

Programmation des paramétrisations physique facilitant mutualisation et tests

Frédéric Hourdin, Nicolas Nanou, Laurent Fairhead, Remi Kazeroni, Yann Meurdesoif, Patryk Kiepas (LMD/LSCE/IPSL/CNRS/SU)

19 mars 2026

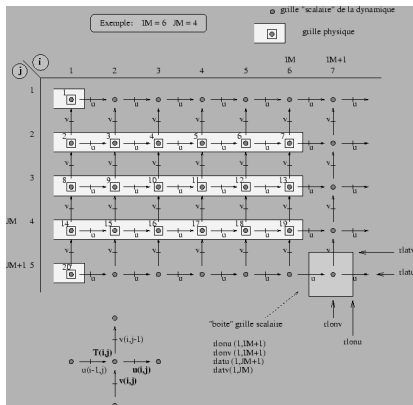
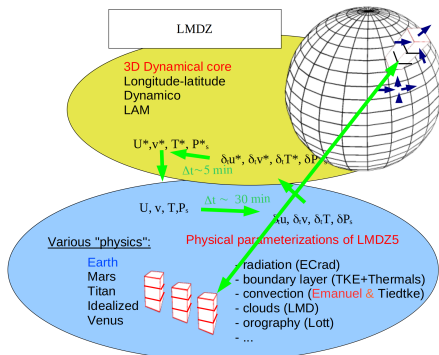
1. Séparation des mondes physique et dynamique
2. Le monde des paramétrisations et l'outil replay
3. GPUmorphosis
4. Un installateur de configuration enseignement/recherche

1. Séparation des mondes physique et dynamique

Pensée dès les premières versions planétaires (1989)

développées sur le "nouveau cœur dynamique" (1985) :

- fichiers séparés start.nc/dynamique startphy.nc/physique
- tableaux et dimensions différentes pour les variables communes
- ne pas démultiplier le point du pôle



1. Séparation des mondes physique et dynamique

pas gratuit :

Fichiers [limit.nc](#) et [startphy.nc](#) pas simple à tacer ou à modifier.

Pas d'échange possible avec les voisins.

Moyenne spatiale des champs u et v aux points scalaires.

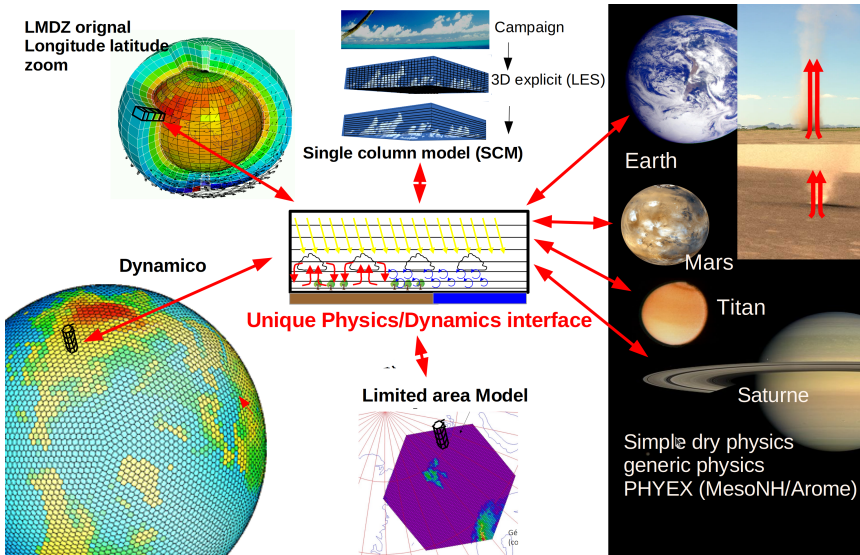
mais :

a forcé une séparation effective totale entre la structure des tableaux entre la dynamique et la physique.

Rôle déterminant pour :

- le multi planète
- le multi dynamiques (3D, 2D longitude-latitude, 1D)
- la parallélisation
- le portage de dynamico (+LAM)
- la GPUisation

1. Séparation des mondes physique et dynamique



2. Le monde des paramétrisations et l'outil replay

Séparation de la physique en 2 mondes motivée par :

- la perspective de la GPUisation
- le souhait de pouvoir sortir les variables internes d'une paramétrisations pendant l'exécution

Le monde des paramétrisations individuelles

Un ensemble de `modules paramX_modA`, `paramX_modB` ...

`use` d'un unique module `paramX_ini` contenant des constantes ...

Les `subroutine` n'ont comme interface que les arguments (`IN/OUT/INOUT`) et le `use paramX_ini`.

`param_ini` n'a comme interface que les arguments

Fortran old school ; boucles explicites ; lisibilité des calculs

Le monde de LMDZ

Appel aux paramétrisations, IO, couplage avec la dynamique, lecture des états initiaux et conditions aux limites ...

2. Le monde des paramétrisations et l'outil replay

`Replay_equip.sh`, script de 650 lignes

Equipe le code d'écriture automatique :

- variables `IN` de l'initialisation → `lmdz_thermcell_ini.bin`
- variables `IN` et `INOUT` de la param → `dump_param_out.bin`
- toutes les variables déclarées → `phys.nc`

Crée un programme `replay1d` :

- relit `lmdz_thermcell_ini.bin` et appelle l'initialisation de la param
- relit en boucle `dump_param_out.bin` et appelle la param
- écrit toutes les variables déclarées → `phys.nc`

Utilisable pour toute paramétrisation ayant été codées en respectant des règles assez simples.

2. Le monde des paramétrisations et l'outil replay

Exemple de la paramétrisation des panaches thermiques

```
MODULE lmdz_thermcell_ini
[...]
```

save

```
real, protected :: r_aspect_thermals=2. ! Aspect ratio for
    thermal celles
```

```
real, protected :: RG,[...]
```

```
[...]
```

CONTAINS

```
SUBROUTINE thermcell_ini([...],RG_in,[...]
```

```
[...]
```

```
RG=rg_in
```

```
[...]
```

```
CALL getin_p('r_aspect_thermals',r_aspect_thermals)
```

2. Le monde des paramétrisations et l'outil replay

Calcul des propriétés du panache

```
subroutine thermcell_main(ngrid,nlay,[...],p_o,ptemp_env,[...]  
[...]  
    real, intent(in), dimension(ngrid,nlay)    :: p_o  
[...]  
    USE lmdz_thermcell_ini, ONLY: RD,RG  
[...]  
    ! deltatw : temperature difference between wake and off-wake  
    ! regions  
    ! dtdwn: source de chaleur due aux descentes (K/s)  
[...]  
    real, intent(in), dimension(ngrid,nlay)    :: p_o  
    real, intent(out), dimension(ngrid,nlay+1) :: fm0,zw2,frac
```

2. Le monde des paramétrisations et l'outil replay

Inclusion de l'écriture des variables IN de l'initialisation

```
open(90,file='lmdz_thermcell_ini.bin',form='unformatted')
write(90) [...],RG_in,RD_in,[...]
close(90)
```

Ecriture des variables IN et INOUT de la param

```
if (first_replay) then
[...]
    open(81,file='dump_param_out.bin',form='unformatted',access='
    direct',recl=rec_length_replay) ! replay automatic include
endif
irec=irec+1
write(81,rec=irec) ngrid,nlay,[...],p_o(1:ngrid,:),[...]
endif
```

```
call iotd_ecrit_seq('p_o',nlay,'p_o in thermcell_main',' ',p_o(1:
ngrid,:))
call iotd_ecrit_seq('fm0',nlay,'fm0 in thermcell_main',' ',fm0(1:
ngrid,:))
```

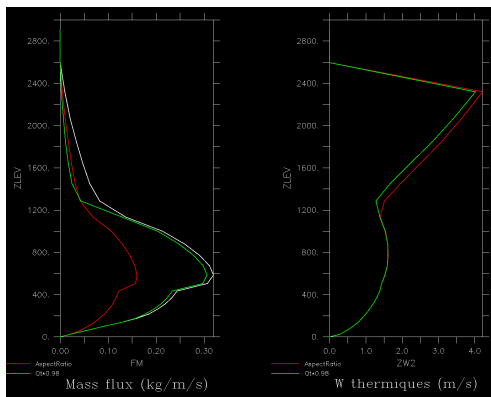
2. Le monde des paramétrisations et l'outil replay

Exemple d'utilisation de Replay

Cas ARM cumulus à 14h LT

Replay : Rapport d'aspect 4 versus 2

Replay : -2% d'eau totale en entrée



2. Le monde des paramétrisations et l'outil replay

Perspectives de l'outil replay

- Pouvoir décortiquer le fonctionnement interne de chaque partie du code
- Tester des modifications des paramétrisations sans couplage avec le reste
- Tester la sensibilité de la paramétrisations off-line
- Tuner en mode LES versus LES, en remplaçant les variables d'entrées lues dans `dump_param_out.bin` par des sorties de la LES

Mutualisation

Les paramétrisation “réplaysées” sont très facilement mutualisables

3. GPUmorphosis



Secrétariat général pour l'investissement



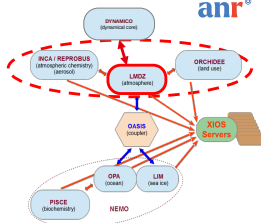
Traduction source-à-source pour le portage GPU

➤ Approche

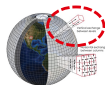
- Exploiter l'**indépendance des calculs en colonne** pour refactoriser les routines de calcul avec un outil de transformation automatique de code
- Développements techniques (portage GPU, optimisation, ...) parallèlement aux développements scientifiques : «**separation of concerns**»
- **Traduction automatique** d'une version de référence CPU d'un modèle en des versions adaptées aux calculs sur accélérateurs pour différentes architectures

➤ Cible initiale : physique atmosphérique de LMDZ

- ✓ Développement d'un outil de traduction source-à-source, **GPU-Morphosis**, d'une **méthodologie de portage incrémental et de sa validation**
- ✓ Application de GPU-Morphosis aux **paramétrisations physiques de LMDZ** après travail de réarchitecture du code Fortran
- ❑ **Objectif** : couplage DYNAMICO-GPU + LMDZ-GPU (sans sorties) => **run atmosphère seule sur GPU**



IPSL ESM MODEL : IPSL-CM7



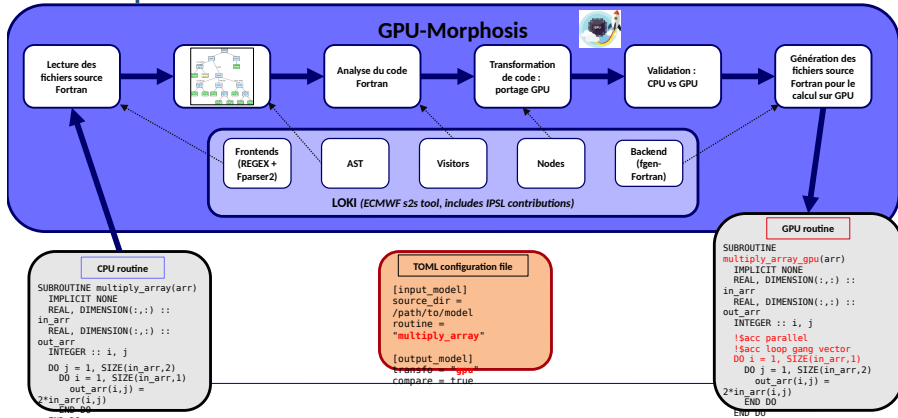
3. GPUmorphosis



Secrétariat général pour l'investissement



GPU-Morphosis - fonctionnement



4. Un installateur de configuration enseignement/recherche

```
./planet.sh -h
Use : ./planet.sh PLANET                                Default : bench
      [--grid|-g IMxJMxLM]                             Default : 32x32x
      [--nday|-n number_of_days]                       Default : 3
      [--parallel 0/1]                                  Default : 0
      - Concerning the zoom -
      [--lon0|-x zoom_center_longitude]                 Default : -120
      [--lat0|-y zoom_center_latitude]                 Default : 0
      [--ext_lon|-dx zoom_extension_longitude]         Default : 120
      [--ext_lat|-dy zoom_extension_latitude]         Default : 60
      [--fact_zoom|-f zoom_factor]                    Default : 1
      [--fact_zoomx|-fx zoom_factor_in_x]              Default : 1
      [--fact_zoomy|-fy zoom_factor_in_y]              Default : 1
```

PLANET among

```
bench : bench for installation. 32x32x39. Coming with start files
tuto  : the simplest one |
clim  : Climatological SSTs |
aqua  : aqua planet (SST function of latitude) | From models/gcm
terra : global continent (simple soil) |
slab  : With a slab ocean |
simple : 20 parameter physics
dev   : teaching parameterization development
SCM[:case1,case2,...] : 1D configurations, with optional cases specif.
```

Conclusions

Général

- Importance des questions informatiques
- Puissance du “source to source”

Replay

- Formalisation de deux mondes dans la physique
- Nombreuses perspectives

GPUmorphosis

- permet un portage non invasif
- nécessite un nettoyage nécessaire et profitable
- toutes les paramétrisations des LMDZ sont portées
- chantier en cours sur la réécriture des moniteurs physique
- nouvel outil replay
- possibilité d'apprendre et utiliser des émulateurs (type réseaux de neurones) en remplacement des routines