ORCHIDEE Training course

Code management, installation, simulation, documentation

22 October 2015, IDRIS

Josefine Ghattas IPSL

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
How to install on a local PC ?

Simulation

What is needed to run in off-line mode
What is produced by ORCHIDEE ?
Parallel mode
Configurations using libIGCM

Finding information

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

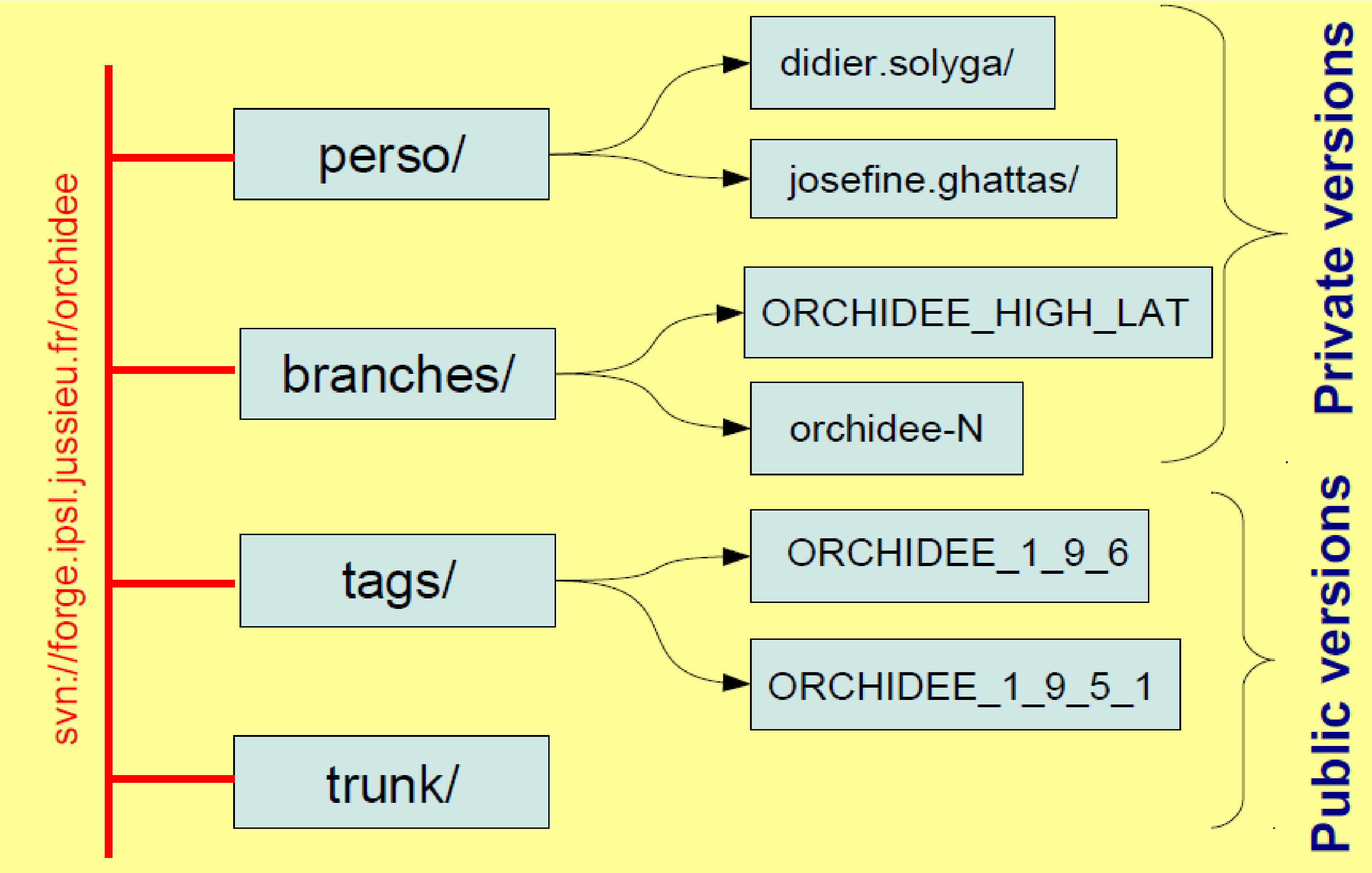
Tag : A tag is a copy of the trunk at a specific moment. Latest tag is ORCHIDEE 1.9.6.

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 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

Different versions of ORCHIDEE



A modification in one directory do not influence the others.

For example, a bug correction in the trunk/ORCHIDEE do not affect branches/ORCHIDEE-MICT

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.

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 version and revision number are needed.

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

Chemin : .
URL : svn://forge.ipsl.jussieu.fr/orchidee/trunk/ORCHIDEE
Racine du dépôt : svn://forge.ipsl.jussieu.fr/orchidee
UUID du dépôt : f489ceea-5127-0410-b15c-c4a6149ed9a7
Révision : 1172
Type de nœud : répertoire
Tâche programmée : normale
Auteur de la dernière modification : anne.cozic
Révision de la dernière modification : 1172
Date de la dernière modification: 2013-02-06 16:14:25 +0100 (mer. 06 févr. 2013)
```

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

see also hands-on exercises this afternoon

- **modipsl is a tool used to install and compile ORCHIDEE**

- 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.

Install and compile
How to install

see also hands-on exercises this afternoon

- **modipsl is a tool used to install and compile ORCHIDEE**

- 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 extract 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,
                                   see following slide
```

Install and compile
Predefined versions with ORCHIDEE

ORCHIDEE_trunk : the latest version on ORCHIDEE/trunk for offline use. For developments this is most 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

ORCHIDEE_TAG : the latest tagged version for off-line use. For the moment this version corresponds to the tag ORCHIEE_1_9_6. Only to be used for comparison with older simulations.

ORCHIDEE_SVN_AR5 : off-line version used for reference CMIP5 simulations. Only to be used for comparisons to older simulations.

Install and compile
Predefined versions with ORCHIDEE

ORCHIDEE_trunk : the latest version on ORCHIDEE/trunk for offline use. Note that this version is not stable and precaution must be taken. *It is possible to take a specific revision number on the trunk.*

LMDZOR_v6 : configuration under development. To be use for coupling the latest version of the trunk ORCHIDEE with LMDZ(atmospheric model).

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

IPSLCM6_rc0 : coupled configuration containing NEMO(ocean model), LMDZ and ORCHIDEE. Close to CMIP6 version.

IPSLCM5_v5 : stable version of the coupled configuration containing NEMO(ocean model), LMDZ and ORCHIDEE version AR5(CMIP5).

Install and compile
Install a branch or private version for off-line

You can modify the version of the ORCHIDEE source before extraction of the 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/ORCHIDEE HEAD 14 ORCHIDEE modeles

For exemple:

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

Install and compile
Which machine ?

- Compiling options of ORCHIDEE is preconfigured at following machines : **obelix** at LSCE, **curie** at TGCC, **ada** at IDRIS, **ciclad** 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 use computer centres you need :

- **a login connected to an existing group**, 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
Compiling with makeorchidee_fcm

- Compiling is done by the script **makeorchidee_fcm** in ORCHIDEE. This compile method is based on the tool FCM.
- Dependencies between modules are determined automatically. Nothing to do 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
- makeorchidee_fcm is launched by the main makefile in the config directory
  - modipsl/config/ORCHIDEE_OL/Makefile

Install and compile
Compiling with makeorchidee_fcm

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 with XIOS:

```
> gmake with_xios
```

After successful compiling, executables are found in :

modipsl/bin/

Compiling off-line driver for older versions such as tag 1.9.6/AR5

```
> cd modipsl/modeles/ORCHIDEE_OL
> gmake orchidee_ol
> gmake teststomate
> gmake forcesoil
```

Install and compile
How to install on local PC

See also https://forge.ipsl.jussieu.fr/orchidee/wiki/HowTo/InstallingORCHIDEE

* Install gfortan compiler

* Install netcdf and compile with the same compiler as above

* Modify path to netcdf in modipsl/util/AA_make.def, target gfortran :

```
#-Q- gfortran  NCDF_INC = where you installed netcdf include files
#-Q- gfortran  NCDF_LIB  = where you installed netcdf library
```

* Modify path to netcdf in modipsl/modeles/ORCHIDEE/arch/arch-gfortran.path

* Create main makefile for target gfortran

```
> cd modipsl/util; ./ins_make -t gfortran
```

Simulation
What is needed to run in off-line mode ?

Executable : orchidee_ol

Parameter text file for run options : run.def

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 : soil parameters
PFTmap.nc : vegetation map
routing.nc : river routing (optional)
floodplains.nc and cartepente2d_15min.nc (optional)
reftemp.nc : temperature (optional in the trunk)

Restart files

driver_rest_in.nc, sechiba_rest_in.nc, stomate_rest_in.nc

Simulation
What is produced by ORCHIDEE ?

Diagnostics **“history files”**
The diagnostics/output are written by IOIPSL or 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
What is produced by ORCHIDEE ?

Restart files

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

1St run, no restart file to start the model Files produced are :

driver_rest_out.nc, sechiba_rest_out.nc, stomate_rest_out.nc

2Nd run, before rename restart files produced in 1st run into :

driver_rest_in.nc, sechiba_rest_in.nc, stomate_rest_in.nc

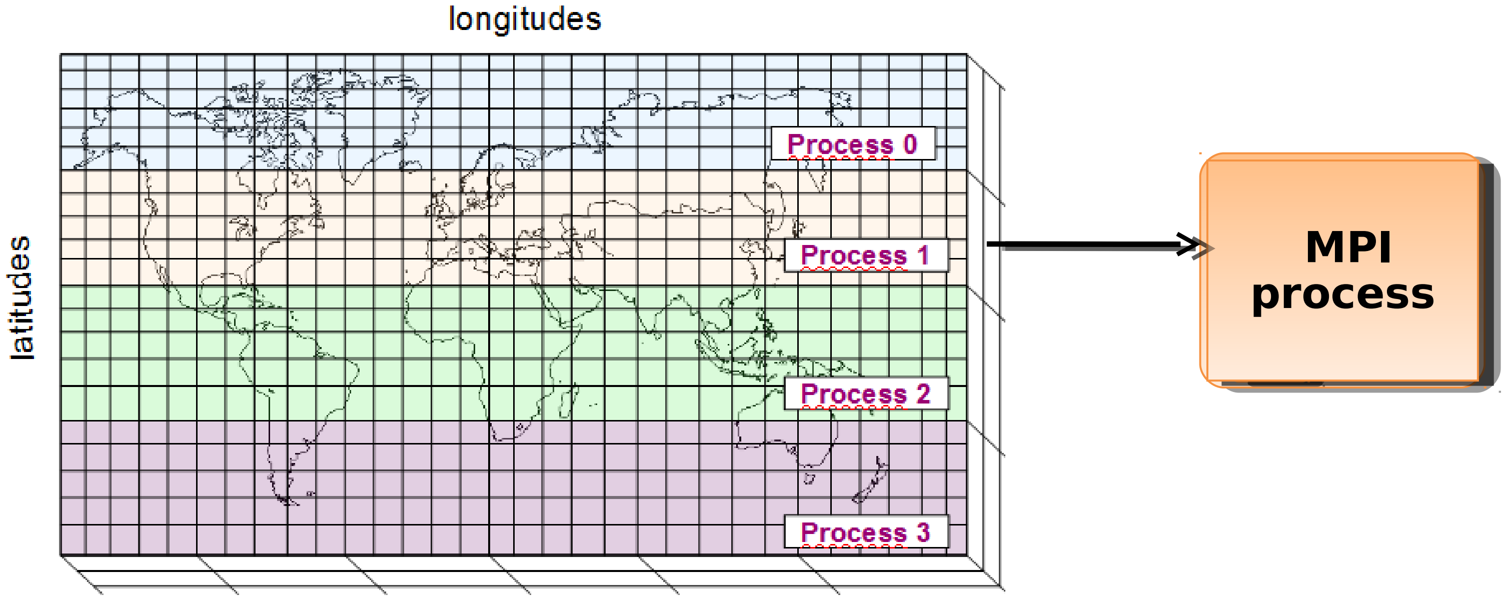
and add following in run.def :

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

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

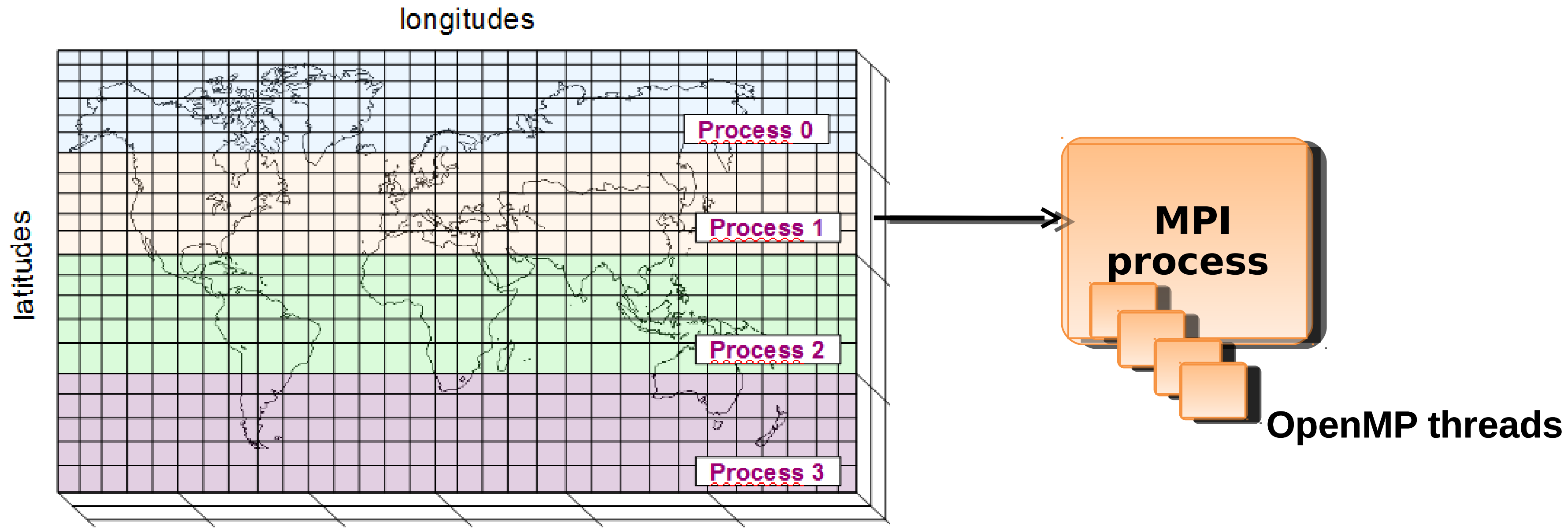
etc...

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.

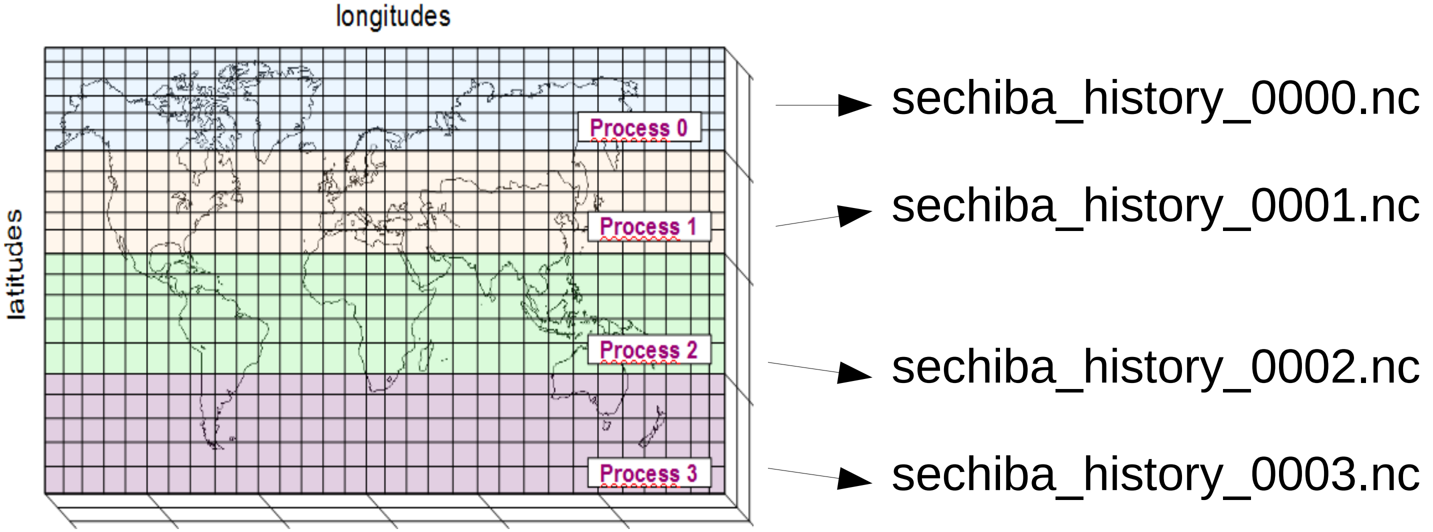
Simulation
Parallel mode



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

- **MPI + OpenMP (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
Parallel mode



sechiba_history_0000.nc

sechiba_history_0001.nc

sechiba_history_0002.nc

sechiba_history_0003.nc

rebuild

sechiba_history.nc

IOIPSL

Simulation
Parallel mode MPI : IOIPSL

Each processor will write history files for its domain. The file will have extension with the rank number of the processes. For example for running on 4 processes :

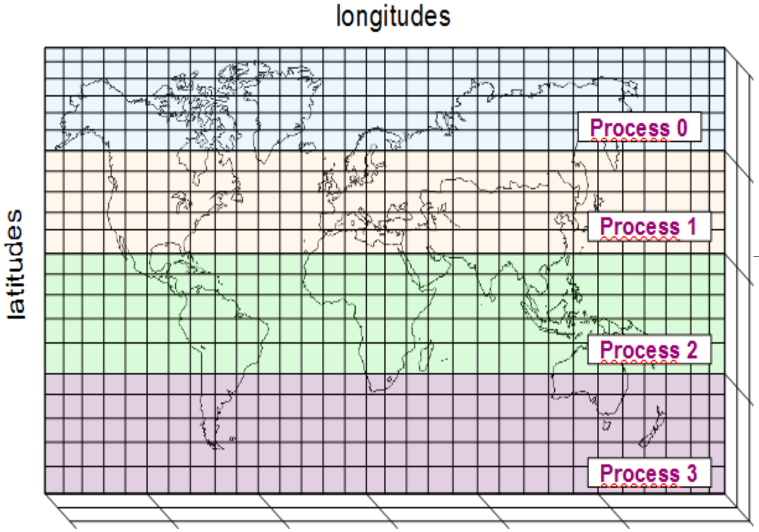
```
sechiba_history_0000.nc
sechiba_history_0001.nc
sechiba_history_0002.nc
sechiba_history_0003.nc
```

History files have to be recombined to the total domain using the tool ***rebuild***. Rebuild is an extension tool to IOIPSL.

Syntax : > rebuild -o outfile.nc inputfile1 inputfile2 inputfile3 ...

```
> rebuild -o sechiba_history.nc sechiba_history_00*
```

Simulation
Parallel mode



► sechiba_history.nc

XIOS

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

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 20years.

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 v5 configurations such as LMDZOR_v5.

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
Recent changes in libIGCM

Since tag libIGCM_v2.7,
Change from using **util/ins_job** info **libIGCM/ins_job**
-> 1 change in config.card necessary

Before in config.card:

```
JobNumProcTot=32
[Executable]
SRF= ("", "")
SBG= ("", "")
OOL= (orchidee_ol, orchidee_ol)
```

Now in config.card:

```
[Executable]
SRF= ("", "")
SBG= ("", "")
OOL= (orchidee_ol, orchidee_ol, 32MPI)
```

Simulation
Recent changes in the configurations

In PARAM/run.def some parameters are modified by the comp.driver (the script). Parameters marked as:

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 : These parameters can not be changed. 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 other parameters directly in PARAM/run.def

= > In coupled configurations the parameters for ORCHIDEE are set in PARAM/orchidee.def_Choi or PARAM/orchidee.def_CWRR

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 core team

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

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

It is possible to have a **“login forge” to the wiki**. This login is used for visualization of personal SVN folder and to edit wiki pages. Write to orchidee-help to get a login. *This is not the same login as for SVN.*

Finding information
Scientific documentation

- The 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

Finding information
“Developer's meeting”

About every 2 months a “developer's meeting” is hold. These meetings consist in a presentation of a specific topic followed by discussions and questions. A short technical presentation might also be given. Meeting place in Jussieu.

See reports and presentations here :
http://forge.ipsl.jussieu.fr/orchidee/wiki/Meetings/Developer

Information about these meetings are done at orchidee-dev email list. **All users/developers of ORCHIDEE are welcome.**