Technical report

SPEEDO

Model description and validation of a flexible coupled model for climate studies

Wilco Hazeleger, Camiel Severijns, Rein Haarsma, Frank Selten, Andreas Sterl

Royal Netherlands Meteorological Institute (KNMI)
P.O. Box 201
3730 AE De Bilt, The Netherlands
email: hazelege@knmi.nl

October 8, 2003

Contents

1 Introduction

In July 2002 an interdivisional project started at KNMI called: "Changes in Patterns of Climate Variability" (PATCH), with subtitle: "Tropics - Extratropics Interaction". The purpose of PATCH is to bring together the expertise of different departments at KNMI to work on the subject of changes of climate patterns. The earth's climate has preferred large scale patterns of climate variability such as the North Atlantic Oscillation (NAO), El Niño Southern Oscillation (ENSO) and Tropical Atlantic Variability (TAV). These patterns vary on different time scales and have a profound impact on regional climate all over the world. For instance, climate in western Europe is strongly influenced by the phase of the NAO. Low frequency changes in these climate patterns have been observed over the last century (e.g. Kushnir, 1994, Hurrell 1995, Zhang et al. 1998, Mehta and Delworth 1995). Different mechanisms have been proposed that force the changes in these climate patterns, ranging from internal atmospheric variability, coupled ocean-atmosphere interactions, remote forcing to external forcing by volcanoes or the sun.

As the large scale climate patterns have a large impact on regional climate, it is important to know and understand how they might change in the future. Much is known about future projections of the global mean climate. It is less clear how these climate patterns will change in spatial scale, whether their frequency will change and what the predictability of these patterns are in a changing climate?

One of the mechanisms that affects patterns of variability in the midlatitudes is tropics-extratropics interaction. Extratropical climate patterns are influenced by tropical climate by so called teleconnections. The physical mechanisms of these teleconnections are, for instance, Rossby wave propagation in the atmosphere from a heat source in the tropics or advection of temperature anomalies in the ocean. At KNMI, we intend to study mechanisms of tropics-extratropics interactions and we will study how these teleconnections change in a changing climate. We will focus on decadal variations in ENSO, decadal variability in TAV and how these tropical low frequency variations affect climate in the Northern Hemisphere, and especially Europe. Changes in teleconnections caused by rising CO_2 concentration in the atmosphere will especially be considered.

For these studies a coupled climate model is required that simulates the tropical and extra-tropical climate credibly. To study mechanisms of variability many model runs are necessary. The approach of constructing ensembles of model runs requires a fast climate model. Also, to study sensitivity to parameter choices, the model must be fast. To study cause and effect relations, different physical processes must be implemented or excluded from the model. This requires flexibility of the coupled model.

In this technical report a flexible coupled climate model is described that is developed for this project (SPEEDy-atmosphere coupled to a range of Ocean models: SPEEDO). First, the different components of the model are described. These components are an atmospheric general circulation model, a land model and different ocean models. The atmospheric general circulation model and the ocean general circulation model have been developed elsewhere and will only shortly be discussed. Then technical aspects of the model framework will be discussed including a short description of how to set up and run the model. Results are provided of a hindcast run with the atmosphere model coupled to the land bucket model and prescribed observed SST variability. These results are compared to different datasets. Finally, surface heat fluxes, radiation and cloud parameters of this model are compared with other models in the Eurocs project (see Siebesma et al., manuscript submitted to QJRMS)

2 Coupled Model

The coupled model consists of an atmosphere, a land and an ocean component. The atmosphere model solves the primitive equation and has a simplified parameterization package. The land and ocean components have different subcomponents. The model permits the use of a prescribed land surface or interactive land model. Similarly, for the ocean, a prescribed ocean, a slab ocean model, a linear shallow water model or a primitive equation model can be used. All models communicate with each other through a coupler which is implemented as a library. The modular setup of the model allows for configuring a model that consists of different subcomponents easily. In the following we describe the different model components.

2.1 Atmosphere model: Speedy

The atmospheric general circulation model that is used is of intermediate complexity. It has a spectral primitive equation dynamical core and a set of simplified parameterization schemes. The model has been developed by at ICTP by F. Molteni and is nicknamed SPEEDY (Simplified Parameterizations primitivE Equation DYnamics). The model has 7 layers in the vertical and has a spectral truncation at total wave number 30. The model contains parameterization schemes for large scale condensation, convection, clouds, radiative fluxes, surface fluxes and vertical diffusion. A five layer version of the model is described in detail by Molteni (2003). We refer to this article for details on the atmospheric model. In the following we describe changes and additions to the model made at KNMI.

2.1.1 KNMI Changes

Firstly, the cloud cover parameterization has been adjusted to improve the simulation of stratiform clouds over the subtropical oceans. Based on the formulation in ECHAM4 (Roeckner et al., 1996), the expression for the fractional horizonal area of grid box covered with clouds is (see also Beersma et al, 2002):

$$c = \frac{r/f - c}{1 - r_0} \tag{1}$$

Here, r is the relative humidity, r_0 is a threshold relative humidity. f is 1.0 over land and sea ice. Over sea it is function of the pressure vertical velocity at 700 hPa, such that f is smaller at descending motions ($\omega < 0.0 \mathrm{Pa\ s^{-1}}$) and when the temperature difference between the surface and 850 hPa is smaller than 11.5 K . The inclusion of the parameter f improved the incoming short wave radiation at the ocean's surface by increasing the cloudiness over the subtropical subsidence areas. Also, based on the ECHAM4 formulation, the critical relative humidity (RH_{cr}) at which clouds form depends on height now:

$$RH_{cr} = RH_{top} + (RH_s - RH_{top})(1 - e^{1-(p/p_s)^{-4}}) \tag{2}$$

Here, RH_s is is specified surface value of 0.3, RH_{top} is specified at 0.99. p and p_s are the pressure and surface pressure respectively.

Secondly, the parameterization of the temperature over sea at 2 m has been changed. The temperature is now a function of both the temperature at 925 hPa and the sea surface temperature. The functional relationship has been derived using data from the NCEP/Reanalysis data (Kalnay et al. 1996) and is a function of latitude. This improved the surface heat fluxes markedly.

Furthermore, some parameter changes have been applied compared to the original Speedy model (see Appendix 2 where the physical parameter list of Speedy is shown). Firstly, the minimum albedo has been set to zero as this would mimic the stratus clouds which are now included by the parameterization described above. Cloud albedo has been raised to 0.55 and it depends on latitude (to simulate the dependence of cloud albedo on the zenith angle of the sun). Different parameters in the radiation scheme have been adjusted to obtain a realistic radiation budget. Also, the bulk transfer coefficient for latent and sensible heat over sea has been raised to 1e-3.

To accomodate climate change experiments, the radiation scheme has been adjusted to incorporate the effect of changes in CO_2 concentration in the atmosphere on the long wave radiation fluxes. The radiative transfer scheme of KRCM (Dorland, 1999) was used to calculate the absorption coefficients in the CO_2 band of the Speedy radiation scheme that match the clearsky long wave radiation flux dependence of KRCM on CO_2 concentration.

The land bucket model that can be coupled to Speedy requires a distinction between snow and rain. Adjustments have been made to accomodate snowfall in the model. When the surface temperature is below the freezing temperature, the precipitation in the model falls as snow and the latent heat for sublimation is used instead of the latent heat of condensation for the associated release of energy.

Finally, adjustments have been made to accomodate coupling with the different model components.

2.2 Ocean models: Slab, G-model, MICOM

At its lower boundary the atmosphere is forced by surface heat and radiative fluxes. Over the ocean, the heat fluxes can be obtained by prescribing the sea surface temperature (SST). For the ocean and atmosphere to interact actively, a more elaborate ocean model must be coupled to the atmosphere model. A hierarchy of ocean models can be used. The models range from a simple, surface heat flux-driven slab ocean model to a full state-of-the-art ocean general circulation model. The different components will be shortly described in the following sections.

2.2.1 Slab ocean model

In the slab ocean model the ocean is represented by a passive mixed layer for which the temperature of the mixed layer T is determined by:

$$\frac{\partial T}{\partial t} = -\frac{Q}{h\rho_w c_p} + F_m \tag{3}$$

Here Q is the net surface heat flux leaving the ocean, h the mixed layer depth, ρ_w the density and c_p the specific heat capacity of sea water. F_m represents the induced heat transport by the ocean and all the other processes nelgected in this balance. To ensure that the climatology of the mixed layer stays close to the observed climatoloty, F_m is diagnosed from equation 3 in a run with Speedy in which T is prescribed:

$$F_m = \frac{\partial T_{clim}}{\partial t} + \frac{Q_{diag}}{h\rho_w c_p}, \tag{4}$$

T_{clim} is the daily mean observed climatological SST obtained from a data set (e.g. COADS, ERA40 or NCEP/Reanalysis), and Q_{diag} is the daily diagnosed net surface heat flux from the model. Q_{diag} represents the ocean heat transport which cannot be simulated in the slab model.

This mixed layer model can be extended by including number of physical processes. Wind induced variations in Ekman currents, turbulent entrainment and barotropic circulation can be incorporated. When these processes are included the mixed layer temperature equation reads:

$$\frac{\partial T}{\partial t} = -\frac{1}{h}(\mathbf{U} \cdot \nabla_h T + w\Delta T + \frac{Q}{\rho_w c_p} + W + F_r) \tag{5}$$

where $\mathbf{U}$ is the horizontal transport in the mixed layer and ∇_h the horizontal part of the gradient operator. w and ΔT are the vertical velocity and the temperature jump at the base of the mixed layer respectively. W is the temperature tendency in the mixed layer due to wind-induced mixing. F_r represents the omitted terms.

The advection terms considered here are Ekman transport U_e and barotropic transport U_b. The vertical integrated horizontal Ekman velocity is given by:

$$\mathbf{U_e} = (U_e, V_e) = \frac{1}{\rho_w(f^2 + r^2)}(f\tau_y + r\tau_x, -f\tau_x + r\tau_y), \tag{6}$$

where f is the Coriolis parameter, τ the wind stress, and r a linear friction term which avoids singularities at the equator. The Ekman pumping derived from this is:

$$w_e = \nabla_h \cdot \mathbf{U_e} = \frac{1}{\rho_w(f^2 + r^2)}(f(\nabla \times \tau)_z + r\nabla \cdot \tau). \tag{7}$$

The barotropic transport $\mathbf{U_b}$ is dererived from a Stommel equation for wind-driven barotropic flow with a linear friction:

$$\kappa_s \nabla^2 \psi + \beta \frac{\partial \psi}{\partial y} = \frac{k \cdot \nabla \times \tau}{\rho_w} \tag{8}$$

Here ψ is the barotropic streamfunction, β is the meridional gradient of the Coriolis parameter, κ_s is a friction coefficient and k is the vertical unit vector. The horizontal velocity derived from the streamfunction is multiplied by the mixed layer thickness in the model.

Finally, the wind-induced mixing is based on the Niiler and Kraus (1977) model for the mixed layer. It can be estimated from the fruction velocity $u_* = \sqrt{\tau/\rho_w}$.

$$W = -\frac{\alpha}{h} u_*^3. \tag{9}$$

The different terms are only evaluated for anomalous wind stress, acting on mean temperature. This mixed layer model coupled to Speedy has been used by Haarsma et al. (2003, submitted to J. Clim).

2.2.2 G-model: linear shallow water model for the tropical Pacific

A step higher in the hierarchy, an ocean model can be used in which thermocline anomalies are determined using a linear 1.5 layer shallow water model of a baroclinic mode on a beta plane. The shallow water model solves for thermocline anomalies in response to wind stress anomalies. In addition, the model contains a sea surface temperature equation, which is an empirical linear equation that relates wind stress anomalies (τ) and thermocline anomalies to the SST anomalies, and it includes a damping term:

$$\frac{dT_a}{dt} = a(x)h_a(x, y) + b(x)\tau(x, y) - c(x)T(x, y) \tag{10}$$

Here T_a is the SST anomaly, h_a is the thermocline anomaly. The parameters a, b and c determine the strength of the feedbacks. They are by default piecewise linear and quadratic functions of longitude tuned such that observed monthly SST anomalies along the equator can be reproduced when observed wind stress anomalies are used. The model domain spans from 30°S to 30°N and contains the Pacific basin with realistic eastern and western boundaries. The model is described in detail by Burgers et al. (2002). For more details on the linear model we refer to this paper. This model is suitable for studying tropical Pacific climate problems.

2.2.3 MICOM: primitive equation ocean model using isopycnal coordinates

Finally, a primitive equation model of the ocean with an extensive parameterization package can be used. The model is the Miami Isopycnic Coordinate Model (MICOM). The model is described by Bleck et al. (1992). Only a brief outline of the model follows here.

The model solves a momentum equation, a layer thickness equation, and tracer equations for temperature and salinity using an isopycnal vertical coordinate and a horizontal C-grid based on the Mercator projection. The model equations and the numeric implementation are described by Bleck et al. (1992). The isopycnal coordinate vertical coordinate implies that each layer has a homogeneous potential density. Only the upper layer has variable density on which the surface forcing acts. This implies that the layer thickness of isopycnal layers with a lower density than the surface layer vanishes and the layers can outcrop against the upper layer. The isopycnal model has the advantage that horizontal mixing occurs allong isopycnals, as observed in the ocean. This avoids spurious diapycnal mixing which occurs in level models unless a proper parameterization scheme is used.

The model is forced at the surface by wind stress, turbulent heat fluxes, radiative fluxes and a freshwater flux. The model contains a bulk mixed layer parameterization for the upper layer based on the Niiler and Kraus (1977) scheme. Entrainment and mixed layer deepening can be driven by wind-induced mixing and/or by turbulent heat loss at the surface. A shallower mixed layer forms when the stabilizing effect of the turbulent heat gain is stronger than the effect of wind-stress on the turbulent kinetic energy budget. There is a small diapycnic diffusion below the mixed layer. Horizontal tracer diffusion, interface diffusion and momentum diffusion are based on Laplacian diffusion schemes.

The version of the MICOM model used at KNMI can be set up for any resolution and basin configuration. Temperature, salinity and layer thickness distributions are initialized from Levitus and Boyer (1994) and Levitus et al. (1994). At the lateral boundaries the thermodynamic properties of the model can be nudged towards observed values. Parameter files allow for changing model parameters easily.

2.3 Land model: Land Bucket Model

The land model that can be coupled to Speedy has been derived from the land model used in the ECBILT model (Opsteegh et al., 1998). A short description of the land model can be found in the technical description of the model (Haarsma et al., 1996). The model consists of grid cells with temperature, soil moisture, soil ice, snow cover, and run off as state variables. There is no horizontal or vertical transport in the model and the heat and moisture budgets are closed, i.e. energy is conserved by the land model. The land bucket model recieves surface fluxes and precipitation from the Speedy model. Because the surface fluxes themselves depend on the land temperature, the land temperature is determined iteratively.

When soil moisture is available it can evaporate. The rate of evaporation depends on the soil moisture content. The initial soil moisture distribution has a realistic geographical distribution (the input file for soil moisture is: `../LBM/Input/sm.nc`). When the land surface temperature drops below 0°C soil ice is formed when soil moisture is available. In that case the latent heat of sublimation is used for the latent heat flux. Vice versa, when the temperatures rises above the freezing point and soil ice is present, it melts. The heat capacity of dry soil is constant, but when the soil contains soil moisture the heat capacity depends on the soil moisture content.

The precipitation falls on the surface as snow when the surface temperature is below 0°C, otherwise as rain. When it falls as snow and the soil is covered with snow, the soil moisture evaporation is halted. Snow can accumulate to a prescribed, adjustable maximum snow depth (e.g. 10 m). Snow in excess of this value is converted to soil ice first. When there is too much soil ice present, the ice is melted and converted to runoff. In that case, the heat budgets are adjusted in accordance with the melting of snow. Furthermore, the albedo of the land is prescribed and spatially variable (input file: `../LBM/Input/albedos.nc`), but when the soil is covered with snow it gets the prescribed albedo of snow.

Rain and melted snow can accumulate in the soil to a maximum prescribed soil moisture content. Excess of soil moisture runs off to the sea.

3 Technical aspects of SPEEDO

In this section we describe some technical aspects of the coupled model. First, some general aspects are reviewed. We discuss the modular set up of different model components and the coupler that is used for communication between the model. To facilitate the user of the model, a short description is given of how to set up the model for a simulation.

3.1 Modular setup

Except for the atmosphere model, the different components of the model have been set up in a Generic Model Framework (GMF). Each model component contains the following flow control

1) initialization phase

2) time stepping loop

  2a) time step of the model itself (the model physics)

  2b) collection of the data and output to a history file according to a time table

  2c) storing of the model state in a restart file

  2d) preparation for the next time step.

Each model model component has a parameter file in which the type of run (e.g. new run, coupled run, restart etc), the calendar type, the start and end data, input/output files and a set of parameters are specified. The parameter file can be edited run time, i.e. the model does not need to be recompiled after changing the parameter file. All input and output data is in NETCDF format. More details on GMF can be found in Appendix 1.

3.2 Coupler

SPEEDO uses a distributed coupler implemented as a library that is linked to each of the component models of coupled system. The coupler supports atmosphere models and both global and regional ocean, land and sea ice models. A coupled model should consist of at least a global atmosphere, ocean and land model. Regional models can be overlayed on the global ocean and land models.

The coupler performs some checks at the start of a run. It ensures that all component models use the same coupling time step and run for the same number of steps. It also checks that each model writes restart files at the same interval. And finally, it determines whether regridding of data is necessary using information on the size and resolution of the grid used by each component model.

The grid mapping files needed by the coupler are computed using the SCRIP tool (developed at Los Alamos National Laboratory). A linear interpolation is used for fields that are non-negative such as precipitation. For all other fields a higher order conservative interpolation scheme is used.

3.3 Setting up and running SPEEDO

The standard distribution consists of 3 tarred and gzipped files. One file contains the MICOM model:

```
micom-2.7.tar.gz
```

This distribution contains the source code of MICOM 2.7, the user manual, several configuration files, ETOPO5 bathymetry data, and a climatology dataset consisting of Levitus (Levitus and Boyer 1994, and Levitus et al. 1994) and COADS data (Da Silva et al. 1994). One can also use a climatology computed from 41 years (1960-2000) of data from the NCEP reanalysis (Kalnay et al. 1996). The input dataset read by MICOM is computed from the climatology with a Ferret script. Note that this script requires Ferret version 5.0 or later. You can download a Ferret binary distribution from the Ferret home page.

The second file contains Speedy, the slab ocean model, the land bucket model, the fixed land model (climatological land temperatures and albedo), the fixed ocean model (climatological SST) and the Sea model (variable SST for hindcasts) as well as configuration files etc.:

```
Speedy7V32.tar.gz
```

In order to compile and run Speedy and MICOM one also needs the following libraries:

the Coupler library (only when coupling with other models)
the Fields library (version 1.1 or higher),
the NetCDF library from the NetCDF page of the UNIDATA web site
the udunits library also from the udunits page of the UNIDATA web site

Links to all these libraries and more information can be found at:
`http://www.knmi.nl/onderzk/CKO/micom-cko.html`.

Finally, the last file contains the code for the G-model:

```
gmodel-3.0.tar.gz
```

3.3.1 Setting up Speedy

To start setting up the model, make a new directory (e.g. SpeedyV32), copy the Speedy file to the directory and unzip and untar the file that contains the atmosphere model:

```
gunzip -c Speedy7V32.tar.gz | tar xvf -
```

The directory structure shown in Figure 1 will be set up.

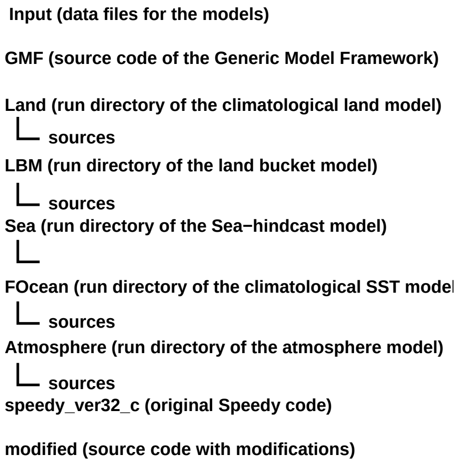


Figure 1: Directory structure of Speedy

Then the atmosphere model must be prepared. In the `Atmosphere/sources` directory the `cls_inphys.h` file contains the physical parameters of the model (see Appendix 2). A default set is specified. In order to run the model properly, `cls_instep.h` file must be edited (see Appendix 3 where the contents of this file is shown). Firstly, the number of months of the run and the start year must be specified. This must be compatible with the ocean model, to set up later. For instance:

```
NMONTHS=1200 !  a 100 year run, note that 360 day calendar is used
IYEAR=1960 !  start year is 1960
```

After editing `cls_instep.h`, the executable can be built using

```
make
```

This builds the exectuable

```
imp.exe
```

The output of the atmosphere model consists of GRADS files which are placed in the `Atmosphere` directory.

3.3.2 Setting up the land model: LBM and Land

Secondly, the land model must be set up. A choice must be made between using prescribed climatological land temperatures (in the `Land` directory) and Land Bucket Model (in the `LBM` directory). After choosing the model, compile the code either in the `Land/sources` or in the `LBM/sources` directory using the `Makefile`:

```
cd ./LBM/sources ;make  or cd ./Land/sources ;make
```

In the `Makefile` the directories of a set of liberaries (`Coupler, Field, udunits`) is default specified, but it might need editing. Also, one can choose between using the Portland compiler (f90) or the Intel compiler (ifc).

After the Land or LBM model has been compiled, the `parameters` file in the `Land` or `LBM` directory must be edited. An example of the parameter file is given in Appendix 2. The type of run, the names of input, output, restart files and model parameters can be edited in this file. Be aware that the number of months used for the run must be the same in all component models, otherwise the coupled model won't run. Note that the parameters that are set in the `parameter` file can be edited run time, that is, the model does not need to be recompiled if one of the parameters is changed.

3.3.3 Setting up the ocean model: FOcean, Sea, SlabOcean

Similary as with the land model, a choice must be made which ocean model to use. Here we describe setting up the simple ocean models. The Micom model is described in the section 3.2.5. There are three simple models:

- FOcean: prescribed, climatological sea surface temperature. This is not really a model, but treated like one in the modular setup of the coupled model.
- Sea: prescribed, variable sea surface temperature (e.g. as derived from NCEP/Reanalysis).
- SlabOcean: fixed depth, mixed layer ocean model.

So, first one must compile the model in the proper directory:

```
cd ./FOcean/sources ;make  or cd ./Sea/sources ;make or cd ./SlabOcean/sources
;make
```

In the `Makefile` the directories of a set of liberaries (`Coupler, Field, udunits`) are default specified, but it might need editing. Also, one can choose between using the Portland compiler (f90) or the Intel compiler (ifc)

Note that before running the slab ocean model, a climatology containing the net heat fluxes must be present (the Qflux). This data is created by first running the atmosphere model coupled to an ocean with climatological, prescribed sea surface temperatures (FOcean). This model creates a net surface heat flux climatology which is needed for the slab ocean model.

After one of the models has been compiled, the parameter file can be edited in the appropriate directory.

Finally, in Sea and FOcean, sea surface temperatures are prescribed. The user may want to change those sea surface temperatures. In the Sea model, `fort.20` contains the SST climatology and `fort.30` the SST anomalies. The SST needs to be on a fixed grid 96 longitudinal and 48 latitudinal points (see the source code at `./Sea/source/Sea.f90` for details). These are binary files. In FOcean, `fort.20` contains the SST climatology.

3.3.4 Setting up the G-model

If one chooses to use the G-model, the file that contains the source code and input files for the G-model must be untarred and unzipped:

```
gunzip -c gmodel-3.0.tar.gz | tar xvf -
```

This creates a directory `Gmodel`. The subdirectory `Model` contains the source code and the executable named `Shallow` can be made using `make` in this directory. A number of adjustable model parameters are set in `starts.f`. These parameters include the square of the velocity of Kelvin waves, the depth of the mean thermocline the biharmonic diffusion and viscosity amongst others.

Originally, the model was created to run in a forced mode or coupled to a statistical atmosphere. Here we describe shortly how to prepare the model. For more information setups we refer to the `readme.txt` file. The file `control.in` controls the type of run this made. The file has the following format:

a one line comment
a one line experiment identifier

0/1	1 means restart run
0/1/2	0 forced run, 1 coupled with statistical atmosphere, 2 coupled with Speedy
n	number of days
yyyymm	The starting year and month

By specifying the coupling to Speedy the model can be run in coupled mode. The model creates several output files of which the most important ones are: `slvano.ext` and `sstano.ext`. These files contain the monthly thermocline depth anomalies and monthly sea surface temperature anomalies respectively. These files are in extra format, that is, unformatted FORTRAN with the following structure:

Record	Contents
1	`yyyymmdd field level size`
2	`(field(i),i=1,size)`

... ...

3.3.5 Setting up MICOM

The Micom file must by untarred and unzipped in a directory (e.g. Micom):

```
gunzip -c micom-2.7-cko.tar.gz | tar xvf -
```

After unzipping and untarring the Micom file in a new directory (e.g. MICOM) the directory structure shown in Figure 2 is set up.

bin

climatology (input files for forcing, initilizations

config (directory with configuration files)

doc (documentation on the Micom model)

include

source (source files of Micom)

Figure 2: Directory structure of Micom

Differently than with the previous models, Micom needs to be configured first. That is, the grid, gridsizes, layer densities, basin size etc has to be specified before the model is compiled. This has to be done in the `config` directory where a few examples are given.

After editing the config file (for instance for the North Atlantic as in `./config/NA`), the model can be configured in the `MICOM` directory with the `configure` script:

```
./configure -with-parameters=NA.
```

This command determines the location of the various commands (compiler, etc.) that are needed to build the MICOM executable. Note that this should be done on the machine on which the executable should run and only needs to be done once on every computer. In the case that the configuration script is not able to determine the location of the NetCDF, udunits, and/or Fields library you should add one or more of the following options to the configure command and run it again:

```
-with-coupler=location-of-coupler-directory
-with-netcdf=location-of-netcdf-include-and-lib-directories
-with-udunits=location-of-udunits-include-and-lib-directories
-with-fields=location-of-fields-include-and-lib-directories
```

Upon successful completion of the configuration script you can find a makefile and a new subdirectory `data.NA` in the directory MICOM. Here forcing files, relaxation files and a `parameters` file are placed. The input files are created by Ferret which regrids the data from climatologies to the Micom grid.

The configure script has several more options. Normally, you won't need to specify any of them. You can obtain a list of all options of the configure script with the command

```
./configure -help
```

After the Micom model has been configured, the model can be compiled using `make` in the `Micom` directory. A `micom-NA` exectubale is made which can be run from the `data.NA` directory.

As with previous models, the `parameters` file in the `data.NA` directory can be edited run time. Micom can also run stand-alone with climatological forcing or in a hindcast mode. Furthermore, for process studies different physical processes can be easily turned off or on in the parameter file. Note that Speedy uses a 360-day calendar, so Micom needs to use a 360-day calendar as well. More information on the Micom model and links can be found at `http://www.knmi.nl/onderzk/CKO/micom-cko.html`.

3.4 Running the coupled model

Before the coupled model can be run one has to decide first which model components to use. The Speedy atmosphere model can be coupled to a land model: LBM or Land and to an ocean model: Sea, FOcean, SlabOcean, G-model, or Micom. So possible combinations are for instance Speedy-LBM-SlabOcean, Speedy-Land-Micom, Speedy-LBM-FOcean etc.

The script called `run_coupled` creates a file `schema` that describes which model is used. To create the `schema` file and start the run one has to type:

```
./run_coupled pgf90 Atmosphere Land Sea
```

Or any other combination of land or ocean model. The `schema` file just contains the working directories of the models and the location of the executables. One can edit the `schema` file and add a line with the working directory of the micom model. An example of a `schema` file is:

```
n0 -wd /usr/z/SpeedyV32/Atmosphere /usr/z/SpeedyV32/Atmosphere/run
n0 -wd /usr/z/SpeedyV32/LBM /usr/z/SpeedyV32/LBM/run
n0 -wd /usr/z/SpeedyV32/SlabOcean /usr/z/SpeedyV32/SlabOcean/run
n0 -wd /usr/z/Micom/data.NA /usr/z/Micom/source/micom-NA
```

In this case the Speedy atmosphere model is coupled to the landbucket model, the slab ocean model and to Micom (Micom overrides the slab ocean such that when Micom has been configured for 1 basin, the slab ocean model will be used elsewhere). This schema file can be easily edited to make other combinations of models.

In order for Micom and Speedy to run properly, remapping files have to be created. Every time a new Micom grid is been configured, this should be done. The files are needed for a proper communication between both models which run on different grids. The files are created by running the `./Coupler/SCRIP/run` script. The bathymetry file of Micom and the directory in which the files need to be put are arguments:

```
./Coupler/SCRIP/run depth.nc /usr/z/Micom/data.NA
```

The script produces the `rmp_MICOM_to_Speedy7_conserv.nc` and `rmp_Speedy7_to_MICOM_conserv.nc` files. The remapping files need to be placed in the directory in which Micom runs.

Before the model can be started a lamserver must run on the platform. This can be done by typing:

```
lamboot
```

With `lamhalt` it can be stopped. Note that all model runs will stop when `lamhalt` is called!!

Finally, to run the coupled model, type the following command:

```
mpirun -c2c -0 schema
```

In all subdirectories history files (in NETCDF), restart files (in NETCDF) and LOG files will be created. The exception is the Speedy model which creates GRADS files. One must make sure that history files do not exist already.

4 General characteristics of a Speedy-LBM-Sea hindcast

In this section we show some results of a run with the Speedy model coupled to the Land Bucket Model. The SST is prescribed and varies in time according to the SST in the NCEP/Reanalysis data (Kalnay et al., 1996). The hindcast has been run from 1960 to 1991. The results are compared with NCEP/Reanalysis, ISCCP (for cloud cover, Bishop and Rossow, 1991) and CMAP (precipitation, Xie and Arkin, 1996). We focus on a fields that are important to simulate well in a climate model. As this model will be used for climate variability studies we include diagnostics on important modes of variability, such as the North Atlantic Oscillation, and the atmospheric response to El Niño.

The climatology of the model shows that SPEEDO is capable of reproducing many aspects of the general circulation credibly. The model errors are of the same order of magnitude as in state-of-the art models. Among the strong points of the model are a good representation of the midlatitude jets, good teleconnections from the tropics to the extratropics and realistic North Atlantic Oscillation variability. Weaknesses of the model include the representation of the stratosphere (which has only 1 layer), the rather weak surface wind response in the tropics to tropical SST anomalies, a poorly simulated Asian monsoon and too much precipitation over land.

In general this model is very suitable for climate studies. Its computational efficiency is attractive and its simulated general circulation is close to the observed circulation. In the following the modelled circulation will be compared to observations.

4.1 Radiation budget

We start with the global averaged radiation balance of the model and of estimates from observations. In Figure 3 we show the top of the atmosphere (TOA) and surface radiation for the Speedy-LBM-Sea model and compare that to the results published by Peixoto and Oort (1991, their figure 6.3). The Speedy-LBM-Sea model is not in equilibrium and has a global average excess of 8 W/m^2. This number goes down with a few W/m^2 if the run is continued. In general, the relative contributions of the different terms in the radiation budget are comparable to estimates from observations. The main errors are found in the outgoing long wave radiation. It doesn't balance the incoming short wave radiaton, which is also reflected in the surface budgets.

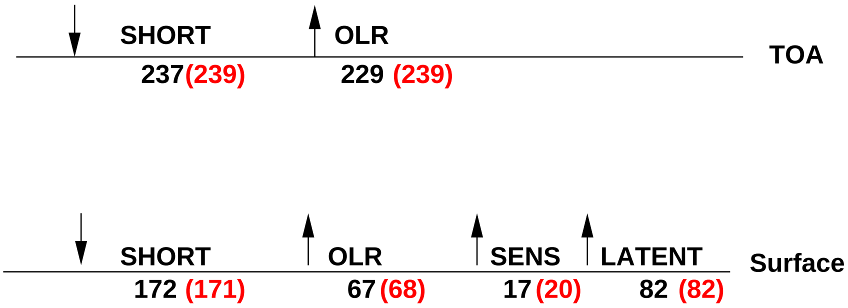
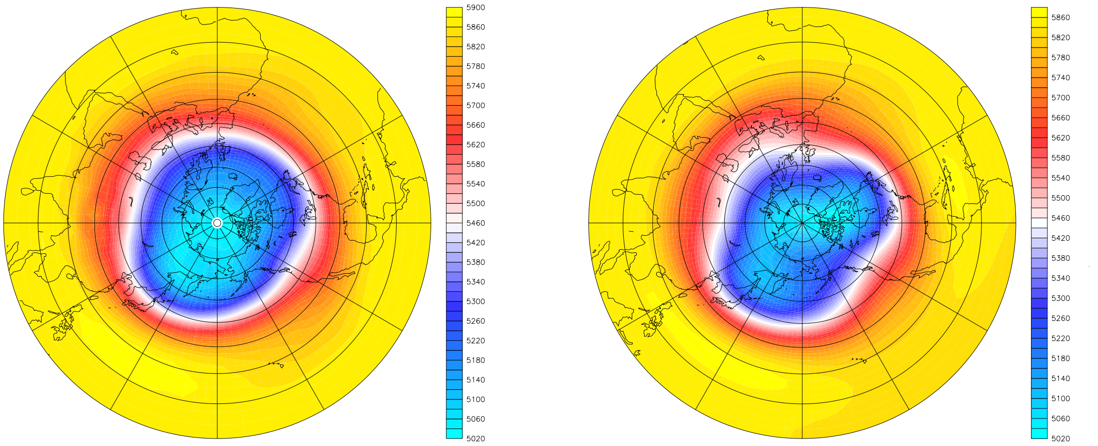


Figure 3: Global averaged radiation budget for top of the atmosphere (TOA) and surface in W/m^2 of the Speedy-LBM-Sea model (black) and Peixoto and Oort (1991) (red).

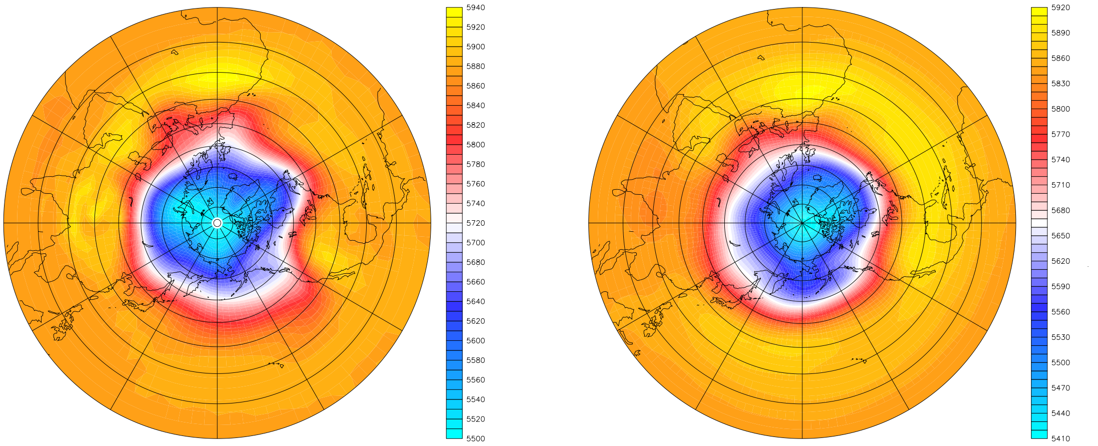
4.2 Geopotential Height

The geopotential height at 500 hPa gives an impression of the large scale circulation in the atmosphere in the midlatitudes. The geopotential height as simulated by the model is compared to results from the NCEP/Reanalysis project (Kalnay et al. 1996). in Figures 4 to 7. The model has bias in the mean geopotential height of about 60m. Despite the bias, the model has winter stationary wave patterns over the Northern Hemisphere that are consistent with the Reanalysis data over the midlatitudes. The ridge with high values over Europe is too weak and the positive excursion of the 500 hPa geopotential height over the Pacific is too strong. The errors are acceptably small in winter, such that the general circulation in the midlatitudes is well simulated. In summer, the circulation deviates most strongly from the Reanalysis data over Asia. As a result, the summer monsoon in Asia is poorly simulated.



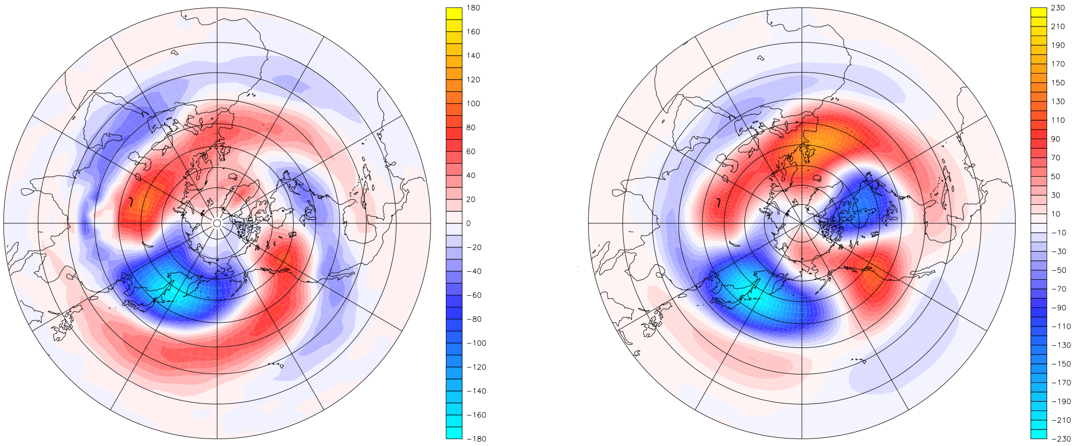
GH500 (DJF) GH500 (DJF, NCEP)

Figure 4: Geopotential height at 500 hPa, December, January, February average. Left: Speedy-LBM-Sea, right: NCEP/Reanalysis



GH500 (JJA) GH500 (JJA, NCEP)

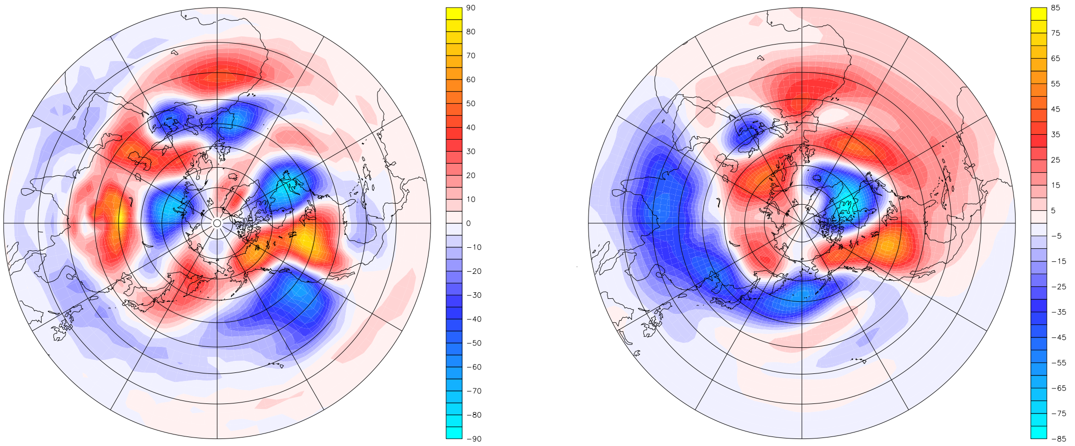
Figure 5: Geopotential height at 500 hPa, June, July, August average. Left: Speedy-LBM-Sea, right: NCEP/Reanalysis



Zonal eddy component of GH500 (DJF)

Zonal eddy component of GH500 (DJF, NCEP)

Figure 6: Eddy component (defined as deviation from zonal mean) of geopotential height at 500 hPa, December, January, February average. Left: Speedy-LBM-Sea, right: NCEP/Reanalysis.



Zonal eddy component of GH500 (JJA)

Zonal eddy component of GH500 (JJA, NCEP)

Figure 7: Eddy component (defined as deviation from zonal mean) of geopotential height at 500 hPa, June, July, August average. Left: Speedy-LBM-Sea, right: NCEP/Reanalysis.

4.3 Zonal winds and wind stresses.

Low level winds and wind stresses at the air/sea interface are important for forcing the ocean and in determining the surface heat fluxes. The zonal winds and wind stresses are compared to those in the NCEP/Reanalysis data in Figures 8 to 11. It is clear that the wind stresses and the 925 hPa winds are too strong in the Northern Hemisphere of the model. Both the westerlies and the trades are stronger than in Reanalysis data. However, the location of the major wind belts is well simulated. Also, the errors over sea are in general smaller than over land. In general the structure of the zonal winds is well simulated compared to Reanalysis data.

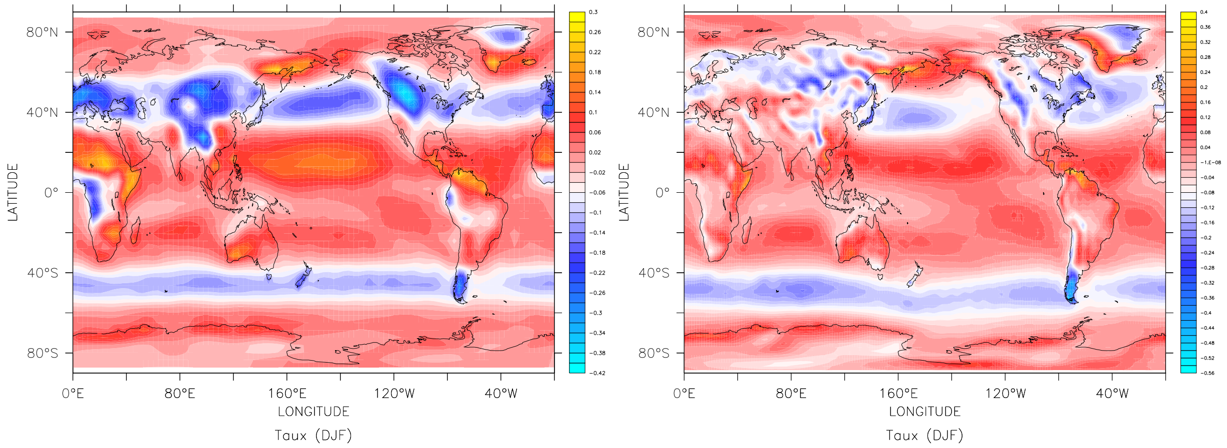


Figure 8: Zonal wind stress, December, January, February average. Left: Speedy-LBM-Sea, right: NCEP/Reanalysis.

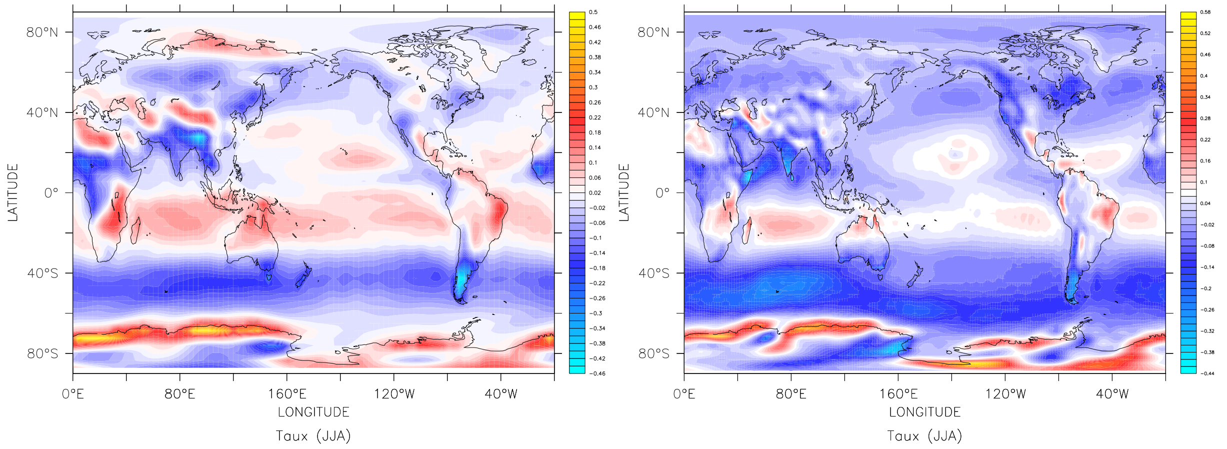


Figure 9: Zonal wind stress, June, July, August average. Left: Speedy-LBM-Sea, right: NCEP/Reanalysis.

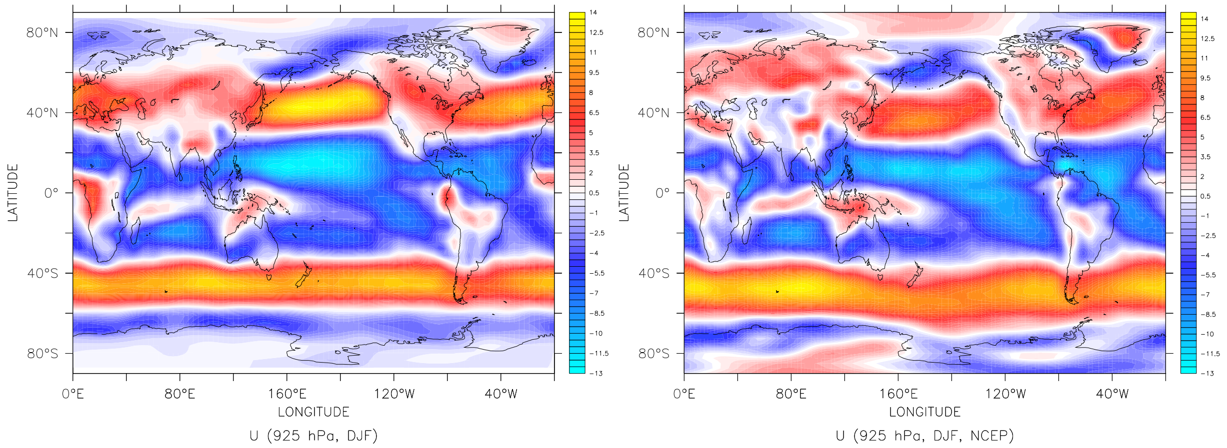


Figure 10: Zonal wind at 925 hPa, December, January, February average. Left: Speedy-LBM-Sea, right: NCEP/Reanalysis.

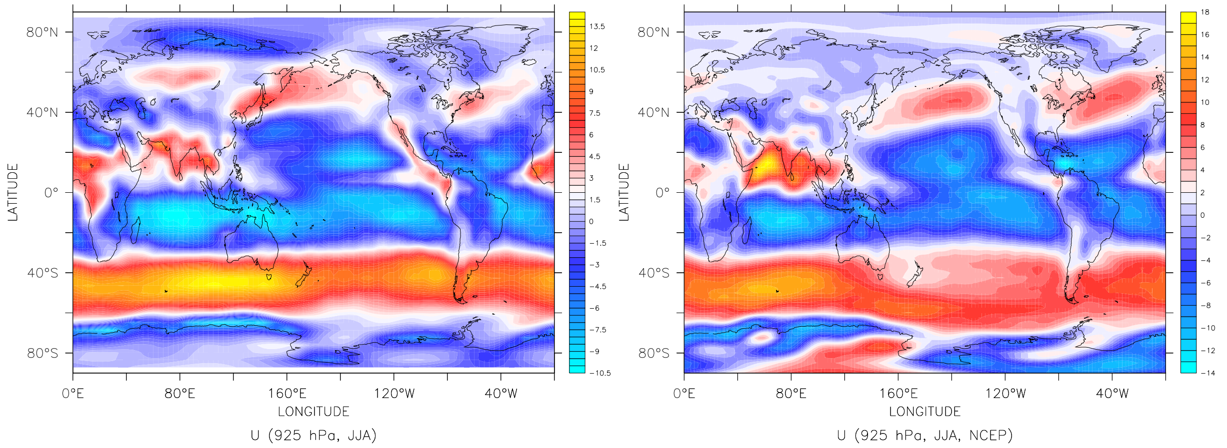
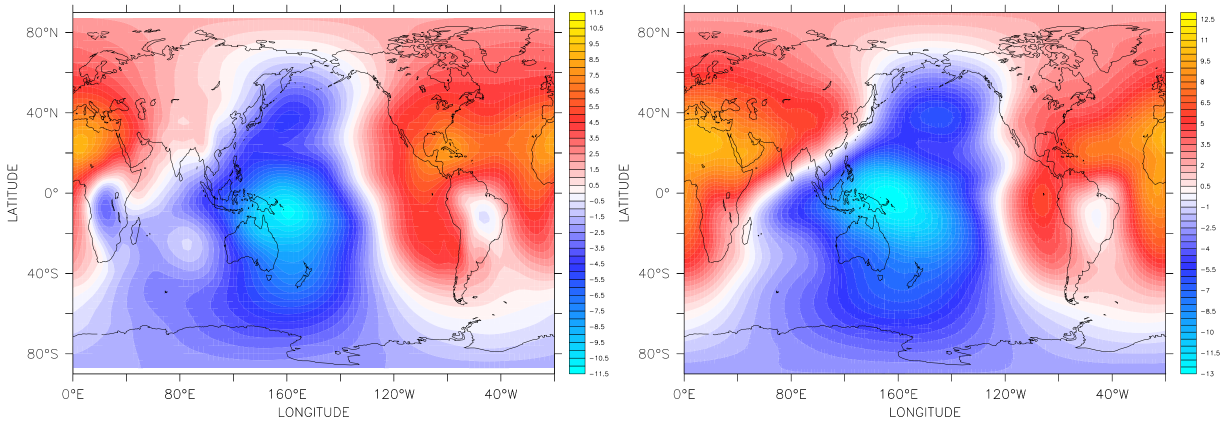


Figure 11: Zonal wind at 925 hPa, June, July, August average. Left: Speedy-LBM-Sea, right: NCEP/Reanalysis.

4.4 Velocity Potential

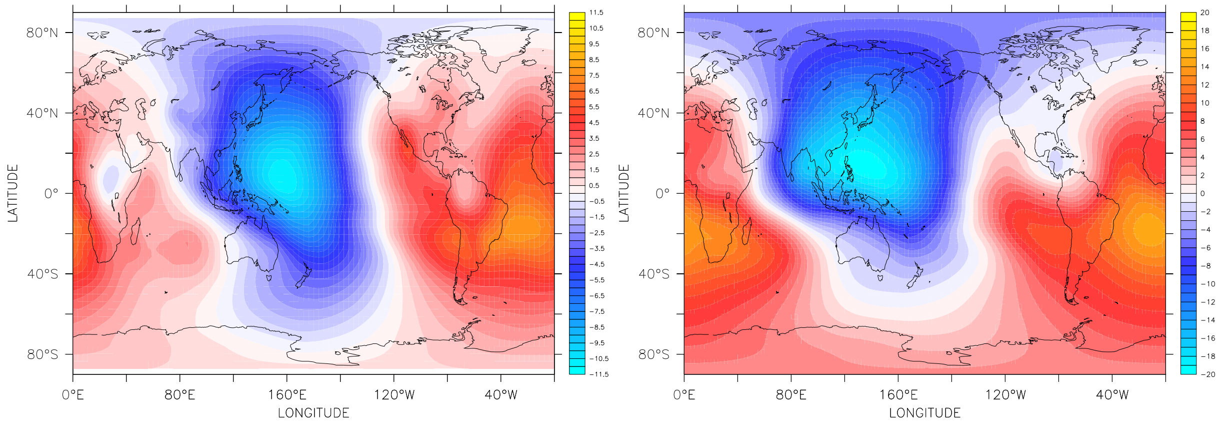
The velocity potential is compared to NCEP/Reanalysis in Figures 12 and 13. It reveals the large scale convergences and divergences of the atmospheric flow and, therefore, the atmospheric large scale overturning cells. The most important feature is the low level convergence and high level convergence in the warm pool area of the Pacific. The model simulates this well. The convergence at high altitudes over Northern Africa in the boreal winter and over the South Atlantic in the boreal summer are also well represented. Departures from the Reanalysis velocity potential are found in the divergence over eastern Africa in boreal winter and the exact location of the hight-altitude convergence of the warm pool in the boreal summer.



Velocity Potential (200 hPa, DJF)

Velocity Potential (200 hPa, DJF, NCEP)

Figure 12: Velocity potential at 200 hPa, December, January, February average. Left: Speedy-LBM-Sea, right: NCEP/Reanalysis.



Velocity Potential (200 hPa, JJA)

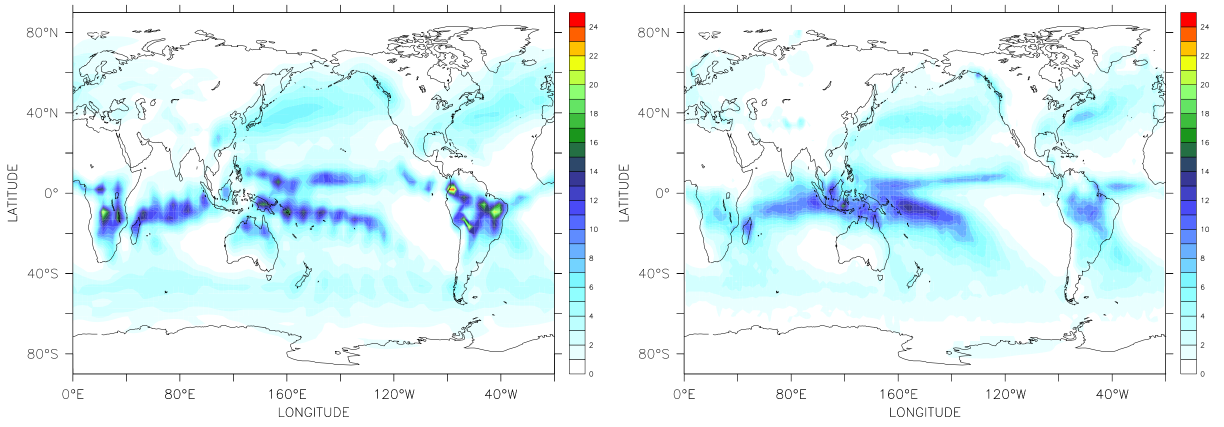
Velocity Potential (200 hPa, JJA, NCEP)

Figure 13: Velocity potential at 200 hPa, June, July, August average. Left: Speedy-LBM-Sea, right: NCEP/Reanalysis.

4.5 Precipitation and cloud cover

Precipitation is compared to the merged analysis and satellite data of CMAP (Xie and Arkin, 1996). The simulated precipitation reveals some noise that might be due the spectral representation of the grid. The general structure is good, with an intense ITCZ north of the equator and clear convergence zones in the South Pacific, Indian Ocean and South Atlantic. Also, large scale precipitation in the storm track regions is simulated. However, there is too much rainfall in the South Pacific Convergence Zone, in a number of points in South America and in southeast Africa in boreal winter. The large precipitation rates near Equador are a common problem of spectral models. Also, it is clear that the summer monsoon over Asia is not well simulated.

The cloud cover is compared to ISCCP data (Bishop and Rossow, 1991). The major bands of high cloud cover are simulated in the model. The new cloud scheme improved this markedly. However, in general, the model simulates not enough cloud. Higher cloud amounts would scrutanize the radiation budget though. The stratiform cloud over the South Pacific and South Atlantic is well simulated, but there is a lack of stratiform cloud in the (sub)tropical North Pacific and North Atlantic. Also, there is too much cloud over the equatorial cold sea surface temperature tongues.



Precipitation (DJF)

Precipitation (DJF, CMAP)

Figure 14: Precipitation, December, January, February average. Left: Speedy-LBM-Sea, right: CMAP.

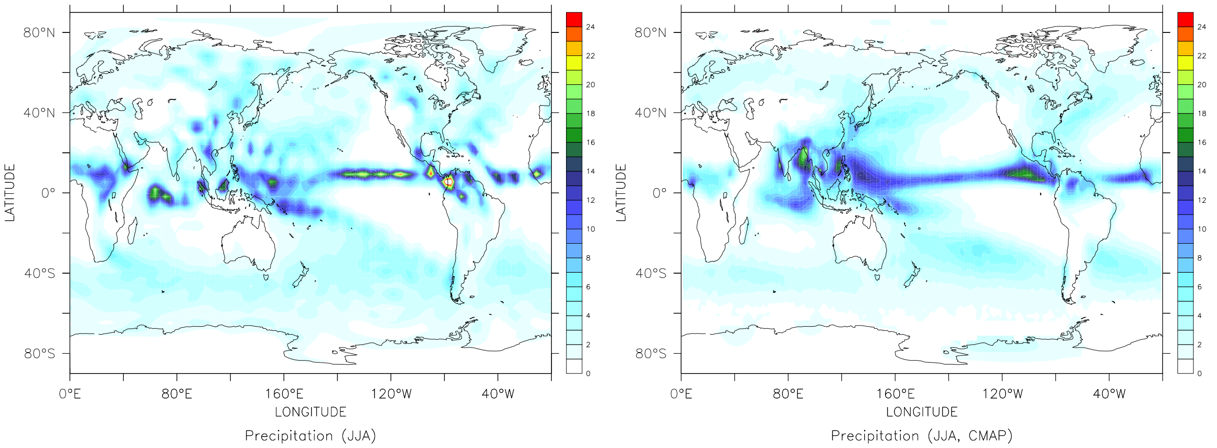


Figure 15: Precipitation, June, July, August average. Left: Speedy-LBM-Sea, right: CMAP.

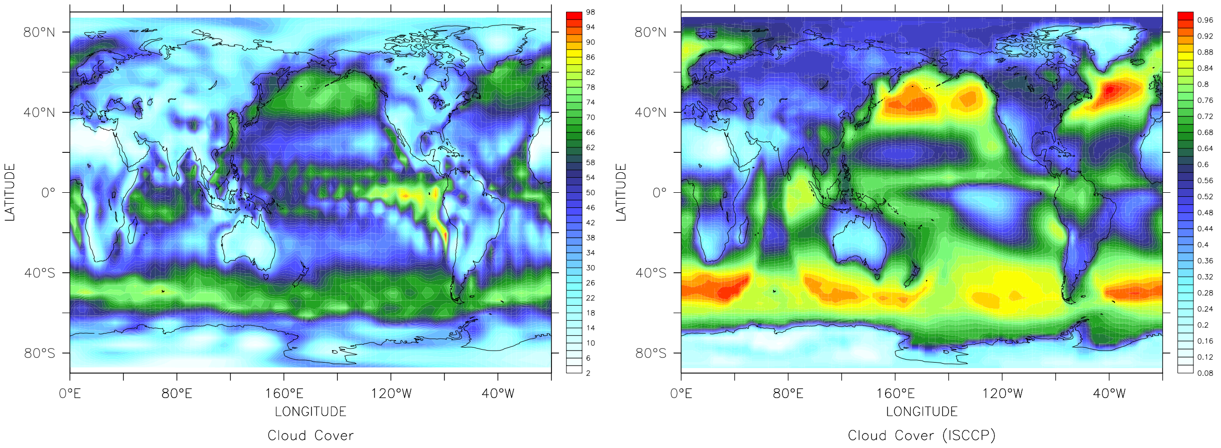


Figure 16: Annual mean cloud cover. Left: Speedy-LBM-Sea, right: ISCCP.

4.6 The NAO and PNA patterns

The main modes of atmospheric variability in the Northern Hemisphere are the North Atlantic Oscillation (NAO) and the Pacific North America pattern (PNA). As the model will be used for climate variability studies, we study how these modes are represented in the model. The location of the North Atlantic Oscillation (NAO) pattern is very well simulated (Fig. 17). The maxima and minima of the dipole correspond well to those in the Reanalysis data. The associated surface temperature and precipitation pattern are comparable to the Reanalysis data as well. That is, the correlation between near surface temperature and the NAO time series has a dipolar structure over Western Europe and along the eastcoast of North America. Furhtermore, the correlation between the NAO time series and precipitation has a banded structure over the North Atlantic and Europe.

The PNA pattern is not as well represented as the NAO. The pattern located too far to the north in this model. However, the exact location depends very much on the domain that is used for the analysis.

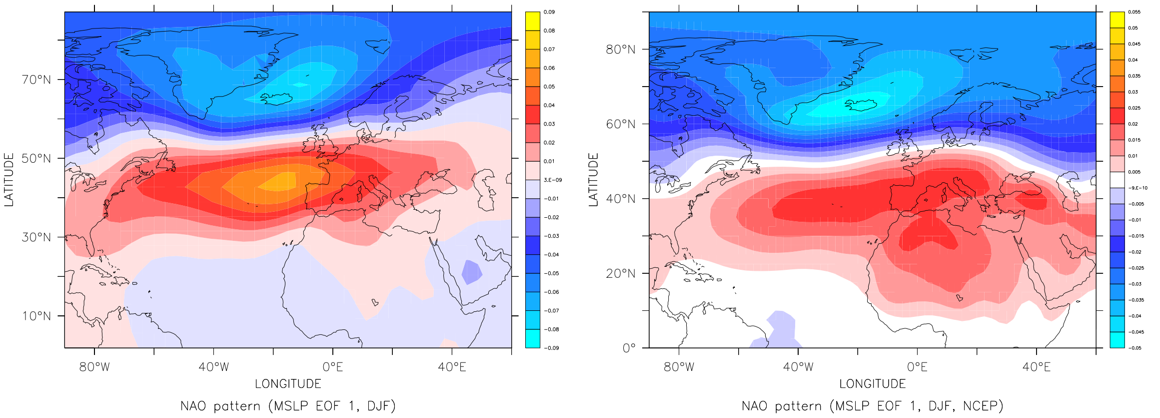


Figure 17: First empirical orthogonal function of December, January, February mean sea level pressure: the NAO pattern. Left: Speedy-Sea-LBM, right: NCEP/Reanalysis.

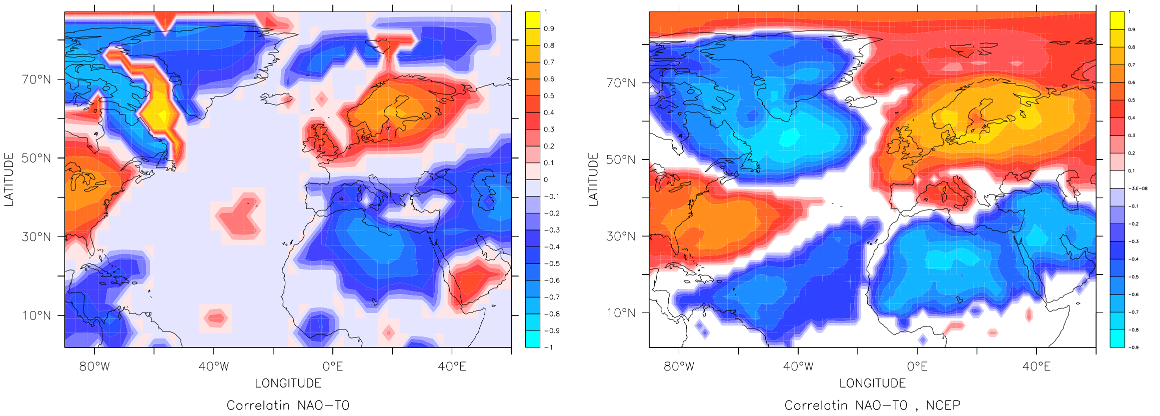


Figure 18: Corrlation of near surface temperature with the NAO time series (principal component of 1st EOF of DJF mean sea level pressure). Left: Speedy-Sea-LBM, right: NCEP/Reanalysis.

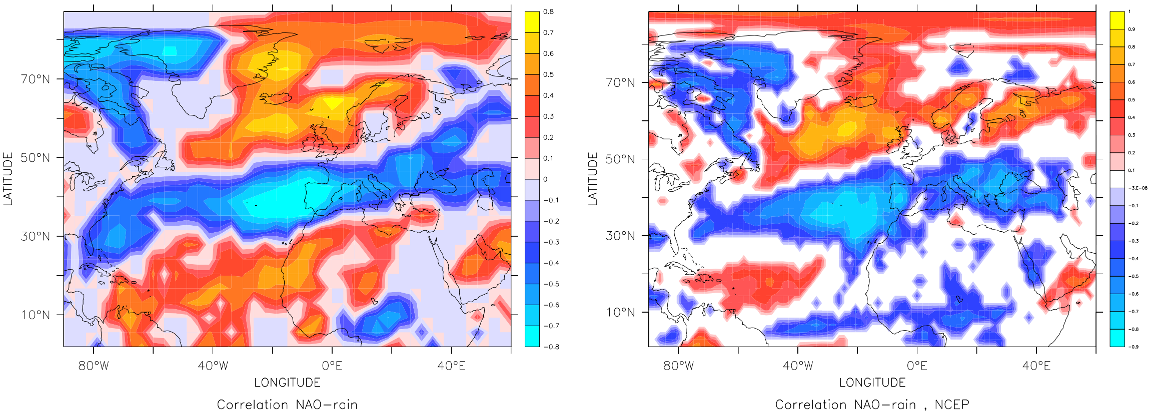


Figure 19: Corrlation of precipitation (right) with the NAO time series (principal component of 1st EOF of DJF mean sea level pressure). Left: Speedy-Sea-LBM, right: NCEP/Reanalysis.

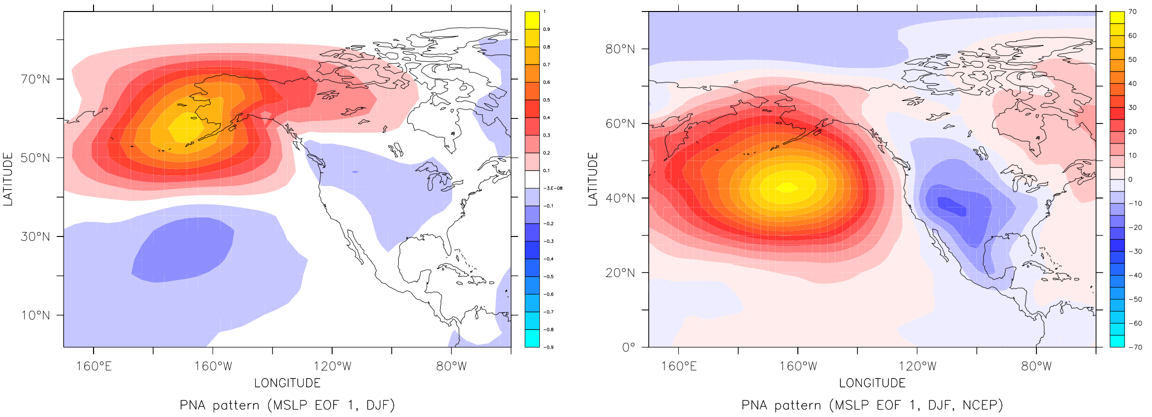


Figure 20: First empirical orthogonal function of December, January, February mean sea level pressure: the PNA pattern. Left: Speedy-Sea-LBM, right: NCEP/Reanalysis.

4.7 El Niño relations

The most important mode of variability in climate on interannual time scales is the El Niño Southern Oscillation. A few results are shown of relationships between atmospheric fields and El Niño. Due to internal variability in the atmosphere, the relationship between tropical and midlatitude variability is not robust. An ensemble of runs would have been needed to find a signficant response.

It is well known that the trade winds relax in response to warming in the eastern tropical Pacific. The model shows this response in the central tropical Pacific in a similar fashion as the Reanalysis data (Fig. 21). The weakening is well captured, but the magnitude of the weakening in the model is weaker (not shown). The strongest response is found in September, October, November.

As shown in many studies, there is a profound remote response to ENSO (e.g. Alexander et al. 2002). Here we show how precipitation and the geopotential height at 500 hPa is related to ENSO. The precipitation anomalies associated with El Niño show a comparable pattern as in the Renalysis. That is, increased precipitation over the central tropical Pacific and reduced rainfall in the western Pacific north of the equator. These changes are connected to a shift of atmospheric convection from the western to central Pacific. The response of the 500 hPa geopotential height shows an overall rise in the tropics and the teleconnections to the midlatitudes. The model captures the wave train response to the north and south. The geopotential height associated with the Aleuthian low is anomalously low in response to El Niño. A positive anomaly is found over Canada. In the South a positive anomaly is found to the southwest of South America and a negative anomaly in the South Atlantic. These features are common to the model and the Reanalysis data showing that the model is well capable of reproducing teleconnection patterns associated with ENSO.

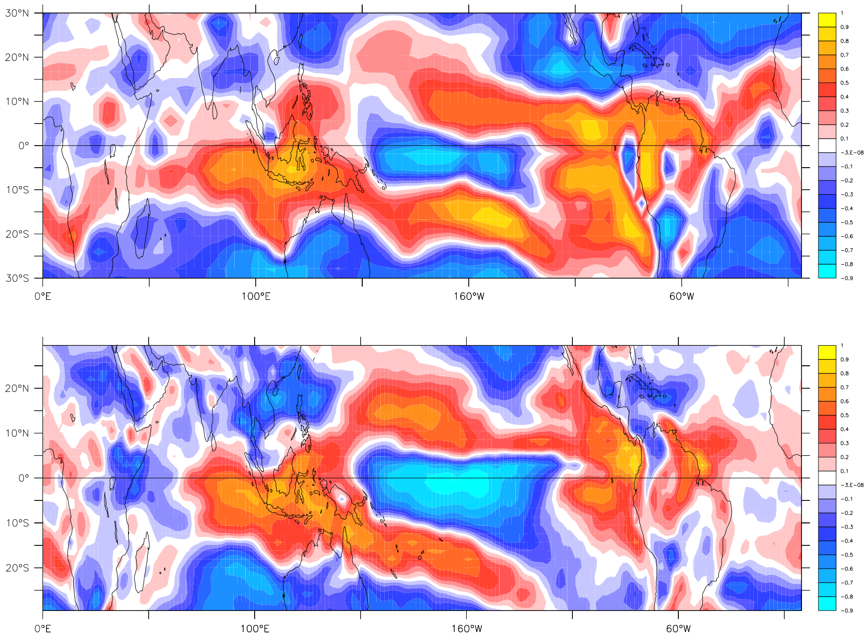


Figure 21: Correlation between NINO3 SST anomalies and zonal windstress, September, October, November average. Top: Speedy-Sea-LBM, Bottom: NCEP/Reanalysis.

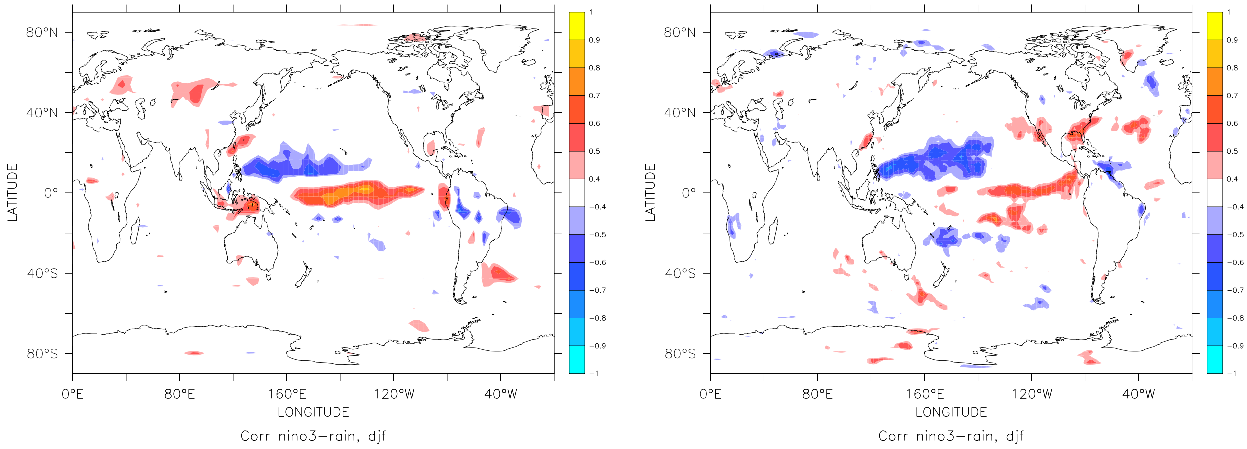


Figure 22: Correlation between NINO3 SST anomalies and precipitation anomalies, December, January, February average. Left: Speedy-Sea-LBM, right: NCEP/Reanalysis.

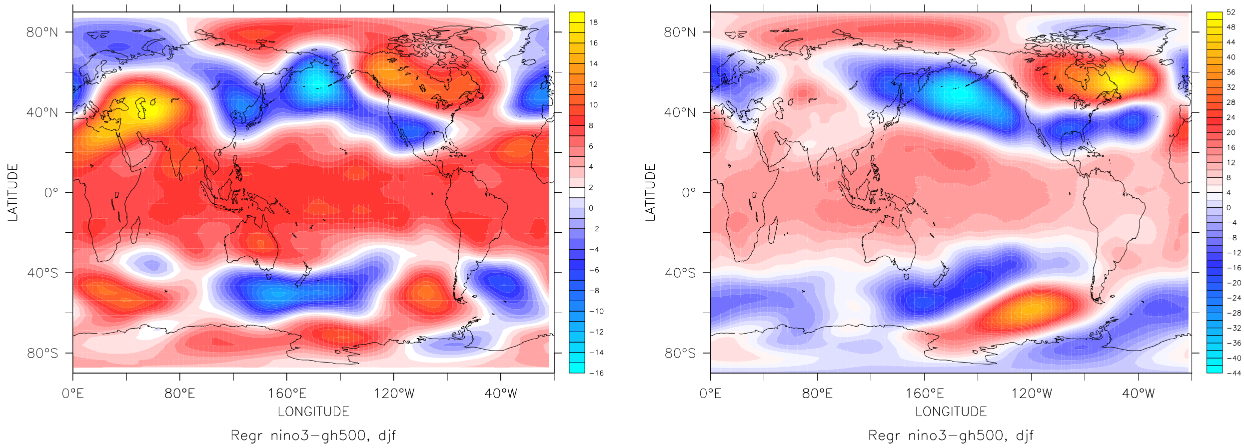


Figure 23: Regression of NINO3 SST anomalies and geopotential height at 500 hPa, December, January, February average. Left: Speedy-Sea-LBM, right: NCEP/Reanalysis.

5 References

Alexander, M.A., I. Blade, M. Newman, J.R. Lanzant, N. Lau and J.D. Scott, 2002: The atmospheric bridge: The influence of ENSO teleconnections on air-sea interaction over the global oceans. *J. Clim.*, **15**, 2205-2231.

Beersma, J.J., R. van Dorland, and J.D. Opsteegh, 2002: Shortwave radiation and cloud parameterizations for intermediate complexity models. *KNMI, Scientific report, WR 2002-02*, 36 pp.

Bishop, J.K.B., and W.B. Rossow, 1991: Spatial and temporal variability of global surface solar irradiance. *J. Geophys. Res.*, **96**, 16,839-16,858

Bleck, R., C. Rooth, D. Hu, and L.T. Smith, 1992: Salinity-driven thermocline transients in a wind- and thermohaline-forced isopycnic coordinate model of the North Atlantic. *J. Phys. Oceanogr.*, **22**, 1486-1505

Burgers, G., M.A. Balmaseda, F.C. Vossepoel, G.J. van Oldenborgh, and P.J. van Leeuwen, 2002: Balanced ocean-date assimilation near the equator. *J. Phys. Oceanogr.*, **32**, 2509-2519.

Da Silva, A.M., C.C. Young, and S. Levitus, *Atlas of surface marine data 1994*, vol 1, *Algorithms and Procedures, NOAA Atlas 6*, 83 pp. Natl. Oceanic and Atmos. Admin., U.S. Dep. of Commer., Washington, D.C., 1994.

Dorland, R. van, 1999: *Radiation and Climate. From Radiative Transfer Modelling to Global Temperature Response.* Thesis RUU, ISBN 90-646-4032-7, 147 pp.

R.J. Haarsma, F.M. Selten, J.D. Opsteegh, G. Lenderink and Q. Liu, 1996: *ECBILT: A coupled atmosphere ocean sea-ice model for climate predictability studies.* Technical report; TR-195, KNMI, The Netherlands.

Hurrell, J. W., 1995: Decadal trends in the North Atlantic Oscillation: regional temperatures and precipitation. *Science*, **269**, 676-679.

Kalnay, E. et al., 1996: The NCEP/NCAR 40-year reanalysis project. *Bull. Amer. Meteorol. Soc.*, **77**, 437-471.

Levitus, S. and T.P. Boyer, 1994: World Ocean Atlas 1994, volume 4: Temperature, *NOAA ATLAS NESDIS 4*, 117 pp.

Levitus, S., R. Burgett, and T.P. Boyer, 1994: World Ocean Atlas 1994, volume 3: Salinity, *NOAA ATLAS NESDIS 3*, 99 pp.

Mehta, V.M. and T. Delworth. 1995. Decadal variability of the tropical Atlantic ocean surface temperature in shipboard measurements and in a global ocean-atmosphere model. *J. Climate*, **9**, 172-190.

Molteni, F., 2003: Atmospheric simulations using a GCM with simplified physical parameterizations. I: model climatoloty and variability in multi-decadal experiments. *Clim. Dyn.*, **20**, 175-191.

Niiler, P.P. and E.B. Kraus, 1977: One-dimensional models of the upper ocean, in Modeling and Prediction of the Upper Layers of the Ocean, edited by E.B. Kraus, pp. 143-172, Pergamom, New York.

Opsteegh, J.D., R.J. Haarsma and F.M. Selten, 1998: ECBILT; A dynamic alternative to mixed boundary conditions in ocean models. **Tellus**, *50A*, 348-367.

Peixoto, J.P. and A.H. Oort, 1991: Physics of Climate. *American Institute of Physics*, 520 pp.

Roeckner, E., K. Arpe, L. Bengtsson, M. Christoph, M. Claussen, L. Dümenil, M. Esch, M. Giorgetta, U. Schlese, U. Schulzweida, 1996: *The atmospheric general circulation model ECHAM4: model description and simulation of present-day climate*, Max-Planck-Institut für Meteorologie, Report 218, Hamburg, 90 pp.

Xie and Arkin, 1996: Analyses of Global Monthly Precipitation Using Gauge Observations, Satellite Estimates, and Numerical Model Predictions. *J. Climate*, **9**, 840 -858.

Zhang, Y., J. M. Wallace, and D. S. Battisti, ENSO-like interdecadal variability: 1900-1993, *J. Climate,* **10,** 1004-1020, 1997.

6 Appendices

APPENDIX 1: Description of the Generic Model Framework

1. Functionality

The Generic Model Framework (GMF) provides the functionality required by generic circulation models. This functionality consists of keeping track of time both as a number of time steps and in a calendar, collecting model data to be output in a history file, and storage and retrieval of the model state from restart files. Output of restart and history data is done according to a time table. The GMF also defines a standard structure for organizing the model code in modules and controls the initialization and time-stepping phases. The flow of control defined in the GMF is as follows:

1. Initialization phase
2. Time stepping loop: repeat while not at the end of the run
   a. Compute one iteration in the model physics
   b. Collect data for the history output and output to history file according to a time table
   c. Store model state in a restart file according to a time table
   d. Prepare for the next iteration

2. The Initialization Phase

In the initialization phase each module of the GMF and the model specific is initialized in the following manner:

1. read all module parameters from the parameter namelist file
2. when a restart file is open, load all resources that need to be recovered from the restart file.
3. compute/initialize additional state of the model; register variables in history
4. check that the model is in valid state

If any of these steps fails, the GMF prints the state of the module as is and aborts the run. This initialization procedure is implemented by a subroutine of the GMF (see example code in section 4 below). N.B.:

- the resources in step 2 consist of the parameters from step 1 plus any other variables that are needed to recover the state of the module, e.g., prognostic variables.
- only parameters are loaded in step 2 that override the values from the parameters namelist file. This depends on the model run type (recovering after crash, continuing after spinup). In case of a new run the restart file is not used.

3. The Time Stepping Loop

The source code below is taken from the time stepping loop of the GMF. Parts that are not relevant here have been removed.

```
SUBROUTINE Model_run
  LOGICAL ::  more
  more = TimeStep_next(nextTimeStepCallback)
  DO WHILE (more)
   CALL Physics_compute
   CALL History_collect(collectHistoryCallback)
   CALL Restart_store(storeInRestartFileCallback)
   more = TimeStep_next(nextTimeStepCallback)
  END DO
  RETURN
END SUBROUTINE Model_run
```

The subroutine call to History_collect gathers all data that should be stored in the history output file. Optional time averaging of the data is done just before the data is written to the history file. A time table in the parameters namelist file determines when the history output file is updated during the run. The call to the Restart_store subroutine creates restart files. Restart files are only created according to the restart time table in the parameters namelist file. At all other moments this call is effectively a no-op. The function TimeStep_next notifies all modules of the GMF and the model to prepare for the next time step and checks whether the end of the run has been reached.

4. Standard Module Structure

The source code below is a typical example of the structure of a module in the GMF. This code is taken from the Land component model. Some explanation of the code is given below each code fragment in italics.

```
MODULE Land
  USE GMF
  USE GMFHistory
  USE GMFModule
  USE GMFResource
  USE GMFSupport
  USE ...
  IMPLICIT NONE
  PRIVATE
```

The declaration of the module. Some modules from the GMF that are often needed in the code are "used". Additional modules might be required. By default everything declared in this module is made private to this module.

```
  PUBLIC ::  Land_compute
  PUBLIC ::  Land_initialize
  PUBLIC ::  Land_collectHistory
  PUBLIC ::  Land_storeInRestartFile
  PUBLIC ::  Land_nextTimeStep
  PUBLIC ::  Land_printState
  ...
```

These are the public standard subroutines required by the GMF. There implementation is given later on.

```
TYPE(ModuleInfo) ::  module_info = ModuleInfo("Land",.FALSE.,.FALSE.)
```

The above variable is used by the GMF to maintain some metainformation about this module.

```
    TYPE(Resource), PARAMETER ::  resources(...)  = &
      & (/ &
      & Resource("iseasc",Restart_Value_Used_On_Recovery), &
      ...      & /)
```

This is a list of variables (resources) that the GMF should recover from a restart file (and store in a restart file when creating one). The second value, Restart_Value_Used_On_Recovery *in this example, indicates in which cases the variable should be taken from the restart file instead of from the parameter file.*

```
    TYPE(HistoryId) ::  net_heat_flux_id, net_heat_flux_anomaly_id
    ...
```

These are identifiers for variables that the GMF should store in the history output file.

```
  CONTAINS
    SUBROUTINE Land_compute
      ...
      RETURN
    END SUBROUTINE Land_compute
```

This subroutine performs the computations that are required for one iteration in the time step loop.

```
    SUBROUTINE Land_Initialize
      USE GMFProcess
      CALL Process_initializeModule(module_info,readParameterFile,readRestartFile,
       & initializationHook,Process_stateIsOkay,printState)
      RETURN
    END SUBROUTINE Land_Initialize
```

This subroutine is called by the GMF to initialize this module. All modules should implement this subroutine as shown here: the subroutine Process_initializeModule *implements the standard procedure for the initialization of a module. In addition to the module's metainformation, the names of the functions that read parameters from the parameter file and restart file should be passed (see below for an implementation example).* The initializationHook *function can be used to complete initialization when all parameters have been read. The last two functions check and print the state of this module.*

```
    SUBROUTINE Land_collectHistory
      IF (output_net_heat_flux)
       CALL History_storeVariable(net_heat_flux_id,hfavel(:,:))
      ...
      RETURN
    END SUBROUTINE Land_collectHistory
```

This subroutine adds data to the history.

```
    SUBROUTINE Land_storeInRestartFile
      USE GMFRestartIO
      CALL RestartIO_storeResource(Trim(module_info    ...
      RETURN
    END SUBROUTINE Land_storeInRestartFile
```

This subroutine stores resources in the restart file.

```
SUBROUTINE Land_nextTimeStep
  USE GMFTimeStep
  CALL synchronize
  RETURN
END SUBROUTINE Land_nextTimeStep
```

This subroutine is called by the GMF to let the module take actions that should be performed before the next iteration of the time step loop.

```
SUBROUTINE Land_printState
  USE GMFProcess
  CALL Process_printModuleState(module_info,printState)
  RETURN
END SUBROUTINE Land_printState
```

This subroutine prints the state of this module. The subroutine Process_printModuleState *adds a header in a standard format. The subroutine printState prints the information that is specific to this module.*

```
!  PRIVATE METHODS
FUNCTION readParameterFile(module_info,parameter_file_unit) RESULT(success)
  LOGICAL ::  success
  TYPE(ModuleInfo), INTENT(in) ::  module_info
  INTEGER, INTENT(in) ::  parameter_file_unit
  INTEGER ::  ios NAMELIST /Land/ iseasc, ...
  READ(unit=parameter_file_unit,nml=Land,iostat=ios)
  success = ( ios == 0 )
  RETURN
END FUNCTION readParameterFile
```

This function is used during initialization to read the parameters of this module from the parameters namelist file. The return value indicates whether this operation was successfully completed.

```
FUNCTION readRestartFile(module_info) RESULT(success)
  USE GMFRestartIO
  LOGICAL ::  success
  TYPE(ModuleInfo), INTENT(in) ::  module_info
  CALL RestartIO_loadResource(Trim(module_info    ...
  success = .TRUE.
  RETURN
END FUNCTION readRestartFile
```

This function is used during initialization to read the resources of this module from a restart file. It is only called when a run is restarted. Each resource loaded here a corresponding store operation should be present in the subroutine Land_storeInRestartFile *above.*

```
FUNCTION initializationHook(module_info) RESULT(success)
  USE GMFTimeStep
  LOGICAL ::  success
  TYPE(ModuleInfo), INTENT(in) ::  module_info
  CALL TimeStep_setSize(86400) !  Time step is one day in seconds
  ...
```

```
    success = registerVariables()
    IF (success) THEN
    ...
    END IF
    RETURN
  END FUNCTION initializationHook
```

This function is called during initiailization after all parameters and resources have been loaded. In this function variables are typically registered in the history. This is also the location to start the coupler.

```
  SUBROUTINE printState(module_info)
    TYPE(ModuleInfo), INTENT(in) ::  module_info
    PRINT "('Seasonal cycle flag (iseasc) :  ',i1)", iseasc
    ...
    RETURN
  END SUBROUTINE printState
```

This subroutine should print all information about this module that is useful to a user of the model.

```
  FUNCTION registerVariables() RESULT(success)
    LOGICAL ::  success
    TYPE(CoordinateSpecification) ::  lon_spec, lat_spec
    TYPE(GridId)                  ::  grid_id
    REAL                          ::  sia(nlat2), wt(nlat2), rad1
    INTEGER                       ::  j
    CALL History_setFillValue(huge)
    lon_spec%name             = "lon"
    lon_spec%description      = "Longitude"
    lon_spec%unit             = "degrees_east"
    lon_spec%regular          = .TRUE.
    lon_spec%number_of_points  = nlon
    lon_spec%start_value      = 0.0
    lon_spec%step_size       = 360.0/nlon
    lat_spec%name = "lat"
    lat_spec%description = "Latitude"
    lat_spec%unit = "degrees_north"
    lat_spec%regular = .FALSE.
    lat_spec%number_of_points = nlat
    CALL gaussl(sia,wt,nlat2)
    DO j=1,nlat2
      rad1 = ASin(sia(j))*Pi/180.0
      lat_spec%values( j) =-rad1
      lat_spec%values(nlat+1-j) = rad1
    ENDDO
    grid_id = History_registerGrid(lon_spec,lat_spec)
    IF (output_net_heat_flux) THEN
     net_heat_flux_id = History_registerVariable("hfx","Net Heat Flux","W/m2",grid_id)
    END IF
    ...      success = .TRUE.
    RETURN
 END FUNCTION registerVariables
```

This function shows how variables can be registered in the history of the GMF. First a fill value is defined, than a longitude and latitude coordinate are specified and a grid is constructed. Hereafter the variables can be registered. Note that multiple grids can be defined.

```
    SUBROUTINE synchronize
      ...
      RETURN
    END SUBROUTINE synchronize
    ...
```

Other private subroutines and functions of the module can be implemented here.

```
END MODULE Land
```

APPENDIX 2: Include file with physics parameters of Speedy:`cls_inphys.h`

```
C--
C--   Constants for physical parametrization routines:

C--   Constants for boundary forcing (common FORCON):

      SOLC   = 342.

      ALBSEA = 0.07
      ALBICE = 0.60
      ALBSN  = 0.60
      SDALB  = 60.

      SWCAP = 0.30
      SWWIL = 0.17

C--   Constants for convection (common CNVCON):

      PSMIN  =  0.8
      TRCNV  =  6.
      RHBL   =  0.9
      RHIL   =  0.7
      ENTMAX =  0.5
      SMF    =  0.8

C--   Constants for large−scale condensation (common LSCCON):

      TRLSC  = 4.
      RHLSC  = 0.9
      DRHLSC = 0.1
      RHBLSC = 0.95

C--   Constants for radiation (common RADCON):

      RHCL1  =  0.30
      RHCL2  =  1.00
      QACL   =  0.20
      WPCL   =  0.3
      PMAXCL = 15.0

      ALBCL  =  0.55 ! Was: 0.35
      ALBMIN =  0.0! Was: ALBCL*0.3

      EPSSW  =  0.015
      EPSLW  =  0.05
      EMISFC =  0.98

      ABSDRY =  0.033
      ABSAER =  0.033
      ABSWV1 =  0.022
      ABSWV2 = 15.000

      ABSCL1 =  0.030 ! Was: 0.015
      ABSCL2 =  0.30  ! Was: 0.15

      ABLWIN =  0.3
      ABLCO2 =  4.0
      ABLWV1 =  1.0  !! 1.4 ! Was: 0.7
      ABLWV2 = 50.0

      ABLCL1 = 24.0 ! Was: 12.0
      ABLCL2 =  1.2 !! 0.6 ! Was: 0.6

C--   Constants for surface fluxes (common SFLCON):

      FWIND0 = 0.6
      FTEMP0 = 1.
      FHUM0  = 0.5

      CDL = 2.4e−3
      CDS = 0.8e−3
      CHL = 1.2e−3 ! Was: 1.2 e−3
      CHS = 1.0e−3 ! Was: 0.8e−3

      VGUST  = 5.
      CTDAY  = 1.0e−2
      DTHETA = 3.
      FSTAB  = 0.67
      HDRAG  = 2000.
      FHDRAG = 0.5

C--   Constants for vertical diffusion and sh. conv. (common VDICON):

      TRSHC  =  6.
      TRVDI  = 24.
      TRVDS  =  6.
      REDSHC =  0.5
      RHGRAD =  0.5
      SEGRAD =  0.1
```

APPENDIX 3: Include file with time step parameters of Speedy: `cls_instep.h`

```
C--
C--   Length of the integration and time stepping constants (common ISTEPS)

      NMONTS = 1200
      NDAYSL = 30
      NSTEPS = 36

      NSTDIA = 36*30
      NSTPPR = 6
      NSTOUT = 36*30
      IDOUT  = 1
      NMONRS = 12

      ISEASC = 1
      IYEAR0 = 1000
      IMONT0 = 1

      NSTRAD = 3
      NSTRDF = 0
      INDRDF = 0

      IALST  = 0
      IASST  = 0
      IAICE  = 0

      ISST0  = 25

C--
C--   Logical flags (common LFLAG1)

      LPPRES = .true.
```

APPENDIX 4: Example parameters file of LBM (similar parameter lists are in the FOcean, Sea, SlabOcean and Micom directories).

```
:
: FRAMEWORK PARAMETERS
:
:
: Model Information
:
: 1. LOGICAL :: write_initial_state_to_history
: 2. INTEGER :: model_run_type
:              0       New run, ignore any restart input file
:              1       Continue a run that was aborted from a restart input file
:                      [Model state is loaded from restart file, parameters file
:              2       Start a run from a spinned up state
:                      [Parameters file overrides restart file contents (in most
: 3. LOGICAL :: redirect_standard_output
: 4. CHARACTER(len=256) :: standard_output_filename
:
 &Model
       write_initial_state_to_history = .false.
       model_run_type                 = 0
       redirect_standard_output       = .true.
       standard_output_filename       = "LOG"
 &END
:
:--------------------------------------------------------------------------------
:
:
: Calendar Information
:
: 1. INTEGER :: calendar_type
:              1       Gregorian calendar      [Normal calendar]
:              2       360 day calendar        [12 months of 30 days]
:              3       365 day calendar        [As normal calendar but without le
:              4       366 day calendar        [12 months of 30.5 days]
: 2. CHARACTER(len=19) :: start_date           [In YYYY-MM-DD hh:mm:ss format]
: 3. CHARACTER(len=19) :: end_date             [In YYYY-MM-DD hh:mm:ss format]
:
:    CalendarDate Fields:
:       INTEGER :: year, month, day,  hour, minute, second
:
 &Calendar
       calendar_type =  2
       start_date    =  "1000-01-01 00:00:00"
       end_date      =  "1100-01-01 00:00:00"
 &END
:
: The following fields are overriden when continuing after a spinup:
:       3. end_date
:
:--------------------------------------------------------------------------------
:
:
: History Output Information
:
: 1. CHARACTER(len=1024) :: history_file_name   [can override restart file; do not
: 2. LOGICAL             :: average             [conditionally overrides restart f
:
 &History
       history_file_name     = "history"
       average               = .TRUE.
 &END
:
: History Output Time Table Information
:
: 1. TYPE(TimeTableDescriptionType) :: table(1:100)
:       [can override restart file, controlled from history output information]
:
:    TimeTableEntryType Fields:
:       CHARACTER(len=19) :: start_date, end_data
:       CHARACTER(len=16) :: increment
:
:    N.B. Last table entry should have end_date set to "End of run"
:
 &HistoryTimeTable
       table(1) = "1000-01-01 00:00:00", "End of run",  "1 month"
 &END
:
:--------------------------------------------------------------------------------
:
:
: Restart Input and Output Information
:
: 1. CHARACTER(len=1024) :: restart_in_file_name        [always overrides restart
: 3. CHARACTER(len=1024) :: restart_out_file_name       [can override restart file
: 4. INTEGER             :: restart_file_policy         [can override restart file
:              0       Never write a restart file
:              1       Overwrite last restart file
:              2       Keep last two restart files
:              3       Keep all restart files
:
 &Restart
       restart_in_file_name  = "restart_input"
       restart_out_file_name = "restart_output"
       restart_file_policy   = 0
 &END
:
: Restart Output Time Table Information
:
: 1. TYPE(TimeTableDescriptionType) :: table(1:100)
:       [can override restart file, controlled from restart output information]
:
:    See history time table information for a description of the time table format
:
 &RestartTimeTable
       table(1,1) = "1000-01-01 00:00:00"
       table(2,1) = "End of run"
       table(3,1) = "1 year"
 &END
:
: The following fields are overriden when continuing after a spinup:
:       All
:
:================================================================================
:
:
: LBM MODEL PHYSICS PARAMETER
:
:
:  1. CHARACTER(len=256) :: land_fraction_filename
:  2. CHARACTER(len=256) :: land_fraction_variable
:  3. CHARACTER(len=128) :: initial_land_temp_filename
:  4. CHARACTER(len=128) :: initial_land_temp_variable
:  5. CHARACTER(len=128) :: max_soil_moisture_filename
:  6. CHARACTER(len=128) :: max_soil_moisture_variable
:  7. CHARACTER(len=128) :: forest_fraction_filename
:  8. CHARACTER(len=128) :: forest_fraction_variable
:  9. REAL(kind=8)       :: max_bottom_moisture
: 10. REAL(kind=8)       :: max_snow_depth
: 11. REAL(kind=8)       :: max_snow_albedo
: 12. REAL(kind=8)       :: max_snow_albedo_depth
: 13. REAL(kind=8)       :: lhcap
: 14. LOGICAL            :: use_land_scenario
: 15. INTEGER            :: land_scenario_start_year
: 16. LOGICAL            :: check_land_model
: 17. LOGICAL            :: output_net_heat_flux
: 18. LOGICAL            :: output_rain_fall
: 19. LOGICAL            :: output_snow_fall
: 20. LOGICAL            :: output_evaporation
: 21. LOGICAL            :: output_soil_moisture
: 22. LOGICAL            :: output_soil_ice
: 23. LOGICAL            :: output_runoff
: 24. LOGICAL            :: output_snow_depth
: 25. LOGICAL            :: output_land_temp
: 26. LOGICAL            :: output_albedo
:
 &LBM
       land_fraction_filename     = "../Input/masks.nc"
       land_fraction_variable     = "FMASK1"
       initial_land_temp_filename = "inputdata/stl0.nc"
       initial_land_temp_variable = "STL0"
       max_soil_moisture_filename = "inputdata/sm.nc"
       max_soil_moisture_variable = "SM"
       forest_fraction_filename   = "inputdata/forest.nc"
       forest_fraction_variable   = "FOREST"
       max_snow_depth             = 10.0
       max_snow_albedo            = 0.7
       max_snow_albedo_depth      = 0.05
       lhcap                      = 8.0e6
       use_land_scenario          = .FALSE.
       land_scenario_start_year   = 1990
       check_land_model           = .FALSE.
       output_net_heat_flux       = .TRUE.
       output_rain_fall           = .TRUE.
       output_snow_fall           = .TRUE.
       output_evaporation         = .TRUE.
       output_soil_moisture       = .TRUE.
       output_soil_ice            = .TRUE.
       output_runoff              = .TRUE.
       output_snow_depth          = .TRUE.
       output_land_temp           = .TRUE.
       output_albedo              = .TRUE.
 &END
:
: The following fields are overriden when continuing after a spinup:
:       None
:
:================================================================================
```