The FMS MOM4 User Guide

Stephen Griffies <`Stephen.Griffies@noaa.gov`>

Niki Zadeh <`Niki.Zadeh@noaa.gov`>

Table of Contents

1. Introduction

The purpose of this section is to provide an overview of MOM and the contents of this web page.

1.1. What is MOM?

The Modular Ocean Model (MOM) is a numerical representation of the ocean's hydrostatic primitive equations. It is designed primarily as a tool for studying the global ocean climate system, as well as capabilities for regional and coastal applications. MOM4 is the latest version of the GFDL ocean model whose origins date back to the pioneering work of Kirk Bryan and Mike Cox in the 1960s-1980s. It is developed and supported by researchers at NOAA's Geophysical Fluid Dynamics Laboratory (GFDL [http://www.gfdl.noaa.gov]), with critical contributions also provided by researchers worldwide. The purpose of this web guide is to provide general information about MOM4, and particular information for how to download and run the code. Here is a table [./MOM_history.pdf] summarizing the algorithmic history of MOM.

1.2. MOM4 Releases

MOM4 has had the following releases (note that MOM4p1 releases are distinguished by release dates).

- MOM4p0a: January 2004

- MOM4p0b: March 2004

- MOM4p0c: August 2004

- MOM4p0d: May 2005

- MOM4p1: 28 September 2007

- MOM4p1: 28 December 2007

- MOM4p1: Dec 2009

For each release, we aim to update MOM4 by enhancing features and documentation, and correcting bugs. Each version is generally compatible with the previous versions. However, as updates are made, we cannot guarantee that all features will bitwise agree across releases. Nonetheless, we do maintain a small selection of ``bitwise-legacy'' code to allow for certain modules to bitwise agreement across versions. As the maintenance of bitwise-legacy features represents an onerous task (e.g., bits change when A+B is altered to B+A), we recommend that researchers beginning new projects start with the most recent version, and that researchers with mature projects carefully test the new code prior to moving forward. Note that each of the MOM4 releases is documented on this web page.

1.3. MOM4 registration

We maintain the MOM4 source code and associated datasets at GForge [http://fms.gfdl.noaa.gov/]. MOM4 users are required to register [https://fms.gfdl.noaa.gov/gf/account/?action=UserAdd] at the GFDL GForge location [http://fms.gfdl.noaa.gov/]. Users need to register only once to get both the source code and datasets of MOM4. Registered users then need to request access [https://fms.gfdl.noaa.gov/extra/request-project-access.cgi] to the relevant project (MOM4p1 is the most recent MOM4 project).

1.4. MOM4 email list

Email concerning MOM4 should be directed to the MOM4-email list located at <oar.gfdl.MOM4p1@noaa.gov>. All questions, comments, and suggestions are to be referred to this list.

An archive of all emails is maintained at the MOM4 email archive [https://fms.gfdl.noaa.gov/gf/project/mom4p1/mailman/?action=index]. Note that by registering at GForge to access the code, you are automatically subscribed to the email list.

1.5. The MOM4 community

Since its release in January 2004, there have been hundreds of registrations with the MOM4 distribution, with each registration generally representing more than one user. This is a sizable user community. This community has proven to be a great resource, especially for users new to MOM4, and those with portability questions, some of which are beyond the abilities of GFDL scientists to answer.

1.6. Efficiency and Portability

The MOM4 team aims to provide code that is efficient, flexible, and transparent for use across a broad range of computer platforms. Balancing these aims is not always simple. For example, some of the most efficient code is also the least transparent. The MOM4 developers are scientists whose main concern is to support MOM4 as a tool for science research. This focus then leads us to weight transparency and portability over efficiency. However, we readily make efficiency modifications that are of a general nature, so please do feel free to volunteer any such changes.

Given the above aims, we have continued to support one avenue for code efficiency involving the allocation of arrays. MOM4 can be compiled in two ways: with static allocation of arrays or dynamic allocation. Static allocation is enabled at compile time via the cpp-preprocessor option MOM4_STATIC_ARRAYS. At GFDL, we have generally found that static allocation executables are faster than dynamic, since compilers like to know beforehand the size of the model arrays. Work on the SGI machines at GFDL has reduced the difference in efficiency between these two compilations. However, details of the model configuration strongly impact the differences in model speed. Additionally, we understand that on some platforms, the dynamic allocation results in faster code than static. Consequently, we have decided to maintain both the static and dynamic options, given the ambiguous results across platforms, compilers, model configurations, ect.

2. An Outline of MOM4

The purpose of this section is to outline certain features of MOM4.

2.1. Documentation

In addition to this online user guide, documentation for MOM4 is provided by the following LaTeX generated postscript documents:

1. A Technical Guide to MOM4 [./ guide4p0.pdf] by <Stephen.Griffies@noaa.gov>, <Matthew.Harrison@noaa.gov>, <Ronald.Pacanowski@noaa.gov>, and <Tony.Rosati@noaa.gov>. This is the primary reference for MOM4p0. It contains details about some of the numerical algorithms and diagnostics. Reference to MOM4p0 in the literature should refer to this document:

   > A Technical Guide to MOM4
   > GFDL Ocean Group Technical Report No. 5
   > S.M. Griffies, M.J. Harrison, R.C. Pacanowski, and A. Rosati
   > NOAA/Geophysical Fluid Dynamics Laboratory
   > August 2004
   > Available on-line at http://www.gfdl.noaa.gov/fms.

2. Elements of MOM4p1 [./guide4p1.pdf] by `<Stephen.Griffies@noaa.gov>` is the primary reference for MOM4p1. It contains details about some of the numerical algorithms and diagnostics. Reference to MOM4p1 in the literature should refer to this document:

> Elements of MOM4p1
> GFDL Ocean Group Technical Report No. 6
> Stephen M. Griffies
> NOAA/Geophysical Fluid Dynamics Laboratory
> Dec 2009
> 444 pages
> Available on-line at http://www.gfdl.noaa.gov/fms.

3. A theoretical rationalization of ocean climate models is provided by Fundamentals of Ocean Climate Models [http://pup.princeton.edu/titles/7797.html]. This book by `<Stephen.Griffies@noaa.gov>` was published by Princeton University Press in August 2004.

2.2. Embedded Documentation

The documentation of most Fortran modules in FMS is inserted right in the source code to enable consistency between the code and documentation. A Perl software tool is used to extract documentation from the source code to create a corresponding html module. For example, documentation for shared/diag_manager/diag_manager.F90 module is shared/diag_manager/diag_manager.html. In general, the embedded documentation is a good starting point to understand the Fortran module, though ultimately the Fortran code is the final source for information.

2.3. MOM4 Characteristics

Although MOM4 shares much in common with earlier versions of MOM, it possesses a number of computational, numerical, and physical characteristics that are noteworthy. The following provides an overview of the main characteristics of MOM4 (please refer to A Technical Guide to MOM4 [./ guide4p0.pdf] and Elements of MOM4p1 [./guide4p1.pdf] for references).

Computational characteristics of MOM4 include the following.

*   MOM4 is coded in Fortran 90 and physical units are MKS.

*   MOM4 meets the code standards set by the GFDL Flexible Modeling System (FMS [http:/ / www.gfdl.noaa.gov/fms]). It also utilizes a substantial number of software infrastructure modules shared by other FMS-based models. In particular, all I/O (e.g., restarts, forcing fields, initial fields) is handled via NetCDF [http://www.unidata.ucar.edu/packages/netcdf/].

*   There are very few cpp-preprocessor option (i.e., `ifdefs`). One is associated with the handling of memory in the model (denoted MOM4_STATIC_ARRAYS in MOM4p1). Another option enables the suite of tracers associated with the MOM4p1 implementation of the GFDL ocean biogeochemistry model. The reason for including this second ifdef is to minimize the compile time required for the many cases when one chooses NOT to use the biogeochemistry (adding the biogeochemistry greatly lengthens the compile time). Other options for vertical coordinates, physical parameterizations, and dynamical choices are handled via namelists and/or settings within tables. Removing `ifdefs` allows for more readable code that possesses a higher level of error checker handling. It also facilitates testing various algorithms using the same executable.

*   2D (latitudinal/longitudinal) horizontal domain decomposition is used for single or multiple parallel processors. Correspondingly, 3D arrays are dimensioned (i,j,k) instead of the (i,k,j) structure used in MOM3 and

earlier. Notably, MOM4 has no memory window or slabs.

Numerical and kinematic/dynamic characteristics of MOM4p0 and MOM4p1 include the following. For a more complete discussion of the differences between MOM4p0 and MOM4p1, please refer to this link for a synopsis [./mom4p1_synopsis.pdf].

- Generalized orthogonal horizontal coordinates are used. GFDL is supporting both the standard spherical coordinates as well as the "tripolar" grid of Murray (1996). Details are provided in A Technical Guide to MOM4 [./guide4p0.pdf].

- Bottom topography is generally represented using the partial cells of Pacanowski and Gnanadesikan (1998). The older full cell approach is available as a namelist in the preprocessing code used to generate the grid specification file.

- The dynamics/kinematics for MOM4p0a and MOM4p0b are based on the non-Boussinesq method of Greatbatch et al. (2002). The Boussinesq option is available via a namelist. This option was jettisoned in MOM4p0c and MOM4p0d for purposes of code brevity, anticipating the move to pressure vertical coordinates. In MOM4p1, the non-Boussinesq dynamics and kinematics is available via pressure-based vertical coordinates. Formulational and algorithmic details are provided in Elements of MOM4p1 [./guide4p1.pdf].

- In MOM4p0, the sole vertical coordinate is the geopotential coordinate. In MOM4p1, the six following vertical coordinates are available: geopotential, zstar, terrain following, pressure, pstar, and terrain following pressure. Formulational and algorithmic details are provided in Elements of MOM4p1 [./guide4p1.pdf]. Refer to this document [./ mom4p1_synopsis.pdf] for a synopsis of the new vertical coordinates available in MOM4p1.

- The time tendency for tracer and baroclinic velocity can be discretized two ways. (1) The first approach uses the traditional leap-frog method along with a Robert-Asselin time filter. This method is available in MOM4p0a, MOM4p0b and MOM4p0c. Note that the Euler forward or Euler backward mixing time step used in earlier MOMs has been eliminated. (2) MOM4p0c and later releases provide an additional time stepping method, which is strongly recommended and now universally used at GFDL. Here, the time tendency is discretized with a two-level forward step, which eliminates the need to time filter. Tracer and velocity are staggered in time. For certain model configurations, this scheme is roughly twice as stable as the leap-frog, thus allowing for a factor of two in computational savings. Without the time filtering needed with the leap-frog, the new scheme conserves total tracer to within numerical roundoff. This scheme shares much in common with time stepping used in the GFDL General Ocean Layer Dynamics (GOLD) and the MIT GCM. It is the default in MOM4p0c and more recent releases. Details of both the leap-frog and two-level schemes are provided in Fundamentals of Ocean Climate Models [http://pup.princeton.edu/titles/7797.html].

- The sole external mode solver is a variant of the Griffies et al. (2001) explicit free surface. There are two time stepping schemes supported: (1) leap-frog and (2) predictor-corrector. The predictor-corrector is more stable and is thus the default method. All model grid cells have time dependent volume (Boussinesq) or mass (nonBoussinesq), thus allowing for total tracer to be conserved to within numerical roundoff. The linearized free surface method used in MOM3 (and many other implementations of the free surface in z-models) has been jettisoned since it precludes precise tracer conservation. Details are provided in Fundamentals of Ocean Climate Models [http://pup.princeton.edu/titles/7797.html] as well as Elements of MOM4p1 [./guide4p1.pdf].

- McDougall et al. (2003) equation of state has been implemented in MOM4p0, with in situ density a function of the local potential temperature, salinity, and hydrostatic pressure (baroclinic pressure plus free surface pressure plus applied pressure from the atmosphere and sea ice). Details are provided in Fundamentals of Ocean Climate Models [http://pup.princeton.edu/titles/7797.html]. The Jackett etal. (2006) equation of state is implemented in MOM4p1, with the Conservative Temperature variable from McDougall (2005) also available. Details are provided in Elements of MOM4p1 [./guide4p1.pdf].

- Tracer advection is available using various schemes. The centered 2nd, 4th, 6th order schemes are available, as documented in The MOM3 Manual of Pacanowski and Griffies (1999) [http://www.gfdl.gov/~smg/MOM/web/guide_parent/guide_parent.html]. The 4th and 6th order schemes assume constant grid spacing, which simplifies the code though compromises accuracy on the more commonly used non-uniform grids. The Quicker scheme documented by Holland et al. (1998) and The MOM3 Manual [http://www.gfdl.gov/~smg/MOM/web/guide_parent/guide_parent.html] is available. Two multi-dimensional flux limited schemes have been ported from the MIT GCM. These schemes are monotonic and have been found to be roughly the same

cost as the Quicker scheme. GFDL researchers have found the Sweby scheme to be satisfying for many applications, such as biogeochemistry. Hence, effort has been made to enhance this scheme's efficiency in MOM4p0c. In MOM4p1, the flux limited as well as unlimited scheme from Prather (1986) is available. This scheme has been found to be quite accurate, though requires the addition of many three-dimensional arrays per tracer. Finally, a multi-dimensional piecewise parabolic scheme (MDPPM) has been ported to MOM4p1 from the MITgcm.

- Tidal forcing from the various lunar and solar components are available to force the free ocean surface. Details are provided in Elements of MOM4p1 [./guide4p1.pdf].

- Open boundary conditions are available to allow open boundaries in either of the north, south, east, or west directions. This module has been extensively rewritten for purposes of code generality, with numerous new options for radiating conditions. This scheme was developed with critical input from modelers using MOM4p1 for regional simulations.

Physical parameterizations available in MOM4 include the following.

- Neutral tracer physics includes Redi neutral diffusion according to Griffies et al. (1998), and Gent-McWilliams stirring according to the Griffies (1998) skew-flux method. Two-dimensional flow dependent diffusivities are available and can be determined in many different ways, such as the depth integrated Eady growth rate and Rossby radius of deformation, as motivated by the ideas of Held and Larichev (1996) and Visbeck et al. (1997). Details are provided in Fundamentals of Ocean Climate Models [http:/ / pup.princeton.edu/titles/7797.html]. In MOM4p1, the treatment of neutral physics near the domain boundaries is available as per Ferrari and McWilliams. Details are provided in Elements of MOM4p1 [./ guide4p1.pdf].

- In addition to the implementation of Gent-McWilliams as a skew tracer diffusion, MOM4p1 provides options which transforms these effects into the momentum equation. The standard method for this transformation is via the approach of Greatbatch and Lamb (1990) and Greatbatch (1998), in which the vertical viscosity is modified. Another approach is due to Aiki, Jacobson, and Yamagata (2004), in which momentum is dissipative with a Rayleigh-like drag term. This latter approach from Aiki etal is experimental and not recommended for general usage.

- Vertical mixing schemes include the time-independent depth profile of Bryan and Lewis (1979), the Richardson number dependent scheme of Pacanowski and Philander (1981), the KPP scheme of Large et al. (1994), and in MOM4p1 the Generalized Ocean Turbulence Model (GOTM) [http://www.gotm.net/] is also available.

- Horizontal friction schemes include uniform and grid dependent viscosity schemes, as well as the shear-dependent Smagorinsky viscosity according to Griffies and Hallberg (2000). The anisotropic scheme of Large et al. (2001) and Smith and McWilliams (2002) has been implemented. Details are provided in Fundamentals of Ocean Climate Models [http://pup.princeton.edu/titles/7797.html].

- Topographically oriented tracer diffusion introduces enhanced diffusion when heavy parcels are above lighter parcels. It is implemented according to the ideas of Beckmann and Döscher (1997) and Döscher and Beckmann(1999). Details are provided in A Technical Guide to MOM4 [./guide4p0.pdf] as well as Elements of MOM4p1 [./guide4p1.pdf].

- The "overflow" scheme of Campin and Goosse (1999) has been implemented, whereby gravitationally unstable fluid parcels are allowed to move downslope via an upwind advection scheme. Details are provided in A Technical Guide to MOM4 [./guide4p0.pdf] as well as Elements of MOM4p1 [./guide4p1.pdf].

- Modifications to the Campin and Goosse (1999) scheme have been implemented in MOM4p1 that allow for the transport of tracers to horizontally distant columns, so long as such is density favored. Details are provided in Elements of MOM4p1 [./guide4p1.pdf].

- Penetration of shortwave radiative heating into the upper ocean is generally attenuated by the inclusion of chlorophyll data (MOM4p0), or via chlorophyll data or an active biogeochemistry model (MOM4p1). Details are provided in A Technical Guide to MOM4 [./guide4p0.pdf].

Miscellaneous features of the code released with MOM4 include the following.

- MOM4 comes with the following tracer packages: (1) ideal age tracer (2) tracers for the OCMIP biotic protocol (3) CFC tracers, (4) an ideal residency time tracer which tags parcels according to physical processes, (5) ideal passive tracer package with online generation of the idealized profile, and (6) the GFDL prognostic ecosystem model TOPAZ.

- MOM4 has numerous diagnostics for checking algorithm and solution integrity. These diagnostics include budgets for energetic consistency, tracer conservation, solution stability, etc. Additional diagnostics are available for numerous fields of relevance to the different physics schemes, as well as term balances.

- MOM4 is distributed with a prognostic sea ice model (SIS), as well as the land model and finite volume atmospheric model used for GFDL's CM2.1 coupled climate model.

2.4. MOM4 and FMS

MOM4 has been coded within GFDL's Flexible Modeling System (FMS [http://www.gfdl.noaa.gov/fms]). Doing so allows for MOM4 developers to use numerous FMS infrastructure and superstructure modules that are shared amongst various atmospheric, ocean, sea ice, land, vegetative, etc. models. Common standards and shared software tools facilitate the development of high-end earth system models, which necessarily involves a wide variety of researchers working on different computational platforms. Such standards also foster efficient input from computational scientists and engineers as they can more readily focus on common computational issues.

The following list represents a sample of the FMS `shared` modules used by MOM4.

- `time manager` [../../shared/time_manager/time_manager.html]: keeps model time and sets time dependent flags

- `coupler` [../../coupler/coupler_main.html]: used to couple MOM4 to other component models

- `I/O` [../../shared/mpp/mpp_io.html]: to read and write data in either NetCDF, ASCII, or native formats

- `parallelization tools` [../../shared/mpp/mpp.html]: for passing messages across parallel processors

- `diagnostic manager` [../../shared/diag_manager/diag_manager.html]: to register and send fields to be written to a file for later analysis

- `field manager` [../../shared/field_manager/field_manager.html]: for integrating multiple tracers and organizing their names, boundary conditions, and advection schemes

The FMS infrastructure (the "Riga version") forms the basis for the December 2009 release of MOM4p1.

The Flexible Modeling System (FMS [http://www.gfdl.noaa.gov/fms]), including MOM4, is free software; you can redistribute it and/or modify it and are expected to follow the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

FMS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with MOM4; if not, write to:

Free Software Foundation, Inc.
59 Temple Place, Suite 330
Boston, MA 02111-1307
USA
or see: http://www.gnu.org/licenses/gpl.html

2.5. MOM4 test cases

MOM4 is distributed with a suite of test cases. Below is an outline of the main test cases available with MOM4p1. Full details are discussed in Elements of MOM4p1 [./guide4p1.pdf].

Warning

Many of the test cases are *NOT* sanctioned for their physical relevance. They are instead provided for the user to learn how to run MOM4, and to verify the numerical and/or computational integrity of the code. *PLEASE* do not assume that the experiments will run for more than the short time selected in the sample runscripts.

`mom4_atlantic`: This regional model tests the open boundary condition option in MOM4p1.

`mom4_bowl`: This experiment has a sector model with a bowl topography.

`mom4_box`: This experiment has a sector model with a flat bottom topography. This is the simplest test case.

`mom4_box_channel`: This idealized experiment has a channel at the southern portion of the domain.

`mom4_core`: This experiment is global with a tripolar grid with roughly "1-degree" resolution and 50 vertical levels. The ocean is coupled to the GFDL sea ice model. The configuration is forced with the Normal Year forcing from the Coordinated Ocean Reference Experiment (data developed by Bill Large and Stephen Yeager at NCAR). This is a large model, and it is similar (though not the same) to the ocean and ice configuration used for the GFDL IPCC simulations.

`mom4_dome`: This is an idealized overflow configuration useful for testing overflow schemes.

`mom4_ebm`: This is a global model configuration coupled to the GFDL energy balance atmosphere plus the GFDL ice model.

`mom4_iom`: This experiment is a regional Indian Ocean model setup during a modeling school in Bangalore, India during October 2004.

`mom4_mk3p5`: This is a global spherical coordinate model which is based on the configuration used by CSIRO Marine and Atmospheric Research in Aspendale, AUS.

`mom4_symmetric_box`: This is an idealized configuration that is symmetric about the equator and uses symmetric forcing. The simulation should thus remain symmetric about the equator.

`mom4_torus`: This is an idealized simulation that is periodic in the x and y directions. It is useful for testing tracer advection schemes.

`CM2p1`: This test case represents a release of the CM2.1 coupled climate model used by GFDL scientists for the IPCC AR4 assessment of climate. This test includes the ocean configuration, land model, atmospheric model, and sea ice model setup as in CM2.1. Note that the original CM2.1 used the MOM4.0 code, whereas the CM2p1 test instead uses MOM4p1. However, GFDL scientists have verified that the climate simulations are compatible.

3. Contributing to MOM4

As with previous versions of MOM, the GFDL-MOM4 developers aim to provide the international climate research community with a repository for robust and well documented methods to simulate the ocean climate system. Consequently, we encourage interested researchers to contribute to MOM4 by commenting on code features, and providing new modules that enhance simulation integrity (e.g., a new physical parameterization or new advection scheme) or increase the model's functionality.

To be included in the GFDL-distributed version of MOM4, code must meet the following criteria:

1. The module must be written in Fortran 90 and minimally touches other parts of the model (i.e., modular).

2. Satisfaction of the FMS [http://www.gfdl.noaa.gov/fms] code specifications outlined in the FMS Developers' Manual [http://www.gfdl.noaa.gov/~vb/FMSManual/FMSManual.html]

3. Compatibility with the MOM4 test cases [#MOM4 test cases]

4. Thorough and pedagogical documentation of the module for inclusion in Elements of MOM4p1 [./guide4p1.pdf] (a Latex document)

5. Comments within the code emulating other parts of the model so that HTML documentation files can be generated by our converter

4. Source code and data sets

The purpose of this section is to outline methods required to obtain the source code and associated datasets.

4.1. Obtaining source code and data sets

The FMS [http://www.gfdl.noaa.gov/fms] development team uses a local implementation of GForge [http://fms.gfdl.noaa.gov] to serve FMS software, located at http://fms.gfdl.noaa.gov. In order to obtain the source code and data sets, you must register [https://fms.gfdl.noaa.gov/gf/account/?action=UserAdd] as an FMS user on our software server. After submitting the registration form on the software server, you should receive an automatically generated confirmation email within a few minutes. Clicking on the link in the email confirms the creation of your account.

After your account has been created, you should log in [https://fms.gfdl.noaa.gov/gf/account/?action=Login] and request access to the MOM4P1 [https://fms.gfdl.noaa.gov/extra/request-project-access.cgi] project. Once the project administrator grants you access, you will receive a second email notification. This email requires action on the part of the project administrator and thus may take longer to arrive. The email will contain a software access password along with instructions for obtaining the release package (a tar file).

You may download the MOM4 tar file from https://fms.gfdl.noaa.gov/gf/project/mom4p1/frs/?action=index.

The data set and outputs for selected MOM4p1 test experiments are available via ftp [ftp://ftp.gfdl.noaa.gov/perm/MOM4/mom4p1_pubrel/exp] More details can be found in the quickstart_guide.html.

4.2. Description of the data sets

There are many datasets provided with the various MOM4 test cases. Most documentation of the datasets can be found in the NetCDF files themselves. We here provide a summary of some of the datasets.

1. The topography data set for the global 1-degree ocean model configuration in `mom4_core` is based on a coarsened version of that kindly provided by Andrew Coward and David Webb at the Southampton Oceanography Centre [http://www.soc.soton.ac.uk/JRD/OCCAM/welcome.html]. Their topography is a montage of that developed by Smith and Sandwell [http://topex.ucsd.edu/marine_topo/mar_topo.html] (1997) by satellite data in the region of 72°S to 72°N, the NOAA (1988) 5-minute global topography ETOPO5 [http://www.ngdc.noaa.gov/mgg/global/etopo5.HTML], and the International Bathymetric Chart of the Arctic Ocean (IBCAO [http://www.ngdc.noaa.gov/mgg/bathymetry/arctic/arctic.html]).

2. The chlorophyll-a density data set was compiled by Colm Sweeney [mailto:Colm.Sweeney@noaa.gov], using data from James A. Yoder [http://www.gso.uri.edu/faculty/yoder.html] and Maureen A. Kennelly at the Graduate School of Oceanography [http://www.gso.uri.edu/], University of Rhode Island [http://www.uri.edu/]. This data set contains monthly chlorophyll concentrations from the SeaWiFS [http://seawifs.gsfc.nasa.gov/SEAWIFS.html] satellite for the period 1999-2001.

3. Temperature and salinity initial and boundary conditions are provided by the NOAA [http://www.noaa.gov] National Oceanographic Data Center (NODC [http://www.nodc.noaa.gov/]) World Ocean Atlas (WOA [http://www.nodc.noaa.gov/OC5/indprod.html]).

4. The Large and Yeager (2004) [http://data1.gfdl.noaa.gov/nomads/forms/mom4/CORE.html] dataset for use in the Coordinated Ocean-ice Reference Experiment (CORE) simulation is available at this link.

All datasets released with MOM4 are in NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format, since this format is widely used in the community. A number of useful tools are available here [http://nco.sourceforge.net/] that allow the user to perform some necessary operations (editting attributes, merging, etc.) on a NetCDF file.

5. Setting up an experiment with MOM4

MOM4 is distributed with code used to generate model grids, initial conditions, and boundary conditions. Each step must be performed prior to running the ocean model. The steps used during this experimental setup stage are generally termed "preprocessing", and the code used for these purposes is under the /preprocessing directory in the MOM4 distribution. The purpose of this section of the User Guide is to outline this code and its usage. Further details of usage and algorithms can be found in the internal documentation within the various preprocessing code modules.

5.1. General comments

We start this section with some general comments regarding the setup of a model experiment.

--Setting up an experiment is critical part to the success of a research or development project with MOM4. It is important that the user take some time to understand each of the many steps, and scrutinize the output from this code.

We have endeavored over the years to provide tools facilitating the ready setup of a new experiment. However, we remain unable to provide code that does everything possible under the sun. Additionally, all features that are provided here may not be fully tested. For these reasons, the preprocessing code continues to evolve as use and functionality evolve. We sincerely appreciate ALL comments about code and documentation, especially comments regarding clarity, completeness, and correctness. Your input is essential for the improvement of the code and documentation.

--Many steps in idealized experiments that were formerly performed while running earlier MOM versions have been extracted from MOM4 and placed into preprocessing. Hence, even if you are running an idealized experiment, it is likely that you will need to perform some if not all of the preprocessing steps discussed here.

--If you have a problem that is not addressed here, then please feel free to query the MOM4 email list. No question is too silly, so please ask!

--All code used to setup an experiment with MOM4 is written in Fortran 90/95 except make_xgrids, which is written in C. Most code is dependent on FMS shared code for the purpose of parallization and interpolation. In addition to the documentation provided here, there are comments within the code to help users debug and to make modifications to suit their purpose.

--Some users make nontrivial changes of general use. With your support, assistance, and maintenance, we will endeavour to include your changes in future releases.

5.2. Creation of the ocean/ice grid (mosaics)

Mosaics is the name given to a general framework to define structured grids at GFDL. The details of this scheme are being incorporated by others outside of GFDL as well. Most of the recent global models developed at GFDL employ the conventions of Mosaic. However, MOM4p1 is backwards compatible, so that it still supports grids generated with older software (discussed here [#Creation of the ocean/ice grid (mosaics)]) For a description of how to create grids using the newer Mosaic tools, which are more general than the older approaches, then please access this link [mosaic_tool.html]

5.3. Creation of the ocean/ice grid (pre-mosaics)

Within GFDL FMS, ocean and ice are assumed to share the same grid. This means that the two models read in the same grid specification file. Even so, the domain decomposition on parallel systems may be different, and indeed they generally are due to different load balance issues between the two models.

Even though the ocean and ice models read the same grid specification file, they use the information from the grid file in a slightly different manner when setting up the respective model's tracer grid. In particular, the ocean model reads the tracer location directly from the arrays (x_T/geolon_t, y_T/geolat_t) written in the grid specification file. In contrast, the GFDL ice model reads (x_vert_T/geolon_vert_t, y_vert_T/geolat_vert_t) from the grid specification file and then averages these four vertex locations to get the tracer location used in the ice model. The result is that diagnostics output from the two models have ocean and ice fields at slightly different locations for cases such as the tripolar grid when the grid is not spherical.

The ocean/ice grid specification file is generated by executing the ocean_grid_generator utility. The ocean_grid_generator utility generates the horizontal grid, vertical grid, and topography. A C-shell script is provided to compile relevant code to generate and run the executable to produce the grid file. To create the desired grid and topography, setting namelist options within the runscript is needed.

The horizontal grid can be conventional lon-lat spherical grid or a reprojected rotated tripolar grid (R. Murray, "Explicit generation of orthogonal grids for ocean models", 1996, J.Comp.Phys., v. 126, p. 251-273.). The choice is controlled by the namelist option "tripolar grid" (true for tripolar grid and false for lon-lat spherical grid). Note that Cartesian beta-plane and f-plane geometries are set up within MOM4, not within the grid generation preprocessing steps discussed here (see mom4/ocean_core/ocean_grids.F90 for beta-plane and f-plane namelist options).

The grid_spec file contains the following horizontal grid information: geographic location of T,E,C and N-cell (Tracer, East, Corner, and North cells), half and full cell lengths (in meters), rotation information between logical (i.e., grid oriented) and geographic east of cell. The complete description of the horizontal grid and namelist option is available in hgrid [../../preprocessing/generate_grids/ocean/hgrid.html]

The vertical grid information includes depth of tracer points and tracer_boundaries. The complete description of namelist option is available in vgrid [../../preprocessing/generate_grids/ocean/vgrid.html]

The topography can be idealized (various examples are provided and others can be easily added through emulating those provided) or remapped from a source topography dataset. The type of topography is specified by the namelist variable "topography". Namelist "topog_depend_on_vgrid" specifies if the topography will depend on the vertical grid or not. To generate a grid for MOM4, "topog_depend_on_vgrid" should always be true. A useful option for those upgrading older models to MOM4 is "adjust_topo". If this option is set to false, there will be no adjustments made to the topography. See topog [../../preprocessing/generate_grids/ocean/topog.html] for further details about topography namelist options.

5.4. The exchange grid for coupled models

"Exchange grid" information is required for coupled models (i.e., ocean/ice coupled to land and/or atmosphere) that employ the GFDL coupler technology. The exchange grid is defined by taking the union of the ocean/ice grid with the atmosphere and land grids. This union is then used to compute area integrals to allow for conservative mapping of fluxes between the component models.
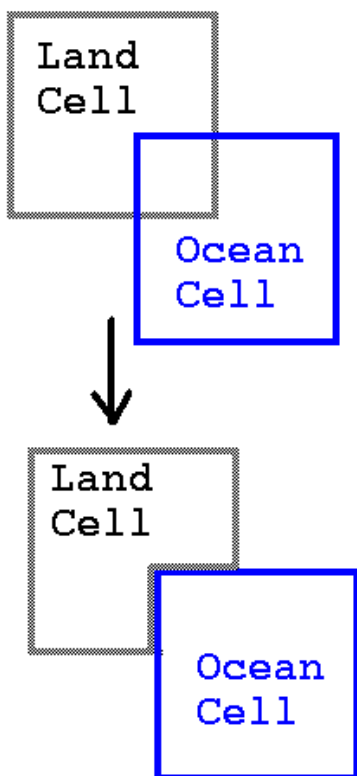
The exchange grid information is generated by executing the make_xgrids

utility. The execution of the make_xgrids utility will generate a netcdf file with the name grid_spec.nc. The grid_spec.nc contains the component model grids as well as the exchange grid information. In particular, the utility make_xgrids generates two exchange grids used by the FMS coupler: one grid for surface fluxes and another for runoff. **make_xgrids** is created by compiling its C source:

```
cc -O -o make_xgrids make_xgrids.c -I/usr/local/include -L/usr/local/lib
-lnetcdf -lm
```

creates the make_xgrids executable from C-source and the netCDF and standard math libraries. It is executed with the command

```
make_xgrids -o ocean_grid.nc -a atmos_grid.nc -l land_grid.nc
```

This execution produces a grid_spec.nc file (input files containing grid information for the ocean/sea-ice, atmosphere and land component models are indicated by the -o, -a and -l flags, respectively). The grid files ocean_grid.nc, atmosphere_grid.nc, and land_grid.nc all can be generated separately through the ocean_grid_generator utility. Normally at GFDL we select the same atmosphere and land model grid, but such is not necessary. When the land and atmosphere grids are the same, then we can reduce the execute command to

```
make_xgrids -o ocean_grid.nc -a atmos_grid.nc
```

If you further decide to choose same ocean, atmosphere and land grid, the execute command will be

```
make_xgrids -o ocean_grid.nc -a ocean_grid.nc
```

make_xgrids expects a netCDF format input specification for each of the component model grid files. For the ice/ocean grid (ocean_grid.nc), the following three fields are required:

1. wet - a 2D array of double precision numbers set to 1.0 where the ice and ocean models are active and 0.0 elsewhere. wet has im indices in the i-direction (pseudo east-west) and jm indices in the j-direction (pseudo north-south). These correspond to the size of the global arrays of temperature, salinity and ice thickness in the coupled climate model.

2. x_vert_T and y_vert_T - 3D double precision arrays (dimensioned im * jm * 4) that contain the longitudes and latitudes (respectively) of the four corners of T- cells. The numbers are in degrees.

For the netCDF format input specification for the atmosphere and land grid (atmos_grid.nc and/or land_grid.nc), x_vert_T and y_vert_t are required.

make_xgrids copies all fields of the ice/ocean grid specification file to its output file, grid_spec.nc, and then appends fields that specify the atmosphere and land model grids and then the surface and runoff exchange grids.

Using the Sutherland-Hodgeman polygon clipping algorithm (reference in next paragraph) for model cell interaction calculation, make_xgrids takes care that the land and ocean grids perfectly tile the sphere. The land model's domain is defined as that part of the sphere not covered by ocean (where wet=0 on the ocean grid). To accomplish this, the land cells must be modified to remove the ocean parts. This is done in make_xgrids by first taking the intersections of atmosphere and land cells. The overlap area between these cells and active ocean cells are then subtracted. Finally, the modified atmosphere/land intersections are aggregated into land cell areas and atmosphere/land exchange cell areas.

Model cell intersections are calculated using the Sutherland-Hodgeman polygon clipping algorithm (Sutherland, I. E. and G. W. Hodgeman, 1974: Reentrant polygon clipping, CACM, 17(1), 32-42.). This algorithm finds the intersection of a convex and arbitrary polygon by successively removing the portion of the latter that is "outside" each boundary of the former. It can be found in many computer graphics text books (e.g., Foley, J. D., A. van Dam, S. K. Feiner, and J. F. Hughes, 1990: Computer graphics: principles and practice, second edition. Addison Wesley, 1174 pp.). The implementation in make_xgrids is particularly simple because the clipping polygon is always a rectangle in longitude/latitude space. For the purpose of finding the line intersections in the clipping operations, the cell boundaries are assumed to be straight lines in longitude/latitude space. This treatment is only perfectly accurate for cells bounded by lines of longitude and latitude.

Spherical areas are calculated by taking the integral of the negative sine of latitude around the boundary of a polygon (Jones, P. W., 1999: First- and second-order conservative remapping schemes for grids in spherical coordinates. Monthly Weather Review, 127, 2204-2210.). The integration pathways are again straight lines in longitude/latitude space. make_xgrids checks that the sphere and the individual cells of the atmosphere and ocean grids are tiled by the surface exchange cells. The fractional tiling errors are reported.

5.5. Initial and Boundary Conditions

After generating the model grid, it is time to generate the initial and boundary conditions (ICs and BCs). These conditions are specific to the details of the model grid, so it is necessary to have the grid specificiation file in hand before moving to the IC and BC generation.

There are two options for ICs and BCs.

--Idealized Conditions. These conditions are based on subroutines that design idealized setups for either initial conditions (e.g., exponential temperature profile) or boundary conditions (e.g., cosine zonal wind stress). Code for these purposes is found in the idealized_ic and idealized_bc directories in the MOM4 distribution. Details of

available namelist choices are in the documentation file idealized_ic.html as well as the comments within the source code itself. Users can readily incorporate their favorite idealized IC or BC into the MOM4 idealized pre-processing step by emulating the code provided.

--Realistic Conditions. These ICs and BCs generally result from a regridding routine to bring, say, the Levitus analysis onto the model grid for initializing a model, or for mapping surface fluxes onto the grid for boundary conditions. Code enabling the regridding functions is found in the preprocessing/regrid_2d, preprocessing/regrid_3d and preprocessing/regrid directories in the MOM4 distribution.

In the remainder of this section, we detail code to generate the ICs and BCs of use for MOM4.

5.5.1. 2d Regridding: the common approach

It is typical for air-sea fluxes of momentum, heat, and mosture to live on a grid distinct from the ocean model grid. In particular, most analysis are placed on a spherical latitude-longitude grid, whereas most global ocean models configured from MOM4 are run with tripolar grids.

When running an ocean or ocean-ice model, it is useful to map the boundary fluxes onto the ocean model grid prior to the experiment. This preprocessing step saves computational time that would otherwise be needed if the fluxes were mapped each time step of a running experiment. To enable this regridding, one should access code in the preprocessing/regrid_2d directory. The original data must be on a latitude-longitude grid to use regrid_2d. The target/destination grid can be either latitude-longitude with arbitrary resolution, or tripolar with arbitrary resolution.

5.5.2. 2d Regridding: the less common approach

In some cases, one may wish to take a set of forcing fields from one tripolar MOM4 experiment and regrid them onto another tripolar MOM4 experiment with different grid resolution. In this case, it is necessary to regrid before running the experiment.

As of the MOM4p0d distribution, there is a regridding tool within the preprocessing/regrid directory that enables one to regrid fields on one tripolar grid to another tripolar grid. Indeed, one can regrid source data from any logically rectangular grid (e.g., latitude-longitude grid or tripolar grid) to a target/destination grid that is any logically rectangular grid.

Note that this is new code, and so has been tested only for particular cases. So the user should be extra careful to scrutinize the results.

5.5.3. Setting the on_grid logical in the data_table

The "on_grid" logical in the data_table indicates whether an input file is on the grid of the model or not.

on_grid=.true. means that the input file is on the same grid as the ocean model. This is the recommended setting for models running with specified atmospheric forcing from data or an analysis product.

on_grid=.false. means the input file has data on a grid differing from the ocean model. This feature is allowed ONLY if the input data lives on a spherical grid. This is a relevant setting if one wishes to keep the input data on their native spherical grid. If the input data is non-spherical, then on_grid=.false. is NOT supported. Instead, it is necessary to preprocess the data onto the ocean model grid.

5.5.4. Regridding river runoff data

The tool preprocessing/runoff_regrid is of use to grid river runoff data onto the ocean model grid. In this case,

runoff is moved to a nearest ocean/land boundary point on the new grid. Note that the source runoff dataset must be on a spherical latitude-longitude grid, whereas the target/destination grid can be spherical or tripolar. The regridding algorithm is conservative.

The conservative regridding scheme used in runoff_regrid is an area average scheme, which is similiar to the algorithm used in coupler flux exchange. If any land point has runoff data, after remapping runoff data onto destination grid, the runoff value of that land point will be moved to the nearest ocean point. Before using this tool, you must use make_xgrids to generate exchange grid information between the source grid and destination grid. The complete description can be found in runoff_regrid.html.

5.5.5. Two ways to specify surface boundary fluxes

There are two ways to specify surface boundary fluxes when using the coupler feature of FMS. One is through flux exchange, and this employs a conservative algorithm as appropriate for running a coupled ocean-atmosphere model. It is assumed that the atmospheric model grid is spherical with arbitrary resolution. The other method is through data override, and this uses a non-conservative scheme. Data override is of use to selectively remove, say, one of the fluxes coming from an atmospheric model and replace this flux with that from data. GFDL modelers have found this feature to be very useful in diagnosing problems with a coupled model.

5.5.6. 3d Regridding for initial conditions or sponges

When generating realistic initial conditions for an ocean experiment, one generally requires the gridding of temperature and salinity, such as from the Levitus analysis product, onto the model's grid. For this purpose, we are in need of vertical grid information in addition to horizontal 2d information required for the surface boundary conditions. Hence, we use the preprocessing/regrid_3d. A similar procedure is required to develop sponge data.

The original data must be on a spherical grid in order to use regrid_3d. If the original data is on a tripolar grid, we should use preprocessing/regrid, which can map data from any logical rectangular grid onto any logical rectangular grid.

5.5.7. Comments on the regridding algorithms

For preprocessing/regrid_3d, preprocessing/regrid_2d and preprocessing/regrid, regridding is accomplished non-conservatively using a nearest neighbor distance weighting algorithm, or bilinear interpolation. The interpolation algorithm is controlled through the namelist option "interp_method".

Bilinear interpolation is recommended for most cases since it provides a smooth interpolation when regridding from coarse grid to fine grid (the usual situation with model destination grids typically having resolution more refined than source data products), and it is more efficient. Efficiency can become a particularly important issue when developing initial and boundary conditions for a refined resolution model.

If the original data is on a tripolar grid, nearest neighbor distance weighting interpolation found in preprocessing/regrid must be used, since bilinear interpolation assumes the original data is on a latitude-longitude grid. For preprocessing/regrid_2d, preprocessing/regrid_3d and preprocessing/regrid using the nearest neighbor distance weighting algorithm, a maximum distance (in radians) can be selected using the namelist value max_dist. Namelist option "num_nbrs" can be adjusted for speed, although for most applications this refinement is not necessary.

The complete namelist description for these algorithms can be found in regrid_2d.html, regrid_3d.html and regrid.html.

5.5.8. Acceptable data formats

When the input data is on a latitude-longitude grid, preprocessing/regrid_2d and preprocessing/regrid_3d can be

used.

When the input data is on a tripolar grid or a latitude-longitude grid, postprocessing/regrid can be used.

For sponge generation, acceptable input data sets must have NetCDF format with COARDS-compliance.

5.6. Time-related issues in forcing files

There are many ways that data can be formatted in time. The FMS tools used to read in time information, and to time interpolate, are quite sophisticated and flexible. Nonetheless, these tools cannot do everything, nor can they know a priori what the modeler intends. It is therefore necessary to maintain certain conventions when preparing the time information for datasets. This section aims to outline some of the common issues involved with time, and to provide a guide for resolving possible problems.

5.6.1. How it works and what to do if it fails

Previous versions of MOM used IEEE binary formats and MOM-specific headers to process forcing data. As of MOM4, data are stored in portable formats (NetCDF currently), and contain standardized metadata per the CF1.0 [http://www.cgd.ucar.edu/cms/eaton/cf-metadata/] convention.

Understanding the functions of Fortran modules that handle metadata and time-related problems will be very helpful in identifying some user's problems. Some of the most frequently used modules are listed below:

mpp_io_mod	Low level I/O (open, close file, write, read,...)
axis_utils_mod	process metadata: identify cartesian axis information (X/Y/Z/T)
time_manager_mod	basic time operations, calendar, increment/decrement time
time_interp_mod	Computes a weight for linearly interpolating between two dates
time_interp_external_mod	top level routines for requesting data
data_override_mod	top level routines for requesting data

5.6.2. Test your forcing files before use

It is likely that you will encounter an error using "off-the-shelf" NetCDF files to force your ocean model. This could be due to inadequate metadata in the forcing files, mis-specification of the DataTable, or errors in the parsing of the axis information by axis_utils or get_cal_time. You'll need some tools to help you diagnose problems and apply the required fix.

The first thing you should do to setup a new forcing file is use the test program: **time_interp_external_mod:test_time_interp_external**. This test program calls time_interp_external at user-specified model times and returns information on how the times were decoded and the resulting interpolation indices and weights. It is STRONGLY suggested that you pass your forcing files through this program before including them in your model configuration. As you gain familiarity with the metadata requirements, you will more easily be able to identify errors and save a lot of time debugging.

The forcing test program is located in `src/preprocessing/test_time_interp_ext`. There is a csh version and a Perl version.

Compilation

```
mkmf -m Makefile -p test_time_interp_ext.exe -t $TEMPLATE -c -Dtest_time_interp_externa
```

running csh version

```
namelist options:
filename='foo.nc'                       ! name of forcing file
fieldname='foo'                         ! name of variable in file
year0=[integer]                         ! initial year to start model calendar
month0=[integer]                        ! initial month to start model calendar
day0=[integer]                          ! initial day to start model calendar
days_inc=[integer]                      ! increment interval for model calendar
ntime=[integer]                         ! number of model "timesteps"
cal_type=['julian','noleap','360day']  ! model calendar
```

running perl version

```
 test_time_interp_ext.pl -f 'foo.nc' -v 'foo' [--nt [int] --year0 [int] --month0 [int] -
```

Modifying the file metadata should hopefully prove straightforward. The NCO operators [http:/ / nco.sourceforge.net] need to be installed on your platform. The utility "ncatted" is most useful for modifying or adding metadata. If for some reason, you are unable to install the NCO operators, you can use the NetCDF utilities "ncgen" and "ncdump" which come with the NetCDF package.

5.6.3. Common metadata problems

Can't identify cartesian axis information

axis_utils_mod:get_axis_cart should return the cartesian information. If this fails, you will get a somewhat cryptic error message: "file/fieldname could not recognize axis atts in time_interp_external". The best solution is to add the "cartesian_axis" attribute to the axes, e.g. "ncatted -a cartesian_axis,axis_name,c,c,"X" foo.nc".

Calendar attribute does not exist

This is a required attribute. time_manager_mod:get_cal_time converts time units appropriate to the specified calendar to the model time representation. If the "calendar" attribute does not exist, an error message appears "get_cal_time: calendar attribute required. Check your dataset to make sure calendar attribute exists " Use a ncatted command such as: "ncatted -a calendar,time_axis_name,c,c,"julian" foo.nc"

Currently, the FMS time_manager does not support the Gregorian calendar. So, for instance if you have forcing data that are encoded using the Gregorian calendar which has an average year length of 365.2425 days compared with the Julian calendar with an average year length of 365.25 days, assuming Julian calendar encoding will result in a drift of 0.75 days/100 years. If your forcing times are referenced to an early date such as "0001-01-01" your times will drift by 15 days by the year 2000. Until the Gregorian calendar is implemented in the FMS code, the recommended solution is to change the reference date in the forcing dataset using an application such as Ferret

5.7. Scalability of MOM4 code

Scalability of a complex model like MOM4 is the correlation between the number of processing elements (PE) and the run time. One would expect that run time decreases as the number of PEs increases. It is important, however, to note that there are a number of important factors that can affect scalability considerably:

- MOM4 test cases are designed for testing the code integrity, they are not set for scalability study or "production" purpose. Changes should be made if one wants to study scalability of the code.

- diag_freq (in MOM4p0) and diag_step (in MOM4p1) set the time steps at which numerical diagnostics (e.g., energy, tracer budgets) are computed. The user needs to set this value to be equal to the time step at the end of the experiment, so that only a single instance of the diagnostics is evaluated. For example, if time step is 1 hour and run length is 4 days, diag_freq in MOM4p0 (diag_step in MOM4p1) should be set to 96.

- diag_table contains all fields that will be saved in history files. The frequency of saving and number of fields can afect the total run time greatly. So when testing performance, it is recommended that the researcher reduce the output of history files.

- Scalability is also dependent on the configuration of the computing platform: Ethernet card, interconnect between PEs, implementation of MPI, memory hierarchy, version of compiler, ...

- In examining the total run time the overheads due to initialization and termination should be extracted from total runtime for scalability study since they contain a lot of I/O activities.

5.8. Minimizing idle processors

When running an ocean-ice model with atmospheric data forcing, it may happen that some ocean-ice tasks contain only land points, in which case these processors stand idle. FMS provides a feature to mask land-only tasks and to thus run the model without allocating a processor for these land-only domains. For example, if a processor domain layout 30x50 is specified, but say 500 subdomains are land only, you may run the model with 1500-500=1000 MPI tasks only. Note that you do not get your results faster when eliminating land-only processors, but compute costs may be reduced.

In the following example, an "x" represents a land-only subdomain, whereas a dot indicates a domain with a nonzero number of "wet" ocean points:

x . x

x . .

. . .

In this example, three idle processors could be left out (two in the northern-most row, and one in the second row) since there are no wet ocean points on these "x" domains. The ocean-ice model may thus be run with only 6 MPI tasks rather than nine. To enable this feature of FMS, the following variables in the namelist coupler_nml must be specified.

coupler_nml

...

do_atmos = .false.

....

n_mask = 3

layout_mask = 3, 3

mask_list = 1,2, 1,3, 3,3

If a mask list is specified, the domain layout of the ocean and the ice model is automatically the same, as specified with layout_mask. Running the model, only 6 MPI tasks need to be started.

For large and complex model topography, it is very tedious to specify the mask list by hand. To simplify this

task, the preprocessing tool "check_mask" can be used after the model grid and topography are specified, i.e., when grid_spec.nc is generated. To do so requires adding the following to topog_nml during the preprocessing stage

topog_nml

check_mask=.true.

The preprocessing code prints out a number of proposals for possible domain layouts, number of land-only sub-domains, and templates for setting the mask_list. You may copy these layouts to coupler_nml, but note that you must replace spaces by a comma in "mask_list".

You may configure "check_mask" from the namelist check_mask_nml. There are two variables "max_pe" and "halo" required to perform the mask checking. The default settings are

max_pe = 128 halo = 1

If you intend to employ more processors, then increase max_pe. If you want to use higher order advection schemes, then increase halo to halo=2 or even halo=3. Otherwise, the halo information of masked domains may leak into ocean domains, in which case the model will fail, with errors reporting land values at ocean points.

In the postprocessing step, the single netCDF-output files from the MPI tasks can be combined into one file using the tool mppnccombine [#mppnccombine]. The latest version adds "missing" values at the location of land-only domains. Hence, the only remaining trace of masking out land-only processors is in your account files, where some compute time is saved.

6. Postprocessing regrid tool

6.1. Introduction to Postprocessing

For many analysis applications, it is sufficient, and often preferable, to have output on the model's native grid (i.e., the grid used to run the simulation). Accurate computation of budgets, for example, must be done on the model's native grid, preferably online during integration. MOM4 provides numerous online diagnostics for this purpose.

Many applications, such as model comparison projects, require results on a common latitude-longitude spherical grid. Such facilitates the development of difference maps. For this purpose, we have developed a tool to regrid scalar and vector fields from a tripolar grid to a spherical grid. In principle, this tool can be used to regrid any logically rectangular gridded field onto a spherical grid. However, applications at GFDL have been limited to the tripolar to spherical regrid case.

In general, regridding is a difficult task to perform accurately and without producing noise or spurious results. The user should carefully examine regridding results for their physical integrity. Problems occur, in particular, with fields near MOM4's partial bottom step topography in the presence of realistic topography and land/sea geometry. Indeed, we were unable to find a simple algorithm to handle regridding in the vertical that did not produce egregious levels of noise. Hence, the regridding tool provided with MOM4 only handles horizontal regridding. The regridded data will thus be on the source vertical grid.

Model comparisons should ideally be performed only after regridding output using the same regridding algorithm. Unfortunately, such is not generally the case since there is no standard regridding algorithm used in the modeling community.

6.2. How to use the regridding tool

The regridding algorithm provided with the MOM4 distribution is located in the directory **postprocessing/regrid**

The algorithm accepts data from any logically rectangular grid (e.g., tripolar or latitude-longitude) and regrids to a spherical latitude-longitude grid. When the data is on the tracer cell (T-cell), the regridding interpolation is conservative. Thus, total heat, salt, and passive tracer remain the same on the two grids. However, when data is located at another position:

- corner or C-cell as for a B-grid horizontal velocity component
- east or E-cell as for an eastward tracer flux
- north or N-cell as for a northward tracer flux

then regridding is accomplished non-conservatively using a nearest neighbor distance weighting algorithm. It is for this reason that computationally accurate results are only available when working on the model's native grids.

The regridding tool reads grids information from a netcdf file, specified by the namelist "grid_spec_file". "grid_spec_file" contains source grid, destination grid and exchange grid information.

- source grid: src_grid.nc. This is the model's native grid. It results from running preprocessing grid generation code.
- destination grid: dst_grid.nc. This is the spherical latitude-lontitude grid. This grid can also be obtained from running preprocessing grid generation code. Be sure that the tripolar option is set to false to ensure that dst_grid.nc is spherical.
- exchange grid: grid_spec.nc. This is the union of the source grid and destination grid. The exchange grid is needed for conservative regridding. The same conservative regridding algorithm is used for coupled models with FMS. The tool to construct the exchange grid is know as "make_xgrids". It is located in the preprocessing directory. After grid_spec.nc is generated, it should be passed to the regrid tool through namelist "grid_spec_file"(No need to pass src_grid.nc and dst_grid.nc to the regrid tool).

To create the exchange grid, execute the command

make_xgrids -o src_grid.nc -a dst_grid.nc

The exchange grid creates a file grid_spec.nc. It has new fields with names:

AREA_ATMxOCN, DI_ATMxOCN, DJ_ATMxOCN, I_ATM_ATMxOCN, J_ATM_ATMxOCN, I_OCN_ATMxOCN, J_OCN_ATMxOCN, AREA_ATMxLND, DI_ATMxLND, DJ_ATMxLND, I_ATM_ATMxLND, J_ATM_ATMxLND, I_LND_ATMxLND, J_LND_ATMxLND, AREA_LNDxOCN, DI_LNDxOCN, DJ_LNDxOCN, I_LND_LNDxOCN, J_LND_LNDxOCN, I_OCN_LNDxOCN, J_OCN_LNDxOCN, xba, yba, xta, yta, AREA_ATM, xbl, ybl, xtl, ytl, AREA_LND, AREA_LND_CELL, xto, yto, AREA_OCN

It is critical that src_grid.nc DO NOT already have any of the above new exchange grid fields. If they do, then these fields should be removed using netcdf tools [http://nco.sourceforge.net/] such as ncks.

After the grid_spec.nc file is generated, it is passed into the regrid program through the nml option "grid_spec_file".

The regrid program reads model data from a netcdf file, which is specfied by the namelist variable "src_data". Again, src_data fields are gridded according to src_grid.nc. The number of fields to be regridded is specified by num_flds. The name of the fields (e.g., temp, salt) to be regridded is specified by the namelist variable "fld_name". Each field can be a scalar or vector. If a vector, then specify by vector_fld. Vector fields should always be paired together (e.g., u,v components to the horizontal current). The output file is a netcdf file specified by the namelist variable "dst_data".

The complete namelist option description is available in regrid.html or the code itself.

7. Preparing the runscript

7.1. The runscript

A runscript is provided in each test case [#MOM4 test cases] directory (`mom4/exp/$test_case`) for each test case [#MOM4 test cases]. Details can be found in quickstart_guide.html.

Incorporated in the FMS infrastructure is `MPP` [.. / .. / shared/ mpp/ mpp.html] (Massively Parallel Processing), which provides a uniform message-passing API interface to the different message-passing libraries. If MPICH is installed, the user can compile the MOM4 source code with MPI. If the user does not have MPICH or the communications library, the MOM4 source code can be compiled without MPI by omitting the `CPPFLAGS` value –`Duse_libMPI` in the example runscript.

7.2. The diagnostics table

The diagnostics table allows users to specify the sampling rates and choose the output fields prior to executing the MOM4 source code. It is included in the `input` directory for each test case [#mom4 test cases] (`mom4/exp/$test_case/input`). A portion of a sample MOM4 diagnostic table is displayed below. Reference diag_manager.html [.. / .. / shared/ diag_manager/ diag_manager.html] for detailed information on the use of diag_manager.

```
"Diagnostics for MOM4 test case"
1980 1 1 0 0 0
#output files
"ocean_month",1,"months",1,"hours","Time"
"ocean_snap",1,"days",1,"hours","Time"
#####diagnostic field entries####
#=================================================================
# ocean model grid quantities (static fields and so not time averaged))
"ocean_model","geolon_t","geolon_t","ocean_month" "all",.false.,"none",2
"ocean_model","geolat_t","geolat_t","ocean_month","all",.false.,"none",2
#=================================================================
# prognostic fields
"ocean_model","temp","temp","ocean_month","all", "max", "none",2
"ocean_model","age_global","age_global","ocean_month","all","min","none",2
#=================================================================
# diagnosing tracer transport
"ocean_model","temp_xflux_sigma","temp_xflux_sigma","ocean_month","all",.true.,"none
"ocean_model","temp_yflux_sigma","temp_yflux_sigma","ocean_month","all",.true.,"none
#=================================================================
# surface forcing
"ocean_model","sfc_hflux","sfc_hflux","ocean_month","all",.true.,"none",2
"ocean_model","sfc_hflux_adj","sfc_hflux_adj","ocean_month","all",.true.,"none",2
#=================================================================
# ice model fields
"ice_model", "FRAZIL",   "FRAZIL",      "ice_month", "all", .true., "none", 2,
"ice_model", "HI",    "HI",   "ice_month", "all", .true., "none", 2
#-----------------------------------------------------------------
```

The diagnostics manager module, `diag_manager_mod` [.. / .. / shared/ diag_manager/ diag_manager.html], is a set of simple calls for parallel diagnostics on distributed systems. It provides a convenient set of interfaces for writing data to disk in NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format. The diagnostics manager is packaged with the MOM4 source code. The FMS diagnostic manager can handle scalar fields as well as arrays. For more information on the diagnostics manager, reference diag_manager.html [.. / .. / shared/ diag_manager/ diag_manager.html].

7.3. The field table

The MOM4 field table is used to specify tracers and their advection schemes, cross-land tracer mixing, cross-land insertion, and other options. The field table is included in the runscript as a namelist and is written to an output file upon execution of the runscript.

```
"prog_tracers","ocean_mod","temp"

horizontal-advection-scheme = quicker
vertical-advection-scheme = quicker
restart_file  = ocean_temp_salt.res.nc
/




"prog_tracers","ocean_mod","salt"

horizontal-advection-scheme = mdfl_sweby
vertical-advection-scheme = mdfl_sweby
restart_file  = ocean_temp_salt.res.nc
/




"tracer_packages","ocean_mod","ocean_age_tracer"

names = global
horizontal-advection-scheme = mdfl_sweby
vertical-advection-scheme = mdfl_sweby
restart_file  = ocean_age.res.nc
min_tracer_limit=0.0
/




"namelists","ocean_mod","ocean_age_tracer/global"

slat = -90.0
nlat =  90.0
wlon =   0.0
elon = 360.0
/




"xland_mix","ocean_mod","xland_mix"
"xland","Gibraltar","ixland_1=274,ixland_2=276,jxland_1=146,jxland_2=146,kxland_1=1,kxla
"xland","Gibraltar","ixland_1=274,ixland_2=276,jxland_1=147,jxland_2=147,kxland_1=1,kxla
"xland","Black-Med","ixland_1=305,ixland_2=309,jxland_1=151,jxland_2=152,kxland_1=1,kxla
"xland","Black-Med","ixland_1=306,ixland_2=309,jxland_1=151,jxland_2=153,kxland_1=1,kxla

"xland_insert","ocean_mod","xland_insert"
"xland","Gibraltar","ixland_1=274,ixland_2=276,jxland_1=146,jxland_2=146,kxland_1=1,kxla
"xland","Gibraltar","ixland_1=274,ixland_2=276,jxland_1=147,jxland_2=147,kxland_1=1,kxla
"xland","Black-Med","ixland_1=305,ixland_2=309,jxland_1=151,jxland_2=152,kxland_1=1,kxla
"xland","Black-Med","ixland_1=306,ixland_2=309,jxland_1=151,jxland_2=153,kxland_1=1,kxla
```

```
"diff_cbt_enhance","ocean_mod","diff_cbt_enhance"
"diffcbt","Gibraltar","itable=274,jtable=146,ktable_1=1,ktable_2=18,diff_cbt_table=0.01"
"diffcbt","Gibraltar","itable=276,jtable=146,ktable_1=1,ktable_2=18,diff_cbt_table=0.01"
"diffcbt","Gibraltar","itable=274,jtable=147,ktable_1=1,ktable_2=18,diff_cbt_table=0.01"
"diffcbt","Gibraltar","itable=276,jtable=147,ktable_1=1,ktable_2=18,diff_cbt_table=0.01"
"diffcbt","Black-Med","itable=305,jtable=151,ktable_1=1,ktable_2=6,diff_cbt_table=0.01"
"diffcbt","Black-Med","itable=309,jtable=152,ktable_1=1,ktable_2=6,diff_cbt_table=0.01"
"diffcbt","Black-Med","itable=306,jtable=151,ktable_1=1,ktable_2=6,diff_cbt_table=0.01"
"diffcbt","Black-Med","itable=309,jtable=153,ktable_1=1,ktable_2=6,diff_cbt_table=0.01"/
```

In the first section of the field table, the user can specify tracers to be used in the simulation. Although there is no limit to the number of tracers specified, temperature (temp) and salinity (salt) must be included. The user may also define the horizontal and vertical tracer advection schemes. For more information on the field manager, reference field_manager.html [../../shared/field_manager/field_manager.html].

In climate modeling, it is often necessary to allow water masses that are separated by land to exchange tracer and surface height properties. This situation arises in models when the grid mesh is too coarse to resolve narrow passageways that in reality provide crucial connections between water masses. The cross-land mixing and cross-land insertion establishes communication between bodies of water separated by land. The communication consists of mixing tracers and volume between non-adjacent water columns. Momentum is not mixed. The scheme conserves total tracer content, total volume, and maintains compatibility between the tracer and volume budgets. The grid points where this exchange takes place, and the rates of the exchange, are specified in the field table.

For some cases, it is necessary to set a large vertical tracer diffusivity at a specified point in the model, say next to a river mouth to ensure fresh water is mixed vertically. These diffusivities are specified in the field table.

For a technical description of cross-land tracer mixing and insertion, please reference A Technical Guide to MOM4 [./guide4p0.pdf].

7.4. mppnccombine

Running the MOM4 source code in a parallel processing environment will produce one output NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] diagnostic file per processor. mppnccombine [../../postprocessing/mppnccombine.c] joins together an arbitrary number of data files containing chunks of a decomposed domain into a unified NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] file. If the user is running the source code on one processor, the domain is not decomposed and there is only one data file. mppnccombine [../../postprocessing/mppnccombine.c] will still copy the full contents of the data file, but this is inefficient and mppnccombine [../../postprocessing/mppnccombine.c] should not be used in this case. Executing mppnccombine [../../postprocessing/mppnccombine.c] is automated through the runscripts [#runscript]. The data files are NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format for now, but IEEE binary may be supported in the future.

mppnccombine [../../postprocessing/mppnccombine.c] requires decomposed dimensions in each file to have a domain_decomposition attribute. This attribute contains four integer values: starting value of the entire non-decomposed dimension range (usually 1), ending value of the entire non-decomposed dimension range, starting value of the current chunk's dimension range and ending value of the current chunk's dimension range. mppnccombine also requires that each file have a NumFilesInSet global attribute which contains a single integer value representing the total number of chunks (i.e., files) to combine.

The syntax of mppnccombine [../../postprocessing/mppnccombine.c] is:

```
mppnccombine [-v] [-a] [-r] output.nc [input ...]
```

Table 1. mppnccombine arguments

-v	print some progress information
-a	append to an existing NetCDF [http:/ / www.unidata.ucar.edu/packages/netcdf/] file
-r	remove the '.####' decomposed files after a successful\ run

An output file must be specified and it is assumed to be the first filename argument. If the output file already exists, then it will not be modified unless the option is chosen to append to it. If no input files are specified, their names will be based on the name of the output file plus the extensions '.0000', '.0001', etc. If input files are specified, they are assumed to be absolute filenames. A value of 0 is returned if execution is completed successfully and a value of 1 indicates otherwise.

The source of mppnccombine [../../postprocessing/mppnccombine.c] is packaged with the MOM4 module in the postprocessing directory. mppnccombine.c should be compiled on the platform where the user intends to run the FMS MOM4 source code so the runscript [#runscript] can call it. A C compiler and NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] library are required for compiling mppnccombine.c:

```
cc -O -o mppnccombine -I/usr/local/include -L/usr/local/lib mppnccombine.c -lnetc
```

8. Examining the output

8.1. Sample model output

Sample MOM4p1 model output data files are available at GFDL ftp site [ftp://ftp.gfdl.noaa.gov/perm/MOM4/mom4p1_pubrel/exp]. Output files are classified into three subdirectories:

- ascii: the description of the setup of the run and verbose comments printed out during the run.

- restart: the model fields necessary to initialize future runs of the model.

- history: output of the model, both averaged over specified time intervals and snapshots.

Note that these output files are compressed using **tar**. All .tar files should be decompressed for viewing. The decompress command is:

```
tar -xvf filename.tar
```

8.2. Analysis tools

There are several graphical packages available to display the model output. These packages vary widely depend-

ing on factors, such as the number of dimensions, the amount and complexity of options available and the output data format. The data will first have to be put into a common format that all the packages can read. FMS requires the data to be stored in NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format since it is so widely supported for scientific visualization. The graphical package is also dependent upon the computing environment. For ocean modeling, ncview [http:/ / meteora.ucsd.edu/ ~pierce/ ncview_home_page.html], Ferret [http:/ / ferret.wrc.noaa.gov/Ferret/] and GrADS [http://grads.iges.org/grads/] are most commonly used.