ORCHIDEE SEMINAR :

Modifying parameters values and adding PFT's at running time : presentation of a new version of ORCHIDEE

Didier Solyga
31/01/2011

Plan

- Why a new version of ORCHIDEE ?
- The characteristics of the new version
- The concept of Metaclass
- How to download and how to use this version ?
- For developpers only
- Perspectives
- Some links

Why a new version of ORCHIDEE ?

- To have the possibility to change the number of PFTs => better representation of the diversity of vegetation, specially useful for regional studies
- To have the possibility to modify any parameter value you want
- To sum up, to have a more flexible and user-friendly version of ORCHIDEE

Characteristics of the externalized version

- Based on ORCHIDEE 1_9_5
- Possibility to change the values of 300 parameters (more or less) without having to recompile the code
- The number of PFTs (parameter 'nvm' in ORCHIDEE) is dynamic : you could choose 'nvm' between 2 and X (X is any number you want) thanks to the metaclass (based on an idea of N.Viovy)
- Documentation with a small tutorial

The concept of Metaclass (1)

- Concerns only the PFT-parameters
- All the 13 PFTs are defined by a set of common parameters (so one parameter is defined by 13 values in ORCHIDEE). The values could be different for each PFT. These values are fixed in the AR5 version.
- How could we change the number of PFTs and the values of the corresponding parameters ? => by linking a PFT to a Metaclass (=MTC)

The concept of Metaclass (2)

- The MTC is an overlayer of the PFT. A MTC is defined by the same set of parameters followed by the PFTs. We implemented 13 MTCs (matching to the current 13 PFTs in ORCHIDEE)
- How does it work ? The idea is to associate one PFT to one MTC. When you do this, the PFT will heritate automatically all the defaults values for each parameter of the corresponding MTC.

How to download it ?

- Create a new folder, go into it and type in a shell :
  svn co http://forge.ipsl.jussieu.fr/igcmg/svn/modipsl/trunk modipsl
- Then go to modipsl/util and modify mod.def (see next slides)
- After that do : ./model ORCHIDEE_SVN_EXT
- Execute : ./ins_make (it will create the Makefile)
- Go to modipsl/modeles/ORCHIDEE_OL and compile in the following order : gmake ; gmake forecesoil and gmake testomate
- Create your own job in modipsl/config/ORCHIDEE_OL (see the presentation of Martial_Mancip)
- Execute: ./ins_job in modipsl/util
- It is done !

The mod.def file (1)

Edit mod.def and add the line in red :

```
#- Repository informations
#-
#-S- 1 cvs anonymous@cvs.ipsl.jussieu.fr:/home/ioipsl/CVSROOT
#-S- 2 cvs sechiba@cvs.ipsl.jussieu.fr:/home/ssipsl/CVSREP
#-S- 3 cvs lmdzbrowse@cvs.lmd.jussieu.fr:/home/cvsroot
#-S- 4 cvs opa@cvs.ipsl.jussieu.fr:/home/opalod/CVSROOT
#-S- 5 cvs nemo@cvs.ipsl.jussieu.fr:/home/opalod/NEMOCVSROOT
#-S- 6 cvs inca@cvs.ipsl.jussieu.fr:/home/incaipsl/CVSROOT
#-S- 7 svn http://forge.ipsl.jussieu.fr/nemo/svn
#-S- 8 svn http://forge.ipsl.jussieu.fr/igcmg/svn
#-S- 9 svn --username inca http://forge.ipsl.jussieu.fr/inca/svn
#-S- 10 svn http://forge.ipsl.jussieu.fr/libigcm/svn
#-S- 11 svn http://svn.lmd.jussieu.fr/LMDZ
#-S- 12 svn http://forge.ipsl.jussieu.fr/ioserver/svn
#-S- 13 svn http://forge.ipsl.jussieu.fr/fcm/svn
#-S- 14 svn --username sechiba svn://forge.ipsl.jussieu.fr/orchidee
```

PS : You don't need to do this if you have updated or downloaded the last version of libIGCM

The mod.def file (2)

Edit mod.def and add (or copy) these lines :

```
#-H- ORCHIDEE_EXT_SVN Modele ORCHIDEE with Externalisation of the parameters
#-H- ORCHIDEE_EXT_SVN ORCHIDEE_EXT with tag ORCHIDEE_1_9_5
#-H- ORCHIDEE_EXT_SVN IOIPSL with tag 2.2.1
#-H- ORCHIDEE_EXT_SVN libIGCM tag libIGCM_v1_9
#-H- ORCHIDEE_EXT_SVN FCM svn version PATCHED/FCM_V1.2
#-M- ORCHIDEE_EXT_SVN Martial.Mancip@ipsl.jussieu.fr
#-C- ORCHIDEE_EXT_SVN IOIPSL/tags/v2_2_1/src   HEAD         8  IOIPSL/src      modeles
#-C- ORCHIDEE_EXT_SVN tags/libIGCM_v1_9        HEAD         10 libIGCM         .
#-C- ORCHIDEE_EXT_SVN PATCHED/FCM_V1.2         HEAD         13 tools/FCM/V1.2  .
#-C- ORCHIDEE_EXT_SVN branches/ORCHIDEE_EXT/ORCHIDEE                 ?      14 .
#-C- ORCHIDEE_EXT_SVN branches/ORCHIDEE_EXT/ORCHIDEE_OL              ?      14 .
#-C- ORCHIDEE_EXT_SVN trunk/ORCHIDEE_OL/OOL_SEC_STO  ?      14 config/ORCHIDEE_OL/OOL_SEC_STO .
#-C- ORCHIDEE_EXT_SVN trunk/ORCHIDEE_OL/OOL_SEC      ?      14 . config/ORCHIDEE_OL
#-C- ORCHIDEE_EXT_SVN trunk/ORCHIDEE_OL/FORCESOIL    ?      14 . config/ORCHIDEE_OL
#-C- ORCHIDEE_EXT_SVN trunk/ORCHIDEE_OL/TESTSTOMATE  ?      14 . config/ORCHIDEE_OL
#-C- ORCHIDEE_EXT_SVN trunk/ORCHIDEE_OL/SPINUP       ?      14 . config/ORCHIDEE_OL
#-C- ORCHIDEE_EXT_SVN trunk/ORCHIDEE_OL/FLUXNET      ?      14 . config/ORCHIDEE_OL
```

PS : this advice is available if you want to dowload ORCHIDEE_FM (resp. ORCHIDEE-N):
for ORCHIDEE_FM replace the orange line by FM_SVN (resp OCN_SVN) and the violet
one by orchidee_FM (resp orchidee-N)

How to launch a job? (for beginners only)

- Go to the folder in ../modipsl/config/ORCHIDEE_OL
- Focuse on the job called OOL_SEC and OOL_SEC_STO
- OOL_SEC = sechiba only and OOL_SEC_STO = sechiba + stomate
- Copy one of into a folder My_own_job (or any other name)
- Go into My_own_job and delete Job_SECHIBA (or Job_SECHSTOM) and run.card.init
- Modify config.card (see course M.Mancip), then go to ../modipsl/util/ and execute ./ins_job
- Come back to My_own_job folder, you are almost ready to launch your first job!

The file orchidee.def

- Each job contains a folder called /PARAM.
- In the PARAM folder you will find a file called orchidee.def (old version).
- This is the file you have to modify for using the externalisation of the parameters. You will find the parameters externalized for the AR5 version.

Some syntax rules and things you should to know

- Each parameter in the code is associated to a key word in capital letters in orchidee.def . Generally, the names of the parameters and the key words matche exactly. There exists some exceptions (you could find the list in the documentation folder)
- For a scalar parameter : all you have to do is to add the config key in orchidee.def and set the new value.
- Syntax rules for arrays : in orchidee.def, you have two possibilities to define an array :
  1) global definition : **ARRAY = 1, 2, 3, 4**
  2) component by component : **ARRAY__01 = 1**
  ARRAY__02 = 2
  (! double underscore) **ARRAY__03 = 3**
  ARRAY__04 = 4

How can I add a new PFT ?

- To add a new PFT in PARAM/orchidee.def, follow these steps :
  1) Set : **NVM = X** (X is an integer representing the number of PFTS)
  2) Fill an array called PFT_TO_MTC like this :
  PFT_TO_MTC__01 = 01
  ...
  PFT_TO_MTC__X = ...
  (you have to write all the nvm components)
  3) Set the flag IMPOSE_VEG to true if nvm is different to 13 : **IMPOSE_VEG = y**
- **One rule : set always PFT_TO_MTC__01 = 01**

Example of orchidee.def

```
# Should the vegetation be prescribed
# This flag allows the user to impose a vegetation distribution
#  and its characterisitcs. It is espacially interesting for 0D
#  simulations. On the globe it does not make too much sense as
#  it imposes the same vegetation everywhere
IMPOSE_VEG = y
# default = n

   # Number of vegetation types (by default 13) :
   NVM = 15

PFT_TO_MTC__01 = 1
PFT_TO_MTC__02 = 2
PFT_TO_MTC__03 = 3
PFT_TO_MTC__04 = 4
PFT_TO_MTC__05 = 5
PFT_TO_MTC__06 = 6
PFT_TO_MTC__07 = 7
PFT_TO_MTC__08 = 8
PFT_TO_MTC__09 = 9
PFT_TO_MTC__10 = 10
PFT_TO_MTC__11 = 11
PFT_TO_MTC__12 = 12
PFT_TO_MTC__13 = 13
PFT_TO_MTC__14 = 5
PFT_TO_MTC__15 = 7
```

```
# PFT_TO_MTC :
# Here is the list of the MTC implemenented in the
model
# & (/ 'bare ground                          ', &        !  1
# &    'tropical  broad-leaved evergreen  ', &    !  2
# &    'tropical  broad-leaved raingreen  ', &      !  3
# &    'temperate needleleaf   evergreen  ', &     !  4
# &    'temperate broad-leaved evergreen  ', &    !  5
# &    'temperate broad-leaved summergreen', &  !  6
# &    'boreal    needleleaf   evergreen  ', &          !  7
# &    'boreal    broad-leaved summergreen', &    !  8
# &    'boreal    needleleaf   summergreen', &       !  9
# &    '          C3          grass      ', &          ! 10
# &    '          C4          grass      ', &          ! 11
# &    '          C3          agriculture', &          ! 12
# &    '          C4          agriculture'  /)        ! 13
```

How can I modify a pft-parameter?

- Let us interest to the pft-parameter **VCMAX_OPT**. The standard values for the 13 MTCs are :
  (/ undef, 65., 65., 35., 45., 55., 35.,
  45., 35., 70., 70., 70., 70. /)

- What happened when I set PFT_TO_MTC__14 = 5 ?
  => Your 14th PFT will be associated as the 5th MTC : temperate broad-leaved evergreen

- What will be the value of VCMAX_OPT__14 ?
  => It will share the value of the 5th MTC, here 45.

- How can I overwrite it ?
  => Set simply in orchidee.def : **VCMAX_OPT__14 = 40**.

How can I modify an array or a scalar parameter?

- For an no-pft array parameter : Let us consider the parameter **ALBSOIL_VIS.** Its dimension is 9 and its default values are:
  (/0.18, 0.16, 0.16, 0.15, 0.12, 0.105, 0.09, 0.075, 0.25/)
  If you want to modify the third value of this parameter set in orchidee.def :
  ALBSOIL_VIS__03 = 0.25
- For an scalar parameter, it is very easy. Let us consider the parameter called **HCRIT_LITTER**. Its standard value is **0.08**. If you want to overwrite it by 0.075, set in orchidee.def :
  HCRIT_LITTER = 0.075

How can I know which parameters I can modify?

- Look at the folder called Doc_externalised_version in ../modipsl/modeles/ORCHIDEE_OL
- It contains : Guide for beginner (in pdf and doc format) and three .txt files called orchidee_run_parameters, sechiba_parameters_default_values and stomate_parameters_default_values
- Guide for beginner goes deeper this presentation, so you should read it to know how you can control the values you imposed for example.
- The three .txt files give the list of all the parameters and their default values (eventually the key words)

How can I check the values of the parameters read by ORCHIDEE?

- First step : go to the folder ../IGCM_OUT/OL2/*JobName*/OOL/Debug/
- Here you will find three different types of files (for example) :
SECHSTOM_19820101_19820131_out_orchidee_ol
SECHSTOM_19820101_19820131_run.def
SECHSTOM_19820101_19820131_used_run.def
- The most useful is :
SECHSTOM_19820101_19820131_used_run.def
- It lists all the parameters externalised in ORCHIDEE. It indicates also if the value of a parameter comes from the orchidee.def or is default.

Interpretating the comments in the use_run.def file

- You could find three kind of comments in the _use_run.def file illustrated with three examples.
- You imposed a value for a scalar parameter or all the values of an array:
  # Values of LIMIT_NORTH comes from run.def
- You kept the defaults values for an array or a scalar :
  # Values of LAI_MAX are all defaults
- You imposed some values for an array parameter :
  # Values of ALBSOIL_VIS are a mix of run.def and defaults
  => Problem : it doesn't indicate which one is a default value or not

Some other specifications

- The array **PFT_TO_MTC** is **the only array you have to fill completely** if you choose a number of PFTs different of 13 or not in the standard configuration. Although the programm will stop.
- You can impose as many values as you want.
- You still can use orchidee.def in the old way (= without changing any parameter) with this version. We implemented default values for NVM and for PFT_TO_MTC. In this case, it corresponds to usual configuration of ORCHIDEE.
- If you want to launch a job with default values and not having to comment the values you imposed, set the flag called **IMPOS_PARAM** to true.

For developpers only

- We tried to centralize all the parameters in the folder : modipsl/modeles/ORCHIDEE/src_parameters/
- This folder was reorganised and contains only three files : pft_parameters.f90, constantes_mtc.f90 (all default values) and constantes.f90 (scalar and other arrays parameters) => more easier maintenance of the model (adding a MTC or correcting a parameter for example)

Perspectives

- Validation with FLUXNET sites and spin-up job
- Convergence with 1_9_5_1 version
- Use it with a veget map with more ou less than 13 PFts
- Improve and update the documentation
- Find some testers
- Spatialization of some parameters for taking into account the regional variability

Some Links

- On syntax of run.def file :

  http://www.ipsl.jussieu.fr/~ioipsl/WWW2/index.html

  - The presentation of Martial Mancip about the libIGCM library for ORCHIDEE :

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

  - My work on the externalisation of the parameters:

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