User Manual for the LMD Martian Atmospheric General Circulation Model

L'équipe modélisation du LMD
(LMD)

5 avril 2004

Attention : ce document, adapté de la documentation du modèle martien, est en cours de mise à jours ! ! !

Table des matières

Chapitre 1

Introduction

Ce document est un manuel d'utilisation du modèle de circulation générale atmosphérique développé par le Laboratoire de Météorologie Dynamique du CNRS, et utilisé comme composante atmosphérique du modèle intégré du système terre de l'IPSL. Il correspond à la version du modèle diffusée à partir de avril 2004, LMDZ4.

Il correspond, par rappor à la version précédente LMDZ.3.3, à une fusion des branches "couplée" (à l'océan et à la végétation) et "traceurs" (couplée au module de chimie INCA).

Le chapitre 2 du présent document, à lire avant tout autre, décrit l'essentiel du modèle. Le modèle est partagé entre deux parties relativement indépendantes : (1) Le code hydrodynamique, commun a toutes les atmosphère (Terre, Mars, etc...) qui intègre dans le temps et sur la sphère les équations de la mécanique des fluides, et (2) un jeu de paramétrisations physiques martiennes, qui inclut par exemple le calcul du transfert radiatif dans l'atmosphère et du mélange turbulent dans la couche limite.

Le chapitre 3 contient une brève description scientifique du code hydrodynamique. Il est suivi (Chapitre 4) en cours d'élaboration sur les paramétrisations physiques.

Ce document contient ensuite une description de l'écriture informatique du modèle accompagnée d'un manuel d'utilisation informatique pour la compilation et pour l'exécution du modèle (Chapitre 5)

Le chapitre 6 décrit la description des Entrées-sorties du modèle avec en entrée les fichiers nécessaires à l'initialisation du modèle (état de l'atmosphère à l'instant $t0$ plus un jeu de conditions aux limites) et en sortie, les "fichiers histoires", archives de l'histoire de l'écoulement atmosphérique telle qu'elle est simulée par le modèle, les "fichiers diagfi", ..., les "fichiers stats" , moyennes diurnes ... Les moyens d'éditer ou de visualiser ces fichiers (l'éditeur "ncdump" ou le logiciel graphique "grads") sont aussi décrits.

Enfin, pour un **premier contact avec le modèle** (voir section 7), une dernière partie permet de guider l'utilisateur dans une première simulation (choix des conditions initiales et des paramètres, et visualisation des fichiers de sortie).

Chapitre 2

Aperçu du modèle

2.1 Principes de base

Le modèle de circulation générale (GCM ou MCG) calcule l'évolution temporelle de diverses **variables** (liste ci-dessous) contrôlant ou décrivant la météorologie et le climat martien en différents points d'un **"maillage" 3D** (voir ci-dessous) qui couvre l'ensemble de l'atmosphère.

A partir d'un état initial, le modèle calcule l'évolution de ces variables, pas de temps après pas de temps :

– A l'instant t, on connait la variable X_t (par exemple la température) en un point de l'atmosphère.

– On calcule l'évolution (les **tendances**) $(\frac{\partial X}{\partial t})_1$, $(\frac{\partial X}{\partial t})_2$, etc... due à chaque phénomène physique, calculé par une **paramétrisation** de chacun de ces phénomènes (par exemple le chauffage due a l'absorption du rayonnement solaire).

– Au pas de temps suivant $t + \delta t$, on calcule $X_{t+\delta t}$ à partir de X_t et des $(\frac{\partial X}{\partial t})$. C'est l'**"integration"** des variables dans le temps. (Par exemple, $X_{t+\delta t} = X_t + \delta t(\frac{\partial X}{\partial t})_1 + \delta t(\frac{\partial X}{\partial t})_2 + ...$)

L'essentiel du modèle consiste donc à calculer les tendances $(\frac{\partial X}{\partial t})$ **dues aux divers phénomènes paramétrés**.

2.2 Séparation Dynamique-Physique

En pratique, le modèle 3D fonctionne en deux parties :
- d'une part la **partie dynamique** contenant la résolution numérique des équations générales de la circulation atmosphérique. Cette partie est (y compris au niveau informatique) commune au modèle terrestre et martien, et plus généralement à toutes les atmosphères de type terrestre.
- d'autre part la **partie physique** propre à la planète considérée, qui calcule le forçage de la circulation et le détail du climat en chaque point.

Les calculs concernant la partie dynamique sont réellement effectués sur une grille 3D avec des échanges horizontaux entre mailles, alors que la partie physique peut être vue informatiquement comme une juxtaposition de "colonnes" d'atmosphère n'interagissant pas entre elles (schéma 2.1).

Les parties dynamiques et physiques traitent des variables de nature différente, sur une grille informatiquement différente. L'intégration temporelle des variables se fait selon un schéma numérique différent (simple comme ci-dessus pour la physique, plus élaboré, Schéma "Matsuno-Leapfrog", pour la dynamique. Les pas de temps sont aussi différents. Le pas de temps physique est I_{physiq} fois plus long que le pas de temps dynamique, car la

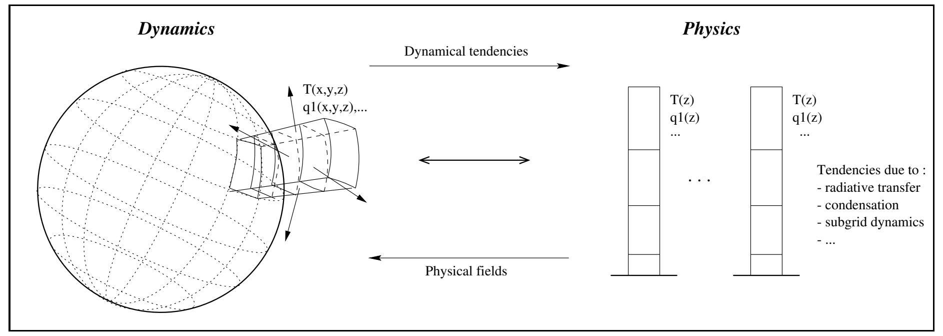


FIG. 2.1 – Interface physique/dynamique

résolution des équations de la dynamique réclame un pas de temps plus court que le calcul du forçage par la physique.

En pratique, le programme principal (`gcm.F`) effectue un certain nombre d'initialisation (lecture des fichiers de configuration, lecture de l'état initial, calculs realtifs à la définition de la grille, etc.) puis appeller le sous-programme (`leapfrog.F`), qui gère l'intégration temporelle du modèle. Lors du déroulement du calcul de l'évolution temporelle, à chaque pas de temps dynamique, ce programme effectue les appels suivants :

1. Appel du sous-programme qui gère le calcul des tendance totales ($\frac{\partial X}{\partial t}$) due à la dynamique (`caldyn.F`)
2. Intégration de ces tendances dynamiques pour calculer l'évolution des variables au pas de temps suivants (sous-programme `integrd.F`)
3. Tous les I_{physiq} pas de temps dynamique, appel du sous-programme d'interface (`calfis.F`) avec le modèle physique (`physiq.F`), qui calcule l'évolution de certaines variables purement physiques (ex : Tsurf) et retourne les tendances ($\frac{\partial X}{\partial t}$) dues à la physique
4. Intégration des variables physiques (sous-programme `addfi.F`)
5. De même, calcul puis intégration des tendances dues à la dissipation horizontale, de la "sponge layer", etc... (cf. section suivante).

Remarque : On peut faire tourner la partie physique seule pour un calcul 1-D pour une seule colonne avec le programme `testphys1d.F`.

2.2.1 Grille horizontale

On utilise pour chaque partie une grille différente. On a représenté figure 2.2 la numérotation des grilles physiques et dynamiques ainsi que les différentes positions possibles des variables sur ces grilles. Afin de repérer les coordonnées d'une variable (sur un point de grille, au dessus, en dessous, à droite, ou à gauche), on utilise les coordonnées **rlonu, rlatu, rlonv et rlatv** (longitude et latitude, en radian).

Pour la grille dynamique, on répète en i=IM+1 les valeurs de i=1 (périodicité en longitude). Les valeurs aux pôles sont quant à elles dupliquées IM+1 fois. En revanche, pour la grille physique, il n'y a qu'une valeur aux pôles et pas de périodicité en longitude. En pratique, les calcul se font pour une suite de NGRID colonnes atmosphériques avec NGRID=IM$\times$(JM-1) +2

2.2.2 Maillage vertical

Le GCM a été écrit originellement sur des coordonnées $\sigma = p/p_s$ qui présentent l'avantage d'avoir un domaine constant (1 à la surface et 0 au sommet de l'atmosphère) quelque

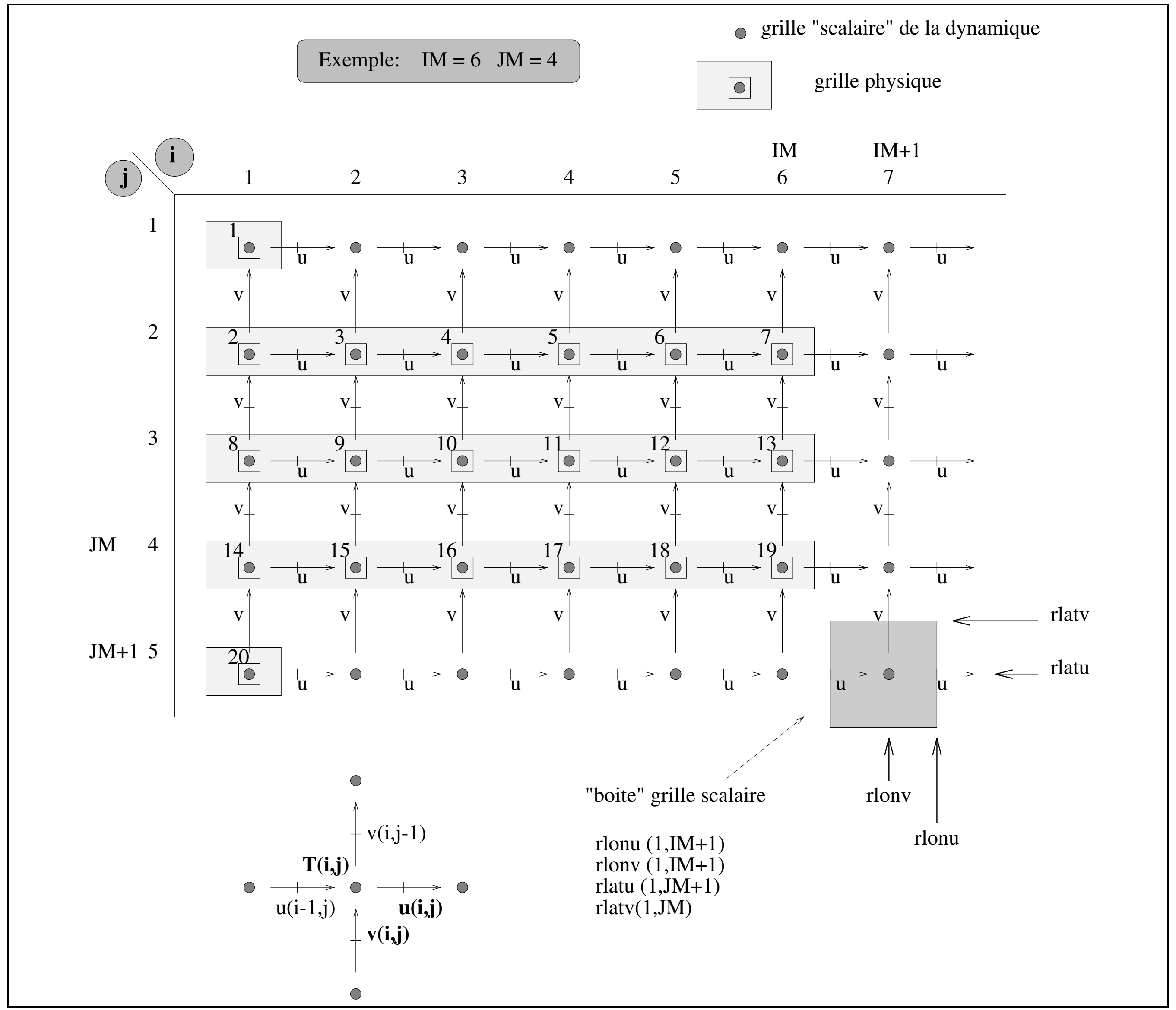


FIG. 2.2 – Grilles dynamique et physique pour une resolution horizontale 6 × 7. Dans la dynamique (mais pas dans la physique) Les vents u et v sont sur une grille dynamique décalée. Les autres variables sont sur la grille "scalaire" dynamique. La physique utilise cette même grille scalaire pour toutes les variables, sauf que les points sont indexés en 1 seul vecteur contenant NGRID=2+(JM-1)×IM points en comptant à partir du pôle nord. NB. Dans le programme fortran, on utilise les variables suivantes : `iim=IM` , `iip1=IM+1`, `jjm=JM` , `jjp1=JM+1`.

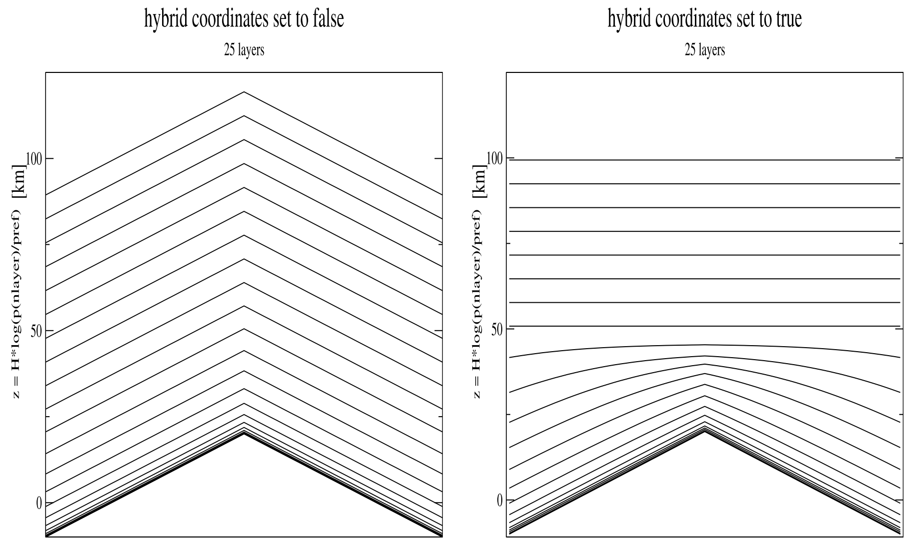


FIG. 2.3 – Comparaison entre coordonnées non hybrides et hybrides

soit le relief sous-jacent. Pourtant, il est vite apparu que de telles coordonnées perturbent de façon significative la représentation de la dynamique stratosphérique dans la mesure òu le relief est visible dans le système de coordonnées jusqu'au sommet du modèle. Une solution élégante a été trouvée à ce problème : l'emploi de coordonnées hybrides équivalentes σ près de la surface et p plus haut. La figure 2.3 illustre l'interêt de l'utilisation des coordonnées hybrides comparée à l'utilisation de coordonnées classiques. La répartition des couches verticales est irrégulière, afin de permettre notamment une plus grande précision au niveau du sol. On décrit généralement à l'aide de 25 couches l'atmosphère jusqu'à une hauteur de l'ordre de 80 km, ou 32 couches pour des simulations jusqu'à 120 km. La première couche décrit alors les quelques premiers mètres au dessus du sol, tandis que les dernières décrivent plusieurs kilomètres. La figure 2.4 décrit les variables du maillage vertical.

La figure 2.4 décrit les variables du maillage vertical.

2.3 Variables du modèle

2.3.1 Partie dynamique

Les variables d'états dynamiques sont la température atmosphérique, la pression au sol, les vents, les concentrations des traceurs. En pratique, la formulation choisie pour résoudre les équations de la dynamique (cf. chapitre 3) est optimisée en utilisant les variables moins "naturelles" suivantes :

- **theta** la température potentielle, reliée à **T** la temperature par $\theta = T(P/Pref)^{-\kappa}$ avec $\kappa = R/C_p$ (notez que κ est nommée `kappa` dans le code dynamique, et `rcp` dans le code physique). On prend Pref=610Pa sur Mars.
- **ps** la pression au sol.
- **masse** la masse d'atmosphère dans chaque maille.

```
DYNAMIQUE                                              PHYSIQUE
---------                                              --------
(variables ap bp)                                      (niveaux de pression)


ap(llm+1)=0,bp(llm+1)=0  ***********************      plev(nlayer+1)=0

aps(llm),bps(llm)        .. llm-1 ........  nlayer    play(nlayer)

ap(llm),bp(llm)          ***********************      plev(nlayer)

aps(llm-1),bps(llm-1)    .. llm-1 ........  nlayer-1  play(nlayer-1)

ap(llm-1),bp(llm-1)      ***********************      plev(nlayer-1)
                                  :
                                  :
                                  :
                                  :

aps(2), bps(2)           ..... layer 2 ...........    play(2)

ap(2), bp(2)             ************************     plev(2)

aps(1), bps(1)           ..... layer 1 ...........    play(1)

ap(1)=1                  ******** surface *******     plev(1)= Psurf
```

1

FIG. 2.4 – Description du maillage vertical des `llm = nlayer` couches atmosphériques dans le code informatique (`llm` est la variable utilisée dans la partie dynamique et `nlayer` est dans la partie physique). Les variables `ap, bp` et `ap_s bp_s` indiquent les niveaux hybrides aux niveaux inter-couches et au milieu des couches, respectivement. La pression à l'intercouche est Play(l)=ap(l)+bp(l) $\times$ Ps et la pression au milieu de la couche est definie par Plev(l)=aps(l)+bps(l) $\times$ Psurf, (avec ici `Psurf` la pression au sol). En coordonnées non hybrides (en fait un cas particulier des coordonnées hybrides), aps=0 et bps=P/Psurf. En coordonnées hybrides, bps=0 à partir de 50km. On peut choisir de faire tourner le modèle en coordonnées hybrides ou non en mettant la variable `hybrid` de run.def à True ou False.

- **ucov** et **vcov** les vents zonal et meridional covariants. Ces variables sont reliées aux vents "naturels" par `ucov = cu * u` et `vcov = cv * v`, avec `cu` et `cv` des constantes qui ne dépendent que de la latitude (cf. Annexe A).
- **q01, q02**, etc... le rapport de mélange des traceurs dans l'atmosphère (typiquement kg/kg).

ucov et **vcov**, variables "vectorielles", sont stockées respectivement sur les grilles u et v "décalées" de la dynamique (voir section 2.2). **theta**, **q01**, **ps**, **masse** variables scalaires, sont stockées sur la grille dite "scalaire" de la dynamique.

2.3.2 Partie physique

Dans la physique, les variables d'états de la dynamique sont transmises via une interface qui interpole les vents sur la grille scalaire (qui correspond à la grille physique) et transforme les variables dynamiques en variables plus "naturelles". On retrouve ainsi les vents **u** et **v** ($m.s^{-1}$), la température **T** (K), le champ de pression au milieu des couches **play** et aux niveaux intercouches **plev** (Pa), les traceurs **q01, q02**, etc... (kg/kg).

De plus, la partie physique gère en plus l'évolution de variables d'états internes comme la température du sol, la couverture de neige, ... Ces variables ne sont jamais remontées dans le moniteur dynamique.

2.3.3 Les traceurs

Par défaut, dans le modèle terrestre, les deux premiers traceurs sont l'eau vapeur et l'eau liquide. Le nom des traceurs peut etre modifié au moyen du fichier traceur.def **A completer**

Chapitre 3

Code Dynamique 3-D

Nul besoin de lire cette partie technique pour travailler avec le GCM !

3.1 Discrétisation des équations de la dynamique

Extrait de la note de Robert Sadourny, Phu Le Van et Frédéric Hourdin, Laboratoire de Météorologie Dynamique.

Le modèle climatique du LMD est bâti, comme tous les modèles de circulation générale atmosphérique, sur la résolution numérique des équations primitives de la météorologie décrites dans de nombreux ouvrages (6). L'analyse présentée ici a été menée sur la nouvelle version de la dynamique du LMD écrite par Phu Le Van (11) sur une formulation de Robert Sadourny. Cette formulation diffère de l'ancienne essentiellement par deux points : dans la nouvelle formulation, la répartition des points en longitude et en latitude peut être changée arbitrairement. L'autre modification porte sur la répartition des points aux pôles[1].

La coordonnée verticale du modèle est la pression normalisée par sa valeur à la surface : $\sigma = p/p_s$. On utilise en fait σ aux niveaux inter-couches et $s = \sigma^\kappa$ au milieu des couches. On note X et Y les coordonnées horizontales :

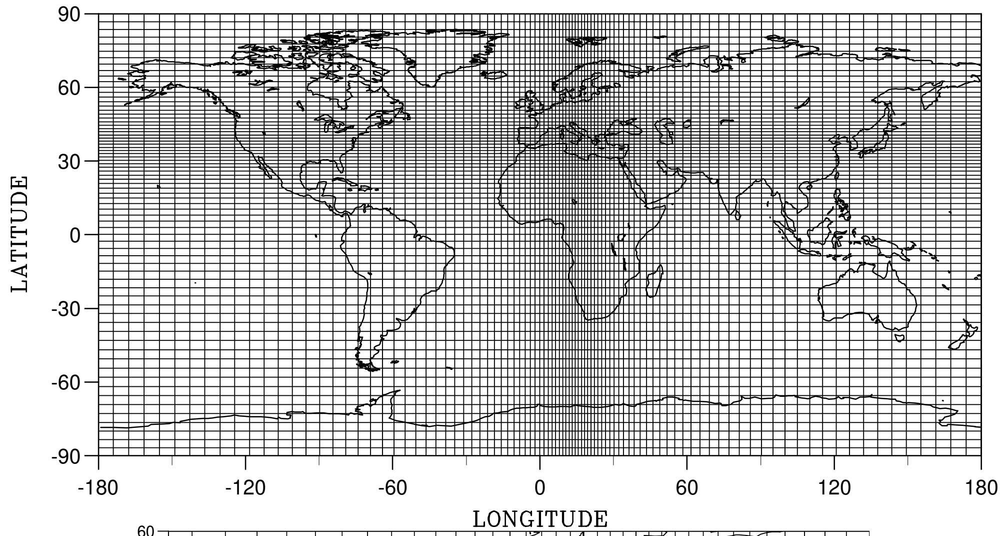
X (resp. Y) est une fonction biunivoque de la longitude λ (resp. de la latitude ϕ). Ces deux fonctions peuvent être choisies de façon arbitraire dans le modèle LMDZ ce qui permet d'effectuer un zoom sur une région du globe particulière. Une grille de ce type est montrée sur la Figure 3.1. Les variables scalaires (température potentielle $\theta = c_p T/{p_s}^\kappa$, géopotentiel Φ et pression de surface p_s) sont évaluées aux points correspondant à des couples de valeurs entières $(X, Y) = (i, j)$. Les variables dynamiques sont décalées par rapport aux variables scalaires en utilisant une grille C dans la définition de Arakawa (**?**) : le vent zonal est calculé aux points $(X, Y) = (i + 1/2, j)$ et le vent méridien aux points $(X, Y) = (i, j + 1/2)$. La disposition des variables sur la grille est illustrée sur la Figure 3.2.

On utilise en fait les composantes covariantes ($\tilde{u}$ et $\tilde{v}$) et contravariantes ($\tilde{\tilde{u}}$ et $\tilde{\tilde{v}}$) du vent définies par

$$\begin{array}{llll} \tilde{u} = c_u u & \text{et} \quad \tilde{\tilde{u}} = u/c_u & \text{avec} & c_u = a \cos\phi \, (d\lambda/dX) \\ \tilde{v} = c_v v & \text{et} \quad \tilde{\tilde{v}} = v/c_v & \text{avec} & c_v = a \, (d\phi/dY) \end{array} \tag{3.1}$$

où u et v sont les composantes physiques du vecteur vent horizontal. On introduit également :

[1]Aux pôles sont calculés : le vent méridien dans l'ancienne formulation et les variables scalaires dans la nouvelle.



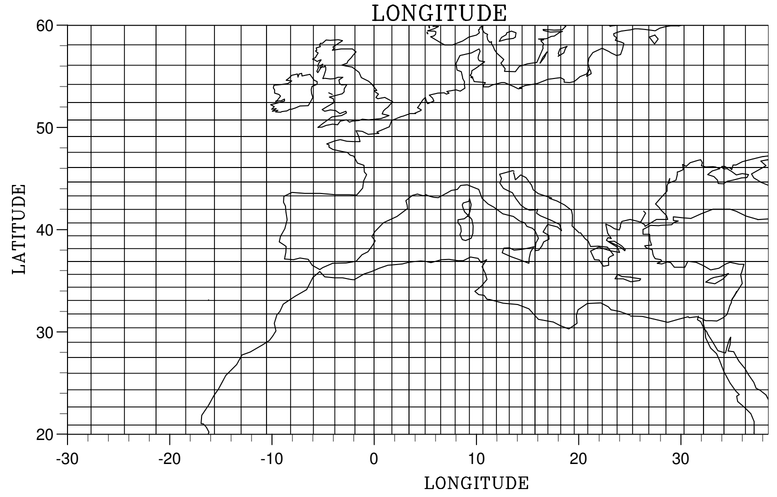


FIG. 3.1 – Grille obtenue avec 96 points en longitude et 73 en latitude et un zoom d'un facteur 3 centré sur la méditérannée (grille utilisée au laboratoire par Ali Harzallah)

.

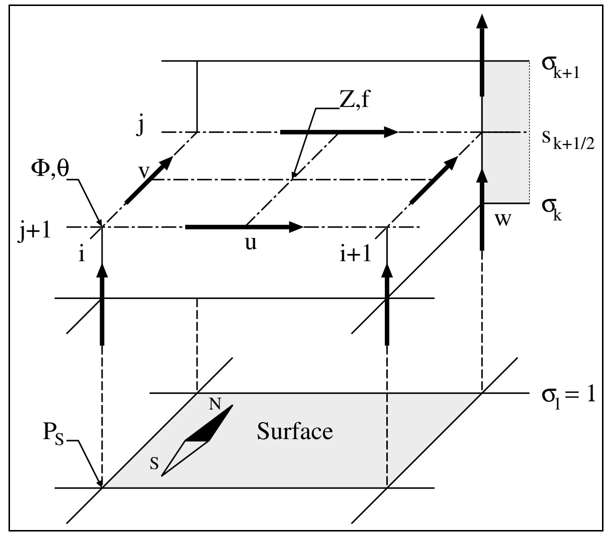


FIG. 3.2 – Disposition des variables dans la grille du LMD

la pression extensive :

$\tilde{p}_s$ (pression au sol multipliée par l'aire de la maille).

les trois composantes du flux de masse :

$$U = \overline{\tilde{p}_s}^X \tilde{\tilde{u}},\ V = \overline{\tilde{p}_s}^Y \tilde{\tilde{v}} \text{ et } W = \tilde{p}_s \dot{\sigma} \text{ avec } \dot{\sigma} = \frac{d\sigma}{dt} \tag{3.2}$$

le facteur de Coriolis multiplié par l'aire de la maille :

$f = 2\Omega \sin \phi c_u c_v$
où Ω est la vitesse de rotation de la planète.

la vorticité potentielle absolue :

$$Z = \frac{\mathcal{F}\left(\delta_X \tilde{v} - \delta_Y \tilde{u}\right) + f}{\overline{\tilde{p}_s}^{X,Y}} \tag{3.3}$$

l'énergie cinétique

$$K = \frac{1}{2}\left(\overline{\tilde{u}\tilde{\tilde{u}}}^X + \overline{\tilde{v}\tilde{\tilde{v}}}^Y\right) \tag{3.4}$$

La notation δX signifie simplement qu'on effectue la différence entre deux points consécutifs suivant la direction X. La notation $\overline{a}^X$ signifie qu'on prend la moyenne arithmétique de la quantité a suivant la direction X. $\mathcal{F}$ est un filtre longitudinale appliqué dans les régions polaires. Les équations discrétisées sont écrites sous la forme suivante :

équations du mouvement :

$$\frac{\partial \tilde{u}}{\partial t} - \overline{Z}^Y \overline{V}^{X,Y} + \delta_X \mathcal{F}\left(\Phi + K\right) + s\overline{\theta}^X \delta_X \mathcal{F}\left({p_s}^\kappa\right) - \frac{\overline{\tilde{u}_a}^{Y,Y} \delta_Z \overline{W}^X}{\overline{\tilde{p}_s}^X \delta_Z \sigma} + \frac{\delta_Z\left(\overline{W}^X \overline{\tilde{u}_a}^Z\right)}{\overline{\tilde{p}_s}^X \delta_Z \sigma} = S_{\tilde{u}} \tag{3.5}$$

oú $\tilde{u}_a$ est la composante zonale covariante du vecteur vent absolu : $\tilde{u}_a = \tilde{u} + c_u a\Omega\cos\phi$ et

$$\frac{\partial \tilde{v}}{\partial t} + \overline{Z}^X \overline{U}^{X,Y} + \delta_Y \mathcal{F}\left(\Phi + K\right) + s\overline{\theta}^Y \delta_Y \mathcal{F}\left({p_s}^\kappa\right) - \frac{\overline{\tilde{v}}^{X,X} \delta_Z \overline{W}^Y}{\overline{\tilde{p}_s}^Y \delta_Z\sigma} + \frac{\delta_Z\left(\overline{W}^Y \overline{\tilde{v}}^Z\right)}{\overline{\tilde{p}_s}^X \delta_Z\sigma} = S_{\tilde{v}} \tag{3.6}$$

équation thermodynamique :

$$\frac{\partial\left(\tilde{p}_s\theta\right)}{\partial t} + \mathcal{F}\left[\delta_X\left(\overline{\theta}^X U\right) + \delta_Y\left(\overline{\theta}^Y V\right)\right] + \frac{\delta_Z\left(\overline{\theta}^Z W\right)}{\delta_Z\sigma} = S_\theta \tag{3.7}$$

équation hydrostatique :

$$\delta_Z\Phi = -p_s^\kappa \overline{\theta}^z \delta_Z s \tag{3.8}$$

équations de continuité :

$$\frac{\partial p_s}{\partial t} = \mathcal{F}\left[\sum_z \delta_Z\sigma\left(\delta_X U + \delta_Y V\right)\right] \tag{3.9}$$

$$\delta_Z W = -\delta_Z\sigma\left[\mathcal{F}\left(\delta_X U + \delta_Y V\right) + \frac{\partial p_s}{\partial t}\right] \tag{3.10}$$

On a noté S les termes sources dans les différentes équations. Dans ces termes sources, on distingue 1) d'une part les paramétrisations physiques mentionnées plus haut et qui font intervenir pour une maille donnée du modèle, tous les points situés sur une même verticale mais ceux-là seulement ; 2) les opérateurs de dissipation horizontale, censés rendre compte des échanges entre échelles explicitement représentées dans le modèle et échelles sous-mailles. Ces opérateurs ont la structure de Laplaciens agissant sur des plans horizontaux c'est à dire qu'il font intervenir un voisin de chaque côté dans les deux directions horizontales. Cet opérateur est généralement itéré pour le rendre plus sélectif en échelle (plus on itère un laplacien et plus son effet sur les petites échelles devient important relativement).

3.2 Filtres aux hautes latitudes

Extrait adapté de Forget et al. [1999]

At high latitude a filter is applied near the singularity in the grid at the pole in order to satisfy the Courant-Friedrichs-Lewy numerical stability criterion without going to an excessively small timestep. In the original version of the dynamical code a classical Fourier filter was used, but we found that because the Martian polar atmosphere appears to be much more dynamically unstable than the Earth's polar atmosphere, a more efficient formulation (based on the grouping of adjacent gridpoints together) was necessary to avoid numerical instability.

En pratique, la technique suivante est utilisée dans la subroutine nommée groupeun.F *:*

- *Les points sont regroupés par paquets de* 2^{ngroup} *aux poles (e.g.* **ngroup**=*3* $\rightarrow$ *paquets de 8), puis* $2^{ngroup-1}$, $2^{ngroup-2}$, *etc... aux latitude plus basse en s'eloignant du pôle*
- *Plus* **ngroup** *est élevé, plus le lissage est efficace, et plus le modèle est stable.*
- *MAIS, il faut* **iim** *divisible par* 2^{ngroup} *! ! !*

3.3 Dissipation

Extrait adapté de Forget et al. [1999]

In the LMD grid point model, nonlinear interactions between explicitly resolved scales and subgrid-scale processes are parameterized by applying a scale-selective horizontal dissipation operator based on an n time iterated Laplacian Δ^n. For the grid point model, for instance, this can be written $\partial q/\partial t = ([-1]^n/\tau_{\text{diss}})(\delta x)^{2n}\Delta^n q$ where δx is the smallest horizontal distance represented in the model and τ_{diss} is the dissipation timescale for a st ructure of scale δx. These operators are necessary to ensure the grid point model numerical stability. In practice, the operator is separately applied to (1) potential temperature, (2) the divergence of the flow, and (3) its vorticity. We respectively use $n = 2$, $n = 1$, and $n = 2$ in the grid point model.

Note : En pratique, les valeurs des n et de τ_{diss} sont réglables et prescrites au début de chaque run dans le fichier de définition du run "run.def" (cf. 6.2.2)

3.4 Sponge layer

Uniquement pour le modèle martien. Adapté de Forget et al. [1999]

In the upper levels a sponge layer is also used in both models in an attempt to reduce spurious reflections of vertically propagating waves from the model top. Unlike the traditional Rayleigh friction formulation, this operates as a linear drag solely on the eddy components of the vorticity and divergence fields and is not scale-selective. The timescales on which it operates are typically half a day, 1 day, and 2 days at the three uppermost levels, respectively.

Note : les valeurs des "timescale" de la sponge layer et son extension en altitude sont réglables et prescrites au début de chaque run dans le fichier de définition du run "run.def" (cf. 6.2.2)

Chapitre 4

Les paramétrisations physiques du modèle LMDZ4

références

4.1 A FAIRE

Chapitre 5

Organisation informatique et compilation

Le modèle du LMD est organisé sur un répertoire de base. Ce répertoire est associé à la variable d'environnement **LMDGCM**. L'exemple ci-dessous montre l'affectation de cette variable en csh sous UNIX pour la version de référence du modèle au LMD.

```
=========================================================================
Environment
===========

csh
setenv LMDGCM ~/LMDZ4
setenv IOIPSLDIR /u/hourdin/IOIPSLmaf/src       # IOIPSL software
setenv NCDFLIB /distrib/local/netcdf/lib        # for Netcdf
setenv NCDFINC -I/distrib/local/netcdf/include  # for Netcdf

bash
export LMDGCM=~/LMDZ4
export IOIPSLDIR=/u/hourdin/IOIPSLmaf/src       # IOIPSL software
export NCDFLIB=/distrib/local/netcdf/lib        # for Netcdf
export NCDFINC=-I/distrib/local/netcdf/include  # for Netcdf


=========================================================================
```

Voici très sommairement le contenu de ce répertoire.

```
=========================================================================
Source files
============

cd $LMDGCM
content :
- libf/bibio     -> some IO routines
       dyn3d     -> routines of the dynamical core
       filtrez   -> longitudinal filter for polar regions.
       grid      -> dimensiosn (iim, jjm, llm) in (X, y, z)
       phylmd    -> routines of the physical package
       phylolo   -> alternative routines of the physical package
```

```
- create_make_gcm    ->  a script generating a makefile for the make
                         unix command.
- makegcm            ->  a unix script to compile the model.

=========================================================================
```

5.1 Organisation des fichiers sources du modèle

Les fichiers sources du modèle sont stockés dans différents sous-répertoires au sein du répertoire **libf**. Ces sous-répertoires correspondent à différentes parties du modèle :

grid : constitué essentiellement du fichier "dimensions.h", qui contient les paramètres définissant la grille du modèle, à savoir le nombre de points en longitude (IIM), en latitude (JJM) et sur la verticale (LLM), ainsi que le nombre de traceurs (NQMX) advectés dans la dynamique (par exemple 2 pour l'eau et la vapeur d'eau pour la version de base du modèle terrestre). Le second fichier contient les fonction $\lambda(I)$, $\phi(J)$ reliant la longitude et la latitude aux indices du modèle, ainsi que leurs dérivées.

dyn3d : contient les sous-programmes de la dynamique.

bibio : contient quelques subroutines génériques utiles à toutes les parties.

phylmd : contient les sources des paramétrisations physiques.

filtrez : contient les sources du filtre en longitude apppliqués dans les hautes latitude, là où le critère de stabilité de Courant-Friedrich-Levy est violé.

5.2 Ecriture informatique

Le modèle est écrit en **FORTRAN-77** et en **FORTRAN-90**.
– Les sources des programmes sont écrites dans des fichiers **"fichier.F" ou "fichier.F90"**. Le suffixe .F est le suffixe standard pour un fichier FORTRAN qui doit être passé par le **pré-processeur-C (cpp)** avant compilation. La plupart des compilateurs FORTRAN reconnaissent automatiquement ce suffixe. Les fichiers .F et .F90 sont compilés en FORTRAN-90 mais en respectant le format FORTRAN-77 pour les .F et FORTRAN-90 pour les .F90.
Un fichier source correspond généralement à un programme ou sous-programme FORTRAN ou, rarement, à un petit ensemble cohérent de sous-programmes

– Les constantes sont placées dans des COMMON situés dans des fichiers "include" communs **"fichier.h"** ou **"fichier.inc"**.
– En règle générale, les variables sont passées de subroutine en subroutine comme arguments (et jamais en COMMON)
– Dans certaines parties du code, pour des raisons "historiques", nous appliquons en partie la norme suivante : dans une subroutine, les variables (ex : `name`) passées en argument par le programme appelant ont un préffixe `p` (ex : `pname`) tandis que les variables locales ont un préfixe `z` (ex : `zname`). En conséquence, de nombreuses variables change de préfixe (et donc de nom) en passant du (sous-) programme appelant au sous-programme appelé.

5.3 Organigramme du modèle

physiq.F

1. Initialisation
   phyeta0.F,surfini.F,iniorbit.F, initracer.F,solarlong.F

1.5 Calculation of mean mass and cp, R and thermal conduction coeff
   concentration.F

2. Calculation of the radiative tendencies : radiative transfer
   (longwave and shortwave) for CO2 and dust.
   dustopacity.F and callradite.F

8. Gravity wave and subgrid scale topography drag.
   calldrag_noro.F

10. Vertical diffusion (turbulent mixing).
   vidfc.F

12. Convective adjustment
   convadj.F

14. Condensation and sublimation of carbon dioxide.
   newcondens.F

7. TRACERS :
   6a. water and water ice: ***watercloud.F***
   6b. call for photochemistry when tracers are chemical species: ***callchim.F***
   6c.other scheme for tracer (dust) transport (lifting, sedimentation): ***dustdevil.F, callsedim.F***
   6d. updates (CO2 pressure variations, surface budget)

19 Thermosphere
   thermosphere.F

8.5 Surface and sub-surface temperature calculations
   soil.F

9. Writing output files :
   - "startfi", "histfi" (if it's time): ***physdem1.F***
   - saving statistics (if "callstats = .true."): ***wstats.F***
   - dumping eof (if "calleofdump = .true."): ***eofdump.F***
   - output any needed variables in "diagfi" : ***writediagfi.F***

FIG. 5.1 – Organigramme de la fonction physiq.F

.

5.4 Compilation du modèle

Le modèle est compilé au moyen de l'utilitaire MAKE de UNIX. Le fichier makefile qui décrit comment la compilation doit se faire est créé automatiquement par le script

`create_make_gcm`

Cet utilitaire recrée le makefile quand nécessaire, par exemple quand un fichier source a été rajouté ou enlevé depuis la précédente compilation.

Tout ceci est transparent à l'utilisateur. Il lui suffit, pour compiler le modèle, de lancer la commande

`makegcm`

dont l'utilisation est décrite plus loin.

Manuel de la fonction makegcm

model /u/hourdin/LMDZ4.beta

makegcm [Options] prog

Par default, la commande makegcm : ——————————

1. compile une serie de sous programmes se trouvant dans des sous-repertoires de /u/hourdin/LMDZ4.beta/libf. Les sous programmes sont ensuite stokes sur dans des librairies FORTRAN sur /u/hourdin/LMDZ4.beta/libo.

2. Ensuite, makegcm compile le programme prog.f se trouvant par default sur /u/hourdin/LMDZ4.beta/libf/dyn3d et effectue le lien avec l'ensemble des librairies.

La variable '/u/hourdin/LMDZ4.beta' doit etre initialisee dans votre .cshrc ou en dur dans la comande makegcm.

La commande makegcm est faite pour permettre de gerer en parallele des versions differentes du modele, compilees avec des options de compilation et des dimensions differentes sans avoir a chaque fois a recompiler tout le modele.

Les librairies FORTRAN sont stoquees sur le directory /u/hourdin/LMDZ4.beta/libo.

OPTIONS : ——

Les options suivantes peuvent etre definies soit par defaut en editant le "script" makegcm, soit en interactif :

-d imxjmxlm ou im, jm, et lm sont resp. le nombre de longitudes, latitudes et couches verticales.

-t ntrac selectionne le nombre de traceur advectes par la dynamique. Dans les versions courantes du modele terrestre on a par exemple ntrac=2 pour l'eau vapeur et liquide

-c ncorps selectionne le nombre d'especes chimiques advectes par la dynamique.

L'effet des options -d, -t et -c est d'ecraser le fichier /u/hourdin/LMDZ4.beta/libf/grid/dimensions.h qui contient sous forme de 5 PARAMETER FORTRAN les 3 dimensions de la grille horizontale im, jm, lm plus le nombre de traceurs advectes passivement par la dynamique ntrac et le nombre d'especes chimiques ncorps, par un nouveau fichier /u/hourdin/LMDZ4.beta/libf/grid/dimension/dimensions.im.jm.lm.tntrac.cncorps
Si ce fichier n'existe pas encore, il est cree par le script /u/hourdin/LMDZ4.beta/libf/grid/dimension/makdim

-p PHYS pour selectionner le jeu de parametrisations physiques avec lequel on veut compiler le modele. Le modele sera alors compile en prenant les sources des parametrisations physiques dans le repertoire : /u/hourdin/LMDZ4.beta/libf/phyPHYS

-g grille selectionne le type de grille qu'on veut utiliser. L'effet de cette option est d'ecraser le fichier /u/hourdin/LMDZ4.beta/libf/grid/fxyprim.h avec le fichier /u/hourdin/LMDZ4.beta/libf/grid/fxy$_grille.hgrillepeutprendrelesvaleurs$: $1.regpourlagrillereguliere2.sinpouravoirdespointsequidistantsensinusdelalatitude3.newpourpouvoirzoomersurunepartieduglobe$

-O "optimisation fortran" ou les optimisations fortran sont les options de la commande f77

-include path Dans le cas ou on a dans des sous programmes des fichiers include (cpp) qui se trouve sur des repertoires non references par defaut

-adjnt Pour compiler la l'adjoint du code dynamique

-filtre filtre Pour choisir le filtre en longitude dans les regions polaires. "filtre" correspond au nom d'un repertoire se trouvant sur /u/hourdin/LMDZ4.beta/libf. Le filtre standard du modele est "filtrez" qui peut etre utilise aussi bien pour une grille reguliere que pour une grille zoomee en longitude.

-link "-Ldir1 -lfile1 -Ldir2 -lfile2 ..." Pour rajouter un lien avec les librairies FORTRAN libfile1.a, libfile2.a ... se trouvant respectivement sur les repertoires dir1, dir2 ... Si dirn est un repertoire dont le chemin est automatique (comme par exemple /usr/lib ...) il n'est pas besoin de specifier -Ldirn.

Auteur : Frederic Hourdin (hourdin@lmd.ens.fr)

Chapitre 6

Entrées/Sorties

6.1 format NetCDF

Les entrées/sorties du GCM sont écrites au format **NetCDF** (Network Common Data Form). NetCDF est une interface pour le stockage et l'accès aux données géophysiques, et une librairie qui fournit une implémentation de cette interface. La librairie NetCDF définit aussi un format indépendant de la machine pour représenter les données scientifiques. Ensembles, interface, librairie et format permettent la création, l'accès et le partage des données scientifiques. NetCDF a été développé par le centre de programme Unidata à Boulder, Colorado. Les sources sont disponibles gratuitement depuis le site web d'Unidata.

`http://www.unidata.ucar.edu/packages/netcdf/index.html`

Un jeu de données sous format NetCDF se compose d'un seul fichier, car il est auto-descriptif.

6.1.1 L'éditeur de fichiers NetCDF : ncdump

L'éditeur est inclu dans la librairie NetCDF. Par defaut il permet de générer une représentation ASCII en sortie standard du fichier NetCDF spécifié en entrée.

Principales commandes de ncdump

ncdump histmth.nc

édite le fichier NetCDF "histmth.nc".

ncdump -c histmth.nc

Affiche les valeurs des variables coordonnées (variables qui sont aussi des dimensions), ainsi que les déclarations, variables et valeurs d'attribut. Les valeurs des données de variables non coordonnées ne sont pas affichées en sortie.

ncdump -h histmth.nc

Montre seulement l'en-tête informative du fichier, qui est la déclaration des dimensions, variables et attribut, mais pas les valeurs de ces variables. La sortie est identique à celle de l'option -c sauf que les valeurs de variables coordonnées ne sont pas incluses.

ncdump -v var1,...,varn histmth.nc

La sortie inclue les valeurs des variables spécifiées, en plus des déclarations de toutes les dimensions, variables et attributs. Plus d'une variable peut être spécifiée dans la liste suivant cette option. La liste doit être un argument simple pour la commande et ne pas contenir d'espace. Si aucune variable n'est spécifiée, la commande affichera par défaut la valeur de toutes les variables du fichier.

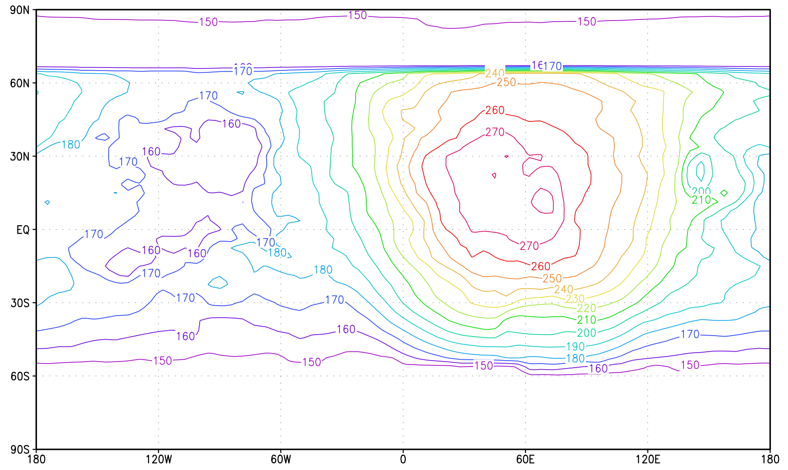


FIG. 6.1 – Exemple de visualisation sous GrADS des données de temperature à un instant donné

6.1.2 Les commandes NCO

L'ensemble de commandes nco permet d'effectuer tout un tas d'opérations sur les fichiers NetCDF. On peut par exemple calculer la moyenne de plusieurs fichiers, extraires des variables, extraire une région particulière du globe. Ces commandes permettent par exemple d'extraire en sortie de simulation un sous-fichier restreint à conserver ou à envoyer à quelqu'un à partir d'un fichier de sortie qui peut $^{e}tretropvolumineux$.

6.1.3 Visualisation graphique des fichiers NetCDF sous GrAds

GrAdS (The Grid Analysis and Display System) est un logiciel graphique du domaine publique qui a été développé par Brian Doty au "Center for Ocean-Land-Atmosphere (COLA)".

Il permet entre autre de visualiser directement des données stockées sous le format NetCDF. On peut voir par exemple figure 6.1 la visualisation sous GrADS des données de température à un instant donné. Cependant, contrairement à NetCDF, GrADS ne reconnait que des fichiers où toutes les variables sont stockées sur la même grille horizontale. Ces variables peuvent être de 1, 2, 3 ou 4 dimensions (X,Y,Z et t).

GrADS est également accessible par le WWW.

`http://grads.iges.org/grads/`

6.1.3.1 Le fichier de données (.nc)

Le fichier de données est un fichier binaire à accès direct. Il comprend deux parties :

- Un en-tête, contenant toutes les informations sur les dimensions, attributs et variables, excepté pour les variables de données (celles qui ne sont pas des variables de dimension).
- Les données elles-mêmes (à taille fixée ou variable).

La figure 6.2 représente le stockage dans le fichier .nc des vents u et v (champs 4D : X,Y,Z et t), et de la pression de surface ps (champ 3D : X,Y et t).

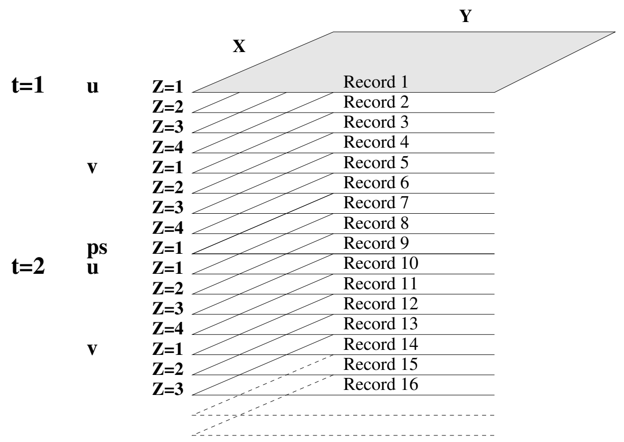


FIG. 6.2 – Exemple de stockage des vents u et v (champs 4D : X,Y,Z et t), et de la pression de surface ps (champ 3D : X,Y et t)

6.2 Entrées

Des exemples de fichiers d'initialisation se trouvent dans le répertoire `$PATH1/LMDZ.MARS/deftank`

Le GCM 3D a besoin en entrée de deux fichiers (NetCDF) d'initialisation :
-start.nc contenant les états initiaux des variables de la dynamique.
-startphy.nc contenant les états initiaux des variables de la physique.

-limit.nc contenant les conditions aux limites en surface (température de surface des océans, rugosité de surface, albédo).
et les fichiers (ascii) de paramètres :
-run.def.def les paramètres principaux contr$^o lantlalongueurdesimulationsetlenomdesautresfichiersdeconfiguration.$
-gcm.def les paramètres de contr$^o ledelapartiedynamique.$
-physiq.def les paramètres de contr$^o ledesparamétrisationsphysiques.$
-traceur.def les paramètres de contr$^o ledestraceurs.$

6.2.1 run.def

```
#
# $Header: /users/lmdz/cvsroot/LMDZ.3.3/run.def,v 1.2.2.3 2002/07/12 14:11:18 lmdzadmin Exp $
#
INCLUDEDEF=physiq.def
INCLUDEDEF=gcm.def
INCLUDEDEF=orchidee.def
## Jour de l'etat initial ( = 350  si 20 Decembre ,par expl. ,comme ici )
dayref=1
##  Annee de l'etat  initial (   avec  4  chiffres   )
anneeref=1979
## Nombre de jours d'integration
nday=30
## periode de sortie des variables de controle (en pas)
iconser=10
## periode d'ecriture du fichier histoire (en jour)
iecri=1
## periode de stockage fichier dyn_hist_ave et dynzon (en jour)
```

```
periodav=30.
```

6.2.2 gcm.def

Un fichier gcm.def typique est donné en exemple ci-dessous. Le choix des variables à règler est simple (e.g. durée de l'intégration nday), tandis que d'autres n'ont pas à être modifiées dans le cadre d'une utilisation normale. Quelques remarques cependant :

– **day_step**, le nombre de pas par jour est régi par le critère de stabilité dit "CFL", qui dépend de la résolution horizontale du modèle. Sur Mars, en théorie, le GCM peut tourner avec `day_step`=400 en 64×48, mais le mod'ele gagne en stabilité avec un chiffre plus élevé : `day_step`=960 est recommandé en 64×48. D'aprés le critère CFL, `day_step` doit varier proportionellement à la résolution : on prendra par exemple `day_step`=480 en résolution horizontale 32×24. `day_step` doit aussi être divisible par "iperiod"

– **tetagdiv, tetagrot, tetatemp** contrôlent l'intensité de la dissipation. Il est préférable de limiter l'intensité de la dissipation (tetagdiv, tetagrot, tetatemp ne doivent pas être trop petites). Cependant, le modèle diverge si tetagdiv, tetagrot, tetatemp sont trop élevés, dans le cas d'une atmosphère chargée en poussière en particulier.
Exemple utilisable avec nitergdiv=1 et nitergrot=niterh=2 :
- en 64×48 tetagdiv=6000 s ; tetagrot=tetatemp=30000 s
- en 32×24 : tetagdiv=3000 s ; tetagrot=tetatemp=9000 s

– **iflag**$_phys = 1 sigcm.eappellelaphysiqueet0 sionneveutfairetournerqueladynamique.Cetteoptionestpratiquenotammentencasdeplantagedumodèlepourdistinguerunplantagenumériqueliéaupasdetempsouàladissipation(quiadebonnechancedesemaintenirm^emesansphysique)d'unplantageliéàlaphysiqueouàdesproblèmesinformatiques.$

– **iphysiq** correspond au pas de temps physique. En pratique, on fait généralement en sorte que ce pas de temps soit de l'ordre d'une demi-heure. On prend donc iphysiq= day_step/48

Contenu de gcm.def :

```
## $Header$
## nombre de pas par jour (multiple de iperiod) ( ici pour  dt = 1 min )
day_step=480
iapp_tracvl=5
## periode pour le pas Matsuno (en pas)
iperiod=5
## periode de la dissipation (en pas)
idissip=5
## choix de l'operateur de dissipation (star ou  non star )
lstardis=y
## nombre d'iterations de l'operateur de dissipation   gradiv
nitergdiv=1
## nombre d'iterations de l'operateur de dissipation  nxgradrot
nitergrot=2
## nombre d'iterations de l'operateur de dissipation  divgrad
niterh=2
## temps de dissipation des plus petites long.d ondes pour u,v (gradiv)
tetagdiv=43200.
## temps de dissipation des plus petites long.d ondes pour u,v(nxgradrot)
tetagrot=21600.
## temps de dissipation des plus petites long.d ondes pour  h ( divgrad)
tetatemp=21600.
## coefficient pour gamdissip
coefdis=0.
## choix du shema d'integration temporelle (Matsuno ou Matsuno-leapfrog)
purmats=n
## avec ou sans physique
```

```
iflag_phys=1
## periode de la physique (en pas)
iphysiq=10
## frequence (en  jours ) de l'ecriture du fichier histphy
ecritphy=0.02
##  Cycle diurne  ou non
cycle_diurne=y
##  Soil Model  ou non
soil_model=y
##  Choix ou non  de  New oliq
new_oliq=y
##  Orodr  ou  non   pour l orographie
ok_orodr=y
##  Orolf  ou  non   pour l orographie
ok_orolf=y
##   Si = .T. ,  lecture du fichier limit avec la bonne annee
ok_limitvrai=n
## Nombre  d'appels des routines de rayonnements ( par jour)
nbapp_rad=12
##  Flag  pour la convection (1 pour LMD, 2 pour Tiedtke, 3 KE, 4 KE vect)
iflag_con=3
## longitude en degres du centre du zoom
clon=0.
## latitude en degres du centre du zoom
clat=0.
## facteur de grossissement du zoom,selon longitude
grossismx=1.0
## facteur de grossissement du zoom ,selon latitude
grossismy=1.0
##  Fonction  f(y)  hyperbolique  si = .true.  , sinon  sinusoidale
fxyhypb=y
## extension en longitude  de la zone du zoom  ( fraction de la zone totale)
dzoomx=0.0
## extension en latitude de la zone  du zoom  ( fraction de la zone totale)
dzoomy=0.0
##raideur du zoom en  X
taux=3.
##raideur du zoom en  Y
tauy=3.
##  Fonction  f(y) avec y = Sin(latit.) si = .true. , sinon y = latit.
ysinus=y
```

6.2.3 physiq.def

```
#
## $Header: /users/lmdz/cvsroot/LMDZ.3.3/physiq.def,v 1.1.2.1 2002/07/12
14:11:18 lmdzadmin Exp $
#
#
# Automatically generated make config: don't edit
#

OCEAN=force
VEGET=n
```

```
OK_journe=y
OK_mensuel=y
OK_instan=n
#
# parametres KE
#
epmax = .99
ok_adj_ema = n
iflag_clw = 1
#
# parametres nuages
#
cld_lc_lsc = 2.6e-4
cld_lc_con = 2.6e-4
cld_tau_lsc = 3600.
cld_tau_con = 3600.
ffallv_lsc = 1.
ffallv_con = 1.
coef_eva = 2.e-5
reevap_ice = y
iflag_cldcon = 3
iflag_pdf = 1
fact_cldcon = 1.
facttemps = 1.e-4
ok_newmicro = y
ratqsbas = 0.005
ratqshaut = 0.33
rad_froid = 35
rad_chau1 = 12
rad_chau2 = 11
if_ebil = 2
R_ecc = 0.016715
R_peri = 102.7
R_incl = 23.441
solaire = 1365.
co2_ppm = 348.
#RCO2 = co2_ppm * 1.0e-06  * 44.011/28.97
#RCO2 = 348. * 1.0e-06  * 44.011/28.97
#RCO2 =   5.286789092164308E-04
CH4_ppb = 1650.
#RCH4 = 1.65E-06* 16.043/28.97
#RCH4 =   9.137366240938903E-07
N2O_ppb = 306.
#RN2O = 306.E-09* 44.013/28.97
#RN2O =    4.648939592682085E-07
CFC11_ppt = 280.
#RCFC11 = 280.E-12* 137.3686/28.97
#RCFC11 =    1.327690990680013E-09
CFC12_ppt = 484.
#RCFC12 = 484.E-12* 120.9140/28.97
#RCFC12 =    2.020102726958923E-09
#
# parametres simulateur ISCCP
#
```

```
#top_height = 1 ou 3
top_height = 1
#overlap = 1, 2 ou 3
overlap = 3
#cdmmax
cdmmax = 2.5E-3
#cdhmax
cdhmax = 2.0E-3
#kstable en dehors des terres
ksta = 1.0e-10
#kstable sur terres
ksta_ter = 1.0e-7
#ok_kzmin : calcul Kzmin dans la CL de surface
ok_kzmin=y

iflag_thermals = 0
nsplit_thermals = 10
iflag_pbl=1
```

6.2.4 traceur.def

```
4
10 10 q01
10 10 q02
10 10 q03
10 10 q04
```

6.2.5 les fichiers d'initialisation : start et startphy

Les fichiers start et startphy, comme tous les fichier NetCDF du GCM, sont bâtis sur le même modèle (voir la composition d'un fichier NetCDF figure 6.3). Ils contiennent :
- un en-tête constitué d'une variable "contrôle" suivie d'une série de variables définissant la grille (physique ou dynamique).
- une série de variables non temporelles donnant des informations sur les conditions de surface de la planète.
- une variable "temps" donnant les valeurs des différents instants auxquels les variables temporelles sont stockées (une seule valeur de temps (t=0) pour start et startphy, puisqu'il s'agit des conditions intiales).
- les variables d'état du modèle sur la grille correspondante aux différents instants de la simulation sélectionnés (réglage avec un pas de temps) pour les fichiers de sortie en cours de simulation (histmth, histADEF, etc...).

Visualisons à l'aide de l'éditeur ncdump le contenu d'un fichier start :

ncdump -h start.nc

```
netcdf start {
dimensions:
index = 100 ;
rlonu = 97 ;
rlatu = 72 ;
rlonv = 97 ;
```

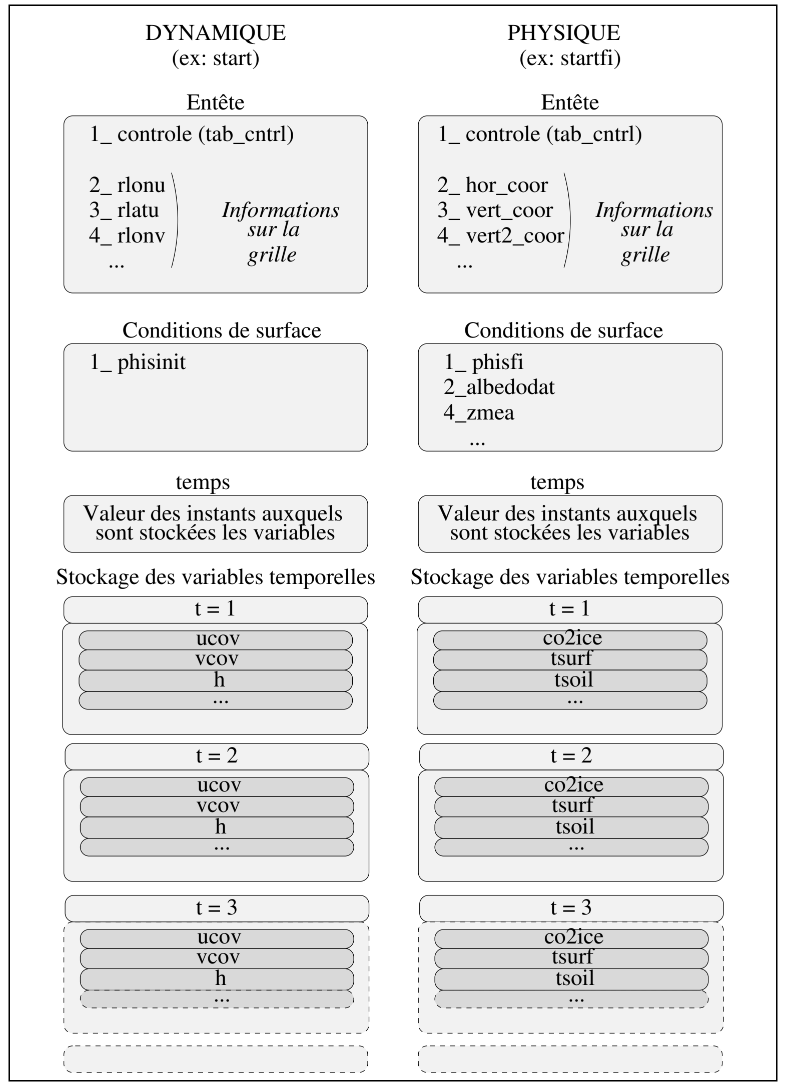


FIG. 6.3 – Organisation des fichiers NetCDF

```
rlatv = 71 ;
sigs = 19 ;
sig = 20 ;
temps = UNLIMITED ; // (1 currently)
variables:
float controle(index) ;
controle:title = "Parametres de controle" ;
float rlonu(rlonu) ;
rlonu:title = "Longitudes des points U" ;
float rlatu(rlatu) ;
rlatu:title = "Latitudes des points U" ;
float rlonv(rlonv) ;
rlonv:title = "Longitudes des points V" ;
float rlatv(rlatv) ;
rlatv:title = "Latitudes des points V" ;
float nivsigs(sigs) ;
```

```
nivsigs:title = "Numero naturel des couches s" ;
float nivsig(sig) ;
nivsig:title = "Numero naturel des couches sigma" ;
float ap(sig) ;
ap:title = "Coefficient A pour hybride" ;
float bp(sig) ;
bp:title = "Coefficient B pour hybride" ;
float presnivs(sigs) ;
float cu(rlatu, rlonu) ;
cu:title = "Coefficient de passage pour U" ;
float cv(rlatv, rlonv) ;
cv:title = "Coefficient de passage pour V" ;
float aire(rlatu, rlonv) ;
aire:title = "Aires de chaque maille" ;
float phisinit(rlatu, rlonv) ;
phisinit:title = "Geopotentiel au sol" ;
float temps(temps) ;
temps:title = "Temps de simulation" ;
temps:units = "days since 1979-12-01 00:00:00" ;
float ucov(temps, sigs, rlatu, rlonu) ;
ucov:title = "Vitesse U" ;
float vcov(temps, sigs, rlatv, rlonv) ;
vcov:title = "Vitesse V" ;
float teta(temps, sigs, rlatu, rlonv) ;
teta:title = "Temperature" ;
float q01(temps, sigs, rlatu, rlonv) ;
q01:title = "Traceurs q01" ;
float q02(temps, sigs, rlatu, rlonv) ;
q02:title = "Traceurs q02" ;
float q03(temps, sigs, rlatu, rlonv) ;
q03:title = "Traceurs q03" ;
float q04(temps, sigs, rlatu, rlonv) ;
q04:title = "Traceurs q04" ;
float masse(temps, sigs, rlatu, rlonv) ;
masse:title = "C est quoi ?" ;
float ps(temps, rlatu, rlonv) ;
ps:title = "Pression au sol" ;

// global attributes:
:title = "Fichier demmarage dynamique" ;
}
```

Listing du contenu d'un fichier startphy :

ncdump -h startphy.nc

```
netcdf startphy {
dimensions:
index = 100 ;
points_physiques = 6722 ;
horizon_vertical = 127718 ;
variables:
float controle(index) ;
controle:title = "Parametres de controle" ;
```

```
float longitude(points_physiques) ;
longitude:title = "Longitudes de la grille physique" ;
float latitude(points_physiques) ;
latitude:title = "Latitudes de la grille physique" ;
float masque(points_physiques) ;
masque:title = "masque terre mer" ;
float FTER(points_physiques) ;
FTER:title = "fraction de continent" ;
float FLIC(points_physiques) ;
FLIC:title = "fraction glace de terre" ;
float FOCE(points_physiques) ;
FOCE:title = "fraction ocean" ;
float FSIC(points_physiques) ;
FSIC:title = "fraction glace mer" ;
float TS01(points_physiques) ;
TS01:title = "Temperature de surface No.01" ;
float TS02(points_physiques) ;
TS02:title = "Temperature de surface No.02" ;
float TS03(points_physiques) ;
TS03:title = "Temperature de surface No.03" ;
float TS04(points_physiques) ;
TS04:title = "Temperature de surface No.04" ;
float Tsoil01srf01(points_physiques) ;
Tsoil01srf01:title = "Temperature du sol No.01srf01" ;
float Tsoil02srf01(points_physiques) ;
Tsoil02srf01:title = "Temperature du sol No.02srf01" ;
float Tsoil03srf01(points_physiques) ;
Tsoil03srf01:title = "Temperature du sol No.03srf01" ;
float Tsoil04srf01(points_physiques) ;
Tsoil04srf01:title = "Temperature du sol No.04srf01" ;
float Tsoil05srf01(points_physiques) ;
Tsoil05srf01:title = "Temperature du sol No.05srf01" ;
float Tsoil06srf01(points_physiques) ;
Tsoil06srf01:title = "Temperature du sol No.06srf01" ;
float Tsoil07srf01(points_physiques) ;
Tsoil07srf01:title = "Temperature du sol No.07srf01" ;
float Tsoil08srf01(points_physiques) ;
Tsoil08srf01:title = "Temperature du sol No.08srf01" ;
float Tsoil09srf01(points_physiques) ;
Tsoil09srf01:title = "Temperature du sol No.09srf01" ;
float Tsoil10srf01(points_physiques) ;
Tsoil10srf01:title = "Temperature du sol No.10srf01" ;
float Tsoil11srf01(points_physiques) ;
Tsoil11srf01:title = "Temperature du sol No.11srf01" ;
float Tsoil01srf02(points_physiques) ;
Tsoil01srf02:title = "Temperature du sol No.01srf02" ;
float Tsoil02srf02(points_physiques) ;
Tsoil02srf02:title = "Temperature du sol No.02srf02" ;
float Tsoil03srf02(points_physiques) ;
Tsoil03srf02:title = "Temperature du sol No.03srf02" ;
float Tsoil04srf02(points_physiques) ;
Tsoil04srf02:title = "Temperature du sol No.04srf02" ;
float Tsoil05srf02(points_physiques) ;
Tsoil05srf02:title = "Temperature du sol No.05srf02" ;
```

```
float Tsoil06srf02(points_physiques) ;
Tsoil06srf02:title = "Temperature du sol No.06srf02" ;
float Tsoil07srf02(points_physiques) ;
Tsoil07srf02:title = "Temperature du sol No.07srf02" ;
float Tsoil08srf02(points_physiques) ;
Tsoil08srf02:title = "Temperature du sol No.08srf02" ;
float Tsoil09srf02(points_physiques) ;
Tsoil09srf02:title = "Temperature du sol No.09srf02" ;
float Tsoil10srf02(points_physiques) ;
Tsoil10srf02:title = "Temperature du sol No.10srf02" ;
float Tsoil11srf02(points_physiques) ;
Tsoil11srf02:title = "Temperature du sol No.11srf02" ;
float Tsoil01srf03(points_physiques) ;
Tsoil01srf03:title = "Temperature du sol No.01srf03" ;
float Tsoil02srf03(points_physiques) ;
Tsoil02srf03:title = "Temperature du sol No.02srf03" ;
float Tsoil03srf03(points_physiques) ;
Tsoil03srf03:title = "Temperature du sol No.03srf03" ;
float Tsoil04srf03(points_physiques) ;
Tsoil04srf03:title = "Temperature du sol No.04srf03" ;
float Tsoil05srf03(points_physiques) ;
Tsoil05srf03:title = "Temperature du sol No.05srf03" ;
float Tsoil06srf03(points_physiques) ;
Tsoil06srf03:title = "Temperature du sol No.06srf03" ;
float Tsoil07srf03(points_physiques) ;
Tsoil07srf03:title = "Temperature du sol No.07srf03" ;
float Tsoil08srf03(points_physiques) ;
Tsoil08srf03:title = "Temperature du sol No.08srf03" ;
float Tsoil09srf03(points_physiques) ;
Tsoil09srf03:title = "Temperature du sol No.09srf03" ;
float Tsoil10srf03(points_physiques) ;
Tsoil10srf03:title = "Temperature du sol No.10srf03" ;
float Tsoil11srf03(points_physiques) ;
Tsoil11srf03:title = "Temperature du sol No.11srf03" ;
float Tsoil01srf04(points_physiques) ;
Tsoil01srf04:title = "Temperature du sol No.01srf04" ;
float Tsoil02srf04(points_physiques) ;
Tsoil02srf04:title = "Temperature du sol No.02srf04" ;
float Tsoil03srf04(points_physiques) ;
Tsoil03srf04:title = "Temperature du sol No.03srf04" ;
float Tsoil04srf04(points_physiques) ;
Tsoil04srf04:title = "Temperature du sol No.04srf04" ;
float Tsoil05srf04(points_physiques) ;
Tsoil05srf04:title = "Temperature du sol No.05srf04" ;
float Tsoil06srf04(points_physiques) ;
Tsoil06srf04:title = "Temperature du sol No.06srf04" ;
float Tsoil07srf04(points_physiques) ;
Tsoil07srf04:title = "Temperature du sol No.07srf04" ;
float Tsoil08srf04(points_physiques) ;
Tsoil08srf04:title = "Temperature du sol No.08srf04" ;
float Tsoil09srf04(points_physiques) ;
Tsoil09srf04:title = "Temperature du sol No.09srf04" ;
float Tsoil10srf04(points_physiques) ;
Tsoil10srf04:title = "Temperature du sol No.10srf04" ;
```

```
float Tsoil11srf04(points_physiques) ;
Tsoil11srf04:title = "Temperature du sol No.11srf04" ;
float DELTAT(points_physiques) ;
DELTAT:title = "Ecart de la SST (pour slab-ocean)" ;
float QS01(points_physiques) ;
QS01:title = "Humidite de surface No.01" ;
float QS02(points_physiques) ;
QS02:title = "Humidite de surface No.02" ;
float QS03(points_physiques) ;
QS03:title = "Humidite de surface No.03" ;
float QS04(points_physiques) ;
QS04:title = "Humidite de surface No.04" ;
float QSOL(points_physiques) ;
QSOL:title = "Eau dans le sol (mm)" ;
float ALBE01(points_physiques) ;
ALBE01:title = "albedo de surface No.01" ;
float ALBE02(points_physiques) ;
ALBE02:title = "albedo de surface No.02" ;
float ALBE03(points_physiques) ;
ALBE03:title = "albedo de surface No.03" ;
float ALBE04(points_physiques) ;
ALBE04:title = "albedo de surface No.04" ;
float ALBLW01(points_physiques) ;
ALBLW01:title = "albedo LW de surface No" ;
float ALBLW02(points_physiques) ;
ALBLW02:title = "albedo LW de surface No" ;
float ALBLW03(points_physiques) ;
ALBLW03:title = "albedo LW de surface No" ;
float ALBLW04(points_physiques) ;
ALBLW04:title = "albedo LW de surface No" ;
float EVAP01(points_physiques) ;
EVAP01:title = "Evaporation de surface No.01" ;
float EVAP02(points_physiques) ;
EVAP02:title = "Evaporation de surface No.02" ;
float EVAP03(points_physiques) ;
EVAP03:title = "Evaporation de surface No.03" ;
float EVAP04(points_physiques) ;
EVAP04:title = "Evaporation de surface No.04" ;
float SNOW01(points_physiques) ;
SNOW01:title = "Neige de surface No.01" ;
float SNOW02(points_physiques) ;
SNOW02:title = "Neige de surface No.02" ;
float SNOW03(points_physiques) ;
SNOW03:title = "Neige de surface No.03" ;
float SNOW04(points_physiques) ;
SNOW04:title = "Neige de surface No.04" ;
float RADS(points_physiques) ;
RADS:title = "Rayonnement net a la surface" ;
float solsw(points_physiques) ;
solsw:title = "Rayonnement solaire a la surface" ;
float sollw(points_physiques) ;
sollw:title = "Rayonnement IF a la surface" ;
float fder(points_physiques) ;
fder:title = "Derive de flux" ;
```

```
float rain_f(points_physiques) ;
rain_f:title = "precipitation liquide" ;
float snow_f(points_physiques) ;
snow_f:title = "precipitation solide" ;
float RUG01(points_physiques) ;
RUG01:title = "rugosite de surface No." ;
float RUG02(points_physiques) ;
RUG02:title = "rugosite de surface No." ;
float RUG03(points_physiques) ;
RUG03:title = "rugosite de surface No." ;
float RUG04(points_physiques) ;
RUG04:title = "rugosite de surface No." ;
float AGESNO01(points_physiques) ;
AGESNO01:title = "Age de la neige" ;
float AGESNO02(points_physiques) ;
AGESNO02:title = "Age de la neige" ;
float AGESNO03(points_physiques) ;
AGESNO03:title = "Age de la neige" ;
float AGESNO04(points_physiques) ;
AGESNO04:title = "Age de la neige" ;
float ZMEA(points_physiques) ;
float ZSTD(points_physiques) ;
float ZSIG(points_physiques) ;
float ZGAM(points_physiques) ;
float ZTHE(points_physiques) ;
float ZPIC(points_physiques) ;
float ZVAL(points_physiques) ;
float RUGSREL(points_physiques) ;
float TANCIEN(horizon_vertical) ;
float QANCIEN(horizon_vertical) ;
float RUGMER(points_physiques) ;
RUGMER:title = "Longueur de rugosite sur mer" ;
float CLWCON(points_physiques) ;
CLWCON:title = "Eau liquide convective" ;
float RNEBCON(points_physiques) ;
RNEBCON:title = "Nebulosite convective" ;
float RATQS(points_physiques) ;
RATQS:title = "Ratqstitle" ;

// global attributes:
:title = "Fichier redemmarage physique" ;
}
```

En-têtes physiques et dynamiques

Il y a deux types d'en-têtes : un pour les fichiers physiques et un pour les fichiers dynamiques. Cet en-tête commence toujours par une variable "contrôle" (décrite un peu plus loin), que l'on affecte différemment pour la physique et pour la dynamique. Les autres variables de l'entête concernent les grilles (physique et dynamique). On retrouve donc :

les coordonnées horizontales
- **rlonu**, **rlatu**, **rlonv**, **rlatv** pour la dynamique,
- **lati**, **long** pour la physique,

les coordonnées verticales
- **sig** ,**s** ,**sig_s** pour la dynamique,
- **surf_coor**, **vert2_coor**, **vert_coor**, **hor_coor** pour la physique,

des coefficients de passages de la grille physique à la grille dynamique
- **cu**,**cv** dans l'entête dynamique uniquement

et pour finir l'aire des mailles
- **aire** pour la dynamique,
- **area** pour la physique.

Conditions de surface

Des conditions de surface sont données essentiellement dans les fichiers NetCDF physiques par les variables :
- **phisfi** pour le géopotentiel au sol initial,
- **albedodat** pour l'albedo du sol nu,
- **inertiedat** pour l'inertie thermique du sol,
- **zmea**, **zstd**, **zsig**, **zgam** et **zthe** pour le relief sous-maille.

Pour la dynamique :
- **physinit** pour le géopotentiel au sol initial

Remarque : les variables **phisfi** et **physinit** contiennent les mêmes informations (géopotentiel au sol), mais physfi donne les valeurs du géopotentiel sur la grille physique tandis que physinit les donne sur la grille dynamique.

Variables d'état physiques et dynamiques

Pour des raisons de commodité, on stocke dans les fichiers d'initialisation, non pas les variables "naturelles", mais directement les variables utilisées par le modèle.

Pour la dynamique :

- **ucov** et **vcov** les vents covariants
  Ces variables sont reliées aux vents "naturels" par
  `ucov = cu * u` et `vcov = cv * v`
- **teta** la température potentielle,
  ou plus précisément l'enthalpie potentielle reliée à **T** la temperature par $\theta = T\left(\frac{P}{Pref}\right)^{-} K$
- **q01, q02, etc...** les traceurs,
- **ps** la pression au sol.
- **masse** la masse d'atmosphère dnas chaque maille.

ucov et **vcov**, variables "vectorielles", sont stockées respectivement sur les grilles u et v "décalées" de la dynamique (voir section 2.2).
h, **q01**, **ps**, **masse**, variables scalaires, sont stockées sur la grille dite "scalaire" de la dynamique.

Pour la physique :

- **co2ice** la glace carbonique en surface,
- **tsurf** la température de surface,

- **tsoil** la température dans différentes couches sous la surface,
- **emis** l'emissivité de la surface,
- **q2** la variance du vent,
  ou plus précisement la racine carrée de l'énergie cinétique turbulente.
- **qsurf01, qsurf02, etc...** le budget de "traceur" en surface (kg.m^{-2}),

Toutes ces variables sont stockées sur la grille "physique" (voir section 2.2).

La variable "contrôle"

Qu'ils soient physiques ou dynamiques, les entêtes des fichiers NetCDF du GCM commencent tous par une variable **contrôle**. Cette variables est un tableau de 100 réels (tableau appellé "tab_cntrl" dans le programme), qui contient les paramètres de contrôle du programme. Ces paramètres, différents pour la physique et la dynamique, sont listés ci-dessous. On peut également visualiser le contenu du tableau tab_cntrl en tapant la commande "*ncdump -v controle*".

```
-------------------------------------------------------
tab_cntrl(1)  = FLOAT(iim)      ! nombre de points en longitude
tab_cntrl(2)  = FLOAT(jjm)      ! nombre de points en latitude
tab_cntrl(3)  = FLOAT(llm)      ! nombre de couches
tab_cntrl(4)  = FLOAT(idayref)  ! jour 0
tab_cntrl(5)  = FLOAT(anneeref) ! annee 0
tab_cntrl(6)  = rad     ! rayon de mars(m) ~3397200
tab_cntrl(7)  = omeg    ! vitesse de rotation (rad.s-1)
tab_cntrl(8)  = g       ! gravite (m.s-2) ~3.72
tab_cntrl(9)  = cpp
tab_cntrl(10) = kappa   ! = r/cp  ~0.256793 (=rcp dans physique)
tab_cntrl(11) = daysec ! duree du sol (s)  ~88775
tab_cntrl(12) = dtvr   ! pas de temps de la dynamique (s)
tab_cntrl(13) = etot0  ! energie totale
tab_cntrl(14) = ptot0  ! pression totale
tab_cntrl(15) = ztot0  ! enstrophie totale
tab_cntrl(16) = stot0  ! enthalpie totale
tab_cntrl(17) = ang0   ! moment cinetique
tab_cntrl(18) = pa
tab_cntrl(19) = preff  ! pression de reference ~670

tab_cntrl(20)  = clon       ! longitude en degres du centre du zoom
tab_cntrl(21)  = clat       ! latitude en degres du centre du zoom
tab_cntrl(22)  = grossismx  ! facteur de grossissement du zoom,selon longitude
tab_cntrl(23)  = grossismy  ! facteur de grossissement du zoom ,selon latitude

tab_cntrl(25) = dzoomx    ! extension en longitude de la zone du zoom
tab_cntrl(26) = dzoomy    ! extension en latitude de la zone du zoom
tab_cntrl(28) = taux      ! raideur du zoom en x
tab_cntrl(29) = tauy      ! raideur du zoom en y
```

La variable "contrôle" de l'entête d'un fichier NetCDF physique : startphy

```
c Info sur la grille physique
```

```
      tab_cntrl(1) = float(ngridmx)  ! nombre de points de la grille physique
      tab_cntrl(2) = float(nlayermx) ! nombre de couches
      tab_cntrl(3) = float(idayref)  ! jour 0

c Info sur la Planete Mars pour la dynamique et la physique
      tab_cntrl(5) = rad      ! rayon de mars(m) ~3397200
      tab_cntrl(6) = omeg     ! vitesse de rotation (rad.s-1)
      tab_cntrl(7) = g        ! gravite (m.s-2) ~3.72
      tab_cntrl(8) = mugaz    ! Masse molaire de l'atm (g.mol-1) ~43.49
      tab_cntrl(9) = rcp      !  = r/cp  ~0.256793 (=kappa dans dynamique)
      tab_cntrl(10) = daysec  ! duree du sol (s)  ~88775

      tab_cntrl(11) = dtphys  ! pas de temps de la physique
      tab_cntrl(12) = 0.
      tab_cntrl(13) = 0.

c Info sur la Planete Mars pour la physique uniquement
      tab_cntrl(14) = year_day  ! duree de l'annee (sols) ~668.6
      tab_cntrl(15) = periheli  ! dist.min. soleil-mars (Mkm) ~206.66
      tab_cntrl(16) = aphelie   ! dist.max. soleil-mars (Mkm) ~249.22
      tab_cntrl(17) = peri_day  ! date du perihelie (sols depuis printemps)
      tab_cntrl(18) = obliquit  ! Obliquite de la planete (deg) ~23.98

c Couche limite et Turbulence
      tab_cntrl(19) = z0        ! surface roughness (m) ~0.01
      tab_cntrl(20) = lmixmin   ! longueur de melange ~100
      tab_cntrl(21) = emin_turb ! energie minimale ~1.e-8

c propriete optiques des calottes et emissivite du sol
      tab_cntrl(22) = albedice(1)  ! Albedo calotte nord ~0.5
      tab_cntrl(23) = albedice(2)  ! Albedo calotte sud ~0.5
      tab_cntrl(24) = emisice(1)   ! Emissivite calotte nord ~0.95
      tab_cntrl(25) = emisice(2)   ! Emissivite calotte sud ~0.95
      tab_cntrl(26) = emissiv      ! Emissivite du sol martien ~.95
      tab_cntrl(31) = iceradius(1) ! mean scat radius of CO2 snow (north)
      tab_cntrl(32) = iceradius(2) ! mean scat radius of CO2 snow (south)
      tab_cntrl(33) = dtemisice(1) ! time scale for snow metamorphism (north)
      tab_cntrl(34) = dtemisice(2) ! time scale for snow metamorphism (south)

c Proprietes des poussiere aerosol
      tab_cntrl(27) = tauvis      ! profondeur optique visible moyenne

      tab_cntrl(28) = 0.
      tab_cntrl(29) = 0.
      tab_cntrl(30) = 0.
-------------------------------------------------------
```

6.3 Sorties

6.3.1 fichiers NetCDF de redemarrage restart et restartphy

Pour les utilisations courantes, il est recommandé d'utiliser pour les diagnostics les fichiers de sorties de la physique :

```
histmth.nc  : sorties mensuelles
histday.nc  : sorties journalieres
histhf.nc   : sorties instantanees toutes les 6h
```

Les deux premieres sortent des variables moyennées dans le temps entre deux écritures. Les fréquences mensuelles ou journalières peuvent en fait $^{e}trerégélesdifféremmentpourchacundesdeuxfichiers.$

6.3.2 Le fichier histmth.nc

Entete du fichier histmth.nc

On règle la fréquence de stockage des variables dans le fichier "histmth" par le paramètre **ecritphy** du fichier "gcm.def" (voir 6.2.2)

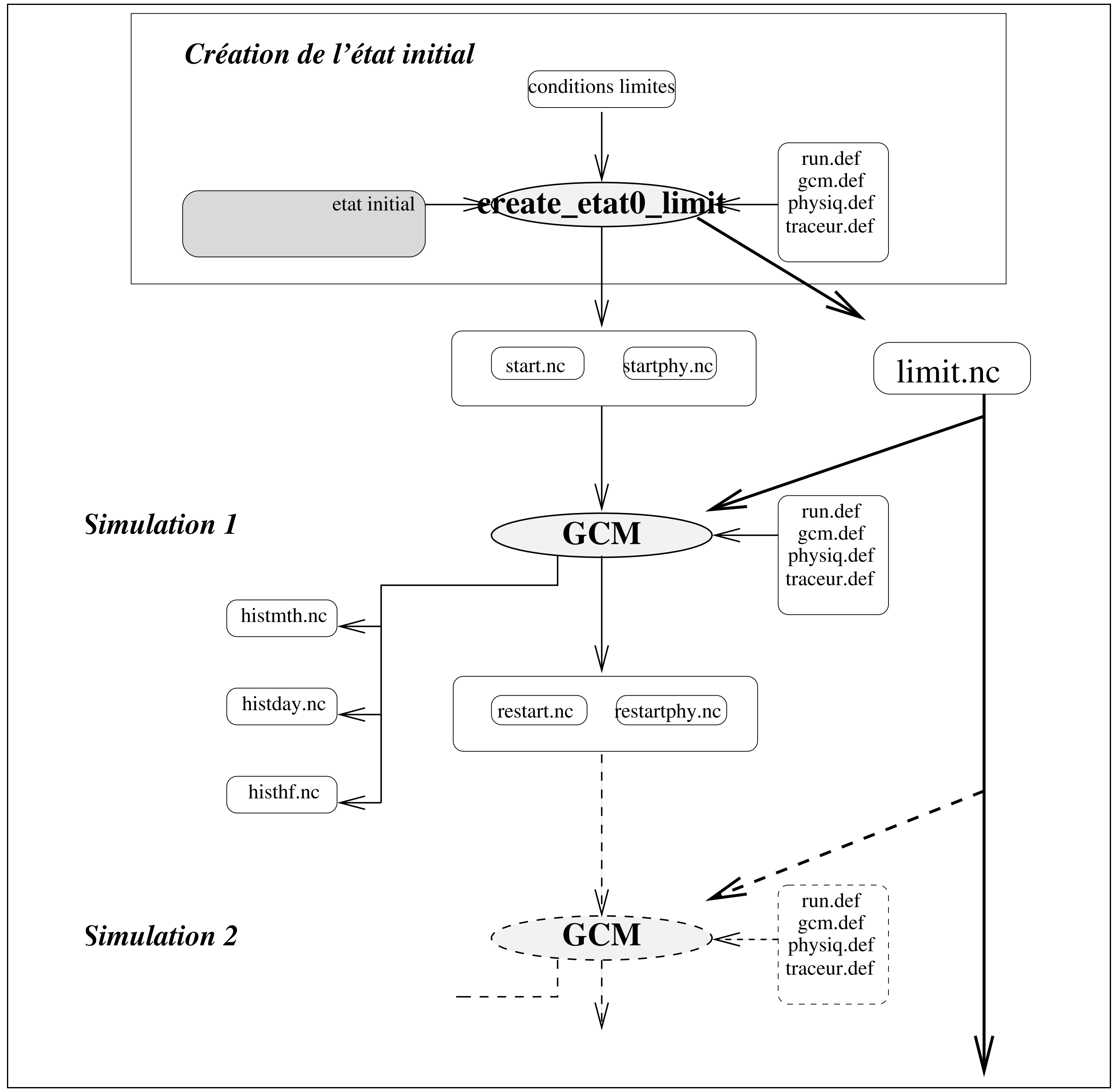


FIG. 6.4 – Entrées/Sorties

Chapitre 7

Faire tourner le modèle : première simulation

Attention manuel martien. Se reporter a la page WEB de LMDZ4

Ce chapitre est destiné aux personnes n'ayant jamais utilisé le modèle du LMD. Le meilleur moyen de prendre contact étant sans doute de réaliser une simulation, voici la marche à suivre. A partir de cet exemple, vous pourrez ensuite modifier à votre gré aussi bien les paramètres de contrôle du run que les conditions initiales. Vous trouverez dans le répertoire tous les fichiers nécessaires à l'installation, la compilation et l'execution du modèle .

7.1 Installer le modèle

- Recopiez chez vous le répertoire de base du modèle stocké dans (Son contenu est décrit section 5).

- Affectez les variables d'environnement pour le modèle :

LMDGCM Path du répertoire où vous avez installé le modèle (Path complet de)

```
setenv  LMDGCM $PATH1/LMDZ.MARS
```

LIBOGCM Path du répertoire (ce répertoire est à créer : *libo* par exemple) dans lequel seront stockées les librairies objets créées lors de la compilation du modèle.

```
setenv  LIBOGCM $PATH2/libo
```

- Installez NetCDF et affectez les variables d'environnement NCDFINC et NCDFLIB :

La dernière version du package NetCDF est disponible sur le web à l'adresse suivante :.

```
http://www.unidata.ucar.edu/packages/netcdf/faq.html#howtoget
```

Elle contient à la fois de quoi créer la librairie objet *libnetcdf.a*, les fichiers includes nécessaires, et de quoi compiler les logiciels NetCDF de base, *ncdump et ncgen*.

Afin que le modèle, lors de la compilation, trouve les librairies et includes correspondant au type de machine utilisée, il convient de déclarer les variables d'environnement **NCDFINC** et **NCDFLIB**.

NCDFLIB Nom du répertoire contenant la librairie objet (libnetcdf.a) et **NCDFINC** nom du répertoire contenant les fichiers include (netcdf.inc) NetCDF pour les différentes plateformes.

```
setenv NCDFINC $PATH3/include
setenv NCDFLIB $PATH3/lib
```

- Installez GrADS

Pour les gens travaillant au LMD, il suffit, grâce au brillantissime Laurent Fairhead, pour avoir accès à GrADS et à son environnement, d'ajouter dans son *.cshrc* (c'est à priori fait dans *.env_soft*) :

```
source /distrib/local/grads/grads.env
```

- Recopiez **makegcm** le script pour la compilation (dans $HOME/bin par exemple). Ce script est disponible dans (Ou demander au LMD).

- Enfin, assurez vous que vous avez bien accès à tous les executables nécessaires pour l'utilisation du modèle. Pensez a indiquer le chemin correspondant dans le PATH.

- fonction UNIX make
- le compilateur Fortran f90
- ncdump
- grads

7.2 Compiler le modèle

- Pour compiler par exemple le modèle martien en résolution 64x48x25, tapez (conformément au manuel de la fonction makegcm décrit section 5.4)

```
makegcm -d 64x48x25 -p mars gcm
```

On récupère l'executable **gcm.e** (le modèle compilé) dans le répertoire où on a lancé la commande makegcm.

- Autre exemple : Pour compiler le modèle martien avec 2 traceurs :

```
makegcm -d 64x48x25 -t 2 -p mars gcm
```

- Autre exemple : Pour compiler le modèle martien pour vérifier les dépassement de tableaux (debuggage : attention, le modèle est alors très lent !) :

```
makegcm -d 64x48x25 -p mars -O "-C" gcm
```

7.3 Les fichiers d'entrée (état initial)

- Récuperez dans le répertoire

```
$PATH1/LMDZ.MARS/deftank
```

le fichier de paramètres **run.def** (décrits section 6.2) nécessaire au modèle. Récupérez en plus les fichiers **callphys.def** et **esasig.def OU z2sig.def** pour le modèle martien. Les paramètres sont affectés par des valeurs usuelles, et il est conseillé de réaliser une première simulation sans les modifier.

- Recopiez les fichiers **start et startfi** (décrits section 6.2) stockés au LMD (demander) ou créer ces fichiers à partir d'une "banque d'états initiaux **start_archive** (cf. section 7.8). Ils contiennent des conditions initiales de scénario standard et permettront de valider l'installation du modèle.

7.4 Faire tourner le modèle

Placez dans un même répertoire le programme **gcm.e** et les fichiers d'entrée, puis tapez :

```
gcm.e
```

7.5 Visualisation des fichiers de sortie

Si vous n'avez jamais utilisé le logiciel graphique **GrAds**, il est vivement conseillé de consacrer une petite demi-heure à se familiariser avec lui en effectuant la demonstration prévue à cet effet. Facile, rapide, elle permet d'utiliser les commandes de base. Pour cela, allez lire le fichier

`/distrib/local/grads/sample`

Prenons l'exemple de la visualisation des fichiers histmoy, histphy.
Les fichiers NetCDF histmoy.nc, histphy.nc... sont directement exploitables sous GrAdS grâce à l'utilitaire gradsnc, dont l'appel est invisible à l'utilisateur.
Pour visualiser par exemple la carte de la température dans la 5ème couche à partir du fichier histmoy :

- Visualisation sous GrAdS :

`grads` *return*

return (pour ouvrir une fenêtre landscape)

`ga-> sdfopen histmoy.nc`

`ga-> query file` (display des infos sur le fichier ouvert, dont le nom des variables stockées. Racourcis : *q file*)

`ga-> set z 5` (fixe l'altitude à la 5ème couche)

`ga-> set t 1` (fixe le temps à la première valeur stockée)

`ga-> query dims` (Indique les valeurs fixées pour les 4 dimensions. Racourcis : *q dims*)

`ga-> display temp` (display la carte de température pour la 5ème couche et pour la première valeur de temps stockée. Raccourcis : *d T*)

`ga-> clear` (pour effacer l'écran. Raccourcis : *c*)

`ga-> set gxout shaded` (mode d'affichage plein)

`ga-> display temp`

`ga-> set gxout contour` (retour au mode contour pour afficher les niveaux)

`ga-> display temp` (superpose les contours si on ne repasse pas par la commande clear)

7.6 Relancer la simulation

A la fin d'une simulation, le modèle génère des fichiers **restart** qui contiennent l'état final du modèle. Ces fichiers (au même format que les fichiers start) peuvent donc servir de conditions initiales à une nouvelle simulation.
Il suffit de les renommer :
mv restart.nc start.nc
mv restartfi.nc startfi.nc
et de relancer la simulation.

7.7 Simulations enchainées

Dans la pratique, il est recommandé d'enchainer une série de simulations de quelques jours ou quelques dizaines de jours (ou centaines à basse résolution).
Pour cela, un script nommé `run0` est disponible dans

`$PATH1/LMDZ.MARS/deftank`

Utilisation :

- Régler la durée de chaque simulation dans `run.def`
- Régler le nombre max de simulation dans l'entête de `run0`
- Recopier les fichiers de démarrage `start.nc startfi.nc` sous le nom `start0.nc startfi0.nc`.
- Taper : `run0`

`run0` lancera ainsi une série de simulations qui génèreront des fichiers de sortie indéxés (e.g. `start1, startfi1, diagfi1`, etc..) dont les fichiers `lrun1, lrun2`, etc. contenant la redirection de l'écran avec les informations sur le déroulement du run.

NOTA : pour redémarrer une série de simulations après une première série (par exemple pour partir de `start5 et startfi5`), il faut simplement inscrire l'indice des fichiers initiaux (e.g. 5) dans le fichier nommé `num_run`. Si `num_run` existe, le modèle démarrera de l'indice contenu dans `num_run`. Sinon, il demarre de `start0 et startfi0`.

NOTA : Un script est disponible pour effectuer des runs annuels avec 12 saisons de 30^o en longitude solaire comme pour la base de données (script `run_mcd` dans le répertoire `deftank` également). Ce script fonctionne avec le script run0. Il suffit de régler le nombre de simulation à 1 dans run0. Il faut recopier run.def en run.def.ref et mettre nday à 9999 dans ce dernier fichier. Il faut commenter les N mois dejà effectués et écrire N dans `num_run` si l'on veut rédémarrer à partir de startN.c On lance ensuite `run_mcd`.

7.8 Créer et changer les conditions initiales

7.8.1 Utilisation du programme "newstart"

De nombreux paramètres du modèles (par exemple, la profondeur optique de la poussière) sont stockés dans les états initiaux (fichiers NetCDF `start` et `startfi`). Pour changer ces paramètres, ou plus généralement changer la résolution du modèle, il faut utiliser le programme **newstart**.

Ce programme est aussi utilisé pour créer un état initial. En pratique, dans la plupart des cas, on reprend un ancien état initial que l'on modifie avec `newstart` .

le programme **newstart** se compile comme le GCM à la résolution voulue. Par exemple :

```
makegcm -d 64x48x25 -p mars newstart
```

Il suffit ensuite de lancer

```
newstart.e
```

Le programme donne alors deux options :

```
 A partir de quoi souhaitez vous creer vos etats initiaux ?
     0 - d un fichier start_archive
     1 - d un fichier start
```

- \- L'option "1" permet de modifier des fichiers `start.nc, startfi.nc`
- \- L'option "0" permet de de lire les informations necessaires à la création d'un nouvel état initial à partir du fichier `start_archive.nc` .

Il suffit ensuite de répondre aux questions du menu déroulant.

Le programme **newstart.e** crée les fichiers `newstart.nc` , `newstartfi.nc` , qu'il faut généralement renommer.

7.8.2 Création du fichier archive d'états initiaux : start_archive.nc

Le fichier archive `start_archive.nc` est crée à partir des fichiers `start.nc, startfi.nc` par le programme **start2archive**. Le programme **start2archive** se compile comme le GCM à la résolution voulue. Par exemple :

```
makegcm -d 64x48x25 -p mars start2archive
```

Il suffit ensuite de lancer `start2archive.e`

7.8.3 Changement de la résolution horizontale ou verticale

Exemple, pour créer des états initiaux à la résolution 32×24×25 à partir de fichiers NetCDF `start` et `startfi` à la résolution 64×48×32 :

- Créer le fichier `start_archive.nc` avec **start2archive.e** compilé à la résolution 64×48×25 avec **l'ancien fichier** `esasig.def` **OU** `z2sig.def` **précédemment utilisé**
- Créer les fichiers `newstart.nc`, `newstartfi.nc` avec **newstart.e** compilé à la résolution 32×24×15, avec **le nouveau fichier** `esasig.def` **OU** `z2sig.def`

Un processus équivalent permet de changer le nombre de traceurs, de créer un état initial pour un zoom, etc...

Chapitre 8

Version 1D du modèle martien

La partie physique du modèle peut être utilisée pour effectuer des simulations 1-D (une colonne d'atmosphère) réaliste. En pratique, la simulation est gérée à partir d'un programme principal `testphys1d.F` qui lui même, après initialisation, appelle la subroutine directrice de la physique `physiq.F` décrite dans les chapitres précédents.

8.1 Compilation

- Pour compiler par exemple le modèle martien en 1-D avec 25 couches, tapez (conformèment au manuel de la fonction makegcm décrit section 5.4)

```
makegcm -d 25 -p lmd testphys1d
```

On récupère l'executable **testphys1d.e** (le modéle compilé) dans le repertoire où on a lancé la commande makegcm.

A FAIRE

Bibliographie

[1] R. T. Clancy and S. W. Lee. A new look at dust and clouds in the Mars atmosphere : Analysis of emission-phase function sequences from global Viking IRTM observations. *Icarus*, 93 :135–158, 1991.

[2] F. Forget. Improved optical properties of the Martian atmospheric dust for radiative transfer calculations in the infrared. *Geophys. Res. Lett.*, 25 :1105–1109, 1998.

[3] F. Forget, F. Hourdin, R. Fournier, C. Hourdin, O. Talagrand, M. Collins, S. R. Lewis, P. L. Read, and J.-P. Huot. Improved general circulation models of the Martian atmosphere from the surface to above 80 km. *J. Geophys. Res.*, 104 :24,155–24,176, 1999.

[4] F. Forget, F. Hourdin, and O. Talagrand. CO_2 snow fall on Mars : Simulation with a general circulation model. *Icarus*, 131 :302–316, 1998.

[5] Y. Fouquart and B. Bonnel. Computations of solar heating of the Earth's atmosphere : A new parametrization. *Contrib. Atmos. Phys.*, 53 :35–62, 1980.

[6] J. R. Holton. *An introduction to dynamic meteorology*, volume 23 of *Internal geophysics series*. Academic Press, second edition, 1979.

[7] C. Hourdin, J.-L. Dufresnes, R. Fournier, and F. Hourdin. Net exchange reformulation of radiative transfer in the CO_2 15μm band on mars. Article in preparation, 2000.

[8] F. Hourdin. A new representation of the CO_2 15 μm band for a Martian general circulation model. *J. Geophys. Res.*, 97(E11) :18,319–18,335, 1992.

[9] F. Hourdin and A. Armengaud. Test of a hierarchy of finite-volume schemes for transport of trace species in an atmospheric general circulation model. *Mon. Wea. Rev.*, 127 :822–837, 1999.

[10] F. Hourdin, P. Le Van, F. Forget, and O. Talagrand. Meteorological variability and the annual surface pressure cycle on Mars. *J. Atmos. Sci.*, 50 :3625–3640, 1993.

[11] P. Le Van. Description de directives pour l'utilisation de la nouvelle dynamique du Modèle de Circulation Générale. Technical Report 147, Laboratoire de Météorologie Dynamique du CNRS, 24 rue Lhomond/75 231 Paris CEDEX 05, Note interne 1989.

[12] S. R. Lewis, M. Collins, P. L. Read, F. Forget, F. Hourdin, R. Fournier, C. Hourdin, O. Talagrand, and J.-P. Huot. A climate database for Mars. *J. Geophys. Res.*, 104 :24,177–24,194, 1999.

[13] F. Lott and M. Miller. A new sub-grid scale orographic drag parametrization : its formulation and testing. *Q. J. R. Meteorol. Soc.*, 123 :101–128, 1997.

[14] M. E. Ockert-Bell, J. F. Bell III, C. McKay, J. Pollack, and F. Forget. Absorption and scattering properties of the Martian dust in the solar wavelengths. *J. Geophys. Res.*, 102 :9039–9050, 1997.

[15] N. O. Rennò, M. L. Burkett, and M. P. Larkin. A simple thermodynamical theory for dust devils. *J. Atmos. Sci.*, 55 :3244–3252, 1998.

[16] O. B. Toon, C. P. McKay, T. P. Ackerman, and K. Santhanam. Rapid calculation of radiative heating rates and photodissociation rates in inhomogeneous multiple scattering atmospheres. *J. Geophys. Res.*, 94 :16,287–16,301, 1989.