ORCHIDEE Training course

Code management, installation, simulation, documentation

24 November 2016, IDRIS

Josefine Ghattas IPSL

Finding information

Wiki and web site, Email lists
Developer's and user's meeting
Scientific documentation

Code management through SVN

What is SVN ?, Different versions of ORCHIDEE
How to know which version I use ?

Install and compile

How to install, predefined versions with ORCHIDEE
Install a branch or private version for off-line
TGCC and IDRIS, How to compile

Simulation

Input and output files
Parallel mode
Configurations using libIGCM

Coding Guidelines

Finding information
Wiki and web site

Official web site

http://labex.ipsl.fr/orchidee

ORCHIDEE wiki

On the wiki you find useful information about on-going developments and help to use the model. Lots of information is found in HowTo section

http://forge.ipsl.jussieu.fr/orchidee/wiki

You need a **“login forge” to be used on the wiki**. This login is needed to see the full content of the wiki, to edit the texte and also to see the SVN repository on the web interface. Write to orchidee-help to get a login.

Finding information
Email lists @ipsl.jussieu.fr

All ORCHIDEE user's are invited to subscribe to:

orchidee-dev	Discussion and information about ORCHIDEE
platform-users	Ask and answer questions about libIGCM Information about IPSL tools

2 email addresses for contact:

orchidee-help	For technical questions
orchidee-project	To contact the ORCHIDEE project team

See how to subscribe :
http://forge.ipsl.jussieu.fr/orchidee/wiki/GroupActivities/Contact

Finding information
“Developer's meeting”

All users and developers of ORCHIDEE are welcome to ORCHIDEE developer's meetings organized about every 2 months. These meetings consist in a presentation of a specific topic followed by discussions and questions. Meeting place in Jussieu but often a videoconference is set up.

See reports and presentations here :

http://forge.ipsl.jussieu.fr/orchidee/wiki/GroupActivities/Meetings/Developer

Information about these meetings are done at orchidee-dev email list.

Finding information
Scientific documentation

- A scientific documentation is integrated in the code.

- It is generated using Doxygen, the documentation is generated in both pdf and html format.

- see here latest compiled version of the documentation :

https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation

Code management through SVN
What is SVN ?

- All different versions of ORCHIDEE are managed through SVN.

- SVN, also called subversion, is **a versioning system** that helps to keep track of different versions. With subversion it is always possible to extract all previous versions of a specific directory.

- Main repository : svn://forge.ipsl.jussieu.fr/orchidee

Code management through SVN
Different versions of ORCHIDEE

Trunk : The main version of ORCHIDEE. Each modification on the trunk has a specific revision number.

- ***Each revision is not fully validated***
- ***Some revision numbers are more tested, see wiki:***

https://forge.ipsl.jussieu.fr/orchidee/wiki/ReferenceSimulations

Branch : a version under development by a group of people. Different branches exist. A branch is a copy of the trunk that is modified. The goal is to integrate developements on the branch in the trunk.

Perso : each user can have one or several personal versions on svn. A personal version is a copy from the trunk, tag or branch at a specific moment. The user is responsible of his own version. ***Updates, bug corrections and re-integration in the trunk is not automatic and might be difficult.***

Code management through SVN
How to know which version I use ?

- When you communicate with other people, it is necessary to know exactly which version you use.

- Both <u>version</u> and <u>revision number</u> are needed.

```
> cd modipsl/modeles/ORCHIDEE
> svn info

Path: .
URL: svn://forge.ipsl.jussieu.fr/orchidee/trunk/ORCHIDEE
Repository Root: svn://forge.ipsl.jussieu.fr/orchidee
Repository UUID: f489ceea-5127-0410-b15c-c4a6149ed9a7
Revision: 3917
Node Kind: directory
Schedule: normal
Last Changed Author: fabienne.maignan
Last Changed Rev: 3917
Last Changed Date: 2016-11-18 14:34:53 +0100 (Fri, 18 Nov 2016)
```

Version : trunk/ORCHIDEE

Revision number : 1172

Code management through SVN
Login SVN

- trunk and tags are public versions. No personal login is needed. Login public : sechiba

- **personal login : firstname.lastname**

All user/developer can ask for a personal login and a personal folder SVN. This login is used for accessing branches and personal folders. **A personal folder is only read and writeable for the owner.**

A personal folder is useful when starting a long development, for example for a phd or when a modified version of ORCHIDEE is used for an article. A personal folder helps for the

    - traceability of code
    - back up

Install and compile
How to install using modipsl

see also hands-on exercises this afternoon

- modipsl is a tool used to install and compile predefined configurations, for example ORCHIDEE offline or ORCHIDEE coupled to LMDZ

- modipsl contains scripts for extraction of predefined configurations, creation of makefiles, creation of job and some more. modipsl is also a empty file tree that will receive the models and tools.

- use **./model *config*** to download a specific configuration

```
> svn co http://forge.ipsl.jussieu.fr/igcmg/svn/modipsl/trunk modipsl
> cd modipsl/util
> ./model -h                    # list predefined configurations
> ./model config                # extract a predefined configuration
```

Install and compile
Predefined configurations with ORCHIDEE

ORCHIDEE_trunk : the latest version on ORCHIDEE/trunk for offline use. For new developments this is often the version to use. Note :

- You get the latest version of the trunk of ORCHIDEE : not always fully tested
- The trunk changes often : if you extract today and again next week there might be differences in the source code due to evolution of the trunk
- You can use this target to extract a branch, see further slides

ORCHIDEE_AR5 : off-line version used for reference CMIP5 simulations. Contains only source code without scripts for running.

Install and compile
Predefined configurations with ORCHIDEE

LMDZOR_v6 : configuration under development. To be use for coupling the latest version of the trunk ORCHIDEE with LMDZ(atmospheric model). Currently corrsponding to IPSLCM6.0.5. Follow changes here:
http://forge.ipsl.jussieu.fr/igcmg_doc/wiki/DocHconfigBlmdzorv6

LMDZOR_v5 : old stable version with ORCHIDEE and LMDZ version corresponding to AR5 simulations (CMIP5).

IPSLCM6.0.5-LR : coupled configuration containing NEMO(ocean model), LMDZ and ORCHIDEE. Latest official version.

IPSLCM6.0.6-LR : Under construction. To be announced next week hopefully..

Install and compile
Install a branch or personal version

You can modify the version of the ORCHIDEE source before extraction of a configuration. In **modipsl/util/mod.def**, modify line:

```
#-C- ORCHIDEE_trunk  trunk/ORCHIDEE        HEAD  14  ORCHIDEE  modeles
```

into

```
#-C- ORCHIDEE_trunk  trunk/ORCHIDEE        2724   14  ORCHIDEE  modeles
```

or

```
#-C- ORCHIDEE_trunk  branches/xxx/yyy      HEAD  14  ORCHIDEE  modeles
```

For exemple:

```
#-C- ORCHIDEE_trunk     branches/ORCHIDEE-MICT/ORCHIDEE    HEAD   14 ORCHIDEE modeles
```

Install and compile
Different platforms

- Compiling options of ORCHIDEE is preconfigured at following platforms :

obelix at LSCE
curie at TGCC
ada at IDRIS
ciclad and climserv at IPSL

- Compiling at other machines need more time for installing pre-request (compiler, netcdf,..)

- The script modipsl/util/ins_make creates makefiles for the target machine.

Install and compile
TGCC and IDRIS

To work on the TGCC or IDRIS computing centres you need:

- **a login connected to an existing group/project**, discuss with your supervisor/colleagues

- **CPU resources** in this group, yearly demand, discuss with your supervisor/colleagues

- **knowledge about the environment** at these centres : different machines, file systems, etc..

- **knowledge about how to calculate CPU consumption**

Install and compile
obelix

ORCHIDEE_trunk and LMDZOR_v6 can be used at obelix.

- **Note 1.** The Job PBS directive for the core distribution needs to be changed if using more than 8 MPI. For example if using 32MPI, the headers will be

  #PBS -l nodes=1:ppn=32

  but needs to be changed to

  #PBS -l nodes=4:ppn=8

- **Note 2.** When running LMDZ at obelix, change in run.def to:

  use_filtre_fft=n

- **Note 3.** Hybrid MPI-OpenMP mode has not been tested

Install and compile
ciclad and climserv

ORCHIDEE_trunk and LMDZOR_v6 can be used in standard configuration with modipsl and libIGCM on ciclad and climserv. See specific information here :
http://forge.ipsl.jussieu.fr/igcmg_doc/wiki/DocBenvDipslAciclad

- **Note 1.** Modification in default compilation of XIOS needed to using netcdf sequential, add **--netcdf_lib netcdf4_seq** in modipsl/config/AA_make on the line make_xios as follow:

```
(cd  ../../modeles/XIOS ; ./make_xios --netcdf_lib netcdf4_seq
--prod --arch ${FCM_ARCH} --job 8 ; cp bin/xios_server.exe ../../bin/. ; )
```

- **Note 2.** Change PBS directive as for obelix, for 32 MPI set

```
#PBS -l nodes=4:ppn=8
```

Install and compile
Compiling ORCHIDEE

- **Compiling is done with a main Makefile in modipsl/config/ORCHIDEE_OL** directory (or LMDZOR_vx)
- The main Makefile launch compilation of XIOS, IOIPSL and finally ORCHIDEE
- Inside the main Makefile, the script **makeorchidee_fcm is launched to compile ORCHIDEE**. This compile script is based on the tool FCM.
- Dependencies between modules are determined automatically. No modifications are needed if you add a module in one of the existing src_ directories.
- **Specific platform dependent compile options are set in modipsl/modeles/ORCHIDEE/arch/** directory. 2 files per platform: arch-ifort_LSCE.fcm and arch-ifort_LSC.path

Install and compile
makeorchidee_fcm

makeorchidee_fcm: a script launched by the main Makefile in config/ORCHIDE_OL

Example 1 : compile at curie(TGCC) for MPI parallel run mode

```
./makeorchidee_fcm -parallel mpi -arch X64_CURIE -driver
```

Example 2 : compile at ada(IDRIS) for MPI-OpenMP parallel run mode

```
./makeorchidee_fcm -parallel mpi_omp -arch X64_ADA -driver
```

Example 3 : compile at obelix(LSCE)

```
./makeorchidee_fcm -arch ifort_LSCE -driver
```

Example 4 : compile using gfortran compiler for sequential run mode
First make sure that the files arch/gfortran.fcm and arch/gfortran.path are suitable for your environement especially the path to netcdf library.

```
./makeorchidee_fcm -parallel seq -arch gfortran -driver
```

Example 5 : clean files created during previous compilation

```
./makeorchidee_fcm -clean
```

Install and compile
How to compile ?

Compiling off-line driver for the trunk version

```
> cd modipsl/config/ORCHIDEE_OL
> gmake
```

or to compile without XIOS (only IOIPSL)

```
> gmake without_xios
```

After successful compiling,
executables are found in :

modipsl/bin/

Install and compile
Steps to follow for installion at a new platform

- 1. Install modipsl and the configuration ORCHIDEE_trunk

- 2. Modify compile options in following files:
  - modipsl/util/**AA_make.gdef** (used for compilation of IOIPSL)
  - modipsl/modeles/**XIOS/arch/arch-*yourtarget*.[fcm/path/env]**
  - modipsl/modeles/**ORCHIDEE/arch/arch-*yourtarget*.[fcm/path]**

Note: the variable FCM_ARCH in AA_make.gdef is the name of the arch files in ORCHIDEE/arch and XIOS/arch.

- 3. Recreate makefiles with target chosen above and compile as usual

```
cd modipsl/util; ./ins_make -t yourtarget
```

=> Requirements are MPI and netCDF4 library.
Additional requirements: parallel library NetCDF4/HDF5

=> It is possible to compile and use without XIOS and without MPI.

Simulation
Input files to run in off-line mode

Executables : orchidee_ol + xios_server.exe (optional)

Parameter files : run.def : text file with run options
iodef.xml, context_orchidee.xml, field_def_orchidee.xml, file_def_orchidee.xml (optionals)

Boundary conditions

forcing_file.nc : climate forcing variables
PFTmap.nc : vegetation map (optional)

Initial state files - <u>if no restart files</u>

soils_param.nc, soils_param_usdatop.nc(optional)
PFTmap.nc : vegetation map
routing.nc : river routing (optional)
floodplains.nc and cartepente2d_15min.nc (optional)
reftemp.nc : temperature (optional)
alb_bg.nc : background albedo (optional)

Restart files (input/output)

driver_rest_in.nc, sechiba_rest_in.nc, stomate_rest_in.nc

Simulation
Output files produced by ORCHIDEE

Restart files

By default, the model produces 3 restart files : for the driver, sechiba and stomate components. Theses files are used to restart the model for next period.

1St execution, no restart file to start the model Files produced are :
driver_rest_out.nc, sechiba_rest_out.nc, stomate_rest_out.nc

2Nd execution, before rename restart files produced in 1st run into :
driver_rest_in.nc, sechiba_rest_in.nc, stomate_rest_in.nc
and add following in run.def :

```
SECHIBA_restart_in =     sechiba_rest_in.nc
STOMATE_RESTART_FILEIN = stomate_rest_in.nc
RESTART_FILEIN =         driver_rest_in.nc
```

3rd execution, rename restart files produced in 2nd run

etc...

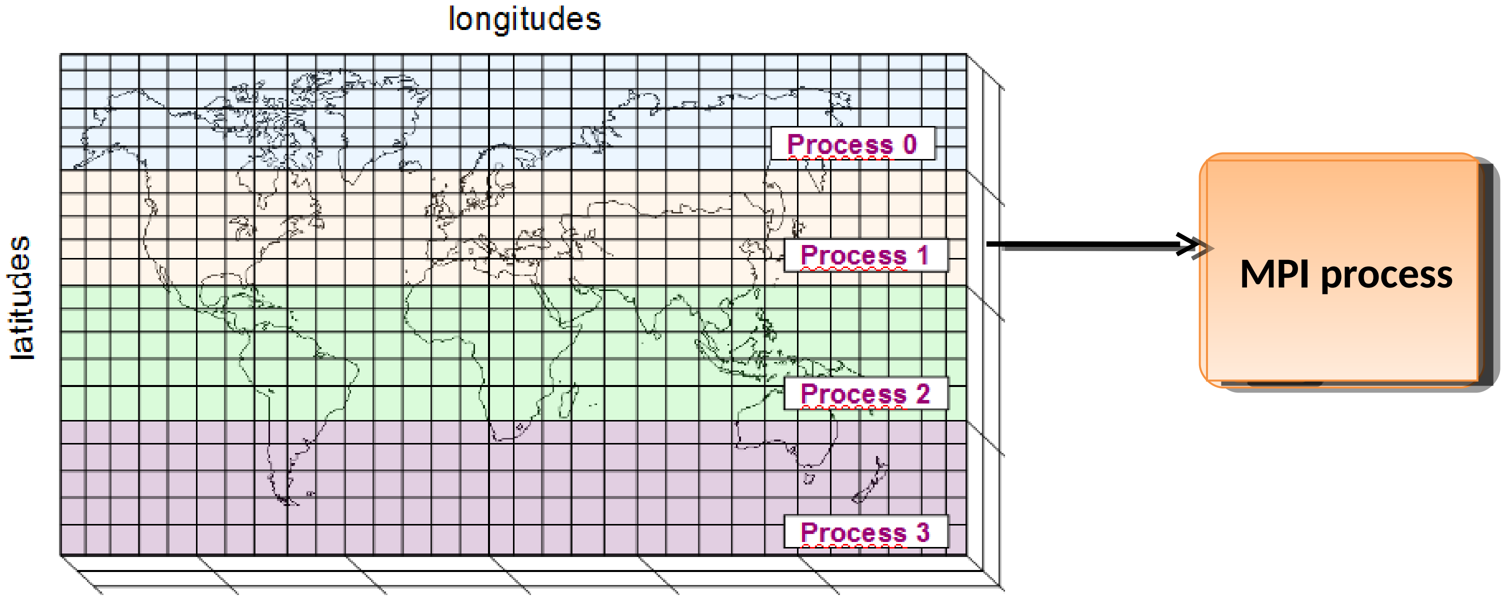
Simulation
Output files produced by ORCHIDEE

Diagnostics **"history files"**
The diagnostics/output are written by XIOS in netCDF format.
Different files, with different levels of output can be configured :

sechiba_history.nc
sechiba_out_2.nc
stomate_history.nc
stomate_ipcc_history.nc

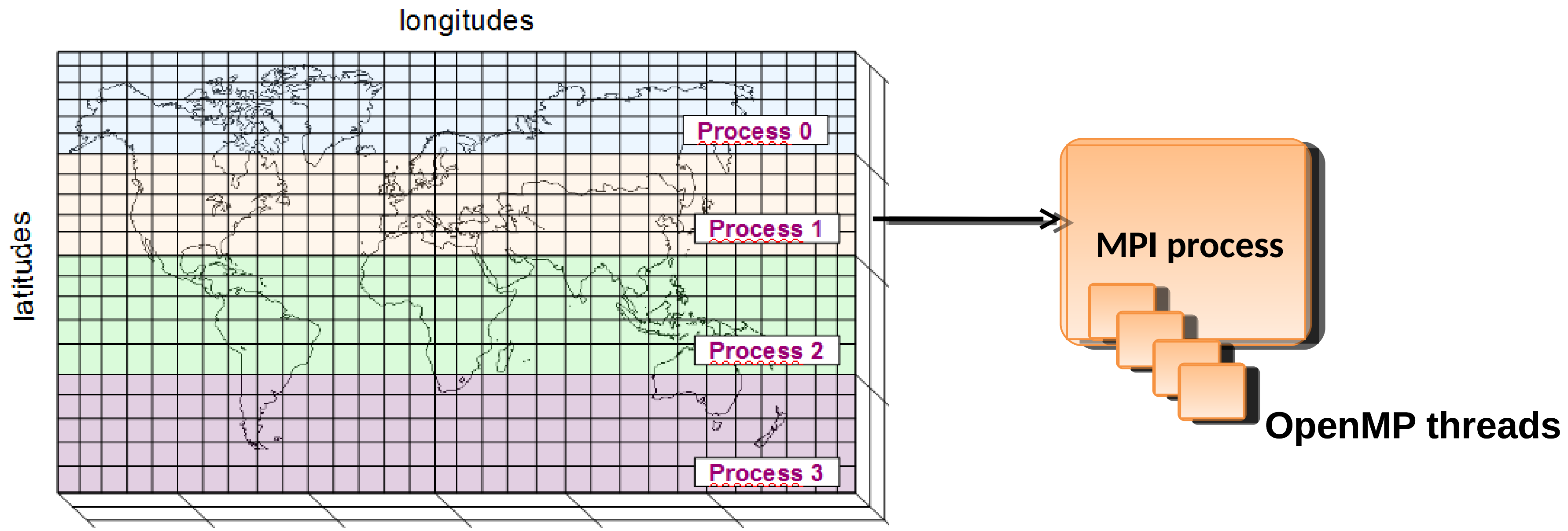
See hands on exercises how to change level and frequency.

Simulation
Parallel mode



- **MPI parallelisation (distributed memory)** : the global domain is divided into sub-domains, each MPI process treats one sub-domain on one computing core.

Simulation
Parallel mode



- **MPI parallelisation (distributed memory)** : the global domain is divided into sub-domains, each MPI process treats one sub-domain on one computing core.

- **Hybbrid MPI + OpenMP (distributed + shared memory)** : each MPI process treats one sub-domain, each MPI process run OMP threads, each OMP threads treats a new subdivision of the sub-domain. *Only possible in coupled mode with LMDZ.*

Simulation
Configurations using libIGCM

libIGCM is a script library developed at IPSL
S. Denvil in charge

Training courses are frequently given to use libIGCM

See documentation here:
http://forge.ipsl.jussieu.fr/igcmg_doc/Doc

Simulation
ORCHIDEE_OL configuration using libIGCM

Content in **modipsl/config/ORCHIDEE_OL**, in the trunk:

OOL_SEC_STO
OOL_SEC
SPINUP_ANLAYTIC
FORCESOIL
TESTSTOMATE

Classic submit directories for different experiments using libIGCM

Obsolete : Not longer maintained in the trunk

ENSEMBLE
SPINUP

More complex configurations used for FLUXNET simulations

AA_make
AA_make.ldef
Makefile

Files for compilation

Simulation
ORCHIDEE_OL configuration using libIGCM

OOL_SEC_STO
Experiment set up with sechiba and stomate

OOL_SEC
Experiment set up with sechiba only

SPINUP_ANLAYTIC
Experiment set up with sechiba, stomate and spinup_analytic activate. In this experiment, the forcing is set to loop over 10 years.

FORCESOIL and TESTSTOMATE
Obsolete experiments, replaced by spinup_analytic. To use these experiments you need first to produce specific forcing files.

Simulation
Differences in ORCHIDEE_OL

The configuration ORCHIDEE_OL have some differences compared to the coupled v6 configurations such as LMDZOR_v6.

No need to create the submission directory: They already exist. Choose one of the existing directories and copy it to a new name, modify and create the main job using ../../../libIGCM/ins_job.

No DRIVER directory: The comp.driver files are found in COMP directory.

Simulation
Modify parameters

In PARAM/run.def some parameters are modified by orchidee_ol.driver, sechiba.driver and stomate.driver. These parameters are always marked equal AUTO or AUTOBLOCKER

AUTO : These parameters can be changed using options in *comp.card* or *config.card.* You can also change them directly in the run.def file, for this case the drivers will not change them again.

AUTOBLOCKER : The job will stop if you modify these parameters. They are set by the *comp.driver* mainly using the information from config.card.

For example, in PARAM/run.def:

```
STOMATE_RESTART_FILEIN = _AUTOBLOCKER_
XIOS_ORCHIDEE_OK = _AUTO_
```

=> You can add or modify parameters directly in PARAM/run.def

Coding Guidelines

All new developements in the ORCHIDEE trunk must follow the Coding Guidelines.

See https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation

- **Comments in english**
- **Indentation**
- **Key words in capital letters**
- **Module and subroutine description part**
- **...**