

TD1 : prise en main du modèle.

Formation LMDZ

November 23, 2012

Ce premier TD vise principalement à l'installation et à la prise en main d'une version du modèle LMDZ. L'énoncé est disponible sous forme de pdf sur

```
wget http://www.lmd.jussieu.fr/~lmdz/Distrib/TD1.pdf
```

ce qui peut vous permettre de faire des copier/coller.

Si vous travaillez sur les machines du réseau LMD, vous allez vous connecter sous le login "tdlmdz" dont le mot de passe vous sera donné de vive voix. Loguez vous puis créez sur le disque local à la machine un répertoire

```
cd /home/tdlmdz
mkdir LMDZ201211_votre_nom
cd LMDZ201211_votre_nom
```

1 Execution du script install.sh

La première étape consiste à récupérer le script install.sh disponible sur la page du LMD, puis à l'exécuter *en aveugle* (après avoir changé les droits pour le rendre exécutable) :

```
wget http://www.lmd.jussieu.fr/~lmdz/Distrib/install.sh
chmod +x install.sh
./install.sh
```

Par défaut, `install.sh` essaie d'installer le modèle avec le compilateur `gfortran`. Mais la version installée sur le réseau local est trop ancienne. Le script vous invite donc à remplacer `gfortran` par un autre compilateur. Choisir `compilo=g95` puis relancer `./install.sh`.

Ce script est sensé s'exécuter sans encombre jusqu'à la fin. Si ce n'est pas le cas, demandez de l'aide sans plus attendre. Le script se termine par des lignes comme

```
#####
La simulation test est terminée sur ...
vous pouvez la relancer : cd ...
ou ./bench.sh
#####
```

Pendant le déroulement du script, qui peut **prendre** plusieurs minutes, vous verrez apparaître à la fois des lignes associées à la récupération des différents éléments via des `wget` soit des lignes associées à la compilation. Le déroulement du script se termine par le lancement d'une simulation test sur 5 jours avec une grille régulière 48×36-L19.

Pendant l'installation, vous êtes invités à explorer les répertoires sur une autre fenêtre. Au lancement de `install.sh`, vous verrez se créer le répertoire **LMDZtesting** dans le quel vous trouverez **modipsl** – qui contient l'arborescence du modèle – et **netcdf-4.0.1** – qui contient la librairie **netcdf** installée par `install.sh` –. Dans **modipsl**, vous trouvez en particulier le répertoire **modeles** et, à l'intérieur de ce dernier, le répertoire **LMDZ5**. Vous rendre sur celui-ci. Une fois le cas test lancé, vous verrez apparaître en plus le répertoire **BENCH48x36x19/** sur lequel vous pourrez commencer à visualiser les sorties avant même la fin de la simulation.

Regarder le contenu du répertoire. Faire quelques sorties à partir du fichier **histday.nc**.

2 Changement de version du modèle

Le test que vous venez d'effectuer a été réalisé avec la dernière version *testing* du modèle, celle du 16 octobre 2012, associée à la *release svn* 1669, ce que vous pouvez vérifier à la fin du fichier

```
http://www.lmd.jussieu.fr/~lmdz/Distrib/LISMOI.testing
```

ou sur la page web des versions testing

```
http://lmd.lmd.jussieu.fr/utilisateurs/distribution-du-modele/versions-intermediaires
```

ou encore directement sur le répertoire LMDZ :

```
cd LMDZtesting/modips1/modeles/LMDZ5
svn info
```

On peut, à l'aide de `svn`, changer de version du modèle en se rappelant sur une *release svn* ultérieure ou antérieure (on va le faire ici à titre d'exercice, bien qu'employer la dernière testing, celle que vous venez d'installer, soit la bonne façon de travailler). Pour cela, placez vous dans le répertoire **LMDZ5** et lancez la commande:

```
svn update --revision 1665
```

pour revenir à la *release svn* 1665 (la *testing* précédente); Attention: lors de la mise à jour de la version par `svn`, un conflit sur *makegcm* sera signalé, répondre "mf" (mine-full: pour signaler que vous souhaitez conserver votre version personnelle¹ du script **makegcm**).

Toujours depuis le répertoire **LMDZ5** , lancer la commande `./makegcm -d 48x36x19 -v true gcm` L'option `-v true` n'est pas obligatoire. Elle indique qu'on compile avec Orchidee mais on désactive en pratique Orchidee au moment du lancement dans ce TD. Seule une partie des routines vont être recompilées. Il faut relancer une seconde fois la commande² pour que le programme **gcm.e** soit produit.

Une fois le modèle recompilé, lancer une nouvelle simulation. Pour ce faire, créer un nouveau sous-répertoire de **LMDZ5** et y copier les fichiers de conditions au limites et fichiers de paramètres (***.def**, **star*.nc**, **limit.nc**) depuis le répertoire **BENCH48x36x19** , ainsi que l'exécutable **gcm.e** fraîchement compilé.

Puis lancer `./gcm.e > listing 2>&1` (la redirection des affichages du modèle dans un fichier texte permet de garder une trace de l'exécution). La simulation terminée, on peut comparer, à l'aide de la commande `diff` , le résultat des simulations (comparer les fichiers **listing** , **restart.nc** et **restartphy.nc** des deux simulations). Il se trouve qu'à la date de la formation 2012, les changements entre les dernières *testing* ne sont pas sensés changer les résultats numériques (fichiers **restart.nc** et **restartphy.nc**), seuls quelques changements dans les affichages informatifs (fichiers **listing**) devraient apparaître.

Une fois cet exercice fini, revenir à la dernière version *testing* du modèle en se plaçant dans le répertoire **LMDZ5** :

```
svn update
```

A noter que utiliser `svn update` sans préciser de numéro de version permet de se mettre à jour sur la version la plus récente de la branche (à utiliser régulièrement et sans modération pour rester à jour avec les évolutions du modèle!).

¹Ce conflit provient du fait que lors de l'installation via le script *install.sh*, ce dernier modifie le script *makegcm*, mais que par ailleurs certaines lignes (sans conséquences) du script **makegcm** ont également changées d'une *release svn* à l'autre. Du coup `svn` ne sait pas quelles modifications conserver et vous demande de trancher.

²Le fait de devoir parfois lancer deux fois la commande **makegcm** afin de générer l'exécutable *gm.e* provient d'un léger défaut d'estimation des interdépendances entre fichiers sources du script.

3 Lancement d'un test de sensibilité

Ensuite, on change quelque chose dans un `.def`. On peut par exemple désactiver les paramétrisations de l'effet de l'orographie sous-maille en désactivant les clefs `ok_orodr` et `ok_orolf` dans `physiq.def`. On peut aussi changer la constante d'auto-conversion d'eau nuageuse en pluie avec la constante `cld_tau_lsc` dans `physiq.def`, la concentration du CO₂ atmosphérique, etc.

Relancer ensuite une simulation (dans un nouveau répertoire) et regarder les différences entre les résultats des deux simulations.

4 Modification des sorties

Les sorties du modèle sont contrôlées dans le fichier `config.def` par les lignes suivantes :

```
phys_out_filekeys=      y      y      n      y      n
phys_out_filenames=    histmth histday histhf  histins histLES
phys_out_filetimesteps= 5day   1day   1hr    6hr    6hr
phys_out_filelevels=   10     5     0     4     4
phys_out_filetypes=    ave(X) ave(X)  ave(X) inst(X) inst(X)
```

Avec ce réglage, vous avez sorti un fichier `histmth.nc` qui contient en fait une moyenne sur 5 jours et un fichier `histday.nc` contenant des moyennes journalières (5 états).

Vous pouvez regarder la liste des variables disponibles en exécutant la commande

```
ncdump -h histmth.nc | grep long_name
```

dans le répertoire où la simulation de test a tourné.

Vous pouvez aussi relancer la simulation en sortant quelques variables à plus haute fréquence (T2m, pression de surface, précipitation ...). Par exemple en modifiant les lignes :

```
phys_out_filekeys=      y      y      y      y      y
phys_out_filenames=    histmth histday histhf  histts  histhfm
phys_out_filetimesteps= 10day   1day   1hr    1ts    1hr
phys_out_filelevels=   10     5     0     0     0
phys_out_filetypes=    ave(X) ave(X)  inst(X) inst(X) ave(X)
flag_t2m=              10     10     1     1     1
flag_psol=             10     10     1     1     0
flag_precip=          10     10     0     1     1
```