



# *Les sorties dans LMDZ*

## Plan de l'exposé

- Généralités
- Fichiers «histoire»
- Fichiers «restart»
- Les fichiers de contrôle

**[http://www.lmd.jussieu.fr/~idelkadi/201112\\_Formation\\_LMDZ\\_Output.pdf](http://www.lmd.jussieu.fr/~idelkadi/201112_Formation_LMDZ_Output.pdf)**



# *Les sorties dans LMDZ*

## Généralités

On intègre dans les «sorties» du modèles :

- les fichiers «**histoires**» qui recueillent des variables diagnostiques instantanées ou moyennées à différentes fréquences
- les fichiers «**restart**» qui permettent de redémarrer ou étendre une simulation
- «le fichier d'output» recueillant toutes les impressions de contrôle et messages d'erreurs



# *Les sorties dans LMDZ*

## Généralités :

Les fichiers sorties «histoire» et «redémarrage» dans LMDZ sont tous au format **NetCDF** en utilisant soit directement la librairie **NetCDF** soit la librairie **IOIPSL**.

**IOIPSL** : c'est une librairie pour les entrées/sorties développée à l'IPSL, utilisant le format NetCDF.

En ce qui concerne les sorties «histoire», elle permet de faire des opérations (moyennisation par exemple) sur les diagnostics avant leur écriture.

La sortie comporte deux étapes :

- une étape de définition des diagnostics à sortir (lancée en début de simulation)
- une étape de calcul et d'écriture des diagnostics (qui se déroule au long de la simulation)

Comme aide au débogage, il existe aussi un mécanisme qui permet de sortir des variables au format **GrADS**.



# *Les sorties dans LMDZ*

## Généralités :

La séparation entre module dynamique et module physique, et le fait qu'on peut faire tourner (l'un sans l'autre) ==>

Les deux parties ont chacune leurs sorties «**histoire**» et sorties «**restart**» propres :

- «**restart.nc**» pour la dynamique et «**restartphy.nc**» pour la physique
- Les sorties «**histoire**» de la dynamique concernent uniquement les variables d'état (U, V, T, Q, Ps) et à deux fréquences de sorties : instantanées et moyennées.

Dans le cas d'une simulation parallélisée, chaque process sort un fichier «histoire» sur son domaine. Il est donc nécessaire de reconstruire un fichier «histoire» global à partir de ces différents domaines. L'utilitaire **rebuild** (disponible dans la librairie **IOIPSL**) permet cette reconstruction.



# *Les sorties dans LMDZ*

## Les sorties «histoire» physique :

- Mécanisme qui permet de contrôler individuellement la sortie des variables dans 5 fichiers de base + 1 spécifique :
  - **histmth.nc, histday.nc, histhf.nc, histins.nc, histLES.nc**
  - **histstn.nc**
- Il existe un certain nombre de fichiers sorties spécifiques :
  - **histmthNMC.nc, histdayNMC.nc, histhfNMC.nc**
  - **histmthCOSP.nc, histdayCOSP.nc, histhfCOSP.nc, histISCCP.nc**
  - **paramLMDZ\_phy.nc**
  - **histrac.nc**



# *Les sorties dans LMDZ*

## Les sorties «histoire» physique (Principe) :

- A chaque fichier histoire «histI.nc» prévu est associé un niveau de sortie, **lev\_histI**
- A chaque diagnostic «D» pouvant être «sorti» est associé un niveau de sortie, **flag\_D**  
**flag\_D ≤ lev\_histI ==>** le diagnostic «D» est défini et écrit dans le fichier «histI.nc»
- Ces clés de contrôle sont définies dans les fichiers **config.def/output.def**
- Chaque fichier peut être paramétré indépendamment pour :
  - Activer ou non sa sortie
  - Définir son nom,
  - Définir sa fréquence de sortie,
  - Définir si l'on sort sa valeur moyennée, instantanée, max ou min
  - Sortie sur tout le globe ou sur un domaine limité
  - ....



# Les sorties dans LMDZ

## Les sorties «histoire» physique (Pratique) :

Dans le fichier **config.def** ou **output.def**, pour :

- Définir les clés des différents fichiers :

**# Sortie ou non des fichiers :**

```
phys_out_filekeys=      y      y      n      y      n      n
```

**# Noms des fichiers :**

```
phys_out_filenames = histmth histday histhf histins histLES histstn
```

**# Niveaux de sorties**

```
phys_out_filelevels=   5      2      2      4      4      5
```

**# Opération sur variables**

```
phys_out_filetypes=  ave(X)  ave(X)  ave(X)  inst(X) inst(X) inst(X)
```

**# Fréquence de l'opération (mounths, mth, mois, day, days, jour, jours, heure, mn, TS, ...)**

```
phys_out_filetimesteps=30day  1day  6hr  3hr  6hr  1TS
```

- Définir les clés pour les champs à sortir :

**# Vertical wind**

```
flag_vitw      =      2      3      7      6      6      8
```

```
name_vitw      =      vitw
```



# Les sorties dans LMDZ

## Les sorties «histoire» physique (Pratique) :

Dans le fichier **config.def** ou **output.def**, pour :

# Noms des fichiers :

```
phys_out_filenames = histmth      histday      histhf      histins      histLES      histstn
```

# Activer ou non les sorties sur un domaine limité

```
reg_out_fkeys =          n          n          y          n          n          n
```

# Longitude min et max du domaine

```
phys_out_lonmin =      -180        -180         0         -180        -180        -180
```

```
phys_out_lonmax =      +180        +180         90         +180        +180        +180
```

# Latitude min et max du domaine

```
phys_out_latmin =      -90         -90         -30         -90         -90         -90
```

```
phys_out_latmax =      +90         +90         +40         +90         +90         +90
```

# Niveau vertical min et max

```
phys_out_levmin=       1           1           1           1           1           1
```

```
phys_out_levmax=      39          39          3           39          39          39
```

Nombre de variables gérées par ce mécanisme ~

**300 champs 2D et 150 champs 3D**





# Les sorties dans LMDZ

Les sorties «histoire» physique (Comment rajouter une sortie ?) :

Il faut intervenir dans deux fichiers :

- **libf/phylmd/phys\_output\_mod.F90** où :
  - on va définir le niveau de sortie et le nom du nouveau champs
  - puis le définir, au sens NetCDF, dans les différents fichiers histoire
- **libf/phylmd/phys\_output\_write.h** où est gérée :
  - l'écriture du champs (variable du code correspondant)
  - l'opération à effectuer sur la variable (moyenne, ...)

Exemple sur la variable «tsol» :

```
.../libf/phylmd/phys_output_mod.F90
```

```
type(ctrl_out),save :: o_tsol=ctrl_out((/ 1,1,1,5,10,10 /),'tsol') !Déclaration
```

```
CALL histdef2d(iff,clef_stations(iff),o_tsol%flag,o_tsol%name, &  
              "Surface Temperature","K") !Définitions dans le fichier
```

```
.../libf/phylmd/phys_output_write.h
```

```
IF (o_tsol%flag(iff)<=lev_files(iff)) THEN  
CALL histwrite_phy(nid_files(iff),clef_stations(iff),  
$                 o_tsol%name,itau_w,zxtsol)  
ENDIF !Ecriture de la variable
```



# Les sorties dans LMDZ

## Les sorties «histoire» physique (fichiers spécifiques) :

- Il existe plusieurs autres fichiers spécifiques de sortie «histoire» physique. Ils sont créés en utilisant IOIPSL de façon classique :

Pour chaque fichier «**hsthfNMC.nc**» :

- la création du fichier et la définition des champs ==> `"ini_hsthfNMC.h"`

- l'écriture des champs dans une routine ==> `"write_hsthfNMC.h"`

**hsthfNMC.nc histdayNMC.nc histmthNMC.nc** : sorties sur des niveaux de pression standards (1000., 925., 850., 700., .600., 500., 400., 300., 250., 200., 15000., 100., 70., 50., 30., 20., 10.hPa)

**paramLMDZ\_phy.nc** : paramètres utilisés dans la simulation et différents bilans

**histISCCP.nc histdayCOSP.nc hsthfCOSP.nc histmthCOSP.nc** : sorties simulateurs d'observables

**histrac.nc** : sorties traceurs

- Attention, la création de ces fichiers est la plupart du temps contrôlés par des clés CPP.

```
.../libf/physlmd/physiq.F :  
#ifdef histNMC  
#include "ini_hsthfNMC.h"  
.....  
#endif
```



# Les sorties dans LMDZ

Les sorties «histoire» physique (fichiers spécifiques) :

- **hismthNMC.nc, histdayNMC.nc, histhfNMC.nc** : sorties sur des niveaux de pression standards

Clés pour activer ou non la sortie de ces fichiers (dans **config.def**) :

- **ok\_histNMC = y y y**

- **histdayCOSP.nc histhfCOSP.nc hismthCOSP.nc** : sorties simulateurs d'observables COSP

- Clés pour activer ou non la sortie de ces fichiers (dans **config.def**)

**ok\_hismthCOSP= y # Sortir ou non le fichier hismthCOSP.nc**

**ok\_histdayCOSP= y # Sortir ou non le fichier histdayCOSP.nc**

**ok\_histhfCOSP = y # Sortir ou non le fichier histhfCOSP.nc**

- Clés pour activer ou non la sortie des diagnos (dans **cosp\_output\_nl.txt**) :

.....

**clcalipso = .false. !Lidar Cloud Fraction**

**cllcalipso = .true. !Lidar Low-level Fraction**

**clmcalipso = .true. !Lidar Mid-level Fraction**

**clhcalipso = .true. !Lidar High-level Fraction**

.....



# Les sorties dans LMDZ

## Les sorties «restart» :

Les modules dynamique et physique écrivent chacun un fichier de redémarrage ([start.nc](#) et [startphy.nc](#)) qui permet de redémarrer une autre simulation ou d'étendre celle que l'on vient de faire :

Ces fichiers sauvegardent toutes les variables dont le modèle a besoin d'un pas de temps sur l'autre, ceci afin de bien pouvoir redémarrer d'un état du modèle équivalent à l'état du modèle qui s'est terminé. (On parle en pratique de «**1+1=2**»)

D'un point de vue informatique, les routines impliquées sont :

- `.../libf/dyn3d/dynredem.F` pour la dynamique (la routine relisant l'état initial étant `.../libf/dyn3d/dynetat0.F`)
- `.../libf/phy1md/phyredem.F` pour la physique (avec la routine correspondante `.../libf/phy1md/phyetat0.F`)

Ces routines n'utilisent pas la bibliothèque **IOIPSL**, et «attaquent» directement la librairie **NetCDF**.



# Les sorties dans LMDZ

## Les sorties «de contrôle» :

La majeure partie des sorties de contrôle s'imprime sur la sortie standard, suite à des `print*`, ... ou des `write(*,*) 'ma variable =', ..` Fortran.

Un mécanisme est prévu pour limiter les impressions de contrôle et pour qu'elles se retrouvent dans un fichier plutôt qu'à l'écran. Il suffit :

- d'inclure dans toute nouvelle routine, le fichier **iniprint.h** qui définit et met en commun 2 paramètres :
  - **Lunout** : un numéro d'unité correspondant à un fichier de sortie  
(`lunout ≠ 6` ==> un fichier `lmdz.out` est créé et assigné à ce numéro)
  - **prt\_level** : un niveau de sortie

La valeur de ces deux paramètres peut ensuite être modifiée dans le fichier **run.def**

- d'utiliser ensuite dans la routine où l'on veut sortir des impressions de contrôle, des lignes du type :

```
IF (prt_level>9) WRITE(lunout,*) 'pas de convection'
```

en gardant les petites valeurs de **prt\_level** pour les messages vraiment importants

Les sorties de vérification des fichiers \*.def utilisés (`run.def`, `physiq.def`, `config.def`, `gcm.def`, ...) : Mécanisme de IOIPSL permettant d'écrire les différents paramètres effectivement lus et utilisés dans le fichier **used\_run.def**



## *Les sorties dans LMDZ*

Perspective :

**XIOS (XML-IO-SERVER)** : Librairie en XML pour gérer les entrées/sorties dans le modèle.

- Intègre les mêmes fonctions que IOIPSL
- Plus facile à utiliser, ...