Development of a fast Coupled General Circulation Model (FORTE) for climate studies, implemented using the OASIS coupler

Bablu Sinha[1]
Robin S. Smith[2]

Southampton Oceanography Centre

Internal Document No. 81

Southampton Oceanography Centre

Southampton SO14 3ZH, U.K.

[1] James Rennell Division, A laboratory of the Natural Environment Research Council
[2] School of Ocean and Earth Science, University of Southampton

Abstract

We have developed an Atmosphere-Ocean Coupled General Circulation Model using physically rigorous component atmosphere and ocean models. The model has relatively low horizontal resolution but runs approximately 10 times faster than high resolution climate models (e.g. the Met Office climate model), yet includes all physical processes thought to be important to climate. The speed of the model allows many experiments to be done and allows a detailed study of processes influencing climate. Additionally the geometry, orography and bottom topography of the model are easily configurable and long timescales (decades to centuries) can be simulated with ease. The model is complementary to high-resolution climate models and will be an important tool in understanding processes occurring in high-resolution simulations.

The coupler makes use of message passing technology, is portable, and has been tested on both homogeneous and heterogeneous computer architectures.

This document is an account of the technical task of producing the coupled model using given component ocean and atmosphere models and using the generic coupler OASIS. Included are all the instructions required to install, configure, compile and run the model.

Preliminary results showing the evolution of a 100-year reference simulation of present-day climate are presented.

CONTENTS

1. Introduction

This document is an account of the technical task of producing a coupled ocean-atmosphere climate model using given component ocean and atmosphere models and using the generic coupler OASIS. Included are all the instructions required to install, configure, compile and run FORTE - Fast Ocean and Rapid Troposphere Experiment model.

FORTE is a coarse resolution ocean-atmosphere coupled general circulation model capable of integrations over timescales ranging from days to thousands of years. It is also capable of a high degree of flexibility - the model has been designed with the intention of performing experiments using highly idealised configurations, so it can accept new continental geometry, land-surface orography and ocean bottom topography in a user-friendly manner, and the horizontal resolution is also easy to change. Its principal components are ocean and atmosphere General Circulation Models (GCMs) capable of (and originally designed with the intention of) solo operation. The ocean model is MOMA (Webb, 1993), a derivative of the well-known Modular Ocean Model code which is based on the GFDL model (see Pacanowski and Griffies, 1998). The atmosphere is adapted from IGCM3 (Intermediate General Circulation Model 3) a spectral atmosphere GCM developed at Reading University (see Hoskins and Simmons, 1975, Forster et al., 2000). They are coupled together using OASIS (Terray et al., 2000), a very flexible coupler that allows the models to pass data between each other whilst running.

By and large, the basic form and parameterisations of each model remain unchanged. The greatest change is the addition of the coupling routines to each program. Every timestep, the fields to be passed are accumulated into arrays, then once per model day the average is taken and the models pass each other their data. The frequency of coupling can be altered by the user if desired (the atmospheric model can optionally allow a diurnal cycle, however for simplicity this is not used). MOMA gives a daily averaged sea-surface temperature (SST), IGCM3 passes radiative (short- and long-wave), sensible and latent heat fluxes calculated from the previous day's SST, zonal and meridional windstresses and a waterflux. These fields are first passed to the coupler, where they are treated (see below), and passed on to the other model. In this way, the surface boundary conditions for each model are supplied, and evolve with the model rather than being fixed externally.

As already mentioned, IGCM3 is a spectral model, meaning the primitive equations are solved by expressing the unknown fields (i.e. temperature, velocity etc.) as the weighted sum of a set of spherical harmonic functions. The resolution of the model is then expressed as the number of functions at which the sum is truncated (e.g. T21 means 21 spherical harmonic functions are retained). When it evaluates variables in real space, the spectral atmosphere model uses a Gaussian grid, which for T21 spectral resolution is approximately 5.625x5.625 degrees (see Bourke, 1972 and references therein for a detailed explanation of the spectral method and Gaussian grid). MOMA solves the oceanic primitive equations using finite difference methods on a simple latitude-longitude grid (the example run described later was performed at 4*4 degree resolution). Most of the coupler treatment involves accurately mapping the quantities to be passed from one grid to the other - this is done automatically via a system of areas and weights pre-supplied to the coupler. It is also important to note that the two models 'count' their latitudes in opposite senses - MOMA (and OASIS) from South to North, and IGCM3 from North to South. Both models count longitude from West to East, with the Pacific in the middle. The necessary flip of the fields is achieved in a separate process to the weighting. OASIS also deals with minor issues such as the atmosphere's requirement that the SST it sees is in Kelvin rather than degrees Celsius.

The basic technical task is to pass fields (data) from one model to another whilst both are executing (possibly on different machines). More specifically, considering the example configuration described in this document, the task is to transfer the MOMA sea surface temperature array (90x45 gridpoints, linear lat-long grid) to the IGCM3 sea-surface temperature array (64x32 gridpoints, Gaussian grid in latitude, linear grid in longitude) with interpolation from one grid to the other in between. Similarly x and y components of windstress plus total heat and freshwater fluxes must be transferred to MOMA from IGCM3.

In order to be able to do this, OASIS requires information on the grids of both models and their respective land masks. It must also employ a technique whereby separate executing processes are able to pass data to each other. It also requires information on which fields are to be passed and how frequently, and it must know the respective timestepping information of each model in order to synchronise data transfer.

2.1 Basic requirements to run the models

The user is assumed to have a working knowledge of unix/linux operating systems, Fortran 77 and Fortran 90. It is also useful to have some knowledge of unix scripts and makefiles. For reasons of brevity, only a basic introduction to OASIS, and the two component models is given here, the user is advised to read the references quoted in conjunction with this report. A knowledge of the nupdate code maintenance utility is an advantage but not essential. Similarly, the user may need to learn the basics of HDF and NetCDF data formats in order to be able to visualise output from the model. Knowledge of the message passing techniques used by OASIS (namely PVM and/or MPI) are not necessary in order to run the model, but could be useful for troubleshooting.

1 Operating systems: Both models and OASIS require either unix or linux operating systems. Linux is freeware and can be obtained from various websites. Details of setting up the model are provided in section 2.3.

2 Compilers: Both model codes are written in Fortran 77, therefore an f77 compiler must be available. If IGCM3 is to be run on a linux platform, the compiler options -r8 and -Msave are essential and must be supported by the compiler employed. OASIS is written predominantly in Fortran 90, and hence an f90 compiler is required, but contains one program unit written in C, therefore a C compiler must also be installed.

3 Message passing: OASIS requires PVM (Parallel Virtual Machine) or MPI (Message Passing Interface, see http://www.lam-mpi.org/).

4 Code management: In order to maintain the model codes and avoid haphazard changes, both MOMA and IGCM3 use the nupdate utility. Some information on how to use nupdate can be obtained from the UGAMP website (ugamp.nerc.ac.uk - search for nupdate). Contact the persons named on the UGAMP website in order to find out how to obtain the nupdate code and executable.

5 Data formats: MOMA makes use of HDF (hierarchical data format) and/or NetCDF (see http://www.unidata.ucar.edu/packages/netcdf/software.html). IGCM3 requires NetCDF only.

6 Other: In order to change the continental geography, NAG libraries are required, although, once configured (or in the example configuration supplied), NAG libraries are not required to run the model.

7 Visualisation: NetCDF files can be displayed and the data manipulated using the FERRET utility (see from http://ferret.wrc/noaa.gov/Ferret for details and downloadable software). Both NetCDF and HDF format data can also be manipulated using FORTRAN/C programs. Some software is available at SOC to do this. Contact the authors for details.

2.2 Platforms and architectures suitable for FORTE

The coupled model has been tested on an SGI Origin 2000 using both MPI and PVM message passing techniques. Another tested configuration involves a network consisting of a PC connected to an Origin 2000. The IGCM3 component runs on the PC using the linux operating system, whilst the ocean component and OASIS run on the SGI Origin 2000. In order to achieve this, it was necessary to install a version of PVM which is compatible with the rest of the system (PVM3.4 was found to work well - it is helpful if someone experienced in installing such packages is on hand to troubleshoot during installation). In our case the right version of PVM was included as part of the linux operating system software. Note also that the unix commands rsh and rlogin must be supported by the network. In addition use of the commands rsh and rlogin must not prompt for a password as the run would then stall. This requires modification of the unix/linux .rhosts file. The other essential elements are to obtain a linux compiler which supports the -r8, -byteswapio and -Msave options. Essentially any system capable of running PVM/MPI should be suitable, and each component can run on a separate machine if required. The nupdate facility, used to organise the IGCM3 and MOMA codes is useful but not essential, as both run scripts can be modified to compile the source code directly. **It is important to note that the IGCM3 model as released is set up to run on a PC running linux. Modifications to the IGCM3 and OASIS run scripts are necessary to allow the model to run exclusively on an Origin 2000 (see section 3.1.1.4). Additionally the IGCM3 utility routines in igcm3_linux/kd/linux/igcm1/lib/src should be recompiled for an SGI rather than for a linux architecture. The relevant alternative makefiles are stored in the directories aux, blas, fft and util.**

2.3 Installing the model

The model comes as four gzipped tarfiles: igcm3.tar.gz, oasis.tar.gz, auxfiles.tar.gz and moma.tar.gz. Each must be unzipped (e.g. gunzip igcm3.tar.gz), the resulting file is a tarfile and the model code and other files must be recovered using the tar command (tar xvf igcm3.tar). The resulting directory structures created are illustrated in Figures 1, 4, 7 and 9. If OASIS and MOMA are to be run on an Origin 2000 (as with the example configuration) then the Moma directory must then be moved wholesale to oasis_version_2.3/toyclim/Moma. IGCM3 can be placed in any convenient location if it is to be run on the same machine. If it is to be run on a different architecture, it must of course be located on the machine on which it is to run.

3.1 The component models

3.1.1 IGCM3

Briefly, IGCM3 is a primitive equation spectral atmosphere model, incorporating realistic coastlines and orography. The spectral method of solution was outlined in the introduction. The prognostic variables are temperature and humidity, vorticity and divergence. The wind field is obtained diagnostically from the latter two. The model is driven by solar radiation incident at the top of the atmosphere and employs a multi-band radiation scheme, including the effects of water vapour, carbon dioxide and ozone. Convection is dealt with via a Betts miller scheme with low, medium and high layer clouds. The model in standalone mode incorporates a land surface scheme consisting of a surface and deep soil temperature and moisture content. Each land gridbox is also assigned a vegetation index, which determines the roughness length and albedo. The surface boundary condition over land is thus provided by the land surface scheme and boundary layer sensible and latent heat fluxes and windstresses can be calculated using bulk aerodynamic formulae. Over ocean areas the standalone model incorporates a 25m-deep slab mixed layer model (temperature only, no salinity) with fixed albedo and roughness length. Thus boundary layer fluxes can also be calculated over ocean areas. For the coupled model the mixed layer model over ocean areas is disabled, and the atmosphere is given the SST derived from MOMA at one-day intervals. It should be noted that in the standalone model, water is not necessarily conserved as the soil moisture content is only allowed to increase to a fixed maximum value after which excess precipitation is assumed to enter the sea, which is considered an effectively infinite reservoir. In the coupled mode we wish to conserve moisture entering and leaving the ocean and so it is necessary to account for the excess precipitation over land and deposit the water at the coast as river runoff. Details of the runoff scheme employed are given below (section 3.1.1.7). See Forster et al. (2000) and references therein for details of the parameterisation schemes outlined above. The model's main drawback is that it quite slow to run and cannot be parallelised. It is therefore more efficient to run it on a single fast processor (e.g. on a desktop PC) than on an array of slow processors (e.g. an Origin 2000).

3.1.1.1 Resolution and grid

The vertical resolution is 22 sigma layers (values are listed Table 1), whilst the currently implemented spectral resolution is T21. There are 16 Gaussian latitudes per hemisphere, resulting in a real-space resolution of approximately 5.625x5.625 degrees. Changing the vertical resolution (number of layers) is possible but not recommended without major input from the model developers. It is not a trivial process, because a large number of physical processes are dependent on the number of layers (in particular the cloud parameterisation) and they may need to be changed if the number of layers are changed. On the other hand a T42 version of the model is supplied with the code and can be substituted for the T21 atmosphere without difficulty (contact B. Hoskins/Piers Forster of Reading University for information on how to do this). However the grid parameters will change and a new set of interpolation weights will have to be calculated (see section 3.2.3 on OASIS below). A list of the 32 latitudes of the model and the north and south bounding latitudes of the gridboxes is given in Table 1.

index	□-level	index	□-level
1	0.0010	12	0.2965
2	0.0055	13	0.3380
3	0.0140	14	0.3855
4	0.0280	15	0.4415
5	0.0500	16	0.5080
6	0.0815	17	0.5840
7	0.1180	18	0.6680
8	0.1540	19	0.7560
9	0.1895	20	0.8410
10	0.2245	21	0.9165
11	0.2595	22	0.9755

Table 1. IGCM3 s-levels. Level 22 is the surface level. To convert to pressure levels, the sigma levels must be multiplied by the surface pressure.

index	latitude	S-B'ndary	N-B'ndary	index	latitude	S-B'ndary	N-B'ndary
1	-85.76	-90.00	-83.21	17	2.77	0.00	5.54
2	-80.27	-83.21	-77.61	18	8.31	5.54	11.08
3	-74.74	-77.61	-72.05	19	13.84	11.08	16.62
4	-69.21	-72.05	-66.50	20	19.38	16.62	22.16
5	-63.68	-66.50	-60.95	21	24.92	22.16	27.70
6	-58.14	-60.95	-55.41	22	30.46	27.70	33.24
7	-52.61	-55.41	-49.86	23	36.00	33.24	38.78
8	-47.07	-49.86	-44.32	24	41.53	38.78	44.32
9	-41.53	-44.32	-38.78	25	47.07	44.32	49.86
10	-36.00	-38.78	-33.24	26	52.61	49.86	55.41
11	-30.46	-33.24	-27.70	27	58.14	55.41	60.95
12	-24.92	-27.70	-22.16	28	63.68	60.95	66.50
13	-19.38	-22.16	-16.62	29	69.21	66.50	72.05
14	-13.84	-16.62	-11.08	30	74.74	72.05	77.61
15	-8.31	-11.08	-5.54	31	80.27	77.61	83.21
16	-2.77	-5.54	0.00	32	85.76	83.21	90.00

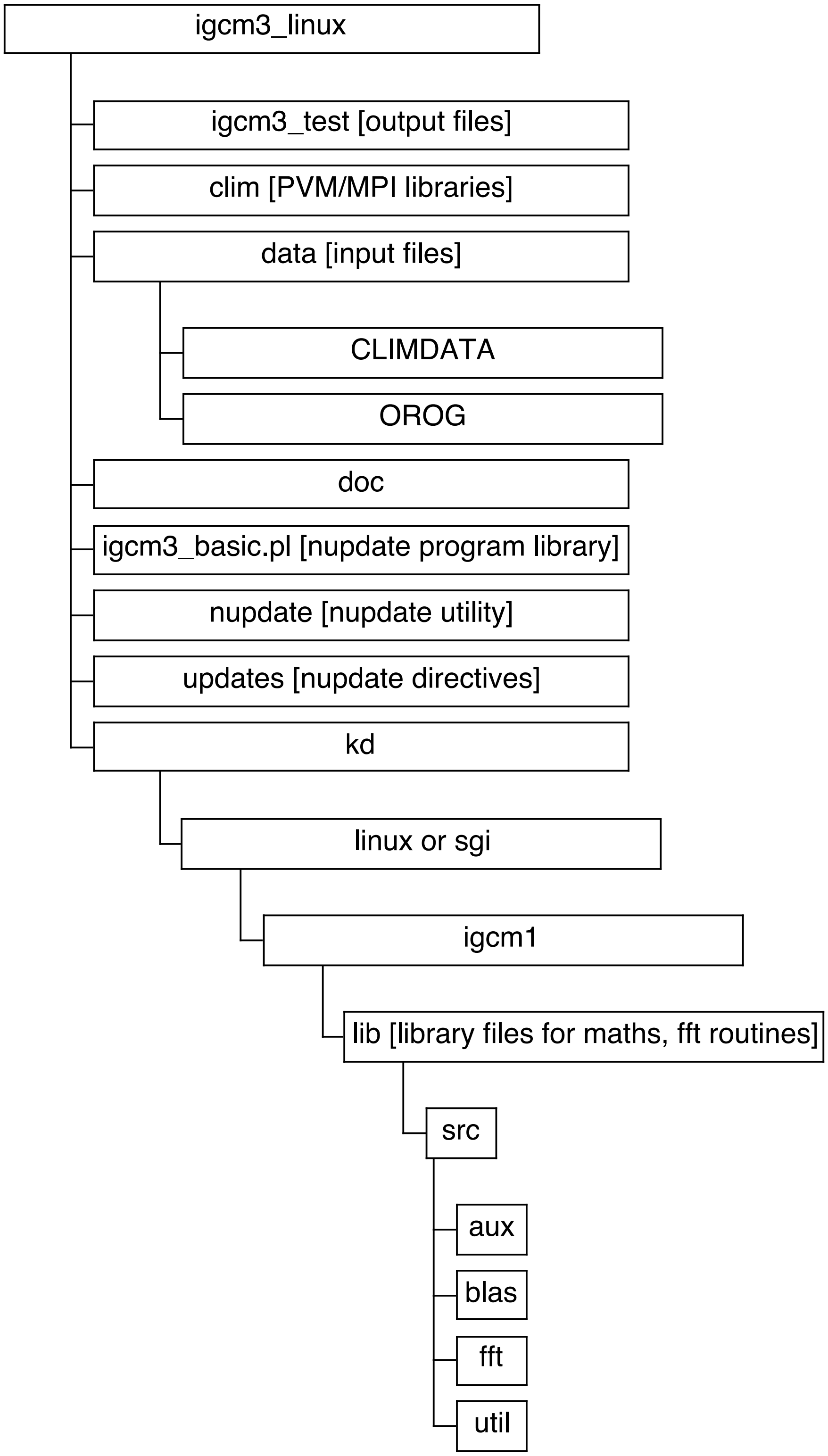
Table 2. IGCM3 latitudes and gridbox boundaries.

The IGCM3 longitudinal grid is regular with a grid spacing of 5.625 degrees, the central latitude of the first box (i=1) being 2.8125.

3.1.1.2 Directory structure

The IGCM3 (standalone) code is in directory igcm3_linux/igcm3-basic.pl and the modification files applied to create the coupled model are in igcm3_linux/updates (refer to Figure 1). The code is maintained by use of the utility nupdate, which is included in igcm3_linux/nupdate. The compilation/run file is run_igcm3-oasis. In addition to the source code IGCM3 needs a lot of ancillary files to run. These include restart files for atmosphere and boundary layer, look up tables for vegetation types and climatological values of gas concentrations. The input control file is called data but many parameters are input via namelists, set in the runfile.

Figure 1. Igcm3 directory structures


igcm3_linux
igcm3_test [output files]
clim [PVM/MPI libraries]
data [input files]
CLIMDATA
OROG
doc
igcm3_basic.pl [nupdate program library]
nupdate [nupdate utility]
updates [nupdate directives]
kd
linux or sgi
igcm1
lib [library files for maths, fft routines]
src
aux
blas
fft
util

3.1.1.3 IGCM3 land mask and orography and how to alter it

Changing orography and coastlines requires use of a special program supplied by Piers Forster of Reading University. Any changes must be implemented slowly through a series of runs to allow the model to adjust (the model can go unstable if too large a perturbation is applied). Note that in order to allow the model to work with arbitrary topography it was necessary alter the Planck function to deal with temperatures above 350K by linear interpolation. It was not found necessary to interpolate the Planck function below its current lower limit of 250K.

The IGCM3 land mask and orography (on the real-space grid of 64x32 gridpoints) is stored in a file called t21.59 which resides in directory igcm3_linux/data/OROG. It has a sister file called t21.50, which describes the land mask in spectral space. Both are necessary input files to run IGCM3. In fact the gridpoint file consists of an array of 64x33 values representing the orographic height in meters above sea level. The 17th line of 64 values, representing values exactly on the equator, is not used, instead lines 1-16 represent the first 16 latitude lines of the model (counting from the North pole southwards) and lines 18-33 represent the final 16 latitude lines (counting from the equator to the South pole). A value of zero denotes an ocean gridpoint. The land mask for IGCM3 can be altered using the following programs. It should be noted that for consistency between the atmosphere and ocean models, once the choice of land mask is defined for IGCM3, the land mask for MOMA is automatically determined.

t21.59 must be copied to auxfiles-neat before running the model. The file must then be altered either by hand or using some other method (e.g. the Fortran program readorog.f, this is not part of the standard package, but can be obtained from the authors if required). The following procedure must then be followed.

1) Compile and run the program grid_spec.f. This must use the option -r8 and requires NAG libraries (version 18 currently) to be linked. This creates as output the orography in spectral space. The output file name is fort.50 which must be renamed t21.50.

2) Copy t21.50 and t21.59 to igcm3_linux/data/OROG, remembering to save the original files.

Clearly if the IGCM3 land mask changes during this procedure then a new land mask for MOMA must be generated. This is achieved by compiling the programs in the progs directory (issue the command f77 progs/*) and running the resultant executable file (type a.out). Among other output files, the file ocean.kmtc will appear in the OUTPUTS directory. This is the new MOMA land mask/topography file, but as yet it only contains the land mask, all sea points are set to zero. The file must now be combined with a topography dataset. This can be done using the program hadint.f (input files hadcm3.tdepths and ocean.kmt) which interpolates the HadCM3 topography (contained in the file hadcm3.tdepths) onto the MOMA grid and puts in values where the MOMA grid indicates sea points. However this has the potential to create difficulties when HadCM3 and MOMA disagree on which are land points and which are sea, and the output of the program must be examined carefully and altered by hand if necessary. Once the final MOMA topography is decided on it should be copied to the Moma/Src directory and renamed ocean.kmt, ready to be used.

For the T21 model resolution the land mask is illustrated in Figure 2. The orography is illustrated in Figure 3 along with the river runoff scheme. **It is important to note that currently IGCM3 must have an island at the North Pole, since MOMA will not work without one. The necessary southward extent of the island is dictated by the latitudinal extent of MOMA's domain as detailed in section 3.1.2.3. We hope to address this in future versions of the model.**

```
1111111111111111111111111111111111111111111111111111111111111111
0000000000000000000000000000000000000000000000001111111111111000
0000000000000000111110000000000000000000000100011110001111111000
0001111000111111111111111111111110001111011111011101100111110000
0011111111111111111111111111111100011111111111111001100110000000
0011011111111111111111111100000000001000111111110001100000000000
1011111111111111111111111100000000000000011111111111110000000001
1111111111111111111111111000000000000000001111111111100000000000
1000111111111111111111111000000000000000001111111111000000000001
1100001111111111111111111000000000000000000001111111000000000000001
1111111111111111111110000000000000000000000011110100000000000001
1111111111011111111110000000000000000000000001100000000000000011
1111111011100111011100000000000000000000000000110000000000000011
1111111110000010001100000000000000000000000000001111000000000011
1111111110000000000000100000000000000000000000000111000000000011
0011111110000000001011000000000000000000000000000011100000000000
0011111100000000000010000000000000000000000000000011111100000000
0001111100000000000000000000000000000000000000000011111110000000
0001111100000000000000011000000000000000000000000001111110000000
0001111010000000000000111100000000000000000000000001111100000000
0001110000000000000000011111100000000000000000000001111000000000
0000110000000000000000011111100000000000000000000001110000000000
0000000000000000000000000110000000000000000000000001100000000000
0000000000000000000000000000000000000000000000000011000000000000
0000000000000000000000000000000000000000000000000011000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000011110111111111111110000000000000000000000001000000000000
1111111111111111111111111111100000000000111101111111100000000111
1111111111111111111111111111100000000111111111111111000011011111
1111111111111111111111111111111111110111111111111111111111111111
```

Figure 2. IGCM3 land mask.

3.1.1.4 IGCM3 scripts for compiling, editing source code and running

a) Running under linux

Unix scripts that compile and run IGCM3 reside in the top directory of IGCM3. The file run_igcm3-pgi compiles and runs the model in standalone mode (i.e. without coupling to OASIS and MOMA). This is useful for spinning up prior to coupling. The file run_igcm3-oasis compiles and/or runs the model depending on the options selected (a listing of this file is given in Appendix C, lines which require changes by the user are highlighted). For an initial run set COMPILE (line 26) equal to yes, and set RUN (line 27) equal to yes. For a restart run, where the code has been compiled previously, COMPILE should be set to no. Line 30 (USERPATH=/storage/igcm3_linux) sets the path to the model code, and must be changed appropriately. Other important lines to note are EXEC=IGCM-Linux (line 55) which sets the name of the executable to IGCM-Linux. The executable is placed in directory igcm3_test/igcm3_oasis after compilation. Note that some lines may need modification for different architectures/configurations, for example lines 35 and 36 (NETCDF_ROOT /usr/src/PGnetcdf, NETCDFDIR=$NETCDF_ROOT/lib) set the path for the NetCDF libraries. These are not supplied with the model and must be installed on the user's system. Line 67-80 (echo "*IDENT igcm-add" > updates et seq) reads in user-defined changes to the model code from the file igcm3_soc/updates/common.upd and other .upd files in nupdate format. Different files can be substituted if the user desires to modify the model code in any way. The nupdate executable is supplied with the model, but no instructions on how to use nupdate are supplied. For this see the UGAMP website as explained in section 2.1. Line 89 (&INPRN KRUN=1382400,BEGDAY=0.0,KITS=0,TSPD=64,PNU=0.02) sets the number of timesteps per model day (TSPD=64) and the number of timesteps for the run (KRUN=1382400). BEGDAY=0.0 allows the model to read in data from the last restart file, 0.0 must be reset to the day number of the restart file. Even when executing an initial run, it is recommended to use the supplied restart file, as starting from a state of rest requires a long run and is rather sensitive to initial configuration (e.g. changed orography or land mask). Line 100 (&INPOP RNTAPE=1.0, KOUNTP=1920, KOUNTH=1920, KOUNTR=1920, KOUNTE=1…) sets the intervals for output of restart files and diagnostics. KOUNTR=1920 outputs restart files every 1920 timesteps for example. The only other lines that the casual user is likely to need to change are lines 189, 192 and 195. Line 189 sets the name of the restart file to use (in this case called restart.nodate.12), line 192 sets the name of the land surface model restart file to use (restart.nodate.17), and line 195 sets the name of the vegetation file (vegetation.oasis), which contains the surface albedo/roughness length information and can be changed for different land configurations.

b) Running on an Origin 2000

All the modifications above apply, but the specific changes which need to be made are on line 45 where 'linux' must be replaced with 'sgi' in the line KD=$USERPATH/kd/linux. Also lines 147-162 will probably need to be changed as the compiler command and compilation flags for an Origin 2000 are different to those for linux. Consult the authors for details if necessary. Finally, as mentioned in section 2.2, the utility routines in igcm3_linux/kd/sgi/igcm1/lib/src need to be recompiled.

3.1.1.5 IGCM3 input files and climatologies

IGCM3 requires a large number of input files and generates a variety of output files. These are best summarised in a table (Table 3). In addition to the input files tabulated, IGCM3 uses climatological information (for example ozone concentration) which is stored in the igcm3_linux/data/CLIMDATA directory. As the casual user will be unlikely to need to change these, they are not described. However, for simulations with changes to greenhouse gas forcing and related effects changes will be necessary. For more information on this contact the authors directly.

3.1.1.6 Vegetation file

The file igcm3_linux/data/vegetation.oasis contains an array of values, one for each gridpoint. The values (1-25) determine the values of albedo and roughness length via lookup tables contained in the model (See Table 4). These values need to be modified in the event of changing the land mask configuration. This can be done by modifying the Fortran program readsvege.f, included in the same directory.

file	location	Description
t21.59*	data/ OROG	Gridpoint orography file
t21.50*	data/ OROG	Spectral orography file
vegetation.oasis	data	Vegetation index file
runoff.mask*	data/ OROG	River runoff catchment areas and outflow points
restart.nodate.12	data	Main restart file
restart.nodate.17	data	Surface restart file
data*	igcm3_test	Created by run script
gastab.lblm	data	Gas concentration profiles for radiation scheme
plfunc	data	Planck function tabulation
wv.199	data	Water vapour absorption table
gastab_nbm	data	Narrow band gas conc. profiles
oceanflux_22feb*	data	Ocean heat-flux correction (stand alone model)

Output file	Output interval	location	description
fort.2	-	igcm3_test	Standard output
flxocean	KOUNTR	igcm3_test	Flux fields on atmos. grid
fort.11	KOUNTR	igcm3_test	Main restart file (time series)
fort.19	KOUNTR	igcm3_test	Surface restart file (time series
fort.12	KOUNTR	igcm3_test	Main restart file (final)
fort.17	KOUNTR	igcm3_test	Surface restart file (final)
avtemp.nc	KOUNTH	igcm3_test	Average surface temperature
press.nc	KOUNTH	igcm3_test	Average surface pressure
temp.nc	KOUNTH	igcm3_test	Inst. 3D temperature field
humid.nc	KOUNTH	igcm3_test	Inst. surface humidity field
uwind.nc	KOUNTH	igcm3_test	Inst. 3D zonal wind
vwind.nc	KOUNTH	igcm3_test	Inst. 3D meridional wind
suwind.nc	KOUNTH	igcm3_test	Inst. surface zonal windstress
svwind.nc	KOUNTH	igcm3_test	Inst. surface meridional windstress
swlw.nc	KOUNTH	igcm3_test	Inst. net radiative flux
sensheat.nc	KOUNTH	igcm3_test	Inst. sensible heat flux
latheat.nc	KOUNTH	igcm3_test	Inst. latent heat flux
water.nc	KOUNTH	igcm3_test	Inst. water flux (inc. runoff)
fort.9	KOUNTH	igcm3_test	History file

Table 3. IGCM3 input and output files. Output files with suffix .nc are NetCDF files, all others are binary files except fort.2 which is ASCII. Input files marked with an asterisk (*) are ASCII, others are binary.

INDEX	Albedo (snow-covered)	Albedo (bare)	Roughness Length	INDEX	Albedo (snow-covered)	Albedo (bare)	Roughness Length
1 (desert)	0.8	0.1	0.001	13	0.7	0.19	0.12
2	0.8	0.75	1.0E-4	14	0.6	0.2	0.12
3	0.2	0.06	3.0E-4	15	0.8	0.2	0.12
4	0.2	0.14	1.0	16	0.7	0.12	0.12
5	0.3	0.12	1.2	17	0.7	0.17	0.12
6	0.3	0.13	1.0	18	0.7	0.19	0.12
7	0.3	0.13	1.0	19	0.7	0.19	0.12
8	0.2	0.13	1.2	20	0.7	0.25	0.12
9	0.3	0.13	1.0	21	0.4	0.18	1.5
10	0.5	0.17	0.4	22	0.8	0.15	0.12
11	0.5	0.16	0.4	23	0.7	0.12	0.12
12	0.5	0.16	0.4	24 (ocean)	0.8	0.35	3.0E-3

Table 4. Characteristics of the 24 vegetation types used by IGCM3.

3.1.1.7 River runoff

Every timestep, the variables to be passed from IGCM3 to MOMA are accumulated into separate arrays, with an averaging factor, and these averaged arrays are the quantities passed. Just before they are passed, two additional subroutines are called. One converts the latent heat and rainfall into comparable units and totals up the global difference between the two to check that water is being conserved. The other augments the precipitation field that is about to be passed to the ocean model by moving the rain over land grid points to predetermined coastal points - this is meant to simulate river runoff for the coupled model, which otherwise has no way of returning to the sea water that has fallen onto the land. In order to do this, the model grid is divided up into a number of fixed regions, with the rainfall over each region being divided equally between a number of coastal 'runoff' gridboxes for each region. The number and locations of the predetermined points, and the division into fixed regions, are based on the runoff scheme of Weaver et al., 1998. The model obtains this information from an input file called igcm3_linux/data/OROG/runoff.grid (in the same directory as the orography files t21.59 and t21.50), except for the runoff at the poles which is hard-wired into the model at present. Figure 3 depicts the catchment areas and predetermined points where the runoff appears in the ocean. The IGCM3 orography is also illustrated in this Figure.

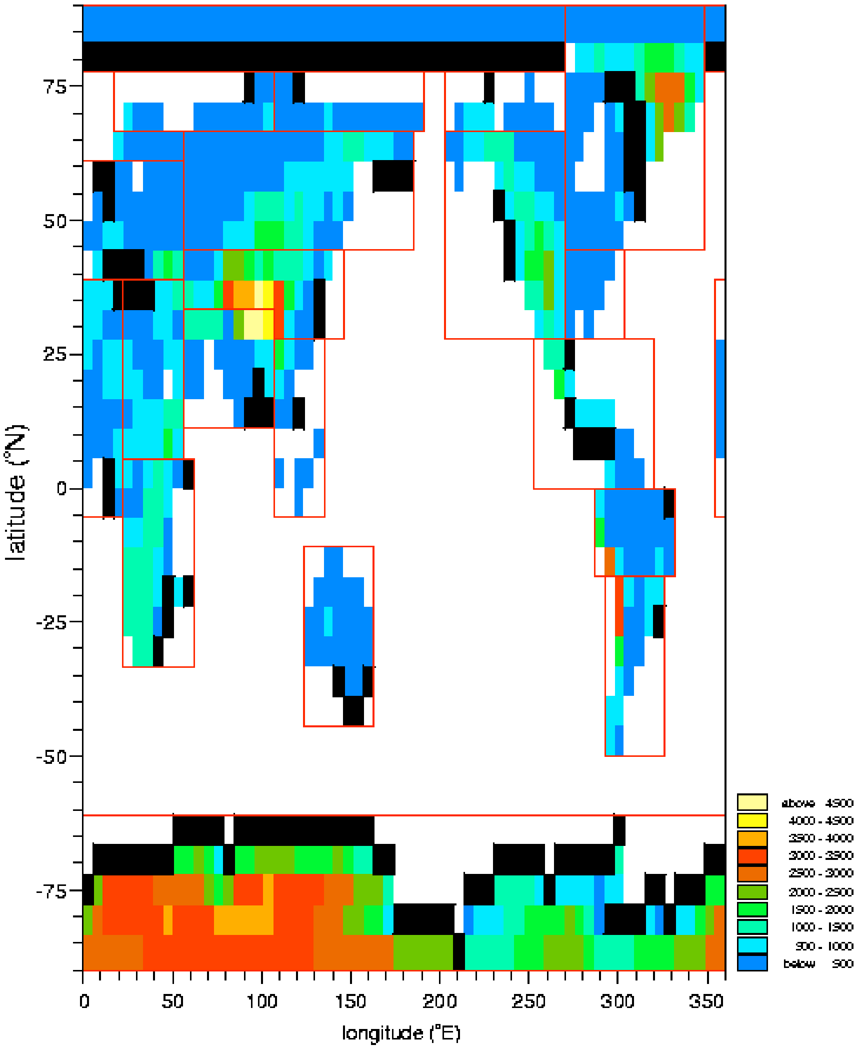
3.1.1.8 Changing the runoff scheme

This can be done by simply altering the file runoff.grid either with a Fortran program or by hand. runoff.grid is an ASCII file with 20 entries (at present) of the following form:

```
23 29 19  23
4
26 23
29 23
27 24
28 24
```

Line 1 gives the (I,J) model gridbox co-ordinates of the lower left corner of the catchment area, followed by the (I,J) co-ordinates of the upper right corner (all catchment areas are rectangular - non-rectangular regions can broken down into a number of rectangular ones). Line 2 gives the number of coastal runoff gridpoints over which to equally distribute the water falling onto the catchment area. The subsequent four lines then give the coordinates of the coastal gridpoints. The entry is then repeated 20 times for the 20 different catchment areas (excluding the poles). If the number of catchment areas needs to be changed, then the file igcm3_linux/updates/alterwater.upd needs to be altered (specifically lines 13 and 107: do iarea=1,20 must be changed). If the polar catchment areas and runoff points are to be changed then alterwater.upd would need to be changed accordingly (the authors can advise on this).

Figure 3. IGCM3 orography and runoff. Colours denote orography in meters. Red lines indicate catchment areas for precipitation over land. Black areas are ocean gridboxes affected by runoff. White areas are ocean gridboxes not affected by runoff. Runoff from a given catchment area is divided equally between the black (ocean) gridpoints contained in the area. The only exception is that the runoff gridbox in the Gulf of Mexico is considered to drain the area containing the south-eastern United States, and not the area of Central America.



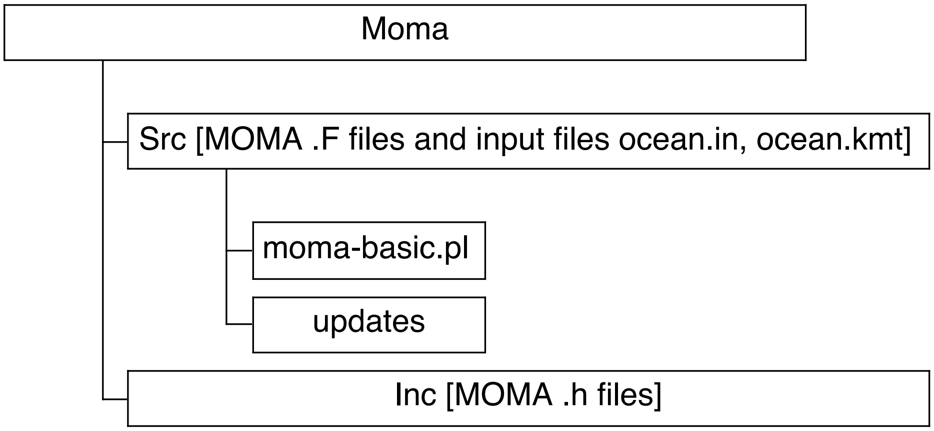
3.1.2 MOMA

MOMA is a primitive equation ocean model using standard second order finite difference methods. It employs a free-surface Bryan-Cox formulation based on MOM. The user can input realistic coastlines and topography. The model is highly efficient, particularly on multiprocessor machines (e.g. Origin 2000 on which it has been tested). The prognostic variables are the temperature, salinity, ocean currents and free surface height. The model in standalone mode is forced with climatological surface windstresses and relaxed to monthly climatological values of temperature and salinity at the surface with a timescale of 50 days. In coupled mode, the surface forcing (windstress, heat and freshwater fluxes) is given once a day from the atmospheric model via OASIS. The main limitations of the model are the lack of isopycnic mixing and an explicit mixed layer scheme, and lack of a pole patch. We hope to address these deficiencies in the future.

3.1.2.1 Directory structure

The MOMA source code is in directory oasis_version_2.3/toyclim/Moma/Src in the form of .F files (refer to Figure 4). However, to allow for easier program management, the source code is also included as a nupdate program library (in oasis_version_2.3/toyclim /Moma/Src/moma-basic.pl) which is accessed by the OASIS run script and the MOMA makefile. Update files, containing alterations of the basic program library are stored in /Moma/Src/updates. Include files are in oasis_version_2.3/toyclim/Moma/Inc.The input (control) file is ocean.in and the coastlines and topography file is ocean.kmt, both reside in oasis_version_2.3/toyclim /Moma/Src. The model depths are adapted from the Met Office Unified Model (Gordon et al 2000) as described in the IGCM3 section above.

Figure 4. Moma directory structure



3.1.2.2 Land mask and bottom topography

The procedure for creating the MOMA land mask and bottom topography file (ocean.kmt) was described above in section 3.1.1.3 and so does not need to be repeated here. However it should be noted, that in the current implementation, a MOMA gridbox is set to ocean if its *centre* falls within an atmospheric gridbox which is over land (see Figure 9). Also for the purposes of the example run, the minimum depth of topography was 1700m, to simplify treatment of the coastal shelves

3.1.2.3 Resolution and grid

level	depth	thickness	level	depth	thickness	level	depth	thickness
1	15.00	30.00	6	481.70	192.11	11	2360.06	585.36
2	53.08	46.15	7	705.13	254.76	12	2987.28	669.09
3	110.62	68.93	8	996.48	327.95	13	3693.03	742.41
4	195.04	99.93	9	1365.36	409.81	14	4464.06	799.65
5	315.33	140.63	10	1818.82	497.11	15	5281.94	836.10

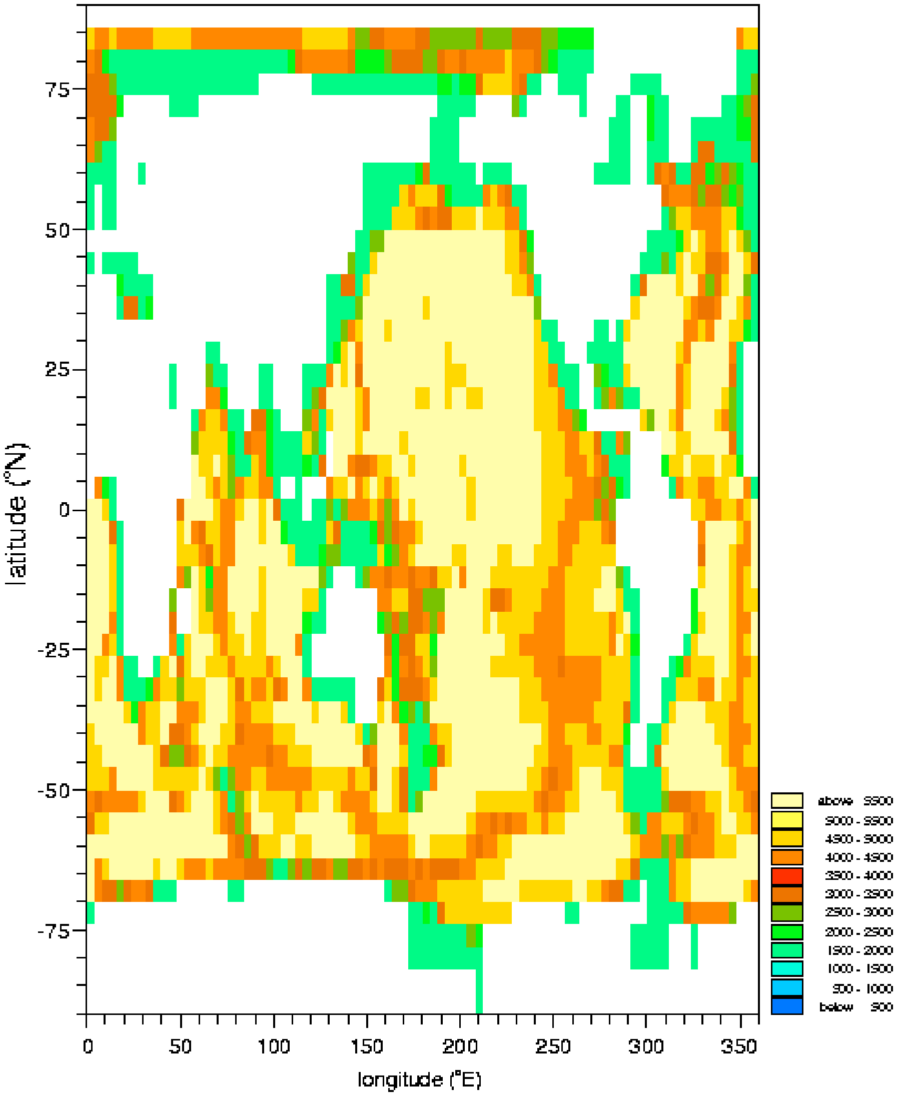
Table 5. MOMA depth levels and cell thicknesses in meters.

Currently 15 levels in the vertical are implemented. Table 5 lists the central depths and cell thicknesses for each level. These are set on lines 266-267 and 269 of the file Moma.F. The number of gridpoints (including inactive halo points) in the zonal and meridional directions, imt and jmt, are set in the file Moma/Inc/param.h. These can be altered as desired. The gridsize in the x and y directions (dxdeg and dydeg) will also need to be altered as will the latitude origin, stlat (currently set at -92.0 including the halo point). The central latitude of the first (J=1, halo point) gridbox is at 90 degrees South. The last (J=45, halo point) gridbox is centred on 86 degrees North**. Since MOMA has no pole-patch, the halo points must be set to land (kmt=0) or errors will occur.** This is usually ensured by having an island at the North Pole in the IGCM3 land mask, but could conceivably not be the case if MOMA's domain ended without overlapping the final gridbox of the IGCM3 land mask**. It is therefore the user's responsibility to make sure that the polar island in the IGCM3 land mask extends sufficiently far south so that it overlaps with the northernmost MOMA gridboxes.** The central longitude of the first (I=1) gridbox is at -2 degrees East. The starting longitude is -4 including the halo point. Information on the vertical grid is obtainable by viewing the source code moma.F (lines 276-290 list the thicknesses of the vertical gridboxes, from which their depths can easily be calculated). For 4x4 degree resolution the model land mask is illustrated in Figure 5. The bottom topography is illustrated in Figure 6.

```
111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
000000000000000000000000000000000000000000000000000000000000000000001111111111111111111000
000000000000000000000000000000000000000000000000000000000000000000001111111111111111111000
000000000000000000000001111110000000000000000000000000000000001100001111110000111111111100 0
000001111110000111111111111111111111111111111110000011111001111111011110011000111111100000
000011111111111111111111111111111111111111111100001111111111111111111100011000111000000000
000011111111111111111111111111111111111111111100001111111111111111111100011000111000000000
000011101111111111111111111111111111000000000000000010000111111111110000011000000000000000
010011111111111111111111111111111111000000000000000000000001111111111111111110000000000000
010011111111111111111111111111111111000000000000000000000001111111111111111110000000000000
111111111111111111111111111111111110000000000000000000000000111111111111111000000000000000
010000011111111111111111111111111110000000000000000000000000111111111111100000000000000000
111100000111111111111111111111110000000000000000000000000000011111111111000000000000000000
111100000111111111111111111111110000000000000000000000000000011111111111000000000000000000
111111111111111111111111111111110000000000000000000000000000000111110010000000000000000000
111111111111111100111111111111110000000000000000000000000000000011100000000000000000000011
111111111110111100011110011110000000000000000000000000000000000000110000000000000000000011
111111111110111100011110011110000000000000000000000000000000000000110000000000000000000011
111111111111110000000100001110000000000000000000000000000000000000011111100000000000000011
111111111111110000000000000000001000000000000000000000000000000000000000111100000000000011
111111111111110000000000000000001000000000000000000000000000000000000000111100000000000011
100011111111110000000000011011100000000000000000000000000000000000000000111100000000000000
000011111111000000000000000001000000000000000000000000000000000000000001111111110000000000
000001111111000000000000000000000000000000000000000000000000000000000001111111111000000000
000001111111000000000000000000000000000000000000000000000000000000000001111111111000000000
000001111111000000000000000000000111000000000000000000000000000000000000011111111000000000
000001111110110000000000000000001111110000000000000000000000000000000000001111110000000000
000001111110110000000000000000001111110000000000000000000000000000000000001111110000000000
000001111110000000000000000000111111111000000000000000000000000000000000001111100000000000
000000011000000000000000000000111111111000000000000000000000000000000000001111000000000000
000000000000000000000000000000000000111000000000000000000000000000000000001110000000000000
000000000000000000000000000000000000111000000000000000000000000000000000001110000000000000
000000000000000000000000000000000000000000000000000000000000000000000000011000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000011000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000111111001111111111111111111100000000000000000000000000000000001000000000000000
011111111111111111111111111111111111111111000000000000000111111100111111110000000000000111
111111111111111111111111111111111111111111000000000001111111111111111111100000111011111111
111111111111111111111111111111111111111111000000000001111111111111111111100000111011111111
111111111111111111111111111111111111111111111111111011111111111111111111111111111111111111
111111111111111111111111111111111111111111111111111011111111111111111111111111111111111111
```

Figure 5. MOMA land mask.

Figure 6. MOMA land mask and bottom topography. Colours denote the ocean depth in meters, white areas denote land.



3.1.2.4 MOMA compilation, editing source code and running

MOMA is compiled and run by the same script that runs OASIS (section 3.2.4) and does not require it's own script when running for the first time. Subsequently, if modifications are made to MOMA, then it would be tedious to have to recompile all the OASIS files again. In this case it is better to recompile MOMA either by using the makefile supplied in the Moma/Src directory (if the .F file has been modified directly) or by using the nupdate command (if the nupdate directive files have been modified - see sections 3.1.1.4 above and 3.2.4 below). The OASIS script can then be run with its compile options turned off.

3.1.2.5 MOMA input and output files

These are summarised in Table 6. The output files are a mixture of NetCDF, HDF and ASCII and binary files. The input files are ASCII or HDF. The MOMA control file ocean.in is more-or-less self-explanatory (reproduced in Appendix D). The main parameters which the user is likely to change are the total run length (days), the name of the restart file to use (fnrest='d0002.hdf') and the intervals at which data is output (snapd=1.0, archd= 1.,). init=.true., and restrt=.false. should be altered according to whether the run is a restart or not. Finally, the size of the baroclinic and barotropic timesteps in seconds are set by the parameters dtts=3600.0, dtuv=3600.0, dtbt = 100.0.

Input file	location	description
ocean.in	Moma/Src	Control file
ocean.kmt or a restart file (e.g. d00001.hdf)	Moma/Src	orography file or restart file

Output file	Output interval	location	description
fort.61	-	toyclim/wkdir	Standard output
d00002.hdf	archts	toyclim/wkdir	Main restart file (time series)
sstatmos	archts	toyclim/wkdir	Surface temp. on ocean grid
temp.nc	archsnap	toyclim/wkdir	Inst. 3D temp.
sal.nc	archsnap	toyclim/wkdir	Inst. 3D salinity
uvel.nc	archsnap	toyclim/wkdir	Inst. 3D zonal velocity
vvel.nc	archsnap	toyclim/wkdir	Inst. 3D meridional velocity

Table 6. MOMA input and output files. Files with suffix .nc are NetCDF format, those with suffix .hdf are HDF format. sstatmos is binary, all others are ASCII format.

3.1.2.6 Ice parameterisation

Every model day, values for the heat fluxes (radiative, latent and sensible), water flux and winds are imported from the coupler, and in the routine setvbc where the top and bottom boundary conditions for the model are set these values are used instead of those from the internal data set.

Since the coupled model lacks a sea-ice model a crude ice parameterisation is also included here - if the sea surface temperature is less than -1.96 degrees then the ocean is insulated from all atmospheric influences for that grid point.

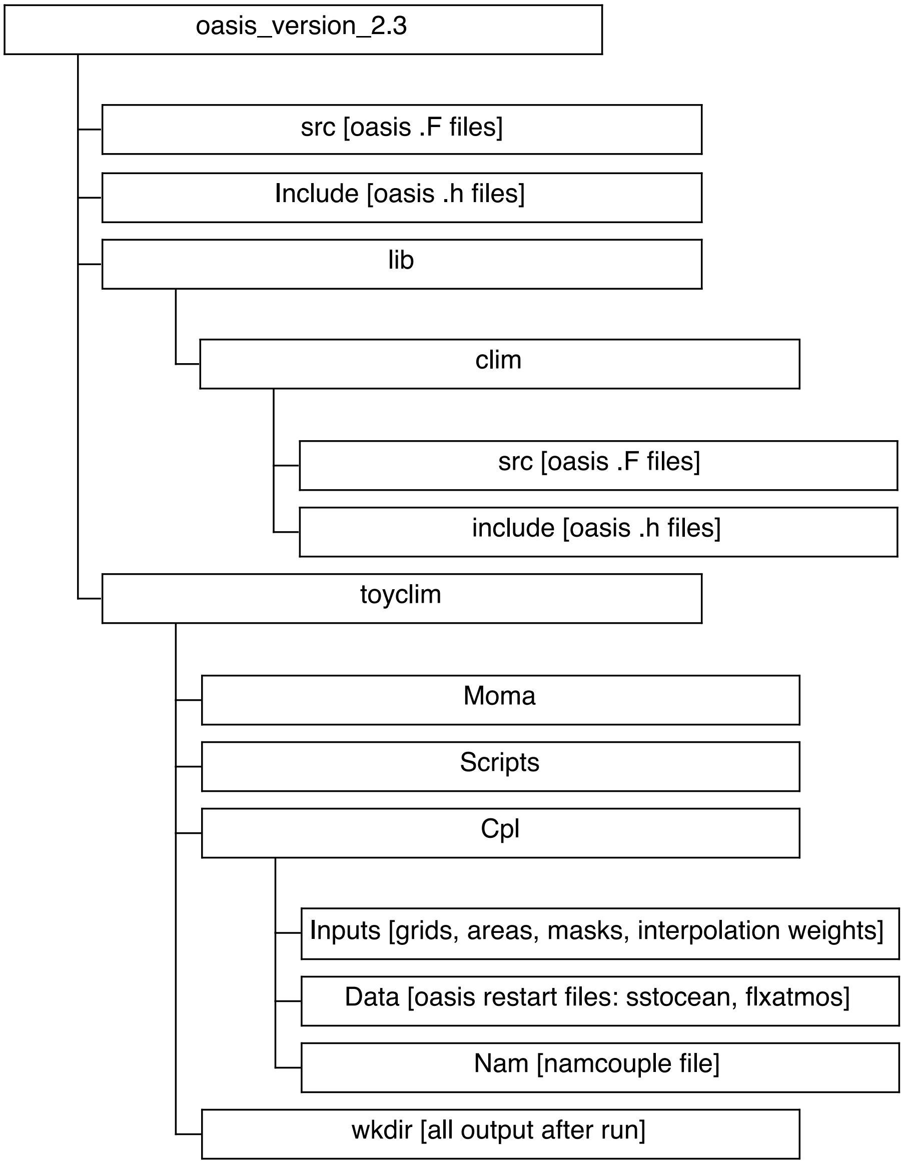
3.2 The OASIS coupler

The generic coupler OASIS (Terray et al, www.cerfacs.fr) is designed to link the processes running the two models (atmosphere and ocean). There is a choice of 5 communications techniques given in OASIS. We have used the CLIM technique (based on message passing using PVM/MPI-2) and will concentrate on this technique in what follows. It should be noted that using the MPI-2 message passing rather than PVM has not yet been successfully implemented at SOC, the OASIS team have recently released OASIS version 2.5 which uses MPI-1. The PVM library allows different unix processes to become members of a group. A library of Fortran routines (e.g. pvm_send) can then be incorporated into the Fortran code, which can specify which processes to send or receive from, and which data to send (e.g. the process id and name, and an array of sea surface temperature). This will be clearer after reading section 3.2.7 below.

3.2.1 Directory structure

The OASIS directory structure is depicted schematically in Figure 7. The model codes are stored in the OASIS directory (oasis_version_2.3/src). Note that OASIS versions 2.4 and 2.5 are now also available from the OASIS website www.cerfacs.fr). Moma should be moved wholesale to oasis_version_2.3/toyclim/Moma before starting to run the model. Running and output generally occur in toyclim/wkdir, the exception being IGCM3, which outputs to its own directory.

Figure 7. Oasis directory structure
(only relevant directories are shown for conciseness)



3.2.2 Input and output files

OASIS requires the following input files: namcouple, sstocean, flxatmos, masks, grids, areas, maskr and at31topa (see Table 7). Examples of these files come with the OASIS release. For users to generate new files for their own applications, programs have been written in auxfiles-neat. oasis_version_2.3/toyclim/Cpl/Data contains masks, areas, grids, and at31topa.

oasis_version_2.3/toyclim/Cpl/Nam contains the namcouple control file. oasis_version_2.3/toyclim/Cpl/Inputs contains the restart files flxatmos and sstocean. After a run, all the OASIS output files reside in the oasis_version_2.3/toyclim/wkdir directory. The output files are cplout (OASIS diagnostic output), oasis.timex (unix system timing information for the run) and Oasis.prt (PVM/MPI diagnostics). In order to begin a restart run, the only OASIS input files that require changing are namcouple (on line 65 if the length of the run is to change - see next section for more details) and the restart files sstocean and flxatmos. These must be obtained from the output files of MOMA and IGCM3 respectively. MOMA produces an output file called sstatmos (surface fields on ocean grid at last timestep) which can be renamed and used as the new restart file in place of sstocean whilst IGCM3 produces fluxocean (flux fields on atmospheric grid at last timestep) which must be renamed and used as the new restart file in place of flxatmos.

Input file	location	description
namcouple	Cpl/Nam	Control file
sstocean	Cpl/Inputs	Restart file for ocean SST
flxatmos	Cpl/Inputs	Restart file for atmos. Heat/water fluxes
masks	Cpl/Data	Model land masks
grids	Cpl/Data	Model latitudes and longitudes
areas	Cpl/Data	Areas of model gridboxes
maskr	Cpl/Data	Reduced-grid information (not used)
at31topa	Cpl/Data	Interpolation weights

Output file	location	description
cplout	Toyclim/wkdir	Standard output
Oasis.prt	Toyclim/wkdir	PVM/MPI diagnostics
oasis.timex	Toyclim/wkdir	Unix system diagnostics

Table 7. OASIS input and output files. Apart from namcouple, all input files are in binary. Refer to the OASIS manual for an explanation of their structure. Both output files are ASCII.

3.2.3 Generation of OASIS auxiliary files

These need to be present in the working directory before the models run. The script launching the application will generally copy these files from the directory they reside in to the working directory. However, for each configuration of the grids in the model, the auxiliary files must be generated anew.

Procedure for calculating auxiliary files (and MOMA land mask) - refer to Figure 8.

1. Copy the land mask/orography file T21.59 to directory auxfiles-neat
2. In directory progs edit file parameter.h, putting in the correct number of gridpoints for MOMA (nxo and nyo), and altering values of the southernmost latitude (yoffs) and westernmost longitude(xoff) and the x- and y- grid thicknesses (dxo and dyo). Changing the resolution of IGCM3 is more complicated and the authors of this document should be consulted if this is required.

3. In directory auxfiles-neat compile programs: f77 progs/*.f
4. Run resulting executable: a.out
5. Copy at31topa to at31topa in directory Cpl/Data, ready for reading by OASIS.
6. Copy areas to areas in directory Cpl/Data, ready for reading by OASIS.
7. Copy grids to grids in directory Cpl/Data, ready for reading by OASIS.
8. Merge ocean.kmtc with HadCm3 topography and copy to Moma/Src/ocean.kmt (see section 3.1.1.3)
9. Copy masks to Cpl/Data/masks, ready for reading by OASIS.

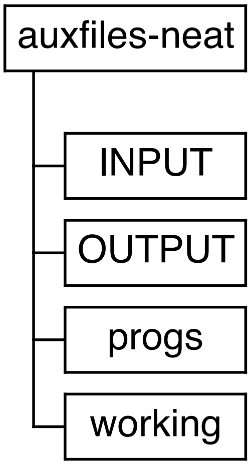
The names of the subroutines (in directory progs) creating specific files are as follows in case the user wishes to make modifications:

mozaicweights.f - creates interpolation weights file at31topa (see Appendix A).

makeigcm3mask.f and readigcmcoords.f - together create land mask file ocean.kmtc for MOMA based on IGCM3 land mask.

writeareas.f, writegrids.f, writemasks.f - respectively create the areas, grids and masks files required as input files to OASIS.

Figure 8. Auxfiles-neat directory structure



3.2.4 OASIS scripts and namcouple file

The run script for use with the O2000-linux architecture configuration is oasis_version_2.3/toyclim/Scripts/script.4deg.linux_atm (see appendix E). The lines that the user needs to be aware of are highlighted. Some of these will potentially need to be changed depending on the user's requirements. Firstly HDF_ROOT needs to already be defined (line 38) - normally HDF can be set up in the user's .cshrc file, alternatively a line setting HDF_ROOT can be inserted before line 38. PVM_ROOT (line 49) should be altered to point to the location of PVM on the user's system. Similarly CDF_ROOT must point to the location of NetCDF on the user's machine. HOMEOA (line 79) must be altered to point to the location of OASIS (e.g. simply ~/oasis_version_2.3 if OASIS is installed in the user's home directory). Lines 207-210 set the files containing the update commands to alter the MOMA source code. The user may need to add more commands of the same sort if modifications to the basic code is required (for example if the user wants to put a passive tracer in MOMA). The location of nupdate on the user's machine must be substituted in line 212. Lines 251, 258 and 260 copy input files for OASIS from the directories where they are stored to the working directory. For example oasis_version_2.3/toymodel/Cpl/Data/grids.4t is copied to the file grids in oasis_version_2.3/toyclim/wkdir. If the user has generated new files, then the names of the files to be copied must be altered. The correct namcouple file (in this case namcouple.moma.igcm) must be supplied to the script (line 266). Finally, the PVM hostfile must know the IP address of the machine running IGCM3 (line 289), 72-39 in this specific case. This number must be changed to the last two numbers in the user's IP address, for example If the IP address is 139.166.242.44, the line should be changed to echo 242-44 >> hostfile.$PVM_ARCH. Note however, if running the whole model (including IGCM3) on an Origin 2000, that this line should be deleted.

The namcouple file itself is reproduced in appendix F. The user will rarely need to make radical changes to namcouple, the main ones being changing the length of the run (line 65); and the numbers of gridpoints in the ocean model (lines 134, 190, 212, 237, 263, 288, 313, 335, 357, 379). For more advanced changes (e.g. adding/removing a new exchange field, changing the interpolation/extrapolation techniques, or changing the coupling frequency) it is recommended that the OASIS manual is consulted.

3.2.5 Summary of fields exchanged by OASIS and OASIS symbolic names

We now come to the actual fields passed by OASIS from model to model, and the mechanics of conversion between grids. OASIS maintains a list of all 2D arrays which are passed to it from the models and which it sends to the models. Each array has a symbolic name and they are connected up in pairs. For example the ocean model code has a real*8 array sstoc of dimension 92, 45 which contains the daily averaged sea surface temperature, whilst the atmosphere model has an array zsst which has dimension 64, 32 which contains the surface temperature used by the atmosphere model. OASIS gives sstoc the symbolic name SOSSTSST and zsst the symbolic name SISUTESU and ensures that when it receives array sstoc from MOMA, it receives it as SOSSTSST. From its internal list it then knows it has to interpolate SOSSTSST to SISUTESU and then send SISUTESU to IGCM3. The atmosphere receives the data as array zsst and can then carry on integrating. In this way two fields are regularly sent from MOMA to IGCM3 (the SST and one dummy field which is currently not used for anything, but could be used for, say, sea-ice in the future). Eight fields are passed from IGCM3 to MOMA. These are the latent heat flux, net radiative heat flux (long wave and short wave), net water flux, sensible heat, and the u- and v- components of the windstress twice. Therefore the second field ocean field and the last two atmosphere fields are redundant in the current set-up. Note that the windstresses are transferred from the atmosphere grid to the T-grid of the ocean (not the U-grid). This is a minor error, which should be rectified in the future. The importance of the OASIS symbolic names to the casual user is that they are used to identify the fields in the namcouple file.

The exchange fields, in the correct order that they're swapped (OASIS ordering) are summarized in Table 8.

Field number	Field description	MOMA array	Symbolic name	Symbolic name	IGCM3 array
1	SST	sstoc	SOSSTSST	SISUTESU	zsst
2 (not used)	Sea-ice	sieoc	SOICECOV	SIICECOV	glace
3	Latent heat flux	heatlat	CONSFTOT	SONSHLDO	heatlat
4	Radiative heat flux	swlw	COSHFTOT	SOSHFLDO	swlw
5	Water flux	water	COWATFLU	SOWAFLDO	exwater
6	Sensible heat flux	sensible	CORUNOFF	SORUNOFF	sensible
7	U-wind/u-grid	ztaux	COZOTAUX	SOZOTAUX	taux
8	V-wind/v-grid	ztauy	COMETAUY	SOMETAUY	tauy
9 (not used)	U-wind/v-grid	ztaux	COZOTAUV	SOZOTAUV	taux
10 (not used)	V-wind/u-grid	ztauy	COMETAUU	SOMETAUU	tauy

Table 8. OASIS exchange fields.

3.2.6 OASIS field treatments

In addition to passing fields from one model to another, OASIS performs a number of operations on each field before passing the data across. These include consistency checks, extrapolation of values over land points, interpolation from one grid to another, converting degrees Celsius to Kelvin, and changing the North-South ordering of the arrays since the models count their latitudes in opposite senses. For a full description of these operations, consult the OASIS manual. Here we note that each operation has a name and requires a number of parameters. The namcouple file has an entry for each pair of exchanged fields together with the list of operations performed during the exchange. A brief description of each operation appears below, together with the values of the parameters required.

MASK, EXTRAP: screen off land points and replace with extrapolations from sea values.
Parameter values: 999.999 NINENN 4 1 2

MOZAIC: performs an interpolation between the two different grids based on an area-weighted average of overlapping gridpoints.
Parameter values MOMA to IGCM3: at31topa 91 2 48
IGCM3 to MOMA: at31topa 91 1 48

BLASNEW: changes SST to Kelvin for IGCM3.
Parameter values: 1. 1 CONSTANT 273.15

REVERSE: changes MOMA/OASIS field ordering to IGCM3.
Parameter values: NORSUD WSTEST (2nd parameter is the orientation of the target grid)

INVERT: flips IGCM3 field ordering to OASIS/MOMA order.
Parameter values: NORSUD WSTEST (2nd parameter is the orientation of the source grid)

CHECKIN, CHECKOUT: perform consistency checks.
Parameter values: none required.

There follows a summary of the operations performed on each pair of fields in the order OASIS performs them:

1 CHECKIN MASK EXTRAP MOZAIC BLASNEW REVERSE CHECKOUT
2 CHECKIN MOZAIC CHECKOUT (not used)
3 INVERT CHECKIN MASK EXTRAP MOZAIC CHECKOUT
4 INVERT CHECKIN MASK EXTRAP MOZAIC CHECKOUT
5 INVERT CHECKIN MASK EXTRAP MOZAIC CHECKOUT
6 INVERT CHECKIN MASK EXTRAP MOZAIC CHECKOUT
7 INVERT CHECKIN MOZAIC CHECKOUT
8 INVERT CHECKIN MOZAIC CHECKOUT
9 INVERT CHECKIN MOZAIC CHECKOUT
10 INVERT CHECKIN MOZAIC CHECKOUT

3.2.7 Overview of modifications to ocean and atmosphere models

In general terms, both MOMA and IGCM3 models have a main loop which is executed every timestep. At the end of each timestep all variables have been updated so this is a good time to transfer data between models. When data is sent, it is labelled with the timestep of the originating model. Clearly it is simpler if the models have identical timesteps but this is not essential as long as each model is able to compute the timestep of the other model from which it requires data. A schematic will illustrate this. Let us assume that the timestep in both models is 1 hour and the run is for 2 days. Fields are exchanged every model day therefore OASIS will perform two cycles.

Atmosphere	OASIS	Ocean	
	OASIS cycle 1 begins		
Na=1		No=1	timestep completed
Na=2		No=2	timestep completed
Na=3		No=3	timestep completed
.		.	
.		.	
.		.	
Na=22		No=22	timestep completed
Na=23		No=23	timestep completed
Na=24		No=24	timestep completed
Atmosphere sends heatflux labelled '24'			
		Ocean sends sst labelled '24'	
	OASIS receives heatflux and sst labelled '24'		
	OASIS cycle 2 begins		
	OASIS sends heatflux and sst labelled '24'		
Na=25		No=25	timestep completed
Atmosphere receives sst labelled '24' Updates sst			
		Ocean receives heatflux labelled '24' Updates heatflux	
Na=26		No=26	timestep completed
Na=27		No=27	timestep completed
Na=28		No=28	timestep completed
.		.	
.		.	
.		.	
Na=46		No=46	timestep completed
Na=47		No=47	timestep completed
Na=48		No=48	timestep completed

Thus the following 4 steps can be identified.

1. Identify heatfluxes, windstresses and freshwater fluxes in IGCM3 and at which point in the program these are updated and available to be sent to OASIS for a given timestep (see above)
2. Copy each field into a 2-D array and send to OASIS using message passing.
3. Receive fields from OASIS
4. Convert to model units and incorporate into boundary conditions.

In general very few modifications need to be made to the models. Before the main loop, a set-up routine must be called to establish communications with OASIS. Just before the end of the main loop, data must be sent to OASIS and received from OASIS. The calls are to black-box routines already written and supplied with OASIS. In the case of the ocean model they are inicmo.F (starts communication), flx.F (receives 6 flux fields together) stpcmo.F (sends sst and sea ice) and tau.F (receives windstresses). For the atmosphere model they are incma.F, fromcpl.F, and intocpl.F. A few parameter files have to be included in the models. A summary of all the changes made to IGCM3, MOMA and OASIS is given in Appendix B.

3.2.8 Grids and interpolation

The interpolation of data from atmospheric to ocean grids and vice-versa is performed using the mozaic technique. This is an area-weighted interpolation designed to conserve heat and freshwater in the coupled system as a whole. For each target gridpoint, a list of gridpoints on the source grid whose areas have nonzero overlap with the target gridpoint is compiled. The areas of overlap are computed and the value of the variable on the target gridpoint is calculated as the overlap-area-weighted mean of the values on the source gridpoints with nonzero overlap. The process of calculating the weights is done offline using the subprogram mozaicweights.f, in directory auxfiles-neat/progs. Currently, the program only works provided the atmosphere gridboxes are larger than the ocean gridboxes: If an occasion arose where an atmospheric gridbox was wholly contained by an ocean gridbox the algorithm would fail. This eventuality is unlikely given the expected applications of the model, but the program should nevertheless be corrected as soon as possible. A detailed account of the procedure for calculating the weights is given in Appendix A.

3.2.9. Land-sea mismatches at coastal boundaries

The fields the models pass are global ones - that is to say that the field of temperatures that the ocean passes to the atmosphere includes land temperatures, although the ocean knows nothing about them and they are set to an arbitrary value of -2, and the fields that the atmosphere gives to the ocean include values for over land, despite the fact that these are of no use to the ocean model (see Figure 9). It was left to the separate models to decide which values of the by now re-mapped fields were valid 'sea' values, and to discard the others. This process did not seem to function satisfactorily however, with some coast points of the ocean model picking up land values of heat fluxes and over-heating to extremes. The solution became apparent in a coupler option that allows the 'land' points to be screened out on the original grid, before being remapped and passed. The screened out points are given values by extrapolating from the nearest sea points. The fields actually passed at the end of the model day are therefore rather odd, consisting of valid data over sea, and entirely spurious data over land, but that spurious data is better fitted to resolving the grid mismatches that cause problems near the coast. Since the atmosphere knows not to use the new extrapolated land data coming from the sea model, no harm is done (although this may cause some flux-conservation problems later on, it was felt that removing the extreme values caused by the grid mismatches was more important). In future, the 'tiling' technique, allowing partial atmospheric gridboxes, will be implemented. This will ensure flux conservation, whilst also permitting coastal sea gridpoints to receive realistic fluxes.

4. Step by step procedure to run model

4.1 Initial run

The following steps are required to get a run started:

a Change run length in seconds in oasis_version_2.3/toyclim/Cpl/Nam/namcouple.latest to the required number. E.g. for 8760 steps of 1 hour each, the required number is 32536000
b Change KRUN in the IGCM3 run script (e.g. run_igcm3-oasis) to the required number (e.g. 8760). Note this is the number of timesteps to run regardless of the number already run, as reported in the restart file.
e Change number of days to run in Moma/Src/ocean.in to the required number. Unlike IGCM3 this must take into account the number of days already run, as stated in the restart file, e.g. for 8760 steps of 1 hour each, days must be set to 365 greater than the total reached by the last run.
f Ensure that the correct input files are present in the right directories for both models and OASIS. It will be necessary to run programs in the auxfiles_neat directory to do this. The output must then be copied to the appropriate directories (see instructions in OASIS section 3.2.3 above).
g Change directory to oasis_version_2.3/toyclim/Scripts and run script.4deg.linux_atm
h Change directory to igcm3_linux and run_igcm3-oasis. Note that this directory might be on another machine.

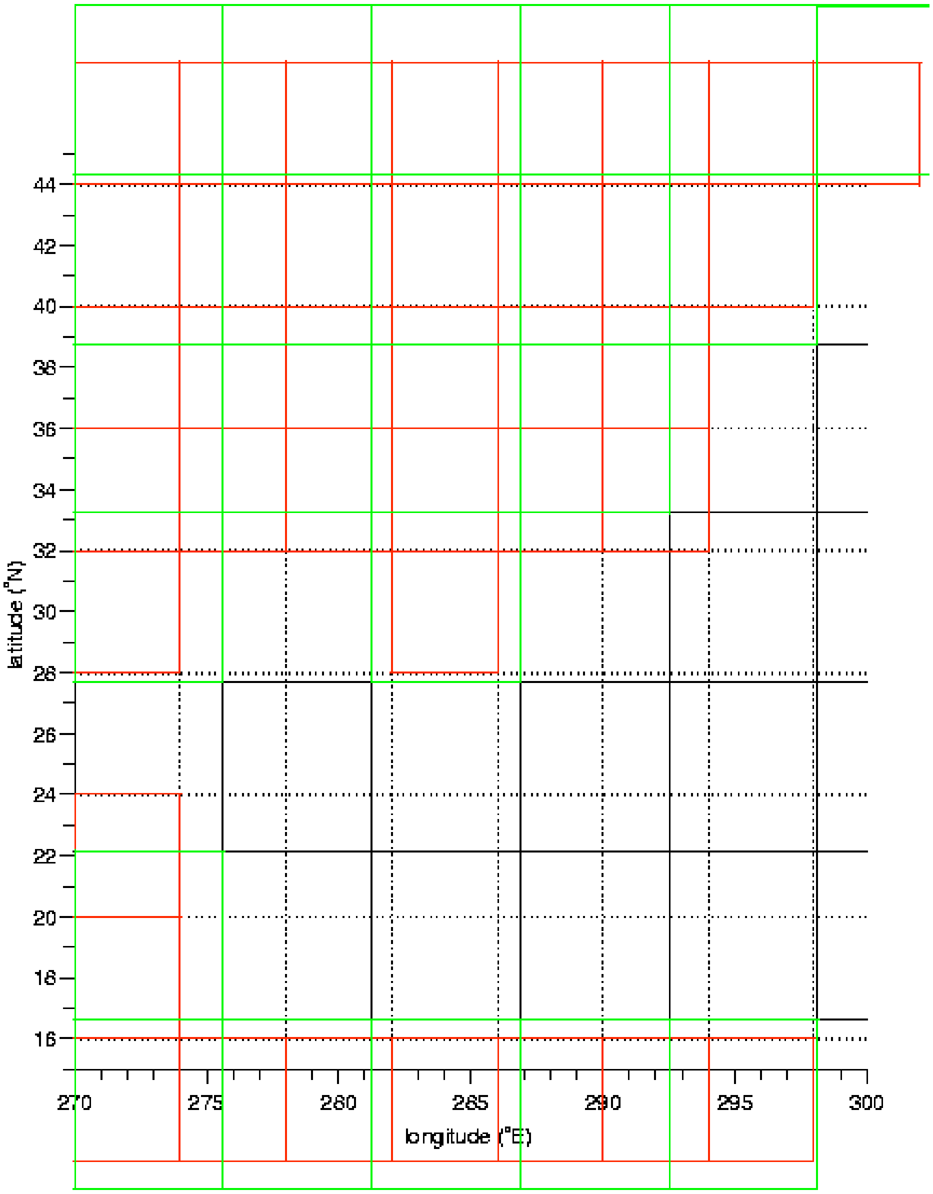
At the end of the run, all output files for OASIS and MOMA will be left in to oasis_version_2.3/toyclim/wkdir, and all output files for IGCM3 will be left in igcm3-linux/igcm3_test.

Optionally, the user may wish to spin up the ocean and atmosphere models separately in standalone mode before running them coupled together. This can be done by suitably modifying the update files in the respective models, but the details are beyond the scope of this report. Refer to the authors for details.

4.2 Restart run

Similar to the above, but this time, the final restart files generated by the previous run must be copied to the input files of the new run (taking care to save the original files first). The compilation options in the run scripts can be switched off leaving only the run options on. It may be necessary to remove all files and directories from oasis_version_2.3/toyclim/wkdir before running. NOTE: when running IGCM3 on a PC with linux, it is essential to include the compilation flag -byteswapio (line 150 of run_igcm3-oasis). This is because linux uses littlendian representation for binary files (the user should check this is true for his/her PC). As most other computers (e.g. Origin 2000) use bigendian representation, this would lead to problems unless the linux compilation flag -byteswapio is used to effectively allow the computer to use bigendian representation. If this is not done it will be necessary to byteswap files that are output in binary by IGCM3 and manually copied to be read in by OASIS (namely flxocean). When direct communication via PVM/MPI is in progress there are no big/littlendian problems, as the communication technique automatically deals with these issues. It is only when manual copying across architectures is performed that problems can arise.

Figure 9. Illustration of MOMA and IGCM3 grids and land masks over the Gulf of Mexico region. Solid black lines represent IGCM3 gridboxes over ocean, solid green lines represent IGCM3 gridboxes over land. Dotted black lines represent MOMA ocean gridboxes, solid red lines represent MOMA land gridboxes.



4.3 Restarting from a crash

Normally, IGCM3 uses output files fort.12 or fort.17 to restart from as these contain the final output fields at the end of the run (see section 3.1.1.5 and Table 3 above). However in the event of a crash (e.g. system failure) these files are not available and in order to restart from the last available restart data (which will be stored in fort.11 and fort.19) it is necessary to manually create new fort.12 and fort.17 files using the utility programs PULL.fort.11.f and PULL.fort.19.f present in igcm3_linux/data. These will create new versions of fort.12 and fort.17, which can then be used for a restart in the normal way. MOMA restart files each contain only one time level and so there is no problem following a crash.

4.4 Changing land configuration and/or MOMA resolution

The following steps are necessary to change land/orography/bottom topography configuration.

a Modify the IGCM3 orography/land mask file, generate the spectral version of the orography and generate the new MOMA land mask and bottom topography and oasis input files(see sections 3.1.1.3 and 3.2.3).
b Alter the vegetation file appropriately (section 3.1.1.6)
c Alter the runoff mask (section 3.1.1.7)
d Begin as for an initial run (section 4.1)

If only the MOMA resolution is to be changed, it is only necessary to generate the new MOMA land mask and bottom topography, and the new oasis input files. If the resolution is changed, MOMA may be started from a state of rest, to avoid the need for a restart file at the new resolution, by modifying the input file ocean.in (section 3.1.2.5 and Appendix D).

5. Results from a 100yr control experiment

The results presented come from an attempt to use the model under present-day conditions in order to provide a first indication of the speed and accuracy of the model. The model was run for 100years (taking about 8 days wallclock time) with the land mask as shown in Figures 2 and 5 and the bottom topography and orography as shown in Figures 3 and 6, intended to reproduce the main features of the continental configuration we see today. The atmosphere was spun up for 10 years using the mixed layer ocean (and flux adjustments) that come with it (using the run script run_igcm3-pgi). The ocean was spun up for 50 years by restoring to the surface fields of salinity and temperature interpolated from the Levitus (1998) climatological dataset and forcing with the Hellerman (1983) wind dataset. The coupled model was then run without any flux corrections. Figures 10 and 11 show the time series of the global mean surface temperature and salinity respectively. In the case of the temperature, after an initial adjustment period of order 5 years, the model shows a drift of order 0.8 degrees Celsius per century. The salinity takes longer to adjust initially, with a change of about 0.2 psu in the first 50 years or so, followed by a smaller change of order 0.1 psu over the next 50 years. The model will be run on for longer, but it seems likely that flux corrections will be necessary to produce a stable climate. This is not unexpected or uncommon with component models of such resolution. Despite the drift, the simulated surface temperature structure (Figure 12) is sensible and reproduces the Antarctic Circumpolar Current and subtropical gyre circulations well. The salinity structure (Figure 13) is less realistic: although the model displays high salinity bands in the subtropics, the real-world difference between the Pacific and more saline Atlantic in not reproduced and the Arctic Ocean is not fresh enough. The model meridional overturning circulation reflects this, with vigorous overturning in the pacific. The Atlantic meridional overturning (Figure 14) is at least realistic. Here, we do not comment further on the results shown. We hope to evaluate the model in detail in a forthcoming paper.

Figure 10. Time series of globally averaged sea surface temperature (degrees Celsius).

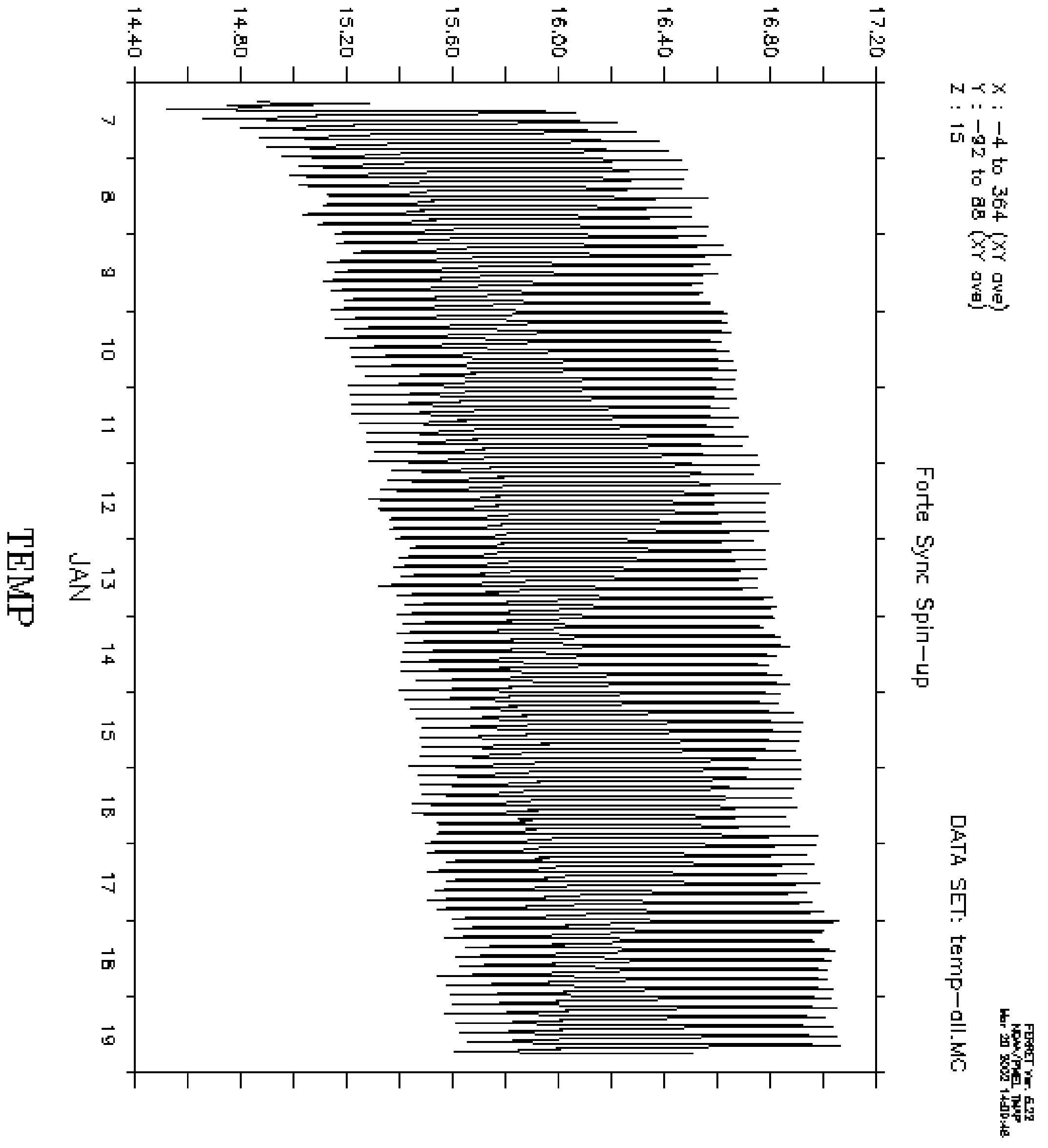


Figure 11. Time series of globally averaged sea-surface salinity (psu).

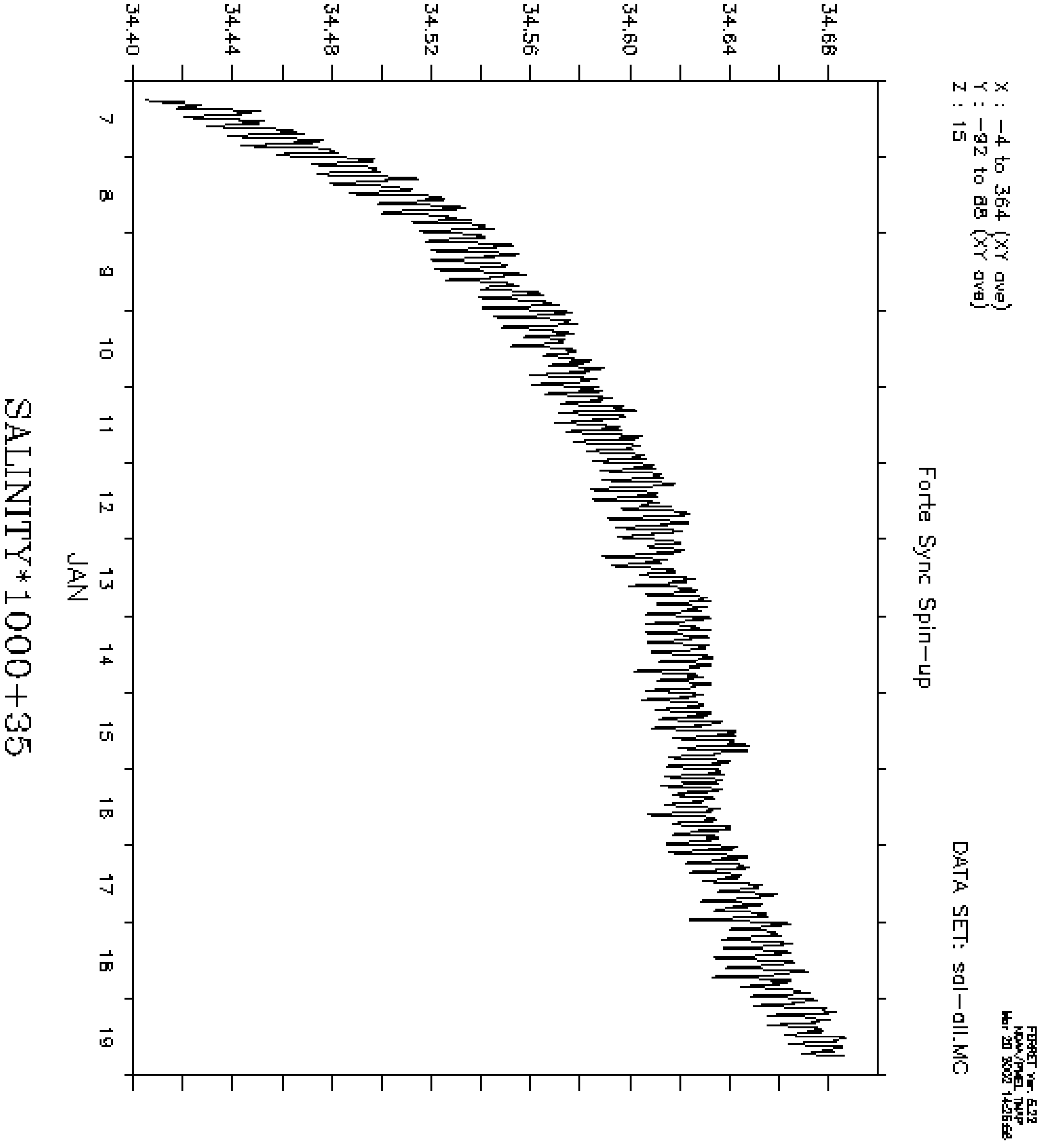


Figure 12. Sea surface temperature (degrees Celsius) averaged over final year of integration.

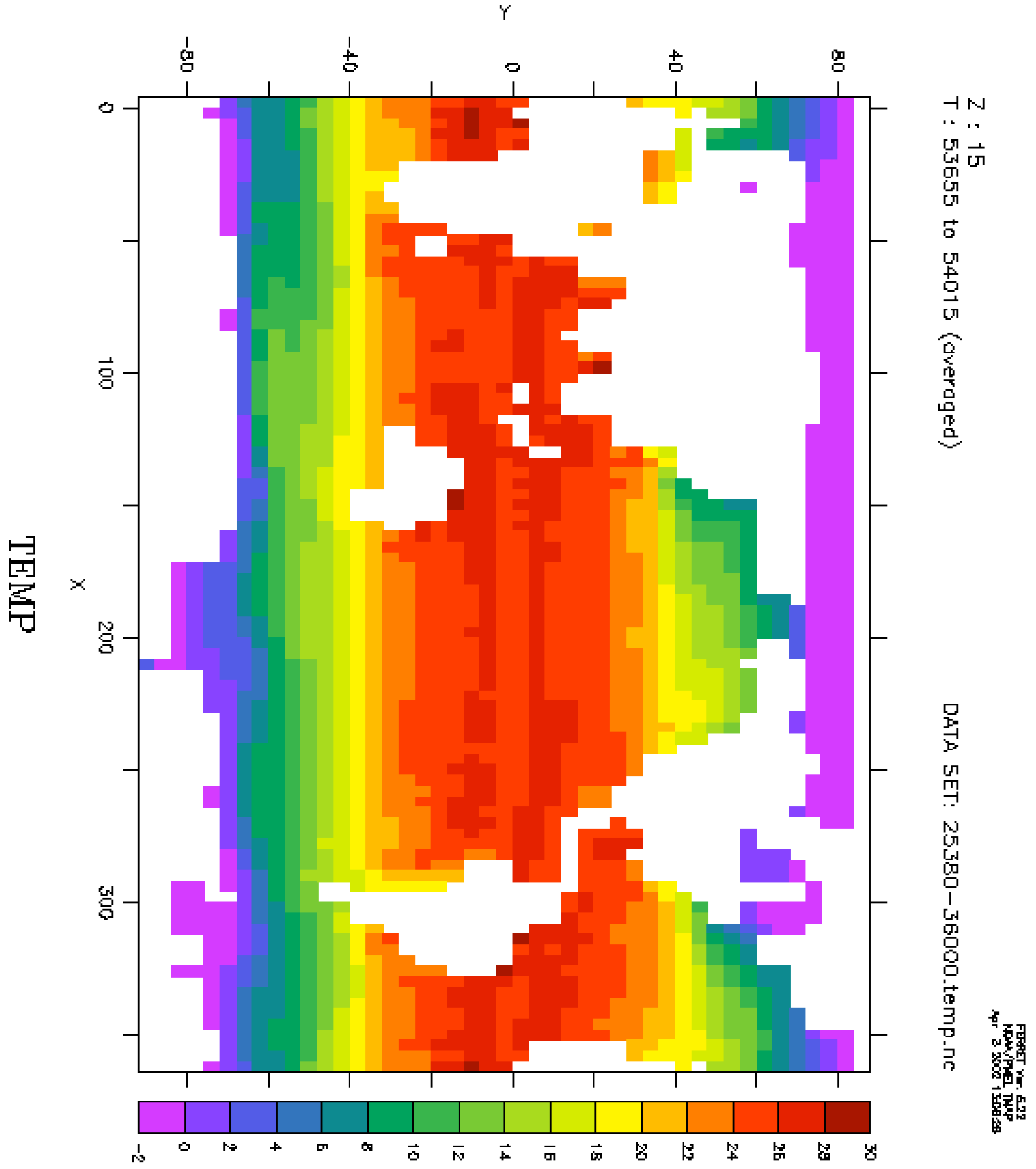


Figure 13. Sea surface salinity (psu) averaged over final year of integration.

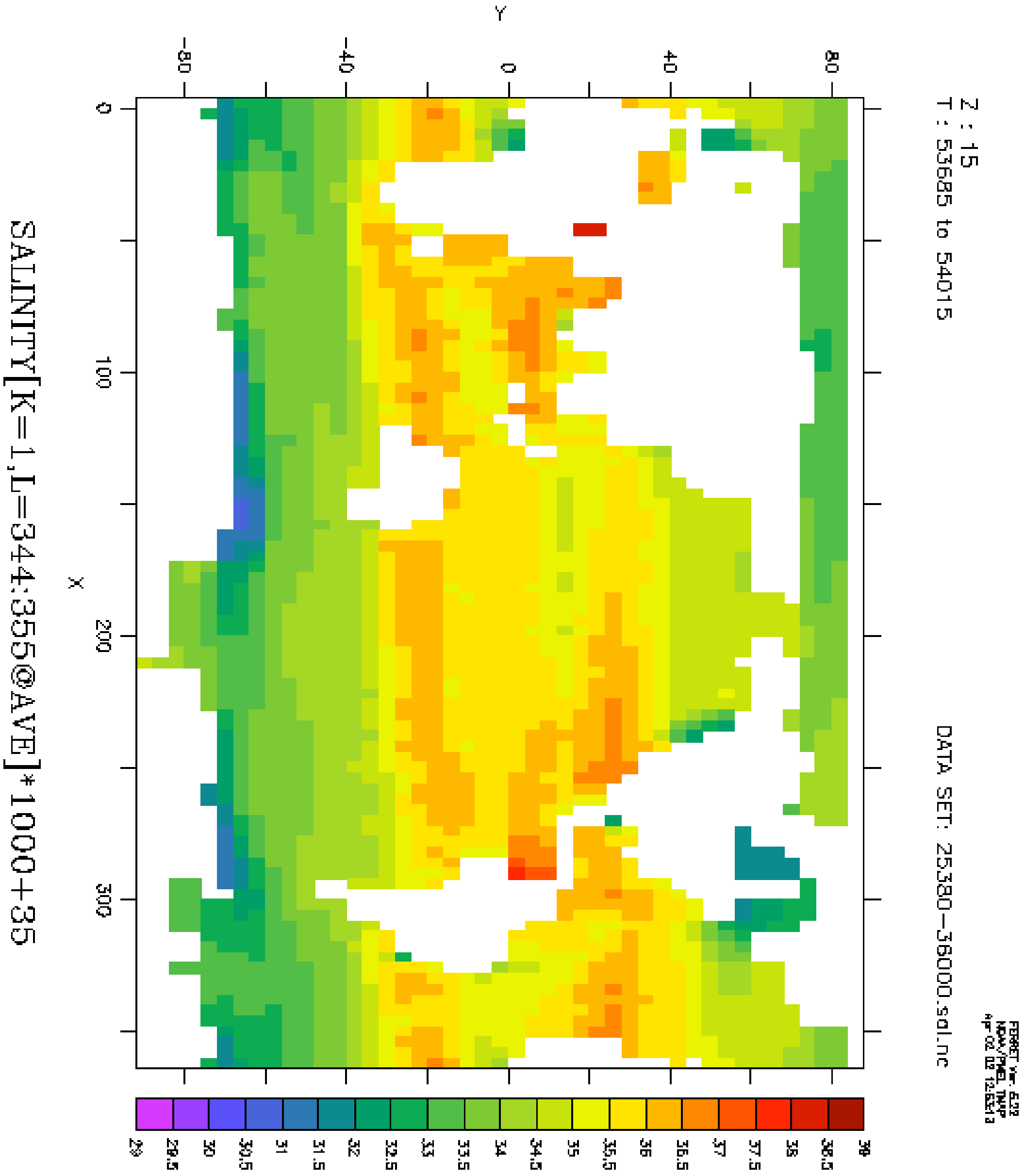
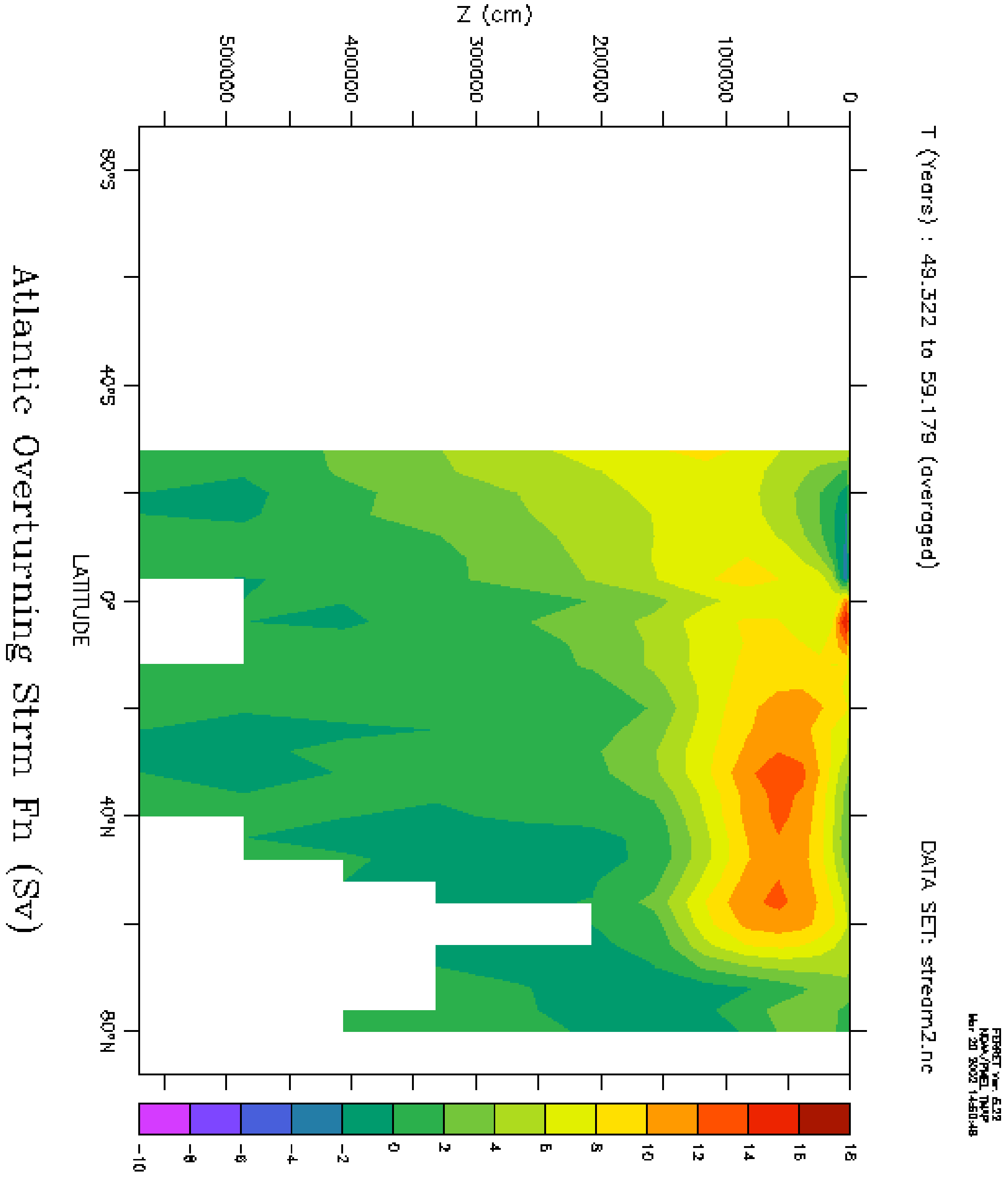


Figure 14. Atlantic meridional overturning circulation averaged over final year of integration.



Acknowledgements

B. Sinha carried out this work in his role as a core team scientist of the Coupled Ocean-Atmosphere Processes and European Climate (COAPEC) thematic programme of the Natural Environment Research Council (NERC). R. Smith was supported by a NERC Research Studentship. We thank Brian Hoskins, Keith Shine, Piers Forster and Manoj Joshi for help and advice regarding IGCM3. We thank David Webb and Andrew Coward for supplying MOMA and providing useful advice on coupling issues. We also acknowledge use of depth data supplied through COAPEC by the Hadley Centre for Climate Prediction and Research. We are grateful to John Stark for assisting with computing and visualisation.

References

Bourke, W (1972) An efficient, one-level, primitive equation spectral model. *Monthly Weather Review*, **100**(9), 683-689.

Forster, PM de F, Blackburn, M, Glover, R, Shine, KP (2000) An examination of climate sensitivity for idealised climate change experiments in an intermediate general circulation model. *Climate Dynamics*, **16**, 833-849.

Gordon, C, Cooper, C, Senior, CA, Banks, H, Gregory, JM, Johns, TC, Mitchell, JFB, Wood, RA (2000) The simulation of SST, sea ice extents and ocean heat transports in a version of the Hadley Centre coupled model without flux adjustments. *Climate Dynamics*, **16**, 147-168.

Hellerman S, Rosenstein M (1983) Normal Monthly Wind Stress Over The World Ocean With Error-Estimates. Journal Of Physical Oceanography , **13** (7): 1093-1104.

Hoskins, BJ, Simmons, AJ (1975) A multi-layer spectral model and the semi-implicit method. *Quart. J. R. Met. Soc.* **101**, 637-655.

Levitus S, Boyer, TP (1998) *World Ocean Atlas 1998, Volume 4, Temperature.* . NOAA Atlas, NESDIS 2, 99pp.

Levitus S, Burgett, R, Boyer, TP (1998) *World Ocean Atlas 1998, Volume 3, Salinity.* . NOAA Atlas, NESDIS 3, 99pp.

Packanowski, RC, Griffies, SM (1998) **MOM 3.0 Manual**, NOAA/Geophysical Fluid Dynamics Laboratory, Princeton, USA, 08542, 680pp.

Valcke, S., Terray, L, Piacentini, A (2000) OASIS 2.4 User's Guide. CERFACS, 85pp.

Webb, DJ (1993) An ocean model code for array processor computers. *Computers and Geosciences*, **22**(5), 569-578.

Webb, DJ, De Cuevas, BA, Coward, AC (1998) The first main run of the OCCAM global ocean model. *Southampton Oceanography Centre Internal Document No. 34, 44pp (Unpublished manuscript).*

Weaver AJ, Eby M, Augustus FF, Wiebe EC (1998) Simulated influence of carbon dioxide, orbital forcing and ice sheets on the climate of the Last Glacial Maximum**.** *Nature*, 394 (6696): 847-853.

APPENDIX A: Calculation of weights between atmospheric and ocean grids

Due to the fact that both atmosphere and ocean models are implemented on rectangular latitude-longitude grids, calculation of the weights is somewhat simplified. Figure A.1 shows part of the ocean grid at 4-degree resolution (red lines) and part of the T21 atmosphere grid (green lines). Clearly there are nine possibilities. The central red box is entirely contained within the central green box, hence only one atmospheric box contributes in this case. The four red boxes horizontally and vertically adjacent to the central one are entirely contained by two green boxes, and the remaining four red boxes (diagonally adjacent to the central one) are entirely contained by four green boxes. Taking as an example the lower left red box diagonally adjacent to the central one, we see that it is overlapped by four atmospheric gridpoints. Therefore the heat flux given to the ocean model by the atmosphere model should be a weighted average of the four overlapping gridpoints, with the weighting given by the areas of overlap. The area of overlap with the atmospheric gridbox on the lower left is highlighted in blue. The weight is given by $\text{Area}_{\text{Overlap}}$ /$\text{Area}_{\text{Oceanbox}}$ Using the well-known formula that the area on a sphere enclosed by two latitudes ($□_2$ and $□_1$) and two longitudes ($□_2$ and $□_1$) equals $2□R^2[\ \sin□_2\ -\ \sin□_1\ ][\ (□_2 - □_1]/360$. Where R is the radius of the earth and all latitudes and longitudes are in degrees, it is easy to show that the area of the ocean box is given by

$$\text{Area}_{\text{Oceanbox}} = 2□R^2[\ \sin(18.0) - \sin(14.0)\ ]\ (288.0 - 284.0)/360.0$$

and the area of overlap is given by

$$\text{Area}_{\text{Overlap}} = 2□R^2[\ \sin(16.620) - \sin(14.0)\ ]\ (286.750 - 284.0)/360.0$$

Similarly, in transferring the ocean model SST to the atmospheric model, the atmospheric gridbox on the lower left would be supplied a weighted average of the SST's of the four overlapping ocean gridboxes. In this case, the weighting of the lower left ocean gridbox already considered would be $\text{Area}_{\text{Overlap}}$ /$\text{Area}_{\text{Atmosbox}}$ where

$$\text{Area}_{\text{Atmosbox}} = 2□R^2[\ \sin(16.620) - \sin(11.080)\ ]\ (286.875 - 281.250)/360.0$$

Note that the atmospheric gridboxes can be overlapped by a four, six or nine oceanic gridboxes, whereas the oceanic gridboxes can be overlapped by one, two or four atmospheric gridboxes.

The main task of mozaicweights .f, the program that calculates and stores the weights is then to distinguish which of the cases above holds for each gridbox. The algorithm is as follows.

1. Target grid = ocean, which atmosphere points contribute and by how much? For a given ocean point, go through all the atmosphere points and ask the following questions:

Is the western longitude of the atmos gridbox to the west of the western longitude of the ocean gridbox AND is the eastern longitude of the atmos gridbox east of the eastern longitude of the ocean gridbox? If so the ocean gridbox is entirely contained zonally by the ATMOS gridbox. This is illustrated below, where A_W, A_W are the west, east bounding longitudes of the atmospheric gridbox, where O_W, O_W are the west, east bounding longitudes of the atmospheric gridbox and the x's represent the area of overlap :

| | | |

A_W O_W O_E A_E

xxxxxxxxxxxxx

Is the western longitude of the atmos gridbox to the west of the western longitude of the ocean gridbox BUT the eastern longitude of the atmos gridbox falls between the eastern and western longitudes of the ocean gridbox? If so the ocean gridbox is partially covered by the atmos gridbox which therefore contributes with a certain weight. The area of overlap is bounded to the west by the western longitude of the ocean gridbox, and to the east by the eastern longitude of the ATMOS gridbox:

| | | |

A_W O_W A_E O_E

xxxxxxx

Is the eastern longitude of the atmos gridbox to the east of the eastern longitude of the ocean gridbox BUT the western longitude of the atmos gridbox falls between the eastern and western longitudes of the ocean gridbox? If so the ocean gridbox is partially covered by the atmos gridbox which therefore contributes with a certain weight. The area of overlap is bounded to the east by the eastern longitude of the ocean gridbox, and to the west by the western longitude of the ATMOS gridbox:

| | | |

O_W A_W O_E A_E

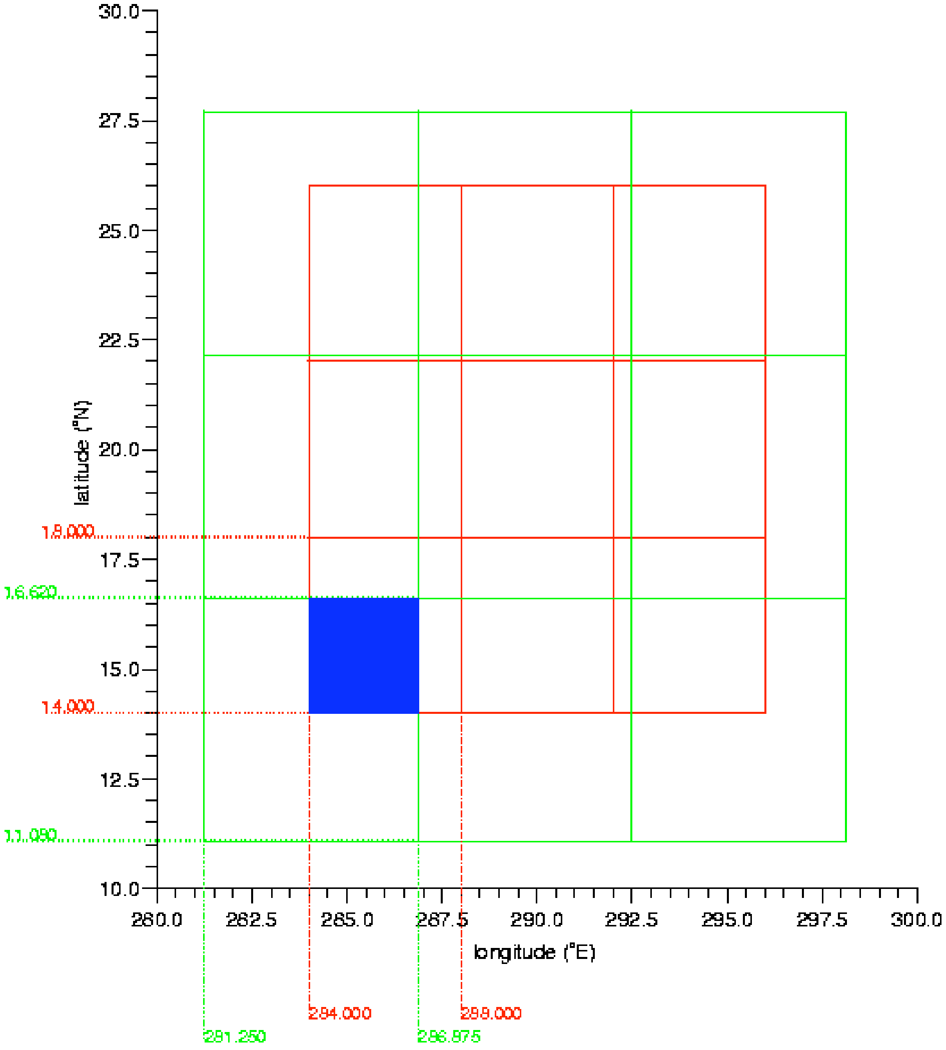
xxxxxxx

Similarly in the North-South directions we consider three possibilities:

```
|--  A_N            |-- O_N            |-- A_N
|                   |                  |
|                   |-- A_N   x        |
|--  O_N     x      |-- O_S   x        |
|            x      |                  |
|            x      |                  |
|--  O_S     x      |                  |
|                   |                  |
|                   |                  |
|                   |                  |-- O_N  x
|--  A_S            |                  |-- A_S  x
|                   |                  |
|                   |-- A_S            |-- O_S
```

2. Target grid = atmosphere, which ocean points contribute and by how much? For a given atmosphere point, go through all the ocean points and follow the same procedure as above, i.e. identify which of the nine configurations is realised. The only difference is that the weight is now the area of overlap divided by the area of the ATMOS gridpoint.

Figure A.1. Illustration of calculation of interpolation weights. Solid green lines are IGCM3 gridboxes, solid red lines are MOMA gridboxes, blue area is overlapping area between an ocean and atmospheric gridbox.



APPENDIX B: summary of changes to model codes and OASIS

a) Summary of changes made to original IGCM3 code

Function planck extended up past 350K by linear extrapolation from the end of the lookup table.

Every timestep: heat, wind, water arrays are accumulated into special arrays with an average factor for swapping (E and P are also accumulated into separate arrays for more accurate E-P calculation)

Before coupling, the rain array is rearranged to account for runoff - land rain moved to coast points using special file runoff.grid.

Subroutine inicmc sets up PVM/MPI.

Subroutine fromcpl imports surface temperature, subroutine intocpl exports in heat, water and wind fluxes.

NetCDF routines linked to existing history file output times.

Run with LOLDBL.TRUE - LOC.FALSE - LSL.FALSE

b) Summary of changes made to the original MOMA code

Subroutines tau and flx import IGCM3 fields from OASIS every nexco timesteps.

Subroutine stpcmo exports surface temperature and dummy field to OASIS every nexco timesteps.

Subroutine setvbc altered:

```
if t > -1.96
        stf(1) altered to: heat fluxes/specific heat (swlw + latheat + sensible)/specific heat
        stf(2) altered to: E-P (water)
else
        no forcing (this is the ice parameterisation)
endif
```

similarly, for winds :

smf(1) altered to: zonal wind stress (ztaux)
smf(2) altered to: meridional wind stress (ztauy)

Resolution set to 4 degrees, latitude offset to -92. (imt=92, jmt=45 in param.h). Baroclinic timestep set to 1350 seconds for both dtts, dtuv. Barotropic timestep (dtbt) set to 100 seconds.

hdf.F altered to allow 5-figure output to label .hdf files

c) Summary of changes made to the original OASIS code

It was necessary to replace PVMDATAINPLACE to PVMDATADEFAULT in line 151 of CLIM_Start.F (in directory oasis_version2.3/lib/clim/src - see OASIS user guide p69 on CLIM_Start).

APPENDIX C: IGCM3 run script for linux

```
ls
#! /bin/ksh
set -xve
##########################
#                        #
#  FULL PHYSICS IGCM3_1  #
# 2m MIXED LAYER OCEAN   #
# Piers Forster 3-3-99   #
# Sylvia Hare Aug 2000   #
# 1 year control         #
# integration, from 1st  #
#  Jan.                  #
#                        #
##########################
set +v
############################################# USER SWITCHES.
#
# set up $MACHINE option and
# set $RUNSTORE for directory where you want the source code to go
# this directory must already exist.
# the experiment goes into $EXPDIR

MACHINE=linux

NUPDATE=yes                   # Nupdate to a new fortran code, or reuse the
existing one
COMPILE= yes # Compile (yes/no)
RUN= yes # run an executable already created

EXPID=igcm3_oasis                   # Experiment identifer.
USERPATH=/storage/igcm3_linux
RUNSTORE=$USERPATH/igcm3_test
EXPDIR=$RUNSTORE/$EXPID       # Experiment directory.
OUTPUTDIR=$USERPATH/igcm3_test
```

```
NETCDF_ROOT=/usr/src/PGnetcdf
NETCDFDIR=$NETCDF_ROOT/lib

PVM_ROOT=/usr/share/pvm3
PVM_ARCH=LINUX
PVMLIB=$PVM_ROOT/lib/$PVM_ARCH

#############################################
if [ $MACHINE = LINUX ]
then
 KD=$USERPATH/kd/LINUX
 MODEL=$USERPATH
fi
 CLIMDIR=$MODEL/data/CLIMDATA/NEW
 nupdate=nupdate
 LIB=/usr/lib

OROGDIR=$MODEL/data/OROG
INIDATA=$MODEL/data

EXEC=IGCM-Linux                    # Full path of the executable
#                                # either to be created if COMPILE=yes
#                                # or to be run if COMPILE=no

############################################# UPDATE DIRECTIVES.
#
[ ! -d $EXPDIR ]    &&   mkdir -p $EXPDIR
[ ! -d $OUTPUTDIR ]    &&   mkdir -p $OUTPUTDIR
cd $RUNSTORE
[ ! -d $EXPDIR/climdata ]    &&   mkdir -p $EXPDIR/climdata
if [ $NUPDATE = yes ]
then
echo "*IDENT igcm-add" > updates
cat   $MODEL/updates/common.upd >> updates
cat   $MODEL/updates/stepreport.upd >> updates
cat  $MODEL/updates/levels22.upd >> updates
cat  $MODEL/updates/linux.upd >> updates
cat  $MODEL/updates/planckextra.upd >> updates
#cat  $MODEL/updates/temptest.upd >> updates
cat  $MODEL/updates/oasis.upd >> updates
cat  $MODEL/updates/oasis-backend.upd >> updates
cat  $MODEL/updates/av-accumulate.upd >> updates
cat  $MODEL/updates/netcdf-call.upd >> updates
cat  $MODEL/updates/netcdf-routine22.upd >> updates
cat  $MODEL/updates/alterwater.upd >> updates
#cat $MODEL/updates/timereset.upd >> updates
fi

#
```

```
############################################# NAMELIST DATA.
#
cat  <<  /EOF  >  $EXPDIR/data
 &INPPL
 &END
 &INPRN KRUN=1382400,BEGDAY=0.0,KITS=0,TSPD=64,PNU=0.02
 TDISS=0.250,NDEL=8,BEGDOY=0.0
 LFLUX=.TRUE.,
 LSTRETCH=.TRUE.,
 L22L=.TRUE.,LCLIM=.FALSE.,LPERPET=.FALSE.,LOROG=.TRUE.,
 LSHORT=.FALSE.,LTVEC=.TRUE., LBALAN=.TRUE.,
 LRSTRT=.TRUE.,
 LRESTIJ=.FALSE.,
 LNOISE=.FALSE.,
 LMASCOR=.TRUE.,LMASPRT=.FALSE.,LMASOLD=.TRUE.
 &END
 &INPOP RNTAPE=1.0,KOUNTP=1920,KOUNTH=1920,KOUNTR=1920,KOUNTE=1,NLAT=16,
 NTRACO=1,RNTAPO=1.0
 &END
 &INPHYS LBL=.TRUE.,LCR=.TRUE.,LLR=.TRUE.,LCUBM=.TRUE.,LCBADJ=.TRUE.,
      LRD=.TRUE.,LVD=.TRUE.,LSL=.TRUE.,LOC=.FALSE.,
      LNOICE=.FALSE.,LOLDBL=.FALSE.,LNNSK=.TRUE.,LCOND=.FALSE.,
      CBADJT=3.
 &END
 &INPRS
 &END
 &INPRSIJ
 &END
 &INPBL
 KBAL=5,LTBAL=.TRUE.
 &END
 &INQ LRH=.FALSE., LNSURF=.FALSE.
 &END
 &INRADLW LLBLM=.TRUE.,GAS(1)=1,GAS(2)=1,GAS(3)=1,GAS(4)=0,GAS(5)=0,
      GAS(6)=0,GAS(7)=0,GAS(8)=0,VMRCO2=358.0E-6,VMRCH4=1.72E-6,
      VMRN2O=312.0E-9,VMRHALO=7.97E-12
 &END
/EOF


#
############################################# ERROR PROCESSING FUNCTION.
#
ABORT ()
{
echo '!!!!!!!!!! ERROR PROCESSING !!!!!!!!!!'
set +e
\cp fort.2 $EXPDIR/results_fail
exit 1
}
#
############################################# COMPILE AND RUN PROGRAM.
#
if [ $NUPDATE = yes ]
```

```
then
[ -f update*log ] && \rm -f up*log
    $nupdate  -p $MODEL/igcm3-basic.pl  -c igcm3_oasis      \
              -i updates \
             -f -o sq       ||  ABORT NUPDATE
fi

if [ $COMPILE = yes ]
then
if [ $MACHINE = linux ]
then
    MACH=linux
    fflags='-byteswapio -r8 -O2 -Msave -Mdalign -Kieee -g'
fi

cd $USERPATH/Clim
    pgf77 $fflags -c  *.F
    gcc -c wallclk.c

cd $RUNSTORE
    pgf77 $fflags -o $EXPDIR/$EXEC igcm3_oasis.f ../Clim/*.o \
      -L $KD/igcm1/lib -l$MACH'pgfft' -l$MACH'pgblas' \
      -l$MACH'pgutil' -l$MACH'pgaux' -L $NETCDFDIR -lnetcdf \
      -L $PVMLIB -lfpvm3 -lgpvm3 -lpvm3  || ABORT f77
fi
# \rm update
#
#

if [ $RUN = yes ]
then

cd $EXPDIR
     [ -f fort.41 ] && \rm -f fort.41
     ln  -s $INIDATA/ocflux_25m_igcm3_1  fort.41
     [ -f fort.33 ] && \rm -f fort.33
     ln  -s $INIDATA/wv.199  fort.33
     [ -f fort.35 ] && \rm -f fort.35
     ln  -s $INIDATA/gastab_lblm  fort.35  # line by line model tables
     [ -f fort.36 ] && \rm -f fort.36
     ln  -s $INIDATA/gastab_nbm  fort.36   # narrow band model tables
     [ -f fort.37 ] && \rm -f fort.37
     ln  -s $INIDATA/plfunc  fort.37
     [ -r climdata ] && \rm -f climdata/*.dat
     ln -s $CLIMDIR/*.dat climdata
     [ -r orogdata ] && \rm -f orogdata
     ln -s $OROGDIR orogdata
     [ -f fort.7 ] && \rm -f fort.7
     ln  -s data  fort.7
     [ -f fort.10 ] && \rm -f fort.10
#     ln -s $INIDATA/restart.12 fort.10
     ln -s $INIDATA/restart.nodate.12 fort.10
     [ -f fort.18 ] && \rm -f fort.18
#     ln -s $INIDATA/restart.17 fort.18
```

```
     ln -s $INIDATA/restart.nodate.17 fort.18
     [ -f column.dat ] && \rm -f column.dat
     [ -f fort.31 ] && \rm -f fort.31
     ln -s $INIDATA/vegetation.oasis fort.31
    time ./$EXEC                          ||  ABORT EXECUTE
    set +e                                   # Disable exit on error.
    ls -alF    $EXPDIR                       # List files.
#    \mv fort.9  $OUTPUTDIR/history              # Save history
#    \mv fort.17 $OUTPUTDIR/surfhist
#    \mv fort.2 $OUTPUTDIR/fort.2
#    \mv fort.63 $OUTPUTDIR/fort.63
#    \rm $EXPDIR/*
    \rm $EXPDIR/climdata/*
    \rmdir $EXPDIR/climdata
    \rm $RUNSTORE/src/*
    \rmdir $RUNSTORE/src
    \rm $RUNSTORE/tmpsrc/*
    \rmdir $RUNSTORE/tmpsrc
fi
exit 0                                       # Successful termination.
#
############################################## END OF JOB.
```

APPENDIX D: MOMA control file - ocean.in

```
File 'ocean.in'.  Input file for MOMA model.
 ****    version for free surface MOMA code
 ****    2 day test run.

 &contrl
     init=.true.,         fnrest='d0002.hdf',
     days = 2.0,          restrt=.false.,
     nmix=  50,           eb=.true.,       ncon=1,         acor = 0.6,
     tsi=0.041666666,       snapd=1.0,       archd= 1.0,
     ftrest = 'hdf',      ftsnap = 'hdf', ftarch = 'hdf',
     dgnstc=10.0,         dchkbd = 0.1
 &end
 &eddy
      am     = 1.0e9,   ah     = 2.0e7,
      fkph   = 20.0,    fkpm   = 1.0,    cdbot = 0.001
 &end
 &tsteps
     dtts=3600.0,       dtuv=3600.0,     dtbt = 100.0
 &end
```

APPENDIX E: OASIS RUN SCRIPT

```
#!/bin/sh
##############################################################################
#
```

```
#                              script_comp_run
#
#############################################################################
#
#
# Script to submit to run the toy model using the CLIM communication
technique
#
### To run the toyclim model, the user has to adapt parts 1,2,3,4 and 5
### in the USER INTERFACE
#
#############################################################################
######################### USER INTERFACE ####################################
#############################################################################
###
### 1- Adapt the following QSUB directives:
###
###    On a CRAY, you should use:
###       #QSUB -r toyclim -me -eo -o ../OUTPUTS/toyclim_output
###       #QSUB -s /bin/ksh
###       #QSUB -lT 2000 -lt 1500 -lM 32mw
###    On a O2000, you should use:
###       #QSUB -r toyclim -me -eo -o ../OUTPUTS/toyclim_output
###       #QSUB -s /bin/ksh
###       #QSUB -lT 2000 -lt 1500 -lM 32mw
###    On a VPP, you should use:
#@$-r toyclim -me -eo -o ../OUTPUTS/toyclim_output
#@$-s /bin/ksh
#@$-lM 1500Mb -lV 0mb -lP 0 -lT 1800 -lt 1700
#
###
### 2- Adapt the following exported variables depending on the computer used,
###    if not defines for your machine below:
###
MACHINE=${HOST_NAME}
### On a O2000:
      DOPTNS="-mp -apolist -Dcyclic -Dde_checkbd -Dncsa_hdf -DREAL_8" ;
      FC="f77   -64 -O2 " ;
      INCLUDE=-I${HDF_ROOT}/include/IRIX64 ;
      HDFLIBS="${HDF_ROOT}/lib/IRIX64/libmfhdf.a
${HDF_ROOT}/lib/IRIX64/libdf.a ${HDF_ROOT}/lib/IRIX64/libz.a
${HDF_ROOT}/lib/IRIX64/libjpeg.a" ;
      F90=f90 ;
      OPTSF90="-64 -O -r8 -OPT:Olimit=2607" ;
      F77=${F90} ;
      OPTSF77=${OPTSF90} ;
###   LDR=${F90} ;
      OPTSLDR=${OPTSF90} ;
      CC="cc -64 ";
      AR=ar;
        ARFLAGS=rv;
      PVM_ROOT=/users/phd/rssmi/two/pvm3
      PVM_ARCH="SGI64";
      PVMLIB="-L${PVM_ROOT}/lib/${PVM_ARCH} -lfpvm3 -lgpvm3 -lpvm3" ;
      PVMINCLUDE=${PVM_ROOT}/include/fpvm3.h ;
      PVMSOPW= ;
      ASSCOM="echo ";
        CDF_ROOT=/users/phd/rssmi/two/netcdf
      NETCDFLIB="-L${CDF_ROOT}/lib/ -lnetcdf" ;
      PATH="$PATH:${PVM_ROOT}/lib/${PVM_ARCH}:${PVM_ROOT}/bin/${PVM_ARCH}" ;
      MACH=IEEE ;
      LDR="f77 -64 -mp" ;
      echo "$MACHINE OK" ;
#
###
### 3- In order to avoid "undefined external", define the dummy *.f
###    files to be used for OASIS and for the models (the dummy file
###    svipcdummy.c is automatically compiled by the Makefile_clim) .
###
###    On a CRAY, you should use:
###       export DUMOASIS='dummypipe.f dummysvipc.f' ;;
###
###    On a machine other than a CRAY, you should use:
###       export DUMOASIS='dummypipe.f dummysvipc.f dummysrc.f' ;;
#
### On a machine other than a CRAY
#DUMOASIS='dummypipe.f dummysvipc.f dummysrc.f' ;
DUMOASIS='dummypipe dummysvipc dummysrc' ;
#
###
### 4- Define the path of OASIS and of the toy coupled model
HOMEOA=/net/stommel/local/rssmi/oasis_version2.3
HOMETOY=$HOMEOA/toyclim
#
### 5- Define the path of the working directory
DIRWORK=$HOMETOY/wkdir
#
set -x
date
###############################################################################
###################### END OF USER INTERFACE ##################################
###############################################################################
#
# Computer name for PVM hostfile construction
LISTE=$MACHINE
# Get user id
UID=`id | sed 's/uid=//' | awk -F\( '{ print $1}'`
#
# Create working directory DIRWORK if necessary and go in there
#
[ -d $DIRWORK ] || mkdir $DIRWORK
cd $DIRWORK
#
# Remove all files but not directories
# \rm *
#
################# PREPARATION OF OASIS MAKEFILE (if 1st run) ###############
#
```

```
if [ ! -d Clim ]
then
    mkdir Cpl
    cd Cpl
    cp -r $HOMEOA/include .
    cp -r $HOMEOA/lib .
    cp -r $HOMEOA/src .
    cp -r $HOMETOY/Cpl/Make/Makefile_clim .
    #
    # Prepare makefile for coupler
    #
    cat <<EOF >Makefile
#
LIBOASIS     =     ${DIRWORK}/Cpl/liboasis2.2.a
LIBPVM       =     ${OPTSLDR} ${PVMLIB}
CMD   =     oasis
F77   =     ${F77}
F90   =     ${F90}
CC    =     ${CC}
OPTSF77      =     ${OPTSF77}
OPTSF90      =     ${OPTSF90}
AR    =     ${AR}
ARFLAGS      =     ${ARFLAGS}
PVM_ARCH=       ${PVM_ARCH}

SRCD = \\
EOF

#
# complete Makefile with dummies files sources
#
for DUMMY in $DUMOASIS
do
echo "\c" >>Makefile
echo "\$(DUMMYDIR)/${DUMMY}.f \c" >>Makefile
done
echo "" >>Makefile
cat <<EOF >>Makefile
OBJD = \\
EOF

#
# complete Makefile with dummies files object
#
for DUMMY in $DUMOASIS
do
DUMMYO=${DUMMY}.o
echo "\c" >>Makefile
echo "\$(DUMMYDIR)/${DUMMYO} \c" >>Makefile
done
echo "" >>Makefile

    cat Makefile_clim >>Makefile
    cd ..
```

```
fi
#
############################### COMPILATION ################################
#
# 1- Compile CLIM objects that will be used by both models
#
mkdir Clim
cd Clim
cp $HOMEOA/lib/clim/src/*.[Fc] .
cp $HOMEOA/lib/clim/include/*.h .
cp $HOMETOY/Oce/Inc/clim.h .
cp $PVMINCLUDE .
$FC $OPTSF77 -c *.F
$CC -D$PVM_ARCH  -c wallclk.c
cd ..
#
# 2- Compile the atmospheric model
#
mkdir Atm
cd Atm
cp $HOMETOY/Atm/Inc/*.h .
cp $HOMETOY/Atm/Src/*.[fF] .

$F77 $OPTSF77 $OPTSLDR -o toyatm *.[fF] ../Clim/*.o ${PVMLIB}
\rm *.o
#
ls toyatm && banner 'atm ok'
cd ..
cp Atm/toyatm .
 #
# 3- Compile the oceanic model
#
mkdir Moma
cd Moma
cp $HOMETOY/Moma/Inc/*.h .
cp $HOMETOY/Moma/Src/*.h .
cp $HOMETOY/Moma/Src/*.[fF] .
cp $HOMETOY/Moma/Src/ocean.in .
cp $HOMETOY/Moma/Src/ocean.kmt .

# nupdate to a new Moma.F
cp -r $HOMETOY/Moma/Src/updates .
cp -r $HOMETOY/Moma/Src/moma-basic.pl .
rm Moma.F

echo "*IDENT moma-add" > momadd.upd
cat updates/M-netcdf.upd >> momadd.upd
cat updates/M-oasis-inout.av.upd >> momadd.upd
cat updates/M-oasis-influence-ice.upd >> momadd.upd
#cat updates/M-oasis-accel.upd >> momadd.upd
```

```
/users/phd/rssmi/two/nupdate/nupdate -p moma-basic.pl -i momadd.upd -c
moma -a F -f -o sq
# nupdating done

${FC} ${DOPTNS} -r8 -c ${INCLUDE} *.F
${LDR} moma.o hdf.o inicmo.o flx.o tau.o stpcmo.o halte.o locread.o
locwrite.o ../../wkdir/Clim/*.o -o moma ${HDFLIBS} ${PVMLIB} ${NETCDFLIB}
\rm *.o
#
ls moma && banner 'moma ok'
cd ..
cp Moma/moma .
cp Moma/ocean.in .
cp Moma/ocean.kmt .
#
# 4- Compile the coupler OASIS:
#
pwd
cd Cpl
#
make
#exit
ls oasis && banner 'cpl ok'
cd ..
cp Cpl/oasis .
#
####################### INPUT and AUXILIARY DATA FILES
#######################
#
# Grid and analysis auxiliary data files
#
# grids: grids t u v f for both models
# masks: masks t u v f for both models
# maskr: reduced gird masks
# areas: mesh surfaces for both models
# at31topa: mozaic file with addresses and weights for interpolation from
#           ocean to atmosphere used for sst and icecover
# runoff31: mozaic file with adresses and weights for interpolation from
#           atmosphere to ocean used for runoff field
#
for nom in grids masks maskr areas at31topa runoff31
do
    cp $HOMETOY/Cpl/Data/${nom}.4t ${nom}
    # on Cray C90 or J90 "assign -F f77 -N ieee_dp f:$nom" has to be done
    eval ${ASSCOM} f:${nom}
done
#
# Restart input files: flxatmos sstocean
#
cp $HOMETOY/Cpl/Inputs/flxatmos.4t  flxatmos
eval ${ASSCOM} flxatmos
cp $HOMETOY/Cpl/Inputs/sstocean.4t  sstocean
eval ${ASSCOM} sstocean
```

```
ls sstocean && banner 'cpl restart ok'
#
# OASIS input file namcouple
#
cp $HOMETOY/Cpl/Nam/namcouple.moma.igcm  namcouple.tmp
# Adapt the type of machine for the present run
sed "s/CRAY/$MACH/" namcouple.tmp > namcouple
\rm namcouple.tmp
#
############################## PVM SET-UP ###################################
#
# hostfile generation for PVM automatic construction
#
lo=$LOGNAME
dx=$PWD/pvmd.$PVM_ARCH
ep=$PWD:$PATH
wd=$PWD
#
rm -f hostfile.$PVM_ARCH
#
LISTENEW=$(`echo $LISTE | sed "s/\(.*\).idris.fr/\1/"`)
#for node in $LISTENEW
for node in $LISTE
do
echo "#"node=$node >> hostfile.$PVM_ARCH
#echo $node lo=$lo dx=$dx ep=$ep wd=$wd ${PVMSOPW} >> hostfile.$PVM_ARCH
echo $node lo=$lo dx=$dx wd=$wd ${PVMSOPW} >> hostfile.$PVM_ARCH
echo 72-39 >> hostfile.$PVM_ARCH
done
cat hostfile.$PVM_ARCH
#
# pvmd.$PVM_ARCH generation
#
rm -f  pvmd.$PVM_ARCH
echo export PVM_ARCH=$PVM_ARCH >> pvmd.$PVM_ARCH
echo export PVM_ROOT=$PVM_ROOT >> pvmd.$PVM_ARCH
echo export PATH=$PATH >> pvmd.$PVM_ARCH
echo pvmd3 '$*' >> pvmd.$PVM_ARCH
chmod +x pvmd.$PVM_ARCH
#
# pvm.$PVM_ARCH generation
#
rm -f  pvm.$PVM_ARCH
echo export PVM_ARCH=$PVM_ARCH >> pvm.$PVM_ARCH
echo export PVM_ROOT=$PVM_ROOT >> pvm.$PVM_ARCH
echo export PATH=$PATH >> pvm.$PVM_ARCH
echo pvm >> pvm.$PVM_ARCH
chmod +x pvm.$PVM_ARCH
cat pvm.$PVM_ARCH
#
# cleaning
#
cat /tmp/pvm[ld].$UID
rm /tmp/pvm[ld].$UID
```

```
kill -9 `ps -ealf | grep $LOGNAME | awk '/pvmgs|pvmd3/{print $4}'`
#
###################### LAUNCHING THE TOY COUPLED MODEL
#######################
#
# PVM lauching
#
./pvmd.${PVM_ARCH} hostfile.${PVM_ARCH} &
sleep 5
#
# PVM information
#
cat /tmp/pvm[ld].$UID
ps -ealf | head -1
ps -ealf | grep $LOGNAME | grep pvm
#
# OASIS and model launching
#
#timex toyatm 2>atmos.timex &
timex oasis 2>oasis.timex &
timex moma 2>moma.timex &
banner "start IGCM"
```

APPENDIX F: OASIS CONTROL FILE: NAMCOUPLE

```
# This is a typical input file for OASIS 2.3.  Don't hesitate to ask
# precisions or make suggestions (OASIShelp@cerfacs.fr). This file can
# be used AS IT IS to run the CLIM toy model (toyclim). To run on a
# Cray the only thing to change is IEEE for IEEE under $MACHINE.
# The OASIS I/O routines decrypt the file without
# namelist or specific format. Any line beginning with # is ignored.
#
 $SEQMODE
# This has to do with the time strategy. If you have all the models
# running simultaneously, you must put 1.
# Otherwise, if you have n models running sequentially, you put n
#
              1
 $END
#########################################################################
 $MACHINE
# This describes the type of machine you run OASIS on.
# - if it is a cray, put IEEE; otherwise put IEEE
#
                IEEE
 $END
#########################################################################
 $CHANNEL
# This describes the type of message passing you want to use.
# - if you use named pipes + binary files (for synchro and data respectively)
#   you must write PIPE
# - if you use sockets for both synchro. and data (use of the Cerfacs library
#   CLIM based on PVM3.3), you must write CLIM
# - If you use system V shared memory segments and semaphores (for data and
#   synchro respectively), you must write SIPC
# - If you use OASIS as just an interpolator (i.e no models), you
#   must write NONE (furthermore you need to set NBMODEL to 0)
                CLIM
 $END
##########################################################################
 $NFIELDS
# This is the total number of fields being exchanged.
# 2 fields  Ocean -> Atm + 8 fields Atm. -> Ocean
# For the definition of the fields, see under $STRINGS keyword
#
             10
 $END
##########################################################################
 $JOBNAME
# This is an acronym for this given simulation
# (3 characters)
   CLI
 $END
###########################################################################
 $NBMODEL
# This gives you the number of models running in this experiment +
# their names (6 characters).
#
  2  toyatm   toyoce
 $END
############################################################################
 $RUNTIME
# This gives you the total simulated time for this run in seconds
#
#   3 days
#    259200
#    604800
#   30 days
#    2592000
     1296000
#   5 days
#    432000
#   10 days
#    864000
#    8640000
#   60 days
#    5184000
#   360 days
#    31104000
#    62208000
#    93312000
#    155520000
#   3600 days
#    311040000
#    622080000
#    777600000
#    1555200000
#   60 years
#    1866240000
 $END
###########################################################################
 $INIDATE
# This is the initial date of the run. It is regularly updated by the
# program. This is important if, for example, the SST field coming from
# a Pacific OGCM needs to be completed with climatological data
# of the right date.
#
  010101
 $END
###########################################################################
 $MODINFO
# Indicates if a header must be encapsulated within the field brick
# (YES or NOT)
  NOT
 $END
###########################################################################
 $NLOGPRT
# Index of printing level in output file cplout: 0 = no printing
#  1 = main routines and field names when treated, 2 = complete output
 2
 $END
###########################################################################
 $STRINGS
#
# The above variables are the general parameters for the experiment.
# Everything below has to do with the fields being exchanged
# For each field, the first 2 lines are descriptors of the field, the
# related grid, the related model and data files.
# The third line gives the list of analysis to be performed  and the
# following lines give specific parameters  for each analysis.
# See the documentation for the analyses available in OASIS and for the
# relevant lines.
#
############################################################################
#                      OCEAN  --->>>  ATMOS
#                      --------------------
# Field 1 : sea surface temperature
#
#   First line:
# 1) and 2) Symbolic names for the field before and after interpolation
#           (8 characters maximum)
# 3) Label number for internal OASIS output (cf blkdata.f)
# 4) Exchange frequency for the field in seconds (here 1 day)
# 5) Number of analysis to be performed
# 6) 7) 8) and 9) restart input binary file names + related unit numbers
# 10) Field status (EXPORTED or AUXILARY)
#
SOSSTSST SISUTESU 1 86400  7  sstocean   sstatmos  35   96  EXPORTED
#
#   Second line:
# 1) 2) 3) and 4) Number of long. and lat. on initial and final grids
# 5) and 6) locator prefix (4 characters) used to read the parameters
#           of the source and target grid
# 7) Index of the sequential position of the model generating the field
#    Meaningfull only if the SEQMODE input is > 1.
# 8) Flag used to delay the exchange of the given field in the case of models
#    running simultaneously (n = number of coupling timestep delay).
# 9) Flag to compute an extra timestep at the end (1 yes, 0 no)
# 10) Flag to compute the field integral in analyses CHECKIN and CHECKOUT
#     (1 yes, 0 no)
#
92  45    64  32    topa  at31     1     0      0      1
P  2  P  0
SERIAL
#
# List of analyses
#
#CHECKIN  MOZAIC   BLASNEW   CHECKOUT  REVERSE
 CHECKIN MASK EXTRAP MOZAIC  BLASNEW   REVERSE  CHECKOUT
#CHECKIN MOZAIC  BLASNEW   CHECKOUT
#
# Specific parameters for each analysis
#
# Mask operation
    999.999
#EXTRAP operation for $CMETHOD = NINENN
    NINENN      4       1     1
# Mozaic: 1) mapping filename 2) connected unit 3) dataset rank 4) Maximum
#         number of neighbors used
#
 at31topa      91      2     48
#
# Blasnew: go from Celsius to Kelvin
# 1) mult. coeff for initial field 2) nb of additional fields
# 3) names of additional field, 4) value of multiplicative coefficient
#
  1.   1
  CONSTANT     273.15
#
# Reverse: 1) and 2) describes the ordering of the target model
#
  NORSUD   WSTEST
#
# Glored: Since version 2.3, the information on the reduced grid
# has to be NOxx WHERE xx is half the number of latitude lines
#
#NO16 2 1 2
#
############################################################################
#
############################################################################
# Field 2 : Sea ice extent
#
SOICECOV SIICECOV 2 86400 3  sstocean   sstatmos  35   96  EXPORTED
92  45    64  32    topa  at31     1    0     0    1
P  2  P  0
```

```
SERIAL
#
CHECKIN  MOZAIC  CHECKOUT
#
 at31topa   91    2   48
#
#NORSUD   WSTEST
#
#NO16 2 1 2
#
############################################################################
#                      ATMOSPHERE  --->>>  OCEAN
#                      -------------------------
############################################################################
#
############################################################################
#
# Field 3 : Non solar heat flux
#
CONSFTOT SONSHLDO 6 86400  6   flxatmos  flxocean  95   31   EXPORTED
92  45    64  32    at31   topa    1     0     0     1
P  0  P  2
SERIAL
#
#REDGLO  INVERT CHECKIN   MASK EXTRAP   INTERP  CHECKOUT
#CHECKIN   MOZAIC  CHECKOUT
 INVERT CHECKIN MASK EXTRAP  MOZAIC CHECKOUT
#INVERT CHECKIN MOZAIC CHECKOUT
#
#NO16   SEALAND
#
#
 NORSUD   WSTEST
 999.999
 NINENN  4 1 2
 at31topa   91    1   48
#BILINEAR  G    SCALAR
#
############################################################################
#
############################################################################
#
# Field 4 : Solar heat flux
#
COSHFTOT  SOSHFLDO   7   86400  6   flxatmos  flxocean  95   31  EXPORTED
92  45        64  32    at31   topa  1  0   0   1
P  0  P  2
SERIAL
#
#INVERT   CHECKIN  MASK EXTRAP   INTERP   CHECKOUT
#CHECKIN   MOZAIC  CHECKOUT
 INVERT CHECKIN MASK EXTRAP  MOZAIC  CHECKOUT
#INVERT CHECKIN MOZAIC  CHECKOUT
#
```

```
#
#NO16    SEALAND
#
#
 NORSUD    WSTEST
 999.999
 NINENN  4 1 2
 at31topa   91    1   48
#BILINEAR  G  SCALAR
#
############################################################################
#
############################################################################
#
# Field 5 : water flux
#
COWATFLU SOWAFLDO 25 86400  6   flxatmos  flxocean  95   31  EXPORTED
92  45        64  32    at31    topa    1    0     0     1
P  0  P  2
SERIAL
#
#REDGLO  INVERT  CHECKIN  MASK EXTRAP  INTERP BLASNEW  CHECKOUT
#CHECKIN   MOZAIC  CHECKOUT
 INVERT CHECKIN MASK EXTRAP  MOZAIC CHECKOUT
#INVERT CHECKIN MOZAIC CHECKOUT
#
#
#NO16     SEALAND
 NORSUD    WSTEST
 999.999
 NINENN  4 1 2
 at31topa   91    1   48
#BILINEAR  G    SCALAR
#1.157407e-05  0
#
############################################################################
#
############################################################################
#
# Field 6 : Runoff
#
CORUNOFF  SORUNOFF   32  86400    6   flxatmos  flxocean  95   31  EXPORTED
92  45     64  32    at31   topa   1     0     0   1
P  0  P  2
SERIAL
#
#REDGLO  INVERT  CHECKIN  MOZAIC  BLASNEW CHECKOUT
#CHECKIN   MOZAIC  CHECKOUT
 INVERT CHECKIN MASK EXTRAP  MOZAIC CHECKOUT
#INVERT CHECKIN MOZAIC CHECKOUT
#
#
#NO16     LANDSEA
 NORSUD    WSTEST
 999.999
 NINENN  4 1 2
 at31topa   91    1   48
#runoff31   93   1    41
#1.157407e-05  0
#BILINEAR  G    SCALAR
#########################################################################
#
#########################################################################
#
# Field 7 : zonal wind stress -> ugrid
#
COZOTAUX  SOZOTAUX   23  86400  4   flxatmos  flxocean   95  31  EXPORTED
92  45     64  32    at31     uopa  1  0    0    1
P  0  P  2
SERIAL
#
#REDGLO  INVERT CHECKIN  MASK EXTRAP INTERP CHECKOUT
#CHECKIN   MOZAIC  CHECKOUT
#INVERT CHECKIN MASK EXTRAP  MOZAIC CHECKOUT
 INVERT CHECKIN MOZAIC CHECKOUT
#
#
#NO16     SEALAND
 NORSUD    WSTEST
#999.999
#NINENN  4 1 3
 at31topa   91    1   48
#BILINEAR  G   VECTOR
#
#########################################################################
#
# Field 8 : meridional wind stress -> vgrid
#
COMETAUY  SOMETAUY   24  86400  4   flxatmos  flxocean 95   31  EXPORTED
92  45     64  32    at31     vopa    1    0     0     1
P  0  P  2
SERIAL
#
#REDGLO  INVERT CHECKIN  MASK EXTRAP   INTERP  CHECKOUT
#CHECKIN   MOZAIC  CHECKOUT
#INVERT CHECKIN MASK EXTRAP  MOZAIC CHECKOUT
 INVERT CHECKIN MOZAIC CHECKOUT
#
#
#NO16     SEALAND
 NORSUD    WSTEST
#999.999
#NINENN  4 1 4
 at31topa   91    1   48
#BILINEAR  G   VECTOR
#
#########################################################################
#
```

```
# Field 9 : zonal wind stress -> vgrid
#
COZOTAUV  SOZOTAUV   23  86400  4   flxatmos  flxocean   95  31  EXPORTED
92   45    64  32    at31     vopa  1  0    0    1
P  0  P  2
SERIAL
#
#REDGLO  INVERT CHECKIN  MASK EXTRAP INTERP CHECKOUT
#CHECKIN   MOZAIC  CHECKOUT
#INVERT CHECKIN MASK EXTRAP  MOZAIC CHECKOUT
 INVERT CHECKIN MOZAIC CHECKOUT
#
#
#NO16     SEALAND
 NORSUD    WSTEST
#999.999
#NINENN  4 1 4
 at31topa   91    1   48
#BILINEAR  G    VECTOR
#
#########################################################################
#
# Field 10 : meridional wind stress -> ugrid
#
COMETAUU  SOMETAUU   24  86400  4   flxatmos  flxocean 95   31  EXPORTED
92   45    64  32    at31     uopa    1    0     0     1
P  0  P  2
SERIAL
#
#REDGLO  INVERT CHECKIN  MASK EXTRAP   INTERP  CHECKOUT
#CHECKIN   MOZAIC  CHECKOUT
#INVERT CHECKIN MASK EXTRAP  MOZAIC CHECKOUT
 INVERT CHECKIN MOZAIC CHECKOUT
#
#
#NO16     SEALAND
 NORSUD    WSTEST
#999.999
#NINENN  4 1 3
 at31topa   91    1   48
#BILINEAR  G  VECTOR
#
##########################################################################
 $END
```