
F_{x\,i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}}} \times \\
&\times \left[\frac{1}{h^2_{\psi i+\frac{1}{2},j+\frac{1}{2}}}\delta_i\left(\Delta y\Delta z h_\psi^2\overline{\nu_H D_T}^{\psi}\right)_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{1}{h^2_{\mu\,i+\frac{1}{2},j+\frac{1}{2}}}\delta_j\left(\Delta x\Delta z h_\mu^2\overline{\nu_H D_S}^{\mu}\right)_{i+\frac{1}{2},j+\frac{1}{2}}\right], \\
F_{y\,i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}}} \times \\
&\times \left[\frac{1}{h^2_{\psi i+\frac{1}{2},j+\frac{1}{2}}}\delta_i\left(\Delta y\Delta z h_\psi^2\overline{\nu_H D_S}^{\psi}\right)_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{1}{h^2_{\mu\,i+\frac{1}{2},j+\frac{1}{2}}}\delta_j\left(\Delta x\Delta z h_\mu^2\overline{\nu_H D_T}^{\mu}\right)_{i+\frac{1}{2},j+\frac{1}{2}}\right].
\end{aligned} \tag{5.42}$$

When the grid point $(i-\frac{1}{2}, j+\frac{1}{2})$ is defined as a (vertically partial) land (Figure 5.6a), the velocity gradients at the wall are calculated as follows.

$$\begin{aligned}\left(\frac{\partial u}{\partial x}\right)_{i,j+\frac{1}{2}} &= \frac{u_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x^{-}_{ij}},\\ \left(\frac{\partial v}{\partial x}\right)_{i,j+\frac{1}{2}} &= \frac{v_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x^{-}_{ij}},\end{aligned} \tag{5.43}$$

where Δx^{-}_{ij} is the length between the points $(i, j+\frac{1}{2})$ and $(i+\frac{1}{2}, j+\frac{1}{2})$. The contribution of this wall to the force is:

$$\begin{aligned}F^{W}_{x_{i+\frac{1}{2},j+\frac{1}{2}}} &= -\frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}} h^{2}_{\psi_{i+\frac{1}{2},j+\frac{1}{2}}}} \Delta y_{i,j+\frac{1}{2}} \Delta \tilde{z}_{i,j+\frac{1}{2}} h^{2}_{\psi_{i-\frac{1}{2},j}} \nu_{Hi-\frac{1}{2},j} \frac{u_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x^{-}_{i+\frac{1}{2},j+\frac{1}{2}}},\\ F^{W}_{y_{i+\frac{1}{2},j+\frac{1}{2}}} &= -\frac{1}{\Delta V_{i+\frac{1}{2},j+\frac{1}{2}} h^{2}_{\psi_{i+\frac{1}{2},j+\frac{1}{2}}}} \Delta y_{i,j+\frac{1}{2}} \Delta \tilde{z}_{i,j+\frac{1}{2}} h^{2}_{\psi_{i-\frac{1}{2},j}} \nu_{Hi-\frac{1}{2},j} \frac{v_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x^{-}_{i+\frac{1}{2},j+\frac{1}{2}}},\end{aligned} \tag{5.44}$$

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

5.2.5 Vertical viscosity

Only the harmonic scheme is considered. The vertical momentum flux is assumed to be proportional to the vertical gradient of velocity. For the upper part of a U-cell at the $(k+\frac{1}{2})$th vertical level, the momentum flux (positive upward) is calculated as follows:

$$-\left(\nu_v \frac{\partial u}{\partial z}\right)_k = -\nu_{vk} \frac{u_{k-\frac{1}{2}} - u_{k+\frac{1}{2}}}{\Delta z_k},$$

where $\Delta z_k = (\Delta z_{k-\frac{1}{2}} + \Delta z_{k+\frac{1}{2}})/2$, and $\Delta z_{k+\frac{1}{2}}$ is the thickness of the U-cell (`dzu`). Similarly, the momentum flux in the lower part of the U-cell is calculated as follows:

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

where ν_{vk+1} is set to zero if the $(k+\frac{3}{2})$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 at the bottom and near the sea surface are not considered when evaluating fluxes for simplicity.

To calculate viscosity, the divergence of the momentum flux is first calculated. The expression for the vertical viscosity term is

$$\frac{\partial}{\partial z}\left(\nu_v \frac{\partial u}{\partial z}\right)_{k+\frac{1}{2}} = \frac{\left(\nu_v \frac{\partial u}{\partial z}\right)_k - \left(\nu_v \frac{\partial u}{\partial z}\right)_{k+1}}{\widehat{\Delta z}_{k+\frac{1}{2}}} = \frac{\nu_{vk}(u_{k-\frac{1}{2}} - u_{k+\frac{1}{2}})}{\Delta z_k \widehat{\Delta z}_{k+\frac{1}{2}}} - \frac{\nu_{vk+1}(u_{k+\frac{1}{2}} - u_{k+\frac{3}{2}})}{\Delta z_{k+1} \widehat{\Delta z}_{k+\frac{1}{2}}} \tag{5.45}$$

where the variation of the grid thickness for the U-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 5.6(b)). Note that the first term on the r.h.s. of equation (5.45) is set to zero in calculating the viscosity term for the vertical level of $\frac{1}{2}$ ($k=0$).

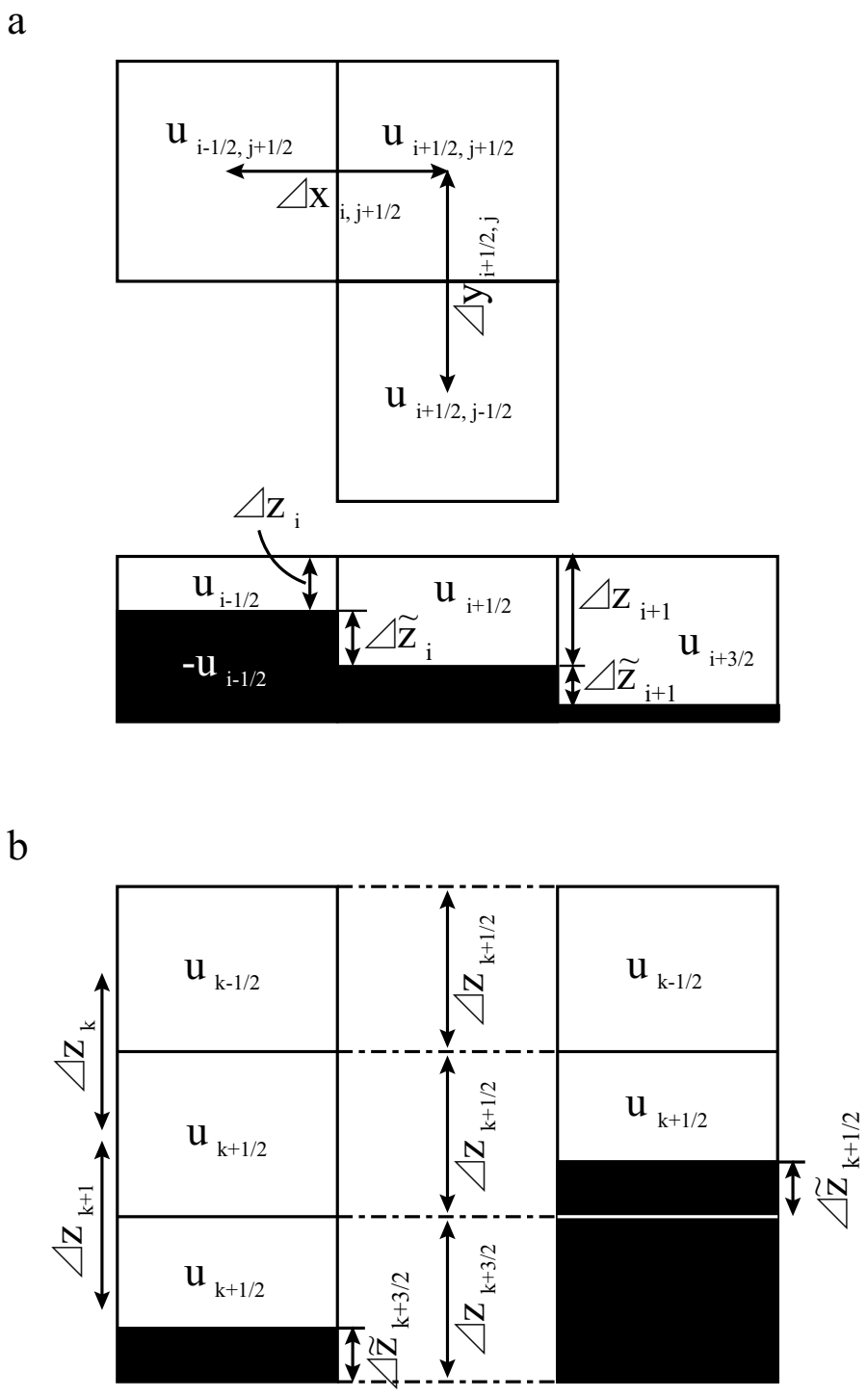

Figure 5.6. (a) A schematic distribution of grids for horizontal viscosity. Upper: Plan view. Lower: Side view. The shadings denote solid earth. (b) A schematic distribution of grids for vertical viscosity. Side views. Left: The lower adjacent layer $(k+\frac{3}{2})$ has a sea bed. Right: The U-cell $(k+\frac{1}{2})$ has a sea bed.

5.2.6 Bottom friction

When a U-cell in the $(k+\frac{1}{2})$th layer contains solid earth (Figure 5.6(b) right), the stress from the lower boundary (τ_x^b, τ_y^b) is calculated following Weatherly (1972). The specific expression is as follows.

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

where C_{btm} is a dimensionless constant. Viscous stress at the lower boundary has a magnitude proportional to the square of the flow speed at the U-cell and an angle $(\theta_0+\pi)$ relative to the flow direction.

In MRI.COM,

$$\begin{aligned} C_{\mathrm{btm}} &= 1.225\times10^{-3} \\ \theta_0 &= \pm\pi/18 \quad [\mathrm{rad}] \qquad (\equiv \quad 10[^\circ]), \end{aligned}$$

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

References

Delhez, E. J. M., and E. Deleersnijder, 2007: Overshootings and spurious oscillations caused by biharmonic mixing., *Ocean Modell.*, *17*, 183–198.

Griffies, S., and R. Hallberg, 2000: Biharmonic friction with a Smagorinsky-like viscosity for use in large-scale eddy-permitting ocean models, *Mon. Weather Rev.*, *128*, 2935–2946.

Ishizaki, H., and T. Motoi, 1999: Reevaluation of the Takano-Oonishi scheme for momentum advection on bottom relief in ocean models, *J. Atmos. Oceanic Tech.*, *16*, 1994–2010.

Smagorinsky, J., 1963: General circulation experiments with the primitive equations: I. The basic experiment, *Mon. Weather Rev.*, *91*, 99–164.

Smith, R. D., and J. C. McWilliams, 2003: Anisotropic horizontal viscosity for ocean models., *Ocean Modell.*, *5*, 129–156.

Weatherly, G. L., 1972: A study of the bottom boundary layer of the Florida current, *J. Phys. Oceanogr.*, *2*, 54–72.

Chapter 6 Temperature and salinity equations

This chapter formulates a basic part of the temperature and salinity prediction procedure, except for ocean surface processes (related to mixed layer and surface fluxes). First, the finite difference expression of the flux form is explained (Section 6.1). The formulation of each component of the flux (advection and diffusion) is then given (Sections 6.2 and 6.3). The advection scheme mainly focuses on the Quadratic Upstream Interpolation for Convective Kinematics (QUICK; Leonard, 1979) scheme. Other advection schemes, QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979; option `QUICKEST`), the Uniformly Third-Order Polynomial Interpolation Algorism (UTOPIA; Leonard et al., 1993; option `UTOPIA`), and the Second Order Moment (SOM; Prather, 1986; option `SOMADVEC`) are explained in Chapter 13. Finally, a convective adjustment scheme is explained (Section 6.4).

6.1 Flux form

The equations governing the time change of temperature and salinity (differential form) are presented in equations (2.14) to (2.29) of Chapter 2. The finite difference forms of (2.14) and (2.15) are given by calculating fluxes through each cell face, and setting their divergence and convergence to be the time change at the grid cell (Figure 6.1) as follows:

$$\begin{aligned} T^{n+1}_{i,j,k+\frac{1}{2}} \Delta V^{n+1}_{i,j,k+\frac{1}{2}} &= T^{n-1}_{i,j,k+\frac{1}{2}} \Delta V^{n-1}_{i,j,k+\frac{1}{2}} \\ &\quad + 2\Delta t \left\{ FX_{i-\frac{1}{2},j,k+\frac{1}{2}} - FX_{i+\frac{1}{2},j,k+\frac{1}{2}} + FY_{i,j-\frac{1}{2},k+\frac{1}{2}} - FY_{i,j+\frac{1}{2},k+\frac{1}{2}} + FZ_{i,j,k+1} - FZ_{i,j,k} \right\}, \end{aligned} \tag{6.1}$$

where T is temperature or salinity, ΔV is the volume of the grid cell, and FX, FY, and FZ represent (*flux*) $\times$ (*area of the grid boundary*). The fluxes include contributions by advection and diffusion. Note that ΔV from $k = 1$ to $k =$ `ksgm` varies with time (Chapter 4).

The volume of the grid cell is written as follows:

$$\begin{aligned} \Delta V_{i,j,k} &= \mathtt{a\_tr}_{i-\frac{1}{2},j-\frac{1}{2}} \Delta z_{i-\frac{1}{2},j-\frac{1}{2},k} + \mathtt{a\_tl}_{i+\frac{1}{2},j-\frac{1}{2}} \Delta z_{i+\frac{1}{2},j-\frac{1}{2},k} \\ &\quad + \mathtt{a\_br}_{i-\frac{1}{2},j+\frac{1}{2}} \Delta z_{i-\frac{1}{2},j+\frac{1}{2},k} + \mathtt{a\_bl}_{i+\frac{1}{2},j+\frac{1}{2}} \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k}, \end{aligned} \tag{6.2}$$

where Δz is the thickness of the corresponding velocity cell (in which fluctuations of the surface height are considered). The horizontal area of the left-lower quarter of the T-cells at (i, j) (corresponding to the southwestern part in geographical coordinates) is represented by $\mathtt{a\_tr}_{i-\frac{1}{2},j-\frac{1}{2}}$. Similarly, $\mathtt{a\_tl}_{i+\frac{1}{2},j-\frac{1}{2}}$ is the right-lower (southeastern), $\mathtt{a\_br}_{i-\frac{1}{2},j+\frac{1}{2}}$ is the left-upper (northwestern), and $\mathtt{a\_bl}_{i+\frac{1}{2},j+\frac{1}{2}}$ is the right-upper (northeastern) quarter of a T-cell at (i, j). See also Section 3.5.

6.2 Advection

There are several choices for tracer advection schemes. For horizontal advection, the Quadratic Upstream Interpolation for Convective Kinematics (QUICK; Leonard, 1979), the Uniformly Third-Order Polynomial Interpo-

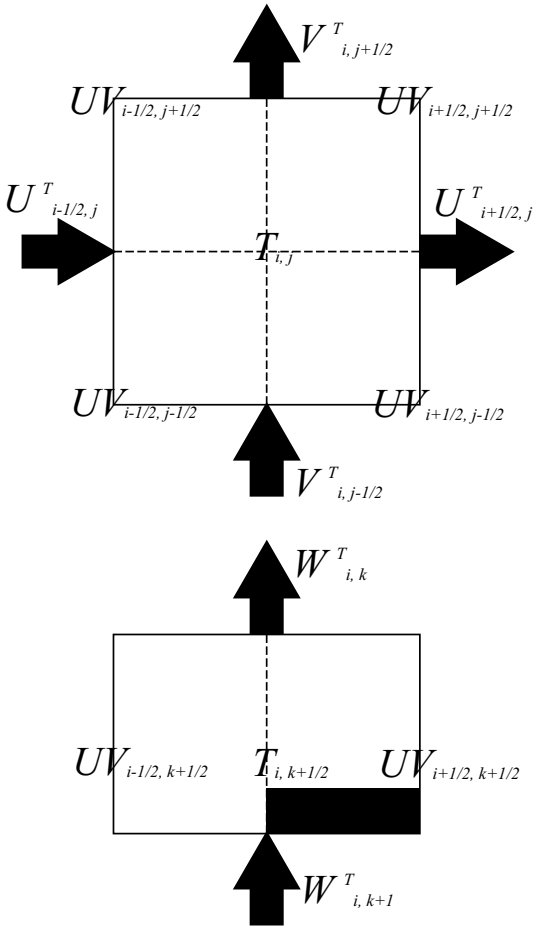

Figure 6.1. Grid arrangement around TS-Box (Upper: Views from the upper, Below: views from the horizontal). Fluxes represented by an arrow are calculated.

lation Algorithm (UTOPIA; Leonard et al., 1993), and the Second Order Moment (SOM; Prather, 1986) schemes are available. For vertical advection, QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979), is available in addition to QUICK and SOM. When SOM is selected, it should be used for both horizontal and vertical directions. Refer to Leonard (1979) for a comparison between QUICK and QUICKEST. Leonard et al. (1993) provides a detailed explanation of the extension from QUICKEST to UTOPIA. The procedure for calculating UTOPIA is based on Leonard et al. (1994). This section explains the QUICK scheme. Explanations for QUICKEST, UTOPIA, and SOM are given in Chapter 13.

Fluxes due to advection are given as follows:

$$FXA_{i+\frac{1}{2},j,k+\frac{1}{2}} = U^T_{i+\frac{1}{2},j,k+\frac{1}{2}} T_{i+\frac{1}{2},j,k+\frac{1}{2}}, \tag{6.3}$$

$$FYA_{i,j+\frac{1}{2},k+\frac{1}{2}} = V^T_{i,j+\frac{1}{2},k+\frac{1}{2}} T_{i,j+\frac{1}{2},k+\frac{1}{2}}, \tag{6.4}$$

$$FZA_{i,j,k+1} = W^T_{i,j,k+1} T_{i,j,k+1}, \tag{6.5}$$

where horizontal volume transport U^T and V^T are defined as follows, using (3.4) to (3.9),

$$U^T_{i+\frac{1}{2},j,k+\frac{1}{2}} = \frac{\Delta y_{i+\frac{1}{2},j}}{2}\left(u_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}\Delta z_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\Delta z_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right), \tag{6.6}$$

$$V^T_{i,j+\frac{1}{2},k+\frac{1}{2}} = \frac{\Delta x_{i,j+\frac{1}{2}}}{2}\left(v_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\Delta z_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} + v_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\Delta z_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right). \tag{6.7}$$

Vertical volume transport W^T is then obtained by diagnostically solving (3.3). Moreover, the vertical velocity w, which is necessary for using QUICKEST, is calculated as follows (w is not needed, except for QUICKEST):

$$W^T_{i,j,k+1} = w_{i,j,k+1} \times \mathtt{areat}_{i,j,k+\frac{1}{2}}, \tag{6.8}$$

where

$$\begin{aligned}\mathtt{areat}_{i,j,k+\frac{1}{2}} = {} & \mathtt{a\_tr}_{i-\frac{1}{2},j-\frac{1}{2}} \times eu_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} + \mathtt{a\_tl}_{i+\frac{1}{2},j-\frac{1}{2}} \times eu_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} \\ & + \mathtt{a\_br}_{i-\frac{1}{2},j+\frac{1}{2}} \times eu_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} + \mathtt{a\_bl}_{i+\frac{1}{2},j+\frac{1}{2}} \times eu_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}.\end{aligned} \tag{6.9}$$

An array *eu* 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 grid 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 $T_{i,j,k+1}$. The centered finite difference scheme uses the average between two neighboring points of temperature and salinity as the grid boundary value. In the QUICK scheme, the grid boundary value is interpolated by a quadratic function, using three points, with one of them added from the upstream side (Figure 6.2).

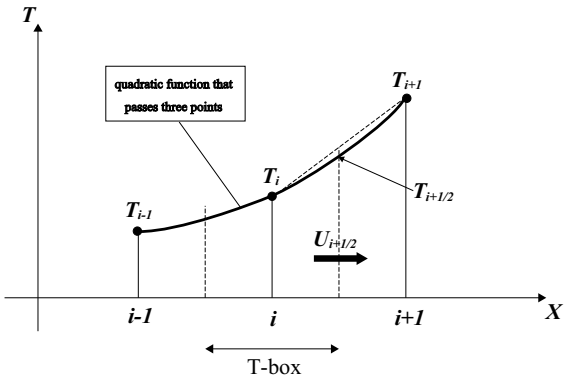

Figure 6.2. Schematic for interpolation: T_i indicates the representative temperature currently calculated at T-point, and $T_{i+\frac{1}{2}}$ is the grid boundary value. In the QUICK scheme, $T_{i+\frac{1}{2}}$ is interpolated by a quadratic function that passes T_i and the neighboring T-point values, T_{i-1} and T_{i+1}.

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

$$T_{i+\frac{1}{2},j,k+\frac{1}{2}} = \frac{\Delta x_i T_{i+1,j,k+\frac{1}{2}} + \Delta x_{i+1} T_{i,j,k+\frac{1}{2}}}{\Delta x_{i+1} + \Delta x_i} - \frac{\Delta x_{i+1}\Delta x_i}{4} c_{i+\frac{1}{2},j,k+\frac{1}{2}}, \tag{6.10}$$

$$T_{i,j+\frac{1}{2},k+\frac{1}{2}} = \frac{\Delta y_j T_{i,j+1,k+\frac{1}{2}} + \Delta y_{j+1} T_{i,j,k+\frac{1}{2}}}{\Delta y_{j+1} + \Delta y_j} - \frac{\Delta y_{j+1}\Delta y_j}{4} d_{i,j+\frac{1}{2},k+\frac{1}{2}}, \tag{6.11}$$

$$T_{i,j,k+1} = \frac{\Delta z_{k+\frac{1}{2}} T_{i,j+1,k+\frac{3}{2}} + \Delta z_{k+\frac{3}{2}} T_{i,j,k+\frac{1}{2}}}{\Delta z_{k+\frac{3}{2}} + \Delta z_{k+\frac{1}{2}}} - \frac{\Delta z_{k+\frac{1}{2}}\Delta z_{k+\frac{3}{2}}}{4} e_{i,j,k+1}, \tag{6.12}$$

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

$$\begin{aligned}
c_{i+\frac{1}{2},j,k+\frac{1}{2}} &= \frac{\Delta x_i \delta_x \delta_x T_{i,j,k+\frac{1}{2}}}{2\overline{\Delta x_i}^x} (= c_p), \quad \text{if} \quad U^T_{i+\frac{1}{2},j,k+\frac{1}{2}} > 0, \\
&= \frac{\Delta x_{i+1} \delta_x \delta_x T_{i+1,j,k+\frac{1}{2}}}{2\overline{\Delta x_{i+1}}^x} (= c_m), \quad \text{if} \quad U^T_{i+\frac{1}{2},j,k+\frac{1}{2}} < 0, \\
d_{i,j+\frac{1}{2},k+\frac{1}{2}} &= \frac{\Delta y_j \delta_y \delta_y T_{i,j,k+\frac{1}{2}}}{2\overline{\Delta y_j}^y} (= d_p), \quad \text{if} \quad V^T_{i,j+\frac{1}{2},k+\frac{1}{2}} > 0, \\
&= \frac{\Delta y_{j+1} \delta_y \delta_y T_{i,j+1,k+\frac{1}{2}}}{2\overline{\Delta y_{j+1}}^y} (= d_m), \quad \text{if} \quad V^T_{i,j+\frac{1}{2},k+\frac{1}{2}} < 0, \\
e_{i,j,k+1} &= \frac{\Delta z_{k+\frac{3}{2}} \delta_z \delta_z T_{i,j,k+\frac{3}{2}}}{2\overline{\Delta z_{i,j,k+\frac{3}{2}}}^z} (= e_p), \quad \text{if} \quad W^T_{i,j,k+1} > 0, \\
&= \frac{\Delta z_{k+\frac{1}{2}} \delta_z \delta_z T_{i,j,k+\frac{1}{2}}}{2\overline{\Delta z_{i,j,k+\frac{1}{2}}}^z} (= e_m), \quad \text{if} \quad W^T_{i,j,k+1} < 0.
\end{aligned} \tag{6.13}$$

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

$$\begin{aligned}
\delta_x A_i &\equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta x_i} \quad , \quad \delta_x A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_i}{\Delta x_{i+\frac{1}{2}}}, \\
\overline{A_i}^x &\equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2} \quad , \quad \overline{A_{i+\frac{1}{2}}}^x \equiv \frac{A_{i+1} + A_i}{2}.
\end{aligned} \tag{6.14}$$

Letting c_p, d_p, and e_p represent their values for positive velocity at the grid boundary and c_m, d_m, and e_m represent their values for negative velocity at grid boundary and taking

$$c_a = c_m + c_p \tag{6.15}$$

$$c_d = c_m - c_p \tag{6.16}$$

$$d_a = d_m + d_p \tag{6.17}$$

$$d_d = d_m - d_p \tag{6.18}$$

$$e_a = e_m + e_p \tag{6.19}$$

$$e_d = e_m - e_p, \tag{6.20}$$

we obtain

$$\begin{aligned}
FXA_{i+\frac{1}{2},j,k+\frac{1}{2}} =& U^T_{i+\frac{1}{2},j,k+\frac{1}{2}} \left[\frac{\Delta x_{i,j} T_{i+1,j,k+\frac{1}{2}} + \Delta x_{i+1,j} T_{i,j,k+\frac{1}{2}}}{\Delta x_{i+1,j} + \Delta x_{i,j}} - \frac{\Delta x_{i+1,j} \Delta x_{i,j}}{8} c_{a\,i+\frac{1}{2},j,k+\frac{1}{2}} \right] \\
&+ |U^T_{i+\frac{1}{2},j,k+\frac{1}{2}}| \frac{\Delta x_{i+1,j} \Delta x_{i,j}}{8} c_{d\,i+\frac{1}{2},j,k+\frac{1}{2}},
\end{aligned} \tag{6.21}$$

$$\begin{aligned}
FYA_{i,j+\frac{1}{2},k+\frac{1}{2}} =& V^T_{i,j+\frac{1}{2},k+\frac{1}{2}} \left[\frac{\Delta y_{i,j} T_{i,j+1,k+\frac{1}{2}} + \Delta y_{i,j+1} T_{i,j,k+\frac{1}{2}}}{\Delta y_{i,j+1} + \Delta y_{i,j}} - \frac{\Delta y_{i,j+1} \Delta y_{i,j}}{8} d_{a\,i,j+\frac{1}{2},k+\frac{1}{2}} \right] \\
&+ |V^T_{i,j+\frac{1}{2},k+\frac{1}{2}}| \frac{\Delta y_{i,j+1} \Delta y_{i,j}}{8} d_{d\,i,j+\frac{1}{2},k+\frac{1}{2}},
\end{aligned} \tag{6.22}$$

$$\begin{aligned}
FZA_{i,j,k+1} =& W^T_{i,j,k+1} \left[\frac{\Delta z_{i,j,k+\frac{1}{2}} T_{i,j,k+\frac{3}{2}} + \Delta z_{i,j,k+\frac{3}{2}} T_{i,j,k+\frac{1}{2}}}{\Delta z_{i,j,k+\frac{3}{2}} + \Delta z_{i,j,k+\frac{1}{2}}} - \frac{\Delta z_{i,j,k+\frac{3}{2}} \Delta z_{i,j,k+\frac{1}{2}}}{8} e_{a\,i,j,k+1} \right] \\
&+ |W^T_{i,j,k+\frac{1}{2}}| \frac{\Delta z_{i,j,k+\frac{3}{2}} \Delta z_{i,j,k+\frac{1}{2}}}{8} e_{d\,i,j,k+1}.
\end{aligned} \tag{6.23}$$

Equation (6.21) can be rewritten as

$$FXA_{i+\frac{1}{2},j,k+\frac{1}{2}} \simeq U^T_{i+\frac{1}{2},j,k+\frac{1}{2}} \tilde{T}_{i+\frac{1}{2},j,k+\frac{1}{2}} + A_Q \frac{\partial^3 T_{i+\frac{1}{2},j,k+\frac{1}{2}}}{\partial x^3}, \tag{6.24}$$

where $\tilde{T}_{i+\frac{1}{2},j,k+\frac{1}{2}}$ is the value of T at the grid boundary interpolated by the cubic polynomial, and

$$A_Q = |U^T_{i+\frac{1}{2},j,k+\frac{1}{2}}| \frac{\Delta x_{i+1} \Delta x_{i+\frac{1}{2}} \Delta x_i}{8}. \tag{6.25}$$

Although the time integration for advection is done by the leap-frog scheme, the second term on the r.h.s. of (6.24) 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 $w_{i,j,k} > 0$ and the T-point at $(i, j, k+\frac{3}{2})$ is below the bottom. The upstream-side weighting ratio is given by the user as the namelist parameter `vupp`.

6.3 Diffusion

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 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.

Three types of horizontal diffusion, harmonic horizontal diffusion (default), biharmonic horizontal diffusion (option `TRCBIHARM`), and isopycnal diffusion (option `ISOPYCNAL`), 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. This is realized by merely setting two mixing coefficients for the isopycnal diffusion tensor (Griffies, 1998). An anisotropic GM scheme (Smith and Gent, 2004, option `GMANISOTROP`), which gives greater diffusivity only in the direction of the current vector, is also available.

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. However, biharmonic diffusion is not recommended in eddy-less models because this would result in numerical instability. Harmonic horizontal diffusion is 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. Isopycnal diffusion mixes tracers along neutral surfaces. The GM scheme represents eddy-induced transports in isopycnal layers, mimicking transport caused by baroclinic instability. 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.

6.3.1 Vertical diffusion

Vertical diffusion assumes that vertical diffusion flux is proportional to the vertical gradient of temperature and salinity. By default, vertical diffusivity is given as a function of depth. Its profile is stored in the one dimensional array $\mathtt{vdbg}(1:\mathtt{km})$ (e.g., Tsujino et al., 2000). A three dimensional distribution can be set by selecting option `VMBG3D` 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.

In addition to these static profiles, the following processes give vertical diffusivity coefficients for every model time step.

- Surface mixed layer models.
- Vertical component of isopycnal diffusion,
- Enhanced diffusivity where the stratification is unstable (option `DIFAJS`).
- Enhanced diffusivity around rivermouths to avoid negative salinity (by setting `flg_enhance_vm_rivmouth` of namelist `nrivermouth` to be `.true.`). This scheme is needed when positive definiteness is not guaranteed by a tracer advection algorithm.

The vertical diffusion for "this" time step is taken as the largest of the above estimations. When one or more of the above mixing schemes is employed, a backward (implicit) scheme is used in the time integration (Section 12.5) because high diffusivity is expected. Otherwise, a forward scheme is used.

The finite difference form is as follows:

$$FZD_{i,j,k+1} = -\kappa_z \, \mathtt{areat}_{i,j,k+1} \, \delta_z T_{i,j,k+1}, \tag{6.26}$$

where

$$\delta_z T_{i,j,k+1} \equiv \frac{T_{i,j,k+\frac{1}{2}} - T_{i,j,k+\frac{3}{2}}}{\Delta z_{k+1}} \tag{6.27}$$

Noted 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+1} when calculating the gradient.

6.3.2 Harmonic horizontal diffusion

Harmonic horizontal diffusion assumes that diffusion fluxes are proportional to the gradient of temperature and salinity. The finite differences of the fluxes are as follows:

$$FXD_{i+\frac{1}{2},j,k+\frac{1}{2}} = -\kappa_H \Delta y_{i+\frac{1}{2},j} \overline{\Delta z_{i+\frac{1}{2},j,k+\frac{1}{2}}}^{\,y} \delta_x T_{i+\frac{1}{2},j,k+\frac{1}{2}}, \tag{6.28}$$

$$FYD_{i,j+\frac{1}{2},k+\frac{1}{2}} = -\kappa_H \Delta x_{i,j+\frac{1}{2}} \overline{\Delta z_{i,j+\frac{1}{2},k+\frac{1}{2}}}^{\,x} \delta_y T_{i,j+\frac{1}{2},k+\frac{1}{2}}, \tag{6.29}$$

where

$$\delta_x T_{i+\frac{1}{2},j,k+\frac{1}{2}} \equiv \frac{T_{i+1,j,k+\frac{1}{2}} - T_{i,j,k+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j}}, \tag{6.30}$$

$$\delta_y T_{i,j+\frac{1}{2},k+\frac{1}{2}} \equiv \frac{T_{i,j+1,k+\frac{1}{2}} - T_{i,j,k+\frac{1}{2}}}{\Delta y_{i,j+\frac{1}{2}}}. \tag{6.31}$$

6.3.3 Biharmonic horizontal diffusion

Biharmonic horizontal diffusion assumes that diffusion flux is proportional to the gradient of the Laplacian of temperature and salinity. The finite difference of the flux is as follows:

$$FXD_{i+\frac{1}{2},j,k+\frac{1}{2}} = \kappa_b \Delta y_{i+\frac{1}{2},j} \overline{\Delta z_{i+\frac{1}{2},j,k+\frac{1}{2}}}^{\,y} \delta_x \nabla^2 T_{i+\frac{1}{2},j,k+\frac{1}{2}}, \tag{6.32}$$

$$FYD_{i,j+\frac{1}{2},k+\frac{1}{2}} = \kappa_b \Delta x_{i,j+\frac{1}{2}} \overline{\Delta z_{i,j+\frac{1}{2},k+\frac{1}{2}}}^{\,x} \delta_y \nabla^2 T_{i,j+\frac{1}{2},k+\frac{1}{2}}, \tag{6.33}$$

where

$$\nabla^2 T_{i,j,k+\frac{1}{2}} = \frac{\Delta x_{i,j} \Delta y_{i,j}}{\Delta V_{i,j,k+\frac{1}{2}}} (\delta_x \overline{\Delta z_{i,j,k+\frac{1}{2}}}^{\,y} \delta_x T_{i,j,k+\frac{1}{2}} + \delta_y \overline{\Delta z_{i,j,k+\frac{1}{2}}}^{\,x} \delta_y T_{i,j,k+\frac{1}{2}}). \tag{6.34}$$

6.3.4 Isopycnal diffusion

In isopycnal diffusion, 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 temperature and salinity in each direction. Using diffusion tensor $\mathbf{K}$, each flux component is written as $F^m(T) = -K^{mn}\partial_n T$, and then

$$\mathbf{K} = \frac{\kappa_I}{1+S^2}\begin{pmatrix} 1+S_y^2+\varepsilon S_x^2 & (\varepsilon-1)S_xS_y & (1-\varepsilon)S_x \\ (\varepsilon-1)S_xS_y & 1+S_x^2+\varepsilon S_y^2 & (1-\varepsilon)S_y \\ (1-\varepsilon)S_x & (1-\varepsilon)S_y & \varepsilon+S^2 \end{pmatrix}, \tag{6.35}$$

where

$$\mathbf{S} = (S_x, S_y, 0) = \begin{pmatrix} -\dfrac{\dfrac{\partial\rho}{\partial x}}{\dfrac{\partial\rho}{\partial z}}, -\dfrac{\dfrac{\partial\rho}{\partial y}}{\dfrac{\partial\rho}{\partial z}}, 0 \end{pmatrix}, \tag{6.36}$$

$$S = |\,\mathbf{S}\,|, \tag{6.37}$$

$$\varepsilon = \frac{\kappa_D}{\kappa_I} \tag{6.38}$$

(Redi, 1982).

The finite difference is based on Cox (1987) except for the small isopycnal slope approximation. The finite difference form of three components of the gradient of temperature and salinity in calculating the east-west component of flux $FXD_{i+\frac{1}{2},j,k}$ is as follows:

$$(\delta_x T)_{i+\frac{1}{2},j,k+\frac{1}{2}} = \delta_x T_{i+\frac{1}{2},j,k+\frac{1}{2}}, \tag{6.39}$$

$$(\delta_y T)_{i+\frac{1}{2},j,k+\frac{1}{2}} = \overline{\delta_y T_{i+\frac{1}{2},j,k+\frac{1}{2}}}^{xy}, \tag{6.40}$$

$$(\delta_z T)_{i+\frac{1}{2},j,k+\frac{1}{2}} = \overline{\delta_z T_{i+\frac{1}{2},j,k+\frac{1}{2}}}^{xz}. \tag{6.41}$$

Similarly, the north-south component is given by

$$(\delta_x T)_{i,j+\frac{1}{2},k+\frac{1}{2}} = \overline{\delta_x T_{i,j+\frac{1}{2},k+\frac{1}{2}}}^{xy}, \tag{6.42}$$

$$(\delta_y T)_{i,j+\frac{1}{2},k+\frac{1}{2}} = \delta_y T_{i,j+\frac{1}{2},k+\frac{1}{2}}, \tag{6.43}$$

$$(\delta_z T)_{i,j+\frac{1}{2},k+\frac{1}{2}} = \overline{\delta_z T_{i,j+\frac{1}{2},k+\frac{1}{2}}}^{yz}, \tag{6.44}$$

and the vertical component is given by

$$(\delta_x T)_{i,j,k+1} = \overline{\delta_x T_{i,j,k+1}}^{xz}, \tag{6.45}$$

$$(\delta_y T)_{i,j,k+1} = \overline{\delta_y T_{i,j,k+1}}^{yz}, \tag{6.46}$$

$$(\delta_z T)_{i,j,k+1} = \delta_z T_{i,j,k+1} \tag{6.47}$$

(Figure 6.3).

The density gradient in the calculation of each component of the diffusion tensor can be obtained by replacing T in the above equation 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 $k+1$ for the vertical component.

In addition, the upper bound on the isopycnal slope $S_{\max}$ is set because a nearly vertical isopycnal slope and the resultant low horizontal diffusivity could cause numerical instability. If $|\mathbf{S}| > S_{\max}$, $\partial_z\rho$ in all flux components is replaced so as to satisfy $|\mathbf{S}| = S_{\max}$.

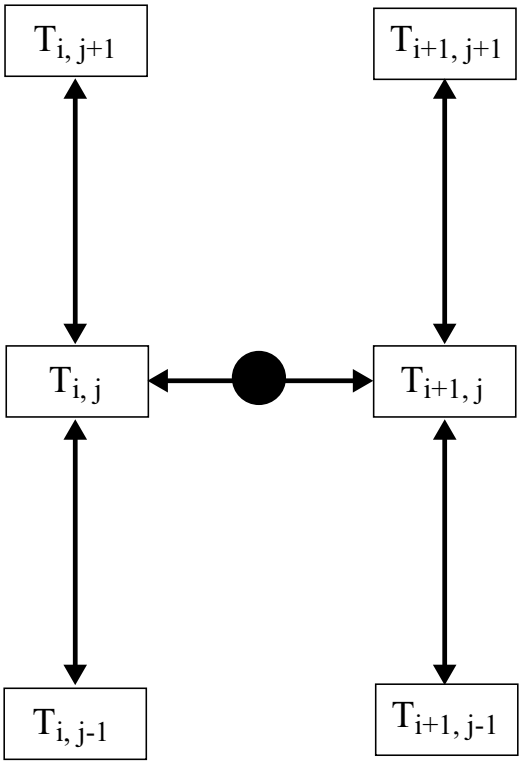

Figure 6.3. The way of calculating the gradient at the circle $(i+\frac{1}{2}, j, k+\frac{1}{2})$ in isopycnal diffusion: the east-west gradient is indicated by an arrow through the circle, and the north-south gradient is given by averaging four arrows in the vertical direction.

Griffies et al. (1998) noted a problem in the finite difference expression of the isopycnal diffusion as implemented in the GFDL-model by Cox (1987). The problem is that two properties, the down-gradient orientation of the diffusive fluxes along the neutral directions and the zero isoneutral diffusive flux of locally referenced density, are not guaranteed because of 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.

6.3.5 Gent and McWilliams parameterization

The Gent and McWilliams (1990) parameterization represents transports of temperature and salinity 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:

$$R(T) = \partial_m (J^{mn} \partial_n T) \tag{6.48}$$

Diffusion tensor J^{mn} is expressed as the sum of the symmetric component $K^{mn} = (J^{mn} + J^{nm})/2$ and the anti-symmetric component $A^{mn} = (J^{mn} - J^{nm})/2$. Isopycnal diffusion has the form of a symmetric diffusion tensor.

Convergence of a skew flux caused by the anti-symmetric component $F^m_{\rm skew} = -A^{mn}\partial_n T$ is as follows:

$$\begin{aligned} R_A(T) &= \partial_m(A^{mn}\partial_n T) \\ &= \partial_m(A^{mn})\partial_n T \\ &= \partial_n(\partial_m A^{mn} T), \end{aligned} \tag{6.49}$$

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 $\mathbf{F}_{\rm adv} = \mathbf{u}_* T$ and $R_A(T) = -\mathbf{u}_* \cdot \nabla T$ since $\mathbf{u}_*$ is divergence free by definition.

The Gent and McWilliams parameterization is given by

$$\mathbf{u}_* = -\frac{\partial}{\partial\rho}(\kappa_T \nabla_\rho h) \Big/ \frac{\partial h}{\partial\rho}, \tag{6.50}$$

where h is the depth of the neutral surface ($\rho = $ const). This velocity is expressed in the depth coordinate as

$$\mathbf{u}_* = (-\partial_z(\kappa_T S_x), -\partial_z(\kappa_T S_y), \nabla_h \cdot (\kappa_T \mathbf{S})) \tag{6.51}$$

(Gent et al., 1995).

Generally, the corresponding anti-symmetric tensor $\mathbf{A}$ can not be uniquely determined. Here, from $u^1_* = -\partial_y A^{21} - \partial_z A^{31} = -\partial_z(\kappa_T S_x)$, we choose $A^{21} = 0$, $A^{31} = \kappa_T S_x$. Likewise, we adopt $A^{12} = 0$, $A^{32} = \kappa_T S_y$. A specific form of $\mathbf{A}$ for the Gent and McWilliams parameterization is given as follows:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & -\kappa_T S_x \\ 0 & 0 & -\kappa_T S_y \\ \kappa_T S_x & \kappa_T S_y & 0 \end{pmatrix}. \tag{6.52}$$

The flux due to advection can be expressed using a vector streamfunction, $\Psi = (-\kappa_T S_y, \kappa_T S_x, 0)$, which produces $\mathbf{u}_*$ in (6.51):

$$\mathbf{F}_{\rm adv} = \mathbf{u}_* T = T(\nabla \times \Psi).$$

The skew diffusive expression for the flux using tensor $\mathbf{A}$ in (6.52) is

$$\mathbf{F}_{\rm skew} = -\mathbf{A}\nabla T = -(\nabla T) \times \Psi = \mathbf{F}_{\rm adv} - \nabla \times (T\Psi).$$

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).

6.3.6 Anisotropic Gent-McWilliams scheme

Using unit vector $\hat{\mathbf{n}} = (n_x, n_y)$ in an arbitrary direction, the two-dimensional anisotropic diffusion tensor is defined as follows:

$$\mathbf{K}_2 = \begin{pmatrix} L & M \\ M & N \end{pmatrix} = \begin{pmatrix} \kappa_A n_x^2 + \kappa_B n_y^2 & \kappa_B n_x n_y \\ \kappa_B n_x n_y & \kappa_B n_x^2 + \kappa_A n_y^2 \end{pmatrix}, \tag{6.53}$$

where κ_A is the diffusivity in the $\hat{\mathbf{n}}$ direction, and κ_B is that in the direction normal to $\hat{\mathbf{n}}$. This is applied to the anti-symmetric tensor in the Gent-McWilliams scheme, and the following expression is obtained (Smith and Gent, 2004),

$$\mathbf{A}' = \begin{pmatrix} 0 & 0 & -LS_x - MS_y \\ 0 & 0 & -MS_x - NS_y \\ LS_x + MS_y & MS_x + NS_y & 0 \end{pmatrix}. \tag{6.54}$$

In the choice of option GMANISTROP, $\hat{\mathbf{n}}$ is set in the direction of the local horizontal velocity. The value of κ_A is read from the namelist njobpi (variable name aitd). The ratio of κ_B/κ_A is read from the namelist nmlgmanisotrop (variable name cscl_isotrop). The default value of cscl_isotrop is set to 1/2.

6.4 Convective adjustment

Convective adjustment is realized by replacing the density (temperature and salinity) that is statically unstable (the upper density exceeds the lower density) in a water column with the averaged density between neighboring levels (neutralization), considering that interior convection occurs in that place. Most of the realistic phenomena represented by the convective adjustment include the developing mixed layer due to surface cooling during winter. Convective adjustment also includes the case in which dense bottom water flows out the sill and flows down along the slope. Moreover, the convective adjustment includes the practical effect that it suppresses disturbances caused by the numerical calculation error and smoothes the distribution.

In general, there are three numerical schemes for convective adjustment. 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 the option DIFAJS. The vertical diffusivity between the unstable grid points is set to 10000 $\mathrm{cm^2\,s^{-1}}$. 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).

6.4.1 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 6.4). The level at the vertical boundary of a T-cell corresponds to the integer k. The index α_k indicates an unstable part within a water column: $\alpha_k = 1$ if it is unstable at the level between $k-\frac{1}{2}$ and $k+\frac{1}{2}$, and $\alpha_k = 0$ if it is neutral or stable. The index β_k memorizes the mixed part: $\beta_k = 1$ at the boundary where it is neutral as a result of mixing, and $\beta_k = 0$ elsewhere. Variables TU_k, SU_k, and VU_k and TL_k, SL_k, and VL_k are temperature, salinity and volume accumulated by multiplying α above the level k and below the level k, respectively, and are expressed by the following recursive relation.

$$
\begin{aligned}
VU_1 &= \alpha_1 V_{\frac{1}{2}}, \\
VU_2 &= \alpha_2(V_{1+\frac{1}{2}} + \alpha_1 V_{\frac{1}{2}}) = \alpha_2(V_{1+\frac{1}{2}} + VU_1), \\
&\cdots, \\
VU_k &= \alpha_k(V_{k-\frac{1}{2}} + VU_{k-1}), \\
&\cdots, \\
VU_{KM-1} &= \alpha_{KM-1}(V_{KM-1-\frac{1}{2}} + VU_{KM-2}),
\end{aligned} \tag{6.55}
$$

and

$$
\begin{aligned}
VL_{KM-1} &= \alpha_{KM-1} V_{KM-\frac{1}{2}}, \\
VL_{KM-2} &= \alpha_{KM-2}(V_{KM-1-\frac{1}{2}} + \alpha_{KM-1} V_{KM-\frac{1}{2}}) = \alpha_{KM-2}(V_{KM-1-\frac{1}{2}} + VL_{KM-1}), \\
&\cdots, \\
VL_k &= \alpha_k(V_{k+\frac{1}{2}} + VL_{k+1}), \\
&\cdots, \\
VL_1 &= \alpha_1(V_{1+\frac{1}{2}} + VL_2),
\end{aligned} \tag{6.56}
$$

where $V_{k+\frac{1}{2}}$ denotes a volume of the cell at the level $k+\frac{1}{2}$. In a similar way, other quantities are expressed as follows:

$$
\begin{aligned}
&TU_1 = \alpha_1 T_{\frac{1}{2}} V_{\frac{1}{2}}, \quad TU_k = \alpha_k(T_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + TU_{k-1}), \\
&SU_1 = \alpha_1 S_{\frac{1}{2}} V_{\frac{1}{2}}, \quad SU_k = \alpha_k(S_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + SU_{k-1}), \\
&TL_{KM-1} = \alpha_{KM-1} T_{KM-\frac{1}{2}} V_{KM-\frac{1}{2}}, \; TL_k = \alpha_k(T_{k+\frac{1}{2}} V_{k+\frac{1}{2}} + TL_{k+1}), \\
&SL_{KM-1} = \alpha_{KM-1} S_{KM-\frac{1}{2}} V_{KM-\frac{1}{2}}, \; SL_k = \alpha_k(S_{k+\frac{1}{2}} V_{k+\frac{1}{2}} + SL_{k+1}),
\end{aligned} \tag{6.57}
$$

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

$$
\begin{aligned}
VU_k + VL_k &= V_{k-\frac{1}{2}} + V_{k+\frac{1}{2}}, \\
TU_k + TL_k &= T_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + T_{k+\frac{1}{2}} V_{k+\frac{1}{2}}, \\
SU_k + SL_k &= S_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + S_{k+\frac{1}{2}} V_{k+\frac{1}{2}},
\end{aligned}
$$

indicating a volume and accumulated temperature and salinity in an unstable part and

$$
\begin{aligned}
TM_{k-\frac{1}{2},k+\frac{1}{2}} &= \frac{TU_k + TL_k}{VU_k + VL_k}, \\
SM_{k-\frac{1}{2},k+\frac{1}{2}} &= \frac{SU_k + SL_k}{VU_k + VL_k},
\end{aligned} \tag{6.58}
$$

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 $\alpha_{k-2} = \alpha_{k+1} = 0$,

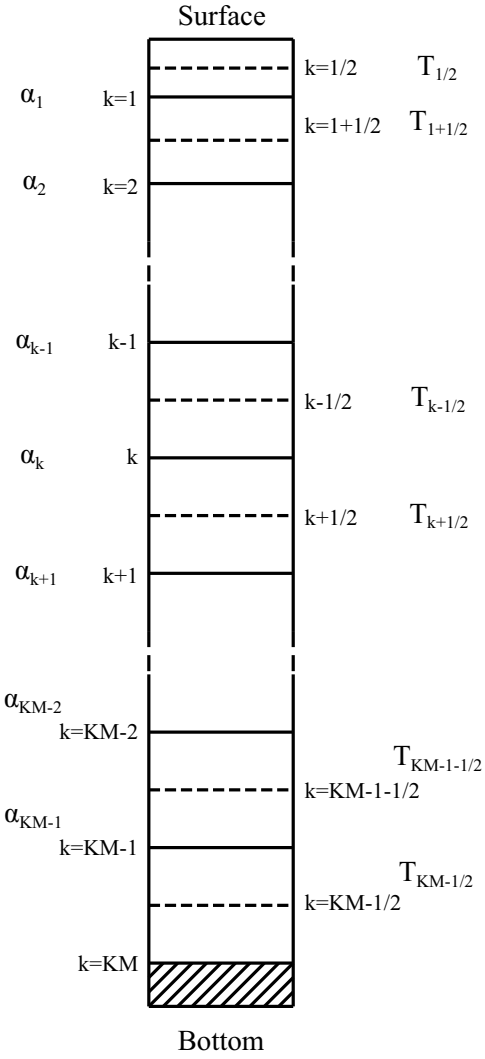

Figure 6.4. Reference vertical grid points in Section 6.4

$$\begin{aligned}
VU_{k-1}+VL_{k-1} &= VU_k+VL_k \\
&= V_{k-1-\frac{1}{2}}+V_{k-\frac{1}{2}}+V_{k+\frac{1}{2}}, \\
TU_{k-1}+TL_{k-1} &= TU_k+TL_k \\
&= T_{k-1-\frac{1}{2}}V_{k-1-\frac{1}{2}}+T_{k-\frac{1}{2}}V_{k-\frac{1}{2}}+T_{k+\frac{1}{2}}V_{k+\frac{1}{2}}, \\
SU_{k-1}+SL_{k-1} &= SU_k+SL_k \\
&= S_{k-1-\frac{1}{2}}V_{k-1-\frac{1}{2}}+S_{k-\frac{1}{2}}V_{k-\frac{1}{2}}+S_{k+\frac{1}{2}}V_{k+\frac{1}{2}},
\end{aligned} \tag{6.59}$$

and

$$\begin{aligned}
TM_{k-1-\frac{1}{2},k+\frac{1}{2}} &= \frac{TU_{k-1}+TL_{k-1}}{VU_{k-1}+VL_{k-1}} = \frac{TU_k+TL_k}{VU_k+VL_k}, \\
SM_{k-1-\frac{1}{2},k+\frac{1}{2}} &= \frac{SU_{k-1}+SL_{k-1}}{VU_{k-1}+VL_{k-1}} = \frac{SU_k+SL_k}{VU_k+VL_k}.
\end{aligned} \tag{6.60}$$

These are averages of the three layer, $k-1-\frac{1}{2}$, $k-\frac{1}{2}$, and $k+\frac{1}{2}$.

6.4.2 Numerical procedure

[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 (6.55) to (6.57), 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 (6.58) and substituted for the original values of T and S. This change modifies the density at the intermediate depth in [1].

[4] The value of α^n is stored in β^n. $\beta^n = 1$ is set if $\alpha^n = 1$, or $\alpha^n = 0$ and $\beta^{n-1} = 1$, and otherwise $\beta^n = 0$. This is presented by the following:

$$\beta_k^n = \alpha_k^n + \beta_k^{n-1}(1-\alpha_k^n). \tag{6.61}$$

[5] The static stability is judged only for $\beta_k^n = 0$. Let $\alpha_k^{n+1} = 1$ if statically unstable, and 0 otherwise. If there is no unstable part, the procedure for that water column is completed.

[6] For a water column which still includes an unstable part, modification for α_k^{n+1} is done using β_k^n by the following. After the procedure [2], any instability will be found only at the bottom of the part that is neutral as a result of prior mixing. In that case, the neutral part must be treated as an unstable part, that is, $\alpha_k^{n+1} = 1$. On the other hand, no more procedure is needed if the upper and lower end is stable, giving $\alpha_k^{n+1} = 0$. This is done by a recursive formula going down and up in the following.

$$\begin{aligned} &\gamma_1 = \alpha_1^{(n+1)},\ \gamma_k = \alpha_k^{(n+1)} + (1-\alpha_k^{(n+1)})\beta_k^{(n)}\gamma_{k-1} \\ &\alpha_{KM-1}^{(n+1)} = \gamma_{KM-1},\ \alpha_k^{(n+1)} = \gamma_k + (1-\gamma_k)\beta_k^{(n)}\alpha_{k+1}^{(n+1)}, \end{aligned} \tag{6.62}$$

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 6.1 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 (6.62), as described in [6]. Note that $\beta_k^0 = 0$, though there is no description in the table.

Table 6.1. Example of the convective adjustment procedure

n	k	α		VU	VL	VU+VL	TU+TL	β
1	1	1	1	$\mathbf{V}_{\frac{1}{2}}$	$\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{T}_{\frac{1}{2}}\mathbf{V}_{\frac{1}{2}}+\mathbf{T}_{1\frac{1}{2}}\mathbf{V}_{1\frac{1}{2}}+\mathbf{T}_{2\frac{1}{2}}\mathbf{V}_{2\frac{1}{2}}$	1
	2	1	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}$	$\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{T}_{\frac{1}{2}}\mathbf{V}_{\frac{1}{2}}+\mathbf{T}_{1\frac{1}{2}}\mathbf{V}_{1\frac{1}{2}}+\mathbf{T}_{2\frac{1}{2}}\mathbf{V}_{2\frac{1}{2}}$	1
	3	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
	5	1	1	$\mathbf{V}_{4\frac{1}{2}}$	$\mathbf{V}_{5\frac{1}{2}}$	$\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\mathbf{T}_{4\frac{1}{2}}\mathbf{V}_{4\frac{1}{2}}+\mathbf{T}_{5\frac{1}{2}}\mathbf{V}_{5\frac{1}{2}}$	1
2	1	0	1	$\mathbf{V}_{\frac{1}{2}}$	$\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\sum_{k=0}^{3}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	2	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}$	$\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\sum_{k=0}^{3}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	3	1	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\sum_{k=0}^{3}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	4	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	1
3	1	0	1	$\mathbf{V}_{\frac{1}{2}}$	$\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	2	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}$	$\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	3	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	4	1	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	5	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}$	$\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1

References

Cox, M. D., 1987: Isopycnal diffusion in a z-coordinate ocean model, *Ocean Modell.*, *74*, 1-5.

Gent, P. R., and J. C. McWilliams, 1990: Isopycnal mixing in ocean circulation models, *J. Phys. Oceanogr.*, *20*, 150-155.

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.

Griffies, S. M., 1998: The Gent-McWilliams skew flux, *J. Phys. Oceanogr.*, *28*, 831-841.

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.*, *28*, 805-830.

Holland, W. R., J. C. Chow, and F. O. Bryan, 1998: Application of a third-order upwind scheme in the NCAR ocean model., *J. Climate*, *11*, 1487–1493.

Ishizaki, H., 1997: A massive treatment scheme of complete convection for ocean models, unpublished manuscript.

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.

Leonard, B. P., M. K. MacVean, and A. P. Lock, 1993: Positivity-Preserving Numerical Schemes for Multidimensional Advection, *NASA Tech. Memo.*, *106055, ICOMP-93-05*, 62pp.

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*, 27pp.

Prather, M. J., 1986: Numerical advection by conservation of second-order moments, *J. Geophys. Res.*, *91*, 6671-6681.

Redi, M. H., 1982: Oceanic isopycnal mixing by coordinate rotation, *J. Phys. Oceanogr.*, *12*, 1154–1158.

Smith, R. D., and P. R. Gent, 2004: Anisotropic GM parameterization for ocean models, *J. Phys. Oceanogr.*, *34*, 2541–2564.

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,doi:10.1029/2002GL015633.

Tsujino, H., H. Hasumi, and N. Suginohara, 2000: Deep pacific circulation controlled by vertical diffusivity at the lower thermocline depth, *J. Phys. Oceanogr.*, *30*, 2853–2865.

Part III

Additional Processes

Chapter 7 Mixed layer model

The surface boundary layer is made turbulent by wind and other factors injecting momentum, so vertical mixing could be induced even if the stratification is stable. However, this phenomenon is not expressed by the fundamental equation of the model.* Therefore, general circulation models express the effects of turbulent phenomena by using synoptic states (variables of velocity and temperature in the model). In MRI.COM, the vertical viscosity and diffusivity are set to large values in the uppermost layer or calculated at each time step by using mixed layer models. Three mixed layer models are supported: the turbulence closure model by Mellor and Yamada (1982) and Mellor and Blumberg (2004) introduced in Section 7.1, the turbulence closure model by Noh and Kim (1999) in Section 7.2, and K-profile parameterization by Large et al. (1994) in Section 7.3. Table 7.1 lists the notation of the vertical viscosity and diffusivity as well as physical properties that are predicted or diagnosed in each mixed layer model.

Table 7.1. Physical properties predicted and diagnosed in each mixed layer model and their notations used for explanation in each section

Physical property (name of variable in MRI.COM)	Mellor and Yamada	Noh and Kim	KPP
coefficient of diffusion (`avdsl`)	K_H	K_H	K_x
coefficient of viscosity (`avm`)	K_M	K_M	K_x
turbulent velocity (`q`: only Mellor-Yamada)	q	-	-
turbulent kinetic energy (`eb`: only Noh and Kim)	$q^2/2$	E	-
vertical turbulent scale (`alo`: except for KPP)	l	l	-

7.1 Mellor and Yamada's Turbulence Closure Model

7.1.1 Turbulence Closure Model

The physical properties in the basic equations of motion for a Boussinesq fluid are separated into averaged components and perturbed components, and then the equations are time averaged. The expressions for averaged velocity U, averaged pressure P, and averaged potential temperature Θ are

$$\frac{\partial U_i}{\partial x_i} = 0, \tag{7.1}$$

$$\frac{DU_j}{Dt} + \varepsilon_{jkl} f_k U_l = \frac{\partial}{\partial x_k}(-\langle u_k u_j \rangle) - \frac{1}{\rho_0}\frac{\partial P}{\partial x_j} - g_j \frac{\rho}{\rho_0} + \frac{\partial}{\partial x_k}\left[2\nu \frac{1}{2}\left(\frac{\partial U_k}{\partial x_j} + \frac{\partial U_j}{\partial x_k}\right)\right], \tag{7.2}$$

$$\frac{D\Theta}{Dt} = \frac{\partial}{\partial x_k}(-\langle u_k \theta \rangle) + \frac{\partial}{\partial x_k}\left(\kappa \frac{\partial \Theta}{\partial x_k}\right), \tag{7.3}$$

*It could be expressed by relaxing the hydrostatic approximation but this is for future work.

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 letters. The statistical averages of the turbulent components are represented by $\langle\,\rangle$. The equation for salinity is similar to that for temperature (7.3). 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).

The evolution of large scale physical quantities could be obtained by calculating the statistically averaged turbulent component ($\langle\,\rangle$) expressed above for each time step, but the calculation is very complicated and unknown higher order terms arise when evolution equations for statistically averaged turbulent components are derived (e.g., to calculate the covariance of turbulent velocities, the equation for a turbulent velocity component is multiplied by another component of turbulent velocity and then statistically averaged). Thus, the expression is not closed, although it should be closed at a certain level (usually called closure). The expression closed in the second order of the turbulent component is named second moment closure and is popular in modeling turbulence. Following Kantha and Clayson (2000), the second moment closure is expressed as

$$\frac{D\langle u_i u_j\rangle}{Dt} + \frac{\partial}{\partial x_k}\Big[\langle u_k u_i u_j\rangle - \nu\frac{\partial\langle u_i u_j\rangle}{\partial x_k}\Big] + \frac{1}{\rho_0}\Big[\frac{\partial\langle p u_i\rangle}{\partial x_j} + \frac{\partial\langle p u_j\rangle}{\partial x_i}\Big] + f_k(\varepsilon_{jkl}\langle u_l u_i\rangle + \varepsilon_{ikl}\langle u_l u_j\rangle)$$
$$= \Big[-\langle u_k u_i\rangle\frac{\partial U_j}{\partial x_k} - \langle u_k u_j\rangle\frac{\partial U_i}{\partial x_k}\Big] - \frac{\beta}{\rho_0}(g_j\langle u_i\theta\rangle + g_i\langle u_j\theta\rangle) + \Big\langle\frac{p}{\rho_0}\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big)\Big\rangle - 2\nu\Big\langle\frac{\partial u_i}{\partial x_k}\frac{\partial u_j}{\partial x_k}\Big\rangle, \quad (7.4)$$

$$\frac{D\langle u_j\theta\rangle}{Dt} + \frac{\partial}{\partial x_k}\Big[\langle u_k u_j\theta\rangle - \kappa\Big\langle u_j\frac{\partial\theta}{\partial x_k}\Big\rangle - \nu\Big\langle\theta\frac{\partial u_j}{\partial x_k}\Big\rangle\Big] + \frac{1}{\rho_0}\Big\langle\theta\frac{\partial p}{\partial x_j}\Big\rangle + \varepsilon_{jkl}f_k\langle u_l\theta\rangle$$
$$= \Big[-\langle u_j u_k\rangle\frac{\partial\Theta}{\partial x_k} - \langle u_k\theta\rangle\frac{\partial U_j}{\partial x_k}\Big] - \frac{\beta}{\rho_0}g_j\langle\theta^2\rangle - (\kappa+\nu)\Big\langle\frac{\partial u_j}{\partial x_k}\frac{\partial\theta}{\partial x_k}\Big\rangle, \quad (7.5)$$

$$\frac{D\langle\theta^2\rangle}{Dt} + \frac{\partial}{\partial x_k}\Big[\langle u_k\theta^2\rangle - \kappa\Big\langle\frac{\partial\theta^2}{\partial x_k}\Big\rangle\Big] = -2\langle u_k\theta\rangle\frac{\partial\Theta}{\partial x_k} - 2\kappa\Big\langle\frac{\partial\theta}{\partial x_k}\frac{\partial\theta}{\partial x_k}\Big\rangle. \quad (7.6)$$

Mellor and Yamada (1982) reduced the higher order terms as follows. It should be noted that it is not unique.

Based on Rotta's (1951a,b) hypothesis of energy redistribution, the covariances of pressure and velocity gradients are assumed to be linear functions of Reynolds stress:

$$\Big\langle\frac{p}{\rho_0}\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big)\Big\rangle = -\frac{q}{3l_1}\Big(\langle u_i u_j\rangle - \frac{\delta_{ij}}{3}q^2\Big) + C_1 q^2\Big(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\Big), \quad (7.7)$$

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 Kolmogolov's hypothesis of local isotropy in small eddies the energy dissipation is modeled as follows,

$$2\nu\Big\langle\frac{\partial u_i}{\partial x_k}\frac{\partial u_j}{\partial x_k}\Big\rangle = -\frac{2}{3}\frac{q^3}{\Lambda_1}\delta_{ij}, \quad (7.8)$$

where Λ_1 is the length scale.

The redistribution of temperature and the dissipation of heat are modeled as follows in the same form as above,

$$\Big\langle\frac{p}{\rho_0}\frac{\partial\theta}{\partial x_j}\Big\rangle = -\frac{q}{3l_2}\langle u_j\theta\rangle \quad (7.9)$$

$$(\kappa+\nu)\Big\langle\frac{\partial u_j}{\partial x_k}\frac{\partial\theta}{\partial x_k}\Big\rangle = 0, \quad (7.10)$$

where l_2 is the length scale. The dissipation of temperature variance is

$$2\kappa\Big\langle \frac{\partial\theta}{\partial x_k}\frac{\partial\theta}{\partial x_k}\Big\rangle = -2\frac{q}{\Lambda_2}\langle\theta^2\rangle, \tag{7.11}$$

where Λ_2 is the length scale.

In order to avoid the higher order problems, the turbulent velocity diffusion term and the other higher order terms are modeled as follows:

$$\langle u_k u_i u_j\rangle = \frac{3}{5}lqS_q\Big(\frac{\partial\langle u_i u_j\rangle}{\partial x_k}+\frac{\partial\langle u_i u_k\rangle}{\partial x_j}+\frac{\partial\langle u_j u_k\rangle}{\partial x_i}\Big), \tag{7.12}$$

$$\langle u_k u_j\theta\rangle = -lqS_{u\theta}\Big(\frac{\partial\langle u_k\theta\rangle}{\partial x_j}+\frac{\partial\langle u_j\theta\rangle}{\partial x_k}\Big), \tag{7.13}$$

$$\langle u_k\theta^2\rangle = -lqS_\theta\frac{\partial\langle\theta^2\rangle}{\partial x_k}, \tag{7.14}$$

where S_q, $S_{u\theta}$, and S_θ are non-dimensional numbers and can be set as constants or functions of certain parameters. Other relations are $\langle p\theta\rangle = 0$ and $\langle pu_i\rangle = 0$.

The essence of the Mellor-Yamada mixed layer model is that the above length scales are related linearly to each other:

$$(l_1,\Lambda_1,l_2,\Lambda_2) = (A_1,B_1,A_2,B_2)l, \tag{7.15}$$

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 $(A_1,B_1,A_2,B_2,C_1) = (0.92,16.6,0.74,10.1,0.08)$.

7.1.2 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 (7.33) 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 is employed as the surface boundary layer model by MRI.COM and 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 physical quantity become

$$\frac{DU}{Dt}+\frac{\partial}{\partial z}\langle uw\rangle = -\frac{1}{\rho_0}\frac{\partial P}{\partial x}+fV, \tag{7.16}$$

$$\frac{DV}{Dt}+\frac{\partial}{\partial z}\langle vw\rangle = -\frac{1}{\rho_0}\frac{\partial P}{\partial y}-fU, \tag{7.17}$$

$$0 = -\frac{1}{\rho_0}\frac{\partial P}{\partial z}-g\frac{\rho}{\rho_0}, \tag{7.18}$$

$$\frac{D\Theta}{Dt}+\frac{\partial}{\partial z}(\langle w\theta\rangle) = 0. \tag{7.19}$$

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

$$\frac{D}{Dt}\Big(\frac{q^2}{2}\Big)-\frac{\partial}{\partial z}\Big[lqS_q\frac{\partial}{\partial z}\Big(\frac{q^2}{2}\Big)\Big]=P_s+P_b-\varepsilon, \tag{7.20}$$

where

$$P_s=-\langle wu\rangle\frac{\partial U}{\partial z}-\langle wv\rangle\frac{\partial V}{\partial z} \tag{7.21}$$

is the term for energy produced by the vertical shear of the averaged flow,

$$P_b=-g\langle w\rho\rangle/\rho_0 \tag{7.22}$$

is the term for energy produced by buoyancy, and

$$\varepsilon=q^3/\Lambda_1 \tag{7.23}$$

is the energy dissipation term.

The algebraic equations for the statistically averaged values, which are expressed by other second-moment turbulent quantities, are given below.

$$\langle u^2\rangle=\frac{q^2}{3}+\frac{l_1}{q}\Big[-4\langle wu\rangle\frac{\partial U}{\partial z}+2\langle wv\rangle\frac{\partial V}{\partial z}-2P_b\Big], \tag{7.24}$$

$$\langle v^2\rangle=\frac{q^2}{3}+\frac{l_1}{q}\Big[2\langle wu\rangle\frac{\partial U}{\partial z}-4\langle wv\rangle\frac{\partial V}{\partial z}-2P_b\Big], \tag{7.25}$$

$$\langle w^2\rangle=\frac{q^2}{3}+\frac{l_1}{q}\Big[2\langle wu\rangle\frac{\partial U}{\partial z}+2\langle wv\rangle\frac{\partial V}{\partial z}+4P_b\Big], \tag{7.26}$$

$$\langle uv\rangle=\frac{3l_1}{q}\Big[-\langle uw\rangle\frac{\partial V}{\partial z}-\langle vw\rangle\frac{\partial U}{\partial z}\Big], \tag{7.27}$$

$$\langle wu\rangle=\frac{3l_1}{q}\Big[-(\langle w^2\rangle-C_1q^2)\frac{\partial U}{\partial z}-g\langle u\rho\rangle\Big], \tag{7.28}$$

$$\langle vw\rangle=\frac{3l_1}{q}\Big[-(\langle w^2\rangle-C_1q^2)\frac{\partial V}{\partial z}-g\langle v\rho\rangle\Big], \tag{7.29}$$

$$\langle u\theta\rangle=\frac{3l_2}{q}\Big[-\langle uw\rangle\frac{\partial\Theta}{\partial z}-\langle w\theta\rangle\frac{\partial U}{\partial z}\Big], \tag{7.30}$$

$$\langle v\theta\rangle=\frac{3l_2}{q}\Big[-\langle vw\rangle\frac{\partial\Theta}{\partial z}-\langle w\theta\rangle\frac{\partial V}{\partial z}\Big], \tag{7.31}$$

$$\langle w\theta\rangle=\frac{3l_2}{q}\Big[-\langle w^2\rangle\frac{\partial\Theta}{\partial z}-g\langle\theta\rho\rangle\Big], \tag{7.32}$$

$$\langle\theta^2\rangle=-\frac{\Lambda_2}{q}\langle w\theta\rangle\frac{\partial\Theta}{\partial z}. \tag{7.33}$$

Some of the terms in these equations can be further simplified as follows:

$$-\langle uw\rangle = K_M \frac{\partial U}{\partial z}, \tag{7.34}$$

$$-\langle vw\rangle = K_M \frac{\partial V}{\partial z}, \tag{7.35}$$

$$-\langle \theta w\rangle = K_H \frac{\partial \Theta}{\partial z}, \tag{7.36}$$

$$K_M = lqS_M, \tag{7.37}$$

$$K_H = lqS_H. \tag{7.38}$$

This simplification means that the vertical turbulent fluxes are proportional to the gradient of the large scale values. 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 (7.37) and (7.38).

Assuming that the potential density is linearly related to the potential temperature (and salinity), the simultaneous equations for S_M and S_H are derived as follows:

$$\begin{aligned} S_M[6A_1A_2G_M] + S_H[1 - 3A_2B_2G_H - 12A_1A_2G_H] &= A_2, \\ S_M[1 + 6A_1^2G_M - 9A_1A_2G_H] - S_H[12A_1^2G_H + 9A_1A_2G_H] &= A_1(1 - 3C_1), \end{aligned} \tag{7.39}$$

where

$$G_M \equiv \frac{l^2}{q^2}\left[\left(\frac{\partial U}{\partial z}\right)^2 + \left(\frac{\partial V}{\partial z}\right)^2\right], \tag{7.40}$$

$$G_H \equiv \frac{l^2}{q^2}\frac{g}{\rho_0}\frac{\partial \tilde{\rho}}{\partial z}, \tag{7.41}$$

and $\partial\tilde{\rho}/\partial z$ is the vertical gradient of potential density.

Using S_M and S_H, K_M and K_H are then obtained from (7.37) and (7.38) by determining q and l.

The turbulent velocity q is obtained by solving the following expression that is modified from (7.20) using the above results:

$$\frac{\partial}{\partial t}\left(\frac{q^2}{2}\right) - \frac{\partial}{\partial z}\left[K_E\frac{\partial}{\partial z}\left(\frac{q^2}{2}\right)\right] = K_M\left[\left(\frac{\partial U}{\partial z}\right)^2 + \left(\frac{\partial V}{\partial z}\right)^2\right] + \frac{g}{\rho_0}K_H\frac{\partial\tilde{\rho}}{\partial z} - \varepsilon, \tag{7.42}$$

where

$$K_E = lqS_q \tag{7.43}$$

and advection terms are neglected. In MRI.COM, S_q is set proportional to S_M ($S_q \propto S_M$). We adopt the form $S_q = S_{qc}S_M/S_{Mn}$, where $S_{qc} = 0.2$ and $S_{Mn} = 0.3927$. With this choice, $S_q = 0.2$ when the stratification is neutral ($G_H = 0$).

The sea surface boundary condition for the turbulent kinetic energy follows Mellor and Blumberg (2004):

$$K_q\frac{\partial q^2}{\partial z} = 2\alpha_{\mathrm{CB}}u_\tau^3, \tag{7.44}$$

where $\alpha_{\mathrm{CB}} = 100$ and u_τ is the frictional velocity defined as $u_\tau \equiv (\tau_s/\rho_s)^{1/2}$ by using the surface stress (τ_s) and the sea surface density (ρ_s). This flux boundary condition is analytically converted to the condition for q at the sea surface:

$$q^2 = (15.8\alpha_{\mathrm{CB}})^{2/3}u_\tau^2. \tag{7.45}$$

The vertical scale of the turbulence (master length scale) 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. The formula used in MRI.COM is a diagnosis for cases without surface wind wave effects:

$$l = \gamma \int_{z_b}^{0} |z'| q dz' / \int_{z_b}^{0} q dz', \tag{7.46}$$

where $\gamma = 0.2$, and z_b is the depth of the bottom. 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). Roughness parameter $z_{\rm w}$ due to surface wind waves is given by Mellor and Blumberg (2004) as follows:

$$z_w = \beta_{\rm w} \frac{u_\tau^2}{g}, \quad \beta_{\rm w} = 2.0 \times 10^5. \tag{7.47}$$

For a depth of $|z| < z_w$,

$$l = \max(\kappa z_w, \kappa z), \tag{7.48}$$

where κ is the von Karman constant ($\kappa = 0.4$). For depths exceeding z_w, (7.46) is employed.

7.1.3 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 (7.46), (7.47), and (7.48), the turbulent kinetic energy ($q^2/2$) is solved using (7.42), (7.15), (7.23), and (7.45), 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 12.5). The vertical viscosity and diffusivity for the time step n+1 are estimated using q, l, and (7.37) to (7.39). The vertical scale of the turbulence (master length scale) based on (7.46) 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 grid cell (i, j, k). The specific expression for the discretized form of the turbulent kinetic energy ($E = q^2/2$) equation is as follows:

$$\begin{aligned}
\frac{E_k^{n+1} - E_k^n}{\Delta t} =& \frac{1}{\Delta z_k} \Bigg[\frac{K_{E_{k-\frac{1}{2}}} (E_{k-1}^{n+1} - E_k^{n+1})}{\Delta z_{k-\frac{1}{2}}} - \frac{K_{E_{k+\frac{1}{2}}} (E_k^{n+1} - E_{k+1}^{n+1})}{\Delta z_{k+\frac{1}{2}}} \Bigg] \\
&+ K_{Mk} \frac{(U_{k-\frac{1}{2}}^{n+1} - U_{k+\frac{1}{2}}^{n+1})(\tilde{U}_{k-\frac{1}{2}} - \tilde{U}_{k+\frac{1}{2}})}{\Delta z_k^2} + K_{Mk} \frac{(V_{k-\frac{1}{2}}^{n+1} - V_{k+\frac{1}{2}}^{n+1})(\tilde{V}_{k-\frac{1}{2}} - \tilde{V}_{k+\frac{1}{2}})}{\Delta z_k^2} \\
&- K_{Hk} \frac{B_{k-\frac{1}{2}}^{n+1} - B_{k+\frac{1}{2}}^{n+1}}{\Delta z_k} - 2E_k^{n+1} q_k^n / B_1 l_k^n,
\end{aligned} \tag{7.49}$$

where $\tilde{u} = (u^{n+1} + u^n)/2$ 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 (7.46), (7.47), and (7.48).
2. Update the turbulent kinetic energy using (7.49).

3. Solve the algebraic equation for S_M and S_H using (7.39).
4. Calculate the vertical viscosity and diffusivity for the next time step using (7.37), (7.38), and (7.43).
5. Calculate the master length scale using (7.46) for the next time step.

7.2 Turbulent mixed layer model by Noh and Kim (1999)

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 that by Mellor and Yamada and is classified as a second moment closure model.

7.2.1 Fundamental equation

The equations for the zonal and meridional components of the velocity, U, V, buoyancy $B = -g\Delta\rho/\rho_o$, and turbulent energy E in the large scale fields are

$$\frac{DU}{Dt} = -\frac{\partial}{\partial z}\langle uw\rangle - \frac{1}{\rho_0}\frac{\partial P}{\partial x} + fV, \tag{7.50}$$

$$\frac{DV}{Dt} = -\frac{\partial}{\partial z}\langle vw\rangle - \frac{1}{\rho_0}\frac{\partial P}{\partial y} - fU, \tag{7.51}$$

$$\frac{DB}{Dt} = -\frac{\partial}{\partial z}\langle bw\rangle + \frac{\partial R}{\partial z}, \tag{7.52}$$

$$\frac{DE}{Dt} = -\frac{\partial}{\partial z}\Big\langle w\Big(\frac{p}{\rho_0} + uu + vv + ww\Big)\Big\rangle - \langle uw\rangle\frac{\partial U}{\partial z} - \langle vw\rangle\frac{\partial V}{\partial z} + \langle bw\rangle - \varepsilon, \tag{7.53}$$

where R is the downward short wave 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,

$$\frac{DU}{Dt} = \frac{\partial}{\partial z}\Big(K_M\frac{\partial U}{\partial z}\Big) - \frac{1}{\rho_0}\frac{\partial P}{\partial x} + fV, \tag{7.54}$$

$$\frac{DV}{Dt} = \frac{\partial}{\partial z}\Big(K_M\frac{\partial V}{\partial z}\Big) - \frac{1}{\rho_0}\frac{\partial P}{\partial y} - fU, \tag{7.55}$$

$$\frac{DB}{Dt} = \frac{\partial}{\partial z}\Big(K_H\frac{\partial B}{\partial z}\Big) - \frac{\partial R}{\partial z}, \tag{7.56}$$

$$\frac{DE}{Dt} = \frac{\partial}{\partial z}\Big(K_E\frac{\partial E}{\partial z}\Big) + K_M\frac{\partial U}{\partial z}\frac{\partial U}{\partial z} + K_M\frac{\partial V}{\partial z}\frac{\partial V}{\partial z} - \Big(K_H\frac{\partial B}{\partial z}\Big) - \varepsilon. \tag{7.57}$$

The central problem is how to determine the viscosity, diffusivity (K_M, K_H, K_E), and turbulent energy dissipation rate (ε). By using the typical velocity scale ($q = (2E)^{1/2}$) and the vertical length scale (l) of the turbulence, we assume the following

$$K_M = Sql, \tag{7.58}$$

$$K_H = S_B ql, \tag{7.59}$$

$$K_E = S_E ql, \tag{7.60}$$

$$\varepsilon = Cq^3 l^{-1}. \tag{7.61}$$

The constants (S, S_B, S_E, C) are obtained from experiments and it is assumed that $S = S_0 = 0.39$, $Pr = Pr_0 = S/S_B = 0.8$, $\sigma = S/S_E = 1.95$, and $C = C_0 = 0.06$ for neutral stratification.

As for the influence of the stratification, 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,

$$K \sim q l_b \sim q l \mathrm{Ri}_t{}^{-1/2}, \tag{7.62}$$

where Ri_t is the turbulent Richardson number

$$\mathrm{Ri}_t = (Nl/q)^2. \tag{7.63}$$

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 since the internal waves are induced by the injected turbulent energy, resulting in their horizontal propagation. It could also be considered that the local turbulent energy dissipation becomes large.

The following equation is used for S so that it satisfies (7.62) when Ri_t is large:

$$S/S_0 = (1 + \alpha \mathrm{Ri}_t)^{-1/2}, \tag{7.64}$$

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 energy dissipation (C) is set as follows:

$$C/C_0 = (1 + \alpha \mathrm{Ri}_t)^{1/2}. \tag{7.65}$$

The effect of stratification on the Prandtl number (Pr) is set following Noh et al. (2005):

$$Pr/Pr_0 = (1 + \beta \mathrm{Ri}_t)^{1/2}, \tag{7.66}$$

where β is a tuning parameter and 0.5 is used following Noh et al. (2005).

The vertical scale of turbulence is given by

$$l = \frac{\kappa(|z| + z_0)}{(1 + \kappa(|z| + z_0)/h)}, \tag{7.67}$$

where z_0 is the sea surface roughness ($z_0 = 1$[m]), z is the depth, and h is the mixed layer depth. The vertical scale becomes longer as the mixed layer becomes deeper.

The boundary conditions are as follows:

$$K_M \frac{\partial U}{\partial z} = \frac{\tau}{\rho_0}, \tag{7.68}$$

$$K_H \frac{\partial B}{\partial z} = Q_0, \tag{7.69}$$

$$K_E \frac{\partial E}{\partial z} = m u_*^3, \tag{7.70}$$

where m is a tuning parameter and $m = 100$ is recommended by Noh and Kim (1999).

In the case of unstable stratification ($N^2 < 0$), $K_M = K_H = 1.0[\mathrm{m^2 s^{-1}}]$ and K_E is estimated from the turbulent velocity scale and the vertical length scale in the model. This treatment is due to the difference between the time scales of the vertical convection and the development of turbulence.

7.2.2 Implementation

Equation (7.57) 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 12.5). 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 grid cell $(i, j, k-\frac{1}{2})$. The specific expression for the discretized form of the turbulent kinetic energy (E) equation is as follows:

$$\begin{aligned}
\frac{E^{n+1}_{k+\frac{1}{2}} - E^{n}_{k+\frac{1}{2}}}{\Delta t} =& \frac{1}{\Delta z_{k+\frac{1}{2}}}\left[\frac{K_{Ek}(E^{n+1}_{k-\frac{1}{2}} - E^{n+1}_{k+\frac{1}{2}})}{\Delta z_k} - \frac{K_{Ek+1}(E^{n+1}_{k+\frac{1}{2}} - E^{n+1}_{k+\frac{3}{2}})}{\Delta z_{k+1}}\right] \\
&+ \frac{1}{2}K_{Mk}\frac{(u^{n+1}_{k-\frac{1}{2}} - u^{n+1}_{k+\frac{1}{2}})(u^{n+1}_{k-\frac{1}{2}} - u^{n}_{k+\frac{1}{2}})}{\Delta z_k \Delta z_{k+\frac{1}{2}}} + \frac{1}{2}K_{Mk+1}\frac{(u^{n+1}_{k+\frac{1}{2}} - u^{n+1}_{k+\frac{3}{2}})(u^{n}_{k+\frac{1}{2}} - u^{n+1}_{k+\frac{3}{2}})}{\Delta z_{k+1} \Delta z_{k+\frac{1}{2}}} \\
&+ \frac{1}{2}K_{Mk}\frac{(v^{n+1}_{k-\frac{1}{2}} - v^{n+1}_{k+\frac{1}{2}})(v^{n+1}_{k-\frac{1}{2}} - v^{n}_{k+\frac{1}{2}})}{\Delta z_k \Delta z_{k+\frac{1}{2}}} + \frac{1}{2}K_{Mk+1}\frac{(v^{n+1}_{k+\frac{1}{2}} - v^{n+1}_{k+\frac{3}{2}})(v^{n}_{k+\frac{1}{2}} - v^{n+1}_{k+\frac{3}{2}})}{\Delta z_{k+1} \Delta z_{k+\frac{1}{2}}} \\
&- \frac{1}{2}K_{Hk}\frac{B^{n+1}_{k-\frac{1}{2}} - B^{n+1}_{k+\frac{1}{2}}}{\Delta z_k} - \frac{1}{2}K_{Hk+1}\frac{B^{n+1}_{k+\frac{1}{2}} - B^{n+1}_{k+\frac{3}{2}}}{\Delta z_{k+1}} - 2CE^{n+1}_{k+\frac{1}{2}} q^{n}_{k+\frac{1}{2}} / l^{n}_{k+\frac{1}{2}}.
\end{aligned} \tag{7.71}$$

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 (7.67),
2. Update the turbulent kinetic energy using (7.71),
3. Calculate the vertical viscosity and diffusivity for the next time step.

7.3 K Profile Parameterization (KPP)

7.3.1 Outline

The K-profile parameterization (KPP) (Figure 7.1, Equation 7.79) is a method to determine the coefficients of vertical viscosity and diffusivity. First, the turbulent vertical velocity scale in the mixed layer (the boundary layer) is determined following the Monin-Obukhov similarity law near the boundary. This scale is then multiplied by the mixed layer thickness and the non-dimensional profile function that are separately obtained to produce the vertical viscosity and diffusivity. The coefficient ν_x below the mixed layer is set to different values from those within the mixed layer and connected continuously. In this way, KPP differs from a series of turbulent closure model represented in the previous subsections. The KPP scheme is an application of the nonlocal K-profile model (Treon and Mahrt, 1986) used in the atmospheric model to the ocean model by Large et al. (1994). The KPP subroutine in MRI.COM is based on that of the NCEP ocean model (NCOM). The paramterization for mixing due to salt fingering is not adopted and a vertically one-dimensional background profile is given at `vdbg.F90` by user (see Chapter 16).

The time tendency of an averaged value X due to the turbulent eddy is expressed by the vertical divergence of turbulent flux $\langle wx \rangle$,

$$\partial_t X = -\partial_z \langle wx \rangle, \tag{7.72}$$

where X represents the time averaged components of the velocity component U, V, temperature T, salinity S, buoyancy B, and so on, and x represents the turbulent components of the velocity component u, v, temperature t, salinity s, buoyancy b, and so on. Furthermore, w is the vertical velocity due to the turbulent eddy (positive upward). Hereinafter, the momentum component is expressed as m and the scalar property as s in some cases.

In the KPP scheme, the turbulent flux within the mixed layer is expressed by the vertical gradient term of X and the nonlocal transport term.[†] That is

$$\langle wx \rangle = -K_x(\partial_z X - \gamma_x). \tag{7.73}$$

The vertical viscosity and diffusivity coefficients K_x and the nonlocal transport γ_x are calculated in the following order by the KPP subroutine of MRI.COM.

- Calculation of the sea surface fluxes (momentum and buoyancy) $\langle wx_0 \rangle$
- Calculation of the stability scale L
- Calculation of the mixed layer thickness h
- Calculation of the non-dimensional universal function ϕ_x
- Calculation of the turbulent vertical velocity scale w_x
- Calculation of the vertical viscosity and diffusivity coefficient within the mixed layer K_x
- Connection of the coefficient in the mixed layer to the coefficient ν_x below the mixed layer base K_x^*
- Calculation of the vertical viscosity and diffusivity coefficient due to shear instability ν_x^s
- Adopt the larger value of K_x^* and ν_x^s for the vertical viscosity and diffusivity coefficient
- Calculation of the nonlocal transport γ_x

7.3.2 Monin-Obukhov similarity law

The Monin-Obukhov similarity law applies to the boundary layer near the sea surface. In this boundary layer, only the distance from the sea surface $d(=-z)$ and the sea surface flux $\langle wx_0 \rangle$ are important. From these parameters, the following three basic turbulent parameters are determined,

- Frictional velocity:

$$u^{*2} = (\langle wu_0 \rangle^2 + \langle wv_0 \rangle^2)^{1/2} = |\vec{\tau}_0|/\rho_0, \tag{7.74}$$

- Turbulent scale of scalar (temperature and salinity):

$$s^* = -\langle ws_0 \rangle / u^*, \tag{7.75}$$

[†]Nonlocal means the phenomenon in which the material is transported upwards (that is opposite to the gradient of the averaged fields) due to the gradient induced by the turbulent component $\partial x/\partial z$ even though the gradient of the averaged fields $\partial X/\partial z$ is locally positive. See Figure 7.1(a).

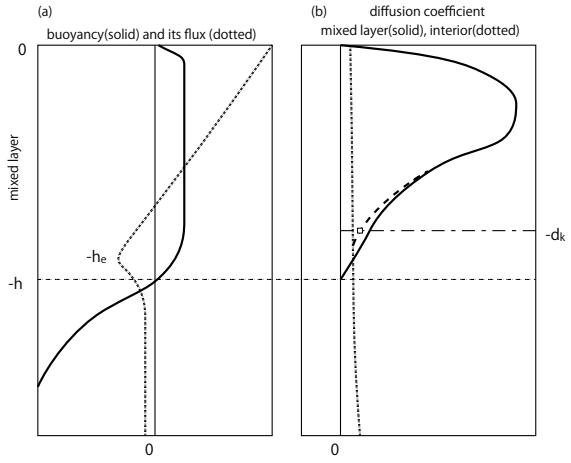

Figure 7.1. Schematic diagram of KPP. (a) Relative buoyancy (solid line) and buoyancy flux (dotted line) profiles. Upward buoyancy flux (non-local transport; Section 7.3.7) could occur in the boundary layer even if the stratification is neutral. Also indicated are the entrainment depth (h_e) and mixed layer base (h). (b) The boundary layer diffusivity profile K_x (solid line; equation (7.79), and the interior diffusivity profile (dotted line). The diffusivity at depth (d_k) immediately above the mixed layer base (h) is set to the value marked by □, and the boundary layer profile and the interior profile are connected by the dashed line (see Section 7.3.4).

- Monin-Obukhov stability scale:

$$L = u^{*3}/(\kappa B_f), \tag{7.76}$$

where $\vec{\tau}_0$ is the sea surface wind stress, ρ_0 is density, $\kappa = 0.4$ is the von Karman constant, and B_f is the surface buoyancy flux (positive downward and thus negative means unstable). It should be noted that the direction of B_f is opposite to that within the ocean. Note also that the fluxes need not be constant in the boundary layer ($d < \varepsilon h$ [$\varepsilon \ll 1$, in general $\varepsilon \sim 0.1$]), but are affected by the sea surface flux $\langle wx_0 \rangle$ and derived properties u^*, s^*, L. In this case, the non-dimensional profiles of velocity and scalar are defined as functions of stability parameter $\zeta = d/L$:

$$\phi_m(\zeta) = \frac{\kappa d}{u^*}\partial_z(U^2+V^2)^{1/2}, \tag{7.77}$$

$$\phi_s(\zeta) = \frac{\kappa d}{s^*}\partial_z S. \tag{7.78}$$

These functions are determined empirically based on observations.

7.3.3 Coefficients of vertical viscosity and diffusivity

The profile of vertical viscosity and diffusivity K_x in the mixed layer is defined as the product of the turbulent vertical velocity scale w_x and the non-dimensional vertical shape function $G(\sigma)$. Since the mixing becomes more effective due to the turbulent eddies as the mixed layer becomes thicker, the coefficient K_x is set linearly proportional to h:

$$K_x(\sigma) = h w_x(\sigma) G(\sigma), \tag{7.79}$$

where $\sigma = d/h$ (fractional depth within the mixed layer) is the non-dimensional vertical coordinate. The vertical shape $G(\sigma)$ is approximated by a third order polynomial (O'Brien, 1970),

$$G(\sigma) = a_0 + a_1\sigma + a_2\sigma^2 + a_3\sigma^3. \tag{7.80}$$

Since the turbulent eddy does not cross the sea surface, $K_x = 0$ at $\sigma = 0$. Thus, $a_0 = 0$ in equation (7.80).

The Monin-Obukhov similarity law is applied in the boundary layer ($\sigma < \varepsilon[= 0.1]$). Assuming that the turbulent flux $\langle wx \rangle$ is linear (Lumley and Panofski, 1964; Tennekes, 1973), we have

$$w_x(\sigma)(a_1 + a_2\sigma) = \frac{\kappa u^*}{\phi_x(\zeta)} \frac{\langle wx(d) \rangle}{\langle wx_0 \rangle} \tag{7.81}$$

from (7.73) with $\gamma_x = 0$, (7.78), and (7.79). Equation (7.81) holds if

$$w_x(\sigma) = \frac{\kappa u^*}{\phi_x(\zeta)}. \tag{7.82}$$

To avoid w_x becoming too large, w_x is set to be constant below the boundary layer ($\sigma = \varepsilon(\sim 0.1)$) under the unstable condition ($\zeta(= d/L) < 0$),

$$\begin{aligned} w_x(\sigma) &= \frac{\kappa u^*}{\phi_x(\varepsilon h/L)} \quad \varepsilon < \sigma < 1 \quad \zeta < 0, \\ w_x(\sigma) &= \frac{\kappa u^*}{\phi_x(\sigma h/L)} \quad \text{otherwise.} \end{aligned} \tag{7.83}$$

The non-dimensional profile function ϕ_x is determined as a function of the stability parameter $\zeta(= d/L)$ based on experiments. They are determined so that w_x could be scaled by κu^* for a neutrally stable case ($h/L \leq 0$), and so that w_x could be larger (smaller) than κu^* for an unstable (stable) case. Large et al. (1994) employ the following expressions.

$$\begin{aligned} \phi_m &= \phi_s = 1 + 5\zeta \qquad 0 \leq \zeta, \\ \phi_m &= (1 - 16\zeta)^{-1/4} \qquad \zeta_m \leq \zeta < 0, \\ \phi_m &= (a_m - c_m\zeta)^{-1/3} \qquad \zeta < \zeta_m, \\ \phi_s &= (1 - 16\zeta)^{-1/2} \qquad \zeta_s \leq \zeta < 0, \\ \phi_s &= (a_s - c_s\zeta)^{-1/3} \qquad \zeta < \zeta_s. \end{aligned} \tag{7.84}$$

Here, $(\zeta_s, c_s, a_s, \zeta_m, c_m, a_m) = (-1.0, 98.96, -28.86, -0.2, 8.38, 1.26)$.

Assuming a linear profile for the turbulent flux $\langle wx \rangle$ (flux decreases linearly from the sea surface value), equation (7.81) is

$$\langle wx(\sigma) \rangle / \langle wx_0 \rangle = 1 - \beta_r\sigma/\varepsilon = a_1 + a_2\sigma. \tag{7.85}$$

From this, we have $a_1 = 1$ and $a_2 = -\beta_r/\varepsilon$. Using the condition at the mixed layer base, $G(1) = \partial_\sigma G(1) = 0$, and assuming $\varepsilon = 0.1$, we have $a_2 = -2$, $a_3 = 1$, and $\beta_r = 0.2$.

7.3.4 Coefficients of vertical viscosity and diffusion at the base of the mixed layer

In the KPP scheme, viscosity and diffusion coefficients in the mixed layer (K_x) are connected to the background profile (ν_x) at the base of the mixed layer.

The coefficient at the boundary (level k) is determined by using the following equations:

$$\delta = (h - d_{k-\frac{1}{2}})/(d_{k+\frac{1}{2}} - d_{k-\frac{1}{2}}) \quad d_{k-\frac{1}{2}} < h < d_{k+\frac{1}{2}}, \tag{7.86}$$

$$K_x^* = (1-\delta)^2 K_x(d_{k-\frac{1}{2}}) + \delta^2 \nu_x(d_k) \quad d_{k-\frac{1}{2}} < h \leq d_k, \tag{7.87}$$

$$K_x^* = (1-\delta)^2 K_x(d_{k-\frac{1}{2}}) + \delta^2 K_x(d_k) \quad d_k < h \leq d_{k+\frac{1}{2}}, \tag{7.88}$$

$$\Lambda_x = (1-\delta)\nu_x(d_{k-\frac{1}{2}}) + \delta K_x^*. \tag{7.89}$$

The vertical viscosity and diffusion coefficients at depth d_k that corresponds to the boundary between the mixed layer base (h) and the interior region is presented by Λ_x ($\Box$ in Figure 7.1(b)). The coefficient K_x^* is set to yield large coefficients there. When $d_{k-\frac{1}{2}} < h < d_k$ (when the mixed layer is shallower than d_k), $K_x(d_k)$ that appears in (7.88) is not defined and is replaced by the background value $\nu_x(d_k)$.

7.3.5 Thickness of the mixed layer

The thickness h of the mixed layer is estimated from the vertical profiles of buoyancy $B(d)$ and velocity $\vec{V}(d)$. Here, it is determined as the depth where the bulk Richardson number referred to the sea surface,

$$Ri_b(d) = \frac{(B_r - B(d))d}{|\vec{V}_r - \vec{V}(d)|^2 + V_t^2(d)}, \tag{7.90}$$

equals the reference value Ri_c (0.3 in MRI.COM), where B_r and $\vec{V}_r$ are the buoyancy and the velocity in the uppermost layer. The quantity V_t/d is called the turbulent velocity shear and is expressed as follows:

$$V_t^2(d) = \frac{C_v(-\beta_T)^{1/2}}{Ri_c\kappa^2}(c_s\varepsilon)^{-1/2}dNw_s, \tag{7.91}$$

where $C_v = 1.8$ and $\beta_T = -0.2$. The turbulent velocity shear is introduced to achieve entrainment at the mixed layer base under strong stratification.

7.3.6 Mixing due to shear instability

In the stratified interior, mixing could occur locally due to shear instabilities. The tendency for shear instability is measured by the local gradient Richardson number,

$$Ri_g = \frac{N^2}{(\partial_z U)^2 + (\partial_z V)^2}. \tag{7.92}$$

When Ri_g is below a critical value Ri_0, the vertical velocity shear overcomes the stabilizing effect of the buoyancy gradient. This process is parameterized using a vertical mixing coefficient ν_x^s given by the following formula and is the same for viscosity and diffusivity.

$$\nu_x^s/\nu^0 = 1 \quad Ri_g < 0 \tag{7.93}$$

$$\nu_x^s/\nu^0 = [1 - (Ri_g/Ri_0)^2]^{p_1} \quad 0 < Ri_g < Ri_0 \tag{7.94}$$

$$\nu_x^s/\nu^0 = 0 \quad Ri_0 < Ri_g \tag{7.95}$$

Here, $\nu^0 = 50 \times 10^{-4}$ [$\mathrm{m^2\,s^{-1}}$], $Ri_0 = 0.7$, and $p_1 = 3$.

The final vertical mixing coefficient for "this" time step is taken as the larger of this shear instability mixing coefficient and the one obtained by connecting boundary layer and background interior (Section 7.3.4).

7.3.7 Nonlocal Transport

If the stratification is locally stable or neutral, the turbulent buoyancy flux should be downward or zero according to the r.h.s. of (7.73 [$\gamma_x = 0$]). However, an upward turbulent buoyancy flux exists if the sea surface buoyancy flux is unstable (upward; Figure 7.1(a)). This is called a nonlocal transport (or counter gradient transport).

In the mixed layer, the turbulence exhibits a nonlocal character, and the local buoyancy flux depends on the boundary layer parameters such as the sea surface flux $\langle wb \rangle$ and the thickness of the mixed layer h in addition to the local gradient. Furthermore, nonlocal transport has a non-zero value only for the scalar under unstable forcing (Deardroff, 1972). Here, using the parameterization of Mailhôt and Benoit (1982), the nonlocal transport γ_s for the scalar under the unstable forcing is calculated by

$$\gamma_s = C^* \frac{\langle ws_0 \rangle}{w^* h}, \tag{7.96}$$

where $C^* = 10$. These are summarized as follows:

$$\begin{aligned} \gamma_x &= 0 && \zeta \geq 0, \\ \gamma_m &= 0 && \zeta < 0, \\ \gamma_s &= C_s \frac{\langle ws_0 \rangle}{w_s(\sigma) h} && \zeta < 0, \\ \gamma_\theta &= C_s \frac{\langle w\theta_0 \rangle + \langle w\theta_R \rangle}{w_s(\sigma) h} && \zeta < 0, \end{aligned} \tag{7.97}$$

where

$$C_s = C^* \kappa (c_s \kappa \varepsilon)^{1/3}, \tag{7.98}$$

and $\langle w\theta_R \rangle$ expresses the absorption of short wave radiation, which will be detailed in Chapter 8.

References

Burchard, H., 2002: Energy-conserving discretization of turbulent shear and buoyancy production, *Ocean Modell.*, *4*, 347-361.

Deardroff, J. W., 1972: Theoretical expression for the counter-gradient vertical heat flux, *J. Geophys. Res.*, *77*, 5900-5904.

Kantha, L. H., and C. A. Clayson, 2000: *Small Scale Processes in geophysical Fluid Flows*, Academic Press, 888pp.

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.

Lumley, J. A., and H. A. Panofsky, 1964: *The structure of the atmospheric turbulence*, 239pp., John Wiley, New York.

Mailhôt, J., and R. Benoit, 1982: A finite-element model of the atmospheric boundary layer suitable for use with numerical weather prediction models, *J. Atmos. Sci.*, *39*, 2249-2266.

Mellor, G. L., and A. Blumberg, 2004: Wave breaking and ocean surface layer thermal response, *J. Phys. Oceanogr.*, *34*, 693-698.

Mellor, G. L., and T. Yamada, 1982: Development of a turbulence closure model for geophysical fluid problems, *Rev. Geophys. Space Phys.*, *20*, 851-875.

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.*, *32*, 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.

O'Brien, J. J., 1970: A note on the vertical structure of the eddy exchange coefficient in the planetary boundary layer, *J. Atmos. Sci.*, *27*, 1213-1215.

Rotta, J. C., 1951a: Statistische Theotie nichthomogener Turbulenz, *Z. Phys.*, *129*, 547-572.

Rotta, J. C., 1951b: Statistische Theotie nichthomogener Turbulenz, *Z. Phys.*, *131*, 51-77.

Tennekes, H., 1973: A model for the dynamics of the inversion above a convective boundary layer, *J. Atmos. Sci.*, *30*, 558-567.

Treon, I. B., and L. Mahrt, 1986: A simple model of the atmospheric boundary layer; sensitivity to surface evaporation, *Boundary Layer Meteorol.*, *37*, 129-148.

Chapter 8 Sea surface fluxes

Sea surface momentum, heat, and fresh water fluxes are needed to drive an ocean model. This chapter explains how those fluxes are treated in MRI.COM. The treatment is designed for the stand-alone simulation driven by atmospheric reanalysis data or observed data and simulation using a coupled atmosphere-ocean model.

By default, a stand-alone simulation is driven by momentum fluxes and temperature and salinity fluxes derived by restoring the model sea surface temperature and salinity to the observed ones. It would be natural to use sea surface momentum, heat and fresh water fluxes based on observation when a realistic simulation is intended. However, the simulated sea surface temperature and salinity do not always agree with the observed ones due to the following reasons.

1. Sea surface heat and fresh water fluxes derived from observations include large errors.
2. The ocean stand-alone simulations do not include feedback mechanisms from the ocean to the atmosphere.
3. Since ocean models have dynamical errors, they do not necessarily capture all of the phenomena.

Thus, it is not common in a stand-alone simulation to drive the model by fluxes without a feedback process, such as flux data derived from atmospheric reanalysis data. Rather, it is common to calculate fluxes using a bulk formula with sea surface meteorological elements derived from atmospheric reanalysis data.

For a simulation using a coupled atmosphere-ocean model, all surface fluxes are calculated by the atmospheric component and fluxes on the oceanic grid points are given by a flux coupler. Users are referred to Yukimoto et al. (2010) for the treatment of fluxes in a coupled model.

Section 8.1 describes momentum flux. The surface forcing terms for temperature and salinity are presented in Section 8.2, and their components, (heat flux (Section 8.3), fresh water flux (Section 8.4), and the equivalent fluxes under invariable first layer volume condition (Section 8.5)) are detailed in the succeeding sections. Several options are available for the choice of the bulk formula to calculate momentum, latent and sensible heat, and evaporative fluxes. Each bulk formula is detailed in Section 8.6 with description about a general formulation for the bulk transfer coefficient. Finally, the work flow in MRI.COM is summarized in Section 8.7.

8.1 Momentum flux (surface stress)

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 below sea ice), and is expressed as a body force to the first level velocity in the model algorithm

$$(F_x, F_y) = \frac{(\tau_x, \tau_y)}{\rho_0 \Delta z_{\frac{1}{2}}}, \tag{8.1}$$

where $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer and τ_x and τ_y are zonal and meridional stress.

8.1.1 Input of wind stress data

By default, wind stresses are given as external data and are read from a file. (This input file is always required, and thus a file filled with zeros is necessary when wind stress is not applied.) The input data should be in units of $[\mathrm{dyn}\cdot\mathrm{cm}^{-2}]$. The order of operations is as follows:

- Wind stress data are read from a file in `force.F90` in the following format:

  `read(unit = iwind)` τ_x, τ_y.

- Time-interpolated in `mkflux.F90` to produce surface stress.
- A part of the surface stress might be modified by the stress from sea ice.
- Surface stress is applied to the first level horizontal velocity in `clinic.F90` as follows:

$$\frac{\partial}{\partial t}(u_1, v_1) = \ldots + \frac{1}{\rho_0 \Delta z_{\frac{1}{2}}}(\tau_x, \tau_y). \tag{8.2}$$

In the last of the above operations, (u_1, v_1) is horizontal velocity at the first level and $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer.

8.1.2 Calculating wind stress using a bulk formula

If the model option `TAUBULK` is specified, the wind stress is calculated based on a bulk formula. In this option, zonal and meridional wind velocities at 10 m in units of $[\mathrm{cm}\cdot\mathrm{s}^{-1}]$ should be prepared as external data, instead of the default wind stress data. Each component of wind stress is calculated with a bulk transfer coefficient C_D (see Section 8.6) as follows:

$$\tau = \rho_a C_D |\mathbf{U}_a - \mathbf{U}_s|(\mathbf{U}_a - \mathbf{U}_s), \tag{8.3}$$

where ρ_a is the density of air, $\mathbf{U}_a$ is the wind vector at 10 m height, and $\mathbf{U}_s$ is the first level velocity.

8.2 Sea surface forcing for temperature and salinity

8.2.1 Temperature

The contribution of surface forcing (F_z^T) to the first layer temperature is expressed as

$$\frac{\partial T}{\partial t}\Big|_{k=\frac{1}{2}} = \ldots + \frac{F_z^T}{\Delta z_{\frac{1}{2}}}, \tag{8.4}$$

where $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer of the T-cell. The term (F_z^T) consists of the net sea surface heat flux gained by the ocean (Q_{NET}), heat transport due to fresh water flux, and restoring of SST (T_o) to a specific value (T^*):

$$F_z^T = \frac{Q_{NET}}{\rho_0 C_p} + (W_{AO} + W_{Ibot})\cdot T_o + W_{Isurf}\cdot T_{\mathrm{freeze}} - \frac{1}{\gamma_t}(T_o - T^*)\Delta z_{\frac{1}{2}}. \tag{8.5}$$

The heat flux term is converted to heat transport using the reference density (ρ_0) and specific heat (C_p) of sea water. The water transported through the air-sea interface (W_{AO}) and ice bottom (W_{Ibot}) is assumed to have the first level temperature (T_o). The water transported from the ice surface ($W_{Isurf} > 0$) is assumed to have the freezing point

temperature (T_{freeze}). The fourth term of the r.h.s. is a restoring term to the observed SST. The parameter γ_t is a restoring time (in units of seconds) and should be specified in the namelist `njobbdy` as `rtmsc` in units of days. The restoring term is converted to heat transport by multiplying the thickness of the first layer of T-cell ($\Delta z_{\frac{1}{2}}$).

By default, surface temperature forcing consists only of the restoring term. By specifying the option `HFLUX`, sea surface heat fluxes are calculated using the observed atmospheric variables. The fresh water flux terms are included when the option `WFLUX` is specified.

If the option `SFLUXW` is specified, fresh water flux is not added to the ocean and the expression for the heat transport term is modified as follows:

$$F_z^T = \frac{Q_{NET}}{\rho_0 C_p} - W_{Isurf} \cdot (T_o - T_{\text{freeze}}) - \frac{1}{\gamma_t}(T_o - T^*)\Delta z_{\frac{1}{2}}. \tag{8.6}$$

See Section 8.5 for the derivation.

Net sea surface heat flux Q_{NET} is given by

$$Q_{NET} = Q_{SH} + Q_{LO} + Q_{LA} + Q_{SN} + Q_{\text{ice}}, \tag{8.7}$$

where Q_{SH} is the net shortwave radiation flux, Q_{LO} is the net longwave radiation flux, Q_{LA} is the latent heat flux, and Q_{SN} is the sensible heat flux. The last term on the r.h.s., Q_{ice}, is due to exchange of heat with sea ice and will be detailed in Chapter 9.

Downward heat flux is defined as positive. Three of the first four components of the heat flux, Q_{LO}, Q_{LA}, and Q_{SN}, depend on sea surface temperature, and thus they are calculated using the model SST.

8.2.2 Salinity

The contribution of surface forcing (F_z^S) to the first layer salinity is expressed as

$$\frac{\partial S}{\partial t}\Big|_{k=\frac{1}{2}} = \ldots + \frac{F_z^S}{\Delta z_{\frac{1}{2}}}, \tag{8.8}$$

where $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer of the T-cell.

The explicit surface forcing (F_z^S) consists of salt transport due to formation and melting of sea ice and restoring of SSS (S_o) to a specific value (S^*):

$$F_z^S = (W_{Ibot} + W_{Isurf}) \cdot S_I - \frac{1}{\gamma_s}(S_o - S^*)\Delta z_{\frac{1}{2}}. \tag{8.9}$$

The water exchanged with ice ($W_{Ibot} + W_{Isurf}$) is assumed to have low salinity ($S_I = 4.0$ [psu]). The parameter γ_s is a restoring time (in units of seconds) and should be specified in the namelist `njobbdy` as `rtmsc` in units of days (the same as temperature). The restoring term is converted to salinity transport by multiplying the thickness of the first layer of the T-cell ($\Delta z_{\frac{1}{2}}$). The fresh water flux modifies the volume but not the salt content of the surface layer, changing the salinity of the surface layer (see also Chapter 4).

By default, surface salinity forcing consists only of the restoring term. When the option `WFLUX` is specified, the effect of the fresh water flux on salinity could be included.

If the option `SFLUXW` 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:

$$F_z^S = -W_{AO} \cdot S_o - (W_{Ibot} + W_{Isurf}) \cdot (S_o - S_I) - \frac{1}{\gamma_s}(S_o - S^*)\Delta z_{\frac{1}{2}}, \tag{8.10}$$

where S_o is the first level salinity. See Section 8.5 for the derivation.

8.3 Heat flux

8.3.1 Shortwave radiation flux

By default, the downward shortwave radiation (Q_{SH}^{down}) is read as external data. A part of the insolating shortwave radiation ($\alpha_o Q_{SH}^{\mathrm{down}}$) is reflected at the sea surface, and the remainder penetrates into the ocean interior as the net shortwave radiation flux ($Q_{SH}(0) = (1-\alpha_o)Q_{SH}^{\mathrm{down}}$), where α_o is the albedo at the ocean surface. More than 50% of the insolating radiation (near-infrared band) is absorbed within a depth of 1 m below the sea surface, but the remainder (visible and ultraviolet bands) penetrates further into the ocean interior and affects the subsurface temperature structure. According to Paulson and Simpson (1977), the shortwave radiation flux penetrating into the ocean interior is given by

$$Q_{SH}(z) = Q_{SH}(0)[R\exp(z/\zeta_1) + (1-R)\exp(z/\zeta_2)], \tag{8.11}$$

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). Vertical convergence of the penetrating shortwave radiation energy $\partial Q_{SH}(z)/\partial z$ is converted to heat within each layer.

Using option `SOLARANGLE` enables us to represent the shortwave radiation flux including the effect of the insolation angle. This scheme is based on Ishizaki and Yamanaka (2010). In this case, if the depth (z) on the r.h.s. of Equation (8.11) is replaced by the penetrating distance from the sea surface, the shortwave radiation is given by

$$Q_{SH}(z) = Q_{SH}(0)[R\exp(z/\zeta_1 \sin\theta_w) + (1-R)\exp(z/\zeta_2 \sin\theta_w)], \tag{8.12}$$

where θ_w is the penetrating angle in the ocean interior. When option `SOLARANGLE` is specified in coupled models, Equation (8.12) is replaced by

$$Q_{SH}(z) = Q_{SHb}(z) + Q_{SHd}(z), \tag{8.13}$$

where $Q_{SHb}(z)$ and $Q_{SHd}(z)$ are the shortwave radiation fluxes due to direct solar radiation and diffuse solar radiation. Those fluxes are expressed as follows:

$$Q_{SHb}(z) = Q_{SHb}(0)[R_b\exp(z/\zeta_1 \sin\theta_w) + (1-R_b)\exp(z/\zeta_2 \sin\theta_w)], \tag{8.14}$$

$$Q_{SHd}(z) = Q_{SHd}(0)[R_d\exp(z/\zeta_1) + (1-R_d)\exp(z/\zeta_2)], \tag{8.15}$$

where R_b and R_d are the ratio of near-infrared radiation to total radiation for direct solar radiation and diffuse solar radiation. $Q_{SHb}(0)$ and $Q_{SHd}(0)$ are the net direct and diffuse solar radiation fluxes at the sea surface.

There are three options for the sea surface albedo. The first option (`albedo_choice = 1` in the namelist `njobalb`) is a constant value, which should be specified as `alb` in the namelist `njobalb`. The second option (`albedo_choice = 2`) is based on Large and Yeager (2008) and is given by

$$\alpha_o = 0.069 - 0.011\cos(2\phi), \tag{8.16}$$

where ϕ is latitude. The third option (`albedo_choice = 3`) is based on Baker and Li (1995) and is given by

$$\alpha_o = 0.06 + 0.0421x^2 + 0.128x^3 - 0.04x^4 + \left(\frac{3.12}{5.68+U} + \frac{0.074x}{1.0+3.0U}\right)x^5, \tag{8.17}$$

where $x = 1 - \sin\theta_a$ (θ_a is a height angle of the Sun), and U is the surface wind speed [$\mathrm{m \cdot s^{-1}}$].

8.3.2 Shortwave radiation flux based on chlorophyll concentration

Recent studies indicate that solar radiation absorption and local heating within the upper ocean are strongly influenced by the chlorophyll concentration. Using both the `CHLMA94` and `NPZD` options enables us to use the 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

$$Q_{SH}(z) = Q_{SH}(0)[R\exp(-z/0.267\sin\theta_w) + (1-R)\{V_1\exp(-z/\zeta_1) + V_2\exp(-z/\zeta_2)\}], \tag{8.18}$$

The first exponential is for the infrared waveband ($>$ 750nm), which is not influenced by biological materials. The second and third exponentials are for the ultraviolet and visible bands ($<$ 750nm). V_1, V_2, ζ_1, and ζ_2 are calculated from an empirical relationship as a function of chlorophyll concentration (chl [mg $\cdot$ m^{-3}]) as follows:

$$V_1 = 0.321 + 0.008C + 0.132C^2 + 0.038C^3 - 0.017C^4 - 0.007C^5, \tag{8.19}$$
$$V_2 = 0.679 - 0.008C - 0.132C^2 - 0.038C^3 + 0.017C^4 + 0.007C^5, \tag{8.20}$$
$$\zeta_1 = 1.540 - 0.197C + 0.166C^2 - 0.252C^3 - 0.055C^4 + 0.042C^5, \tag{8.21}$$
$$\zeta_2 = 7.925 - 6.644C + 3.662C^2 - 1.815C^3 - 0.218C^4 + 0.502C^5, \tag{8.22}$$

where $C = log_{10}(chl)$. It is noted that $V_1 + V_2 = 1$.

When the option `SOLARANGLE` is added to the options mentioned above, the shortwave radiation is slightly modified by

$$Q_{SH}(z) = Q_{SH}(0)[R\exp(-z/0.267\sin\theta_w) + (1-R)\{V_1\exp(-z/\zeta_1\sin\theta_w) + V_2\exp(-z/\zeta_2\sin\theta_w)\}]. \tag{8.23}$$

In coupled models, Equation (8.23) is replaced by

$$Q_{SH}(z) = Q_{SHb}(z) + Q_{SHd}(z), \tag{8.24}$$

where $Q_{SHb}(z)$ and $Q_{SHd}(z)$ are the shortwave radiation fluxes due to direct solar radiation and that due to diffuse solar radiation. Those fluxes are expressed as follows:

$$Q_{SHb}(z) = Q_{SHb}(0)[R_b\exp(-z/0.267\sin\theta_w) + (1-R_b)\{V_1\exp(-z/\zeta_1\sin\theta_w) + V_2\exp(-z/\zeta_2\sin\theta_w)\}], \tag{8.25}$$
$$Q_{SHd}(z) = Q_{SHd}(0)[R_d\exp(-z/0.267) + (1-R_d)\{V_1\exp(-z/\zeta_1) + V_2\exp(-z/\zeta_2)\}], \tag{8.26}$$

where R_b and R_d are the ratios of the near-infrared part to the total radiation for direct and diffuse solar radiation, respectively.

8.3.3 Longwave radiation flux

Longwave radiation is calculated by removing the upward radiation due to the observed SST T_o^{OBS} (external data) from the net longwave radiation Q_{LO}^{OBS} (external data) and by adding the upward radiation due to the model SST.

$$Q_{LO} = Q_{LO}^{OBS} + e_m\sigma((T_o^{OBS} + 273.16)^4 - (T_o + 273.16)^4). \tag{8.27}$$

Here, $e_m = 0.97$ is the emissivity for sea water, and $\sigma = 5.67\times10^{-5}$[erg$\cdot$s$^{-1}\cdot$cm$^{-2}\cdot$K^{-4}] is the Stefan-Boltzmann constant.

Using option `LWDOWN` enables us to use the downward longwave radiation as external data. In this case, dummy data should be prepared for the SST data as external data.

8.3.4 Latent and sensible heat fluxes

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 8.6).

$$Q_{LA} = -\rho_a L C_E U_{10}(q_s - q_a) \tag{8.28}$$

and

$$Q_{SN} = -\rho_a C_{pa} C_H U_{10}(T_o - T_a). \tag{8.29}$$

Here, ρ_a is the atmospheric density, L is the latent heat for evaporation, q_a is the specific humidity at the sea surface, q_s is the saturated specific humidity of the sea surface temperature, T_a is the sea surface air temperature, and U_{10} is the scalar wind speed. The quantity C_{pa} is the specific heat for the atmosphere.

MRI.COM can provide three calculation methods for bulk transfer coefficients: Kondo (1975), Large and Yeager (2004), and Kara et al. (2002). In the following, we explain the elements necessary for calculating these bulk transfer coefficients and calculation of the latent and sensible heat fluxes.

(a) Kondo (1975) (`BULKKONDO2`)

The bulk formula based on Kondo (1975) uses sea surface pressure P_s [hPa], sea surface air temperature T_a [°C], sea surface specific humidity q_a[g·g^{-1}], and scalar wind speed U_{10}[cm·s^{-1}] as input data.

Elements necessary for calculating the latent and sensible heat fluxes are obtained as follows:

1. Saturated water vapor pressure e_s [hPa] for sea surface temperature (T_o [°C])

$$e_s = 0.98 \times 6.1078 \times 10^{7.5T_o/(237.3+T_o)}, \tag{8.30}$$

2. Saturated specific humidity for sea surface temperature q_s [g·g^{-1}]

$$q_s = \frac{0.62197 e_s}{P_s - 0.378 e_s}, \tag{8.31}$$

3. Latent heat of evaporation L [erg·g^{-1}]

$$L = 4.186 \times 10^7 (594.9 - 0.5T_o), \tag{8.32}$$

4. Atmospheric density ρ_a [g·cm^{-3}]

$$\rho_a = 1.205 \times 10^{-3}. \tag{8.33}$$

(b) Large and Yeager (2004) (`BULKNCAR`)

The bulk formula based on Large and Yeager (2004) uses the same input data and elements as those of Kondo (1975).

(c) Kara et al. (2002) (`BULKKARA`)

The bulk formula based on Kara et al. (2002) uses four elements: sea surface pressure P_s [hPa], sea surface air temperature T_a [°C], sea surface dew point temperature T_d[°C], and scalar wind speed U_{10} [cm·s^{-1}].

Elements necessary for calculating the latent and sensible heat fluxes are obtained as follows:

1. Water vapor pressure at the sea surface e_a[hPa]

$$e_a = 6.1121 \exp[(18.729 - T_d/227.3) T_d/(T_d + 257.87)], \tag{8.34}$$

2. Saturated water vapor pressure e_s[hPa] for sea surface temperature (T_o[°C])

$$e_s = 0.9815 \times 6.1121 \exp[(18.729 - T_o/227.3)T_o/(T_o + 257.87)], \tag{8.35}$$

3. Specific humidity at the sea surface q_a[g·g^{-1}]

$$q_a = \frac{0.62197 e_a}{P_s - 0.378 e_a}, \tag{8.36}$$

4. Latent heat of evaporation L[erg·g^{-1}]

$$L = 10^{10}(2.501 - 0.00237 T_o), \tag{8.37}$$

5. Air density ρ_a[g·cm^{-3}]

$$\rho_a = 10^3 P_s/[R_g(T_a + 273.16)(1.0 + 0.61 q_a)]. \tag{8.38}$$

Here, R_g is the gas constant for dry air.

The saturated specific humidity for sea surface temperature q_s[g·g^{-1}] is given by Equation (8.31), which is equivalent to Kondo (1975).

8.4 Freshwater flux

8.4.1 Introduction

Freshwater flux through the sea surface is caused mainly by precipitation (P), evaporation (E), river discharge (R), and formation-melting of sea ice (I). The fresh water fluxes are included in the model when option `WFLUX` is specified. In this case, precipitation and river discharge data should be prepared as input data. A free surface model can deal with freshwater flux explicitly and calculate the salinity change using a volume change of the first layer.

Even with the `WFLUX` option, the salinity restoring term could be added to the water flux by specifying the option `SFLUXR` to avoid unexpected salinity drift. The salinity restoring is converted to the equivalent fresh water flux in this case. It is possible to transform all freshwater fluxes to salinity flux by specifying option `SFLUXW`, in order to prevent the thickness of the first layer (nominally several meters) from becoming zero or unrealistically large due to excess precipitation or evaporation.

8.4.2 Calculating freshwater flux

Freshwater flux through the sea surface F^W[cm·s^{-1}] is given by

$$F^W = P - E + R + I, \tag{8.39}$$

where precipitation P is given as external data, and evaporation E is calculated in the model,

$$E = \rho_a C_E U_{10}(q_s - q_a) = -Q_{LA}/L. \tag{8.40}$$

River discharge R is given as external data when `RUNOFF` is selected. Freshwater flux I due to the formation and melting of the sea ice is added when the sea ice model is used (`ICE`) (see Chapter 9 for the sea ice model). We get $F^W = P - E$ when only `WFLUX` is selected.

Since F^W 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. By converting the restoring term to the fresh water flux, the expression for F^W becomes

$$F^W = P - E + R + I + \frac{1}{\gamma_s}\frac{S_o - S^*}{S_0}. \tag{8.41}$$

In this case, the salinity flux should be

$$F_z^S = (W_{Ibot} + W_{Isurf}) \cdot S_I, \tag{8.42}$$

where W_{Ibot} is the water transported from the ice bottom, W_{Isurf} is the water transported through the ice surface, and S_I is the salinity of the water exchanged with ice.

The global mean of F^W should not necessarily be zero, and thus the volume of the first layer averaged globally may continue to increase or decrease. To avoid this, the integral of the globally averaged freshwater flux is set to zero by selecting option `WADJ`,

$$F_{ADJ}^W = F^W - \overline{F^W}, \tag{8.43}$$

where $\overline{F^W}$ is the global mean of $F^W (= P - E(+R) + \gamma_s(S_o - S^*)/S_0)$.

8.5 Equivalent surface temperature and salinity fluxes for constant first layer volume

Precipitation, evaporation, and river discharge ($W_{AO} = P - E + R$) are assumed to have the first level temperature (T_o) 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 ($T_{\rm freeze}$). The water exchanged with ice is assumed to have low salinity ($S_I = 4.0$ [psu]). Note that the freezing point temperature ($T_{\rm freeze}$) is given by mS_I, where $m = -0.0543$ [K / psu].

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 by the following.

$$F_{WF}^T = (W_{AO} + W_{Ibot}) \cdot T_o + W_{Isurf} \cdot T_{\rm freeze}, \tag{8.44}$$

$$F_{WF}^S = (W_{Ibot} + W_{Isurf}) \cdot S_I. \tag{8.45}$$

If the surface fresh water flux is not added to the first layer to avoid an unexpected sea level rise or fall during a long-term integration, equivalent temperature and salinity fluxes should be imposed.

The conservation equations for the first layer heat and salinity are considered. Temperature and salinity are assumed to evolve from the old values $(T_{\rm old}, S_{\rm old})$ to the new values $(T_{\rm new}, S_{\rm new})$ due to fresh water flux.

If a volume change is allowed,

$$V_{\rm new} T_{\rm new} = V_{\rm old} T_{\rm old} + (W_{AO} T_{\rm old} + W_{Ibot} T_{\rm old} + W_{Isurf} T_{\rm freeze}) \cdot \Delta A \cdot \Delta t, \tag{8.46}$$

$$V_{\rm new} S_{\rm new} = V_{\rm old} S_{\rm old} + (W_{Ibot} S_I + W_{Isurf} S_I) \cdot \Delta A \cdot \Delta t, \tag{8.47}$$

$$V_{\rm new} = V_{\rm old} + (W_{AO} + W_{Ibot} + W_{Isurf}) \cdot \Delta A \cdot \Delta t, \tag{8.48}$$

where ΔA is the horizontal area.

If a volume change is *not* allowed,

$$V_{\rm old}T_{\rm new} = V_{\rm old}T_{\rm old} + F^T_{WF}\cdot\Delta A\cdot\Delta t, \tag{8.49}$$

$$V_{\rm old}S_{\rm new} = V_{\rm old}S_{\rm old} + F^S_{WF}\cdot\Delta A\cdot\Delta t, \tag{8.50}$$

where F^T_{WF} and F^S_{WF} are temperature and salinity flux for the constant volume case.

Removing $T_{\rm new}$, $S_{\rm new}$, and $V_{\rm new}$ from the above equations, we have,

$$\begin{aligned} &V_{\rm old}\{V_{\rm old}T_{\rm old} + (W_{AO}T_{\rm old} + W_{Ibot}T_{\rm old} + W_{Isurf}T_{\rm freeze})\cdot\Delta A\cdot\Delta t\} \\ &\quad = \{V_{\rm old} + (W_{AO} + W_{Ibot} + W_{Isurf})\cdot\Delta A\cdot\Delta t\}(V_{\rm old}T_{\rm old} + F^T_{WF}\cdot\Delta A\cdot\Delta t), \end{aligned} \tag{8.51}$$

$$\begin{aligned} &V_{\rm old}\{V_{\rm old}S_{\rm old} + (W_{Ibot}S_I + W_{Isurf}S_I)\cdot\Delta A\cdot\Delta t\} \\ &\quad = \{V_{\rm old} + (W_{AO} + W_{Ibot} + W_{Isurf})\cdot\Delta A\cdot\Delta t\}(V_{\rm old}S_{\rm old} + F^S_{WF}\cdot\Delta A\cdot\Delta t). \end{aligned} \tag{8.52}$$

The fluxes for the constant volume case are given by

$$F^T_{WF} = -\frac{V_{\rm old}}{V_{\rm old} + W\cdot\Delta A\cdot\Delta t}W_{Isurf}\cdot(T_{\rm old} - T_{\rm freeze}), \tag{8.53}$$

$$F^S_{WF} = -\frac{V_{\rm old}}{V_{\rm old} + W\cdot\Delta A\cdot\Delta t}\{W_{AO}\cdot S_{\rm old} + (W_{Ibot} + W_{Isurf})\cdot(S_{\rm old} - S_I)\}, \tag{8.54}$$

where $W = W_{AO} + W_{Ibot} + W_{Isurf}$.

If $|V_{\rm old}| \gg |W\cdot\Delta A\cdot\Delta t|$, we have,

$$F^T_{WF} = -W_{Isurf}\cdot(T_{\rm old} - T_{\rm freeze}), \tag{8.55}$$

$$F^S_{WF} = -W_{AO}\cdot S_{\rm old} - (W_{Ibot} + W_{Isurf})\cdot(S_{\rm old} - S_I). \tag{8.56}$$

8.6 Bulk transfer coefficient

This section briefly describes how to calculate sea surface fluxes using a bulk formula. For details, refer to Kantha and Clayson (2000) and Large and Yeager (2004).

8.6.1 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 require specific interactions 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:

$$\tau = -\rho_a\overline{wu} = \rho_a u_*^2, \tag{8.57}$$

$$H_S = -\rho_a c_p\overline{w\theta} = \rho_a c_p u_*\theta_*, \tag{8.58}$$

$$H_L = -\rho_a L_E\overline{wq} = \rho_a L_E u_* q_*, \tag{8.59}$$

$$E = -\rho_a\overline{wq} = \rho_a u_* q_* = H_L/L_E, \tag{8.60}$$

where a bar over a variable denotes a covariance between the turbulent component of vertical velocity and the turbulent component of each physical property, ρ_a is air density, u_* is friction velocity, $\theta_* \equiv H_S/(\rho_a c_p u_*)$ 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_p 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:

$$\tau = \rho_a C_{\mathrm{Dh}} (U_a - U_s)^2, \tag{8.61}$$

$$H_S = -\rho_a c_p |U_a - U_s| C_{\mathrm{Hh}} (T_s - T_a), \tag{8.62}$$

$$H_L = -\rho_a L_E |U_a - U_s| C_{\mathrm{Eh}} (q_s - q_a), \tag{8.63}$$

$$E = \rho_a |U_a - U_s| C_{\mathrm{Eh}} (q_s - q_a), \tag{8.64}$$

where U_a is the wind velocity at height of z_{h}, and U_s is the velocity component in the U_a direction at the sea surface. The subscript "a" means a value of each physical property at $z = z_{\mathrm{h}}$, and the subscript "s" means the value just above the sea surface.

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 (8.65), (8.66), and (8.67) and the similarity law of Monin-Obukhov mentioned below,

$$C_{\mathrm{Dh}} = \frac{u_*^2}{(U_a - U_s)^2}, \tag{8.65}$$

$$C_{\mathrm{Hh}} = -\frac{H_S/\rho_a c_p}{|U_a - U_s|(T_s - T_a)} = \frac{u_* \theta_*}{|U_a - U_s|(T_s - T_a)}, \tag{8.66}$$

$$C_{\mathrm{Eh}} = \frac{E/\rho_a}{|U_a - U_s|(q_s - q_a)} = \frac{u_* q_*}{|U_a - U_s|(q_s - q_a)}. \tag{8.67}$$

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 velocity, air temperature, and humidity can be written as follows:

$$\frac{\kappa z}{u_*}\frac{\partial U}{\partial z} = \phi_M\left(\frac{z}{L}\right), \tag{8.68}$$

$$\frac{\kappa z}{\theta_*}\frac{\partial T}{\partial z} = \phi_H\left(\frac{z}{L}\right), \tag{8.69}$$

$$\frac{\kappa z}{q_*}\frac{\partial q}{\partial z} = \phi_E\left(\frac{z}{L}\right), \tag{8.70}$$

where $\kappa = 0.4$ is the von Karman constant, and L is the Monin-Obukhov length scale

$$L = \frac{u_*^3}{\kappa Q_b} = -\frac{u_*^3 T_v}{\kappa g \overline{w\theta_v}}. \tag{8.71}$$

In (8.71), $Q_b = -g\overline{w\theta_v}/T_v$ is the buoyancy flux, T_v is the virtual temperature ($T_v = T(1 + 0.608\,q)$), and $\overline{w\theta_v}$ is the net heat flux including the water vapor flux:

$$\overline{w\theta_v} = \overline{w\theta}(1 + 0.608\,q) + 0.608\,T\,\overline{wq}. \tag{8.72}$$

In (8.68) to (8.70), $\zeta = z/L$ 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 (8.68), (8.69), and (8.70) vertically, we have the following.

$$U(z)-U_s = \frac{u_*}{\kappa}\left[\ln\frac{z}{z_0}-\Psi_M(\zeta)\right] \tag{8.73}$$

$$T(z)-T_s = \frac{\theta_*}{\kappa}\left[\ln\frac{z}{z_{0\mathrm{T}}}-\Psi_H(\zeta)\right] \tag{8.74}$$

$$q(z)-q_s = \frac{q_*}{\kappa}\left[\ln\frac{z}{z_{0\mathrm{E}}}-\Psi_E(\zeta)\right], \tag{8.75}$$

where

$$\Psi_{M,H,E}(\zeta)=\int_0^\zeta\left[1-\phi_{M,H,E}(\zeta')\frac{d\zeta'}{\zeta'}\right]. \tag{8.76}$$

In (8.73) to (8.75), z_0, $z_{0\mathrm{T}}$, and $z_{0\mathrm{E}}$ 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,

$$C_{\mathrm{Dh}}=\frac{\kappa^2}{\left[\ln\frac{z_\mathrm{h}}{z_0}-\Psi_M(\zeta_h)\right]^2}, \tag{8.77}$$

$$C_{\mathrm{Hh,Eh}}=\frac{\kappa^2}{\left[\ln\frac{z_\mathrm{h}}{z_0}-\Psi_M(\zeta_h)\right]\left[\ln\frac{z_\mathrm{h}}{z_{0\mathrm{T,0E}}}-\Psi_{H,E}(\zeta_h)\right]}. \tag{8.78}$$

If neutrally stable ($\zeta=0$), the bulk transfer coefficient is a function of the roughness length only and is expressed as follows:

$$C_{\mathrm{DNh}}=\frac{\kappa^2}{\left[\ln\frac{z_\mathrm{h}}{z_{0\mathrm{N}}}\right]^2}, \tag{8.79}$$

$$C_{\mathrm{HNh,ENh}} = \frac{\kappa^2}{\left[\ln\frac{z_\mathrm{h}}{z_{0\mathrm{N}}}\right]\left[\ln\frac{z_\mathrm{h}}{z_{0\mathrm{TN,0EN}}}\right]} \tag{8.80}$$

$$= \frac{C_{\mathrm{DNh}}}{\left[1-\frac{1}{\kappa}C_{\mathrm{DNh}}^{1/2}\ln\frac{z_{0\mathrm{TN,0EN}}}{z_{0\mathrm{N}}}\right]} \tag{8.81}$$

$$= \frac{\kappa C_{\mathrm{DNh}}^{1/2}}{\left[\ln\frac{z_\mathrm{h}}{z_{0\mathrm{TN,0EN}}}\right]}. \tag{8.82}$$

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 and stable bulk transfer coefficients at a height of 10 m by eliminating the roughness length as follows:

$$C_{\mathrm{Dh}}=\frac{C_{\mathrm{DN10}}}{\left[1+\frac{1}{\kappa}C_{\mathrm{DN10}}C_{\mathrm{DN10}}^{-1/2}\left(\ln\frac{z_\mathrm{h}}{z_{10}}-\Psi_M(\zeta_h)\right)\right]^2}, \tag{8.83}$$

$$C_{\mathrm{Hh,Eh}}=\frac{C_{\mathrm{HN10,EN10}}\left(\frac{C_{\mathrm{Dh}}}{C_{\mathrm{DN10}}}\right)^{1/2}}{\left[1+\frac{1}{\kappa}C_{\mathrm{HN10,EN10}}C_{\mathrm{DN10}}^{-1/2}\left(\ln\frac{z_\mathrm{h}}{z_{10}}-\Psi_{H,E}(\zeta_h)\right)\right]}, \tag{8.84}$$

where z_{10} means $z = 10$[m].

The neutral bulk transfer coefficients at a height of 10 m (C_{DN10}, C_{HN10}, and C_{EN10}) are often estimated, according to the stability, as a function of velocity at 10 m. 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 and stable case and use a bulk coefficient at 10 m in the neutral case,
- 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 and neutral case into one with a height and a stability at which wind speeds are observed.

8.6.2 Kondo (1975) `BULKKONDO2`

When 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 estimated at the same height. In this case, 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`, you may specify `BULKITER` in `configure.in` to do this iteration. The observed heights are specified as `altu` (wind), `altt` (temperature), and `altq` (humidity) in the namelist `njbkondo`.

The following procedure is executed:

1. Estimate the neutral bulk transfer coefficient at 10 m assuming that the observed wind speed is at 10 m in the neutral case. When the 10 m wind speed is already known, the neutral bulk transfer coefficient is given as follows:
$$C_{\mathrm{DN10}} = 10^{-3}\{a_d + 10^{-2} b_d U_{10}^{p_d}\}, \tag{8.85}$$
where a_d, b_d, and p_d are nondimensional numbers depending on the wind speed (Table 8.1).

2. Recalculate the roughness length using this neutral bulk transfer coefficient at 10 m and correct the height for wind speed once again:
$$z_0 = \exp\left\{\ln z_{10} - \kappa C_{\mathrm{DN10}}^{-1/2}\right\}, \tag{8.86}$$
$$u_{10} = u(z)\ln(10/z_0)/\ln(z/z_0). \tag{8.87}$$

3. Re-estimate the neutral bulk transfer coefficient at 10 m.

4. Convert the bulk transfer coefficient to that in the neutral case at the observed height:
$$C_{\mathrm{DNh}} = \frac{\kappa^2}{\left[\kappa C_{\mathrm{DN10}}^{-1/2} - \ln(z_{10}/z_{\mathrm{h}})\right]^2} \tag{8.88}$$

$$C_{\mathrm{HNh,ENh}} = \frac{\kappa C_{\mathrm{DNh}}^{1/2}}{\kappa C_{\mathrm{DN10}}^{1/2}/C_{\mathrm{HN10,EN10}} + \ln(z_{\mathrm{h}}/z_{10})}, \tag{8.89}$$

where z_{10} means $z = 10[\mathrm{m}]$.

5. Estimate the bulk transfer coefficient in which stability is considered using the neutral bulk transfer coefficient obtained above.

The neutral bulk transfer coefficient at 10 m is given as follows:

$$C_{\mathrm{DN10}} = 10^{-3}\{a_d + 10^{-2} b_d U_{10}^{p_d}\}, \tag{8.90}$$

$$C_{\mathrm{EN10}} = 10^{-3}\{a_e + 10^{-2} b_e U_{10}^{p_e} + c_e (10^{-2} U_{10} - 8)^2\}, \tag{8.91}$$

$$C_{\mathrm{NN10}} = 10^{-3}\{a_h + 10^{-2} b_h U_{10}^{p_h} + c_h (10^{-2} U_{10} - 8)^2\}, \tag{8.92}$$

where C_{DN10}, C_{EN10}, and C_{HN10} are bulk transfer coefficients for neutral atmospheric stability. and $a_{e,h}$, $b_{e,h}$, $c_{e,h}$, and $p_{e,h}$ are nondimensional numbers depending on wind speed (Table 8.1).

Table 8.1. Non-dimensional parameters used in calculating the bulk transfer coefficient based on Kondo (1975)

U_{10} $(\mathrm{m \cdot s^{-1}})$	a_d	a_e	a_h	b_d	b_e	b_h	c_e	c_h	p_d	p_e	p_h
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:

$$s = s_0 \frac{|s_0|}{|s_0| + 0.01}, \tag{8.93}$$

where

$$s_0 = \frac{T_o - T_a}{10^{-4} U_{10}^2}. \tag{8.94}$$

The atmospheric boundary layer is unstable if $s > 0$ $(T_o - T_a > 0)$ and stable if $s < 0$ $(T_o - T_a < 0)$. Finally, the bulk transfer coefficient depending on the stability is given as follows:

$$C_D = \begin{cases} 0 & s < -3.3 \\ C_{\mathrm{DNh}}\{0.1 + 0.03s + 0.9\exp(4.8s)\} & -3.3 \le s < 0 \\ C_{\mathrm{DNh}}(1.0 + 0.47\sqrt{s}) & 0 \le s, \end{cases} \tag{8.95}$$

$$C_E = \begin{cases} 0 & s < -3.3 \\ C_{\mathrm{ENh}}\{0.1 + 0.03s + 0.9\exp(4.8s)\} & -3.3 \le s < 0 \\ C_{\mathrm{ENh}}(1.0 + 0.63\sqrt{s}) & 0 \le s, \end{cases} \tag{8.96}$$

$$C_H = \begin{cases} 0 & s < -3.3 \\ C_{\mathrm{HNh}}\{0.1 + 0.03s + 0.9\exp(4.8s)\} & -3.3 \le s < 0 \\ C_{\mathrm{HNh}}(1.0 + 0.63\sqrt{s}) & 0 \le s. \end{cases} \tag{8.97}$$

8.6.3 Large and Yeager (2004) BULKNCAR

In the bulk formula based on Large and Yeager (2004), 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 the option `BULKITER` in `configure.in`. The observed heights are specified as `altu` (wind) `altt` (temperature), and `altq` (humidity) in the namelist `njbncar`.

The bulk transfer coefficient at 10 m in the neutral case is given as follows:

$$10^3 C_{\mathrm{DN10}} = \frac{2.70}{U_{\mathrm{10N}}} + 0.142 + \frac{U_{\mathrm{10N}}}{13.09}, \tag{8.98}$$

$$10^3 C_{\mathrm{HN10,EN10}} = 18.0\sqrt{C_{\mathrm{DN10}}}, \ \text{stable } \zeta > 0, \tag{8.99}$$

$$10^3 C_{\mathrm{HN10,EN10}} = 32.7\sqrt{C_{\mathrm{DN10}}}, \ \text{unstable } \zeta \leq 0, \tag{8.100}$$

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:

$$\theta_v = \theta(z_\theta)(1 + 0.608\, q(z_q)) \tag{8.101}$$

Next, calculate the bulk transfer coefficient at 10 m in the neutral and stable cases assuming that the first guess for the 10 m wind speed in the neutral and stable cases is $U_{\mathrm{10N}} = U(z_u)$ (Equations (8.98) to (8.100)). 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,

$$u_* = \sqrt{\frac{\tau}{\rho_a}} = \sqrt{C_{\mathrm{DN10}}}\, U(z_u), \tag{8.102}$$

$$\theta_* = \frac{H_S}{\rho_a c_p u_*} = \frac{C_{\mathrm{HN10}}}{\sqrt{C_{\mathrm{DN10}}}} (\theta(z_\theta) - T_s), \tag{8.103}$$

$$q_* = \frac{E}{\rho_a u_*} = \frac{C_{\mathrm{EN10}}}{\sqrt{C_{\mathrm{DN10}}}} (q(z_q) - q_{\mathrm{sat}}(T_s)), \tag{8.104}$$

where $q_{\mathrm{sat}}(T_s)$ 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,

$$\zeta = \frac{\kappa g z}{u_*^2} \left[\frac{\theta_*}{\theta_v} + \frac{q_*}{(q(z_q) + 0.608^{-1})} \right]. \tag{8.105}$$

The integral of the non-dimensional profile function is expressed as

$$\Psi_M(\zeta) = \Psi_H(\zeta) = -5\zeta, \tag{8.106}$$

if it is stable ($\zeta \geq 0$), and

$$\Psi_M(\zeta) = 2\ln\left(\frac{1+X}{2}\right) + \ln\left(\frac{1+X^2}{2}\right) - 2\tan^{-1}(X) + \frac{\pi}{2}, \tag{8.107}$$

$$\Psi_H(\zeta) = 2\ln\left(\frac{1+X^2}{2}\right), \tag{8.108}$$

if it is unstable ($\zeta < 0$). In the above,

$$X = (1 - 16\zeta)^{1/4}. \tag{8.109}$$

Using these values, convert the wind speed to that at 10 m in the neutral and stable cases, and convert the temperature and specific humidity to those at a height where the wind speed is observed,

$$U_{10\mathrm{N}} = U(z_u)\left(1 + \frac{\sqrt{C_{\mathrm{DN10}}}}{\kappa}\left[\ln\frac{z_u}{z_{10}} - \Psi_M(\zeta_u)\right]\right)^{-1}, \tag{8.110}$$

$$\theta(z_u) = \theta(z_\theta) - \frac{\theta_*}{\kappa}\left[\ln\frac{z_\theta}{z_u} + \Psi_H(\zeta_u) - \Psi_H(\zeta_\theta)\right], \tag{8.111}$$

$$q(z_u) = q(z_q) - \frac{q_*}{\kappa}\left[\ln\frac{z_q}{z_u} + \Psi_H(\zeta_u) - \Psi_H(\zeta_q)\right], \tag{8.112}$$

$$\tag{8.113}$$

where z_{10} means $z = 10$[m]. Estimate the bulk coefficient at 10 m in the neutral and stable cases using $U_{10\mathrm{N}}$, and then obtain the bulk coefficient at a height (z_u) where wind speed is observed,

$$C_{\mathrm{Du}} = \frac{C_{\mathrm{DN10}}}{\left[1 + \frac{1}{\kappa}C_{\mathrm{DN10}}C_{\mathrm{DN10}}^{-1/2}\left(\ln\frac{z_u}{z_{10}} - \Psi_M(\zeta_u)\right)\right]^2}, \tag{8.114}$$

$$C_{\mathrm{Hu}} = \frac{C_{\mathrm{HN10}}\left(\frac{C_{\mathrm{Du}}}{C_{\mathrm{DN10}}}\right)^{1/2}}{\left[1 + \frac{1}{\kappa}C_{\mathrm{HN10}}C_{\mathrm{DN10}}^{-1/2}\left(\ln\frac{z_u}{z_{10}} - \Psi_H(\zeta_u)\right)\right]}, \tag{8.115}$$

$$C_{\mathrm{Eu}} = \frac{C_{\mathrm{EN10}}\left(\frac{C_{\mathrm{Du}}}{C_{\mathrm{DN10}}}\right)^{1/2}}{\left[1 + \frac{1}{\kappa}C_{\mathrm{EN10}}C_{\mathrm{DN10}}^{-1/2}\left(\ln\frac{z_u}{z_{10}} - \Psi_H(\zeta_u)\right)\right]}. \tag{8.116}$$

Repeat the procedures to calculate the bulk coefficients using these bulk coefficients with temperature and specific humidity at $z = z_u$, and recalculate the scales for virtual temperature, friction velocity, temperature, and specific humidity (Equations of (8.102), (8.103), and (8.104)).

8.6.4 Kara et al. (2002) `BULKKARA`

In the bulk formula based on Kara et al. (2002), bulk transfer coefficients C_D (momentum), C_E (latent heat), and C_H (sensible heat) are calculated using a polynomial fitting as follows to reduce the calculation cost due to iteration:

$$C_{0D} = 10^{-3}(0.692 + 0.071U_{10} - 0.00070U_{10}^2), \tag{8.117}$$

$$C_{1D} = 10^{-3}(0.083 - 0.0054U_{10} + 0.000093U_{10}^2), \tag{8.118}$$

$$C_D = C_{0D} + C_{1D}(T_o - T_a), \tag{8.119}$$

$$C_{0H} = 10^{-3}(0.8195 + 0.0506U_{10} - 0.0009U_{10}^2), \tag{8.120}$$

$$C_{1H} = 10^{-3}(-0.0154 + 0.5698/U_{10} - 0.6743/U_{10}^2), \tag{8.121}$$

$$C_E = C_{0H} + C_{1H}(T_o - T_a), \tag{8.122}$$

$$C_H = 0.95C_E, \tag{8.123}$$

where $2.5[\mathrm{m}\cdot\mathrm{s}^{-1}] \leq U_{10} \leq 32.5[\mathrm{m}\cdot\mathrm{s}^{-1}]$.

Flux correction factors may be specified by setting `close_budgets`, `corr_factor_n`, and `corr_factor_s` in namelist `njbkara`.

8.6.5 Bulk coefficient over sea ice

A similar treatment is necessary even over sea ice, but it is common to replace the neutral and stable bulk coefficients at 10 m with constants, as described in Large and Yeager (2004). Then, bulk coefficients and wind speed at each height are calculated based on the stability.

Large and Yeager (2004) uses

$$C_{\mathrm{DN10}} = C_{\mathrm{HN10}} = C_{\mathrm{EN10}} = 1.63 \times 10^{-3}, \tag{8.124}$$

but the default in MRI.COM uses, following Mellor and Kantha (1989),

$$C_{\mathrm{DN10}} = 3.0 \times 10^{-3} \tag{8.125}$$

$$C_{\mathrm{HN10}} = C_{\mathrm{EN10}} = 1.5 \times 10^{-3}. \tag{8.126}$$

The wind stress could be separately calculated over sea ice under the option `TAUBULK`. Users should specify `flg_strsi` = `.true.` in `ice_main_cat.F90`.

8.7 Work flow in MRI.COM

8.7.1 Momentum flux

- Use of bulk formula (`TAUBULK`)
  1. Input of external data (velocity vectors at 10 m height) in `force.F90`.
     (a) Time interpolation of external data in `mkflux.F90`.
     (b) Calculation of wind stress (τ_x, τ_y) (Equation (8.3)) in `bulk.F90` (called from `mkflux.F90`).
     (c) Calaculation of ice-water stress and total surface stress based on area-weighted average in `ice_dyn.F90` (called from the sea ice part (`ICE`), see Chapter 9).
  2. Update velocity for the first layer (Equation (8.2)) in `clinic.F90`.
- Use of external data (`default`)
  1. Input of external data in `force.F90`.
  2. Time interpolation of external data in `mkflux.F90`.
  3. Calaculation of ice-water stress and total surface stress based on area-weighted average in `ice_dyn.F90` (called from the sea ice part (`ICE`), see Chapter 9).
  4. Update velocity for the first layer (Equation (8.2)) in `clinic.F90`.
- Coupled model
  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. Calaculation of ice-water stress and total surface stress based on area-weighted average in `ice_dyn.F90` (called from the sea ice part (`ICE`), see Chapter 9).
4. Update velocity for the first layer (Equation (8.2)) in `clinic.F90`.

8.7.2 Temperature (heat) flux

- Heat flux type (`HFLUX`)

  1. Input of external data in `force.F90`
     (a) `read`(`unit` = `ihflx`) $Q_{SH}^{OBS}, Q_{LO}^{OBS}, T_o^{OBS}$
     (b) `read`(`unit` = `ibulk`) T_a, q_a (T_d, if `TDEW`), U_{10}, P_s
  2. Calculation of F_T in `mkflux.F90`
     (a) Time interpolation of external data.
     (b) Calculation of Q_{SH} (Equation (8.11)).
     (c) Calculation of Q_{LO} （Equation (8.27)).
     (d) Calculation of Q_{LA} and Q_{SN} (Equation (8.28)-(8.29)) in `bulk.F90` (called from `mkflux.F90`).
     (e) Preparation for Q_{SH} and $(Q_{LO} + Q_{LA} + Q_{SN})$.
     (f) Heat flux between sea ice and sea water (Q_{ice}) in `ice_main_cat.F90` (called from the sea ice part (`ICE`), see Chapter 9).
     (g) Calculation of F_{WF}^T (Equation (8.44) or (8.55)).
  3. Update first layer temperature using F^T (Equation (8.4)) in `tracer.F90`.
     – Update temperature at the depth deeper than the first layer using shortwave absorption (Equation (8.11)).

- Restoring sea surface temperature type (`default`)

  1. Calculation of F^T (the last term on the r.h.s. of Equation (8.5)) in `mkflux.F90`.
  2. Update the first layer temperature using F^T (Equation (8.4)) in `tracer.F90`.

- Coupled model

  1. Getting fluxes from the atmospheric model at the beginning of a coupling cycle in `cgcm_scup__get_a2o` (`cgcm_scup.F90`).
  2. Calculation of F_T in `mkflux.F90`
     (a) Getting fluxes for this step from `get_flux_a2o` (`get_fluxes.F90`).
     (b) Preparation for Q_{SH} and $(Q_{LO} + Q_{LA} + Q_{SN})$.
     (c) Heat flux between sea ice and sea water (Q_{ice}) in `ice_main_cat.F90` (called from the sea ice part (`ICE`), see Chapter 9).
     (d) Calculation of F_{WF}^T (Equation (8.44) or (8.55)).
  3. Update first layer temperature using F^T (Equation (8.4)) in `tracer.F90`.
     – Update temperature at the depth deeper than the first layer using shortwave absorption (Equation (8.11)).

8.7.3 Salinity and fresh water flux

- Freshwater flux type `WFLUX`

  1. Input of external data in `force.F90`:
     (a) `read(ipcpr)` P
     (b) `read(irnof)` R, when `RUNOFF`
  2. Calculation of F^W and F^S in `mkflux.F90`:
     (a) Time interpolation of external data.
     (b) Calculation of E (Equation (8.40)) in `bulk.F90` (called from `mkflux.F90`).
     (c) Freshwater and salinity fluxes between sea ice and sea water I and F^S (Equation (8.42)) in `ice_main_cat.F90` (called from the sea ice part (`ICE`), see Chapter 9).
     (d) Restoring term for sea surface salinity and conversion to F^W (Equation (8.41)).
     (e) Calculation of F^W_{ADJ} for the option `WADJ` (Equation (8.43)).
     (f) Set $F^W = 0$ and reevaluate F^S for option `SFLUXW` (Equation (8.10)).
  3. Update salinity of the first layer:
     (a) Update the first layer volume using F^W (or F^W_{ADJ}) in `surface_integ.F90`.
     (b) Update the first layer salinity using F^S (Equation (8.8)) in `tracer.F90`.

- Restoring sea surface salinity (`default`)

  1. Calculation of F^S (the last term on the r.h.s. of Equation (8.9)) in `mkflux.F90`
  2. Update the first layer salinity (Equation (8.8)) in `tracer.F90`

- Coupled model

  1. Getting fluxes from the atmospheric model at the beginning of a coupling cycle in `cgcm_scup__get_a2o` (`cgcm_scup.F90`).
  2. Calculation of F^W and F^S in `mkflux.F90`:
     (a) Getting fluxes for this step from `get_flux_a2o` (`get_fluxes.F90`).
     (b) Freshwater and salinity fluxes between sea ice and sea water I and F^S (Equation (8.42)) in `ice_main_cat.F90` (called from the sea ice part (`ICE`), see Chapter 9).
     (c) Calculation of restoring term for sea surface salinity and conversion to F^W (Equation (8.41)).
     (d) Calculation of F^W_{ADJ} for the option `WADJ` (Equation (8.43)).
     (e) Set $F^W = 0$ and reevaluate F^S for the option `SFLUXW` (Equation (8.10)).
  3. Update salinity of the first layer:
     (a) Update the first layer volume using F^W (or F^W_{ADJ}) in `surface_integ.F90`.
     (b) Update the first layer salinity using F^S (Equation (8.8)) in `tracer.F90`.

8.8 Remarks

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 use. For example, a bias of several $W \cdot m^{-2}$ in heat flux greatly affects the thickness of sea ice, meaning that accuracy on the order of several $W \cdot m^{-2}$ 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 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.

8.9 Appendix

8.9.1 Unit of constants

Symbol and Name		Numeral	Unit	
			Model(CGS)	S I
ρ_o	sea water density	1.0	$\times g \cdot cm^{-3}$	$\times 10^3 Kg \cdot m^{-3}$
ρ_a	air density	1.205×10^{-3}	$\times g \cdot cm^{-3}$	$\times 10^3 Kg \cdot m^{-3}$
C_p	sea water specific heat	3.99	$10^7 erg \cdot g^{-1} \cdot K^{-1}$	$10^3 J \cdot Kg^{-1} \cdot K^{-1}$
C_{pa}	air specific heat	1.00467	$10^7 erg \cdot g^{-1} \cdot K^{-1}$	$10^3 J \cdot Kg^{-1} \cdot K^{-1}$
α_o	sea surface albedo	`alb`		
e_m	sea water emissivity	0.97		
σ	Stefan-Boltzmann constant	5.67	$10^{-5} erg \cdot s^{-1} \cdot cm^{-2} \cdot K^{-4}$	$10^{-8} J \cdot s^{-1} \cdot m^{-2} \cdot K^{-4}$
R_g	gas constant for dry air	2.871	$10^6 erg \cdot g^{-1} \cdot K^{-1}$	$10^2 J \cdot Kg^{-1} \cdot K^{-1}$

8.9.2 Unit of variables

Symbol and Name		Unit	
		Model(CGS)	S I
Q_{**}	Sea surface heat flux	$\times g \cdot s^{-3}$	$\times 10^{-3} W \cdot m^{-2}$
			$10^{-3} J \cdot s^{-1} \cdot m^{-2}$
F^W	Freshwater flux	$cm \cdot s^{-1}$	$10^{-2} m \cdot s^{-1}$
τ_*	Momentum flux (wind stress)	$dyn \cdot cm^{-2}$	$10^{-1}\ N \cdot m^{-2}$
$\Delta z_{\frac{1}{2}}$	the first layer thickness	cm	$10^{-2} m$
T_*	temperature	°C	°C
P_s	sea surface pressure	hPa (not in CGS)	hPa
U_{10}	scalar wind speed	$cm \cdot s^{-1}$	$10^{-2} m \cdot s^{-1}$
ρ_*	density	$g \cdot cm^{-3}$	$10^3 Kg \cdot m^{-3}$
q_*	specific humidity	$g \cdot g^{-1}$	$Kg \cdot Kg^{-1}$
e_*	water vapor pressure	hPa (not in CGS)	hPa
L	evaporation latent heat	$erg \cdot g^{-1}$	$J \cdot Kg^{-1}$

References

Baker, H. W., and Z. Li, 1995: Improved simulation of clear-sky shortwave radiative transfer in the CCC-GCM., *J. Climate*, *8*, 2213-2223.

Ishizaki, H., and G. Yamanaka, 2010: Impact of explicit sun altitude in solar radiation on an ocean model simulation, *Ocean Modell.*, *xxx*, in press.

Jerlov, N. G., 1976: *Marine Optics*, 231pp., Elsevier.

Kantha, L. H., and C. A. Clayson, 2000: *Small Scale Processes in Geophysical Fluid Flows.*, 888pp., Academic Press.

Kara, A. B., P. A. Rochford, and H. E. Hurlburt, 2002: Air-sea flux estimates and the 1997-1998 ENSO event., *Boundary-Layer Meteorol.*, *103*, 439-458.

Kondo, J., 1975: Air-sea bulk transfer coefficients in diabatic conditions., *Boundary-Layer Meteorol.*, *9*, 91-112.

Large, W. G., and S. G., Yeager, 2008: The global climatology of an interannually varying air-sea flux data set., *Climate Dynamics*, *DOI 10.1007/s00382-008-0441-3*, 24pp.

Large, W. G., and S. G., Yeager, 2004: *Diurnal to decadal global forcing for ocean and sea-ice models: the data sets and flux climatologies.*, NCAR Technical Note: NCAR/TN-460+STR. CGD Division of the National Center for Atmospheric Research.

Mellor, G. L., and L. Kantha, 1989: An ice-ocean coupled model, *J. Geophys. Res.*, *94*, 10937-10954.

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.

Paulson, C. A., and J. J. Simpson, 1977: Irradiance measurements in the upper ocean, *J. Phys. Oceanogr.*, *7*, 952-956.

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), pp.306..

Yukimoto, S., and coauthors, 2010: *Meteorological Research Institute-Earth System Model v1 (MRI-ESM1) - Model Description* -, Technical Reports of the Meteorological Research Institute, No.64, in press.

Chapter 9 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 9.1 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 9.2, remapping among thickness categories in Section 9.3, dynamics in Section 9.4, advection in Section 9.5, and ridging in Section 9.6. Discretization issues are described in Section 9.7, and finally some technical issues are presented in Section 9.8.

9.1 Outline

The sea ice part of an ocean model gives surface boundary conditions. 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 nor included there, such as categorizing by thickness, ridging, and rheology, we adopt those of the Los Alamos sea ice model (CICE; Hunke and Lipscomb, 2006).

The fundamental property that defines the state of sea ice is the fractional area as a function of location (x,y) and thickness (h_I). The equation for this distribution function ($g(x,y,h_I)$) is

$$\frac{\partial g}{\partial t} = \frac{\partial}{\partial h_I}(fg) - \frac{1}{h_\mu h_\psi}\Big(\frac{\partial (g h_\psi u_I)}{\partial \mu} + \frac{\partial (g h_\mu v_I)}{\partial \psi}\Big) + \chi, \tag{9.1}$$

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. 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 (χ; 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:

$$a_n = \int_{H_{n-1}}^{H_n} g dh. \tag{9.2}$$

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 ice and snow have one layer with the heat capacity for sea ice but without heat capacity for snow. The heat capacity for sea ice is due to brine and is represented by the temperature at the center of the ice. It is assumed that sea ice has the same energy (temperature) throughout the whole layer. Figure 9.1 and Table 9.1 summarize symbols used in this chapter and their variable names in the source code of MRI.COM.

Table 9.1. Physical quantities used in the sea ice part (cf. Figure 9.1) and their variable names in the source code.

	Meaning	Variable name	Variable name of average
h_I	ice thickness	`hicen`	`hiceo`
h_s	snow thickness	`hsnwn`	`hsnwo`
A	area fraction (compactness)	`aicen`	`a0iceo`
Ah_I	average ice thickness (volume)	`hin`	`hi`
Ah_s	average snow thickness (volume)	`hsn`	`hsnw`
T_3	skin temperature of upper surface	`tsfcin`	`tsfci`
T_1	temperature of ice	`t1icen`	-
T_0	skin temperature of lower surface	`t0icen`	`t0iceo` (under the sea ice)
T_{0L}	skin temperature of sea surface at open leads	`t0icen`	`t0icel` (open leads)
S_0	skin salinity of lower surface	`s0n`	`s0`
S_{0L}	skin salinity of sea surface at open leads	`s0n`	`s0l`
Q_{IO}	heat flux on the ice side of the ice bottom	`fheatn`	-
Q_{AO}	heat flux on the air side at open leads	`fheat`	-
F_{TI}	heat flux on the ocean side of the ice bottom	`ftio`	-
F_{TL}	heat flux on the ocean side at open leads	`ftao`	-
F_{SI}, F_{SL}	salinity flux driving first layer of the ocean model	-	`sfluxi` $= F_{SI} + F_{SL}$
W	fresh water flux driving first layer of the ocean model	-	`wfluxi` $= W_{IO} + W_{AO} + W_{RO} + W_{FR}$
W_{AI}	fresh water flux due to snow fall at the upper surface of ice	-	`snowfall`
W_{AI}	fresh water flux due to sublimation at the upper surface of ice	`sublim`	-
W_{IO}	fresh water flux due to freezing and melting at the bottom of ice	`wio`	-
W_{AO}	fresh water flux due to freezing at open leads	`wao`	-
W_{RO}	fresh water flux due to melting at the upper surface of ice	-	`wrss, wrsi` (snow, ice)
W_{FR}	fresh water flux due to formation of frazil ice	-	`wrso`
u_I	zonal component of ice pack velocity	-	`uice`
v_I	meridional component of ice pack velocity	-	`vice`

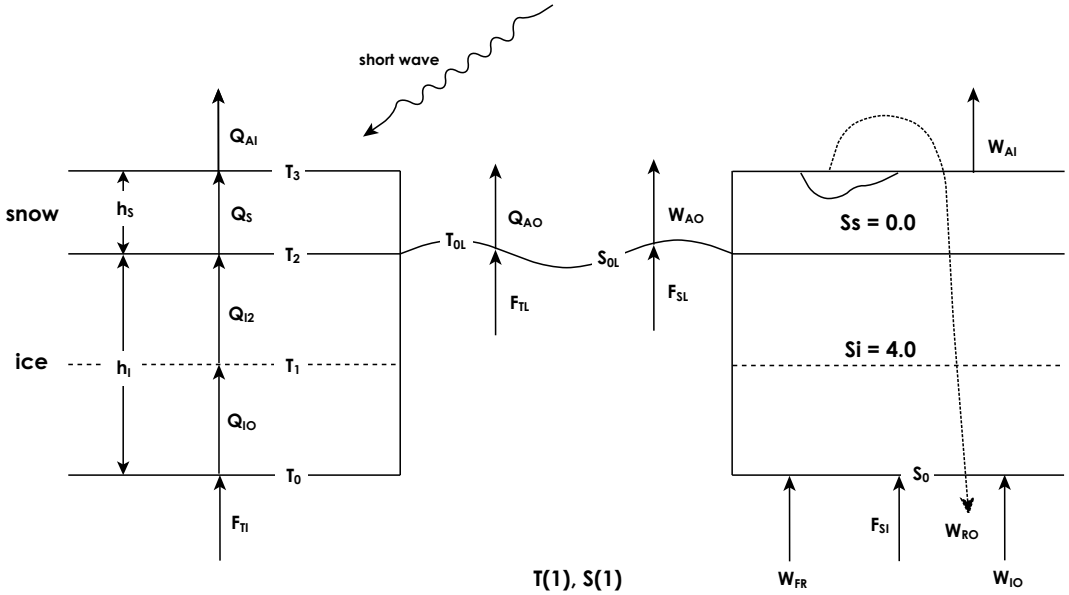

Figure 9.1. Meaning of symbols and their locations. The left side is related to heat flux, and the right side is related to fresh water flux. Sea ice is separated into the part that originated from sea water (thickness: h_I and salinity: S_I = 4.0 [psu]) and the part that originated from snow (thickness: h_s). The former is further divided into upper and lower halves. Sea ice thus 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_{TI}. 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 9.1.

9.2 Thermodynamic processes

In considering thermodynamics, thermal energy of sea ice should be defined. The basis of energy (i.e., zero energy) is defined here as that of sea water at 0°C. The thermal energy (enthalpy; $E(T,r)$) of sea ice that has temperature $T(<0)$°C and brine (salt water) fraction r is the negative of the energy needed to raise the temperature to 0°C and melt all of it:

$$E(T,r) = r(C_{po}T) + (1-r)(-L_F + C_{pi}T), \tag{9.3}$$

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 brine 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.

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.

9.2.1 Formation of new sea ice

a. From a sea water below the freezing point temperature

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:

$$\rho_o \Delta z_1 [C_{po} T] = \rho_I h_I \{ r(C_{po} T_{\text{freeze}}) + (1-r)(-L_F + C_{pi} T_{\text{freeze}}) \} + \rho_o (\Delta z - \rho_I h_I / \rho_o)(C_{po} T_{\text{freeze}}), \tag{9.4}$$

where $r = mS_I / T_{\text{freeze}}$. Using the above equation, we compute the thickness of the new ice:

$$h_I = \frac{\rho_o \Delta z}{\rho_I (1-r)} \frac{C_{po}(T_{\text{freeze}} - T)}{L_F + T_{\text{freeze}}(C_{po} - C_{pi})}. \tag{9.5}$$

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., $h_I > 10[\text{cm}]$), $h_I^{\text{new}} = h_I$ and $A = 1$.
- For a grid cell where sea ice already exists, h_I is added to each category and open water.

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 an ocean 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 9.1.

b. Input of iceberg

Input of iceberg (F_{iceberg}) is given as a water mass flux per unit area ($\text{kg}\,\text{m}^{-2}\,\text{s}^{-1}$). The thickness of the new ice (h_I) is give by

$$h_I = F_{\text{iceberg}} \Delta t / \rho_I, \tag{9.6}$$

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.

9.2.2 Air-ice interface

a. Heat flux at the upper surface of ice (Q_{AI})

The surface heat flux (Q_{AI}) is expressed as follows:

$$Q_{AI} = Q_{SI} + Q_{LI} - (1-\alpha_I) SW - LW + \varepsilon_I \sigma (T_3 + 273.16)^4. \tag{9.7}$$

We will now examine each component.

i) Short wave

The downward short wave 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 the more sophisticated albedo scheme of the Los Alamos sea ice model (CICE; Hunke and Lipscomb, 2006) by choosing model option `CALALBSI`. We briefly describe how the incoming short wave radiation is treated by CICE. The downward short wave 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 9.2 for the specific value of i_0.
- The part penetrating into the ice ($= (1-\alpha_I)i_0 SW$) 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 is absorbed into the ocean.

The albedos and penetration coefficients of CICE are listed on Table 9.2. The property f_{snow} is the snow fraction of the upper surface of the ice, which is expressed as follows:

$$f_{\text{snow}} = \frac{h_s}{h_s + h_{snowpatch}}, \tag{9.8}$$

where h_s is the snow thickness and $h_{snowpatch} = 0.02$m.

If the ice thickness (h_I) is less than $h_{\text{ref}} = 0.5$[m], the albedo of thin ice is computed as

$$\alpha_{thinice} = \alpha_o + \beta(\alpha_{coldice} - \alpha_o), \tag{9.9}$$

where

$$\beta = \frac{\arctan(a_r h_I)}{\arctan(a_r h_{\text{ref}})}, \quad a_r = 4.0, \tag{9.10}$$

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

$$\alpha_{meltice} = \alpha_{thinice} - \gamma_i(T_3 + 1.0), \tag{9.11}$$

$$\alpha_{meltsnow} = \alpha_{coldsnow} - \gamma_s(T_3 + 1.0), \tag{9.12}$$

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

$$\alpha_i = \alpha_{meltice}(1 - f_{\text{snow}}) + \alpha_{meltsnow} f_{\text{snow}}. \tag{9.13}$$

The albedos for visible and near infra-red wave lengths are computed separately. If the short wave 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.

ii) Long wave

The downward long wave radiation from the atmosphere is represented by LW in (9.7). The black body radiation from the sea surface is $\varepsilon_I \sigma (T_3 + 273.16)^4$, where ε_I is emissivity, and σ is the Stefan-Boltzmann constant. Hereinafter, we use $LW_I = LW - \varepsilon_I \sigma (T_3 + 273.16)^4$ as the net longwave radiation.

iii) Sensible heat flux

Table 9.2. Albedo and surface transparency of the albedo scheme of CICE.

	near infra-red ($>$ 700nm)	visible ($<$ 700nm)
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 short wave flux through the ice surface (i_0)	0.0	$0.7 \times (1.0 - f_{\rm snow})$

The sensible heat flux (Q_{SI} in (9.7)) is computed using a bulk formula:

$$Q_{SI} = \rho_a C_{pa} C_{HAI} U_{10} (T_3 - T_A), \tag{9.14}$$

where ρ_a is the density of air, C_{pa} is the specific heat of air, C_{HAI} is the bulk transfer coefficient at the air-ice interface (Section 8.6), 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 9.1).

iv) Latent heat flux

The latent heat flux (Q_{LI} in (9.7)) is computed using a bulk formula:

$$Q_{LI} = \rho_a L_s C_{HAI} U_{10} (q_i - q_A), \tag{9.15}$$

where L_s is the latent heat of sublimation, q_i is the saturation humidity at the ice surface temperature T_3, and q_A is the specific humidity of air. Section 9.9.1 details a computing method for q_i.

Fresh water loss due to sublimation is computed as

$$W_{AI} = \rho_a C_{HAI} U_{10} (q_i - q_A)/\rho_o. \tag{9.16}$$

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

$$Q_S = \frac{k_s}{h_s}(T_2 - T_3), \tag{9.17}$$

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:

$$Q_{I2} = \frac{k_I}{h_I/2}(T_1 - T_2), \tag{9.18}$$

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

$$T_2 = \frac{\frac{k_s}{h_s} T_3 + \frac{k_1}{h_I/2} T_1}{\frac{k_s}{h_s} + \frac{k_1}{h_I/2}}. \tag{9.19}$$

In the lower half of the ice layer, the heat flux is computed as follows:

$$Q_{IO} = \frac{k_I}{h_I/2}(T_0 - T_1). \tag{9.20}$$

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°C for snow and mS_I [°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. A melting rate is computed as

$$W_{RO} = (Q_{AI} - Q_S)/(\rho_o L_3) \tag{9.21}$$

where $L_3 = L_F$ for snow melt and

$$L_3 \equiv [E(T_3,1) - E(T_1,r_1)] \tag{9.22}$$

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.

e. Notes

- All precipitation on sea ice is assumed to be snow.
- Melted water is assumed to run off to the ocean.
- If all pre-existing ice is melted, the residual surface heat flux Q_{AI} is added to the ice-ocean flux Q_{IO}.

f. 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 (9.7) and (9.18) into $Q_{AI} = Q_{I2}$ with $T_3 \to T_3 + \delta T_3$,

$$\begin{aligned}\frac{k_I}{h_I/2}\{T_1 - (T_3 + \delta T_3)\} &= Q_{LI}(T_3 + \delta T_3) + Q_{SI}(T_3 + \delta T_3) \\ &\quad -(1-\alpha_I)(1-i_0)SW - LW + \varepsilon_I \sigma\{(T_3 + \delta T_3) + 273.16\}^4.\end{aligned} \tag{9.23}$$

By expanding the specific heat, latent heat, and black body radiation in a Taylor series, we have,

$$\begin{aligned}\frac{k_I}{h_I/2}(T_1 - (T_3 + \delta T_3)) &= Q_{LI} + Q_{SI} + \frac{\partial Q_{LI}}{\partial T_3}\delta T_3 + \frac{\partial Q_{SI}}{\partial T_3}\delta T_3 - (1-\alpha_I)(1-i_0)SW \\ &\quad -LW_I + 4\varepsilon_I \sigma (T_3 + 273.16)^3 \delta T_3.\end{aligned} \tag{9.24}$$

Using this, we compute δT_3 and add it to T_3 to obtain a new temperature:

$$\delta T_3 = \frac{-Q_{SI} - Q_{LI} + (1-\alpha_I)(1-i_0)SW + LW_I + \frac{k_I}{h_I/2}(T_1 - T_3)}{\frac{\partial Q_{LI}}{\partial T_3} + \frac{\partial Q_{SI}}{\partial T_3} + 4\varepsilon_I \sigma (T_3 + 273.16)^3 + \frac{k_I}{h_I/2}}, \tag{9.25}$$

and

$$T_3^{new} = T_3^{old} + \delta T_3, \tag{9.26}$$

where

$$\frac{\partial Q_{SI}}{\partial T_3} = \rho_a C_{pa} C_{HAI} U_{10}, \tag{9.27}$$

$$\frac{\partial Q_{LI}}{\partial T_3} = \rho_a L_s C_{HAI} U_{10} \frac{\partial q_i}{\partial T_3}. \tag{9.28}$$

Note that the dependency of L_s on temperature (Section 9.9.1) 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 9.9.1.

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:

$$\begin{aligned} Q_{AI} &= -Q_{SI} - Q_{LI} + (1-\alpha_I)(1-i_0)SW + LW_I, \\ Q_{I2} &= \frac{k_I}{h_I/2}(T_1 - T_3^{new}), \\ L_3 &= [E(T_3^{new}, 1) - E(T_1, r_1)], \\ \Delta h_I &= \frac{(Q_{AI} - Q_{I2})\Delta t}{\rho_i L_3} = W_{RO}\Delta t. \end{aligned} \tag{9.29}$$

If there is snow, the above procedure is performed for the snow surface ($T_3^{new} = 0[^\circ\mathrm{C}]$, $L_3 = L_F$). If all the snow melts away, the residual heat (E_{res}) is used to melt the ice:

$$\begin{aligned} E_{res} &= (Q_{I2} - Q_{AI})\Delta t - h_s \rho_s L_F, \\ L_3 &= [E(mS_I, 1) - E(T_1, r_1)], \\ \Delta h_I &= -E_{res}/\rho_i L_3 = W_{RO}\Delta t. \end{aligned} \tag{9.30}$$

9.2.3 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:

$$\rho_I h_I \Big[\frac{\partial}{\partial t} E(T_1, r_1) + u_{Ii} \frac{\partial}{\partial x_i} E(T_1, r_1) \Big] = Q_{IO} - Q_{I2} + [SW_{surface} - SW_{bottom}], \tag{9.31}$$

where

$$SW_{surface} = (1-\alpha_I) i_0 SW, \tag{9.32}$$

$$SW_{bottom} = (1-\alpha_I) i_0 SW \times \exp(-\kappa_i h_I). \tag{9.33}$$

The above equation can be solved explicitly without causing serious problems when the time step is not too long.

9.2.4 Ice-ocean interface

Melting and freezing at the ice-ocean interface is computed using heat fluxes at the interface as depicted in Figure 9.1. The ice-covered area and the open water are treated separately. The solution method slightly differs from that of Mellor and Kantha (1989).

In open water, the heat flux on the air side of the air-ocean interface (Q_{AO}) is computed as in Chapter 8,

$$Q_{AO}^{all} = Q_{SO}(T) + Q_{LO}(T) - (1-\alpha_o)SW - LW + \varepsilon_o \sigma (T + 273.16)^4. \tag{9.34}$$

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, short wave radiation is assumed to pass through the skin layer without absorption and is excluded from the evaluation of the freezing rate in open water:

$$Q_{AO} = Q_{SO} + Q_{LO} - LW + \varepsilon_o \sigma (T + 273.16)^4. \tag{9.35}$$

This operation causes the short wave 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, some 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.

In the ice-covered area, the heat flux on the ice side of the ice-ocean interface (Q_{IO}) is computed according to (9.20).

Melting and freezing occur due to the imbalance between fluxes above and below the interface:

$$F_{T_I} = Q_{IO} - W_{IO}\rho_o L_o, \tag{9.36}$$

$$F_{T_L} = Q_{AO} - W_{AO}\rho_o L_o, \tag{9.37}$$

where

$$L_o \equiv [E(T_0, 1) - E(T_1, r_1)] (= L_F). \tag{9.38}$$

The heat flux that drives the first level of the ocean model is given by

$$F_T = (AQ_{IO} + (1-A)Q_{AO}) - W_O \rho_o L_o, \tag{9.39}$$

where

$$W_O \equiv AW_{IO} + (1-A)W_{AO}. \tag{9.40}$$

The flux balance for salt is written as follows:

$$F_{S_I} = W_{IO}(S_I - S), \tag{9.41}$$

$$F_{S_L} = W_{AO}(S_I - S). \tag{9.42}$$

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). By doing so, the equation to solve for S_0 becomes linear. 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 salinity flux caused by melting and freezing at the ocean surface that drives the first level of the ocean model is given by

$$F_S = (AW_{IO} + (1-A)W_{AO})(S_I - S). \tag{9.43}$$

Note that the effect of melting at the upper surface of the ice could also be included in the driving salt flux for the ocean model:

$$F_S{}' = -A\{W_{ROice}(S_I - S) - W_{ROsnow}S\}. \tag{9.44}$$

However, fluxes on the oceanic side of the interface (F_{T_I}, F_{T_L}, F_{S_I}, and F_{S_L}) could be obtained as the boundary conditions ($z \to 0$) for the molecular boundary layer:

$$F_{T_I}/(\rho_o C_{po}) = -C_{T_z}(T_{0I} - T), \tag{9.45}$$

$$F_{T_L}/(\rho_o C_{po}) = -C_{T_z}(T_{0L} - T), \tag{9.46}$$

and

$$F_{S_I} = -C_{S_z}(S_{0I} - S), \tag{9.47}$$

$$F_{S_L} = -C_{S_z}(S_{0L} - S), \tag{9.48}$$

where

$$C_{T_z} = \frac{u_\tau}{(P_{rt}k^{-1}\ln(-z/z_0) + B_T)}, \tag{9.49}$$

$u_\tau \equiv (\tau_{IO_x}^2 + \tau_{IO_y}^2)^{1/4}\rho_o^{-1/2}$ is the friction velocity, $k = 0.4$ is von Karman's constant, z_0 is the roughness parameter, (τ_{IOx}, τ_{IOy}) is the stress vector at the ocean-ice interface, and

$$B_T = b\Big(\frac{z_0 u_\tau}{\nu}\Big)^{1/2} Pr^{2/3}, \tag{9.50}$$

with $Pr \equiv \nu/\alpha_t = 12.9$. The specific values for other parameters are given in Section 9.9.2.

Parameters related to salinity are

$$C_{S_z} = \frac{u_\tau}{(P_{rt}k^{-1}\ln(-z/z_0) + B_S)}, \tag{9.51}$$

and

$$B_S = b\Big(\frac{z_0 u_\tau}{\nu}\Big)^{1/2} Sc^{2/3}, \tag{9.52}$$

where $Sc \equiv \nu/\alpha_b = 2432$.

Roughness parameter z_0 is computed as follows:

$$\ln z_0 = A\ln z_{0I} + (1-A)\ln z_{0L}, \tag{9.53}$$

where

$$z_{0I} = 0.05\frac{h_I}{h_{I\mathrm{lim}}}, \quad h_{I\mathrm{lim}} = 3.0\,[\mathrm{m}], \tag{9.54}$$

and

$$z_{0L} = 0.016\,\frac{\rho_o}{\rho_a}\,\frac{u_\tau^2}{g}. \tag{9.55}$$

The roughness parameter below ice (9.54) is also used for open water in MRI.COM.

The above equations are solved simultaneously to obtain melting and freezing temperatures at the upper surface of the ocean under the following constraint:

$$\begin{aligned} W_O &= 0 \quad & A &= 0 \\ T_0 &= mS_0 \quad & A &> 0, \end{aligned} \tag{9.56}$$

where m defines the freezing line as a function of salinity.

Solving procedure

We first solve for S_{0I} and S_{0L} using (9.36), (9.41), (9.45), (9.47), (9.37), (9.42), (9.46), and (9.48):

$$S_{0I} = \frac{C_{S_z}S + (\rho_o C_{po}C_{T_z}T - Q_{IO})(S_I - S)/\rho_o L_o}{C_{S_z} + \rho_0 C_{po}C_{T_z}m(S_I - S)/\rho_o L_o}, \tag{9.57}$$

$$S_{0L} = \frac{C_{S_z}S + (\rho_o C_{po}C_{T_z}T - Q_{AO})(S_I - S)/\rho_o L_o}{C_{S_z} + \rho_o C_{po}C_{T_z}m(S_I - S)/\rho_o L_o}. \tag{9.58}$$

Using S_{0I} and S_{0L}, T_{0I} and T_{0L} are computed from (9.56), F_{T_I} and F_{T_L} are computed from (9.45) and (9.46), F_{S_I} and F_{S_L} are computed from (9.47) and (9.48), and finally W_{IO} and W_{AO} are computed from (9.36) and (9.37). Since W_{AO} is positive (only freezing is allowed in open water), $W_{AO} = 0$ when $W_{AO} < 0$. In this case, $S_{0_L} = S$ from (9.42) and (9.48). Furthermore, $F_{T_L} = Q_{AO}$ in (9.37) 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 heat and salt fluxes are computed from (9.39) and (9.43).

$z \to 0$:

$$-\kappa_V \frac{\partial T}{\partial z} = F_T, \tag{9.59}$$

$$-\kappa_V \frac{\partial S}{\partial z} = F_S, \tag{9.60}$$

where κ_V is the vertical diffusivity for the surface level of the ocean model.

If $-W_{IO}\Delta t > h_I$, 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 computed by

$$h_I = -\frac{\rho_o}{\rho_I} W_{IO}\Delta t \tag{9.61}$$

and F_{T_I} is obtained as

$$F_{T_I} = Q_{IO} - W_{IO}\rho_o L_o. \tag{9.62}$$

When $-W_{IO}\Delta t < h_I$, 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 $-(1-\Psi)W_{IO}\Delta t$ is used to reduce the thickness holding the fractional area. The new thickness is $h_I^{'}$ and $\Psi W_{IO}\Delta t$ is used to reduce compactness holding the thickness:

$$A' = A(1 + \Psi W_{IO}\Delta t / h_I^{'}). \tag{9.63}$$

It does not seem that there is a widely accepted parameterization scheme for edge melting. According to Steele (1992), bottom and top melting are dominant processes, and thus $\Psi \sim 0.1$ is usually used in MRI.COM.

9.2.5 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:

$$\rho_I h_I + \rho_s h_s = \rho_o h_I. \tag{9.64}$$

At the end of the time step, the above equality is checked, and if

$$\frac{\rho_I h_I + \rho_s h_s}{\rho_0} > h_I, \tag{9.65}$$

then the new ice thickness is set as:

$$h_I^{\text{new}} = \frac{\rho_I h_I + \rho_s h_s}{\rho_o}. \tag{9.66}$$

Since the change in snow thickness is

$$\delta h_s = -\frac{(h_I^{\text{new}} - h_I)\rho_I}{\rho_s}, \tag{9.67}$$

the new snow thickness is obtained as follows,

$$h_s^{\text{new}} = h_s + \delta h_s. \tag{9.68}$$

Since the salinity of sea ice is $S_I = 4.0$ [psu], salt is removed from the first layer of the ocean model:

$$S^{\text{new}} = S^{\text{old}} - S_I \frac{\delta h_s \rho_s}{\Delta z_1 \rho_o}. \tag{9.69}$$

9.3 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 (9.1):

$$\frac{\partial g}{\partial t} = \frac{\partial}{\partial h_I}(fg). \tag{9.70}$$

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 at the upper category boundary H_n is obtained by linear interpolation:

$$F_n = f_n + \frac{f_{n+1} - f_n}{h_{n+1} - h_n}(H_n - h_n). \tag{9.71}$$

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

$$H_n^* = H_n + F_n \Delta t. \tag{9.72}$$

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 and volume:

$$\int_{H_L}^{H_R} g dh = a_n, \tag{9.73}$$

$$\int_{H_L}^{H_R} hg dh = v_n. \tag{9.74}$$

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 (9.73) and (9.74) to yield

$$g_1 \frac{\eta_R^2}{2} + g_0 \eta_R = a_n, \tag{9.75}$$

$$g_1 \frac{\eta_R^3}{3} + g_0 \frac{\eta_R^2}{2} = a_n \eta_n, \tag{9.76}$$

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

$$g_0 = \frac{6a_n}{\eta_R^2}\left(\frac{2\eta_R}{3} - \eta_n\right), \tag{9.77}$$

$$g_1 = \frac{12a_n}{\eta_R^3}\left(\eta_n - \frac{\eta_R}{2}\right). \tag{9.78}$$

The values of the thickness distribution function at category boundaries are given as follows

$$g(0) = \frac{6a_n}{\eta_R^2}\Big(\frac{2\eta_R}{3} - \eta_n\Big), \tag{9.79}$$

$$g(\eta_R) = \frac{6a_n}{\eta_R^2}\Big(\eta_n - \frac{\eta_R}{3}\Big). \tag{9.80}$$

Equation (9.79) gives $g(0) < 0$ when the thickness is in the right third of the thickness range or $\eta_n > 2\eta_R/3$. Equation (9.80) 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 $H_C = 3h_n - H_L$ and g is set to zero for $[H_C,\, H_R]$. In this case, $\eta_R = H_C - H_L$ in (9.77) and (9.78). 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 (9.77) and (9.78).

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

$$\Delta a_n = \int_{H_n}^{H_n^*} g dh \tag{9.81}$$

and

$$\Delta v_n = \int_{H_n}^{H_n^*} hg dh. \tag{9.82}$$

If $H_n^* < H_n$, ice is transferred from category $n+1$ to n. The transferred area Δa_n and volume Δv_n are

$$\Delta a_n = \int_{H_n^*}^{H_n} g dh \tag{9.83}$$

and

$$\Delta v_n = \int_{H_n^*}^{H_n} hg dh. \tag{9.84}$$

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

$$\Delta a_0 = \int_0^{\Delta h_0} g dh. \tag{9.85}$$

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 $H_N < h$.

9.4 Dynamics

9.4.1 Momentum equation for ice pack

The momentum equation for an ice pack with mass $\rho_I A h_I$ is

$$\rho_I \frac{\partial}{\partial t}(A h_I u_I) - \rho_I A h_I f v_I = -\rho_I A h_I g \frac{1}{h_\mu}\frac{\partial h}{\partial \mu} + F_\mu(\sigma) + A(\tau_{AIx} + \tau_{IOx}), \tag{9.86}$$

$$\rho_I \frac{\partial}{\partial t}(A h_I v_I) + \rho_I A h_I f u_I = -\rho_I A h_I g \frac{1}{h_\psi}\frac{\partial h}{\partial \psi} + F_\psi(\sigma) + A(\tau_{AIy} + \tau_{IOy}), \tag{9.87}$$

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 τ_{AI} and τ_{IO} are stresses exerted by the atmosphere and ocean.

9.4.2 Stresses at top and bottom

The stress at the top is wind stress:

$$\tau_{AI} = c_a \rho_a |\mathbf{U}_a - \mathbf{u}_I|[(\mathbf{U}_a - \mathbf{u}_I)\cos\theta_a + \mathbf{k} \times (\mathbf{U}_a - \mathbf{u}_I)\sin\theta_a], \tag{9.88}$$

where $\mathbf{U}_a$ is the surface wind vector, c_a 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.

Stress at the bottom is ocean stress:

$$\tau_{IO} = c_w \rho_o |\mathbf{U}_w - \mathbf{u}_I|[(\mathbf{U}_w - \mathbf{u}_I)\cos\theta_o + \mathbf{k} \times (\mathbf{U}_w - \mathbf{u}_I)\sin\theta_o], \tag{9.89}$$

where $\mathbf{U}_w$ is the velocity of the first level of the ocean model, c_w 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.

9.4.3 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 Ducowicz (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 media 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

$$\frac{1}{E}\frac{\partial \sigma_{ij}}{\partial t} + \frac{1}{2\eta}\sigma_{ij} + \frac{\eta - \zeta}{4\eta\zeta}\sigma_{kk}\delta_{ij} + \frac{P}{4\zeta}\delta_{ij} = \dot{\varepsilon}_{ij}, \quad i,j = 1,2, \tag{9.90}$$

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. ($\dot{\varepsilon}_{ij}$) is the strain rate tensor, expressed in Cartesian coordinate as:

$$\dot{\varepsilon}_{ij} = \frac{1}{2}\Big(\frac{\partial u_{Ii}}{\partial x_j} + \frac{\partial u_{Ij}}{\partial x_i}\Big). \tag{9.91}$$

The divergence, tension, and shear of the strain rate are defined as follows:

$$D_D = \dot{\varepsilon}_{11} + \dot{\varepsilon}_{22},\ D_T = \dot{\varepsilon}_{11} - \dot{\varepsilon}_{22},\ D_S = 2\dot{\varepsilon}_{12}. \tag{9.92}$$

The equation for the stress tensor for $\sigma_1 = \sigma_{11} + \sigma_{22}$ and $\sigma_2 = \sigma_{11} - \sigma_{22}$ is given by

$$\frac{1}{E}\frac{\partial \sigma_1}{\partial t} + \frac{\sigma_1}{2\zeta} + \frac{P}{2\zeta} = D_D, \tag{9.93}$$

$$\frac{1}{E}\frac{\partial \sigma_2}{\partial t} + \frac{\sigma_2}{2\eta} = D_T, \tag{9.94}$$

$$\frac{1}{E}\frac{\partial \sigma_{12}}{\partial t} + \frac{\sigma_{12}}{2\eta} = \frac{1}{2}D_S. \tag{9.95}$$

In generalized orthogonal coordinates, divergence, tension, and shear of the strain rate are expressed by

$$D_D = \frac{1}{h_\mu h_\psi}\Big[\frac{\partial (h_\psi u)}{\partial \mu} + \frac{\partial (h_\mu v)}{\partial \psi}\Big], \tag{9.96}$$

$$D_T = \frac{h_\psi}{h_\mu}\frac{\partial}{\partial \mu}\Big(\frac{u}{h_\psi}\Big) - \frac{h_\mu}{h_\psi}\frac{\partial}{\partial \psi}\Big(\frac{v}{h_\mu}\Big), \tag{9.97}$$

$$D_S = \frac{h_\mu}{h_\psi}\frac{\partial}{\partial \psi}\Big(\frac{u}{h_\mu}\Big) + \frac{h_\psi}{h_\mu}\frac{\partial}{\partial \mu}\Big(\frac{v}{h_\psi}\Big). \tag{9.98}$$

The internal stress is obtained as the divergence of the internal stress tensor,

$$F_\mu = \frac{1}{2}\Big[\frac{1}{h_\mu}\frac{\partial \sigma_1}{\partial \mu} + \frac{1}{h_\mu h_\psi^2}\frac{\partial (h_\psi^2 \sigma_2)}{\partial \mu} + \frac{2}{h_\mu^2 h_\psi}\frac{\partial}{\partial \psi}\big(h_\mu^2 \sigma_{12}\big)\Big], \tag{9.99}$$

$$F_\psi = \frac{1}{2}\Big[\frac{1}{h_\psi}\frac{\partial \sigma_1}{\partial \psi} - \frac{1}{h_\mu^2 h_\psi}\frac{\partial (h_\mu^2 \sigma_2)}{\partial \psi} + \frac{2}{h_\mu h_\psi^2}\frac{\partial}{\partial \mu}\big(h_\psi^2 \sigma_{12}\big)\Big]. \tag{9.100}$$

$$\tag{9.101}$$

The viscous parameters are obtained from the concentration and velocity as follows:

$$\zeta = \frac{P}{2\Delta}, \tag{9.102}$$

$$\eta = \frac{P}{2e^2\Delta}, \tag{9.103}$$

$$\Delta = \Big[D_D^2 + \frac{1}{e^2}(D_T^2 + D_S^2)\Big]^{1/2}. \tag{9.104}$$

The pressure of the ice is a function of ice concentration and thickness:

$$P = P^* A h_I e \exp[-c^*(1-A)], \tag{9.105}$$

where P^* is the scaling factor for pressure, c^* is a parameter that defines the dependency on concentration, and e is the axis ratio of the elliptic yield curve ($e = 2$).

The elastic parameter E is given by

$$E = \frac{2E_o \rho_I A h_I}{\Delta t_e^2}\min(\Delta x^2, \Delta y^2), \tag{9.106}$$

where E_o is a tuning factor that satisfies $0 < E_o < 1$, Δt_e is the time step for ice dynamics, and Δx^2 and Δy^2 are the zonal and meridional grid widths.

9.4.4 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 non-ice area:

$$\nu_V \Big(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z} \Big) \Big|_{k=0} = -\frac{A}{\rho_o} (\tau_{IOx}, \tau_{IOy}) + \frac{(1-A)}{\rho_o} (\tau_{AOx}, \tau_{AOy}). \tag{9.107}$$

Note that (τ_{IOx}, τ_{IOy}) is reversed in sign because it is defined by (9.89) as the stress on the ice.

9.4.5 Solution procedure

Given the surface wind vector and the surface velocity of the ocean needed to compute surface stresses, the momentum equations ((9.86) and (9.87)) and the equations for stress tensors ((9.93), (9.94), and (9.95)) are solved.

First, the stress tensor is computed using the equations for stress tensors, the momentum equation is then solved using the stress tensor. Basically, the implicit method is used for prognostic variables for each equation. For example, stress tensor σ_1 is solved for σ_1^{m+1} as follows:

$$\frac{1}{E} \frac{\sigma_1^{m+1} - \sigma_1^m}{\Delta t} + \frac{\sigma_1^{m+1}}{2\zeta^m} + \frac{P}{2\zeta^m} = D_D^m. \tag{9.108}$$

Note that strain rate tensors and viscous parameters are updated every time step using a new velocity.

The momentum equations are solved using σ^{m+1} above:

$$\begin{aligned} \rho_I A h_I \frac{u_I^{m+1} - u_I^m}{\Delta t} &= \rho_I A h_I f v_I^{m+1} - \rho_I A h_I g \frac{1}{h_\mu} \frac{\partial h}{\partial \mu} + F_\mu(\sigma^{m+1}) + A\tau_{AIx} \\ &\quad + A c_w \rho_o |\mathbf{U}_w - \mathbf{u}_I^m| [(U_w - u_I^{m+1}) \cos\theta_o + (V_w - v_I^{m+1}) \sin\theta_o], \end{aligned} \tag{9.109}$$

$$\begin{aligned} \rho_I A h_I \frac{v_I^{m+1} - v_I^m}{\Delta t} &= -\rho_I A h_I f u_I^{m+1} - \rho_I A h_I g \frac{1}{h_\psi} \frac{\partial h}{\partial \psi} + F_\psi(\sigma^{m+1}) + A\tau_{AIy} \\ &\quad + A c_w \rho_o |\mathbf{U}_w - \mathbf{u}_I^m| [(V_w - v_I^{m+1}) \cos\theta_o - (U_w - u_I^{m+1}) \sin\theta_o]. \end{aligned} \tag{9.110}$$

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,

$$\tau_{IO} = c_w \rho_o |\mathbf{U}_w^{n-1} - \mathbf{u}_I^{m+1}| [(\mathbf{U}_w^{n-1} - \mathbf{u}_I^{m+1}) \cos\theta_o + \mathbf{k} \times (\mathbf{U}_w^{n-1} - \mathbf{u}_I^{m+1}) \sin\theta_o]. \tag{9.111}$$

The time step of the ice dynamics is limited by the phase speed of the elastic wave. To damp the elastic waves during the subcycle, the ice dynamics is subcycled several tens of steps during one ocean model time step.

9.5 Advection

Fractional area, snow volume, ice volume, ice energy, and ice surface temperature (optional; set `flg_advec_tskin` to be `.true.` in `mod_seaice_cat.F90`) of each category are advected. A multidimensional positive definite advection transport algorithm (MPDATA; Smolarkiewicz, 1984) is used.

The advection equation for a property (α) is given by

$$\frac{\partial \alpha}{\partial t} + \mathscr{A}(\alpha) = 0, \tag{9.112}$$

where $\mathscr{A}$ is an advection operator defined as

$$\mathscr{A}(\alpha) = \frac{1}{h_\mu h_\psi}\left\{\frac{\partial (h_\psi u\alpha)}{\partial \mu} + \frac{\partial (h_\mu v\alpha)}{\partial \psi}\right\}. \tag{9.113}$$

The specific representation for α is A for fractional area, Ah_s for snow volume, Ah_I for ice volume, and Ah_IE for ice energy.

In MPDATA, (9.112) is first solved to obtain a temporary value using the upstream scheme with a mid-point velocity between time levels n and $n+1$. Using this temporary value, an anti-diffusive velocity is computed as

$$\begin{aligned}
\tilde{u}_{i+\frac{1}{2},j} &= \frac{1}{\alpha^*}\left[\frac{1}{2}\left(|u^{n+\frac{1}{2}}|\Delta x - \Delta t(u^{n+\frac{1}{2}})^2\right)\frac{\partial \alpha^*}{\partial x} - \frac{1}{2}\Delta t u^{n+\frac{1}{2}} v^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial y}\right] \\
&\quad -\frac{1}{2}\Delta t u^{n+\frac{1}{2}}\left(\frac{\partial u^{n+\frac{1}{2}}}{\partial x} + \frac{\partial v^{n+\frac{1}{2}}}{\partial y}\right), \qquad (9.114) \\
\tilde{v}_{i,j+\frac{1}{2}} &= \frac{1}{\alpha^*}\left[\frac{1}{2}\left(|v^{n+\frac{1}{2}}|\Delta y - \Delta t(v^{n+\frac{1}{2}})^2\right)\frac{\partial \alpha^*}{\partial y} - \frac{1}{2}\Delta t u^{n+\frac{1}{2}} v^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial x}\right] \\
&\quad -\frac{1}{2}\Delta t v^{n+\frac{1}{2}}\left(\frac{\partial u^{n+\frac{1}{2}}}{\partial x} + \frac{\partial v^{n+\frac{1}{2}}}{\partial y}\right). \qquad (9.115)
\end{aligned}$$

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.

9.6 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:

$$\begin{aligned}
b(h) &= \frac{2}{G^*}\left(1 - \frac{G(h)}{G^*}\right) \quad \text{if } G(h) < G^* \qquad (9.116) \\
&= 0 \quad \text{otherwise}, \qquad (9.117)
\end{aligned}$$

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 for a range $[H_{n-1},\, H_n]$ as

$$a_{P_n} = \frac{2}{G^*}(G_n - G_{n-1})\left(1 - \frac{G_{n-1} - G_n}{2G^*}\right). \tag{9.118}$$

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 $a_0 > G^*$, 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:

$$f_m^{\text{area}} = \frac{H_R - H_L}{H_{\max} - H_{\min}}, \tag{9.119}$$

where $H_L = \max(H_{m-1}, H_{\min})$ and $H_R = \min(H_m, H_{\max})$. The fractional volume that is distributed in category m is:

$$f_m^{\text{vol}} = \frac{(H_R)^2 - (H_L)^2}{(H_{\max})^2 - (H_{\min})^2}. \tag{9.120}$$

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

$$R_{\text{net}} = \frac{C_s}{2}(\Delta - |D_D|) - \min(D_D, 0), \tag{9.121}$$

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:

$$R_{\text{net}} = \left[a_{P_0} + \sum_{n=1}^{N} a_{P_n}\left(1 - \frac{1}{k_n}\right)\right] R_{\text{tot}}. \tag{9.122}$$

Since R_{net} is computed from (9.121), R_{tot} is computed from (9.122). 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.

9.7 Discretization

9.7.1 Advection (MPDATA)

In MPDATA, tracer (α in Figure 9.2) is updated following a three-step procedure.

1. A temporary value is computed using an upstream scheme.

The tracer fluxes at the side boundaries of the unit grid cell where the tracer is defined are:

$$\begin{aligned} F_x(\alpha_{i,j}^n, \alpha_{i+1,j}^n, u_{i+\frac{1}{2},j}^n) &= \left[\alpha_{i,j}^n (u_{i+\frac{1}{2},j}^n + |u_{i+\frac{1}{2},j}^n|) + \alpha_{i+1,j}^n (u_{i+\frac{1}{2},j}^n - |u_{i+\frac{1}{2},j}^n|)\right] \frac{\Delta y}{2}, \\ F_y(\alpha_{i,j}^n, \alpha_{i,j+1}^n, v_{i,j+\frac{1}{2}}^n) &= \left[\alpha_{i,j}^n (v_{i,j+\frac{1}{2}}^n + |v_{i,j+\frac{1}{2}}^n|) + \alpha_{i,j+1}^n (v_{i,j+\frac{1}{2}}^n - |v_{i,j+\frac{1}{2}}^n|)\right] \frac{\Delta x}{2}, \end{aligned} \tag{9.123}$$

where

$$u_{i+\frac{1}{2},j} = \frac{1}{2}\Big(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}}\Big), \tag{9.124}$$

$$v_{i,j+\frac{1}{2}} = \frac{1}{2}\Big(v_{i+\frac{1}{2},j+\frac{1}{2}} + v_{i-\frac{1}{2},j+\frac{1}{2}}\Big). \tag{9.125}$$

The zonal flux is defined at the closed circle and the meridional flux is defined at the closed square in Figure 9.2. Using this, the temporary value (α^*) is computed using an upstream scheme:

$$\begin{aligned}\frac{(\alpha^*_{i,j}-\alpha^n_{i,j})\Delta S_{i,j}}{\Delta t} =& F_x(\alpha^n_{i-1,j},\alpha^n_{i,j},u^{n+\frac{1}{2}}_{i-\frac{1}{2},j}) - F_x(\alpha^n_{i,j},\alpha^n_{i+1,j},u^{n+\frac{1}{2}}_{i+\frac{1}{2},j})\\ &+ F_y(\alpha^n_{i,j-1},\alpha^n_{i,j},v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}) - F_y(\alpha^n_{i,j},\alpha^n_{i,j+1},v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}).\end{aligned} \tag{9.126}$$

2. Compute the anti-diffusive transport velocity.

Using the temporary value computed in the first step, the anti-diffusive transport velocity is computed as follows:

$$\begin{aligned}\tilde{u}_{i+\frac{1}{2},j} &= \frac{1}{2}\left[\frac{1}{\alpha^{*(2)}_{i+\frac{1}{2},j}}\left(|u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}|\Delta x - \Delta t(u^{n+\frac{1}{2}}_{i+\frac{1}{2},j})^2\right)\left(\frac{\partial\alpha^*}{\partial x}\right)_{i+\frac{1}{2},j} - \frac{1}{\alpha^{*(6)}_{i+\frac{1}{2},j}}\Delta t u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\left(v^{n+\frac{1}{2}}\frac{\partial\alpha^*}{\partial y}\right)_{i+\frac{1}{2},j}\right]\\ &\quad -\frac{1}{2}\Delta t u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i+\frac{1}{2},j},\end{aligned} \tag{9.127}$$

$$\begin{aligned}\tilde{v}_{i,j+\frac{1}{2}} &= \frac{1}{2}\left[\frac{1}{\alpha^{*(2)}_{i,j+\frac{1}{2}}}\left(|v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}|\Delta y - \Delta t(v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}})^2\right)\left(\frac{\partial\alpha^*}{\partial y}\right)_{i,j+\frac{1}{2}} - \frac{1}{\alpha^{*(6)}_{i,j+\frac{1}{2}}}\Delta t v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(u^{n+\frac{1}{2}}\frac{\partial\alpha^*}{\partial x}\right)_{i,j+\frac{1}{2}}\right]\\ &\quad -\frac{1}{2}\Delta t v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j+\frac{1}{2}},\end{aligned} \tag{9.128}$$

where

$$\alpha^{*(2)}_{i+\frac{1}{2},j} = \frac{1}{2}(\alpha^*_{i,j}+\alpha^*_{i+1,j}), \tag{9.129}$$

$$\alpha^{*(2)}_{i,j+\frac{1}{2}} = \frac{1}{2}(\alpha^*_{i,j}+\alpha^*_{i,j+1}), \tag{9.130}$$

$$\alpha^{*(6)}_{i+\frac{1}{2},j} = \frac{1}{6}(\alpha^*_{i,j-1}+\alpha^*_{i+1,j-1}+2\alpha^*_{i,j}+2\alpha^*_{i+1,j}+\alpha^*_{i,j+1}+\alpha^*_{i+1,j+1}), \tag{9.131}$$

$$\alpha^{*(6)}_{i,j+\frac{1}{2}} = \frac{1}{6}(\alpha^*_{i-1,j}+\alpha^*_{i-1,j+1}+2\alpha^*_{i,j}+2\alpha^*_{i,j+1}+\alpha^*_{i+1,j}+\alpha^*_{i+1,j+1}), \tag{9.132}$$

$$\left(\frac{\partial\alpha^*}{\partial x}\right)_{i+\frac{1}{2},j} = \frac{\alpha^*_{i+1,j}-\alpha^*_{i,j}}{\Delta x}, \tag{9.133}$$

$$\left(\frac{\partial\alpha^*}{\partial y}\right)_{i+\frac{1}{2},j} = \frac{\alpha^*_{i,j+1}-\alpha^*_{i,j}}{\Delta y}, \tag{9.134}$$

$$\begin{aligned}\left(v^{n+\frac{1}{2}}\frac{\partial\alpha^*}{\partial y}\right)_{i+\frac{1}{2},j} =& \frac{1}{4}\Big[v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(\frac{\partial\alpha^*}{\partial y}\right)_{i,j+\frac{1}{2}} + v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\left(\frac{\partial\alpha^*}{\partial y}\right)_{i,j-\frac{1}{2}}\\ &+ v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(\frac{\partial\alpha^*}{\partial y}\right)_{i+1,j+\frac{1}{2}} + v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\left(\frac{\partial\alpha^*}{\partial y}\right)_{i+1,j-\frac{1}{2}},\Big]\end{aligned} \tag{9.135}$$

$$\begin{aligned}\left(u^{n+\frac{1}{2}}\frac{\partial\alpha^*}{\partial x}\right)_{i,j+\frac{1}{2}} =& \frac{1}{4}\Big[u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\left(\frac{\partial\alpha^*}{\partial x}\right)_{i+\frac{1}{2},j} + u^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\left(\frac{\partial\alpha^*}{\partial x}\right)_{i-\frac{1}{2},j}\\ &+ u^{n+\frac{1}{2}}_{i+\frac{1}{2},j+1}\left(\frac{\partial\alpha^*}{\partial x}\right)_{i+\frac{1}{2},j+1} + u^{n+\frac{1}{2}}_{i-\frac{1}{2},j+1}\left(\frac{\partial\alpha^*}{\partial x}\right)_{i-\frac{1}{2},j+1}\Big],\end{aligned} \tag{9.136}$$

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n}_{i,j}=\frac{\left(u^{n}_{i+\frac{1}{2},j}\Delta y_{i+\frac{1}{2},j}-u^{n}_{i-\frac{1}{2},j}\Delta y_{i-\frac{1}{2},j}+v^{n}_{i,j+\frac{1}{2}}\Delta x_{i,j+\frac{1}{2}}-v^{n}_{i,j-\frac{1}{2}}\Delta x_{i,j-\frac{1}{2}}\right)}{\Delta S_{i,j}}, \tag{9.137}$$

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = \frac{1}{2}\left[\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j}+\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i+1,j}\right], \tag{9.138}$$

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} = \frac{1}{2}\left[\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j}+\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j+1}\right]. \tag{9.139}$$

3. Update tracer using the anti-diffusive velocity starting from the temporary value.

$$\begin{aligned}\frac{(\alpha^{n+1}_{i,j}-\alpha^{*}_{i,j})\Delta S_{i,j}}{\Delta t} =& F_x(\alpha^{*}_{i-1,j},\alpha^{*}_{i,j},\tilde{u}_{i-\frac{1}{2},j})-F_x(\alpha^{*}_{i,j},\alpha^{*}_{i+1,j},\tilde{u}_{i+\frac{1}{2},j})\\ &+F_y(\alpha^{*}_{i,j-1},\alpha^{*}_{i,j},\tilde{v}_{i,j-\frac{1}{2}})-F_y(\alpha^{*}_{i,j},\alpha^{*}_{i,j+1},\tilde{v}_{i,j+\frac{1}{2}}).\end{aligned} \tag{9.140}$$

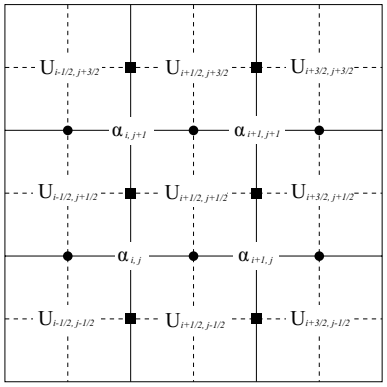

Figure 9.2. Position for tracer (α) and velocity (U). Area and thickness are defined at tracer points. Zonal fluxes are computed at closed circles and meridional fluxes are computed at closed squares. The budget is computed for a unit grid cell for α.

9.7.2 Momentum equation

Specific forms of discretization for properties related to internal stress are given here.

The strain rate tensor ($\dot{\varepsilon}$) and stress tensor (σ) are defined at tracer points (Figure 9.3).

Components (divergence, tension, and shear) of the strain rate tensor are expressed in a discretized form as follows:

$$\begin{aligned}(D_D)_{i,j} =& \frac{1}{\Delta x_{i,j}\Delta y_{i,j}}\Big(\frac{\Delta y_{i+\frac{1}{2},j}}{2}(u_{I_{i+\frac{1}{2},j+\frac{1}{2}}}+u_{I_{i+\frac{1}{2},j-\frac{1}{2}}})-\frac{\Delta y_{i-\frac{1}{2},j}}{2}(u_{I_{i-\frac{1}{2},j+\frac{1}{2}}}+u_{I_{i-\frac{1}{2},j-\frac{1}{2}}})\\ &+\frac{\Delta x_{i,j+\frac{1}{2}}}{2}(v_{I_{i+\frac{1}{2},j+\frac{1}{2}}}+v_{I_{i-\frac{1}{2},j+\frac{1}{2}}})-\frac{\Delta x_{i,j-\frac{1}{2}}}{2}(v_{I_{i+\frac{1}{2},j-\frac{1}{2}}}+v_{I_{i-\frac{1}{2},j-\frac{1}{2}}})\Big),\end{aligned}$$

$$
\begin{aligned}
(D_T)_{i,j} &= \frac{1}{2}\Big[\frac{\Delta y_{i,j+\frac{1}{2}}}{\Delta x_{i,j+\frac{1}{2}}}\Big(\frac{u_{I_{i+\frac{1}{2},j+\frac{1}{2}}}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}-\frac{u_{I_{i-\frac{1}{2},j+\frac{1}{2}}}}{\Delta y_{i-\frac{1}{2},j+\frac{1}{2}}}\Big)+\frac{\Delta y_{i,j-\frac{1}{2}}}{\Delta x_{i,j-\frac{1}{2}}}\Big(\frac{u_{I_{i+\frac{1}{2},j-\frac{1}{2}}}}{\Delta y_{i+\frac{1}{2},j-\frac{1}{2}}}-\frac{u_{I_{i-\frac{1}{2},j-\frac{1}{2}}}}{\Delta y_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big]\\
&\quad -\frac{1}{2}\Big[\frac{\Delta x_{i+\frac{1}{2},j}}{\Delta y_{i+\frac{1}{2},j}}\Big(\frac{v_{I_{i+\frac{1}{2},j+\frac{1}{2}}}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}-\frac{v_{I_{i+\frac{1}{2},j-\frac{1}{2}}}}{\Delta x_{i+\frac{1}{2},j-\frac{1}{2}}}\Big)+\frac{\Delta x_{i,j-\frac{1}{2}}}{\Delta y_{i,j-\frac{1}{2}}}\Big(\frac{v_{I_{i-\frac{1}{2},j+\frac{1}{2}}}}{\Delta x_{i-\frac{1}{2},j+\frac{1}{2}}}-\frac{v_{I_{i-\frac{1}{2},j-\frac{1}{2}}}}{\Delta x_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big],
\end{aligned}
$$

$$
\begin{aligned}
(D_S)_{i,j} &= \frac{1}{2}\Big[\frac{\Delta y_{i,j+\frac{1}{2}}}{\Delta x_{i,j+\frac{1}{2}}}\Big(\frac{v_{I_{i+\frac{1}{2},j+\frac{1}{2}}}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}-\frac{v_{I_{i-\frac{1}{2},j+\frac{1}{2}}}}{\Delta y_{i-\frac{1}{2},j+\frac{1}{2}}}\Big)+\frac{\Delta y_{i,j-\frac{1}{2}}}{\Delta x_{i,j-\frac{1}{2}}}\Big(\frac{v_{I_{i+\frac{1}{2},j-\frac{1}{2}}}}{\Delta y_{i+\frac{1}{2},j-\frac{1}{2}}}-\frac{v_{I_{i-\frac{1}{2},j-\frac{1}{2}}}}{\Delta y_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big]\\
&\quad +\frac{1}{2}\Big[\frac{\Delta x_{i+\frac{1}{2},j}}{\Delta y_{i+\frac{1}{2},j}}\Big(\frac{u_{I_{i+\frac{1}{2},j+\frac{1}{2}}}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}-\frac{u_{I_{i+\frac{1}{2},j-\frac{1}{2}}}}{\Delta x_{i+\frac{1}{2},j-\frac{1}{2}}}\Big)+\frac{\Delta x_{i,j-\frac{1}{2}}}{\Delta y_{i,j-\frac{1}{2}}}\Big(\frac{u_{I_{i-\frac{1}{2},j+\frac{1}{2}}}}{\Delta x_{i-\frac{1}{2},j+\frac{1}{2}}}-\frac{u_{I_{i-\frac{1}{2},j-\frac{1}{2}}}}{\Delta x_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big].
\end{aligned}
$$

The internal stress is defined at velocity points and computed from stress tensor as follows:

$$
\begin{aligned}
(F_\mu)_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{2}\Big[\frac{1}{2}\Big(\frac{(\sigma_1)_{i+1,j+1}+(\sigma_1)_{i+1,j}-(\sigma_1)_{i,j+1}-(\sigma_1)_{i,j}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\Big)\\
&\quad +\frac{1}{2}\Big(\frac{\Delta y^2_{i+1,j+\frac{1}{2}}[(\sigma_2)_{i+1,j+1}+(\sigma_2)_{i+1,j}]-\Delta y^2_{i,j+\frac{1}{2}}[(\sigma_2)_{i,j+1}+(\sigma_2)_{i,j}]}{\Delta y^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\Big)\\
&\quad +\Big(\frac{\Delta x^2_{i+\frac{1}{2},j+1}[(\sigma_{12})_{i+1,j+1}+(\sigma_{12})_{i,j+1}]-\Delta x^2_{i+\frac{1}{2},j}[(\sigma_{12})_{i+1,j}+(\sigma_{12})_{i,j}]}{\Delta x^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}\Big),
\end{aligned}
\tag{9.141}
$$

$$
\begin{aligned}
(F_\psi)_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{2}\Big[\frac{1}{2}\Big(\frac{(\sigma_1)_{i+1,j+1}+(\sigma_1)_{i,j+1}-(\sigma_1)_{i+1,j}-(\sigma_1)_{i,j}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}\Big)\\
&\quad -\frac{1}{2}\Big(\frac{\Delta x^2_{i+\frac{1}{2},j+1}[(\sigma_2)_{i+1,j+1}+(\sigma_2)_{i,j+1}]-\Delta x^2_{i+\frac{1}{2},j}[(\sigma_2)_{i+1,j}+(\sigma_2)_{i,j}]}{\Delta x^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}\Big)\\
&\quad +\Big(\frac{\Delta y^2_{i+1,j+\frac{1}{2}}[(\sigma_{12})_{i+1,j+1}+(\sigma_{12})_{i+1,j}]-\Delta y^2_{i,j+\frac{1}{2}}[(\sigma_{12})_{i,j+1}+(\sigma_{12})_{i,j}]}{\Delta y^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\Big).
\end{aligned}
\tag{9.142}
$$

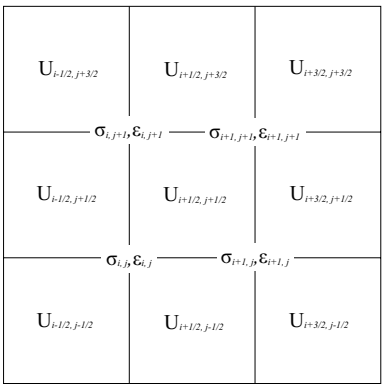

Figure 9.3. Position of variables used by dynamics scheme.

9.8 Technical issues

9.8.1 Source codes

The sea ice part consists of the following programs.

ice_bulk.F90: computes fluxes at the air-ice interface

ice_cat_albedo.F90: computes surface albedo and fraction of short wave penetrating into ice

ice_cat_com.F90: defines variables and sets parameters

ice_cat_bulk.F90: computes surface fluxes using bulk formula

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

The ice model could only be used as a part of the ocean model. Among the subroutines included in the above programs, only `si_initialize` and `simain` in `ice_main_cat.F90` are called from the ocean model. `si_initialize` initializes the ice model by reading parameters, creating grid cells, and reading restart, which is called from the subroutine `ogcm__ini` of `ogcm.F90`. `simain` is the main program and calls subroutines of the ice model, which is called from the subroutine `mkflux` of `mkflux.F90`.

To use the ice part, the ocean model options `ICE`, `SIDYN`, `ICECAT`, should be selected. If `SIDYN` is not selected, dynamics is not solved and ice drifts with a third of the surface ocean velocity. By selecting `CALALBSI`, the albedo scheme of CICE is used. Otherwise, the constant albedo is used.

Other things to be noted are:

- The ice model uses the forward scheme in time integration. The ice model is not called in the backward part of the Matsuno (Euler backward) scheme.
- The output of history and restart for the category integrated/averaged state is automatically done at the same time as the ocean model for separate files (Chapter 16).

9.8.2 Coupling with an atmospheric model

In a coupled mode, 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). To use all the heat fluxes given by the atmospheric component, fluxes are globally adjusted at each subcycle time step of the oceanic component to account for changes in ice area during the subcycle. See Yukimoto et al. (2010) for details.

In the atmospheric component, temperature in the atmospheric boundary layer and at the ice surface (T_3(`tsfcin`)) are computed along with heat flux in snow layer ($Q_S = Q_{I2}$ (`fheatu`)) using ice temperature (T_1(`t1icen`)), snow thickness (h_s(`hsnwn`)) and ice thickness (h_I(`hicen`)) given by the oceanic component.

All the informations needed by the oceanic component are received via `cgcm_scup__get_a2o` in `ogcm.F90` at the beginning of the coupling cycle. The information needed by the ice part is extracted via `get_fluxi_a2o` in `ice_flux.F90`. The main part of the ice is solved using surface fluxes and ice surface temperature (optional) from the atmospheric component. The properties needed by the atmospheric component are sent via `cgcm_scup__put_o2a` in `ogcm.F90` at the end of the coupling cycle.

9.8.3 Job parameters (namelist)

The run-time job parameters (namelist) are listed on Table 9.3.

Table 9.3. namelist for the ice model

variable name	group	description	usage
file_ice_restart_in	infli	restart file of average for input	used only in OGCM
file_ice_restart_out_temp	outflir	restart file of average for output	used only in OGCM
file_ice_hist_temp	outflih	history file of average for output	used only in OGCM
irstrt	njobpsi	set initial state	1:read restart, 0: start without ice
nstepi	njobpsi	time steps to be proceed	same as OGCM if not specified
int_bgtice	njobpsi	the interval of time step by which water budget is written	0 for no output
ibyri	njobpsi	start year of this series of integration	same as OGCM if not specified
ibmni	njobpsi	start month of this series of integration	same as OGCM if not specified
ibdyi	njobpsi	start day of this series of integration	same as OGCM if not specified
adtdi	njobidyn	time step interval for dynamics (in minutes)	about a tenth of the baroclinic time step of the ocean model
hbound(0:ncat)	njobpscat	category boundary	
lsicat_volchk	njobpscat	flag for checking mass conservation	.false. by default
num_hint_ic	nhsticint	the number of time step intervals of history output of categorized ice	not exceed 3

variable name	group	description	usage
maxnum_hist_ic	nhsticfile	maximum histories allowed to be written to one file	given for each num_hint_ic (same for below)
nwrt_hist_ic	nhsticfile	the interval of time step for history output	the number of time steps or -1 for monthly output
imin_hist_ic	nhsticfile	western end for history output	
imax_hist_ic	nhsticfile	eastern end for history output	
jmin_hist_ic	nhsticfile	southern end for history output	
jmax_hist_ic	nhsticfile	northern end for history output	
file_ice_hist_ic_temp	nhsticfile	core part of the file name for history	
file_icecat_restart_in	inflic	restart file name of categorized ice for input	
num_rst_ic	outflic	the number of files allowed to be output	
maxnum_rst_ic	outflic	the maximum snap shots allowed to be written to one file	
nwrt_rst_ic	outflic	the interval of time step for snap shot output	the number of time steps or -1 for monthly output
file_icecat_restart_out_temp	outflic	core part of the file name for snap shot	
alb_ice_visible_t0	njobalbsi	visible ice albedo for thicker ice	0.78
alb_ice_nearIR_t0	njobalbsi	near infrared ice albedo for thicker ice	0.36
alb_snw_visible_t0	njobalbsi	visible, cold snow albedo	0.98
alb_snw_nearIR_t0	njobalbsi	near infrared, cold snow albedo	0.70
alb_ice_visible_dec_ratio	njobalbsi	visible ice albedo declination rate	0.075 [$^\circ C^{-1}$]
alb_ice_nearIR_dec_ratio	njobalbsi	near infrared ice albedo declination rate	0.075 [$^\circ C^{-1}$]
alb_snw_visible_dec_ratio	njobalbsi	visible snow albedo declination rate	0.10 [$^\circ C^{-1}$]
alb_snw_nearIR_dec_ratio	njobalbsi	near infrared ice albedo declination rate	0.15 [$^\circ C^{-1}$]
hi_ref	njobalbsi	the maximum ice thickness to which connection function is used	0.50 [m]
atan_ref	njobalbsi	the base value of the tangent hyperbolic connection function	4.0
tsfci_t0	njobalbsi	the temperature at which ice albedo is equated to that of ocean	0.0 [°C]
tsfci_t1	njobalbsi	the temperature at which ice albedo is started to decline to that of the ocean	-1.0 [°C]

variable name	group	description	usage
fsnow_patch	njobalbsi	area fraction of snow on melting bare ice	0.02

9.9 Appendix

9.9.1 Saturation water vapor pressure and latent heat

The latent heat of sublimation and the saturation specific humidity is computed according to Appendix 4 of Gill (1982) as follows.

The saturation vapor pressure e_{w} (in units [hPa]) of pure water vapor over a plane water surface is given by

$$\log_{10} e_{\mathrm{w}}(t) = (0.7859 + 0.03477t)/(1 + 0.00412t). \tag{9.143}$$

In air, the partial pressure $e_{\mathrm{w}}^{'}$ of water vapor at saturation is not exactly e_{w} but is given by

$$e_{\mathrm{w}}^{'} = f_{\mathrm{w}} e_{\mathrm{w}}. \tag{9.144}$$

The value of f_{w} is given by

$$f_{\mathrm{w}} = 1 + 10^{-6} p(4.5 + 0.0006t^2), \tag{9.145}$$

where p is the pressure (in units [hPa]).

The saturation vapor pressure e_{i} of pure water vapor over ice is given by,

$$\log_{10} e_{\mathrm{i}}(t) = \log_{10} e_{\mathrm{w}}^{'}(t) + 0.00422t. \tag{9.146}$$

The current version does not include this correction. Thus, $e_{\mathrm{i}}^{'} = e_{\mathrm{w}}^{'}$.

The saturation specific humidity q is by definition:

$$e_{\mathrm{i}}^{'}/p_s = q/(\varepsilon + (1-\varepsilon)q), \tag{9.147}$$

where ε is the molecular weight ratio between water vapor and air:

$$\varepsilon = m_{\mathrm{w}}/m_{\mathrm{a}} = 18.016/28.966 = 0.62197, \tag{9.148}$$

and p_s is sea level pressure (units in [hPa]).

Using this, the saturation specific humidity is given as

$$q = \varepsilon e_{\mathrm{i}}^{'}/(p_s - (1-\varepsilon)e_{\mathrm{i}}^{'}). \tag{9.149}$$

The latent heat of sublimation is given by

$$L_s = 2.839 \times 10^6 - 3.6(T_3 + 35)^2 \,\mathrm{Jkg}^{-1}. \tag{9.150}$$

Surface temperature of ice is computed using semi-implicit method, where an expression for the partial derivative of the saturation specific humidity by temperature is needed as in (9.28).

First, using equation (9.149),

$$\frac{\partial q_i}{\partial t} = \frac{\varepsilon p_s}{\{p_s - (1-\varepsilon)e_{\mathrm{i}}^{'}\}^2} \frac{\partial e_{\mathrm{i}}^{'}}{\partial t}, \tag{9.151}$$

where $\partial e_{\mathrm{i}}^{'}/\partial t$ is expressed by setting $f_w = 1$ as

$$\frac{\partial e_{\mathrm{i}}^{'}}{\partial t} = \ln 10 \cdot 10^{g(t)} \cdot g^{'}(t), \tag{9.152}$$

where $g(t) = (0.7859 + 0.03477t)/(1 + 0.00412t)$ and $g^{'}(t) = \partial g^{'}(t)/\partial t$.

9.9.2 Physical constant, parameters

Since the ice part is coded in SI units, constants and parameters are written in SI units.

Thermodynamics

parameter	value	variable name in MRI.COM
Thermal ice conductivity	$k_I = 2.04\,\mathrm{J\,m^{-1}\,s^{-1}K^{-1}}$	CKI
Thermal snow conductivity	$k_s = 0.31\,\mathrm{J\,m^{-1}\,s^{-1}K^{-1}}$	CKS
Specific heat of sea water	$C_{po} = 3990\,\mathrm{J\,kg^{-1}\,K^{-1}}$	CP0
Specific heat of air	$C_{pa} = 1004.67\,\mathrm{J\,kg^{-1}\,K^{-1}}$	CPAIR
Specific heat of ice	$C_{pI} = 2093\,\mathrm{J\,kg^{-1}\,K^{-1}}$	CPI
Specific heat of snow	$C_{ps} = 0.0\,\mathrm{J\,kg^{-1}\,K^{-1}}$	—
Stefan Boltzmann constant	$\sigma = 5.67 \times 10^{-8}\,\mathrm{W\,m^{-2}\,K^{-4}}$	STBL
Albedo of open ocean surface	$\alpha_o = 0.1$	ALBW
Albedo of ice	$\alpha_I = 0.6$	ALBI
Albedo of snow	$\alpha_s = 0.75$	ALBS
Emissivity of ocean surface	$\varepsilon_o = 0.97$	EEW
Emissivity of ice surface	$\varepsilon_I = 1.0$	EEI
Emissivity of snow surface	$\varepsilon_s = 1.0$	EES
bulk transfer coefficient	$C_{HAI} = 1.5 \times 10^{-3}$	CHAI
Latent heat of fusion	$L_F = 3.347 \times 10^5\,\mathrm{J\,kg^{-1}}$	ALF
Latent heat of sublimation	equation (9.150)	RLTH
constants for fusion phase	$m = -0.0543\,\mathrm{K/ppt}$	XMXM
equation: $T_f = mS + nz$	$n = -0.000759\,\mathrm{K\,m^{-1}}$	XNXN
Ice roughness parameter	$z_{oI} = 0.05 h_I/3$	Z0
Salinity of sea ice	$S_I = 4.0\,\mathrm{psu}$	SI
von Karman's constant	$k = 0.4$	XK
Thickness/compactness diffusion of ice	$\kappa_H = 1.0 \times 10^3\,\mathrm{m^2\,s^{-1}}$	AKH
Scaling factor for κ_H	North : 3.0, South : 3.0	FKHDN, FKHDS
Seawater kinematic viscosity	$\nu = 1.8 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$	ANU
Seawater heat diffusivity	$\alpha_t = 1.39 \times 10^{-7}\,\mathrm{m^2\,s^{-1}}$	AT
Seawater salinity diffusivity	$\alpha_b = 6.8 \times 10^{-10}\,\mathrm{m^2\,s^{-1}}$	AS
Turbulent Prandtl number	$P_{rt} = 0.85$	PRT
b in eqs (9.50),(9.52)	$b = 3.14$	AB

Dynamics

parameter	value	variable name in MRI.COM
Reference water density	$\rho_o = 1036\,\mathrm{kg\,m^{-3}}$	RO
Reference air density	$\rho_a = 1.205\,\mathrm{kg\,m^{-3}}$	ROAIR
Reference ice density	$\rho_I = 900\,\mathrm{kg\,m^{-3}}$	RICE
Reference snow density (ratio between snow and water)	$\rho_s = 330\,\mathrm{kg\,m^{-3}}$	RDSW
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)	$c_a = 1.5 \times 10^{-3}$	CDRGAI
drag coefficient (ice-ocean)	$c_w = 5.5 \times 10^{-3}$	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 northern/southern hemisphere)	WIANGL
air turning angle	θ_a	–

References

Gill, A. E., 1982: *Atmosphere-Ocean Dynamics*, Academic Press, 662pp.

Hibler, W. D., III, 1979: A dynamic thermodynamic sea ice model, *J. Phys. Oceanogr.*, *9*, 815–846.

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. Weather Rev.*, *130*, 1848–1865.

Hunke, E. C, and J. K. Dukowicz, 1997: An Elastic-Viscous-Plastic Model for Sea Ice Dynamics, *J. Phys. Oceanogr.*, *94*, 1849–1867.

Hunke, E. C, and W. H. Lipscomb, 2006: *CICE: the Los Alamos Sea Ice Model Documentation and Software User's Manual*, available at http://climate.lanl.gov/source/projects/climate/Models/CICE/index.shtml, 59pp.

Mellor, G. L., and L. Kantha, 1989: An Ice-Ocean Coupled Model, *J. Geophys. Res.*, *94*, 10,937–10,954.

Smolarkiewicz, P. K., 1984: A fully multidimensional positive definite advection transport algorism with small implicit diffusion, *J. Comput. Phys.*, *54*, 325–362.

Steele, M., 1992: Sea ice melting and floe geometry in a simple ice-ocean model, *J. Geohys. Res.*, *97*, 17,729–17,738.

Yoshimura, H., and S. Yukimoto, 2008: Development of simple couplar (Scup) for earth sysyem modeling, *Pap. Meteor. Geophys.*, *59*, 19-29, doi:10.2467/mripapers.59.19.

Yukimoto, S., and coauthors, 2010: *Meteorological Research Institute-Earth System Model v1 (MRI-ESM1)*, Technical Reports of the Meteorological Research Institute, No.64, in press.

Chapter 10 Bottom Boundary Layer (BBL)

MRI.COM has the option of adopting a simple bottom boundary layer model (the option BBL). This chapter describe the formulation and usage of this model.

10.1 General description

In general, z-coordinate models cannot properly reproduce either dense overflows from the Nordic-Scotland-Greenland ridges to the Atlantic oceans or the dense downslope flows from the continental slope around Antarctica. 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 world ocean tend to exhibit a warming bias. To mitigate these deficiencies, MRI.COM incorporates a simple bottom boundary layer (BBL) model, used in Nakano and Suginohara (2002). This BBL model lies at the bottom of the normal ocean grid along the bottom topography. The following components of 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.

10.2 Grid arrangement

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 the z-coordinate model (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 diffusion with the surrounding waters. The incorporation of BBL remedies these problems.

The BBL grid cells in MRI.COM are arranged as in Figure 10.1. The grid cells in MRI.COM are composed of U-grid cells, but it is intuitively easier to understand the grid arrangement of BBL as if they are located at T-grid cells. This is because the depth where the pressure gradient is evaluated is defined as the average depth between the neighboring BBL cells. (In Figure 10.1a, the pressure gradient used to calculate the velocity between the BBLs with different depths is evaluated at the averaged depth between these BBLs.) Assume that the number of the vertical grids, `km`, is 50 before the BBL option is applied. When the BBL option is applied, the number of vertical grids, `km`, becomes 51 (50 + 1). The master BBL cells are set at the lowest level of each array regardless of the actual depths of the BBL cells. This arrangement is suitable for calculating the horizontal advection between the BBL cells. In addition, to easily express the interaction between the BBL cells and the inner ocean cells, we place

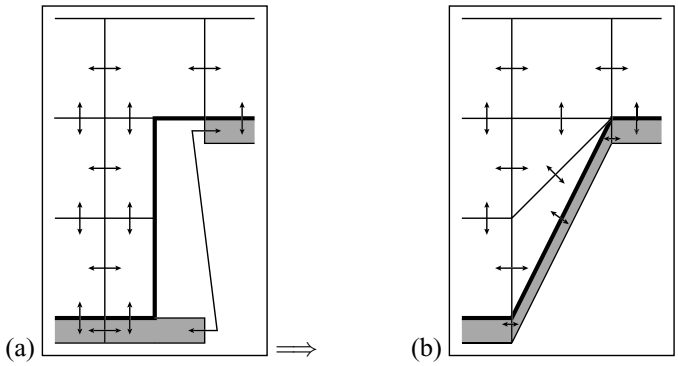

Figure 10.1. (a) Arrangement of BBL(T-box). (b) Schematic arrangement that is identical to (a).

a dummy BBL cell in each vertical column just below the bottom grid of the inner cell and copy the temperature, salinity, and velocities from the master BBL cell.

10.3 Pressure gradient terms

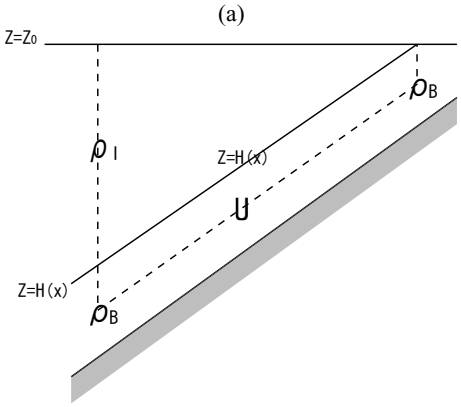

Figure 10.2. Evaluation of horizontal pressure gradient.

Consider the case in which water in the inner grids is uniform, and there is a dense layer along the bottom slope. From the physical point of view, the horizontal pressure gradient should move the dense water downward along the slope in the non-rotational frame. However, in the original MRI.COM code without the BBL option, the pressure gradient is zero in the lowermost inner cells along the bottom slope. In the BBL cells, the horizontal pressure gradient is calculated as follows when the slope is smooth:

$$\left.\frac{\partial p}{\partial x}\right|_{z=const} = \left.\frac{\partial p}{\partial x}\right|_{z=H(x)} + \frac{\partial p}{\partial z}\frac{\partial H}{\partial x}, \tag{10.1}$$

$$= \left.\frac{\partial p}{\partial x}\right|_{z=H(x)} - g\,\rho|_{z=H(x)}\,\frac{\partial H}{\partial x}. \tag{10.2}$$

In the above case, the density of the water in the inner cells is uniform ($\rho = \rho_I$), and there is a dense layer ($\rho = \rho_B$)

along the bottom slope. The horizontal pressure gradient becomes

$$\left.\frac{\partial p}{\partial x}\right|_{z=H(x)} - g\rho_B \frac{\partial H}{\partial x} = \frac{\partial [p_{z=z_0} + gH(x)\rho_I]}{\partial x} - g\rho_B \frac{\partial H}{\partial x}, \tag{10.3}$$

$$= \left.\frac{\partial p}{\partial x}\right|_{z=z_0} - g(\rho_B - \rho_I)\frac{\partial H}{\partial x}, \tag{10.4}$$

where $p_{z=z_0}$ is the the pressure at $z = z_0$ in Figure 10.2. The second term on the r.h.s. of Eq. (10.4) represents the effect of the slope. The larger the difference in the density or the steeper the slope of the topography, the larger the horizontal pressure gradient becomes. When there is no difference in the density between the inner cells and BBL cells, the pressure gradient is zero regardless of the slope of the bottom topography.

10.4 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. Jian 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:

$$-fv = -\frac{\partial p}{\partial x} - \alpha u, \tag{10.5}$$

$$fu = -\frac{\partial p}{\partial y} - \alpha v, \tag{10.6}$$

After some algebra we obtain

$$-f\left(v - \frac{\alpha}{f}u\right) = -fv' = -\frac{\partial p}{\partial x}, \tag{10.7}$$

$$f\left(u + \frac{\alpha}{f}v\right) = fu' = -\frac{\partial p}{\partial y}, \tag{10.8}$$

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 the 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.

10.5 Usage

In this section, we show how to use the BBL model in the MRI.COM.

- Add the 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 available).
- Set the lowest layer of `dz.F90` as the depth of BBL. Fifty to one hundred meters is recommended for the depth of BBL, which roughly corresponds to the observed thickness of the BBL layer near the Denmark Strait.
- Add `ho4bbl`, `exnnbbl` to `file_topo` after `ho4`, `exnn`. 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 available. Thus `exnnbbl` should be equal to or less than one.
- Set the area where the Rayleigh drag coefficient is applied. The default setting is 2500m in the North Atlantic and 4000 m around the Southern Ocean.

In the default setting, the temperature and salinity in the BBL cells are set to those in the cells k=`km`. When the option `INILEV` is used, the temperature and salinity in the BBL cells are set to those in the lowermost inner cells.

Example of `configure.in` with the BBL option

```
DEFAULT_OPTIONS="OMIP FREESURFACE UTOPIA ULTIMATE ZQUICKEST ZULTIMATE
CYCLIC ISOPYCNAL MELYAM HFLUX WFLUX RUNOFF CLMFRC HIST HISTFLUX ICE
SIDYN INILEV BBL"
IMUT=184
JMUT=171
KM=48
KSGM=1
KBBL=1
SLAT0=-84.D0
SLON0=0.D0
DXTDGC=2.D0
DYTDGC=1.D0
ITMSC=0
ITMSCB=0
ISRSTB=NSDAY
NPARTA=4
NPLAT=75.D0
NPLON=320.D0
```

The above example includes `BBL` in `DEFAULT_OPTIONS` and `KBBL` = 1.

A sample Fortran program of the topography data follows.

A Fortran program to create topography data with BBL

```
      integer(4) :: ho4(imut,jmut),exnn(imut,jmut)
      integer(4) :: ho4bbl(imut,jmut),exnnbbl(imut,jmut)
      write(inidt) ho4, exnn
      write(inidt) ho4bbl,exnnbbl
```

10.6 Usage notes

10.6.1 Limit of the area where BBL model should be applied

Ideally, the BBL model should be applied to the globe, and its behavior should change due to temperature, salinity, velocity, and topography. 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, we choose to apply the BBL model in areas where the BBL model is important.

10.6.2 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.

10.6.3 Notes for the program code

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 `atexl` and `aexl` is always very confusing. At the bottom cell (k=`km`), we set `atexl`=1 and `aexl`=1 while we set `atexl`=1 and `aexl`=0 at the lowest ocean cells. In the program `energy.F90`, the treatment of `atexl`=1 in the BBL model needs special care to avoid double counts.

Reference

Jiang, L., and R. W. Garwood Jr., 1996: Three-dimensional simulations of overflows on continental slopes, *J. Phys. Oceanogr.*, *26*, 1214-1233.

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.

Winton, M., R. Hallberg, and A. Gnanadesikan, 1998: Simulation of density-driven frictional flow in z-coordinate ocean models, *J. Phys. Oceanogr.*, *28*, 2163–2174.

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 calculated in 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, short wave 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 process 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 their 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.

The carbon cycle component follows the OCMIP protocols (Orr et al., 1999) whose authority is recognized in the community. 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), and they cannot be used without the carbon cycle component. The biogeochemical model of MRI.COM is largely based on Schmittner (2008) when an NPZD model is adopted as an ecosystem component.

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. The unit of mmol/m^3 is used in some models such as Oschlies (2001). 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, PO_4, and O_2) are used.

$$\frac{\partial[\mathrm{DIC}]}{\partial t} = \mathscr{A}([\mathrm{DIC}]) + \mathscr{D}([\mathrm{DIC}]) + Sb([\mathrm{DIC}]) + Jv([\mathrm{DIC}]) + Jg([\mathrm{DIC}]), \tag{11.1}$$

$$\frac{\partial[\mathrm{Alk}]}{\partial t} = \mathscr{A}([\mathrm{Alk}]) + \mathscr{D}([\mathrm{Alk}]) + Sb([\mathrm{Alk}]) + Jv([\mathrm{Alk}]), \tag{11.2}$$

$$\frac{\partial[\mathrm{PO_4}]}{\partial t} = \mathscr{A}([\mathrm{PO_4}]) + \mathscr{D}([\mathrm{PO_4}]) + Sb([\mathrm{PO_4}]), \tag{11.3}$$

$$\frac{\partial[\mathrm{O_2}]}{\partial t} = \mathscr{A}([\mathrm{O_2}]) + \mathscr{D}([\mathrm{O_2}]) + Sb([\mathrm{O_2}]) + Jg([\mathrm{O_2}]), \tag{11.4}$$

$$\frac{\partial[\mathrm{NO_3}]}{\partial t} = \mathscr{A}([\mathrm{NO_3}]) + \mathscr{D}([\mathrm{NO_3}]) + Sb([\mathrm{NO_3}]), \tag{11.5}$$

$$\frac{\partial[\mathrm{PhyPl}]}{\partial t} = \mathscr{A}([\mathrm{PhyPl}]) + \mathscr{D}([\mathrm{PhyPl}]) + Sb([\mathrm{PhyPl}]), \tag{11.6}$$

$$\frac{\partial[\mathrm{ZooPl}]}{\partial t} = \mathscr{A}([\mathrm{ZooPl}]) + \mathscr{D}([\mathrm{ZooPl}]) + Sb([\mathrm{ZooPl}]), \tag{11.7}$$

$$\frac{\partial[\mathrm{Detri}]}{\partial t} = \mathscr{A}([\mathrm{Detri}]) + \mathscr{D}([\mathrm{Detri}]) + Sb([\mathrm{Detri}]), \tag{11.8}$$

where $\mathscr{A}()$ is advection, $\mathscr{D}()$ is diffusion, and $Sb()$ is source minus sink due to the biogeochemical activities. The square brackets mean dissolved concentration in mol/m^3 of the substance within them. The terms represented by $Jg()$ and $Jv()$ are the air-sea gas fluxes at the sea surface, and they appear only in the uppermost layer. The term $Jg()$ is calculated based on the OCMIP protocol by using the air-sea gas transfer velocity and concentration in the seawater. The term $Jv()$ 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 Jg and Jv, we follow the protocol of OCMIP, which is described in detail in Najjar and Orr (1998), and Orr (1999). The program to calculate them is based on the subroutine downloaded from the OCMIP website. We have modified this subroutine so that it can be used in the vector oriented 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 $Jg([\mathrm{DIC}])$ and $Jg([O_2])$ appear only in the uppermost layer. When these fluxes are expressed as $Fg([\mathrm{DIC}])$ and $Fg([O_2])$, $Jg([\mathrm{DIC}])$ and $Jg([O_2])$ are given as follows:

$$Jg([\mathrm{DIC}]) = \frac{Fg([\mathrm{DIC}])}{\Delta z_1}, \tag{11.9}$$

$$Jg([O_2]) = \frac{Fg([O_2])}{\Delta z_1}, \tag{11.10}$$

where

$$Fg([\mathrm{DIC}]) = \mathrm{K}_w^{\mathrm{CO_2}} * ([\mathrm{CO_2}]_{\mathrm{sat}} - [\mathrm{CO_2}]_{\mathrm{surf}}), \tag{11.11}$$

$$Fg([O_2]) = \mathrm{K}_w^{\mathrm{O_2}} * ([\mathrm{O_2}]_{\mathrm{sat}} - [\mathrm{O_2}]_{\mathrm{surf}}), \tag{11.12}$$

$$[\mathrm{CO_2}]_{\mathrm{sat}} = \alpha^{\mathrm{C}} * \mathrm{pCO_2atm} * P/P_o, \tag{11.13}$$

$$[\mathrm{O_2}]_{\mathrm{sat}} = [\mathrm{O_2}]_{\mathrm{sato}} * P/P_o. \tag{11.14}$$

Here a standard gas transfer formulation is adopted. Next, we elaborate on the above equations.

Piston velocity

Parameters $\mathrm{K}_w^{\mathrm{CO_2}}$ and $\mathrm{K}_w^{\mathrm{O_2}}$ are the air-sea gas exchange transfer (Piston) velocity and are diagnosed as follows.

$$\mathrm{K}_w^{\mathrm{CO_2}} = (1-A)[X_{conv} a U_{10}^2](Sc^{\mathrm{CO_2}}/660)^{-1/2}, \tag{11.15}$$

$$\mathrm{K}_w^{\mathrm{O_2}} = (1-A)[X_{conv} a U_{10}^2](Sc^{\mathrm{O_2}}/660)^{-1/2}, \tag{11.16}$$

where

- A is the fraction of the sea surface covered with ice,
- U_{10} is 10m scalar wind speed,
- a is the coefficient of 0.337, consistent with a piston velocity in cm/hr, and specified in the OCMIP protocol,
- $X_{conv} = 1/(3.6 \times 10^5)$, is a constant factor to convert the piston velocity from [cm/hr] to [m/s],
- $Sc^{\mathrm{CO_2}}$ and $Sc^{\mathrm{O_2}}$ are the Schmidt numbers for CO_2 and O_2. They are computed using the formulation of Wannikhof (1992) for CO_2 and that of Keeling et al. (1998) for O_2.

Computing CO_2 and O_2 concentrations at the surface

The concentration of CO_2 is computed as follows.

- $[\mathrm{CO_2}]_{\mathrm{surf}}$ is diagnosed every step from [DIC],[Alk], temperature, and salinity at the surface.
- α^{C} is the solubility of CO_2, which is diagnosed from sea surface temperature and salinity.
- $[\mathrm{O_2}]_{\mathrm{sato}}$ is the saturation oxygen concentration before the variations in total pressure are taken into account and is diagnosed from the sea surface temperature and salinity.

- pCO_{2atm} is specified. (In the OCMIP climatology, it is set to 280ppm.)
- P is the sea surface pressure in units of [hPa] and $P_o = 1013.25$ [hPa] is the standard surface atmospheric pressure.

Diagnosis of [CO_2] at the surface

Diagnosis of [CO_2] at the surface is the most complex of the above calculations and has the heaviest computational burden. 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:

$$\begin{aligned} [\mathrm{DIC}] &= [CO_2] + [H_2CO_3] + [HCO_3^-] + [CO_3^{2-}] && (11.17)\\ &= [CO_2^*] \qquad\qquad + [HCO_3^-] + [CO_3^{2-}]. && (11.18) \end{aligned}$$

In the OCMIP protocol, the following equations are solved to obtain [CO_2^*].

The equilibrium expressions for dissociation are:

$$K_1 = \frac{[H^+][HCO_3^-]}{[CO_2^*]} \qquad K_2 = \frac{[H^+][CO_3^{2-}]}{[HCO_3^-]}, \tag{11.19}$$

$$K_B = \frac{[H^+][B(OH)_4^-]}{[B(OH)_3]}, \tag{11.20}$$

$$K_{1P} = \frac{[H^+][H_2PO_4^-]}{[H_3PO_4]} \qquad K_{2P} = \frac{[H^+][HPO_4^{2-}]}{[H_2PO_4^-]} \qquad K_{3P} = \frac{[H^+][PO_4^{3-}]}{[HPO_4^{2-}]}, \tag{11.21}$$

$$K_{Si} = \frac{[H^+][SiO(OH)_3^-]}{[Si(OH)_4]}, \tag{11.22}$$

$$K_W = [H^+][OH^-], \tag{11.23}$$

$$K_S = \frac{[H^+]_F[SO_4^{2-}]}{[HSO_4^-]}, \tag{11.24}$$

and

$$K_F = \frac{[H^+]_F[F^-]}{[HF]}, \tag{11.25}$$

where [H^+] is the hydrogen ion concentration in seawater 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:

$$[H^+] = [H^+]_F \left(1 + \frac{S_T}{K_S} + \frac{F_T}{K_F}\right), \tag{11.26}$$

$$\text{and} \qquad [H^+]_T = [H^+]_F \left(1 + \frac{S_T}{K_S}\right). \tag{11.27}$$

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:

$$[DIC] = [CO_2^*] + [HCO_3^-] + [CO_3^{2-}], \tag{11.28}$$

$$[B_T] = [B(OH)_3] + [B(OH)_4^-], \tag{11.29}$$

$$[P_T] = [H_3PO_4] + [H_2PO_4^-] + [HPO_4^{2-}] + [PO_4^{3-}], \tag{11.30}$$

$$[Si_T] = [Si(OH)_4] + [Si(OH)_3^-], \tag{11.31}$$

$$[S_T] = [HSO_4^-] + [SO_4^{2-}], \tag{11.32}$$

and

$$[F_T] = [HF] + [F^-]. \tag{11.33}$$

Alkalinity used in this calculation is defined as follows:

$$\begin{aligned}[Alk] \quad = \quad & [HCO_3^-] + 2[CO_3^{2-}] + [B(OH)_4^-] + [OH^-] + [HPO_4^{2-}] + 2[PO_4^{3-}] + [SiO(OH)_3^-] \\ & -[H^+]_F - [HSO_4^-] - [HF] - [H_3PO_4].\end{aligned} \tag{11.34}$$

These expressions exclude the effect 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 using the Newton-Raphson method. The concentration [Si_T] is not predicted in the biogeochemical model adopted in MRI.COM but rather is specified as a typical value of 7.68375×10^{-3}mol/m^3. (The sensitivity to [Si_T] is much less than that to other variables).

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 the option `SFLUXW` or `SFLUXR` 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

$$\texttt{sflux}(i,j) = -(P-E) * S(i,j,1)/\Delta z, \tag{11.35}$$

where $S(i,j,1)$ and Δz are the salinity and thickness of the uppermost layer. Note that the variable $\texttt{sflux}(i,j)$ 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:

$$Jv(\mathrm{DIC}(i,j,1)) \quad = \quad \texttt{sflux}(i,j)/S(i,j,1) * \mathrm{DIC}(i,j,1), \tag{11.36}$$

$$Jv(\mathrm{Alk}(i,j,1)) \quad = \quad \texttt{sflux}(i,j)/S(i,j,1) * \mathrm{Alk}(i,j,1). \tag{11.37}$$

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 OCMIP protocol, the global averaged salinity S_g is used instead of $S(i,j,1)$ in equations(11.36,11.37). In addition, globally integrated $Jv(\mathrm{DIC})$ and $Jv(\mathrm{Alk})$ are set to 0. In MRI.COM, these modifications are not the default considering the use in regional ocean models.

11.4 Obata and Kitamura model

This section was contributed by A. Obata.

The Obata and Kitamura model used in MRI.COM simply represents the source and sink terms of [DIC], [Alk], [PO_4], 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 $rL[PO4]^2/([PO4]+k)$, where r is a proportional factor ($r = 0.9[\mathrm{yr}^{-1}]$), L is the insolation normalized by the annual mean insolation on the equator, and k is the half-saturation constant ($k = 0.4[\mathrm{molkg}^{-1}]$). The values of r and k are adjusted to reproduce the optimum atmospheric CO_2 concentration and ocean biogeochemical distribution for the preindustrial state of the model. The relationship between the changes in the chemical composition of seawater and the composition of particulate organic matter (POM) is assumed to follow the Redfield ratio $P:N:C:O_2 = 1:16:106:-138$ (Redfield et al., 1963). The rain ratio of calcite to particulate organic carbon (POC) is 0.09, which is in the range proposed by Yamanaka and Tajika (1996). The surface thickness where the export production occurs is fixed at 60 m. The vertical distribution of POM and calcite vertical flux below a depth of 100 m is proportional to $(z/100\mathrm{m})^{-0.9}$ and $\exp(-z/3500\mathrm{m})$ (z is the depth in meters), respectively, following the work of Yamanaka and Tajika (1996). The remineralization of POM (RemiP for phosphorus) and the dissolution of calcite (SolnCa) at depth are parameterized by these fluxes. Oxygen saturation is prescribed at the sea surface. The solubility of oxygen is computed from the formula of Weiss (1970). Source and sink terms of Sb() representing the above processes are as follows.

$$Sb([\mathrm{DIC}]) = 106 * \mathrm{RemiP} + \mathrm{SolnCa} - 106 * \mathrm{ExprodP} \tag{11.38}$$

$$Sb([\mathrm{Alk}]) = 2 * \mathrm{SolnCa} + 16 * \mathrm{ExprodP} - 16 * \mathrm{RemiP} \tag{11.39}$$

$$Sb([\mathrm{PO_4}]) = \mathrm{RemiP} - \mathrm{ExprodP} \tag{11.40}$$

$$Sb([\mathrm{O_2}]) = -138 * Sb([\mathrm{PO_4}]) \tag{11.41}$$

11.5 NPZD model

The NPZD model used in MRI.COM is constructed on the assumptions that the biological composition ratio is nearly constant (Redfield ratio) and that the concentration of the biology can be estimated by nitrogen or phosphorus. The prognostic variables of nitrogen (NO_3), phytoplankton (PhyPl), zooplankton (ZooPl), and detritus (Detri) are normalized in terms of nitrogen 1 mol/m^3. For example, [PhyPl] represents the concentration of phytoplankton estimated in terms of nitrogen in one cubic meter (N mol /m^3). The increase and decrease of carbon can be diagnosed by multiplying by R_{cn} (ratio of C:N).

Source and sink terms Sb() calculated in the NPZD model are as follows. Those for DIC and Alk, $Sb(\mathrm{DIC})$ and

$Sb(\mathrm{Alk})$, used for calculating the carbon cycle, are described later in this section.

$$Sb([\mathrm{PhyPl}]) = \mathrm{Priprod} - \mathrm{MortP1} - \mathrm{MortP2} - \mathrm{GrP2Z} \tag{11.42}$$

$$Sb([\mathrm{ZooPl}]) = assim * \mathrm{GrP2Z} - \mathrm{Excrtn} - \mathrm{MortZ} \tag{11.43}$$

$$Sb([\mathrm{Detri}]) = [(1 - assim) * \mathrm{GrP2Z} + \mathrm{MortP2} + \mathrm{MortZ}] - \mathrm{RemiD} - w_{detri}\frac{\partial \mathrm{Detri}}{\partial z} \tag{11.44}$$

$$Sb([\mathrm{NO_3}]) = \mathrm{MortP1} + \mathrm{Excrtn} + \mathrm{RmeiD} - \mathrm{PriProd} \tag{11.45}$$

$$Sb([\mathrm{PO_4}]) = R_{pn} * Sb([\mathrm{NO_3}]) \tag{11.46}$$

$$Sb([\mathrm{O_2}]) = -R_{on} * R_{np} * Sb([\mathrm{PO_4}]) \tag{11.47}$$

There is no input from the atmosphere such as nitrogen fixation in the above equations, so the sum of these five equations becomes zero at each grid point except for the term for detritus sinking ($-w_{detri}\frac{\partial \mathrm{Detri}}{\partial z}$). The term for detritus sinking expresses the biological pump, whose role is to remove nutrients from the upper layers and transport them into the deep ocean where the plankton cannot use the nutrients. When vertically integrated, the sum of each grid is 0 even though this sinking term is included. The nutrients are transported horizontally through physical processes such as advection and diffusion.

In general, the nitrate limit is more severe than the phosphate limit 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

- $\mathrm{Priprod} = J(I,N,P) * [\mathrm{PhyPl}]$
  Primary production expresses photosynthesis (described in detail in the next subsection).

- $\mathrm{MortP1} = \phi_P * [\mathrm{PhyPl}]$
  The conversion of mortality phytoplankton directly into nutrients. This term was introduced by Oschlies (2001) to increase the primary production of subtropical gyre, where the nutrient limit is severe.

- $\mathrm{MortP2} = \phi_{PP} * [\mathrm{PhyPl}]^2$
  The conversion from phytoplankton to detritus (normal mortality of phytoplankton).

- $\mathrm{GrP2Z} = G(P) * [\mathrm{ZooPl}]^2$
  The grazing of zooplankton. There are a number of parameterizations of grazing. In this formulation, this is given as $G(P) = g * \varepsilon * [\mathrm{PhyPl}]^2/(g + \varepsilon * [\mathrm{PhyPl}]^2)$. Among the grazing, the ratio $assim$ is used for the growth of zooplankton, and the remainder $(1 - assim)$ is converted to detritus.

- $\mathrm{Excrtn} = d * [\mathrm{ZooPl}]$
  Excretion of zooplankton. The excretion is dissolute and directly returned to nutrients ($\mathrm{NO_3}$).

- $\mathrm{MortZ} = \phi_Z * [\mathrm{ZooPl}]$
  The conversion from the zooplankton to detritus (mortality of zooplankton).

- $\mathrm{RemiD} = \phi_D * [\mathrm{Detri}]$
  Remineralization of detritus. This is converted to nutrients through the activity of bacteria.

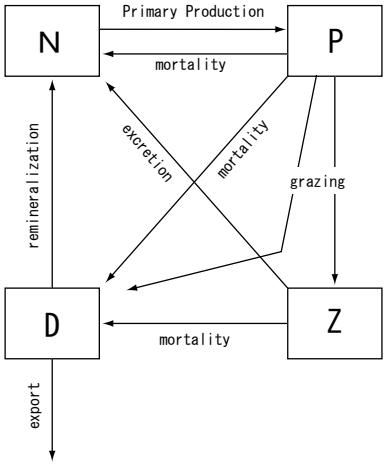

Figure 11.1. Schematic of NPZD model

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:

$$J(I,\mathrm{N},\mathrm{P}) = min\,(J_I,J_N,J_P)\,, \tag{11.48}$$

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:

$$J_I = \frac{J_{max}\alpha I}{[J_{max}^2 + (\alpha I)^2]^{1/2}}. \tag{11.49}$$

Here, J_{max} is the light-saturated growth, which depends on temperature based on Eppley (1972) as

$$J_{max} = a \cdot b^{cT}, \tag{11.50}$$

where $a = 0.6\mathrm{day}^{-1}$, $b = 1.066$, and $c = 1(^\circ\mathrm{C})^{-1}$. Note that the default values in MRI.COM are based on Schmittner (2008) and differ from these values (see Table 11.1). Equation (11.49) is called Smith-type growth. The coefficient α in the equation is "the initial slope of photosynthesis versus irradiance (P-I) curve," that is,

$$\alpha = \lim_{I\to 0}\frac{\partial J_I}{\partial I}. \tag{11.51}$$

Thus, it represents how sensitive J_I is to the irradiance when the light is weak.

Irradiance (I) depends on the angle of incidence and the refraction and absorption in the seawater.

$$I = I_{z=0}\ \mathrm{PAR}\ \exp\left(-k_w\tilde{z} - k_e\int_0^{\tilde{z}} P dz\right), \tag{11.52}$$

where $I_{z=0}$ denotes the downward shortwave radiation at the sea surface, PAR is the photosynthetically active radiation ratio (0.43) and $\tilde{z} = z/\cos\theta = z/\sqrt{1-\sin^2\theta/1.33^2}$ is the effective vertical coordinate (positive downward) with 1.33 as the refraction index according to Snell's law relating the zenith angle of incidence in air (θ) to the angle of incidence in water. The angle of incidence θ is a function of the latitude ϕ and declination δ.

For the nutrient-limited growth rate (J_N and J_P), we adopt the Optimal Uptake (OU) equation instead of the classic Michaelis-Menten (MM) equation. For the classic MM equation, the nitrate-limited growth rate is expressed as

$$J_N = J_{MM} = \frac{J_{max} N}{K_N + N}, \tag{11.53}$$

where K_N is a half-saturation constant for NO_3 uptake. In contrast, the Optimal Uptake (OU) equation for a nitrate is expressed as follows:

$$J_N = J_{OU} = \frac{V_0 N}{N + 2\sqrt{\frac{V_0}{A_0} N} + \frac{V_0}{A_0}}, \tag{11.54}$$

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_O and A_O so that the rates of uptake, J_{MM} and J_{OU}, are equal at $N = K_N$. In addition, we fix the ratio $V_0/A_0 = \alpha_{OU}$, where α_{OU} is determined from fitting the data. This requires

$$V_0 = 0.5 \left(1 + \sqrt{\frac{\alpha_{OU}}{K_N}}\right)^2 J_{max}. \tag{11.55}$$

Finally, we obtain

$$J_{OU} = \frac{V_0 N}{N + 2\sqrt{\alpha_{OU} N} + \alpha_{OU}}. \tag{11.56}$$

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 ($CaCO_3$), in molar numbers according to

$$Sb([\mathrm{DIC}]) = Sb([\mathrm{PO_4}]) R_{cp} - Sb([\mathrm{CaCO_3}]), \tag{11.57}$$

$$Sb([\mathrm{Alk}]) = -Sb([\mathrm{NO_3}]) - 2 \cdot Sb([\mathrm{CaCO_3}]). \tag{11.58}$$

Thus, only these source and sink terms of DIC and Alk are estimated. Since $[PO_4]$ and $[NO_3]$ are prognostic variables, their source and sink are explicitly calculated by the biological model. In contrast, the downward movement of $CaCO_3$ is much faster than the modeled downward velocity of water mass, so $[CaCO_3]$ is not a prognostic variable, and its source (Pr) and sink (Di) are diagnosed by the following equation,

$$Sb([\mathrm{CaCO_3}]) = Pr([\mathrm{CaCO_3}]) - Di([\mathrm{CaCO_3}]). \tag{11.59}$$

Following Schmittner et al. (2008), the source term ($Pr([\mathrm{CaCO_3}])$) of calcium carbonate is determined by the production of detritus as follows:

$$Pr([\mathrm{CaCO_3}]) = [(1 - assim) * [\mathrm{GrP2Z}] + [\mathrm{MortP2}] + [\mathrm{MortZ}]] R_{\mathrm{CaCO_3/POC}} R_{C:N}, \tag{11.60}$$

where *assim*, GrP2Z, MortP2, and MortZ are as described above. The sink term ($Di([\mathrm{CaCO_3}])$) of calcium carbonate is parameterized as

$$Di([\mathrm{CaCO_3}]) = \int Pr([\mathrm{CaCO_3}]) dz \cdot \frac{d}{dz} \left(\exp\left(-z/D_{\mathrm{CaCO_3}}\right)\right), \tag{11.61}$$

which expresses an instantaneous sinking with an e-holding depth of D_{CaCO_3}=3500m. In this equation, z is positive downward. This depth of 3500m was estimated by Yamanaka and Tajika (1996) to reproduce the observed nutrient profile. This value is standard and also is used in the simple biological model in the protocol of OCMIP. The vertical integral of the source minus sink should be zero. Thus, when the sea bottom appears before the sum becomes zero, the remaining calcium carbonate is assumed to be dissolved in the lowermost layer. By using the ratio $R_{CaCO_3/POC}$ = 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

There are three options, `CARBON`, `NPZD`, and `CBNHSTRUN`. Options `NPZD` and `CBNHSTRUN` require `CARBON`.

- When option `NPZD` is not used in `configure.in`, the simple biological model of Obata and Kitamura (2003) is applied as the ecosystem component.
  - `numtrc_p`=4 should be specified in `configure.in`. Using a different number causes the program to stop.
  - When used in the NEC super computer, the compile option of `FFLAGS` = `-pvctlexpand = 6` should be used to activate the in-line expansion. If this option is not used, the model works but quite slowly due to the low efficiency in the vectorization. The configurations in restart and history file are set in namelist `inflpt`, `outfpt`, `outfph`:

```
namelist /inflpt/ fn_ptrc_in   (1:numtrc_p+np_inout)
namelist /outfpt/ fn_ptrc_out  (1:numtrc_p+np_inout)
namelist /outfph/ fn_ptrc_hist (1:numtrc_p+np_hist)
np_inout = 2  np_hist  = 5 (when NPZD is used, this number is 6)
fn_ptrc_in  (1) : restart file for dic     (input)
            (2) : restart file for alk     (input)
            (3) : restart file for po4     (input)
            (4) : restart file for o2      (input)
            (5) : restart file for pco2o   (input)
            (6) : restart file for pco2a   (input)
fn_ptrc_out (1) : restart file for dic     (output)
            (2) : restart file for alk     (output)
            (3) : restart file for po4     (output)
            (4) : restart file for o2      (output)
            (5) : restart file for pco2o   (output)
            (6) : restart file for pco2a   (output)
fn_ptrc_hist(1) : history file for dic
            (2) : history file for alk
            (3) : history file for po4
            (4) : history file for o2
            (5) : history file for pco2o
            (6) : history file for pco2a
            (7) : history file for oaco2
```

```
              (8) : history file for gswd
```

np_inout and np_hist is automatically set in ogfile.F90.

- When CBNHSTRUN is used, the partial pressure of atmospheric CO_2, pCO2a, should be applied as additional atmospheric forcing, and the namelist inflpco2a that indicates file name should be specified. The input intervals and the file size should be the same as other atmospheric forcing files for heat flux (see Chapter 16).
  namelist /inflpco2a/ file_pco2a_ref
  file_pcof2a_ref: Partial pressure of atmospheric CO_2 (ppt)

- When CBNHSTRUN is not used in the ocean-only model, the ocean interacts with the one-box model that contains the uniform partial pressure of atmospheric CO_2. In this case, namelist njobco2io should be specified as follows.
  namelist /njobco2io/ file_atmco2_in, file_atmco2_out
  file_atmco2_in : Partial pressure of atmospheric CO_2 (ppt) in the one-box model at the start of the integration
  file_atmco2_out: Partial pressure of atmospheric CO_2 (ppt) in the one-box model at the end of the integration

- When NPZD is used, an NPZD model is used as the ecosystem component.
  - numtrc_p=8 should be described in configure.in. When another value is specified, the model stops.
  - The compile option related to the in-line expansion should appear as FFLAGS $=$ −pvctlexpand $=$ 10.
  - When the model option CHLMA94 is used, the chlorophyll concentration is considered to calculate the shortwave penetration following Morel and Antoine (1994). See section 8.3.2.
  - The settings of restart and history files are specified in namelist inflpt, outfpt, outfph as follows:

```
namelist /inflpt/ fn_ptrc_in   (1:numtrc_p+np_inout)
namelist /outfpt/ fn_ptrc_out  (1:numtrc_p+np_inout)
namelist /outfph/ fn_ptrc_hist (1:numtrc_p+np_hist)
        np_inout = 2
        np_hist  = 6
fn_ptrc_in  (1) : restart file for dic     (input)
            (2) : restart file for alk     (input)
            (3) : restart file for po4     (input)
            (4) : restart file for o2      (input)
            (5) : restart file for no3     (input)
            (6) : restart file for PhyPl   (input)
            (7) : restart file for ZooPl   (input)
            (8) : restart file for Detri   (input)
            (9) : restart file for pco2o   (input)
           (10) : restart file for pco2a   (input)
fn_ptrc_out (1) : restart file for dic     (output)
```

```
                (2) : restart file for alk     (output)
                (3) : restart file for po4     (output)
                (4) : restart file for o2      (output)
                (5) : restart file for no3     (output)
                (6) : restart file for PhyPl   (output)
                (7) : restart file for ZooPl   (output)
                (8) : restart file for Detri   (output)
                (9) : restart file for pco2o   (output)
               (10) : restart file for pco2a   (output)
fn_ptrc_hist   (1) : history file for dic
               (2) : history file for alk
               (3) : history file for po4
               (4) : history file for o2
               (5) : history file for no3
               (6) : history file for PhyPl
               (7) : history file for ZooPl
               (8) : history file for Detri
               (9) : history file for pco2o
              (10) : history file for pco2a
              (11) : history file for oaco2
              (12) : history file for eprdc
              (13) : history file for gswd
              (14) : history file for pprdc
```

The parameters of NPZD are set in namelist `nbioNPZD`. 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 chlorophyl (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, [s].

```
 namelist /nbioNPZD/ &
  & alphabio,abio,bbio,cbio,dkcbio,dkwbio,rk1bioNO3,rk1bioPO4,&
  & gbio,epsbio,phiphy,phiphyq,a_npz,phizoo,d_npz,remina, &
  & w_detr,fac_wdetr, &
  & c_mrtn, Rcn, Ron,Rnp, dp_euph, dp_mrtn, dp_eprdc, Rcaco3poc, Dcaco3
```

11.7 Program structure

```
 ogcm__ini
 |
 +-- rdinit
```

```
        |
        +-- ptrc_init
              |
              +-- obgc_init
                    |
                    +-- (obgcinit0)
                    |
                    +-- cbn_readdt
                    |
                    +-- co2calc

ogcm__run
  |
  +-- part_1
  |     |
  |     +-- cbn_flx
  |     |     |
  |     |     +-- o2flux
  |     |     |
  |     |     +-- co2flux
  |     |           |
  |     |           +-- co2calc
  |     |
  |     +-- tracer
  |           |
  |           +-- bio_calc
  |
  +-- part_2
  |     |
  |     +-- cbn_rewrit
  |
  +-- writdt
        |
        +-- cbn_writdt
```

Reference

Dugdale, R. C., 1967: Nutrient limitation in the sea: dynamics, identification, and significance, *Limnol. Oceanogr.*, *12*, 685–695.

Eppley, R., 1972: Temperature and phytoplankton growth in the sea, *Fish. Bull.*, *70*, 1063–1085.

Keeling, R. F., B. B. Stephens, R. G. Najjar, S. C. Doney, D. Archer and M. Heimann, 1998: Seasonal variations

in the atmospheric O_2/N_2 ratio in relation to the air-sea exchange of O_2, *Global Biogeochem. Cycles*, *12*, 141-164.

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.

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.

Najjar, R., J. C. Orr, 1998: Design of OCMIP-2 simulations of chlorofluorocarbons, the solubility pump and common biogeochemistry, *Internal OCMIP Report*, *LSCE/CEA Saclay*, 25 pp, Gif-sur-Yvette, France.

Obata, A., and Y. Kitamura, 2003: Interannual variability of the sea-air exchange of CO_2 from 1961 to 1998 simulated with a global ocean circulation-biogeochemistry model, *J. Geophys. Res.*, *108*, 3377,doi:10.1029/2001 JC001088,2003.

Orr, J., R. Najjar, C. Sabine, and F. Joos, 1999: Abiotic-HOWTO, *Internal OCMIP Report*, *LSCE/CEA Saclay*, 25 pp, Gif-sur-Yvette, France.

Oschlies, A., 2001: Model-derived estimates of new production: New results point toward lower values, *Deep-Sea Res. II*, *48*, 2173–2197.

Redfield, A. C., B. H. Ketchum, and F. A. Richards, 1963: *The influence of organisms on the composition of sea water*, in The sea, vol. 2, edited by M. N. Hill, pp. 26-77, Wiley-Intersci., New York..

Schmittner, A., A. Oschlies, H. Matthews, and E. Galbraith, 2008: Future changes in climate, ocean circulation, ecosystems, and biogeochemical cycling simulated for a business-as-usual CO_2 emission scenario until year 4000 AD, *Global Biogeochem. Cycles*, *22*, GB1013,doi:10.1029/2007GB002953.

Schmittner, A., A. Oschlies, H. Matthews, and E. Galbraith, 2009: Correction to "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*, 23. GB3005,doi:10.1029/2009GB003577

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.*, , in press.

Wanninkhof, R., 1992: Relationship between wind speed and gas exchange over the ocean, *J. Geophys. Res.*, *97*, 7373-7382.

Weiss, R. F., 1970: The solubility of nitrogen, oxygen and argon in water and seawater, *Deep-Sea Res.*, *17*, 721–735.

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.

Table 11.1. Parameters used for the NPZD ecosystem component (NPZD).

namelist name		description	unit	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
rk1bioPO4	:	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_wdetri times that in the n th level,		
	:	w_detri 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 nonphotosynthetical 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	[min]	# 10.d0
	:	growth rate.		
	:	This must be a divisor of the time step for tracer.		

Part IV

Miscellaneous

Chapter 12 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 12.1. Sections 12.2 and 12.3 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 12.4. An implicit method for solving the diffusion equation is described in Section 12.5. Durran (1999) treats the basics of finite difference methods for solving differential equations of advection and diffusion in geophysical fluid dynamics.

12.1 Diffusion equation

As an example, consider an initial-boundary value problem expressed by a one-dimensional diffusion equation (heat conductive equation),

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2}. \tag{12.1}$$

Given $T(x,0) = f(x)$ as the initial distribution and $T(0,t) = T(L,t) = 0$ as the boundary condition, the analytical solution is

$$T(x,t) = \sum_{m=0}^{\infty} f_m e^{-\kappa k_m^2 t} \sin(k_m x), \tag{12.2}$$

where

$$f_m = \frac{2}{L}\int_0^L f(x)\sin(k_m x)dx, \quad k_m = \frac{m\pi}{L}. \tag{12.3}$$

Next, consider the finite difference method to get the solution numerically. In the finite difference method, grids are set with a finite increment in space and time, and each term in the equation is evaluated at each grid using $T_j^n = T(x_j, t_n)$. For example,

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \kappa \frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{\Delta x^2}, \tag{12.4}$$

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 (12.1) in the limit $\Delta t \to 0, \Delta x \to 0$ (**consistency**).

If the initial distribution is assumed to be $f(x) = T_0 \sin k_1 x$, the solution of the finite difference equation (12.4) for $t = t_n$ is

$$T_j^n = \lambda^n T_0 \sin k_1 x_j, \tag{12.5}$$

where

$$\lambda = 1 - \frac{2\kappa\Delta t}{\Delta x^2}(1 - \cos k_1 \Delta x). \tag{12.6}$$

In order to suppress oscillation and divergence of the solution (**stability**), $0 < \lambda < 1$ is necessary and Δx and Δt must be set to satisfy this condition. This solution is identical to the analytical solution in the limit of $\Delta t \to 0, \Delta x \to 0$ (**convergence**).

To summarize, the finite difference method that satisfies consistency, stability, and convergence is the necessary condition for an accurate solution.

12.2 Finite difference expressions for time derivatives

The following four finite difference expressions are employed for the time derivatives in MRI.COM:

$$\text{forward}: \quad \frac{T^{n+1}-T^n}{\Delta t} = F(T^n) \tag{12.7}$$

$$\text{backward}: \quad \frac{T^{n+1}-T^n}{\Delta t} = F(T^{n+1}) \tag{12.8}$$

$$\text{Matsuno}: \quad \frac{T^{*n+1}-T^n}{\Delta t} = F(T^n), \quad \frac{T^{n+1}-T^n}{\Delta t} = F(T^{*n+1}) \tag{12.9}$$

$$\text{leap-frog}: \quad \frac{T^{n+1}-T^{n-1}}{2\Delta t} = F(T^n). \tag{12.10}$$

The scheme used in the previous section is the forward scheme. The forward, backward, and Matsuno schemes use the values at two time levels and are accurate to $O(\Delta t)$, while the leap-frog scheme uses three time levels and is accurate to $O(\Delta t^2)$. Basically, the leap-frog scheme is employed in MRI.COM because of its higher order accuracy.

However, the leap-frog scheme cannot be applied to the diffusion equation. A solution by the finite difference method using the leap-frog scheme is given by

$$T_j^n = (T_a\lambda_a^n + T_b\lambda_b^n)\sin k_1 x_j, \tag{12.11}$$

where

$$\lambda_a = -\frac{-\alpha+\sqrt{\alpha^2+4}}{2}, \quad \lambda_b = -\frac{-\alpha-\sqrt{\alpha^2+4}}{2} \quad \left(\alpha \equiv \frac{4\kappa\Delta t}{\Delta x^2}(1-\cos k_1\Delta x)\right). \tag{12.12}$$

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 (12.6). In such a case, the backward scheme is used for vertical diffusion and viscosity (implicit method; see Section 12.5). 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 12.5).

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 (`matsuno_int` of the namelist group `njobp`). It should be noted that the Matsuno scheme needs twice as many numerical operations as the forward and leap-frog schemes.

12.3 Finite difference expression for space derivatives

Let us consider a one-dimensional advection equation,

$$\frac{\partial T}{\partial t} = -u\frac{\partial T}{\partial x}, \tag{12.13}$$

where u is a constant velocity. The solution is

$$T(x,t) = T(x-ut,0). \tag{12.14}$$

Using the leap-frog scheme for time differencing, the finite difference equation can be written as follows:

$$\frac{T_j^{n+1} - T_j^{n-1}}{2\Delta t} = -u\frac{T_{j+\frac{1}{2}}^n - T_{j-\frac{1}{2}}^n}{\Delta x}, \tag{12.15}$$

where $T_{j-\frac{1}{2}}^n$ and $T_{j+\frac{1}{2}}^n$ are the values at the left and right (in the same way for up-down and front-rear grid cells) 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 popular formulations,

$$\text{upstream finite difference :} \quad T_{j-\frac{1}{2}}^n = T_{j-1}^n (u>0), \;\; T_{j-\frac{1}{2}}^n = T_j^n (u<0), \tag{12.16}$$

$$\text{central finite difference :} \quad T_{j-\frac{1}{2}}^n = \frac{T_{j-1}^n + T_j^n}{2}. \tag{12.17}$$

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 (12.15) is

$$\frac{T_j^{n+1} - T_j^{n-1}}{2\Delta t} = -u\frac{T_{j+1}^n - T_{j-1}^n}{2\Delta x}. \tag{12.18}$$

Assuming the solution to be $T(x,t) = \tau(t)e^{-ikx}$,

$$\tau^{n+1} = \tau^{n-1} + 2i\alpha\tau^n, \;\; \text{where } \alpha \equiv \frac{u\Delta t}{\Delta x}\sin k\Delta x. \tag{12.19}$$

It is stable (neutral) if $|\alpha| \leq 1$. To be stable for any wave number,

$$\left|\frac{u\Delta t}{\Delta x}\right| \leq 1 \tag{12.20}$$

must be satisfied (**CFL condition**). However, if $\tau^n = \tau^0 e^{-in\Delta\theta}$,

$$\Delta\theta = -\sin^{-1}[\mu \sin k\Delta x] (\text{where } \mu \equiv \frac{u\Delta t}{\Delta x}). \tag{12.21}$$

Expanding the r.h.s. by a Taylor expansion we obtain

$$\begin{aligned} \Delta\theta &\simeq -\mu\sin k\Delta x - \frac{1}{6}(\mu \sin k\Delta x)^3 \\ &\simeq -\mu k\Delta x + \frac{\mu(k\Delta x)^3}{6} - \frac{\mu^3(k\Delta x)^3}{6} \\ &= -\mu k\Delta x\left\{1 - \frac{(k\Delta x)^2}{6}(1-\mu^2)\right\}. \end{aligned} \tag{12.22}$$

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 5.

Using the upstream finite difference, the finite difference equation (12.15) is

$$\frac{T_j^{n+1} - T_j^{n-1}}{2\Delta t} = -u\frac{T_j^n - T_{j-1}^n}{\Delta x}. \tag{12.23}$$

Expanding the r.h.s. by a Taylor expansion we obtain:

$$-u\frac{\partial T}{\partial x} + \frac{u\Delta x}{2}\frac{\partial^2 T}{\partial x^2} + O(\Delta x^2). \tag{12.24}$$

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

$$T_{j-\frac{1}{2}}^n = \frac{-T_{j-2}^n + 6T_{j-1}^n + 3T_j^n}{8}(u > 0), \quad T_{j-\frac{1}{2}}^n = \frac{3T_{j-1}^n + 6T_j^n - T_{j+1}^n}{8}(u < 0). \tag{12.25}$$

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 13.

12.4 Finite differencing of advection-diffusion equation

According to the above restriction, when advection-diffusion equations (2.14) and (2.15) are 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:

$$\frac{T^{n+1} - T^{n-1}}{2\Delta t} = -\mathscr{A}(T^n) + \mathscr{D}(T^{n-1}) \tag{12.26}$$

$$\frac{S^{n+1} - S^{n-1}}{2\Delta t} = -\mathscr{A}(S^n) + \mathscr{D}(S^{n-1}). \tag{12.27}$$

When the vertical diffusion term is very large, $\mathscr{D}(T^{n+1})$ is used instead of $\mathscr{D}(T^{n-1})$. This formula is an implicit scheme and is described in the next section.

12.5 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 7. 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 temperature equation using the leap-frog scheme. The diffusion term is written separately using the

$(n-1)$ step for the horizontal direction and the $(n+1)$ step for the vertical direction,

$$(T^{n+1}-T^{n-1})/2\Delta t = -\mathscr{A}(T^n)+\mathscr{D}_H(T^{n-1})+\mathscr{D}_V(T^{n+1}). \tag{12.28}$$

Putting all the terms involving T^{n+1} on the l.h.s.,

$$T^{n+1}-2\Delta t\mathscr{D}_V(T^{n+1}) = T^{n-1}+2\Delta t(-\mathscr{A}(T^n)+\mathscr{D}_H(T^{n-1})) \tag{12.29}$$

is obtained, which is an algebraic equation for T^{n+1}. The finite difference form is rewritten specifically as

$$\begin{aligned} T_k^{n+1}-2\Delta t\frac{1}{\Delta z_k}\left(\kappa_{k-\frac{1}{2}}(T_{k-1}^{n+1}-T_k^{n+1})/\Delta z_{k-\frac{1}{2}}-\kappa_{k+\frac{1}{2}}(T_k^{n+1}-T_{k+1}^{n+1})/\Delta z_{k+\frac{1}{2}}\right) \\ = T_k^{n-1}+2\Delta t(-\mathscr{A}(T_k^n)+\mathscr{D}_H(T_k^{n-1})). \end{aligned} \tag{12.30}$$

By putting

$$a=\frac{2\Delta t\,\kappa_{k-\frac{1}{2}}}{\Delta z_k\Delta z_{k-\frac{1}{2}}},\quad b=1+a+c,\quad c=\frac{2\Delta t\,\kappa_{k+\frac{1}{2}}}{\Delta z_k\Delta z_{k+\frac{1}{2}}}, \tag{12.31}$$

we get,

$$-aT_{k-1}^{n+1}+bT_k^{n+1}-cT_{k+1}^{n+1} = T_k^{n-1}+2\Delta t(-\mathscr{A}(T_k^n)+\mathscr{D}_H(T_k^{n-1})). \tag{12.32}$$

Setting $\mathscr{F}_k \equiv -\mathscr{A}(T_k^n)+\mathscr{D}_H(T_k^{n-1})$, this is expressed in the matrix form as

$$\begin{pmatrix} b & -c & & & & & \\ -a & b & -c & & & & \\ & -a & b & -c & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -a & b & -c & \\ & & & & -a & b & -c \\ & & & & & -a & b \end{pmatrix} \begin{pmatrix} T_1^{n+1} \\ T_2^{n+1} \\ T_3^{n+1} \\ \vdots \\ T_{KM-2}^{n+1} \\ T_{KM-1}^{n+1} \\ T_{KM}^{n+1} \end{pmatrix} = \begin{pmatrix} T_1^{n-1}+2\Delta t\mathscr{F}_1 \\ T_2^{n-1}+2\Delta t\mathscr{F}_2 \\ T_3^{n-1}+2\Delta t\mathscr{F}_3 \\ \vdots \\ T_{KM-2}^{n-1}+2\Delta t\mathscr{F}_{KM-2} \\ T_{KM-1}^{n-1}+2\Delta t\mathscr{F}_{KM-1} \\ T_{KM}^{n-1}+2\Delta t\mathscr{F}_{KM} \end{pmatrix}. \tag{12.33}$$

The l.h.s. has the form of the tri-diagonal matrix.

12.5.1 A solution of tri-diagonal matrix

In general, simultaneous linear equations for n variables with tri-diagonal matrix coefficients

$$\begin{pmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & B_{n-1} & C_{n-1} \\ & & & A_n & B_n \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n-1} \\ X_n \end{pmatrix} = \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_{n-1} \\ D_n \end{pmatrix} \tag{12.34}$$

are solved using the Thomas method, which is modified from LU decomposition,

$$P_1 = C_1/B_1 \tag{12.35}$$

$$Q_1 = D_1/B_1 \tag{12.36}$$

$$P_k = \frac{C_k}{B_k - A_k P_{k-1}} \qquad (2 \le k \le n-1) \tag{12.37}$$

$$Q_k = \frac{D_k - A_k Q_{k-1}}{B_k - A_k P_{k-1}} \qquad (2 \le k \le n) \tag{12.38}$$

$$X_n = Q_n \tag{12.39}$$

$$X_k = Q_k - P_k X_{k+1} \qquad (1 \le k \le n-1). \tag{12.40}$$

References

Durran, D. R., 1999: *Numerical methods for wave equations in geophysical fluid dynamics*, Springer-Verlag, 465pp.

Chapter 13 Tracer advection schemes

The default tracer advection scheme of MRI.COM is the Quadratic Upstream Interpolation for Convective Kinematics (QUICK; Leonard, 1979) as described in Chapter 6. Other options with higher accuracy are described in this chapter.

Section 13.1 describes the QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979) for vertical advection (option `QUICKEST`). Section 13.2 describes the Uniformly Third-Order Polynomial Interpolation Algorithm (UTOPIA; Leonard et al., 1993) for horizontal advection (option `UTOPIA`). The UTOPIA scheme is a two-dimensional generalization of the QUICKEST scheme. For these schemes, a flux limiter that prevents unrealistic extrema should be used (Leonard et al., 1994). A flux limiter for the QUICKEST scheme is used when option `ZULTIMATE` is specified. A flux limiter for the UTOPIA scheme is used when option `ULTIMATE` is specified.

The above schemes seek to improve the accuracy by refining the finite-difference expression at the cell faces. There is another approach that seeks to improve the accuracy by considering the distribution within the cell. The second order moment (SOM; Prather, 1986) scheme takes this approach and is available in MRI.COM through the option `SOMADVEC` (Section 13.3).

13.1 QUICKEST for vertical advection

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

Consider a one-dimensional equation of advection for incompressible fluid

$$\frac{\partial T}{\partial t} + \frac{\partial}{\partial z}(wT) = 0, \tag{13.1}$$

where w is a constant. Although the velocities are not constants 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) and bottom $(k+1)$ faces of the vertical cells. The following relation holds for the vertical grid spacings:

$$\Delta z_k = \frac{\Delta z_{k+\frac{1}{2}} + \Delta z_{k-\frac{1}{2}}}{2}. \tag{13.2}$$

In QUICKEST, the distribution of tracer T is defined using the second order interpolations, and the mean value during a time step at the cell face (boundary of two adjacent tracer cells) is calculated. The coefficients for the

second order interpolation are calculated first. A Taylor expansion of T about point z_k gives

$$T_{k-\frac{3}{2}} = c_0 + c_1\left(\frac{\Delta z_{k-\frac{3}{2}}}{2} + \Delta z_{k-\frac{1}{2}}\right) + c_2\left(\frac{\Delta z_{k-\frac{3}{2}}}{2} + \Delta z_{k-\frac{1}{2}}\right)^2 + O(\Delta z^3), \tag{13.3}$$

$$T_{k-\frac{1}{2}} = c_0 + c_1\frac{\Delta z_{k-\frac{1}{2}}}{2} + c_2\frac{\Delta z_{k-\frac{1}{2}}^2}{4} + O(\Delta z^3), \tag{13.4}$$

$$T_{k+\frac{1}{2}} = c_0 - c_1\frac{\Delta z_{k+\frac{1}{2}}}{2} + c_2\frac{\Delta z_{k+\frac{1}{2}}^2}{4} + O(\Delta z^3), \tag{13.5}$$

$$T_{k+\frac{3}{2}} = c_0 - c_1\left(\frac{\Delta z_{k+\frac{3}{2}}}{2} + \Delta z_{k+\frac{1}{2}}\right) + c_2\left(\frac{\Delta z_{k+\frac{3}{2}}}{2} + \Delta z_{k+\frac{1}{2}}\right)^2 + O(\Delta z^3). \tag{13.6}$$

Coefficients c_0, c_1, and c_2 can be solved using three of the four equations (13.3), (13.4), (13.5), and (13.6). The three upstream-side equations are chosen. When $w > 0$ ($w < 0$), equations (13.4), (13.5), and (13.6) ((13.3), (13.4), and (13.5)) are used. The solution is as follows.

$$c_0 = \frac{T_{k-\frac{1}{2}}\Delta z_{k+\frac{1}{2}} + T_{k+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}}{2\Delta z_k} - \frac{\Delta z_{k+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}}{4}c_2, \tag{13.7}$$

$$c_1 = \frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_k} - \frac{\Delta z_{k-\frac{1}{2}} - \Delta z_{k+\frac{1}{2}}}{2}c_2, \tag{13.8}$$

$$c_2 = \begin{cases} \frac{1}{\Delta z_k + \Delta z_{k+1}}\left(\frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_k} - \frac{T_{k+\frac{1}{2}} - T_{k+\frac{3}{2}}}{\Delta z_{k+1}}\right) & (w > 0), \\ \frac{1}{\Delta z_{k-1} + \Delta z_k}\left(\frac{T_{k-\frac{3}{2}} - T_{k-\frac{1}{2}}}{\Delta z_{k-1}} - \frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_k}\right) & (w < 0). \end{cases} \tag{13.9}$$

Next, equation (13.1) is integrated over one time step and one grid cell.

$$\int_{t^n}^{t^{n+1}} dt \int_{z_{k+1}}^{z_k} dz \frac{\partial T}{\partial t} = -\int_{t^n}^{t^{n+1}} dt \int_{z_{k+1}}^{z_k} dz \frac{\partial}{\partial z}(wT). \tag{13.10}$$

The r.h.s. of (13.10) can be written as

$$-\int_{t^n}^{t^{n+1}} dt(w_u T_u - w_l T_l), \tag{13.11}$$

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

$$\int_{t^n}^{t^{n+1}} dt\, T_l = \int_{-w_l\Delta t}^{0} [c_0^n + c_1^n\xi + c_2^n\xi^2 + O(\Delta z^3)]\frac{d\xi}{w_l}. \tag{13.12}$$

Thus expression (13.11) becomes

$$-\Delta t(w_u\overline{T_u^n} - w_l\overline{T_l^n}) + O(\Delta z^3 w\Delta t), \tag{13.13}$$

where

$$\begin{aligned} \overline{T_l^n} &= \frac{1}{w_l\Delta t}\int_{-w_l\Delta t}^{0}(c_0^n + c_1^n\xi + c_2^n\xi^2)d\xi \\ &= c_0^n - \frac{c_1^n}{2}w_l\Delta t + \frac{c_2^n}{3}w_l^2\Delta t^2. \end{aligned} \tag{13.14}$$

Using up to the second order terms of a Taylor expansion, the l.h.s. of (13.10) can be written as follows:

$$\int_{t^n}^{t^{n+1}} dt \int_{z_{k+1}}^{z_k} dz \frac{\partial T}{\partial t} = \Delta z_k\left[T_{k+\frac{1}{2}}^{n+1} - T_{k+\frac{1}{2}}^{n} + \frac{\Delta z_{k+\frac{1}{2}}^2}{24}(T_{zz}{}_{k+\frac{1}{2}}^{n+1} - T_{zz}{}_{k+\frac{1}{2}}^{n}) + O(\Delta z^3)\right], \tag{13.15}$$

where

$$\begin{aligned} {T_{zz}}_{k+\frac{1}{2}}^{n+1} - {T_{zz}}_{k+\frac{1}{2}}^{n} &= \Delta t \left.\frac{\partial T_{zz}}{\partial t}\right|_{k+\frac{1}{2}}^{n} + O(\Delta t^2) \\ &= -\Delta t \frac{\partial^2}{\partial z^2}\left[\frac{\partial}{\partial z}(uT)\right]_{k+\frac{1}{2}}^{n} + O(\Delta t^2) \\ &= -\Delta t \frac{\partial}{\partial z}(wT_{zz})_{k+\frac{1}{2}}^{n} + O(\Delta t^2) \\ &= -\Delta t \left\{\frac{w_u {T_{zz}}_u^n - w_l {T_{zz}}_l^n}{\Delta z_{k+\frac{1}{2}}}\right\} + O(w\Delta t\Delta x) + O(\Delta t^2). \end{aligned} \tag{13.16}$$

The expression for the r.h.s. of (13.15) becomes

$$\Delta z_k \left[T_{k+\frac{1}{2}}^{n+1} - T_{k+\frac{1}{2}}^{n} - \frac{\Delta z_{k+\frac{1}{2}}^2}{24}\Delta t \frac{w_u {T_{zz}}_u^n - u_l {T_{zz}}_l^n}{\Delta z_{k+\frac{1}{2}}} + O(\Delta z^3)\right] + O(w\Delta t\Delta x^3) + O(\Delta z^3 \Delta t^2). \tag{13.17}$$

Based on (13.13) and (13.17), the discretized forecasting equation is expressed as follows:

$$T_k^{n+1} = T_k^n - \frac{\Delta t}{\Delta z_{k+\frac{1}{2}}}\left[w_u \overline{T_u^n} - w_l \overline{T_l^n} - \frac{\Delta z_{k+\frac{1}{2}}^2}{24}(w_u {T_{zz}}_u^n - w_l {T_{zz}}_l^n)\right] + O(\alpha\Delta z^3) + O(\Delta z^2 \Delta t^2), \tag{13.18}$$

where

$$\alpha \equiv \frac{w\Delta t}{\Delta z} < 1, \tag{13.19}$$

$${T_{zz}}_l^n = 2c_2 + O(\Delta z). \tag{13.20}$$

The accuracy of equation (13.18) is $\max(O(\Delta z^3), O(\Delta z^2 \Delta t^2))$.

13.2 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 MRI.COM, 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 boarderlines of the velocity cells into four boxes with (almost) identical area.

Consider an equation of advection:

$$\frac{\partial T}{\partial t} + \frac{1}{h_\mu h_\psi}\frac{\partial}{\partial \mu}(h_\psi u T) + \frac{1}{h_\mu h_\psi}\frac{\partial}{\partial \psi}(h_\mu v T) = 0. \tag{13.21}$$

Integrated over a tracer cell and for one time step,

$$\begin{aligned} &\int_{\psi_L - \Delta\psi_L/2}^{\psi_L + \Delta\psi_L/2} d\psi \int_{\mu_L - \Delta\mu_L/2}^{\mu_L + \Delta\mu_L/2} d\mu (\chi^{n+1} - \chi^n) \\ = \;& -\Delta t (u_r^n \overline{T_r^n}\Delta y_r - u_l^n \overline{T_l^n}\Delta y_l + v_u^n \overline{T_u^n}\Delta x_u - v_d^n \overline{T_d^n}\Delta x_d), \end{aligned} \tag{13.22}$$

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 (13.22), 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 13.1 for the label of the point):

$$\chi = \chi_L + a_{10}(\mu - \mu_L) + a_{20}(\mu - \mu_L)^2 + a_{01}(\psi - \psi_L) + a_{02}(\psi - \psi_L)^2 + a_{11}(\mu - \mu_L)(\psi - \psi_L). \tag{13.23}$$

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

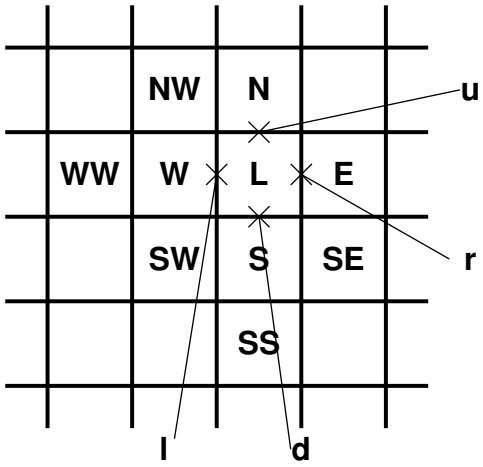

Figure 13.1. Labels of tracer grid points (upper case characters) and faces (lower case characters).

$$\chi_E = \chi_L + a_{10}\Delta\mu_r + a_{20}\Delta\mu_r^2, \tag{13.24}$$

$$\chi_W = \chi_L - a_{10}\Delta\mu_l + a_{20}\Delta\mu_l^2, \tag{13.25}$$

$$\chi_N = \chi_L + a_{01}\Delta\psi_u + a_{02}\Delta\psi_u^2, \tag{13.26}$$

$$\chi_S = \chi_L - a_{01}\Delta\psi_d + a_{02}\Delta\psi_d^2, \tag{13.27}$$

where

$$\Delta\psi_u \equiv \frac{\Delta\psi_L + \Delta\psi_N}{2}, \tag{13.28}$$

$$\Delta\psi_d \equiv \frac{\Delta\psi_L + \Delta\psi_S}{2}, \tag{13.29}$$

$$\Delta\mu_r \equiv \frac{\Delta\mu_L + \Delta\mu_E}{2}, \tag{13.30}$$

$$\Delta\mu_l \equiv \frac{\Delta\mu_L + \Delta\mu_W}{2}. \tag{13.31}$$

Using these known values, the following parameters are obtained,

$$a_{10} = \frac{\Delta\mu_l \dfrac{\chi_E - \chi_L}{\Delta\mu_r} + \Delta\mu_r \dfrac{\chi_L - \chi_W}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}, \tag{13.32}$$

$$a_{20} = \frac{\dfrac{\chi_E - \chi_L}{\Delta\mu_r} - \dfrac{\chi_L - \chi_W}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}, \tag{13.33}$$

$$a_{01} = \frac{\Delta\psi_d \dfrac{\chi_N - \chi_L}{\Delta\psi_u} + \Delta\psi_u \dfrac{\chi_L - \chi_S}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}, \tag{13.34}$$

$$a_{02} = \frac{\dfrac{\chi_N - \chi_L}{\Delta\psi_u} - \dfrac{\chi_L - \chi_S}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{13.35}$$

Substituting (13.23) into the l.h.s. of (13.22) yields

$$\Delta\mu_L\Delta\psi_L \left[\chi_L^{n+1} - \chi_L^n + \frac{\Delta\mu_L^2}{12}(a_{20}^{n+1} - a_{20}^n) + \frac{\Delta\psi_L^2}{12}(a_{02}^{n+1} - a_{02}^n) \right]. \tag{13.36}$$

Using equation (13.21), the following approximation is allowed:

$$a_{20}^{n+1} - a_{20}^{n} = -\Delta t \left[\frac{h_{\psi r} u_r^n T_{\mu\mu r}^n - h_{\psi l} u_l^n T_{\mu\mu l}^n}{\Delta\mu_L} + \frac{h_{\mu u} v_u^n T_{\mu\mu u}^n - h_{\mu d} v_d^n T_{\mu\mu d}^n}{\Delta\psi_L} \right], \tag{13.37}$$

$$a_{02}^{n+1} - a_{02}^{n} = -\Delta t \left[\frac{h_{\psi r} u_r^n T_{\psi\psi r}^n - h_{\psi l} u_l^n T_{\psi\psi l}^n}{\Delta\mu_L} + \frac{h_{\mu u} v_u^n T_{\psi\psi u}^n - h_{\mu d} v_d^n T_{\psi\psi d}^n}{\Delta\psi_L} \right], \tag{13.38}$$

where $T_{\mu\mu r}^n$ is the value of the second order derivative at the right face **r**, whose expression is similar to that of c_{20} described later. Therefore, under a suitable approximation,

$$T_L^{n+1} = T_L^n - \frac{\Delta t}{\Delta S_L}(u_r^n \tilde{T}_r^n \Delta y_r - u_l^n \tilde{T}_l^n \Delta y_l + v_u^n \tilde{T}_u^n \Delta x_u - v_d^n \tilde{T}_d^n \Delta x_d), \tag{13.39}$$

where

$$\tilde{T}_l^n = \overline{T_l^n} - \frac{\Delta\mu_L^2}{24} T_{\mu\mu l}^n - \frac{\Delta\psi_L^2}{24} T_{\psi\psi l}^n, \tag{13.40}$$

$$\tilde{T}_d^n = \overline{T_d^n} - \frac{\Delta\mu_L^2}{24} T_{\mu\mu d}^n - \frac{\Delta\psi_L^2}{24} T_{\psi\psi d}^n. \tag{13.41}$$

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 13.2, and the values of T^n are given as the second order interpolation about **l** of Figure 13.1. Similar operations will be used to obtain the expression for $\overline{T_d^n}$.

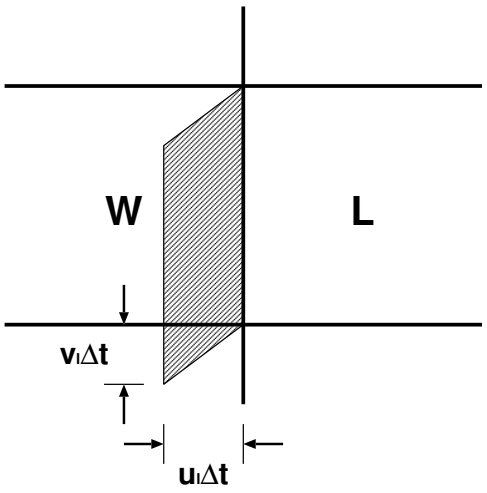

Figure 13.2. Area used to average tracer values for the face **l**

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

$$T^n|_{\mathbf{l}} = c_{00} + c_{10}(\mu - \mu_l) + c_{20}(\mu - \mu_l)^2 + c_{01}(\psi - \psi_L) + c_{02}(\psi - \psi_L)^2 + c_{11}(\mu - \mu_l)(\psi - \psi_L), \tag{13.42}$$

$$T^n|_{\mathbf{d}} = d_{00} + d_{10}(\mu - \mu_l) + d_{20}(\mu - \mu_l)^2 + d_{01}(\psi - \psi_L) + d_{02}(\psi - \psi_L)^2 + d_{11}(\mu - \mu_l)(\psi - \psi_L). \tag{13.43}$$

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

$$T_L^n = c_{00} + c_{10}\frac{\Delta\mu_L}{2} + c_{20}\frac{\Delta\mu_L^2}{4}, \tag{13.44}$$

$$T_W^n = c_{00} - c_{10}\frac{\Delta\mu_W}{2} + c_{20}\frac{\Delta\mu_W^2}{4}, \tag{13.45}$$

$$T_E^n = c_{00} + c_{10}\left(\Delta\mu_L + \frac{\Delta\mu_E}{2}\right) + c_{20}\left(\Delta\mu_L + \frac{\Delta\mu_E}{2}\right)^2, \tag{13.46}$$

$$T_{WW}^n = c_{00} - c_{10}\left(\Delta\mu_W + \frac{\Delta\mu_{WW}}{2}\right) + c_{20}\left(\Delta\mu_W + \frac{\Delta\mu_{WW}}{2}\right)^2, \tag{13.47}$$

$$T_N^n = T_L^n + c_{01}\Delta\psi_u + c_{02}\Delta\psi_u^2 + c_{11}\frac{\Delta\mu_L}{2}\Delta\psi_u, \tag{13.48}$$

$$T_S^n = T_L^n - c_{01}\Delta\psi_d + c_{02}\Delta\psi_d^2 - c_{11}\frac{\Delta\mu_L}{2}\Delta\psi_d, \tag{13.49}$$

$$T_{NW}^n = T_W^n + c_{01}\Delta\psi_u + c_{02}\Delta\psi_u^2 - c_{11}\frac{\Delta\mu_W}{2}\Delta\psi_u, \tag{13.50}$$

$$T_{SW}^n = T_W^n - c_{01}\Delta\psi_d + c_{02}\Delta\psi_d^2 + c_{11}\frac{\Delta\mu_W}{2}\Delta\psi_d. \tag{13.51}$$

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

$$u_l^n > 0, \quad v_l^n > 0 \quad \Rightarrow \quad \mathbf{L}, \mathbf{W}, \mathbf{WW}, \mathbf{S}, \mathbf{NW}, \mathbf{SW} \tag{13.52}$$

$$u_l^n < 0, \quad v_l^n > 0 \quad \Rightarrow \quad \mathbf{L}, \mathbf{W}, \mathbf{E}, \mathbf{N}, \mathbf{S}, \mathbf{SW} \tag{13.53}$$

$$u_l^n > 0, \quad v_l^n < 0 \quad \Rightarrow \quad \mathbf{L}, \mathbf{W}, \mathbf{WW}, \mathbf{N}, \mathbf{NW}, \mathbf{SW} \tag{13.54}$$

$$u_l^n < 0, \quad v_l^n < 0 \quad \Rightarrow \quad \mathbf{L}, \mathbf{W}, \mathbf{E}, \mathbf{N}, \mathbf{S}, \mathbf{NW} \tag{13.55}$$

From equations (13.44) and (13.45),

$$c_{00} = \frac{\Delta\mu_W T_L^n + \Delta\mu_L T_W^n}{2\Delta\mu_l} - c_{20}\frac{\Delta\mu_L\Delta\mu_W}{4}, \tag{13.56}$$

$$c_{10} = \frac{T_L^n - T_W^n}{\Delta\mu_l} - c_{20}\frac{\Delta\mu_L - \Delta\mu_W}{2}. \tag{13.57}$$

When $u_l^n > 0$, from (13.44) and (13.47),

$$c_{20} = \frac{\dfrac{T_L^n - T_W^n}{\Delta\mu_l} - \dfrac{T_W^n - T_{WW}^n}{\Delta\mu_{ll}}}{\Delta\mu_l + \Delta\mu_{ll}}, \tag{13.58}$$

$$\text{where} \quad \Delta\mu_{ll} \equiv \frac{\Delta\mu_W + \Delta\mu_{WW}}{2}.$$

Using equations (13.50) and (13.51),

$$c_{02} = \frac{\dfrac{T_{NW}^n - T_W^n}{\Delta\psi_u} - \dfrac{T_W^n - T_{SW}^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{13.59}$$

When $u_l^n < 0$, from (13.45) and (13.46),

$$c_{20} = \frac{\dfrac{T_E^n - T_L^n}{\Delta\mu_r} - \dfrac{T_L^n - T_W^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{13.60}$$

Using equations (13.48) and (13.49),

$$c_{02} = \frac{\frac{T_N^n - T_L^n}{\Delta\psi_u} - \frac{T_L^n - T_S^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{13.61}$$

When $v_l^n > 0$, from (13.49) and (13.51),

$$c_{01} = \frac{\Delta\mu_W(T_L^n - T_S^n) + \Delta\mu_L(T_W^n - T_{SW}^n)}{2\Delta\mu_l\Delta\psi_d} + c_{02}\Delta\psi_d, \tag{13.62}$$

$$c_{11} = \frac{T_{SW}^n - T_W^n - T_S^n + T_L^n}{\Delta\mu_l\Delta\psi_d}. \tag{13.63}$$

When $v_l^n < 0$, from (13.48) and (13.50),

$$c_{01} = \frac{\Delta\mu_W(T_N^n - T_L^n) + \Delta\mu_L(T_{NW}^n - T_W^n)}{2\Delta\mu_l\Delta\psi_u} - c_{02}\Delta\psi_u, \tag{13.64}$$

$$c_{11} = \frac{T_N^n - T_L^n - T_{NW}^n + T_W^n}{\Delta\mu_l\Delta\psi_u}. \tag{13.65}$$

Next, using equation (13.43), the T^n values at eight points around **d** are

$$T_L^n = d_{00} + d_{01}\frac{\Delta\psi_L}{2} + d_{02}\frac{\Delta\psi_L^2}{4}, \tag{13.66}$$

$$T_S^n = d_{00} - d_{01}\frac{\Delta\psi_S}{2} + d_{02}\frac{\Delta\psi_S^2}{4}, \tag{13.67}$$

$$T_N^n = d_{00} + d_{01}\left(\Delta\psi_L + \frac{\Delta\psi_N}{2}\right) + d_{02}\left(\Delta\psi_L + \frac{\Delta\psi_N}{2}\right)^2, \tag{13.68}$$

$$T_{SS}^n = d_{00} - d_{01}\left(\Delta\psi_S + \frac{\Delta\psi_{SS}}{2}\right) + d_{02}\left(\Delta\psi_S + \frac{\Delta\psi_{SS}}{2}\right)^2, \tag{13.69}$$

$$T_E^n = T_L^n + d_{10}\Delta\mu_r + d_{20}\Delta\mu_r^2 + d_{11}\frac{\Delta\psi_L}{2}\Delta\mu_r, \tag{13.70}$$

$$T_W^n = T_L^n - d_{10}\Delta\mu_l + d_{20}\Delta\mu_l^2 - d_{11}\frac{\Delta\psi_L}{2}\Delta\mu_l, \tag{13.71}$$

$$T_{SE}^n = T_S^n + d_{10}\Delta\mu_r + d_{20}\Delta\mu_r^2 - d_{11}\frac{\Delta\psi_S}{2}\Delta\mu_r, \tag{13.72}$$

$$T_{SW}^n = T_S^n - d_{10}\Delta\mu_l + d_{20}\Delta\mu_l^2 + d_{11}\frac{\Delta\psi_S}{2}\Delta\mu_l. \tag{13.73}$$

From equations (13.66) and (13.67),

$$d_{00} = \frac{\Delta\psi_S T_L^n + \Delta\psi_L T_S^n}{2\Delta\psi_d} - d_{02}\frac{\Delta\psi_L\Delta\psi_S}{4}, \tag{13.74}$$

$$d_{01} = \frac{T_L^n - T_S^n}{\Delta\psi_d} - d_{02}\frac{\Delta\psi_L - \Delta\psi_S}{2}. \tag{13.75}$$

When $v_d^n > 0$, from (13.66) and (13.69),

$$d_{02} = \frac{\frac{T_L^n - T_S^n}{\Delta\psi_d} - \frac{T_S^n - T_{SS}^n}{\Delta\psi_{dd}}}{\Delta\psi_d + \Delta\psi_{dd}}, \tag{13.76}$$

$$\text{where} \quad \Delta\psi_{dd} \equiv \frac{\Delta\psi_S + \Delta\psi_{SS}}{2}.$$

From (13.72) and (13.73),

$$d_{20} = \frac{\frac{T_{SE}^n - T_S^n}{\Delta\mu_r} - \frac{T_S^n - T_{SW}^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{13.77}$$

When $v_d^n < 0$, from (13.67) and (13.68),

$$d_{02} = \frac{\dfrac{T_N^n - T_L}{\Delta\psi_u} - \dfrac{T_L^n - T_S^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{13.78}$$

From (13.70) and (13.71),

$$d_{20} = \frac{\dfrac{T_E^n - T_L^n}{\Delta\mu_r} - \dfrac{T_L^n - T_W^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{13.79}$$

When $u_d^n > 0$, from (13.71) and (13.73),

$$d_{10} = \frac{\Delta\psi_S(T_L^n - T_W^n) + \Delta\psi_L(T_S^n - T_{SW}^n)}{2\Delta\psi_d\Delta\mu_l} + d_{20}\Delta\mu_l, \tag{13.80}$$

$$d_{11} = \frac{T_L^n - T_S^n - T_W^n + T_{SW}^n}{\Delta\psi_d\Delta\mu_l}. \tag{13.81}$$

When $u_d^n < 0$, from (13.70) and (13.72),

$$d_{10} = \frac{\Delta\psi_S(T_E^n - T_L^n) + \Delta\psi_L(T_{SE}^n - T_S^n)}{2\Delta\psi_d\Delta\mu_r} - d_{20}\Delta\mu_r, \tag{13.82}$$

$$d_{11} = \frac{T_E^n - T_{SE}^n - T_L^n + T_S^n}{\Delta\psi_d\Delta\mu_r}. \tag{13.83}$$

The value of $\overline{T_l^n}$ is the average of T^n over the hatched area of Figure 13.2. Defining

$$\xi_l^n = \frac{u_l^n}{h_{\mu l}}, \quad \eta_l^n = \frac{v_l^n}{h_{\psi l}}, \tag{13.84}$$

we have

$$\begin{aligned}
\overline{T_l^n} &= \frac{a}{\xi_l^n\Delta t\Delta\psi_L}\left[\int_{\psi_L-\Delta\psi_L/2}^{\psi_L+\Delta\psi_L/2}\int_{\mu_l-\xi_l^n\Delta t}^{\mu_l} T^n d\mu d\psi\right. \\
&+ \left.\int_{\psi_L-\Delta\psi_L/2-\eta_l^n\Delta t}^{\psi_L-\Delta\psi_L/2}\int_{\mu_l-\xi_l^n\Delta t}^{\mu_l+\frac{\xi_l^n}{\eta_l^n}(\psi-\psi_L+\Delta\psi_L/2)} \{T^n(\psi) - T^n(\psi+\Delta\psi_L)\} d\mu d\psi\right] \\
&= c_{00} - \frac{1}{2}\eta_l^n\Delta t\, c_{01} + \left[\frac{1}{12}\Delta\psi_L^2 + \frac{1}{3}^*(\eta_l^n\Delta t)^2\right] c_{02} \\
&\quad -\frac{1}{2}\xi_l^n\Delta t\, c_{10} + \frac{1}{3}(\xi_l^n\Delta t)^2 c_{20} + \frac{1}{3}^\dagger\xi_l^n\eta_l^n\Delta t^2\, c_{11}.
\end{aligned} \tag{13.85}$$

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,

$$\begin{aligned}
\overline{T_d^n} &= d_{00} - \frac{1}{2}\eta_d^n\Delta t\, d_{01} + \frac{1}{3}(\eta_d^n\Delta t)^2 d_{02} \\
&\quad -\frac{1}{2}\xi_d^n\Delta t\, d_{10} + \left[\frac{1}{12}\Delta\mu_L^2 + \frac{1}{3}^\ddagger(\xi_d^n\Delta t)^2\right] d_{20} + \frac{1}{3}^\S\xi_d^n\eta_d^n\Delta t^2 d_{11},
\end{aligned} \tag{13.86}$$

where

$$\xi_d^n = \frac{u_d^n}{h_{\mu d}}, \quad \eta_d^n = \frac{v_d^n}{h_{\psi d}}. \tag{13.87}$$

* $\frac{1}{2}$ in CCSR model
† $\frac{1}{4}$ in CCSR model
‡ $\frac{1}{2}$ in CCSR model
§ $\frac{1}{4}$ in CCSR model

Therefore,

$$\begin{aligned}\tilde{T_l^n} &= c_{00} - \frac{1}{2}\xi_l^n \Delta t\, c_{10} + \left[\frac{1}{3}(\xi_l^n \Delta t)^2 - \frac{\Delta\mu_L^2}{12}\right] c_{20} \\ &\quad -\frac{1}{2}\eta_l^n \Delta t\, c_{01} + \frac{1}{3}(\eta_l^n \Delta t)^2 c_{02} + \frac{1}{3}\xi_l^n \eta_l^n \Delta t^2\, c_{11}, \qquad (13.88)\\ \tilde{T_d^n} &= d_{00} - \frac{1}{2}\eta_d^n \Delta t\, d_{01} + \left[\frac{1}{3}(\eta_d^n \Delta t)^2 - \frac{\Delta\psi_L^2}{12}\right] d_{02} \\ &\quad -\frac{1}{2}\xi_d^n \Delta t\, d_{10} + \frac{1}{3}(\xi_d^n \Delta t)^2 d_{20} + \frac{1}{3}\xi_d^n \eta_d^n \Delta t^2\, d_{11}. \qquad (13.89)\end{aligned}$$

Finally, we describe how to derive the boundary conditions. Since the face values of the tracers are calculated through the second order interpolation, the value of a tracer at a point over land is sometimes necessary. For that case, the value should be appropriately decided by using the tracer values at the neighboring points in the sea. Since ocean models generally assume that there is no flux of tracers across land-sea boundary, the provisional value over land should be given so as not to create a normal gradient at the boundary.

When the face value of a tracer at boundary **l** is calculated, **W** and **L** are not land, but either **N** or **S** may be land, and either **NW** or **SW** may be land. When **N** or **S** is land, the land-sea boundary runs at the center of **L** in the zonal direction. It is reasonable to assume that the value of land grid **N** or **S** must not cause any meridional tracer gradient at **L** set by second order interpolation using the values at grids **N**, **L**, and **S**. Thus, we set

$$(T_N^n - T_L^n)\Delta\psi_d^2 = (T_S^n - T_L^n)\Delta\psi_u^2. \qquad (13.90)$$

When **NW** or **SW** is a land grid, the following should be assumed.

$$(T_{NW}^n - T_W^n)\Delta\psi_d^2 = (T_{SW}^n - T_W^n)\Delta\psi_u^2 \qquad (13.91)$$

When **WW** is a land grid,

$$(T_{WW}^n - T_W^n)\Delta\mu_l^2 = (T_L^n - T_W^n)\Delta\mu_{ll}^2. \qquad (13.92)$$

When **E** is a land grid,

$$(T_E^n - T_L^n)\Delta\mu_l^2 = (T_W^n - T_L^n)\Delta\mu_r^2. \qquad (13.93)$$

Similar boundary conditions are specified for face **d**.

13.3 Second Order Moment (SOM) scheme

13.3.1 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:

$$f(x,y,z) = a_0 + a_x x + a_{xx}x^2 + a_y y + a_{yy}y^2 + a_z z + a_{zz}z^2 + a_{xy}xy + a_{yz}yz + a_{zx}zx. \qquad (13.94)$$

Prather (1986) expressed the above as a sum of orthogonal functions $K_i(x,y,z)$;

$$f(x,y,z) = m_0 K_0 + m_x K_x + m_{xx}K_{xx} + m_y K_y + m_{yy}K_{yy} + m_z K_z + m_{zz}K_{zz} + m_{xy}K_{xy} + m_{yz}K_{yz} + m_{zx}K_{zx}, \qquad (13.95)$$

where the orthogonal functions are given as follows:

$$\begin{aligned}
K_0 &= 1, \\
K_x(x) &= x - X/2, \\
K_{xx}(x) &= x^2 - Xx + X^2/6, \\
K_y(y) &= y - Y/2, \\
K_{yy}(y) &= y^2 - Yy + Y^2/6, \\
K_z(z) &= z - Z/2, \\
K_{zz}(z) &= z^2 - Zz + Z^2/6, \\
K_{xy}(x,y) &= (x - X/2)(y - Y/2), \\
K_{yz}(y,z) &= (y - Y/2)(z - Z/2), \\
K_{zx}(z,x) &= (z - Z/2)(x - X/2),
\end{aligned} \tag{13.96}$$

and

$$\int K_i K_j dV = 0 \quad (i \neq j). \tag{13.97}$$

The constants for normalization are decided using

$$\int K_x^2 dV = VX^2/12, \quad \int K_{xx}^2 dV = VX^4/180,$$

$$\int K_y^2 dV = VY^2/12, \quad \int K_{yy}^2 dV = VY^4/180,$$

$$\int K_z^2 dV = VZ^2/12, \quad \int K_{zz}^2 dV = VZ^4/180,$$

$$\int K_{xy}^2 dV = VX^2Y^2/144, \quad \int K_{yz}^2 dV = VY^2Z^2/144, \quad \int K_{zx}^2 dV = VZ^2X^2/144.$$

The moments are set by the following expressions:

$$\begin{aligned}
S_0 &= \int f(x,y,z) K_0 dV = m_0 V, \\
S_x &= (6/X) \int f(x,y,z) K_x(x) dV = m_x VX/2, \\
S_{xx} &= (30/X^2) \int f(x,y,z) K_{xx}(x) dV = m_{xx} VX^2/6, \\
S_y &= (6/Y) \int f(x,y,z) K_y(y) dV = m_y VY/2, \\
S_{yy} &= (30/Y^2) \int f(x,y,z) K_{yy}(y) dV = m_{yy} VY^2/6, \\
S_z &= (6/Z) \int f(x,y,z) K_z(z) dV = m_z VZ/2, \\
S_{zz} &= (30/Z^2) \int f(x,y,z) K_{zz}(x) dV = m_{zz} VZ^2/6, \\
S_{xy} &= (36/XY) \int f(x,y,z) K_{xy}(x,y) dV = m_{xy} VXY/4, \\
S_{yz} &= (36/YZ) \int f(x,y,z) K_{yz}(y,z) dV = m_{yz} VYZ/4, \\
S_{zx} &= (36/ZX) \int f(x,y,z) K_{zx}(z,x) dV = m_{zx} VZX/4.
\end{aligned} \tag{13.98}$$

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 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,

$$X-ct \le x \le X,\ \ 0 \le y \le Y,\ \ 0 \le z \le Z, \tag{13.99}$$

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,

$$0 \le x \le X-ct,\ \ 0 \le y \le Y,\ \ 0 \le z \le Z, \tag{13.100}$$

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:

$$\begin{aligned}
{K_0}^L = {K_0}^R &= 1,\\
{K_x}^L = x-(X-ct)/2,\ \ {K_x}^R &= x-(2X-ct)/2,\\
{K_{xx}}^L &= x^2-(X-ct)x+(X-ct)^2/6,\\
{K_{xx}}^R &= x^2-(2X-ct)x+(X-ct)X+(ct)^2/6,\\
{K_y}^L = {K_y}^R &= y-Y/2,\\
{K_{yy}}^L = {K_{yy}}^R &= y^2-Yy+Y^2/6,\\
{K_{xy}}^L &= [x-(X-ct)/2](y-Y/2),\\
{K_{xy}}^R &= [x-(2X-ct)/2](y-Y/2).
\end{aligned} \tag{13.101}$$

The basic quantities for calculating the moments are

$$\begin{aligned}
{m_0}^R &= m_0+\bar{K}_x^R m_x+\bar{K}_{xx}^R m_{xx},\\
{m_x}^R &= m_x+2\bar{K}_x^R m_{xx},\\
{m_{xx}}^R &= m_{xx},\\
{m_y}^R &= m_y+2\bar{K}_x^R m_{xy},\\
{m_{yy}}^R &= m_{yy},\\
{m_{xy}}^R &= m_{xy},
\end{aligned} \tag{13.102}$$

where $\bar{K}$ is the average of the new orthogonal function:

$$\begin{aligned}
\bar{K}_x^L = -ct/2,\ \ \bar{K}_x^R &= (X-ct)/2,\\
\bar{K}_{xx}^L = ct(2ct-X)/6,\ \ \bar{K}_{xx}^R &= (X-ct)(X-2ct)/6.
\end{aligned}$$

The moments in the right part to be removed are expressed as follows:

$$\begin{aligned}
{S_0}^R &= \alpha[S_0+(1-\alpha)S_x+(1-\alpha)(1-2\alpha)S_{xx}],\\
{S_x}^R &= \alpha^2[S_x+3(1-\alpha)S_{xx}],\\
{S_{xx}}^R &= \alpha^3 S_{xx},\\
{S_y}^R &= \alpha[S_y+(1-\alpha)S_{xy}],\\
{S_{yy}}^R &= \alpha S_{yy},\\
{S_{xy}}^R &= \alpha^2 S_{xy},
\end{aligned} \tag{13.103}$$

where $\alpha = \alpha^R = ct/X = V^R/V$. The moments in the remaining part are expressed as follows:

$$\begin{aligned}
{S_0}^L &= (1-\alpha)[S_0 - \alpha S_x - \alpha(1-2\alpha)S_{xx}], \\
{S_x}^L &= (1-\alpha)^2(S_x - 3\alpha S_{xx}), \\
{S_{xx}}^L &= (1-\alpha)^3 S_{xx}, \\
{S_y}^L &= (1-\alpha)(S_y - \alpha S_{xy}), \\
{S_{yy}}^L &= (1-\alpha)S_{yy}, \\
{S_{xy}}^L &= (1-\alpha)^2 S_{xy}.
\end{aligned} \tag{13.104}$$

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:

$$S_0 = {S_0}^R + {S_0}^L,$$

$$S_x = \alpha {S_x}^R + (1-\alpha){S_x}^L + 3[(1-\alpha){S_0}^R - \alpha {S_0}^L],$$

$$S_{xx} = \alpha^2 {S_{xx}}^R + (1-\alpha)^2 {S_{xx}}^L + 5\{\alpha(1-\alpha)({S_x}^R - {S_x}^L) + (1-2\alpha)[(1-\alpha){S_0}^R - \alpha {S_0}^L]\},$$

$$S_y = {S_y}^R + {S_y}^L,$$

$$S_{yy} = {S_{yy}}^R + {S_{yy}}^L,$$

$$S_{xy} = \alpha {S_{xy}}^R + (1-\alpha){S_{xy}}^L + 3[(1-\alpha){S_y}^R - \alpha {S_y}^L],$$

where

$$\alpha = \alpha^R = V^R/(V^R + V^L). \tag{13.105}$$

The original moments are conserved through these operations.

Flux limiter

Some limiters are necessary to guarantee that the tracer is positive (negative) definite. 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,

$$S_0 \geq 0, \tag{13.106}$$

$$S_x{}' = \min[+1.5S_0, \max(-1.5S_0, S_x)], \tag{13.107}$$

$${S_{xx}}' = \min[2S_0 - |S_x{}'|/3, \max(|S_x{}'| - S_0, S_{xx})], \tag{13.108}$$

$${S_{xy}}' = \min[+S_0, \max(-S_0, S_{xy})]. \tag{13.109}$$

It should be noted that the application of this limiter does not completely guarantee the tracer will be positive (negative) definite.

13.3.2 Calculating SOM advection in MRI.COM

It should be noted that the coordinate system is not Cartesian in ocean models. Since the coordinate system covers a spherical surface, the x direction in a grid cell is not identical to that in the adjacent cell, for instance.

Thus, the exact conservation of moments cannot be realized. In addition, a tracer-cell including solid earth (sea floor or lateral boundary) is not a cuboid, so the orthogonal functions cannot be defined precisely for such a cell. Nevertheless, the procedures described in the previous subsection can be carried out using the volume of seawater in the non-cuboid grid cell, and the zeroth moment S_0 (total amount of the tracer) is conserved.

As indicated in the expressions in the previous subsections, the volume-integrated moments (S_i) and the fraction of volume to be removed (α) are used in the SOM advection scheme. There are 10 moments for each tracer. The fraction α is calculated using volume transports (`USTARL`, `VSTARL`, and `WLWL`), which are calculated in the subroutine `cont`). Following Prather (1986), the procedures in three directions are executed in order, not simultaneously. 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 procedure in the SOM scheme does not calculate the flux of the tracer across the boundary of grid cells, unlike other advection schemes. The change in the tracer value caused by SOM advection is estimated in the subroutine `tracer` and added to the variable `trcal` directly.

NAMELIST

When the SOM advection scheme is used, the following namelist (`njobsom`) is required (see Table 13.1) on execution.

Table 13.1. Variables defined in namelist `njobsom`

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 *
limiter(numtrc)	flag to set limits of the moments	.true. / .false.
lrstinsom	flag to read the restart file for the initial state	.true. / .false.
lrstoutsom	flag to write the restart file for the final state	.true. / .false.
lsommonitor	flag to monitor the conservation of the moments	.true. / .false.

* i={x, xx, y, yy, z, zz, xy, yz, zx}, n={01, 02, ..., numtrc}; tracer number

References

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.

Leonard, B. P., M. K. MacVean, and A. P. Lock, 1993: Positivity-Preserving Numerical Schemes for Multidimensional Advection, *NASA Tech. Memo.*, *106055, ICOMP-93-05*, 62pp.

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*, 27pp.

Prather, M. J., 1986: Numerical advection by conservation of second-order moments, *J. Geophys. Res.*, *91*, 6671-6681.

Chapter 14 Generalized orthogonal curvilinear coordinate grids

This chapter introduces general orthogonal curvilinear coordinates and presents related calculus.

14.1 Outline

An ocean model 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 written based on the generalized orthogonal coordinate system, the geographic 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 of the model in Section 14.2. 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 the 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 14.3. We can use the functions : $(\lambda, \phi) \Longleftrightarrow (\mu, \psi)$ to convert a scalar quantity as well as sea depth and the Coriolis parameter. The total flux that the ocean receives from the atmosphere should be equal to the total flux that the atmosphere gives to the ocean. The method for conserving the total flux is explored in Section 14.4. The vector operation in a generalized orthogonal coordinate system is concisely described in Section 14.5.

14.2 Generation of orthogonal coordinate system using conformal mapping

We designate the plane that touches the sphere at the North Pole as $\mathbf{S}_\mathrm{N}$. A polar stereographic projection is a conformal transformation in the general sense, so that an orthogonal coordinate system on the sphere is mapped onto an orthogonal coordinate system on $\mathbf{S}_\mathrm{N}$ and the orthogonality is preserved on the reverse transformation (Figure 14.1). Moreover, if $\mathbf{S}_\mathrm{N}$ is assumed to be a complex plane, various conformal transformations can be defined on it. Therefore, applying (i) the polar stereographic projection, (ii) a conformal transformation on the complex plane $\mathbf{S}_\mathrm{N}$, and (iii) the reverse polar stereographic projection to a geographic coordinate grid point (λ, ϕ) on the sphere, an orthogonal coordinate grid point on the sphere can be obtained (Bentzen et al., 1999).

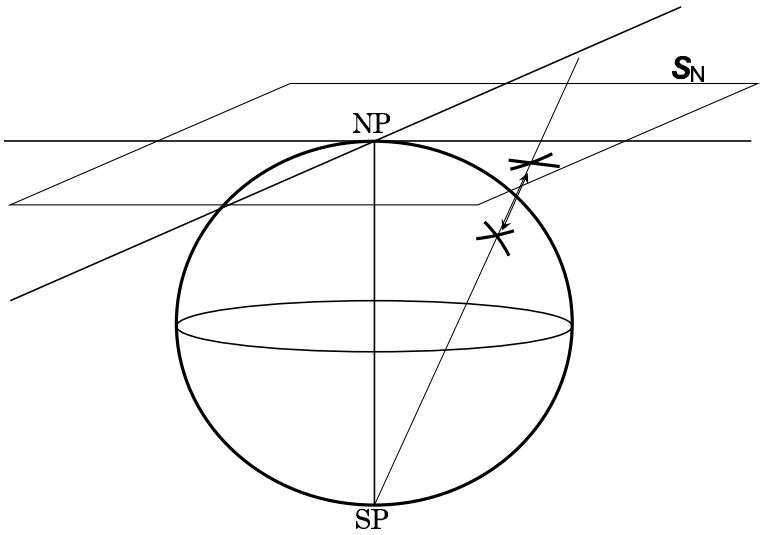

Figure 14.1. Schematic illustration of a Polar stereographic projection (a conformal transformation in the general sense between the sphere and $\mathbf{S}_\mathrm{N}$).

The functions $\mu(\lambda, \phi)$ and $\psi(\lambda, \phi)$ are obtained by the following procedure:

1. From a point (λ, ϕ) on the sphere to a point z on $\mathbf{S}_\mathrm{N}$ (polar stereographic projection). Defining colatitude $\phi' = \pi/2 - \phi$,

$$z = \tan\left(\frac{\phi'}{2}\right) e^{i\lambda}, \tag{14.1}$$

   where the origin of $\mathbf{S}_\mathrm{N}$ corresponds to $\phi' = 0$ ($\phi = \pi/2$), and the positive part of the real axis corresponds to $\lambda = 0$.

2. Conformal mapping M_C on $\mathbf{S}_\mathrm{N}$:

$$\zeta = M_\mathrm{C}(z). \tag{14.2}$$

3. From a point ζ on $\mathbf{S}_\mathrm{N}$ to a point (μ,ψ) on the sphere (reverse polar stereographic projection).

$$\mu = \arg(\zeta), \tag{14.3}$$
$$\psi' = 2\arctan|\zeta|, \tag{14.4}$$
$$\psi = \pi/2 - \psi'. \tag{14.5}$$

Functions $\lambda(\mu,\psi)$ and $\phi(\mu,\psi)$ are obtained by reversing the above procedure.

Defining $\psi' = \pi/2 - \psi$,

$$\zeta = \tan\left(\frac{\psi'}{2}\right) e^{i\mu}, \tag{14.6}$$
$$z = M_\mathrm{C}^{-1}(\zeta), \tag{14.7}$$

and

$$\lambda = \arg(z), \tag{14.8}$$
$$\phi' = 2\arctan|z|, \tag{14.9}$$
$$\phi = \pi/2 - \phi'. \tag{14.10}$$

Thus, when a model coordinate grid point, $(\mu_0 + \Delta\mu \times \mathtt{i}, \psi_0 + \Delta\psi \times \mathtt{j})$ is given, we know the geographic position of the point, Coriolis parameter, etc., at once.

Bentzen et al. (1999) used the linear fraction conversion as a conformal transformation on $\mathbf{S}_\mathrm{N}$. That is,

$$\zeta = M_\mathrm{C}(z) = \frac{(z-a)(b-c)}{(c-a)(b-z)}, \tag{14.11}$$

where the three complex numbers a, b, and c expressed by

$$a = \tan\left(\frac{\phi'_a}{2}\right) e^{i\lambda_a}, \quad b = \tan\left(\frac{\phi'_b}{2}\right) e^{i\lambda_b}, \quad c = \tan\left(\frac{\phi'_c}{2}\right) e^{i\lambda_c}, \tag{14.12}$$

correspond to the three geographic coordinate grid points (λ_a,ϕ_a), (λ_b,ϕ_b), and (λ_c,ϕ_c), which are mapped to the model coordinate grid points, $(\mu,\psi) = (0,\pi/2)$, $(0,-\pi/2)$, $(0,0)$, respectively. Therefore, the singular point $(\mu,\psi) = (0,\pi/2)$ in the model calculation can be put on Greenland, by setting (λ_a,ϕ_a) at 75°N and 40°W. If the option `TRIPOLAR` or `JOT` 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 (`NPLON`, `NPLAT`, `SPLON`, and `SPLAT` in degree) that should be specified in `configure.in`.

When option `TRIPOLAR` is specified, the parameters are set to $\phi_a = \phi_b = 64$°N, $\lambda_a = 80$°E, and $\lambda_b = 100$°W. The transformed grids are used for the region north of 64°N, and the 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 geographical coordinates south of 64°N enables us to do the assimilation and analysis with relative ease (Figure 14.2).

When option `JOT` is specified, the Joukowski conversion is used as a conformal transformation on $\mathbf{S}_\mathrm{N}$. That is,

$$z = M_C^{-1}(\zeta) = \left(\zeta + \frac{\psi_0'^2}{\zeta}\right) e^{i\mu_0}. \tag{14.13}$$

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 14.2 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

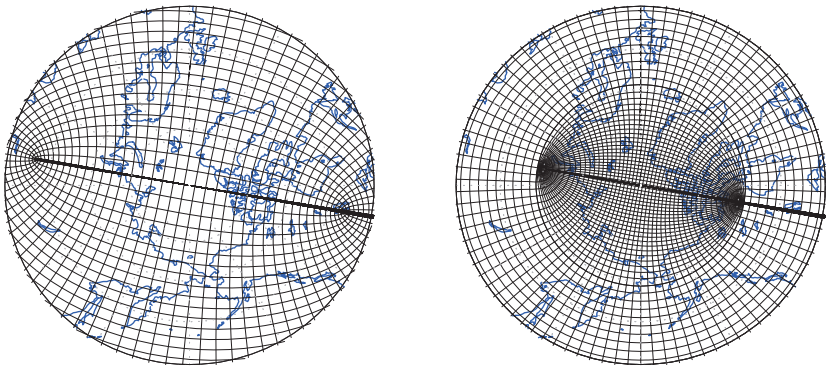

Figure 14.2. Model coordinate grid arrangement in the Arctic sea. Left: Grid system made through the Joukowski conversion (`JOT`). Right: Combination of the coordinates made through the linear fraction conversion and conventional geographic coordinate (`TRIPOLAR`).

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.

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.

14.3 Rotation of vector

A vector expressed in geographic coordinates (λ,ϕ) should be rotated when observed from model coordinates (μ,ψ).

First, we set

$$z = f(\zeta), \quad z = x+iy, \quad \zeta = u+iv, \tag{14.14}$$

$$f'(\zeta) \quad = \quad \frac{\partial x}{\partial u} + i\frac{\partial y}{\partial u} \quad = \quad \frac{\partial y}{\partial v} - i\frac{\partial x}{\partial v}. \tag{14.15}$$

At a certain point $z_0 = f(\zeta_0)$, the angle θ at which a curve $v = v_0$ meets a straight line $y = y_0$ is given by

$$\tan\theta = \left[\frac{\partial y}{\partial x}\right]_{v_0} = \left[\frac{\partial y}{\partial u} \Big/ \frac{\partial x}{\partial u}\right]_{v_0}.$$

Then (see Figure 14.3)

$$\theta = \arg(f'(\zeta_0)). \tag{14.16}$$

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 angle $-\mu_0$ and at point z_0 the meridian $(\lambda = \lambda_0)$ meets the curve $(v = v_0)$ at angle $\lambda_0 - \theta$. The meridian $(\lambda = \lambda_0)$

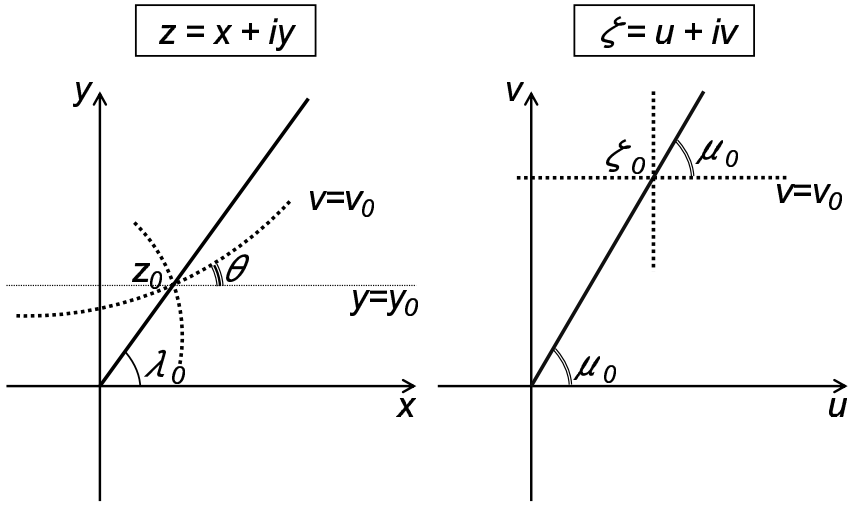

Figure 14.3. A meridian in geographical coordinate (λ, ϕ) (left) and a meridian in model coordinate (μ, ψ) (right).

in λ-ϕ coordinates meets the line $(\mu = \mu_0)$ in μ-ψ coordinates at angle α given as follows::

$$\begin{aligned} \alpha &= -\mu_0 + \lambda_0 - \theta \\ &= \lambda_0 - \mu_0 - \arg(f'(\zeta_0)). \end{aligned} \tag{14.17}$$

Subroutine `rot_mp2lp` defined in `trnsfrm.*.F90` returns $(\cos\alpha, \sin\alpha)$ at a specified grid point of the model. A wind stress vector (τ_x, τ_y) in geographical coordinates should appear in the model ocean described in the μ-ψ coordinate system as $(\tau_x \cos\alpha - \tau_y \sin\alpha, \tau_x \sin\alpha + \tau_y \cos\alpha)$.

14.4 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 14.4). 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.

Grids (I,J) and (M,N) are suitably subdivided into finer grids (i,j) and (m,n). We call these filter grids. A quantity $G'_{i,j}$ is assumed to be homogeneously distributed in the geographic filter grids (i,j) covered by grid (I,J),

$$G'_{i,j} = G_{I,J}.$$

Assume the quantity at a model filter grid $H'_{m,n}$ is equal to that at the nearest geographic filter grid,

$$H'_{m,n} = G'_{i(m,n),j(m,n)}.$$

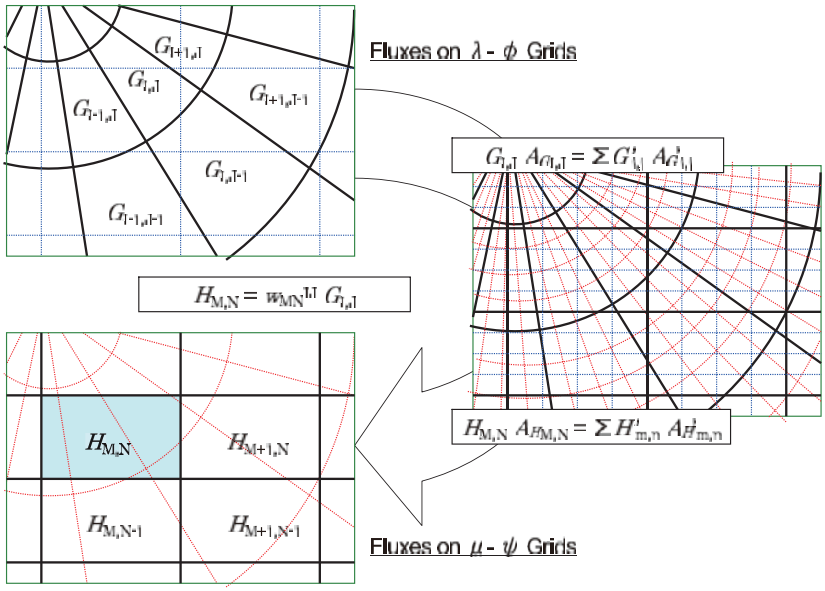

Figure 14.4. Grids (I,J) and (M,N) subdivided into finer grids (i,j) and (m,n).

The quantity at model grid (M,N) is obtained as the area-weighted average:

$$H_{M,N} = \frac{1}{A_{HM,N}} \sum_{m,n} A_{H'm,n} H'_{m,n}, \tag{14.18}$$

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,

$$H_{M,N} = \sum_{I,J} w(M,N,I,J) G_{I,J}. \tag{14.19}$$

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 (14.19) 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 (14.19), to avoid the time-consuming flux conversions in the model calculation.

14.5 Vector operation and differentiation in a general orthogonal coordinate system

To formulate the model equations, we have to know the vector operation and differentiation in general orthogonal coordinates. Some basic formulae used in formulating primitive equations are presented here.

The line element vector $\delta\mathbf{x}$ at a certain point (μ,ψ,r) in an arbitrary general orthogonal coordinate system is expressed as

$$\delta\mathbf{x} = h_\mu \delta\mu \mathbf{e}_\mu + h_\psi \delta\psi \mathbf{e}_\psi + h_r \delta r \mathbf{e}_r, \tag{14.20}$$

where basis vectors $\mathbf{e}_\mu$, $\mathbf{e}_\psi$, and $\mathbf{e}_r$ are mutually orthogonal unit vectors, and h_μ, h_ψ, and h_r are scale factors.

Defining

$$\nabla = \frac{\mathbf{e}_\mu}{h_\mu}\frac{\partial}{\partial\mu} + \frac{\mathbf{e}_\psi}{h_\psi}\frac{\partial}{\partial\psi} + \frac{\mathbf{e}_r}{h_r}\frac{\partial}{\partial r}, \tag{14.21}$$

the gradient of scalar $A(\mu,\psi,r)$ is

$$\nabla A = \frac{\mathbf{e}_\mu}{h_\mu}\frac{\partial A}{\partial\mu} + \frac{\mathbf{e}_\psi}{h_\psi}\frac{\partial A}{\partial\psi} + \frac{\mathbf{e}_r}{h_r}\frac{\partial A}{\partial r}, \tag{14.22}$$

and the divergence of vector $\mathbf{A} = A_\mu\mathbf{e}_\mu + A_\psi\mathbf{e}_\psi + A_r\mathbf{e}_r$ is

$$\nabla\cdot\mathbf{A} = \frac{1}{h_\mu h_\psi h_r}\left[\frac{\partial(h_\psi h_r A_\mu)}{\partial\mu} + \frac{\partial(h_r h_\mu A_\psi)}{\partial\psi} + \frac{\partial(h_\mu h_\psi A_r)}{\partial r}\right]. \tag{14.23}$$

The r component of curl$\mathbf{A}$ is

$$\frac{1}{h_\mu h_\psi}\left[\frac{\partial(h_\psi A_\psi)}{\partial\mu} - \frac{\partial(h_\mu A_\mu)}{\partial\psi}\right]. \tag{14.24}$$

The calculation of velocity advection includes $(\mathbf{a}\cdot\nabla)\mathbf{A}$, where $\mathbf{a}$ is an arbitrary vector ($\mathbf{a} = a_\mu\mathbf{e}_\mu + a_\psi\mathbf{e}_\psi + a_r\mathbf{e}_r$). The μ component of $(\mathbf{a}\cdot\nabla)\mathbf{A}$ is

$$\mathbf{a}\cdot\nabla A_\mu + \frac{A_\psi}{h_\mu h_\psi}\left(a_\mu\frac{\partial h_\mu}{\partial\psi} - a_\psi\frac{\partial h_\psi}{\partial\mu}\right) + \frac{A_r}{h_r h_\mu}\left(a_\mu\frac{\partial h_\mu}{\partial r} - a_r\frac{\partial h_r}{\partial\mu}\right). \tag{14.25}$$

The second and third terms are so-called 'metric' terms in the equation of motion in the spherical coordinates.

These expressions in spherical coordinates (λ,ϕ,r) are shown next. Defining longitude λ, latitude ϕ, and radius of the earth r, scale factors are $h_\lambda = r\cos\phi$, $h_\phi = r$, and $h_r = 1$.
Velocity vector $\mathbf{v}$ is

$$\mathbf{v} = u\mathbf{e}_\lambda + v\mathbf{e}_\phi + w\mathbf{e}_r, \tag{14.26}$$

where $\mathbf{e}_\lambda$, $\mathbf{e}_\phi$, and $\mathbf{e}_r$ are the eastward, northward, and upward unit vectors, respectively, and $(u,v,w) = (r\cos\phi\dot{\lambda}, r\dot{\phi}, \dot{r})$. The gradient of scalar function $A(\lambda,\phi,r)$ is,

$$\nabla A = \frac{\mathbf{e}_\lambda}{r\cos\phi}\frac{\partial A}{\partial\lambda} + \frac{\mathbf{e}_\phi}{r}\frac{\partial A}{\partial\phi} + \mathbf{e}_r\frac{\partial A}{\partial r}, \tag{14.27}$$

where

$$\nabla = \frac{\mathbf{e}_\lambda}{r\cos\phi}\frac{\partial}{\partial\lambda} + \frac{\mathbf{e}_\phi}{r}\frac{\partial}{\partial\phi} + \mathbf{e}_r\frac{\partial}{\partial r}. \tag{14.28}$$

For vector $\mathbf{A} = A_\lambda \mathbf{e}_\lambda + A_\phi \mathbf{e}_\phi + A_r \mathbf{e}_r$,
divergence is

$$\nabla \cdot \mathbf{A} = \frac{1}{r\cos\phi}\left[\frac{\partial A_\lambda}{\partial \lambda} + \frac{\partial(\cos\phi A_\phi)}{\partial \phi}\right] + \frac{\partial(r^2 A_r)}{r^2 \partial r}. \tag{14.29}$$

and the r component of curl$\mathbf{A}$ is

$$[\text{curl}\mathbf{A}]_r = \frac{1}{r\cos\phi}\left[\frac{\partial A_\phi}{\partial \lambda} - \frac{\partial(\cos\phi A_\lambda)}{\partial \phi}\right]. \tag{14.30}$$

The λ component of $(\mathbf{a}\cdot\nabla)\mathbf{A}$ is

$$[(\mathbf{a}\cdot\nabla)\mathbf{A}]_\lambda = \mathbf{a}\cdot\nabla A_\lambda - \frac{A_\phi a_\lambda \tan\phi}{r} + \frac{A_r a_\lambda}{r}. \tag{14.31}$$

The Coriolis force in a generalized orthogonal coordinates (μ, ψ, r) system is given as

$$2\Omega \times \mathbf{v} = (2\Omega_\psi w - 2\Omega_r v)\mathbf{e}_\mu + (2\Omega_r u - 2\Omega_\mu w)\mathbf{e}_\psi + (2\Omega_\mu v - 2\Omega_\psi u)\mathbf{e}_r, \tag{14.32}$$

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 the geographic coordinate (λ, ϕ, r) system.

References

Bentzen, M., G. Evensen, H. Drange, and A. D. Jenkins, 1999: Coordinate transformation on a sphere using conformal mapping, *Mon. Weather Rev.*, *127*, 2733-2740.

Chapter 15 Nesting

In MRI.COM, a high-resolution regional model could be embedded in a low-resolution model using a nesting method. Only one-way nesting from a low- to a high-resolution model is implemented. This chapter briefly describes the nesting method employed by MRI.COM, its program structure, and how to construct a set of low- and high-resolution models in nesting.

15.1 Feature

In a set of nested grid models, a high-resolution model is embedded in a low-resolution model, and values at the side boundary of the high-resolution model are given by the low-resolution model. The side boundary data could be exchanged both off-line and on-line. In off-line mode, the data needed to calculate side boundary values are output to files by first running the low-resolution model, and the high-resolution model is executed reading these data and calculating the side boundary values. In on-line mode, the pre-communicator of Scup (simple coupler) by Yoshimura and Yukimoto (2008) is used to exchange data, and the low-resolution and the high-resolution models are run at the same time.

The main region of the high-resolution model (hereinafter called core region, the hatched region of Figure 15.1) is constructed by connecting tracer points (T-points) of the low-resolution model to form a rectangular region. Two rows or columns of velocity grid cells are added outside the core region at each side boundary (hereinafter called margin; the unhatched region just outside the core region in Figure 15.1), and the side boundary condition is imposed by the low-resolution model. Tracers at the boundary of the core region and one grid outside of it and velocities at two grid points outside the core region are replaced every time step using values from the low-resolution model and linear interpolation. These are referred to as the input interface (Figure 15.1).

A problem arises for velocity points of the high-resolution model between an ocean point and a land point of the low-resolution model, since only data at a single point is available for interpolation. In this case, the high-resolution model velocity perpendicular to the coast line is assumed to be zero at the coast and linearly interpolated from the nearest ocean grid point of the low-resolution model. Furthermore, the high-resolution model velocity tangent to the coast line is set to the same value at the nearest ocean grid point of the low-resolution model. By doing so, the interpolated velocity field satisfies the continuity equation.

The boundary values supplied from the low-resolution model should be enough to compute values at the margin of the high-resolution model using linear interpolation. In practice, three columns or rows are output; they are enough to compute the margin as well as the region one grid equivalent of the low-resolution model inside the boundary of the core region. For the northern boundary, T1-T3 and U1-U3 in Figure 15.1 are sent to the high-resolution model.

In this way, prognostic variables are directly replaced by those from the low-resolution model, which 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 given at the boundary, but the integration is quite stable.

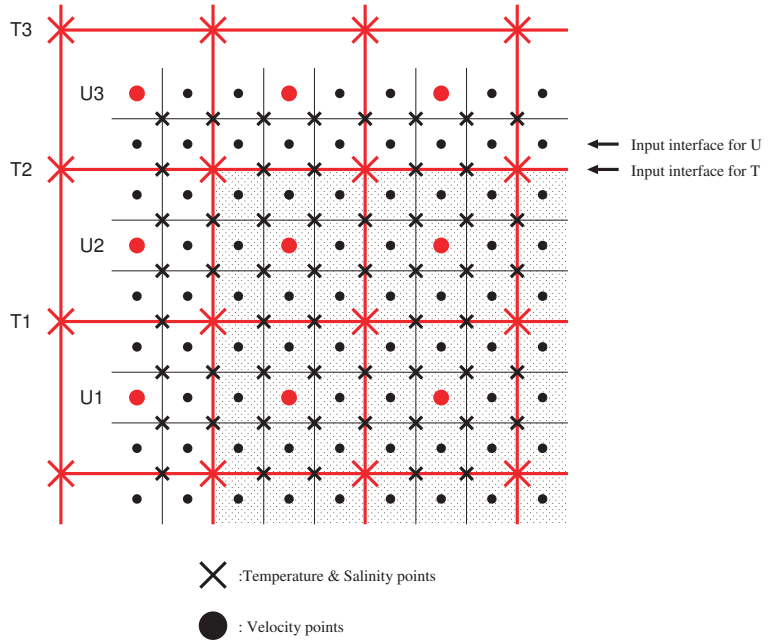

Figure 15.1. Relation between the low-resolution (large symbols) and the high-resolution model (small symbols) model. The hatched region is the main region of the high-resolution model. The tracer points at the boundary of this region and at one grid point outside this boundary and the velocity points at two grid points outside this boundary are replaced by values from the low-resolution model. At the northern boundary, three rows (T1 - T3 and U1 - U3) are sent to the high-resolution model from the low-resolution model.

15.2 Low-resolution model

There is no particular problem in constructing the low-resolution model. Since the grid size ratio between the high- and low-resolution models must be odd, the grid size of the low-resolution model should be determined according to that of the high-resolution model. A grid size ratio of 1:3 or 1:5 is recommended.

In off-line mode, the following files are output from the low-resolution model to be read by the high-resolution model.

- Three rows or columns of baroclinic mode data for each side boundary.
- Three rows or columns of barotropic mode data for each side boundary.
- Land-sea index of the low-resolution model.
- Latitude and longitude of the low-resolution model.

The filenames are set in `parinit.F90` using namelist `outflpar`. In on-line mode, this namelist is not required.

The daily output of side boundary data will work in running the high-resolution model. In on-line mode, side boundary data are sent every time step.

15.3 High-resolution model

15.3.1 Required data

To run the high-resolution model, prepare two data files that contain

- The contribution ratio between low- and high-resolution models around the boundary,
- Two dimensional distribution of horizontal diffusivity and viscosity.

In addition, prepare the following data from the low-resolution model.

- Three rows or columns of baroclinic mode data for each side boundary.
- Three rows or columns of barotropic mode data for each side boundary.
- Land-sea index of the low-resolution model.
- Latitude and longitude of the low-resolution model.

Issues to be carefully considered in creating variable horizontal grid size information and bottom topography are detailed in the next subsection.

Note the following when preparing barotropic boundary data. Since time filtering is used to feedback the result of the barotropic equations to baroclinic modes, the barotropic equations are integrated past the baroclinic time. Thus, the high-resolution model needs future barotropic data in addition to the original output from the low-resolution model. To fulfill this need, prepare barotropic data in one file so that the second data in time sequence can be used as data after the last one for repeating year cycle run. For a historical run, append the second data of the following year of the low-resolution model to the present year data.

15.3.2 Creating data

We briefly describe how to create data that will be read from the high-resolution model.

a. Horizontal grid size

The same grid size as in the low-resolution model should be used in the vertical direction. The high-resolution model is nested in the horizontal directions. The distance between the velocity points of the low-resolution model is divided equally in the high-resolution model. There should be two marginal velocity cells outside the core region at the western and the southern boundary even if the western or the southern boundary is filled by land. However, if the eastern or the northern boundary is filled by land and does not receive boundary data, one marginal velocity cell will be enough.

How to divide low-resolution cells

```
''dxt'' of the low-resolution model (U-point distance: variable name is dxtdeg)
should be divided equally in the high-resolution model.
There should be two marginal U-points in the high-resolution model
(one cell will do if the eastern or northern boundary is filled by land)


 ***** low-resolution model *****

  ifst is the western end grid point number of the low-resolution model
  to be used for interpolation.

                                boundary
                                   |
       (i=ifst)    (i=ifst)   (i=ifst+1) (i=ifst+1)
<------- dxt -------> U <------- dxt -------> U
                                   |
          |-----------+-----------|-----------+-----------|----
                                   |
          T <------- dxu -------> T <------- dxu -------> T
       (i=ifst)    (i=ifst)   (i=ifst+1) (i=ifst+1)
                                   |

 ***** high-resolution model *****
      (nesting ratio is 1:3)
                                   |
                  (1) (1) (2) (2) (3) (3)
                <-dxt->U<-dxt->U<-dxt->U<-dxt->U
                       |---+---|---+---|---+--
                   T<-dxu->T<-dxu->T<-dxu->T
                  (1) (1) (2) (2) (3) (3)
```

How to create a grid size information file (for variable grid size only)

```
! the number of grid points of the high-resolution model
integer(4), parameter :: imut = 535, jmut = 431
integer(4), parameter :: ivgrid = 57 ! arbitrary
real(8) :: dxt(imut), dyt(jmut)      ! dxtdeg, dytdeg in the model

write(unit=ivgrid) (dxt(i),i=1,imut), (dyt(j),j=1,jmut)
```

b. Topography of the high-resolution model

To avoid serious discontinuities in mass transport, the high-resolution model should have the same topography as the low-resolution model around the side boundary of the high-resolution model.

It is recommended that the topography of the high-resolution model should have the same topography as the low-resolution model in the two inner low-resolution velocity cells and the one outer low-resolution velocity cell from the boundary of the core region.

c. Weighting ratio between the low- and high-resolution model

For smoothness, values of the high-resolution model around the side boundary could be given as a weighted average of the low- and high-resolution models. However, we recommend that the weight for the low-resolution model to be unity at the input interface and zero elsewhere.

weighting ratio between the low and high-resolution model

```
! the number of grid points of high-resolution model
integer(4), parameter :: imut = 535, jmut = 431
integer(4), parameter :: ibuffer = 57   ! arbitrary
real(8) :: wbuft(imut,jmut), wbufu(imut,jmut)

write(unit=ibuffer) wbuft, wbufu
```

d. 2-D distribution of diffusion and viscosity coefficient

Diffusion and viscosity coefficients of a high-resolution model should be the same as those of a low-resolution model around the side boundary. The coefficients are made small in the interior.

2-D distribution of diffusion and viscosity coefficient

```
! the number of grid points of the high-resolution model
integer(4), parameter :: imut = 535, jmut = 431
integer(4), parameter :: ihdtsuv = 57   ! arbitrary
real(8) :: hdts(imut,jmut), hduv(imut,jmut)

write(unit=ihdtsuv) hdts, hduv
```

15.4 Usage

15.4.1 Compilation

The model option for the low-resolution model is `PARENT`, and that for the high-resolution model is `SUB`. For on-line mode, the option `NESTONLINE` should be specified along with the name of the model such as `NAME_MODEL` = *modelname*. For a calculation using parallel processors (option `PARALLEL`), only one-dimensional partitioning (in the meridional direction) of model region is supported. The number of zonally partitioned regions (`NPARTX`) must be one. These model options and the model name should be specified in `configure.in`.

The following describes parameters that a user should define before compilation.

a. Low resolution model

The following model parameters should be defined in `parent/parapar.F90` prior to compilation.

```
ifsto = 35   :  the western end point to be used to interpolate data (T-point)
ifedo = 157  :  the eastern end point to be used to interpolate data (T-point)
jfsto = 61   :  the southern end point to be used to interpolate data (T-point)
jfedo = 159  :  the northern end point to be used to interpolate data (T-point)

ipareo = ifedo - ifsto + 1
jpareo = jfedo - jfsto + 1
 ... the total grid number to be sent to the high-resolution model

ioutrts = 5, ioutrtn = 0, ioutrtw = 5, ioutrte = 3
 ... nesting ratio at each side boundary
     south(s), north(n), west(w), east(e)
```

b. High resolution model

Model parameters that should be defined in sub2/parasub.F90 are listed on Table 15.1.

Table 15.1. Parameters to be defined in sub2/parasub.F90

variable name	description
ipmut, jpmut	the number of grid points of the low-resolution model
ifst, ifed, jfst, jfed	define the boundary of the low-resolution model used for interpolation in the high-resolution model (considered in terms of T-points ... should be the same as parapar.F90)
ipare, jpare	the total grid number of the low-resolution model to be used for interpolation in the high-resolution model (... should be the same as parapar.F90)
iinrts, iinrtn, iinrtw, iinrte	nesting ratio at each side boundary
intbcl, intbtr	the time interval in seconds of the boundary data (used only for off-line mode)

```
e.g.
  iinrts=5, iintrn=3, iinrtw=5, iinrte=3
    5 for nesting ratio 5:1
    3 for nesting ratio 3:1
```

```
e.g.,
  intbcl = 21600, intbtr = 21600
```

15.4.2 Running the models

In off-line mode, specify the names of data files and how boundary data are handled in namelist. Users should see parent/parinit.F90 for the low-resolution model and sub2/subinit.F90 for the high-resolution model.

The filename for the weighting ratio and the two-dimensional distribution of horizontal diffusion and viscosity for the high-resolution model (namelist inflsub) should be specified even in on-line mode.

a. Low resolution model

The following namelists are read from `parent/parinit.F90` on execution.

Table 15.2. Namelist for the nested low-resolution model

variable name	group	description	usage
nwrtpc	njobpar	The interval of time steps by which baroclinic mode is written to file	off line only
nwrtpt	njobpar	The interval of time steps by which barotropic mode is written to file	off line only
ircfsto	njobpar	The record number in file at the beginning of the integration (baroclinic mode)	off line only
irtfsto	njobpar	The record number in file at the beginning of the integration (barotropic mode)	off line only
ibfirst	njobpar	Output (1) or no-output(0) the initial state as boundary data	off line only
file_btrs_out	outflpar	the filename of the barotropic mode (southern end)	off line only
file_btrn_out	outflpar	the filename of the barotropic mode (northern end)	off line only
file_btrw_out	outflpar	the filename of the barotropic mode (western end)	off line only
file_btre_out	outflpar	the filename of the barotropic mode (eastern end)	off line only
file_bcls_out	outflpar	the filename of the baroclinic mode (southern end)	off line only
file_bcln_out	outflpar	the filename of the baroclinic mode (northern end)	off line only
file_bclw_out	outflpar	the filename of the baroclinic mode (western end)	off line only
file_bcle_out	outflpar	the filename of the baroclinic mode (eastern end)	off line only
file_lidx_out	outflpar	land-sea index of the low-resolution model	off line only
file_pgrd_out	outflpar	latitude-longitude of the low resolution model	off line only

b. High resolution model

The following namelists are read from `sub2/subinit.F90` on execution.

Table 15.3. namelist for the nested high-resolution model

variable name	group	description	usage
ircfst	njobsub	The record number in file at the beginning of the integration (baroclinic mode)	off line only
irtfst	njobsub	The record number in file at the beginning of the integration (barotropic mode)	off line only
ircend	njobsub	The number of records written in the boundary file (baroclinic mode).	off line only
irtend	njobsub	The number of records written in the boundary file (barotropic mode).	off line only
irtcycle	njobsub	Return to the second record (1) or repeat the last data (0) if the last barotropic data is reached.	off line only
file_btrs	inflp2s	the filename of the barotropic mode (southern end)	off line only
file_btrn	inflp2s	the filename of the barotropic mode (northern end)	off line only

variable name	group	description	usage
file_btrw	inflp2s	the filename of the barotropic mode (western end)	off line only
file_btre	inflp2s	the filename of the barotropic mode (eastern end)	off line only
file_bcls	inflp2s	the filename of the baroclinic mode (southern end)	off line only
file_bcln	inflp2s	the filename of the baroclinic mode (northern end)	off line only
file_bclw	inflp2s	the filename of the baroclinic mode (western end)	off line only
file_bcle	inflp2s	the filename of the baroclinic mode (eastern end)	off line only
file_lidx	inflp2s	land-sea index of the low-resolution model	off line only
file_pgrd	inflp2s	latitude-longitude of the low-resolution model	off line only
file_bwgt	inflsub	weighting ratio between the low and high-resolution model around the boundary	
file_bhdf	inflsub	2-D distribution of diffusion and viscosity of the high-resolution model	

c. On-line mode

In on-line mode, the pre-communicator of Scup (simple coupler) by Yoshimura and Yukimoto (2008) is used to exchange data and the low and high-resolution models are run at the same time. User should tell the coupler how data are exchanged between the low and high-resolution models via a namelist file `NAMELIST_SCUP`. An example of `NAMELIST_SCUP` is listed below. In this example, the model name of the low-resolution model is `GLOBAL` and that of the high-resolution model is `WNP01`.

an example of NAMELIST_SCUP

```
&nam_scup_pre model_put='GLOBAL', model_get='WNP01', type='REAL8' /
&nam_scup_pre var_put='ALONTC',  var_get='ALONTC',  dst_get='ALL' /
&nam_scup_pre var_put='ALATTC',  var_get='ALATTC',  dst_get='ALL' /
&nam_scup_pre var_put='ALONUC',  var_get='ALONUC',  dst_get='ALL' /
&nam_scup_pre var_put='ALATUC',  var_get='ALATUC',  dst_get='ALL' /
&nam_scup_pre var_put='AEXLP',  var_get='AEXLP',  dst_get='ALL' /
&nam_scup_pre var_put='ATXLP',  var_get='ATXLP',  dst_get='ALL' /
&nam_scup_pre var_put='ULS',  var_get='ULS',  dst_get='FIRST' /
&nam_scup_pre var_put='VLS',  var_get='VLS',  dst_get='FIRST' /
&nam_scup_pre var_put='TLS',  var_get='TLS',  dst_get='FIRST' /
&nam_scup_pre var_put='SLS',  var_get='SLS',  dst_get='FIRST' /
&nam_scup_pre var_put='HTS',  var_get='HTS',  dst_get='FIRST' /
&nam_scup_pre var_put='UMS',  var_get='UMS',  dst_get='FIRST' /
&nam_scup_pre var_put='VMS',  var_get='VMS',  dst_get='FIRST' /
&nam_scup_pre var_put='ULN',  var_get='ULN',  dst_get='LAST' /
&nam_scup_pre var_put='VLN',  var_get='VLN',  dst_get='LAST' /
&nam_scup_pre var_put='TLN',  var_get='TLN',  dst_get='LAST' /
&nam_scup_pre var_put='SLN',  var_get='SLN',  dst_get='LAST' /
&nam_scup_pre var_put='HTN',  var_get='HTN',  dst_get='LAST' /
&nam_scup_pre var_put='UMN',  var_get='UMN',  dst_get='LAST' /
&nam_scup_pre var_put='VMN',  var_get='VMN',  dst_get='LAST' /
&nam_scup_pre var_put='ULW',  var_get='ULW',  dst_get='ALL' /
&nam_scup_pre var_put='VLW',  var_get='VLW',  dst_get='ALL' /
&nam_scup_pre var_put='TLW',  var_get='TLW',  dst_get='ALL' /
&nam_scup_pre var_put='SLW',  var_get='SLW',  dst_get='ALL' /
&nam_scup_pre var_put='HTW',  var_get='HTW',  dst_get='ALL' /
&nam_scup_pre var_put='UMW',  var_get='UMW',  dst_get='ALL' /
&nam_scup_pre var_put='VMW',  var_get='VMW',  dst_get='ALL' /
&nam_scup_pre var_put='ULE',  var_get='ULE',  dst_get='ALL' /
&nam_scup_pre var_put='VLE',  var_get='VLE',  dst_get='ALL' /
&nam_scup_pre var_put='TLE',  var_get='TLE',  dst_get='ALL' /
&nam_scup_pre var_put='SLE',  var_get='SLE',  dst_get='ALL' /
&nam_scup_pre var_put='HTE',  var_get='HTE',  dst_get='ALL' /
&nam_scup_pre var_put='UME',  var_get='UME',  dst_get='ALL' /
&nam_scup_pre var_put='VME',  var_get='VME',  dst_get='ALL' /
&nam_scup_pre var_put='ICECATS',  var_get='ICECATS',  dst_get='FIRST' /
&nam_scup_pre var_put='ICECATN',  var_get='ICECATN',  dst_get='LAST' /
&nam_scup_pre var_put='ICECATW',  var_get='ICECATW',  dst_get='ALL' /
&nam_scup_pre var_put='ICECATE',  var_get='ICECATE',  dst_get='ALL' /
&nam_scup_pre var_put='ICEDYNS',  var_get='ICEDYNS',  dst_get='FIRST' /
&nam_scup_pre var_put='ICEDYNN',  var_get='ICEDYNN',  dst_get='LAST' /
&nam_scup_pre var_put='ICEDYNW',  var_get='ICEDYNW',  dst_get='ALL' /
&nam_scup_pre var_put='ICEDYNE',  var_get='ICEDYNE',  dst_get='ALL' /
&nam_scup_pre model_put='WNP01', model_get='GLOBAL', type='REAL8' /
&nam_scup_pre var_put='ATEXSUB',  var_get='ATEXSUB',  dst_get='ALL' /
&nam_scup_pre var_put='TSUB',  var_get='TSUB',  dst_get='ALL' /
&nam_scup_pre var_put='SSUB',  var_get='SSUB',  dst_get='ALL' /
```

15.5 Program structure

The program structure of a low-resolution model is as follows.

```
ogcm__ini
  |
  +- parinit (parinit.F90)        (initialize low-resolution model)
       |
       +- off-line mode
            |
            +- if (ibfirst == 1)  (output boundary data of the initial state)
                 |
                 +- out[snwe]cli
                 |
                 +- out[snwc]tro

ogcm__run
  |
  +-- part_1 (on-line mode)
  |     |
  |     +-- surfce
  |           |
  |           +-surfce_integ
  |               |
  |               +-DO LOOP_N
  |               |    |
  |               |    +- outmrgnt   (send barotropic boundary data
  |               |         |          to high-resolution model)
  |               |         +-- out[snwe]tro
  |               |
  |               +- outmrgnt        (send time-filtered barotropic boundary
  |                    |              data to high-resolution model)
  |                    +-- out[snwe]tro
  |
  +-- part_2 (on-line mode)
  |     |
  |     +-- outmrgn                (send baroclinic boundary data
  |     |     |                      to high-resolution model)
  |     |     +-- out[snwe]cli
  |     |
  |     +-- outmrgnvd              (send viscosity/diffusivity
  |           |                      to high-resolution model)
  |           +-- out[snwe]vd
  |
  +-- outmrgn (off-line mode at the end of each time step)
  |     |
  |     +-- if (mod(nnmats,nwrtpc)== 0)
  |           |
  |           +-- out[snwe]cli
  |
  +-- outmrgnvd (off-line mode at the end of each time step)
  |     |
  |     +-- if (mod(nnmats,nwrtpc)== 0)
  |           |
  |           +-- out[snwe]vd
  |
  +-- outmrgn (off-line mode at the end of each time step)
        |
        +-- if (mod(nnmats,nwrtpt)== 0)
```

```
|
+-- out[snwe]tro
```

The program structure of a high-resolution model is as follows.

```
ogcm__ini
  |
  +- subinit (subinit.F90)   (initialize high-resolution model)
       |
       +- mknlvect           (make list vectors for interpolation
       |                      of boundary data)
       +- mknlvecv
       |
       +- off-line mode
            |
            +- rd[snwe]cli   (read first two baroclinic boundary data)
            |    |
            |    + inplsubuf (interpolation of boundary data
            |    |            to high-resolution model grid points)
            |    + inplsubvf
            |    |
            |    + inplsubtf
            |
            +- rd[snwc]tro   (read first two baroclinic boundary data)
                 |
                 + inplsubus
                 |
                 + inplsubvs
                 |
                 + inplsubts

ogcm__run
  |
  +- part_1
  |    |
  |    +- surfce
  |         |
  |         +-surfce_integ
  |             |
  |             +- DO LOOP_N
  |             |    |
  |             |    +- inpfacet (read/receive barotropic boundary data)
  |             |         |
  |             |         +- rd[snwe]tro
  |             |         |    |
  |             |         |    + inplsubus
  |             |         |    |
  |             |         |    + inplsubvs
  |             |         |    |
  |             |         |    + inplsubts
  |             |         |
  |             |         +- flmrgn[snwe]t (replace barotropic prognistic variables
  |             |                           to those from the low-resolution model)
  |             +- inpfacet      (read/receive time-filtered barotropic boundary data)
  |                  |
  |                  +- rd[snwe]tro
  |                  |    |
  |                  |    + inplsubus
  |                  |    |
  |                  |    + inplsubvs
  |                  |    |
  |                  |    + inplsubts
```

```
|                 |
|                 +- flmrgn[snwe]t
|
+- part_2
     |
     +- inpface              (read/receive baroclinic boundary data
     |    |                   after ``trcimp'' and before ``stable'')
     |    +- rd[snwe]cli
     |    |    |
     |    |    + inplsubuf
     |    |    |
     |    |    + inplsubvf
     |    |    |
     |    |    + inplsubtf
     |    |
     |    +- flmrgn[snwe]c   (replace baroclinic prognistic variables
     |                         to those from the low-resolution model)
     |
     +- inpfacevd            (read/receive viscosity/diffusivity data
          |                   after mixed layer model)
          +- rd[snwe]vd
          |    |
          |    + inplsubtf
          |
          +- flmrgn[snwe]vd  (replace baroclinic prognistic variables
                              to those from the low-resolution model)
```

References

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.

Yoshimura, H., and S. Yukimoto, 2008: Development of simple coupler (Scup) for earth sysyem modeling., *Pap. Meteor. Geophys.*, *59*, 19-29, doi:10.2467/mripapers.59.19.

Chapter 16 User's Guide

This chapter explains the procedure to run MRI.COM using a long term integration of a global model as a standard case. The description in this chapter is based on MRI.COM version 3.0 (MRICOM-3_0-20091130) and may not correspond to the latest version of MRI.COM, since the run procedure (user interface) is continuously updated. It is recommended that users refer to `README.First` in `src` directory when they start setting up the 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 16.1 and Tables 16.1 and 16.2).
- **Input data:** Topography, surface forcing, and climatology to be read at run time (Section 16.2 and Table 16.3).
- **Execution:** An example shell script and namelist (Section 16.3).
- **Post process:** A description of output files (Section 16.4 and Table 16.4).

The input and main output files are listed in Tables 16.1 through 16.4. Note that cgs units are employed to express physical values in the model.

16.1 Model setup

This section describes the procedure necessary to set up the model and compile its programs. First, prepare `configure.in` that contains the information about the model options and grid sizes. In addition, edit several Fortran 90 files to specify parameters corresponding to the selected model options.

Table 16.1. Files used for compilation and their related program files (see Section 16.1)

parameter to be specified	file name	included from
model size, model option, compile option	`configure.in`	`param.F90` via `configure`
vertical resolution	`dz.F90`	`param.F90.in`
vertical diffusion coefficient	`vdbg.F90`	`rdjobp.F90`
grid of surface forcing data (for model option `FLXINTPOL`)	`intpolpar.F90`	`force.F90`
vertical grid of climatological data (for model option `TSINTPOL`)	`depclim.F90`	`tsclim.F90`
3-D grid of climatological data (for model options `TSINTPOL` and `TSCLDIRECT`)	`tsintpol.F90`	`tsclim.direct.F90`

16.1.1 Files needed for compilation

a. configure.in

The model options and configuration parameters should be specified in `configure.in` as follows.

An example `configure.in` for Global tripolar $1^\circ \times 0.5^\circ$ grid model

```
DEFAULT_OPTIONS="ICE ICECAT SIDYN CALALBSI SFLUXW SFLUXR ISOPYCNAL
 SMAGOR VIS9P DIFAJS NOHKIM VVDIMP UTOPIA ULTIMATE ZQUICKEST ZULTIMATE
 BBL HIST HISTFLUX HFLUX TAUBULK WFLUX RUNOFF Y365D CLMFRC LWDOWN
 BULKNCAR BULKITER INILEV CYCLIC VARIABLE TRIPOLAR PARALLEL FREESURFACE"
#
IMUT=364
JMUT=368
KM=51
KSGM=5
KBBL=1
SLAT0=-78.D0
SLON0=0.D0
NPARTX=8
NPARTY=4
DXTDGC=1.0D0
DYTDGC=0.5D0
NPLAT=64.D0
NPLON=80.D0
SPLAT=64.D0
SPLON=260.D0
```

The parameters required for each major option are listed in Table 16.2 (see also `README.Options`).

Table 16.2. model parameters to be set in `configure.in`

option name	variable name	description
default	IMUT, JMUT, KM	zonal/meridional/vertical grid number
	SLON0, SLAT0	longitude/latitude of the first tracer grid point excluding the boundary land or ghost cells
	DXTDGC, DYTDGC	zonal/meridional grid sizes in degree for the case of uniform increment; unless VARIABLE
TRIPOLAR, JOT	NPLAT, NPLON	geographical latitude and longitude of the displaced North Pole
	SPLAT, SPLON	geographical latitude and longitude of the displaced South Pole
FREESURFACE	KSGM	the number layers in the sea surface sigma-layer; see Chapter 4
	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 calculation using parallel processors: the number of parallel processes should be NPARTX $\times$ NPARTY
TSINTPOL	IMT, JMT, KK	3-D size of climatological data
	SLATC, SLONC	latitude/longitude of the first grid point data of climatology
	DLATC, DLONC	grid increment for latitude/longitude of climatology

option name	variable name	description
FLXINTPOL	IMF, JMF	2-D size of sea surface data
(RUNOFF)	IMROF, JMROF	2-D size of river discharge data
	INTPWIND	horizontal interpolation type for the wind data; 1:linear, 2:third order spline
passive tracer	NUMTRC_P	only when any passive tracer is calculated (NUMTRC_P $\geq$ 1)

b. dz.F90

Describe the discretization in the vertical direction in dz.F90. The widths of the discrete cells that fill the vertical column should be written from top to bottom. Usually the depth of the bottom level is determined first, and the column is separated into discrete cells. In general, the levels are finer from the sea surface to about 1000 m and become coarser towards the bottom. Tracer and velocity levels are placed at the middle level of the cell (Chapter 3). The number of vertical cells is described as KM in configure.in. The file dz.F90 is included by param.F90.

Example for dz.F90

```
real(8), parameter :: dz(km) = (/&
  & 3.0d2, 4.0d2, 6.0d2, 8.0d2, 1.0d3, &
       ............
  & 2.5d4, 2.5d4, 3.75d4, 7.50d4 /)
```

c. depclim.F90

For option TSINTPOL, define the information about the depth of climatological temperature and salinity data in depclim.F90. It is used to interpolate the data to model grid points. Enter the number of grid points (IMT, JMT, and ,KK) and horizontal grid-point information (DLATC, DLONC, SLATC, and SLONC) in configure.in (Table 16.2). A sample can be found in directory src/intpolpar.

d. tsintpol.F90

When option TSCLDIRECT is selected in addition to TSINTPOL, define the information about the 3-D grid of climatological temperature and salinity data in tsintpol.F90. With this choice, the data should be prepared in direct access files. As in the default case, enter the number of grid points (IMT, JMT, and KK) and horizontal grid-point information (DLATC, DLONC, SLATC, and SLONC) in configure.in (Table 16.2). But the horizontal grid of data may be defined in tsintpol.F90, overriding the above definition. A sample can be found in directory src/intpolpar.

e. intpolpar.F90

For option FLXINTPOL, define the information about the latitude and longitude of the surface data intpolpar.F90. It is used to interpolate the data to the model grids points for sea surface flux calculation. Enter the number of zonal and meridional grid points (IMF and JMF) in configure.in (Table 16.2). Samples for the datasets of NCEP-NCAR and JRA25 reanalyses and CORE can be found in the directory src/intpolpar.

f. vdbg.F90

The file vdbg.F90 is necessary if the vertical diffusion equation is solved using the implicit method (option VVDIMP). While the vertical diffusion coefficient is replaced by the large one from the mixed layer model and the convection scheme (option DIFAJS), the lower limit (vdbg(1 : km)) should be set in vdbg.F90. The sample of vdbg.F90 and the sample program to prepare it (vdmricom.F90) are in the directory src/vd. Here, the vertical diffusion coefficient used in the GFDL-MOM is shown.

Example for vdbg.F90

```
do k = 1, km
  vdbg(k) = .8d0 + 1.05d0 * datan(4.5d0*(dep(k+1)*1.d-5-2.5d0)) / pi
end do
```

16.1.2 Compilation of the model

The standard compiling script is prepared as compile.sh in the src directory. The part depending on the system (OS, Fortran compiler, and compiler option) in compile.sh and configure have to be edited by users for their computer (hardware) environment.

To compile the programs, execute compile.sh by issuing a shell prompt from the front-end machine. The function of compile.sh is to create param.F90 and Makefile from configure.in, param.F90.in, and Makefile.in 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 param.F90. The Makefile is created from Makefile.in. Finally, make is carried out, and the executable file ogcm is obtained. 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.

16.2 Preparation of input data files for execution

The topographic data (file_topo), temperature and salinity climatological data (file_tscl), and wind stress data (file_wind) are always necessary.

According to user's the specification of model options and job parameters, the following additional data files should be prepared:

- zonal and meridional grid spacing (units in degrees, file_vgrid; for VARIABLE)
- information about the area and the length for each grid (file_scale; except for SPHERICAL)
- information about grid-wise nudging "on" or "off" (file_frc; for namelist parameter iforcev = 1)
- sea surface radiation data (file_hflx; for HFLUX)
- sea surface meteorological data (file_bulk; for HFLUX + selected bulk formula, see Chapter 8)
- precipitation data (file_prcp; for WFLUX)
- river discharge data (file_rnof; for RUNOFF)

Table 16.3. Main input data files and their related program files (see Section 16.2)

subject	file name specified in (namelist)	read from
run-time job parameters (as namelist parameters)	standard input (or redirection from file)	rdjobp.F90 etc.
variable horizontal grid spacing (for the model option VARIABLE)	file_vgrid (inflg)	stmdlp.F90
topography	file_topo (infltopo)	rdbndt.F90
area and length scale for each grid	file_scale (inflscl)	rdbndt.F90
climatological data of temperature and salinity	file_tscl (infltscl)	tsclim.F90
on/off of nudging for tracer points	file_frc (njobbdy)	rstcoef.F90
wind stress	file_wind (inflw)	force.F90
surface radiation data	file_hflx (inflh)	force.F90
surface meteorological data	file_bulk (inflh)	force.F90
precipitation data	file_prcp (inflp)	force.F90
river discharge data	file_rnof (inflo)	force.F90
sea-ice fractional area data	file_icec (inflic)	force.F90
initial values for restart	file_restart_in (infla)	rdinit.F90
(for ice averaged over all categories)	file_ice_restart_in (infli)	mod_seaice.F90
(for ice for each thickness category)	file_icecat_restart_in (inflic)	ice_restart.F90

- sea-ice fractional area data (file_icec; for ICECLIM)

16.2.1 Topographic and grid spacing data

a. Topography

The topographic data consist of the single precision integer array HO4(IMUT, JMUT) that contains the sea floor depths of the velocity grid points (in cm) and the single precision 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)
write(unit=inidt) ho4, exnn
```

For an experiment with a realistic topography, the model topography is usually prepared by averaging the depth data of ETOPO2 over each grid cell. ETOPO2 is 2-minute-grid topography data. An example of the topography for global $1^\circ \times 0.5^\circ$ model created by this method is shown in Figure 16.1.

On 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. The topography data is read from the file file_topo (namelist infltopo) in rdbndt.F90.

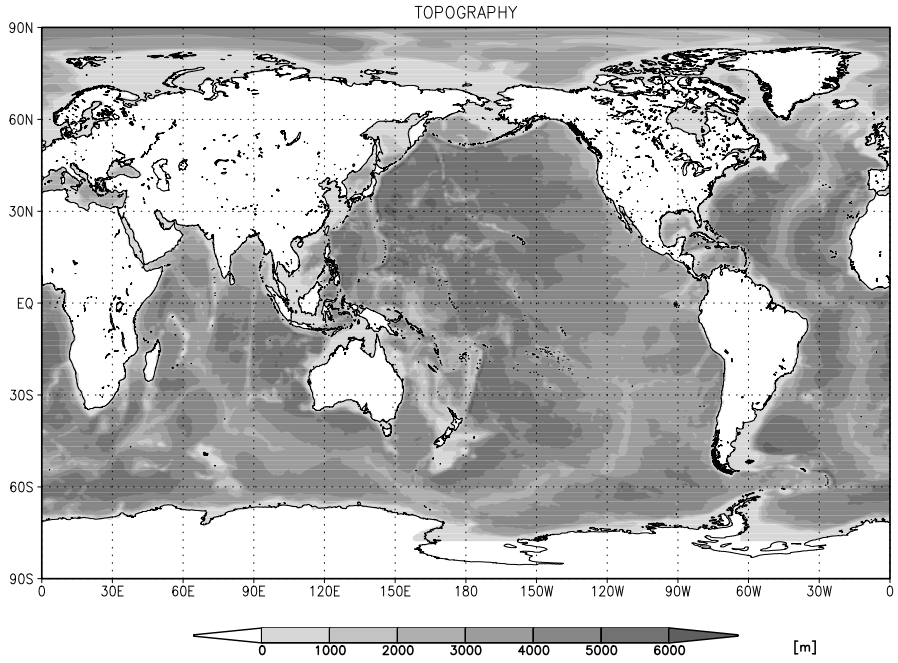

Figure 16.1. Example of ocean model topography (global $1^\circ \times 0.5^\circ$ grid model).

b. Grid spacing

The grid spacing should be prepared when variable grid spacing is used for either the zonal or meridional direction (the model option `VARIABLE`). The units are in degrees. It is read from the file `file_vgrid` (namelist `inflg`) in `stmdlp.F90`.

Format of grid spacing data (file_vgrid; `VARIABLE`)

```
real(8) :: dxtdeg(imut), dytdeg(jmut) ! grid increment for T-point
write(ivgrid) dxtdeg, dytdeg
```

c. Grid cell area and distance

When the model grid points are defined based on the general orthogonal coordinates, the quarter cell area and distance should be prepared. The units are in cgs. It is read from the file `file_scale` (namelist `inflscl`) in `rdbndt.F90`.

When spherical coordinates are used (option `SPHERICAL`), e.g., the grids are defined on geographical latitude and longitude, the grid information is analytically calculated in `stmdlp.F90`, and the file `file_scale` is not necessary.

Format of grid cell area and distance (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)
write(unit=N_T) a_bl  ! U-box area of bottom-left 1/4 grid
write(unit=N_T) a_br  ! U-box area of bottom-right 1/4 grid
write(unit=N_T) a_tl  ! U-box area of top-left 1/4 grid
write(unit=N_T) a_tr  ! U-box area of top-right 1/4 grid
write(unit=N_T) dx_bl ! U-box length of bottom-left 1/4 grid
write(unit=N_T) dx_br ! U-box length of bottom-right 1/4 grid
write(unit=N_T) dx_tl ! U-box length of top-left 1/4 grid
write(unit=N_T) dx_tr ! U-box length of top-right 1/4 grid
write(unit=N_T) dy_bl ! U-box length of bottom-left 1/4 grid
write(unit=N_T) dy_br ! U-box length of bottom-right 1/4 grid
write(unit=N_T) dy_tl ! U-box length of top-left 1/4 grid
write(unit=N_T) dy_tr ! U-box length of top-right 1/4 grid
```

16.2.2 Climatological data

a. Default

By default, define the climatological temperature and salinity data at model grid points (i.e., IMT = IMUT, JMT = JMUT, KK = KM). However, when the option TSINTPOL is selected, data with uniform grid spacing are read and interpolated in the model. In this case, the grid point information (DLATC, DLONC, SLATC, and SLONC) should be specified in configure.in (Table 16.2), and the data grid points (alonc, alatc) are calculated in tsclim.F90.

The climatological data are read from the file file_tscl (namelist infltscl) in tsclim.F90.

Format of climatological data (file_tscl) for default case

```
integer(4), parameter :: imn = 12
real(4) :: ttlev(imt,jmt,kk,imn),tslev(imt,jmt,kk,imn)

do m = 1, imn
  write(unit=itscl) (((ttlev(i,j,k),i=1,imt),j=1,jmt),k=1,kk), &
   &                (((tslev(i,j,k),i=1,imt),j=1,jmt),k=1,kk)
enddo
```

b. Option TSCLDIRECT

When option TSCLDIRECT is selected, data should be prepared in direct access files. Monthly data should be prepared in separate files as follows.

At run time, another namelist njobpts is required to handle data. The variable itsrepeat specifies whether the parepared data are repeatedly used (=1) or interannual variations are assumed (=0). When itsrepeat = 0, the month 00 (13) means the December (January) of the previous (next) year. The variable itsmonfst specifies the first data for "this" run, the start time is between mid-(itsmonfst) and mid-(itsmonfst − 1).

Format of climatological data (file_tscl) for option TSCLDIRECT

```
      integer(4), parameter :: imn = 12
      character(128) :: file_tscl_month
      real(4) :: ttlev(imt,jmt,kk,imn), tslev(imt,jmt,kk,imn)

      ! separate file for each month
      do m = 1, imn
        write(file_tscl_month,'(1a,i2.2)') trim(file_tscl), m
        open (unit=itscl,file=file_tscl_month,access=direct,recl=4*imt*jmt*kk)
        write(unit=itscl,rec=1) (((ttlev(i,j,k),i=1,imt),j=1,jmt),k=1,kk)
        write(unit=itscl,rec=2) (((tslev(i,j,k),i=1,imt),j=1,jmt),k=1,kk)
        close(itscl)
      end do
```

16.2.3 Nudging (body forcing) data

The time integration could be conducted restoring temperature and salinity of the specified grid points to the climatological data read from the file `file_tscl` (often referred to as nudging or body forcing). This is done by setting the namelist parameter `iforcev` (namelist `njobp`) to one. In this case, the data that contain information about the grid-wise "on (=1)" or "off (=0)" of nudging should be prepared in a default case. In a case of `TSINTPOL`, data that contain the grid number of points where nudging is done should be prepared. It is read from the file `file_frc` (namelist `njobbdy`) in `rstcoef.F90`. The restoring time for nudging should be specified as `rtmscb` (namelist `njobbdy`) in units of days.

Format for nudging grid points (file_frc) for TSINTPOL case

```
#ifdef OGCM_TSINTPOL  ! when using interpolated climatological data,
                      ! for example a large size model

      ! numbf : the number of grid points for nudging
      !         numbf is a namelist parameter in the model (rstcoef.F90)
      integer(4), save :: numbf = 1000
      namelist /nblarge/ numbf
      read(unit=5,nml=nblarge,iostat=ios)

      ! iposbf : the grid number of grids where nudging is done
      integer(4):: iposbf(numbf)

      write(ifrcdt) (iposbf(n),n=1,numbf)   ! serial form

      ! in the subroutine rstcoef, iposbf(numbf) is treated as follows and
      ! the nudging grid points at (i,j) are defined.

      do n = 1, numbf
        j = iposbf(n)/imut + 1
        i = iposbf(n) - (j-1)*imut
        chfb(i,j,1:km) = chfbc ! coefficient for nudging
                               ! (chfbc = 1.0d0 / rtmscb / 86400)
      enddo
#endif /* OGCM_TSINTPOL */
```

Format for nudging grid points (file_frc) for default case

```
#ifndef OGCM_TSINTPOL      ! for a small size model
#ifdef OGCM_BF2D           ! for two dimensional distribution

      ! chf2d : 2-D on(=1) off(=0) information for nudging
      ! imut, jmut are the numbers of grid points in the model domain

      real(4) :: chf2d(imut,jmut)   ! single precision

      write(ifrcdt) chf2d   ! serial form
                            ! distributed to each node
                            ! and stored in chfb(imx,jmx,km)
                            ! multiplied by chfbc
                            ! (chfbc = 1.0d0 / rtmscb / 86400)

#else /* OGCM_BF2D */      ! for three dimensional distribution

      ! chf3d : 3-D on(=1) off(=0) information for nudging
      ! imut, jmut, km are the numbers of grids in the model domain

      real(8)   :: chf3d(imut,jmut,km)

      write(ifrcdt) chf3d   ! serial form
                            ! read using the subroutine restart_read
                            ! distributed to each node
                            ! and stored in chfb(imx,jmx,km)
                            ! multiplied by chfbc
                            ! (chfbc = 1.0d0 / rtmscb / 86400)
#endif /* OGCM_BF2D */
#endif /* OGCM_TSINTPOL */
```

16.2.4 Atmospheric forcing data

By default, the surface forcing data are read at a uniform time interval (`isrstb`; in seconds), which should be specified in namelist `ifrcd` (`force.F90`) along with a parameter `ifna` that specifies the record number of surface forcing data corresponding to the nearest future from the start time of "this" run. A leap year is set according to the calendar subroutine. When climatological data is repeatedly used, specify the option `CLMFRC`. When monthly data are used, specify the option `MONFRC`.

The following data files should be prepared according to the chosen model options. Each file is opened only once at the beginning of the run-time and thus should contain all data needed for that run.

a. Wind stress data (mandatory): file_wind (namelist inflw)

zonal component: `wsx4(imf,jmf)` [dyn$\cdot$cm^{-2}]

meridional component: `wsy4(imf,jmf)` [dyn$\cdot$cm^{-2}]

```
real(4) :: wsx4(imf,jmf),wsy4(imf,jmf)

do irec = 1, nrec ! ''nrec'' is  the number of data
  ! data formation
  write(unit=iwind,rec=irec) wsx4, wsy4 ! direct form
enddo
```

b. Radiation data (HFLUX): file_hflx (namelist inflh)

short wave: `qsh4(imf,jmf)` [$\mathrm{erg \cdot s^{-1} \cdot cm^{-2}} = 10^{-3}\mathrm{W \cdot m^{-2}}$]

long wave: `qlo4(imf,jmf)` [$\mathrm{erg \cdot s^{-1} \cdot cm^{-2}} = 10^{-3}\mathrm{W \cdot m^{-2}}$]

sea surface temperature: `sst4(imf,jmf)` [°C]; in the case of LWDOWN (only downward radiation) dummy data are used

```
real(4) :: qsh4(imf,jmf), qlo4(imf,jmf)
real(4) :: sst4(imf,jmf)

do irec = 1, nrec  ! ''nrec'' is  the number of data
  ! data formation
  write(unit=ihflx,rec=irec) qsh4, qlo4, sst4 ! direct form
enddo
```

c. Data needed to calculate the latent and sensible heat fluxes using bulk formula (HFLUX): file_bulk (namelist inflh)

air temperature: `sat4(imf,jmf)` [°C]

specific humidity: `qar4(imf,jmf)`

wind speed: `wdv4(imf,jmf)` [$\mathrm{cm \cdot s^{-1}}$]

sea surface pressure: `slp4(imf,jmf)` [hPa]

```
real(4) :: sat4(imf,jmf), qar4(imf,jmf)
real(4) :: wdv4(imf,jmf), slp4(imf,jmf)

do irec = 1, nrec ! ''nrec'' is  the number of data
  ! data formation
  write(unit=ibulk,rec=irec) sat4,qar4,wdv4,slp4
enddo
```

d. Precipitation data (WFLUX): file_prcp (namelist inflp)

precipitation: `pcp4(imf,jmf)` [$\mathrm{cm \cdot s^{-1}}$]

Note: fresh water flux data should be scaled by pure water density (1.036 $\mathrm{g\,cm^{-3}}$)

```
real(4) :: pcp4(imf,jmf)

do irec = 1, nrec ! ``nrec'' is  the number of data
  ! data formation
  write(unit=ipcpr,rec=irec) pcp4
enddo
```

e. River discharge data (RUNOFF): file_rnof (namelist inflo)

river discharge: `rof4(imrof,jmrof)` [$\mathrm{cm \cdot s^{-1}}$]

Note: fresh water flux data should be scaled by pure water density ($1.036\ \mathrm{g\,cm^{-3}}$)

```
real(4) :: rof4(imrof,jmrof) ! imrof, jmrof could be different than imf, jmf

do irec = 1, nrec ! ``nrec'' is  the number of data
  ! data formation
  write(unit=irnof,rec=irec) rof4
enddo
```

f. Sea-ice fractional area data (ICECLIM): file_icec (namelist inflic)

sea-ice fractional area: `aic4(imf,jmf)`

```
real(4) :: aic4(imf,jmf)

do irec = 1, nrec ! ``nrec'' is  the number of data
  ! data formation
  write(unit=iicec,rec=irec) aic4
enddo
```

16.3 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 job have to be given to the standard input or redirection from a file.

This section presents an example shell script for the global tri-polar grid model. The following three files are used in this example.

- `run.sh` :: A shell script that submits the job to a batch request controlling system NQS.
- `runogcm.sh` :: The shell script that includes the command `ogcm`, the binary that executes the model.
- `NAMELIST` :: The namelist file that is sent to `ogcm` via the standard input.

For example, use the following command to submit a job to the batch request controlling system.

```
% qsub run.sh
```

Example of shell script that should be submitted (`run.sh`)

```
#!/bin/sh
#PBS -l cpunum_job=8
#PBS -l memsz_job=10gb
#PBS -l cputim_job=120000
#PBS -A K0001
#PBS -N coreI
#
RUNDIR=${HOME}/MRICOM/coreI/run # The directory of this shell script

cd ${RUNDIR}

echo ----------------------------------------------------------------------
echo START
date
echo ----------------------------------------------------------------------

pwd

/usr/bin/mpirun -np 8 ./runogcm.sh # this example is for the 8MPI job

echo ----------------------------------------------------------------------
echo END
date
echo ----------------------------------------------------------------------
```

An example shell script runogcm.sh that is called from run.sh

```
#!/bin/sh
#
# RUNDIR ... the directory of this shell script
#            set to reconfirm the path of the environmental variables
#            for the sub processes

RUNDIR=$HOME/MRICOM/coreI/run

cd $RUNDIR

if [ ! -s logs ]; then
  mkdir ./logs   # output directory for execution logs (standard output)
fi

if [ ! -s logs/ftrace ]; then
  mkdir ./logs/ftrace # output directory for job diagnoses by the system
fi

F_FTRACE=FMT1; export F_FTRACE
F_SETBUF06=0; export F_SETBUF06
F_RECLUNIT=BYTE; export F_RECLUNIT
FTRACEDIR=logs/ftrace; export FTRACEDIR

./ogcm  > logs/out.txt 2> logs/stderr.txt < NAMELIST
```

```
=======================        Example for NAMELIST        ==========================
                         (see README.Namelist for details)

# basic parameters for model time integration
&njobp
  nfirst=-1,
  nstep=48
  neng=-1,
  nneng=-1,
  nwrit=48
  nwrt2=-1,
  mampai=1,
  mmpai2=1,
  nkeisu=48
  adtuv=30,
  adtts=30,
  adtsf=30,
  hdts=1.0d7,
  hduv=0.0d0,
  hduv2=0.0d0,
  vduv=0.1d0,
  alpha=1.0d0,
  gamma=51*1.0d0,
  hupp=0.5d0,
  vupp=0.7d0,
  ifrcsf=1,
  iforcv=0,
  ispin=0,
  matsuno_int=12,
  nocbgt=0,
  flg_timemonitor=.false.
/
```

```
# (ibyr,ibmn,ibdy) model integration starting date (year, month, day)
&njobpt
  ibyr=1001,
  ibmn=1,
  ibdy=1,
/

# file name for variable grid spacing (option VARIABLE)
&inflg
 file_vgrid='../data/vgrid.d'
/

# Smagorinsky parameterization for horizontal viscosity (option SMAGOR)
# (cscl) scaling factor of Smagorinsky viscosity coefficient

&njobsmg
 cscl = 3.5d0
/

# parameters for isopycnal diffusion, diapycnal diffusion, layer thickness diffusion
&njobpi
  ai=1.0d7
  ad=0.1d0
  aitd=5.0d6
/

# the number of restart and history files which is output during the job
&numfl
  num_restart=1,
  num_hist=12
/

# the names of the mandatory input data files
# snapshot data to start integration
&infla
  file_restart_in='result/restart.1000'
/

# topography
&infltopo
  file_topo='../data/topo.d'
/

#scale factor
&inflscl
  file_scale='../data/scale_factor.d'
/

# temperature and salinity climatology
&infltscl
  file_tscl='../data/tsclim.d'
/

# nudging
# if iforcv (/njobp/) = 0, dummy file name for file_frc should be specified
# (rtmsc)  restoring time for surface forcing [day]
# (rtmscb) restoring time for body forcing [day]
# (kmb)    the vertical level below which body forcing (nudging) is applied for all area
# (kmb_cnst) the vertical level below which body forcing (nudging) is applied
```

```
#              for the area specified in (file_frc)
&njobbdy
  file_frc='../data/bforce.d',
  rtmsc=8.0d0,
  rtmscb=30.0d0,
  kmb=52,
  kmb_cnst=2,
/

# the name of the final state (restart) file
&outfr
  file_restart_out_temp='result/restart.1001',
/

# The time step (in minutes) for the free surface equation and the way of time-filtering
&njobpf
  adttr=1,
  ntflt=-1,
/

# surface forcing data

# (isrstb) the time interval of the surface forcing data
# (ifna) the first record number for the sea surface data used for ``this'' run
#        the start time is between (ifna) and (ifna - 1)
&ifrcd
  isrstb=21600
  ifna=1,
/

# wind stress
&inflw
  file_wind='../data/file_wind_core.grd',
/

# the names of the input data files for the sea surface flux
&inflh
  file_hflx='../data/file_hflux_core.grd',
  file_bulk='../data/file_bulk_core.grd',
/

# the name of the input data file for precipitation data
&inflp
  file_prcp='../data/file_prcp_core.grd',
/

# the name of the input data file for river discharge data
&inflo
  file_rnof='../data/file_rnof_core.grd',
/

# These parameters set up the diffusion around the river mouth
&nrivermouth
  CFLlim=0.2d0,
  nspreadnum=0,
  flg_enhance_vm_rivmouth=.false.,
  avdrmax=1.0d4,
  dep_rivmix=30.0d2,
  afc1=10.0d0,
```

```
  afc2=7.0d0,
  sal_lim=5.d0,
  hdsal=1.d8,
/

# the name of the output data file for mean states
&outfh
  file_hist_temp='result/hist.1001',
/

# the name of the output data file for flux mean states
&outff
  file_hflux_temp='result/hflux.1001',
/

# These parameters set the sea surface albedo
&njobalb
  alb=0.066d0,
  albedo_choice=1
/

# basic properties for sea ice
&njobpsi
  irstrt=1,
  akh=1.0d2,
  int_bgtice=0,
  nstepi=1488,
  ibyri=1001,
  ibmni=1,
  ibdyi=1,
/

# the time step interval for ice dynamics
&njobidyn
  adtdi=1.0d0
/

# These parameters set the sea-ice albedo
&njobalbsi
  alb_ice_visible_t0=0.8d0,
  alb_ice_nearIR_t0=0.52d0,
  alb_snw_visible_t0=0.98d0,
  alb_snw_nearIR_t0=0.70d0,
  alb_ice_visible_dec_ratio=0.075d0,
  alb_ice_nearIR_dec_ratio=0.075d0,
  alb_snw_visible_dec_ratio=0.10d0,
  alb_snw_nearIR_dec_ratio=0.15d0,
  hi_ref=0.5d0,
  atan_ref=3.0d0,
  tsfci_t0=-1.0d0,
  tsfci_t1=0.0d0,
  fsnow_patch=0.02d0,
/

# The name of the sea ice restart file (input)
&infli
  file_ice_restart_in='result/ice_restart.1000',
/
```

```
# The name of the sea ice restart file (output)
&outfir
  file_ice_restart_out_temp='result/ice_restart.1001',
/

# The name of the sea ice history file
&outfih
  file_ice_hist_temp='result/ice_hist.1001',
/

# These parameters set the sea-ice thickness category
&njobpscat
  hbound=0.0d0,0.6d0,1.4d0,2.4d0,3.6d0,30.0d0,
  lsicat_volchk=.false.
/

# the number of intervals by which mean states of thickness-categorized ice are calculated
&nhsticint
  num_hint_ic=1
/

# These parameters set output of the mean states for thickness-categorized sea-ice
&nhsticfile
  maxnum_hist_ic=1,
  nwrt_hist_ic=-1,
  imin_hist_ic=1,
  imax_hist_ic=364,
  jmin_hist_ic=1,
  jmax_hist_ic=368,
  file_ice_hist_ic_temp='result/sicathsta.1001'
/

# The name of the restart file (input) of the thickness-categorized sea ice
&inflic
  file_icecat_restart_in='result/sicatrsta.1000'
/

# These parameters set the output of final state for the thickness-categorized sea ice
&outflic
  num_rst_ic=1,
  maxnum_rst_ic=1,
  nwrt_rst_ic=-1,
  file_icecat_restart_out_temp='result/sicatrsta.1001'
/
================================================================================
```

Table 16.4. Main output data files and their related program files (see Section 16.4)

subject	file name specified in namelist as	output from
restart data	file_restart_out_temp	writdt.F90
(for ice averaged over all categories)	file_ice_restart_out_temp	writdt.F90
(for ice for each thickness category)	file_icecat_restart_out_temp	ice_restart.F90
mean values	file_hist_temp	writdt.F90
(for sea surface fluxes)	file_hflux_temp	writdt.F90
(for ice averaged over all categories)	file_ice_hist_temp	writdt.F90
(for ice for each thickness category)	file_ice_hist_ic_temp	ice_hist.F90

16.4 Structure of output files

This section summarizes the format of the final state (restart) data to resume the model integration and that of the mean state (history) data used for monitoring and analyses (Table 16.4).

In addition to these, we are developing a simple module that outputs a mean state or a snap shot by only specifying a namelist at run time. For example, to have monthly mean temperature, add a following namelist entry,

An example namelist for monthly mean temperature output

```
&nmlhs_t
  fname='result/hs_tt'
  wrtint=-1
  [undef_mask=-9.99e33]
  [ymdhm=2]
/
```

where `fname` specifies the basename of file, `wrtint` specifies the output interval in terms of integration time step (-1 for monthly output), and `ymdhm` specifies the depth of calendar date used in the file name. In the above example, the temperature averaged for a month of `mm` of a year `yyyy` is output to file `result/hs_tt.yyyymm`. See `README.Namelist` for items available.

16.4.1 Snapshot (restart)

Restart for the ocean model

```
integer(4) :: last, month, iday, ihour, imin
real(8) :: ahour(km), aday(km), ayear(km), pd(km), pm(km+1), dmn(km)
real(8) :: ddmna, over
real(8) :: u(imut,jmut,km), v(imut,jmut,km)
real(8) :: t(imut,jmut,km), s(imut,jmut,km)
real(8) :: avd(imut,jmut,km), avm(imut,jmut,km)
real(8) :: avq(imut,jmut,km), eb(imut,jmut,km)
real(8) :: dbuf(imut,jmut,3)
integer(4) :: nu ! device number

! The informations below are written to a single file in the following order
! by serial form for the default case.

! last   : the total number of the integrated time steps
!        : at the time of writing the restart data
! month, iday, ihour, imin : month, day, hour and minute for restart data
! over   : the parameter of relaxation for barotropic stream function
!        : (dummy data for FREESURFACE)
! ahour, aday, ayear : total integrated hour, day and year from the start
!                    : of the time integration (ahour is master)
! pd, pm : averaged pressure at each vertical level
!        : used for the calculation of the equation of state
!        : pd: T-points
!        : pm: middle point between adjacent T-points
! ddmna  : averaged density of the whole ocean
! dmn    : averaged density at each vertical level

write(nu) last, month, iday, ihour, imin, over, &
   & ahour, aday, ayear, pd, pm, ddmna, dmn

! In the case of SPLITREST the followings are written in the separate files
!  after the above informations about time.

write(nu) u        ! zonal velocity
write(nu) v        ! meridional velocity
write(nu) t        ! temperature
write(nu) s        ! salinity
write(nu) dbuf     ! 1: sea surface height
                   ! 2: vertically integrated zonal transport
                   ! 3: vertically integrated meridional transport
write(nu) avd      ! the vertical diffusion coeff. in the case of VVDIMP
write(nu) avm      ! the vertical viscosity coeff. in the case of VVDIMP
write(nu) avq      ! the vertical diffusion coeff. of turbulent kinetic energy
                   ! in the case of NOHKIM
write(nu) eb       ! in the case of NOHKIM turbulent kinetic energy
```

For sea ice, the grid-averaged state and the thickness-categorized states are written to separate files.

Restart for grid-averaged sea-ice

```
integer(4) :: last, month, iday, ihour, imin ! from the ocean main part
real(8) :: a0iceo(imut,jmut) ! sea-ice concentration
real(8) :: hiceo(imut,jmut)  ! sea-ice thickness
real(8) :: hi   (imut,jmut)  ! averaged thickness = a0iceo * hiceo
real(8) :: hsnwo(imut,jmut)  ! snow depth
real(8) :: hsnw (imut,jmut)  ! averaged snow depth = a0iceo * hsnwo
real(8) :: t0iceo(imut,jmut) ! skin temperature beneath the sea ice
real(8) :: t0icel(imut,jmut) ! skin temperature in the open leads
real(8) :: tsfci (imut,jmut) ! sea-ice surface temperature
real(8) :: uice(imut,jmut), vice(imut,jmut) ! drift vector
! stress tensor
real(8) :: sigma1(imut,jmut), sigma2(imut,jmut), sigma3(imut,jmut)

real(8) :: dbuf(imut,jmut,10)

integer(4) :: nui ! device number

write (nui) last, month, iday, ihour, imin
dbuf(1:imut,1:jmut,1) = a0iceo(1:imut,1:jmut)
dbuf(1:imut,1:jmut,2) = hi(1:imut,1:jmut)
dbuf(1:imut,1:jmut,3) = hiceo(1:imut,1:jmut)
dbuf(1:imut,1:jmut,4) = hsnw(1:imut,1:jmut)
dbuf(1:imut,1:jmut,5) = hsnwo(1:imut,1:jmut)
dbuf(1:imut,1:jmut,6) = t0iceo(1:imut,1:jmut)
dbuf(1:imut,1:jmut,7) = t0icel(1:imut,1:jmut)
dbuf(1:imut,1:jmut,8) = tsfci(1:imut,1:jmut)
dbuf(1:imut,1:jmut,9) = uice(1:imut,1:jmut)
dbuf(1:imut,1:jmut,10) = vice(1:imut,1:jmut)
write(nui) dbuf(1:imut,1:jmut,1:10)
dbuf(1:imut,1:jmut,1) = sigma1(1:imut,1:jmut)
dbuf(1:imut,1:jmut,2) = sigma2(1:imut,1:jmut)
dbuf(1:imut,1:jmut,3) = sigma3(1:imut,1:jmut)
write(nui) dbuf(1:imut,1:jmut,1:3)
```

Restart data for thickness-categorized sea-ice

```
integer(4) :: nstep_job, month, iday, ihour, imin
real(8)    :: aihour ! total integrated time
integer(4), parameter :: ncat = 5 ! number of thickness categories
! ice concentration
real(8) :: aicen (1:imut,1:jmut,0:ncat), a0iceo(imut,jmut)
! ice thickness
real(8) :: hicen (1:imut,1:jmut,0:ncat), hiceo (imut,jmut)
! averaged sea-ice thickness
real(8) :: hin   (1:imut,1:jmut,0:ncat), hi    (imut,jmut)
! snow depth
real(8) :: hsnwn (1:imut,1:jmut,0:ncat), hsnwo (imut,jmut)
! averaged snow thickness
real(8) :: hsn   (1:imut,1:jmut,0:ncat), hsnw  (imut,jmut)
! ice surface temperature
real(8) :: tsfcin(1:imut,1:jmut,0:ncat), tsfci (imut,jmut)
! ice temperature
real(8) :: t1icen(1:imut,1:jmut,0:ncat)
! sea surface skin temperature
real(8) :: t0icen(1:imut,1:jmut,0:ncat)
! sea surface skin salinity
real(8) :: s0n   (1:imut,1:jmut,0:ncat)
! skin temperature beneath the sea ice
real(8) :: t0iceo(imut,jmut)
! skin temperature in the open leads
real(8) :: t0icel(imut,jmut)
! stress tensor
real(8) :: sigma1(imut,jmut), sigma2(imut,jmut), sigma3(imut,jmut)

integer(4) :: nu_icecat_rst ! device number

write (nu_icecat_rst) nstep_job, month, iday, ihour, imin, aihour

do m = 0, ncat
  write(nu_icecat_rst) aicen(1:imut,1:jmut,m)
  write(nu_icecat_rst) hin  (1:imut,1:jmut,m)
  write(nu_icecat_rst) hsn  (1:imut,1:jmut,m)
  write(nu_icecat_rst) hicen(1:imut,1:jmut,m)
  write(nu_icecat_rst) hsnwn(1:imut,1:jmut,m)
  write(nu_icecat_rst) tsfcin(1:imut,1:jmut,m)
  write(nu_icecat_rst) t1icen(1:imut,1:jmut,m)
  write(nu_icecat_rst) t0icen(1:imut,1:jmut,m)
  write(nu_icecat_rst) s0n(1:imut,1:jmut,m)
end do

 write(nu_icecat_rst) a0iceo(1:imut,1:jmut)
 write(nu_icecat_rst) hi(1:imut,1:jmut)
 write(nu_icecat_rst) hiceo(1:imut,1:jmut)
 write(nu_icecat_rst) hsnw(1:imut,1:jmut)
 write(nu_icecat_rst) hsnwo(1:imut,1:jmut)
 write(nu_icecat_rst) t0iceo(1:imut,1:jmut)
 write(nu_icecat_rst) t0icel(1:imut,1:jmut)
 write(nu_icecat_rst) tsfci(1:imut,1:jmut)
 write(nu_icecat_rst) uice(1:imut,1:jmut)
 write(nu_icecat_rst) vice(1:imut,1:jmut)
 write(nu_icecat_rst) sigma1(1:imut,1:jmut)
 write(nu_icecat_rst) sigma2(1:imut,1:jmut)
 write(nu_icecat_rst) sigma3(1:imut,1:jmut)
```

16.4.2 Averaged value (history)

Ocean model history data (HIST)

```
integer(4) :: nkai, month, iday, ihour, imin, mdays

real(4) :: um(imut,jmut,km), vm(imut,jmut,km)
real(4) :: tm(imut,jmut,km), sm(imut,jmut,km)
real(4) :: hm(imut,jmut)

integer(4) :: nu_hst ! device number

! The informations below are written to a single file in the following order
! by serial form for the default case.

! nkai   : the total number of the integrated time steps
!        : at the time of writing the data
! month, iday, ihour, imin : month, day, hour and minute
!                          : at the time of writing the data
! mdays  : the data are averaged over mdays

write(nu_hst) nkai, month, iday, ihour, imin, mdays

! In the case of SPLITHIST the data below are written in a different file
! after the above information about time

write(nu_hst) um     ! zonal velocity
write(nu_hst) vm     ! meridional velocity
write(nu_hst) tm     ! temperature
write(nu_hst) sm     ! salinity
write(nu_hst) hm     ! sea surface height
```

sea surface flux averaged value history file (HISTFLUX)

```
integer(4) :: nkai, month, iday, ihour, imin, mdays

real(4) :: sqlw  (imut,jmut)  ! longwave radiation (downward minus upward)
real(4) :: sqsn  (imut,jmut)  ! sensible heat flux
real(4) :: sqla  (imut,jmut)  ! latent heat flux
real(4) :: shflux(imut,jmut)  ! total heat flux
real(4) :: swflux(imut,jmut)  ! total fresh water flux
real(4) :: sstrx(imut,jmut), sstry(imut,jmut) ! sea surface stress

integer(4) :: nuf_hst ! device number

write(nuf_hst) nkai, month, iday, ihour, imin, mdays
write(nuf_hst) sqlw  (1:imut,1:jmut)
write(nuf_hst) sqsn  (1:imut,1:jmut)
write(nuf_hst) sqla  (1:imut,1:jmut)
write(nuf_hst) shflux(1:imut,1:jmut)
write(nuf_hst) swflux(1:imut,1:jmut)
write(nuf_hst) sstrx (1:imut,1:jmut)
write(nuf_hst) sstry (1:imut,1:jmut)
```

For sea ice, the grid-averaged state and the thickness-categorized states are written to separate files.

Grid-averaged sea-ice history file (`ICE`)

```
integer(4) :: nkai, month, iday, ihour, imin, mdays

real(4) :: sar(imut,jmut)  ! sea-ice concentration
real(4) :: shi(imut,jmut)  ! averaged sea-ice thickness
real(4) :: ssn(imut,jmut)  ! averaged snow depth
real(4) :: sti(imut,jmut)  ! sea-ice surface temperature
real(4) :: uice(imut,jmut), vice(imut,jmut) ! drift vector

real(4) :: sbuf(imut,jmut,6)

integer(4) :: nui_hst ! device number

write (nui_hst) nkai, month, iday, ihour, imin, mdays
sbuf(1:imut,1:jmut,1) = shi(1:imut,1:jmut)
sbuf(1:imut,1:jmut,2) = ssn(1:imut,1:jmut)
sbuf(1:imut,1:jmut,3) = sar(1:imut,1:jmut)
sbuf(1:imut,1:jmut,4) = sti(1:imut,1:jmut)
sbuf(1:imut,1:jmut,5) = sui(1:imut,1:jmut)
sbuf(1:imut,1:jmut,6) = svi(1:imut,1:jmut)
write(nui_hst) sbuf(1:imut,1:jmut,1:6)
```

Thickness-categorized sea-ice history file (ICECAT)

```
integer(4) :: nstep_job, month, iday, ihour, imin, mdays
integer(4), parameter :: ncat = 5 ! number of thickness category

real(4) :: shi0(1:imut,1:jmut) ! sum of averaged sea-ice thickness
                               ! for all categories
real(4) :: shs0(1:imut,1:jmut) ! sum of snow depth for all category
real(4) :: sai0(1:imut,1:jmut) ! sum of concentration for all category
real(4) :: sui0(1:imut,1:jmut), svi0(1:imut,1:jmut) ! sea-ice drift velocity

real(4) :: sain(1:imut,1:jmut,0:ncat) ! sea-ice concentration
real(4) :: st0n(1:imut,1:jmut,0:ncat) ! sea surface skin temperature
real(4) :: ss0n(1:imut,1:jmut,0:ncat) ! sea surface skin salinity
real(4) :: shin(1:imut,1:jmut,1:ncat) ! averaged sea-ice thickness
real(4) :: shsn(1:imut,1:jmut,1:ncat) ! averaged snow depth
real(4) :: stsn(1:imut,1:jmut,1:ncat) ! sea-ice surface temperature
real(4) :: st1n(1:imut,1:jmut,1:ncat) ! sea-ice temperature

integer(4) :: nu_icecat_hst ! device number

write(nu_icecat_hst) nstep_job, month, iday, ihour, imin, mdays

write(nu_icecat_hst) = shi0(1:imut,1:jmut)
write(nu_icecat_hst) = shs0(1:imut,1:jmut)
write(nu_icecat_hst) = sai0(1:imut,1:jmut)
write(nu_icecat_hst) = sui0(1:imut,1:jmut)
write(nu_icecat_hst) = svi0(1:imut,1:jmut)

write(nu_icecat_hst) = sain(1:imut,1:jmut,0)
write(nu_icecat_hst) = st0n(1:imut,1:jmut,0)
write(nu_icecat_hst) = ss0n(1:imut,1:jmut,0)

do m = 1, ncat
  write(nu_icecat_hst) = shin(1:imut,1:jmut,m)
  write(nu_icecat_hst) = shsn(1:imut,1:jmut,m)
  write(nu_icecat_hst) = sain(1:imut,1:jmut,m)
  write(nu_icecat_hst) = stsn(1:imut,1:jmut,m)
  write(nu_icecat_hst) = st1n(1:imut,1:jmut,m)
  write(nu_icecat_hst) = st0n(1:imut,1:jmut,m)
  write(nu_icecat_hst) = ss0n(1:imut,1:jmut,m)
end do
```

16.5 Appendix

16.5.1 Model options

The model options are as follows. 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`. In the source program an expression like `OGCM_PARALLEL` is used but here `OGCM_` is omitted for option expressions.

Table 16.5. Description of Model Options

Model option	Description
`BBL`	uses the bottom boundary layer model
`BF2D`	reads 2-D distribution of "on"/"off" for nudging
`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.
`BULKKARA`	Kara (2000) is used for the surface flux bulk formula.
`BULKKONDO2`	Kondo(1975) is used for the surface flux bulk formula.
`BULKNCAR`	Large and Yeager (2004) is used for the surface flux bulk formula. This option corresponds to the COREs. (*) `BULKKARA`, `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 10m. (*) 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`
`CHFDIST`	employs horizontal distribution of surface restoring time for SST and SSS
`CHLMA94`	shortwave penetration scheme with chlorophyll concentration by Morel and Antoine (1994) (*) use with `NPZD` and `SOLARANGLE`
`CLMFRC`	uses climatological wind stress and surface heat flux (no leap year, repeating one year cycle)
`CYCLIC`	uses zonally cyclic condition
`DIAGTRANSP`	outputs time averaged vertically integrated transport `UM`, `VM` (*) available only for `FREESURFACE`
`DIFAJS`	sets large vertical diffusion coefficient (1.0 $m^2 \cdot s^{-1}$) between unstable points instead of convective adjustment
`FLXINTPOL`	interpolates the surface forcing data to model grid points

Model option	description
FREESURFACE	free surface (if undefined, rigid-lid)
FSEB	free surface equations are advanced using the Eular-backward time-integration
FSMOM	uses MOM free surface scheme
FSVISC	calculates viscosity explicitly in the barotropic momentum equation
	(*) FSEB, FSMOM, FSVISC are available only in the case of FREESURFACE
GMANISOTROP	Anisotropic horizontal variation of thickness diffusion is used
	(*) use with ISOPYCNAL
GMVAR	Horizontal thickness diffusion is allowed to vary in horizontal
HFLUX	calculates sea surface heat flux using bulk formula
HIST	outputs averaged state of temperature, salinity and velocity
HISTFLUX	outputs averaged state of surface fluxes
HISTVAR	outputs variance of temperature, salinity and velocity
	(*) available only for HIST
ICE	sea ice is included
ICECAT	sea ice is categorized according to its thickness
ICECLIM	reading climatological sea-ice fractional area from file
INILEV	sets Levitus climatological three-dimensional data for the initial value
ISOPYCNAL	uses isopycnal diffusion and Gent-McWilliams' parameterization for eddy
	induced tracer transport velocity (thickness diffusion)
KPP	uses KPP for mixed layer model
LWDOWN	Long wave radiation data include only downward component instead of default net radiation
MELYAM	uses Mellor and Yamada Level 2.5 for mixed layer model
MON30D	sets one month as 30 days and one year as 360 days
MONFRC	uses monthly forcing for surface data
NESTONLINE	low resolution (PARENT) and high resolution (SUB) models exchange data on-line
	(*) use with SCUP
NOHKIM	uses Noh's mixed layer model
NOMATSUNO	uses forward finite difference instead of Matsuno scheme
NPZD	NPZD process is included
	(*) use with CARBON
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
PRAJS	adjusts the freshwater flux to suppress the increase/decrease of total ocean volume
	(*) available only for WFLUX
RUNOFF	uses river runoff data
	(*) available only for WFLUX
SALCNSVRS	Salinity restoring flux is corrected so that globally integrated salinity flux become zero
	at each time step

Model option	description
	(*) use with ”SFLUXR”
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 (EVP)
	(*) available only for ICE
SLIMIT	Tapers thickness diffusion near the sea surface
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 short wave penetration
SOMADVEC	uses second order moment advection by Prather (1986)
SPHERICAL	calculates scale factor semi-analytically for the spherical coordinates
SPLITHIST	output averaged temperature, salinity, velocity and sea surface height to separate files
	(*) available only for HIST
SPLITREST	input(output) initial(final) state data from(to) separate files for each properties
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
TRCBIHARM	uses biharmonic operator for horizontal diffusion
	(*) Should not be used with ISOPYCNAL
TSCLDIRECT	Temperature and salinity climatology is read from direct access files
	naming convention is (file_tscl)mm (mm = month)
	see README.Namelist
TSINTPOL	interpolates climatological temperature and salinity data to model grids
ULTIMATE	applies ultimate limiter for the calculation of the horizontal advection of temperature and salinity
	(*) available only for for UTOPIA
UTOPIA	uses UTOPIA scheme for horizontal advection of temperature and salinity
VARHID	sets the horizontal (isopycnal) diffusion coefficient and viscosity as functions of grid size
VARIABLE	variable horizontal grid spacing
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 option ISOPYCNAL is selected

Model option	description
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
Y365	no leap year
ZQUICKEST	uses QUICKEST for the vertical advection of temperature and salinity
ZULTIMATE	applies ultimate limiter for the vertical advection of temperature and salinity
	(*) available only for ZQUICKEST
CGCM	used as an ocean module for a coupled model
MOVE	used as ocean module for data assimilation (MOVE) system
SCUP	use simple coupler (SCUP) library
SCUPCGCM	used as an ocean module for a coupled model using scup for communication

References

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.

Acknowledgments

The development of MRI.COM was first proposed by Yoshiteru Kitamura. During the course of its development, significant contributions were made by Seiji Yukimoto and Atsushi Obata.

The authors owe special thanks to Hiroyasu Hasumi of the University of Tokyo for his valuable suggestions. Application of the EVP dynamics to the sea ice model, of σ-coordinates to the free surface formulation, and of high-accuracy advection schemes to the tracer equations would not have been successful without his support.

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, and Noriya Yoshioka for their numerous contributions to the original models that MRI.COM is based on.

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 of MRI contributed to the development and refinement of this model.

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 a new version 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) of Climate Variability and Predictability (CLIVAR) are gratefully acknowledged.

気象研究所技術報告一覧表

第１号　バックグラウンド大気汚染の測定法の開発（地球規模大気汚染特別研究班，1978）
Development of Monitoring Techniques for Global Background Air Pollution. (MRI Special Research Group on Global Atmospheric Pollution, 1978)

第２号　主要活火山の地殻変動並びに地熱状態の調査研究（地震火山研究部，1979）
Investigation of Ground Movement and Geothermal State of Main Active Volcanoes in Japan. (Seismology and Volcanology Research Division, 1979)

第３号　筑波研究学園都市に新設された気象観測用鉄塔施設（花房龍男・藤谷徳之助・伴野　登・魚津　博，1979）
On the Meteorological Tower and Its Observational System at Tsukuba Science City. (T. Hanafusa, T. Fujitani, N. Banno, and H. Uozu, 1979)

第４号　海底地震常時観測システムの開発（地震火山研究部，1980）
Permanent Ocean－Bottom Seismograph Observation System. (Seismology and Volcanology Research Division, 1980)

第５号　本州南方海域水温図－400m（又は 500m）深と 1,000m 深－（1934－1943 年及び 1954－1980 年）（海洋研究部，1981）
Horizontal Distribution of Temperature in 400m (or 500m) and 1,000m Depth in Sea South of Honshu, Japan and Western －North Pacific Ocean from 1934 to 1943 and from 1954 to 1980. (Oceanographical Research Division, 1981)

第６号　成層圏オゾンの破壊につながる大気成分及び紫外日射の観測（高層物理研究部，1982）
Observations of the Atmospheric Constituents Related to the Stratospheric ozon Depletion and the Ultraviolet Radiation. (Upper Atmosphere Physics Research Division, 1982)

第７号　83 型強震計の開発（地震火山研究部，1983）
Strong－Motion Seismograph Model 83 for the Japan Meteorological Agency Network. (Seismology and Volcanology Research Division, 1983)

第８号　大気中における雪片の融解現象に関する研究（物理気象研究部，1984）
The Study of Melting of Snowflakes in the Atmosphere. (Physical Meteorology Research Division, 1984)

第９号　御前崎南方沖における海底水圧観測（地震火山研究部・海洋研究部，1984）
Bottom Pressure Observation South off Omaezaki, Central Honsyu. (Seismology and Volcanology Research Division and Oceanographical Research Division, 1984)

第 10 号　日本付近の低気圧の統計（予報研究部，1984）
Statistics on Cyclones around Japan. (Forecast Research Division, 1984)

第 11 号　局地風と大気汚染質の輸送に関する研究（応用気象研究部，1984）
Observations and Numerical Experiments on Local Circulation and Medium－Range Transport of Air Pollutions. (Applied Meteorology Research Division, 1984)

第 12 号　火山活動監視手法に関する研究（地震火山研究部，1984）
Investigation on the Techniques for Volcanic Activity Surveillance. (Seismology and Volcanology Research Division, 1984)

第 13 号　気象研究所大気大循環モデルーⅠ（MRI・GCM－Ⅰ）（予報研究部，1984）
A Description of the MRI Atmospheric General Circulation Model (The MRI・GCM－Ⅰ). (Forecast Research Division, 1984)

第 14 号　台風の構造の変化と移動に関する研究－台風 7916 の一生－（台風研究部，1985）
A Study on the Changes of the Three - Dimensional Structure and the Movement Speed of the Typhoon through its Life Time. (Typhoon Research Division, 1985)

第 15 号　波浪推算モデル MRI と MRI－Ⅱの相互比較研究－計算結果図集－（海洋気象研究部，1985）
An Intercomparison Study between the Wave Models MRI and MRI－Ⅱ －A Compilation of Results－ (Oceanographical Research Division, 1985)

第 16 号　地震予知に関する実験的及び理論的研究（地震火山研究部，1985）
Study on Earthquake Prediction by Geophysical Method. (Seismology and Volcanology Research Division, 1985)

第 17 号　北半球地上月平均気温偏差図（予報研究部，1986）
Maps of Monthly Mean Surface Temperature Anomalies over the Northern Hemisphere for 1891－1981. (Forecast Research Division, 1986)

第 18 号　中層大気の研究（高層物理研究部・気象衛星研究部・予報研究部・地磁気観測所，1986）
Studies of the Middle Atmosphere. (Upper Atmosphere Physics Research Division, Meteorological Satellite Research Division, Forecast Research Division, MRI and the Magnetic Observatory, 1986)

第 19 号　ドップラーレーダによる気象・海象の研究（気象衛星研究部・台風研究部・予報研究部・応用気象研究部・海洋研究部，1986）
Studies on Meteorological and Sea Surface Phenomena by Doppler Radar. (Meteorological Satellite Research Division, Typhoon Research Division, Forecast Research Division, Applied Meteorology Research Division, and Oceanographical Research Division, 1986)

第 20 号　気象研究所対流圏大気大循環モデル（MRI・GCM－Ⅰ）による 12 年間分の積分（予報研究部，1986）
Mean Statistics of the Tropospheric MRI・GCM－Ⅰbased on 12－year Integration. (Forecast Research Division, 1986)

第 21 号　宇宙線中間子強度 1983－1986（高層物理研究部，1987）
Multi－Directional Cosmic Ray Meson Intensity 1983－1986. (Upper Atmosphere Physics Research Division, 1987)

第 22 号　静止気象衛星「ひまわり」画像の噴火噴煙データに基づく噴火活動の解析に関する研究（地震火山研究部，1987）
Study on Analysis of Volcanic Eruptions based on Eruption Cloud Image Data obtained by the Geostationary Meteorological satellite (GMS). (Seismology and Volcanology Research Division, 1987)

第 23 号　オホーツク海海洋気候図（篠原吉雄・四竈信行，1988）
Marine Climatological Atlas of the sea of Okhotsk. (Y. Shinohara and N. Shikama, 1988)

第 24 号　海洋大循環モデルを用いた風の応力異常に対する太平洋の応答実験（海洋研究部，1989）
Response Experiment of Pacific Ocean to Anomalous Wind Stress with Ocean General Circulation Model. (Oceanographical Research Division, 1989)

第 25 号　太平洋における海洋諸要素の季節平均分布（海洋研究部，1989）
Seasonal Mean Distribution of Sea Properties in the Pacific. (Oceanographical Research Division, 1989)

第 26 号　地震前兆現象のデータベース（地震火山研究部，1990）
Database of Earthquake Precursors. (Seismology and Volcanology Research Division, 1990)

第 27 号　沖縄地方における梅雨期の降水システムの特性（台風研究部，1991）
Characteristics of Precipitation Systems During the Baiu Season in the Okinawa Area. (Typhoon Research Division, 1991)

第 28 号　気象研究所・予報研究部で開発された非静水圧モデル（猪川元興・斉藤和雄，1991）
Description of a Nonhydrostatic Model Developed at the Forecast Research Department of the MRI. (M. Ikawa and K. Saito, 1991)

第 29 号　雲の放射過程に関する総合的研究（気候研究部・物理気象研究部・応用気象研究部・気象衛星・観測システム研究部・台風研究部，1992）
A Synthetic Study on Cloud－Radiation Processes. (Climate Research Department, Physical Meteorology Research Department, Applied Meteorology Research Department, Meteorological Satellite and Observation System Research Department, and Typhoon Research Department, 1992)

第 30 号　大気と海洋・地表とのエネルギー交換過程に関する研究（三上正男・遠藤昌宏・新野　宏・山崎孝治，1992）
Studies of Energy Exchange Processes between the Ocean－Ground Surface and Atmosphere. (M. Mikami, M. Endoh, H. Niino, and K. Yamazaki, 1992)

第 31 号　降水日の出現頻度からみた日本の季節推移－30 年間の日降水量資料に基づく統計－（秋山孝子，1993）
Seasonal Transition in Japan, as Revealed by Appearance Frequency of Precipitating-Days. －Statistics of Daily Precipitation Data During 30 Years－(T. Akiyama, 1993)

第 32 号　直下型地震予知に関する観測的研究（地震火山研究部，1994）
Observational Study on the Prediction of Disastrous Intraplate Earthquakes. (Seismology and Volcanology Research Department, 1994)

第 33 号　各種気象観測機器による比較観測（気象衛星・観測システム研究部，1994）
Intercomparisons of Meteorological Observation Instruments. (Meteorological Satellite and Observation System Research Department, 1994)

第 34 号　硫黄酸化物の長距離輸送モデルと東アジア地域への適用（応用気象研究部，1995）
The Long－Range Transport Model of Sulfur Oxides and Its Application to the East Asian Region. (Applied Meteorology Research Department, 1995)

第 35 号　ウインドプロファイラーによる気象の観測法の研究（気象衛星・観測システム研究部，1995）
Studies on Wind Profiler Techniques for the Measurements of Winds. (Meteorological Satellite and Observation System Research Department, 1995)

第 36 号　降水・落下塵中の人工放射性核種の分析法及びその地球化学的研究（地球化学研究部，1996）
Geochemical Studies and Analytical Methods of Anthropogenic Radionuclides in Fallout Samples. (Geochemical Research Department, 1996)

第 37 号　大気と海洋の地球化学的研究（1995 年及び 1996 年）（地球化学研究部，1998）
Geochemical Study of the Atmosphere and Ocean in 1995 and 1996. (Geochemical Research Department, 1998)

第 38 号　鉛直２次元非線形問題（金久博忠，1999）
Vertically 2-dmensional Nonlinear Problem (H. Kanehisa, 1999)

第 39 号　客観的予報技術の研究（予報研究部，2000）
Study on the Objective Forecasting Techniques（Forecast Research Department, 2000）

第 40 号　南関東地域における応力場と地震活動予測に関する研究（地震火山研究部，2000）
Study on Stress Field and Forecast of Seismic Activity in the Kanto Region（Seismology and Volcanology Research Department, 2000）

第 41 号　電量滴定法による海水中の全炭酸濃度の高精度分析および大気中の二酸化炭素と海水中の全炭酸の放射性炭素同位体比の測定（石井雅男・吉川久幸・松枝秀和，2000）
Coulometric Precise Analysis of Total Inorganic Carbon in Seawater and Measurements of Radiocarbon for the Carbon Dioxide in the Atmosphere and for the Total Inorganic Carbon in Seawater（I.Masao, H.Y.Inoue and H.Matsueda, 2000）

第 42 号　気象研究所／数値予報課統一非静力学モデル（斉藤和雄・加藤輝之・永戸久喜・室井ちあし，2001）
Documentation of the Meteorological Research Institute / Numerical Prediction Division Unified Nonhydrostatic Model (Kazuo Saito, Teruyuki Kato, Hisaki Eito and Chiashi Muroi，2001)

第 43 号　大気および海水中のクロロフルオロカーボン類の精密測定と気象研究所クロロフルオロカーボン類標準ガスの確立（時枝隆之・井上(吉川)久幸，2004）
Precise measurements of atmospheric and oceanic chlorofluorocarbons and MRI chlorofluorocarbons calibration scale (Takayuki Tokieda and Hisayuki Y. Inoue，2004)

第 44 号　PostScript コードを生成する描画ツール"PLOTPS"マニュアル（加藤輝之，2004）
Documentation of "PLOTPS": Outputting Tools for PostScript Code (Teruyuki Kato，2004)

第 45 号　気象庁及び気象研究所における二酸化炭素の長期観測に使用された標準ガスのスケールとその安定性の再評価に関する調査・研究（松枝秀和・須田一人・西岡佐喜子・平野礼朗・澤　庸介・坪井一寛・堤　之智・神谷ひとみ・根本和宏・長井秀樹・吉田雅司・岩野園城・山本　治・森下秀昭・鎌田匡俊・和田　晃，2004）
Re-evaluation for scale and stability of CO_2 standard gases used as long-term observations at the Japan Meteorological Agency and the Meteorological Research Institute (Hidekazu Matsueda, Kazuto Suda, Sakiko Nishioka, Toshirou Hirano, Yousuke, Sawa, Kazuhiro Tuboi, Tsutumi, Hitomi Kamiya, Kazuhiro Nemoto, Hideki Nagai, Masashi Yoshida, Sonoki Iwano, Osamu Yamamoto, Hideaki Morishita, Kamata, Akira Wada，2004)

第 46 号　地震発生過程の詳細なモデリングによる東海地震発生の推定精度向上に関する研究（地震火山研究部, 2005）
A Study to Improve Accuracy of Forecasting the Tokai Earthquake by Modeling the Generation Processes (Seismology and Volcanology Research Department, 2005)

第 47 号　気象研究所共用海洋モデル（MRI.COM）解説（海洋研究部, 2005）
Meteorological Research Institute Community Ocean Model (MRI.COM) Manual (Oceanographical Research Department, 2005)

第 48 号　日本海降雪雲の降水機構と人工調節の可能性に関する研究（物理気象研究部・予報研究部, 2005）
Study of Precipitation Mechanisms in Snow Clouds over the Sea of Japan and Feasibility of Their Modification by Seeding (Physical Meteorology Research Department, Forecast Research Department, 2005)

第 49 号　2004 年日本上陸台風の概要と環境場（台風研究部, 2006）
Summary of Landfalling Typhoons in Japan, 2004 (Typhoon Research Department, 2006)

第 50 号　栄養塩測定用海水組成標準の 2003 年国際共同実験報告（青山道夫, 2006）
2003 Intercomparison Exercise for Reference Material for Nutrients in Seawater in a Seawater Matrix (Michio Aoyama, 2006)

第 51 号　大気および海水中の超微量六フッ化硫黄(SF_6)の測定手法の高度化と SF_6 標準ガスの長期安定性の評価（時枝隆之、石井雅男、斉藤 秀、緑川 貴, 2007）
Highly developed precise analysis of atmospheric and oceanic sulfur hexafluoride (SF_6) and evaluation of SF_6 standard gas stability (Takayuki Tokieda, Masao Ishii, Shu Saito and Takashi Midorikawa, 2007)

第 52 号　地球温暖化による東北地方の気候変化に関する研究（仙台管区気象台, 環境・応用気象研究部, 2008）
Study of Climate Change over Tohoku District due to Global Warming (Sendai District Meteorological Observatory, Atmospheric Environment and Applied Meteorology Research Department, 2008)

第 53 号　火山活動評価手法の開発研究（地震火山研究部, 2008）
Studies on Evaluation Method of Volcanic Activity (Seismology and Volcanology Research Department, 2008)

第 54 号　日本における活性炭冷却捕集およびガスクロ分離による気体計数システムによる ^{85}Kr の測定システムの構築および 1995 年から 2006 年の測定結果（青山道夫, 藤井憲治, 廣瀬勝己, 五十嵐康人, 磯貝啓介, 新田 済, Hartmut Sartorius, Clemens Schlosser, Wolfgang Weiss, 2008）
Establishment of a cold charcoal trap-gas chromatography-gas counting system for ^{85}Kr measurements in Japan and results from 1995 to 2006 (Michio Aoyama, Kenji Fujii, Katsumi Hirose, Yasuhito Igarashi, Keisuke Isogai, Wataru Nitta, Hartmut Sartorius, Clemens Schlosser, Wolfgang Weiss, 2008)

第 55 号　長期係留による 4 種類の流速計観測結果の比較（中野俊也, 石崎 廣, 四竈信行, 2008）
Comparison of Data from Four Current Meters Obtained by Long-Term Deep-Sea Moorings (Toshiya Nakano, Hiroshi Ishizaki and Nobuyuki Shikama, 2008)

第 56 号　CMIP3 マルチモデルアンサンブル平均を利用した将来の海面水温・海氷分布の推定（水田 亮, 足立恭将, 行本誠史, 楠 昌司, 2008）
Estimation of the Future Distribution of Sea Surface Temperature and Sea Ice Using the CMIP3 Multi-model Ensemble Mean (Ryo Mizuta, Yukimasa Adachi, Seiji Yukimoto and Shoji Kusunoki, 2008)

第 57 号　閉流路中のフローセルを用いた分光光度法自動分析装置による海水の高精度 pH_T 測定（斉藤 秀, 石井雅男, 緑川 貴, 井上（吉川）久幸, 2008）
Precise Spectrophotometric Measurement of Seawater pH_T with an Automated Apparatus using a Flow Cell in a Closed Circuit（Shu Saito, Masao Ishii, Takashi Midorikawa and Hisayuki Y. Inoue, 2008）

第58号　栄養塩測定用海水組成標準の2006年国際共同実験報告（青山道夫,J. Barwell-Clarke, S. Becker, M. Blum, Braga E.S., S. C. Coverly, E. Czobik, I. Dahllöf, M. Dai, G. O Donnell, C. Engelke, Gwo-Ching Gong, Gi-Hoon Hong, D. J. Hydes, Ming-Ming Jin, 葛西広海, R. Kerouel, 清本容子, M. Knockaert, N. Kress, K. A. Krogslund, 熊谷正光, S. Leterme, Yarong Li, 増田真次, 宮尾 孝, T. Moutin, 村田昌彦, 永井直樹, G. Nausch, A. Nybakk, M. K. Ngirchechol, 小川浩史, J. van Ooijen, 太田秀和, J. Pan, C. Payne, O. Pierre-Duplessix, M. Pujo-Pay, T. Raabe, 齊藤一浩, 佐藤憲一郎, C. Schmidt, M. Schuett, T. M. Shammon, J. Sun, T. Tanhua, L. White, E.M.S. Woodward, P. Worsfold, P. Yeats, 芳村 毅, A. Youénou, Jia-Zhong Zhang, 2008）
2006 Inter-laboratory Comparison Study for Reference Material for Nutrients in Seawater（M. Aoyama, J. Barwell-Clarke, S. Becker, M. Blum, Braga E. S., S. C. Coverly, E. Czobik, I. Dahllöf, M. H. Dai, G. O. Donnell, C. Engelke, G. C. Gong, Gi-Hoon Hong, D. J. Hydes, M. M. Jin, H. Kasai, R. Kerouel, Y. Kiyomono, M. Knockaert, N. Kress, K. A. Krogslund, M.

Kumagai, S. Leterme, Yarong Li, S. Masuda, T. Miyao, T. Moutin, A. Murata, N. Nagai, G. Nausch, M. K. Ngirchechol, A. Nybakk, H. Ogawa, J. van Ooijen, H. Ota, J. M. Pan, C. Payne, O. Pierre-Duplessix, M. Pujo-Pay, T. Raabe, K. Saito, K. Sato, C. Schmidt, M. Schuett, T. M. Shammon, J. Sun, T. Tanhua, L. White, E.M.S. Woodward, P. Worsfold, P. Yeats, T. Yoshimura, A. Youénou, J. Z. Zhang, 2008）

気象研究所技術報告は，1978年（昭和53）年の初刊以来，気象研究所が必要の都度発行する刊行物であり，気象研究所の研究計画に基づき実施した研究に関する手法、データ、結果等についてのまとめ、または、すでに公表した研究論文類をとりまとめ総合的報告としたものを掲載する。

気象研究所技術報告　ISSN 0386-4049
第 59 号

平成22年2月　発行

編集兼
発行者　気象研究所

〒305-0052　茨城県つくば市長峰１－１
TEL(029)853-8535

印刷者　朝日印刷株式会社つくば支社
〒305-0046 茨城県つくば市東2-11-15