

Documentation du modèle de circulation générale  
atmosphérique LMDZ

14 novembre 2019

# Table des matières

<b>1</b>	<b>Généralités</b>	<b>4</b>
1.1	LMDZ : un modèle de circulation générale atmosphérique . . . . .	4
1.2	Composante atmosphérique du modèle de climat . . . . .	5
1.3	LMDZ et les planètes . . . . .	5
1.4	Découpage et raccordement des parties dynamique et physique . . . . .	5
1.5	Généralités informatiques . . . . .	5
<b>2</b>	<b>Le cœur dynamique de LMDZ</b>	<b>8</b>
2.1	Les équations primitives . . . . .	8
2.2	La grille horizontale . . . . .	8
2.3	La coordonnée verticale . . . . .	8
2.3.1	Coordonnées hybrides . . . . .	8
2.3.2	Masse des mailles . . . . .	8
2.3.3	La fonction d'Exner . . . . .	9
2.4	Discrétisation des équations primitives . . . . .	10
2.4.1	Flux de masse . . . . .	10
2.4.2	Les autres équations . . . . .	12
2.5	L'intégration temporelle . . . . .	13
2.6	Advection des traceurs . . . . .	13
2.7	Le filtrage des hautes latitudes . . . . .	13
2.8	La dissipation horizontale . . . . .	13
2.9	Guidage . . . . .	14
2.9.1	Principe . . . . .	14
2.9.2	Calcul de l'intensité du rappel $\alpha$ . . . . .	17
2.9.3	Fonctionnement . . . . .	18
2.9.4	Fichiers de guidage . . . . .	20
2.9.5	Résumé des paramètres . . . . .	21
2.10	Les conditions aux limites et les paramètres de contrôle . . . . .	21
2.11	L'interface entre dynamique et physique . . . . .	21
2.12	Variables d'états et sorties . . . . .	21
<b>3</b>	<b>La physique du modèle terrestre</b>	<b>23</b>
3.1	Le moniteur de la physique . . . . .	23
3.2	La couche limite diffuse . . . . .	23
3.3	Le modèle du thermique . . . . .	23
3.4	La convection nuageuse profonde . . . . .	23
3.5	Poches froides . . . . .	23
3.6	Ajustement convectif sec . . . . .	25

3.7	L'interface avec les surfaces . . . . .	25
3.8	Les traceurs dans la partie physique du modèle . . . . .	25
3.9	La condensation à grande échelle . . . . .	25
3.10	Le transfert radiatif . . . . .	25
3.11	Les variables d'état et les entrées/sorties de la physique . . . . .	25
	3.11.1 Lecture d'un fichier contenant le champ d'ozone . . . . .	25
3.12	Modèle uni-colonne . . . . .	29

Pour toutes les parties où c'est pertinent, la section se termine par un arbre des procédures concernées, procédures qu'on répertorie par ailleurs pour l'index général. Notamment, pour chacune des paramétrisations physiques, on met :

- principes physiques ;
- équations ;
- discrétisation ;
- arbres des procédures concernées ;
- paramètres de contrôle dans les `.def`.

# Chapitre 1

## Généralités

### 1.1 LMDZ : un modèle de circulation générale atmosphérique

*(Généralités sur ce qu'est un modèle de circulation générale atmosphérique. Equations primitives, paramétrisations physiques. Historique de LMDZ)*

LMDZ correspond à la seconde génération d'un modèle de climat développé depuis une trentaine d'années au LMD et décrit initialement par [SL84]. Ce modèle est plus modulaire et flexible<sup>1</sup> que son prédécesseur. Le "Z" de son nom se réfère à la capacité de raffinement de la grille (Zoom) obtenue grâce à une écriture généralisée de la formulation numérique avec un maillage dont les facteurs d'élongation dans les deux directions horizontales peuvent être choisis arbitrairement.

Comme la plupart des modèles de climat, le modèle LMDZ intègre sur la sphère et dans le temps les "équations primitives de la météorologie". Ces équations sont une version des équations de Navier Stokes simplifiées en supposant que l'atmosphère est à tout moment en équilibre hydrostatique sur la verticale et en négligeant les variations verticales de la géométrie horizontale (hypothèse de couche mince). Le moment cinétique par rapport à l'axe des pôles est par exemple calculé en utilisant comme distance à l'axe  $a \cos \phi$  – où  $a$  est le rayon de la planète et  $\phi$  la latitude – plutôt que  $(a + z) \cos \phi$ , qui tiendrait compte de l'altitude  $z$  d'une particule d'air au dessus de la surface.

Le modèle du LMD est bâti sur une discrétisation en différences finies de ces équations avec certaines propriétés intéressantes du schéma numérique comme la conservation de la masse, la conservation du moment cinétique par la composante axi-symétrique de l'écoulement et de la vorticit  barotrope. On revient sur ces aspects en toute fin de chapitre.

Ces équations ne peuvent pas  tre int gr es jusqu'  l' chelle visqueuse. En pratique, dans les mod les globaux, on utilise des mailles de quelques dizaines   quelques centaines de kilom tres suivant les applications. L'impact des  chelles sous-mailles sur la grande  chelle doit donc  tre repr sent  au travers de param trisations. Il faut  galement repr senter des processus fondamentaux comme le transfert de rayonnement visible et infrarouge dans l'atmosph re, les processus

---

1. La flexibilit  est ici une notion positive!

nuageux ou les interactions avec la surface. Dans le jargon des modélisateurs, ces paramétrisations sont regroupées dans la “physique” du modèle, par opposition avec le code représentant la “dynamique” de grande échelle. Dans le cas de l’utilisation du modèle de circulation pour d’autres planètes, c’est cette partie physique qui doit être largement modifiée, et notamment le calcul du transfert radiatif.

## **1.2 LMDZ : la composante atmosphérique du modèle de climat de l’IPSL**

Généralités sur la modélisation du climat et du changement climatique.  
Couplage avec les autres composantes.

## **1.3 LMDZ et les planètes**

## **1.4 Découpage et raccordement des parties dynamique et physique**

## **1.5 Généralités informatiques**

### **Modularité**

L’organisation du modèle LMDZ est très étroitement calquée sur la distinction entre partie “dynamique” – la seule partie où soient pris en compte des échanges horizontaux entre des mailles du modèle – et la partie “physique” qui peut être vue comme une juxtaposition de “colonnes” d’atmosphère.

Cette spécificité de la partie physique est exploitée en ce sens que le codage de toutes les paramétrisations est fait avec un indice interne muet qui représente la grille horizontale. Cette organisation est illustrée sur la Fig. 1.1. Chacune des parties, physique et dynamique, a ses propres fichiers de conditions initiales et de sorties.

Cette écriture se prête à la fois à la vectorisation et à la parallélisation des codes. Cette approche permet également de disposer de façon transparente d’une version unidimensionnelle du modèle de circulation. Cette approche permet également de disposer de façon transparente d’une version unidimensionnelle du modèle de circulation. Pour disposer d’un modèle unidimensionnel, il suffit d’écrire un programme dans lequel on initialise des profils météorologiques sur un point particulier du globe et dans lequel on appelle ensuite en boucle le moniteur physique.

Autre atout important, cette conception modulaire permet de gérer en parallèle des “physiques” différentes interfacées avec le même code dynamique. Ce point est essentiel pour les études menées au LMD sur Mars et sur Titan. Enfin, on peut noter qu’il existe aussi une version bidimensionnelle, latitude-altitude, du noyau dynamique du modèle qui a été abondamment utilisée sur Titan comme on le voit à la fin de ce document.

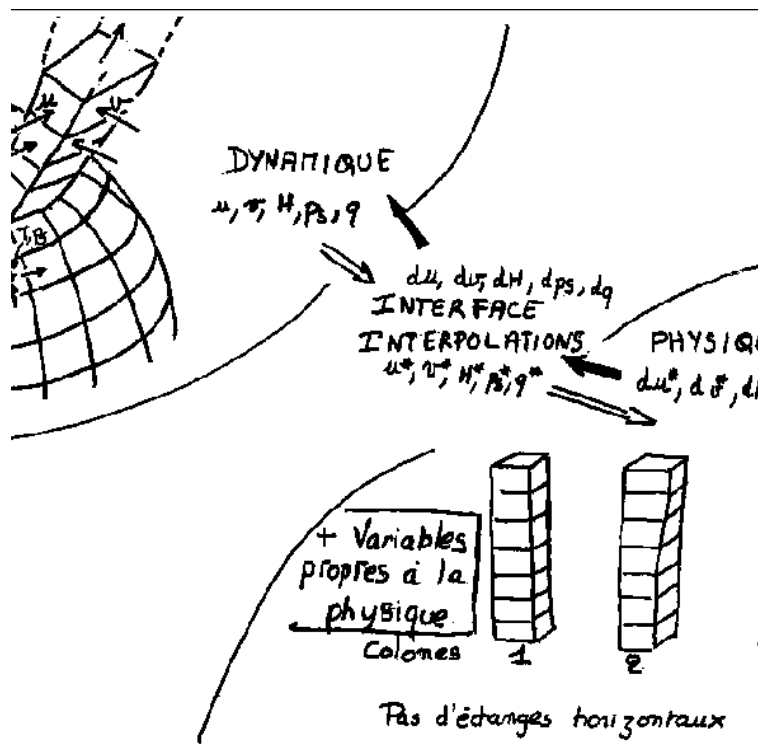


FIGURE 1.1 – Séparation entre les parties dynamique 3D et physique 1D du code LMDZ.

### Le code : architecture, compilation

Lorsque vous sortez LMDZ4 du dépôt SVN, vous obtenez au plus haut niveau la liste suivante de fichiers et répertoires.

```

arch/      create_make_gcm*  libf/      offline.def  physiq.def
bld.cfg   gcm.def             makegcm*   orchidee.def run.def
build_gcm* guide.def          makelmdz_fcm* output.def  traceur.def

```

(l'étoile signale les exécutables). Les huit fichiers `.def` servent à l'exécution des programmes `create_etat0_limit` et `gcm`. Le répertoire `libf` contient les fichiers sources en Fortran. Les autres fichiers et répertoire de cette liste :

```

arch/  bld.cfg  build_gcm*  create_make_gcm*  makegcm*  makelmdz_fcm*

```

servent à la compilation. Le répertoire `arch` contient les informations spécifiques aux différents compilateurs et machines. Deux méthodes de compilation indépendantes peuvent être utilisées. L'une utilise les scripts `create_make_gcm` et `makegcm`, l'autre les fichiers `makelmdz_fcm`, `bld.cfg` et `build_gcm`. Dans le répertoire `libf`, la logique de la répartition dans les sous-répertoires est la suivante :

**bibio** : en principe des procédures pour les entrées-sorties, ou des procédures numériques générales (indépendantes de LMDZ) ;

**dyn3d** : la partie dynamique du programme, dans sa configuration séquentielle ;  
**dyn3dpar** : l'équivalent de **dyn3d** pour la configuration parallèle ;  
**filtrez** : le filtre longitudinal, pour les régions polaires ;  
**grid** : le maillage ;  
**phylmd** : la partie physique du programme.

Normalement, en dehors de **dyn3d** et **dyn3dpar**, les fichiers sont les mêmes pour les configurations séquentielle et parallèle (en particulier la partie physique est la même).

Tout le programme est écrit en Fortran. Il a pour ambition d'être conforme à la norme Fortran 95. Les fichiers ont pour suffixes **.F** (format fixe, avec directives pour le préprocesseur), **.f90** (format libre, sans directive pour le préprocesseur), **.F90** (format libre, avec directives pour le préprocesseur) ou **.h** (fichiers inclus, normalement conformes à la fois au format fixe et au format libre).

Quelques détails sur la compilation avec **make1mdz\_fcm**. Les macros destinées au préprocesseur sont choisies d'une part dans le script **make1mdz\_fcm**, d'autre part dans le fichier de configuration **arch.fcm**. Les macros choisies par **make1mdz\_fcm** sont dans la variable **CPP\_KEY**. Les macros choisies par **arch.fcm** sont dans la variable **FPP\_DEF**. Dans le fichier **blg.cfg**, on lit que ces deux ensembles de macros sont utilisés par le préprocesseur :

```
bld::tool::fppkeys    %CPP_KEY %FPP_DEF
```

**make1mdz\_fcm** choisit les macros selon la configuration du programme, indépendamment du compilateur. Pour la configuration par défaut (c'est-à-dire sans autre option que **-arch** sur la ligne de commande), **make1mdz\_fcm** sélectionne les macros **CPP\_IOIPSL** et **CPP\_PHYS**.

La résolution spatiale du modèle est spécifiée dans le fichier **grid/dimensions.h**. **make1mdz\_fcm** crée le fichier **dimensions.h** via le script **grid/dimension/makdim**.



## Chapitre 2

# Le cœur dynamique de LMDZ

### 2.1 Les équations primitives

### 2.2 La grille horizontale

### 2.3 La coordonnée verticale

#### 2.3.1 Coordonnées hybrides

La coordonnée verticale est définie de façon implicite en donnant la pression dans la couche  $l$  du modèle sous la forme

$$p_l = A_l + B_l p_s \quad (2.1)$$

On ne définit donc pas de coordonnée verticale proprement dite ; autrement dit, la coordonnée verticale est directement l'indice de la couche variant de 1 près de la surface à  $N$  en haut du modèle.

Si on prend près de la surface  $A \sim 0$  et  $B \sim 1$  et vers le haut du modèle  $A \sim 0$  et  $B \sim 0$  avec  $A \gg B p_s$ , on voit qu'on remplit bien les conditions requises plus haut. ????? peut-être faut-il quand même reprendre le chapeau de ce § que tu as enlevé.

#### 2.3.2 Masse des mailles

Les niveaux de pression du modèle définis par la relation 2.1 sont définis aux interfaces entre les couches du modèle avec  $p_1 = p_s$  et  $p_{N+1} = 0$  où  $N$  est le nombre de couches dans le modèle (exactement comme la disposition des niveaux  $\sigma$  dans l'ancienne version). La masse d'air contenue dans une maille du modèle comprise entre les niveaux  $p_k$  et  $p_{k+1}$  n'est autre que

$$m_k = a \frac{p_k - p_{k+1}}{g} \quad (2.2)$$

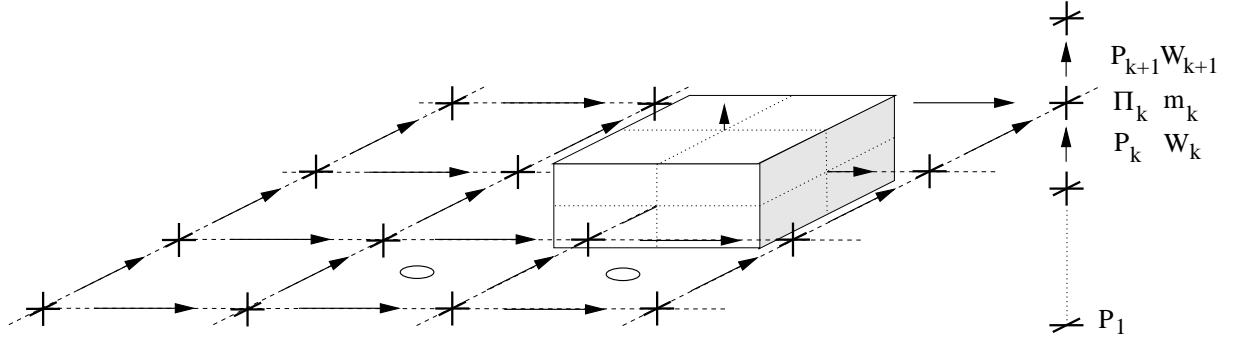


FIGURE 2.1 – Dispositon des variables dans la grille du LMD

( $a = c_u c_v$  est l'aire de la maille) ou encore

$$m_k = \frac{a}{g} [A_k - A_{k+1} + (B_k - B_{k+1}) p_s] \quad (2.3)$$

( $m$  remplace la pression extensive dans l'ancienne formulation).

### 2.3.3 La fonction d'Exner

On présente ici le calcul de la fonction d'Exner  $\Pi = C_p p^\kappa$ . Au lieu de calculer  $\Pi$  au milieu des couches à partir par exemple d'une pression extrapolée au milieu des couches à partir des niveaux de pression aux interfaces  $p_\kappa$  (ce qui est entre autres très coûteux numériquement), on propose d'utiliser une relation supplémentaire entre des niveaux  $p_k$  et la fonction d'Exner qui découle directement de considérations sur les différentes formes d'énergie dans une colonne d'atmosphère (les mêmes considérations qui étaient à l'origine du choix de la distribution relative des niveaux  $\sigma$  et des niveaux  $s$  dans l'ancienne formulation).

En effet, une propriété importante de l'approximation hydrostatique est la proportionnalité des contenus en énergie totale, interne et potentielle, à savoir qu'on a par exemple

$$\int_0^\infty \Phi \rho dz = \int_0^\infty RT \rho dz \quad (2.4)$$

Cette relation n'est plus évidente dans la formulation numérique. Cependant, sous réserve de choisir une définition adéquate pour  $\Pi$ , on peut garantir l'identité suivante :

$$\sum_{l=1}^N \Phi_l m_l - \Phi_s = \sum_{l=1}^N RT_l m_l = \sum_{l=1}^N \kappa \theta_l \Pi_l m_l \quad (2.5)$$

A partir de l'équation hydrostatique 2.29, on peut écrire

$$\Phi_l = \Phi_s + \sum_{k=1}^l \left[ \bar{\theta}^z \delta_z \Pi \right]_k \quad (2.6)$$

avec la convention (utilisée dans le modèle) :

$$\left[ \bar{\theta}^z \right]_1 = \theta_1 \quad (2.7)$$

L'énergie potentielle de gravité totale de la colonne s'écrit donc

$$\sum_{l=1}^N \Phi_l m - \Phi_s = \sum_{l=1}^N m_l \sum_{1 \leq k \leq l} \left[ \bar{\theta}^z \delta_z \Pi \right]_k \quad (2.8)$$

$$= \sum_{k=1}^N \sum_{k \leq l \leq N} \left[ \bar{\theta}^z \delta_z \Pi \right]_k m_l \quad (2.9)$$

$$= \sum_{k=1}^N \left[ \bar{\theta}^z \delta_z \Pi \right]_k \times \frac{a}{g} p_k \quad (2.10)$$

$$= \frac{a}{g} \sum_{k=1}^N \theta_k \left[ \overline{p \delta_z \Pi}^z \right]_k \quad (2.11)$$

avec les conventions

$$\left[ \overline{p \delta_z \Pi}^z \right]_1 = \frac{p_2 (\delta_z \Pi)_2}{2} + p_s (\delta_z \Pi)_1 \quad (2.12)$$

et

$$\left[ \overline{p \delta_z \Pi}^z \right]_N = \frac{p_{N-1} (\delta_z \Pi)_{N-1}}{2} \quad (2.13)$$

On voit alors que l'équation 2.5 peut être satisfaite simplement si on choisit les niveaux  $\Pi$  suivant la relation  $\overline{p \delta_z \Pi}^z = \kappa \Pi \delta_z p$  ou encore

$$\frac{a}{g} \overline{p \delta_z \Pi}^z = \kappa \Pi m \quad (2.14)$$

**Remarque :** Dans l'ancienne formulation, où les niveaux de pression étaient définis par  $p = \sigma p_s$  et où on introduisait  $s = \sigma^\kappa$  au milieu des couches, la relation précédente s'écrivait simplement

$$\left[ \overline{\sigma \delta_z s}^z \right]_N = \frac{\sigma_{N-1} \delta_z s_{N-1}}{2} \quad (2.15)$$

et était utilisée justement pour définir les niveaux  $s$ . La nouvelle formulation est donc finalement exactement équivalente à l'ancienne pour le cas particulier où on choisit  $B = \sigma$  et  $A = 0$  pour définir les niveaux de pression.

## 2.4 Discrétisation des équations primitives

### 2.4.1 Flux de masse

On peut alors introduire les trois composantes du flux de masse :

$$U = \bar{m}^X \tilde{u}, \quad V = \bar{m}^Y \tilde{v} \quad \text{et} \quad W \quad (2.16)$$

où le flux de masse vertical  $W$  est défini à partir de

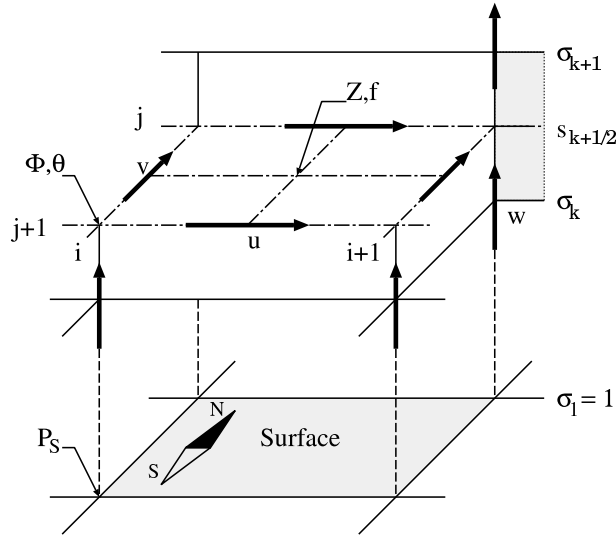


FIGURE 2.2 – Disposition des variables dans la grille du LMD

### l'équation de continuité<sup>1</sup>

$$\frac{\partial m}{\partial t} + \delta_x U + \delta_y V + \delta_z W = 0 \quad (2.17)$$

Comme dans l'ancienne version, pour résoudre cette équation, on commence par calculer la convergence de masse cumulée depuis le sommet de l'atmosphère jusqu'au niveau considéré :

$$\Omega_k = \sum_{l=k}^N (\delta_x U + \delta_y V) \quad (2.18)$$

(qu'on peut écrire de façon plus synthétique  $\delta_z \omega = -\delta_x U - \delta_y V$ ). La convergence au sol  $\Omega_1$  donne accès à l'évolution de la pression de surface

$$\frac{a}{g} \frac{\partial p_s}{\partial t} = \Omega_1 \quad (2.19)$$

et finalement, en remarquant que

$$\frac{\partial m}{\partial t} = -\frac{a}{g} \delta_z B \frac{\partial p_s}{\partial t} = -\delta_z B \Omega_1 \quad (2.20)$$

on obtient en intégrant l'équation de continuité

$$-\delta_z B \Omega_1 - \delta_z \Omega + \delta_z W = 0 \quad (2.21)$$

depuis le sommet de l'atmosphère jusqu'au niveau  $k$  :

$$1 \quad (2.22)$$

1. La notation  $\delta_X$  signifie simplement qu'on effectue la différence entre deux points consécutifs suivant la direction  $X$ . La notation  $\bar{a}^X$  signifie qu'on prend la moyenne arithmétique de la quantité  $a$  suivant la direction  $X$ .

ce qui donne l'évolution temporelle de la pression de surface à partir de la convergence du vent horizontale dans la colonne d'air. On introduit la dérivée temporelle de la pression de surface ainsi calculée dans l'équation 2.17

$$\frac{\partial p_s}{\partial t} \delta_z B + \delta_x U + \delta_y V + \delta_z W = 0 \quad (2.23)$$

qui s'intègre alors verticalement l'équation 2.17 à partir du haut du modèle pour obtenir le flux de masse vertical à tous les niveaux.

## 2.4.2 Les autres équations

En introduisant :

**le facteur de Coriolis multiplié par l'aire de la maille :**  $f = 2\Omega \sin \phi c_u c_v$   
où  $\Omega$  est la vitesse de rotation de la planète.

**la vorticité potentielle absolue :**

$$Z = \frac{\delta_x \tilde{v} - \delta_y \tilde{u} + f}{\bar{m}^{X,Y}} \quad (2.24)$$

**l'énergie cinétique**

$$K = \frac{1}{2} \left( \overline{\tilde{u}\tilde{u}}^X + \overline{\tilde{v}\tilde{v}}^Y \right) \quad (2.25)$$

**les équations du mouvement** prennent la forme suivante :

$$\frac{\partial \tilde{u}}{\partial t} - \bar{Z}^Y \bar{V}^{X,Y} + \delta_x (\Phi + K) + \bar{\theta}^X \delta_x \Pi + \frac{\overline{\bar{W}^X}^Z \delta_z \tilde{u}}{\bar{m}^X} = S_{\tilde{u}} \quad (2.26)$$

$$\frac{\partial \tilde{v}}{\partial t} + \bar{Z}^X \bar{U}^{X,Y} + \delta_y (\Phi + K) + \bar{\theta}^Y \delta_y \Pi + \frac{\overline{\bar{W}^Y}^Z \delta_z \tilde{v}}{\bar{m}^Y} = S_{\tilde{v}} \quad (2.27)$$

où  $\Pi = C_p p^\kappa$  est la fonction d'Exner au milieu de la maille.

**L'équation thermodynamique** s'écrit simplement (comme l'équation de continuité) :

$$\frac{\partial (m\theta)}{\partial t} + \delta_x (\bar{\theta}^X U) + \delta_y (\bar{\theta}^Y V) + \delta_z (\bar{\theta}^Z W) = S_\theta \quad (2.28)$$

Enfin,

**l'équation hydrostatique :**

$$\delta_z \Phi = -\bar{\theta}^z \delta_z \Pi \quad (2.29)$$

avec  $\bar{\theta}_1^z = \theta_1$  et  $\delta_z \Pi_1 = p_s^\kappa - \Pi_1$ .

## 2.5 L'intégration temporelle

## 2.6 Advection des traceurs

## 2.7 Le filtrage des hautes latitudes

## 2.8 La dissipation horizontale

Dans la procédure `inidissip`, dans le cas `ok_strato`, le tableau `zvert` est calculé à partir de la fonction :

$$f(z) = 1 + \frac{\text{dissip\_factz} - 1}{2} \left( 1 + \text{th} \frac{z - \text{dissip\_zref}}{\text{dissip\_deltaz}} \right)$$

avec les valeurs par défaut :

`dissip_factz` = 4 (sans dimension)

`dissip_deltaz` = 10 km

`dissip_zref` = 30 km

et  $z = (8\text{km}) \times \ln(p_{\text{ref}}/p)$ . La fonction  $f$  est strictement croissante sur  $\mathbb{R}$  de 1 à `dissip_factz`. Elle vaut  $\frac{\text{dissip\_factz}+1}{2}$  quand la pseudo-altitude  $z$  vaut `dissip_zref`. Cf. figure (2.3).

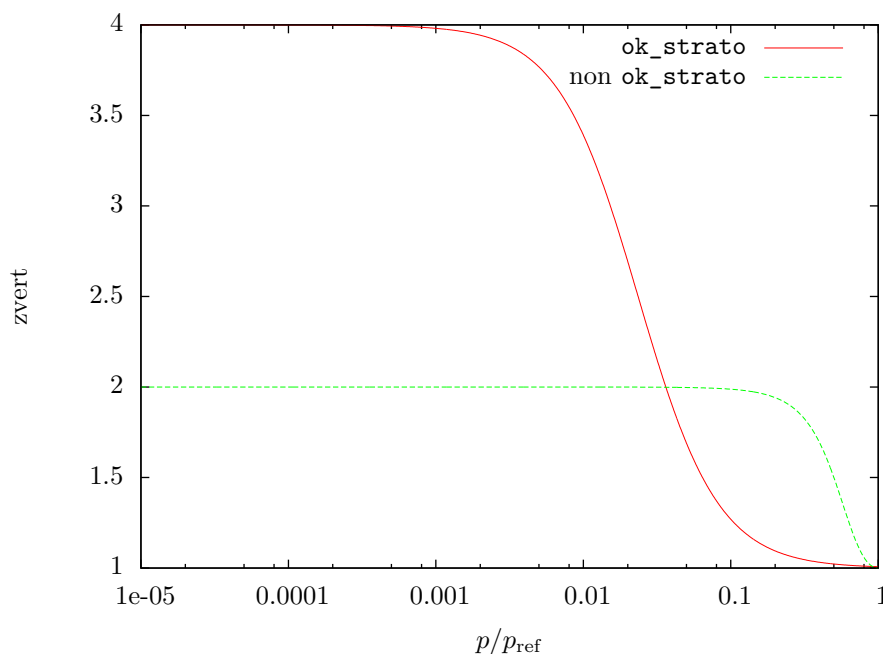


FIGURE 2.3 – Variation verticale du coefficient de dissipation, dans `inidissip`.

## 2.9 Guidage

Pour les explications, on suppose qu'on guide la variable  $u$  du modèle par une variable  $u_g$ .

### 2.9.1 Principe

On peut ajouter à certains champs du modèle une relaxation vers un état de "guidage"  $u_g$  :

$$\frac{du}{dt} = -\frac{u - u_g}{\tau}$$

où  $\tau$  est un temps de relaxation. On pourrait résoudre exactement cette équation avec une fonction linéaire  $u_g$  :

$$u_g(t) = \gamma t + u_{g0}$$

où  $\gamma$  et  $u_{g0}$  sont des constantes connues. La solution est :

$$u = u_g - \gamma\tau + (u_0 - u_{g0} + \gamma\tau) \exp(-t/\tau) \quad (2.30)$$

Considérons que l'intervalle de temps sur lequel  $u_g$  est linéaire est  $[t = 0, t_f]$  (l'intervalle entre deux lectures du champ de guidage). On a donc :

$$\gamma = \frac{u_{gf} - u_{g0}}{t_f}$$

Intéressons-nous à deux dates,  $t$  et  $t + \delta t$ , comprises entre  $t = 0$  et  $t_f$ . Nous voulons écrire  $u(t + \delta t)$  en fonction de  $u(t)$ . À partir de l'équation (2.30) :

$$u(t + \delta t) = u_g(t + \delta t) - \gamma\tau + [u(t) - u_g(t) + \gamma\tau] \exp\left(-\frac{\delta t}{\tau}\right)$$

En faisant apparaître  $u_g(t + \delta t)$  à la place de  $u_g(t)$  :

$$u(t + \delta t) = u_g(t + \delta t) \left[1 - \exp\left(-\frac{\delta t}{\tau}\right)\right] + u(t) \exp\left(-\frac{\delta t}{\tau}\right) - \beta\gamma\delta t \quad (2.31)$$

avec :

$$\beta := \frac{\tau}{\delta t} - \left(1 + \frac{\tau}{\delta t}\right) \exp\left(-\frac{\delta t}{\tau}\right) \quad (2.32)$$

$\beta$  est une fonction de  $\frac{\delta t}{\tau}$ , et on peut montrer que  $\beta$  est borné, compris entre 0 et environ 0,3. Cf. tableau (2.1). Le terme en  $\gamma$ , dans l'équation (2.31), qui tient compte de la variation de  $u_g$ , n'est actuellement pas pris en compte dans LMDZ. Donc, dans LMDZ, on écrit simplement :

$$\delta u = -\alpha(u - u_g)$$

avec :

$$\alpha = 1 - \exp(-\delta t/\tau)$$

$\frac{\delta t}{\tau}$	0	$x_1 \approx 1,8$	$+\infty$
$\beta$	0	$\approx 0,3$	0

TABLE 2.1 – Guidage, variations du coefficient  $\beta$  apparaissant lorsque l'on tient compte de la variation temporelle du champ vers lequel on guide, équation (2.32).  $x_1$  est la solution de  $e^x = x^2 + x + 1$ .

Le guidage peut aussi être en moyenne zonale au lieu de local (`guide_zon = .true.`). Faisons l'approximation que  $u_g$  ne dépend pas du temps sur un pas de temps du guidage. En notant la moyenne zonale avec une barre<sup>2</sup>, on a à résoudre :

$$(\partial_t u)(\lambda, t) = -\frac{\bar{u}(t) - \bar{u}_g}{\tau(\lambda)} \quad (2.33)$$

où  $\tau$  est une fonction connue de  $\lambda$ . Posons :

$$\tau' := \frac{1}{1/\tau}$$

En prenant la moyenne zonale de (2.33), on obtient :

$$\frac{d\bar{u}}{dt}(t) = -\frac{\bar{u}(t) - \bar{u}_g}{\tau'}$$

D'où :

$$\bar{u}(t) - \bar{u}_g = (\bar{u}_0 - \bar{u}_g)e^{-t/\tau'}$$

En remplaçant dans (2.33) et en intégrant par rapport au temps :

$$u(\lambda, t) = u_0(\lambda) + \frac{\tau'}{\tau(\lambda)}(\bar{u}_g - \bar{u}_0)(1 - e^{-t/\tau'}) \quad (2.34)$$

Nous pouvons récrire cette équation entre  $t$  et  $t + \delta t$  :

$$u(\lambda, t + \delta t) = u(\lambda, t) + \frac{\tau'}{\tau(\lambda)}[\bar{u}_g - \bar{u}(t)] \left[ 1 - \exp\left(-\frac{\delta t}{\tau'}\right) \right]$$

Est-il intéressant d'utiliser le guidage en moyenne zonale avec un rappel dépendant de la longitude (c'est-à-dire, en pratique, avec une grille zoomée en longitude) ? En faisant tendre  $t$  vers l'infini dans l'équation (2.34), on obtient :

$$u_\infty(\lambda) = u_0(\lambda) + \frac{\tau'}{\tau(\lambda)}(\bar{u}_g - \bar{u}_0)$$

Si  $\tau$  est une constante alors :

$$u_\infty(\lambda) = u_0(\lambda) + \bar{u}_g - \bar{u}_0$$

<sup>2</sup>. La moyenne zonale est prise suivant un niveau du gcm. Si `guide_reg = .true.`, elle est calculée entre `lon_min_g` et `lon_max_g`



C'est-à-dire que, par rapport à  $u_0$ ,  $u_\infty$  est simplement décalé uniformément pour que sa moyenne zonale soit  $\bar{u}_g$ . Mais si  $\tau$  n'est pas constant alors le décalage à l'infini n'est pas uniforme, il peut même être arbitrairement grand pour certaines longitudes. Illustrons cet effet sur un exemple. Si  $\tau$ , périodique de période  $2\pi$ , est défini sur  $[-\pi, \pi]$  par :

$$\tau(\lambda) = \tau_1[1 + 100 \exp(-\lambda^2)]$$

où  $\tau_1$  est une constante quelconque, et si :

$$u_0(\lambda) = U \sin(\lambda) + \bar{u}_0$$

alors on peut obtenir le graphe de la figure (2.4) pour certaines valeurs de  $U$ ,  $\bar{u}_0$  et  $\bar{u}_g$ . Le guidage zonal avec un rappel dépendant de la longitude ne

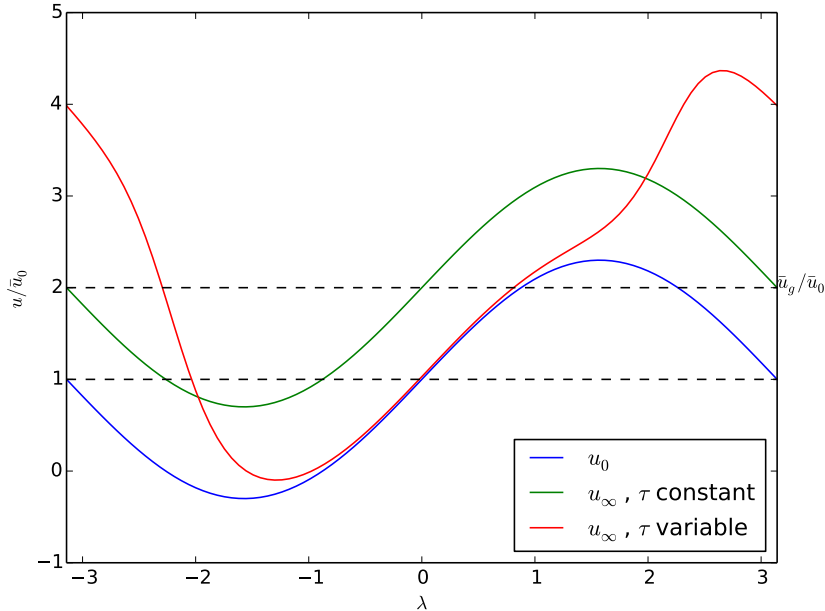


FIGURE 2.4 – Exemple de guidage zonal.  $U/\bar{u}_0 = 1,3$ ,  $\bar{u}_g/\bar{u}_0 = 2$ .

semble ainsi pas avoir de sens. Nous restreignons donc dans LMDZ l'utilisation du guidage zonal à une grille régulière en longitude. Alors  $\tau$  ne dépend pas de  $\lambda$ ,  $\tau' = \tau$  et on a alors simplement :

$$\delta u = -\alpha(\bar{u} - \bar{u}_g)$$

avec :

$$\alpha = 1 - \exp(-\delta t/\tau)$$

Une troisième possibilité est d'ajouter un forçage fixe (`guide_add = .true.`) :

$$\frac{du}{dt} = u_g/\tau$$

Le contrôle sur l'état final est moins bon, mais le forçage est mieux connu et on n'atténue pas la variabilité. En négligeant la variation de  $u_g$  sur un pas de temps de guidage, la solution est :

$$u(t + \delta t) = u(t) + u_g \frac{\delta t}{\tau}$$

donc :

$$\delta u = \alpha u_g$$

avec  $\alpha = \delta t / \tau$ . Dans la procédure `tau2alpha`, nous distinguons donc le cas de forçage fixe :

$$\alpha = \frac{\delta t}{\tau}$$

des autres cas :

$$\alpha = 1 - \exp\left(-\frac{\delta t}{\tau}\right)$$

Dans le cas d'un guidage vers un état moyen (cycle saisonnier...), il est possible d'effectuer une première simulation avec relaxation, de sauvegarder le champ de guidage  $du/dt$ , puis de l'utiliser ensuite comme forçage fixe dans une deuxième simulation.

### 2.9.2 Calcul de l'intensité du rappel $\alpha$

On ajoute  $\alpha(u_g - u)$  au champ  $u$  à chaque pas de temps du guidage, soit tous les `iperiod`. Le paramètre d'entrée est le temps de relaxation  $\tau$ , donné en jours. On a donc :

$$\alpha = 1 - \exp\left(-\frac{\text{dtvr } \text{iperiod}}{\text{daysec } \tau}\right)$$

#### Variations spatiales

Pour chaque variable guidée, on donne deux paramètres  $\tau_{min}$  et  $\tau_{max}$ . Il y a ensuite trois possibilités pour le calcul de  $\alpha$  :

**Grille régulière.** Si la grille du modèle est régulière et `guide_reg = .false.`,  $\tau$  prend la valeur constante  $\tau_{max}$ .

**Guidage régional.** Si `guide_reg = .true.`,  $\tau = \tau_{min}$  à l'intérieur de la région de guidage, et 0 à l'extérieur. La région de guidage est définie par les paramètres `lat_min_g`, `lat_max_g`, `lon_min_g`, `lon_max_g`; la transition se fait par des fonctions `tanh` de pente `1 / tau_lat`, `1 / tau_lon`.

**Grille zoomée.** Si la grille du modèle est zoomée, la valeur de  $\tau$  varie entre  $\tau_{min}$  (à l'extérieur du zoom) et  $\tau_{max}$  (au centre du zoom)<sup>3</sup> :

$$\frac{1}{\tau} = \frac{x}{\tau_{max}} + \frac{1-x}{\tau_{min}}$$

3. Pour obtenir un temps de rappel constant malgré une grille zoomée, utiliser `guide_reg = .true.`

Le coefficient  $x$  varie par une loi de puissance :

$$x = \left( \frac{A_{max} - A}{A_{max} - A_{min}} \right)^\gamma$$

où  $A$  est lié aux dimensions de la maille :

$$A = \sqrt{(\delta\lambda)^2 + (\delta\phi)^2}$$

L'exposant  $\gamma$  dépend de l'importance du zoom :

$$\gamma = -\ln(2) / \ln \left( \frac{A_{max} - 2A_{min}}{A_{max} - A_{min}} \right)$$

Le paramètre optionnel `gamma4` donne une limite supérieure de 4 à  $\gamma$ . La transition est donc moins raide et se fait plus à l'extérieur du zoom.

### Forçage fixe

Pour un forçage fixe (`guide_add = .true.`),  $\alpha$  est constant et égal à  $1/\tau_{min}$ , sauf dans le cas d'un guidage régional. Avec  $\tau_{min} = 1$  jour, On ajoute exactement le champ de guidage. Pour  $\tau_{min} = 2$  jours, on ajoute la moitié etc.

### Couche limite

La valeur de  $\alpha$  est par défaut constante sur la verticale. Si on ne veut pas guider dans la couche limite (pour laisser le modèle déterminer sa structure et les flux de surface), on peut désactiver le paramètre `guide_BL`. On a alors une modulation verticale de  $\alpha$ , qui devient faible dans la couche limite ( $\sigma < 0.85$ ) :

$$\alpha = \alpha * \left( 1 - \tanh \left( \frac{\sigma - 0.85}{0.05} \right) \right) / 2$$

## 2.9.3 Fonctionnement

### Appel

La routine `GUIDE_MAIN` est appelée depuis `LEAPFROG` avec la période du pas Matsuno : `iperiod` (5 pas de temps de la dynamique en standard). Le pas de temps est donc  $dt = dtvr \cdot iperiod$ , mais c'est le compteur de pas de temps de la dynamique `itau` qui est utilisé à l'intérieur du guide. Au premier appel, on appelle la routine `GUIDE_INIT` qui initialise les variables et alloue la mémoire. On calcule aussi les coefficients  $\alpha$  et, si `ini_anal = .true.`, on initialise les champs du GCM à ceux de guidage.

### Lecture du guidage

La lecture des fichiers de guidage se fait dans `GUIDE_READ` ou `GUIDE_READ2D` à une fréquence donnée par `iguide_read`. Si `iguide_read = n > 0`, la lecture se fait  $n$  fois par jour, le premier état étant lu au premier appel de `GUIDE_MAIN`. Si `iguide_read = n < 0`, la lecture est tous les  $n$  jours. Si `iguide_read = 0`, la lecture se fait une seule fois.

Le nouvel état est stocké dans les variables *unat2, tnat2, etc*, l'ancien dans *unat1, tnat1, etc*. Ces variables sont sur la grille originale des fichiers de guidage (mêmes niveaux verticaux) et gardent les mêmes unités. Si le fichier de guidage est en 2 dimensions (lat - P, `guide_2D = .true.`), les variables sont étendues à l'identique en longitude.

### Conversion du guidage

A un intervalle de temps donné par `iguide_int`, les variables de guidage sont interpolées dans `GUIDE_INTERP` sur les niveaux de pression du gcm, et converties dans les mêmes unités. Les nouvelles variables sont *ugui1, ugui2, tgui1, tgui2, etc*.

**Interpolation.** La pression au milieu des couches est recalculée en chaque point du modèle. Pour les variables de guidage, la pression à chaque niveau est soit donnée par les niveaux fixes, soit calculée à partir de *ps, ap* et *bp* si `guide_modele = .true.` L'interpolation verticale est faite ensuite dans `PRES2LEV`.

Le paramètre `guide_invertp` indique si il est nécessaire d'inverser l'ordre des niveaux verticaux (premier niveau au sommet ou à la surface). Au premier appel, les niveaux de pression du guidage et du modèle sont envoyés vers le fichier de sortie pour vérification de la valeur de `guide_invertp`.

Les fichiers de guidage étant déjà sur la grille horizontale du modèle, il n'y a pas besoin de faire d'interpolation horizontale en ligne. Par contre, il peut être nécessaire d'inverser l'ordre des points en latitude (`guide_inverty`) si le premier point en latitude est au pôle sud.

Pour vérifier le besoin d'inverser ou pas sur la verticale et l'horizontale, les champs de *u* et *T* du modèle et du guidage (non interpolé) sont sauvegardés au premier pas de temps dans des fichiers `ucov.nc, teta.nc, unat.nc, et tnat.nc`.

**Conversion.** Les variables de guidage sont ensuite converties dans les unités du modèle :

**Vent zonal :** Multiplication par la résolution en longitude *cu* pour convertir en *ucov*.

**Vent méridien :** Multiplication par la résolution en latitude *cv* pour convertir en *vcov*

**Température :** Multiplication par  $(P/P_{ref})^\kappa$  pour convertir en température potentielle  $\theta$ , sauf si `guide_teta = .true.`

**Humidité :** Si `guide_hr = .true.` : on suppose que la variable de guidage *qnat* est une humidité relative (en %). On calcule l'humidité spécifique saturante *qsat* à partir de la température et pression du modèle. Le forçage en vapeur d'eau est alors  $qgui = qnat \cdot 0.01 \cdot qsat$ .

### Ajout du forçage

A un instant *t* entre deux lectures du fichier de guidage aux temps  $t_1, t_2$ , le guidage est calculé à partir d'une moyenne de ces deux états :

$$u_g = \frac{ugui_1(t_2 - t) + ugui_2(t - t_1)}{t_2 - t_1}$$

On calcule ensuite pour chaque variable guidée un champ *fadd* à ajouter aux variables du modèle. Pour *u* par exemple :

Variable	Nom Fichier	Nom variable	Unité	Dimensions
Vent zonal	u.nc	UWND	$\text{m.s}^{-1}$	rlonu,rlatu
Vent méridien	v.nc	VWND	$\text{m.s}^{-1}$	rlonv,rlatv
Température	T.nc	AIR	K	rlonv,rlatu
Humidité	hur.nc	RH	% ou $\text{kg} / \text{kg}$	rlonv,rlatu
Pression surface	ps.nc	SP	Pa	rlonv,rlatu
Niveaux pression	apbp.nc	AP, BP	Pa (AP)	

TABLE 2.2 – Fichiers de guidage

- Par défaut  $fadd = \alpha \cdot (u_g - ucov)$ .
- Si `guide_add = .true.`,  $fadd = \alpha \cdot u_g$
- Si `guide_zon = .true.`,  $fadd$  est remplacé par sa moyenne en longitude entre `lon_min_g` et `lon_max_g` :  $fadd = \alpha \cdot ([u_g] - [u])$ .

Si la pression au sol est modifiée, la valeur de la masse d'air de chaque couche (*masse*) est recalculée.

### Sauvegarde du forçage

Si `guide_sav = .true.`, la valeur instantanée de  $fadd$  est sauvegardée à une fréquence donnée par `iguide_sav`. Le champ  $fadd \cdot 86400/dt$  est envoyé à `GUIDE_OUT`. Le champ sauvegardé est donc de dimension  $du/dt$  avec  $dt$  en  $\text{jour}^{-1}$ . Il peut être réutilisé directement comme forçage avec `guide_add = .true.` et  $\tau = 1$ .

## 2.9.4 Fichiers de guidage

### Temps et Variables

On utilise un fichier par variable guidée. Les noms de fichiers et de variables sont données dans le tableau 2.2. Les différents états de guidage sont dans le même fichier. Le premier état correspond à l'instant initial de la simulation, on doit donc avoir par exemple pour une simulation de 30 jours à un état par jour 31 états en tout par fichier.

### Grille horizontale

Les fichiers de guidage doivent être sur la même grille horizontale que le modèle. Cette grille est différente pour les variables  $u$ ,  $v$ , et scalaires. Pour l'interpolation horizontale, les grilles se trouvent soit dans un fichier `start.nc`, soit dans `grilles_gcm.nc`.

Exemple de script *Ferret* pour l'interpolation :

```
use "grilles_gcm.nc"
use "$fileu"
use "$filev"
use "$filet"

let uwnd=$varu
let vwnd=$varv
```

```

let air=$vart

define grid/like=uwnd[d=2]/x=grille_u[d=1]/y=grille_u[d=1] grille_u
define grid/like=vwnd[d=3]/x=grille_v[d=1]/y=grille_v[d=1] grille_v
define grid/like=air[d=4]/x=grille_v[d=1]/y=grille_u[d=1] grille_T

save/clobber/file=T.nc air[d=4,g=grille_T,i=1:iip1,j=1:jjp1,$t0]
repeat/$t1tn save/file="T.nc"/append air[d=4,g=grille_T,i=1:iip1,j=1:jjp1]

save/clobber/file=v.nc vwnd[d=3,g=grille_v,i=1:$iip1,j=1:$jjm,$t0]
repeat/$t1tn save/file="v.nc"/append vwnd[d=3,g=grille_v,i=1:iip1,j=1:jjm]

save/clobber/file=u.nc uwnd[d=2,g=grille_u,i=1:iip1,j=1:jjp1,$t0]
repeat/$t1tn save/file="u.nc"/append uwnd[d=2,g=grille_u,i=1:iip1,j=1:jjp1]

```

On peut remplacer (`grilles_gcm.nc`, `grille_u`, `grille_v`) par (`start.nc`, `vitv`, `vitv`). Des scripts unix sont disponibles pour créer des fichiers de guidage à partir de réanalyses ou de sorties de LMDZ.

### Niveaux de pression

Les fichiers de guidage peuvent avoir un nombre de niveaux verticaux quelconque, mais identiques d'un fichier à l'autre. Il y a ensuite deux options :

**Niveaux de pression constante** Dans ce cas, on lit la pression dans un des fichiers de guidage, le nom de la variable étant `LEVEL` (`ncep = .true.`) ou `PRESSURE` (`ncep = .false.`). L'unité est l'hPa.

**Niveaux variables** Si `guide_modele = .true.`, la pression en chaque point à chaque niveau est calculée à partir de la pression à la surface donnée dans `ps.nc` et de coefficients  $ap$  et  $bp$  :  $p = ap + bp * ps$ . Ces coefficients sont lus dans le fichier `apbp.nc`<sup>4</sup>

### 2.9.5 Résumé des paramètres

Cf. tableau (2.3).

## 2.10 Les conditions aux limites et les paramètres de contrôle

## 2.11 L'interface entre dynamique et physique

## 2.12 Variables d'états et sorties

---

4. Dans le modèle,  $ap$  et  $bp$  sont définis aux intercouches. Pour créer `apbp.nc`, on peut simplement faire une moyenne de  $ap$ ,  $bp$  entre deux niveaux. Ces variables se trouvent par exemple dans `start.nc`.

Nom	Type	Description	Défaut
guide add	logic	forçage fixe	.FALSE.
guide zon	logic	guidage en moyenne zonale	.FALSE.
guide sav	logic	sauvegarde du forçage	.FALSE.
ini anal	logic	état initial = guidage	.FALSE.
<i>Variables guidées</i>			
guide u	logic	guidage de u	.TRUE.
guide v	logic	guidage de v	.TRUE.
guide T	logic	guidage de T	.TRUE.
guide Q	logic	guidage de Q	.TRUE.
guide P	logic	guidage de P surf	.TRUE.
guide hr	logic	guidage de Q par H.R.	.TRUE.
guide teta	logic	guidage de T par Teta	.FALSE.
<i>Fichiers de guidage</i>			
guide modele	logic	niveaux P variables	.FALSE.
guide invertp	logic	inversion niveaux verticaux	.TRUE.
guide invertv	logic	inversion Nord-Sud	.TRUE.
guide 2D	logic	fichier guidage 2D	.FALSE.
<i>Constantes de rappel (en jours)</i>			
tau min u	real	Cste min pour u	0.02
tau max u	real	Cste max pour u	10.
<i>idem pour tau min v, T, Q, P</i>			
gamma4	logic	Zone sans rappel élargie	.FALSE.
guide BL	logic	Guidage dans couche limite	.TRUE.
<i>Guidage régional (unités en degrés)</i>			
guide reg	logic	guidage régional	.FALSE.
lat min g	real	Latitude mini guidage	- 90.
lat max g	real	Latitude maxi guidage	90.
lon min g	real	Longitude mini guidage	- 180.
lon max g	real	Longitude maxi guidage	180.
tau lat	real	raideur lat guide reg	5.
tau lon	real	raideur lon guide reg	5.
<i>Fréquence des opérations (en jour<sup>-1</sup> si &lt; 0, jour si &gt; 0)</i>			
iguide read	int	Fréquence de lecture guidage	4
iguide int	int	Fréquence de conversion guidage	4
iguide sav	int	Fréquence de sauvegarde forçage	4

TABLE 2.3 – Constantes et Paramètres du guidage

## Chapitre 3

# La physique du modèle terrestre

### 3.1 Le moniteur de la physique

### 3.2 La couche limite diffuse

### 3.3 Le modèle du thermique

### 3.4 La convection nuageuse profonde

concvl appelle `cva_driver`. `cva_driver` a été créé à la révision 879 (en 2008), lors de la mise en place de la nouvelle physique. C'est une alternative à `cv_driver`, qui fait partie de l'ancienne physique.

Dans `cva_driver`, les variables dont le nom finit par 1 sont définies sur tout le globe. Les variables correspondantes sans le 1 sont compressées.

Dans la procédure `cv3_unsat`, le calcul de `b`, dans le cas où `cvflag_ice` est faux, s'écrit, avec les notations de [Ema91], si  $b > 0$  et  $E \geq 0$  :

$$\frac{db}{dp} = \frac{\sigma_s E}{M_p + 0,1\sigma_d} + \frac{10C_p T}{L_v \sigma_d \theta} \frac{d\theta}{dp}$$

### 3.5 Poches froides

La principale procédure est `wake`. `wake` est appelé uniquement par `calwake`, une seule fois par exécution de `calwake`, et `calwake` est appelé uniquement par `physiq`, une seule fois par exécution de `physiq`.

Les articles de référence sont [GL10] et [GLC10]. Cf tableau (3.1).

En notant  $\langle \cdot \rangle_z$  la moyenne verticale entre  $z = 0$  et  $z = h_w$ , la formule utilisée pour WAPE dans la procédure `wake` est :

$$\text{WAPE} = -gh_w \frac{\langle \delta\theta \rangle_z + \left(\frac{1}{\epsilon} - 1\right) (\langle \delta\theta \rangle_z \langle \bar{q}_v^x \rangle_z + \langle \bar{\theta}^x \rangle_z \langle \delta q_v \rangle_z + \langle \delta\theta \rangle_z \langle \delta q_v \rangle_z)}{\langle \bar{\theta}_v^x \rangle_z}$$

Comparer à l'équation (5) de [GL10].



TABLE 3.1 – Correspondance entre les symboles de [GLC10] et les variables de la procédure wake (par ordre alphabétique de nom de variable). Le « u » dans av\_thvu, qu, thu, tu est pour *undisturbed* (et non *updraft*).

variable dans wake	symbole dans l'article
amdwn	$M_p$
amup	$M_c$
av_dth	$\frac{1}{h_w} \int_0^{h_w} \delta\theta \, dz$
av_thvu	$\frac{1}{h_w} \int_0^{h_w} \overline{\theta_v^x} \, dz$
deltaqv	$\delta q_v$
deltatw	$\delta T$
dth	$\delta\theta$
fip	ALP
hw	$h_w$
omg	$\delta\omega$
pi	$(p/p_{\text{ref}})^\kappa$
ptop	$p_w$
pupper	$p_h$
qe	$\bar{q}_v$
qu	$\bar{q}_{v,x}$
sigmaw	$\sigma_w$
te	$\bar{T}$
the	$\bar{\theta}$
thu	$\bar{\theta}_x$
tu	$\bar{T}_x$
wape	WAPE
wdens	$D_{\text{wk}}$

Dans la procédure wake, les variables hw, ptop et ktop (indice de pression) correspondent au même niveau vertical.  $h_w$  est calculé par :

$$h_w = \frac{2 \int_{p_w}^{p_s} \delta\theta \frac{dp}{\rho g}}{\min_{[p_w, p_s]} \delta\theta}$$

qui correspond bien à l'altitude où  $\delta\theta$  s'annule si  $\delta\theta$  croît linéairement avec  $z$  à partir d'une valeur strictement négative à la surface.

Calcul de `ptop_provis` : c'est une interpolation linéaire de la pression pour  $\delta\theta = 0,2$  K entre les points  $(\delta\theta_{k-1}, p_{k-1})$  et  $(\delta\theta_k, p_k)$ , avec :

$$\begin{aligned} \delta\theta_{k-1} &< 0,2 \text{ K} \\ \delta\theta_k &\geq 0,2 \text{ K} \end{aligned}$$

$\overline{\theta}_v^x$  est donné par :

`thu * (1 + epsim1 * qu)`

où `epsim1` vaut  $\frac{1}{\epsilon} - 1$ .

### 3.6 Ajustement convectif sec

### 3.7 L'interface avec les surfaces

### 3.8 Les traceurs dans la partie physique du modèle

### 3.9 La condensation à grande échelle

### 3.10 Le transfert radiatif

### 3.11 Les variables d'état et les entrées/sorties de la physique

Le champ d'ozone est contenu dans la variable `wo`. La variable `wo` est utilisée dans la procédure `physiq` et déclarée dans le module `phys_state_var_mod`. Au premier passage dans `physiq`, la procédure `phys_state_var_init` est appelée, qui alloue `wo`. Puis, si le modèle n'est pas couplé à INCA, `wo` n'est modifié que par `ozonecm` ou bien lecture d'un fichier. Cf. figure (3.1).

#### 3.11.1 Lecture d'un fichier contenant le champ d'ozone

Nous partons d'un fichier NetCDF contenant le champ d'ozone en fonction de la latitude, de la pression et du mois de l'année. Le maillage en latitude et pression peut être quelconque. Le champ d'ozone est remaillé pour LMDZ, en latitude et en temps, dans `ce01`. Le remaillage en temps est une interpolation linéaire produisant des valeurs pour chaque 360°d'année à partir des valeurs

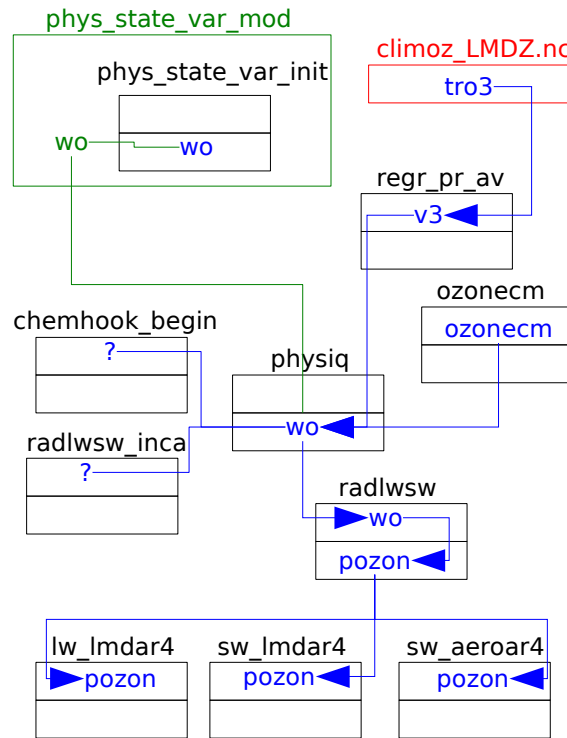


FIGURE 3.1 – Flot de la variable `wo`, contenant le champ d’ozone. Un rectangle peut correspondre à une procédure, une module ou un fichier. La couleur verte est associée aux modules, la couleur noire aux procédures, la couleur rouge aux fichiers. Une procédure est représentée par l’empilement de deux rectangles : le rectangle supérieur est l’interface de la procédure, le rectangle inférieur contient les données locales. Une variable de module est en vert au niveau du module. Une variable de module utilisée ou définie dans une procédure, et donc visible dans cette procédure par utilisation (*use association*) ou descente (*host association*) est réécrite au niveau de cette procédure en bleu. Les variables autres que les variables de module sont aussi en bleu. Si la procédure est une fonction alors, dans l’interface, le nom de la fonction est repris pour désigner son résultat. Une ligne bleue désigne une lecture ou une écriture dans un fichier ou une affectation. Une ligne verte désigne une association par utilisation ou descente. Une flèche à l’une ou l’autre extrémité d’une ligne indique le sens du transfert de la donnée (le transfert peut dans certains cas se faire dans les deux sens).

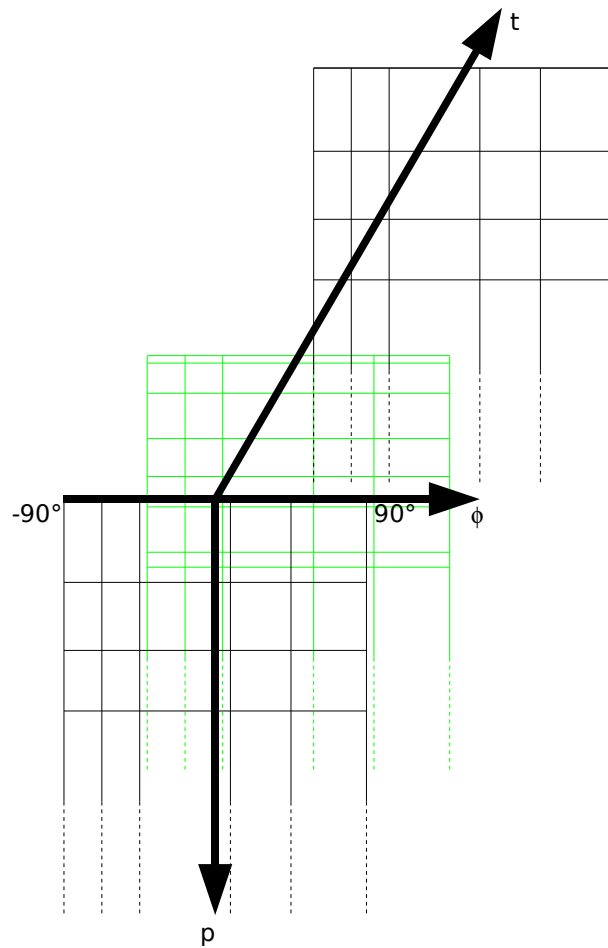


FIGURE 3.2 – Remaillage du champ d’ozone pour LMDZ. Remaillage par moyenne en latitude et pression. Remaillage par interpolation linéaire en temps. Les valeurs mensuelles, en noir, sont interprétées comme les valeurs aux milieux des mois. En vert, le remaillage latitude-pression à un jour intermédiaire. Le remaillage en latitude est le même à toutes les dates. Le remaillage en pression change chaque jour.

mensuelles. Le remaillage vertical est fait une fois par 360°d'année dans `gcm`. Nous négligeons donc l'effet sur le remaillage de la variation temporelle de pression à la surface pendant un 360°d'année. Cf. figure (3.2).

Le fichier lu par `ce01` doit s'appeler `climoz.nc`. `ce01` crée un nouveau fichier NetCDF : `climoz_LMDZ.nc`.

Justification d'un choix de programmation. Pour l'interpolation en temps dans `ce01`, nous avons choisi de permettre un fichier d'entrée contenant 12 ou 14 mois. Cf. sur le site de LMDZ : Accueil → Documentation → Guides → LMDZ pas à pas → Utilisation comme boîte noire → Le champ d'ozone. Une autre possibilité aurait été de ne permettre que 12 mois dans `climoz.nc` et de lire trois fichiers `climoz.nc` dans `ce01` : une année entière d'un fichier, le mois de décembre seulement d'un autre fichier et le mois de janvier seulement du dernier fichier. Ce serait simplifier un peu la préparation des fichiers `climoz.nc` et compliquer un peu la programmation dans `ce01`. Ce serait aussi alourdir un peu l'utilisation de `ce01` : il lui faudrait systématiquement trois fichiers de climatologie en entrée, dont deux seraient éventuellement des liens symboliques vers le troisième.

Le programme `gcm` lit `climoz_LMDZ.nc`, qui doit contenir la fraction molaire d'ozone en fonction de la latitude, de la pression et de la date. Le maillage en latitude doit déjà être celui du programme `gcm`. La date est représentée par un indice compris entre 1 et 360, indépendamment du calendrier utilisé dans `gcm`. Lorsque le programme utilise un calendrier à 360 jours, l'indice correspond au jour de l'année. Pour un autre calendrier, l'indice correspond à une période d'un trois-cent-soixantième d'année, différente d'un jour.

À chaque passage dans la procédure `physiq`, en fonction de la date courante, nous décidons de quel état de l'ozone nous avons besoin, c'est-à-dire de quelle valeur de l'indice temporel comprise entre 1 et 360. Cette valeur souhaitée de l'indice est dans la variable `ro3i` de `physiq`. Le nom `ro3i` signifie : “*required ozone index*”. `ro3i` est donc recalculé à chaque passage dans `physiq`. (Cette variable n'a pas à être rémanente.) Une autre variable de `physiq`, `co3i`, contient l'indice temporel du champ d'ozone déjà contenu dans la variable `wo`. Le nom `co3i` signifie : “*current ozone index*”. `co3i` est initialisé à 0 à la compilation et cette valeur nulle est reconnue comme signifiant que `wo` n'a pas encore été défini.

Calcul de `ro3i`. Notons ici  $L$  la durée de l'année courante et  $t$  la durée écoulée depuis le début de l'année. ( $t$  est la durée exacte, a priori pas un nombre entier de jours.) Cf. figure (3.3). Nous avons :

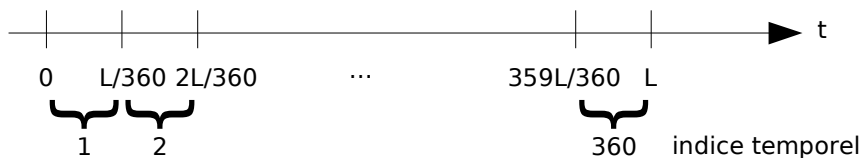


FIGURE 3.3 – Indice temporel de l'ozone en fonction de la durée écoulée depuis le début de l'année.

$$\text{ro3i} = E\left(\frac{t}{L/360}\right) + 1$$

( $E$  est la partie entière.) En principe  $t < L$  donc  $ro3i \leq 360$ . Dans `physiq`, nous ajoutons un test pour le cas où  $t/L \times 360$  donnerait une valeur supérieure ou égale à 360, par erreur d'arrondi. Dans `physiq`,  $t$  est donné par :

```
days_elapsed + jh_cur - jh_1jan
```

et  $L$  est donné par `ioget_year_len(year_cur)`. Ces deux expressions sont en jours.

Comme la variable `jh_cur` dans `physiq` a le statut OpenMP de variable privée, nous privatisons aussi `co3i` (et `ro3i`).

Le programme `gcm` ne lit pas une fois pour toutes les 360 états temporels du champ d'ozone, il lit chaque état au moment où il en a besoin, donc tous les 360<sup>e</sup>d'année.

### 3.12 Modèle uni-colonne

`ok_flux_surf` ne peut valoir vrai que pour le 1D.

# Bibliographie

- [Ema91] Kerry A. Emanuel. A Scheme for Representing Cumulus Convection in Large-Scale Models. *Journal of the Atmospheric Sciences*, 48(21) :2313–2335, 1991.
- [GL10] J.-Y. Grandpeix and J.P. Lafore. A density current parameterization coupled to Emanuel’s convection scheme. Part I : The models. *Journal of the Atmospheric Sciences*, 67(4) :881–897, 2010.
- [GLC10] J.-Y. Grandpeix, J.P. Lafore, and F. Cheruy. A density current parameterization coupled to Emanuel’s convection scheme. Part II : 1D simulations. *Journal of the Atmospheric Sciences*, 67(4) :898–922, 2010.
- [SL84] R. Sadourny and K. Laval. January and July performance of the LMD general circulation model. In A. Berger and C. Nicolis, editors, *New perspectives in climate modeling*, pages 173–197. Elsevier, 1984.