# ORCHIDEE Training course

Code management, installation, simulation, documentation

6 November 2014, 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 MPI
- 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.***

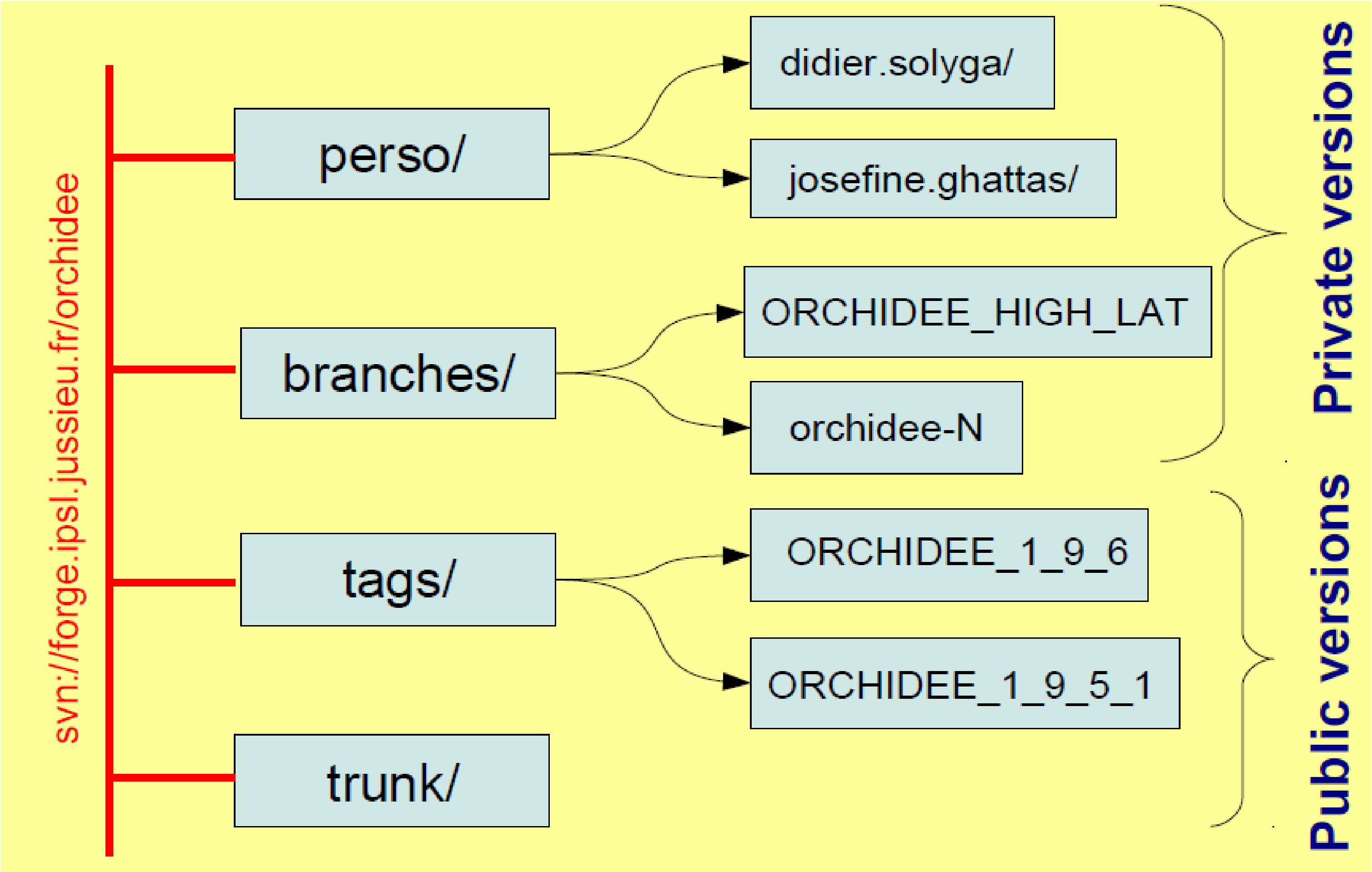
**Tag** : **Validated version of ORCHIDEE.** A tag is a copy of the trunk at a specific moment. Latest tag is ORCHIDEE 1.9.6.

**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 version and revision number 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 <u>version</u> and <u>revision number</u> 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. 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_name*** 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_name        # extract a predefined configuration,
                               see following slide
```

# Install and compile
# **Predefined versions with ORCHIDEE**

**ORCHIDEE_TAG** : the latest tagged version for off-line use. For the moment this version corresponds to the tag ORCHIEE_1_9_6.

**ORCHIDEE_SVN_AR5** : off-line version used for reference CMIP5 simulations.

**ORCHIDEE_trunk** : the latest version on ORCHIDEE/trunk for offline use. Note that this version is not stable and precaution must be taken.

# Install and compile
# **Predefined versions with ORCHIDEE**

**ORCHIDEE_TAG** : the latest tagged version for off-line use. For the moment this version corresponds to the tag ORCHIEE_1_9_6.

**ORCHIDEE_SVN_AR5** : off-line version used for reference CMIP5 simulations.

**ORCHIDEE_trunk** : the latest version on ORCHIDEE/trunk for offline use. Note that this version is not stable and precaution must be taken.

**LMDZOR_v5** : this is a configuration using LMDZ5(atmospheric model) and ORCHIDEE version AR5.

**LMDZOR_v5.2** same as LMDZOR_v5 but with ORCHIDEE trunk

**IPSLCM5_v5** : coupled configuration containing NEMO(ocean model), LMDZ and ORCHIDEE version AR5. Different versions of IPSLCM5_v5 exist also.

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

1) **Default method**: directory ORCHIDEE/src_driver must exist

```
> svn co http://forge.ipsl.jussieu.fr/igcmg/svn/modipsl/trunk modipsl
> cd modipsl/util
> vi mod.def
  Modify line containing :
#-C- ORCHIDEE_trunk  trunk/ORCHIDEE               HEAD 14 ORCHIDEE modeles
  into
#-C- ORCHIDEE_trunk  branches/xxx/ORCHIDEE HEAD 14 ORCHIDEE modeles
> ./model ORCHIDEE_trunk
```

2) **If the version do not contain src_driver**, then driver sources are found in ORCHIDEE_OL directory. Use the script modipsl/util/recup_my_ORCHIEEE. For example :

```
./recup_my_ORCHIDEE  josefine.ghattas branches/Hydrology
```

# Install and compile
# **Where is the model installed ?**

Source code

- **modipsl/modeles/ORCHIDEE** ORCHIDEE Fortran sources
- **modipsl/modeles/IOIPSL** Fortran library for input/output issues, to be linked to the model
- modipsl/modeles/LMDZ Atmospheric model, if installing a coupled configuration

# Install and compile
# **Where is the model installed ?**

<u>Source code</u>

- **modipsl/modeles/ORCHIDEE** ORCHIDEE Fortran sources
- **modipsl/modeles/IOIPSL** Fortran library for input/output issues, to be linked to the model
- modipsl/modeles/LMDZ Atmospheric model, if installing a coupled configuration

<u>Scripts</u>

- **modipsl/libIGCM** a ksh script library used for simulation

- **modipsl/config/ORCHIDEE_OL** scripts to run with libIGCM in off-line mode

or for use in coupled mode with LMDZ :
- modipsl/config/LMDZOR_v5

# Install and compile

**libIGCM** is a script library developed at IPSL
**Training courses are frequently given** to use libIGCM
Next date : spring 2015

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

Scripts

- **modipsl/libIGCM** a script library used for simulation

- **modipsl/config/ORCHIDEE_OL** scripts to run with libIGCM in off-line mode

or for use in coupled mode with LMDZ :
- modipsl/config/LMDZOR_v5

# 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 these computer centres you need :

- **a login connected to an existing group**, ask your supervisor

- **CPU resources** in this group, yearly demand, ask your supervisor

- 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
# **Compiling with standard makefiles**

- Old method: **A Makefile in each source directory**
- Standard makefiles created based on AA_make and AA_make.gdef. Main makefile is found in ORCHIDEE.
- Specific platform dependent compile options are set in modipsl/util/AA_make.gdef
- Dependencies must be declared in the AA_make files.

# Install and compile
## **How to compile ?**

Compiling off-line driver for the trunk version

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

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

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

After successful compiling, executables are found in :

modipsl/bin

# 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
```

* Compile as usual: cd modipsl/config/ORCHIDEE_OL; gmake

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

**Executable :** orchidee_ol

**Parameter text file for run options :** run.def

**Inital state files - <u>if no restart files</u>**

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

**Restart files**

driver_rest_in.nc, sechiba_rest_in.nc, stomate_rest_in.nc

**Boundary conditions**

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

# Simulation
# What is produced by ORCHIDEE ?

**Diagnostics** **“history files”**
The diagnostics/output are written by IOIPSL in netCDF format. Different files, with different levels of output are configured in run.def.

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**
**4th run, rename restart files produced in 3rd run**
**etc...**

# Simulation
# **Parallel mode MPI**

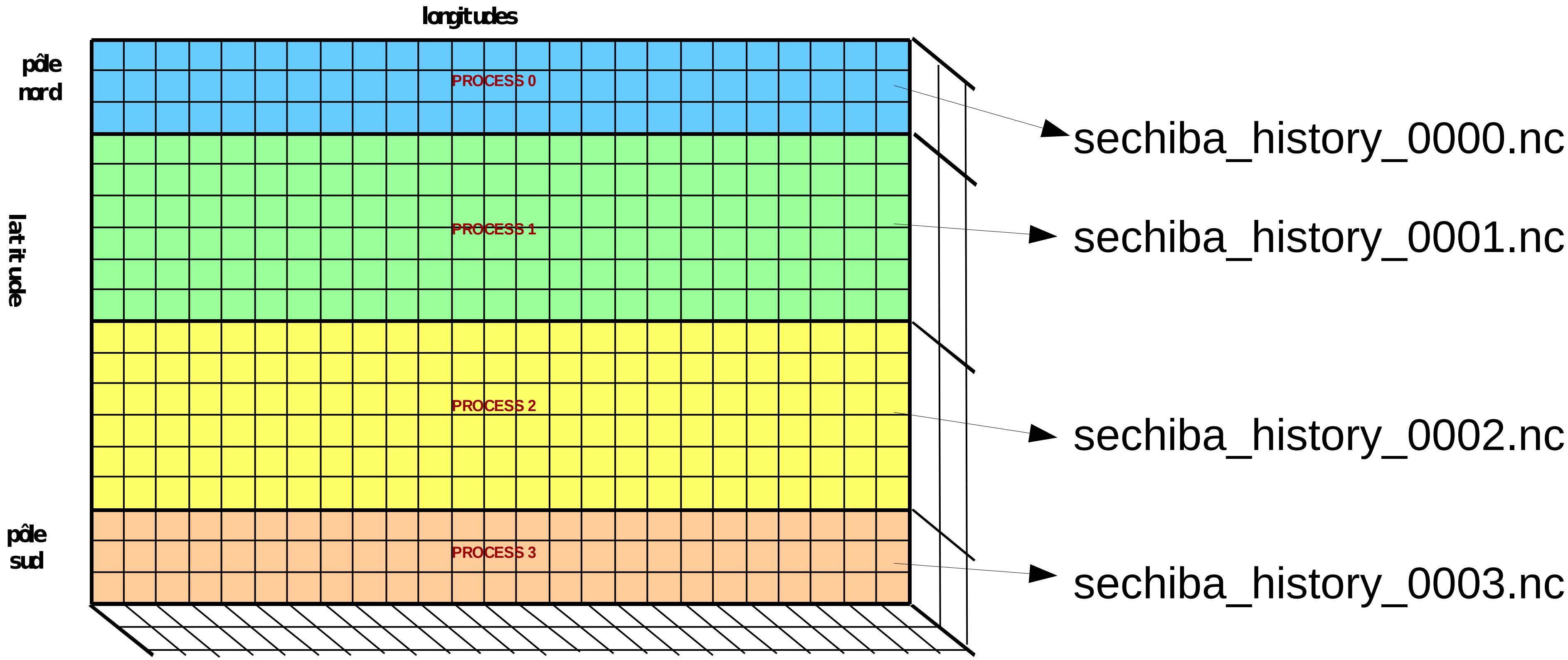
- The model can be run in sequential or parallel mode.

- For parallel mode with MPI, compiling must be done with preprocessing key CPP_PARA and linking must be done with MPI library.

- Running is done on several processes, communication is done with MPI (Message Passing Interface library). Each processes will calculate a part of the total domain.

- Hybrid mode OpenMP – MPI exists in the trunk and can be used in coupled mode with LMDZ.

# Simulation
# **Parallel mode MPI**



sechiba_history_0000.nc

sechiba_history_0001.nc

sechiba_history_0002.nc

sechiba_history_0003.nc

# Simulation
# **Parallel mode MPI**

**Each processes 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
# **Configurations using libIGCM**

**libIGCM is a script library developed at IPSL, responsable S. Denvil**

**Training courses are frequently given** to use libIGCM
Next date : spring 2015

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<br>OOL_SEC<br>SPINUP_ANLAYTIC<br>FORCESOIL<br>TESTSTOMATE | Classic submit directories for different experiments using libIGCM |
| ENSEMBLE<br>SPINUP | More complex configurations used for FLUXNET simulations |
| AA_make<br>AA_make.ldef<br>Makefile | Files for compiling |

# 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
# **Differencies in ORCHIDEE_OL**

The configuration ORCHIDEE_OL have some differecies 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 as wanted and create the main job using ../../../util/ins_job.

**Changes in PARAM/run.def will always be taken into account.** No variables are set to AUTO.

- In LMDZOR_v5, some variables are set =AUTO. In that case they will always be set by the comp.driver.

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

# Simulation
# **SPINUP_ANALYTIC**

SPINUP_ANALYTIC uses a new option in libIGCM to cycle over a set of years. The variables CyclicBegin and CyclicEnd is added in config.card to activate this possiblity. The variable CyclicYear is then avaiable and can be used in orchidee_ol.card.

```
In SPINUP_ANALYTIC/config.card:
#-- Start and End of Job
DateBegin=1901-01-01
DateEnd=2240-12-31
# Forcing data between 1901 and 1920
CyclicBegin=1901
CyclicEnd=1920

In SPINUP_ANALYTIC/COMP/orchidee_ol.card
[BoundaryFiles]
List=   (${R_BC}/.../cruncep_${CyclicYear}.nc, forcing_file.nc)
```

# Finding information
# **Email lists @ipsl.jussieu.fr**

3 email lists exist for inscriptions :

| | |
|---|---|
| - orchidee-dev | discussion and information |
| - orchidee-users | some general information |
| - orchidee-meeting | |

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 give. 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.**