LABORATOIRE J.A. DIEUDONNÉ - NANTES UNIVERSITÉ

MÉTHODE DE COLLOCATION EN ENVIRONNEMENT PARALLÈLE POUR LA SIMULATION INTENSIVE D'ÉCOULEMENTS.

Rapport de stage

Etudiant
MARIOT Clément

Tuteur
Pr **Abide** Stéphane

19 septembre 2024

Table des matières

Mes remerciements vont à mon tuteur Stéphane Abide pour ce projet. Tout au long de ce travail, il a su m'apporter un soutien constant, une disponibilité, une écoute, une confiance et des conseils précieux et avisés à la hauteur de ses compétences et de ses réelles qualités humaines.

Ils s'adressent de même à Florence Marcotte. Pour son aide sur les aspects les plus physiques de ce projet. Sa considération vont me permettre de poursuivre ce projet.

Mes remerciements s'adressent également à Gilles Scarella pour son aide sur le choix et l'utilisation des outils informatiques pour le développement du code de calcul.

Ils s'adressent aussi à Jean-Marc Lacroix pour ses conseils de développement et ses lumières sur les considérations sur le génie logiciel. Je le remercie d'avoir partagé ses connaissances s'étendant du developpement de code de calcul jusqu'aux questions de plus bas niveau tel que le matériel.

1 Introduction

Dans ce chapitre, le système d'équations régissant l'écoulement de fluides incompressibles sera présenté. Ce système apporte un couplage entre le champ de vitesse et le champ de pression. Pour surmonter cette contrainte, on va détailler l'algorithme de projection originalement présenté par Chorin en 1967 [2], puis l'amélioration de l'ordre de convergence apportée par l'algorithme de projection à pression incrémentale. Enfin, pour corriger les erreurs commises sur les parois pour le champ de pression, un dernier algorithme de projection sera détaillé : l'algorithme de projection amélioré.

1.1 Les équations de Navier-Stokes

1.1.1 Descriptions eulérienne et lagrangienne

Le premier obstacle à la mise en équation du phénomène d'écoulement de fluide incompressible est la transcription des lois physiques décrites avec la technique lagrangienne en description eulérienne.

La description lagrangienne est l'une des deux techniques qui permettent de caractériser un écoulement. Elle consiste à suivre dans le temps les particules fluides le long de leur trajectoire. Néanmoins la description eulérienne qui repose sur le champ des vitesses est préférée car celle-ci permet de décrire l'écoulement dans une région donnée par opposition à la technique lagrangienne où ce sont les volumes élémentaires qui sont décrits.

On fixe les notations suivantes pour une grandeur représentée par la fonction f :

$$\begin{aligned} &f(x,t) && \text{pour la description eulérienne} \\ &F(X,t) = f(x(X),t) && \text{pour la description lagrangienne} \end{aligned}$$

Pour $t \in \mathbb{R}^+$ le temps écoulé depuis un temps de référence choisi, $x \in \mathbb{R}^3$ une position dans l'espace et X un volume élémentaire que l'on appelle aussi particule fluide.

Ainsi on définit deux dérivées temporelles distinctes, la dérivée relativement à une position x constante notée $\dfrac{\partial}{\partial t}$ et la dérivée relativement à une unique particule fluide X notée $\dfrac{D}{Dt}$. On rappelle la règle de dérivation composite pour une fonction dérivable f :

$$\frac{D}{Dt}f(x,t) = \frac{D}{Dt}(t)\frac{\partial}{\partial t}f + \frac{D}{Dt}(x)\frac{\partial}{\partial x}f \tag{1}$$

Par exemple l'accélération d'une particule fluide en description eulérienne s'exprime de la manière suivante, pour $U \in \mathbb{R}^3$ le vecteur vitesse d'une particule fluide, d'après (1) :

$$\begin{aligned}\frac{DU}{Dt} &= \frac{\partial U}{\partial t} + \sum_{i=1}^{3} U_i \frac{\partial U}{\partial x_i} \\ &= \frac{\partial U}{\partial t} + U \cdot \nabla U\end{aligned} \tag{2}$$

Ainsi on remarque que, même à l'état stationnaire ($i.e. \frac{\partial U}{\partial t} = 0$), l'accélération de la particule fluide peut être non nulle en conséquence du terme d'advection.

Pour un volume V de fluide dont la surface S se déplace à une vitesse U_s on obtient, grâce au théorème de Newton-Leibniz, pour une fonction F intégrable :

$$\frac{D}{Dt}\int_V F dv = \int_V \frac{\partial F}{\partial t} dv + \int_S F U_s \cdot ds = \int_V \frac{\partial F}{\partial t} + \nabla \cdot (UF)\, dv \tag{3}$$

1.1.2 Conservation de la masse

Ainsi on considère un volume de masse $m = \int_V \rho dv$. En appliquant le principe de conservation de la masse, on obtient sur un volume V quelconque :

$$0 = \frac{Dm}{Dt} = \frac{D}{Dt}\int_V \rho dv = \int_V \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (U\rho)\right] dv \tag{4}$$

Cette équation de conservation locale de masse doit être vérifiée pour tout volume de fluide V. Ainsi (4) et l'hypothèse d'incompressibilité du fluide ($i.e.$ la densité ρ est constante) donnent :

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0 \\
\Leftrightarrow\ &\frac{\partial \rho}{\partial t} + U \cdot \nabla \rho + \rho \nabla \cdot U = 0 \\
\Leftrightarrow\ &\frac{\partial \rho}{\partial t} + \rho \nabla \cdot U = 0 \\
\Leftrightarrow\ &\nabla \cdot U = 0
\end{aligned}
$$

On obtient ainsi la première équation décrivant l'écoulement d'un fluide incompressible, $i.e.$ la condition d'incompressibilité :

$$\nabla \cdot U = 0 \text{ sur } V \tag{5}$$

1.1.3 Conservation du moment cinétique

Pour l'application de la conservation du moment cinétique, on introduit le calcul suivant pour $F \in \mathbb{R}^3$:

$$
\begin{aligned}
\frac{D}{Dt}\int_V \rho F &= \int_V \frac{\partial \rho F}{\partial t} + \nabla \cdot (\rho F U)\, dv \\
&= \int_V F\frac{\partial \rho}{\partial t} + \rho \frac{\partial F}{\partial t} + F\left(\nabla \cdot (\rho U)\right) + \rho\left(U \cdot \nabla\right) F dv \\
&= \int_V \rho \frac{\partial F}{\partial t} + \rho\left(U \cdot \nabla\right) F dv \qquad \text{par hypothèse d'incompressibilité} \\
&= \int_V \rho \frac{DF}{Dt} dv
\end{aligned}
$$

L'hypothèse d'incompressibilité implique que $F\left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U)\right) = 0$.

Avant d'appliquer la conservation du moment, un bilan des forces est nécessaire. Pour les forces volumiques, on inclut uniquement la gravité, notée :

$$F_g = \int_V \rho g dv \tag{6}$$

Les forces surfaciques sont données par le tenseur de contraintes σ, ainsi pour une surface S d'un volume matériel V, on note les forces surfaciques comme :

$$F_s = \int_{\partial V} \sigma \cdot dS = \int_V \nabla \cdot \sigma dv \tag{7}$$

Ainsi la conservation du moment cinétique donne :

$$\begin{aligned} &\frac{D}{Dt}\int_V \rho U dv = \int_V \rho g dv + \int_V \nabla \cdot \sigma dv \\ \Leftrightarrow &\int_V \left[\rho \frac{\partial U}{\partial t} + \rho U \cdot \nabla U - \rho g - \nabla \cdot \sigma\right] dv = 0 \end{aligned}$$

Avec le même argument qu'utilisé plus haut, comme le volume V est quelconque, on obtient l'équation locale de conservation du moment, qui est l'équation de Cauchy pour le mouvement d'un milieu continu :

$$\rho \frac{\partial U}{\partial t} + \rho U \cdot \nabla U = \rho g + \nabla \cdot \sigma \tag{8}$$

Plus spécifiquement, pour un fluide, le tenseur de contrainte s'écrit :

$$\sigma_{i,j} = -p\delta_{i,j} + \tau_{i,j} \tag{9}$$

Pour p la pression du fluide, δ est le symbole de Kronecker et τ est le tenseur de contrainte de cisaillement. Ce tenseur est défini par la loi de comportement suivante, avec μ la viscosité dynamique du fluide :

$$\tau_{i,j} = \mu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) \tag{10}$$

Ainsi en reprenant l'expression des forces surfaciques (7) :

$$\nabla \cdot \sigma = \nabla \cdot \left(-pI + \mu \left(\nabla U + \nabla U^t\right)\right) \tag{11}$$

La condition d'incompressibilité donne l'identité $\nabla \cdot (\nabla U^t) = \nabla (\nabla \cdot U) = 0$. Ainsi (11) mène à :

$$\nabla \cdot \sigma = -\nabla p + \mu (\nabla \cdot (\nabla U)) = \mu \nabla^2 - \nabla p \tag{12}$$

On remplace dans l'équation de Cauchy, pour le mouvement d'un milieu continu (8) :

$$\rho\left(\frac{\partial U}{\partial t}+(U\cdot\nabla)U\right)=-\nabla p+\rho g+\mu\nabla^2 U \tag{13}$$

Or $\mu=\rho\nu$ avec ν la viscosité cinématique, ainsi (13) se réécrit :

$$\frac{\partial U}{\partial t}+(U\cdot\nabla)\,U=-\frac{1}{\rho}\nabla p+g+\nu\nabla^2 U \tag{14}$$

Si ρ est constante alors le système est fermé, sinon il faut ajouter une équation d'état de la thermodynamique, par exemple pour lier les variables $p,\rho,T,...$

Ainsi, pour ρ constante, on obtient le système d'équations de Navier-Stokes décrivant l'écoulement d'un fluide incompressible.

$$\begin{cases} \dfrac{\partial U}{\partial t}+(U\cdot\nabla)\,U-\nu\nabla^2 U=-\dfrac{1}{\rho}\nabla p+g \\ \nabla\cdot U=0 \end{cases} \tag{15}$$

Pour fermer le problème, il est nécessaire d'ajouter des conditions aux limites compatibles.

1.2 Présentation des algorithmes de projection

Le système d'équations de Navier-Stokes (15) pour les fluides incompressibles montre deux difficultés : la condition d'incompressibilité ($i.e.\nabla\cdot U=0$) et la gestion du couplage entre le champ de vitesse U et de pression p. Pour surmonter ces obstacles, divers algorithmes ont été proposés. Nous allons nous concentrer de manière non-exhaustive sur les algorithmes de projection.

1.2.1 Algorithme de projection

Le premier algorithme de cette famille est proposé en 1967 par Chorin dans [2]. Le principe général de l'algorithme est d'estimer provisoirement le champ de vitesse $\tilde{U}$, puis de projeter ce champ de vitesse pour imposer la contrainte d'incompressibilité, pour enfin en déduire le champ de pression.

On donne l'algorithme détaillé dans la suite, découplant le système d'équations de Navier-Stokes pour les fluides incompressibles en plusieurs problèmes de Helmholtz. Ainsi on réécrit

le problème adimensionné sur Ω, un ouvert de $\mathbb{R}^3$, avec Γ la frontière de Ω dans $\mathbb{R}^3$, pour $T \in \mathbb{R}^+$ un temps final, pour U le champ de vitesse, p le champ de pression.

$$\begin{cases} \dfrac{\partial U}{\partial t} + NL(U) - \nu \nabla^2 U + \nabla p = f \text{ sur } \Omega \times [0,T] \\ \nabla \cdot U = 0 \text{ sur } \Omega \times [0,T] \\ U|_{t=0} = U_0 \text{ sur } \Omega \\ U = W \text{ sur } \Gamma \end{cases} \tag{16}$$

Avec $NL(U) = (U \cdot \nabla) U$ qui contient le terme non-linéaire du problème. De plus, on pose une discrétisation temporelle uniforme telle que le k-ème instant t_k est défini par $t_k = k\Delta t$ à partir d'un instant de référence.

Pour commencer Chorin, propose d'estimer un champ de vitesse provisoire $\tilde{U}$ du prochain itéré temporel $\tilde{U}^{k+1}$, à partir du champ de vitesse précédent U^k, par le problème de Helmholtz suivant :

$$\frac{1}{\Delta t}\left(\tilde{U}^{k+1} - U^k\right) - \nu \nabla^2 \tilde{U}^{k+1} = f(t^{k+1}) - NL(U^k) \tag{17}$$

Avec la condition aux limites suivante :

$$\tilde{U}^{k+1} = W^{k+1} \tag{18}$$

Puis l'algorithme de Chorin donne le système suivant :

$$\begin{cases} \dfrac{1}{\Delta t}\left(U^{k+1} - \tilde{U}^{k+1}\right) + \nabla p^{k+1} = 0 \\ \nabla \cdot U^{k+1} = 0 \\ U^{k+1} \cdot n = W^{k+1} \cdot n \text{ sur } \Gamma \end{cases} \tag{19}$$

Ainsi en appliquant la condition d'incompressibilité sur U^{k+1}, on en déduit le champ de pression p^{k+1} :

$$\nabla^2 p^{k+1} = \frac{1}{\Delta t}\left(\nabla \cdot \tilde{U}^{k+1}\right) \tag{20}$$

Cela permet ainsi, grâce à (19), de calculer le nouveau champ de vitesse U^{k+1} :

$$U^{k+1} = \tilde{U}^{k+1} - \frac{1}{\Delta t}\nabla p^{k+1} \tag{21}$$

L'estimation d'erreur suivante est donnée par [6] :

$$||U^{ex}_{\Delta t} - U_{\Delta t}||_{l^\infty([L^2(\Omega)]^3)} + ||U^{ex}_{\Delta t} - \tilde{U}_{\Delta t}||_{l^\infty([L^2(\Omega)]^3)} \leqslant C\Delta t$$
$$||p^{ex}_{\Delta t} - p_{\Delta t}||_{l^\infty(L^2(\Omega))} + ||U^{ex}_{\Delta t} - \tilde{U}_{\Delta t}||_{l^\infty([H^1(\Omega)]^3)} \leqslant C\Delta t^{\frac{1}{2}}$$

Rannacher montre dans [12] que cet algorithme commet une erreur sur le champ de pression en imposant $\nabla p^{k+1} \cdot n|_\Gamma = 0$. La principale faiblesse de cet algorithme est l'estimation du champ de pression. Cela conduit à une modification de l'algorithme de projection de Chorin.

1.2.2 Algorithme de projection à pression incrémentale

L'algorithme de projection à pression incrémentale, consiste à prendre en compte le champ de pression de l'instant précédent p^k, dans l'estimation de la vitesse provisoire $\tilde{U}^{k+1}$. On utilise la *Backward Differentiation Formula* proposée par Curtiss et Hirschfelder en 1952 [3]. Ainsi l'estimation de la vitesse provisoire est donnée par :

$$\frac{1}{2\Delta t}\left(3\tilde{U}^{k+1} - 4U^k + U^{k-1}\right) - \nu\nabla^2\tilde{U}^{k+1} = f(t^{k+1}) - 2NL(U^k) + NL(U^{k-1}) - \nabla p^k \tag{22}$$

Avec la condition aux limites est :

$$\tilde{U}^{k+1} = W^{k+1} \tag{23}$$

Puis, pour obtenir les champs de pression et de vitesse actuels p^{k+1} et U^{k+1}, le système suivant doit être résolu :

$$\begin{cases} \frac{1}{2\Delta t}\left(3U^{k+1} - 3\tilde{U}^{k+1}\right) + \nabla\left(p^{k+1} - p^k\right) = 0 \\ \nabla \cdot U^{k+1} = 0 \\ U^{k+1} \cdot n = W^{k+1} \cdot n \text{ sur } \Gamma \end{cases} \tag{24}$$

De même manière que précédemment, le champ de pression p^{k+1} est donné en appliquant la contrainte d'incompressibilité sur la première équation, aboutissant :

$$\nabla^2 p^{k+1} = \frac{3}{2\Delta t}\nabla \cdot \tilde{U}^{k+1} + \nabla^2 p^k \tag{25}$$

Cela permet ainsi de déterminer le nouveau champ de vitesse U^{k+1}.

Voici les résultats sur l'erreur commise démontrés par J.-L. Guermond dans [7] :

$$||U^{ex}_{\Delta t} - U_{\Delta t}||_{l^\infty([L^2(\Omega)]^3)} + ||U^{ex}_{\Delta t} - \tilde{U}_{\Delta t}||_{l^\infty([L^2(\Omega)]^3)} \leqslant C\Delta t^2$$
$$||p^{ex}_{\Delta t} - p_{\Delta t}||_{l^\infty(L^2(\Omega))} + ||U^{ex}_{\Delta t} - \tilde{U}_{\Delta t}||_{l^\infty([H^1(\Omega)]^3)} \leqslant C\Delta t$$

Ceux-ci sont valides dans le cas où la première itération utilise un schéma temporel d'ordre 1, avant de passer sur un schéma *BDF2* d'ordre 2. Ainsi il est nécessaire de commencer par estimer la pression initiale p^0, grâce à la relation :

$$\nabla^2 p^0 = \nabla \cdot f(t^0) \text{ avec } \frac{\partial}{\partial n} p^0|_\Gamma = \left(f(t^0) + \nu\nabla^2 U^0\right) \cdot |_\Gamma \tag{26}$$

Ainsi, là où l'erreur du premier algorithme est d'imposer $\nabla p^{k+1} \cdot n|_\Gamma = 0$, l'algorithme de projection à pression incrémentale impose $\nabla\left(p^{k+1} - p^k\right) \cdot n|_\Gamma = 0$. Cela revient à imposer la contrainte suivante :

$$\nabla p^{k+1} \cdot n|_\Gamma = \nabla p^k \cdot n|_\Gamma = \nabla p^0 \cdot n|_\Gamma$$

Cette condition n'étant pas physique, elle introduit des erreurs numériques sur les bords qui limitent la précision de l'algorithme.

1.2.3 Algorithme de projection amélioré

L'algorithme de projection amélioré, présenté par S. Hugues et A. Randriamampianina [8], repose sur le calcul d'un champ de pression provisoire, pour éviter la contrainte imposée par les précédents algorithmes. Ainsi, pour un instant t^{k+1}, on note $\tilde{p}^{k+1}$ le champ de pression provisoire. L'algorithme de projection amélioré commence par la détermination de ce champ de pression provisoire :

$$\begin{cases} \nabla^2 \tilde{p}^{k+1} = \nabla \cdot \left[-2NL(U^k) + NL(U^{k-1}) + f(t^{k+1}) \right] \text{ sur } \Omega \\ \dfrac{\partial}{\partial n} \tilde{p}^{k+1} = \phi \cdot n \text{ sur } \Gamma \end{cases} \tag{27}$$

$$\phi = \frac{-3W^{k+1} + 4U^k - U^{k-1}}{2\Delta t} - 2NL(U^k) + NL(U^{k-1}) + \nu\nabla^2 \left(2U^k - U^{k-1} \right) + f(t^{k+1}) \tag{28}$$

Puis, comme dans les précédents algorithmes de projection, le champ de vitesse provisoire est déterminé par l'équation suivante :

$$\frac{1}{2\Delta t} \left(3\tilde{U}^{k+1} - 4U^k + U^{k-1} \right) - \nu\nabla^2 \tilde{U}^{k+1} = -2NL(U^k) + NL(U^{k-1}) - \nabla \tilde{p}^{k+1} + f(t^{k+1}) \tag{29}$$

Où la condition suivante est imposée sur le bord Γ :

$$\tilde{U}^{k+1} = W^{k+1} \tag{30}$$

Enfin, les champ de pression et de vitesse sont déterminés de manière identique aux précédents algorithmes de projection, à partir du système suivant :

$$\begin{cases} \dfrac{1}{2\Delta t} \left(3U^{k+1} - 3\tilde{U}^{k+1} \right) + \nabla \left(p^{k+1} - \tilde{p}^{k+1} \right) = 0 \\ \nabla \cdot U^{k+1} = 0 \\ U^{k+1} \cdot n = W^{k+1} \cdot n \text{ sur } \Gamma \end{cases} \tag{31}$$

Tel que le champ de pression p^{k+1} soit résolu par un problème de Helmholtz obtenu en appliquant la condition d'incompressibilité, pour enfin en déduire le champ de vitesse U^{k+1}.

2 La méthode numérique

Cette section détaille les deux discrétisations qui seront utilisées. La première se construit sur la base des polynômes de Tchebyshev, pour pouvoir insérer des conditions aux limites non-périodiques de type Dirichlet, Neumann ou Robin. La seconde discrétisation se base sur une approximation par série de Fourier pour traiter un cas périodique. Pour chaque direction, une discrétisation doit être choisie, cela permet de choisir des géométries mixtes pour la nature des conditions aux limites. Ensuite, la résolution d'un problème de Helmholtz en 1D sera montrée puis son extension naturelle au cas 3D. Enfin, pour les directions non-périodiques, une attention particulière sera portée sur l'intégration des conditions aux limites.

2.1 Discrétisation par polynômes de Tchebyshev

Cette discrétisation est moins répandue que son équivalent spectral mais celle-ci permet de traiter des problèmes avec des conditions aux limites non-périodiques de nature variée. Ainsi, on note T_n le polynôme de Tchebyshev d'ordre n, défini par récurrence :

$$\begin{cases} T_0 = & 1 \\ T_1 = & x \\ T_{n+1} = & 2xT_n - T_{n-1} \end{cases}$$

Ces polynômes sont définis de manière équivalente par :

$$T_k(x) = \cos\left(k \cos^{-1}(x)\right) \tag{32}$$

La famille des $(T_i)_{0 \leqslant i \leqslant N}$ forment une base de $\mathbb{R}_N[X]$ et sont orthogonaux deux à deux pour le produit scalaire $(\cdot, \cdot)_\omega$ tel que, pour $u, v \in L^2(\Omega)$:

$$(u, v)_\omega = \int_{-1}^{1} uv\omega dx \tag{33}$$

Où ω est un poids tel que :

$$\omega(x) = (1 - x^2)^{-\frac{1}{2}} \tag{34}$$

Soit $N \in \mathbb{N}$, on discrétise Ω, un intervalle fermé connexe de $\mathbb{R}$, par les $N+1$ points de Gauss-Lobatto, notés x_i, pour $i = 0, \ldots, N$:

$$x_i = \cos(\frac{i\pi}{N}) \tag{35}$$

Or, le polynôme de Tchebyshev T_N atteint ses extrema ± 1 aux points de Gauss-Lobatto. On notera que ce sont aussi les zéros du polynôme $(1 - x^2)T'_N(x)$. Ainsi, en considérant l'approximation par série de polynômes de Tchebyshev suivante de la fonction réelle u :

$$u_N(x) = \sum_{j=0}^{N} \hat{u}_j T_j(x) \tag{36}$$

Où $(\hat{u}_j)_{j=0,\ldots,N}$ sont les coefficients de Tchebyshev de l'approximation. Ils sont déterminés en imposant que $\forall i \in [\![0\ ;\ N]\!]$

$$u_N(x_i) = u(x_i) \tag{37}$$

Ce qui signifie que l'approximation est exacte en chaque point de Gauss-Lobatto, ils sont nommés les points de collocation. Ainsi le polynôme défini par (36) est le polynôme d'interpolation de Lagrange de degré N sur les points de Gauss-Lobatto, donc l'approximation peut se réécrire :

$$u_N(x) = \sum_{j=0}^{N} h_j(x) u(x_j) \tag{38}$$

Où $h_j(x)$ est le polynôme de degré N défini par :

$$h_j(x) = \frac{(-1)^{j+1}(1 - x^2)T'_N(x)}{\bar{c}_j N^2 (x - x_j)} \tag{39}$$

$$\bar{c}_j = \begin{cases} 2 & \text{si } j = 0 \\ 1 & \text{si } 1 \geqslant j \geqslant N - 1 \\ 2 & \text{si } j = N \end{cases} \tag{40}$$

(39) se démontre par le fait que les points de collocation x_j sont les zéros de $(1-x^2)T'_N(x)$

et en remarquant que :

$$\lim_{x \to x_j} \frac{(1-x^2)T'_N(x)}{x - x_j} = (-1)^{j+1}\bar{c}_j N^2 \tag{41}$$

Cela permet de choisir comme inconnues la valeur en chaque point de collocation $u(x_i)$ à la place des coefficients $\hat{u}_k$.

Pour la résolution d'EDO, il est nécessaire d'introduire un moyen de différencier ce type de série de manière exacte aux points de collocation. En réécrivant (38) sur les points de collocation, on obtient :

$$u_N(x_i) = \sum_{j=0}^{N} h_j(x_i)u(x_j) \qquad i = 0, ..., N \tag{42}$$

Comme $h_j(x)$ est de classe C^∞ et en notant $h_j^{(p)}(x_i) = d_{i,j}^{(p)}$ avec $p \in \mathbb{N}$ on a :

$$u_N^{(p)}(x_i) = \sum_{j=0}^{N} d_{i,j}^{(p)}u(x_i) \qquad i = 0, ..., N \tag{43}$$

Les coefficients $d_{i,j}^{(p)}$ se déterminent en évaluant la p-ième dérivée de h_j sur le point de collocation x_i. Ces coefficients sont donnés dans [11] pour la différenciation d'ordre 1 et 2 que l'on rappelle ici :

$$\begin{aligned}
d_{i,j}^{(1)} &= \frac{\bar{c}_i}{\bar{c}_j}\frac{(-1)^{i+j}}{(x_i - x_j)} && \text{pour } 0 \leqslant i, j \leqslant N, i \neq j \\
d_{i,i}^{(1)} &= -\frac{x_i}{2(1-x_i^2)} && \text{pour } 1 \leqslant i \leqslant N-1 \\
d_{0,0}^{(1)} &= -d_{N,N}^{(1)} = \frac{2N^2+1}{6}
\end{aligned}$$

$$
\begin{aligned}
d^{(2)}_{i,j} &= \frac{(-1)^{i+j}}{\bar{c}_j}\frac{x_i^2 + x_i x_j - 2}{(1-x_i^2)(x_i - x_j)^2} && \text{pour } 1 \leqslant i \leqslant N-1, 0 \leqslant j \leqslant N, i \neq j \\
d^{(2)}_{i,i} &= -\frac{(N^2-1)(1-x_i^2)+3}{3(1-x_i^2)^2} && \text{pour } 1 \leqslant i \leqslant N-1 \\
d^{(2)}_{0,j} &= \frac{2}{3}\frac{(-1)^j}{\bar{c}_j}\frac{(2N^2+1)(1-x_j)-6}{(1-x_j)^2} && \text{pour } 1 \leqslant j \leqslant N \\
d^{(2)}_{N,j} &= \frac{2}{3}\frac{(-1)^{j+N}}{\bar{c}_j}\frac{(2N^2+1)(1-x_j)-6}{(1+x_j)^2} && \text{pour } 0 \leqslant j \leqslant N-1 \\
d^{(2)}_{0,0} &= -d^{(2)}_{N,N} = \frac{N^4-1}{15}
\end{aligned}
$$

Ces coefficients peuvent être écrits sous forme matricielle notée D^2. Ainsi sur les points de collocation (*i.e.* les points de Gauss-Lobatto) $(x_i)_{0\leqslant i\leqslant N}$, le vecteur $U = (u(x_0), \dots u(x_N))^t$ peut être différencié sur ces points tel que :

$$
\begin{pmatrix} u''(x_0) \\ \vdots \\ u''(x_N) \end{pmatrix} = D^2 U \tag{44}
$$

2.2 Discrétisation par série de Fourier

Pour résoudre le problème de Helmholtz (62) avec des conditions aux limites périodiques, on utilise des approximations par série de Fourier, dites spectrales, en notant pour u une solution à un problème de Helmholtz 1D :

$$
u_{ex}(x) = \sum_{m=-\infty}^{\infty} \widehat{u}_m e^{imx} \tag{45}
$$

Où $(\widehat{u}_m)_{m\in\mathbb{Z}}$ sont les coefficients spectraux de l'expansion en série de Fourier de la solution u_{ex}. Ainsi, en tronquant la série avec $N = 2K+1$ permet d'approcher la solution u_{ex} avec une discrétisation uniforme, contrairement à la discrétisation nécessaire à la méthode de collocation de Tchebyshev, tel que :

$$
u(x) = \sum_{m=-K}^{K} \widehat{u}_m e^{imx} \tag{46}
$$

Grâce à diverses stratégies basées sur les transformées de Fourier notées FFT (*Fast Fou-*

rier Transform) les coefficients $(\widehat{u}_m)_{m\in\mathbb{Z}}$ sont calculés efficacement. Ainsi, on peut différencier u, tel que pour $p \in \mathbb{N}$:

$$u^{(p)} = \sum_{m=-K}^{K} (im)^p \, \widehat{u} e^{imx} \tag{47}$$

Cela permet de baser la résolution du problème de Helmholtz 1D sur le même principe que pour la discrétisation de Tchebyshev. On note D la FFT donnant les coefficients spectraux et D^{-1} la transformée inverse pour revenir dans l'espace physique. Ainsi, pour le problème de Helmholtz avec conditions aux limites périodiques, on répète l'algorithme utilisé dans l'approximation par polynômes de Tchebyshev. De ce fait, à la place de la diagonalisation de l'opérateur $\tilde{D} = P\Lambda P^{-1}$, la matrice de passage P est remplacée par une FFT et une FFT inverse pour P^{-1}. La matrice diagonale Λ est remplacée par la multiplication du coefficient $\widehat{u}_m$ par $-m^2$.

Cette résolution est utilisée dans de nombreux codes de calcul car son efficacité repose, par construction, sur l'implémentation de l'algorithme de FFT. Or, celui-ci est au centre de la recherche informatique depuis de longues années, par exemple, le benchmark du Top500 des machines contient une mesure de l'efficacité des calculateurs sur l'application de FFT.

De ce fait, l'amélioration de la performance du code de calcul repose sur celle des algorithmes et de l'implémentation de FFT. L'idée de reposer la résolution du problème sur le même schéma, que ce soit pour une discrétisation par polynômes de Tchebyshev ou par série tronquée de Fourier, permet de construire un code de calcul qui rend possible un choix entre les directions périodiques et non-périodiques sans contraintes particulières.

Ainsi, un des principaux avantages de cet algorithme est cette agilité à résoudre différents problèmes de Helmholtz avec des conditions aux limites de nature variée. Mais cela se fait au prix d'une efficacité réduite dans la configuration périodique pour les trois directions, face à des algorithmes spécialement construits sur les FFT bénéficiant de leur faible coût mémoire et grande vitesse de calcul.

Donc, on remarque ici un choix dans le développement du code de calcul, où la recherche de l'efficacité ne se fait pas au prix de la généralité de celui-ci.

2.3 Problème de Helmholtz

La résolution de ce type de problème est cruciale pour les algorithmes de projection. Dans cette section seront construits les solveurs pour les problèmes de cette nature. Pour des raisons de généralisation, le solveur présenté est associé à un problème non-périodique. Cependant, pour un problème périodique, les quelques corrections à mettre en place, sont la définition de l'opérateur $\tilde{D}$ donnée dans la section précédente et la suppression de l'intégration des conditions aux limites.

2.3.1 Problème en 1D

On pose le problème de Helmholtz suivant sur $\Omega = [-1, 1]$ avec $\sigma, \nu, \alpha_{L,R}, \beta_{L,R} \in \mathbb{R}$

$$\begin{cases} \sigma u + \nu u'' = S \quad \text{sur} \quad \Omega \\ \alpha_L u(-1) + \beta_L u'(-1) = f_L \\ \alpha_R u(+1) + \beta_R u'(+1) = f_R \end{cases}$$

De plus le terme source S est dépendant du temps et de l'espace, de même que les valeurs aux bords f_L et f_R.

On discrétise Ω par les $N+1$ points de Gauss-Lobatto $(x_i)_{0 \leqslant i \leqslant N}$. On pose comme inconnues la valeur de la solution aux points de collocation notée $u_i = u(x_i)$. Ainsi lorsque que l'on écrit le problème sur les points de collocations :

$$\begin{cases} \sigma u_i + \nu \sum_{j=0}^{N} d^{(2)}_{i,j} u_j = S_i \quad 1 \leqslant i \leqslant N-1 \\ \alpha_R u_0 + \beta_R \sum_{p=0}^{N} d^{(1)}_{0,p} u_p = f_R \\ \alpha_L u_N + \beta_L \sum_{p=0}^{N} d^{(1)}_{N,p} u_p = f_L \end{cases}$$

Les deux dernières lignes vont nous servir à exprimer u_0 et u_N en fonction des points de

collocations intérieurs, ainsi :

$$\begin{cases} \alpha_R u_0 + \beta_R (d^{(1)}_{0,0} u_0 + d^{(1)}_{0,N} u_N) &= f_R - \beta_R \sum_{p=1}^{N-1} d^{(1)}_{0,p} u_p \\ \alpha_L u_N + \beta_L (d^{(1)}_{N,0} u_0 + d^{(1)}_{N,N} u_N) &= f_L - \beta_L \sum_{p=1}^{N-1} d^{(1)}_{N,p} u_p \end{cases}$$

Ce système linéaire se réécrit :

$$\begin{pmatrix} \alpha_R + \beta_R d^{(1)}_{0,0} & \beta_R d^{(1)}_{0,N} \\ \beta_L d^{(1)}_{N,0} & \alpha_L + \beta_L d^{(1)}_{N,N} \end{pmatrix} \begin{pmatrix} u_0 \\ u_N \end{pmatrix} = \begin{pmatrix} f_R - \beta_R \sum_{p=1}^{N-1} d^{(1)}_{0,p} u_p \\ f_L - \beta_L \sum_{p=1}^{N-1} d^{(1)}_{N,p} u_p \end{pmatrix} \tag{48}$$

On note :

$$CL = \begin{pmatrix} \alpha_R + \beta_R d^{(1)}_{0,0} & \beta_R d^{(1)}_{0,N} \\ \beta_L d^{(1)}_{N,0} & \alpha_L + \beta_L d^{(1)}_{N,N} \end{pmatrix} \tag{49}$$

CL est inversible lorsque des conditions limites de type Dirichlet, Neumann ou Robin sont choisies. Ainsi, on a :

$$\begin{pmatrix} u_0 \\ u_N \end{pmatrix} = CL^{-1} \begin{pmatrix} f_R - \beta_R \sum_{p=1}^{N-1} d^{(1)}_{0,p} u_p \\ f_L - \beta_L \sum_{p=1}^{N-1} d^{(1)}_{N,p} u_p \end{pmatrix} \tag{50}$$

Donc le système sur les points intérieurs de collocations s'écrit :

$$\begin{cases} \sigma u_i + \nu \sum_{j=0}^{N} d^{(2)}_{i,j} u_j = & S_i \quad 1 \leqslant i \leqslant N-1 \\ \begin{pmatrix} u_0 \\ u_N \end{pmatrix} & = CL^{-1} \begin{pmatrix} f_R - \beta_R \sum_{p=1}^{N-1} d^{(1)}_{0,p} u_p \\ f_L - \beta_L \sum_{p=1}^{N-1} d^{(1)}_{N,p} u_p \end{pmatrix} \end{cases}$$

Or on peut réécrire la première ligne de ce système pour $i \in [\![1, N-1]\!]$:

$$\begin{aligned}
\sigma u_i + \nu \sum_{j=1}^{N-1} d_{i,j}^{(2)} u_j &= S_i - d_{i,0}^{(2)} u_0 - d_{i,N}^{(2)} u_N \\
&= S_i \\
&\quad - d_{i,0}^{(2)} \left[CL_{1,1}^{-1} \left(f_R - \beta_R \sum_{p=1}^{N-1} d_{0,p}^{(1)} u_p \right) + CL_{1,2}^{-1} \left(f_L - \beta_L \sum_{p=1}^{N-1} d_{N,p}^{(1)} u_p \right) \right] \\
&\quad - d_{i,N}^{(2)} \left[CL_{2,1}^{-1} \left(f_R - \beta_R \sum_{p=1}^{N-1} d_{0,p}^{(1)} u_p \right) + CL_{2,2}^{-1} \left(f_L - \beta_L \sum_{p=1}^{N-1} d_{N,p}^{(1)} u_p \right) \right]
\end{aligned}$$

D'où, en séparant les points intérieurs au domaine, des points sur les frontières de Ω notées Γ :

$$\begin{aligned}
&S_i - d_{i,0}^{(2)} \left(CL_{1,1}^{-1} f_R + CL_{1,2}^{-1} f_L \right) - d_{i,N}^{(2)} \left(CL_{2,1}^{-1} f_R + CL_{2,2}^{-1} f_L \right) \\
&= \sigma u_i + \nu \sum_{j=1}^{N-1} \Big[d_{i,j}^{(2)} - d_{i,0}^{(2)} \left(CL_{1,1}^{-1} \beta_R d_{0,j}^{(1)} - CL_{1,2}^{-1} \beta_L d_{N,j}^{(1)} \right) \\
&\qquad - d_{i,N}^{(2)} \left(CL_{2,1}^{-1} \beta_R d_{0,j}^{(1)} - CL_{2,2}^{-1} \beta_L d_{N,j}^{(1)} \right) \Big] u_j
\end{aligned}$$

Ces $N-1$ relations déterminent la valeur de la solution u sur les points intérieurs du domaine Ω. Cela s'écrit sous forme du système linéaire suivant :

$$\tilde{D}\tilde{U} = \tilde{S} \tag{51}$$

Tel que :

$$\tilde{D} = \sigma I + \nu D^2 \tag{52}$$

$$D_{i,j}^2 = d_{i,j}^{(2)} - d_{i,0}^{(2)} \left(CL_{1,1}^{-1} \beta_R d_{0,j}^{(1)} - CL_{1,2}^{-1} \beta_L d_{N,j}^{(1)} \right) - d_{i,N}^{(2)} \left(CL_{2,1}^{-1} \beta_R d_{0,j}^{(1)} - CL_{2,2}^{-1} \beta_L d_{N,j}^{(1)} \right)$$

$$\tilde{U} = \begin{pmatrix} u_1 \\ \vdots \\ u_{N-1} \end{pmatrix} \tag{53}$$

$$\tilde{S} = \begin{pmatrix} S_1 - d^{(2)}_{1,0}\left(CL^{-1}_{1,1}f_R + CL^{-1}_{1,2}f_L\right) - d^{(2)}_{1,N}\left(CL^{-1}_{2,1}f_R + CL^{-1}_{2,2}f_L\right) \\ \vdots \\ S_i - d^{(2)}_{i,0}\left(CL^{-1}_{1,1}f_R + CL^{-1}_{1,2}f_L\right) - d^{(2)}_{i,N}\left(CL^{-1}_{2,1}f_R + CL^{-1}_{2,2}f_L\right) \\ \vdots \\ S_{N-1} - d^{(2)}_{N-1,0}\left(CL^{-1}_{1,1}f_R + CL^{-1}_{1,2}f_L\right) - d^{(2)}_{N-1,N}\left(CL^{-1}_{2,1}f_R + CL^{-1}_{2,2}f_L\right) \end{pmatrix} \tag{54}$$

D'après [11] la matrice D^2 possède $N-1$ valeurs propres distinctes (non-nulle si on ne pose pas de conditions de Neumann à droite et à gauche) donc D^2 est diagonalisable. On note P et P^{-1} les matrices de passage, ainsi que Λ la matrice diagonale contenant les valeurs propres. D'où (51) donne :

$$\left(\sigma I + \nu P \Lambda P^{-1}\right)\tilde{U} = \tilde{S} \tag{55}$$

$$\left(\sigma I + \nu \Lambda\right) P^{-1}\tilde{U} = P^{-1}\tilde{S} \tag{56}$$

Ainsi la solution du système linéaire (51) s'écrit comme $\tilde{U} = PV$, tel que pour $i \in [\![1, N-1]\!]$:

$$V_i = \frac{(P^{-1}\tilde{S})_i}{\sigma + \nu\lambda_i} \tag{57}$$

2.3.2 Problème en 3D

On considère ici l'extension en trois dimensions du précédent problème de Helmholtz sur $\Omega = [-1, 1]^3$:

$$\sigma u + \nabla^2 u = S \tag{58}$$

Le conditions limites sont similaires à celles posées pour le cas en 1D $i.e.$ soit $\alpha^{x,y,z}_{L,R}, \beta^{x,y,z}_{L,R} \in \mathbb{R}$ tels que :

$$\begin{cases} \alpha_L^x u + \beta_L^x u' = f_L^x & \text{sur } \Gamma_x^- \\ \alpha_R^x u + \beta_R^x u' = f_R^x & \text{sur } \Gamma_x^+ \\ \alpha_L^y u + \beta_L^y u' = f_L^y & \text{sur } \Gamma_y^- \\ \alpha_R^y u + \beta_R^y u' = f_R^y & \text{sur } \Gamma_y^+ \\ \alpha_L^z u + \beta_L^z u' = f_L^z & \text{sur } \Gamma_z^- \\ \alpha_R^z u + \beta_R^z u' = f_R^z & \text{sur } \Gamma_z^+ \end{cases} \tag{59}$$

Le problème est séparable, ainsi on approxime chaque direction par une série tronquée de polynômes de Tchebyshev dans chaque direction, de degré maximal N_x, N_y et N_z respectivement pour la direction x, y et z. De plus, on introduit l'opérateur de produit tensoriel suivant pour A, B, C des tenseurs d'ordre 2 représentant des opérateurs linéaires dans les direction x, y ou z et ϕ un tenseur d'ordre 3 qui est le champ de données :

$$(A \otimes B \otimes C)\phi = A_{i,p} B_{j,q} C_{k,r} \phi_{p,q,r} = R \tag{60}$$

Avec R un tenseur d'ordre 3, de taille $N_x \times N_y \times N_z$, tel que :

$$[R]_{i,j,k} = [(A \otimes B \otimes C)\phi]_{i,j,k} = \sum_{p=1}^{N_x} \sum_{q=1}^{N_y} \sum_{r=1}^{N_z} A_{i,p} B_{j,q} C_{k,r} \phi_{p,q,r} \tag{61}$$

Ainsi, en reprenant les notations précédentes, on a sur les points de collocation intérieurs de Ω :

$$(\tilde{D}_x \otimes \tilde{D}_y \otimes \tilde{D}_z)\tilde{U} = \tilde{S} \tag{62}$$

Où $\tilde{D}_{x,y,z}$ représente l'opérateur réduit de dérivation d'ordre 2 dans la direction x, y ou z, $\tilde{S}$ est le terme source réduit pour résoudre les points de collocation intérieur. Ainsi, après diagonalisation de chacun des opérateurs direction par direction on obtient :

$$\tilde{S} = \left(\left[\sigma I_x + \nu P_x \Lambda_x P_x^{-1}\right] \otimes I_y \otimes I_z\right) \tilde{U} \tag{63}$$

$$+ \left(I_x \otimes \left[\sigma I_y + \nu P_y \Lambda_y P_y^{-1}\right] \otimes I_z\right) \tilde{U} \tag{64}$$

$$+ \left(I_x \otimes I_y \otimes \left[\sigma I_z + \nu P_z \Lambda_z P_z^{-1}\right]\right) \tilde{U} \tag{65}$$

Ce qui donne le système linéaire suivant :

$$(P_x^{-1} \otimes P_y^{-1} \otimes P_z^{-1})\tilde{S} = \left([\sigma I_x + \nu \Lambda_x] P_x^{-1} \otimes I_y \otimes I_z \right) \tilde{U} \tag{66}$$

$$+ \left(I_x \otimes [\sigma I_y + \nu \Lambda_y] P_y^{-1} \otimes I_z \right) \tilde{U} \tag{67}$$

$$+ \left(I_x \otimes I_y \otimes [\sigma I_z + \nu \Lambda_z] P_z^{-1} \right) \tilde{U} \tag{68}$$

Notons $\widehat{\Lambda}_{x,y,z} = (\sigma I_x + \nu \Lambda_x \otimes I_y \otimes I_z) + (I_x \otimes \sigma I_y + \nu \Lambda_y \otimes I_z) + (I_x \otimes I_y \otimes \sigma I_z + \nu \lambda_z)$, ainsi :

$$(P_x^{-1} \otimes P_y^{-1} \otimes P_z^{-1}) \;\; \tilde{S} = \widehat{\Lambda}_{x,y,z} (P_x^{-1} \otimes P_y^{-1} \otimes P_z^{-1}) \tilde{U} \tag{69}$$

Ainsi, on note V le tenseur suivant :

$$\widehat{V} = (P_x^{-1} \otimes P_y^{-1} \otimes P_z^{-1}) \tilde{U} \tag{70}$$

De même :

$$\widehat{S} = (P_x^{-1} \otimes P_y^{-1} \otimes P_z^{-1}) \tilde{S} \tag{71}$$

Cela permet d'appliquer $\widehat{\Lambda}_{x,y,z}^{-1}$ sur $\widehat{S}$, donnant en indiciel :

$$\widehat{V}_{i,j,k} = \frac{\widehat{S}_{i,j,k}}{\nu \left(\lambda_{x,i} + \lambda_{y,j} + \lambda_{z,k} \right) + \sigma} \tag{72}$$

Avec $i \in [\![1, N_x - 1]\!]$, $j \in [\![1, N_y - 1]\!]$ et $k \in [\![1, N_z - 1]\!]$

Pour obtenir $\tilde{U}$, on remonte dans l'espace physique :

$$\tilde{U} = (P_x \otimes P_y \otimes P_z) \, \widehat{V} \tag{73}$$

2.3.3 Intégration des conditions limites

La reconstruction se fait de la même manière que pour le problème 1D, à ceci près que l'on doit effectuer la reconstruction valeurs aux bords dans le bon espace propre de chaque direction.

Pour les matrices CL_x, CL_y et CL_z, leur direction respective est donnée par l'indice mais

leur construction est identique direction par direction au cas 1D.

Ainsi, pour reconstruire les valeurs aux bords, on obtient le système suivant avec un raisonnement identique au cas en 1D.

$$\begin{cases} \begin{pmatrix} u_{0,j,k} \\ u_{N_x,j,k} \end{pmatrix} = CL_x^{-1} \begin{pmatrix} f_R^x - \beta_R^x \sum_{i=1}^{N_x-1} d_{0,i}^{(1)} u_{i,j,k} \\ f_L^x - \beta_L^x \sum_{i=1}^{N_x-1} d_{N_x,i}^{(1)} u_{i,j,k} \end{pmatrix} \text{ avec } j \in [\![1, N_y - 1]\!] \text{ et } k \in [\![1, N_z - 1]\!] \\ \begin{pmatrix} u_{i,0,k} \\ u_{i,N_y,k} \end{pmatrix} = CL_y^{-1} \begin{pmatrix} f_R^y - \beta_R^y \sum_{j=1}^{N_y-1} d_{0,j}^{(1)} u_{i,j,k} \\ f_L^y - \beta_L^y \sum_{j=1}^{N_y-1} d_{N_y,j}^{(1)} u_{i,j,k} \end{pmatrix} \text{ avec } i \in [\![1, N_x - 1]\!] \text{ et } k \in [\![1, N_z - 1]\!] \\ \begin{pmatrix} u_{i,j,0} \\ u_{i,j,N_z} \end{pmatrix} = CL_z^{-1} \begin{pmatrix} f_R^z - \beta_R^z \sum_{k=1}^{N_z-1} d_{0,k}^{(1)} u_{i,j,k} \\ f_L^z - \beta_L^z \sum_{k=1}^{N_z-1} d_{N_z,k}^{(1)} u_{i,j,k} \end{pmatrix} \text{ avec } i \in [\![1, N_x - 1]\!] \text{ et } j \in [\![1, N_y - 1]\!] \end{cases} \tag{74}$$

Cela permet de remplacer pour $i \in [\![1, N_x - 1]\!], j \in [\![1, N_y - 1]\!]$ et $k \in [\![1, N_z - 1]\!]$, les points $u_{0,j,k}, u_{N_x,j,k}, u_{i,0,k}, u_{i,N_y,k}, u_{i,j,0}$ et u_{i,j,N_z} que l'on nommera par la suite les points de collocation sur les bords. Ainsi, pour (62) on a :

$$\tilde{U}_{i,j,k} = u_{i,j,k} \text{ pour } i \in [\![1, N_x - 1]\!], j \in [\![1, N_y - 1]\!] \text{ et } k \in [\![1, N_z - 1]\!] \tag{75}$$

$$\tilde{D_x} = \sigma I_{N_x-1} + \nu D_x^2 \tag{76}$$

$$\tilde{D_y} = \sigma I_{N_y-1} + \nu D_y^2 \tag{77}$$

$$\tilde{D_z} = \sigma I_{N_z-1} + \nu D_z^2 \tag{78}$$

Tels que pour $i, l \in [\![1, N_x - 1]\!]$, $j, m \in [\![1, N_y - 1]\!]$ et $k, n \in [\![1, N_z - 1]\!]$:

$$\begin{aligned}
\left(D_x^2\right)_{i,l} &= d_{i,l}^{(2)} - d_{i,0}^{(2)} \left(CL_{x,1,1}^{-1}\beta_R^x d_{0,l}^{(1)} - CL_{x,1,2}^{-1}\beta_L^x d_{N_x,l}^{(1)}\right) - d_{i,N_x}^{(2)} \left(CL_{x,2,1}^{-1}\beta_R^x d_{0,l}^{(1)} - CL_{x,2,2}^{-1}\beta_L^x d_{N_x,l}^{(1)}\right) \\
\left(D_y^2\right)_{j,m} &= d_{j,m}^{(2)} - d_{j,0}^{(2)} \left(CL_{y,1,1}^{-1}\beta_R^y d_{0,m}^{(1)} - CL_{y,1,2}^{-1}\beta_L^y d_{N_y,m}^{(1)}\right) - d_{j,N_y}^{(2)} \left(CL_{y,2,1}^{-1}\beta_R^y d_{0,m}^{(1)} - CL_{y,2,2}^{-1}\beta_L^y d_{N_y,m}^{(1)}\right) \\
\left(D_z^2\right)_{k,n} &= d_{k,n}^{(2)} - d_{k,0}^{(2)} \left(CL_{z,1,1}^{-1}\beta_R^z d_{0,n}^{(1)} - CL_{z,1,2}^{-1}\beta_L^z d_{N_z,n}^{(1)}\right) - d_{k,N_z}^{(2)} \left(CL_{z,2,1}^{-1}\beta_R^z d_{0,n}^{(1)} - CL_{z,2,2}^{-1}\beta_L^z d_{N_z,n}^{(1)}\right)
\end{aligned}$$

Enfin, pour le terme source réduit $\tilde{S}$:

$$\tilde{S}_{i,j,k} = S_{i,j,k} + \tilde{R}_{i,j,k} \tag{79}$$

Où S contient les valeurs du terme source aux points de collocation. $\tilde{R}$ est un tenseur d'ordre 3, de taille $(N_x - 1) \times (N_y - 1) \times (N_x - 1)$, qui contient les contributions supplémentaires provenant de la réduction du problème aux points de collocation intérieurs, comme dans l'expression de (54) pour le cas en $1D$.

$$\begin{aligned}
\tilde{R}_{i,j,k} = &- d_{i,0}^{(2)} \left(CL_{x,1,1}^{-1} f_R^x + CL_{x,1,2}^{-1} f_L^x\right) - d_{i,N_x}^{(2)} \left(CL_{x,2,1}^{-1} f_R^x + CL_{x,2,2}^{-1} f_L^x\right) \\
&- d_{j,0}^{(2)} \left(CL_{y,1,1}^{-1} f_R^y + CL_{y,1,2}^{-1} f_L^y\right) - d_{j,N_y}^{(2)} \left(CL_{y,2,1}^{-1} f_R^y + CL_{y,2,2}^{-1} f_L^y\right) \\
&- d_{k,0}^{(2)} \left(CL_{z,1,1}^{-1} f_R^z + CL_{z,1,2}^{-1} f_L^z\right) - d_{k,N_z}^{(2)} \left(CL_{z,2,1}^{-1} f_R^z + CL_{z,2,2}^{-1} f_L^z\right)
\end{aligned}$$

Cela permet de prendre en compte la nature et valeur des conditions imposées aux limites du problème. En résumé, cet algorithme résout le problème sur les noeuds intérieurs en prenant en compte les conditions limites, pour ensuite reconstruire les noeuds frontières, grâce au système (74).

3 Implémentation

Le code construit, nommé Tchebycube, est écrit en Fortran90 avec l'utilisation des dernières normes pour construire sa couche supérieure. On utilise des objets pour rendre Tchebycube le plus facilement manipulable et le plus accessible possible. Néanmoins, les routines de calcul principales , i.e. les différents solveurs des problèmes de Helmholtz, sont écrites avec comme priorité l'efficacité, de telle sorte que celles-ci soient seulement maniées par l'utilisateur à l'aide des objets des couches supérieures. De plus, les considérations sur la parallélisation de Tchebycube, ainsi que les futures perspectives de son amélioration, sont discutées dans cette section.

3.1 Langage objet

La motivation de construire des objets pour ce code de calcul est de faciliter son utilisation. On prendra, comme exemple, la construction de deux types dérivées mères : `MESH` et `OPERATOR`. Ceux-ci permettent de manier les classes filles de ces objets pour utiliser la discrétisation des polynômes de Tchebyshev ou celle d'une décomposition par série tronquée de Fourier. Cela s'appuie sur le concept de type dérivé et d'héritage.

3.1.1 Les types dérivés

Les types dérivés extensibles sont apportés par la norme 2003, ainsi cela permet, par exemple, de déclarer un type `MESH` pour le maillage du domaine Ω. Ce processus conduit à un tableau nommé `MAILLAGE` de taille 3 dont chaque entrée est de type `MESH`. Chaque composante de `MAILLAGE` contient donc les données nécessaires pour mailler le problème dans une direction.

De plus, pour chaque type dérivé, des procédures internes peuvent y être définies. Elles s'appuient sur les données contenues dans ces objets. Par exemple, le type `OPERATOR` a plusieurs procédures internes permettant de différencier un champ, dans la direction spécifique de cet opérateur. Cette procédure nommée `D1` s'appelle pour `op` de type `OPERATOR` s'appelle de la manière suivante : `op%D1([arg])`

Le code de calcul Tchebycube est écrit de manière à construire des types dérivés de

plus en plus complexes sur les types les plus simples. Le but ici est d'expliciter la chaîne de construction de chaque type dérivé.

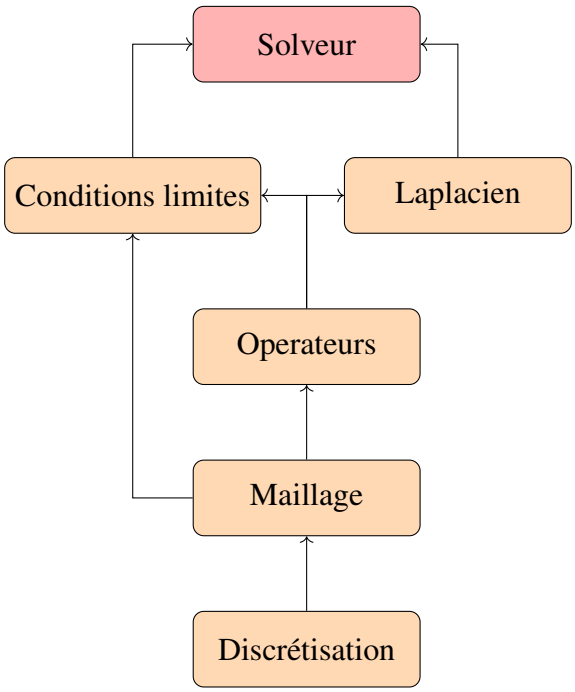

FIGURE 1 – Schéma de la hierarchie des types dérivés de Tchebycube.

En orange sont les types décrivant une unique direction et en rouge le solveur utilisant les trois directions simultanément. On remarque que le choix de la discrétisation est crucial. En effet, celui-ci se répercute sur la construction de tous les types suivants.

3.1.2 Classe et héritage

Le choix de la programmation orientée objet prend son importance avec la figure (1). En effet, l'idéal serait que pour un choix de discrétisation donné pour chaque direction (*i.e.* périodique ou non-périodique) le choix de type dérivé correspondant soit implicite, pour que le solveur correspondant à la géométrie désirée soit utilisé. L'utilisation de la notion d'héritage s'impose pour permettre cela. Ainsi, la notion d'héritage utilisée dans Tchebycube est assez simple, dans les faits l'idée est de créer une classe abstraire mère pour chaque type dérivé, puis une classe fille pour le cas périodique et une autre pour le cas non-périodique. Pour illustrer cela, on prend l'exemple de la classe de type dérivé `OPERATOR`, définie comme suivant :

```
TYPE , ABSTRACT :: T_OPERATOR_BASE
```

```
    INTEGER :: AXIS
    INTEGER :: N
CONTAINS
    ! fonctionnalité pour l'utilisateur
    PROCEDURE(EVAL_OPERATOR), DEFERRED, PASS(THIS) :: D1
    PROCEDURE(EVAL_OPERATOR), DEFERRED, PASS(THIS) :: D2
    PROCEDURE(EVAL_OPERATOR), DEFERRED, PASS(THIS) :: ID
END TYPE T_OPERATOR_BASE
```

L'attribut `ABSTRACT` permet de définir une classe mère qui n'est pas utilisable hors du processus d'héritage. Ainsi, chaque classe fille de ce type dérivé est dotée d'un champ de donnée `N` et `AXIS`. De plus, l'attribut `DEFERRED` indique que chaque classe héritée de celle-ci doit avoir définie cette procédure interne avec l'interface spécifique `EVAL_OPERATOR`.

Cela mène à la classe fille, pour une discrétisation d'une direction non-périodique, nommée `T_OPERATOR_TCHEBY` définie par :

```
TYPE , EXTENDS(T_OPERATOR_BASE)   ::  T_OPERATOR_TCHEBY
    TYPE(T_DISCR_TCHEBY) :: DISCR_D1
    TYPE(T_DISCR_TCHEBY) :: DISCR_D2
    TYPE(T_DISCR_TCHEBY) :: DISCR_ID
CONTAINS
    PROCEDURE :: INIT_OPERATOR_TCHEBY => INIT_OPERATOR_TCHEBY
    PROCEDURE :: D1 => T_OPERATOR_TCHEBY_D1
    PROCEDURE :: D2 => T_OPERATOR_TCHEBY_D2
    PROCEDURE :: ID => T_OPERATOR_TCHEBY_ID
    PROCEDURE :: GET_D1, GET_D2, GET_ID
END TYPE T_OPERATOR_TCHEBY
```

L'attribut `EXTENDS` permet d'indiquer l'héritage depuis le type dérivé `T_OPERATOR_BASE`. De plus, ce type ajoute comme champ de donnée trois `T_DISCR_TCHEBY`. On utilise des pointeurs pour définir les procédures héritées `D1`, `D2` et `ID`, pour nommer de manière explicite ces routines.

Ainsi, en construisant de manière similaire le type dérivé `T_OPERATOR_FOURIER`, l'utilisation du polymorphisme permet de faire appel à la routine `D1`, par exemple, sans se poser la question du type de discrétisation choisi sur chaque direction. D'où, pour une variable `OP_X` définie comme un type dérive `OPERATOR`, cela permet de faire appel à la routine `OP_X%D1` sans avoir besoin de savoir si une discrétisation de Tchebyshev ou de Fourier est considérée.

3.2 Parallélisation

La parallélisation du code de calcul pour le protocole MPI se repose sur la librairie 2DecompFFT [9]. Elle consiste à décomposer les données selon une grille $C \times R$, où C est le nombre de processeur par colonne et R celui dénombrant le nombre de processus par ligne. Ainsi, tous les processeurs ont accès aux données dans une direction et ont la charge d'une fraction du domaine total. Par exemple, lors de l'initialisation de 2DecompFFT les données sont alignées selon la direction x $i.e.$ on se place dans le pencil x :

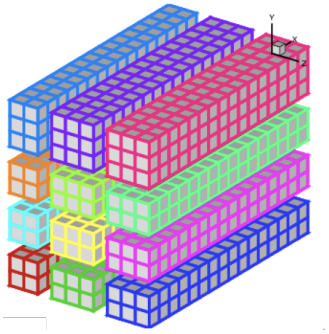

FIGURE 2 – Distribution des données dans le pencil x pour $C = 4$ et $R = 3$

3.2.1 Résolution sur les noeuds intérieurs

Comme montré précédemment dans (71), on commence par aligner dans le pencil x les données, afin d'y appliquer $P_x^{-1} \otimes I_y \otimes I_z$. Cela est réalisable par chaque processeur car comme montré sur la figure (2), pour chaque point de collocation intérieur, le processus correspondant a accès à toutes les données dans la direction x nécessaire à ce produit tensoriel. Puis, grâce à la routine `TRANSPOSE_X_TO_Y` les données sont alignées dans le pencil y (voir figure(3)) pour y appliquer sur les données $I_x \otimes P_y^{-1} \otimes I_z$.

Ensuite on passe dans le pencil z avec la routine `TRANSPOSE_Y_TO_Z` alignant les données comme sur la figure (4) pour y appliquer $I_x \otimes I_y \otimes P_z^{-1}$

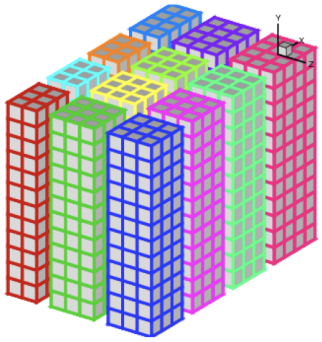

FIGURE 3 – Distribution des données dans le pencil y pour $C = 4$ et $R = 3$

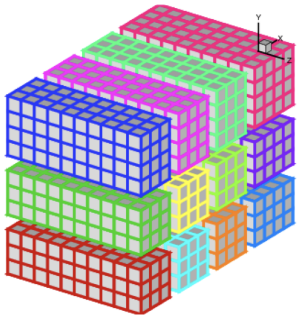

FIGURE 4 – Distribution des données dans le pencil z pour $C = 4$ et $R = 3$

Dans cette configuration, on calcule $\widehat{V}$ comme décrit par l'équation (72). Enfin, pour obtenir $\tilde{U}$, il suffit d'utiliser la relation (73). Cela revient à appliquer $I_x \otimes I_y \otimes P_z$, puis de transposer les données dans le pencil y pour y appliquer $I_x \otimes P_y \otimes I_z$. Pour ensuite, finir par transposer les données dans le pencil x et appliquer $P_x \otimes I_y \otimes I_z$.

3.2.2 Intégration points de collocation sur Γ

Outre le temps de calcul des six produits tensoriels, on remarque que la résolution d'un problème de Helmholtz est constituée de quatre transpositions des données. Celles-ci sont construites sur la base du protocole MPI `MPI_ALLTOALLV`. Ainsi, ces transpositions engendrent de lourdes communications pour un nombre élevé de processeurs. Donc pour reconstruire la solution à partir des points de collocation intérieurs, par la relation (74), il faut réaliser quatre transpositions supplémentaires, pour aligner les données et reconstruire les valeurs sur les bords. Pour éviter cela, la résolution sur Γ se fait en même temps que celle sur les points intérieurs, permettant de faire quatre transpositions d'un tenseur de taille $N_x + 1 \times N_y + 1 \times N_z + 1$ au lieu de de quatre transpositions d'un tenseur de taille $N_x - 1 \times N_y - 1 \times N_z - 1$ puis de quatre autres transpositions de taille $2\,(N_y \times N_z + N_x \times N_z + N_x \times N_y)$.

Notons pour $i \in [\![1, N_x - 1]\!], j \in [\![0, N_y]\!]$ et $k \in [\![0, N_z]\!]$:

$$(S)_i = \left(\begin{array}{c|c|c} f_R^y & f_R^y & f_R^y \\ \hline f_L^z & \widehat{S} & f_R^z \\ \hline f_L^y & f_L^y & f_L^y \end{array} \right) \tag{80}$$

$$S_{0,j,k} = f_L^x$$

$$S_{N_x,j,k} = f_R^x$$

Ainsi, en appliquant le produit tensoriel dans chaque direction avec la matrice de passage associée au tenseur S, on obtient le tenseur $\tilde{S}$. Cela permet de multiplier terme à terme $\tilde{S}$ avec le tenseur contenant les valeurs propres Λ_x, Λ_y et Λ_z. Puis, de revenir dans l'espace physique en appliquant les produits tensoriels entre ce tenseur et respectivement direction par direction avec la matrice de passage P_z, P_y et P_x.

3.2.3 Transformée de Fourier rapide

Pour le moment, les transformées de Fourier rapides (FFT) sont utilisées par l'intermédiaire de la bibliothèque FFTW3 [4]. Cette bibliothèque est écrite en C mais celle-ci comporte un wrapper permettant de faire appel à ses routines en Fortran.

Cette bibliothèque repose sur le concept de définir des plans, où la taille et le type de l'objet sont donnés, de même que le sens de la transformée de Fourier.

La routine la plus directe utilisée est `DFFTW_EXECUTE_DFT_R2C`, qui s'initialise avec la création de plan `DFFTW_PLAN_DFT_R2C_1D`, pour une transformée d'un vecteur réel aux coefficients spectraux complexes. Cet algorithme de FFT s'utilise dans une unique direction, ainsi celui-ci est utilisé lors des routines de différenciation contenu dans le type dérivé hérité de la classe `OPERATOR`. Pour appliquer la FFT inverse, donc revenir dans l'espace physique, un plan doit être construit avec `DFFTW_PLAN_DFT_C2R_1D`, puis exécuter par `DFFTW_EXECUTE_DFT_C2R`.

Néanmoins, cette routine implique plusieurs boucles pour parcourir le tenseur de données pour y appliquer, dans la direction choisie, vecteur par vecteur, la routine de FFT. Par exemple,

pour un volume de données de taille $N_x \times N_y \times N_z$ à transformer par FFT dans la première direction requiert $N_y \times N_z$ itérations de cette procédure.

Pour éviter ces boucles et tirer le maximum de performance de l'optimisation des algorithmes de FFT, une autre routine est à disposition dans la bibliothèque FFTW3. Cette procédure permet d'appliquer une FFT sur plusieurs variables à la fois, si celles-ci sont écrites en mémoires par intervalles constants sans être forcement contiguës. Ainsi, l'exécution se fait grâce à la routine `DFFTW_EXECUTE_DFT` mais la construction des plans utilise une procédure plus complexe : `DFFTW_PLAN_MANY_DFT`.

En plus des arguments identiques à `DFFTW_PLAN_DFT_R2C_1D` donnant la taille de la FFT à appliquer et le sens de celle-ci, les arguments tels que `istride` et `idist`, pour les données en entrée, et `ostride` et `odist`, pour celles en sortie, concernent la répartition des données en mémoire. L'argument `istride` donne l'écart mémoire entre deux éléments d'une FFT, tandis que `idist` indique l'écart entre les premières données de deux FFT. Pour illustrer cela on utilise le tenseur T de dimension $3 \times 3 \times 3$ dont l'arrangement des données est décrit par la figure (5).

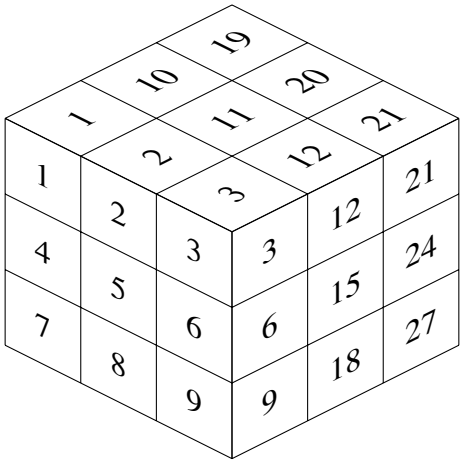

FIGURE 5 – Numérotation mémoire de chaque donnée de T

Supposons que les données soient alignées d'abord selon la direction x, puis y et enfin z.

Ainsi, pour produire un plan pour exécuter une FFT dans la direction x, `istride` vaut 1 soit la distance entre deux données et `idist` est égale à 3 pour la distance entre deux FFT. De même, pour produire une FFT dans la direction z on donne `istride`$= 9$ et `idist`$= 1$. Cependant pour construire une FFT dans la direction y une boucle est malgré tout nécessaire. En effet, dans cet exemple pour exécuter les neuf FFT dans cette direction, `istride`$= 3$

mais `idist` est variable. Celui-ci vaut 3 pour les trois premières FFT, puis doit valoir -5 pour appliquer la quatrième FFT. Ainsi, une boucle sur la direction x est nécessaire pour utiliser cette routine. Malgré cela, cette procédure reste plus performante que de réaliser chaque FFT individuellement avec la routine `DFFTW_EXECUTE_DFT_R2C`.

3.3 Perspectives

Lors de ce stage, un processus de vérification sur la capacité du code de calcul Tchebycube à mener des opérations sur des champs de données et à résoudre un problème de Helmholtz a été construit. Mais des mesures de performance et de scalabilité doivent être encore faites pour situer ce code, notamment, par rapport à DEDALUS [1]. Prochainement, l'ajout de nouvelles géométries telle qu'un domaine en coordonnées cylindriques et possiblement de géométries quelconques permettrait d'étendre les phénomènes reproductibles par Tchebycube. De plus, une mesure sur l'intérêt de calculer par des transformées de Fourier rapides à partir d'une certaine taille du problème est encore à qualifier quantitativement. Ainsi, l'état actuel du code de calcul permet de manière robuste de résoudre un problème de Helmholtz avec diverses conditions aux limites en parallèle. Celui-ci propose une grande ergonomie, permise par une programmation orientée objet, autorisant un utilisateur de manier Tchebycube sans rentrer dans le code source du solveur. De plus, la parallélisation est explicite grâce à la bibliothèque 2DecompFFT [9].

4 Les forces de Coriolis

De nombreux phénomènes physiques, particulièrement en astronomie, sont décrit comme des systèmes soumis aux effets de leur propre rotation. Les forces de Coriolis prennent en importance proportionnellement à la vitesse de cette rotation.

Dans ce chapitre ces forces seront intégrées au système d'équation de Navier-Stokes pour les fluides incompressibles. Cet ajout, historiquement traité de manière explicite, sera ensuite modifié, sous certaine hypothèse, pour être calculé de manière implicite. Ce processus semble permettre d'explorer des configurations physiques dont les effets de la rotation sont plus important. Cela sera vérifié sur un cas analytique pour le cas explicite et implicite.

4.1 Formulation du problème

Pour intégrer les forces de Coriolis, on part du système d'équations de Navier-Stokes pour les fluides incompressibles sur Ω :

$$\begin{cases} \dfrac{\partial V}{\partial t} + (V \cdot \nabla) V - \nu \nabla^2 V + \nabla q = f \\ \nabla \cdot V = 0 \end{cases} \tag{81}$$

On change de référentiel pour le référentiel non-inertiel en rotation avec le système étudié, ainsi, (81) se réécrit :

$$\begin{cases} \dfrac{\partial U}{\partial t} + (U \cdot \nabla) U - \nu \nabla^2 U + \nabla q + 2\underline{\omega} \times U + \underline{\omega} \times (\underline{\omega} \times r) = f \\ \nabla \cdot U = 0 \end{cases} \tag{82}$$

Avec $\underline{\omega}$ le vecteur de vitesse angulaire et r le vecteur depuis le centre de coordonnées. Les forces de Coriolis sont décrites par le terme $2\underline{\omega} \times U$ et les forces centrifuges le sont par $\underline{\omega} \times (\underline{\omega} \times r)$. Ainsi, en utilisant l'identité suivante :

$$\underline{\omega} \times (\underline{\omega} \times r) = -\nabla \frac{1}{2} (\underline{\omega} \times r)^2 \tag{83}$$

Et en posant $q = p - \dfrac{1}{2} (\underline{\omega} \times r)^2$, le système (82) se réécrit :

$$\begin{cases} \dfrac{\partial U}{\partial t} + (U \cdot \nabla)\, U - \nu \nabla^2 U + \nabla p + 2\underline{\omega} \times U = f \\ \nabla \cdot U = 0 \end{cases} \tag{84}$$

Le code de calcul a été construit de manière à simuler un système fluide en rotation autour de l'axe z, donc $\underline{\omega}$ a la forme suivante.

$$\underline{\omega} = \begin{pmatrix} 0 \\ 0 \\ \omega \end{pmatrix} \tag{85}$$

Ainsi, le terme de Coriolis se réécrit :

$$2\underline{\omega} \times U = \begin{pmatrix} -\omega U_y \\ \omega U_x \\ 0 \end{pmatrix} \tag{86}$$

Cela apporte une nouvelle difficulté, le couplage entre la composante U_x et U_y du champ de vitesse.

4.2 Terme de Coriolis explicite

Pour répondre à cette nouvelle contrainte, la voie la plus rapide consiste à expliciter ce terme et l'intégrer dans le calcul des termes non-linéaires. Ainsi, pour les trois précédents algorithmes de projection, on modifie le terme $NL(U^k)$ comme suit :

$$NL(U^k) = \left(U^k \cdot \nabla\right) U^k + 2\underline{\omega} \times U^k \tag{87}$$

Mais l'inconvénient de cette explicitation apparaît car $\omega \approx \dfrac{1}{E_k}$ où E_k est le nombre d'Ekman représentant le rapport entre les forces de viscosité et la force de Coriolis. Ainsi, lorsque que le nombre d'Ekman diminue ($i.e.$ la vitesse de rotation augmente) l'importance de ce terme augmente. Donc l'explicitation de celui-ci détériore grandement l'approximation pour un nombre d'Ekman très faible.

4.3 Terme de Coriolis implicite

Pour permettre d'explorer des nombres d'Ekman de plus en plus faibles, une autre solution consiste à impliciter le terme de Coriolis. Ainsi, pour les algorithmes de projection, l'étape d'estimation de vitesse provisoire, pour un schéma temporel du premier ordre, se réécrit telle que :

$$\frac{1}{\Delta t}\left(\tilde{U}^{k+1} - U^k\right) - \nu\nabla^2\tilde{U}^{k+1} + 2\underline{\omega} \times \tilde{U}^{k+1} = f(t^{k+1}) - NL(U^k) \tag{88}$$

En projetant (88) sur chacun des axes, on obtient le système suivant :

$$\begin{cases} \frac{1}{\Delta t}\left(\tilde{U}_x^{k+1} - U_x^k\right) - \nu\frac{\partial\tilde{U}_x^{k+1}}{\partial x^2} - 2\omega U_y^k = f_x(t^{k+1}) - NL_x(U^k) \\ \frac{1}{\Delta t}\left(\tilde{U}_y^{k+1} - U_y^k\right) - \nu\frac{\partial\tilde{U}_y^{k+1}}{\partial y^2} + 2\omega U_x^k = f_y(t^{k+1}) - NL_y(U^k) \\ \frac{1}{\Delta t}\left(\tilde{U}_z^{k+1} - U_z^k\right) - \nu\frac{\partial\tilde{U}_z^{k+1}}{\partial z^2} = f_z(t^{k+1}) - NL_z(U^k) \end{cases} \tag{89}$$

Mon tuteur S. Abide propose, pour tenir compte du couplage des composantes U_x et U_y, de poser une variable complexe $\tilde{\phi}^{k+1} = \tilde{U}_x^{k+1} + i\tilde{U}_y^{k+1}$. Par la suite, on nomme $(L1)$ le première ligne du système (89) et $(L2)$ la seconde ligne. Ainsi, en multipliant par le nombre complexe i $(L2)$ puis en y additionnant $(L1)$, on a, avec la notation :

$$f_x(t^{k+1}) + if_y(t^{k+1}) - \left(NL_x(U^k) + iNL_y(U^k)\right) = f_\phi(t^{k+1}) - NL_\phi(U^k) \tag{90}$$

$$\begin{aligned} f_\phi(t^{k+1}) - NL_\phi(U^k) &= \frac{1}{\Delta t}\left(\tilde{U}_x^{k+1} + i\tilde{U}_y^{k+1} - U_x^k - iU_y^k\right) - \nu\left(\frac{\partial\tilde{U}_x^{k+1}}{\partial x^2} + i\frac{\partial\tilde{U}_y^{k+1}}{\partial y^2}\right) - 2\omega U_y^k + 2i\omega U_x^k \\ &= \frac{1}{\Delta t}\left(\tilde{\phi}^{k+1} - \phi^k\right) - \nu\nabla^2\tilde{\phi}^{k+1} - 2i\omega\phi^k \end{aligned}$$

Ainsi, un solveur complexe permet de résoudre ce couplage et donc d'impliciter le terme de Coriolis, pour ce cas précis.

Mais il faut ensuite calculer le champ de pression, à partir du système suivant :

$$\begin{cases} \frac{1}{\Delta t}\left(U^{k+1}-\tilde{U}^{k+1}\right)+\underline{\omega}\times\left(U^{k+1}-\tilde{U}^{k+1}\right)+\nabla p^{k+1}=0 \\ \nabla\cdot U^{k+1}=0 \\ U^{k+1}\cdot n|_{\Gamma}=W^{k+1}\cdot n|_{\Gamma} \end{cases} \tag{91}$$

En réécrivant la première ligne de ce système, avec les notations de [10] pour la matrice 3×3 de l'opérateur $\mathbf{M}$, $\mathbf{M}=[I+\Delta t\underline{\omega}\times]$, cela donne :

$$\Delta t\nabla p^{k+1}=-\left(U^{k+1}+\Delta t\underline{\omega}\times U^{k+1}\right)+\tilde{U}^{k+1}+\Delta t\underline{\omega}\times\tilde{U}^{k+1} \tag{92}$$

$$\nabla p^{k+1}=\frac{1}{\Delta t}\mathbf{M}\left(\tilde{U}^{k+1}-U^{k+1}\right) \tag{93}$$

Or, d'après l'article d'Olshanskii [10], on a, en notant $\tilde{\omega}=\Delta t\underline{\omega}$:

$$\mathbf{M}^{-1}=\frac{1}{1+|\tilde{\omega}|^2}[I+\tilde{\omega}\otimes\tilde{\omega}-\tilde{\omega}\times] \tag{94}$$

Dont l'opération $\otimes$ représente le produit terme à terme des vecteurs, ainsi $(\tilde{\omega}\otimes\tilde{\omega})_{i,j}=\tilde{\omega}_i\tilde{\omega}_j$.

Vérifions cette identité, pour cela on note :

$$\tilde{\omega}=\begin{pmatrix}\omega_x\\ \omega_y\\ \omega_z\end{pmatrix} \tag{95}$$

Ainsi, pour $X=(x,y,z)^t$

$$\tilde{\omega}\times X=\begin{pmatrix}\tilde{\omega}_y z-\tilde{\omega}_z y\\ \tilde{\omega}_z x-\tilde{\omega}_x z\\ \tilde{\omega}_x y-\tilde{\omega}_y x\end{pmatrix} \tag{96}$$

Cela permet donc d'écrire, composante par composante, l'opérateur suivant :

$$[\tilde{\omega}\times]=\begin{pmatrix}0 & -\omega_z & \omega_y\\ \omega_z & 0 & -\omega_x\\ -\omega_y & \omega_x & 0\end{pmatrix} \tag{97}$$

$$\mathbf{M} = \begin{pmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{pmatrix} \tag{98}$$

Cela permet aussi d'expliciter $\mathbf{M}^{-1}$:

$$\mathbf{M}^{-1} = \frac{1}{1+|\tilde{\omega}|^2} \begin{pmatrix} 1+\omega_x^2 & \omega_x\omega_y + \omega_z & \omega_x\omega_z - \omega_y \\ \omega_x\omega_y - \omega_z & 1+\omega_y^2 & \omega_y\omega_z + \omega_x \\ \omega_x\omega_z + \omega_y & \omega_y\omega_z - \omega_x & 1+\omega_z^2 \end{pmatrix} \tag{99}$$

Ce qui permet de confirmer que $\mathbf{M}\mathbf{M}^{-1} = \mathbf{M}^{-1}\mathbf{M} = I$

Ainsi, en appliquant l'hypothèse d'incompressibilité sur (93), on obtient :

$$\mathbf{M}^{-1}\nabla p^{k+1} = \frac{1}{\Delta t}\left(\tilde{U}^{k+1} - U^{k+1}\right) \tag{100}$$

$$\nabla \cdot \mathbf{M}^{-1}\nabla p^{k+1} = \frac{1}{\Delta t}\nabla \cdot \tilde{U}^{k+1} \tag{101}$$

Or, en rappelant que $\underline{\omega}$ est un vecteur constant, donc $\underline{\omega} \times \nabla p^{k+1} = \nabla \times \left(p^{k+1}\underline{\omega}\right)$, d'où l'identité suivante :

$$\nabla \cdot \left(\underline{\omega} \times \nabla p^{k+1}\right) = 0 \tag{102}$$

Ainsi, (101) se réécrit avec $\mathcal{M} = \frac{1}{1+|\tilde{\omega}|^2}[I + \tilde{\omega} \otimes \tilde{\omega}]$

$$\nabla \cdot \mathcal{M}\nabla p^{k+1} = \frac{1}{\Delta t}\nabla \cdot \tilde{U}^{k+1} \tag{103}$$

Or, d'après les hypothèses précédentes, la rotation est seulement autour de l'axe z, ainsi :

$$\tilde{\omega} = \begin{pmatrix} 0 \\ 0 \\ \Delta t\omega \end{pmatrix} \tag{104}$$

Ce qui simplifie l'expression de $\mathcal{M}$ pour en faire une matrice diagonale.

$$\mathcal{M} = \begin{pmatrix} \dfrac{1}{1+|\tilde{\omega}|^2} & 0 & 0 \\ 0 & \dfrac{1}{1+|\tilde{\omega}|^2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{105}$$

Ainsi, (101) devient un problème de Helmholtz avec une pondération sur l'application de l'opérateur de divergence entre la différenciation des deux premières directions et la dernière.

5 Validation

Une des tâches les plus chronophages lors de la construction du code de calcul est la vérification de celui-ci. Pour cela, chaque propriétés doit être vérifiées et un maximum de symétries doit être brisées„ pour aller confronter la moindre ligne de Tchebycube. Pour réaliser cela, la précision spéctrale sera recherchée, permettant d'expliciter le processus de développement en détaillant une erreur dans le programme. Puis, une comparaison entre l'algorithme de projection originellement proposé par Chorin, l'algorithme de projection à pression incrémentale et l'algorithme de projection amélioré sur un cas analytique sera proposé. Ensuite, cette comparaison sera renouvelée en intégrant au problème les forces de Coriolis. Enfin, le cas académique de convection de Rayleigh-Bénard sera présenté sans effet des forces de Coriolis.

5.1 Précision spectrale

La propriété clef de la méthode numérique est la précision spectrale de la solution, celle-ci est donc parmi les premiers aspects de l'algorithme qui sont vérifiés. En effet, Gottlieb et Orszag [5], pour un intervalle connexe et borné I et pour $1 \leqslant p \leqslant \infty$, donnent l'estimation de convergence suivante :

$$||u - u_N||_{L^p(I)} \leqslant CN^{-m}||u^{(m)}||_{L^p(I)} \tag{106}$$

Pour u la solution exacte, u_N l'approximation et C une constante indépendante de N. Ainsi, pour une solution infiniment différenciable, l'erreur d'approximation est inférieur à toutes puissances de $\frac{1}{N}$, la convergence est donc exponentielle et la précision est dite spectrale. Pour vérifier cela, on pose comme solution exacte au système d'équation de Navier-Stokes :

$$U_{ex} = \begin{pmatrix} -2sin^2(\pi x)\, sin(2\pi y)\, sin(2\pi z)\, (1 + cos^2(4\pi t)) \\ sin(2\pi x)\, sin^2(\pi y)\, sin(2\pi z)\, (1 + cos^2(4\pi t)) \\ sin(2\pi x)\, sin(2\pi y)\, sin^2(\pi z)\, (1 + cos^2(4\pi t)) \end{pmatrix} \tag{107}$$

$$p_{ex} = \frac{1}{\pi} sin^2(\pi x) sin^2(\pi y) sin^2(\pi z) \tag{108}$$

Les erreurs sur le champ de pression et le champ de vitesse de la figure (6) sont mesurées sur le système d'équations de Navier-Stokes pour les fluides incompressibles, pour $P_r = 1$, $R_a = 10^8$, $\Delta t = 7.8125 \times 10^{-5}$ et $T_{final} = 0.5$. Un algorithme de projection à pression incrémentale est utilisé avec des conditions aux limites de Dirichlet pour le champ de vitesse et de Neumann pour le champ de pression.

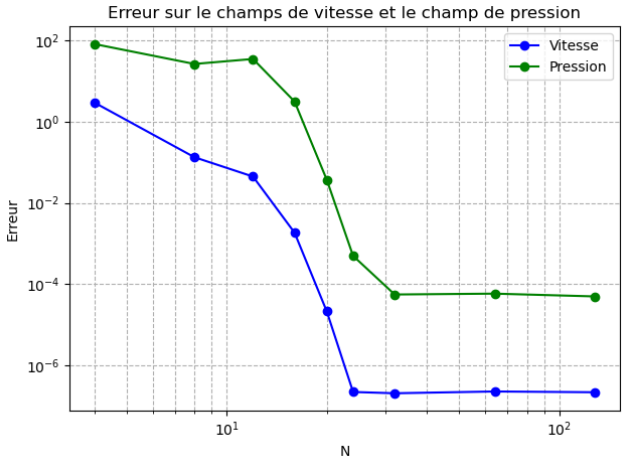

FIGURE 6 – Erreur sur le système de Navier-Stokes en fonction de la résolution N

Sur la figure (6), la décroissance exponentielle de l'erreur en fonction de N la finesse de la discrétisation est visible pour N allant de 4 à 24. Cependant, l'erreur stagne après cette finesse en raison du schéma temporel. Pour voir la convergence dite spectrale il faudrait aussi diminuer le pas de temps Δt, car la précision spectrale est valide qu'en espace et pas en temps.

5.2 Exemple de bug

Je souhaite présenter dans les détails une erreur trouvée lors de la vérification de la résolution du problème de Helmholtz : $\sigma u + \nu \nabla^2 u = S$, dans une configuration non-périodique dans les trois directions.

Pour commencer, on construit une solution U_{ex} sur le domaine $\Omega = [-1, 1]^3$. Ainsi, comme une approximation par série tronquée de polynômes de Tchebyshev est exacte sur les combinaisons linéaires de polynôme de Tchebyshev, on choisit comme forme de solution exacte :

$$U_{ex} = \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} \sum_{k=0}^{N_z} \alpha_{i,j,k} T_i(x) T_j(y) T_k(z) \tag{109}$$

Avec N_x, N_y et N_z la résolution dans chaque direction, $\alpha_{i,j,k}$ les coefficients des polynômes de Tchebyshev T_k de degré k.

Pour briser les symétries, les coefficients $\alpha_{i,j,k}$ sont tirés de manière aléatoire, de même que les résolutions N_x, N_y et N_z. Les six conditions aux limites du domaine Ω sont aussi randomisées entre une condition de type Dirichlet, Neumann ou Robin. Enfin, les paramètres du problème ν et σ sont aussi tirés aléatoirement.

Ainsi, on s'attend à ce que la solution de ce problème soit approchée à l'ordre du zéro machine, soit en utilisant la double précision, de l'ordre de 10^{-16}. Lors des premières vérifications, l'ordre de l'erreur commise atteint 10^{-6}. Cela a permis de trouver une des erreurs de programmation les plus subtiles de ce code de calcul.

En regardant l'erreur commise sur chaque point de collocation, on remarque que l'erreur commise atteint le zéro machine sur les points intérieurs de collocation et cela permet de situer le problème sur les bords du domaine.

En effet, comme expliquer dans la parallélisation de l'algorithme, les valeurs aux bords sont écrites sur les six "faces" du tenseur de calcul S. Mais cela pose un problème sur les douze arêtes et huit sommets du tenseur S. Sur la jonction de deux conditions limites un choix doit être fait entre les deux valeurs à imposer. Pour illustrer cela, on utilise une discrétisation de Ω en $3 \times 3 \times 3$:

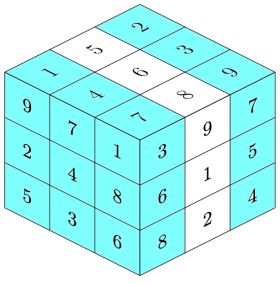

FIGURE 7 – Valeurs aux limites dans la direction x

Ainsi, cela crée un ordre de priorité sur le remplissage des valeurs aux bords. Mais cette hiérarchie est à prendre en compte lors du retour dans l'espace physique, c'est-à-dire lors de l'application par produits tensoriels des matrices de passages P_z, P_y et P_x respectivement

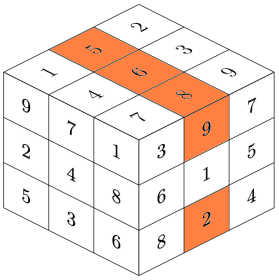

FIGURE 8 – Valeurs aux limites dans la direction y

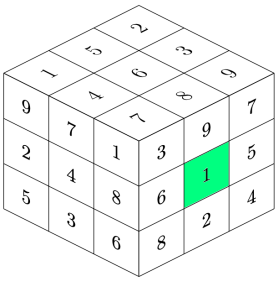

FIGURE 9 – Valeurs aux limites dans la direction z

dans les directions z, y et x. Pour résumer le processus, les points de collocation de la figure (7) contiennent les valeurs aux limites dans la direction x puis les points sur la figure (8) pour les valeurs dans la direction y. Avant de commencer les produits tensoriels pour passer dans l'espace propre les points de la figure (9) sont initiés avec les valeurs aux bords dans la direction z.

Puis, après avoir appliquer Λ pour reconstruire les valeurs aux bords et le produit avec P_z, on applique les relations (74) sur les points (9), puis, après application de la matrice de passage P_y, on utilise ce système sur les points de la figure (8) et enfin après le produit tensoriel avec P_x.

Ainsi, cette vérification a permis de mettre en lumière que l'ordre de remplissage des sommets et des arêtes est important, pour imposer les conditions aux limites avec les valeurs correspondantes.

5.3 Comparaison des algorithmes de projection

On construit une solution exacte au problème (16) sur $\Omega = [-1, 1]^3$ avec des conditions aux limites Dirichlet homogènes.

$$U_{ex} = \left(1 + cos^2(4\pi t)\right) \begin{pmatrix} -2sin^2\left(\pi x\right) sin\left(2\pi y\right) sin\left(2\pi z\right) \\ sin\left(2\pi x\right) sin^2\left(\pi y\right) sin\left(2\pi z\right) \\ sin\left(2\pi x\right) sin\left(2\pi y\right) sin^2\left(\pi z\right) \end{pmatrix} \tag{110}$$

$$p_{ex} = \frac{1}{\pi} sin^2(\pi x) sin^2(\pi y) sin^2(\pi z) \tag{111}$$

Comme le problème est posé sur le gradient du champ de pression, avec des conditions de Neumann, l'erreur est mesurée entre le gradient calculé et exact de ce champ car celui-ci est défini à une constante près.

Ainsi, on prend un maillage $32 \times 32 \times 32$ sur Ω pour comparer l'erreur commise sur le champ de vitesse et le gradient du champ de pression à $T_{final} = 0.5$ et $\nu = \dfrac{Pr}{Ra}$ avec $Pr = 1$ le nombre adimensionnel de Prandtl et $Ra = 10^8$ le nombre de Rayleigh.

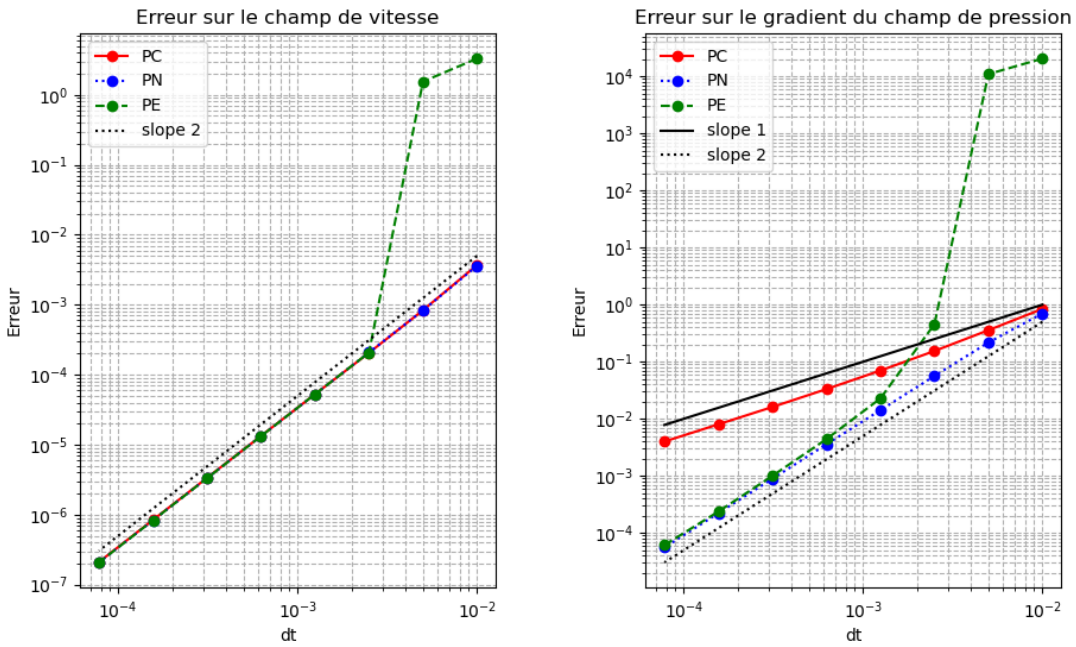

FIGURE 10 – Comparaison de divers algorithmes de projection

PC est l'algorithme de projection proposé par Chorin, quant à PN, il représente l'algorithme de projection à pression incrémentale. Enfin, PE est le diminutif de l'algorithme de projection amélioré.

Ainsi, d'après la figure ci-dessus (10), l'algorithme de projection amélioré semble nécessiter une condition sur le pas de temps Δt plus forte par rapport aux deux autres algorithmes. De plus, les trois algorithmes sont mesurés empiriquement comme étant d'ordre 2 en temps,

sans distinction entre ceux-ci pour l'approximation du champ de vitesse. Donc dans ce cas de figure le résultat donné par Guermond [7] n'est pas optimal sur ce champ. La différence la plus flagrante entre ces algorithmes se situe sur le gradient du champ de pression. En effet, l'algorithme de Chorin est empiriquement d'ordre 1 en temps, tandis que les algorithmes de projection à pression incrémentale et de projection amélioré sont empiriquement d'ordre 2 en temps. Au vu de cela, lors de l'ajout des forces de Coriolis dans le modèle de la section suivante, seulement les algorithmes de projection à pression incrémentale et amélioré seront utilisés car ils apportent, pour ce cas analytique, une meilleure approximation du gradient du champ de pression.

5.4 Comparaison des algorithmes de projection avec les forces de Coriolis

On reprend la solution exacte utilisée pour la comparaison entre l'algorithme de projection à pression incrémentale et l'algorithme de projection amélioré.

$$U_{ex} = \begin{pmatrix} -2sin^2(\pi x)\, sin(2\pi y)\, sin(2\pi z)\, (1 + cos^2(4\pi t)) \\ sin(2\pi x)\, sin^2(\pi y)\, sin(2\pi z)\, (1 + cos^2(4\pi t)) \\ sin(2\pi x)\, sin(2\pi y)\, sin^2(\pi z)\, (1 + cos^2(4\pi t)) \end{pmatrix} \tag{112}$$

$$p_{ex} = \frac{1}{\pi} sin^2(\pi x) sin^2(\pi y) sin^2(\pi z) \tag{113}$$

L'algorithme de projection à pression incrémentale est noté PN, pour le terme de Coriolis traité explicitement, et PNI, lorsque celui-ci est implicite. De même, l'algorithme de projection amélioré est noté PE et sa version implicitant le terme de Coriolis est abrégée en PEI. La première chose que l'on remarque est que la diminution du nombre d'Ekman induit une perte de stabilité de l'algorithme. Mais celle-ci est moins prononcée lorsque la force de Coriolis est implicite (un critère de stabilité sur Δt reste à déterminer).

De plus, à nombre d'Ekman équivalent, pour les deux algorithmes l'erreur commise est supérieure pour chacune des versions explicites des algorithmes de projections relativement à leur version implicite. Enfin, une étude plus approfondie est à mener mais un nombre d'Ekman plus faible (*i.e.* avec des effets de rotations plus prépondérants) l'algorithme de projection amélioré avec implicitation du terme de Coriolis est empiriquement d'ordre 3 en

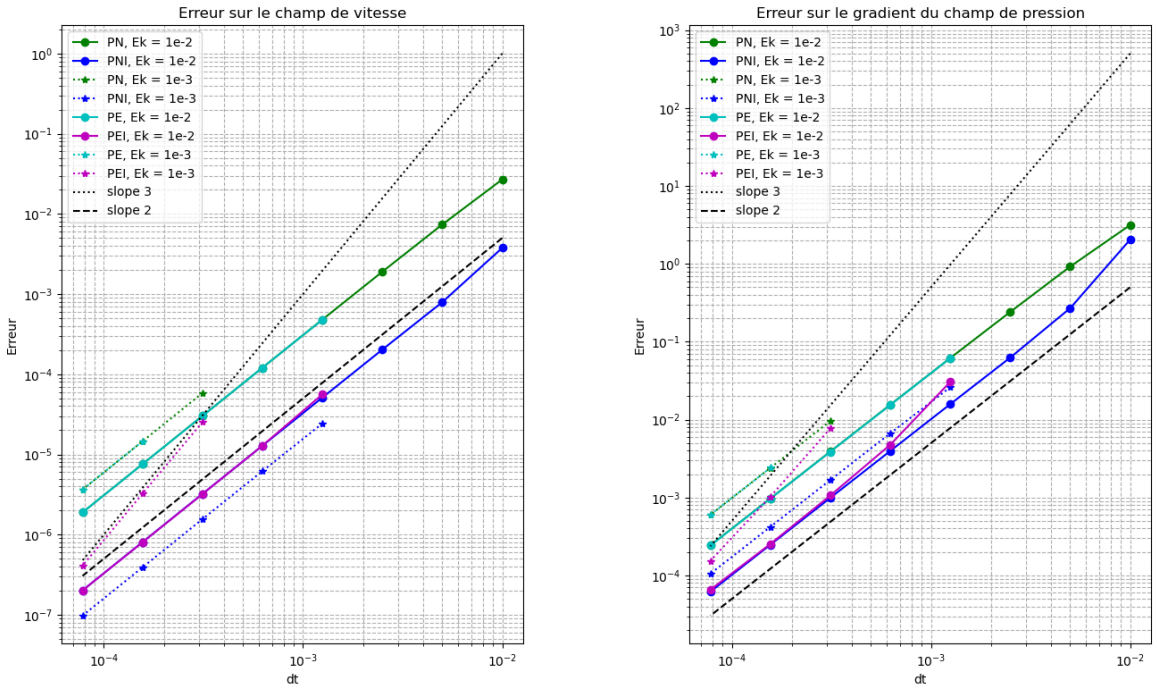

FIGURE 11 – Comparaison pour différents Ek, $T_{final} = 0.5, Pr = 1$ et $Ra = 10^8$

temps en comparaison avec les autres algorithmes qui montrent un ordre 2 en temps.

5.5 Exemple d'écoulement

La convection de Rayleigh-Bénard repose sur une instabilité entre un gradient de température sur le domaine Ω et deux conditions aux limites de Dirichlet non-homogènes. La figure (12) le résume, une mise en mouvement globale du fluide est attendue. De plus, pour un temps de calcul assez long, une phénomène de retournement peut être observable.

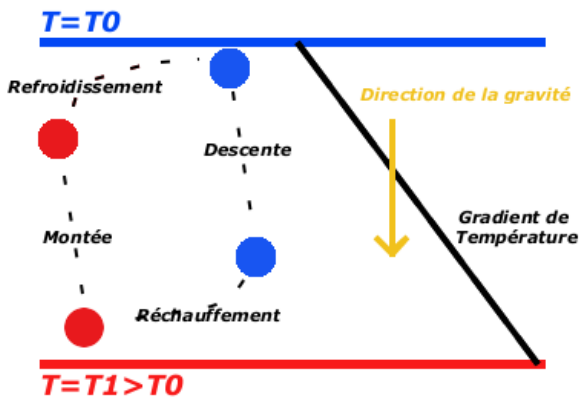

FIGURE 12 – Schéma de convection

Le système à résoudre est le système d'écoulement de fluides incompressibles de Navier-Stokes auquel une équation sur le champ de température T est ajoutée, dont le nombre adimensionnel est λ pour la conduction thermique.

$$\begin{cases} \dfrac{\partial U}{\partial t} + (U \cdot \nabla) U - \nu \nabla^2 U = -\dfrac{1}{\rho} \nabla p + g \\ \dfrac{\partial T}{\partial t} + (U \cdot \nabla) T - \lambda \nabla^2 T = 0 \\ \nabla \cdot U = 0 \end{cases} \tag{114}$$

La simulation se porte sur le domaine $\Omega = [-1, 1]^3$. Pour le champ de pression, les conditions aux limites sont toutes de type Neumann homogènes. Quant au champ de vitesse, on impose du Dirichlet homogène pour garantir la propriété d'étanchéité des surfaces. Enfin le champ de température est doté de conditions aux limites de Neumann homogène dans les deux premières directions et de condition de Dirichlet non-homogène dans la direction z.

Les paramètres de l'écoulement sont $\nu = \sqrt{\dfrac{P_r}{R_a}} = \sqrt{\dfrac{0.7}{10^7}}$ et $\lambda = \dfrac{1}{R_a} = \dfrac{1}{10^7}$. La discrétisation temporelle est donnée par $\Delta t = 10^{-2}$ pour un temps final à $T_{final} = 60$. Pour des raisons de ressources informatiques la simulation suivante est réalisée sur une résolution $64 \times 64 \times 64$ parallélisée sur 16 processeurs du calculateur AMU de Aix-Marseille, sur une période d'une heure.

Les figures suivantes sont une coupe normale à l'axe y située au centre du domaine Ω.

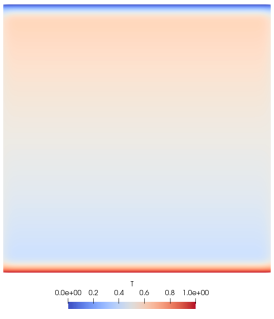

FIGURE 13 – Coupe de Ω à $t = 1$

La figure (13) illustre les conditions limites Dirichlet non-homogènes que l'on remarque sur le haut et le bas de la coupe. De plus, le gradient de température est encore visible 1 seconde après le temps initial.

Sur la figure (14) les premières instabilités apparaissent malgré que le gradient de température se maintienne globalement.

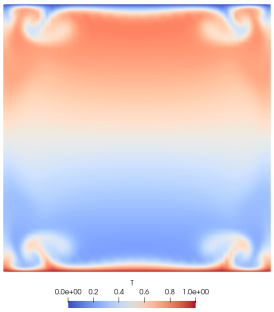

FIGURE 14 – Coupe de Ω à $t = 10$

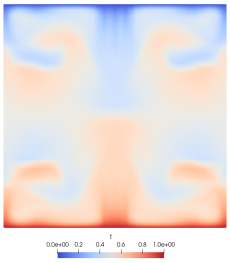

FIGURE 15 – Coupe de Ω à $t = 17.5$

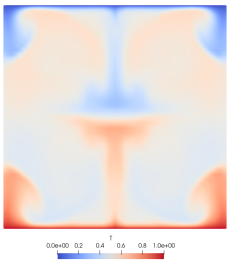

FIGURE 16 – Coupe de Ω à $t = 20$

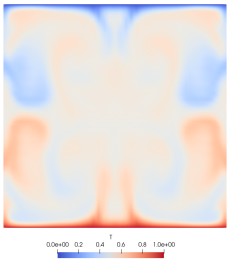

FIGURE 17 – Coupe de Ω à $t = 22.5$

Les figures (15), (16), (17) et (18) permettent d'observer plusieurs phénomènes de convection. Ce cas académique d'écoulement ne fait pas ressortir de soucis de programmation du code de calcul Tchebycube.

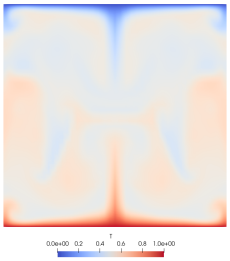

FIGURE 18 – Coupe de Ω à $t = 25$

Néanmoins, la simulation est sous résolue. Cela est visible sur les franges ascendantes dans les phénomènes de convection, particulièrement sur la figure (16). De ce fait, cette simulation est à but purement illustrative, une si faible résolution ne permet pas d'étude de cet écoulement. Pour aller chercher une simulation de production, des temps d'intégration bien plus long sont nécessaires et une résolution d'au minimum $512 \times 512 \times 512$ pour une configuration non-périodique dans chaque direction. Pour l'ajout d'une ou plusieurs directions périodiques une résolution plus grande dans ces directions doit être envisagée. En effet, la résolution de l'écoulement dans les directions périodiques se reposant sur l'usage des transformées de Fourier rapides, elle sera plus rapide que dans le cas non-périodique. Ainsi, la résolution de la simulation peut être augmentée, au moins dans ces directions.

6 Conclusion

Le code de calcul développé lors de ce stage utilise des méthodes numériques au moins aussi précises que les méthodes d'ordre élevé. Cependant, cela se fait au prix d'une dépendance entre chaque point de collocation du domaine. Celle-ci impose une attention particulière lors de la parallélisation de l'algorithme, pour des applications dans le calcul haute performance. Ces difficultés sont en partie surmontées grâce à plusieurs outils tels que les bibliothèques 2DecompFFT [9] et FFTW3 [4]. Malgré ces travaux, un faisceau de présomption pèse sur la scalabilité de cette implémentation souffrant de lourdes communications. Ce défaut est observable sur les mesures faites sur les algorithmes de FFT, qui reposent sur le même principe de parallélisation : le découpage en pencils.

De plus, le travail produit sur les effets des forces de Coriolis ouvre une voie à explorer. En effet, cette méthode implicitant les termes de ces forces pourrait enrichir l'étude de ces écoulements spécifiques, notamment à faible nombre d'Ekman. Mais cela est encore à l'étape de prospection et des études plus approfondies doivent être conduites pour réfuter ou renforcer ces premiers résultats.

Enfin, d'un point de vue plus personnel, ce stage m'a donné un aperçu du monde de la recherche, de ses enjeux et de ses moyens. En plus de l'enrichissement de mes connaissances sur la mécanique des fluides, la construction de Tchebycube m'a donné une vision d'ensemble sur le développement d'un code de calcul. Particulièrement les aspects matériel et logiciel ont été nouveaux pour moi. Ainsi, ces six mois ont approfondi ma compréhension dans le domaine informatique autant d'un point de vue software que hardware, dans le domaine physique avec une approche approfondie de la mécanique des fluides et dans le domaine des méthodes numérique avec les méthodes spéctrales.

Références

[1] Keaton J. BURNS et al. “Dedalus : A flexible framework for numerical simulations with spectral methods”. In : *Physical Review Research* 2.2, 023068 (avr. 2020), p. 023068. DOI : 10.1103/PhysRevResearch.2.023068. arXiv : 1905.10388 [astro-ph.IM].

[2] Alexandre Joel CHORIN. “The numerical solution of the Navier-Stokes equations for an incompressible fluid”. In : *Bulletin of the American Mathematical Society* 73.6 (1967), p. 928-931.

[3] CF CURTISS et JO HIRSCHFELDER. “Integration of Stiff Equations”. In : *Proceedings of the National Academy of Sciences of the United States of America* 38.3 (mars 1952), p. 235-243. ISSN : 0027-8424. DOI : 10.1073/pnas.38.3.235. URL : https://europepmc.org/articles/PMC1063538.

[4] M. FRIGO et S.G. JOHNSON. “The Design and Implementation of FFTW3”. In : *Proceedings of the IEEE* 93.2 (fév. 2005), p. 216-231. ISSN : 0018-9219. DOI : 10.1109/JPROC.2004.840301.

[5] David GOTTLIEB et Steven A ORSZAG. *Numerical analysis of spectral methods : theory and applications*. SIAM, 1977.

[6] J.L. GUERMOND, P. MINEV et Jie SHEN. “An overview of projection methods for incompressible flows”. In : *Computer Methods in Applied Mechanics and Engineering* 195.44 (2006), p. 6011-6045. ISSN : 0045-7825. DOI : https://doi.org/10.1016/j.cma.2005.10.010. URL : https://www.sciencedirect.com/science/article/pii/S0045782505004640.

[7] Jean-Luc GUERMOND. “Un résultat de convergence d’ordre deux en temps pour l’approximation des équations de Navier-Stokes par une technique de projection incrémentale”. fre. In : *ESAIM : Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique* 33.1 (1999), p. 169-189. URL : http://eudml.org/doc/193909.

[8] Sandrine HUGUES et Anthony RANDRIAMAMPIANINA. “An improved projection scheme applied to pseudospectral methods for the incompressible Navier-Stokes equations”.

In : *International Journal for Numerical Methods in Fluids* 28 (1998), p. 501-521. URL : https://api.semanticscholar.org/CorpusID:122964813.

[9] Ning. LI et Sylvain LAIZET. “2DECOMP&FFT - A Highly Scalable 2D Decomposition Library and FFT Interface”. In : 2010. URL : https://api.semanticscholar.org/CorpusID:62453043.

[10] Maxim A. OLSHANSKII, Andriy SOKOLOV et Stefan TUREK. “Error Analysis of a Projection Method for the Navier–Stokes Equations With Coriolis Force”. In : *Journal of Mathematical Fluid Mechanics* 12 (2010), p. 485-502. URL : https://api.semanticscholar.org/CorpusID:5931832.

[11] Roger PEYRET. *Spectral methods for incompressible viscous flow*. T. 148. Springer, 2002.

[12] Rolf RANNACHER. “On Chorin’s projection method for the incompressible Navier-Stokes equations”. In : *The Navier-Stokes Equations II—Theory and Numerical Methods : Proceedings of a Conference held in Oberwolfach, Germany, August 18–24, 1991*. Springer. 2006, p. 167-183.