PRISM

Project for Integrated Earth System Modelling

An Infrastructure Project for Climate Research in Europe

funded by the European Commission

under Contract EVR1-CT2001-40012

The
IPSL_CM4 Coupled Model
Adaptation Guide

Edited by:

Marie-Estelle Demory

PRISM-Report Series-9

Final Edition

(last change: November 30, 2004)

About this document

- This report was written in agreement with Marie-Alice Foujols and Arnaud Caubel.
- The chapter II 1 was written by Claire Levy and Gurvan Madec.
- The chapter II 2 was written in agreement with Jerome Cuny.
- The chapter II 3 was written in agreement with Laurent Fairhead and is based on the LMDZ documentation.
- The chapter II 4 was written according to the paper of Krinner et al. (2003), 'A dynamic global vegetation model for studies of the coupled atmosphere-biosphere system'.
- This report is available on the PRISM web page at IPSL: http://www.ipsl.jussieu.fr/~prisipsl/Docs/IPSL_CM4_PRISM_report.pdf

Persons to contact

- IPSL_CM4: Marie-Alice Foujols (foujols@ipsl.jussieu.fr) and Arnaud Caubel (caubel@lsce.saclay.cea.fr)
- OPA-LIM: The OPA system team: opatlod@lodyc.jussieu.fr
- LMDZ-ORCHIDEE: Laurent Fairhead (fairhead@lmd.jussieu.fr)

Introduction

The climate coupled configuration called IPSL_CM4 and used at Pierre-Simon Laplace Institute (IPSL) is composed of the oceanic component OPA, the sea ice component LIM, the atmospheric component LMDZ and the land surface component ORCHIDEE, coupled through OASIS coupler.

This document gives some information about this configuration and its adaptation to the PRISM system. The two first parts give a brief description about this coupled model and its components. The third part is more technical and stresses on the adaptation of the models sources. At last, the fourth part is a step-by-step guide to help the user to compile and run IPSL_CM4 with the PRISM environment.

I. The coupled configuration IPSL_CM4

IPSL_CM4 is a climate coupled model used at ISPL. It is adapted to the PRISM system according to the standard conventions required by the PRISM project. The atmospheric and oceanic components of IPSL_CM4 are coupled via a coupler called OASIS and developed at CERFACS.

The figure 1 explains how this coupled configuration works and details the exchanged fields between the atmosphere and the ocean. The arrows correspond to the exchanges between the components. The oceanic and atmospheric fields are exchanged through the coupler OASIS, while OPA and LIM or LMDZ and ORCHIDEE exchange their data directly. In the frames are noted the fields which are respectively sent by LMDZ-ORCHIDEE (in green) and OPA-LIM (in blue) to the coupler and then by the coupler to OPA-LIM (green) and LMDZ-ORCHIDEE (blue).

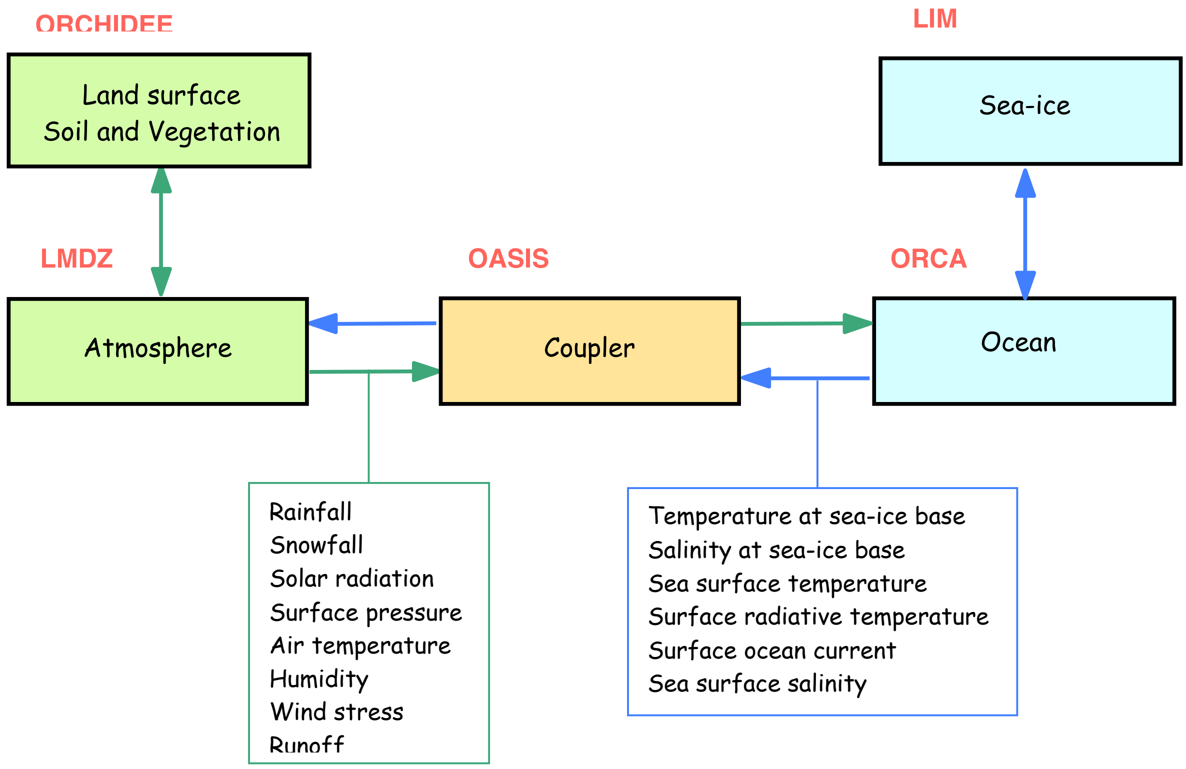


Figure 1: IPSL_CM4 coupled configuration available in the PRISM software

The meshes between the atmospheric and oceanic models are different. These differences reside in the size and the cells number of the mesh and in its organization. The coupler needs then to apply transformations and an interpolation method to reorder the mesh before exchanging each field. Moreover, the coupler verifies the coherence between the fields before and after being transformed and interpolated and allows the good functioning of the coupled simulation.

II. Models description

1. The oceanic component: OPA

➢ The OPA Oceanic General circulation model

The OPA system is developed at LODYC (Laboratoire d'Océanographie Dynamique…). It is a primitive equation model of both the regional and global ocean circulation. It is intended to be a flexible tool for studying ocean and its interactions with the others components of the earth climate system (atmosphere, sea-ice, biogeochemical tracers, ...) over a wide range of space and time scale. Prognostic variables are the three-dimensional velocity field and the thermohaline variables. The distribution of variables is a three dimensional Arakawa-C-type grid using prescribed *z*- or *s*-levels. Various physical choices are available to describe ocean physics, including a 1.5 turbulent closure for the vertical mixing. OPA is interfaced with a sea-ice model, a passive tracer model and, via the OASIS coupler, with several atmospheric general circulation models. In addition, it can be run on many different computers, including shared and distributed memory multiprocessor computers.

➢ The 3D dynamical core

The ocean is a fluid which can be described to a good approximation by the primitive equations, i.e. the Navier-Stokes equations along with a non-linear equation of state which couples the two active tracers (temperature and salinity) to the fluid velocity, plus the following additional assumptions made from scale considerations : spherical earth approximation; thin-shell approximation ; turbulent closure hypothesis ; Boussinesq hypothesis ; hydrostatic hypothesis; incompressibility hypothesis.

The primitive equations are written using a tensorial formalism so that any orthogonal curvilinear coordinate system which preserves the local vertical can be used.

The basic idea of numerical methods consists in discretizing differential equations on a three dimensional grid and computing the time evolution of each variable for each grid point. Ocean models are usually written in finite difference form. Such a method provides a legible computer code, easy to update, and is able to deal with the complex boundary conditions formed by the coastline geometry and the bottom topography.

The OPA reference manual describes in detail the ocean physics as taken in account by the model (explicitly or using sub-grid parametrization) as well as boundary conditions (surface, bottom, lateral), numerical schemes and computer implementation.

➢ The configurations used in IPSL_CM4: ORCA2_LIM and ORCA4_LIM

ORCA is the generic name given to global ocean configurations using the OPA System. Its specificity lies on the horizontal curvilinear mesh used to overcome the North Pole singularity found for geographical meshes.

The common geographical coordinate system has a singular point at the North Pole which cannot be easily treated in a global model without filtering. A solution consists in introducing an appropriate coordinate transformation which shifts the singular point on land [Madec and Imbard 1996, Murray 1996].

Space-time domain

• The horizontal resolution available through the standard configuration is ORCA2. It is based on a 2 degrees Mercator mesh, (i.e. variation of meridian scale factor as cosinus of the latitude). In the northern hemisphere the mesh has two poles so that the ratio of anisotropy is nearly one everywhere. The mean grid spacing is about 2/3 of the nominal value. An other resolution (ORCA4) is available. In the ORCA2 and ORCA4 configurations the meridional grid spacing is increased near the equator to improve the equatorial dynamics. Figures in pdf format of mesh and bathymetry can be found and downloaded at http://www.lodyc.jussieu.fr/opa/ORCA_mesh.html

• The vertical domain spreads from the surface to a depth of 5000m. There are 31 levels, with 10 levels in the top 100m. The vertical mesh is deduced from a mathematical function of z (Madec and Imbard 1996). The ocean surface corresponds to the w-level k=1, and the ocean bottom to the w-level k=31. The last T-level (k=31) is thus always in the ground. The depths of the vertical levels and the associated scale factors can be viewed at http://www.lodyc.jussieu.fr/opa/Docu_Free/ORCA_slide/Depth_e3.jpg

• The time step depends on the resolution. It is 1h36' for ORCA2 so that there are 15 time steps in one day.

Ocean Physics (for ORCA2_LIM in coupled configurations)

• Horizontal diffusion on momentum: the eddy viscosity coefficient depends on the geographical position. It is taken as 40000. m2/s, reduced in the equator regions (2000 m2/s) excepted near the western boundaries.

• Isopycnal diffusion on tracers: the diffusion acts along the isopycnal surfaces (neutral surface) with a eddy diffusivity coefficient of 2000. m2/s.

• Eddy induced velocity parameterization with a coefficient that depends on the growth rate of baroclinic instabilities (it usually varies from 15 m2/s to 3000m2/s).

• Lateral boundary conditions: zero fluxes of heat and salt and no-slip conditions are applied through lateral solid boundaries.

• Bottom boundary condition: zero fluxes of heat and salt are applied through the ocean bottom. The Beckmann diffusive bottom boundary layer parameterization is applied along continental slopes. A linear friction is applied on momentum.

• Convection: the vertical eddy viscosity and diffusivity coefficients are increased to 100 m^2/s in case of static instability.

• Ocean surface: a free surface formulation is used [Roullet, G., and G. Madec, 2000. J. Geophys. Res. 105].

• Forcing: the ocean receives heat, freshwater, and momentum fluxes from the atmosphere and/or the sea-ice. The sea-ice LIM component is used (see LIM documentation). The solar radiation penetrates the top meters of the ocean. The downward irradiance I(z) is formulated with two extinction coefficients [Paulson and Simpson, 1977], whose values correspond to a Type I water in Jerlov's classification (i.e the most transparent water).

-Dkey_orca_r4 or Dkey_orca_r2	resolution option (4 or 2 degrees)
-Dkey_ice_lln	+ use the LIM sea ice model
-Dkey_freesurf_cstvol	Free surface formulation
-Dkey_zdftke	tke 1.5 turbulent closure scheme
-Dkey_flxqsr	+ solar radiation penetration
-Dkey_trahdfiso	tracer diffusion along neutral surfaces
-Dkey_trahdfcoef2d	tracer latitude and longitude dependent coef.
-Dkey_dynhdfcoef3d	dynamic latitude and longitude dependent coef.
-Dkey_trahdfeiv	eddy induced advection
-Dkey_convevd	+ enhanced vertical diffusion
-Dkey_temdta	temperature data file (Levitus)
-Dkey_saldta	salinity data file (Levitus)
-Dkey_saldta_monthly	+ salinity data file (Levitus monthly)
-Dkey_diaznl	Diagnostics: zonal fluxes
-Dkey_diahth	Diagnostics: thermocline depth, and depth of the 20 degrees isotherm
-Dkey_diaeiv	Diagnostics: Eddy induced velocities
-Dkey_monotasking	monotasking option
-Dkey_coupled	coupled version - Boundary conditions: momentum, heat and fresh water fluxes (coupled)

The OPA OGCM (Ocean General Ciculation Model) references in papers and other publications can be found in the OPA reference manual (Madec et al. (1998)) or in http://www.lodyc.jussieu.fr/opa/.

2. The sea-ice component: LIM

LIM is the Louvain-la-Neuve sea ice model, which is a dynamic–thermodynamic model specifically designed for climate studies. This model is embedded in the ocean model ORCA and is the sea ice component of ORCALIM. Its evolution is managed at the LODYC. A brief description of the model is given here; further details can be found in the descriptions by Fichefet and Morales Maqueda (1997, 1999).

The physical processes governing the evolution of sea ice can be conceptually divided into two parts.

The first one concerns the thermodynamic growth or decay of the ice, which can be considered to depend only on the vertical response of the ice layer to the exchanges with the atmosphere and the ocean. The horizontal thermodynamic processes, such as the horizontal heat conduction through ice, can be safely neglected because of the much larger horizontal scales.

The representation of the vertical growth and decay is based on a 3-layer model (one layer for snow and two layers for ice), which takes into account sensible and latent heat storage in the snow-ice system and simulates the changes of snow and ice thickness in response to surface and bottom heat fluxes. The variation of ice compactness due to thermal processes is a function of the energy balance of the surface layer in the region occupied by leads.

The second part concerns the ice dynamics and transport, which can be considered at large scale as horizontal processes, the ice velocity being homogenous from the bottom to the top of the ice.

For calculating ice dynamics, sea ice is considered to behave as a viscous-plastic continuum. At the ice-ocean interface, the sensible heat flux is proportional to the temperature difference between the surface layer and its freezing point and to the friction. The ice-ocean stress is taken to be a quadratic function of the relative velocity between ice and the uppermost level of the ocean. Considering salt and freshwater exchanges between ice and ocean, brine is released to the ocean when ice is formed, while freshwater is transferred to the ocean when sea ice or snow melts.

This distinction between thermodynamics and dynamics is purely formal since they are intrinsically coupled. The ice growth is a function of the ice thickness and concentration (the smaller the ice thickness and concentration, the faster the ice growth and decay which in turns depends strongly on the advection pattern. Conversely, the motion of sea ice is to a great extent affected by the ice-thickness distribution which roughly controls the amount of stress the ice can sustain.

The sea ice model runs on the same grid than the ocean model.
Details about this model can be found in http://www.climate.be/tools/clio.html.

3. The atmospheric component: LMDZ

LMDZ is the General Circulation Model (GCM) developed at LMD (Laboratoire de Météorologie Dynamique) and used in the IPSL coupled model. One of its features is a modular writing with, in particular, a quite generic interface between its hydrodynamic part and its physical parameterizations. This has allowed the LMD to use a set of parameterizations developed "in-house", to adapt a set from ECMWF and to use other parameterizations suitable for the study of the atmosphere of some of the other planets of the solar system.

The hydrodynamic code solves the primitive equations by the finite difference method. These equations are discretized on an Arakawa-C-type grid. Vertically, the model uses hybrid coordinates.

Another feature of the model is that it is possible to use a "zoomed in grid" on a particular region thus increasing the local resolution of the model without increasing the number of points, and thus the computing time, used in the model.

In the IPSL coupled configuration, the model uses a regular grid horizontally but with two standard resolutions: a 'low resolution' of 72 longitudinal points, 46 latitudinal points and 19 vertical levels, and a 'medium resolution' of 96 points in longitude, 72 in latitude and 19 vertical levels.

In order to study climate, it is necessary to represent the numerous processes that are not taken into account in the hydrodynamic part of the program that solves the basic fundamental equations of fluid dynamics. These processes are either those that are not resolved because of the low resolutions used (such as turbulence) or are physical phenomena of major importance for the climate that need to be taken into account such as solar radiation and heat transfer.

The main physical parameterizations are listed below:

- The convection scheme was derived from the mass flux scheme developed by Tiedke (1989). It has now been replaced by a scheme developed by Kerry-Emmanuel
- The parameterization of clouds uses a scheme of cloudy water related to a static description of water distribution (LeTreut and Li, 1991).
- The radiative codes are identical to those used at the European Centre (ECMWF).
- The solar radiation uses Fouquart and Bonnel's scheme (1980) while the Morcrette's code is used for the long wave radiation.
- The boundary layer is represented with a model using a 2nd order closure. The land surface model is either a "bucket model" which simulates a homogeneous layer of 150 mm, the humidity being stored in some tank and lost by water streaming when the amount of water exceeds this value; or the ORCHIDEE land surface scheme developed at IPSL.
- The surface temperature is computed within the boundary layer scheme and uses an energy-balance condition.

More details concerning LMDZ can be found in http://www.lmd.jussieu.fr/~lmdz/manuelGCM/main/main.html (in French)

4. The land surface component: ORCHIDEE

ORCHIDEE (Organizing Carbon and Hydrology in Dynamic EcosystEms) is a terrestrial biosphere model which consists of the SVAT (Surface-Vegetation-Atmosphere Transfer scheme) SECHIBA (Schématisation des Echanges Hydriques à l'interface Biosphère-Atmosphère) and of a carbon module including the vegetation dynamics part of the LPJ (Lund-Postdam-Jena) dynamic global vegetation model. In other words, ORCHIDEE is a SVAT coupled to a biogeochemistry and a dynamic biogeography model.

As thus it explicitly simulates the phenomena of the terrestrial carbon cycle that are linked to vegetation and soil decomposition processes, but also changes in vegetation distributions in response to climate change as well as short-time scale interactions between the vegetated land surface and the atmosphere.

ORCHIDEE is based on three different models:

- The SVAT SECHIBA (Ducoudré et al., 1993) has been developed as a set of surface parameterizations for the LMD AGCM (LMDZ). It describes the short-timescale processes (of the order of a few minutes to hours) of energy and water exchanges between the atmosphere and the biosphere, and the soil water budget. There is no parameterization of photosynthesis. The time step of the hydrological module is of the order of 30 minutes. The diurnal cycle is fully described.
- The parameterizations of vegetation dynamics (fire, sapling establishment, light competition, tree mortality, and climatic criteria for the introduction or elimination of plant functional types) have been taken from the dynamic global vegetation model LPJ. The effective time step of the vegetation dynamics parameterizations is one year.
- The other processes such as photosynthesis, carbon allocation, litter decomposition, soil carbon dynamics, maintenance and growth respiration, and phenology form together a third model called STOMATE (Saclay Toulouse Orsay Model for the Analysis of Terrestrial Ecosystems). STOMATE essentially simulates the phenology and carbon dynamics of the terrestrial biosphere. Treating processes that can be described on time scales of a few days (time step is one day), STOMATE makes the link between the fast hydrologic and biophysical processes of SECHIBA and the slow processes of vegetation dynamics described by LPJ.

ORCHIDEE can be run on a stand-alone mode that is forced by climatologic or experimental data, or it can be run coupled to an AGCM.

More information on ORCHIDEE can be found in http://www.ipsl.jussieu.fr/~ssipsl/doc/doc_main.html.

III. Adaptation of IPSL_CM4 to the PRISM system

Adapting the model to the PRISM system means modifying the model sources in order to compile it and run it with the standard PRISM coupler Oasis3. To communicate with Oasis3, the model needs to interface with the PRISM System Model Interface libray PSMILe. More information about OASIS3 and PSMILe can be found in the OASIS3 user's guide (Valcke et al. (2003)).

1. Adaptation of OPA-LIM to the PRISM system

As described in the OPA reference manual, OPA is configured by cpp keys. The one key_coupled_prism must be activated to use this model with the PRISM system.

The routines which have been modified are: src/inicmo.F, src/stpcmo.F, include/flx.coupled.clio.h, include/flx.coupled.h, include/tau.coupled.clio.h, include/tau.coupled.h, include/common.coupled.h and include/parameter.coupled.h

Details:

Subroutine src/inicmo.F:
This routine is used to initialize the coupled mode communication for ocean exchange process identifiers and timestep information between the atmospheric model, the oceanic model and the coupler. In this routine are implemented specific calls to PSMILe for these different phases:

- Initialisation of PSMILe library (calls to prism_init_comp_proto and prism_get_localcomm_proto)
- Partition definition (call to prism_def_partition_proto): definition of a box partition in order to write coupling fields in 2D in the output files.
- I/O coupling field declaration (call to prism_def_var_proto)
- End of definition phase (call to prism_enddef_proto)

In the routines below, I/O coupling field receiving phase of PSMILe is implemented.

Include/flx.coupled.clio.h and flx.coupled.h:
These routines read fluxes from a coupled atmospheric model. So for each flux field, the model calls prism_get_proto.

Include/tau.coupled.clio.h and tau.coupled.h:
These routines read wind stress from a coupled atmospheric model. So the model calls prism_get_proto for each wind stress field.

Subroutine src/stpcmo.F:
This routine is used to send fields to the coupler at each coupling time step. So for each coupling field, the model calls prism_put_proto to implement the PSMILe phase of I/O coupling field sending. At the last time step of the simulation, the communication between the coupler and the model process is terminated by calling prism_terminate_proto.

The OPA's version adapted to the PRISM system is tagged ipsl_cm4_v1_8 on the IPSL server.

2. Adaptation of LMDZ-ORCHIDEE to the PRISM system

The adaptation of LMDZ to the PRISM software is controlled by the cpp key CPP_PSMILE. If this key is activated, the subroutine interface_surf.F90 then uses the module oasis.F90 in the directory phylmd. This module contains three subroutines which gather all the calls to the PSMILe phases:
In the subroutine inicma:

- Initialisation (calls to prism_init_comp_proto and prism_get_localcomm_proto)
- Partition definition (call to prism_def_partition_proto)
- I/O coupling field declaration (call to prism_def_var_proto)
- End of definition phase (call to prism_enddef_proto)

In the subroutine fromcpl:

- I/O coupling field receiving (call to prism_get_proto)

In the subroutine intocpl:

- I/O coupling field sending (call to prism_put_proto)
- Termination (call to prism_terminate_proto)

The LMDZ's version adapted to the PRISM system is tagged IPSL-CM4_PSMILE on the IPSL server.

3. PRISM architecture

As LMDZ-ORCHIDEE and OPA-LIM have their own sources architecture, the IPSL has made a tool which permits to pass from the initial sources architecture to the PRISM one. This script retrieves the components from their own CVS server and reorganizes the model sources in order to be compatible with the PRISM architecture.

This script is detailed in the annexe A.

IV. Step-by-step use of IPSL_CM4 coupled configuration

1. Retrieval of IPSL_CM4 from the PRISM CVS repository

IPSL_CM4 is available on the PRISM CVS repository. To retrieve it, you can type the following lines:

- ➢ Login on the server Bedano: please contact Veronika Gayler (gayler@dkrz.de) or Reiner Vogelsang (reiner@sgi.com) to get the password

cvs –d ":pserver :guest@bedano.cscs.ch:/users/cvs" login

- ➢ Checkout the code sources by entering a module name. This module name <module> can be:
  - IPSL_CM4 to check the code sources, the input data files and the PRISM tools
  - IPSL_CM4SRC to check the code sources and the PRISM tools
  - IPSL_CM4DATA to check the input data files only

cvs –d ":pserver :guest@bedano.cscs.ch:/users/cvs" checkout <module>

- ➢ Logout at the end of your session

cvs –d ":pserver :guest@bedano.cscs.ch:/users/cvs" logout

The modules are also available on the web site: http://prism-cvs.cscs.ch/cgi-bin/cvsweb.cgi : please contact Veronika Gayler (gayler@dkrz.de) or Reiner Vogelsang (reiner@sgi.com) to get login and password.

2. Compilation of the IPSL_CM4 components

• Generation of Makefiles:

The generic Makefiles are available from Bedano for each model. But it is also possible to create them thanks to a tool in prism/util/compile/Append_dependencies. This tool uses a base Makefile_1 that is delivered in the model sources and appends the dependencies of the routines. To create the Makefiles, Append_dependencies requires two parameters: the model name and the name of the sources directory for which the Makefile needs to be created.

• Generation of compile scripts:

The compile scripts need to be created for each component of the coupled configuration. They are generated thanks to the script Create_COMP_models.frm in the directory prism/util/compile/frames. This tool provides a help function by typing Create_COMP_models.frm --help). It accepts seven command line parameters, which must be given in the right order:

- $1=model name
- $2=""/"MPI1"/"NONE": message passing library (MPI-2 is the default)
- $3=""/"-": standard output of the compilation directed to a file / the screen
- $4=""/"-"/"+": error output of the compilation directed to a file / the screen / the standard output file
- $5=""/"node name": node name of the machine where the models are compiled (by default, it is the machine on which the scripts are created, but it is also possible to create the scripts from a different machine. In that case, it is necessary to precise the node name of the compile machine. This can be used if the compile server has not installed the right 'm4' release.)
- $6="version acronym": acronym for the model version. It is used for the executable name and permits to differentiate the versions of a same model
- $7="list of participating models": indicates the name of the models that are used in the configuration

A compile script is needed too to compile the library whose the models are depending. It is then generated thanks to a script Create_COMP_libs.frm in prism/util/compile/frames. It requires 3 parameters: the standard output, the error output and the node name.

Bellow are the lines that can be typed to create the scripts of the IPSL models:

cd prism/util/compile/frames
./Create_COMP_libs.frm "" "" ""
./Create_COMP_models.frm oasis3 "" "" "" "" " " ""
./Create_COMP_models.frm lmdz "" "" "" "" I01 "lmdz orchidee opa lim"
./Create_COMP_models.frm orchidee "" "" "" "" I01 "lmdz orchidee opa lim"
./Create_COMP_models.frm opa "" "" "" "" I01 "lmdz orchidee opa lim"
./Create_COMP_models.frm lim "" "" "" "" I01 "lmdz orchidee opa lim"

• Compilation:

The compile scripts are created in each model directory. They must be run then to begin the compilation. In the IPSL coupled configuration, LIM is defined as a 'sub-model' of OPA and ORCHIDEE as a 'sub-model' of LMDZ. It is not then necessary to compile manually LIM and ORCHIDEE since their compilation will be automatically run by the compilation of OPA and LMDZ.

Below are the lines which can be typed to run the compile scripts which have been created according to the previous commands:

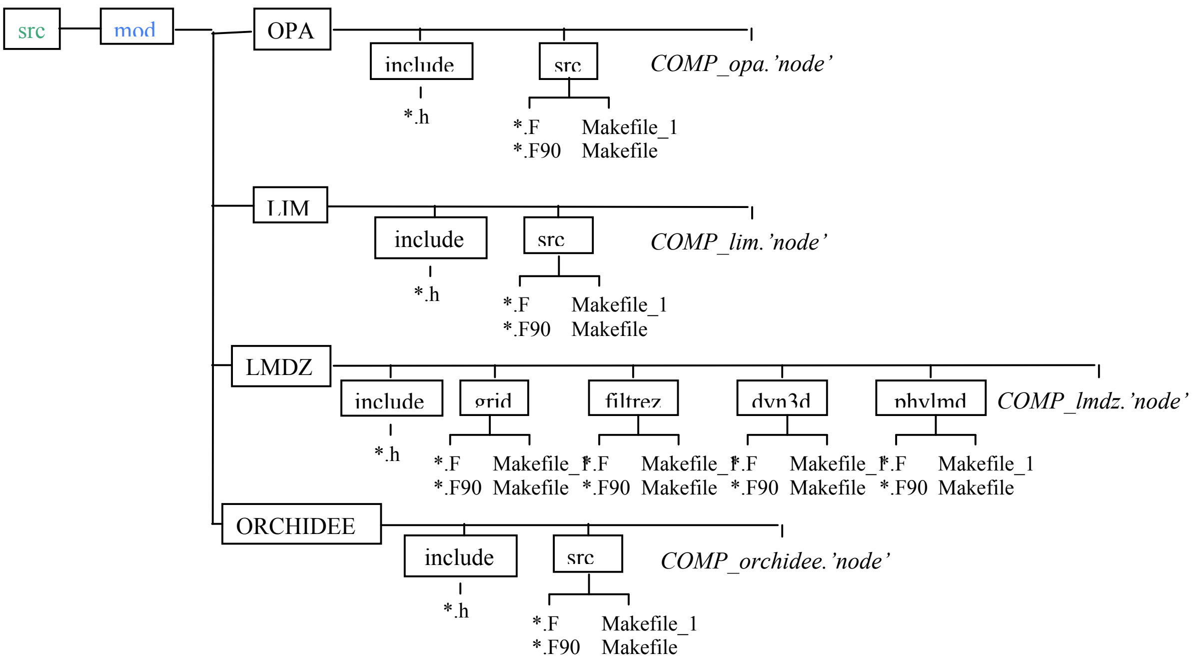
./COMP_oasis3_MPI2.'node'
./COMP_opa_I01.'node'
./COMP_lmdz_I01.'node'

'node' is the name of the compile host. At this time, IPSL_CM4 is adapted to the PRISM system in order to be compiled and run on NEC SX5 at IDRIS, NEC SX6 at CEA and NEC SX6 at DKRZ. This phase is still in progress to adapt it on VPP and SGI. Go to this web page: http://prism.dkrz.de/Workpackages/WP3i/Deliverables/Table_prism_sites.html to know which machines and which coupled configurations are able to run with the PRISM system.

IPSL_CM4 is available at two resolutions: the 'low' resolution, which is orca4 (92x76x31) for OPA and lmd7245 (72x46x19) for LMDZ and the 'medium' resolution, which is orca2 (182x149x31) for OPA and lmd9671 (96x72x19) for LMDZ. By default, the compile script is generated with the low resolution. So the compile scripts *COMP_opa_I01.'node', COMP_lim_I01.'node'* and *COMP_lmdz_I01.'node'* need to be edited to change the resolution.

More details about the PRISM standard compiling environment can be read in the PRISM SCE guide (Legutke and Gayler (2004)).

Below is the arborescence of the models sources directory for each component of IPSL_CM4.



3. Running of IPSL_CM4

The run script is generated by the script Create_TASKS.frm in the directory prism/util/running/frames. It requires 2 parameters: the name of the coupled configuration=ipsl_cm4 and the version of the experiment (type Create_TASKS.frm --help for details) as it is below:

./Create_TASKS.frm ipsl_cm4 I01

A setup file *setup_ipsl_cm4_I01* is then created in the directory prism/util/running/frames/setup. This one contains all parameters of the coupled model experiment that can be specified by the user. It must be edited to change these parameters according to the design of the experiment.

Then Create_TASKS.frm must be run a second time with the same parameters to create the run script, which will be sent automatically to the compute server.

At last, the run script of the experiment can be submitted on the compute server.

The IPSL usually runs the IPSL_CM4 coupled configuration on NEC SX5 at IDRIS (compilation on Rhodes, execution on Uqbar and archiving on Gaya) and NEC SX6 at CEA (Mercure), which are then adapted to the PRISM standard running environment. The setup file *setup_ipsl_cm4_I01* which is created by the first run of Create_TASKS.frm, as described above, is then different from one machine to another. Below is described the setup file as it could be modified to run the job on these two machines:

Running IPSL_CM4 on NEC SX6 at CEA (mercure): *setup_ipsl_cm4_I01*

As Mercure is a machine on which the compilation, the execution and the archiving is possible, the completion of the setup file is then very simple. If the models sources are on /scratchdir/p86demo/IPSL_CM4_2_2/prism:

```
#--  archiving_host - Node name of an archiving machine
archiving_host="" : the files are stored on mercure

#-- home: Permanent file system for the scripts on the computing host
home=/scratchdir/p86demo/IPSL_CM4_2_2/prism/experiments

#-- data: Root directory of the short term data server. Model input and output
#         will be read from/written to this file system of the computing host
data=/scratchdir/p86demo/IPSL_CM4_2_2/prism/experiments
or data=/work/p86demo/IPSL_CM4_2_2/prism/experiments

#-- archive: Root directory of the long term output data archive. - Either a
#            filesystem of the computing host or of a remote archiving host.
#            If ${archive} differs from ${data} model output will be saved in
#            ${archive} and removed from ${data}.
archive=/scratchdir/p86demo/IPSL_CM4_2_2/prism/experiments
or archive=/dmnfs/p86demo/IPSL_CM4_2_2/prism/experiments

#-- archive_in: Root directory of the long term input data archive. It needs
#              to reside on the same machine as the output archive. This
```

#   archive is intended for input data that is needed with several experiments
archive_in=/scratchdir/p86demo/IPSL_CM4_2_2/prism/data
or archive_in=/dmnfs/p86demo/IPSL_CM4_2_2/prism/data

#-- work: Root directory for the temporary working directory
work=/scratchdir/p86demo/IPSL_CM4_2_2/prism/experiments
or work=${TMPDIR}

#-- Compilation
#   compile_server: Node name of the compile-server
#   compile_path: Directory where the executables are stored on the compile-server
compile_server=mercure
compile_path=/scratchdir/p86demo/IPSL_CM4_2_2/prism/SX/bin

Running IPSL_CM4 on NEC SX5 at IDRIS (uqbar): ***setup_ipsl_cm4_I01***

The NEC SX5 is composed of three servers: the compiling server Rhodes, the running server Uqbar and the archiving server Gaya. The completion of the setup file is then a little bit more complicated. Here is a description of what may be written if the models sources are on Rhodes: /home_bis/rech/ces/rces844/IPSL_CM4_2_2/prism:

#-- archiving_host - Node name of an archiving machine
archiving_host=gaya

#-- home: Permanent file system for the scripts on the computing host:
#   this is the home on uqbar (there must be only one directory name,
#   which will be created automatically)
home=/u/rech/ces/rces844/IPSL_CM4_2_2

#-- data: Root directory of the short term data server. Model input and output
#   will be read from/written to this file system of the computing host
#   the workdir on uqbar seems to be the best directory to avoid the disk quota exceeded.
#   (there must be just one directory)
data=/workdir/rech/ces/rces844/IPSL_CM4_2_2

#-- archive: Root directory of the long term output data archive. - Either a
#   filesystem of the computing host or of a remote archiving host.
#   If ${archive} differs from ${data} model output will be saved in
# ${archive} and removed from ${data}.
#   This is a directory on Gaya:
archive=/u/rech/ces/rces844/IPSL_CM4_2_2/prism/experiments

#-- archive_in: Root directory of the long term input data archive. It needs
#   to reside on the same machine as the output archive. This
#   archive is intended for input data that is needed with several experiments
#   This is a directory on Gaya:
archive_in=/u/rech/ces/rces844/IPSL_CM4_2_2/prism/data

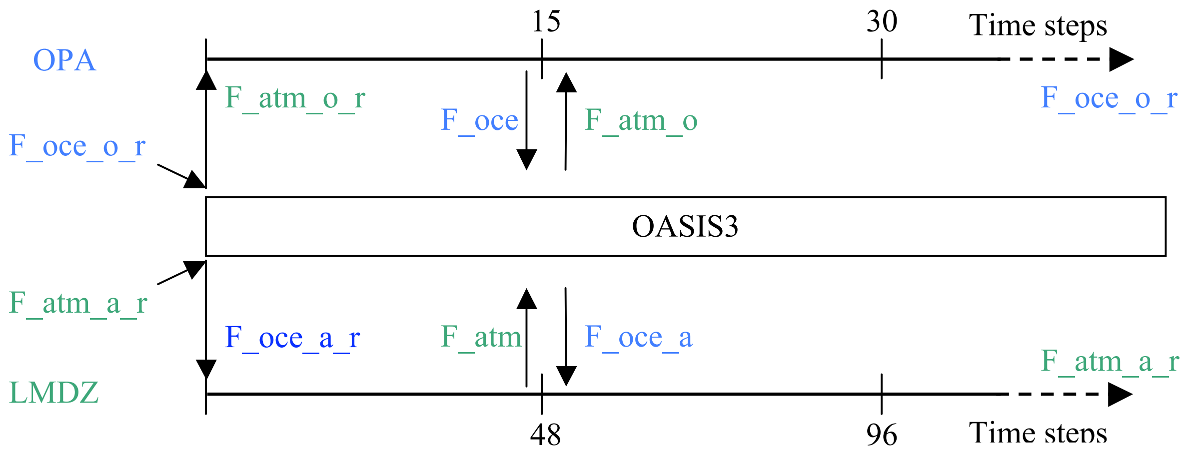
#-- work: Root directory for the temporary working directory

#   The tmpdir directory has to be used to avoid the disk quota exceeded.
work=${TMPDIR}

#-- Compilation
#   compile_server: Node name of the compile-server
#   compile_path: Directory where the executables are stored on the compile-server
compile_server=rhodes
compile_path=/home_b/rech/ces/rces844/IPSL_CM4_2_2/prism/SX/bin

For more details about the PRISM standard running environment, see the PRISM SRE guide (Gayler and Legutke (2004)).

The figure 2 represents the data flows at LMDZ-OASIS-OPA level. The flows are exchanged every day, which corresponds to the 15th time step for OPA and the 48th time step for LMDZ. At the first time step, the fields are read from restart files (F_atm_a_r and F_oce_o_r) by the coupler and are then sent to the models in F_atm_o_r and F_oce_a_r. Then at the first coupling time step, OPA (resp. LMDZ) sends its fields F_oce (resp. F_atm) which, after being interpolated by the coupler, are received by LMDZ (resp. OPA) as F_oce_a (resp. F_atm_o). And at the last time step, each model writes its fields in a restart file (F_oce_o_r for OPA and F_atm_a_r for LMDZ), which is used at the beginning of the next simulation.



F_oce (fields sent by OPA)=SOSSTSSW, SOICECOV, SOICEALW, SOICTEMW F_oce_a (fields received by LMDZ)= SISUTESW, SIICECOV, SIICEALW, SIICTEMW F_atm (fields sent by LMDZ)= COSHFICE, COSHFOCE, CONSFICE, CONSFOCE, CODFLXDT, COTFSICE, COTFSOCE, COTOLPSU, COTOSPSU, CORUNCOA, CORIVFLU, COCALVIN, COTAUXXU, COTAUYYU, COTAUZZU, COTAUXXV, COTAUYYV, COTAUZZV F_atm_o (fields received by OPA)= SOSHFLDO, SOSHFOCE, SONSFLDO, SONSFOCE, SODFLXDT, SOTFSICE, SOTFSOCE, SOTOLPSU, SOTOSPSU, SORUNCOA, SORIVFLU, SOCALVIN, SOTAUXXU, SOTAUYYU, SOTAUZZU, SOTAUXXV, SOTAUYYV, SOTAUZZV F_atm_a_r (At the first time step, the atmospheric fields are sent from the restart file F_atm_a_r to the coupler. After being interpolated, they are sent to OPA in F_atm_o_r) F_oce_o_r (At the first time step, the oceanic fields are sent from the restart file F_oce_o_r to the coupler. After being interpolated, they are sent to LMDZ in F_oce_a_r)

Figure 2: Mechanism of the exchanges between the ocean and the atmosphere during a simulation

The results of the IPSL_CM4 demonstration runs are described in http://www.ipsl.jussieu.fr/~prisipsl/Docs/template_individual_report_IPSL180804.pdf.

References and Bibliography

- Ducoudré, N., Laval, K. and Perrier, A. (1993) 'SECHIBA, a new set of parameterizations of the hydrologic exchanges at the land/atmosphere interface within the LMD atmospheric general circulation model'. Journal of Climate 2(9):248-273.
- Fichefet, T., Morales Maqueda, M.A., 1997. Sensitivity of a global sea ice model to the treatment of ice thermodynamics and dynamics. Journal of Geophysical Research 102 (C6), 12609–12646.
- Fichefet, T., Morales Maqueda, M.A., 1999. Modelling the influence of snow accumulation and snow–ice formation on the seasonal cycle of the Antarctic sea-ice cover. Climate Dynamics 15, 251–268.
- Gayler, V. and Legutke, S., 2004: 'The PRISM Standard Running Environment Guide', PRISM Report Series, no. 4, nn pp. (http://prism.enes.org/Results/Documents/PRISM_Reports/Report4_pdf)
- [7] Guilyardi, E., Polcher, J., Fichefet, T., Madec, G., Marti, O., and Planton, S., Revisiting physical interfaces in Earth system models, 2004.
- Legutke, S. and Gayler, V., 2004: 'The PRISM Standard Compilation Environment Guide', PRISM Report Series, no. 3, nn pp. (http://prism.enes.org/Results/Documents/PRISM_Reports/Report5_pdf)
- Madec, G., P. Delecluse, M. Imbard, and C. Lévy, 1998: OPA 8.1 Ocean General Circulation Model reference manual. *Note du Pôle de modélisation*, Institut Pierre-Simon Laplace (IPSL), France, No 11, 91pp.
- Valcke S., Caubel A., Declat D., Terray L. (2003), OASIS3 Ocean Atmosphere Sea Ice Soil User's Guide. PRISM Report Series, n° 2, 54pp. http://prism.enes.org/Results/Documents/PRISM_Reports/Report2_oasis3_UserGuide.pdf

ANNEXE A:

Script for adapting the architecture of OPA to the PRISM architecture:

```
#!/bin/csh

# The script is run like this:
# 'script_prism_opa -r tag_opa'
# or 'script_prism_opa' (by default: ipsl_cm4_v1_8)

### initialization
### ---------------
set tag_opa="-r ipsl_cm4_v1_8"

### Options
### ---------
if ($#argv > 0) then
  switch ($1:q)
  case -r:
    set tag_opa="-r $2" ; shift ; shift
  endsw
endif

echo 'tag opa=' $tag_opa

### Model sources retieval
### ---------------------------
set HOME=`pwd`
cd $HOME
mkdir OPA_PRISM
cd OPA_PRISM

mkdir src
mkdir src/mod
mkdir src/lib
mkdir src/mod/opa
mkdir src/mod/lim
mkdir src/mod/trc
mkdir src/mod/opa/src
mkdir src/mod/opa/include
mkdir src/mod/lim/src
mkdir src/mod/lim/include
mkdir src/mod/trc/src_trc
```

```
mkdir src/mod/trc/src_sms
mkdir src/mod/trc/include

### ORCALIM
### --------------
cvs -d :pserver:opa@cvs.ipsl.jussieu.fr:/home/opalod/CVSROOT co $tag_opa OPA

if (-r $HOME/OPA_PRISM/OPA/SRC_ORCA) then
  cp -f $HOME/OPA_PRISM/OPA/SRC_ORCA/*.[Ff]* src/mod/opa/src/.
  cp -f $HOME/OPA_PRISM/OPA/SRC_ORCA/*.h    src/mod/opa/include/.
endif
if (-r $HOME/OPA_PRISM/OPA/SRC_UCL) then
  cp -f $HOME/OPA_PRISM/OPA/SRC_UCL/*.[Ff]*  src/mod/lim/src/.
  cp -f $HOME/OPA_PRISM/OPA/SRC_UCL/*.h     src/mod/lim/include/.
endif
if (-r $HOME/OPA_PRISM/OPA/SRC_TRC) then
  cp -f $HOME/OPA_PRISM/OPA/SRC_TRC/*.[Ff]*  src/mod/trc/src_trc/.
  cp -f $HOME/OPA_PRISM/OPA/SRC_TRC/*.h     src/mod/trc/include/.
  cp -f $HOME/OPA_PRISM/OPA/SRC_TRC/trc_sms/*.[Ff]* src/mod/trc/src_sms/.
  cp -f $HOME/OPA_PRISM/OPA/SRC_TRC/trc_sms/*.h    src/mod/trc/include/.
endif

rm -rf $HOME/OPA_PRISM/OPA
```

Script for adapting the architecture of LMDZ to the PRISM architecture:

```
#!/bin/csh

# The script is run like this:
# 'script_prism_lmdz -r tag_lmdz -i tag_orchidee'
# or 'script_prism_lmdz' (by default: tag_lmdz=IPSL-CM4_PSMILE, tag_orchidee=orchidee_1_3 )

### initialization
### ---------------
set tag_lmdz="-r IPSL-CM4_PSMILE"
set tag_orchidee="-r orchidee_1_3"

### Options
### ---------
if ($#argv > 0) then
   switch ($1:q)
   case -r:
      set tag_lmdz="-r $2" ; shift ; shift
   case -i:
      set tag_orchidee="-r $2" ; shift ; shift
   endsw
endif

echo 'tag lmdz='$tag_lmdz
echo 'tag orchidee='$tag_orchidee

### Model sources retrieval
```

```
### ----------------------------
set HOME=`pwd`
cd $HOME
mkdir LMDZ_PRISM
cd LMDZ_PRISM

mkdir src
mkdir src/mod
mkdir src/mod/lmdz
mkdir src/mod/orchidee
mkdir src/mod/lmdz/bibio
mkdir src/mod/lmdz/phylmd
mkdir src/mod/lmdz/dyn3d
mkdir src/mod/lmdz/filtrez
mkdir src/mod/lmdz/grid
mkdir src/mod/lmdz/include
mkdir src/mod/orchidee/src_parameters
mkdir src/mod/orchidee/src_sechiba
mkdir src/mod/orchidee/src_stomate

### LMDZ
### --------
cvs -d :pserver:lmdzbrowse@piaf.lmd.jussieu.fr:/users/lmdz/cvsroot co $tag_lmdz LMDZ.3.3

cp -f $HOME/LMDZ_PRISM/LMDZ.3.3/libf/bibio/*.[Ff]* src/mod/lmdz/bibio/.
cp -f $HOME/LMDZ_PRISM/LMDZ.3.3/libf/phylmd/*.[Ff]* src/mod/lmdz/phylmd/.
cp -f $HOME/LMDZ_PRISM/LMDZ.3.3/libf/phylmd/oasis.psmile src/mod/lmdz/phylmd/oasis.F90
if (-f oasis.F) then
   rm -f src/mod/lmdz/phylmd/oasis.F
endif
cp -f $HOME/LMDZ_PRISM/LMDZ.3.3/libf/dyn3d/*.[Ff]* src/mod/lmdz/dyn3d/.
foreach fili (`ls src/mod/lmdz/dyn3d/*.F`)
   if ($fili != src/mod/lmdz/dyn3d/gcm.F && $fili != src/mod/lmdz/dyn3d/create_etat0_limit.F) then
      set test=` (  head $fili | grep '      PROGRAM' ) `
      if ( "$test" != "" ) then
         rm -f $fili
      endif
   endif
end
cp -f $HOME/LMDZ_PRISM/LMDZ.3.3/libf/filtrez/*.[Ff]* src/mod/lmdz/filtrez/.
cp -r $HOME/LMDZ_PRISM/LMDZ.3.3/libf/grid/dimension src/mod/lmdz/grid/.
cp -f $HOME/LMDZ_PRISM/LMDZ.3.3/libf/*/*.h src/mod/lmdz/include/.
cp -f $HOME/LMDZ_PRISM/LMDZ.3.3/libf/*/*.inc src/mod/lmdz/include/.

rm -rf $HOME/LMDZ_PRISM/LMDZ.3.3


### ORCHIDEE
### ---------------

cvs -d :pserver:sechiba@cvs.ipsl.jussieu.fr:/home/ssipsl/CVSREP co $tag_orchidee ORCHIDEE

cp -f $HOME/LMDZ_PRISM/ORCHIDEE/src_parameters/*.[Ff]* src/mod/orchidee/src_parameters/.
cp -f $HOME/LMDZ_PRISM/ORCHIDEE/src_sechiba/*.[Ff]* src/mod/orchidee/src_sechiba/.
cp -f $HOME/LMDZ_PRISM/ORCHIDEE/src_stomate/*.[Ff]* src/mod/orchidee/src_stomate/.
```

```
rm -rf $HOME/LMDZ_PRISM/ORCHIDEE
```