Hybrid Coordinate Ocean Model (HYCOM)

Version 2.1

User's Guide

Updated: January 2003

A. Wallcraft
Naval Research Laboratory
S.N. Carroll, K.A. Kelly, K.V. Rushing
Planning Systems Incorporated

This User's Guide is available at:
http://panoramix.rsmas.miami.edu/hycom/documentation.html
Report corrections to: wallcraft@nrlssc.navy.mil

Contents

List of Figures

List of Tables

1 Introduction

The Hybrid Coordinate Ocean Model (HYCOM; (Halliwell *et al.*, 1998, 2000; Bleck, 2001) was developed to address known shortcomings in the vertical coordinate scheme of the Miami Isopycnic-Coordinate Ocean Model (MICOM) developed by Rainer Bleck and colleagues. HYCOM is a primitive equation, general circulation model with vertical coordinates that remain isopycnic in the open, stratified ocean. However, the isopycnal vertical coordinates smoothly transition to z-coordinates in the weakly stratified upper-ocean mixed layer, to terrain-following sigma coordinates in shallow water regions, and back to z-level coordinates in very shallow water. The latter transition prevents layers from becoming too thin where the water is very shallow.

The HYCOM user has control over setting up the model domain, generating the forcing fields, and ingesting either the climatology or output fields from other model simulations to use for boundary and interior relaxation. The model is fully parallelized and designed to be portable among all UNIX-based systems.

An important goal in developing HYCOM was to provide the capability of selecting from several vertical mixing schemes for the surface mixed layer and comparatively weak interior diapycnal mixing. The K-Profile Parameterization (KPP, Large *et al.*, 1994; 1997) algorithm was included as the first non-slab mixed layer model because it provides mixing throughout the water column with a transition between the vigorous mixing in the surface boundary layer and the weaker diapycnal mixing in the ocean interior. KPP works on a relatively coarse and unevenly spaced vertical grid, and it parameterizes the influence of a suite of physical processes larger than other commonly used mixing schemes. In the ocean interior, the contribution of background internal wave breaking, shear instability mixing, and double diffusion (both salt fingering and diffusive instability) are parameterized. In the surface boundary layer, the influences of wind-driven mixing, surface buoyancy fluxes, and convective instability are parameterized. The KPP algorithm also parameterizes the influence of nonlocal mixing of temperature (T) and salinity (S), which permits the development of counter gradient fluxes.

Three additional mixed layer models have been incorporated into the HYCOM version 2.1 code: 1) the dynamical instability model of Price *et al.* (1986), 2) the Mellor-Yamada level 2.5 turbulence closure scheme used in the Princeton Ocean Model (POM; Mellor and Yamada, 1982; Mellor, 1998), and 3) the Kraus-Turner (KT) slab model. Other mixed layer models will be included in the near future, such as a turbulence model developed recently by Canuto (2000).

HYCOM versions 1.0 and 2.0 have previously been released as a result of collaborative efforts between the University of Miami, the Los Alamos National Laboratory, and the Naval Research Laboratory (NRL). Ongoing HYCOM research has been funded under the National Oceanographic Partnership Program (NOPP) and the Office of Naval Research (ONR).

2 Description of HYCOM Usage

This manual describes in detail the procedures for running the Hybrid Coordinate Ocean Model Version 2.1 (HYCOM). HYCOM is set up to be domain independent, and all components except the model code will be compiled only once. The model script is configured to allow data files (input and output) to be resident on a different machine (e.g., an archive system). The actual run is from a scratch directory, and files are copied from scratch to permanent storage (possibly on another machine) using whatever commands are associated with the `pget` and `pput` environment variables. If everything is on a single machine (or if the archive directory is NFS mounted on the run machine), `pget` and `pput` can both be `cp` (e.g., `setenv pget /usr/bin/cp`). Otherwise, the ALL/bin directory contains several examples of appropriate `pput` and `pget` commands.

A separate technical manual, the HYCOM User's Manual, compliments this document and contains the mathematical formulation, solution procedure, and code of the model as well as flow charts and descriptions of the programs and sub-programs (Wallcraft *et al.*, 2002).

2.1 Running Environment

Before running HYCOM the following requirements must be met:

- Unix-like Operating System, with the C shell (csh or tcsh) and awk.
- Fortran 90/95 compiler.
- A globally accessible shared file system.
- Memory and disk requirements depend on the domain size.

2.2 Directory Structure

HYCOM 2.1 has been designed so that all of the pre- and post-processing programs are compiled only once. The directory hycom_ALL contains all of the domain-independent pre- and post-processing programs. A second directory contains the data files and scripts needed to run a simulation for a specific domain. For example, everything needed to process and run HYCOM on the Atlantic 2.00 degree domain is found in the hycom_ATLb2.00 directory. These files are used in conjunction with the programs found in the ALL directory to run a simulation. A description of the subdirectories and their contents are listed in Tables 1 and 2 on the following pages.

2.3 I/O File Formats in HYCOM

Almost all HYCOM model input and output uses the standard HYCOM .[ab] format. The ".a" file contains idm*jdm 32-bit IEEE real values for each array, in standard fortran element order, followed by padding to a multiple of 4096 32-bit words (16K bytes), but otherwise with no control bytes/words, and input values of 2.0**100 indicating a data void. Each record is padded to 16K bytes to potentially improve I/O performance on some machines by aligning record boundaries on disk block boundaries.

Table 1: List of Subdirectories under Source Code Directory ALL

Subdirectory	**Contents**
archive	Source code for modifying HYCOM archive files, and converting them to other file formats.
bin	Utilities – this directory should be in user's path.
config	Machine and parallelization-specific part of makefiles.
force	Source code for interpolation of NRL format wind/flux files on "native" grid to HYCOM model grid.
libsrc	Common source files.
meanstd	Source code for forming the mean or mean-squared or standard deviation of a sequence of archive files.
plot	Source code for plotting HYCOM archive files and 2-D fields from any HYCOM ".a" file, using National Center for Atmospheric Research (NCAR) graphics.
plot_ncl	
README.ALL	README files pertaining to all of the domain-independent source code files.
relax	Source code for interpolation of a climatology to a HYCOM model grid for use in boundary relaxation zones or for model initialization.
restart	Source code for reading and writing out restart files.
sample	Source code for sampling HYCOM archive files.
subregion	Source code for extraction of a subregion from an archive file.
topo	Source code for bathymetry processing.

Table 2: List of Subdirectories under Region Directory ALTb2.00

Subdirectory	**Contents**
archive	Scripts for reading HYCOM archive files and writing archive/data/netCDF/restart files.
config	Machine and parallelization specific part of makefiles.
doc	Documentation.
expt_01.0	Old example simulation, hybrid vertical coordinate.
data	Data files for example simulation expt_01.0.
expt(_)01.5	Latest example simulation, hybrid vertical coordinate.
data	Data files for example simulation expt_01.5.
examples	
force	Atmospheric forcing data files.
coads	Generate the COADS forcing files.
offset	Wind offset files and scripts.
plot	Plot the forcing files.
meanstd	Mean and standard deviation of HYCOM surface archived fields.
plot	Example of plotting and scripts.
relax	Relaxation data files and scripts.
010	“Isopycnal” climatology for a HYCOM simulation.
999	Zonal isopycnal depth climatology for HYCOM subroutine POFLAT.
levitus	LEVITUS climatology on HYCOM horizontal grid.
plot	Plot climatology.
sample	Transport across sections from archive files
src_2.0.01_22_one	Source code HYCOM version 2.0.01
src_2.1.00_22_one	Source code, HYCOM version 2.1 for 16 layers and Message Passing Interface (MPI).
test	Programs to test that individual communication routines are working.
subregion	Interpolate bathymetry/archive to new grid/sub-domain.
topo	Grid and bathymetry data files and scripts.
partit	Partition text file for domain decomposition into tiles.
topo_IASd0.50	

The associated small “.b” file is plain text, typically containing a 5-line header followed by one line for each 2-D field in the “.a” file. The format of the per-array line varies, but it typically use ”=” to separate an array description from trailing numeric values. It is good practice to confirm that the minimum and maximum values from “.a” and “.b” files agree on input. The best way to read and write “.a” files is using the standard set of “ZAIO” routines provided in mod_za.F, see Section 15. The ocean model has separate versions of these routines for distributed memory and shared memory machines, but pre and post-processing programs use the (shared memory) version in the subdirectory libsrc in the ALL directory. There are many utilities that act on “.a” files in the bin subdirectory (see Appendix A for a list and description), and almost any program in the ALL subdirectories can act as an example of how to use the “ZAIO” routines.

The only other HYCOM model input is via, *.input, plain text files. The format of these varies, but a common format for a single line of text is a data value followed by the six-character name of the variable in quotes followed by a comment (which is ignored on input). Such lines can be read by the “blkin[ilr]” routines (for integer, logical, real input). These subroutine names come from blkdat.input, which uses this format.

The primary HYCOM model output is archive files, which are in “.[ab]” format and contain all prognostic variables for one time step (or just the surface, depth averaged and layer 1 fields). These can be converted to other file formats, including netCDF, as a post-processing step (see Section 17).

3 The HYCOM Grid

HYCOM uses the "C" grid that was originally used in MICOM although HYCOM's horizontal mesh differs from the one used in MICOM. In the MICOM mesh, the positive x direction is southward and the positive y direction is eastward. The HYCOM mesh was converted to standard Cartesian coordinates, with the x-axis pointing eastward and the y-axis pointing northward. The HYCOM mesh is illustrated below (Figure 3) for pressure (p), velocity component (u and v), and vorticity (Q) grid points. This case is for 7×7 pressure grid points. The grid meshes for the other variables have 8×8 grid points. All fields in this case would therefore be dimensioned 8×8, with the eighth row and column unused for variables on pressure grid points.

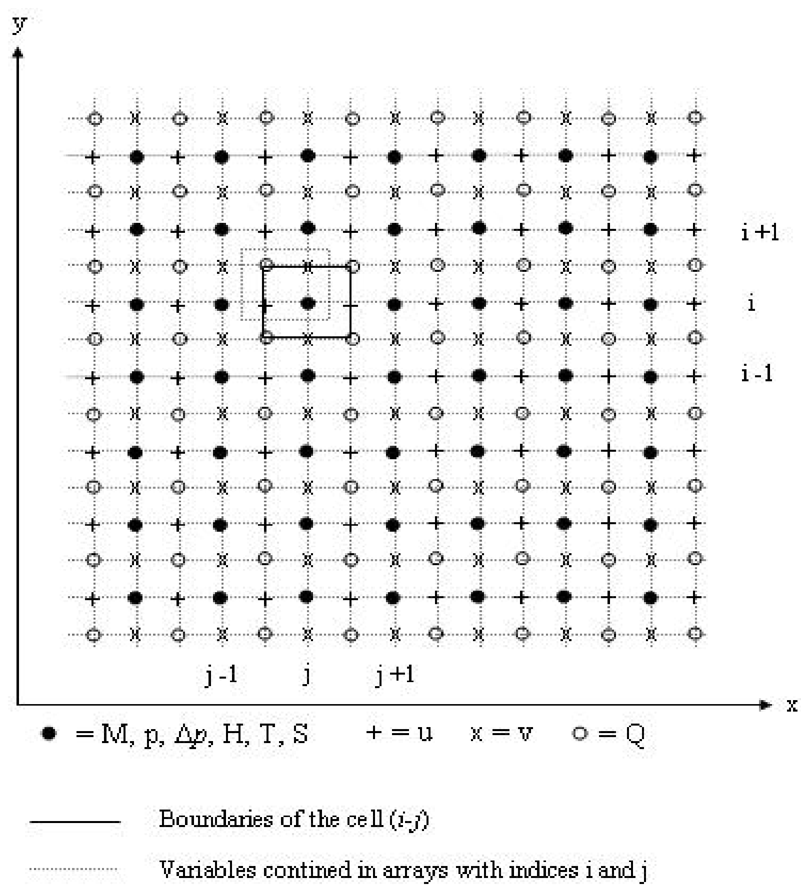

Figure 1: HYCOM mesh.

4 Operating Guidelines

There are several key aspects of setting up HYCOM for a new domain and running a simulation. The basic steps are:

1. Choose a domain and resolution.
2. Build a bathymetry.
3. Interpolate atmospheric forcing to the domain.
4. Choose vertical (isopycnal) structure (in expt_##.#/blkdat.input).
5. Interpolate T/S climatology to the model domain and the chosen vertical structure.
6. Configure and compile the model code.
7. Complete configuration of expt_##.#.
8. Run the simulation.
9. Plot and analyze results.

A new simulation on the same domain typically repeats steps 7 – 9, and if the vertical structure changes, steps 4, 5, and 7 are also repeated. The model code will only need reconfiguring if the number of layers changes.

5 Choosing a Domain and Resolution

The standard HYCOM version 2.1 model has been set up to run for the 2.00 degree Atlantic Ocean domain (ATLb2.00). The file "**regional.grid.[ab]**", which specifies the parameters for the model grid domain, has been provided in the ATLb2.00/topo subdirectory. This file is read at run time by all of the pre- and post-processing programs, so if running HYCOM for a new region, this file will need to be generated. In addition, the file "**dimensions.h**" will need to be modified. If running HYCOM on the ATLb2.00 domain, then no changes need to be made to these files.

To set up HYCOM for a new stand-alone region, the first step is to create a new directory, analogous to the ATLb2.00 directory and subdirectories. To do this, the user must pick a region name, in the format "XXXaN.NN"(e.g., IASb0.50, ATLb2.00, ATLd0.32). "XXX" is an uppercase three letter primary region name, "a" is a lowercase letter secondary region name, and "N.NN" is a three digit grid resolution description. Once the new region directory and subdirectories have been created, the next step is to create the "regional.grid.[ab]" files in the XXXaN.NN/topo subdirectory to describe the location of the new region and grid.

5.1 File "regional.grid.[ab]"

All HYCOM pre- and post-processing programs read regional.grid.b at run-time to get the longitudinal array size (`idm`) and the latitudinal array size (`jdm`) for the particular region being processed. So the script **regional.grid.com** must be run first to generate regional.grid.[ab]. The source code that is called by the script regional.grid.com is domain-independent and located in the topo subdirectory under the ALL directory. The script itself, regional.grid.com, is located in the topo subdirectory under the ATLb2.00 directory.

In the case of ATLb2.00, the grid is "mercator", which is a constant longitudinal grid spacing in degrees but latitudinal grid spacing in degrees varying with cos(latitude) to give a square grid cell in meters. The method of specifying the grid location is as used previously by MICOM and earlier versions of HYCOM. The user must set the grid specification parameters in regional.grid.com (see Figure 2). When run, regional.grid.com calls the program GRID_MERCATOR, which creates the grid definition file.

```
  0     'mapflg' = map flag (0=mercator,2=uniform,4=f-plane)
  1.0   'pntlon' = longitudinal reference grid point on pressure grid
263.0   'reflon' = longitude of reference grid point on pressure grid
  2.0   'grdlon' = longitudinal grid size (degrees)
 11.0   'pntlat' = latitudinal  reference grid point on pressure grid
  0.0   'reflat' = latitude of  reference grid point on pressure grid
  2.0   'grdlat' = latitudinal  grid size at the equator (degrees)
```

Figure 2: Grid specification parameters in "regional.grid.com"

Grid array point (`pntlon`,`pntlat`) is at longitude and latitude location (`reflon`,`reflat`) with equatorial grid spacing of `grdlon` by `grdlat`. For a mercator grid, `grdlon`=`grdlat` and `reflat` is always zero (i.e. `pntlat` indicates the array index of the equator, which need not be in 1:`jdm`). The program GRID_MERCATOR can also produce cylindrical equidistant grids (constant latitudinal grid spacing, perhaps with `grdlon .ne. grdlat`) and f-plane grids. There are other programs in the ALL/topo source directory for non-constant/non-mercator latitudinal grids (GRID_LATITUDE) and global grids with an arctic dipole patch (GRID_PANAM or GRID_LPANAM).

Any orthogonal curvilinear grid can be used, so if the above programs don't meet your needs - just produce your own regional.grid.[ab] in the right format. The easiest way to get the format right is to use the "zaio" routines to write the .a file, just as GRID_MERCATOR does. The contents of the regional.grid.b file can be seen in Figure 3.

```
   57    'idm   ' = longitudinal array size
   52    'jdm   ' = latitudinal  array size
    0    'mapflg' = map flag (-1=unknown,0=mercator,2=uniform,4=f-plane)
plon:  min,max =      263.00000     375.00000
plat:  min,max =      -19.60579      63.11375
qlon:  min,max =      262.00000     374.00000
qlat:  min,max =      -20.54502      62.65800
ulon:  min,max =      262.00000     374.00000
ulat:  min,max =      -19.60579      63.11375
vlon:  min,max =      263.00000     375.00000
vlat:  min,max =      -20.54502      62.65800
pang:  min,max =        0.00000       0.00000
pscx:  min,max =   100565.21875   222389.87500
pscy:  min,max =   100571.95312   222378.59375
qscx:  min,max =   102139.75781   222356.01562
qscy:  min,max =   102146.85938   222344.73438
uscx:  min,max =   100565.21875   222389.87500
uscy:  min,max =   100571.95312   222378.59375
vscx:  min,max =   102139.75781   222356.01562
vscy:  min,max =   102146.85938   222344.73438
cori:  min,max =  -0.0000511824   0.0001295491
pasp:  min,max =        0.99993       1.00005
```

Figure 3: Contents of the file "regional.grid.[ab]"

Three header lines in regional.grid.b identify the domain size and the map projection (which is ignored for most purposes). This is followed by one line for each field in regional.grid.a: (i) the longitude and latitude of all four grids, (ii) the angle of the p grid with respect to a standard latitude and longitude grid, (iii) the grid spacing in meters of all four grids, (iv) the Coriolis parameter (q-grid), and (v) the aspect ratio of the p grid (pasx/pscy).

5.2 Parameter file "dimensions.h"

When choosing a new domain or changing the number of layers, the user must alter the source code file "**dimensions.h**" or select one of the dimensions.h files already made available in HYCOM in the ALTb2.00/src_*_* subdirectory. There are several example versions for different regions available in HYCOM from which the user can choose (see Table 3). Typically, the "_omp" OpenMP version of dimensions.h is appropriate for a single processor and the "_ompi" OpenMP+MPI version is used for any distributed memory configuration (MPI only, SHMEM only, or MPI+OpenMP). To use one of these files, the user must copy the appropriate version to dimensions.h. Alternatively, the user can create a version for a new region by altering the parameters in dimensions.h. These user-tunable parameters and their descriptions are listed in Table 4.

Table 3: Regional **dimensions.h** versions available in HYCOM.

File	**Description**
dimensions_ATLa2.00_omp.h	2.00 degree Atlantic, shared memory.
dimensions_ATLa2.00_ompi.h	2.00 degree Atlantic, distributed memory.
dimensions_ATLd0.32_omp.h	0.32 degree Atlantic, shared memory.
dimensions_ATLd0.32_ompi.h	0.32 degree Atlantic, distributed memory.
dimensions_JESa0.18_omp.h	0.18 degree Japan/East Sea, shared memory.
dimensions_JESa0.18_ompi.h	0.18 degree Japan/East Sea, distributed memory.

5.2.1 Grid dimensions

In order to change the region size or the number of layers, the user can change the parameters `itdm`, `jtdm`, or `kdm` in dimensions.h. The default values for these parameters have been set to a total grid dimension of 57 by 52, and 16 vertical layers for the Atlantic 2.00 degree domain. The user must create a new source code directory and executable every time the values of these parameters are changed. In addition, the user must update the "**regional.grid.b**" file that is used to define the region to setup programs so that it is consistent with dimensions.h.

If memory is plentiful, then `kkwall`, `kknest`, `kkmy25` can all be set to `kdm`. However, if memory is in short supply then `kwall` and/or `kknest` can be set to 1 if wall or nest relaxation is not being used. The parameter `kkmy25` can be set to -1 if the Mellor-Yamada mixed layer is not being used.

Table 4: Region and tiling specific parameters in **dimensions.h**.

Parameter	Description
itdm	Total grid dimension in i direction.
jtdm	Total grid dimension in j direction.
kdm	Grid dimension in k direction.
iqr	Maximum number of tiles in i direction.
jqr	Maximum number of tiles in j direction.
idm	Maximum single tile grid dimension in i direction.
jdm	Maximum single tile grid dimension in j direction.
mxthrd	Maximum number of OpenMP threads.
kkwall	Grid dimension in k direction for wall relax arrays.
kknest	Grid dimension in k direction for nest relax arrays.
kkmy25	Grid dimension in k direction for M-Y 2.5 arrays.

5.2.2 mxthrd

The parameter `mxthrd` in dimensions.h is only important when using OpenMP (`TYPE` = `omp` or `ompi`). OpenMP divides each outer (`i` or `j`) loop into `mxthrd` pieces. Set `mxthrd` as an integer multiple of the number of threads (`omp_num_threads`) used at run time (*i.e.*, of `NOMP`), typically set as `jblk` = (`jdm`+`2*nbdy`+`mxthrd-1`)/`mxthrd` ranges from 5-10. For example, `mxthrd` = 16 could be used with 2, 4, 8 or 16 threads. Other good choices are 12, 24, 32, etc. Large values of `mxthrd` are only optimal for large `idm` and `jdm`. For `TYPE = omp`, use the command `bin/hycom_mxthrd` to aid in selecting the optimal `mxthrd`. It prints out the stripe size (`jblk`) and load-balance efficiency of all sensible `mxthrd` values. Set `mxthrd` larger than `omp_num_threads` to give better land/sea load balance between threads. The directives have not yet been extensively tuned for optimal performance on a wide range of machines, so please report cases to Alan Wallcraft where one or more routines scale poorly. Also send in any improvements to the OpenMP directives.

A separate source code directory and executable are required for each parallelization strategy, or `TYPE`, chosen by `TYPE` = `one`, `omp`, `ompi`, `mpi`, or `shmem`. The `TYPE` also affects how dimensions.h is configured.

5.2.3 Dimensioning tiles

When running on a shared memory machine (`TYPE` = `one` or `omp`), set the parameters `iqr` = `jqr` = 1, `idm` = `itdm`, and `jdm` = `jtdm`. Note that the same OpenMP executable (`TYPE` = `omp`) may be used for a range of processor counts, provided `mxthrd` is chosen appropriately. When running on a distributed memory machine (`TYPE` = `mpi` or `ompi` or `shmem`) set `iqr` and `jqr` to the maximum number of processors used in each dimension, and `idm` and `jdm` to the maximum (worse case) dimensions for any single tile on any targeted number of processors. Note that the same executable may be used for a range of processor counts, provided `iqr`, `jqr`, `idm`, and `jdm` are large enough for each case.

5.3 "poflat.f"

If the user changes regions, the file "**poflat.f**" must also be modified for the particular region. The file poflat.f defines the zonal climatology for the initial state when `iniflg` = 1 (**blkdat.input** parameter; Appendix B). Example versions already available in HYCOM for different regions are listed in Table 5.

Note that this routine does not depend on grid resolution. Copy the appropriate version to poflat.f, or create a version for a new region. There are examples of how to create a new version in the directory relax. All input bathymetry, forcing and boundary relaxation files are also region specific and are selected at run time from blkdat.input.

Table 5: Regional **poflat.f** versions available in HYCOM.

File	**Description**
poflat_ATLa.f	Atlantic to 65N.
poflat_ATLd.f	Atlantic to 70N, including the Mediterranean.
poflat_JESa.f	Japan/East Sea.
poflat_F1Da.f	1-D test case (same profile at all latitudes.)
poflat_F2Da.f	2-D upwelling case (same profile at all latitudes).
poflat_SYMa.f	3-D symmetry case (same profile at all latitudes).

6 Building a Bathymetry

HYCOM 2.1 provides the bathymetry files for the Atlantic 2.00 domain and the scripts that were used to generate these files in the ATLb2.00/topo subdirectory (See Table 6). If the user wants to generate bathymetry files for other regions, the bathymetry on the HYCOM grid can be generated in one of three ways:

1. Data sets from the Earth Topography 2 (ETOPO2) Global Earth Topography from the National Geophysical Data Center (NGDC) can be obtained and used to generate the bathymetry. This data can be accessed at `http://dss.ucar.edu/datasets/ds759.1/`.

2. A bathymetry file for the Atlantic 2.00 degree region can be copied. Since filenames include the region name, the script **new_topo.com** is provided to copy scripts from one region to another.

3. A new bathymetry file can be generated on the HYCOM grid by using the program TOPINT and the 5-minute TerrainBase data set. The TOPINT program is located in the file bathy_05min.f in the subdirectory topo. The Terrain Base data set is available at `ftp://obelix.rsmas.miami.edu/awall/hycom/tbase_for_hycom.tar.gz`.

If the bathymetry is being created for a new region, the newly generated bathymetry files and landsea masks should be placed in the XXXaN.NN/topo subdirectory that was created for the new region.

Table 6: Atlantic 2.00 degree bathymetry files (ATLb2.00/topo subdirectory).

File	**Description**
depth.51x56	MICOM bathymetry file (Atlantic, two-degree).
depth_ATLa2.00_01.[ab]	HYCOM bathymetry file.
depth_ATLa2.00_01.com	Script to create depth_ATLa2.00_01.
depth_ATLa2.00_01.log	From csh depth_ATLa2.00_01.com>& depth_ATLa2.00_01.log.
depth_ATLa2.00_02.[ab]	HYCOM bathymetry file (smoothed 01).
depth_ATLa2.00_02.com	Script to create depth_ATLa2.00_02 (smoothed 01).
depth_ATLa2.00_02.log	From csh depth_ATLa2.00_02.com>& depth_ATLa2.00_02.log.
depth_ATLa2.00_03.[ab]	HYCOM bathymetry file (flat bottom).
depth_ATLa2.00_03.com	Script to create depth_ATLa2.00_03 (flat bottom).
depth_ATLa2.00_03.log	From csh depth_ATLa2.00_03.com>& depth_ATLa2.00_03.log.

6.1 Bathymetry File Naming Convention

Depth files include the region name (i.e., ATLa2.00) so that files from several regions may be collected in one directory. The ending "_01" indicates version 01 of the bathymetry, and this convention allows for up to 99 distinct bathymetries for the same region and resolution. For example, the Atlantic 2.00 degree bathymetry files, "depth_ATLa2.00_02" and "depth_ATLa2.00_03", have the same land or sea boundary as "depth_ATLa2.00_01" but 02 applies a 9-point smoother to the 01 depths and 03 has a flat bottom at 5000 m. Additional smoothing, as in depth_ATLa2.00_02, may not be necessary at two-degree resolution but one or two smoothing passes may be appropriate when using higher resolution.

6.2 Example IASb0.50: Creating a New Bathymetry

An example dataset and scripts have been provided in HYCOM to demonstrate how to create a new bathymetry (Table 7). The dataset is a subset of the Intra Americas 0.50 degree domain based on the 5 minute global TerrainBase data set. The version used here has been extended by 5 degrees across the N and S poles to simplify interpolation near the poles. In this example, depth_IASb0.50_01 is a raw bathymetry that is not used in the model run, but it becomes depth_IASb0.50_02 after editing. Additional smoothing, as was used in depth_ATLb2.00_02, may not be necessary at 2 degree resolution but one or two smoothing passes might be appropriate when using higher resolution. The script **new_topo.com** is provided to copy scripts from one region to another, since filenames include the region name.

Table 7: Example bathymetry (ATLb2.00/topo_IASd0.50).

File	Scripts to date
depth_IASb0.50_01.[ab]	5 minute global terrain base data set for Intra Americas 0.50 degree domain.
depth_IASb0.50_01.com	Script to interpolate 5-minute bathymetry to HYCOM bathymetry.
depth_IASb0.50_01_map.com	Script to map a HYCOM bathymetry.
depth_IASb0.50_02_landmask.com	Script that applies a landmask "_01" to get "_02".
depth_IASb0.50_02_map.com	Script to map the masked HYCOM bathymetry.
landsea_IASb0.50.com	Script to interpolate 5-minute bathymetry to HYCOM landsea mask.
landsea_IASb0.50_modify.com	Script to modify a HYCOM land or sea mask.
regional.grid.[ab]	HYCOM grid location file.
regional.grid.com	Script to create regional.grid.[ab].
regional.grid.log	Log file generated from csh regional.grid.com >&regional.grid.log.

6.2.1 Steps for Generating New Bathymetry

To generate the new bathymetry on the HYCOM grid, the user must follow these steps:

1. Obtain a bathymetry dataset (Example: "depth_IASb0.50_01.[ab]").

2. Run the following scripts:

   a) Run **regional.grid.com**. All programs read regional.grid.b at run-time to get idm and jdm for the particular region being processed. So regional.grid.com must be run first to generate regional.grid.[ab].

   b) Run **depth_IASb0.50_01.com** to interpolate the 5-minute bathymetry to HYCOM bathymetry.

   c) Run **landsea_IASb0.50.com** to interpolate the 5-minute bathymetry to the HYCOM land or sea mask.

   d) Run **depth_IASb0.50_01_map.com** (choose landmask for 02).

   e) Run **depth_IASb0.50_02_landmask.com**. A landmask is optional, but is used to distinguish between the model land or sea boundary (e.g., the 20m isobath) and the actual coastline (at least to the limits of the grid used) on plots. It isn't necessary unless you are using the NCAR graphics-based HYCOMPROC and FIELDPROC.

   f) Run **depth_IASb0.50_02_map.com** to map the HYCOM bathymetry (choose land-sea modifications).

   g) Run **landsea_IASb0.50_modify.com** to modify the HYCOM land or sea mask.

Some steps may need to be iterated to get them right, and plots may also help this process. The source code is domain-independent and therefore located in the ALL/topo subdirectory.

6.3 Using MICOM Bathymetry Files

HYCOM allows for the conversion of MICOM bathymetry files to a corresponding HYCOM bathymetry file using the program T_M2H in topo_m2h.f. Since the MICOM and HYCOM bathymetry files leave out the last row and column (which are always outside the region), the conversion from MICOM bathymetry to HYCOM is simple (HYCOM idm,jdm):

```
do j= 1,jdm-1
  do i= 1,idm-1
    dh(i,j) = dm(jdm-j,i)
  enddo
enddo
```

The file "topo_m2h.f" defines the domain-specific header of the bathymetry file and must be customized for each domain. The suggested resolution for values in the header is at least five significant digits. Some MICOM bathymetry files use an alternative PAKK encoding. If the HYCOM bathymetry from TOPO_M2H does not look correct, try TOPO_MALT2H, which links in pakk_micom.f instead of pakk_hycom.f. The only differences between these files are six lines in the subroutine UNPAKK (lines 107-112).

7 Interpolating Atmospheric Forcing to the Domain

After establishing a bathymetry for the domain, the user must interpolate wind data to the HYCOM grid for the region chosen so that data can be input to the model. In order to obtain input files to run in the HYCOM model, the user must do the following steps:

1. Obtain wind or flux data for the region being run in HYCOM,
2. Create a wind offset file, and
3. Run scripts **coads_mon_wind.com** or **coads_mon_flux.com** that call programs WNDINT or FLXINT to interpolate wind or flux data to the HYCOM grid.

The script **new_force.com** has been provided in the ATLb2.00/force subdirectory. This script can be used to edit the forcing scripts for a new region. These scripts can then be run in the XXXaN.NN/force subdirectory to interpolate atmospheric forcing fields to the new region.

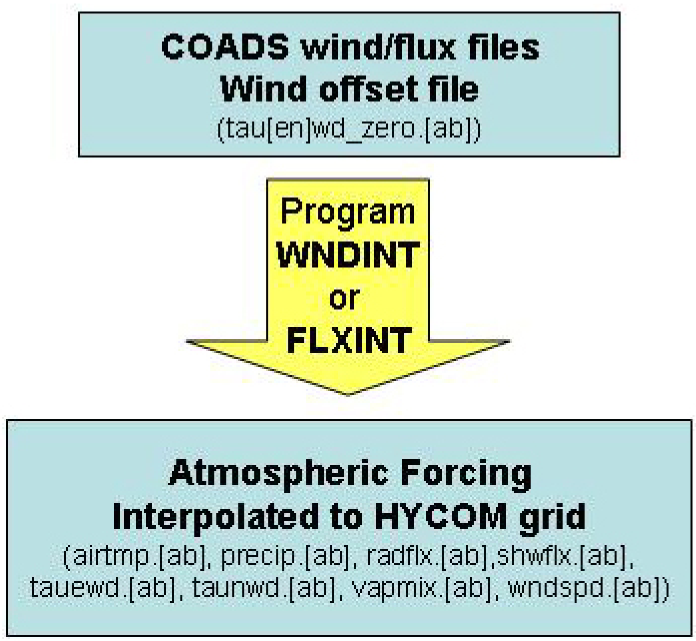

Figure 4: Flow chart of atmospheric forcing interpolation to HYCOM grid scheme.

7.1 COADS wind data

The user can obtain wind data produced by the Comprehensive Ocean-Atmosphere Data Set project (COADS) from the HYCOM ftp site (`ftp://obelix.rsmas.miami.edu/awall/hycom/-coads_for_hycom.tar.gz`). The COADS wind files are formatted in Naval Research Laboratory (NRL) format that consists of a Fortran unformatted sequential file with a single header record identifying wind dates followed by the wind or flux data (**coads_mon_taqaqrqppc.d**). This is a format used at NRL to avoid dealing with multiple wind file formats. All wind sets are converted to this format so that interpolation programs can use a single input routine for any wind set. There can be up to 5,999 sample times in one file and the very first record of the file contains the array size, the array latitude, longitude, the array grid size, the number of samples, and a 6,000 element array that lists the "wind day" of each sample time (and of the next time in sequence beyond the last record). Here "wind day" is days since 00Z December 31 1900. To use the interpolation programs, either convert your atmospheric forcing data to this format or modify the native grid reading subroutines to input fields in their existing format. There are many programs already available to convert wind sets to the required format. To see if the one needed is available , send e-mail to `metzger@hermes.nrlssc.navy.mil`.

The `wind_stat` command located in the bin subdirectory summarizes the contents of the native wind or flux data file. In Figure 5, an example is given of the COADS data file summary. Note that `Qr` is based on COADS constrained net flux, and `Qp=Qr+Qlw`. The wind days (`1111.00` to `1477.00`) and dates (`16.00/1904 - 16.00/1905`) are ignored by the interpolation programs. All that matters is that the file contains 12 monthly sets of fields. The COADS wind grid specification parameters are outlined in Table 8.

```
> wind_stat flux_ieee/coads/coads_mon_taqaqrqppc.d
COADS monthly Ta,Ha,Qr,Qr+Qlw,Pc 366 day
IWI,JWI = 360, 180    XFIN,YFIN =    0.50,  -89.50    DXIN,DYIN = 1.000, 1.000
NREC =   12     WDAY =
  1111.00  1142.00  1171.00  1202.00  1232.00  1263.00  1293.00  1324.00
  1355.00  1385.00  1416.00  1446.00  1477.00
   12 RECORD CLIMATOLOGY STARTING ON  16.00/1904 COVERING   366.00 DAYS

> wind_stat uwm_coads_monmn_unsm.d
COADS mnth clim, 366 day, unsmooth, MKS
IWI,JWI = 360, 180    XFIN,YFIN =    0.50,  -89.50    DXIN,DYIN = 1.000, 1.000
NREC =   12     WDAY =
  1111.00  1142.00  1171.00  1202.00  1232.00  1263.00  1293.00  1324.00
  1355.00  1385.00  1416.00  1446.00  1461.00
   12 RECORD CLIMATOLOGY STARTING ON  16.00/1904 COVERING   350.00 DAYS
```

Figure 5: Example of COADS wind file summary provided by wind_stat.

Table 8: COADS wind grid specification parameters.

Parameter	Description
`iwi`	First dimension of wind or flux grid.
`jwi`	Second dimension of wind or flux grid.
`xfin`	Longitude of first wind or flux grid point.
`yfin`	Latitude of first wind or flux grid point.
`dxin`	Wind or flux longitudinal grid spacing.
`dyin`	Wind or flux latitudinal grid spacing.
	= 0.0; Gaussian grid with `jwi`/2 nodes per hemisphere.
`wmks`	Scale factor from input to MKS stress.
`hmks`	Converts from input air relative humidity units to *kg/kg*.
`rmks`	Converts from input radiation heat flux units to W/m^2 (positive into ocean).
`pmks`	Converts from input precipitation units to m/s into ocean.

7.2 Wind offset

Before interpolating COADS winds to the HYCOM grid, a wind "offset" input file must be read in. The "offset" file allows the annual mean wind to come from a different wind data set and is usually set to zero. Therefore, the first step for wind generation is to create the file "**tau[en]wd_zero.[ab]**" using the script **tauXwd_zero.com** in the offset subdirectory for the specified model region. The offset can also be a different field for each sample time. This allows the combining of a climatology and anomaly field.

7.3 WNDINT and FLXINT

The programs WNDINT and FLXINT carry out the interpolation of wind or flux files from their "native" grid to the HYCOM model grid. The interpolation methods used by the programs are piecewise bilinear or cubic spline. WNDINT and FLXINT are used as part of the standard HYCOM run script to produce the 6 or 12-hourly wind or fluxes for actual calendar days from the operational center wind or flux files. This is done "just in time"(i.e., the files for the next model segment are generated while the current model segment is running).

The only difference in the program WNDINT for the files **wi_100_co.f** and **wi_1125_ec.f** (for the one-degree COADS winds and 1.125-degree ECMWF winds, respectively) is the inclusion of different **WIND*.h** include files that define the native wind and/or flux geometry and units for each file. The most common allowable grid types for global atmospheric data sets are uniform latitude/longitude or uniform in longitude with Gaussian global latitude. Use the script **WIND_update.com** to create source files for other known native grids from the ***_100_co.f** version. The script should be updated when another native grid is added.

Table 9: Atmospheric forcing input and output files.

File	**Description**
Input files	
COADS	Wind or flux file in NRL format.
tau[en]wd_zero.[ab]	Annual mean wind offset file.
Output files	
airtmp.[ab]	Surface air temperature field in degrees C.
precip.[ab]	Precipitation field in m/s.
radflx.[ab]	Radiation heat flux field in W/m^2 (positive into ocean).
shwflx.[ab]	Short-wave radiation heat flux field in W/m^2 (positive into ocean).
tauewd.[ab]	Surface wind stress in N/m^2 (`tau_x`, positive eastwards).
taunwd.[ab]	Surface wind stress in N/m^2 (`tau_y`, positive northwards).
vapmix.[ab]	Water vapor mixing ratio in *kg/kg*.
wndspd.[ab]	Wind speed field at 10 m in m/s.

7.4 Output

The output data sets consist of atmospheric forcing in MKS units. Heat flux is positive into the ocean. The output files consist of the COADS fields interpolated to the HYCOM grid in HYCOM 2.1 array ".a" and header ".b" format. A list of the current HYCOM atmospheric forcing output and input files is listed in Table 9. An example file is provided of wind stress output (See Figure 6).

The output files also include any bias or minimum wind speed constraints. For example, in the subdirectory ATLb2.00/force/coads, compare **coads_mon_flux.com** (zero bias) to **coads_mon_flux+070w.com** (70w bias). An all zero precipitation file input to HYCOM indicates that there should be no evaporation-precipitation surface salinity forcing (See **precip_zero***). By default the output wind stress components are on the "native" u and v grids, but setting NAMELIST variable IGRID=2 will output wind stress component on the pressure grid (which is always used for all other atmospheric forcing fields). When running HYCOM, use `wndflg=1` for u/v winds and `wndflg=2` for winds on the pressure grid (`wndflg` set in blkdat.input). Note that in all cases the HYCOM wind stresses are orientated with respect to the local HYCOM grid (i.e., taunwd (`tau_y`) is only actually "North-ward" when the HYCOM grid is E-W/N-S). The actual surface forcing fields and their units are found in Table 9.

```
ajax 96> cat tauewd.b
COADS monthly, MKS



i/jdm,iref,reflon,equat,gridsz/la =   57   52  1  -97.000    11.00 2.000 0.000
 tau_ewd: month,range = 01  -1.3818002E-01   2.1997201E-01
 tau_ewd: month,range = 02  -1.3790721E-01   1.8771732E-01
 tau_ewd: month,range = 03  -1.3121982E-01   1.4483750E-01
 tau_ewd: month,range = 04  -1.1269118E-01   8.3349183E-02
 tau_ewd: month,range = 05  -1.0441105E-01   7.3009044E-02
 tau_ewd: month,range = 06  -1.3582233E-01   6.2626213E-02
 tau_ewd: month,range = 07  -1.4753306E-01   5.9464872E-02
 tau_ewd: month,range = 08  -1.0999266E-01   6.8312079E-02
 tau_ewd: month,range = 09  -1.0449981E-01   1.0520227E-01
 tau_ewd: month,range = 10  -9.0205058E-02   1.4911072E-01
 tau_ewd: month,range = 11  -8.7021284E-02   1.7461090E-01
 tau_ewd: month,range = 12  -1.2721729E-01   1.9768250E-01

ajax 97> hycom_range tauewd.a 57 52
 min, max =  -0.13818002 0.21997201
 min, max =  -0.1379072 0.18771732
 min, max =  -0.13121982 0.1448375
 min, max =  -0.11269118 0.08334918
 min, max =  -0.10441105 0.073009043
 min, max =  -0.13582233 0.06262621
 min, max =  -0.14753306 0.059464871
 min, max =  -0.10999266 0.06831208
 min, max =  -0.10449981 0.10520227
 min, max =  -0.09020506 0.14911072
 min, max =  -0.087021283 0.1746109
 min, max =  -0.1272173 0.1976825

 12  FIELDS PROCESSED
```

Figure 6: Example COADS output file “tauewd.a”.

8 Choosing a Vertical (Isopycnal) Structure

Vertical structure parameters are selected when editing **blkdat.input** for each model simulation (see Figure 7). The blkdat.input file is located in the ATLb2.00/expt subdirectory. If setting up for a new domain, blkdat.input will need to be edited for the vertical structure chosen by the user.

1. Begin by specifying the fixed vertical grid near the surface through the number of sigma-levels (`nsigma`, which is 0 for all `z`-levels).

2. Next, choose the minimum sigma thickness (parameter `dp00s`) and the minimum (`dp00`). The kth layers minimum `z`-thickness is `dp00f`**(`k-1`)*`dp00`, but if `k` is less than `nsigma` the minimum thickness is the smaller of this value and the larger of `dp00s` and `depth/nsigma`. This approach gives `z`-levels in deep water, optionally going to sigma-levels in coastal water and back to `z`-levels in very shallow water.

3. Finally, select a `z`-level stretching factor (parameter `dp00f`, which is 1 for uniform `z`-levels).

0	'nsigma'	=	number of sigma levels(nhybrd-nsigma z-levels)	
3.0	'dp00 '	=	deep	z-level spacing minimum thickness (m)
12.0	'dp00x '	=	deep	z-level spacing maximum thickness (m)
1.125	'dp00f '	=	deep	z-level spacing stretching factor (1.0=const.space)
3.0	'ds00 '	=	shallow	z-level spacing minimum thickness (m)
12.0	'ds00x '	=	shallow	z-level spacing maximum thickness (m)
1.125	'ds00f '	=	shallow	z-level spacing stretching factor (1.0=const.space)

Figure 7: Example of vertical structure parameters in “blkdat.input”.

9 Interpolating Climatology to the Domain

The next step in setting up HYCOM is to interpolate the temperature and salinity climatology to the model domain and the vertical structure. There are three steps for setting up the climatology for model initialization:

1. Obtain temperature and salinity data files,
2. Interpolate climatology to the HYCOM grid using the program TSZINT, and
3. Convert climatology from z-levels to isopycnals using the program RELAX.

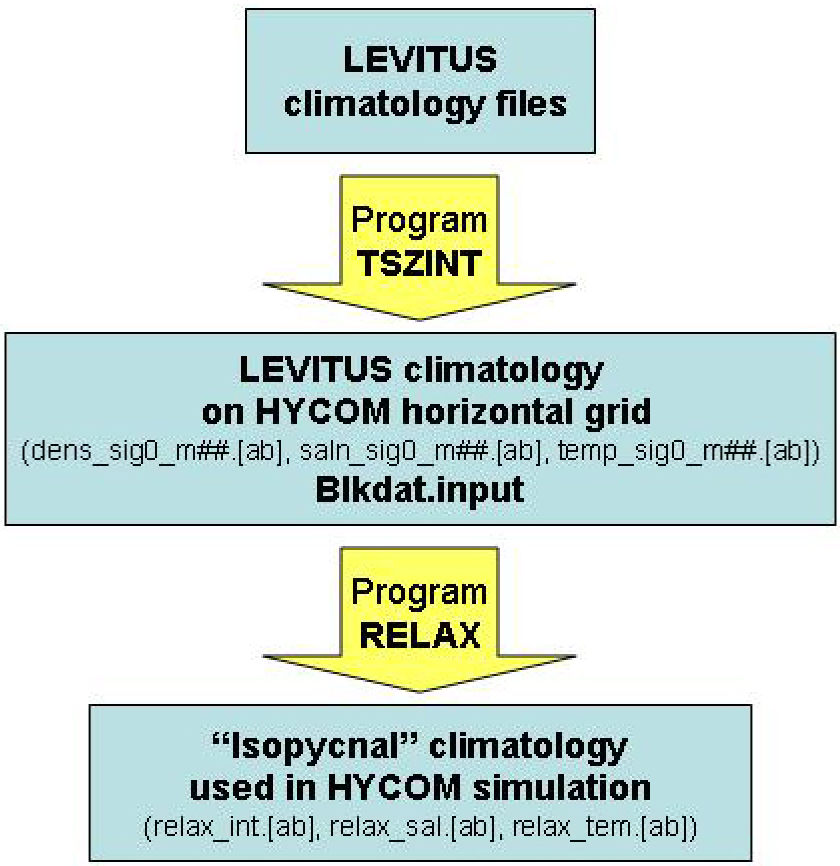

Figure 8: Flowchart of climatology interpolation to HYCOM grid.

9.1 LEVITUS Climatology Files

The climatology data sets are input on their native grid from standard Naval Research Laboratory (NRL) LEVITUS climatology files. LEVITUS files in the required format are available from the HYCOM ftp site: `ftp://obelix.rsmas.miami.edu/awall/hycom/levitus_for_hycom.tar.gz`. The user can use the `clim_stat` command located in the bin subdirectory to list the contents of a native climatology file. All fields in the native file must be defined at all grid points. This includes over land and below the ocean floor. In addition, the potential density vertical profile must be stable at all locations. Table 10 gives the grid specification parameters for the climatology files.

Table 10: Native climatology grid specification parameters.

Parameters	**Description**
`iwi`	First dimension of climatological grid.
`jwi`	Second dimension of climatological grid.
`jwi`	Third dimension of climatological grid (number of `z`-levels).
`xfin`	Longitude of 1^{st} climatological grid point.
`yfin`	Latitude of 1^{st} climatological grid point.
`dxin`	Climatological longitudinal grid spacing.
`dyin`	Climatological latitudinal grid spacing.

9.2 TSZINT - Interpolating to the Domain

The process of interpolating LEVITUS climatology files has been split into two phases. This saves time because the `z`-level climatology does not depend upon the isopycnals chosen by a particular HYCOM simulation. The first phase requires the generation of a formatted model grid climatology file suitable for input to the HYCOM isopycnal climatology generator. The climatology is first interpolated to the HYCOM horizontal grid at its native fixed `z`-levels by the program TSZINT (in "z_levitus.f"). The user must run the script **z_levitus_sig[02].com**, which will call TSZINT. This script is located in the ATLb2.00/relax/levitus subdirectory. The interpolation is performed using either piecewise bilinear or cubic spline. The interpolated climatology is defined at all grid points. Again, this includes land and below the ocean floor, and its potential density vertical profile is stable at all locations. The native LEVITUS climatology is defined using `sigma0`, but the interpolated climatology may be set as `sigma0`, `sigma2`, or `sigma4`.

9.3 RELAX - Interpolating to the Vertical Structure

The second phase is vertical mapping from z-levels to isopycnals, which is based on Rainer Bleck's "re-step" procedure for converting one stair step (i.e., piecewise constant) set of profiles

Table 11: Climatology input and output files.

File	Description
Input files	
dens_sig0_m01.[ab]	Monthly LEVITUS file of potential density interpolated to HYCOM horizontal grid using subroutine TSZINT.
saln_sig0_m01.[ab]	Monthly LEVITUS file of potential salinity interpolated to HYCOM horizontal grid using TSZINT.
temp_sig0_m01.[ab]	Monthly LEVITUS file of potential temperature interpolated to HYCOM horizontal grid using TSZINT.
blkdat.input	Created from a previous **blkdat.input** file using the script blkdat.com.
Output files	
relax.0000_###_00.[ab]	Dummy HYCOM archive from climatology (monthly).
relax_int.[ab]	Climatological interface depth in specified isopycnal coordinates.
relax_sal.[ab]	Potential salinity in specified isopycnal coordinates.
relax_tem.[ab]	Potential temperature in specified isopycnal coordinates.

(in this case between `z`-levels) into another with prescribed density steps. The user must run the script **relax.com**, which calls the program RELAX to perform the conversion of the interpolated climatology to the "isopycnal" climatology required for a particular HYCOM simulation. The region and simulation specific environment variables needed by program RELAX are located in **EXPT.src**. If the experiment or region is changed, this file must be modified. Program RELAX does not depend on the climatology that is being used, provided that all the native climatologies use the same number of `z`-levels in the vertical. Use **relax_sig0.f** for `sigma0` and **relax_sig2.f** for `sigma2` HYCOM density coordinates. The required input file, blkdat.input, is located in the relax/010 subdirectory and can be created from a previous experiment version of blkdat.input using the script **blkdat.com**.

9.4 Output

The output fields generated from this procedure include climatological interface depth, potential temperature and salinity for the specified set of isopycnal layers and minimum near-surface layer thicknesses (Table 11). The fields are output in array (.a) and header (.b) file format. Programs TSZINT and RELAX each handle a single set of climatology fields. Since HYCOM expects six bi-monthly or twelve monthly sets of fields in a single file, the individual output climatology files must be concatenated before use. The output fields can be plotted using the standard HYCOM archive file plot program HYCOMPROC. Note that **relax_zon[02].f**, a special case of **relax_sig[02].f**, writes out zonal interface depth averages. It can be used to calculate region-specific values for zonal initialization via HYCOM's **poflat.f**.

The HYCOM climatology can be used for initializing the model (`iniflg`=2), for surface relaxation to augment surface atmospheric forcing (`trelax`=1 and/or `srelax`=1), and for lateral boundary nudging (`relax`=1). In the latter case, a relaxation mask is required to specify where and how much relaxation to apply. It can be generated by "rmu.f".

9.5 2-D Relaxation Mask for Lateral Boundary Nudging

A 2-D relaxation mask is required for any HYCOM simulation that uses lateral boundary nudging. It is typically zero everywhere except the boundary regions where relaxation is to be applied. In the boundary regions, set the mask to the relaxation scale factor (1/second). Use program RLXMSK (**rmu.f**) to specify the boundary relaxation zones (see **relax_rmu.com)**. Input is up to 99 individual patches and the associated e-folding time in days (converted internally to 1/e-folding-time-in-seconds for assignment to rmu.f).

9.6 Generating Zonal Statistics

The relax/999 subdirectory contains scripts for the user to generate zonal statistics that can be used to customize the "pdat" array in HYCOM subroutine poflat. The poflat subroutine represents the depth of potential density 21.0, 21.5, ... , 28.0 in a specified latitude range. Note that this script is only used for initialization, and only when iniflg=1.

The script **blkdat.com** must be run first to generate a blkdat.input file. The next step is to run the script **relax_zon.com**, which is the primary zonal climatology script that generates the statistics. The script **sig_lat.com** extracts values needed for poflat.

The following sequence of commands are used to generate the statistics:

```
>csh blkdat.com
>csh relax_zon.com >& relax_zon.log
>./sig
>csh sig_lat.com >& sig_lat.log
```

The `./sig` command generates the zonal tables and plots using the gnu plot script **sig.gnu**. The awk script **sig_lat.awk** has been provided for interpolating to a given latitude. The last step is to use the following command:

```
>cut -c 13-18 sig_lat.log | paste -s -d ",,,,,,,,,,,,,,,\n" - | sed
-e 's/ *//g' -e 's/^....../     +/' >! sig_lat.tbl
```

to extract sig_lat.tbl from sig_lat.log. This only needs minor editing for use in poflat (see sig_lat.data).

10 Configuring and Compiling the Model Code

The makefiles necessary to run HYCOM setup programs and the model simulation source machine-specific configuration files that contain the architecture type and/or the parallelization strategy type. The diagnostics in ALL are completely separate from the model code. The first step the user must take is to compile the set-up programs in the ALL directory, which is domain independent:

1. Make sure a setup program configuration file ($(ARCH)_setup) exists for the particular machine architecture being used. If it does not exist, the user will have to create one.
2. Edit **Make_all.src.**
3. Run **Make_all.com** from the ALL root directory.

Then for each model domain the user will need to do the following:

1. If setting up HYCOM for a new stand-alone region, create a new source code directory for the region.
2. Check available machine-specific configuration files ($(ARCH)_$(TYPE)) to be sure a file exists for the particular machine architecture and type of system the model is being run on. If it does not exist, the user will have to create one.
3. Edit the script **Make.com** and the **dimensions.h file.**
4. Run **Make.com** from the source code directory (e.g., ATLb2.00/src_*).

The following subsections of this chapter provide details of configuring the files for the setup programs versus the model domain, and also detail how to edit and run the compilations for each step. Further explanation of compiling the code for the Atlantic 2.00 degree domain or a new domain are also presented.

10.1 Setup Programs

10.1.1 Configuration Files for Setup Programs

HYCOM Version 2.1 has several setup program configuration files available for the makefiles to source. These files are formatted as $(ARCH)_setup, where ARCH defines exactly what machine architecture to target. They are located in the setup program directory ALL under the config subdirectory (See Table 12). These files contain the environmental variables needed to run HYCOM setup programs for a specific machine architecture. For machines not listed in Table 12, new $(ARCH)_setup files must be created by the user. See Table 13 for a list of the environmental variables and their descriptions that must be defined in the $(ARCH)_setup file.

The file $(ARCH)_setup is typically identical to HYCOM's standard configuration file $(ARCH)_one4, if real is real*4. If real is real*8 but real*4 is available, $(ARCH)_setup is like $(ARCH)_one except that the macro REAL4 is set (in addition to REAL8).

Table 12: Setup program configuration files available in HYCOM.

File	**Description**
alpha_setup	Compaq Alpha.
alphaL_setup	Compaq Alpha Linux.
intel_setup	Intel Linux/pgf90.
intelIFC_setup	Intel Linux/ifc (little-endian)
o2k_setup	SGI Origin 2800.
sp3_setup	IBM SMP Power3.
sun_setup	Sun (32-bit).
sun64_setup	Sun (64-bit).
t3e_setup	Cray T3E.

Table 13: Environment variables in configuration files.

Variable	**Description**
`FC`	Fortran 90 compiler.
`FCFFLAGS`	Fortran 90 compilation flags.
`CC`	C compiler.
`CCFLAGS`	C compilation flags.
`CPP`	CPP preprocessor (may be implied by `FC`).
`CPPFLAGS`	CPP -D macro flags (See **README.macros**).
`LD`	Loader.
`LDFLAGS`	Loader flags.
`EXTRALIBS`	Extra local libraries (if any).

Some IBM SP filesystems (e.g. GPFS) cannot be used to compile Fortran modules. If the src directory is on such a filesystem, use TYPE=sp3GPFS instead of TYPE=sp3 (i.e., the configuration file is sp3GPFS_setup instead of sp3_setup). This version does the compile on a non-GPFS filesystem, which is currently set to /scratch/$(USER)/NOT_GPFS. Since all compiles use this directory, only perform one make at a time.

In addition, "make" suffix rules are required for creating object (.o) files from the program files (.c, .f, and .F) (see Figure 9). Note that the rule command lines start with a tab character.

```
#
# rules.
#

.c.o:
      $(CC) $(CPPFLAGS) $(CCFLAGS)  -c $*.c
.f.o:
      $(FC)             $(FCFFLAGS) -c $*.f
.F.o:
      $(FC) $(CPPFLAGS) $(FCFFLAGS) -c $*.F
```

Figure 9: Make suffix rules for creating object files.

10.2 Compiling the Setup Programs

10.2.1 Make_all.src

The first step in compiling the setup programs is to edit the file **Make_all.src** for the correct machine architecture (ARCH). All components of Make_all.src are hard linked together, so the file needs to be edited only once. The */src/Makefiles are configured to key on "../../config/$(ARCH)_setup" for machine-dependent definitions. For example, when running on a Linux PC, ARCH is `intel` and an individual make command might be the following:

```
make zero ARCH=intel >& Make_zero
```

10.2.2 Make_all.com

Once the Make_all.src has been edited, run **Make_all.com** from the ALL root directory using the following command:

```
csh Make_all.com.
```

This creates all of the executables in all of the source directories, except those in the archive/src subdirectory that depend on the NetCDF library. Also, programs in plot/src depend on NCAR graphics and require the ALL/bin/ncargf90 wrapper script to be defined for this machine type (See Section 14.1). Apart from these exceptions, this make should typically only have to occur once for the domain-independent setup programs. Some common source files in the ALL/*/src subdirectories have been hard linked to those in ALL/libsrc. Replicating these common files in all of the source directories avoids issues with compiler-dependent module processing.

If the user wishes to run a complete make from the source in a single source directory rather than from all of the source directories, issue the "`make clean`" command. This deletes all executables, object (.o) and module (.mod) files. A subsequent command "`csh Make_all.com`" (or "`make all`") builds all executables from scratch. The script **Make_clean.com** removes all machine specific executables but should only be required when updating to a new compiler version. Issue the "`csh Make_clean.com`" command in the ALL root directory to run Make_clean.com in each */src directory.

On a new machine type, Make_all.com should be run to recompile all the *.[Ffc] source codes to create executables ending in _machinetype, where "machinetype" is typically the output of `uname`, which is soft linked to the standard executable name. The c-shell scripts `clim_stat`, `wind_stat` and `hycom_sigma` invoke *_machinetype using a hardwired path. The path and possibly the machinetype definition may need modifying for the user's particular setup. The Gnuplot plot package is also used by `hycom_sigma,` and its location must be specified. This can be achieved by invoking the command `csh Make_all.com`. It will generate a warning if the c-shell scripts need modifying.

Make_all.com in ALL/bin does not use Make_all.src, but it should only need editing if you are running Solaris and would prefer 64-bit to 32-bit executables. Running Make_all.com in the ALL root directory invokes ALL/bin/Make_all.com.

10.3 Model Code

10.3.1 Configuration Files for Model Run

The configuration files for the model run are found in the ATLb2.00 directory under the subdirectory config. They are formatted as $(ARCH)_$(TYPE), where ARCH defines the machine architecture and TYPE is the parallelization strategy and precision (ONE, ONE4, SETUP, OMP, MPI, MPISR, or SHMEM). Table 14 provides a list of available machine-specific configuration files.

10.3.2 Compiling the Model Code

The example source directory (src_2.1.03_22_one), and scripts (expt_01.5/*.com) are currently configured for a single processor. To compile HYCOM, simply run Make.com from the src_* directory. Each executable is then created by invoking the following command:

```
./Make.com >& Make.log
```

If HYCOM is being run on a different system configuration, the script Make.com in the src_* directory will have to be edited to define $ARCH appropriately for the machine, and dimensions.h will need to be modified for different shared memory types (one, omp) and distributed memory types (mpi, ompi, shmem) (See Section 5.2). There is no need to create an executable for every parallelization technique, just for the one that you plan to actually use.

Table 14: Machine-specific configuration files available in HYCOM.

File	Description
alpha_one{4}	Compaq Alpha, single processor {,real*4}._
alpha_omp	Compaq Alpha, OpenMP._
alphaL_one{4}	Compaq Alpha Linux, single processor {,real*4}.
intel_one{4}	Intel Linux/pgf90, single processor {,real*4}.
intel_omp	Intel Linux/pgf90, OpenMP.
o2k_one{4}	SGI Origin 2000, single processor {,real*4}.
o2k_omp	SGI Origin 2000, OpenMP.
o2k_mpi	SGI Origin 2000, MPI.
o2k_shmem	SGI Origin 2000, SHMEM.
sp3_one{4}	IBM SMP Power3, single processor {,real*4}.
sp3_omp	IBM SMP Power3, OpenMP.
sp3_q64omp	IBM SMP Power3, OpenMP (64-bit memory model).
sp3_ompi	IBM SMP Power3, OpenMP and MPI.
sp3_mpi	IBM SMP Power3, MPI.
sun64_one{4}	Sun (64-bit), single processor {,real*4}.
sun64_omp	Sun (64-bit), OpenMP.
sun64_ompi	Sun (64-bit), OpenMP and MPI.
sun64_mpi	Sun (64-bit), MPI.
sun_one{4}	Sun (32-bit), single processor {,real*4}.
sun_omp	Sun (32-bit), OpenMP.
sun_ompi	Sun (32-bit), OpenMP and MPI.
sun_mpi	Sun (32-bit), MPI.
t3e_one	Cray T3E, single processor.
t3e_mpi	Cray T3E, MPI.
t3e_shmem	Cray T3E, SHMEM.

Table 15: Macros used in config/$(ARCH)_$(TYPE).

Macro	**Description**
ALPHA	Compaq Alpha (Linux or OSF).
AIX	IBM AIX.
ARCTIC	Fully global domain with Arctic dipole patch.
DEBUG_ALL	Sets all DEBUG_* macros.
DEBUG_TIMER	Printout every time the timer is called for a user routine.
ENDIAN_IO	Swap endian-ness as part of array I/O.
IA32	IA32 Linux (Intel Pentium II/III/IV or AMD Athlon).
MPI	MPI message passing (see MPISR, NOMPIR8, SERIAL_IO, SSEND).
MPISR	Use MPI_SENDRECV (vs non-blocking pt2pt calls).
NOMPIR8	This MPI does not implement mpi_real8.
REAL4	REAL is REAL*4.
REAL8	REAL is REAL*8.
RINGB	Use local synchronization for SHMEM.
SERIAL_IO	Serialize array I/O (MPI, SHMEM).
SHMEM	SHMEM put/get version (see RINGB, SERIAL_IO).
SGI	SGI IRIX64.
SSEND	Use MPI_SSEND and MPI_ISSEND (*vs.* MPI_SEND and MPI_ISEND).
SUN	SUN Solaris.
TIMER	Turn on the subroutine-level wall clock timer.
T3E	Cray T3E.
YMP	Cray YMP/C90/T90/SV1.

Table 16: Macros used in ALL/config/$(ARCH)_setup.

Macro	**Description**
REAL4	REAL is REAL*4, or (with REAL8) REAL*4 is available.
REAL8	REAL is REAL*8.
ALPHA	Compaq Alpha (Linux or OSF).
AIX	IBM AIX.
ENDIAN_IO	Swap endian-ness as part of array I/O.
HPUX	HP HP-UX.
IA32	IA32 Linux (Intel Pentium II/III/IV or AMD Athlon).
SGI	SGI IRIX64.
SUN	SUN Solaris.
T3E	Cray T3E.
YMP	Cray YMP/C90/T90/SV1.

Table 17: Macros set and used internally.

Macro	Description
BARRIER	Set a barrier (need HEADER in same routine).
HEADER	Include a header file of constants (MPI).
MPI_ISEND	MPI, either mpi_isend or mpi_issend (SSEND).
MPI_SEND	MPI, either mpi_send or mpi_ssend (SSEND).
MTYPED	MPI, type for double precision.
MTYPEI	MPI, type for integer.
MTYPER	MPI, type for real.
SHMEM_GETD	SHMEM, get double precision variables.
SHMEM_GETI	SHMEM, get integer variables.
SHMEM_GETR	SHMEM, get real variables.
SHMEM_MYPE	SHMEM, return number of this PE (0...npes-1).
SHMEM_NPES	SHMEM, return number of PE's.

11 Configuring expt_##.#

Steps for a new simulation configuration:

1. **Create a new experiment directory.** An example is ../expt_03.4. If the model output will be copied to another archive machine, the experiment data directory (e.g., expt_03.4/data) must also be created on the archive machine.

2. **Copy new_expt.com** into the /expt_03.4 directory, and edit `DO, DN, O,` and `N` to indicate the old and new directory and experiment number. Edit the `mlist` line, if necessary, to reflect the run sequence (see ../bin/mlist for further information).

3. **Run new_expt.com** by issuing the command "`csh new_expt.com`" in the /expt_03.4 directory. The .awk, .com, LIST, blkdat.input, and ports.input files, corresponding to those in the old experiment directory, will be created. The data subdirectory will also be created. The files ###W.com and ###F.com (where ### is the experiment number) are always present for consistency, but are only used for "just in time" winds and fluxes.

4. **Edit the ###.com file** to document the new experiment and reveal any changes in input filenames (e.g., bathymetry, forcing). If 6 or 12-hourly inter-annual winds/fluxes are being used the forcing is generated "just in time" as part of the run script. The files ###W.com and ###F.com are used to accomplish atmospheric forcing. The directories /data/wind and /data/flux must exist on the scratch file system to turn on the forcing.

5. **Change the run segment size.** If the run segment size has changed, edit the .awk file for the new experiment. This file can handle calendar years by setting `nd` = 365. For example, calendar year 1979 is model year 079, and each model year has 365 or 366 days (the latter in leap years only).

6. **Edit blkdat.input** for the new experiment number and model parameter changes. Refer back to Section 4.1.2 and see Appendix B for parameters in blkdat.input.

7. **Set boundary conditions.** Barotropic boundary conditions, if required, can come from the net transport through one or more "ports", under control of the input text file named ports.input `(lbflag=1)` or from a HYCOM simulation over an enclosing region (`lbflag=2`, not yet implemented). If `lbflag` = 1, editing ports.input is necessary.

8. **Change the number of vertical coordinate surfaces (`kdm`)**:

   - Create a new directory (../src_2.0.00_xx_$TYPE), with xx being the new number of vertical coordinate surfaces and $TYPE the desired type (ONE, OMP, MPI, etc.).
   - Edit dimensions.h and change the parameter `kdm` to its new value.
   - Recompile HYCOM in the source directory.
   - Edit the ".com" file in the new experiment directory to point to the correct source directory for the executable.

9. **Change the model region and/or domain size**. Dimensions.h is the only source code file that should need changing for a new region or a different number of layers. Refer back to Section 4.2 for more details.

10. In order to change to a new model region and/or domain size, see Sections 5 and 22.

12 Running HYCOM

Before beginning a HYCOM model run, the bin directory must be present in the user's primary path. The bin directory contains HYCOM commands and aliases that may be used throughout the run. **Appendix A** provides a complete list of HYCOM utility commands and their definitions. For commands without manual pages, the header of the script or the source code contains usage information. Invoking the command with no arguments will print a single line usage message.

The process of running a simulation is optimized for batch systems, but will also work interactively. The basic procedure is that each invocation of the ocean model results in a run for a fixed length of (model) time (e.g., one month, or three months, or one year, or five years). Each invocation has an associated run script that identifies which year or part year is involved (e.g., 015y005.com or 015y001a.com where 015 is the experiment number)). The initial year is indicated by y followed by three digits, and if there are multiple parts per year this is indicated by a letter following the year digits. All of the scripts mentioned in the following sections can be found in the ATLb2.00/expt subdirectory. The `msub` source codes are located in the ALL/bin subdirectory.

12.1 Generating Model Scripts

Each actual model script is created from a template script using an awk command. For example, **015.awk** modifies the template script **015.com**. The number of years per run can be changed by editing `ny` in 015.awk and `ymx` in 015.com. The 015.awk and 015.com files are presently configured for one year runs, as described by # and C comment lines therein. Actual scripts for single model jobs for the first three years, for example, could be generated manually using the following:

```
awk -f 015.awk y01=1 015.com > 015y001.com
awk -f 015.awk y01=2 015.com > 015y002.com
awk -f 015.awk y01=3 015.com > 015y003.com
```

If 015.awk were configured for six month runs (by setting `np`=2 in 015.awk; where `np` is the number of parts that the year is divided into), the two scripts for the first year could be generated manually using:

```
awk -f 015.awk y01=1 ab=a 015.com > 015y001a.com
awk -f 015.awk y01=1 ab=b 015.com > 015y001b.com
```

12.2 Running in Batch Mode

Manual generation of scripts is rarely necessary. The process has been automated for batch runs. There are several command choices for the user to perform a batch run, depending on the type of queuing system used:

`msub_codine` command and **015cod.com**	(for CODINE batch)
`msub_grd` command and **015grd.com**	(for GRD batch)
`msub_ll` command and **015rll.com**	(for LoadLeveler batch)
`msub_lsf` command and **015lsf.com**	(for LSF batch)
`msub_nqs` command and **015nqs.com**	(for NQS/NQE batch)
`msub_pbs` command and **015rll.com**	(for PBS batch).

Any of these also work with the default `msub` command `msub_csh` for interactive background runs.

These scripts read the first line of the **LIST** file generated by `mlist` (see below). The scripts either generate a new segment script (if the line is of the form 'year segment', such as '001 a', or of the form 'year', such as '001'), or they use the indicated existing script (e.g., 015y001.com). The new script is run, and upon its completion the first line is removed from LIST, and the job either exits (if LIST is empty), or cycles again (based on the number of segments it is configured to run), or is resubmitted for the next segment. The number of segments per job should be chosen based on batch run time limits, and is specified by a foreach loop in the script - this is currently five in 015lsf.com:

```
C
C --- Number of segments specified by '( SEG1 SEG2 ... )' in foreach.
C --- Use '( SEG1 )' for a single segment per run.
C
foreach seg (SEG1 SEG2 SEG3 SEG4 SEG5)
```

Therefore, the first step to running in batch mode is to generate a LIST file for a sequence of years. This is done by invoking the command `mlist`. For example,

```
  mlist 1 30 1
```

generates a list of model years 1 to 30 in steps of 1 year. Note that `mlist` will only be invocable by name if the hycom/ALL/bin directory is in your environment variable $PATH. The advantage of separating out the run list, in LIST, from the batch cycling, via `msub`, is that this gives much finer control of the run process.

The command `msub` (`msub_csh, msub_codine, msub_grd, msub_ll, msub_lsf, msub_nqs, msub_pbs`) then runs the script. For example, type

```
  msub_nqs 015nqs.com 01
```

to run the script for a NQS/NQE queuing system. In this command line, the final 2 digits identify the job number.

When running in batch (in addition to setting the number of segments per job), the batch script will need configuring to request the appropriate resources. This is batch system specific, but it usually involves editing the header of the corresponding batch script. For example, under LoadLeveler, lines that start with "#@" are interpreted by the batch system and 015rll.com is configured to run for two wall hours on a single (4-processor) node with four MPI (or OpenMP) tasks:

```
#!/bin/csh
#
#@ job_name         = XXXrll
#@ output           = $(job_name).log
#@ error            = $(job_name).log
#@ restart          = yes
#@ job_type         = parallel
#@ network.MPI      = css0,not_shared,US
#@ environment      = MP_EUILIB=us
#@ node             = 1
#@ total_tasks      = 4
#@ node_usage       = not_shared
#@ wall_clock_limit = 2:00:00
#@ account_no       = NRLSS018
#@ class            = batch
#@ queue
#
```

Some of the above lines will definitely need editing for your local setup. Note that `msub_ll` inserts the correct name on the "job_name" line, so XXXrll is ok here for any experiment number.

12.3 Dummy*.com

The dummy*.com scripts do nothing at all, and can be inserted into LIST if you want a particular alignment of runs within a .log file. For example, if there are four segments per year and 051lsf.com runs four segments, then normally one year will be in each .log file. However, if the simulation starts in the Summer the first year can still be in the entire first .log file by using the following LIST configuration: `051y001C.com 001 d dummyA.com dummyB.com 002 a 002 b 002 c 002 d`. In this case the first script, 051y001C.com, would differ from the standard 051y001c.com script (automatically generated by `001 c` in LIST) by replacing "LIMITS" with "LIMITI" to generate an initial limits file. The dummy scripts can similarly be used to maintain alignment of runs in the .log file after restarting from a model crash.

Note that if the batch system crashes, there will be RUNNING* files left in this directory that must be deleted before resubmitting the job.

12.4 Set Input Parameters

The file **blkdat.input** contains the input parameters that must be set prior to running HYCOM. Generate this file from a previously written blkdat.input file using the **blkdat.com** script. Edit the blkdat.input file for region-specific and experiment-specific parameter changes before the model is run. **Appendix B** contains definitions for model input parameters in blkdat.input.

13 Calculating Means and Standard Deviations

13.1 HYCOM_MEAN and HYCOM_STD

HYCOM contains two programs, HYCOM_MEAN and HYCOM_STD, that can be used to calculate the mean, mean-squared, and standard deviation of a sequence of archive files. HYCOM_STD calculates the standard deviation of archive files using their mean and mean-squared files that were generated by the program HYCOM_MEAN. The reason for using mean and mean-squared files to generate the standard deviation (rather than generating it from the mean and the original archives) is that this approach allows incremental calculations. For example, you can generate the annual mean and mean-square from each year of the run as it becomes available and later merge them together to form five year mean and mean-square files (and then a five year standard deviation file). This is done with the same HYCOM_MEAN program, which can used incrementally (i.e., a previously calculated mean or mean-squared can be part of the input to form extended means). For example, in eddy resolving cases it would be typical to produce and plot monthly or seasonal means and them combine them into annual and multi-year means and standard deviations.

The layered means are weighted by the layer thickness, but the mixed layer means and all non-layered means are simple means. Weighting by layer thickness is clearly the right approach for isopycnal layers (since the means are then in density space) and is equivalent to a simple mean for any constant thickness layers near the surface. However, layers that are sometimes isopycnal and sometimes near-surface (constant thickness) can be difficult to interpret. Seasonal means may help keep the two layer modes separate.

In order to run HYCOM_MEAN, the script "**010_mn+sq_0020.com**" has been provided in the subdirectory ATLb2.00/meanstd. The script "**010_std_0020.com**" is run to form standard deviations.

13.2 Plotting Mean and Standard Deviation Files

The next section on Plotting Results (Section 14) details how to plot the resulting mean and standard deviation files. Example input files, 010y020MN.* and 010y020SD.* have been provided in the ATLb2.00/plot subdirectory.

Note: Be careful when interpreting vertical sections that use depth (rather than layer number) as the vertical axis. These associate mean layer quantities with the mean location of the layer. If the layer's location or thickness is highly variable, then its mean location and thickness may not be as good a representation of the layer as you expect. For example, suppose a layer were 100 m thick in the summer but inside the mixed layer. Therefore, in the winter it might only be a 10 m z-layer. Then its mean, layer thickness weighted density would be close to its summer (isopycnal) value but its mean thickness would be 55 m which isn't close to either its winter or summer value. It could also be in a location that it almost never occupies in the water column. Using seasonal means should help reduce the impact of such hybrid coordinate variability.

14 Plotting the Results

There are several plotting options available in HYCOM. The user can generate plots from model output, archive files and also from any HYCOM ".a" file produced during the configuration process. The directory ALL/plot contains the source code for plotting HYCOM archive files and 2-D fields using NCAR graphics. Alternatively, fields can be output in several common data formats by programs in ALL/archive. The fields can then be plotted by the user's desired graphics package. See Section 17 for extracting 2-D and 3-D diagnostic field into data files (several formats) suitable for plotting by other graphics packages.

HYCOM has two standard plotting packages available for the user to plot results, HYCOMPROC and FIELDPROC. The plot program HYCOMPROC can plot x-y surface fields, x-y layers, x-z slices and y-z slices. It can also plot a "surface field only" archive file by setting `kk=1` in the source file hycomproc.f. In addition, surface mean or standard deviation files can be plotted.(see: `010srf*.IN` and `../meanstd/README.meanstd`). The directory ALL/relax/plot contains example plots that are produced from the dummy archive version of each monthly climatology using HYCOMPROC.

The program FIELDPROC, which is based on HYCOMPROC and has similar input, will plot any 2-D horizontal scalar field from a HYCOM ".a" data file. It will also plot fixed z-depth x-y plots after first running the program ARCHV2DATA3Z to interpolate hybrid layers from the archive file to a fixed depth. Once the interpolation is performed, the program FIELDPROC can be run(see Section 17.6).

Both of these plotting packages are completely region independent and can display the full domain or a sub-region. The number of plots per frame, spacing of latitude/longitude labels and grid lines, and the location of the contour label and color bar are all specified at run-time. All plots use NCAR graphics and are in logical space (i.e., every grid cell is the same size on the plot).

There are three main steps in using the HYCOM plot programs. These steps are outlined below:

1. Compile the plot programs,
2. Create a plot input file (*.IN),
3. Run the plot program.

In the following sections, these steps will be explained in detail.

14.1 Compiling the Plot Programs

Typically, all executables are created just once by editing the script **Make_all.src** for the correct `ARCH` and then issuing the command "`csh Make_all.com`" as described in Section 10, Configuring and Compiling the Model Code. However, the user can perform individual makes if needed. The plotting makefile is configured to key on ../../config/$(ARCH)_setup for machine-dependent definitions (See Section 10.1). After ascertaining that the correct setup configuration file is present, the user can issue an individual make command. For example, when running on a Linux PC, `ARCH` is "intel" and an individual make command might be

```
make hycomproc ARCH=intel >& Make_hycomproc
```

The same configuration file is used for all pre- and post-processing programs, but in HYCOM the makefile for the source files hycomproc.f and micomproc.f uses NCARGF90 in place of $(LD) in order to link in the NCAR graphics package. The NCARGF90 must be present in the user's path and consistent with $(LDFLAGS) as defined in $(ARCH)_setup. If NCARGF90 does not exist, the user must create the script by editing NCARGF77 (examples for SunOS can be found in the ALL/bin subdirectory). On some machines, a softlink from NCARGF77 to NCARGF90 will be sufficient (i.e., NCARGF77 will also work on f90 object files). If the NCAR graphics package does not exist on the user's machine, it can be downloaded from `http://ngwww.ucar.edu/`. If the user prefers to use another graphics package, the programs HYCOMDATA, ARCHV2DATA and FIELD2DATA have been provided in HYCOM (See Section 17). These programs are located in the ALL/archive directory. They provide similar functionality to HYCOMPROC and FIELDPROC, except that they output data fields in one of several common formats rather than creating plots of the fields. The user must then use the output together with the preferred graphics package to plot the results.

In the makefile, the parameter `opngks_obj` selects the default output type, which can be metafile, PostScript portrait or PostScript landscape. All three output types may be used by making the specific executables (hp_meta, hp_psp, hp_psl or fp_meta, fp_psp and fp_psl).

14.2 Input Files - *.IN

Before running either of the plotting packages, an input file must be generated by the user. An example input file is provided in Appendix C. The input file tells the plotting package what data to plot and how it should be plotted. The first line of the input text file identifies the dummy archive file to plot. For example, the only difference between the summer and winter input files 010y020s.IN and 010y021w.IN is the first line of each file:

`../expt_01.0/data/archv.0020_196_00.b`

versus

`../expt_01.0/data/archv.0021_016_00.b`

The first line identifies the archive file to plot and so must be different in every case. HYCOM 1.0 archive files, which are signaled by their filename (without an ".a" or ".b"), can also be plotted using HYCOMPROC and/or FIELDPROC in this way. Several input files and their output have been generated for the example experiment 1.0 run on the Atlantic 2.00 domain (See directories ATLb2.00/relax/plot and ATLb2.00/force/plot). These files can be used as further examples of *.IN files input to the plot programs.

The input file also allows very fine control over exactly what is plotted. For example, not all layers need to be included in the list of layer-by-layer plots. In fact, the same layer can appear more than once (giving fine control over the plot order). A negative number for the parameter `kf` indicates that the layer number should be displayed on the plot. Otherwise, the layers nominal isopycnal density is displayed. Note that the parameters `noisec` and `nojsec` (the number of z-sections to plot) must be followed by exactly the specified number of `isec` and `jsec` lines, respectively.

Table 18: Color options available using the parameter `ipalet.`

Option	Description
0	Contour lines only, no color
1	Alternate pastel shading of contour intervals
2	Use canonical sst color palette (64 intervals)
3	Use Rainer's color palette (100 intervals)
4	Two-tone shades (64 intervals)
5	NRL's 100 false color palette (100 intervals)
6	NRL's 100 inverted fc palette (100 intervals)

If the color palette is multi-color (`kpalet`>1) and a positive contour interval is specified, then the next input value is '`center`' to identify the central value of the color bar range. The actual range then depends on the number of distinct colors in the palette (either 64 or 100). The color options available in HYCOM are listed in Table 18.

14.3 Aliases

The aliases in alias.src (source alias.src) can be used to simplify the generation of plots. The usage of these aliases is straightforward. For example, the command "`fp2ps coads_airtmp.IN`" will generate the files coads_airtmp.log and coads_airtmp.ps. Any of the aliases can be used to generate plots by following this formulation: Type "*x*p2*yy* *plot*.IN" to create *plot*.log and *plot*.ps or "*x*p2*x* *plot*.IN" to create *plot*.log and X11 window. The plotting aliases and their functions are listed in Table 19.

14.4 Troubleshooting HYCOMPROC

Note that in the contouring subroutine CONREC there is a fixed size workspace buffer. If the run time error "`AREA-MAP ARRAY OVERFLOW`" appears, try increasing the size of the parameter

Table 19: Aliases for HYCOM/MICOM plotting.

Alias	Function
hp2ps	HYCOMPROC to PostScript
hp2gv	HYCOMPROC to PostScript to GhostView (on screen display)
mp2ps	MICOMPROC to PostScript
mp2gv	MICOMPROC to PostScript to GhostView (on screen display)
fp2ps	FIELDPROC to PostScript
fp2gv	FIELDPROC to PostScript to GhostView (on screen display)
hp2x	HYCOMPROC to X11
mp2x	MICOMPROC to X11
fp2x	FIELDPROC to X11

`lgthmp` in the file **conrec.f**. Similarly, "`MCS TOO SMALL`" refers to the parameter `lgthwk` in conrec.f. These parameters are currently set relatively large by default.

14.5 Plotting a Sub-region

In order to plot a subregion, the location of the sub-region is set at run time by specifying the location (`iorign,jorign`) on the full grid of (1,1) on the subregion grid and its size (`idmp,jdmp`). For the full region, `iorign`=`jorign`=1 and `idmp`=`jdmp`=0. All other input parameters can be the same for the full region and a subregion, but note that `isec` and `jsec` are with respect to the subregion rather than the full region.

14.6 Plotting MICOM Archive Files

The plotting program MICOMPROC is used to plot MICOM archive files. The source file **micomproc.f** is identical to **hycomproc.f** except that `lhycom` is false. The parameter `iexpt` must be explicitly specified (since it is not in the archive file) in the input file for MICOMPROC, otherwise it is identical to the HYCOMPROC input file. Since MICOM is in CGS and uses a N-S then W-E grid orientation, the input is immediately rotated (W-E then S-N) and converted to MKS. Note that the bottom topography and all of the input parameters are always from HYCOM.

The advantage of using essentially the same program to plot both models is that the display layout is identical and only one plot program needs to be maintained. Note also that HYCOM now has a MICOM-like isopycnal mode. This produces HYCOM archive files that are plotted using HYCOMPROC. However, the existence of MICOMPROC makes it very easy to compare isopycnal HYCOM simulations with any corresponding actual MICOM cases.

Another option is to convert MICOM (and/or HYCOM 1.0) pakked archive files to HYCOM 2.0 ".[ab]" format using the program HYCOMARCHV(See Section 17.1). The resulting files can then be plotted with HYCOMPROC .

15 HYCOM I/O Routines

Table 20 contains one line descriptions of all HYCOM I/O routines. All of these routines are assumed to be called with identical argument lists by all processors when using SPMD message passing. This is not difficult to arrange, since by default all routines are called in this manner in a SPMD run. The ZAIO routines are used to process HYCOM ".a" files, which contain array data only. The ZAGETC routine is used to process HYCOM ".b" plain text files. These are only opened on the first processor, so under MPI ZAGETC reads a line on the first processor and then broadcasts it to all other processors.

Two versions of each subroutine are provided, **mod_za_mp.F** for message passing, and **mod_za_sm.F** for a single processor (and OpenMP). The appropriate version is included in mod_za.F under control of cpp macros. The routines are configured as a module, and all HYCOM routines should start with "`use mod_za`" to allow HYCOM communication routines to be invoked when required.

A special version of each subroutine is also in ALL/libsrc/mod_za.F. This implements the identical set of subroutines, but for pre- or post-processing programs only. These are all single processor programs, and ALL/libsrc/mod_za.F is therefore similar to mod_za_sm.F except that the array size `idm, jdm` is set at run time. A related API with "za" replaced by "zb" is in ALL/libsrc/mod_zb.F. It is only used when reading in the full domain (via ZAIORD), but writing out a sub-domain (via ZBIOWR).

Table 20: HYCOM I/O Routines

Routine	**Description**
ZAGETC	Gets (read) one line from a text file.
ZAIOST	Initializes all array i/o.
ZAIOPN	Opens a file for array i/o, filename from array i/o unit.
ZAIOPE	Opens a file for array i/o, filename from environment variable.
ZAIOPF	Opens a file for array i/o, filename provided.
ZAIOPI	Is an array i/o unit open?
ZAIOCL	Closes an array i/o unit.
ZAIOFL	Flushes an array i/o unit.
ZAIOIQ	Queries an array i/o unit.
ZAIORW	Rewinds an array i/o unit.
ZAIORD3	Reads an array (3-D).
ZAIORD	Reads an array.
ZAIOSK	Skips an array read.
ZAIOWR3	Writes an array (3-D).
ZAIOWR	Writes an array.

16 HYCOM Communication Routines

Table 21 contains one line descriptions of all HYCOM communication routines. With the exception of XCHALT, all of these routines are assumed to be called with identical argument lists by all processors when using SPMD message passing. This is not difficult to arrange, since by default all routines are called in this manner in a SPMD run. Most communication routines act as implicit barriers that synchronize processor state (i.e., when a processor exits a communication routine, all processors that must communicate with it have entered the same subroutine). In addition, the subroutine XCSYNC has been provided for cases where all processors must enter a critical section of code before the first processor exits.

Two versions of each subroutine are provided, **mod_xc_mp.F** for message passing, and **mod_xc_sm.F** for a single processor. The appropriate version is included in mod_xc.F under control of cpp macros. The routines are configured as a module, and all HYCOM routines should start with "`use mod_xc`" to allow HYCOM communication routines to be invoked when required. The programs in the ATLb2.00/src/TEST subdirectory confirm that individual communication routines are working.

Table 21: HYCOM Communication Routines

Routine	**Description**
XCAGET	Converts an entire 2-D array from tiled to non-tiled layout.
XCAPUT	Converts an entire 2-D array from non-tiled to tiled layout.
XCEGET	Finds the value of `a(ia,ja)` on the non-tiled 2-D grid.
XCEPUT	Fills a single element in the non-tiled 2-D grid.
XCHALT	Emergency stops all processes, called by one process.
XCLGET	Extracts a line of elements from the non-tiled 2-D grid.
XCLPUT	Fills a line of elements in the non-tiled 2-D grid.
XCMAXR	Replaces array `a` with its element-wise maximum over all tiles.
XCMINR	Replaces array `a` with its element-wise minimum over all tiles.
XCSPMD	Initializes processor data structures, called once.
XCSTOP	Stops all processes, called by all processes.
XCSUM	Sum of a 2-D array.
XCSUMJ	Row-sum of a 2-D array.
XCSYNC	Barrier, no processor exits until all arrive (flush STDOUT).
XCTBAR	Syncs with processors ipe1 and ipe2 (internal use only).
XCTILR	Updates the tile overlap halo of a 3-D real array.
XCTMRI	Initializes timers.
XCTMR0	Starts timer.
XCTMR1	Adds time since call to XCTIM0 to timer.
XCTMRN	Registers name of timer.
XCTMRP	Prints all active timers.

17 Programs for Modifying Archive Files

HYCOM contains many programs that can be used to modify HYCOM archive files or convert them to other file formats. These programs are located in the All/archive directory. Several of these source routines are identical to the source files found in the All/plot directory, because both sets of programs are doing similar archive "processing". These are not hardlinked together, so any modifications of a program in one directory must be manually propagated to the other by the user.

Typically, all (non-netCDF) executables are created just once by editing Make_all.src for the correct ARCH and then issuing the command "`csh Make_all.com`" (See Section 10, Configuring and Compiling the Model Code). Executables that use the netCDF library (version 3.5) are created just once by editing Make_ncdf.com for the correct root directory in NCDF and then issuing the command "`csh Make_ncdf.com`". The netCDF library is at: `http://www.unidata.ucar.edu/packages/netcdf/`. These are optional, ignore Make_ncdf.com if you don't want to use NetCDF.

The following sections list the archive programs and describe their usage in HYCOM.

17.1 HYCOMARCHV

The program HYCOMARCHV converts a MICOM or HYCOM 1.0 archive file to HYCOM 2.0. It can also be used to generate a new ".b" file corresponding to an ".a" HYCOM 2.0 archive file (e.g., if the original ".b" file is corrupted). This is illustrated by the script archv_010_0021_016.com, which creates a new archive file as *.[AB]:

../expt_01.0/data/archv.0021_016_00.A
../expt_01.0/data/archv.0021_016_00.B
../expt_01.0/data/archv.0021_016_00.a
../expt_01.0/data/archv.0021_016_00.b

In this case the original ".b" file is correct and so the ".B" is identical (except that `diafx` fields are missing). If the original ".b" file is missing, then a dummy version must be provided which has the correct time step and model day but can have the wrong minimum and maximum values (e.g., Create this by editing any existing ".b" file with the same number of layers).

17.2 TRIM_ARCHV

The program TRIM_ARCHV will modify the number of layers in a HYCOM 2.0 archive file. It is primarily used in the process of generating sub-region archive files for nested boundary conditions (e.g., when the nested (sub-region) domain has a subset of the layers used by the enclosing domain). Layers can only be added at the top of the water column (e.g., for converting isopycnal cases to a hybrid vertical coordinate), or removed at the bottom of the water column (e.g., to remove dense layers that don't exist in the sub-region). See Section 19.1.4.

17.3 MRGL_ARCHV

The program MRGL_ARCHV also modifies the number of layers in a HYCOM 2.0 archive file by combining several layers into one layer. It is primarily used diagnostically (e.g., to plot a water mass that consists of several layers).

17.4 ARCHV2RESTART

The program ARCHV2RESTART creates a HYCOM 2.X restart file from an archive file. Since the archive file only contains one time level, this is duplicated to get the two time levels needed for restart. In addition an example restart file is input to obtain the few fields that are not in the archive file.

17.5 ARCHV2DATA2D and ARCHV2NCDF2D

The programs ARCHV2DATA2D and ARCHV2NCDF2D extract diagnostic layered fields in several file data formats. They are identical except that ARCHV2NCDF2D includes the option of outputting NetCDF files, and requires the NetCDF version 3.5 library. If you don't need NetCDF, then use ARCHV2DATA2D. The NetCDF library (if required) is at:
`http://www.unidata.ucar.edu/packages/netcdf/`.

The program ARCHV2DATA2D extracts 2-dimensional diagnostic fields from an archive file. All of the horizontal (x-y) fields that can be plotted with HYCOMPROC can be written out by ARCHV2DATA2D and like HYCOMPROC the output can be for a sub-region. Output can be formatted, unformatted (BINARY), or [.ab](HYCOM). Note that HYCOM .a files can be converted to "raw" files with an arbitrary data void value using the program HYCOM2RAW located in the bin subdirectory.

17.6 ARCHV2DATA3Z and ARCHV2NCDF3Z

The programs ARCHV2DATA3Z and ARCHV2NCDF3Z extract diagnostic fields at fixed Z-level depths in several file data formats. They are identical except that ARCHV2NCDF3Z includes the option of outputting NetCDF files, and requires the NetCDF version 3.5 library. If you don't need NetCDF, then use ARCHV2DATA3Z.

The program ARCHV2DATA3Z interpolates hybrid layers from an archive file to fixed depths. Like HYCOMPROC, the output can be for a sub-region. Output can be formatted, unformatted (BINARY), or [.ab](HYCOM). The ".ab" files can be plotted directly using FIELDPROC. Many plot packages can read fully raw binary files, but may not handle padded arrays. Note that HYCOM .a files can be converted to "raw" files with an arbitrary data void value using the HYCOM2RAW program in the bin subdirectory.

The data*_010_0021_016.com scripts are currently configured to write HYCOM .[ab] files and to convert the .a file to a "raw" .A file.

17.7 ARCHM* Programs

The ARCHV* programs can read all HYCOM 1.0 and HYCOM 2.X archive files. There are also corresponding ARCHM* programs that read MICOM files. For example, ARCHM2NCDF2D and ARCHM2NCDF2Z provide a NetCDF capability for MICOM archive files. Since MICOM is in CGS and uses a N-S then W-E grid orientation, the input is immediately rotated (to HYCOM's W-E then S-N grid) and converted to MKS. Note that the bottom topography and all the input parameters are always from HYCOM.

17.8 NetCDF Files

NetCDF files are self-describing and can be plotted and otherwise processed by a wide array of packages. This is our recommended format for diagnostic files. The files conform to the NetCDF Climate and Forecast (CF) metadata conventions, which were chosen because they allow for curvilinear grids. This is a new convention that is not yet widely supported, but it is an extension of the popualar COARDS conventions which means that many existing NetCDF packages provide at least partial support.

18 Sampling Transport

18.1 Sampling Transport

In the ALL/sample subdirectory, source code for sampling transport across specified sections from HYCOM archive files or mean archive files is contained. The following scripts are located in the ATLb2.00/sample subdirectory:

010M020-020.com	Script to sample a mean archive file.
010M020-020mn.com	Script to produce statistics from mean transports.
010y020-020.com	Script to sample archive files.
010y020-020mn.com	Script to produce statistics from transports.
link.com*	Script to create softlinks for topography and ./src.

The link.com script is typically edited for each new region and run just once to define softlinks to topography for that region.

18.1.1 BARO_VEL

The program BARO_VEL will extract barotropic (depth averaged) velocity at every point along a list of sections from a sequence of archive files. The resulting plain text barotropic velocity profiles can be plotted by many graphics packages, including gnuplot.

18.1.2 TRANSPORT

The program TRANSPORT is used in the script **010y020-020.com** to sample the tranport across a list of sections from a sequence of archive files. The sections are specified by end points in p-grid array space: `if, il, jf, jl` with either `if=il` or `jf=jl`. Sections are therefore either zonal or meridional. Transports are `+ve` for a net current from right to left when standing at `(if,jf)` facing towards `(il,jl)`. Note that `max(if,il)` can be greater than `idm` in periodic (global) cases and `max(jf,jl)` can be greater than `jdm` in arctic dipole patch (global) cases. It is possible to add several consecutive sections together by using special names for the sections: '`@0`' means skip this section, '`@+`': means add to next section, and '`@-`' means subtract from next section. The '`@+`' and '`@-`' transports are carried over to the first following section that does not have a name starting with '`@`'. These names are processed at the statistics production phase, by MEANTSPT, but are defined and placed into the output sample transport file by the transport program. This output file is plain text, and can be edited if statistics from a different combination of sections is desired than originally specified. For example, '`@0`' would not normally be in the original sample file but can replace the name of sections that are not desired in a particular set of statistics.

18.1.3 TRANSP_MN

The program TRANSP_MN is used in the script **010M020-020.com** to sample the tranport across a list of sections from a mean archive file. Only one mean archive file is input, but the output is a transport sample file covering all the model days that made up the mean - each with identical transport. This file has identical form to those produced by the transport program, but because all times have identical transports the statistics (from MEANTSPT) will have accurate means but zero variability.

18.1.4 TRANSP_MN_2P0

The program TRANSP_MN_2P0 will sample the tranport across a list of sections from a single mean archive file generated by the HYCOM 2.0 MEANSTD program. Only the barotropic transport (i.e., total transport across all layers) is sampled.

18.1.5 MERGETSPT

The program MERGETSPT will merge transport sample files that contain identical sections but for different time periods.

18.1.6 MEANTSPT

The program MEANTSPT is used in **010[yM]020-020mn.com** to produce mean and variability (zero variability from mean archives) statistics from a transport section data file generated by transport or TRANSP_MN (zero variability) or TRANSP_MN_2P0 (mean total transport only). As illustrated in the example scripts, it is possible to combine a set of consecutive layers in the statistics (e.g., to start with 22 layers but write statistics for only five multi-layer combinations).

19 Nesting in HYCOM

The directory ALL/subregion contains domain-independent source code for the extraction of a subregion from an archive file. The target can have the same grid resolution as the original, or be finer than the original by an integer multiplier. The general case (non-integer grid refinement and/or different grid orientation) is not yet supported. The following section details nesting at a finer resolution using the example files for subregion IASb0.50 that are provided in HYCOM. The files and scripts mentioned are located in directories ALL/subregion/src, ALL/topo/src, ATLb2.00/subregion, and IASb0.50/topo. Section 19.2 explains nesting at the same horizontal resolution, and example files for subregion IASd0.30 are provided to the user.

19.1 Nesting at Different Horizontal Grid Resolutions

IASb0.50 is a sub-region of the Atlantic 2.00 degree domain (ATLb2.00) at four times higher resolution, and illustrates how to nest a subregion in a larger HYCOM model region with different horizontal resolution. This is off-line one-way nesting, using boundary conditions similar (but not identical) to those already used for this purpose by MICOM.

19.1.1 Input Files

The IASb0.50 HYCOM model does not "know" about the ATLb2.00 domain. It expects a sequence of IASb0.50 input archive files to supply the data needed for the boundary conditions. In fact, there are two distinct sets of boundary conditions: relaxation to T/S/p in a buffer zone, and application of depth averaged flow at the open boundary. Both are input from archive files, however T/S/p is only available from full 3-D archive files and depth averaged flow is also available from surface archive files. So archive input for depth averaged flow could be more frequent than for T/S/p. Nesting in MICOM was similar, except that relaxation to velocity was also used in the buffer zone and the relaxation e-folding time on p/vel was much shorter than on T/S.

19.1.2 Setting the Resolution of the Nested Domain

The nested domain must be finer than the original by an integer multiplier (`ijgrd`). In addition, subregion `p(1,1)` and `p(idm_out,jdm_out)` must be on the original p-grid (i.e., `idm_out-1` and `jdm_out-1` must be integer multiples of `ijgrd`). The general cases of non-integer grid refinement or different grid orientation are not yet supported. Don't forget to allow for the fact that the buffer zone (typically at least 10 fine grid points) should probably be outside the region of high interest (i.e., make `idm_out` and `jdm_out` larger to allow for this).

Typically the subregion's regional.grid.com script (located in the topo directory) will be similar to that from the enclosing region. The program HYCOM_IJ2LONLAT can be used to find co-located points on the two grids. Since subregion `p(1,1)` must be on the original grid, this is usually the point to reference. For example:

```
>hycom_ij2lonlat 1 1 ~/hycom/IASb0.50/topo/regional.grid.a
 97.000W   3.997N
```

```
>hycom_ij2lonlat 1 13 ~/hycom/ATLb2.00/topo/regional.grid.a
 97.000W   3.997N
```

Obviously, HYCOM_IJ2LONLAT can't be used on the subregion until its regional.grid.[ab] has been produced. But it can be used to identify the location on the enclosing grid that will become the subregion `p(1,1)`, and this can then be used as a guide to configure regional.grid.com for the subregion.

19.1.3 Creating Sub-region Bathymetry

It is advisable to make the sub-region bathymetry and coastline exactly consistent with the coarser enclosing region, both on the open boundary and in the relaxation buffer zone. Everywhere else the bathymetry and coastline can be optimized for the higher resolution. For example, to create the IASb0.50 bathymetry do the following:

1. Generate the grid bathymetry and coastline at the finest resolution possible for the entire region. In this example, the bathymetry that is generated is depth_IASb0.50_02.[ab].
2. Interpolate the coarse enclosing bathymetry to the nested region using the program ISUB_TOPOG. This program is called by the script **depth_IASb0.50_99.com**. The script then produces depth_IASb0.50_99.[ab], which is further edited into depth_IASb0.50_98.[ab].
3. Merge the two bathymetries (02,98) using the program TOPO_MERGE, which selects the coarse depths and coastline in the buffer zone, a combination "near" the buffer zone, and the fine depths and coastline everywhere else. This program is called by the script **depth_IASb0.50_03_merge.com**, which produces the final bathymetry: depth_IASb0.50_03.[ab].

19.1.4 Generating IASb0.50 Archive Files

To generate IASb0.50 archive files from ATLb2.00, do the following:

1. Use the program ISUBREGION to create a finer-grid subregion file from the full region archive file.
2. It isn't necessary in this case, but for some sub-regions the deepest layers of the enclosing model are not present in the sub-region at all. The unnecessary layers can be removed from the sub-region archives by using the program TRIM_ARCHV.

See ATLb2.00/subregion/010y020a_IASb0.50.com for both phases, followed by the generation of a .tar bundle for all nested archive files needed for a one year IASb0.50 model run.

Once the IASb0.50 sub-region archive files are available, they can be used as boundary conditions by using HYCOM 2.1.00 or later, by creating a data/nest subdirectory on the scratch disk and setting two new `blkdat` input variables, '`bnstfq`' and '`nestfq`', non-zero and `lbflag` to 2. For example (from IASb0.50/expt_01.0/blkdat.input):

```
  1.0    'bnstfq' = number of days between baro nesting archive input
  6.0    'nestfq' = number of days between 3-d  nesting archive input
  2      'lbflag' = lateral barotropic boundary flag (0=none, 1=port, 2=input)
  3      'iniflg' = initial state flag (0=levl, 1=zonl, 2=clim, 3=archv)
```

Here, `iniflg` is also set to 3 since the "initial" restart will be from an archive file (see below). The archive to restart conversion is done off-line, so subroutine INICON is never called and `iniflg` is not used. The advantage of setting `iniflg` to 3 (versus the usual 2) is that this may remove the need for relaxation file input or limit this input to surface fields only.

In addition, the location of barotropic boundaries must be specified in the file ports.input. For example (from IASb0.50/expt_01.0/ports.input):

```
 2     'nports' = number of boundary port sections
 1     'kdport' = port orientation (1=N, 2=S, 3=E, 4=W)
39     'ifport' = first i-index
92     'ilport' = last  i-index (=ifport for north/south port)
65     'jfport' = first j-index
65     'jlport' = last  j-index (=jfport for east/west port)
 3     'kdport' = port orientation (1=N, 2=S, 3=E, 4=W)
93     'ifport' = first i-index
93     'ilport' = last  i-index (=ifport for north/south port)
 3     'jfport' = first j-index
64     'jlport' = last  j-index (=jfport for east/west port)
```

This is for two open boundaries on the northern and eastern edges of the region rectangle. Note that northern and southern boundaries are specified on the v-grid (since v-velocity is normal to these boundaries), and eastern and western boundaries are specified on the u-grid. Each boundary location is a grid point just outside the model region. Correctly positioned open boundaries appear as *'s on the iu and iv maps printed in the model run ".log" file. If the boundary locations are not specified correctly the model will stop and the iu and/or iv maps will contain 9's instead of *'s at the locations that are in error. Some errors (e.g., boundaries that are too short) can't be detected by HYCOM, so always check the iu and iv maps when initially configuring a nested domain.

Note that the nesting buffer zone relaxation is completely independent of climatology buffer zone relaxation. Both could be active in the same model run. Nesting barotropic boundary conditions cannot be used in combination with port (`lbflag`=1), specified inflow/outflow forcing.

The very first run with the nested boundaries needs a restart file consistent with the enclosing region. This is obtained from a sub-region archive file using the program RESTART_ARCHV located in directory ALL/restart. An existing restart file is required by this process. If none is available, generate a climatology over the sub-region (just as for a closed domain) and run for 1 day with closed boundaries and `iniflg`=2. Outside the nested buffer zone, the fine and coarse bathymetry and coastline may be significantly different. This might cause problems on restart. One option to ISUBREGION that might help in this case (but not needed normally) is to smooth the layer thicknesses (`smooth`=1). Using a smaller time step for the very first run might also allow it to accept a sub-optimal interpolated restart file. In general, it is a good idea to make the fine and coarse coastlines as compatible as possible - which is most easily done by always generating a fine reference coastline/bathymetry for the enclosing region and then subsampling it to the desired resolution. This won't be possible when dealing with existing bathymetries, but is the recommended way to produce new bathymetries.

19.2 Nesting at the Same Horizontal Resolution

IASd0.32 is a sub-region of ATLd0.32, and illustrates how to nest a subregion in a larger HYCOM model region with the same horizontal resolution. Nesting at the same resolution, as here, is not very interesting, but is a special case of nesting inside a coarser resolution region. Most of the process is the same for any enclosing region. This is off-line one-way nesting, using boundary conditions similar to those already used for this purpose by MICOM.

As in IASb0.50, the IASd0.32 HYCOM model does not "know" about the ATLd0.32 domain. It expects a sequence of IASd0.32 input archive files to supply the data needed for the boundary conditions. These conditions are the same as IASb0.50, explained in Section 19.1.1.

To generate IASd0.32 archive files from ATLd0.32, do the following:

1. Use the program SUBREGION to extract a subregion from a full region HYCOM 2.0 archive file, but note that the result for 3-D archives will have 26-layers. For example, see ATLd0.32/subregion/081y010.com.

2. Use TRIM_ARCHV to reduce from 26 to 22 layers. For example, see IASd0.32/plot/ATLd-0.32_081y010.com

Once the archive files have been generated, the procedure is the same as for nesting HYCOM at a finer horizontal resolution (Section 19.1.4). The archive files are used as boundary conditions, and the location of the barotropic boundaries must be specified in the file ports.input. The first run of the model uses a restart file which is obtained from an archive restart file using the program RESTART_ARCHV.

20 Parallel Processing

HYCOM can run on multiple processors in several ways:

- Using OpenMP threads on a shared memory multi-processor (e.g., Compaq ES40, Sun E10000).
- Using SHMEM on machines with a global shared memory (e.g., Cray T3E, SGI Origin 2800/3800).
- Using MPI on any homogeneous set of "connected" processors (e.g., IBM SP, clusters, all of the above machines).
- Using MPI and OpenMP on a cluster of shared memory multi-processors (e.g., IBM SP Power 3, Compaq AlphaServer SC).

The first step is to compile HYCOM for the desired parallel mode, by setting up a source code directory name ending with `_$TYPE`, where `$TYPE` is the parallelization type (`one, omp, mpi, ompi, or shmem`). The next step is to configure the batch script, 015xxx.com, for the desired configuration. Finally, the run script, 015.com, should be configured for the number of processors and how they are shared between MPI and OpenMP.

20.1 Configuring the Run Script

To configure the run script, there are two environment variables, `NOMP` and `NMPI`, that must be set depending on the type of processing being used. `NOMP` is the number of OpenMP threads, and `NMPI` is the number of MPI tasks. `NOMP` should be set to 0 for no OpenMP, or 1 for interactive OpenMP. `NMPI` should be set to 0 for no MPI. These variables are explicitly set near the top of the script, but note that this explicit setting might be modified based on batch limits. The script is currently configured to do this for LSF, Codine, and GRD batch systems. Explicit 0 values are preserved, and when both `NOMP` and `NMPI` are non-zero the `NOMP` value is preserved (i.e., `NMPI` is modified to conform to the batch limit).

When running on a single processor (`TYPE=one`), set `NOMP` and `NMPI` to 0. When using OpenMP alone (`TYPE=omp`), set `NMPI` to 0 and `NOMP` to the number of OpenMP threads (*i.e.* number of shared memory processors used). When using SHMEM, set `NOMP` to 0 and `NMPI` to the number of SHMEM tasks. When using MPI alone, set `NOMP` to 0 and `NMPI` to the number of MPI tasks. When using MPI and OpenMP, set both `NOMP` and `NMPI` above 1, and the total number of processors is then `$NOMP` times `$NMPI`. Also, be careful to ensure that if a node in the cluster runs (say) `N` MPI tasks it has at least `N*$NOMP` processors. For example, an IBM SP WinterHawk II has 4 processors per node so it can run up 4 MPI tasks per node without OpenMP (`NOMP`=0 or 1), or up to 2 MPI tasks per node with `NOMP`=2, or 1 MPI task per node with `NOMP`=3 or 4.

The model run script assumes that all data files are on a globally accessible shared file system. Some low-cost MPI-based systems (e.g., Beowulf Clusters) do not have a shared file system that is accessible by all nodes. In such cases the script must be modified to use a local file system on each node. When doing this, note that ".a" and ".b" files are both read and written on the very first MPI task only. The script will typically run on the same node as the first MPI task, which means that all ".a" and ".b" files are probably already handled correctly for local disks. However, the ".input" files may need to be broadcast to all nodes (e.g., by rcp) as part of the script.

20.2 patch.input

When using MPI or SHMEM, an additional file, patch.input, is required to control the domain decomposition into tiles. This is assumed to be located at ../topo/partit/depth_*_xxx where "xxx" is `$NMPI` as a 3-digit number.

20.3 Generating "Equal-Ocean" Tile Partitions

When generating "equal-ocean" tile partitions for use with MPI parallelization, all scripts will require renaming and/or editing for a different region or bathymetry. The following steps show how to generate the tile partitions:

1. Edit **depth_ATLa2.00_01_2d.com** to contain the desired numbers of processors.

2. Run using the command:

   `csh depth_ATLa2.00_01_2d.com >& depth_ATLa2.00_01_2d.log`

   This will create the partition text files: **depth_ATLa2.00_01.xxx.**

3. To view the partitions, generate ".ppm" bitmaps with `csh ppm.com` and display them using X-Windows Viewer (XV) (or another bitmap viewer). Note that **xbathy.pal** must be present for this to work.

4. Generate a list of partition statistics with `csh size.com`. This produces **size.lis**, for example:

npes	npe	mpe	idm	jdm	ibig	jbig	nreg	minsea	maxsea
4	2	2	57	52	34	26	0	408	439
8	4	2	57	52	24	26	0	196	222
9	3	3	57	52	33	19	0	168	195
16	4	4	57	52	34	15	0	94	110

5. The maximum values in the four columns `npe, mpe, ibig,` and `jbig` should be entered in src_*_mpi/**dimensions.h** as `iqr, jqr, idm,` and `jdm`. This allows any of the partitions to be used with the same executable.

20.4 Comparing Runs

20.4.1 Pipe.f

The source code in **pipe.f** controls the use of named pipes to compare two identical runs, the "master" vs the "slave". These typically differ only in the number of processors used (i.e., in the values of `NOMP` and `NMPI`). Often the master is on a single processor(i.e., `NMPI`=0 and `NOMP`=0 or 1). The comparison is made for every element of every array sent over the named pipe from the slave to the master. By default, most of the significant model arrays are compared after each major phase of each time step. If an error (i.e., a difference between master and slave runs) is detected, additional calls to COMPARALL or COMPARE can be added in the subroutine that introduced the error to find out exactly which OpenMP loop needs modifying. Named pipes can be difficult to use from Fortran. The current open statement has worked on all machines tried so far, but it might require modification on a new machine.

The first task of each run must be on the same node (i.e., the same O/S image), for the named pipe to work. This will almost always be the case when comparing two OpenMP runs, but can be harder to arrange for MPI runs. The easiest MPI-based named pipe comparisons to make are with a single processor master without MPI, but because this involves two different executables make sure they were created with identical compiler options (typically `TYPE=mpi` with `TYPE=one` and `TYPE=ompi` with `TYPE=omp`).

The script **024y001T_pipe.com** illustrates how to configure an OpenMP test run. It creates a named pipe and two separate data directories, dataT01 and dataT03. The named pipe is linked to PIPE_MASTER in dataT01 and to PIPE_SLAVE in dataT03. The existence of these filenames switches on the named pipe comparison. Finally the two "twin" runs scripts, **024y001T01.com** and **024y001T03.com**, are run in the background and the job waits for both to end. If no errors are detected, 024y001T_pipe.com will end normally. However, if the two runs do not produce exactly identical results the "master" will terminate but the slave will hang and must be killed manually. The location of the difference will be at the end of filename PIPE_base.out in the master scratch dataT01 directory.

The two runs scripts are identical except for the data directory used and the number of threads used. They are almost identical to a standard runs script, 024.com, except that they use 024y001T.limits as their limits file and do not copy their results back to the permanent directory.

If the file PIPE_DEBUG exists in the scratch data directory, it switches on single point (at `itest,jtest`) diagnostic printout to the .log file from every call to COMPARALL. This can be used with or without the named pipe comparison. It is simply using the name-pipe subroutine interface for diagnostic printout. The standard run script includes the following lines that can be uncommented to turn on this capability: #C #C — turn on detailed debugging. #C #touch PIPE_DEBUG

21 Demonstration run of HYCOM

The "experiment 01.5" demonstration run in the directory ATLb2.00/expt_01.5 is configured for a North Atlantic domain with 2-degree horizontal resolution and 16 coordinate surfaces in the vertical. Forcing is from COADS, plus relaxation to Levitus climatology in boundary zones, and relaxation to Levitus surface salinity. Initialization is to summer Levitus climatological interface pressures, salinity, and temperature. The mixed-layer formulation is KPP.

In order to run this example, and later on modify the code for specific applications, the user must do the following (assuming a single processor, i.e., no parallelization):

1. Compile hycom in the ATLb2.00/src_2.1.03_22_one directory, with the command:

   ./Make.com >& Make.log

   Note that this compilation is for exactly 22 layers. A 26-layer HYCOM would be compiled in a different source directory, ATLb2.00/src_2.1.03_26_one, with kdm=26 in dimensions.h.

2. If input and output files are to reside on the same machine as that from which the model is run, modify the script **015.com** located in the directory ATLb2.00/expt_01.5 to replace `pput` and `pget` by `cp` in the lines corresponding to your operating system. (See Section 2 for more information.)

3. Modify the script **015.com** to set `P` as the primary path (default is ./data subdirectory), to set `D` as the data subdirectory (default sets `D` to `P`), and to set `S` as the scratch directory (machine-dependent). If you only have one filesystem on one machine, set `S` to `$D/WORK` (as an example), so that the data and scratch directories are distinct.

4. Create or edit **LIST** to include the sequence of model years to run. For example, `../bin/mlist 1 5 1` will create a LIST file to run the first five years as five one year runs.

5. Defining a special **015y001.limits** file allows the run to start in the summer of the first year. Note that the start date in 015y001.limits should be -180.0, where 0 or `-ve` values indicate an initial run (rather than a restart).

6. Submit the demorun by issuing `../bin/msub 015nqs.com 01` where the appropriate 015xxx.com batch script should be used and the appropriate variant of `msub` for the local batch system should first be made the default via a softlink. Note that `msub_csh` is for running without a batch system, as a background interactive job, and this works with all variants of 015xxx.com.

   (Inclusion of the hycom/ALL/bin directory in the run script's command path does away with the need to specify the full path for any command in that directory. If this directory is not included in the run script's command path, spurious error messages may be generated by use of the null command "C" as a comment indicator. See Appendix A.)

7. The output files will be in the permanent data subdirectory `D` defined in 015.com. Note that this may be on a different machine, depending on how `pput` and `pget` are defined.

For more information on changing run scripts and model configuration, see Sections 12.2, and 11.

22 Setting Up HYCOM for a New Region

The following information is a summary of the steps to take for setting up HYCOM for a new region. These steps have already been incorporated into the previous sections of this document, however, this list presents the information in a concise outline format. All pre- and post-processing programs are now region-independent, but scripts typically still need editing for each new region (e.g., to include the region name and the bathymetry version).

To set up HYCOM for a new stand-alone region:

1. Pick a region name, XXXaN.NN (e.g., IASb0.50, ATLb2.00, ATLd0.32). "XXX" is an uppercase three letter primary region name, "a" is a lowercase one letter secondary region name, and "N.NN" is a three digit grid resolution description.
2. Create XXXaN.NN/topo/regional.grid.[ab] files that describe the location of the region and grid.
3. In XXXaN.NN/topo, generate a bathymetry and a landsea mask.
4. In XXXaN.NN/force, interpolate atmospheric forcing fields to this region.
5. In XXXaN.NN/expt_01.0, choose a vertical structure and implement it in the blkdat.input file.
6. In XXXaN.NN/relax/levitus, interpolate Levitus climatology to this region and bathymetry (still on the Levitus z-levels).
7. In XXXaN.NN/relax/010, interpolate Levitus onto the vertical structure chosen in the experiment's blkdat.input file. Region specific information is in EXPT.src.
8. In XXXaN.NN/src_2.1.03_MM_one (where MM is the number of layers), edit dimensions.h for this domain and number of layers and run Make.com. For multi-cpu runs, replace "_one" with the parallelization type.
9. In XXXaN.NN/expt_01.0, edit scripts as needed and run the simulation.

There are several scripts that aid in migrating region-specific scripts to a new region. These scripts include **new_force.com** in the ATLb2.00/force subdirectory, **new_topo.com** in the ATLb2.00/topo subdirectory, **new_topo.com** in the ATLb2.00/topo/partit subdirectory, and **new_expt.com** in the ATLb2.00/expt_01.5 subdirectory.

23 TECHNICAL REFERENCES

23.1 HYCOM Software Documentation

Wallcraft, A., Halliwell, G., Bleck, R., Carroll, S., Kelly, K., Rushing, K., (2002). Hybrid Coordinate Ocean Model (HYCOM) User's Manual: Details of the numerical code. Technical Report #...

23.2 General Technical Documentation

Bleck, R., (2001). An oceanic general circulation model framed in hybrid isopycnic-Cartesian coordinates. *Ocean Modeling*, **4**: 55-88.

Canuto, V.M., (2000). Ocean Turbulence: A model with shear stratification and salinity. NASA Goddard Institute for Space Studies, Unpublished Manuscript.

Halliwell Jr., G.R., Bleck, R., and Chassignet, E.P., (1998). Atlantic ocean simulations performed using a new Hybrid Coordinate Ocean Model (HYCOM). *EOS, Fall AGU Meeting.*

Halliwell Jr., G.R., Bleck, R., Chassignet, E.P., and Smith, L.T., (2000). Mixed layer model validation in Atlantic ocean simulations using the Hybrid Coordinate Ocean Model (HYCOM). *EOS, 80, OS304.*

Large, W.G., McWilliams, J.C. and Doney, S.C., (1994). Oceanic vertical mixing: A review and a model with a nonlocal boundary layer parameterization. *Rev. Geophys.*, **32**: 363-403.

Large, W.G., Danabasoglu, G., Doney, S.C., and McWilliams, J.C., (1997). Sensitivity to surface forcing and boundary layer mixing in a global ocean model: Annual-mean climatology. *J. Phys. Oceanogr.*, **27**: 2418-2447.

Levitus, S., (1982). Climatological atlas of the world ocean. NOAA/ERL GFDL Professional Paper 13, Princeton, N.J., 173 pp. (NTIS PB83-184093).

Mellor, G.L. and Yamada, T., (1982). Development of a turbulence closure model for geophysical fluid problems. *Rev. Geophys. Space Phys.,* **20**: 851-875.

Mellor, G.L., (1998). User's guide for a three dimensional, primitive equation numerical ocean model. *AOS Program Report*, Princeton University, Princeton, NJ. 34 pp.

Price, J.F., Weller, R.A., and Pinkel, R., (1986). Diurnal cycling: Observations and models of the upper ocean response to diurnal heating, cooling and wind mixing. *J. Geophys. Res.*, **91**: 8411-8427.

24 Acronyms

COADS	Comprehensive Ocean-Atmospheric Data Set project
CODINE	Computing in Distributed Network Environments
CPP	C PreProcessor
cp	Copy
ECMWF	European Center for Medium-range Weather Forecasting
ETOPO2	Earth Topography 2
GPFS	General Parallel File System
GRD	Global Resource Director
HYCOM	HYbrid Coordinate Ocean Model
IAS	Institute for Atmospheric Sciences Model
IBM SMP	A type of computer (operating system)
I/O	Input/Output
KPP	K-Profile Parameterization
LEVITUS	Annual climatology of temperature, salinity and oxygen in the world ocean.
LSF	Load Sharing Facility
MICOM	Miami Isopycnic Coordinate Ocean Model
MKS	Meters Kilograms Seconds measuring system
MLB	Mixed Layer Base
MPI	Message Passing Interface
NetCDF	Network Common Data Format
NCAR	National Center for Atmospheric Research
NCARGF77	NCAR Graphics F77
NFS	Network File System
NGDC	National Geophysical Data Center
NQS	Network Queing System
NQE	Network Queing Environment
NRL	Naval Research Laboratory
PBS	Portable Batch System
PE	Processor Element
RCP	Remote CoPy
RMS	Root Mean Square
S	Salinity
SGI	A type of computer (operating system)
SHMEM	Cray's SHared MEMory access library
T	Temperature
XV	X-Window Viewer

25 APPENDIX A

25.1 HYCOM Utility Commands

The following table lists HYCOM utility commands and their function, and gives the filename where the source code is found. These commands are located in the ALL/bin subdirectory. For commands without manual pages, the header of the script or the source code contains usage information. Also, invoking the command with no arguments will print a single line usage message.

On a new machine type, the script Make_all.com should be run to recompile all the *.[Ffc] source codes to create executables ending in `_machinetype`, where "`machinetype`" is typically the output of uname, which are softlinked to the standard executable name. The c-shell scripts `clim_stat, wind_stat and hycom_sigma` invoke *_machinetype using a hardwired path. The path, and possibly the machinetype definition may need modifing for your particular setup. The gnuplot plot package is also used by hycom_sigma and its location must be specified. This can all be achieved by invoking the command "`csh Make_all.com`". It will warn you if the c-shell scripts need modifying. The script Make_clean.com will remove all machine specific executables, but should only typically be required when updating to a new compiler version.

Table 22: HYCOM Utility Commands

Utility	**Description**
`C`	Used as a "comment" line indicator in model scripts.
`clim_stat`	Lists contents of NRL "native" climatology file.
clim_stat.1	Manual page for `clim_stat`.
clim_stat.f	Source code for `clim_stat_machinetype`.
`echo2`	Echos arguments to `stderr` (softlink).
echo2.c	Source code for `echo2_machinetype`.
`hycom2raw`	Outputs a raw copy of a HYCOM ".a" file (softlink).
hycom2raw.F	Source code for `hycom2raw_machinetype`.
`hycom_alat`	HYCOM grid statistics (softlink).
hycom_alat.f	Source code for `hycom_alat_machinetype`.
`hycom_depth`	HYCOM z-depth statistics (softlink).
hycom_depth.f	Source code for `hycom_depth_machinetype`.
`hycom_expr`	Arithmetic expression of fields (softlink).
hycom_expr.F	Source code for `hycom_expr_machinetype`.

Table 22: HYCOM Utility Commands

Utility	Description
`hycom_ij2lonlat`	Longitude, latitude of an `ip`, `jp` point on the p-grid (softlink).
hycom_ij2lonlat.F	Source code for `hycom_ij2lonlat_machinetype`.
`hycom_lonlat2ij`	Nearest p-grid point to longitude, latitude (softlink).
hycom_lonlat2ij.F	Source code for `hycom_lonlat2ij_machinetype`.
`hycom_mxthrd`	HYCOM OpenMP `mxthrd` statistics (softlink).
hycom_mxthrd.F	Source code for `hycom_mxthrd_machinetype`.
`hycom_nest_dates`	Archive dates needed for nesting (softlink).
hycom_nest_dates.f	Source code for `hycom_nest_dates_machinetype`.
`hycom_print`	Prints a sub-array (softlink).
hycom_print.F	Source code for `hycom_print_machinetype.`
`hycom_profile`	Vertical profile from an archive (softlink).
hycom_profile.F	Source code for `hycom_profile_machinetype.`
`hycom_profile_all`	Vertical profile grid from an archive (softlink).
hycom_profile_all.F	Source code for `hycom_profile_all_machinetype`.
`hycom_profile2z`	Z-space vertical profile from isopycnal profile.
hycom_profile2z.F	Source code for `hycom_profile2z_machinetype`.
`hycom_range`	HYCOM ".a" file statistics (softlink).
hycom_range.F	Source code for `hycom_range_machinetype.`
`hycom_range_ij`	Utility `hycom_range` with location information (softlink).
hycom_range_ij.F	Source code for `hycom_range_ij_machinetype`.
`hycom_rivers`	Rivers within a longitude/latitude box (softlink).
hycom_rivers.F	Source code for `hycom_rivers_machinetype`.
hycom_rivers.d	Global database of rivers.
`hycom_sea_ok`	Same sea points as a bathymetry (softlink).
hycom_sea_ok.F	Source code for `hycom_sea_ok_machinetype`.
`hycom_shift`	Identical after a shift. (softlink).

Table 22: HYCOM Utility Commands

Utility	Description
hycom_shift.F	Source code for hycom_shift_machinetype.
hycom_sigma	Illustrates HYCOM z-Sigma-z.
hycom_sigma.f	Source code for hycom_sigma_machinetype.
hycom_sigma.gnu	Gnuplot script used in hycom_sigma.
hycom_wind_date	Model day to yyyy_ddd_hh (softlink).
hycom_wind_date.f	Source code for hycom_wind_date_machinetype.
hycom_yoflat	HYCOM Mercator grid latitudes (softlink).
hycom_yoflat.f	Source code for hycom_yoflat_machinetype.
hycom_zonal	Mean and root mean square of zonal extents (softlink).
hycom_zonal.F	Source code for hycom_zonal_machinetype.
mdel	Delete a sequence of NQS jobs.
mlist	Create a list of model runs in ./LIST.
msub	Submit a sequence of jobs (softlink to default version).
msub_codine	Submit a CODINE batch job (softlink to msub_grd).
msub_csh	Submit a background csh job.
msub_grd	Submit a GRD batch job.
msub_ll	Submit a LoadLeveler batch job.
msub_lsf	Submit a LSF batch job.
msub_nqs	Submit a NQS batch job.
msub_pbs	Submit a PBS batch job.
ncargf77	NCAR Graphics f77 wrapper (softlink to script).
ncargf77.4.1.1_SunOS	Version 4.1.1 script for Solaris.
ncargf90	NCAR Graphics f90 wrapper (softlink to script).
ncargf90.4.1.1_SunOS	Version 4.1.1 script for Solaris.
pget	Get one file (softlink to default version).
pget_rcp	Get one file using RCP.
pput	Put one file (softlink to default version).
pput_rcp	Put one file using RCP.
wind_stat	Content list of NRL “native” wind file.
wind_stat.1	Manual page for wind_stat.
wind_stat.f	Source code for wind_stat_machinetype.

Table 22: HYCOM Utility Commands

Utility	**Description**
wind_stat_t3e.f	Source code for `wind_stat_t3e`.

26 APPENDIX B

26.1 blkdat.input Model Input Parameters

Table 23: blkdat.input - Model Input Parameters

Parameter	Description
iversn	HYCOM version number ×10.
iexpt	Experiment number ×10.
mapflg	Map flag (0=mercator, 1=rotated, 2=uniform, 3=beta-plane).
idm	Longitudinal array size.
pntlon	Longitudinal reference grid point on pressure grid.
reflon	Longitude of reference grid point on pressure grid.
grdlon	Longitudinal grid size (degrees).
jdm	Latitudinal array size.
pntlat	Latitudinal reference grid point on pressure grid.
reflat	Latitude of reference grid point on pressure grid.
grdlat	Latitudinal grid size at the equator (degrees).
itest	Grid point where detailed diagnostics are desired.
jtest	Grid point where detailed diagnostics are desired.
kdm	Number of layers.
nhybrd	Number of hybrid levels (0=all isopycnal).
nsigma	Number of sigma levels (nhybrd-nsigma z-levels).
dp00s	Sigma spacing minimum thickness (m).
dp00	z-level spacing minimum thickness (m).
dp00x	z-level spacing maximum thickness (m).
dp00f	z-level spacing stretching factor (1.0=no stretching).
saln0	Initial salinity value (psu), only used for iniflg¡2.
kapflg	Thermobaric compressibility flag (-1=none or thflag).
thflag	Reference pressure flag (0=Sigma-0, 2=Sigma-2, 4=Sigma-4).
thbase	Reference density (sigma units).
sigma	Layer density (sigma units).
iniflg	Initial state flag (0=levl, 1=zonl, 2=clim, 3=archv).
jerlv0	Initial Jerlov water type (1 to 5).
yrflag	Days in year flag (0=360, 1=366, 2=366J1, 3=actual).
dsurfq	Number of days between model diagnostics at the surface.
diagfq	Number of days between model diagnostics.
rstrfq	Number of days between model restart output.
baclin	Baroclinic time step (seconds), integer divisor of 86400.
batrop	Barotropic time step (seconds), integer divisor of baclin/2.
hybflg	Hybrid generator flag (0=T&S, 1=th&S, 2=th&T).
advflg	Thermal advection flag (0=T&S, 1=th&S, 2=th&T).

Table 23: blkdat.input - Model Input Parameters

Parameter	Description
slip	= +1 for free-slip, -1 for non-slip boundary conditions.
biharm	Fraction of diffusion that is biharmonic (0.0 to 1.0).
viscos	Deformation-dependent viscosity factor (nondimensional).
veldff	Diffusion velocity (m/s) for momentum dissipation.
thkdff	Diffusion velocity (m/s) for thickness diffusion.
temdff	Diffusion velocity (m/s) for temperature/salinity diffusion.
vertmx	Diffusion velocity (m/s) for momentum at MICOM Mixed Layer Base (MLB).
cbar	Root mean-square (RMS) flow speed (m/s) for linear bottom friction.
cb	Coefficient of quadratic bottom friction.
thkbot	Thickness of bottom boundary layer (m).
sigjmp	Minimum density jump across interfaces (kg/m^3).
tmljmp	Equivalent temperature jump across mixed-layer (degrees C).
thkmin	Minimum mixed-layer thickness (m).
iceflg	Ice model flag (0=none, 1=energy loan model).
mlflag	Mixed layer flag (0=none, 1=KPP, 2=KTa, 3=KTb).
pensol	KT: Activate penetrating solar radiation(0=F,1=T).
dypflg	KT: Diapycnal mixing flag (0=none, 1=KPP, 2=explicit).
mixfrq	KT: Number of time steps between diapycnal mixing calculations.
diapyc	KT: Diapycnal diffusivity x buoyancy frequency (m^2/s^2).
dtrate	KT: Maximum permitted mixed layer detrainment rate (m/day).
shinst	KPP: Activate shear instability mixing (0=F, 1=T).
dbdiff	KPP: Activate double diffusion mixing (0=F, 1=T).
nonloc	KPP: Activate nonlocal bottom layer mixing (0=F, 1=T).
difsmo	KPP: Activate horizontal smooth diffusivity coefficients (0=F, 1=T).
rinfty	KPP: Value for calculating rshear instability.
difm0	KPP: Maximum viscosity due to shear instability (m^2/s).
difs0	KPP: Maximum diffusivity due to shear instability (m^2/s).
difmiw	KPP: Background/internal wave viscosity (m^2/s).
difsiw	KPP: Background/internal wave diffusivity (m^2/s).
dsfmax	KPP: Salt fingering diffusivity factor (m^2/s).
rrho0	KPP: Salt fingering rp=(alpha*delT)/(beta*delS).
ricr	KPP: Critical bulk Richardson number.
cs	KPP: Value for nonlocal flux term.
cstar	KPP: Value for nonlocal flux term.
cv	KPP: Value for turbulent shear contribution to bulk Richardson number.
c11	KPP: Value for turbulent velocity scale.

Table 23: blkdat.input - Model Input Parameters

Parameter	Description
niter	KPP: Iterations for semi-implicit solution. (2 recommended).
clmflg	Climatology frequency flag (6=bimonthly, 12=monthly).
lbflag	Lateral barotropic boundary flag (0=none, 1=port, 2=input).
wndflg	Wind stress input flag (0=none, 1=u/v-grid, 2=p-grid).
flxflg	Thermal forcing flag (0=none, 1=origin, 2=new-flux-calculations).
relax	Activate lateral boundary nudging (0=F, 1=T).
srelax	Activate surface salinity nudging (0=F, 1=T).
trelax	Activate surface temperature nudging (0=F, 1=T).

27 Appendix C

27.1 Sample Input File for Plotting - 990_cs2.IN

```
../990/relax.0000_196_00.a ATLb2.00 000 'iexpt ' = experiment
number x10 (000=from archive file)
  0 'yrflag' = days in year flag (0=360J16,1=366J16,2=366J01,3-actual)
 57 'idm   ' = longitudinal array size
 52 'jdm   ' = latitudinal  array size
 22 'kdm   ' = number of layers
 25.0 'thbase' = reference density (sigma units)
  1 'nperfr' = number of horizontal plots per frame
 20 'lalolb' = spacing of latitude/longitude labels
 10 'lalogr' = spacing of latitude/longitude grid over land (<0 land+sea)
  4 'loclab' = location of the contour label (1=upr,2=lowr,3=lowl,4=upl)
 11 'locbar' = location of the color bar     (1[0-4]=vert,2[0-4]=horiz)
  1 'kpalet' = palete (0=none,1=pastel,2=sst,3=gaudy,4=2tone,5=fc,6=ifc)
  0 'smooth' = smooth fields before plotting (0=F,1=T)
  1 'mthin ' = mask thin layers from plots   (0=F,1=T)
  2 'i_th'   = current vector plotting spacing
  1 'iorign' = i-origin of plotted subregion
  1 'jorign' = j-origin of plotted subregion
  0 'idmp  ' = i-extent of plotted subregion (<=idm; 0 implies idm)
  0 'jdmp  ' = j-extent of plotted subregion (<=jdm; 0 implies jdm)
 -1.0 'botqq ' = bathymetry       contour int (<0 no plot; 0 from field)
 -1.0 'flxqq ' = surf. heat flux  contour int (<0 no plot; 0 from field)
 -1.0 'empqq ' = surf. evap-pcip  contour int (<0 no plot; 0 from field)
 -1.0 'ttrqq ' = surf. temp trend contour int (<0 no plot; 0 from field)
 -1.0 'strqq ' = surf. saln trend contour int (<0 no plot; 0 from field)
 -1.0 'icvqq ' = ice coverage     contour int (<0 no plot; 0 from field)
 -1.0 'ithqq ' = ice thickness    contour int (<0 no plot; 0 from field)
 -1.0 'ictqq ' = ice temperature  contour int (<0 no plot; 0 from field)
 -1.0 'sshqq ' = sea surf. height contour int (<0 no plot; 0 from field)
 -1.0 'bsfqq ' = baro. strmfn.    contour int (<0 no plot; 0 from field)
  0.0 'mthrsh' = mix lay velocity plot threshold (0 no plot; <0 contour int)
 -1.0 'bltqq ' = bnd. lay. thick. contour int (<0 no plot; 0 from field)
 -1.0 'mltqq ' = mix. lay. thick. contour int (<0 no plot; 0 from field)
 -1.0 'sstqq ' = mix. lay. temp.  contour int (<0 no plot; 0 from field)
 -1.0 'sssqq ' = mix. lay. saln.  contour int (<0 no plot; 0 from field)
 -1.0 'ssdqq ' = mix. lay. dens.  contour int (<0 no plot; 0 from field)
  0 'kf    ' = first plot layer (=0 end layer plots; <0 label with layer #)
5500.0 'depth ' = cross section plot depth
  1.0 'vstep ' = velocity contours (1.0 stairstep, to 0.0 gently curved)
 -1.0 'velqq ' = vel  contour int (<0 no temp plot; 0 from field)
```

```
 0.0 'center '= central contoured value (ignored if kpalet<2)
 0.5 'temqq ' = temp contour int (<0 no temp plot; 0 from field)
14.0 'center '= central contoured value (ignored if kpalet<2)
 0.02 'salqq ' = saln contour int (<0 no temp plot; 0 from field)
35.5 'center '= central contoured value (ignored if kpalet<2)
-1.0 'tthqq ' densmp contour int (<0 no temp plot; 0 from field)
26.0 'center '= central contoured value (ignored if kpalet<2)
 1 'mxlflg' = plot mixed layer (0=no-plot,1=plot,2=smooth-plot)
 2 'kpalet' = palete (0=none,1=pastel,2=sst,3=gaudy,4=2tone,5=fc,6=ifc)
 4 'noisec' = number of i cross sections
12 'isec'   = i cross section location
20 'isec'   = i cross section location
36 'isec'   = i cross section location
52 'isec'   = i cross section location
 4 'nojsec' = number of j cross sections
11 'jsec'   = j cross section location
20 'jsec'   = j cross section location
36 'jsec'   = j cross section location
48 'jsec'   = j cross section location
```