The Max-Planck-Institute Global Ocean/Sea-Ice Model MPI-OM

Patrick Wetzel, Helmuth Haak, Johann Jungclaus and Ernst Maier-Reimer

Max-Planck-Institute for Meteorology
Bundestrasse 53
20146 Hamburg, Germany

CONTENTS

1. INTRODUCTION

The Hamburg Ocean Model, MPI-OM is the successor of the Hamburg Ocean Primitive Equation (HOPE) model and has undergone significant development in recent years. Most notable is the treatment of horizontal discretization which has undergone transition from a staggered E-grid to an orthogonal curvilinear C-grid. The treatment of subgridscale mixing has been improved by the inclusion of a new formulation of bottom boundary layer (BBL) slope convection, an isopycnal diffusion scheme, and a Gent and McWilliams style eddy-induced mixing parameterization.

The Hamburg Ocean Model, MPI-OM, is an ocean general circulation model (OGCM) based on the primitive equations with representation of thermodynamic processes. It is capable of simulating the oceanic circulation from small scales (oceanic eddies) to gyre scales, in response to atmospheric forcing fields. For an application on horizontal scales smaller than about 1 km the hydrostatic assumption is no longer valid and the model must in parts be reformulated. The use of an ocean circulation model requires a comprehensive understanding of the ocean physics and the numerical formulation. It is not recommended to use this model as a black box. Many physical processes in the ocean are still not very well understood and are therefore only crudely parameterized. Each new application requires a new consideration of how to specify model parameters, especially coefficients for eddy viscosity and diffusivity. The model is thought to be a framework into which new ideas concerning parameterizations or forcing mechanisms might easily be incorporated. This manual gives a description of the MPI-OM model, in order to help potential users to run the model and to acquire a understanding of the model physics and numerics.

This manual refers to release_1_1 of MPI-OM.

2. USING MPI-OM

This chapter shows how to set up and run the model with a given configuration and resolution. It will also deal with compiling and running the model on different computational platforms.

2.1 Input Files

2.1.1 Grid Files

Grid definition files store information about the model grid and bathymetry. To generate a well working setup needs considerable expertise and requires extensive evaluation. Most users work with exiting, thoroughly tested, model setups. The tools to generated a specific setup are described in chapter 3.1. File formates are described in 8.3 The following grid definition files are needed for MPI-OM:

1. **anta**
   Geographical positions (longitudes and latitudes) of the center and of the edges of the grid cells. Format is EXTRA.

2. **topo**
   Model topography. Format is ASCII.

3. **arcgri**
   Grid point separation. Format is EXTRA.

4. **BEK**
   ASCII file which is used for online diagnostics. It stores the masks for the major ocean basins using a number-code from 0 to 9.

5. **SURSAL and SURTEM**

   Climatological values of surface salinity and temperature from Steele *et al.* (2001), interpolated onto the model grid. Surface salinity and temperature can be restored with a constant relaxation time which is set in the namelist (Table 2.1). Format is EXTRA. Alternatively, a global hydrography from Gouretski and Koltermann (2004) is available.

6. **INITEM and INISAL** Three dimensional data of potential temperature and salinity (Steele *et al.*, 2001), interpolated onto the model grid Starting value for the initialization of MPI-OM. Format is EXTRA. Alternatively, a global hydrography from Gouretski and Koltermann (2004) is available.

7. **runoff_obs**
   Monthly mean river discharge data (currently for 53 positions) in EXTRA format.

8. **runoff_pos**
   Longitudes and latitudes of the river discharge positions in EXTRA format.

Alternatively to runoff_obs and runoff_pos, river discharge information can be in supplied as a forcing field. Please see section 2.1.3 and table 2.5.

2.1.2 Restart File

The current status of the model at the end of a run is stored in the ocean restart file Z37000 (IEEE 8byte EXTRA Format). Parameter ISTART in the namelist 2.1 defines if a run is started from a climatological distribution of temperature and salinity (INISAL and INITEM, see section 2.1 and 3.2) of from an existing restart file. The file contains all scalar and vector variables for the ocean and the sea-ice model.

2.1.3 Forcing Fields

Forcing data sets are provided as daily mean fields. The model is forced with heat, freshwater and momentum fluxes, interpolated onto the model grid. Popular is, for example, the climatological 360 day forcing compiled by the OMIP project. or the forcing generated from the NCEP/NCAR reanalysis (Kalnay, 1996), which starts in 1948. The daily 2 m air and dew point temperatures, precipitation, cloud cover, 10 m wind speed and surface wind stress are taken without modification. Dew point temperature T_{Dew} is derived from specific humidity q and air pressure p according to Oberhuber (1988). Format of the forcing fields is EXTRA. If you are interested in compiling your own forcing, please see section 3.3

1. **GICLOUD**
   Total cloud cover

2. **GIPREC**
   Precipitation

3. **GISWRAD**
   Solar radiation

4. **GITDEW**
   Dew point temperature

5. **GITEM**
   Surface temperature

6. **GIU10**
   10 m wind speed

7. **GIWIX**
   zonal (along index i) wind stress

8. **GIWIY** meridional (along index j) wind stress

9. **GIRIV** river discharge data (optional)

2.1.4 Namelist

Some important tuning parameters can be set in the namelist OCECTL (ASCII file, table 2.1). The depth levels are defined in DZW; NPROCS defines the number of processors assigned for the MPI parallelization (example in section 2.4).

2.2 Output Fields

MPI-OM generates a large number of output files. Most of them are mean values of ocean properties (temperate, salinity ...). In addition, there is output for mean diagnostic and flux variables, as well as grid, forcing or coupling (ECAHM) information. Time averaging can be daily, monthy or yearly and is controlled by the namelist (see 2.1) variable `IMEAN` (table 2.1). Output data is selected with CPP switches (see 2.6.1). Each code is written into a separate file named `fort.`*`unit`* according to the unit the file is written to. The file format is `EXTRA`. At the end of each run a tar file is created from the averaged data files. Mean and diagnostic output is discussed in detail in chapter 7 ”Diagnostic and Mean Output”. Tables 7.2 to 7.7 given an overview on all available output codes.

Parameter	GR03	Comment	Reference
DT	8640.	model time step in seconds	5.2
CAULAPTS	0.	horizontal biharmonic diffusion coefficient for tracers (T,S)	5.2.17
CAULAPUV	0.005	horizontal biharmonic diffusion coefficient for momentum (u,v)	5.2.20
CAH00	1000.	horizontal harmonic diffusion coefficient (isopycnic and GM)	5.2.16, 5.2.17, 5.2.19
DV0	1.E-2	vertical diffusion coefficient for tracers (T,S)	5.2.17
AV0	1.E-2	vertical diffusion coefficient for momentum (u,v)	5.2.20
DBACK	1.E-5	minimum setting for vertical tracers (T,S) diffusion	5.2.17
ABACK	1.E-4	minimum setting for vertical momentum (u,v) diffusion	5.2.20
CWT	5.E-4	wind mixing coefficient	4.17, 5.2.1
CSTABEPS	0.030	minimum wind mixing	5.2.1
CRELSAL	2.0E-7	surface salinity relaxation	4.8, 4.32
CRELTEM	0.	surface temperature relaxation	4.8
CDVOCON	0.05	convection coefficient for tracers (T,S)	5.2.1
CAVOCON	0.0	convection coefficient for momentum (u,v)	5.2.1
NYEARS	0	Number of simulated years in a run	
NMONTS	1	Number of simulated months in a run	
IMEAN	2	Time averaging of output data (3: yearly means, 2: monthly means, 1: daily means)	chapter 7
ISTART	2	Start from initial conditions [0/1], standard [2]	
I3DREST	1	3-D restoring of temperature and salinity to initial conditions	5.2.18

Tab. 2.1: Parameters of the MPI-OM namelist OCECTL, numbers are examples for a GR03 setup.

2.3 Timeseries

A ASCII file named TIMESER containing time series data for some diagnostic variables is generated in each model time step. The content is described in detail in routine `DIAGNOSIS.F90` of the source code. Usually, the TIMESER files from each year are "cat" together to a file named ZEITSER. A Fortran77 program named "plsteig.f" is available which generates a series of plots from one ore many ZEITSER files for evaluation and comparison.

2.4 Quickstart Examples

On a PC or Sun workstation, the most simple way to run the model is to put a precompiled MPI-OM executable and all necessary input and forcing files into one directory and start the model executable manually. In this case the model runs only on one processor and there the executable should have be compiled without parallelization options. If you need to compile your own executable, please refer to section 2.6, for more details on parallelization (running on more than one processor), please see section 2.5.

2.4.1 Run Script

In most cases one wants to run several consecutive years, either by repeating a climatological forcing, or by using different forcing fields for each year. In this case a run script takes care of the right forcing, the right parameters in the namelist and the proper labeling of the output.

If you have received the model from CD or the ZMAW CVS server (see 2.6) you will find a shell-script called "prepare_run_mpiom_omip" which helps to set up a runtime environment. It will create directories in the appropriate places and set links to the necessary input and forcing files. It will also create run-script which can serve as an example for your actual script. The script is written for the MPI/DKRZ environment (Hurrikan) and has to be modified to work on other platforms.

Linux and SUN

An example for a LINUX run-script for one processor, GR30 resolution and climatological OMIP forcing is given below. All necessary files are in one folder and the same climatological forcing is used for all years. It is important to note, that binary input and forcing files are double precision with big endian. The same script can be used on a SUN workstation. Sun uses big endian for binary files by default, so no extra settings are necessary.

```
#!/bin/sh
set verbose
set echo
#
# This script is for a simple runtime setup.
# with one processor on a Linux PC.
# All files are stored in one directory named GR30_$ID
# Script is designed for a 20 layer GR30 grid.
#
#---------------------------------------------------------
#

HOME=/home/patrick ; export HOME
ID=OMIP ; export ID
# for intel ifc compiler
F_UFMTENDIAN=big ; export F_UFMTENDIAN

#
# at the beginning
#
echo 1 > year.asc

#
# Execute 3 times
#
for number in 1 2 3
do

cat > OCECTL  << EOF
 &NPROCS
 nprocx=1
 nprocy=1
  /
 &OCECTL
 DT      = 8640.
 CAULAPTS= 0.
 CAULAPUV= 0.005
 AUS     = 0.
 CAH00   = 1000.
 DV0     = 1.E-2
 AV0     = 1.E-2
 CWT     = 5.E-4
 CSTABEPS= 0.030
 DBACK   = 1.E-5
 ABACK   = 1.E-4
 CRELSAL = 2.0E-7
 CRELTEM = 0.
 CDVOCON = 0.05
 CAVOCON = 0.0
 NYEARS  = 1
 NMONTS  = 0
 IMEAN   = 2
 &END
 &OCEDZW
```

```
  DZW = 20.,20., 20., 30.,40.,50.,70.
        ,90.,120.,150.,180.,210.,250.,300.
        ,400.,500.,600.,700.,900.,1400.
 &END
EOF

set -e
#for fujitsu lf95 compiler (set read/write to big endian)
#ocmod.x -Wl,-T

#for intel ifc compiler
./ocmod.x
set +e

#
# Append the timeseries to ZEITSER
#

cat TIMESER >> ZEITSER
\rm TIMESER

#
# Append the timeseries to ZEITSER
#

\cp Z37000 Z37000_$YEAR
\cp Z37000 Z38000

#
# Rename the output files you want to keep ....
#

ls -al
mv fort.71 $ID\_tho.ext4
mv fort.72 $ID\_sao.ext4
mv fort.73 $ID\_uko.ext4
mv fort.74 $ID\_vke.ext4
mv fort.79 $ID\_eminpo.ext4
mv fort.82 $ID\_zo.ext4
mv fort.84 $ID\_flum.ext4
mv fort.85 $ID\_pem.ext4
mv fort.86 $ID\_sictho.ext4
mv fort.87 $ID\_sicomo.ext4
mv fort.88 $ID\_sicuo.ext4
mv fort.89 $ID\_sicve.ext4
mv fort.90 $ID\_kcondep.ext4
mv fort.130 $ID\_wu10.ext4
mv fort.131 $ID\_tafo.ext4
mv fort.132 $ID\_fclo.ext4
mv fort.133 $ID\_fpre.ext4
mv fort.134 $ID\_fswr.ext4
mv fort.135 $ID\_ftde.ext4
mv fort.136 $ID\_sicsno.ext4
mv fort.137 $ID\_qswo.ext4
mv fort.138 $ID\_qlwo.ext4
mv fort.139 $ID\_qlao.ext4
mv fort.140 $ID\_qseo.ext4
mv fort.141 $ID\_preco.ext4
mv fort.142 $ID\_amld.ext4
mv fort.143 $ID\_psiuwe.ext4
mv fort.144 $ID\_avo.ext4
mv fort.145 $ID\_dvo.ext4
mv fort.146 $ID\_wo.ext4
mv fort.147 $ID\_sictru.ext4
mv fort.148 $ID\_sictrv.ext4
mv fort.149 $ID\_txo.ext4
mv fort.150 $ID\_tye.ext4
mv fort.156 $ID\_zmld.ext4
mv fort.93 $ID\_weto.ext4
mv fort.94 $ID\_gila.ext4
mv fort.97 $ID\_giph.ext4
mv fort.96 $ID\_depto.ext4
mv fort.151 $ID\_dlxp.ext4
mv fort.152 $ID\_dlyp.ext4
mv fort.153 $ID\_deuto.ext4
mv fort.154 $ID\_dlxu.ext4
mv fort.155 $ID\_dlyu.ext4
mv fort.245 $ID\_wtmix.ext4
mv fort.246 $ID\_wgo.ext4
mv fort.159 $ID\_bolx.ext4
mv fort.160 $ID\_boly.ext4
mv fort.247 $ID\_dqswo.ext4
mv fort.248 $ID\_dqlwo.ext4
mv fort.249 $ID\_dqseo.ext4
mv fort.250 $ID\_dqlao.ext4
mv fort.251 $ID\_dqtho.ext4
mv fort.252 $ID\_dqswi.ext4
mv fort.253 $ID\_dqlwi.ext4
mv fort.254 $ID\_dqsei.ext4
mv fort.255 $ID\_dqlai.ext4
mv fort.256 $ID\_dqthi.ext4
mv fort.257 $ID\_dticeo.ext4
mv fort.157 $ID\_tmceo.ext4
mv fort.158 $ID\_tmcdo.ext4
mv fort.303 $ID\_ukomfl.ext4
mv fort.304 $ID\_vkemfl.ext4
mv fort.305 $ID\_rivrun.ext4

#
# tar and move the output and the restart files
#

tar cvf $YEAR.tar Z37000_$YEAR ZEITSER *.ext4
\mv $YEAR.tar $HOME/OUTPUT/GR30_$ID

#
# .... and delete the others.
#
```

```
rm *.ext4 Z37000_$YEAR

#
# add one to the year-counter
#

YEAR=‘expr ${YEAR} + 1‘
echo $YEAR > year.asc

done
```

NEC SX-6 (DKRZ-Hurrikan)

If run on a supercomputer such as the SX-6 of DKRZ (Hurrikan), a special script with the right queuing parameters and environment variables is required. The shell-script ”prepare_run_mpiom_omip” helps to set up a runtime environment on Hurrikan. It will create directories in the appropriate places and set links to the necessary input and forcing files in the Hurrikan ”/pool” directory . It will also create run-script which can serve as a simple example for your actual script. In this example it is assumed that the MPI-OM executable has been compiled only with OpenMP parallelization or without any parallelization. The MPI version is commented out. Please keep in mind that, besides for testing, using only one processor (serial queue) does not make much sense on a supercomputer. Identical to the LINUX/SUN example all necessary files are stored in one directory which in this case is on the SHR directory tree on DKRZ Hurrikan. Output files are tared and moved to the UT directory tree. The DKRZ web-side (www.dkrz.de) also has up-to-date examples on the queues and procedures for all kinds of applications. For more details on parallelization and compiling, please see section 2.5 and 2.6.

```
#! /bin/ksh
#-----------------------------------------------------------------------------
# # 1 CPU (maximum number of CPUs 8)
 # 2 h cputime
 # 1 Gbyte memory
 # job runs on 1 node
 # join err and out to out
 # write output to file "GR30_OMIP.rep"
 # job name
 # you should always specify your email
 # address for error messages etc

#PBS -S /bin/ksh
#
#PBS -N mpiom_omip   # job name
#
```

```
#PBS -l cpunum_prc=1          # 1 CPU (maximum number of CPUs 8)
#PBS -l cputim_job=02:00:00   # 2 h realtime per node
#PBS -l memsz_job=1.0gb       # 1 Gbyte memory (48 GB Memory per node)
#PBS -j o                     # join err and out to out
#
#export MPIPROGINF=ALL_DETAIL
export F_PROGINF=DETAIL
export F_SETBUF=4096

#MPIEXPORT="OMP_NUM_THREADS F_FILEINF" ; export MPIEXPORT
#MPIMULTITASKMIX=ON ; export MPIMULTITASKMIX
OMP_NUM_THREADS=1 ; export OMP_NUM_THREADS

#
#-----------------------------------------------------------------------------
#
#                  Job file to run MPI-OM with OMIP forcing
#
#-----------------------------------------------------------------------------
#
# If a command has a non-zero exit status, execute ERR trap, if set, and exit
#
#set -ex
#
#=============================================================================

expno=test
echo "Experiment: ${expno}"
ncpus=1
nprocx=1
nprocy=1
#
echo "   CPUs: ${ncpus} (nprocx: ${nprocx}, nprocy: ${nprocy})"
#
#-----------------------------------------------------------------------------
#
EXPDIR=/shr/2/m212047/${expno}

# absolute path to directory with plenty of space:
ARCHIVE=/ut/m/m212047/EXPERIMENTS/${expno}

# absolute path to directory with initial data:
INITIAL_DATA=/pool/SX-6/MPI-OM

# horizontal and vertical resolution
GRID=GR30
LEV=L40
#

#-----------------------------------------------------------------------------
#
cd ${EXPDIR}           #  output and rerun files are written into $ARCHIVE
pwd
#-----------------------------------------------------------------------------
```

```
ln -s ${INITIAL_DATA}/${GRID}/${GRID}_arcgri             arcgri
ln -s ${INITIAL_DATA}/${GRID}/${GRID}_topo               topo
ln -s ${INITIAL_DATA}/${GRID}/${GRID}_anta               anta
cp ${INITIAL_DATA}/${GRID}/${GRID}_BEK                   BEK
chmod 755 BEK

ln -s ${INITIAL_DATA}/${GRID}/${GRID}_GIWIX_OMIP365      GIWIX
ln -s ${INITIAL_DATA}/${GRID}/${GRID}_GIWIY_OMIP365      GIWIY
ln -s ${INITIAL_DATA}/${GRID}/${GRID}_GITEM_OMIP365      GITEM
ln -s ${INITIAL_DATA}/${GRID}/${GRID}_GIPREC_OMIP365     GIPREC
ln -s ${INITIAL_DATA}/${GRID}/${GRID}_GISWRAD_OMIP365    GISWRAD
ln -s ${INITIAL_DATA}/${GRID}/${GRID}_GITDEW_OMIP365     GITDEW
ln -s ${INITIAL_DATA}/${GRID}/${GRID}_GIU10_OMIP365      GIU10
ln -s ${INITIAL_DATA}/${GRID}/${GRID}_GICLOUD_OMIP365    GICLOUD

ln -s ${INITIAL_DATA}/${GRID}/${GRID}${LEV}_INITEM_PHC  INITEM
ln -s ${INITIAL_DATA}/${GRID}/${GRID}${LEV}_INISAL_PHC  INISAL
ln -s ${INITIAL_DATA}/${GRID}/${GRID}${LEV}_SURSAL_PHC  SURSAL

ln -s ${INITIAL_DATA}/runoff_obs                         runoff_obs
ln -s ${INITIAL_DATA}/runoff_pos                         runoff_pos
#
#-----------------------------------------------------------------------------

for lll in 1 2 3 4 5 ; do

YEAR=‘cat year.asc‘ ; export YEAR
echo $YEAR
STA=3 ; export STA
RES=0 ; export RES

if [ ${YEAR} -eq 0 ] ; then
STA=2 ; export STA
RES=1 ; export RES
fi

cat > OCECTL  << EOF
&nprocs
 nprocx=${nprocx}
 nprocy=${nprocy}
/
 &ocectl
 dt      = 8640.
 caulapts= 0.
 caulapuv= 0.005
 aus     = 0.
 cah00   = 1000.
 dv0     = 1.e-2
 av0     = 1.e-2
 cwt     = 5.e-4
 cstabeps= 0.03
 dback   = 1.e-5
 aback   = 1.e-4
 crelsal = 5.e-8
 creltem = 0.
 cdvocon = 0.05
 cavocon = 0.0
 nyears  = 1
 nmonts  = 0
 imean   = 2
 istart  = ${STA}
 i3drest = ${RES}
 /
 &ocedzw
 dzw     = 12.,10.,10.,10.,10.,10.,13.,15.,20.,25.,
           30.,35.,40.,45.,50.,55.,60.,70.,80.,90.,
           100.,110.,120.,130.,140.,150.,170.,180.,190.,200.,
           220.,250.,270.,300.,350.,400.,450.,500.,500.,600./
 /
EOF

#-----------------------------------------------------------------------------
#mprun -np ${ncpus} mpiom.x
#
mpiom.x
#
#=============================================================================

cat TIMESER >>ZEITSER ; \rm TIMESER
cp Z37000 Z38000

if [ ‘expr $YEAR % 10‘==10 ]; then
cp Z37000 ${ARCHIVE}/restart/Z37000_${YEAR}
endif


mv fort.71 ${expno}_tho.ext4
mv fort.72 ${expno}_sao.ext4
... otherwise identical to LINUX example ...

tar cvf ${YEAR}.tar ${expno}*.ext4
mv ${YEAR}.tar ${ARCHIVE}/${YEAR}.tar

\rm fort.* ${expno}_*.ext4

YEAR=‘expr ${YEAR} + 1‘
\rm year.asc
echo $YEAR > year.asc

done
echo "submitting next job"
if [ ${YEAR} -le 100 ] ; then
qsub ${EXPDIR}/run_mpiom_hamocc_omip
fi

exit
```

2.5 Parallel Computing with MPI and OpenMP

MPI-OM is designed to run in parallel on different processors. Two ways of parallelization are possible. First, OpenMP (www.openmp.org) which supports shared-memory parallel programming (Several processors on one machine with access to the same shared memory). Second, MPI (Message Passing Interface, www.mpi-forum.org) which additionally supports parallelization across different machines (LINUX cluster) or different nodes (NEC SX-6). MPI and OpenMP libraries are available for almost all architectures including LINUX, SUN and NEC SX-6.

2.5.1 Running the OpenMP version

A model compiled with OpemMP can be started just like the non-parallel examples in section 2.4, only the environment variable

```
OMP_NUM_THREADS=<numprocs>
```

has to be set to the number of available processors.

2.5.2 Running the MPI version

For MPI, the calculated region has to be distributed up among the processors. The distribution along the x and y coordinates has to be defined in the beginning of the namelist OCECTL:

```
  &NPROCS
  nprocx=...
  nprocy=...
  &end
```

Computation is started by first calling the MPI daemon and then starting the MPI execution:

```
/sw/linux/mpi/bin/mpd --daemon &
/sw/linux/mpi/bin/mpiexec -np <numprocs> $HOME/model/model.x
```

The number of processors has to be equal to nprocx*nprocy. Model output is written in:

```
  oceout          for processor 0
  oceout_001      for processor 1
  oceout_002      for processor 2
```

If the number of processors is set to nprocx*nprocy+1 the model is computed in ”debug” mode. The additional processor computes the total area and, on each boundary exchange, there is a check if numbers are identical with the total area computation. For debugging, all numerical optimizations have to be switched off while compiling the model because, depending on the loop-length, optimizations can cause small numerical differences.

On Hurrikan (SX-6), because of better performance it is recommended to use OpemMP if not more than 8 processors (one node) are require.

2.5.3 MPI Examples

Sun and Linux

Please note that you need to have a MPI daemon on your machine which is compatible with the compiler that was used to compile the model. If your institute does not provide an MPI environment, you will probably have to compile the MPI library yourself. Alternatively to using ”mprun” as in this example the daemon and the executable can be started separately,as described above.

```
#! /bin/sh
set echo
set verbose

HOME=/home/patrick ; export HOME
ID=OMIP ; export ID

set echo
set verbose



cat > OCECTL  << EOF
 &NPROCS
 nprocx=4
 nprocy=3
  /
 &OCECTL
 DT      = 8640.
 CAULAPTS= 0.
 CAULAPUV= 0.005
 AUS     = 0.
 CAH00   = 1000.
 DV0     = 1.E-2
 AV0     = 1.E-2
 CWT     = 5.E-4
 CSTABEPS= 0.030
 DBACK   = 1.E-5
 ABACK   = 1.E-4
 CRELSAL = 2.0E-7
 CRELTEM = 0.
 CDVOCON = 0.05
 CAVOCON = 0.0
 NYEARS  = 1
 NMONTS  = 0
 IMEAN   = 2
 /
 &OCEDZW
  DZW = 20.,20., 20., 30.,40.,50.,70.
        ,90.,120.,150.,180.,210.,250.,300.
        ,400.,500.,600.,700.,900.,1400.
 /
EOF

cat >mpi.sh<< END
#! /bin/sh
cd $HOME/OMIP
set -e
./mpiom.x
set +e
END
chmod 755 mpi.sh

mprun -np 12 mpi.sh
```

NEC SX-6 (DKRZ-Hurrikan)

On Hurrikan (SX-6), because of better performance it is recommended to use OpemMP if the model should run on only one node. The following is an example taken from www.dkrz.de for an MPI Job which requests 2 compute nodes with 8 CPUs on each node (i.e. 16 CPUs in total !). The scheduler chooses the nodes dynamically, depending on the load. In the script the nodes are named 0 and 1. A four node run would have the numbers 2 and 3 for the additional nodes.

```
#!/bin/ksh
### PBS -S /bin/ksh          # NQSII Syntax  to set the shell
#PBS -l cpunum_prc=8         #  8 cpus per node
#PBS -l cputim_job=10:00:00  # 10 h cputime per node
#PBS -l memsz_job=6gb        #  6 GB Memory per node
#PBS -T mpisx
#PBS -b 2                    # job runs on 2 nodes
#PBS -j o                    # join err and out to out
#PBS -M myname@mymail.de     # you should always specify your email
                             # address for error messages etc
#PBS -N job_multi16          # job name

/bin/echo " job started at: " \\c
date
/bin/echo " ExecutionHost : " \\c
hostname                     # print name of current host
```

```
EXE=/ipf/x/xnnnnnn/model_mpi.x

mpiexec -host 0 -n 8 -host 1 -n 8 $EXE

/bin/echo " job completed at: " \\c
date
#
# ... data handling similar to the quickstart LINUX example ...
#
```

2.6 Compliling the Model

Most users at some point will have to compile the model them self. The source code is available from CD or, if you are a registered user, from the ZMAW CVS server.

To retrieve the sources from CVS you have to do the following:

```
setenv CVSROOT :pserver:<user-name>@cvs.zmaw.de:/server/cvs/mpiom1
cvs login
cvs checkout mpi-om
```

In any way you will end up with a directory named mpi-om and five subdirectories:

1. **src**
   All sources required to compile the MPI-OM ocean standalone model.

2. **src_hamocc**
   All sources required to compile the HAMOCC marine biogeochemistry together with the MPI-OM ocean model.

3. **make**
   Makefiles required to compile the MPI-OM ocean and the MPI-OM/HAMOCC model.

4. **bin**
   This is where the binaries are stored after compilation.

5. **run**
   Shell scripts to set up a runtime environment.

If you are working within the Max Planck Institute for Meteorology (MPI-MET) or the DKRZ, just go to the make directory and type ”make -f Makefile_mpiom_omip”. The Makefile will automatically detect your machine-type and select an available compiler. If you need to use a different compiler or if you are not working within the MPI-MET or the DKRZ, you will have to provide and select the compiler and, if required, the MPI and the NetCDF libraries yourself. The example below shows parts of the standard Makefile.

```
#-----------------------------------------------------------------------------
DEF =   -DZZNOMPI -DVERSIONGR30 -DZZLEVELS40 -DZZTIMECHECK \
        -DZZYEAR360 -DSOR -DZZRIVER_GIRIV \
        -DMEAN -DRESYEAR -DZZDEBUG_ONEDAY \
        -DQLOBERL -DBULK_KARA \
        -DEISREST -DREDWMICE -DALBOMIP \
        -DISOPYK -DGMBOLUS \
        -DADPO -DSLOPECON_ADPO  \
        -DNURDIF \
        -DDIAG -DZZGRIDINFO -DZZDIFFDIAG -DZZKONVDIAG \
        -DZZCONVDIAG -DZZAMLDDIAG -DTESTOUT_HFL \
        -DZZRYEAR
#-----------------------------------------------------------------------------

PROG = mpiom.x

VPATH = ../src

SRCS = absturz.f90 adisit.f90 adisit1.f90 adisitj.f90 amocpr.f90 aufr.f90 \
        ....

OBJS = absturz.o adisit.o adisit1.o adisitj.o amocpr.o aufr.o \
        ....

# Set up system type
UNAMES := $(shell uname -s)
HOST   := $(shell hostname)

ifeq ($(UNAMES),SunOS)
NETCDFROOT = /pf/m/m214089/yin/local/SunOS64
NETCDF_LIB = -L${NETCDFROOT}/lib -lnetcdf
NETCDF_INCLUDE = -I${NETCDFROOT}/include

MPIROOT = /opt/SUNWhpc
MPI_LIB = -L${MPIROOT}/lib/sparcv9 -R${MPIROOT}/lib/sparcv9 -lmpi
MPI_INCLUDE = -I${MPIROOT}/include
endif

INCLUDES = $(NETCDF_INCLUDE) $(MPI_INCLUDE)
LIBS = $(NETCDF_LIB) $(MPI_LIB)

ifeq ($(UNAMES), SunOS)
#-----------------------------------------------------------------------------
#FOR SUN (SunStudio10 compiler)
F90 = f95
F90FLAGS = $(INCLUDES) -xtypemap=real:64,double:64,integer:32 -fast \
                       -g -xarch=v9b -xchip=ultra3cu  -fpp
# OpenMP: -xopenmp
endif
```

```
LDFLAGS = $(F90FLAGS)

all: $(PROG)

$(PROG): $(OBJS)
$(F90) $(LDFLAGS) -o $@ $(OBJS) $(LIBS)
cp $(PROG) ../bin/.

clean:
rm -f $(PROG) $(OBJS) *.mod i.*.L

.SUFFIXES: $(SUFFIXES) .f90

%.o: %.f90
$(F90) $(F90FLAGS) -c $(DEF) $<

#
#-------------------------------------------------------------------------------
# Dependencies
#
absturz.o: mo_commo1.o mo_commo2.o mo_param1.o mo_units.o mo_parallel.o
...
```

2.6.1 Conditional Compilation

The MPI-OM source code contains a number of cpp flags for conditional compilation. There are different groups of compile options. First, the group which controls the resolution, the configuration for different platforms and different forcing or coupling options. Second, the group which controls different parameterizations of the model physics. Third, the group which controls the model output. All options are described in the following tables.

Key name	Action	Reference
`VERSIONGIN`	grid with high resolution in the Greenland, Iceland and Norwegian Seas	
`VERSIONGR03`	GR grid (poles over Greenland and Antarctica) with nominal resolution of 0.3 °	
`VERSIONGR09`	GR grid with nominal resolution of 0.9 °	
`VERSIONGR15`	GR grid with nominal resolution of 1.5 °	
`VERSIONGR30`	GR grid with nominal resolution of 3.0 °	
`VERSIONGR60`	GR grid with nominal resolution of 6.0 °	
`VERSIONT43`	grid with about T42 resolution and increased resolution at the Equator	
`LEVELS23`	run with 23 layers (default is 20)	
`LEVELS30`	run with 30 layers (default is 20)	
`LEVELS40`	run with 40 layers (default is 20)	

Tab. 2.2: List of cpp flags for grid options.

Key name	Action	Reference
__coupled	coupled to ECHAM5 with PRISM coupler	8.1.2
__synout	enable coupler ASCII output to file oceout	8.1.2
bounds_exch_isend	MPI parallelization boundary exchange	8.1.2
bounds_exch_put	MPI parallelization boundary exchange	8.1.2
use_comm_MPI1	MPI1 parallelization library	8.1.2
use_comm_MPI2	MPI2 parallelization library	8.1.2
SALTCORRECT	correct salinity when coupled to ECHAM5	8.1.4
TEMPCORRECT	correct temperature when coupled to ECHAM5	8.1.4
FLUXCORRECT	correct surface fluxes when coupled to ECHAM5	8.1.4
PBGC	compiling with HAMOCC marine biogeochemistry	
FB_BGC_OCE	downward solar penetration modulated by HAMOCC marine biology; without PBGC: penetration climatology from file	5.2.3
PDYNAMIC_BGC	diagnistic output for HAMOCC	

Tab. 2.3: List of cpp flags for forcing or coupling options.

Key name	Action	Reference
SOR	use successive over-relaxation time stepping (barotropic solver)	5.2.8
ISOPYK	Isopycnal diffusion	
GMBOLUS	Gent and McWilliams style eddy-induced mixing (Gent *et al.*, 1995a)	5.2.15
BOLK05	multiply default Gent and McWilliams Bolus coefficient by 0.5	5.2.15
BOLK025	multiply default Gent and McWilliams Bolus coefficient by 0.25	5.2.15
REDWMICE	reduced eddy mixing energy transfer in presence of sea ice	5.2.1
GMVISETAL	calculate Richardson number coefficients after Visbeck *et al.* (1997)	5.2.15
ADFS	predictor-corrector advection scheme	5.2.14
QUICK	quick-scheme as proposed by Farrow and Stevens (1995)	
QUICK2	quick-scheme with modified boundary treatment	
ADPO	total variation diminishing (TVD) advection scheme (Sweby, 1984).	5.2.13
SLOPECON_ADPO	bottom boundary layer transport scheme for the ADPO advection scheme	5.2.12 and 4.2.1
FREESLIP	no friction on boundaries	
NURDIF	vertical diffusion and no mixing (coefficient is set in namelist)	5.2.1
NURMISCH	only vertical mixing and no diffusion	5.2.1
UMKLAP	convective adjustment instead of vertical mixing and diffusion	5.2.1
PLUME	PLUME convection after Stössel instead of convective adjustment	5.2.1
DBACKGFDL	background vert. diffusion to profile after GFDL	
DBACKGFDL2	background vert. diffusion as mean of DBACK and GFDL	
DBACKPROFIL	set minimum vertical diffusion to profile	
SCHEP	modify horizontal diffusion of momentum	
AULREDSC	reduce grid dependence of horizontal biharmonic diffusion (power of 4 to power of 3) for tracers (T,S)	
AULREDUV	reduce grid dependence of horizontal biharmonic diffusion (power of 4 to power of 3) for momentum (u,v)	
BULK_KARA	use bulk formula from from Kara for surface heat balance	
DASILVA	use bulk formula from from DaSilva for surface heat balance	
DRAGGILL	use bulk formula values from gill (atmosphere ocean dynamics) for surface heat balance	
QLOBERL	compute downward long-wave radiation according to Berliand and Berliand (1952).	
OPEND55	increase exp. scale for downward solar penetration by 2 relative to default	

Tab. 2.4: List of cpp flags for the physical parameterisation.

Key name	Action	Reference
MEAN	enable monthly/daily mean output	section 7
DIAG	enable diagnostic output	
DIFFDIAG	enable diagnostic output for diffusion	table 7.3
CONVDIAG	enable diagnostic output for convection	table 7.1
AMLDDIAG	enable diagnostic output for max. monthly mixed layer depth	table 7.1
FORCEDIAG	enable diagnostic output for surface forcing	table 7.4
KONVDIAG	enable diagnostic output for convection	table 7.1
MFLDIAG	enable diagnostic output for divergence free velocity	table 7.1
GRIDINFO	enable grid information output	table 7.3
PNETCDFO	enable output in NetCDF format	
TESTOUT_HFL	enable diagnostic output for heat-fluxes	table 7.5
OASIS_FLUX_DAILY	enable daily output for OASIS/PRISM coupler fluxes	table 7.7

Tab. 2.5: List of cpp flags for diagnosis.

Key name	Action	Reference
RESTORE_MON	enable surface salinity restoring to monthly climatology	equation 4.32
YEAR360	run with climatological forcing (360 days per year)	section 4.5
RYEAR	run with real year forcing (e.g. NCEP)	section 4.6
RESYEAR	write restart file at the end of a year; default: month	
SW089	multiply incoming short wave radiation by 0.89 (NCEP only)	section 4.6
ALBMELTHI	tune albedo for sea-ice and snow	
ALBMSNO7	tune albedo for snow	
ALBNCEP	tune albedo for NCEP forcing	
ALBOMIP	tune albedo for OMIP forcing	
RIVER_GIRIV	read river runoff data from GIRIV and not from runoff_obs and runoff_pos files	
GLACCALV	read glacier calving for file gletscher_5653	
FORCE_SLP	read sea level pressure forcing (GIPRESS) for flux calculations	
FORCE_DWLW	read downward long-wave radiation forcing (GIDWLW) for flux calculations	
EISTEST	enable sea-ice module test	
DRAGTEST	enable drag coefficient test	
DEBUG	enable debugging	
DEBUG_ONEDAY	set the length of a month to one day for debugging	
TIMECHECK	check the time-step in the parallel version wit MPI	

Tab. 2.6: List of cpp flags for runtime control and testing.

Key name	Action	Reference
NAG	compiling with NAG compiler (Sun OS)	8.1.3
__IFC	compiling with Intel Fortran compiler (Windows and Linux)	
NEC	Run on NEC SX-6 (DKRZ Hurrikan)	

Tab. 2.7: List of cpp flags for compiler and system options.

3. CREATING A NEW SETUP

3.1 Creating the Grid

3.1.1 anta

Step one is the creation of the latitudes and longitudes of the new Arakawa C-grid. We will need this file during the rest of the grid generation procedure. The job script to create a new bipolar grid for MPI-OM is called "mk_anta-etopo2.job". Users need to adjust position and size of the two poles as well as the grid resolution. The program should be run at double precision. The example below is for the GROB60 test-setup.

```
#INPUT PARAMETERS TO BE EDIT BY THE USER
#####################################
#name of the grid
grid=GR60

#1st pole positions
rlat1=72.
rlon1=-40.

#phi determines size of 1st pole
#phi=pi/2 gives no hole
#phi smaller than pi/2 gives increasingly larger hole
#note: size of second pole adjusted with parameter je
phi=1.49

#2nd pole Eurasia
rlat2=-84.
rlon2=60.

#horizontal dimensions
ie=60
je=50
######################################
```

INPUT/OUTPUT:

1. **INPUT: etopo2.ext**
   ETOPO-5 dataset (Digital relief of the Surface of the Earth) in big endian extra format.

2. **OUTPUT: anta**
   Grid file with lat's and lon's from the Arakawa C-grid. This file is needed during the rest of the procedure, e.g. generation of initial and forcing data)

3. **OUTPUT: topo**
   Ocean topography in ASCII format. For coarse resolution setups this file usually needs to be ”corrected” by hand to get a proper representation of the ocean topography, e.g. Bering Strait or Panama etc. All important sill depth should be checked, e.g. Drake Passage or Denmark Strait.

3.1.2 arcgri

Step two is the creation of the grid distances of the new grid. The job script to create the second grid description file for MPI-OM is called ”mk_arcgri.job”. The file contains the grid distances dlxp, dlxp, dlxu, dlyu, dlyu and dlyv of the scalar and vector points of the Arakawa C-grid as well as the local coriolis parameter f. Users need to change the dimensions ie and je accordingly. The program should be run at double precision.

INPUT/OUTPUT:

1. **INPUT: anta**
   The grid file with lat's and lon's from the Arakawa C-grid in big endian extra format.

2. **OUTPUT: arcgri**
   Grid file with distances dlxp, dlxp, dlxu, dlyu, dlyu and dlyv of the scalar and vector points of the Arakawa C-grid as well as the local coriolis parameter in big endian extra format.

3.1.3 BEK

Step three is the creation of BEK file for the new grid. The job script to create the BEK file for MPI-OM is called ”mk_BEK.job”. The file stores the information on the ocean basins which is of importance for the diagnostic output. Users need to change the dimensions ie and je in the script accordingly.

INPUT/OUTPUT:

1. **INPUT: anta**
   The grid file with lat's and lon's from the Arakawa C-grid in big endian extra format.

2. **INPUT: topo**
   Ocean topography in ASCII format.

3. **OUTPUT: BEK**
   Ocean basin information in ASCII format.

4. **OUTPUT: ibek.ext**
   Ocean basin information in EXTRA format.

Attention: The generated BEK file has to be modified by hand afterwards and some diagnostic output requires modifications in the source code!

3.2 Creating the Input

3.2.1 INISAL and INITEM

In step four we generate the initial value data sets for temperature and salinity by interpolating a global ocean climatology onto the model grid. Two global hydrographic climatologies are available. First, the "PHC" climatology with improved arctic ocean information from Steele *et al.* (2001) and second, the new global hydrography from Gouretski and Koltermann (2004). The job script to create the files from the Steele *et al.* (2001) climatology is called "mk_phc.job", the script to create them from the Gouretski and Koltermann (2004) climatology is called "mk_sac.job". The user has to specify the dimensions ie and je and the vertical resolution. The following example is from "mk_phc.job":

```
foreach code ( temp salt )
foreach time ( an )   # opton : mon

if $version == GR60 then
set me=60
set ne=50
set lev2=20
endif
```

INPUT/OUTPUT:

1. **INPUT: anta**
   The grid file with lat's and lon's from the Arakawa C-grid in big endian extra format.

2. **INPUT:PAC or SAC file**
   Ocean climatology in EXTRA format.

3. **OUTPUT: INISAL, INITEM**
   Ocean initial values for temperature and salinity.

3.2.2 SURSAL and SURTEM

The surface restoring maps, SURSAL and SURTEM, have to be extracted from the 3-D INISAL, INITEM by selecting the first level (CDO or EXTRA tools, "sellevel"). A monthly climatology is only available for Steele *et al.* (2001).

3.2.3 runoff_obs and runoff_pos

Positions of river discharge are stored with latitudes and longitudes. Their referring grid points are assigned during runtime. Nevertheless, to make sure the positions fit to the grid runoff_obs and runoff_pos have to be specifically created for each setup. The job script to create the files is called "mk_runoff.job".

INPUT/OUTPUT:

1. **INPUT: anta**
   The grid file with lat's and lon's from the Arakawa C-grid in big endian extra format.

2. **INPUT: runoff.nc**
   Global runoff in NetCDF format from the OMPI project.

3. **INPUT: land_sea_mask.ECMWF.ext4**
   The ECMWF land-sea mask in EXTRA format.

4. **INPUT: ibek.ext**
   Ocean basin information in EXTRA format.

5. **OUTPUT: runoff_obs**
   Monthly mean river discharge data (currently for 53 positions) in EXTRA format.

6. **OUTPUT: runoff_pos**
   Longitudes and latitudes of the river discharge positions in EXTRA format.

3.3 Interpolate the Forcing

Now that all grid information is available we need to provide the daily mean surface fields of heat, freshwater and momentum fluxes (see also.2.1.3). A popular choice is the climatological OMIP forcing as discussed in 4.5. The job script to interpolate the forcing from the OMIP climatology to the choosen model grid is called ”forcing_omip.job”. The user has to specify the dimensions ie and je.

INPUT/OUTPUT:

1. **INPUT: anta**
   The grid file with lat’s and lon’s from the Arakawa C-grid in big endian extra format.

2. **INPUT:OMIP climatology data**
   2m_dewpoint_temperature.nc, east_west_stress.nc, north_south_stress.nc, scalar_wind.nc, total_precipitation.nc, 2m_temperature.nc, mean_sea_level_pressure.nc, total_cloud_cover.nc and total_solar_radiation.nc in NetCDF format.

3. **INPUT: land_sea_mask.ECMWF.ext4**
   The ECMWF land-sea mask in EXTRA format.

4. **OUTPUT: surface forcing fields**
   Total cloud cover: GICLOUD
   Precipitation: GIPREC
   Solar radiation: GISWRAD
   Dew point temperature: GITDEW
   Surface temperature: GITEM
   10 m wind speed: GIU10
   zonal wind stress: GIWIX
   meridional wind stress: GIWIY
   All files in EXTRA format.

Also available is a forcing compiled from the NCEP/NCAR reanalysis (section 4.6). The script is called ”forcing_ncep.job”. The NCEP/NCAR reanalysis provides the same input fields as the OMIP project. The only difference is that ”forcing_ncep.job” has to consider the leap years.

3.4 Modifications in the Source

The generation of a new grid requires some modifications in the source code, encapsulated with an appropriate CPP switch (see 2.6.1). The new grid has to be defined in the following subroutines:

1. **mo_param1.f90**
   Define new global model dimensions.

2. **diag_ini.f90**
   Define new diagnostic locations for timeseries output.

3. **mo_commodiag.f90**
   Define different variables for grid types with different orientation, because the through-flows are different.

4. **diagnosis.f90**
   Grid types are taken into account when calculating the diagnostic output.

4. NUMERICAL BACKGROUND

Most parts of this chapter are taken from Marsland *et al.* (2003) and the HOPE model description by Wolff *et al.* (1997).

4.1 Ocean Primitive Equations

The horizontal momentum balance for a hydrostatic Boussinesq fluid on a rotating sphere is

$$\frac{d\vec{v}_o}{dt} + f(\vec{k} \times \vec{v}_o) = -\frac{1}{\rho_w}\left[\vec{\nabla}_H(p + \rho_w g\zeta)\right] + \vec{F}_H + \vec{F}_V \tag{4.1}$$

where $\vec{v}_o = (u_o, v_o)$ is the oceanic horizontal velocity vector on the orthogonal coordinates, t is the time, f is the Coriolis parameter, $\vec{k}$ is a unit vector normal to the earth's center, ρ_w is a constant reference density, $\vec{\nabla}_H$ is the horizontal gradient operator, p is the internal pressure, g is the acceleration due to gravity and ζ is the sea surface elevation. The total derivative is given by $\frac{d}{dt} = \frac{\partial}{\partial t} + \vec{v}_o \cdot \vec{\nabla}_H + w_o \cdot \frac{\partial}{\partial z}$ where w_o is the vertical component of ocean velocity and $\frac{\partial}{\partial z}$ is the vertical partial derivative. $\vec{F}_H$ and $\vec{F}_V$ are parameterizations of horizontal and vertical eddy viscosity, respectively.

Diagnostic treatment of pressure and density is used to close the momentum balance. Density ρ is taken to be a function of model pressure, temperature and salinity according to the equation of state polynomial defined by the Joint Panel on Oceanographic Tables and Standards (UNESCO, 1983). Potential temperatures are converted to *in situ* temperatures for the density calculation. The pressure is calculated using the hydrostatic equation.

$$\frac{\partial p}{\partial z} = -g\rho \tag{4.2}$$

Forward integration of the ocean model surface elevation is based on a linearized kinematic boundary condition stating that the time rate of change of surface elevation is equal to the vertical component of oceanic velocity at the surface.

$$\frac{\partial \zeta}{\partial t} = w_o|_{z=\zeta} \tag{4.3}$$

The vertical velocity is calculated from the horizontal velocity field using the incompressibility condition.

$$\frac{\partial w_o}{\partial z} = -\vec{\nabla}_H \cdot \vec{v}_o \tag{4.4}$$

Integrating over the entire depth gives the vertical velocity at the sea surface.

$$w_o|_{z=\zeta} = -\vec{\nabla}_H \cdot \int_{-H}^{\zeta} \vec{v}_o \, dz \tag{4.5}$$

Potential temperature θ and salinity S obey the advection-diffusion equations

$$\frac{d\theta}{dt} = \vec{\nabla}_H \cdot \left(\mathbf{K}\vec{\nabla}_H \theta\right) \tag{4.6}$$

$$\frac{dS}{dt} = \vec{\nabla}_H \cdot \left(\mathbf{K}\vec{\nabla}_H S\right) \tag{4.7}$$

where the tensor $\mathbf{K}$ is a subgridscale parameterization of horizontal/isoneutral and vertical/dianeutral diffusion.

The surface fields can be relaxed to climatological fields $(\Theta_\star, S_\star)$ by specifying

$$D_V \frac{\partial \Theta}{\partial z} = \lambda(\Theta_\star - \Theta) \tag{4.8}$$

$$D_V \frac{\partial S}{\partial z} = \lambda(S_\star - S) \tag{4.9}$$

Relaxation time constants can be set in the namelist (2.1). At lateral boundaries and at the sea floor no flux conditions apply for heat and salt.

4.2 Ocean Subgridscale Parameterizations

The coarse horizontal and vertical resolution of OGCMs necessitates the use of subgridscale parameterizations. The MPI-OM model currently has formulations for BBL slope transport, horizontal and vertical viscosity, vertical and isopycnal diffusivity, eddy-induced mixing, and convection, as described below. With respect to the many parameter choices necessary for inclusion of subgridscale processes in the MPI-OM model (and OGCMs more generally), we note the myriad sensitive interactions that such choices have on each other. Since computational constraints do not currently permit a full spanning of parameter space the parameter choices used in the present version of MPI-OM are a necessary blend of those gained via past experience with the HOPE model, those available from the bandwidth of literature values, and those that were tuned (in a sometimes *ad hoc* manner traditionally referred to as the art of numerical modeling) to give iteratively improved results as part of the model development process. The latter is relevant to, for example, our choices of friction, diffusion, convection and wind mixing parameters.

4.2.1 Slope Convection

The thermohaline circulation of the world's ocean is maintained by the sinking of dense water masses in high latitudes. In most cases these water masses form in marginal seas or on shelves and slide down the continental slopes. One particular phenomenon is the 'overflow', where dense near-bottom waters cross sills between ocean basins (e.g. Price and Baringer, 1994). This process is not well resolved in coarse-resolution z-coordinate ocean models owing to a lack of a BBL. In such models the dense water passing over a sill is advected horizontally and placed above lighter water. The resulting convection (parameterized as strong mixing or intensified diffusion) leads to an overestimation of the mixing between the overflow and the ambient water. To overcome this problem Beckmann and Döscher (1997) developed a BBL parameterization where the horizontal flow is redirected as if it flowed along the bottom. This approach to the BBL problem can be avoided by the use of σ-coordinate models (e.g. Ezer and Mellor, 1997), which are been compared to both isopycnal and z-coordinate models in the Dynamics of Overflow, Mixing and Entrainment (DOME) Project. Advantages and disadvantages of the various vertical coordinate systems are discussed by the DYNAMO Group (1997) or on the DOME webpage (www.rsmas.miami.edu/personal/tamay/DOME/dome.html).

In the HOPE model a relatively simple slope convection scheme was introduced by Legutke and Maier-Reimer (2002). In the Legutke and Maier-Reimer scheme the slope convection was considered to be a directional diffusive transport whereby an exchange of tracer occurs between the bottom cells of two adjacent horizontally discretised water columns. The BBL scheme newly introduced into MPI-OM is schematically represented in Fig. 4.1 which shows the BBL transport Tr$_{BBL}$ from a source cell to a target cell. The implementation is similar to that of Beckmann and Döscher (1997) with two notable exceptions. Firstly, the BBL thickness H_{BBL} may be less than the model layer thickness Δz_s of the source cell. That is,

$$H_{BBL} = \min\left(\Delta z_s, BBL_{max}\right) \tag{4.10}$$

where BBL$_{max}$ is a prescribed maximum thickness. The motivation is to limit the BBL transport in coarse vertical resolution ocean models where the deeper grid cells can be of order 1 km thickness. Secondly, as suggested by Campin and Goosse (1999) the BBL transport is redirected to a level of neutral buoyancy rather than to the bottom cell of a horizontally adjacent water column. For a source cell having density ρ_s at model level k$_s$, the target cell level k$_t$ is chosen by the following condition.

$$k_t = \begin{cases} k_\rho & if \quad \rho_t \leq \rho_s \quad and \quad \rho_{t+1} > \rho_s \\ k_{bot} & if \quad \rho_{bot} \leq \rho_s \end{cases} \tag{4.11}$$

Here, the subscript *bot* denotes the deepest wet model layer in a target column. The neutral buoyancy approach is particularly useful in regions

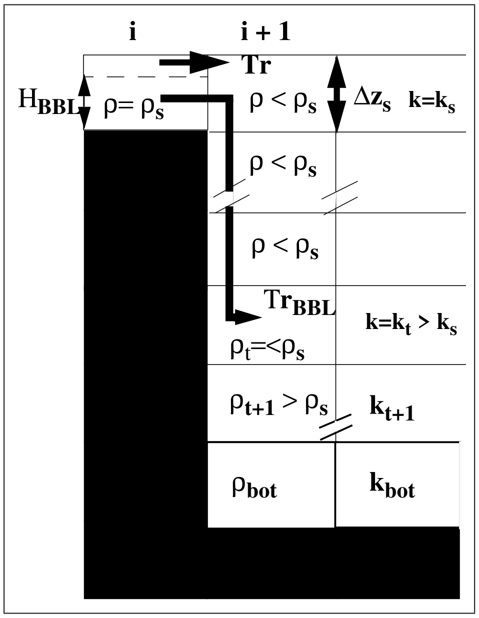

Fig. 4.1: Schematic diagram of the bottom boundary layer advective transport to neutral density level. Model levels are indicated by k where k$_s$ and k$_t$ indicate the levels of the source and target cells respectively. The grid index i refers to either meridional or parallel directions in the curvilinear horizontal discretization. For additional details refer to the text.

where coarse horizontal resolution does not allow for a gradual descent on a staircase-like slope. The resultant BBL transport is given by

$$Tr_{BBL} = u_s \Delta x, y H_{BBL} \tag{4.12}$$

where u_s represents the meridional (parallel) velocity in the model level k$_s$, and $\Delta x, y$ represents the parallel (meridional) source cell width. The remaining advective transport is given by

$$Tr = u_s \Delta x, y (\Delta z_s - H_{BBL}) \tag{4.13}$$

and is purely horizontal. Although not used in this study, MPI-OM also includes an improved version of the BBL scheme where H_{BBL} is calculated locally depending on the stratification and friction velocity using the formulation of Killworth and Edwards (1999).

4.2.2 Eddy Viscosity

The horizontal and vertical eddy viscosity are treated separately. Horizontal eddy viscosity $\vec{F}_H$ is parameterized using a scale-dependent biharmonic formulation

$$\vec{F}_H = -\vec{\nabla}_H \cdot \left(B_H \vec{\nabla}_H \Delta_H \vec{v}_o \right) \tag{4.14}$$

where B_H is a coefficient proportional to the fourth power of the grid spacing.

Vertical eddy viscosity $\vec{F}_V$ is parameterized as

$$\vec{F}_V = \frac{\partial}{\partial z}\left(A_V \frac{\partial}{\partial z}\vec{v}_o\right). \tag{4.15}$$

The eddy coefficient A_V is partially relaxed to the value at the previous time step by use of a time filter to avoid $2\Delta t$ oscillation. Using n and Λ_V to denote the time increment and relaxation coefficient gives:

$$A_V{}^n = (1-\Lambda_V)\,A_V{}^{n-1} + \Lambda_V\left(A_{VO}\,(1+C_{RA}R_i)^{-2} + A_w + A_b\right). \tag{4.16}$$

The time linear relaxation coefficient Λ_V is set to 0.6 in accordance with past experience. Following Pacanowski and Philander (PP; 1981) the Richardson number R_i dependent mixing term includes constant coefficients A_{VO} and C_{RA}. A small constant background viscosity representing mixing by internal wave breaking is denoted by A_b. The PP scheme in its classical form underestimates turbulent mixing close to the surface. Therefore an additional parameterization for the wind induced stirring A_w is included. The near surface wind mixing is proportional to the cube of the local ten meter wind speed V_{10m}, and is reduced in proportion to the fractional sea ice cover I. It decays exponentially with depth and depends on the local static stability $\delta_z\rho$.

$$\begin{aligned} A_w(1) &= (1-I)W_T V_{10m}^3 && (4.17)\\ A_w(k) &= A_w(k-1)\frac{\frac{\lambda}{\Delta z}}{\frac{\lambda}{\Delta z}+\delta_z\rho}e^{\frac{\Delta z}{z_0}} && (4.18)\end{aligned}$$

where k=2,3,...,k_{bot} is the vertical level and Δz is the level thickness; λ, z_0, and W_T are adjustable parameters which were tuned for optimal mixed layer depths.

Tracer diffusion in Eqns 4.6 and 4.7 is represented in two optional ways. The diffusion tensor $\mathbf{K}$ can be chosen either to represent: a) standard horizontal/vertical diffusion

$$\mathbf{K} = D_H \begin{bmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & \epsilon \end{bmatrix} \tag{4.19}$$

with $\epsilon = \frac{D_V}{D_H}$; or b) isoneutral/dianeutral diffusion

$$\mathbf{K} = D_H \begin{bmatrix} 1 & 0 & S_x\\ 0 & 1 & S_y\\ S_x & S_y & \epsilon + S_{dif}^2 \end{bmatrix} \tag{4.20}$$

with $\epsilon = \frac{D_V}{D_H}$ and $S_{dif} = (S_x, S_y, 0) = (\frac{-\delta_x \rho}{\delta_z \rho}, \frac{-\delta_y \rho}{\delta_z \rho}, 0)$.
The transformation follows Redi (1982) with the small slope approximation by Gent *et al.* (1995b). The scheme is numerically implemented following Griffies (1998).

The effect of tracer mixing by advection with the unresolved mesoscale eddies is parameterized after Gent et al. (1995).

The vertical eddy diffusivity coefficient D_V is treated similarly to Eqn 4.16, except for the cubic dependence on the shear instability-dependent (Richardson number) term:

$$D_V{}^n = (1 - \Lambda_D)\, D_V{}^{n-1} + \Lambda_D \left(D_{VO}\, (1 + C_{RD} R_i)^{-3} + D_w + D_b \right). \quad (4.21)$$

As with the vertical viscosity, D_{VO}, C_{RD} and the small background term D_b are constant. The wind-induced term D_w is treated in the same manner as for viscosity.

There are several choices for parameterization of convection currently available in the MPI-OM model. For details please see 5.2.1

4.3 Model Grid

Horizontal discretization of the MPI-OM model is on a staggered Arakawa C-grid (Arakawa and Lamb, 1977). The reference grid is shown in figure 4.2. It consists of prism-shaped finite volumes with the edges aligned with coordinates. Vertical discretization is on a so called ‘z-coordinate’ system (differentiating from pressure or density coordinate systems).

4.3.1 Horizontal discretization

Variables defined on vector points are :

- horizontal velocities (u, v)
- wind stress $\underline{\tau} = (\tau^\phi, \tau^\lambda)$
- coefficients of vertical viscosity

Variables defined on scalar points are :

- potential temperature Θ
- salinity S
- density ρ

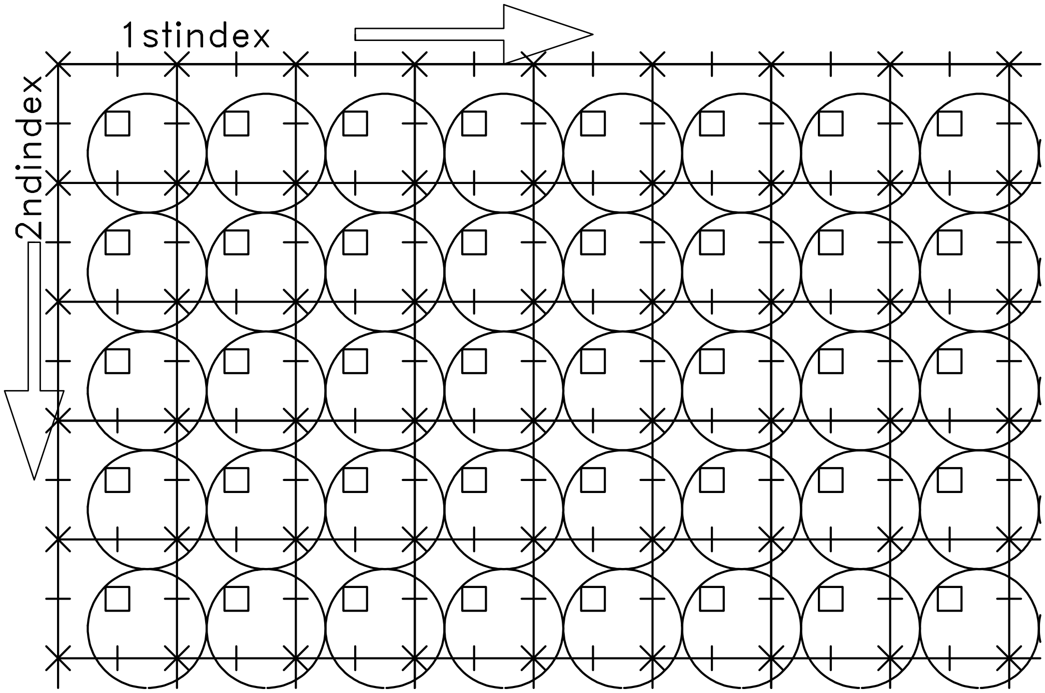

Fig. 4.2: *Layout of the Arakawa C–grid. Boxes denote scalar points, slashes vector points (u in i-direction and v in j-direction) and stars the points where the stream function ψ is defined.*

- pressure p
- vertical velocity w
- coefficients of vertical diffusivity
- sea surface elevation ζ
- heat– and freshwater fluxes across the air sea interface

Diagnostic stream functions are defined in the ψ points of the grid.

Multiple grid-refinements on an orthogonal grid are possible, i.e. MPI-OM calculates a grid using two vectors, one containing information on longitudes and one on latitudes. A high resolution in a specified region may cause very "slim" grid cells elsewhere.

Indexing is done as follows:

West – East : $\texttt{I} = 1, \ldots, \texttt{IE}$,
North – South : $\texttt{J} = 1, \ldots, \texttt{JE}$,
Sea surface – Bottom : $\texttt{K} = 1, \ldots, \texttt{KE}$

The points with west–east indices 1 and 2 correspond to points with indices IE-1 and IE, respectively, when periodic boundaries are chosen.

The spherical geometry of the earth is taken into account by storing all grid distances $\Delta x(\lambda, \phi)$ and $\Delta y(\lambda, \phi)$ on arrays, which are then used in the discretization formulas.

4.3.2 Vertical discretization

The vertical discretization is the same as used in the HOPE model (Wolff *et al.*, 1997), which includes partial vertical grid cells, i.e. at each point in the horizontal grid the deepest wet cell has a uniform thickness that is adjusted to resolve the discretised bathymetry. The surface layer thickness is also adjusted to account for the sea surface elevation and the sea ice/snow draft where appropriate.

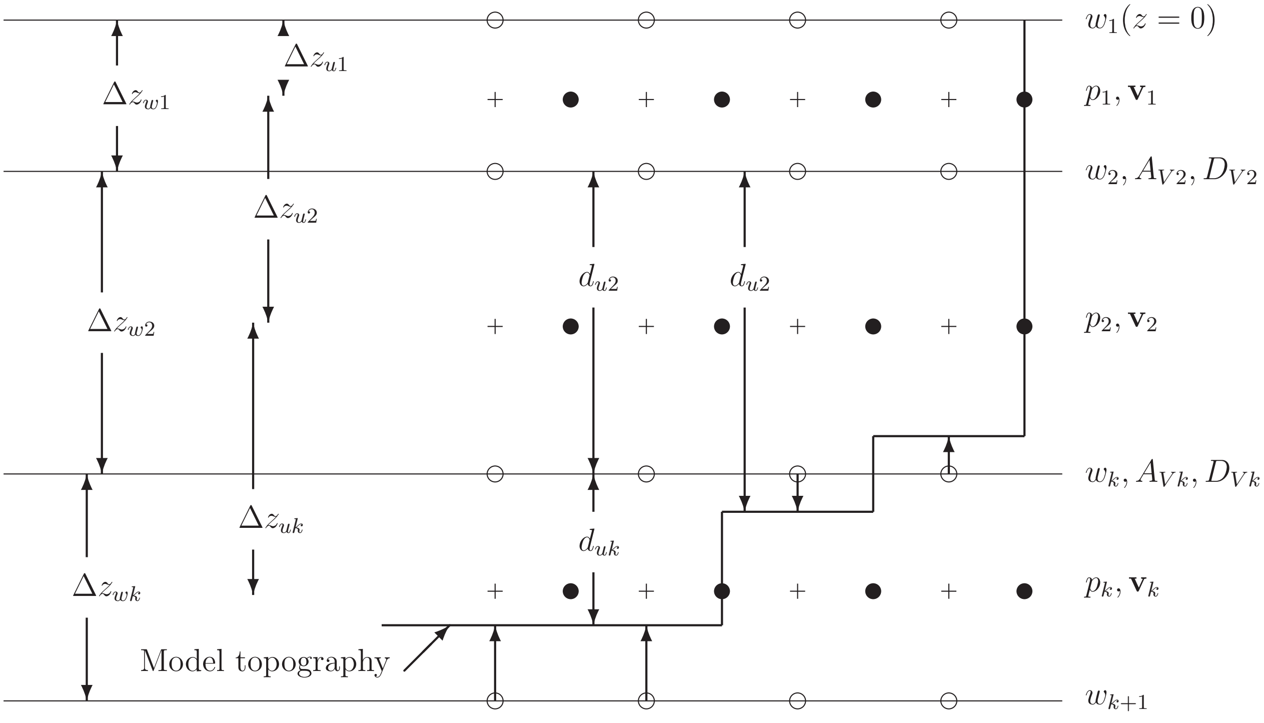

Fig. 4.3: Vertical structure of model grid

The choice of depth values at vector points, and thus the assignment of layer thicknesses, is free. The vertical velocity component is then computed as indicated in fig. 4.2. Model arrays are defined as follows:

Array(*dimensions*) : description

DZW(KE) : Vertical distance between two vertical velocity points (layer thickness). The layer thicknesses are set in the main program via

DATA DZW / .../ and may be changed by the user. All other vertical distances are computed from this array.

$$\texttt{DZW(K)} = \Delta z_{wk} \qquad k = 1, \ldots, \texttt{KE}$$

TIESTW(KE+1) : Actual depth of $w-$ levels (TIESTW(1)=0)

$$\texttt{TIESTW(K)} = \sum_{l=2}^{k} \Delta z_{wl-1} \qquad k = 2, \ldots, \texttt{KE}$$

TIESTU(KE+1) : Actual depth of vector/scalar levels (depend only on k)

$$\texttt{TIESTU(K)} = 0.5 * (\texttt{TIESTW(K+1)} - \texttt{TIESTW(K)}) \qquad k = 1, \ldots, \texttt{KE}$$

DZ(KE) : Vertical distance between two vector/scalar points (for K=1 this is half of the first layer thickness)

$$\texttt{DZ(K)} = \Delta z_{uk} \qquad k = 1, \ldots, \texttt{KE}$$

After a topography dataset has been supplied on the user specified grid (at the positions of vector points) MPI-OM recalculates the actual depth of scalar points as the maximum depth of the four surrounding vector points. Having done this, local layer thicknesses are computed taking into account the adjustment of near bottom vertical velocity points (see fig. 4.3).

The layer thicknesses are stored on arrays DDUE, DDUO and DDPO

$\texttt{DDU}_{E/O}(\texttt{IE}, \texttt{JE}, \texttt{KE})$: Layer thicknesses at vector points

$\texttt{DDPO}(\texttt{IE}, \texttt{JE}, \texttt{KE})$: Layer thicknesses at scalar points

where E/O indicate the vector points in u and v direction.

4.3.3 Curvilinear Coordinate System

The MPI-OM model uses a bipolar orthogonal spherical coordinate system. If the poles are antipodes (diametrically opposed) then the coordinate system is reduced to a rotated spherical grid. Otherwise, orthogonal meridians and parallels are constructed according to the choice of zonal and meridional resolution and are used to define the spatial mesh. Although it may be desirable to maintain ‘quadrature’ of the grid (i.e. within each grid cell the local zonal and meridional grid distances are equal), it is by no means a necessary condition. Two advantages can be achieved by assignment of a radius to the poles. Firstly, land points can be removed from the computational matrix. Secondly, by choosing non-equal pole radii horizontal

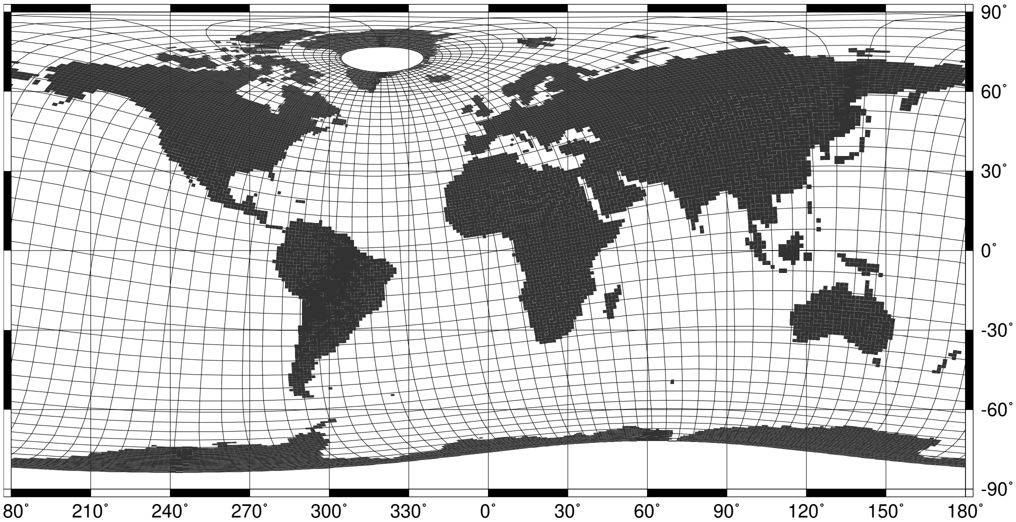

Fig. 4.4: Standard MPI-OM orthogonal curvilinear grid for global climate study applications at MPIfM.

resolution can be concentrated about the pole of smaller radius for regional studies. Implementation of the curvilinear grid is relatively straightforward but does require some additional computational expense. In terms of memory many additional arrays must be added for storage of all terms related to horizontal metrics between both scalar and vector neighboring pairs. The processing time is therefore extended for all operations including horizontal metrics that could otherwise be factored on a regular grid. This condition is omnipresent throughout the model code with the exception of purely vertical operations such as convection.

The standard MPIfM horizontal ocean grid used for climate studies has traditionally been at a spatial resolution approximating spectral truncation T42 with additional equatorial meridional grid refinement (e.g. HOPE, Legutke and Voss, 1999; OPYC, Roeckner at al., 1999). With the 2005 IPCC experiments this has changed to the GROB15 grid with a formal resolution of 1.5 °. In this setup, one pole is located over Greenland and the other over Antarctica. The horizontal resolution gradually varies between 10 km in the Arctic and about 170 km in the Tropics. This arrangement gives highest resolution in the main sinking regions associated with the thermohaline circulation (THC). It has 40 vertical levels with level thickness increasing with depth. Eight layers are within the upper 90 m and 20 are within the upper 600m. Also common is the GR30 setup with a formal resolution of 3.0 °. The models bathymetry was created by interpolation of the ETOPO-5 dataset (Data Announcement 88-MGG-02, Digital relief of the Surface of the Earth. NOAA, National Geophysical Data Center, Boulder, Colorado, 1988) to the model grid. The spectral truncation T42 grid and other grid versions, used for more regionally focused studies, are shown in Marsland *et al.* (2003).

4.4 Bulk Formulae

Simulation with the ocean model requires the specification of heat, fresh water and momentum fluxes at the air/sea interface. Introducing Q_{srf} to denote either Q_w or Q_i in Eqn 6.7 the surface heat balance is given by

$$Q_{srf} = Q^{se}_{srf} + Q^{la}_{srf} + Q^{lw}_{srf} + Q^{sw}_{srf} \tag{4.22}$$

where Q^{se}_{srf}, Q^{la}_{srf}, Q^{lw}_{srf} and Q^{sw}_{srf} are parameterizations of the sensible, latent, long-wave and short-wave heat fluxes, respectively.

Following Oberhuber (1993) the turbulent fluxes are parameterized as

$$\begin{aligned} Q^{se}_{srf} &= \rho_a c_a C_H V_{10m}(T_a - T_{srf}) && (4.23)\\ Q^{la}_{srf} &= \rho_a L_{srf} C_L V_{10m}(q_a - q_{srf}) && (4.24) \end{aligned}$$

Constants ρ_a, c_a and L_{srf} denote the air density, the air specific heat capacity and the latent heat of vaporization or sublimation as appropriate. The 10 m wind speed V_{10m} and 2 m air temperature T_a are taken as prescribed forcing. Variable coefficients of sensible C_H and latent C_L heat transfer are formulated according to Large and Pond (1982). The surface temperature T_{srf} represents either the ocean model upper layer temperature or the sea ice/snow layer skin temperature as in Eqn 6.9. The specific humidity q is a function of water vapor pressure e (units of Pascal) and air pressure p (currently approximated by a constant 1000 hPa in MPI-OM-1).

$$q = (0.623e)/(p - 0.378e) \tag{4.25}$$

At the 2 m level (q_a) the water vapor pressure is a function of dew point temperature, while at the surface (q_{srf}) the saturation vapor pressure is a function of the water or ice/snow surface temperature. In both cases the vapor pressures (e) are calculated according to the formulae of Buck (1981).

The radiant fluxes are parameterized as

$$\begin{aligned} Q^{lw}_{srf} &= \varepsilon\sigma T_a^4(.39 - .05\sqrt{e/100})(1 - \chi n^2) + 4\varepsilon\sigma T_a^3(T_{srf} - T_a) && (4.26)\\ Q^{sw}_{srf} &= (1 - \alpha_{srf})Q^{incsw} && (4.27) \end{aligned}$$

The parameterization of net longwave radiation is based on that of Berliand and Berliand (1952), with the fractional cloud cover n taken as prescribed forcing. The surface thermal emissivity and Stefan-Boltzmann constant are denoted by ε and σ respectively. The saturation vapor pressures e depend on water or sea ice/snow conditions and are also calculated according to the formulae of Buck (1981). The cloudiness factor χ is a modified form of that proposed by Budyko (1974) and is a function of latitude ϕ.

$$\chi = 0.5 + 0.4(\min(|\phi|, 60^\circ))/90^\circ \tag{4.28}$$

The incident shortwave radiation Q^{incsw} is provided as part of the forcing data and implicitly modified by cloud cover in the ERA model. The surface reflectivity α_{srf} in Eqn 4.27 is either that appropriate for open water or takes one of four possible values determined by both the absence or presence of snow and by whether the surface temperature of the sea ice or snow is below 0°C (freezing) or equal to 0°C (melting).

$$\alpha_{srf} = \begin{cases} \alpha_w & \text{open water} \\ \alpha_{im} & \text{sea ice surface and melting} \\ \alpha_{if} & \text{sea ice surface and freezing} \\ \alpha_{sm} & \text{snow surface and melting} \\ \alpha_{sf} & \text{snow surface and freezing} \end{cases} \tag{4.29}$$

Over open water Q_w^{sw} is allowed to penetrate beyond the upper model layer with an exponential decay profile.

The surface freshwater forcing effect on sea level displacement is given by

$$Q_\zeta = P - E + R + G \tag{4.30}$$

where P, E, R and G are fluxes of freshwater in units of m s^{-1} due to precipitation, evaporation, river runoff and glacial meltwater, respectively. For the ocean only simulations considered here P is taken as prescribed forcing, R is taken from the observed mean monthly discharge of the world's 50 largest rivers (Dümenil *et al.*, 1993), and G is neglected. Finally, E is calculated from the latent heat flux (Eqn 4.24) as

$$E = Q_{srf}^{la}/(L_{srf}\rho_w) \tag{4.31}$$

where ρ_w is the density of sea water and L_{srf} is once again the latent heat of vaporization or sublimation as appropriate for water or ice/snow surfaces respectively. The corresponding change in surface model layer salinity ΔS_1 is given by

$$\Delta S_1 = \left(\frac{\Delta z_1 + Q_\zeta}{\Delta z_1}\right) S_1 + (S_1 - S_{obs})/S_R \tag{4.32}$$

where S_R is a time-linear restoring coefficient (units s^{-1}), and S_{obs} is a prescribed observed (monthly or annual) surface salinity (see also equation 4.8). The salinity restoring helps to correct for an unbalanced globally integrated surface freshwater flux, along with errors in the poorly known forcing fields (P, R, and G). The restoring has the positive effect of reducing long term model drift. However, it does damp model variability. A more thorough discussion of the effects of surface relaxation is given by Killworth *et al.*

(2000). The timescale of the restoring is set in the namelist (see table 2.1), restoring to a monthly climatology is an option which requires a CPP switch (Section 2.6.1).

Momentum forcing by atmospheric wind stress over sea ice is applied directly in Eqn. 6.1. For the ocean surface layer velocity $\vec{v_1}$, the wind stress over open water and the ocean-ice stress otherwise result in a change of

$$\frac{\partial \vec{v_1}}{\partial t} = \frac{1-I}{\rho_w \Delta z_1}\vec{\tau_a} + \frac{I}{\rho_w \Delta z_1}\vec{\tau_i} \qquad (4.33)$$

where the ice to ocean stress $\vec{\tau_i} = -\vec{\tau_o}$ from Eqn 6.2.

4.5 OMIP Atmospheric Forcing

The OMIP forcing is a climatological forcing dataset with daily temporal resolution and atmospheric synoptic scale variability. It provides the heat, freshwater and momentum fluxes at the air/sea interface required by the MPI-OM model The forcing data is taken from the German Ocean Model Intercomparison Project (OMIP) climatology, and hereafter is called the OMIP-forcing. OMIP was a collaboration between MPIfM, the German Climate Computing Center (DKRZ) and the Alfred Wegener Institute for Polar and Marine Research. The OMIP project [1] compared the mean state of the global circulation of HOPE with that of the Modular Ocean Model (MOM2; Pacanowski, 1995). Considerable emphasis was also placed on the generation of a surface heat and freshwater flux climatology. The OMIP-forcing was derived from the ECMWF Re-Analysis (ERA; Gibson et al., 1997) 15 year dataset and is documented by Röke (2001).

Three criteria were used for the design of the OMIP-forcing: that the forcing be global; that the forcing resolves timescales associated with weather systems; and that the horizontal resolution of the forcing be somewhat finer than that commonly used in OGCMs. The forcing was constructed by Gaussian filtering of the ERA data to produce a low-frequency and a high-frequency component. The low-frequency components from all years were averaged to create a single mean year. Then the high-frequency component of a single year was superimposed onto the mean year to maintain quasi-realistic synoptic scale space-time variability. The criterion for choosing a particular year for the high-frequency component was to maximize the variance of the OMIP-forcing relative to the variance of the original ERA data. For dynamical consistency it was considered desirable to choose only one high-frequency year for all forcing products (winds, temperatures etc). As different products have maximum variance in different years, the maximization of variance was arbitrarily limited to zonal wind stress. This resulted

[1] www.mpimet.mpg.de/Depts/Klima/natcli/omip/omip_report.html

in the choice of the year 1982, with preservation of 78% of the original ERA variability relative to monthly means.

4.6 NCEP/NCAR Atmospheric Forcing

The NCEP/NCAR reanalysis is a dataset with daily temporal resolution and atmospheric synoptic scale variability (Kalnay, 1996). Daily 2 m air and dew-point temperatures, precipitation, cloud cover, downward shortwave radiation, 10 m wind speed and surface wind stress are available for the full period that is covered by the NCEP/NCAR reanalysis (1948-today). Dew point temperature T_{Dew} is derived from specific humidity q and air pressure p according to Oberhuber (1988).

$$\begin{aligned} e &= q * p/(0.378 * q + 0.623) && (4.34) \\ \alpha &= 1/7.5 * (log_{10} * e/611) && (4.35) \\ T_{Dew} &= (273.16 - (35.86 * \alpha))/(1 - \alpha) && (4.36) \end{aligned}$$

On global average NCEP/NCAR downward short wave radiation is appr. 10% higher than ECMWF reanalysis data and 20% higher than ERBE estimates. To correct for this systematic offset in the NCEP/NCAR downward shortwave radiation a global scaling factor of 0.89 can be applied by using an CPP switch (Section 2.6.1).

5. TIME STEPPING

5.1 Time stepping method

The motions associated with the vertically integrated velocity field (barotropic part) are solved implicitly which damps the external gravity wave mode and thus allows the use of a longer time-step in the integrations. Time stepping in MPI-OM is based on the idea of operator splitting, which is also called *time splitting* or *method of fractional steps*, as described by e.g. Press *et al.* (1988). The method is illustrated with the following example (taken from Press *et al.* (1988)).

Suppose you have an initial value equation of the form

$$\frac{\partial u}{\partial t} = \mathcal{L}u \tag{5.1}$$

where $\mathcal{L}$ is some operator. While $\mathcal{L}$ is not necessarily linear, suppose that it can at least be written as a linear sum of m pieces, which act additively on u,

$$\mathcal{L}u = \mathcal{L}_1 u + \mathcal{L}_2 u + \cdots \mathcal{L}_m u \tag{5.2}$$

Finally, suppose that for each of the pieces, you already know a differencing scheme for updating the variable u from time-step n to time-step $n+1$, valid if that piece of the operator were the only one on the right-hand side. We will write these updating symbolically as

$$\begin{array}{rcl} u^{n+1} & = & \mathcal{U}_1(u^n, \Delta t) \\ u^{n+1} & = & \mathcal{U}_2(u^n, \Delta t) \\ & \cdots & \\ u^{n+1} & = & \mathcal{U}_m(u^n, \Delta t) \end{array} \tag{5.3}$$

Now, one form of operator splitting would be to get from n to $n+1$ by the following sequence of updating:

$$\begin{array}{rcl} u^{n+(1/m)} & = & \mathcal{U}_1(u^n, \Delta t) \\ u^{n+(2/m)} & = & \mathcal{U}_2(u^{n+(1/m)}, \Delta t) \\ & \cdots & \\ u^{n+1} & = & \mathcal{U}_m(u^{n+(m-1)/m}, \Delta t) \end{array} \tag{5.4}$$

This operator splitting is used in *MPI-OM*.

5.2 Timestepping in the Model

The timestepping proceeds by the method of operator splitting or fractional steps as described in Section 5.1. That is, prognostic variables are updated successively in several subroutines. Prescribed forcing is read in at the start of each time-step after which the sea ice dynamics equations are solved by means of functional iteration with under relaxation. Then the sea ice thermodynamics are implemented. The ocean momentum equation is first solved partially for the friction terms and then the advection terms. This results in a partially updated momentum equation which is decomposed into baroclinic and barotropic subsystems. These are solved separately as described below. As in HOPE (Wolff *et al.*, 1997) the prognostic equation for the free surface is solved implicitly, which allows for the model's barotropic time-step to equal the baroclinic time-step.

All subroutine called during one time step are listed in table 5.1 and are described in the following. All symbols are explained in section 8.2 “list of variables”.

5.2.1 octher.f90

boundary forcing

TO DO: The source has to be cleaned up. Encapsulate different parameterizations. CPP flag ”NURMISCH” is identical to ”CDVOCON == 0”. ”NURDIFF” has to be default.

In ice free or non–compact ice covered areas the salinity is changed according to

$$S^{n+1/m}\Big|_{k=1} = S^n\Big|_{k=1} + \Delta t \lambda_S \left(S_{Levitus} - S^n\Big|_{k=1} \right) \cdot (1 - A) \quad (5.5)$$

where A is the sea ice compactness. The salinity is further modified in the sea ice model due to freezing or melting processes. In addition, the sea surface temperature can relaxed to a SST climatology (Levitus *et al.*, 1998a)

$$\Theta^{n+1/m} = \Theta_1^n + \Delta t\, \lambda_T \cdot (\Theta_{Levitus} - \Theta^n\Big|_{k=1}) \quad (5.6)$$

Default for the time constants is $\lambda_S = 3. * 10^{-7}$ and $\lambda_T = 0$. λ_S and λ_T can be modified in the namelist (2.1). Both are weighted with the actual thickness of the first layer with respect to a thickness of 20 m ($\lambda = \lambda_{namelist} * 20\ m/DZW_{(1)}$). Salinity and temperature restoring does only work if MPI-OM is not coupled to ECHAM.

SBR name	Action	Ref.
`octher.f90`	-boundary forcing on salt and temperature -baroclinic pressure -convective adjustment -vertical eddy viscosity and diffusivity coefficients	5.2.1
`-ocice.f90`	Sea ice model as described by Hibler (1979)	ch. 6
`--growth.f90`	Update of ice thickness, compactness, snow depth, upper ocean temperature and salinity due to atmospheric heat and freshwater fluxes.	
`ocwind.f90`	Update ocean velocities with the surface stress (wind forcing).	5.2.2
`octide.f90`	Include forcing from tides (optional).	
`ocmodmom.f90`	Decomposition into barotropic and baroclinic field.	5.2.4
`occlit.f90`	Solution of the baroclinic system.	5.2.6
`bartim.f90`	Calculate sea level hight for the barotropic subsystem with gauss elimination and back-substitution (default).	5.2.7
`troneu.f90`	Calculate sea level hight for the barotropic subsystem with iterative solver (SOR, optional).	5.2.8
`ocvtro.f90`	Calculate new barotropic velocities.	5.2.9
`ocvtot.f90`	Calculate new total velocities u, v and w.	5.2.10
`ocuad.f90`	Advection of momentum analogous to `ocadpo.f90` in u direction.	5.2.11
`ocvad.f90`	Advection of momentum analogous to `ocadpo.f90` in v direction.	
`slopetrans.f90`	Calculate bottom bondary layer (BBL) transport for tracer advection (optional).	5.2.12
`ocadpo.f90`	Computes advection with a second order total variation diminishing (TVD) scheme (Sweby, 1984). Called for temperature and salinity (optional).	5.2.13
`ocadfs.f90`	Compute advection with predictor-corrector scheme or the quick-scheme as proposed by Farrow and Stevens (1995) (optional).	5.2.14
`ocjitr.f90`	Parameterize sub grid-scale eddy-induced tracer transport following Gent *et al.* (1995a) (optional).	5.2.15
`octdiff_base.f90`	Compute tracer-independent matrices for horizontal, isopycnal diffusion.	5.2.16
`octdiff_trf.f90`	Calculate vertical diffusion and harmonic and biharmonic horizontal diffusion (with matrices calculated in `octdiff_base.f90`) for temperature and salinity.	5.2.17
`relax_ts.f90`	3-D restoring of temperature and salinity to initial conditions.	5.2.18
`ocschep.f90`	Harmonic horizontal diffusion of momentum.	5.2.19
`ocvisc.f90`	Vertical diffusion of momentum, bottom friction, biharmonic horizontal momentum diffusion	5.2.20

Tab. 5.1: List of subroutine calls during one time-step.

Discharge from rivers is affecting salinity and sea surface elevation.

$$\begin{aligned} S^{n+1/m}\Big|_{k=1} &= S^n\Big|_{k=1} + \frac{\Delta z_w}{\Delta z_1 + \Delta R_{input}} && (5.7) \\ ZO^{n+1/m} &= ZO + \Delta R_{input} && (5.8) \\ & && (5.9) \end{aligned}$$

Local river input is calculated from discharge data for the given position (see section 2.1):

$$\Delta R_{input} = \frac{discharge * \Delta t}{\Delta x * \Delta y} \qquad (5.10)$$

baroclinic pressure and stability

Hydrostatic pressure and stratification is computed. Potential temperatures Θ are converted to in–situ temperatures T (subroutine `adisitj.f90`) by solving the Gill (2004) formula with a Newton's method. The density is computed in subroutine `rho1j.f90` with the UNESCO (1983) formula.

$$\begin{aligned} p_1 &= g\Delta z_{w1}\, \rho\left(S_1^{n+2/m}, T_1^{n+2/m}, p_{1(ref)}\right) && (5.11) \\ p_k &= p_{k-1} + g\Delta z_{wk}\, \rho\left(S_k^{n+2/m}, T_k^{n+2/m}, p_{k(ref)}\right) && (5.12) \\ \frac{\partial \rho_k}{\partial z} &= \frac{1}{\Delta z_{uk}}(\rho_{k-1} - \rho_k)^{n+2/m} && (5.13) \end{aligned}$$

where the density ρ is calculated using a reference pressure

$$p_{(ref)k} = g\rho_0 h_k \qquad (5.14)$$

where h_k is the depth of layer k.

There are several choices for parameterization of convection currently available in the MPI-OM model.

- Default in cases of unstable stratification is a combination of vertical diffusion and mixing. Other mechanisms are activated with compile flags (table 2.4).
- Compile flag "NURDIFF" disables the default mixing.
- Compile flag "UMKLAP" activates the convective adjustment. Convective adjustment follows Bryan (1969). Traditionally this technique involved the full mixing of vertically adjacent grid cells in the presence of static instability. The MPI-OM formulation is similar but only mixes the upper grid cell with an equivalent thickness of the lower grid

cell. This approach aims to increase the penetrative depth of convection. It is done with only one sweep through the water column per timestep, i.e. for $\rho_z > 0$

$$\Theta_k^{n+3/m} = \frac{\Delta z_{wk-1}\Theta_{k-1}^{n+2/m} + \Delta z_{wk}\Theta_k^{n+2/m}}{\Delta z_{wk-1} + \Delta z_{wk}} \tag{5.15}$$

$$S_k^{n+3/m} = \frac{\Delta z_{wk-1}S_{k-1}^{n+2/m} + \Delta z_{wk}S_k^{n+2/m}}{\Delta z_{wk-1} + \Delta z_{wk}} \tag{5.16}$$

for stable stratification $\rho_z < 0$ (or `ICONVA = 0`)

$$\Theta_k^{n+3/m} = \Theta_k^{n+2/m} \tag{5.17}$$

$$S_k^{n+3/m} = S_k^{n+2/m} \tag{5.18}$$

If convective adjustment was active the stratification array is adjusted.

- Compile flag "PLUME" activates the so-called "PLUME" convection scheme (subroutine `nlopps.f90`). It is based on an original routine by E. Skyllingstad and T. Paluszkiewicz. It is a much more physically based parameterization based on the penetrative plume convection scheme of Paluszkiewicz and Romea (1997). Plume convection was found to significantly improve the deep water characteristics and the simulation of Southern Ocean sea ice in the HOPE model (Kim and Stössel, 2001). However, the penetrative plume convection scheme is computationally quite expensive.

- If the compile flag "NURMISCH" is set, only the Richardson number depending coefficients are used for the diffusion. Richardson number depending coefficients for eddy viscosity and eddy diffusivity are computed every timestep (Pacanowski and Philander, 1981). Additionally, a mixed layer turbulence contribution is included.

$$A_V^{\mathbf{n+1}} = \lambda_V A_V^n + (1-\lambda_V)\left[\frac{A_{V0}}{1+(C_{RA}Ri)^2}\right] + A_B + \delta_{\Delta T}W_T \tag{5.19}$$

$$D_V^{\mathbf{n+1}} = \lambda_V D_V^n + (1-\lambda_V)\left[\frac{D_{V0}}{1+(C_{RD}Ri)^2}\right] + \delta_{\Delta T}W_T \tag{5.20}$$

where $0 \le \lambda \le 1$, A_{V0} and D_{V0} are constant values, A_B is the background mixing (set to 10^{-6} m^2/s^{-1}), W_T is a value for a wind induced mixed layer turbulence (increased turbulent viscosity and diffusivity), $\delta_{\Delta T}$ is a switch which is 1 for a vertical temperature difference to the sea surface temperature smaller than a preset ΔT and 0 elsewhere, and Ri is the local Richardson number

$$Ri = -\frac{g\partial\rho/\partial z}{(\partial u/\partial z)^2 + (\partial v/\partial z)^2} \tag{5.21}$$

Eqs. (5.19) and (5.20) are slightly modified with respect to the original formulations by Pacanowski and Philander (1981).

- If the compile flag "NURMISCH" is not set, in cases of unstable stratification the coefficient is replaced by the mixing term "CDVOCON" which is set in the namelist (table 2.1), if "CDVOCON" is larger that the Richardson coefficient. This leads to a greatly enhanced vertical diffusion in the presence of static instability (e.g. Marotzke, 1991; Klinger at al., 1996). Such an approach avoids the excessive intermediate mixing associated with the traditional adjustment scheme by introducing a timescale associated with the choice of (constant) convective-diffusion coefficient.

5.2.2 ocwind.f90

TO DO: Clean Up, get rid of "SICOMU" and "SICOMV".

Wind stress and ice stress (per unit density) are added to the ocean velocities.

$$u^{n+(1/m)} = u^n + \Delta t \frac{\tau^x_{wind}}{\Delta z_{w1}} \cdot (1-A) + \Delta t \frac{\tau^x_{ice}}{\Delta z_{w1}} \cdot A \tag{5.22}$$

$$v^{n+(1/m)} = v^n + \Delta t \frac{\tau^y_{wind}}{\Delta z_{w1}} \cdot (1-A) + \Delta t \frac{\tau^y_{ice}}{\Delta z_{w1}} \cdot A \tag{5.23}$$

A is the sea ice compactness. The ice stress is computed in subroutine `ocice.f90`.

5.2.3 ocice.f90

The sea ice model is computed in routine `ocice.f90` and its subroutines `growth.f90`, `budget.f90` and `obudget.f90`. A detailed described is given in chapter "Sea Ice Model".

In addition, the penetration of solar radiation is described in `ocice.f90` by a simple vertical profile, constant with latitude and longitude.

$$\frac{I}{I_0} = (1-R)e^{(z/D)} \tag{5.24}$$

The radiation profile is converted to an absorption profile and used to update the temperatures. There is no heat-flux through the bottom of the ocean, so all remaining heat is absorbed in the bottom layer. If the marine biogeochemical model HAMOCC is included, there is an option to calculate the absorption as a function of chlorophyll.

5.2.4 ocmodmom.f90

The velocity fields ($\mathbf{v} = (u, v)$) are decomposed into barotropic (vertically averaged) and baroclinic parts

$$\begin{aligned} \mathbf{V} &= \int_{-H}^{0} \mathbf{v}\, dz && (5.25) \\ \mathbf{v}' &= \mathbf{v} - \frac{1}{H} \int_{-H}^{0} \mathbf{v}\, dz && (5.26) \end{aligned}$$

The definition of $\mathbf{V} = (U, V)$ was chosen to be consistent with the coding, i. e. it is the barotropic transport mode not a velocity.

5.2.5 ocbarp.f90

TO DO: No real function. Fill common block VSE, VZE, USO, UZO. Move to ”bartim.f90” ????

5.2.6 occlit.f90

Solution of the baroclinic system. See section 5.3

5.2.7 bartim.f90

Calculate sea level hight for the barotropic subsystem with an implicit method. The equations are solved with a Gaussian triangulisation method. Requires the subroutines `ocbarp.f90` and `trian.f90`. See section 5.3.

5.2.8 troneu.f90

If the compile flag ”SOR” is set (table 2.4), the equations for the barotropic subsystem are solved by iteration which requires less memory, but considerably more cpu time. Requires the subroutines `itprep.f90` and `trotest.f90`. See section 5.3.

5.2.9 ocvtro.f90

The “new” sea level values are used to calculate new barotropic velocities.

5.2.10 ocvtot.f90

- Baroclinic and barotropic velocities are added to give total velocity fields. The vertical velocity component is calculated from the continuity equation (h is layer thickness)

  $$\begin{aligned}\frac{\partial w^{n+1}}{\partial z} &= -\beta\left[\frac{\partial}{\partial x}(hu^{n+1})+\frac{\partial}{\partial y}(hv^{n+1})\right]\\ &\quad -(1-\beta)\left[\frac{\partial}{\partial x}(hu^{n})+\frac{\partial}{\partial y}(hv^{n})\right]\end{aligned}$$

  (5.27)

  The total velocity field of the new time step is available in this subroutine for further use (e. g. for diagnostics and post–processing).

- A very powerful consistency test for the barotropic implicit system and/or the correct back-substitution is available at this point, i.e. the consistency of sea level change with the vertical velocity at z=0

  $$\frac{\partial \zeta}{\partial t} - w|_{z=0} = 0 \qquad (5.28)$$

  The results of this test should be of round–off–error precision (modified by the effects of twofold differencing, empirically : 10^{-8}).

- Update time levels of velocity and sea level fields.

- Time memory of viscous dissipation according to local rate of strain $\texttt{TURB}_{E/O}$.

5.2.11 ocuad.f90 and ocvad.f90

Advection of momentum analogous to `ocadpo.f90` in u and v direction.

5.2.12 slopetrans.f90

Calculate bottom bondary layer (BBL) transport for tracer advection. For more details see section 4.2.1.

5.2.13 ocadpo.f90

TO DO: BBL and ocadpo are not default. Is it necessary to compute the new vertical velocities in ocadpo (20 times with BGC)?

First, new vertical velocities are calculated, including the BBL transport velocities. Second, advection of scalar traces is computed with a second order total variation diminishing (TVD) scheme (Sweby, 1984). The total variation of a solution is defined as:

$$TV_{(u^{n+1})} \quad = \quad \sum u_{k+1}^{n+1} - u_k^{n+1} \tag{5.29}$$

A difference scheme is defined as being total variation diminishing (TVD) if:

$$TV_{(u^{n+1})} \quad \leq \quad TV_{(u^n)} \tag{5.30}$$

Momentum advection of tracers is by a mixed scheme that employs a weighted average of both central-difference and upstream methods. The weights are chosen in a two step process. First, according to the ratio of the first minus the second spatial derivative over the first spatial derivative of the advected quantity T,

$$r = max\left\{0, \frac{|T'| - |T''|}{|T'|}\right\} \tag{5.31}$$

with $T' = T_{k-1} - T_{k+1}$ and $T'' = T_{k-1} + T_{k+1} - 2 \cdot T_k$. If the second derivative is small usage of central-differencing is save and therefor favorably. In contrast, if the first derivative is small and the second derivative is large there is an extrema in the middle and an upstream scheme is preferred.

In a second step the ratio is weighted with strength of the flow (the time it takes to fully ventilate the grid-box). Water transport in and out of a grid-box is given as:

$$\begin{aligned} U_{in} &= 0.5 \cdot \delta t \cdot \delta x \cdot \delta y \cdot (w_k + |w_{k-1}|) \\ U_{out} &= 0.5 \cdot \delta t \cdot \delta x \cdot \delta y \cdot (|w_{k+1}| - w_{k+1}) \end{aligned} \tag{5.32}$$

The total weight R is defined as:

$$R = min\left\{1, \frac{\delta x \cdot \delta y \cdot \delta z}{U_{in} + U_{out}} \cdot r\right\} \tag{5.33}$$

If the flow is weak and r is small, the magnitude of this ratio is less than 1 and the weights favor usage of central-differencing. With a stronger flow or a larger r the upstream scheme is preferred. The idea is to incorporate the benefit of positive-definiteness of the upstream scheme (and thus limit numerically spurious tracer sources and sinks), while avoiding large implicit numerical diffusion in regions where strong gradients exist in the tracer field.

Advection is computed as follows. Tracer transport in and out of a grid-box is defined as:

$$\begin{aligned} T_{in} &= U_{in} \cdot (1 - R) \cdot T_k + R \cdot 0.5 \cdot (T_k + T_{k-1}) \\ T_{out} &= U_{out} \cdot (1 - R) \cdot T_k + R \cdot 0.5 \cdot (T_k + T_{k+1}) \end{aligned} \tag{5.34}$$

The new tracer concentration T_k^{n+1} is given by the old concentration T_k^n plus tracer in-, and out-fluxes, normalized by the "new" volume of the grid-box (volume plus in-, and out-fluxes of water).

$$
\begin{aligned}
T_k^{n+1} &= (T_k^n \cdot \delta x \cdot \delta y \cdot \delta z + T_{in\,k+1} - T_{in\,k} - T_{out\,k} + T_{out\,k-1}) \cdot \frac{1}{V_{new}} \\
V_{new} &= \delta x \cdot \delta y \cdot \delta z + U_{in\,k+1} - U_{in\,k} - U_{out\,k} + U_{out\,k-1} \qquad (5.35)
\end{aligned}
$$

5.2.14 ocadfs.f90

Compute advection with either a predictor-corrector scheme or with the quick-scheme as proposed by Farrow and Stevens (1995). This routine does not work with BBL transport (5.2.12).

5.2.15 ocjitr.f90

Parameterize sub grid-scale eddy-induced tracer transport following Gent *et al.* (1995a). The advection at the end is done with an upwind scheme.

5.2.16 octdiff_base.f90

Tracer-independent matrices for horizontal, isopycnal diffusion are computed for use in `octdiff_trf.f90`. See also equation 4.20 in chapter 4.

5.2.17 octdiff_trf.f90

Calculate vertical diffusion and harmonic and biharmonic horizontal diffusion (with matrices calculated in `octdiff_base.f90`) for scalar tracers (temperature and salinity).

5.2.18 relax_ts.f90

3-D restoring of temperature and salinity to initial conditions (Levitus *et al.*, 1998b).

5.2.19 ocschep.f90

Harmonic horizontal diffusion of momentum.

5.2.20 ocvisc.f90

TO DO: clean up, TRIDSY to TRIDSY(i,j,k) as in HAMOCC routine octdiff_bgc.f90 ?

Horizontal velocities are modified due to bottom friction, vertical harmonic momentum diffusion and horizontal biharmonic momentum diffusion. The vertical diffusion uses the vertical friction ”avo” calculated in `oocther.f90`.

5.3 Baroclinic and Barotropic Subsystem

Denoting the internal baroclinic pressure divided by reference density as p', and the three dimensional baroclinic velocities as (u', v', w'), the partially updated baroclinic momentum equations can be expressed by

$$\frac{\partial u'}{\partial t} - fv' = \frac{1}{H}\int_{-H}^{\zeta} \frac{\partial p'}{\partial x} dz - \frac{\partial p'}{\partial x} \tag{5.36}$$

$$\frac{\partial v'}{\partial t} + fu' = \frac{1}{H}\int_{-H}^{\zeta} \frac{\partial p'}{\partial y} dz - \frac{\partial p'}{\partial y} \tag{5.37}$$

where x, y, z and t indicate the curvilinear parallel, curvilinear meridional, vertical and temporal dimensions respectively. The local depth is given by H, ζ is the sea surface displacement from the z-coordinate uppermost surface, and f is the Coriolis parameter. Assuming only disturbances of small amplitude (linearization) allows the vertical density advection to be expressed as $\frac{\partial \rho}{\partial t} = -w\frac{\partial \rho}{\partial z}$. This can be combined with the time derivative of the hydrostatic approximation ($\frac{\partial \rho^2}{\partial z \partial t} = \frac{-g}{\rho_0}\frac{\partial \rho}{\partial t}$) to give an equation for the time evolution of the linearized internal baroclinic pressure

$$\frac{\partial^2 p'}{\partial z \partial t} = \frac{wg}{\rho_0}\frac{\partial \rho}{\partial z}. \tag{5.38}$$

Then the baroclinic subsystem is closed with the baroclinic continuity equation.

$$\frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} + \frac{\partial w'}{\partial z} = 0 \tag{5.39}$$

Introducing the superscripts n and $n+1$ to denote old and new time levels respectively, the time discretization of the partially updated linearized baroclinic subsystem can then be written as

$$\begin{aligned} u'^{n+1} - u'^{n} &= \alpha \Delta t \left(fv'^{n+1} + \frac{1}{H}\int_{-H}^{\zeta} p_x'^{n+1} dz - p_x'^{n+1} \right) \\ &+ (1-\alpha)\Delta t \left(fv'^{n} + \frac{1}{H}\int_{-H}^{\zeta} p_x'^{n} dz - p_x'^{n} \right) \end{aligned} \tag{5.40}$$

$$v'^{n+1} - v'^{n} = \alpha\Delta t\left(-fu'^{n+1} + \frac{1}{H}\int_{-H}^{\zeta} p'^{n+1}_{y}dz - p'^{n+1}_{y}\right)$$
$$+(1-\alpha)\Delta t\left(-fu'^{n} + \frac{1}{H}\int_{-H}^{\zeta} p'^{n}_{y}dz - p'^{n}_{y}\right) \quad (5.41)$$
$$p'^{n+1}_{z} - p'^{n}_{z} = \frac{g}{\rho_0}\Delta t\rho_z\left(\beta w'^{n+1} + (1-\beta)w'^{n}\right) \quad (5.42)$$

Here Δt is the model's timestep, and $0 \leq \alpha, \beta \leq 1$ are stability coefficients partially weighting the new velocities to the old velocities. For stability reasons it is required that $\alpha \geq 1 - \alpha$ and $\beta \geq 1 - \beta$. In the semi-implicit case where $\alpha = \beta = \frac{1}{2}$ the system is neutrally stable, but similar to the familiar leapfrog-scheme this tends to produce a computational mode with $2\Delta t$ oscillations. These are suppressed by the choice $\alpha = 0.55$ and $\beta = 0.5$. The system is solved iteratively with the old baroclinic velocities used as a first guess.

For the partially updated barotropic subsystem, the momentum equations are

$$\frac{\partial U}{\partial t} - fV + gH\frac{\partial \zeta}{\partial x} + \int_{-H}^{\zeta} \frac{\partial}{\partial x}p'dz = 0 \quad (5.43)$$
$$\frac{\partial V}{\partial t} + fU + gH\frac{\partial \zeta}{\partial y} + \int_{-H}^{\zeta} \frac{\partial}{\partial y}p'dz = 0 \quad (5.44)$$

where U and V are the partially updated barotropic velocities on the model's curvilinear grid. The barotropic subsystem is closed with a continuity equation accounting for the time derivative of the sea level ζ

$$\frac{\partial \zeta}{\partial t} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = Q_\zeta \quad (5.45)$$

where the forcing term Q_ζ represents the surface freshwater flux (see Eqn 4.30 below). The sea level is partially updated according to Q_ζ before the baroclinic subsystem is solved and so is ignored in the following time discretization.

Denoting the partially updated sea level by ζ', the discretized partially updated barotropic subsystem can then be written as

$$U^{n+1} - U^{n} - f\Delta t\left(\alpha V^{n+1} + (1-\alpha)V^{n}\right)$$
$$+gH\Delta t\left(\alpha\zeta'^{n+1}_{x} + (1-\alpha)\zeta'^{n}_{x}\right) + \Delta t\int_{-H}^{\zeta} p'^{n+1}_{x}dz = 0 \quad (5.46)$$
$$V^{n+1} - V^{n} + f\Delta t\left(\alpha U^{n+1} + (1-\alpha)U^{n}\right)$$
$$+gH\Delta t\left(\alpha\zeta'^{n+1}_{y} + (1-\alpha)\zeta'^{n}_{y}\right) + \Delta t\int_{-H}^{\zeta} p'^{n+1}_{y}dz = 0 \quad (5.47)$$

$$\zeta'^{n+1} - \zeta'^{n} + \Delta t\left(\beta(U^{n+1}_{x} + V^{n+1}_{y}) + (1-\beta)(U^{n}_{x} + V^{n}_{y})\right) = 0 \quad (5.48)$$

where the partial temporal relaxation weights α and β are the same as for the baroclinic subsystem. The set of equations 5.46, 5.47 and 5.48

is rearranged into matrix form and solved by Gaussian elimination with back substitution. Alternatively, when the model dimensions exceed the availability of core computing memory, the matrix can be solved iteratively using successive over-relaxation. For a discussion of how the implicit free surface approach of MPI-OM compares with explicit treatment of the free surface the reader is referred to Griffies *et al.* (2000).

Momentum advection of tracers is by a mixed scheme that employs a weighted average of both central-difference and upstream methods. The weights are chosen according to the ratio of the second spatial derivative over the first spatial derivative of the advected quantity. When the magnitude of this ratio is less than 1 the weights favor usage of central-differencing, and when greater than 1 the upstream scheme is preferred. The idea is to incorporate the benefit of positive-definiteness of the upstream scheme (and thus limit numerically spurious tracer sources and sinks), while avoiding large implicit numerical diffusion in regions where strong gradients exist in the tracer field.

6. SEA ICE MODEL

The sea ice model in routine `ocice.f90` and its subroutines `growth.f90`, `budget.f90` and `obudget.f90` consists of three parts: the dynamics of sea ice circulation, the thermodynamics of sea ice growth and melt and the thermohaline coupling to the ocean model (brine rejection). The following description is mostly identical to Marsland *et al.* (2003) and very similar to the HOPE model description by Wolff *et al.* (1997).

6.1 Sea Ice Dynamics

Sea ice motion is determined by a two-dimensional momentum balance equation.

$$\frac{d\vec{v}_i}{dt} + f(\vec{k} \times \vec{v}_i) = -g\vec{\nabla}\zeta + \frac{\vec{\tau_a}}{\rho_i h_i} + \frac{\vec{\tau_o}}{\rho_i h_i} + \vec{\nabla} \cdot \sigma_{mn} \tag{6.1}$$

Here f, $\vec{k}$, ζ, g, and t are as in Eqn 4.1. Sea ice of thickness h_i and density ρ_i has a velocity $\vec{v}_i$ which responds to wind stress $\vec{\tau_a}$, ocean current stress $\vec{\tau_o}$, and an internal ice stress represented by the two dimensional stress tensor σ_{mn}. It is noted that inclusion of the nonlinear (advective) terms in Eqn 6.1 considerably reduces the model time-step. For the standard MPI-OM setup considered in Section 4.3.3 the time-steps with and without the advective terms are 15 and 36 minutes respectively. To reduce computational expenditure the nonlinear terms were therefore neglected in the simulations considered here. In Eqn. 6.1 the stress terms are in units of N m^{-2}. From above $\vec{\tau_a}$ is taken as prescribed forcing, while from below $\vec{\tau_o}$ is parameterized as

$$\vec{\tau_o} = \rho_w C_W |\vec{v}_1 - \vec{v}_i|(\vec{v}_1 - \vec{v}_i) \tag{6.2}$$

where $\vec{v}_1$ is the upper ocean layer velocity and the constant coefficient of bulk momentum exchange is given by C_W. Currently no turning angles are employed for the atmosphere and ocean/ice stress terms.

The choice of sea ice rheology σ_{mn} determines the way in which ice flows, cracks, ridges, rafts and deforms. Following Hibler (1979) internal sea ice stress is modeled in analogy to a nonlinear viscous compressible fluid obeying the constitutive law

$$\sigma_{mn} = 2\eta\dot{\epsilon}_{mn} + \left\{(\xi - \eta)(\dot{\epsilon}_{11} + \dot{\epsilon}_{22}) - \frac{P_i}{2}\right\}\delta_{mn} \tag{6.3}$$

where $\dot{\epsilon}_{mn}$ is the strain rate tensor and δ_{mn} $(m, n \in \{1, 2\})$ is the Kronecker delta. The internal sea ice pressure P_i is a function of sea ice thickness h_i and subgridscale areal fractional sea ice compactness

$$P_i = P^* h_i e^{-C(1-I)} \tag{6.4}$$

where P^* and C are empirically derived constants. The pressure is related to the nonlinear bulk ξ and shear η viscosities according to:

$$\xi = \frac{P_i}{2\Delta} \; ; \; \eta = \frac{\xi}{e^2} \tag{6.5}$$

$$\Delta = \left[\left(\dot{\epsilon}_{11}^2 + \dot{\epsilon}_{22}^2 \right) \left(1 + \frac{1}{e^2} \right) + 4\frac{\dot{\epsilon}_{12}^2}{e^2} + 2\dot{\epsilon}_{11}\dot{\epsilon}_{22} \left(1 - \frac{1}{e^2} \right) \right]^{\frac{1}{2}} \tag{6.6}$$

Here e is the ratio of the lengths of the principal axes of the yield ellipse (these correspond to the principal components in stress space, i.e. σ_{11} and σ_{22} from Eqn 6.3). The yield ellipse discriminates between linear-viscous (internal) and plastic (boundary) points in stress space, while exterior points cannot be reached. Numerical problems arise when the strain rates are small. Then the Δ in Eqn 6.6 approaches zero, and the viscosities in Eqn 6.5 approach infinity. Following Hibler (1979), the problem is avoided by choosing the viscosities to be a maximum of their function value given in Eqn 6.5, and an empirically chosen maximum value corresponding to the function value when $\Delta = \Delta_{min} = 2.0 \times 10^{-9} s^{-1}$.

6.2 Sea Ice Thermodynamics

Thermodynamics of sea ice involves the determination of the local growth or melt rate at the base of the sea ice and the local melt rate at the surface. To allow for the prognostic treatment of the subgridscale fractional sea ice cover the surface heat balance is solved separately for the ice covered and ice free areas. That is, the net atmospheric heat flux Q_a is weighted according to the open water heat flux Q_w and heat flux over sea ice (or sea ice and snow) Q_i.

$$Q_a = (1 - I)Q_w + IQ_i \tag{6.7}$$

A thermodynamic equilibrium is sought at the interface between the atmosphere and the sea ice/snow layer. An initial solution T^*_{srf} is found for the sea ice/snow layer surface temperature T_{srf} from the energy balance equation

$$Q_i + Q_{cond} = 0. \tag{6.8}$$

The conductive heat flux Q_{cond} within the sea ice/snow layer is assumed to be directly proportional to the temperature gradient across the sea ice/snow layer and inversely proportional to the thickness of that layer (i.e. the so-called zero-layer formulation of Semtner, 1976).

$$Q_{cond} = k_i \frac{(T_{freeze} - T_{srf})}{\tilde{h}_i} \tag{6.9}$$

Here k_i is the thermal conductivity of sea ice, T_{freeze} the freezing temperature of sea water and $\tilde{h}_i$ the effective thermodynamic sea ice thickness of the sea ice/snow layer. This effective thickness is defined to be

$$\tilde{h}_i = \frac{1}{I}\left(h_i + h_s \frac{k_i}{k_s}\right) \tag{6.10}$$

where h_s is the snow layer thickness and k_s is the thermal conductivity of the snow. The ratio of the thermal conductivity of sea ice with respect to that of snow is approximately 7. This means that snow is seven times more effective as an insulator against oceanic heat loss to the atmosphere than sea ice. Hence, even a relatively thin snow cover will result in a much increased effective sea ice thickness. Atmospheric precipitation is converted to snow fall when T_a is below 0°C. Snow loading on the sea ice may result in the submerging of the sea ice/snow interface. In such cases the thickness of the snow draft is converted to sea ice. Since the heat of fusion of snow is slightly greater than the heat of fusion of sea ice this process results in a net heat gain to the sea ice/snow layer. To close the heat balance of the conversion process a small additional amount of snow is also melted.

When the initial solution T^*_{srf} in Eqn 6.9 is greater than 0°C the left-hand side of Eqn 6.8 is recalculated with T_{srf} replaced by 0°C and the resultant energy is used to melt snow and then sea ice from above. In the case where the entire sea ice/snow layer is melted from above any remaining heat is added to Q_w in Eqn 6.7.

To complete the sea ice thermodynamic evolution a heat balance equation must also be applied at the ocean/sea ice and ocean/atmosphere interfaces. The balance equation takes the form

$$\rho_w c_w \Delta z'_1 \frac{\partial \hat{\theta}_1}{\partial t} = (1 - I)Q_w + I(Q_{cond} - h_i \rho_i L_i) \tag{6.11}$$

and is solved for an interim upper layer oceanic temperature $\hat{\theta}_1$. Here c_w is the specific heat capacity of sea water, L_i is the latent heat of fusion of sea ice and $\Delta z'_1$ is the thickness of the upper ocean layer, given by

$$\Delta z'_1 = \Delta z_1 + \zeta - h_{draft} \tag{6.12}$$

where Δz_1 is the defined constant thickness of the ocean model's upper layer. The draft of the sea ice/snow layer h_{draft} is given by

$$h_{draft} = \frac{1}{\rho_w}\left(\rho_i h_i + \rho_s h_s\right). \tag{6.13}$$

where ρ_i and ρ_s are the densities of the sea ice and snow layers respectively. Note that the treatment of a sea ice draft is purely for thermodynamic considerations, and that the ocean momentum balance is not effected. The embedding of sea ice into the upper ocean layer, as opposed to allowing sea ice to exist in multiple ocean layers, is for computational convenience. However, such treatment introduces an upperbound to the sea thickness. In MPI-OM the sea ice draft is not allowed to remain above a local maximum sea ice draft specified as

$$h_{maxdraft} = 0.7(\Delta z_1 + \zeta). \tag{6.14}$$

Any additional sea ice draft is converted to water in a salt (but currently not heat) conserving way. It is noted that this critical sea ice thickness is never reached in simulations using the standard grid of MPI-OM (Section 4.3.3) forced with OMIP (Section 4.5) or NCEP (Section 4.6) surface forcing. For the sea ice undersurface to be in thermal equilibrium with the upper ocean it is required that $T_{melt} \leq \theta_1 \leq T_{freeze}$. To maintain this inequality sea ice/snow is melted when the solution for $\hat{\theta}_1$ from Eqn 6.11 is above T_{melt} and new sea ice is formed when $\hat{\theta}_1$ is below T_{freeze}. For the purposes of this study the effect of salinity on the freezing and melting temperatures is ignored and constant values of T_{freeze} and T_{melt} are used. The model upper layer ocean temperature θ_1 is only allowed to rise above T_{melt} when all of the sea ice/snow layer has been melted within a grid cell. Then the new upper ocean temperature θ_1 and the change in sea ice thickness Δh_i are given by

$$\theta_1 = \hat{\theta}_1 - \min\left\{\frac{h_i \rho_i L_f}{\rho_w c_w \Delta z'_1}, \hat{\theta}_1 - T_{freeze}\right\} \tag{6.15}$$

$$\Delta h_i = \max\left\{(T_{freeze} - \hat{\theta}_1)\frac{\rho_w c_w \Delta z'_1}{\rho_i L_f}, 0\right\}. \tag{6.16}$$

For freezing conditions, the upper ocean temperature and the sea ice thickness change are

$$\theta_1 = T_{freeze} \tag{6.17}$$

$$\Delta h_i = \frac{\hat{\theta}_1 - T_{freeze}}{\rho_i L_f}\rho_w c_w \Delta z'_1. \tag{6.18}$$

Subgridscale thermodynamic processes of sea ice growth and melt are assumed to effect the sea ice compactness within a grid cell in the following ways. When freezing occurs over open water areas the sea ice compactness increases (i.e. leads concentration decreases) at a rate given by

$$\Delta I^{thin} = \max\left\{\frac{\Delta h_i^{thin}(1-I)}{h_o \Delta t}, 0\right\} \tag{6.19}$$

where Δt is the model time-step, $\Delta h_i^{thin} = \Delta t Q_w/(\rho_i L_f)$ is thermohaline coupling to the ocean model the thickness of new sea ice formed and h_o is an arbitrary demarcation thickness (taken to be 0.5 m following Hibler, 1979). When melting of thick sea ice occurs the sea ice compactness decreases (i.e. leads concentration increases) at a rate given by

$$\Delta I^{thick} = \min\left\{\frac{\Delta h_i^{thick} I}{2h_i \Delta t}, 0\right\} \tag{6.20}$$

where Δh_i^{thick} is the change in sea ice thickness due to the melting. This formulation is based on the assumption that sea ice thickness within a grid cell has a uniform distribution between 0 and $2h_i$. The change in compactness of sea ice due to thermodynamic lead opening and closing is then calculated as the sum of both these terms.

$$\frac{\partial I}{\partial t} = \Delta I^{thin} + \Delta I^{thick} \tag{6.21}$$

6.3 Update of Salinity (Brine Rejection)

Completion of the sea ice thermohaline coupling to the ocean model requires consideration of salt and fresh water exchanges during sea ice growth and melt. Sea ice is assumed to have a constant salinity independent of it's age and denoted by S_{ice}. While multi-year Arctic sea ice has a salinity of around 3 psu, thinner ice in both hemispheres has a much higher salinity (Cox and Weeks, 1974; Eicken, 1992). For the simulations with the standard grid (Section 4.3.3) an intermediate value of 5 psu representing this global diversity has been chosen. The ocean model's upper layer salinity S_1 is changed by an amount ΔS due to the surface fresh water flux (modified by snow fall which accumulates on top of the sea ice) and due to sea ice growth or melt, according to:

$$(S_1 + \Delta S)\,\Delta z'^{old} + \frac{\rho_i h_i^{old}}{\rho_w} S_{ice} = S_1 \Delta z'^{new} + \frac{\rho_i h_i^{new}}{\rho_w} S_{ice} \tag{6.22}$$

Here $\Delta z'^{old}$ is the upper ocean layer thickness accounting for sea surface elevation and sea ice draft as in Eqn 6.12, and also for the atmospheric precipitation minus evaporation. $\Delta z'^{new}$ is $\Delta z'^{old}$ modified by the new sea ice draft due to melt or growth, and $h_i^{new} - h_i^{old}$ is the amount of sea ice growth (if positive) or melt (if negative).

6.4 Subroutines

- Subroutine `growth.f90` calculates the ice thickness, compactness, snow depth, upper ocean temperature and salinity due to atmospheric heat and freshwater fluxes.

- Subroutine `budget.f90` calculates the growth rates for the ice covered part of a grid cell with standard bulk formulas. First, the snow or ice skin temperature and the surface residual heat flux at the interface to the atmosphere and at the bottom are computed. Second, surface melt of snow or ice and bottom ablation or aggregation are deduced.

- Subroutine `obudget.f90` calculates growth rates of new ice in the ice free part of a grid cell with bulk formulas or the net atmospheric heat flux at the water-atmosphere interface.

7. DIAGNOSTIC AND MEAN OUTPUT

MPI-OM generates a large number of output files. Most of them are mean values of ocean properties (temperate, salinity ...). In addition, there is mean output for diagnostic and flux variables, as well as grid, forcing or coupling (ECAHM) information. Time averaging is controlled by the namelist (see 2.1) variable `IMEAN`. The number of output can be controlled with CPP switches (see 2.6.1). Each code is written into a separate file named `fort.`*`unit`* according to the unit the file is written to. The file format is `EXTRA`. Tables 7.2 to 7.7 given an overview over all possible output codes. This chapter deals with the structure of the MPI-OM output and the coding behind some of the diagnostic variables of which the meaning might not be strait-forward.

SBR name	Action	CPP flag
`mo_commodiag.f90`	Define variables for the mean diagnostic output.	
`mo_mean.f90`	Define all variables for mean model output.	MEAN
`mo_commconv.f90`	Define variables for the mean depth of convection	CONVDIAG
`diag_ini.f90`	Diagnostic output for the time-series is initialized.	
	begin timestepping	
after `ocvtot.f90`	new total velocities u, v and w are available	
`wrte_mfl.f90`	write divergence free velocity just after the new velocities are computed	MFLDIAG
`diagnosis.f90`	prepare diagnostic output (see 7.1.1)	
after `octdiff_trf.f90`	advection and diffusion of tracers is done	
`wrte_mean.f90`	write mean output	MEAN
	end of one day	
`wrte_amlddiag.f90`	write max. monthly mixed layer depth	AMLDDIAG
	end of one month	
	end of one year	
`wrte_konvdiag.f90`	write convection overturning	KONVDIAG
`wrte_gridinfo.f90`	write grid information	GRIDINFO

Tab. 7.1: List of diagnostic and output subroutine calls in the order in which they are called.

7.1 Subroutines

All subroutine calls for diagnostic and mean output in successive order are listed in table 5.1. A complete list of subroutines called during one time-step. is given in table 5.1.

7.1.1 diagnosis.f90

Compute diagnostic output for the time-series and mean output. Write the time-series once a day.

mixed layer depth

Mixed layer depth (variable ZMLD) is computed based on the density difference criterion $\sum_1^k \delta\rho_{insitu}(z) < 0.125 kg/m^3$; the depth where the density has increased by 0.125 kg/m^3 as compared to the value in the surface box.

The monthly maximum of the variable ZMLD is stored in the variable AMLD (see table 7.2).

barotropic stream function

The vertically integrated, horizontal barotropic stream function (variable PSIUWE) is computed as $\Psi_{(i,j)} = \sum_k \sum_2^j \delta x \cdot \delta z \cdot v_y$. The stream function is also used for the time-series output for the golf stream and the Kuroshio, as well as for Banda, Drake and Bering strait transports.

7.1.2 wrte_mean.f90

Compute the heat flux and the sea ice transport in x- and y-direction. Average (daily, monthly or yearly) and write the mean and diagnostic output.

7.1.3 wrte_mfl.f90

Average (daily, monthly or yearly) and write the u- and v-velocities just after the new total velocities u, v and w have computed in `ocvtot.f90`. At this point in time, the velocity field is divergence free. Velocities written in `wrte_mean.f90` at the end of the time-step have already been updated by various processes such as the slope-convection.

Code	Content	L.	Variable	Unit	fort.x	CPP	CP
2	temperature	40	THO (P,R)	C	71	M	(x)
5	salinity	40	SAO (P,R)	psu	72	M	(x)
3	x velocity	40	UKO (P,R)	m/s	73	M	( )
4	y velocity	40	VKE (P,R)	m/s	74	M	( )
303	x velocity (divergence free)	40	UKOMFL (P,R)	m/s	303	M	(x)
304	y velocity (divergence free)	40	VKEMFL (P,R)	m/s	304	M	(x)
8	insitu density	40	RHO (D,?)	kg/m**3			
6	pressure	40	PO (?)	Pa			
67	freshwater flux by restoring	1	EMINPO	m/s	79	M	
70	total heat-flux	1	FLUM	W/m**2	84	M	(x)
79	total freshwater flux	1	PEM	m/s	85	M	(x)
13	ice thickness	1	SICTHO (P,R)	m	86	M	(x)
15	ice compactness	1	SICOMO (P,R)	frac.	87	M	(x)
35	x ice velocity	1	SICUO (P,R)	m/s	88	M	(x)
36	y ice velocity	1	SICVE (P,R)	m/s	89	M	(x)
141	snow thickness	1	SICSNO (P,R)	m	136	M	(x)
176	heat flux short-wave	1	QSWO (F)	W/m**2	137	M	
177	heat flux long-wave	1	QLWO (F)	W/m**2	138	M	
147	heat flux latent	1	QLAO (F)	W/m**2	139	M	
146	heat flux sensible	1	QSEO (F)	W/m**2	140	M	
65	net freshwater flux + runoff	1	PRECO (F)	m/s	141	M	(x)
1	sea-level	1	ZO (P,R)	m	82	M	(x)
82	sea-level change	1	Z1O	m			
27	hor. bar. stream-function	1	PSIUWE (D)	Sv	143	M	(x)
83	max. monthly mixed layer depth	1	AMLD (D)	m	142	M	(x)
142	sea-ice transport x	1	SICTRU	m**2/s	147	M	(x)
143	sea-ice transport y	1	SICTRV	m**2/s	148	M	(x)
183	mixed layer depth (SJ)	1	zmld	m			
305	River Runoff	1	rivrun	m/s	305	M	
158	mon. mean depth of convection	1	TMCDO	level			

Tab. 7.2: Code Table for MPI-OM mean output.
(x): Reasonable in the coupled setup.
M : Can be switch on with CPP flag MEAN.

7.2 Output Files

The following tables give an overview of all available output fields with variable names, units and the EXTRA format code numbers. Most output is optional an can be switched on with CPP compile flags (table 2.5).

Code	Content	L.	Variable	Unit	fort.x	Cpp	CP
7	ver. velocity	40	WO (P,R)	m/s	146	M D	(x)
69	depth of convection	1	KCONDEP (D)	level	90	K	(x)
110	vertical momentum diffusion	40	AVO	m**2/s	144	M D	(x)
111	vertical T,S diffusion	40	DVO	m**2/s	145	M D	(x)
612	wind mixing	40	WTMIX	m**2/s	245	M D	(X)
207	GM vertical velocity	40	WGO	m/s	246	M G	(x)
?	GM BolX	1	BOLX	?	159	M G	(x)
?	GM BolY	1	BOLY	?	160	M G	(x)

Tab. 7.3: Code Table for MPI-OM mean diagnostic output.
(x): Reasonable in the coupled setup.
M : Can be switch on with CPP flag MEAN.
D : Can be switch on with CPP flag DIFFDIAG.
K : Can be switch on with CPP flag KONVDIAG.
G : Can be switch on with CPP flag GRIDINFO.

Code	Content	L.	Variable	Unit	fort.x	Cpp	CP
92	surface air temperature	1	TAFO (F)	C	131	M F	
164	cloud cover	1	FCLOU (F)		132	M F	
52	surface u-stress	1	TXO (F)	Pa/1025.	149	M F	(X)
53	surface v-stress	1	TYE (F)	Pa/1025.	150	M F	(X)
260	prescr. precipitation	1	FPREC (F)	m/s	133	M F	
80	downward short-wave rad.	1	FSWR (F)	W/m**2	134	M F	
81	dew-point temperature	1	FTDEW (F)	K	135	M F	
171	10m wind-speed	1	FU10 (F)	m/s	130	M F	

Tab. 7.4: Code Table for MPI-OM mean forcing output.
(x): Reasonable in the coupled setup.
M : Can be switch on with CPP flag MEAN.
F : Can be switch on with CPP flag FORCEDIAG.

Code	Content	L.	Variable	Unit	fort.x	Cpp	CP
247	heat-flux sw over water	1	DQSWO	W/m**2	247	M H	
248	heat-flux lw over water	1	DQLWO	W/m**2	248	M H	
249	heat-flux se over water	1	DQSEO	W/m**2	249	M H	
250	heat-flux la over water	1	DQLAO	W/m**2	250	M H	
251	heat-flux net over water	1	DQTHO	W/m**2	251	M H	
252	heat-flux sw over seaice	1	DQSWI	W/m**2	252	M H	
253	heat-flux lw over seaice	1	DQLWI	W/m**2	253	M H	
254	heat-flux se over seaice	1	DQSEI	W/m**2	254	M H	
255	heat-flux la over seaice	1	DQLAI	W/m**2	255	M H	
256	heat-flux net over seaice	1	DQTHI	W/m**2	256	M H	
257	Equi. temp over seaice	1	DTICEO	K	257	M H	

Tab. 7.5: Code Table for MPI-OM mean heat-flux output.
(x): Reasonable in the coupled setup.
M : Can be switch on with CPP flag MEAN.
F : Can be switch on with CPP flag TESTOUT_HFL.

Code	Content	L.	Variable	Unit	fort.x	Cpp	CP
172	landseamask (pressure points)	40	WETO		93	G	(x)
507	landseamask (vector points v)	40	AMSUE		212	G	(x)
508	landseamask (vector points u)	40	AMSUO		213	G	(x)
84	depth at pressure points	1	DEPTO	m	96	G	(x)
484	depth at vector points (u)	1	DEUTO	m	196	G	(x)
584	depth at vector points (v)	1	DEUTE	m	197	G	(x)
184	level thickness (vector u)	40	DDUO	m	198	G	(x)
284	level thickness (vector v)	40	DDUE	m	199	G	(x)
384	level thickness (pressure)	40	DDPO	m	200	G	(x)
85	grid distance x	1	DLXP	m	151	G	(x)
86	grid distance y	1	DLYP	m	152	G	(x)
185	grid distance x (vector u)	1	DLXU	m	201	G	(x)
186	grid distance y (vector u)	1	DLYU	m	202	G	(x)
285	grid distance x (vector v)	1	DLXV	m	203	G	(x)
286	grid distance y (vector v)	1	DLYV	m	204	G	(x)
54	latitude in radians	1	GILA	rad	94	G	(x)
55	longitude in radians	1	GIPH	rad	97	G	(x)
354	latitude in degrees (pressure)	1	ALAT	deg	205	G	(x)
355	longitude in degrees (pressure)	1	ALON	deg	206	G	(x)
154	latitude in degrees (vector u)	1	ALATU	deg	208	G	(x)
155	longitude in degrees (vector u)	1	ALONU	deg	209	G	(x)
254	latitude in degrees (vector v)	1	ALATV	deg	210	G	(x)
255	longitude in degrees (vector v)	1	ALONV	deg	211	G	(x)

Tab. 7.6: Code Table for MPI-OM grid information output.
(x): Reasonable in the coupled setup.
G : Can be switch on with CPP flag GRIDINFO.

Code	Content	L.	Variable	Unit	fort.x	Cpp	CP
270	oasis net heat flux water	1	AOFLNHWO	W/m**2	270	O M OFD	(x)
271	oasis downward short wave	1	AOFLSHWO	W/m**2	271	O M OFD	(x)
272	oasis residual heat flux ice	1	AOFLRHIO	W/m**2	272	O M OFD	(x)
273	oasis conduct. heat flux ice	1	AOFLCHIO	W/m**2	273	O M OFD	(x)
274	oasis fluid fresh water flux	1	AOFLFRWO	m/s	274	O M OFD	(x)
275	oasis solid fresh water flux	1	AOFLFRIO	m/s	275	O M OFD	(x)
276	oasis wind stress water x	1	AOFLTXWO	Pa/1025	276	O M OFD	(x)
277	oasis wind stress water y	1	AOFLTYWO	Pa/1025	277	O M OFD	(x)
278	oasis wind stress ice x	1	AOFLTXIO	Pa/1025	278	O M OFD	(x)
279	oasis wind stress ice x	1	AOFLTYIO	Pa/1025	279	O M OFD	(x)
280	oasis wind speed	1	AOFLWSVO	m/s	280	O M OFD	(x)

Tab. 7.7: Code Table for MPI-OM/ECHAM5 mean coupler output.
(x): Reasonable in the coupled setup.
M : Can be switch on with CPP flag MEAN.
O : Can be switch on with CPP flag OASIS.
OFD : Can be switch on with CPP flag OASIS_FLUX_DAYLY.

8. APPENDIX

8.1 Appendix A Auxiliary Subroutines

8.1.1 trian.f90

Called from `mpiom.f90`. Matrix triangulation to solve a set of linear equations by Gaussian elimination.

8.1.2 mo_parallel

Variables and subroutines for the parallelization. Including the subroutine `bounds_exch` which updates the cyclic boundaries of arrays

8.1.3 mo_mpi

Variables and subroutines for the MPI parallelization.

8.1.4 mo_couple

Variables and subroutines for the coupling to ECHAM.

8.1.5 rho1j.f90

Calculates the density $\rho(S,T,p)$, using the equation of state defined by the Joint Panel on Oceanographic Tables and Standards (UNESCO, 1981). See Gill (1982, appendix A3.1).

This subroutine calculates the density at all scalar "i" points for a given "j", i.e. even on land points. As a consequence the values of T and S on land points should have sensible magnitudes (e.g. S must be positive !).

8.1.6 rho2.f90

Calculates the density $\rho(S, T, p)$, using the equation of state defined by the Joint Panel on Oceanographic Tables and Standards (UNESCO, 1981). See Gill (1982, appendix A3.1).

`rho2.f90` returns the density of one location only and is used to compute the reference stratification.

8.1.7 adisit1.f90

Transformation from potential temperature Θ to in situ temperature T for use in the UNESCO equation of state.

8.1.8 adisitj.f90

Transformation from potential temperature Θ to in situ temperature T for use in the UNESCO equation of state. This subroutine calculates the temperature at all scalar ”i” points for a given ”j”, i.e. even on land points.

8.1.9 diagnosis.f90

Computes the total vertical mass fluxes, layer mean temperatures and kinetic energies in each time-step. Temperatures and kinetic energies are stored for diagnostics in a time series file.

8.1.10 nlopps.f90

Computes the so-called ”PLUME” convection scheme based on an original routine by E. Skyllingstad and T. Paluszkiewicz. Activated by the compile flag ”PLUME”.

8.2 Appendix B List of Variables

8.2.1 Model Constants and Parameters

Symbol	Description	Value
α_{if}	freezing sea-ice albedo	0.75
α_{im}	melting sea-ice albedo	0.70
α_{sf}	freezing snow albedo	0.85
α_{sm}	melting snow albedo	0.70
α_w	sea water albedo	0.10
λ	wind mixing stability parameter	0.03 kg m^{-3}
ε	emissivity of sea water	0.97
ρ_a	density of air	1.3 kg m^{-3}
ρ_i	density of sea-ice	910 kg m^{-3}
ρ_s	density of snow	330 kg m^{-3}
ρ_w	density of sea water	1025 kg m^{-3}
σ	Stefan-Boltzmann constant	5.5×10^{-8} W m^{-2} K^{-4}
Λ_V	eddy viscosity relaxation coefficient	0.6
Λ_D	eddy diffusivity relaxation coefficient	0.6
c_a	specific heat capacity of air	1004 J kg^{-1} K^{-1}
c_w	specific heat capacity of sea water	4.0×10^{3} J kg^{-1} K^{-1}
e	ratio of principle axis of yield ellipse	2.0
g	acceleration due to gravity	9.81 m s^{-2}
k_i	thermal conductivity of sea ice	2.17 W m^{-1} K^{-1}
k_s	thermal conductivity of snow	0.31 W m^{-1} K^{-1}
z_0	wind mixing penetration depth	40 m
A_b	PP background vertical viscosity	1.0×10^{-4} m^2 s^{-1}
A_w	PP wind mixing	5.0×10^{-4} m^2 s^{-1}
A_{VO}	PP vertical viscosity parameter	1.0×10^{-2} m^2 s^{-1}
B_H	biharmonic horizontal viscosity	1.1×10^{-6} s^{-1} $\times$ $(\Delta x^4, \Delta y^4)$
BBL_{max}	maximum BBL thickness	500 m
C	empirical internal ice pressure const.	20
C_{RA}	PP viscosity tuning constant	5.0
C_{RD}	PP diffusivity tuning constant	5.0
C_W	ocean-ice stress bulk transfer	0.0045
D_b	PP background vertical diffusivity	1.0×10^{-5} m^2 s^{-1}
D_H	harmonic horizontal diffusion	2.5×10^{-3} m s^{-1} $\times$ $(\Delta x, \Delta y)$
D_w	PP wind mixing	5.0×10^{-4} m^2 s^{-1}
D_{VO}	PP vertical diffusivity parameter	1.0×10^{-2} m^2 s^{-1}
L_f	latent heat of fusion	2.5×10^{6} J kg^{-1}
L_s	latent heat of sublimation	2.834×10^{6} J kg^{-1}
L_v	latent heat of vaporization	2.5×10^{6} J kg^{-1}
P^*	empirical internal ice pressure const.	5000 N m^{-1}
S_{ice}	salinity of sea-ice	5 psu
T_{freeze}	freezing temperature of sea water	-1.9°C
T_{melt}	melting temperature of sea ice/snow	0°C
W_T	wind mixing amplitude parameter	5.0×10^{-4} m^2 s^{-1}

Tab. 8.1: Constants and parameters used in the GROB3 setup of the ocean/sea ice model. Table is taken from Haak (2004).

8.2.2 Global Parameters

Parameter	Description	GROB3
IE	number of zonal grid points	122
JE	number of meridional grid points	101
KE	number of vertical levels	40

Tab. 8.2: Global values for version GROB3

8.2.3 Model Variables

TO DO: Name all vector point defined variables in the U/V convention.

U/V denote the vector points in u and v direction on the C-grid. For every array that is indexed by U/V there exist two arrays. Sometimes, U/V points are still called E/O which used to denotes the EVEN/ODD parity of the north–south index of the HOPE model E-grid.

Name	Symbol	Description
AVO	A_V	vertical eddy viscosity
DVO	D_V	vertical eddy diffusivity
WO	w	vertical velocity
WPO	$p'^{n+1,l}$	pressure in baroclinic subsystem iteration

Tab. 8.3: 3–dimensional arrays (IE,JE,KE+1)

Name	Symbol	Description
$\mathtt{DDU}_{E/O}$	d_{uk}	layer thicknesses at vector points
DDPO	d_{wk}	layer thicknesses at scalar points
PO	p	pressure
SAO	S	salinity
STABIO	$-\partial\rho/\partial z$	negative of vertical density gradient (stability)
THO	Θ	potential temperature
UKO	u', u	baroclinic zonal velocity / total zonal velocity
VKE	v', v	baroclinic merid. velocity / total merid. velocity
UOO	u	total zonal velocity
VOE	v	total meridional velocity

Tab. 8.4: 3–dimensional arrays (IE,JE,KE)

Name	Symbol	Description
DEPTO	H_p	total depth at scalar points
DEUT$_{E/O}$	H_u	total depth at vector points
DEUTI$_{E/O}$	$1./H_u$	inverse of total depth at vector points
DLX$_{U/V}$	Δx_ζ	zonal distance of 2 vector-points
DLY$_{U/V}$	Δy	meridional distance of 2 vector/scalar points
DPY$_{E/O}$	$\Delta t/\Delta y$	
DTDXP$_{E/O}$	$\Delta t/\Delta x_\zeta$	
DTDXU$_{E/O}$	$\Delta t/\Delta x_u$	
DTDYO	$\Delta t/\Delta y$	
EMINPO	$E-P$	Evaporation minus precipitation
PX$_{E/O}$IN	$\int p_x dz$	vertically integrated zonal pressure gradient
PY$_{E/O}$IN	$\int p_y dz$	vertically integrated meridional pressure gradient
TXO	τ^x	wind-stress zonal component
TYE	τ^y	wind-stress meridional component
U1$_{E/O}$	U	barotropic zonal velocity (also $(1-\beta)U$)
V1$_{E/O}$	V	barotropic meridional velocity (also $(1-\beta)V$)
USO		$\beta(\Gamma_U + f\alpha\Delta t\Gamma_V)$ see 5.2.9 and 5.2.10
VSE		$\beta(\Gamma_V - f\alpha\Delta t\Gamma_U)$ see 5.2.9 and 5.2.10
UZO	Γ_U	see 5.2.9 and 5.2.8
VZE	Γ_V	see 5.2.9 and 5.2.8
Z1O	ζ^n	sea level old time step
ZO	ζ^{n+1}	sea level new time step

Tab. 8.5: 2–dimensional arrays (IE,JE)

Name	Dimension	Symbol	Description
ALAT	JE*2	ϕ	latitudes
ALONG	IE*2+6	λ	longitudes
DZ	KE+1	Δz_u	vertical distances vector points
DI	KE+1	$1/\Delta z_u$	
DZW	KE	Δz_w	vertical distances vertical velocity points
DWI	KE	$1/\Delta z_w$	
SAF	KE	S_{ref}	reference salinity
TAF	KE	Θ_{ref}	reference temperature
TIESTU	KE+1		vertical level of vector/scalar points see section 4.3.2
TIESTW	KE+1		vertical level of vertical velocity points see section 4.3.2
TRIDSY	IE,KE,3		coefficients of vertical tridiagonal system
NUM	IE,JE*2		consecutively numbered oceanic scalar points
PGL	2*KBB+1,ILL	A	barotropic system matrix and elimination factors

Tab. 8.6: variables with various dimensions

Name	Symbol	Description
HIBET$_{E/O}$	η	nonlinear shear viscosity of ice
HIBZET$_{E/O}$	ζ	nonlinear bulk viscosity of ice
SICOMO	A	sea ice compactness
SICPO	P_I	internal sea ice pressure
SICSHO		sea ice velocity shear
SICTHO	h_I	sea ice thickness
SICUO	u_I	zonal sea ice velocity
SICVE	v_I	meridional sea ice velocity

Tab. 8.7: Sea ice model variables (2–D)

8.3 Appendix C File Formats

- EXTRA :
  EXTRA is a binary format which was developed at the University of Hamburg. It contains the describing variables date, code, level and field size. It does not contain any grid description. EXTRA is described in detail in the DKRZ Technical Report No. 6.

- NetCDF :
  NetCDF (network Common Data Form) is an interface for array-oriented data access and a library that provides an implementation of the interface. The netCDF library also defines a machine-independent format for representing scientific data. For a detailed description please refer to:

  `http://my.unidata.ucar.edu/content/software/netcdf/index.html`

- ASCII :
  ASCII is an abbreviation for American Standard Code for Information Interchange, developed through the American National Standards Institute. ASCII is a scheme of binary notation for machine-readable data.

BIBLIOGRAPHY

Arakawa, A. and V. R. Lamb, 1977. Computational design of the basic dynamical processes of the UCLA general circulation model. *Methods Comput. Phys.*, **17**, 173–265.

Beckmann, A. and R. Döscher, 1997. A method for improved representation of dense water spreading over topography in geopotential-coordinate models. *J. Phys. Oceanogr.*, **27**, 581–591.

Berliand, M. E. and T. G. Berliand, 1952. Determining the net long-wave radiation of the earth with consideration of the effects of cloudiness. Isv. Akad. Nauk. SSSR Ser. Geofis. 1.

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

Buck, A. L., 1981. New equations for computing vapor pressure and enhancement factor. *J. Appl. Met.*, **20**, 1527–1532.

Budyko, M. I., 1974. *Climate and life*. Academic Press, Int. Geophys. Ser.

Campin, J. M. and H. Goosse, 1999. Parameterization of density-driven downsloping flow for a coarse-resolution ocean model in z-coordinate. *Tellus*, **51A**, 412–430.

Cox, G. F. N. and W. F. Weeks, 1974. Salinity variations in sea ice. *J. Glaciol.*, **13**, 109–120.

Dümenil, L., K. Isele, H.-J. Liebscher, U. Schröder, M. Schumacher and K. Wilke, 1993. Discharge data from 50 selected rivers for GCM validation. Report 100, Max-Planck-Institut für Meteorologie, Hamburg, Germany.

DYNAMO Group, 1997. DYNAMO Dynamics of North Atlantic Models: Simulation and assimilation with high resolution models. Report 294, Institut für Meereskunde, Kiel, Germany.

Eicken, H., 1992. Salinity profiles of Antarctic sea ice: Field data and model results. *J. Geophys. Res.*, **97**, 15545–15557.

Ezer, T. and G. L. Mellor, 1997. Simulations of the Atlantic Ocean with a free surface sigma coordinate ocean model. *J. Geophys. Res.*, **102**, 15647–15657.

Farrow, D. and D. Stevens, 1995. A new tracer advection scheme for bryan and cox type ocean general circulation models. *J. Phys. Oceanogr.*, **25**(7), 1731–1741.

Gent, P., J. Willebrand, T. McDougall and J. McWilliams, 1995a. Parameterizing eddy-induced tracer transport in ocean circulation models. *J. Phys. Oceanogr.*, **25**, 463–474.

Gent, P. R., J. Willebrand, T. McDougall and J. C. McWilliams, 1995b. Parameterizing eddy-induced tracer transports in ocean circulation models. *J. Phys. Oceanogr.*, **25**, 463–474.

Gibson, J. K., P. Kållberg, S. Uppala, A. Hernandez, A. Nomura and E. Serrano, 1997. ERA description. ECMWF Re-analysis Proj. Rep. Ser. 1, Eur. Cent. for Medium-Range Weather Forecasts, Reading, England.

Gill, A., 2004. In *Atmosphere-Ocean Dynamics* (edited by W. Donn), Volume 30 of *International Geophysics Series*. Academic Press, Orlando, Florida, U.S.A.

Gouretski, V. V. and K. P. Koltermann, 2004. Woce global hydrographic climatologyges. Technical Report 35, Bundesamt fr Seeschifffahrt und Hydrographie, Hamburg, Germany.

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

Griffies, S. M., C. Böning, F. O. Bryan, E. P. Chassignet, R. Gerdes, H. Hasumi, A. Hirst, A.-M. Treguir and D. Webb, 2000. Developments in ocean climate modelling. *Ocean Modelling*, **2**, 123–192.

Haak, H., 2004. *Simulation of Low-Frequency Climate Variability in the North Atlantic Ocean and the Arctic*, Volume 1. Max Planck Institute for Meteorology.

Hibler, W. D., 1979. A dynamic thermodynamic sea ice model. *J. Phys. Oceanogr.*, **9**, 815–846.

Kalnay, E. et al., 1996. The NCEP/NCAR 40 year-reanalysis project. *Bull. Amer. Meteor. Soc.*, **77**, 437–470.

Killworth, P. D. and N. R. Edwards, 1999. A turbulent bottom boundary layer code for use in numerical ocean models. *J. Phys. Oceanogr.*, **29**, 1221–1238.

Killworth, P. D., D. A. Smeed and A. J. G. Nurser, 2000. The effects on ocean models of relaxation toward observations at the surface. *J. Phys. Oceanogr.*, **30**, 160–174.

Kim, S.-J. and A. Stössel, 2001. Impact of subgrid-scale convection on global thermohaline properties and circulation. *J. Phys. Oceanogr.*, **31**, 656–674.

Klinger, B. A., J. Marshall and U. Send, 1996. Representation of convective plumes by vertical adjustment. *J. Geophys. Res.*, **101**, 18175–18182.

Large, W. G. and S. Pond, 1982. Sensible and latent heat flux measurements over the ocean. *J. Phys. Oceanogr.*, **12**, 464–482.

Legutke, S. and E. Maier-Reimer, 2002. The impact of downslope water-transport parameterization in a global ocean general circulation model. *Clim. Dyn.*, **18**, 611–623.

Legutke, S. and R. Voss, 1999. The Hamburg atmosphere–ocean coupled circulation model ECHO–G. Technical Report 18, German Climate Computer Center (DKRZ), Hamburg, Germany.

Levitus, S., T. P. Boyer, M. E. Conkright, T. O'Brien, J. Antonov, C. Stephens, L. Stathoplos, D. Johnson and R. Gelfeld, 1998a. World Ocean Database 1998: Volume 1: Introduction. NOAA Atlas NESDIS 18, Ocean Climate Laboratory, National Oceanographic Data Center, U.S. Gov. Printing Office, Wash., D.C.

Levitus, S., T. P. Boyer, M. E. Conkright, T. O'Brien, J. Antonov, C. Stephens, L. Stathoplos, D. Johnson and R. Gelfeld, 1998b. World Ocean Database 1998: Volume 1: Introduction. NOAA Atlas NESDIS 18, Ocean Climate Laboratory, National Oceanographic Data Center, U.S. Gov. Printing Office, Wash., D.C.

Marotzke, J., 1991. Influence of convective adjustment on the stability of the thermohaline circulation. *J. Phys. Oceanogr.*, **21**, 903–907.

Marsland, S. J., H. Haak, J. H. Jungclaus, M. Latif and F. Roeske, 2003. The Max-Planck-Institute global ocean/sea ice model with orthogonal curvilinear coordinates. *Ocean Modelling*, **5**, 91–127.

Oberhuber, J., 1988. An atlas based on the COADS data set: The budget of heat, buoyancy and turbulent kinetic energy at the surface of the global ocean. Technical Report 15, Max-Planck Institut für Meteorologie (MPI).

Oberhuber, J. M., 1993. Simulation of the Atlantic circulation with a coupled sea ice-mixed layer-isopycnal general circulation model. Part I: Model description. *J. Phys. Oceanogr.*, **23**, 808–829.

Pacanowski, R. C., 1995. MOM 2, Documentation, User's Guide and Reference Manual, Version 1.0. GFDL Ocean Technical Report 3, GFDL, Princeton.

Pacanowski, R. C. and S. G. H. Philander, 1981. Parameterization of vertical mixing in numerical-models of tropical oceans. *J. Phys. Oceanogr.*, **11**, 1443–1451.

Paluszkiewicz, T. and R. D. Romea, 1997. A one-dimensional model for the parameterization of deep convection in the ocean. *Dynamics Atmos. and Oceans*, **26**, 95–130.

Press, W. H., B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, 1988. *Numerical Recipes in C. The Art of Scientific Computing*. Cambridge University Press.

Price, J. F. and M. Baringer, 1994. Outflows and deep water productions by marginal seas. *Prog. Oceanogr.*, **25**, 162–200.

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

Roeckner, E., L. Bengtsson and J. Feichter, 1999. Transient climate change simulations with a coupled atmosphere-ocean GCM including the tropospheric sulfur cycle. *J. Climate*, **12**, 3004–3032.

Röke, F., 2001. An atlas of surface fluxes based on the ECMWF Re-Analysis - a climatological dataset to force global ocean general circulation models. Report 323, Max-Planck-Institut für Meteorologie, Hamburg, Germany.

Semtner, A. J., 1976. A model for the thermodynamic growth of sea ice in numerical investigations of climate. *J. Phys. Oceanogr.*, **6**, 379–389.

Steele, M., R. Morley and W. Ermold, 2001. Phc: A global ocean hydrography with a high-quality arctic ocean. *J. Climate*, **14**(9), 2079–2087.

Sweby, P., 1984. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.*, **21**, 995–1011.

UNESCO, 1983. Algorithms for computation of fundamental properties of seawater. UNESCO Technical Papers in Marine Science 44, UNESCO.

Visbeck, M., J. Marshall, T. Haine and M. Spall, 1997. Specification of eddy transfer coefficients in coarse-resolution ocean circulation models. *J. Phys. Oceanogr.*, **27**(3), 381–402.

Wolff, J.-O., E. Maier-Reimer and S. Legutke, 1997. The Hamburg Ocean Primitive Equation Model HOPE. Technical Report 13, German Climate Computer Center (DKRZ), Hamburg, Germany.