Préconditionnement de systèmes linéaires symétriques définis positifs

Application à la simulation numérique d'écoulements océaniques tridimensionnels

Thèse

présentée à la faculté des sciences
pour obtenir le grade de docteur ès sciences par

Julien Straubhaar

soutenue avec succès le 19 avril 2007
et acceptée sur proposition du jury

Prof. Olivier Besson	directeur de thèse
Prof. Eric Blayo	rapporteur (Grenoble)
Prof. Martin J. Gander	rapporteur (Genève)
Prof. Michel Benaïm	examinateur (Neuchâtel)

Institut de mathématiques, Université de Neuchâtel,
Rue Emile–Argand 11, CH–2009 Neuchâtel

Université de Neuchâtel

FACULTE DES SCIENCES
Secrétariat-Décanat de la faculté
- Rue Emile-Argand 11
- CP 158
- CH-2009 Neuchâtel

IMPRIMATUR POUR LA THESE

Préconditionnement de systèmes linéaires symétriques définis positifs. Application à la simulation numérique d'écoulements océaniques tridimensionnels

Julien STRAUBHAAR

UNIVERSITE DE NEUCHATEL

FACULTE DES SCIENCES

La Faculté des sciences de l'Université de Neuchâtel,
sur le rapport des membres du jury

MM. O. Besson (directeur de thèse),
M. Benaïm, M. Gander (Genève)
et E. Blayo (Grenoble)

autorise l'impression de la présente thèse.

Neuchâtel, le 8 mai 2007

Le doyen :
T. Ward

UNIVERSITE DE NEUCHATEL
FACULTE DES SCIENCES
Secrétariat-décanat de la faculté
Rue Emile-Argand 11 - CP 158
CH-2009 Neuchâtel

■ Téléphone : +41 32 718 21 00 ■ Fax : +41 32 718 21 03 ■ E-mail : secretariat.sciences@unine.ch ■ www.unine.ch/sciences

à Léonie

Remerciements

Je tiens à remercier M. Olivier Besson pour tout ce qu'il m'a apporté. Il a su susciter mon intérêt pour les mathématiques appliquées. Fort de son expérience scientifique tant au niveau théorique que pratique, M. Olivier Besson a su me suggérer des problèmes intéressants et pertinents qui sont à l'origine de ce document et me guider tout au long de ce travail. Son enthousiasme a été la base d'un climat de travail serein et propice à la réalisation de cette thèse. Il m'a fait confiance en me laissant travailler de manière autonome tout en étant toujours disponible lorsque je rencontrais des difficultés. Ses précieux conseils m'ont permis de débloquer des situations délicates. M. Olivier Besson est également parvenu, grâce à son optimisme, à me redonner de la motivation lorsque celle–ci baissait.

Je remercie les membres du jury, MM. Eric Blayo, Martin J. Gander et Michel Benaïm du temps consacré à la lecture de ce document et de l'intérêt porté à ce travail. Merci aux rapporteurs pour leurs commentaires enrichissants.

Le centre suisse de calcul scientifique (CSCS) a mis à ma disposition les ressources informatiques nécessaires à la mise en œuvre parallèle des logiciels développés lors de ce travail. Je remercie ses collaborateurs d'avoir bien voulu répondre à mes questions techniques.

Je souhaite aussi remercier tous mes collègues de l'institut de mathématiques pour l'ambiance chaleureuse qu'ils y font régner. Je remercie notamment Souleye Kane, le doctorant précédent de M. Besson, pour les discussions partagées autour de ce travail et Benjamin Bergé pour ses conseils et nos pauses ludiques.

Je remercie David Ardia pour ses conseils en LaTeX qui m'ont permis d'améliorer la mise en page de cette thèse.

Enfin, mes remerciements s'adressent à ma famille et mes amis qui m'ont apporté leur soutien. Merci à ma femme qui m'a patiemment écouté tout au long de ce travail.

Mai 2007,
Julien Straubhaar

Table des matières

Introduction

Mots clés. Méthode du gradient conjugué ; préconditionneurs ; A–orthogonalisation ; moindres carrés ; parallélisation ; équations de Navier–Stokes ; méthodes de prédicteur–correcteur ; éléments finis ; écoulements océaniques tridimensionnels ; simulations numériques ; océan Atlantique nord.

Keywords. Conjugate gradient method ; preconditioners ; A–orthogonalization ; least squares ; parallelization ; Navier–Stokes equations ; predictor–corrector methods ; finite elements ; three dimensional ocean circulations ; numerical simulations ; north Atlantic ocean.

L'eau recouvre la majorité de notre planète... La compréhension et la connaissance de la Terre commencent par celles de l'eau !

La circulation de l'eau joue un rôle important dans de nombreux phénomènes physiques et biologiques. Outre l'activité tectonique, l'érosion par l'eau façonne le relief des fonds lacustres, marins et océaniques. Leur structure est influencée par le transport et l'accumulation de sédiments. Les courants ont un impact sur la biodiversité aquatique, sur la végétation des fonds mais aussi sur certaines espèces animales migrant au fil des saisons selon la température de l'eau. Si le déplacement des masses d'eau participe au développement de phénomènes météorologiques locaux, il est, à l'échelle océanique, le principal facteur déterminant le climat. À plus petite échelle, la connaissance de l'écoulement permet d'anticiper la propagation de polluants (marée noire par exemple) et ainsi de minimiser les dégâts causés à l'environnement par ceux-ci.

Le réchauffement climatique dû à l'activité de l'Homme est certainement un acteur principal de l'avenir de notre planète. La connaissance des écoulements océaniques est ainsi primordiale pour prévoir les impacts et les conséquences sur les différentes régions du globe.

La circulation de l'eau dans les océans est notamment influencée par les différences de densité de l'eau. La salinité et la température de l'eau engendrent des

variations de densité qui induisent la circulation océanique thermohaline (lente et à large échelle)[16, 55]. Les modèles de circulation thermohaline négligent l'influence des vents.

Plusieurs méthodologies peuvent être considérées pour obtenir des simulations d'écoulements.
Des modèles numériques basés sur l'assimilation de données consistent à reconstituer un écoulement en respectant au mieux les mesures existantes (salinité, température), en surface et *in situ*. Le projet de recherche français *Mercator Océan* (http ://www.mercator-ocean.fr) et le projet européen *AWI* (http ://www.awi.de/en/home) développent notamment, avec cette approche, un système d'océanographie opérationnelle permettant d'analyser et de prédire les océans (courants, température, salinité, etc.).
Des modèles pour des domaines peu profonds (*shallow water*) consistent à intégrer verticalement (selon la profondeur) les équations régissant l'écoulement, afin de calculer les vitesses moyennes horizontales des courants (modèle bidimensionnel)[40, 41, 59].

Nous proposons dans ce travail de calculer des écoulements océaniques tridimensionnels globaux, induits par les vents en surface et en considérant l'eau à densité constante et uniquement des contraintes physiques sur les bords du domaine.

Les équations de Navier–Stokes décrivent le mouvement des fluides [23, 37, 47, 35, 20]. Nous considérons ces équations pour des bassins peu profonds comme les lacs, les mers, les océans [15, 5, 40]. Ces domaines sont caractérisés par le fait que le rapport de la profondeur sur l'étendue horizontale est très petit (inférieur à 1%). Des calculs de courants en trois dimensions pour le lac de Neuchâtel et le lac Léman sont présentés dans [13] dans le cas où les tractions en surface dues aux vents constituent le moteur du système. Le but de ce travail est de simuler, dans ce contexte, des écoulements tridimensionnels pour de "grands" bassins (échelle océanique) et ainsi apporter des résultats originaux. Nous présentons le cas de l'océan Atlantique nord.

La discrétisation des équations de Navier–Stokes proposée donne des systèmes linéaires symétriques définis positifs de très grande taille, que nous résolvons avec l'algorithme du gradient conjugué (voir par exemple [3, 19, 46]). Pour cela, des techniques de préconditionnement et des logiciels parallèles sont nécessaires.

Cette thèse est constituée de deux parties. Dans la première, nous présentons une nouvelle classe de préconditionneurs pour l'algorithme du gradient conjugué. Des résultats théoriques sont donnés, des tests numériques sont effectués et la mise en œuvre sur des machines parallèles est étudiée. La seconde partie est consacrée à la simulation d'écoulements océaniques tridimensionnels. Nous y présentons en détails les techniques utilisées pour la résolution des équations de Navier–Stokes non stationnaires. Un écoulement en trois dimensions dans l'océan Atlantique nord est obtenu en utilisant un logiciel parallèle et les méthodes de préconditionnement développées dans la première partie. Des cartes de courants sont présentées.

Les logiciels parallèles nécessaires à ce travail ont été développés sur les machines parallèles CRAY XT3 du CSCS (Swiss National Supercomputing Center), voir le site web

http ://www.cscs.ch.

Introduction à la première partie

Le préconditionnement de systèmes linéaires symétriques définis positifs (SDP) est le sujet de la première partie.

Le but est de résoudre le système linéaire

$$Ax = b$$

où A est une matrice SDP, creuse (*i.e.* ayant une faible proportion de coefficients non nuls) et de grande taille. Pour cela, nous utilisons l'algorithme du gradient conjugué. C'est une méthode itérative dont la vitesse de convergence est contrôlée par la condition $\kappa = \lambda_{max}/\lambda_{min}$ de la matrice A, λ_{min} et λ_{max} étant respectivement la plus petite et la plus grande valeur propre de A. La m–ème approximation x_m donnée par l'algorithme vérifie [46]

$$\|x_m - \hat{x}\|_A \leqslant 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m \|x_0 - \hat{x}\|_A ,$$

où $\hat{x} = A^{-1}b$ est la solution du système et $\|y\|_A = (y^t A y)^{-1/2}$.
Préconditionner le système $Ax = b$ consiste à le remplacer par le système SDP équivalent

$$TAT^t\tilde{x} = Tb \ , \quad x = T^t\tilde{x}.$$

La matrice T (préconditionneur) est choisie de sorte que la condition de $\widetilde{A} = TAT^t$ soit proche de 1. Afin de pouvoir mettre en œuvre l'algorithme, le préconditionneur doit aussi être creux pour des raisons de place en mémoire. Des techniques de préconditionnement sont proposées dans [46, 8, 38, 36, 10, 27].

Dans ce document, nous proposons une nouvelle classe de préconditionneurs aisément parallélisables, voir aussi [54]. Elle est basée sur le procédé d'A–orthogonalisation de Gram–Schmidt utilisé par [10, 9, 11], des approximations au sens des moindres carrés et une méthode optimale de remplissage extraite de [27]. Des paramètres permettent de choisir la précision souhaitée et de contrôler le remplissage. Un résultat sur la condition du système préconditionné est donné (théorème 3.1 et corollaire 3.2). Une version couplée et une version par blocs sont considérées.

Nous présentons des tests numériques et des comparaisons avec d'autres préconditionneurs (chapitre 4).

Finalement, la parallélisation des algorithmes pour les préconditionneurs développés est étudiée (chapitre 5), voir aussi [53]. La performance est évaluée en termes de *speed–up* et d'*efficience* [25].

Introduction à la deuxième partie

La simulation d'écoulements océaniques tridimensionnels fait l'objet de la deuxième partie. Nous y appliquons les méthodes de préconditionnement développées dans la première partie.

Partant des équations de Navier–Stokes, une démarche permettant d'obtenir une simulation numérique est présentée de manière détaillée.

Au chapitre 6, nous énonçons les équations de Navier–Stokes non stationnaires (voir [23, 37, 47, 35, 20]) pour un fluide incompressible anisotrope à densité constante dans un bassin peu profond. Le modèle prend en compte la force de Coriolis (due à la rotation de la Terre), l'attraction de la Terre et les tractions des vents en surface, qui constituent le moteur de l'écoulement. Un vecteur de viscosité turbulente intègre l'influence du rapport d'aspect du bassin (*i.e.* $\varepsilon = h_0/d$, où h_0 est la profondeur maximale et d le diamètre horizontal) de sorte à vérifier asymptotiquement l'approximation hydrostatique [15, 5, 40]. Ceci donne le caractère anisotrope du fluide.

La discrétisation temporelle est traitée au chapitre 7. La formule de différentiation rétrograde d'ordre 2 ou la formule d'Euler implicite est utilisée pour approcher la dérivée en temps. Deux méthodes de projection sont proposées pour dissocier le calcul de la vitesse et de la pression, voir [32, 43, 56]. Chaque pas de temps est décomposé en deux sous–pas, le premier consiste en une étape de prédiction et le second en une étape de correction. Ce sont des méthodes à pas fractionnaires qui sont aussi appelées méthodes de *prédicteur–correcteur*.

Le principe de la première méthode présentée (prédicteur–correcteur (I)) est le suivant. Une prédiction de la vitesse vérifiant les conditions de bord est calculée sans prendre en compte la condition d'incompressibilité. Cette prédiction est ensuite projetée sur un espace de fonctions à divergence nulle (caractérisation de l'incompressibilité), ce qui se réalise en apportant une correction de pression à la vitesse prédite [29].

La seconde méthode présentée (prédicteur–correcteur (II)) tient compte de l'incompressibilité à l'étape de prédiction et des conditions de bord à l'étape de projection [33].

Des analyses de ces méthodes sont présentées dans [32, 29, 33, 28, 30, 31, 48, 57].

Ainsi, avec la méthode (I), nous obtenons des vitesses respectant l'incompressibilité du fluide mais ne satisfaisant que partiellement les conditions de bord, alors que la méthode (II) fournit des vitesses vérifiant les conditions de bord mais ne garantissant pas l'incompressibilité du fluide. Nous privilégierons la méthode (II) qui donnent expérimentalement des vitesses plus régulières que la méthode (I).

Nous donnons encore dans le chapitre 7 des formulations faibles des problèmes obtenus.

Au chapitre 8, nous proposons une discrétisation spatiale avec des éléments finis [21] de type Q_2 pour le calcul des vitesses horizontales (*i.e.* les deux premières composantes du vecteur vitesse) et de type Q_1 pour le calcul de la vitesse verticale (*i.e.* la troisième composante du vecteur vitesse) et la pression. La résolution numérique des équations de (Navier–)Stokes par des éléments finis est largement traitée dans la littérature, par exemple [24, 42, 12, 51].

Nous obtenons alors des problèmes faibles approchés, systèmes linéaires SDP creux. La construction des matrices et des seconds membres correspondants est donnée en détails. La matrice pour le calcul de la pression étant mal conditionnée, une méthode de pénalisation est considérée (chapitres 7 et 8).

Dans le dernier chapitre, le cas de l'océan Atlantique nord est traité, grâce à des données (bathymétrie, vents) fournies par le projet de recherche *Mercator Océan* (http ://www.mercator-ocean.fr). Un logiciel parallèle et les méthodes de préconditionnement de la première partie nous permettent d'obtenir des résultats que nous présentons sur plusieurs figures réalisées avec le logiciel *AVS/Express* (http ://www.avs.com).

Première partie

Préconditionnement de systèmes linéaires symétriques définis positifs

CHAPITRE 1

Méthode du gradient conjugué préconditionné

Considérons le système linéaire

$$Ax = b \tag{1.1}$$

où $A = (a_{ij})$ est une matrice réelle symétrique définie positive (SDP) de taille $n \times n$ et b un vecteur de $\mathbb{R}^n$. Dans ce chapitre, l'algorithme du gradient conjugué est présenté : c'est la méthode numérique qui va être utilisée pour résoudre ce type de système. De plus, le préconditionnement d'un tel système est abordé de manière générale.

1.1 Méthode du gradient conjugué (GC)

La méthode du gradient conjugué due, à Hestenes et Stiefel (1952), est une méthode itérative permettant de résoudre (1.1) : des *solutions approchées* $x_k \in \mathbb{R}^n$ et leur *résidu* $r_k = b - Ax_k$ sont calculés successivement ($k = 0, 1, ...$). Le résidu permet de contrôler l'erreur commise et donne un critère d'arrêt : lorsque sa norme est suffisamment petite, la solution approchée correspondante est retenue.

L'algorithme de cette méthode est donné par l'algorithme 1, où $(\,.\,|\,.\,)$ désigne le produit scalaire usuel de $\mathbb{R}^n$ et *tol* une tolérance fixée par l'utilisateur.

Présentons brièvement le principe de cette méthode (pour les détails voir l'annexe A et [3, 19, 46]). Une matrice SDP B fournit le produit scalaire $(\,x\,|\,y\,)_B := (\,x\,|\,By\,)$ et la norme associée $\|.\|_B$. La solution $\hat{x} = A^{-1}b$ de (1.1) est le minimum de l'application

$$f(x) = \|x - \hat{x}\|_A^2 = \|b - Ax\|_{A^{-1}}^2 = (\,x\,|\,Ax\,) - 2\,(\,b\,|\,x\,) + (\,b\,|\,\hat{x}\,)\,.$$

Dans l'algorithme GC, x_{k+1} est obtenu en cherchant depuis x_k et dans la direction d_k le minimum de cette application.

Comme $\nabla f(x) = 2(Ax-b)$, le résidu r_k indique la direction de plus grande pente de f au point x_k ; nous choisissons $d_0 = r_0$, mais, pour $k \geqslant 0$, ce n'est pas la direction r_{k+1} qui est retenue : nous déterminons $\beta_k \in \mathbb{R}$ de sorte que la *direction de descente* $d_{k+1} := r_{k+1} + \beta_k d_k$ soit A–orthogonale à d_k (*i.e.* $(\, d_{k+1} \,|\, d_k \,)_A = 0$). Pour comprendre ce choix, étudions un peu l'application f.

Algorithme GC

Soit $x_0 \in \mathbb{R}^n$ donné
Calculer $r_0 = b - Ax_0$ et poser $d_0 = r_0$
Si $\|r_0\| \,/\, \|b\| < tol$: STOP
Pour $k \geqslant 0$, faire :
  $\alpha_k = (\, r_k \,|\, r_k \,) \,/\, (\, d_k \,|\, Ad_k \,)$
  $x_{k+1} = x_k + \alpha_k d_k$
  $r_{k+1} = r_k - \alpha_k A d_k$
  Si $\|r_{k+1}\| \,/\, \|b\| < tol$: STOP
  $\beta_k = (\, r_{k+1} \,|\, r_{k+1} \,) \,/\, (\, r_k \,|\, r_k \,)$
  $d_{k+1} = r_{k+1} + \beta_k d_k$
Fin pour k

Algorithme 1.

Pour $c \geqslant 0$, notons

$$S_c = \{x \in \mathbb{R}^n \mid f(x) = c\}.$$

Considérons une base orthonormée $\mathcal{B} = \{\xi_1, \ldots, \xi_n\}$ de $\mathbb{R}^n$ formée de vecteurs propres de A, avec

$$A\xi_i = \lambda_i \xi_i, \;\; i = 1, \ldots, n,$$

où $0 < \lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n$ sont les valeurs propres de A.
En écrivant $x = \sum_{i=1}^n x_i \xi_i$, $b = \sum_{i=1}^n b_i \xi_i$ et $\hat{x} = \sum_{i=1}^n \hat{x}_i \xi_i$ dans cette base, nous avons $\hat{x}_i = b_i \lambda_i^{-1}$, $i = 1, \ldots, n$ et nous obtenons

$$S_c = \left\{ x = \sum_{i=1}^n x_i \xi_i \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^n \lambda_i (x_i - \hat{x}_i)^2 = c \right\}.$$

La surface de niveau S_c de f est donc un ellipsoïde centré en $\hat{x}$, de demi-axe $c^{1/2} \lambda_i^{-1/2}$ dans la direction ξ_i pour $i = 1, \ldots, n$.
La correction apportée à r_{k+1} pour obtenir la direction d_{k+1} "compense l'écrasement" de ces ellipsoïdes (marqué lorsque $\lambda_1 \ll \lambda_n$).
Le cas $n = 2$, $A = \mathrm{diag}(1, 9)$, $b = 0$ et $x_0 \in S_1$ est représenté sur la figure 1.

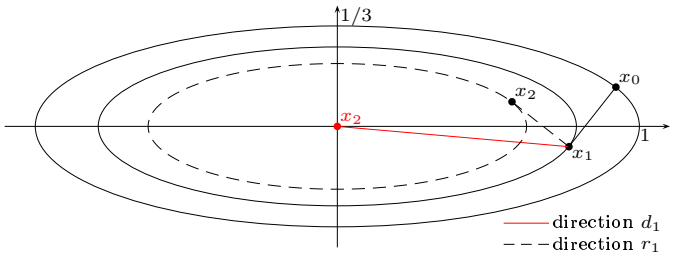

Figure 1.

1.1.1 Propriétés de la méthode du gradient conjugué

Les vecteurs d_k sont A–orthogonaux deux à deux (ils sont de directions conjuguées), *i.e.* $(\, d_i \,|\, d_j \,)_A = 0$ si $i \neq j$. Les résidus r_k sont deux à deux orthogonaux (relativement au produit scalaire usuel). Cette dernière propriété implique que l'algorithme converge en au plus n itérations (le résidu doit alors être nul) ! Cependant, ceci est théorique, en effet, comme nous le verrons plus loin, en pratique, il est possible que l'algorithme ne converge pas après n itérations (à cause des erreurs d'arrondi dues à la précision fixe du codage des nombres réels dans un ordinateur). De plus, même si l'algorithme converge, lorsque n est grand, la durée des calculs peut être élevée.

Le théorème suivant montre que la vitesse de convergence de l'algorithme GC est contrôlée par la condition de la matrice du système.

Théorème 1.1 *[46, p. 194] Si $\hat{x} = A^{-1}b$ est la solution de (1.1), alors les solutions approchées x_m, $m \geqslant 0$, de la méthode du gradient conjugué vérifient*

$$\|x_m - \hat{x}\|_A \leqslant 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^m \|x_0 - \hat{x}\|_A$$

*où $\|x\|_A = (\, x \,|\, Ax \,)^{1/2}$ est la $A-$norme du vecteur $x \in \mathbb{R}^n$ et $\kappa = \lambda_{max}/\lambda_{min}$ est la condition de la matrice A (*i.e. *le rapport entre sa plus grande et sa plus petite valeur propre).*

Par conséquent, plus la condition κ de la matrice A est proche de 1, plus le nombre d'itérations est petit.

Les preuves de ces propriétés et du théorème 1.1 figurent dans l'annexe A.

1.2 Méthode du gradient conjugué préconditionné

Pour améliorer la vitesse de convergence de la méthode du gradient conjugué, le théorème 1.1 suggère l'idée suivante : remplacer le système (1.1) en un système équivalent dont la matrice reste SDP et a une condition petite. Une telle transformation est appelée *préconditionnement* de (1.1). Pratiquement, nous cherchons à ce que la matrice du nouveau système soit "proche" de la matrice identité. Tout d'abord, présentons l'algorithme du gradient conjugué préconditionné.

1.2.1 Premier préconditionnement

Soit T une matrice réelle régulière de taille $n \times n$. Le système

$$TAT^t \tilde{x} = Tb, \;\; x = T^t \tilde{x} \tag{1.2}$$

est équivalent au système (1.1). La matrice T est appelée *préconditionneur (de type 1)* du système (1.1).
Notons $\widetilde{A} = TAT^t$ et $\tilde{b} = Tb$. Comme T est régulière et que A est SDP, la matrice $\widetilde{A}$ du système (1.2) ($\widetilde{A}\tilde{x} = \tilde{b}$) est encore SDP et la méthode du gradient conjugué peut être utilisée pour résoudre ce nouveau système. Notons $\tilde{x}_k$ les solutions approchées, $\tilde{r}_k = \tilde{b} - \widetilde{A}\tilde{x}_k$ les résidus et $\tilde{d}_k$ les directions de descente

obtenus en appliquant l'algorithme GC au système $\widetilde{A}\tilde{x} = \tilde{x}$ et posons $x_k = T^t\tilde{x}_k$, $r_k = b - Ax_k$, $d_k = T^t\tilde{d}_k$ et $s_k = T^t \cdot Tr_k$. Nous avons alors

$$\begin{aligned}
\tilde{r}_k &= Tb - TAT^t\tilde{x}_k = T(b - Ax_k) = Tr_k, \\
d_0 &= T^t\tilde{d}_0 = T^t\tilde{r}_0 = T^t \cdot Tr_0 = s_0, \\
(\,\tilde{r}_k \,|\, \tilde{r}_k\,) &= (\,Tr_k \,|\, Tr_k\,) = (\,r_k \,|\, s_k\,), \\
\left(\tilde{d}_k \,|\, \widetilde{A}\tilde{d}_k\right) &= \left(\tilde{d}_k \,|\, TAT^t\tilde{d}_k\right) = (\,d_k \,|\, Ad_k\,), \\
\tilde{\alpha}_k &= (\,\tilde{r}_k \,|\, \tilde{r}_k\,) \,/\, \left(\tilde{d}_k \,|\, \widetilde{A}\tilde{d}_k\right) = (\,r_k \,|\, s_k\,) \,/\, (\,d_k \,|\, Ad_k\,), \\
x_{k+1} &= T^t\tilde{x}_{k+1} = T^t(\tilde{x}_k + \tilde{\alpha}_k\tilde{d}_k) = x_k + \tilde{\alpha}_k d_k, \\
r_{k+1} &= T^{-1}\tilde{r}_{k+1} = T^{-1}(\tilde{r}_k - \tilde{\alpha}_k\widetilde{A}\tilde{d}_k) = r_k - \tilde{\alpha}_k Ad_k, \\
\tilde{\beta}_k &= (\,\tilde{r}_{k+1} \,|\, \tilde{r}_{k+1}\,) \,/\, (\,\tilde{r}_k \,|\, \tilde{r}_k\,) = (\,r_{k+1} \,|\, s_{k+1}\,) \,/\, (\,r_k \,|\, s_k\,), \\
d_{k+1} &= T^t\tilde{d}_{k+1} = T^t(\tilde{r}_{k+1} + \tilde{\beta}_k\tilde{d}_k) = s_{k+1} + \tilde{\beta}_k d_k.
\end{aligned}$$

Nous obtenons alors l'algorithme du gradient conjugué donné par l'algorithme 2.

Algorithme GCP$_1$: Avec T régulière

Soit $x_0 \in \mathbb{R}^n$ donné
Calculer $r_0 = b - Ax_0$, $s_0 = T^t \cdot Tr_0$ et poser $d_0 = s_0$
Si $\|r_0\| \,/\, \|b\| < tol$: STOP
Pour $k \geqslant 0$, faire :
  $\tilde{\alpha}_k = (\,r_k \,|\, s_k\,) \,/\, (\,d_k \,|\, Ad_k\,)$
  $x_{k+1} = x_k + \tilde{\alpha}_k d_k$
  $r_{k+1} = r_k - \tilde{\alpha}_k Ad_k$
  Si $\|r_{k+1}\| \,/\, \|b\| < tol$: STOP
  $s_{k+1} = T^t \cdot Tr_{k+1}$
  $\tilde{\beta}_k = (\,r_{k+1} \,|\, s_{k+1}\,) \,/\, (\,r_k \,|\, s_k\,)$
  $d_{k+1} = s_{k+1} + \tilde{\beta}_k d_k$
Fin pour k

Algorithme 2.

Remarque 1.1 *Si $\hat{x} = A^{-1}b$ est la solution de (1.1) et $\widehat{\tilde{x}}$ celle de (1.2), nous avons, d'après le théorème 1.1,*

$$\begin{aligned}
\|x_m - \hat{x}\|_{\widetilde{A}} &= \left\|T^t(\tilde{x}_m - \widehat{\tilde{x}})\right\|_{\widetilde{A}} \leqslant \|T^t\|_{\widetilde{A}} \left\|\tilde{x}_m - \widehat{\tilde{x}}\right\|_{\widetilde{A}} \\
&\leqslant 2\,\|T^t\|_{\widetilde{A}} \left(\frac{\sqrt{\tilde{\kappa}} - 1}{\sqrt{\tilde{\kappa}} + 1}\right)^m \left\|\tilde{x}_0 - \widehat{\tilde{x}}\right\|_{\widetilde{A}} \\
&\leqslant 2\,\|T^t\|_{\widetilde{A}} \,\|(T^t)^{-1}\|_{\widetilde{A}} \left(\frac{\sqrt{\tilde{\kappa}} - 1}{\sqrt{\tilde{\kappa}} + 1}\right)^m \|x_0 - \hat{x}\|_{\widetilde{A}}
\end{aligned}$$

où $\tilde{\kappa}$ est la condition de la matrice $\widetilde{A} = TAT^t$ et $\|S\|_{\widetilde{A}} = \sup_{x \neq 0}(\|Sx\|_{\widetilde{A}} \,/\, \|x\|_{\widetilde{A}})$ est une norme opérateur de la matrice $S \in \mathbb{M}_n(\mathbb{R})$. Ainsi, la vitesse de convergence de l'algorithme GCP$_1$ est contrôlée par la condition $\tilde{\kappa}$ de $\widetilde{A}$.

1.2.2 Deuxième préconditionnement

Soit M une matrice SDP de taille $n \times n$. Le système

$$M^{-1}Ax = M^{-1}b \tag{1.3}$$

est aussi équivalent au système (1.1). La matrice M est appelée *préconditionneur (de type 2)* du système (1.1).
Comme la matrice M est supposée SDP, elle est orthogonalement diagonalisable de valeurs propres $\mu_1, \ldots, \mu_n$ toutes strictement positives : il existe une matrice régulière orthogonale Q (*i.e.* $Q^{-1} = Q^t$) telle que $M = QDQ^t$, où $D = \mathrm{diag}(\mu_1, \ldots, \mu_n)$. Notons $D^{1/2} = \mathrm{diag}(\mu_1^{1/2}, \ldots, \mu_n^{1/2})$, $M^{1/2} = QD^{1/2}Q^t$ et $M^{-1/2} = (M^{1/2})^{-1}$. Il est clair que $M^{1/2}$, $M^{-1/2}$ et M^{-1} sont SDP et que $(M^{1/2})^2 = M$ et $(M^{-1/2})^2 = M^{-1}$. Le système (1.3) est donc équivalent à

$$M^{1/2}M^{-1}AM^{-1/2}M^{1/2}x = M^{1/2}M^{-1}b,$$

c'est–à–dire à

$$M^{-1/2}AM^{-1/2}\tilde{x} = M^{-1/2}b,\ x = M^{-1/2}\tilde{x}. \tag{1.4}$$

Comme $M^{-1/2}$ est symétrique (et régulière), le système (1.4) est de la même forme que (1.2) avec $T = T^t = M^{-1/2}$ et l'algorithme 3 est obtenu.

Algorithme GCP2 : Avec M SDP

Soit $x_0 \in \mathbb{R}^n$ donné
Calculer $r_0 = b - Ax_0$, résoudre $Ms_0 = r_0$ et poser $d_0 = s_0$
Si $\|r_0\| / \|b\| < tol$: STOP
Pour $k \geqslant 0$, faire :
  $\tilde{\alpha}_k = (\, r_k \,|\, s_k \,) / (\, d_k \,|\, Ad_k \,)$
  $x_{k+1} = x_k + \tilde{\alpha}_k d_k$
  $r_{k+1} = r_k - \tilde{\alpha}_k Ad_k$
  Si $\|r_{k+1}\| / \|b\| < tol$: STOP
  Résoudre $Ms_{k+1} = r_{k+1}$
  $\tilde{\beta}_k = (\, r_{k+1} \,|\, s_{k+1} \,) / (\, r_k \,|\, s_k \,)$
  $d_{k+1} = s_{k+1} + \tilde{\beta}_k d_k$
Fin pour k

Algorithme 3.

Alors que dans l'algorithme GCP1 le calcul des s_k est direct ($s_k = T^t \cdot Tr_k$), dans l'algorithme GCP2 les s_k s'obtiennent indirectement ($Ms_k = r_k$) si l'inverse de M n'est pas connu ; cette étape peut être difficilement réalisable (voire irréalisable) et réduire la qualité de l'algorithme de manière significative. Les préconditionneurs de type 1 sont donc privilégiés.

1.3 Intérêts du préconditionnement

L'intérêt d'utiliser des préconditionneurs est évidemment de gagner du temps ! Le temps de calcul nécessaire pour résoudre le système (1.1) avec une méthode du gradient conjugué préconditionné se répartit entre

- le temps utilisé pour la construction du préconditionneur ;

- le temps utilisé pour la résolution du système préconditionné avec GCP1 (ou GCP2).

Si plusieurs systèmes linéaires SDP où seul le second membre change sont à résoudre, le préconditionnement peut apporter un gain de temps significatif. En effet, le préconditionneur se calcule une seule fois, puis, ce dernier permet d'améliorer la vitesse de convergence lors de la résolution de chaque système. Ainsi, lorsque le nombre de systèmes est grand, le temps de calcul du préconditionneur est négligeable.

Des systèmes linéaires SDP sont utilisés pour résoudre numériquement de nombreuses équations aux dérivées partielles non stationnaires. Un pas de temps est fixé et la solution d'un système linéaire est cherchée à chaque pas de temps. Ainsi, lorsque la matrice du système est indépendante du temps et que le nombre de pas de temps considérés est élevé, la situation décrite ci–dessus a lieu.

Remarque 1.2 *Un préconditionneur peut aussi être utile pour obtenir la convergence de la méthode du gradient conjugué (lorsque celle-ci ne converge pas, ce qui peut se produire en pratique).*

1.4 Type de matrices

La matrice A du système (1.1) sera considérée de grande taille et creuse, c'est–à–dire ayant une grande proportion de coefficients nuls. (Par exemple A d'ordre $n = 100'000, 300'000, \ldots$, avec 0,1% de coefficients non nuls.) C'est le cas lors de la modélisation de nombreux phénomènes physiques.

Pour des raisons de place en mémoire et de temps de calcul, les préconditionneurs doivent également être creux.

CHAPITRE 2

Quelques préconditionneurs connus

2.1 Préconditionneur diagonal

Le préconditionneur diagonal, de type 2, est donné par la matrice

$$M = \text{diag}(a_{11}, a_{22}, \ldots, a_{nn}).$$

Bien qu'il soit de type 2, son inverse est connu ($M^{-1} = \text{diag}(a_{11}^{-1}, \ldots, a_{nn}^{-1})$) et le calcul de $s_k = M^{-1} r_k$ dans l'algorithme GCP2 est direct. Ce préconditionneur peut être considéré de type 1 avec

$$T = \text{diag}(a_{11}^{-1/2}, \ldots, a_{nn}^{-1/2}).$$

2.2 Préconditionneur tridiagonal

Le préconditionneur tridiagonal, de type 2, est donné par la matrice

$$M = \begin{pmatrix} a_{11} & a_{12} & & \\ a_{21} & \ddots & \ddots & \\ & \ddots & \ddots & a_{n-1\,n} \\ & & a_{n\,n-1} & a_{nn} \end{pmatrix}.$$

Plus précisément, si $M = LL^t$ est la décomposition de Cholesky de M, la matrice L est triangulaire inférieure et ses coefficients sont nuls hors de la diagonale principale et de la sous-diagonale. La construction de L est rapide et, dans l'algorithme GCP2, le vecteur s_k vérifiant $Ms_k = r_k$ est obtenu en résolvant $Ly = r_k$ (détermination des composantes du vecteur y de la première à la dernière) puis $L^t s_k = y$ (détermination des composantes du vecteur s_k de la dernière à la première). La matrice L étant bidiagonale, la durée de chaque itération reste petite.

Remarque 2.1 *Nous pouvons procéder de la même manière avec une largeur de bande p fixée,* i.e. *avec $M = (m_{ij})$, où $m_{ij} = a_{ij}$ si $|i-j| \leqslant p$ et $m_{ij} = 0$ sinon.*

2.3 Décomposition de Cholesky incomplète

Un préconditionneur peut être construit à l'aide d'une décomposition de Cholesky incomplète.

Théorème 2.1 *[38, 46] Soit A une M–matrice[1] symétrique de taille $n \times n$ et $S \subset \{(i,j) \mid 1 \leqslant i,j \leqslant n,\ i \neq j\}$ un ensemble d'indices symétrique (*i.e. *$(i,j) \in S \iff (j,i) \in S$). Alors il existe une unique matrice triangulaire inférieure $L = (l_{ij})$ et une unique matrice symétrique $R = (r_{ij})$ telles que*

$$\begin{aligned} l_{ij} &= 0 \quad \textit{si } (i,j) \in S, \\ r_{ij} &= 0 \quad \textit{si } (i,j) \notin S \end{aligned}$$

*et telles que $A = LL^t - R$; de plus cette décomposition est régulière (*i.e. *LL^t est monotone et R est à coefficients positifs).*

La construction de L est donnée dans [38, 46]. Cette matrice fournit un préconditionneur de type 2, $M = LL^t$. Dans l'algorithme GCP2, s_k s'obtient en résolvant $Ly = r_k$ puis $L^t s_k = y$.
Remarquons qu'il est nécessaire de choisir l'ensemble S assez "gros" pour que la matrice L soit creuse et sa construction réalisable en un temps raisonnable. La matrice A étant creuse en général, nous pouvons par exemple considérer $S = \{(i,j) \mid a_{ij} = 0\}$; ce choix est appelé *A-remplissage.*

2.4 Inverse approché factorisé

Soit $A = L_A L_A^t$ la décomposition de Cholesky de A. L'idée est de construire une matrice $G = (g_{ij})$ triangulaire inférieure, approximation de L_A^{-1}, de sorte que GAG^t soit proche de la matrice identité. La matrice $T = G$ est alors un préconditionneur de type 1.

Présentons brièvement le principe de la construction de G décrite dans [36]. Un ensemble d'indices P contenu dans la partie inférieure et contenant la diagonale (*i.e.* $\{(i,i) \mid 1 \leqslant i \leqslant n\} \subset P \subset \{(i,j) \mid 1 \leqslant j \leqslant i \leqslant n\}$) est considéré et la norme de Frobenius de $I - GL_A$,

$$\|I - GL_A\|_F = \mathrm{Tr}((I - GL_A)(I - GL_A)^t)^{1/2},$$

est minimisée sous la contrainte

$$g_{ij} = 0\ , \quad \text{si } (i,j) \notin P.$$

Les équations suivantes sont obtenues

$$(GL_A L_A^t)_{ij} = (L_A^t)_{ij}\ , \quad (i,j) \in P,$$

[1] A est une M–matrice si ses coefficients hors de la diagonale sont négatifs et si A est monotone (*i.e.* régulière et d'inverse à coefficients positifs), voir [58].

c'est–à–dire, comme P est contenue dans la partie inférieure et que L_A^t est triangulaire supérieure,

$$(GA)_{ij} = \begin{cases} 0 & \text{si } (i,j) \in P,\ i \neq j, \\ (L_A)_{ii} & \text{si } i = j. \end{cases}$$

Comme les coefficients $(L_A)_{ii}$ sont inconnus, nous cherchons $\widetilde{G}$ vérifiant

$$(\widetilde{G}A)_{ij} = 0 \text{ si } (i,j) \in P,\ i \neq j, \tag{2.1}$$

$$(\widetilde{G}A)_{ii} = 1 \text{ pour } i = 1, \ldots, n, \tag{2.2}$$

$$\widetilde{G}_{ij} = 0 \text{ si } (i,j) \notin P, \tag{2.3}$$

puis posons $G = D\widetilde{G}$, où D est une matrice diagonale telle que $(GAG^t)_{ii} = 1$ pour $i = 1, \ldots, n$. Si $D = \mathrm{diag}(d_1, \ldots, d_n)$, alors, pour $i = 1, \ldots, n$,

$$\begin{aligned} 1 &= (GAG^t)_{ii} = (D\widetilde{G}A\widetilde{G}^t D)_{ii} = d_i^2 (\widetilde{G}A\widetilde{G}^t)_{ii} = d_i^2 \sum_{k=1}^{n} (\widetilde{G}A)_{ik} \widetilde{G}_{ik} \\ &= d_i^2 \sum_{k:(i,k)\in P} \underbrace{(\widetilde{G}A)_{ik}}_{\delta_{ik}} \widetilde{G}_{ik} = d_i^2 \widetilde{G}_{ii}, \end{aligned}$$

c'est–à–dire $d_i^2 = 1/\widetilde{G}_{ii}$. Nous obtenons alors, dans l'algorithme GCP1, $s_k = \widetilde{G}^t D^2 \widetilde{G} r_k$.

Les équations (2.1) à (2.3) définissent n systèmes indépendants : un pour chaque ligne de $\widetilde{G}$ (l'algorithme de sa construction est donc aisément parallélisable (voir chapitre 5)). Notons, pour $i = 1, \ldots, n$,

$$\mathcal{J}_i = \{ j \mid (i,j) \in P \}$$

l'ensemble des indices de colonne des éléments de la i–ème ligne de $\widetilde{G}$ à déterminer ($P = \cup_{i=1}^n \mathcal{J}_i$). Pour i fixé, avec

$$\mathcal{J} = \mathcal{J}_i = \{ j_1 < j_2 < \ldots < j_m = i \},$$

la i–ème ligne de $\widetilde{G}$ est déterminée par

$$(\widetilde{G}A)_{ij_l} = \delta_{lm} \text{ pour } l = 1, \ldots, m, \tag{2.4}$$

$$(\widetilde{G})_{ij} = 0 \text{ si } j \notin \mathcal{J}. \tag{2.5}$$

En notant $A_{\mathcal{J}\mathcal{J}}$ la matrice SDP de taille $m \times m$ obtenue en traçant dans A les lignes et colonnes n'appartenant pas à $\mathcal{J}$ et $\widetilde{G}_{i\,\mathcal{J}} = (\widetilde{G}_{ij_1}, \ldots, \widetilde{G}_{ij_m})$, (2.4) et (2.5) se récrivent matriciellement

$$\widetilde{G}_{i\,\mathcal{J}} A_{\mathcal{J}\mathcal{J}} = (0, \ldots, 0, 1), \tag{2.6}$$

où le membre de droite est un vecteur de longueur m. Le système (2.6) peut être résolu à l'aide de la décomposition de Cholesky de la matrice $A_{\mathcal{J}\mathcal{J}}$ (qui est SDP) en supposant que m ne soit pas trop grand.

Reste à choisir l'ensemble P (ou les ensembles $\mathcal{J}_i$, $i = 1, \ldots, n$). La matrice A étant creuse en général, un choix possible est le A-remplissage $P = \{(i,j) \mid 1 \leqslant j \leqslant i \leqslant n,\ a_{ij} \neq 0\}$. Remarquons qu'avec ce choix, l'ensemble $\mathcal{J}_i \times \mathcal{J}_i$ n'est pas nécessairement contenu dans $P \cup \overline{P}$ où $\overline{P} = \{(i,j) \mid (j,i) \in P\}$, *i.e.* les matrices $A_{\mathcal{J}_i\mathcal{J}_i}$ peuvent contenir des zéros.

CHAPITRE 3

Préconditionneurs utilisant une méthode de Gram–Schmidt conjuguée et des approximations au sens des moindres carrés

Dans ce chapitre, une nouvelle classe de préconditionneurs est développée. Elle se base sur un procédé d'A–orthogonalisation (ou méthode de Gram–Schmidt conjuguée) incomplet [10, 9, 11], des approximations au sens des moindres carrés et un méthode de remplissage optimale [27].

3.1 Méthode de Gram–Schmidt conjuguée

Considérons le produit scalaire donné par la matrice SDP A

$$(\, x \,|\, y \,)_A = (\, x \,|\, Ay \,) = x^t \cdot A \cdot y$$

ainsi que la norme associée $\|.\|_A$. Appliquons à la base canonique $\{e_1, \ldots, e_n\}$ de $\mathbb{R}^n$ le procédé d'orthogonalisation de Gram–Schmidt relativement à ce produit scalaire (appelé aussi procédé de Gram–Schmidt conjugué ou procédé d'A–orthogonalisation). Nous obtenons la base A–orthogonale $\{z_1, \ldots, z_n\}$ de $\mathbb{R}^n$ (*i.e.* $(\, z_i \,|\, z_j \,)_A = 0$ si $i \neq j$ et $z_i \neq 0$) définie par

$$\begin{aligned} z_1 &= e_1, \\ z_k &= e_k - \sum_{i=1}^{k-1} \frac{(\, e_k \,|\, z_i \,)_A}{(\, z_i \,|\, z_i \,)_A} z_i, \;\; k = 2, \ldots, n. \end{aligned} \tag{3.1}$$

La matrice

$$Z = (z_1, \ldots, z_n),$$

dont les colonnes sont les vecteurs z_k, est triangulaire supérieure de diagonale unitaire et satisfait la relation

$$Z^t \cdot A \cdot Z = D,$$

où $D = \mathrm{diag}(p_1, \ldots, p_n)$, avec $p_k = z_k^t \cdot A \cdot z_k = (\, z_k \,|\, z_k \,)_A = \|z_k\|_A^2$. (Remarquons que $A = (Z^{-1})^t \cdot D \cdot Z^{-1}$ fournit la décomposition de Cholesky de A.) Notons

$$p_{ik} = (\, e_i \,|\, z_k \,)_A = (\, a_i \,|\, z_k \,),$$

où $a_i = Ae_i$ désigne la i–ème colonne de A. Comme les vecteurs z_k sont A–orthogonaux deux à deux, il suit de (3.1) $p_k = p_{kk}$. L'algorithme de Gram–Schmidt conjugué est donné par l'algorithme 4.

Algorithme GSC

$z_i^{(0)} = e_i,\ i = 1, \ldots, n$
Pour $k = 1, \ldots, n$, faire :
  Pour $i = k, \ldots, n$, faire :
    $p_i^{(k)} = \left(a_i \,|\, z_k^{(k-1)} \right)$
  Fin pour i
  Pour $i = k+1, \ldots, n$, faire :
    $z_i^{(k)} = z_i^{(k-1)} - \left(p_i^{(k)} / p_k^{(k)} \right) z_k^{(k-1)}$
  Fin pour i
Fin pour k

Algorithme 4.

Nous avons $p_i^{(k)} = p_{ik}$ et, à la fin, $z_k = z_k^{(k-1)}$ et $p_k = p_k^{(k)}$. La quatrième ligne de l'algorithme GSC peut être remplacée par $p_i^{(k)} = \left(a_k \,|\, z_i^{(k-1)} \right)$. En effet, en utilisant (3.1) et la symétrie de A, il suit

$$\begin{aligned} \left(a_i \,|\, z_k^{(k-1)} \right) &= (\, a_i \,|\, e_k \,) - \sum_{j=1}^{k-1} \frac{(\, a_k \,|\, z_j \,)}{p_j} (\, a_i \,|\, z_j \,) \\ &= \left(a_k \,\middle|\, e_i - \sum_{j=1}^{k-1} \frac{(\, a_i \,|\, z_j \,)}{p_j} z_j \right) \\ &= \left(a_k \,|\, z_i^{(k-1)} \right) \end{aligned}$$

pour $i \geqslant k$.

L'algorithme GSC fournit l'inverse $A^{-1} = Z \cdot D^{-1} \cdot Z^t$ de A et avec $T = D^{-1/2} Z^t$, nous avons $TAT^t = I$.

3.1.1 Obtention d'un préconditionneur

L'idée pour obtenir un préconditionneur est de considérer une approximation de la matrice Z (encore notée Z). Elle fournira, avec la matrice diagonale correspondante $D = \mathrm{diag}((\, z_1 \,|\, Az_1 \,), \ldots, (\, z_n \,|\, Az_n \,))$, une estimation $M^{-1} = Z \cdot D^{-1} \cdot Z^t$ de A^{-1} et un préconditionneur

$$T = D^{-1/2} \cdot Z^t \tag{3.2}$$

pour le système (1.2) et l'algorithme GCP1 satisfaisant $TAT^t \approx I$. Remarquons que la matrice $M = Z^{-t} D Z^{-1}$ pour le système (1.3) donne le même préconditionneur puisque $M^{-1} = Z D^{-1} Z^t = T^t \cdot T$.

Considérons un ensemble d'indices P contenu dans la partie supérieure et contenant la diagonale, *i.e.*

$$\{(i,i) \mid 1 \leqslant i \leqslant n\} \subset P \subset \{(i,j) \mid 1 \leqslant i \leqslant j \leqslant n\} \tag{3.3}$$

et imposons

$$Z_{ij} = 0 \text{ si } (i,j) \notin P. \tag{3.4}$$

Dans les sections suivantes, différentes méthodes pour déterminer les coefficients non nuls de la matrice Z sont présentées.

3.2 Gram–Schmidt conjugué incomplet

Une des idées de [10] consiste à utiliser l'algorithme GSC en ignorant les coefficients de Z dont les indices ne sont pas dans P, où P satisfaisant (3.3) et (3.4) est donné. L'algorithme 5 (avec la notation $Z = (z_{ij})$) est obtenu.

Algorithme GSC INC

$Z = I$ (initialisation)
Pour $k = 1, \ldots, n$, faire :
  Pour $i = k, \ldots, n$, faire :
    $p_i = a_{ik}$
    Pour $j = 1, \ldots, k-1$, faire :
      Si $(j,i) \in P$: $p_i = p_i + a_{jk} z_{ji}$
    Fin pour j
  Fin pour i } $p_i^{(k)} = \left(a_k \mid z_i^{(k-1)} \right)$
  Pour $i = k+1, \ldots, n$, faire :
    $t = p_i / p_k$
    Pour $j = 1, \ldots, k$, faire :
      Si $(j,i) \in P$ et $(j,k) \in P$: $z_{ji} = z_{ji} - t z_{jk}$
    Fin pour j
  Fin pour i } $z_i^{(k)} = z_i^{(k-1)} - t z_k^{(k-1)}$
Fin pour k

Algorithme 5.

La matrice A étant creuse, nous pouvons par exemple choisir le A-remplissage $P = \{(i,j) \mid 1 \leqslant i \leqslant j \leqslant n,\ a_{ij} \neq 0\}$.

3.3 Approximation au sens des moindres carrés

Considérons un ensemble d'indices P satisfaisant (3.3) et construisons la matrice triangulaire supérieure de diagonale unitaire Z avec les contraintes (3.4). Supposons les colonnes $z_1, \ldots, z_{k-1}$ construites et montrons comment obtenir z_k. Posons $z_k(k) = Z_{kk} = 1$ et notons

$$\mathcal{J} = \mathcal{J}_k = \{j \mid (j,k) \in P,\ j \neq k\}$$

l'ensemble des indices des composantes de z_k à déterminer. Supposons $\mathcal{J}$ non vide ; notons $\mathcal{J} = \{j_1 < \ldots < j_p\} \subset \{1, \ldots, k-1\}$ et

$$y = (y_1, \ldots, y_p)^t = z_k(\mathcal{J}) = (Z_{j_1 k}, \ldots, Z_{j_p k})^t. \tag{3.5}$$

Nous cherchons z_k A–orthogonal aux vecteurs $z_1, \ldots, z_{k-1}$, c'est–à–dire, z_k vérifiant

$$(z_k \,|\, Az_i) = 0, \ i = 1, \ldots, k-1. \tag{3.6}$$

Avec les notations précédentes, nous avons

$$(z_k \,|\, Az_i) = (Az_k \,|\, z_i) = (a_k \,|\, z_i) + \sum_{l=1}^{p} y_l \, (a_{j_l} \,|\, z_i)$$

(où $a_{j_l} = Ae_{j_l}$ désigne la j_l–ème colonne de A) et (3.6) se récrit

$$\sum_{l=1}^{p} (a_{j_l} \,|\, z_i) \, y_l = - (a_k \,|\, z_i), \ i = 1, \ldots, k-1.$$

C'est un système linéaire de $k-1$ équations à p inconnues $(y_1, \ldots, y_p)$ qui s'écrit matriciellement

$$By = c \tag{3.7}$$

où $B = (b_{il})$ est la matrice de taille $(k-1) \times p$ avec $b_{il} = (a_{j_l} \,|\, z_i)$ et $c = (c_1, \ldots, c_{k-1})^t$ le vecteur de $\mathbb{R}^{k-1}$ avec $c_i = - (a_k \,|\, z_i)$.

Considérons $Z_{k-1} = (Z_{ij})_{1 \leqslant i,j \leqslant k-1}$ la sous-matrice mineure principale de Z d'ordre $k-1$, $\widetilde{A}$ la matrice de taille $(k-1) \times p$ obtenue en gardant dans A les $k-1$ premières lignes et les colonnes d'indices $j_1, \ldots, j_p$ ($\widetilde{A}_{il} = a_{ij_l}$) et $\tilde{a}_k$ le vecteur de $\mathbb{R}^{k-1}$ formé des $k-1$ premières composantes de a_k, la k–ème colonne de A. Comme les composantes $k, k+1, \ldots, n$ des vecteurs $z_1, \ldots, z_{k-1}$ sont nuls, il est clair que nous avons

$$B = Z_{k-1}^t \cdot \widetilde{A}$$

et

$$c = -Z_{k-1}^t \cdot \tilde{a}_k.$$

La matrice Z_{k-1} étant régulière (triangulaire supérieure de diagonale unitaire), le système (3.7) est équivalent à

$$\widetilde{A} y = -\tilde{a}_k. \tag{3.8}$$

Ce système n'admet en général pas de solution ($p \leqslant k-1$). Nous allons donc chercher sa solution au sens des moindres carrés, *i.e.* le vecteur $y \in \mathbb{R}^p$ qui minimise

$$\left\| \widetilde{A} y + \tilde{a}_k \right\|^2. \tag{3.9}$$

Cette solution y est donnée par le système

$$\widetilde{A}^t \cdot \widetilde{A} y = - \widetilde{A}^t \cdot \tilde{a}_k. \tag{3.10}$$

Comme la matrice A est SDP, la sous-matrice mineure A_{k-1} l'est aussi, ses colonnes sont donc linéairement indépendantes et $\widetilde{A}^t \cdot \widetilde{A}$ est SDP. Ainsi (3.10) admet une unique solution.

L'équation (3.10) est résolue à l'aide de la décomposition QR de $\widetilde{A}$. Les colonnes $\tilde{a}_1, \ldots, \tilde{a}_p$ de $\widetilde{A}$ étant linéairement indépendantes, cette matrice s'écrit $\widetilde{A} =$

QR, où $Q = (q_1, \ldots, q_p)$ est une matrice de taille $(k-1) \times p$ vérifiant $Q^t \cdot Q = I_p$ (*i.e.* les colonnes $q_1, \ldots, q_p$ de Q sont orthonormées) et $R = (r_{ij})$ une matrice carrée régulière d'ordre p triangulaire supérieure. Nous obtenons Q et R en appliquant le procédé d'orthonormalisation de Gram–Schmidt à la famille $\{\tilde{a}_1, \ldots, \tilde{a}_p\}$ (algorithme 6).

Décomposition QR
$r_{11} = \|\tilde{a}_1\|$
$q_1 = \tilde{a}_1 / r_{11}$
Pour $l = 2, \ldots, p$, faire :
$\quad r_{jl} = (\, \tilde{a}_l \,|\, q_j \,),\ j = 1, \ldots, l-1$
$\quad \tilde{q}_l = \tilde{a}_l - \sum_{j=1}^{l-1} r_{jl} q_j$
$\quad r_{ll} = \|\tilde{q}_l\|$
$\quad q_l = \tilde{q}_l / r_{ll}$
Fin pour l

Algorithme 6.

Il suffit alors de résoudre

$$Ry = -Q^t \tilde{a}_k \tag{3.11}$$

pour obtenir la solution de (3.10). En effet, comme $Q^t \cdot Q = I$,

$$\widetilde{A}^t \cdot \widetilde{A} y = -\widetilde{A}^t \tilde{a}_k \iff R^t Q^t \cdot QRy = -R^t Q^t \tilde{a}_k \iff Ry = -Q^t \tilde{a}_k.$$

Rappelons que le vecteur $y \in \mathbb{R}^p$ minimisant (3.9) est $y = z_k(\mathcal{J})$ (voir (3.5)), c'est–à–dire, le vecteur $\tilde{z}_k = (Z_{1k}, \ldots, Z_{k-1,k})^t$ avec $Z_{j_l k} = y_l$, $l = 1, \ldots, p$ et $Z_{ik} = 0$ si $i \notin \mathcal{J}$ réalise le minimum

$$m = \min_{\substack{u \in \mathbb{R}^{k-1} \\ u_i = 0,\ i \notin \mathcal{J}}} \|A_{k-1} u + \tilde{a}_k\| \,. \tag{3.12}$$

Reste à choisir, pour chaque $k \in \{2, \ldots, n\}$, l'ensemble $\mathcal{J}_k$ des indices des composantes de la k–ème colonne de Z pouvant être non nulles (coefficient Z_{kk} mis à part) (ou l'ensemble $P = \cup_{k=2}^{n} \mathcal{J}_k \cup \{(k,k) \mid k = 1, \ldots, n\}$).

Remarque 3.1 *Le calcul de la k–ème colonne z_k de la matrice Z est fait à partir de (3.8) où n'apparaît plus les colonnes précédentes $z_1, \ldots, z_{k-1}$; ainsi les colonnes de Z peuvent être calculées* ***indépendamment*** *les unes des autres, ce qui mène immédiatement à un algorithme parallélisable pour cette méthode (voir chapitre 5).*

3.4 Choix des coefficients non nuls

Proposons différentes manières de choisir les ensembles $\mathcal{J}_k$ (ou l'ensemble P).

***A*–remplissage :** Le choix le plus simple est de prendre

$$P = \{(i,j) \mid 1 \leqslant i \leqslant j \leqslant n,\ a_{ij} \neq 0\}.$$

Si la matrice A est creuse, la cardinalité de chaque $\mathcal{J}_k$ est petite.

Remplissage diagonal : Une autre possibilité est de fixer la cardinalité maximale p_{max} de chaque $\mathcal{J}_k$ et considérer un remplissage de Z contre la diagonale,

$$P = \{(i,j) \mid 0 \leqslant j - i \leqslant p_{max}\}.$$

Remplissage optimal : Choisissons les $\mathcal{J}_k$ plus judicieusement. Pour cela, inspirons nous de [27]. Fixons k, $2 \leqslant k \leqslant n$. L'ensemble $\mathcal{J} = \mathcal{J}_k$ et le vecteur z_k sont construits de sorte que le minimum m de (3.12) soit plus petit qu'un nombre ε donné. Pour cela, nous fixons le nombre maximal p_{max} d'indices dans $\mathcal{J}$ et le pas de remplissage s ($1 \leqslant s \leqslant p_{max}$) ; nous procédons alors de la manière suivante :

(i) initialiser $\mathcal{J} = \emptyset$,

(ii) calculer le vecteur $\tilde{z}_k$ réalisant le minimum m de (3.12),

(iii) si $m \leqslant \varepsilon$ ou $|\mathcal{J}| \geqslant p_{max}$: sortir de la boucle,

(iv) ajouter s indices à $\mathcal{J}$ et nous retournons au point (ii).

À la sortie de la boucle, nous posons $z_k = (\tilde{z}_k^t, 1, 0, \ldots, 0)^t \in \mathbb{R}^n$ (l'ensemble $\mathcal{J}$ vérifie alors $|\mathcal{J}| \leqslant p_{max} + s - 1$). Remarquons que la condition $m \leqslant \varepsilon$ du point (ii) évite un remplissage superflu de la matrice Z.

Il reste à décrire au point (iv) le choix des indices supplémentaires pour un ensemble $\mathcal{J}$ donné. Le but est d'ajouter les indices qui réduisent au plus le minimum de (3.12). Notons

$$r = A_{k-1}\tilde{z}_k + \tilde{a}_k,$$

où $\tilde{z}_k$ réalise le minimum de (3.12), *i.e.* $\|r\| = \min\{\|A_{k-1}u + \tilde{a}_k\| \mid u \in \mathbb{R}^{k-1},\ u_i = 0,\ i \notin \mathcal{J}\}$. Considérons le sous-espace $W = \langle A_{k-1}e_j \mid j \in \mathcal{J}\rangle$; comme $A_{k-1}\tilde{z}_k$ est la meilleure approximation de $-\tilde{a}_k$ dans W, r est orthogonal à W (voir par exemple [2, p. 217]), *i.e.*

$$(\, r \,|\, A_{k-1}e_j \,) = 0 \ \ \forall j \in \mathcal{J}. \tag{3.13}$$

Posons $\mathcal{L} = \{1 \leqslant l \leqslant k-1 \mid r_l \neq 0\}$ et pour chaque $l \in \mathcal{L}$, considérons l'ensemble $\mathcal{M}_l = \{1 \leqslant j \leqslant k-1 \mid a_{lj} \neq 0,\ j \notin \mathcal{J}\}$. Alors

$$\widetilde{\mathcal{J}} = \bigcup_{l \in \mathcal{L}} \mathcal{M}_l \tag{3.14}$$

est l'ensemble des indices "utiles" à ajouter à $\mathcal{J}$ pour réduire $\|r\|$. Remarquons que si $r \neq 0$, alors $\widetilde{\mathcal{J}} \neq \emptyset$. En effet, supposons $\widetilde{\mathcal{J}} = \emptyset$; alors pour tout $l \in \mathcal{L}$, $\mathcal{M}_l = \emptyset$, *i.e.* $a_{lj} = 0$ si $j \notin \mathcal{J}$. Donc, $(\, r \,|\, A_{k-1}e_j \,) = \sum_{l \in \mathcal{L}} r_l a_{lj} = 0$, pour tout $j \notin \mathcal{J}$. Ainsi, avec (3.13), r est orthogonal à toutes les colonnes de A_{k-1} et, comme A_{k-1} est régulière, il suit $r = 0$.

Pour chaque $j \in \widetilde{\mathcal{J}}$, le nombre $\mu_j \in \mathbb{R}$ qui réalise le minimum

$$\rho_j = \min_{\mu \in \mathbb{R}} \|r + \mu A_{k-1}e_j\|$$

est le réel qui annule la dérivée de la fonction quadratique

$$f_j(\mu) = \|r + \mu A_{k-1}e_j\|^2 = \|r\|^2 + 2\mu\,(\, r \,|\, A_{k-1}e_j \,) + \mu^2 \,\|A_{k-1}e_j\|^2\,,$$

c'est–à–dire

$$\mu_j = -\frac{(\, r \,|\, A_{k-1}e_j \,)}{\|A_{k-1}e_j\|^2}.$$

Nous avons alors

$$\begin{aligned} \rho_j^2 &= f_j(\mu_j) = \|r\|^2 - 2\frac{(\, r \,|\, A_{k-1}e_j \,)^2}{\|A_{k-1}e_j\|^2} + \frac{(\, r \,|\, A_{k-1}e_j \,)^2}{\|A_{k-1}e_j\|^2} \\ &= \|r\|^2 - \frac{(\, r \,|\, A_{k-1}e_j \,)^2}{\|A_{k-1}e_j\|^2}. \end{aligned}$$

Pour chaque $j \in \widetilde{\mathcal{J}}$, le poids

$$\omega_j = \frac{(\, r \,|\, A_{k-1}e_j \,)^2}{\|A_{k-1}e_j\|^2}$$

est associé à j. Plus le poids ω_j est grand, plus l'indice j sera "efficace". Par conséquent, s indices de $\widetilde{\mathcal{J}}$ sont sélectionnés parmi ceux de plus grand poids et ajoutés à $\mathcal{J}$; si $|\widetilde{\mathcal{J}}| \leqslant s$, tous les indices de $\widetilde{\mathcal{J}}$ sont ajoutés à $\mathcal{J}$. En cas d'égalité de poids, l'indice le plus proche de k (de la diagonale) est choisi.

Avec les deux ensembles disjoints $\mathcal{J} = \{j_1, \ldots, j_p\}$ et $\widetilde{\mathcal{J}} = \{\tilde{j}_1, \ldots, \tilde{j}_s\}$, montrons comment procéder pour calculer le vecteur $\tilde{z}_k$ réalisant le minimum

$$\min_{\substack{u \in \mathbb{R}^{k-1} \\ u_i = 0,\ i \notin \mathcal{J} \sqcup \widetilde{\mathcal{J}}}} \|A_{k-1}u + \tilde{a}_k\|.$$

Posons $y = (Z_{j_1 k}, \ldots, Z_{j_p k}, Z_{\tilde{j}_1 k}, \ldots, Z_{\tilde{j}_s k})^t$. Pour utiliser (3.11), la décomposition $\widehat{A} = \widehat{Q}\widehat{R}$ de la matrice

$$\widehat{A} = (A_{k-1}e_{j_1}, \ldots, A_{k-1}e_{j_p}, A_{k-1}e_{\tilde{j}_1}, \ldots, A_{k-1}e_{\tilde{j}_s}),$$

est calculée à partir de la décomposition $\widetilde{A} = QR$ de $\widetilde{A} = (A_{k-1}e_{j_1}, \ldots, A_{k-1}e_{j_p})$, connue de l'étape précédente. Comme nous avons (en utilisant la notation par blocs)

$$\widehat{A} = (\widetilde{A}, \widehat{A}_{12}) = (Q, \widehat{Q}_{12}) \begin{pmatrix} R & \widehat{R}_{12} \\ 0 & \widehat{R}_{22} \end{pmatrix},$$

il suffit de prolonger la décomposition $\widetilde{A} = QR$, c'est–à–dire de calculer seulement les s dernières colonnes de $\widehat{Q}$ et $\widehat{R}$. Remarquons que la suite d'indices $j_1, \ldots, j_p, \tilde{j}_1, \ldots, \tilde{j}_s$ n'est pas nécessairement croissante.

3.5 Résultats théoriques

Le préconditionneur $T = D^{-1/2}Z^t$ (voir (3.2)) construit par l'une des méthodes précédentes fournit la matrice

$$S = TAT^t = D^{-1/2} \cdot Z^t \cdot A \cdot Z \cdot D^{-1/2}, \tag{3.15}$$

pour le système préconditionné. Considérons la construction de la matrice Z décrite à la section 3.3 avec le remplissage optimal (section 3.4). Quelques propriétés théoriques de la matrice $S = (s_{ij})$ sont données dans cette section, en particulier une majoration de sa condition.

Théorème 3.1 *Pour $k = 2, \ldots, n$, notons m_k le minimum de l'équation (3.12) réalisé par le vecteur formé des $k-1$ premières composantes de la k–ème colonne de Z. Supposons que $m_k \leqslant \varepsilon$ pour $k = 2, \ldots, n$ et notons $\lambda_{min} = \lambda_{min}(A)$ la plus petite valeur propre de A. Alors*

$$\begin{aligned} |s_{ik}| &\leqslant \varepsilon/\lambda_{min}, \text{ si } i \neq k, \\ s_{kk} &= 1, \ 1 \leqslant k \leqslant n \end{aligned}$$

et

$$\begin{aligned} \|S - I\|_F, \ \|S - I\|_2 &\leqslant \sqrt{n(n-1)} \cdot \varepsilon/\lambda_{min}, \\ \|S - I\|_1 &\leqslant (n-1)\varepsilon/\lambda_{min}, \end{aligned}$$

où $\|X\|_F = (Tr(XX^t))^{1/2} = (Tr(X^t \cdot X))^{1/2}$ est la norme de Frobenius de X et $\|X\|_i = sup_{\|y\|_i=1} \|X \cdot y\|_i$ est la norme opérateur provenant de la norme $\|.\|_i$ de $\mathbb{R}^n$, $i = 1, 2$, avec $\|y\|_1 = \sum_{j=1}^n |y_j|$, $\|y\|_2 = \left(\sum_{j=1}^n y_j^2\right)^{1/2}$.

Preuve. Lorsque la norme n'est pas mentionnée explicitement, la norme euclidienne de $\mathbb{R}^n$ est utilisée. D'après (3.15), nous avons

$$s_{ik} = \frac{(\, z_i \,|\, Az_k \,)}{\|z_i\|_A \, \|z_k\|_A}.$$

Il est clair que $s_{kk} = 1$ pour tout k. Comme S est symétrique, il suffit de montrer l'inégalité $|s_{ik}| \leqslant \varepsilon/\lambda_{min}$ pour $i < k$. Fixons i, k, avec $i < k$ et pour $x \in \mathbb{R}^n$, notons $\tilde{x}$ le vecteur de $\mathbb{R}^{k-1}$ formé des $k-1$ premières composantes de x. Comme Z est triangulaire supérieure de diagonale unitaire, nous avons

$$(\, z_i \,|\, Az_k \,) = (\, \tilde{z}_i \,|\, A_{k-1}\tilde{z}_k + \tilde{a}_k \,).$$

Comme par hypothèse $m_k = \|A_{k-1}\tilde{z}_k + \tilde{a}_k\| \leqslant \varepsilon$, en utilisant l'inégalité de Cauchy–Schwarz, nous obtenons

$$|\,(\, z_i \,|\, Az_k \,)\,| \leqslant \|\tilde{z}_i\| \cdot \varepsilon = \|z_i\| \cdot \varepsilon. \tag{3.16}$$

Le quotient de Rayleigh pour la matrice A défini par $\mu(x) = (\, x \,|\, Ax \,) / (\, x \,|\, x \,)$, $x \neq 0$ satisfait $\lambda_{min} = \min_{x \neq 0} \mu(x)$ et $\lambda_{max} = \max_{x \neq 0} \mu(x)$ (voir par exemple [46]). Ainsi, nous avons

$$\frac{\|z_i\|}{\|z_i\|_A} = \frac{1}{\sqrt{\mu(z_i)}} \leqslant \frac{1}{\sqrt{\lambda_{min}}} \tag{3.17}$$

et

$$\|z_k\|_A = \sqrt{\mu(z_k)} \cdot \|z_k\| \geqslant \sqrt{\lambda_{min}} \cdot \|z_k\|. \tag{3.18}$$

Avec les estimations (3.16),(3.17) et (3.18), nous obtenons

$$|s_{ik}| = \frac{|\,(\, z_i \,|\, Az_k \,)\,|}{\|z_i\|_A \, \|z_k\|_A} \leqslant \frac{\|z_i\| \cdot \varepsilon}{\|z_i\|_A \cdot \|z_k\|_A} \leqslant \frac{\varepsilon}{\lambda_{min} \cdot \|z_k\|}.$$

Comme $\|z_k\| \geqslant 1$, la première partie du théorème est prouvée.

Notons e_k le k–ème vecteur de la base canonique de $\mathbb{R}^n$. Par ce qui précède, nous avons

$$\|(S-I)e_k\|_2^2 = \sum_{i=1}^{n}(s_{ik}-\delta_{ik})^2 = \sum_{i\neq k}(s_{ik})^2 \leqslant (n-1)\frac{\varepsilon^2}{\lambda_{min}^2}$$

et

$$\|(S-I)e_k\|_1 = \sum_{i\neq k}|s_{ik}| \leqslant (n-1)\frac{\varepsilon}{\lambda_{min}}.$$

Par conséquent, pour la norme de Frobenius, nous obtenons

$$\|S-I\|_F^2 = \mathrm{Tr}((S-I)^t\cdot(S-I)) = \sum_{k=1}^{n}\|(S-I)e_k\|_2^2 \leqslant n(n-1)\frac{\varepsilon^2}{\lambda_{min}^2}.$$

Pour $i=1,2$, nous avons

$$\begin{aligned}\|S-I\|_i &= \sup_{\|x\|_i=1}\|(S-I)x\|_i = \sup_{\|x\|_i=1}\left\|\sum_{k=1}^{n}x_k(S-I)e_k\right\|_i \\ &\leqslant \sup_{\|x\|_i=1}\sum_{k=1}^{n}|x_k|\cdot\|(S-I)e_k\|_i .\end{aligned}$$

Ainsi

$$\|S-I\|_1 \leqslant \sup_{\|x\|_1=1}\sum_{k=1}^{n}|x_k|\cdot\|(S-I)e_k\|_1 \leqslant (n-1)\frac{\varepsilon}{\lambda_{min}}.$$

Comme $\|x\|_1 \leqslant \sqrt{n}\,\|x\|_2$ pour tout $x\in\mathbb{R}^n$ et $\sup_{\|x\|_2=1}\|x\|_1 = \sqrt{n}$, nous obtenons

$$\begin{aligned}\|S-I\|_2 &\leqslant \sup_{\|x\|_2=1}\sum_{k=1}^{n}|x_k|\cdot\|(S-I)e_k\|_2 \leqslant \sqrt{(n-1)}\frac{\varepsilon}{\lambda_{min}}\sup_{\|x\|_2=1}\|x\|_1 \\ &= \sqrt{n(n-1)}\frac{\varepsilon}{\lambda_{min}}.\end{aligned}$$

■

Corollaire 3.2 *Avec les hypothèses du théorème précédent et $\delta = (n-1)\varepsilon/\lambda_{min}(A)$, nous avons*

$$|\lambda - 1| \leqslant \delta$$

pour toute valeur propre λ de S. En particulier, si $\delta < 1$, la condition $\kappa(S)$ de S vérifie

$$\kappa(S) \leqslant \frac{1+\delta}{1-\delta}.$$

Preuve. Soit λ une valeur propre de S et v un vecteur propre de S de valeur propre λ avec $\|v\|_1 = 1$. Alors, par le théorème précédent,

$$|\lambda-1| = \|(\lambda-1)v\|_1 = \|(S-I)v\|_1 \leqslant \|S-I\|_1\cdot\|v\|_1 \leqslant \delta.$$

(Remarquons que le même raisonnement avec la norme $\|.\|_2$ fournit l'inégalité avec $\delta = \sqrt{n(n-1)}\cdot\varepsilon/\lambda_{min}(A)$, ce qui est moins bon.) ■

3.6 Variantes

Dans cette section, quelques variantes pour obtenir des préconditionneurs sont présentées.

3.6.1 Préconditionneurs couplés

Deux préconditionneurs peuvent être appliqués successivement, T_1 à la matrice A, puis T_2 à la matrice $\widetilde{A} = T_1 A T_1^t$. Le préconditionneur résultant est

$$T = T_2 T_1$$

et la matrice du système préconditionné est

$$T_2 \widetilde{A} T_2^t = T_2 T_1 A T_1^t T_2^t = T A T^t.$$

Le cas du préconditionneur diagonal pour T_1 a l'avantage de conserver la structure de la matrice, *i.e.* les positions des coefficients non nuls de A et $\widetilde{A}$ sont les mêmes, ce qui permet aisément d'appliquer un second préconditionneur T_2.

3.6.2 Traitement par blocs

Dans ce cas, une décomposition par blocs de la matrice A est considérée. Soit $A_1, \ldots, A_M$ les blocs diagonaux, où A_i est une matrice d'ordre m_i, avec $m_1 + \ldots + m_M = n$ (l'ordre de A). Pour chaque bloc A_i un préconditionneur T_i est construit. Alors $T = \mathrm{diag}(T_1, \ldots, T_M)$ est un préconditionneur de A.

Remarque 3.2 *Considérons le préconditionneur de la section 3.3 avec le remplissage optimal (voir section 3.4). Pour le calcul de la k–ème colonne, les "meilleurs" indices sont sélectionnés parmi les $k-1$ positions au–dessus de la diagonale. Ainsi, plus k est grand, plus le calcul de la k–ème colonne est long. Avec un traitement par blocs, le temps de calcul est réduit et le préconditionneur obtenu reste de bonne qualité (voir chapitre 4).*

CHAPITRE 4

Tests numériques

Dans ce chapitre, l'algorithme du gradient conjugué (GC, GCP1, GCP2) est testé pour le système $Ax = b$ avec les préconditionneurs des chapitres précédents :

0. AUCUN ;
1. DIAG : préconditionneur diagonal, $T = \text{diag}(a_{11}^{-1/2}, \ldots, a_{nn}^{-1/2})$ (voir section 2.1) ;
2. TRIDIAG : préconditionneur tridiagonal (voir section 2.2) ;
3. CHO INC : décomposition de Cholesky incomplète avec A-remplissage (voir section 2.3 et [38, 46]) ;
4. IAF : inverse approché factorisé avec A-remplissage (voir section 2.4 et [36]) ;
5. GSC INC : Gram–Schmidt conjugué incomplet avec A-remplissage (voir section 3.2 et [10]) ;
6. GSC MC (A) : Gram–Schmidt conjugué avec approximations de moindres carrés et A-remplissage (voir section 3.3 et 3.4) ;
7. GSC MC (DIAG) : Gram–Schmidt conjugué avec approximations de moindres carrés et remplissage diagonal (voir section 3.3 et 3.4) ;
8. GSC MC (OPT) : Gram–Schmidt conjugué avec approximations de moindres carrés et remplissage optimal (voir section 3.3 et 3.4) ;
9. DIAG + GSC MC (OPT) : préconditionneur couplé, diagonal puis GSC MC (OPT) ;
10. GSC MC (OPT) BLOCS : préconditionneur GSC MC (OPT) par blocs ;
11. DIAG + GSC MC (OPT) BLOCS : préconditionneur DIAG + GSC MC (OPT) par blocs ;

Pour les préconditionneurs GSC MC (OPT) et DIAG + GSC MC (OPT), la tolérance est fixée à $\varepsilon \approx \lambda_{min}/(n-1)$ de sorte que le nombre δ du corollaire 3.2 soit proche de 1, avec $\lambda_{min} = \lambda_{min}(A)$ pour le préconditionneur GSC MC (OPT) et

$\lambda_{min} = \lambda_{min}(T_1 A T_1^t)$, où $T_1 = \text{diag}(a_{11}^{-1/2}, \ldots, a_{nn}^{-1/2})$ pour le préconditionneur DIAG + GSC MC (OPT). Pour le préconditionneur GSC MC (OPT) BLOCS (resp. DIAG + GSC MC (OPT) BLOCS), la même tolérance ε que pour le préconditionneur GSC MC (OPT) (resp. DIAG + GSC MC (OPT)) sera considérée, bien que les valeurs propres des blocs diagonaux ne sont pas les mêmes que celle de la matrice entière. De plus, si M blocs sont pris en compte, leur ordre sera $m_1 = \ldots = m_r = q + 1$ et $m_{r+1} = \ldots = m_M = q$, où $n = q \cdot M + r$ est la division euclidienne de n par M (q, r sont entiers avec $0 \leqslant r < M$).

Considérons les matrices tests SDP du tableau 1 ; leur ordre est noté n et le nombre de coefficients non nuls contenu dans la partie supérieure (ou inférieure) est noté *NNZ*. Les structures de ces matrices (*i.e.* les positions des coefficients non nuls) sont représentées sur les figures 2–4).

Matrice (A)	n	*NNZ*
nos1	237	627
bcsstk27	1224	28675
s1rmt3m1	5489	112505

Tableau 1.

Ces matrices peuvent être trouvées sur les sites web

http ://math.nist.gov/MatrixMarket,
http ://www.cise.ufl.edu/research/sparse/matrices.

Pour chaque test, le second membre $b = (1, \ldots, 1)^t \in \mathbb{R}^n$ est considéré et la tolérance de convergence *tol* est fixée à 10^{-8} dans l'algorithme du gradient conjugué. Le point de départ choisi est $x_0 = 0$.

Pour chaque matrice, les résultats sont présentés dans des tableaux. Le nombre d'itérations pour la résolution est noté *It*. Toutefois, si une erreur intervient lors de la construction du préconditionneur (resp. lors de la résolution), la notation "EP" (resp. "EGC") est utilisée. Si la norme du résidu est plus grande que la tolérance *tol* après n itérations, la notation "$> n$" est utilisée.

Des estimations de la valeur propre maximale λ_{max} et de la valeur propre minimale λ_{min} sont données, ce qui permet d'obtenir la condition κ de la matrice TAT^t du système préconditionné (voir (1.2)). (La méthode de la puissance et de la puissance inverse (voir [52, p. 471-472]) est utilisée pour estimer ces valeurs.)

Les temps de calculs t_{prec} pour la construction du préconditionneur et t_{res} pour la résolution sont donnés. Les tests sont effectués sur une machine *Intel® Pentium® 4* cadencée à 2.66 GHz.

Enfin, le nombre de coefficients non nuls *nnz* des préconditionneurs GSC MC (DIAG), GSC MC (OPT), DIAG + GSC MC (OPT), GSC MC (OPT) BLOCS et DIAG + GSC MC (OPT) BLOCS figure dans les tableaux. Remarquons que pour les préconditionneurs CHO INC, IAF, GSC INC et GSC MC (A) nous avons *nnz* = *NNZ*.

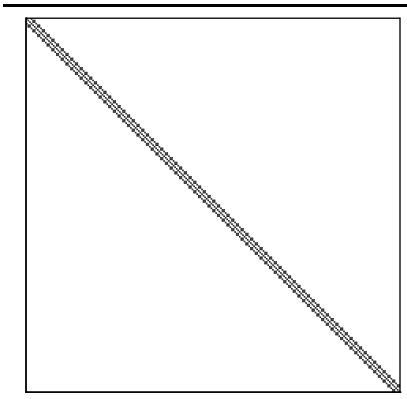

Figure 2: ***nos1.***

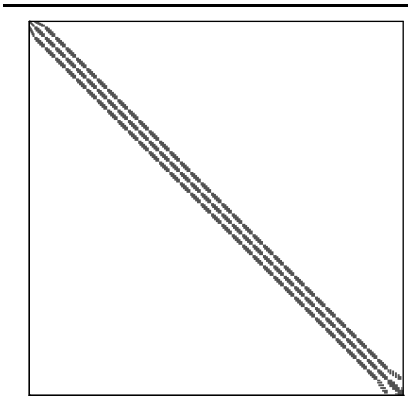

Figure 3: ***bcsstk27.***

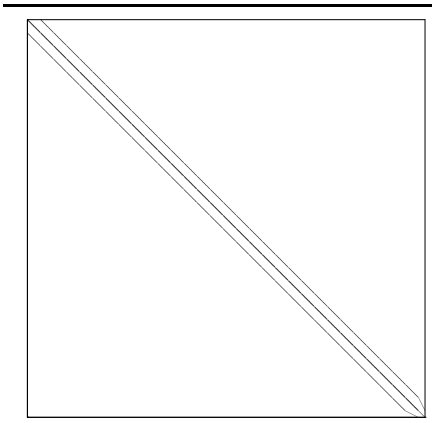

Figure 4: ***s1rmt3m1.***

4.1 Premier test : matrice *nos1*

Les résultats obtenus pour la matrice *nos1* sont présentés dans les tableaux 2–9. Notons que les temps de calculs (t_{prec} et t_{res}) sont ici peu significatifs vu le petit nombre de coefficients non nuls dans la matrice.

Les premiers tests (tableaux 2–5) mettent en évidence qu'*en pratique* l'algorithme du gradient conjugué ne converge pas nécessairement et montrent l'efficacité des préconditionneurs GSC MC (DIAG), GSC MC (OPT), DIAG +GSC MC (OPT). En effet, ce sont les seuls qui font converger l'algorithme pour cette matrice. Remarquons que dans le cas du préconditionneur GSC MC (DIAG), pour une petite valeur de p_{max}, l'algorithme ne converge pas ; ceci est aussi vrai pour une valeur de p_{max} trop élevée. L'explication vient probablement des erreurs d'arrondis.

Préc.	*It*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
AUCUN	>237	0.00E+00	1.00E−02	1.23E+02	2.46E+09	1.99E+07
DIAG	>237	0.00E+00	1.00E−02	5.09E−07	2.00E+00	3.93E+06
TRIDIAG	>237	0.00E+00	2.00E−02	—	—	—
CHO INC	EGC	—	—	—	—	—
IAF	>237	0.00E+00	2.00E−02	1.92E−06	1.69E+00	8.79E+05
GSC INC	>237	0.00E+00	0.00E+00	?	?	?
GSC MC (A)	>237	0.00E+00	3.00E−02	1.65E−06	2.63E+00	1.60E+06

Tableau 2.

Préconditionneur : GSC MC (DIAG)							
p_{max}	*It*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
1	>237	473	0.00E+00	4.00E−02	5.09E−07	2.08E+00	4.08E+06
5	>237	1407	1.00E−02	3.00E−02	2.16E−06	3.05E+00	1.41E+06
10	115	2252	1.00E−02	2.00E−02	2.83E−05	2.05E+00	7.25E+04
20	58	4767	3.00E−02	1.00E−02	2.04E−04	1.89E+00	9.25E+03
50	25	10812	1.10E−01	0.00E+00	4.74E−03	1.64E+00	3.46E+02
100	145	18887	5.70E−01	9.00E−02	3.17E−07	1.28E+01	4.06E+07
200	>237	27537	1.84E+00	1.60E−01	?	1.84E+01	?

Tableau 3.

Préconditionneur : GSC MC (OPT) avec ε =5.23E−01, $s = 1$							
p_{max}	*It*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
1	>237	471	1.00E−02	3.00E−02	1.31E−06	2.65E+00	2.02E+06
5	>237	1391	3.00E−02	3.00E−02	2.15E−06	4.25E+00	1.97E+06
10	>237	2496	5.00E−02	3.00E−02	3.05E−06	2.69E+00	8.80E+05
20	179	4556	1.30E−01	3.00E−02	4.43E−06	2.47E+00	5.57E+05
50	44	9536	7.40E−01	1.00E−02	1.20E−04	2.21E+00	1.84E+04
100	14	14067	2.20E+00	1.00E−02	1.75E−02	1.73E+00	9.85E+01
200	2	15720	3.20E+00	0.00E+00	1.00E+00	1.00E+00	1.00E+00

Tableau 4.

Préconditionneur : DIAG + GSC MC (OPT) avec ε =2.16E−09, $s = 1$							
p_{max}	*It*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
1	>237	471	1.00E−02	1.00E−02	1.20E−06	2.02E+00	1.67E+06
5	151	1391	2.00E−02	1.00E−02	9.25E−06	1.79E+00	1.93E+05
10	112	2496	4.00E−02	1.00E−02	1.30E−05	1.80E+00	1.39E+05
20	98	4556	1.00E−01	1.00E−02	1.33E−05	2.01E+00	1.51E+05
50	33	9536	5.00E−01	1.00E−02	1.61E−04	1.88E+00	1.17E+04
100	15	14067	1.40E+00	1.00E−02	5.30E−03	1.72E+00	3.25E+02
200	2	15720	2.17E+00	0.00E+00	1.00E+00	1.00E+00	1.00E+00

Tableau 5.

Dans les tableaux 6 et 7 figurent les résultats obtenus pour les préconditionneurs GSC MC (OPT) et DIAG + GSC MC (OPT) avec le paramètre p_{max} fixé et différentes valeurs pour le paramètre s ; la raison pour laquelle la construction du préconditionneur échoue (EP) est que l'ensemble $\widetilde{\mathcal{J}}$, voir (3.14), est vide. Augmenter le paramètre ε permettrait de calculer entièrement le préconditionneur.

Préconditionneur : GSC MC (OPT) avec ε =5.23E−01, $p_{max} = 50$							
s	It	nnz	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	37	9536	4.60E−01	0.00E+00	2.40E−04	2.09E+00	8.68E+03
3	>237	9643	3.10E−01	5.00E−02	?	4.08E+00	?
4	EP	—	—	—	—	—	—
5	>237	9802	1.80E−01	6.00E−02	?	6.26E+00	?
50	>237	9802	1.70E−01	6.00E−02	?	6.26E+00	?

Tableau 6.

Préconditionneur : DIAG + GSC MC (OPT) avec ε =2.16E−09, $p_{max} = 50$							
s	It	nnz	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	25	9536	2.60E−01	1.00E−02	8.75E−05	1.91E+00	2.19E+04
3	147	9643	2.30E−01	4.00E−02	2.44E−06	2.54E+00	1.04E+06
4	EP	—	—	—	—	—	—
5	EP	—	—	—	—	—	—
50	EP	—	—	—	—	—	—

Tableau 7.

Les résultats pour les préconditionneurs blocs GSC MC (OPT) BLOCS et DIAG + GSC MC (OPT) BLOCS sont donnés dans les tableaux 8 et 9.

Préconditionneur : GSC MC (OPT) BLOCS avec ε =5.23E−01, $p_{max} = 10$, $s = 1$							
Blocs	It	nnz	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	>237	2386	4.00E−02	3.00E−02	3.07E−06	2.68E+00	8.75E+05
3	>237	2275	3.00E−02	4.00E−02	3.12E−06	2.67E+00	8.58E+05
4	>237	2164	2.00E−02	2.00E−02	3.14E−06	2.65E+00	8.46E+05
6	216	1942	2.00E−02	2.00E−02	3.24E−06	2.52E+00	7.78E+05
8	191	1721	2.00E−02	1.00E−02	3.44E−06	2.30E+00	6.68E+05
16	179	1080	0.00E+00	2.00E−02	2.54E−06	2.00E+00	7.87E+05
24	>237	757	1.00E−02	1.00E−02	1.70E−06	2.00E+00	1.18E+06
32	>237	594	1.00E−02	1.00E−02	1.24E−06	2.00E+00	1.61E+06

Tableau 8.

Préc. : DIAG + GSC MC (OPT) BLOCS avec ε =2.16E−09, $p_{max} = 10$, $s = 1$							
Blocs	It	nnz	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	120	2386	3.00E−02	2.00E−02	7.95E−06	2.00E+00	2.52E+05
3	130	2275	2.00E−02	1.00E−02	7.34E−06	2.04E+00	2.78E+05
4	136	2164	2.00E−02	1.00E−02	6.43E−06	2.04E+00	3.17E+05
6	149	1942	1.00E−02	1.00E−02	5.21E−06	2.04E+00	3.92E+05
8	158	1721	1.00E−02	2.00E−02	4.31E−06	2.04E+00	4.73E+05
16	184	1080	1.00E−02	1.00E−02	2.54E−06	2.00E+00	7.87E+05
24	>237	757	0.00E+00	3.00E−02	1.70E−06	2.00E+00	1.18E+06
32	>237	594	0.00E+00	3.00E−02	1.24E−06	2.00E+00	1.61E+06

Tableau 9.

4.2 Deuxième test : matrice *bcsstk27*

Les résultats obtenus pour la matrice *bcsstk27* sont présentés dans les tableaux 10–17.

Préc.	It	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
AUCUN	1190	0.00E+00	5.50E−01	1.44E+02	3.47E+06	2.41E+04
DIAG	253	0.00E+00	1.40E−01	2.12E−03	4.37E+00	2.06E+03
TRIDIAG	253	0.00E+00	1.20E−01	—	—	—
CHO INC	24	5.00E−02	4.00E−02	—	—	—
IAF	89	7.00E−02	1.10E−01	5.31E−03	1.71E+00	3.22E+02
GSC INC	99	2.00E−02	1.10E−01	9.24E−03	3.12E+00	3.38E+02
GSC MC (A)	213	6.23E+00	2.80E−01	1.67E−03	3.17E+00	1.90E+03

Tableau 10.

Préconditionneur : GSC MC (DIAG)							
p_{max}	It	nnz	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
1	343	2447	2.20E−01	2.60E−01	1.18E−03	4.65E+00	3.93E+03
5	331	7329	3.60E−01	2.10E−01	9.90E−04	3.90E+00	3.94E+03
10	401	13409	8.10E−01	3.20E−01	4.60E−04	3.37E+00	7.33E+03
20	439	25494	2.71E+00	5.20E−01	4.81E−04	3.98E+00	8.27E+03
50	343	61149	3.13E+01	7.00E−01	1.00E−03	5.13E+00	5.10E+03
100	179	118574	2.10E+02	8.60E−01	3.61E−03	4.47E+00	1.24E+03

Tableau 11.

Préconditionneur : GSC MC (OPT) avec ε =1.17E−01, $s = 1$							
p_{max}	It	nnz	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
1	144	2447	1.16E+00	8.00E−02	4.19E−03	2.47E+00	5.89E+02
5	119	7329	8.74E+00	9.00E−02	6.19E−03	2.62E+00	4.23E+02
10	105	13409	2.64E+01	7.00E−02	6.90E−03	2.52E+00	3.65E+02
20	93	25487	8.60E+01	1.00E−01	8.22E−03	2.36E+00	2.87E+02
50	70	61139	5.20E+02	1.30E−01	1.27E−02	2.01E+00	1.58E+02
100	57	118549	1.87E+03	1.90E−01	2.09E−02	1.95E+00	9.32E+01

Tableau 12.

Préconditionneur : DIAG + GSC MC (OPT) avec ε =1.73E−06, $s = 1$							
p_{max}	It	nnz	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
1	139	2447	1.18E+00	8.00E−02	4.56E−03	2.46E+00	5.39E+02
5	103	7329	7.69E+00	7.00E−02	5.06E−03	1.89E+00	3.73E+02
10	84	13409	2.16E+01	9.00E−02	7.46E−03	1.69E+00	2.26E+02
20	70	25494	6.47E+01	9.00E−02	1.15E−02	1.69E+00	1.47E+02
50	52	61149	3.74E+02	1.20E−01	2.00E−02	1.71E+00	8.57E+01
100	40	118570	1.49E+03	1.40E−01	3.01E−02	1.56E+00	5.18E+01

Tableau 13.

Le préconditionneur GSC INC a $nnz = 28675$ coefficients non nuls et nous obtenons 99 itérations (voir tableau 10). Avec le préconditionneur DIAG + GSC MC (OPT), $p_{max} = 20$ (tableau 13), le remplissage est $nnz = 25494$ et 70 itérations sont nécessaires pour obtenir la convergence. Donc cette dernière méthode donne les meilleurs résultats pour ce type de préconditionneurs.

Préconditionneur : GSC MC (OPT) avec ε =1.17E−01, $p_{max} = 50$							
s	It	nnz	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	75	61145	2.31E+02	1.40E−01	1.24E−02	2.03E+00	1.63E+02
3	>1224	62319	1.97E+02	2.53E+00	?	7.08E+00	?
4	>1224	63492	1.61E+02	2.38E+00	?	6.13E+00	?
5	76	61146	1.19E+02	1.50E−01	1.16E−02	2.08E+00	1.80E+02
10	77	61146	8.18E+01	1.50E−01	1.06E−02	2.23E+00	2.11E+02
25	EP	—	—	—	—	—	—
50	EP	—	—	—	—	—	—

Tableau 14.

Préconditionneur : DIAG + GSC MC (OPT) avec ε =1.73E−06, $p_{max} = 50$							
s	*It*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	53	61149	1.76E+02	8.00E−02	1.80E−02	1.62E+00	8.96E+01
3	119	62322	1.31E+02	2.30E−01	1.25E−02	3.58E+00	2.86E+02
4	450	63494	1.12E+02	8.70E−01	9.29E−10	3.53E+00	3.80E+09
5	52	61149	1.01E+02	1.10E−01	1.69E−02	1.46E+00	8.63E+01
10	52	61149	6.93E+01	1.00E−01	1.54E−02	1.42E+00	9.22E+01
25	58	61196	5.34E+01	1.10E−01	1.34E−02	1.87E+00	1.39E+02
50	EP	—	—	—	—	—	—

Tableau 15.

Préconditionneur : GSC MC (OPT) BLOCS avec ε =1.17E−01, $p_{max} = 20$, $s = 1$							
Blocs	*It*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	95	25128	7.60E+01	8.00E−02	7.81E−03	2.36E+00	3.02E+02
4	99	24409	5.72E+01	1.00E−01	7.21E−03	2.36E+00	3.27E+02
8	104	23554	2.78E+01	1.30E−01	5.64E−03	2.30E+00	4.08E+02
16	126	21251	7.47E+00	1.40E−01	3.79E−03	2.10E+00	5.54E+02
32	168	16849	1.46E+00	1.40E−01	1.95E−03	2.19E+00	1.12E+03
64	192	10151	2.70E−01	1.40E−01	1.81E−03	2.47E+00	1.37E+03
128	200	5907	4.00E−02	1.30E−01	1.95E−03	2.64E+00	1.35E+03

Tableau 16.

Préc. : DIAG + GSC MC (OPT) BLOCS avec ε =1.73E−06, $p_{max} = 20$, $s = 1$							
Blocs	*It*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	76	25135	5.70E+01	7.00E−02	1.04E−02	1.92E+00	1.85E+02
4	85	24416	4.75E+01	8.00E−02	8.93E−03	1.98E+00	2.22E+02
8	96	23572	2.43E+01	1.30E−01	6.47E−03	2.08E+00	3.22E+02
16	125	21277	6.84E+00	1.50E−01	3.97E−03	2.04E+00	5.14E+02
32	168	16876	1.43E+00	1.30E−01	1.95E−03	2.18E+00	1.12E+03
64	192	10187	2.70E−01	1.40E−01	1.81E−03	2.47E+00	1.37E+03
128	200	5914	7.00E−02	1.50E−01	1.95E−03	2.64E+00	1.35E+03

Tableau 17.

Avec les mêmes paramètres p_{max}, ε et s, pour le préconditionneur GSC MC (OPT) (resp. DIAG + GSC MC (OPT)), lorsque le nombre de blocs augmente, le nombre d'itérations augmente aussi, voir tableaux 12 et 16 (resp. 13 et 17).

De plus, en comparant le tableau 16 avec le tableau 17, nous voyons que le premier préconditionnement diagonal (tableau 17) a peu d'effet lorsque le nombre de blocs est suffisamment grand.

4.3 Troisième test : matrice *s1rmt3m1*

Les résultats obtenus pour la matrice *s1rmt3m1* sont présentés dans les tableaux 18–25.

À nouveau pour cette matrice, le préconditionneur DIAG + GSC MC (OPT) avec $p_{max} = 10$ (tableau 21) donne de meilleurs résultats que le préconditionneur GSC INC (tableau 18) : 309 itérations avec $nnz = 60323$ pour la première méthode et 338 itérations avec $nnz = 112505$ pour la seconde.

Préc.	*It.*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
AUCUN	>5489	0.00E+00	1.07E+01	3.80E−01	9.67E+05	2.55E+06
DIAG	926	0.00E+00	1.80E+00	6.87E−06	3.65E+00	5.31E+05
TRIDIAG	785	0.00E+00	1.70E+00	—	—	—
CHO INC	225	1.30E−01	1.33E+00	—	—	—
IAF	329	2.10E−01	1.61E+00	2.34E−05	1.76E+00	7.52E+04
GSC INC	338	1.20E−01	1.66E+00	3.20E−05	2.66E+00	8.33E+04
GSC MC (A)	1380	2.23E+01	6.65E+00	3.22E−06	5.54E+00	1.72E+06

Tableau 18.

Préconditionneur : GSC MC (DIAG)							
p_{max}	*It.*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
1	902	10977	4.50E+00	2.31E+00	6.33E−06	3.57E+00	5.64E+05
5	967	32919	5.98E+00	2.84E+00	6.44E−06	3.76E+00	5.84E+05
10	1052	60324	8.91E+00	3.82E+00	6.35E−06	4.57E+00	7.20E+05
20	1280	115059	1.88E+01	6.15E+00	5.29E−06	5.34E+00	1.01E+06
50	2436	278664	8.23E+01	2.04E+01	4.62E−06	1.59E+01	3.45E+06

Tableau 19.

Préconditionneur : GSC MC (OPT) avec ε =6.92E−05, $s = 1$							
p_{max}	*It.*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
1	826	10976	1.49E+01	2.11E+00	1.01E−05	4.45E+00	4.38E+05
5	1221	32918	5.63E+01	3.62E+00	3.75E−06	5.46E+00	1.45E+06
10	1312	60323	1.41E+02	4.80E+00	4.11E−06	6.33E+00	1.54E+06
20	1010	115058	4.43E+02	4.90E+00	6.36E−06	5.28E+00	8.31E+05
50	663	278663	2.66E+03	5.61E+00	1.38E−05	4.23E+00	3.06E+05

Tableau 20.

Préconditionneur : DIAG + GSC MC (OPT) avec ε =1.25E−09, $s = 1$							
p_{max}	*It.*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
1	596	10976	1.50E+01	1.50E+00	1.11E−05	2.72E+00	2.44E+05
5	372	32918	5.55E+01	1.07E+00	1.75E−05	1.82E+00	1.04E+05
10	309	60323	1.37E+02	1.14E+00	2.55E−05	1.83E+00	7.17E+04
20	243	115058	4.38E+02	1.21E+00	3.72E−05	1.69E+00	4.55E+04
50	178	278663	2.33E+03	1.51E+00	6.56E−05	1.50E+00	2.28E+04

Tableau 21.

Préconditionneur : GSC MC (OPT) with ε =6.92E−05, $p_{max} = 20$							
s	*It.*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	981	115058	2.07E+02	4.97E+00	7.41E−06	5.85E+00	7.89E+05
3	>5489	120526	1.53E+02	2.80E+01	?	8.40E+00	?
4	980	115058	1.11E+02	4.93E+00	6.66E−06	5.20E+00	7.82E+05
5	931	115058	9.29E+01	4.66E+00	7.25E−06	4.81E+00	6.64E+05
10	1396	115067	5.39E+01	6.88E+00	?	4.89E+00	?
20	EP	—	—	—	—	—	—

Tableau 22.

Préconditionneur : DIAG + GSC MC (OPT) avec ε =1.25E−09, $p_{max} = 20$							
s	*It.*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	248	115058	1.97E+02	1.23E+00	3.60E−05	1.72E+00	4.78E+04
3	331	120526	1.47E+02	1.75E+00	3.08E−05	2.61E+00	8.50E+04
4	267	115058	1.04E+02	1.31E+00	3.34E−05	1.80E+00	5.39E+04
5	279	115058	8.89E+01	1.38E+00	3.20E−05	1.83E+00	5.71E+04
10	292	115067	4.93E+01	1.44E+00	2.91E−05	1.80E+00	6.17E+04
20	350	117200	3.70E+01	1.83E+00	2.55E−05	2.24E+00	8.76E+04

Tableau 23.

Préconditionneur : GSC MC (OPT) BLOCS avec ε =6.92E−05, $p_{max} = 50$, $s = 1$							
Blocs	*It.*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	662	276128	2.42E+03	5.54E+00	1.37E−05	4.11E+00	3.01E+05
4	638	271406	1.69E+03	5.32E+00	1.35E−05	3.92E+00	2.91E+05
8	577	262217	6.42E+02	4.62E+00	1.59E−05	3.23E+00	2.03E+05
16	452	243731	2.10E+02	3.47E+00	1.94E−05	2.48E+00	1.28E+05
32	574	207705	6.96E+01	3.94E+00	1.03E−05	2.22E+00	2.17E+05
64	586	174468	3.93E+01	3.56E+00	9.89E−06	2.24E+00	2.27E+05
128	594	111071	1.30E+01	2.66E+00	9.56E−06	2.24E+00	2.35E+05
256	630	59269	2.16E+00	2.30E+00	8.93E−06	2.35E+00	2.63E+05
512	666	31706	3.20E−01	1.89E+00	7.91E−06	2.38E+00	3.01E+05

Tableau 24.

Préc. : DIAG + GSC MC (OPT) BLOCS avec ε =1.25E−09, $p_{max} = 50$, $s = 1$							
Blocs	*It.*	*nnz*	t_{prec}	t_{res}	λ_{min}	λ_{max}	κ
2	203	276128	2.09E+03	1.72E+00	5.86E−05	1.84E+00	3.14E+04
4	235	271406	1.45E+03	1.94E+00	4.83E−05	1.86E+00	3.85E+04
8	286	262220	5.62E+02	2.32E+00	3.63E−05	1.97E+00	5.44E+04
16	398	243736	1.52E+02	3.09E+00	2.00E−05	2.03E+00	1.01E+05
32	573	207762	4.34E+01	3.98E+00	1.03E−05	2.22E+00	2.17E+05
64	586	174664	2.85E+01	3.58E+00	9.89E−06	2.24E+00	2.27E+05
128	594	111121	1.09E+01	2.70E+00	9.56E−06	2.24E+00	2.35E+05
256	630	59269	2.08E+00	2.21E+00	8.93E−06	2.35E+00	2.63E+05
512	667	31706	3.40E−01	1.91E+00	7.91E−06	2.38E+00	3.01E+05

Tableau 25.

Pour les préconditionneurs GSC MC (OPT) BLOCS et DIAG + GSC MC (OPT) BLOCS, lorsque le nombre de blocs augmente, le gain de temps pour le calcul du préconditionneur devient considérable (tableaux 24 et 25).

4.4 Observations

De manière générale, comme le prédisait le théorème 1.1, nous observons que plus la condition de la matrice du système préconditionné est proche de 1, plus le nombre d'itérations est petit.

Les préconditionneurs GSC MC (OPT) et DIAG + GSC MC (OPT) sont intéressants car leurs paramètres les rendent flexibles et permettent de prendre en compte différentes exigences lors de leur construction. Le second semble meilleur et plus stable. Le temps de calcul pour la construction de ces préconditionneurs est élevé ; cependant, ils sont facilement parallélisables et ce temps de calcul peut être réduit par ce biais (voir remarque 3.1 et chapitre 5). Lorsque le pas

de remplissage s est augmenté les préconditionneurs obtenus sont de mauvaise qualité. Le traitement par blocs est le meilleur moyen d'économiser du temps de calcul pour la construction du préconditionneur. Mais, lorsque le nombre de blocs augmente, le nombre d'itérations augmente aussi ; ceci est compensé par le nombre de coefficients non nuls dans le préconditionneur : nnz décroît. Par conséquent le temps utilisé pour la résolution est assez stable.

CHAPITRE 5

Parallélisation

Un programme informatique est dit *parallèle* lorsque son exécution emploie plusieurs processeurs simultanément. L'intérêt de paralléliser un programme est évidemment de gagner du temps. Donnons un exemple simple : le calcul de la somme de 1'000'000 de nombres réels. Avec un programme sériel (*i.e.* un seul processeur), le processeur fait toutes les additions ; avec par exemple dix processeurs, chacun additionnera 100'000 nombres, communiquera son résultat aux autres processeurs et il restera à additionner les dix sommes partielles.

Un algorithme est plus ou moins bien parallélisable. Donnons deux exemples simples.

- Le produit d'une matrice et d'un vecteur $y = Ax$ est bien parallélisable : les composantes du vecteur y se calculent indépendamment les unes des autres. Avec quatre processeurs par exemple et A de taille $100'000 \times 100'000$, les 25'000 premières lignes de A sont mémorisées sur le premier processeur, les 25'000 suivantes sur le deuxième, les 25'000 suivantes sur le troisième, les 25'000 dernières sur le quatrième et le vecteur x sur tous les processeurs ; ensuite chaque processeur calculent simultanément les composantes correspondantes du vecteur y.
- La résolution de $Lx = b$ où L est une matrice $n \times n$ triangulaire inférieure et inversible est mal parallélisable. En effet, le calcul de la solution $x = (x_1, \ldots, x_n)^t \in \mathbb{R}^n$ se fait comme suit : la composante x_1 est déterminée, puis la composante x_2 et ainsi de suite jusqu'à la composante x_n. Le calcul de $x_k = (b_k - (L_{k,1}x_1 + L_{k,2}x_2 + \ldots + L_{k,k-1}x_{k-1}))/L_{kk}$ nécessite la connaissance de $x_1, \ldots, x_{k-1}$. Cette dernière somme peut être parallélisée (lorsque k est grand) et la parallélisation est de plus en plus bénéfique lorsque k augmente, mais inefficace au début.

Pour bien paralléliser un programme, il faut distribuer le "travail" sur les différents processeurs de manière la plus équitable possible afin d'éviter que des

processeurs attendent pendant que les autres terminent leurs calculs. Remarquons qu'un programme n'est en général pas entièrement parallélisable (par exemple la lecture de fichiers sous format séquentiel se fait par un processeur pendant que les autres attendent).

De plus, dans un programme parallèle, il y deux type d'applications : celles de l'utilisateur et celles qui gèrent les *communications* entre les processeurs. Ces dernières sont collectées dans une librairie informatique qui permet d'utiliser un environnement parallèle.

5.1 Speed–up et efficience

Pour évaluer la performance d'un logiciel parallèle, le speed–up et l'efficience sont utilisés [25]. Le *speed–up* est le rapport $S_p = T_1/T_p$, où T_p désigne la durée d'exécution du logiciel sur p processeurs. Le cas idéal est $S_p = p$ (lorsque le nombre de processeurs est multiplié par p, le temps de calcul est divisé par p). L'*efficience* est le taux $E_p = S_p/p$ (c'est le rendement par rapport au cas idéal). Lorsque le test utilisant le moins de processeurs en emploie m, le speed–up est remplacé par $S_p(m) = mT_m/T_p$ de sorte que $S_p(m) = p$ décrive toujours le cas idéal et l'efficience reste $E_p(m) = S_p(m)/p$.

5.2 Algorithme du gradient conjugué préconditionné et parallélisation

Les algorithmes présentés dans les chapitres précédents sont-ils parallélisables ?

La construction des préconditionneurs CHO INC (voir section 2.3 et [38, 46]) et GSC INC (voir section 3.2 et [10]) sont mal parallélisables. De plus, pour le préconditionneur CHO INC, la résolution avec l'algorithme GCP2 (voir section 1.2.2) est de nouveau mal parallélisable (voir le deuxième exemple en début de chapitre).

Par contre, la construction des préconditionneurs IAF et des différents GSC MC sont naturellement parallélisables.

Quant à la partie de résolution, les algorithmes GC et GCP1 sont bien parallélisables : les produits matrice vecteur en sont les opérations essentielles. Par contre, comme mentionné ci–dessus, l'algorithme GCP2 est en général mal parallélisable.

Dans la suite de ce chapitre, l'algorithme du gradient conjugué préconditionné GCP1 avec des préconditionneurs du type GSC MC est étudié sur des machines parallèles. La construction du préconditionneur et la résolution avec l'algorithme GCP1 constituent deux parties distinctes, traitées séparément. Des tests numériques sont effectués et des courbes de speed–up tracées afin d'évaluer la performance de la parallélisation.

5.3 Matériel informatique

Les tests numériques présentés dans ce chapitre ont été effectués sur les machines parallèles CRAY XT3 du CSCS (Swiss National Supercomputing Center). De plus amples informations sont disponibles sur

http ://www.cscs.ch.

Les codes des programmes sont écrits en *fortran* et les librairies MPI (Message Passing Interface, voir [22, 26]) et SHMEM sont utilisées pour gérer l'environnement parallèle. De la documentation peut être trouvée sur

http ://www.cray.com.

5.4 Construction du préconditionneur

Supposons que p processeurs numérotés de 0 à $p-1$ sont utilisés. La matrice Z (voir (3.2)) de taille $n \times n$ du préconditionneur GSC MC (OPT) est distribuée de la manière suivante. La j–ème colonne de Z est calculée par le processeur $j-1 \mod p$. Puisque Z est triangulaire supérieure, cette répartition garantit l'équilibre du remplissage et de la charge de calcul sur chaque processeur. Une distribution continue de Z (*i.e.* les $\lfloor n/p \rfloor (+1)$ sur le processeur 0, les $\lfloor n/p \rfloor (+1)$ suivantes sur le processeur 1, et ainsi de suite) serait un mauvais choix ; en effet, les premiers processeurs auraient fini de calculer leur partie bien avant les derniers. Pour la construction de la k–ème colonne de Z, la sous-matrice principale A_{k-1} d'ordre $k-1$ et les $k-1$ premières composantes de la k–ème colonne de A sont utilisées (voir chapitre 3). Par conséquent, la matrice A est mémorisée entièrement sur chaque processeur afin d'éviter une quantité prohibitive de communications. La copie de la matrice A sur tous les processeurs et un test d'erreur à la fin des calculs constituent toutes les communications nécessaires (réalisées avec la librairie MPI).

Remarque 5.1 *Le format de stockage CSR (ou CSC) (voir section 5.5.1) est utilisé pour la matrice A. Chaque partie locale de Z est mémorisée sous format CSC.*

Pour la version par blocs de ce préconditionneur, la même méthode est appliquée pour chaque bloc successivement : le bloc correspondant dans A est copié sur tous les processeurs et un test d'erreur termine le calcul du bloc considéré. Si M blocs sont considérés et $n = q \cdot M + r$ est la division euclidienne de n par M (q, r sont entiers avec $0 \leqslant r < M$), les r premiers blocs sont d'ordre $q+1$ et les autres d'ordre q.

5.4.1 Évaluation de la performance

Considérons la matrice test SDP *gyro_k*. C'est une matrice d'ordre $n = 17361$ avec $NNZ = 519620$ coefficients non nuls dans sa partie supérieure (ou inférieure). Sa structure est donnée sur la figure 5. Cette matrice peut être obtenue sur

http ://www.cise.ufl.edu/research/sparse/matrices.

Figure 5: ***gyro_k.***

Considérons le préconditionneur couplé DIAG + GSC MC (OPT) avec les paramètres $p_{max} = 10, 20, 50, 100$, ε =9.09E−13 et $s = 1$ (voir chapitre 3). Les remplissages obtenus avec la version standard (*i.e.* préc. avec un bloc) et la version par blocs sont donnés dans le tableau 26 ; le nombre d'éléments non nuls dans Z est noté *nnz*.

p_{max}	*nnz*		
	1 bloc	16 blocs	32 blocs
10	189702	179641	170015
20	360174	327797	297542
50	863688	729669	607914
100	1683529	1309240	993436

Tableau 26.

Le temps de calcul (T_p), le speed–up (S_p) et l'efficience (E_p) pour la construction de ces préconditionneurs sont présentés dans les tableaux 27–30. Le nombre p est le nombre de processeurs utilisés. Les courbes de speed–up et d'efficience pour le calcul du préconditionneur avec différentes valeurs du nombre de blocs sont tracés sur les figures 6–11. Le nombre p est placé sur l'abscisse et le speed–up S_p ou l'efficience E_p sur l'ordonnée.

Pour la version standard, un bloc d'ordre 17361 est considéré. Dans la version avec 16 blocs, le premier bloc est d'ordre 1086 et les 15 suivants sont d'ordre 1085. Dans la version avec 32 blocs, les 17 premiers blocs sont d'ordre 543 et les 15 autres sont d'ordre 542.
Le nombre de colonnes calculées dans un bloc varie de un au maximum d'un processeur à un autre. Par exemple, dans la version avec 32 blocs avec 128 processeurs : puisque $542 = 128 \cdot 4 + 30$, au total, les processeurs 0 à 29 calculent $32 \cdot 5 = 160$ colonnes, le processeur 30 calcule $17 \cdot 5 + 15 \cdot 4 = 145$ colonnes et les processeurs 31 à 127 calculent $32 \cdot 4 = 128$ colonnes. Ceci explique que la version par blocs est moins efficiente que la version avec un seul bloc.

Pour le calcul de chaque bloc, deux communications (copie d'une partie de A et test d'erreur) nécessitent la synchronisation des processeurs. Par conséquent,

l'augmentation du nombre de blocs réduit la qualité de la performance (voir figures 6–11).
En comparant les figures 6–11 et le tableau de remplissage 26, nous observons, qu'en général, lorsque la quantité de calculs augmentent, la qualité de la performance est aussi croissante : la performance est de bonne qualité pour les grandes matrices.

$p_{max} = 10$									
p	1 bloc			16 blocs			32 blocs		
	T_p	S_p	E_p	T_p	S_p	E_p	T_p	S_p	E_p
1	1.31E+03	1.00E+00	1.00E+00	2.44E+02	1.00E+00	1.00E+00	8.92E+01	1.00E+00	1.00E+00
2	6.54E+02	2.00E+00	1.00E+00	1.23E+02	1.99E+00	9.94E−01	4.52E+01	1.97E+00	9.86E−01
4	3.28E+02	3.99E+00	9.96E−01	6.24E+01	3.92E+00	9.79E−01	2.31E+01	3.86E+00	9.65E−01
8	1.64E+02	7.97E+00	9.96E−01	3.23E+01	7.55E+00	9.44E−01	1.23E+01	7.24E+00	9.04E−01
16	8.32E+01	1.57E+01	9.83E−01	1.73E+01	1.41E+01	8.84E−01	6.83E+00	1.31E+01	8.17E−01
32	4.26E+01	3.07E+01	9.59E−01	9.62E+00	2.54E+01	7.93E−01	4.14E+00	2.16E+01	6.74E−01
64	2.16E+01	6.05E+01	9.45E−01	5.53E+00	4.42E+01	6.90E−01	2.70E+00	3.31E+01	5.17E−01
128	1.15E+01	1.13E+02	8.85E−01	3.71E+00	6.59E+01	5.15E−01	1.99E+00	4.48E+01	3.50E−01

Tableau 27.

$p_{max} = 20$									
p	1 bloc			16 blocs			32 blocs		
	T_p	S_p	E_p	T_p	S_p	E_p	T_p	S_p	E_p
1	3.46E+03	1.00E+00	1.00E+00	6.72E+02	1.00E+00	1.00E+00	2.26E+02	1.00E+00	1.00E+00
2	1.74E+03	1.99E+00	9.97E−01	3.38E+02	1.99E+00	9.94E−01	1.15E+02	1.98E+00	9.88E−01
4	8.69E+02	3.98E+00	9.96E−01	1.71E+02	3.93E+00	9.83E−01	5.86E+01	3.87E+00	9.66E−01
8	4.36E+02	7.94E+00	9.93E−01	8.95E+01	7.51E+00	9.39E−01	3.11E+01	7.28E+00	9.10E−01
16	2.21E+02	1.57E+01	9.80E−01	4.72E+01	1.43E+01	8.91E−01	1.70E+01	1.33E+01	8.32E−01
32	1.14E+02	3.03E+01	9.47E−01	2.62E+01	2.57E+01	8.02E−01	9.97E+00	2.27E+01	7.09E−01
64	5.92E+01	5.84E+01	9.13E−01	1.49E+01	4.52E+01	7.06E−01	6.27E+00	3.61E+01	5.64E−01
128	3.04E+01	1.14E+02	8.90E−01	9.53E+00	7.05E+01	5.51E−01	4.35E+00	5.20E+01	4.06E−01

Tableau 28.

$p_{max} = 50$									
p	1 bloc			16 blocs			32 blocs		
	T_p	S_p	E_p	T_p	S_p	E_p	T_p	S_p	E_p
1	1.41E+04	1.00E+00	1.00E+00	2.67E+03	1.00E+00	1.00E+00	8.08E+02	1.00E+00	1.00E+00
2	7.07E+03	1.99E+00	9.96E−01	1.34E+03	1.99E+00	9.95E−01	4.08E+02	1.98E+00	9.89E−01
4	3.55E+03	3.97E+00	9.93E−01	6.77E+02	3.95E+00	9.87E−01	2.08E+02	3.88E+00	9.71E−01
8	1.78E+03	7.92E+00	9.90E−01	3.55E+02	7.52E+00	9.41E−01	1.10E+02	7.33E+00	9.17E−01
16	9.00E+02	1.56E+01	9.78E−01	1.85E+02	1.45E+01	9.05E−01	5.96E+01	1.36E+01	8.47E−01
32	4.60E+02	3.06E+01	9.56E−01	1.02E+02	2.62E+01	8.20E−01	3.43E+01	2.36E+01	7.36E−01
64	2.36E+02	5.96E+01	9.32E−01	5.73E+01	4.67E+01	7.29E−01	2.10E+01	3.84E+01	6.00E−01
128	1.20E+02	1.17E+02	9.13E−01	3.56E+01	7.51E+01	5.86E−01	1.37E+01	5.90E+01	4.61E−01

Tableau 29.

$p_{max} = 100$									
p	1 bloc			16 blocs			32 blocs		
	T_p	S_p	E_p	T_p	S_p	E_p	T_p	S_p	E_p
1				7.78E+03	1.00E+00	1.00E+00	2.13E+03	1.00E+00	1.00E+00
2	2.22E+04	2.00E+00	1.00E+00	3.91E+03	1.99E+00	9.94E−01	1.07E+03	1.98E+00	9.92E−01
4	1.11E+04	4.00E+00	1.00E+00	1.97E+03	3.95E+00	9.88E−01	5.48E+02	3.89E+00	9.73E−01
8	5.56E+03	7.98E+00	9.97E−01	1.02E+03	7.59E+00	9.49E−01	2.90E+02	7.34E+00	9.18E−01
16	2.81E+03	1.58E+01	9.87E−01	5.32E+02	1.46E+01	9.14E−01	1.55E+02	1.37E+01	8.57E−01
32	1.43E+03	3.10E+01	9.68E−01	2.89E+02	2.69E+01	8.40E−01	8.85E+01	2.41E+01	7.53E−01
64	7.40E+02	6.00E+01	9.37E−01	1.63E+02	4.77E+01	7.46E−01	5.34E+01	3.99E+01	6.24E−01
128	3.78E+02	1.17E+02	9.16E−01	1.00E+02	7.77E+01	6.07E−01	3.48E+01	6.13E+01	4.79E−01

Tableau 30.

Speed–up, construction, 1 bloc

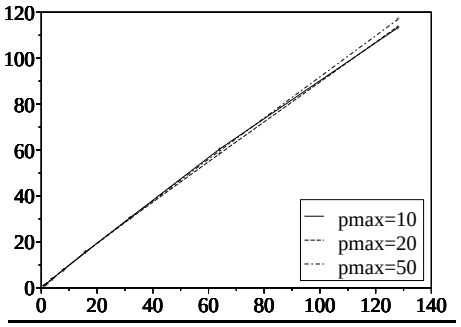

Figure 6.

Efficience, construction, 1 bloc

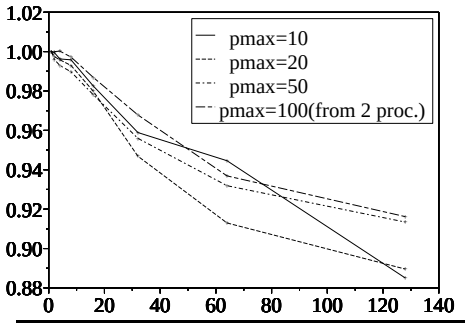

Figure 7.

Speed–up, construction, 16 blocs

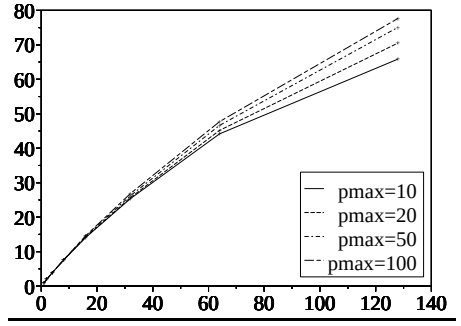

Figure 8.

Efficience, construction, 16 blocs

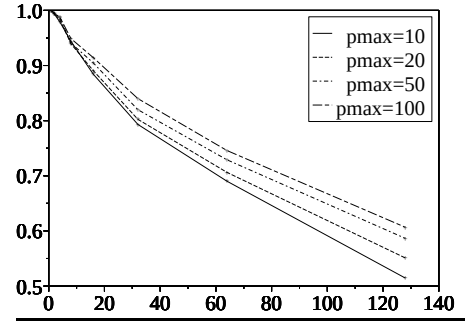

Figure 9.

Speed–up, construction, 32 blocs

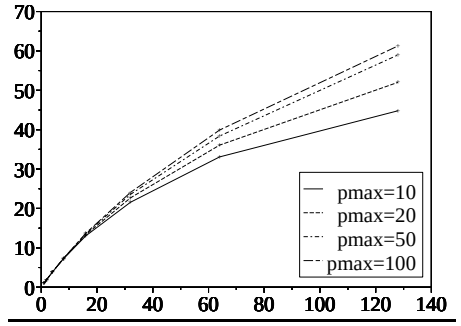

Figure 10.

Efficience, construction, 32 blocs

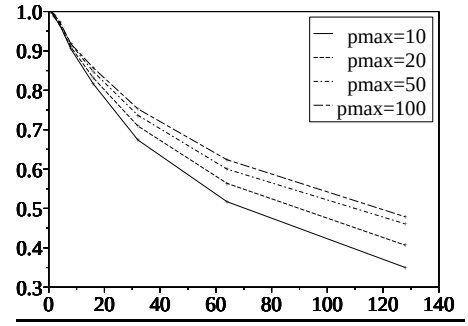

Figure 11.

Notons que sur la figure 7, la courbe pour $p_{max} = 100$ représente les valeurs de $E_p(2)$.

5.5 Résolution avec l'algorithme du gradient conjugué préconditionné

Dans cette section, un préconditionneur $T = D^{-1/2}Z^t$ (du type (DIAG +) GSC MC) est supposé construit et la résolution avec l'algorithme GCP1 parallélisé est étudiée. Pour cela, introduisons brièvement les méthodes de stockage des matrices et vecteurs, quelques procédures de communication et quelques calculs simples.

5.5.1 Stockage

Soit B une matrice $m \times n$ creuse et NNZ le nombre de coefficients non nuls dans B. Cette matrice peut être décrite par les trois vecteurs

1) ab : vecteur de réels de longueur NNZ contenant les coefficients non nuls de B, ligne par ligne ;
2) jab : vecteur d'entiers de longueur NNZ contenant les indices de colonne des coefficients correspondants dans ab ;
3) iab : vecteur d'entiers de longueur $m+1$ contenant les pointeurs du début de chaque ligne dans ab et jab, *i.e.* les valeurs de ab et jab des positions $iab(k)$ à $iab(k+1)-1$ se rapportent à la k–ème ligne. La première composante du vecteur iab est $iab(1) = 1$ et la dernière $iab(m+1) = NNZ + 1$.

Ce format de stockage est appelé format CSR (Compressed Sparse Row).

En échangeant les rôles des lignes et colonnes et en remplaçant m par n dans le format CSR, nous obtenons le format CSC (Compressed Sparse Column).
Pour plus de détails sur les formats de stockage, voir [46] et le logiciel *sparskit* développé par Y. SAAD sur

http ://www-users.cs.umn.edu/~saad,

voir également [7], disponible sur **ftp.netlib.org/templates/templates.ps**.

Supposons maintenant que p processeurs numérotés de 0 à $p-1$ sont utilisés et considérons une matrice creuse M de taille $n \times n$. Dans ce cas, une décomposition continue par lignes de la matrice M peut être considérée : les q_0 premières lignes sont stockées sur les processeur 0, les q_1 suivantes sur le processeur 1, ..., et les q_{p-1} dernières sur le processeur $p-1$, où $q_0 = \ldots = q_{r-1} = q+1$, $q_r = \ldots = q_{p-1} = q$ et $n = q \cdot p + r$ est la division euclidienne de n par p. La partie locale de la matrice M stockée sur le processeur i est notée $M_{loc}^{(i)}$, sa taille est $q_i \times n$ et le format de stockage CSR est utilisé.

De manière similaire, une décomposition continue par colonnes de la matrice M peut être considérée. La partie locale de la matrice M sur le processeur i est alors de taille $n \times q_i$ et le format de stockage CSC est utilisé.

Pour un vecteur u de $\mathbb{R}^n$, une décomposition continue par composantes peut être considérée et le vecteur local $u_{loc}^{(i)} \in \mathbb{R}^{q_i}$ est mémorisé sur le processeur i. Ceci est aussi utilisé pour une matrice diagonale D, puisqu'elle est caractérisée par un vecteur ; dans ce cas, $D_{loc}^{(i)}$ désigne la matrice diagonale locale de taille $q_i \times q_i$ sur le processeur i.

Remarque 5.2 *Lorsque les décompositions continues (par lignes, colonnes ou composantes) ci–dessus sont utilisées, la notation $M_{loc}^{(i)}$ peut être remplacée par M_{loc}, $u_{loc}^{(i)}$ par u_{loc}, $D_{loc}^{(i)}$ par D_{loc} et q_i par n_{loc} ; ainsi, chaque processeur a ses propres matrices locales M_{loc}, D_{loc}, son propre vecteur local u_{loc} et sa propre dimension n_{loc}.*

5.5.2 Quelques procédures de communication

Présentons ici brièvement le concept de trois procédures de communication provenant de la librairie MPI qui seront utilisées dans l'algorithme GCP1 parallélisé. Pour les détails, voir [22, 26].
Les deux premières procédures donnent lieu à des communications globales, c'est–à–dire, **tous les** processeurs appellent la procédure et reçoivent son résultat en retour.

- Procédure MPI_ALLREDUCE. Si, pour chaque i de 0 à $p-1$, v_i est un vecteur de $\mathbb{R}^l$ mémorisé sur le processeur i, cette procédure permet d'obtenir la somme $v = v_0 + \ldots + v_{p-1}$ sur **tous les** processeurs.
- Procédure MPI_ALLGATHERV. Si, pour chaque i de 0 à $p-1$, v_i est un vecteur de $\mathbb{R}^{l_i}$ mémorisé sur le processeur i, cette procédure permet d'obtenir le vecteur v de longueur $l = l_0 + \ldots + l_{p-1}$ construit en rassemblant v_0, $\ldots$, v_{p-1} bout à bout (*i.e.* $v = (v_0, \ldots, v_{p-1})$) sur **tous les** processeurs.

La procédure suivante est une procédure de communication de type *one–side*, **seulement un** processeur l'appelle.

- Procédure MPI_GET. Cette procédure permet au processeur qui l'appelle d'obtenir une copie d'une variable (ou d'un vecteur) mémorisée sur un autre processeur.

Remarque 5.3 *La librairie* SHMEM *peut aussi être utilisée pour les communications. Par exemple, la procédure* SHMEM_GET *correspond à la procédure* MPI_GET *pour cette librairie.*

5.5.3 Quelques calculs simples

Dans cette section, le calcul du produit $u = M \cdot v$ où M est une matrice de taille $n \times n$ et v un vecteur de $\mathbb{R}^n$ est présenté dans plusieurs situations.

Dans les cas où un seul processeur est utilisé, l'algorithme 7 (resp. 8) donne le calcul de $u = M \cdot v$ avec la matrice M mémorisée sous format CSR (resp. CSC) à l'aide des trois vecteurs am, jam et iam (voir section 5.5.1).

$u = Mv$, M en format CSR
Pour i de 1 à n, faire :
$\quad w = 0$
$\quad$ Pour j de $iam(i)$ à $iam(i+1)-1$, faire :
$\quad\quad w = w + am(j) \cdot v(jam(j))$
$\quad$ Fin pour j
$\quad u(i) = w$
Fin pour i

Algorithme 7.

$u = Mv$, M en format CSC

$u(i) = 0,\ i = 1, \ldots, n$
Pour i de 1 à n, faire :
  $w = v(i)$
  Pour j de $iam(i)$ à $iam(i+1) - 1$, faire :
    $u(jam(j)) = u(jam(j)) + am(j) \cdot w$
  Fin pour j
Fin pour i

Algorithme 8.

Supposons maintenant que p processeurs sont utilisés et que la matrice M et le vecteur v sont mémorisés selon la décomposition continue de la section 5.5.1. Les algorithmes 9, 10 et 11 effectuent le produit $u = M \cdot v$ dans trois situations.

Algorithme 9. Décomposition par lignes, M_{loc} en format CSR, communications globales

1. Utiliser MPI_ALLGATHERV pour obtenir le vecteur v sur tous les processeurs à partir des vecteurs locaux v_{loc}.
2. Calculer $u_{loc} = M_{loc} \cdot v$ sur chaque processeur en utilisant l'algorithme 7.

Algorithme 10. Décomposition par lignes, M_{loc} en format CSR, communications de type one–side

1. En utilisant la procédure MPI_GET, chaque processeur copie uniquement les composantes nécessaires des vecteurs v_{loc} mémorisés sur les autres processeurs afin d'effectuer le produit local ci–dessous.
2. Calculer $u_{loc} = M_{loc} \cdot v$ sur chaque processeur en utilisant l'algorithme 7.

Algorithme 11. Décomposition par colonnes, M_{loc} en format CSC

1. Calculer $w = M_{loc} \cdot v_{loc} \in \mathbb{R}^n$ sur chaque processeur en utilisant l'algorithme 8.
2. Utiliser la procédure MPI_ALLREDUCE pour obtenir le vecteur u sur tous les processeurs à partir des p vecteurs locaux w de chaque processeur.

Notons qu'à la fin des algorithmes 9 et 10, chaque processeur a seulement la partie locale u_{loc} du vecteur u dans sa mémoire, tandis qu'à la fin de l'algorithme 11, le vecteur u est entièrement mémorisé sur tous les processeurs.

Remarque 5.4 *Si l'algorithme 7, 9 ou 10 (resp. 8 ou 11) est employé pour une matrice M mémorisée sous format CSC (resp. CSR), le vecteur $u = M^t \cdot v$ est obtenu.*

5.5.4 L'algorithme du gradient conjugué préconditionné pas à pas

Puisque la matrice du système A est symétrique, les formats CSR et CSC sont exactement les mêmes pour cette matrice. D'après sa construction par colonnes, la matrice Z du préconditionneur $T = D^{-1/2}Z^t$ est mémorisée sous format CSC. Supposons que la décomposition continue de la section 5.5.1 est utilisée

pour mémoriser la matrice A (décomposition par lignes, format CSR), la matrice Z (décomposition par colonnes, format CSC) et la matrice diagonale D^{-1} (décomposition par composantes sur la diagonale).

Reprenons l'algorithme GCP1 qui peut être récrit comme suit :

Algorithme GCP1 : Avec T régulière

Soit $x_0 \in \mathbb{R}^n$ donné
Calculer $r_0 = b - Ax_0$, $s_0 = T^t \cdot Tr_0$ et poser $d_0 = s_0$
Si $\|r_0\| / \|b\| < tol$: STOP
Calculer $\tilde{\alpha}_0 = (\, r_0 \,|\, s_0 \,) / (\, d_0 \,|\, Ad_0 \,)$, $x_1 = x_0 + \tilde{\alpha}_0 d_0$ et $r_1 = r_0 - \tilde{\alpha}_0 Ad_0$
Si $\|r_1\| / \|b\| < tol$: STOP
Pour $k \geqslant 1$, faire :
    $s_k = T^t \cdot Tr_k$
    $\tilde{\beta}_{k-1} = (\, r_k \,|\, s_k \,) / (\, r_{k-1} \,|\, s_{k-1} \,)$
    $d_k = s_k + \tilde{\beta}_{k-1} d_{k-1}$
    $\tilde{\alpha}_k = (\, r_k \,|\, s_k \,) / (\, d_k \,|\, Ad_k \,)$
    $x_{k+1} = x_k + \tilde{\alpha}_k d_k$
    $r_{k+1} = r_k - \tilde{\alpha}_k Ad_k$
    Si $\|r_{k+1}\| / \|b\| < tol$: STOP
Fin pour k

Algorithme 12.

Donnons en détails les étapes d'une itération (un passage dans la boucle) avec $T = D^{-1/2} Z^t$ (et r, s, $t_{old} = (\, r \,|\, s \,)$, d, x connus de l'itération précédente) :

$$\tilde{s} = Z^t r \tag{5.1a}$$
$$\hat{s} = D^{-1} \tilde{s} \tag{5.1b}$$
$$s = Z \hat{s} \tag{5.1c}$$
$$t = (\, r \,|\, s \,) \tag{5.1d}$$
$$\tilde{\beta} = t / t_{old} \tag{5.1e}$$
$$t_{old} = t \tag{5.1f}$$
$$d = s + \tilde{\beta} d \tag{5.1g}$$
$$q = Ad \tag{5.1h}$$
$$\tilde{\alpha} = t / (\, d \,|\, q \,) \tag{5.1i}$$
$$x = x + \tilde{\alpha} d \tag{5.1j}$$
$$r = r - \tilde{\alpha} q \tag{5.1k}$$
$$\text{Si } \|r\|^2 / \|b\|^2 < tol^2 \text{ : sortir de la boucle.} \tag{5.1l}$$

Supposons qu'au début, le vecteur r est distribué sur les processeurs selon la décomposition continue considérée. L'étape (5.1a) est effectuée en utilisant l'algorithme 9 ou l'algorithme 10. Ensuite, chaque processeur calcule

$$\hat{s}_{loc} = D_{loc}^{-1} \tilde{s}_{loc}$$

et l'étape (5.1c) est réalisée en utilisant l'algorithme 11. Le vecteur s est alors entièrement mémorisé sur tous les processeurs.

Pour l'étape (5.1d), le produit scalaire $(\, r_{loc} \,|\, s_{loc} \,)$ est calculé localement sur chaque processeur et la procédure MPI_ALLREDUCE est employée pour obtenir la somme $(\, r \,|\, s \,)$.
Les étapes (5.1e) à (5.1g) sont effectuées globalement par chaque processeur. L'étape (5.1h) est locale, *i.e.* chaque processeur calcule

$$q_{loc} = A_{loc} d.$$

Le produit scalaire de l'étape (5.1i) est obtenu comme à l'étape (5.1d). Ensuite, les étapes (5.1j) et (5.1k) sont locales, les calculs suivants sont effectués sur chaque processeur :

$$\begin{aligned} x_{loc} &= x_{loc} + \tilde{\alpha} d_{loc}, \\ r_{loc} &= r_{loc} - \tilde{\alpha} q_{loc}. \end{aligned}$$

Pour le test (5.1l), le produit scalaire $(\, r \,|\, r \,)$ est calculé comme aux étapes (5.1d) et (5.1i).

Remarque 5.5 *Considérons la version par blocs du préconditionneur GSC MC (OPT) avec M blocs. Supposons que $p = M$ processeurs sont utilisés pour l'algorithme GCP1. Dans ce cas, il y a exactement un bloc sur chaque processeur et les trois premières étapes ((5.1a) à (5.1c)) peuvent être effectuées localement sans communication. Si ici Z_{loc} désigne la matrice bloc de taille nloc × nloc (propre à chaque processeur) sur la diagonale de Z, en utilisant les algorithmes 7 et 8 pour la première et la troisième étapes respectivement, nous obtenons*

$$\begin{aligned} \tilde{s}_{loc} &= Z_{loc}^t r_{loc}, \\ \hat{s} &= D_{loc}^{-1} \tilde{s}, \\ s_{loc} &= Z_{loc} \hat{s}_{loc}. \end{aligned}$$

Le vecteur s n'est alors mémorisé que partiellement sur les processeurs. Par conséquent, l'étape (5.1g) est réalisée localement,

$$d_{loc} = s_{loc} + \tilde{\beta} d_{loc},$$

et l'étape (5.1h) est effectuée en utilisant l'algorithme 9 (ou 10).

5.5.5 Évaluation de la performance

La matrice test SDP et les préconditionneurs de la section 5.4.1 sont considérés.

Le second membre $b = (1, \ldots, 1)^t$ et le point de départ $x_0 = 0$ sont choisis. La tolérance de convergence *tol* est fixée à 10^{-8}. Pour chaque test, le nombre d'itérations nécessaires est donné. Puisque ce nombre varie un peu, le speed–up considéré est $S_p = (T_1/It_1)/(T_p/It_p)$, où It_p est le nombre d'itérations avec p processeurs. Ainsi, les comparaisons sont faites par itération. L'efficience ne sera pas donnée ici afin de réduire les tableaux.

L'algorithme de résolution est appelé *GCP1 global* si l'algorithme 9 est utilisé pour la première étape (5.1a) (voir section précédente) et *GCP1 one–side* si cette étape est effectuée en utilisant l'algorithme 10.

Remarque 5.6 *Les communications de type one–side sont effectuées avec la procédure* SHMEM_GET *(voir remarque 5.3). Effectivement, par expérience, la librairie* SHMEM *est plus efficiente que la librairie* MPI *pour ce type de communication.*

Les résultats des tests numériques sont donnés dans les tableaux 31–38.

GCP1 global, $p_{max} = 10$

p	1 bloc			16 blocs			32 blocs		
	It_p	T_p	S_p	It_p	T_p	S_p	It_p	T_p	S_p
1	4328	3.06E+01	1.00E+00	5561	3.88E+01	1.00E+00	6315	4.37E+01	1.00E+00
2	4468	1.85E+01	1.70E+00	5596	2.28E+01	1.71E+00	6316	2.54E+01	1.72E+00
4	4400	1.23E+01	2.54E+00	5528	1.44E+01	2.67E+00	6328	1.75E+01	2.51E+00
8	4294	9.24E+00	3.29E+00	5678	1.27E+01	3.13E+00	6331	1.42E+01	3.08E+00
16	4470	8.91E+00	3.55E+00	5691	1.13E+01	3.50E+00	6328	1.28E+01	3.43E+00
32	4453	8.38E+00	3.76E+00	5737	1.07E+01	3.72E+00	6415	1.20E+01	3.71E+00
64	4288	7.55E+00	4.02E+00	5680	1.07E+01	3.71E+00	6276	1.17E+01	3.72E+00

Tableau 31.

GCP1 one–side, $p_{max} = 10$

p	1 bloc			16 blocs			32 blocs		
	It_p	T_p	S_p	It_p	T_p	S_p	It_p	T_p	S_p
1	4328	3.06E+01	1.00E+00	5561	3.90E+01	1.00E+00	6315	4.36E+01	1.00E+00
2	4468	2.08E+01	1.52E+00	5596	2.08E+01	1.88E+00	6316	2.61E+01	1.67E+00
4	4400	2.65E+01	1.18E+00	5528	1.39E+01	2.79E+00	6328	1.78E+01	2.46E+00
8	4294	5.00E+01	6.08E−01	5678	9.88E+00	4.03E+00	6331	1.38E+01	3.17E+00
16	4470	4.88E+01	6.48E−01	5691	9.00E+00	4.44E+00	6328	1.20E+01	3.64E+00
32	4453	5.69E+01	5.54E−01	5737	3.10E+01	1.30E+00	6415	9.21E+00	4.81E+00
64	4288	4.76E+01	6.37E−01	5680	3.61E+01	1.10E+00	6276	2.43E+01	1.78E+00

Tableau 32.

GCP1 global, $p_{max} = 20$

p	1 bloc			16 blocs			32 blocs		
	It_p	T_p	S_p	It_p	T_p	S_p	It_p	T_p	S_p
1	3546	3.02E+01	1.00E+00	5208	4.29E+01	1.00E+00	6008	4.78E+01	1.00E+00
2	3665	1.81E+01	1.72E+00	5217	2.41E+01	1.78E+00	5974	2.74E+01	1.73E+00
4	3660	1.16E+01	2.70E+00	5152	1.55E+01	2.74E+00	6201	1.82E+01	2.71E+00
8	3671	8.46E+00	3.70E+00	5328	1.25E+01	3.52E+00	6237	1.39E+01	3.58E+00
16	3689	7.80E+00	4.03E+00	5162	1.05E+01	4.05E+00	5911	1.18E+01	3.98E+00
32	3688	7.08E+00	4.44E+00	5017	9.59E+00	4.31E+00	5909	1.18E+01	4.00E+00
64	3649	6.66E+00	4.67E+00	5326	1.02E+01	4.32E+00	6163	1.10E+01	4.48E+00

Tableau 33.

GCP1 one–side, $p_{max} = 20$

p	1 bloc			16 blocs			32 blocs		
	It_p	T_p	S_p	It_p	T_p	S_p	It_p	T_p	S_p
1	3546	3.04E+01	1.00E+00	5208	4.31E+01	1.00E+00	6008	4.82E+01	1.00E+00
2	3665	2.17E+01	1.45E+00	5217	2.30E+01	1.88E+00	5974	2.93E+01	1.64E+00
4	3660	3.02E+01	1.04E+00	5152	1.42E+01	3.01E+00	6201	1.98E+01	2.51E+00
8	3671	5.51E+01	5.72E−01	5328	1.06E+01	4.16E+00	6237	1.58E+01	3.18E+00
16	3689	5.23E+01	6.05E−01	5162	8.57E+00	4.99E+00	5911	1.27E+01	3.72E+00
32	3688	6.50E+01	4.87E−01	5017	3.03E+01	1.37E+00	5909	8.66E+00	5.47E+00
64	3649	5.78E+01	5.42E−01	5326	3.94E+01	1.12E+00	6163	2.35E+01	2.10E+00

Tableau 34.

GCP1 global, $p_{max} = 50$

p	1 bloc			16 blocs			32 blocs		
	It_p	T_p	S_p	It_p	T_p	S_p	It_p	T_p	S_p
1	2535	3.28E+01	1.00E+00	4815	5.66E+01	1.00E+00	5838	6.27E+01	1.00E+00
2	2532	1.83E+01	1.80E+00	4816	3.17E+01	1.79E+00	5881	3.54E+01	1.78E+00
4	2532	1.09E+01	3.01E+00	4836	1.97E+01	2.88E+00	5800	2.27E+01	2.74E+00
8	2529	7.91E+00	4.14E+00	4823	1.34E+01	4.24E+00	5835	1.57E+01	4.00E+00
16	2448	5.65E+00	5.61E+00	4738	1.07E+01	5.21E+00	5669	1.28E+01	4.74E+00
32	2549	5.26E+00	6.28E+00	4754	9.62E+00	5.81E+00	5799	1.17E+01	5.31E+00
64	2522	4.74E+00	6.89E+00	4739	8.93E+00	6.24E+00	5895	1.09E+01	5.83E+00

Tableau 35.

GCP1 one–side, $p_{max} = 50$

p	1 bloc			16 blocs			32 blocs		
	It_p	T_p	S_p	It_p	T_p	S_p	It_p	T_p	S_p
1	2535	3.31E+01	1.00E+00	4815	5.69E+01	1.00E+00	5838	6.28E+01	1.00E+00
2	2532	2.38E+01	1.39E+00	4816	3.07E+01	1.85E+00	5881	3.90E+01	1.62E+00
4	2532	3.41E+01	9.69E−01	4836	1.85E+01	3.09E+00	5800	2.62E+01	2.38E+00
8	2529	5.14E+01	6.42E−01	4823	1.26E+01	4.54E+00	5835	1.82E+01	3.45E+00
16	2448	5.31E+01	6.02E−01	4738	8.70E+00	6.43E+00	5669	1.50E+01	4.06E+00
32	2549	6.21E+01	5.35E−01	4754	3.26E+01	1.72E+00	5799	9.21E+00	6.78E+00
64	2522	6.08E+01	5.42E−01	4739	4.21E+01	1.33E+00	5895	2.43E+01	2.61E+00

Tableau 36.

GCP1 global, $p_{max} = 100$

p	1 bloc			16 blocs			32 blocs		
	It_p	T_p	S_p	It_p	T_p	S_p	It_p	T_p	S_p
1	1770	3.64E+01	1.00E+00	4464	7.57E+01	1.00E+00	5487	7.77E+01	1.00E+00
2	1771	1.95E+01	1.87E+00	4457	4.12E+01	1.83E+00	5700	4.44E+01	1.82E+00
4	1818	1.17E+01	3.20E+00	4487	2.48E+01	3.06E+00	5511	2.74E+01	2.85E+00
8	1769	7.02E+00	5.19E+00	4523	1.63E+01	4.70E+00	5702	1.94E+01	4.17E+00
16	1818	5.37E+00	6.97E+00	4503	1.24E+01	6.15E+00	5690	1.42E+01	5.67E+00
32	1818	4.54E+00	8.24E+00	4447	9.97E+00	7.56E+00	5699	1.26E+01	6.41E+00
64	1827	3.91E+00	9.62E+00	4355	9.11E+00	8.11E+00	5737	1.14E+01	7.12E+00

Tableau 37.

GCP1 one–side, $p_{max} = 100$

p	1 bloc			16 blocs			32 blocs		
	It_p	T_p	S_p	It_p	T_p	S_p	It_p	T_p	S_p
1	1770	3.65E+01	1.00E+00	4464	7.62E+01	1.00E+00	5487	7.81E+01	1.00E+00
2	1771	2.60E+01	1.41E+00	4457	4.04E+01	1.88E+00	5700	5.03E+01	1.62E+00
4	1818	3.63E+01	1.03E+00	4487	2.39E+01	3.21E+00	5511	3.19E+01	2.46E+00
8	1769	4.57E+01	7.98E−01	4523	1.51E+01	5.12E+00	5702	2.33E+01	3.49E+00
16	1818	5.48E+01	6.84E−01	4503	1.03E+01	7.46E+00	5690	1.89E+01	4.28E+00
32	1818	6.01E+01	6.24E−01	4447	3.02E+01	2.51E+00	5699	9.94E+00	8.17E+00
64	1827	5.72E+01	6.59E−01	4355	4.37E+01	1.70E+00	5737	2.46E+01	3.32E+00

Tableau 38.

Comme dans la section 5.4.1, sur les graphes des figures 12–19, le nombre p de processeurs utilisés est représenté en abscisse et le speed–up S_p en ordonnée.

Speed–up, GCP1 global, 1 bloc

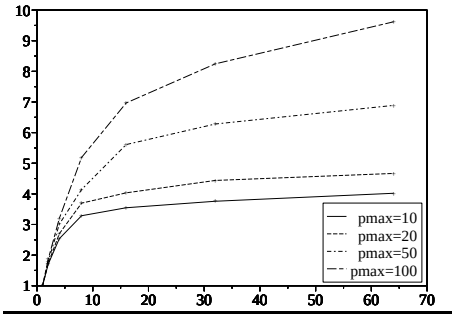

Figure 12.

Speed–up, GCP1 one–side, 1 bloc

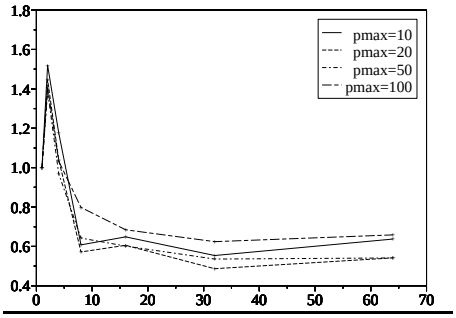

Figure 13.

Speed–up, GCP1 global, 16 blocs

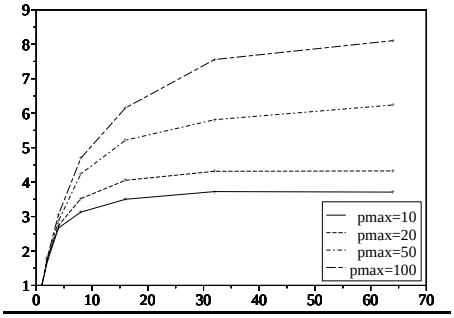

Figure 14.

Speed–up, GCP1 one–side, 16 blocs

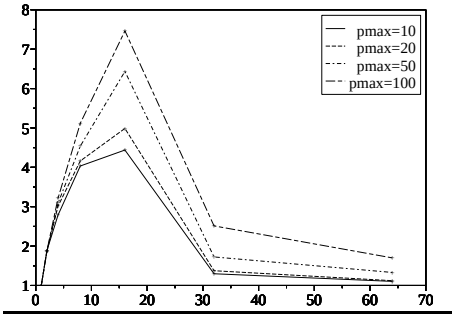

Figure 15.

Speed–up, GCP1 global, 32 blocs

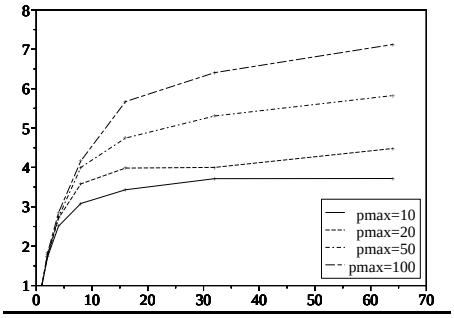

Figure 16.

Speed–up, GCP1 one–side, 32 blocs

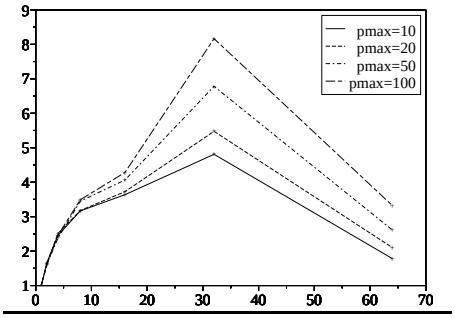

Figure 17.

Les figures 18 et 19 permettent de comparer les algorithmes GCP1 global (gathering) et GCP1 one–side (one–sided) dans le cas du préconditionneur par blocs avec $p_{max} = 100$.

Speed–up, GCP1, 16 blocs, $p_{max} = 100$

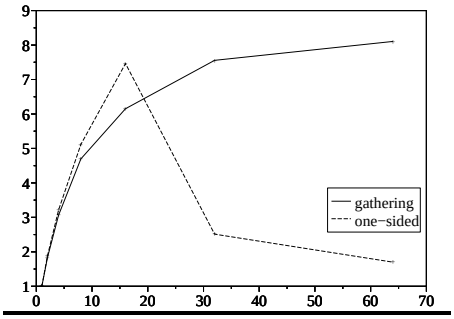

Figure 18.

Speed–up, GCP1, 32 blocs, $p_{max} = 100$

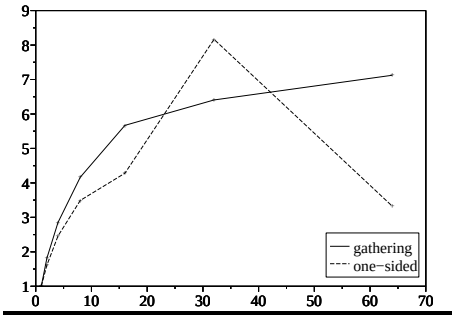

Figure 19.

Les communications de type one–side rendent l'algorithme GCP1 plus performant que les communications globales seulement lorsque le nombre de blocs et le

nombre de processeurs utilisés sont les mêmes. Dans cette situation, aucune communication n'est nécessaire pour l'étape (5.1a) dans GCP1 avec l'algorithme 10. D'après la remarque 5.5, l'appel de la procédure MPI_ALLREDUCE à l'étape (5.1c) peut être remplacée par un appel de la procédure MPI_ALLGATHERV à l'étape (5.1h). Ainsi, le temps de calcul peut être réduit comme le montrent les tableaux 39 et 40. (Le nombre d'itérations ne varie pas selon le type de communication utilisé.)

16 blocs et processeurs				
Algorithme	$p_{max} = 10$	$p_{max} = 20$	$p_{max} = 50$	$p_{max} = 100$
global	1.13E+01	1.05E+01	1.07E+01	1.24E+01
one–side	9.00E+00	8.57E+00	8.70E+00	1.03E+01
selon remarque 5.5	5.82E+00	5.59E+00	5.95E+00	7.85E+00

Tableau 39: Temps de calcul (T_{16}).

32 blocs et processeurs				
Algorithme	$p_{max} = 10$	$p_{max} = 20$	$p_{max} = 50$	$p_{max} = 100$
global	1.20E+01	1.18E+01	1.17E+01	1.26E+01
one–side	9.21E+00	8.66E+00	9.21E+00	9.94E+00
selon remarque 5.5	4.75E+00	4.51E+00	5.03E+00	6.11E+00

Tableau 40: Temps de calcul (T_{32}).

Comme pour la construction du préconditionneur (section 5.4.1), lorsque la quantité de calcul augmente, la qualité de la performance croît aussi. Ici, la performance est plutôt médiocre. Cependant, en connaissant le nombre de processeurs utilisés, le traitement par blocs pour la classe de préconditionneurs GSC MC (OPT) permet d'obtenir un algorithme du gradient conjugué efficace (suivant la remarque 5.5, voir tableaux 39 et 40).

Deuxième partie

Simulation numérique d'écoulements océaniques tridimensionnels

CHAPITRE 6

Description du problème

Le but de cette deuxième partie est de simuler des écoulements océaniques en trois dimensions. Dans ce chapitre, les équations à résoudre sont données et les différents paramètres du problème sont décrits.

Le mouvement d'un fluide est décrit par les équations de Navier–Stokes, voir [23, 37, 47, 35, 20]. Nous considérons ici le cas d'un fluide incompressible anisotrope à densité constante dans un bassin peu profond, soumis à la force de Coriolis, à l'attraction de la Terre et aux tractions des vents à la surface. La résolution des équations du mouvement dans un bassin peu profond explique le caractère anisotrope du fluide, voir section 6.6 et [15, 40].

6.1 Bassin peu profond

Les océans sont des bassins *peu profonds*, c'est–à–dire que leur diamètre horizontal d est beaucoup plus grand que leur profondeur h_0, le quotient $\varepsilon = h_0/d$, appelé *rapport d'aspect*, est "petit" (de l'ordre du millième). Les lacs, les mers ou encore les flaques d'eau sont d'autres exemples de tels bassins. Pour l'océan Atlantique nord, le diamètre horizontal est $d \approx 5000$ km et la profondeur $h_0 \approx 5$ km, ce qui donne un rapport d'aspect $\varepsilon \approx 0.001$. Le rôle du rapport d'aspect dans le calcul des courants est décrit dans la section 6.6.

6.2 Les équations de Navier–Stokes

Considérons un bassin peu profond. C'est un domaine (*i.e.* ouvert connexe) borné de $\mathbb{R}^3$,

$$W = \{\xi = (\xi_1, \xi_2, \xi_3) \in \mathbb{R}^3 \ : \ (\xi_1, \xi_2) \in G_s, \ -h(\xi_1, \xi_2) < \xi_3 < 0\},$$

où G_s (domaine de $\mathbb{R}^2$) est la surface du bassin et $h : G_s \to \mathbb{R}^+$ donne la profondeur au point de G_s. Notons encore

$$G_b = \partial W \setminus G_s = \{\xi = (\xi_1, \xi_2, \xi_3) \in \mathbb{R}^3 \ : \ (\xi_1, \xi_2) \in G_s, \ \xi_3 = -h(\xi_1, \xi_2)\}$$

le fond du bassin.

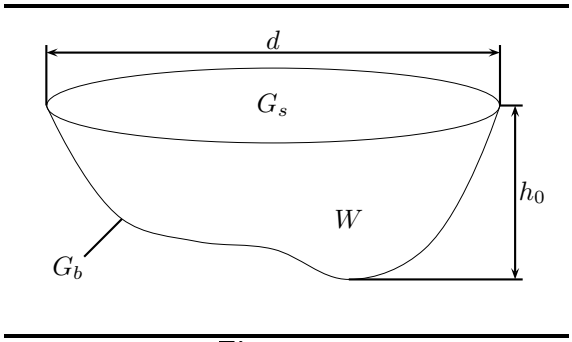

Figure 20.

Décrivons maintenant les grandeurs physiques intervenant dans le problème. La vitesse du fluide au point $\xi \in W$ et au temps $t \geqslant 0$ est notée

$$u = u(\xi, t) = (u_1(\xi, t), u_2(\xi, t), u_3(\xi, t))^t.$$

La densité du fluide est notée ρ et supposée constante ($\rho = 10^3 \ Kg \cdot m^{-3}$ pour l'eau). Le vecteur de viscosité dynamique (resp. cinématique) turbulente (voir section 6.6) est noté $\eta = (\eta_1, \eta_2, \eta_3)$ (resp. $\nu = (\nu_1, \nu_2, \nu_3)$). Les vecteurs (constants) η et ν vérifient la relation $\nu_i = \eta_i \rho^{-1}$, $i = 1, 2, 3$. À la surface du bassin G_s, le fluide est soumis à des tractions $\tilde{\tau}_1 = \tilde{\tau}_1(\xi, t)$ et $\tilde{\tau}_2 = \tilde{\tau}_2(\xi, t)$ induites par les vents, dans les directions ξ_1 et ξ_2 respectivement. Posons $\tau_i = \tilde{\tau}_i \rho^{-1}$, $i = 1, 2$. De plus, dans le bassin W, un champ de force, décrit par l'accélération $f = (f_1, f_2, f_3) = f(\xi, t)$, agit sur le fluide. La pression totale exercée sur le fluide est composée de la pression hydrostatique $\rho g \xi_3$, où $g > 0$ est l'accélération de la pesanteur (scalaire), et de la pression hydrodynamique P. Notons

$$p = p(\xi, t) = g\xi_3 + \frac{1}{\rho} P(\xi, t).$$

Les grandeurs introduites sont données avec unité dans le tableau 42.

Notation	Description	Unité (SI)		
$u = (u_1, u_2, u_3)$	Vitesse	$[u_i]$	$=$	$m \cdot s^{-1}$
ρ	Densité	$[\rho]$	$=$	$Kg \cdot m^{-3}$
$\eta = (\eta_1, \eta_2, \eta_3)$	Vecteur de viscosité dynamique turbulente	$[\eta_i]$	$=$	$Kg \cdot m^{-1} \cdot s^{-1}$
$\nu = (\nu_1, \nu_2, \nu_3)$	Vecteur de viscosité cinématique turbulente ($\nu_i = \eta_i \rho^{-1}$)	$[\nu_i]$	$=$	$m^2 \cdot s^{-1}$
$\tilde{\tau} = (\tilde{\tau}_1, \tilde{\tau}_2)$	Traction à la surface	$[\tilde{\tau}_i]$	$=$	$Kg \cdot m^{-1} \cdot s^{-2}$
$\tau = (\tau_1, \tau_2)$	$\tau_i = \tilde{\tau}_i \rho^{-1}$	$[\tau_i]$	$=$	$m^2 \cdot s^{-2}$
$f = (f_1, f_2, f_3)$	Champ de force	$[f_i]$	$=$	$m \cdot s^{-2}$
g	Accélération de la pesanteur	$[g]$	$=$	$m \cdot s^{-2}$
p	ρp : Pression totale	$[p]$	$=$	$m^2 \cdot s^{-2}$

Tableau 42.

Les tractions s'expriment aussi en Pascal, $[\tilde{\tau}_i] = Pa = N \cdot m^{-2} = Kg \cdot m^{-1} \cdot s^{-2}$ et la viscosité cinématique en Poiseuille, $[\tilde{\nu}_i] = Pl = Pa \cdot s = N \cdot m^{-2} \cdot s = Kg \cdot m^{-1} \cdot s^{-1}$.

Remarque 6.1 *Les tractions $\tilde{\tau}_1, \tilde{\tau}_2$ sont généralement calculées à partir des vitesses horizontales v_1, v_2 des vents en surface par une formule du type*

$$\tilde{\tau}_i = C \cdot (v_1^2 + v_2^2)^{1/2} \cdot v_i, \ i = 1, 2,$$

C étant un coefficient de traînée estimé expérimentalement [40, 18].

Les variables u et p sont déterminées par les équations de Navier–Stokes [23, 37, 47, 35, 20] pour un fluide incompressible anisotrope à densité constante

$$\frac{\partial u}{\partial t} + (\, u \,|\, \nabla \,)\, u - \Delta_\nu u + \nabla p = f \qquad \text{dans } W, \tag{6.1a}$$

$$\operatorname{div} u = 0 \qquad \text{dans } W, \tag{6.1b}$$

$$u = 0 \qquad \text{sur } G_b, \tag{6.1c}$$

$$\nu_3 \frac{\partial u_1}{\partial \xi_3} = \tau_1, \ \nu_3 \frac{\partial u_2}{\partial \xi_3} = \tau_2, \ u_3 = 0 \qquad \text{sur } G_s, \tag{6.1d}$$

où $\nabla = (\frac{\partial}{\partial \xi_1}, \frac{\partial}{\partial \xi_2}, \frac{\partial}{\partial \xi_3})^t$ désigne le gradient pour les coordonnées spatiales,

$$(\, u \,|\, \nabla \,)\, u = ((\, u \,|\, \nabla u_1 \,), (\, u \,|\, \nabla u_2 \,), (\, u \,|\, \nabla u_3 \,))^t$$

et

$$\Delta_\nu u = (\Delta_\nu u_1, \Delta_\nu u_2, \Delta_\nu u_3)^t, \ \Delta_\nu u_i = \sum_{j=1}^{3} \nu_j \frac{\partial^2 u_i}{\partial \xi_j^2}.$$

Le terme $-\Delta_\nu u$ traduit les effets de viscosité et l'équation $\operatorname{div} u = 0$ est la condition d'incompressibilité. Le terme non linéaire $(\, u \,|\, \nabla \,)\, u$ est le propos de la remarque 6.3.

Le champ de force agissant dans le bassin est dû à la rotation de la Terre (force de Coriolis),

$$f = -2\omega \wedge u,$$

où ω est la vitesse angulaire de la Terre exprimée en radian par seconde. Ainsi, l'équation (6.1a) devient

$$\frac{\partial u}{\partial t} - \Delta_\nu u = -(\, u \,|\, \nabla \,)\, u - 2\omega \wedge u - \nabla p.$$

Généralisons la condition de bord sur le fond du bassin (6.1c). Pour $i = 1, 2, 3$, considérons une partie $G_i \subset G_b$ telle que $G_b \setminus G_i$ soit contenu dans un plan parallèle à l'axe ξ_i ; la composante u_i de la vitesse peut être laissée libre sur $G_b \setminus G_i$ sans qu'il y ait de flux sortant ou entrant dans le bassin W et nous pouvons ainsi remplacer la condition (6.1c) par

$$u_1 = 0 \text{ sur } G_1, \ u_2 = 0 \text{ sur } G_2, \ u_3 = 0 \text{ sur } G_3 \ ;$$

en particulier, avec encore $u_3 = 0$ sur G_s, le vecteur u vérifie

$$(u \,|\, n_W) = 0 \text{ sur } \partial W, \tag{6.2}$$

où n_W est le vecteur normal unité sortant de W.

Remarque 6.2 *La vitesse n'est pas fixée à zéro sur les parties de G_b qui sont en fait des bords virtuels du bassin, lorsque W est une partie d'un bassin plus grand avec lequel il n'y a pas d'échange de fluide. Par exemple, nous pouvons supposer que pour l'océan Atlantique nord, le bord vertical situé à l'équateur est un tel bord virtuel. En effet la force de Coriolis est parallèle à ce bord (voir (6.4)) et les vents sont principalement orientés d'est en ouest (alizés).*

Ainsi, les équations de Navier–Stokes (6.1) deviennent

$$\frac{\partial u}{\partial t} - \Delta_\nu u = -(u \,|\, \nabla)\, u - 2\omega \wedge u - \nabla p \qquad \text{dans } W, \tag{6.3a}$$

$$\operatorname{div} u = 0 \qquad \text{dans } W, \tag{6.3b}$$

$$u_1 = 0 \text{ sur } G_1,\ u_2 = 0 \text{ sur } G_2,\ u_3 = 0 \qquad \text{sur } G_3, \tag{6.3c}$$

$$\nu_3 \frac{\partial u_1}{\partial \xi_3} = \tau_1,\ \nu_3 \frac{\partial u_2}{\partial \xi_3} = \tau_2,\ u_3 = 0 \qquad \text{sur } G_s. \tag{6.3d}$$

De plus, nous considérons la condition initiale (en $t = 0$)

$$u(.,0) = u_0 \text{ dans } W,$$

avec u_0 donnée à divergence nulle. La pression initiale $p(.,0)$ est alors déterminée à une constante additive près par (6.3a) (nous pouvons la choisir de moyenne nulle).

6.3 Système de coordonnées

Considérons le repère orthonormé fixe $\Sigma = \{O, e_1, e_2, e_3\}$ avec l'origine O au centre de la Terre et e_3 suivant l'axe de rotation de la Terre dirigé vers le nord. Au point de la Terre (supposée sphérique) de latitude α et de longitude φ est attaché le repère orthonormé $\Sigma' = \{O', e'_1, e'_2, e'_3\}$ (voir figure 21), où

$$\begin{aligned} e'_1 &= (-\sin\varphi, \cos\varphi, 0), \\ e'_2 &= (-\sin\alpha\cos\varphi, -\sin\alpha\sin\varphi, \cos\alpha), \\ e'_3 &= (\cos\alpha\cos\varphi, \cos\alpha\sin\varphi, \sin\alpha). \end{aligned}$$

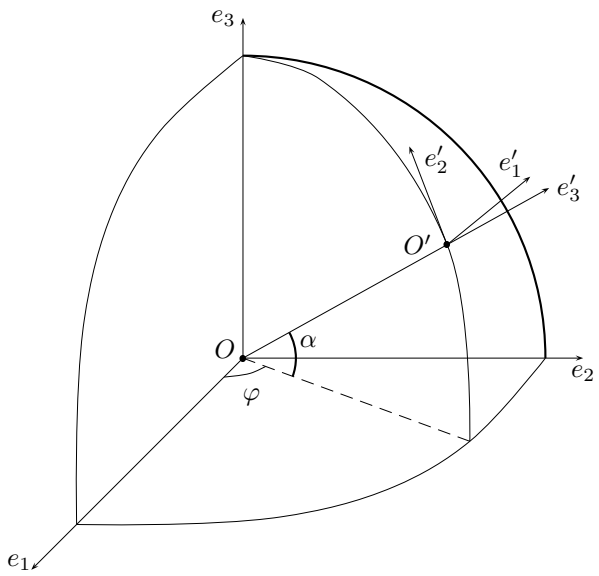

Figure 21.

Les équations de Navier–Stokes (6.3) sont écrites relativement aux coordonnées locales (ξ_1, ξ_2, ξ_3) dans le repère Σ'.

Remarque 6.3 *Nous travaillons avec des coordonnées euleriennes* $\xi = (\xi_1, \xi_2, \xi_3)$, *c'est–à–dire dans un repère fixe dans le temps. Dans un repère lagrangien attaché au fluide (repère mobile dans le temps) de coordonnées* $\zeta = (\zeta_1, \zeta_2, \zeta_3) = \zeta(\xi, t)$, *nous avons*

$$\frac{d\zeta}{dt}(\xi, t) = u(\zeta(\xi, t), t)$$

et il suit

$$\frac{du}{dt}(\zeta, t) = \frac{\partial u}{\partial t}(\zeta, t) + \sum_{j=1}^{3} \frac{\partial u}{\partial \xi_j}(\zeta, t) u_j(\zeta, t) = \left[\frac{\partial u}{\partial t} + (\, u \,|\, \nabla \,)\, u\right](\zeta, t).$$

Ainsi, le terme non linéaire $(\, u \,|\, \nabla \,)\, u$ *des équations de Navier–Stokes provient du choix du repère de coordonnées fixe dans le temps.*

6.4 Force de Coriolis

Dans le repère Σ, la vitesse angulaire de la Terre est le vecteur $\omega = (0, 0, \omega_3)$, avec $\omega_3 = 2\pi/(24 \cdot 3600)$ (en radians par seconde). Si $a = (a_1, a_2, a_3)$ est un vecteur de $\mathbb{R}^3$ écrit dans le repère Σ, la matrice de la transformation linéaire $x \mapsto a \wedge x$ est, dans le repère Σ,

$$T_a = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}$$

et, dans le repère Σ', $T'_a = S^t T_a S$, où $S = (e'_1, e'_2, e'_3)$ est la matrice orthogonale dont les colonnes sont e'_1, e'_2 et e'_3.
Ainsi, si $u = u(\xi)$ est le vecteur vitesse exprimé en coordonnées locales (*i.e.* dans le repère Σ'), le terme donnant la force de Coriolis s'écrit, en coordonnées locales,

$$-2\omega \wedge u = T'_{-2\omega} u = 2\omega_3(u_2 \sin\alpha - u_3 \cos\alpha, -u_1 \sin\alpha, u_1 \cos\alpha). \tag{6.4}$$

La composante u_3 de la vitesse (*i.e.* la vitesse verticale des courants) étant petite par rapport aux vitesses horizontales, l'expression (6.4) peut être approchée par

$$T'_{-2\omega} u \approx T_{-2\omega'} = 2\omega_3(u_2 \sin\alpha, -u_1 \sin\alpha, 0),$$

où $\omega' = (0, 0, \omega_3 \sin\alpha)$ est la projection de ω sur l'axe porté par e'_3.

6.5 Renormalisation du bassin

Pour $i = 1, 2, 3$, notons d_i la dimension du bassin dans la direction ξ_i, *i.e.*

$$\begin{aligned} d_i &= \sup_{\xi,\zeta \in G_s} |\xi_i - \zeta_i|, \ i = 1, 2, \\ d_3 &= \sup_{\xi \in G_s} h(\xi). \end{aligned}$$

Remarquons que $d_3 \ll d_i$, $i = 1, 2$. Considérons les nouvelles coordonnées spatiales

$$x_i = \xi_i / d_i, \ i = 1, 2, 3$$

et le domaine Ω, correspondant à W,

$$\Omega = \{x = (x_1, x_2, x_3) \in \mathbb{R}^3 \ : \ (d_1 x_1, d_2 x_2, d_3 x_3) \in W\},$$

dont les dimensions dans les trois directions sont du même ordre de grandeur. Notons $\Gamma_s = \{(x_1, x_2) \in \mathbb{R}^2 \ : \ (d_1 x_1, d_2 x_2) \in G_s\}$ la surface de Ω, $\Gamma_b = \partial\Omega \setminus \Gamma_s$ le fond et $\Gamma_i = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \ : \ (d_1 x_1, d_2 x_2, d_3 x_3) \in G_i\}$ la partie de Γ_b correspondant à G_i après renormalisation, $i = 1, 2, 3$.

Nous avons

$$\begin{aligned} \frac{\partial u_i}{\partial \xi_j} &= \sum_{k=1}^{3} \frac{\partial u_i}{\partial x_k} \frac{\partial x_k}{\partial \xi_j} = d_j^{-1} \frac{\partial u_i}{\partial x_j}, \ 1 \leqslant i, j \leqslant 3, \\ \frac{\partial p}{\partial \xi_j} &= d_j^{-1} \frac{\partial p}{\partial x_j}, \ 1 \leqslant j \leqslant 3 \end{aligned}$$

et, en notant

$$\begin{aligned} \nabla_\alpha f &= (\alpha_1 \frac{\partial f}{\partial x_1}, \alpha_2 \frac{\partial f}{\partial x_2}, \alpha_3 \frac{\partial f}{\partial x_3}), \\ \mathrm{div}_\alpha \, u &= \sum_{i=1}^{3} \alpha_i \frac{\partial u_i}{\partial x_i}, \end{aligned}$$

pour $\alpha = (\alpha_1, \alpha_2, \alpha_3) \in \mathbb{R}^3$, les équations de Navier–Stokes (6.3) s'écrivent, dans ces nouvelles coordonnées,

$$\frac{\partial u}{\partial t} - \Delta_\lambda u = -\left(u \,|\, \nabla_\mu\right) u - 2\omega \wedge u - \nabla_\mu p \text{ dans } \Omega, \tag{6.5a}$$

$$\operatorname{div}_\mu u = 0 \text{ dans } \Omega, \tag{6.5b}$$

$$u_1 = 0 \text{ sur } \Gamma_1,\ u_2 = 0 \text{ sur } \Gamma_2,\ u_3 = 0 \text{ sur } \Gamma_3, \tag{6.5c}$$

$$\lambda_3 \frac{\partial u_1}{\partial x_3} = \Theta_1,\ \lambda_3 \frac{\partial u_2}{\partial x_3} = \Theta_2,\ u_3 = 0 \text{ sur } \Gamma_s, \tag{6.5d}$$

où $\lambda = (d_1^{-2}\nu_1, d_2^{-2}\nu_2, d_3^{-2}\nu_3)$, $\mu = (d_1^{-1}, d_2^{-1}, d_3^{-1})$ et $\Theta_i = d_3^{-1}\tau_i,\ i = 1, 2$.
La condition initiale reste

$$u(., 0) = u_0 \text{ dans } \Omega.$$

Soit n_Ω le vecteur normal unité sortant de Ω. Le vecteur n_W en un point (ξ_1, ξ_2, ξ_3) de ∂W et le vecteur n_Ω au point correspondant $(x_1, x_2, x_3) = (\mu_1\xi_1, \mu_2\xi_2, \mu_3\xi_3)$ de $\partial\Omega$ via la renormalisation satisfont la relation

$$(n_W)_i = C \cdot \mu_i \cdot (n_\Omega)_i,\ i = 1, 2, 3, \tag{6.6}$$

où $C = \left(\sum_{i=1}^3 \mu_i^2 (n_\Omega)_i^2\right)^{-1/2}$. Ainsi la relation (6.2) devient, après renormalisation,

$$\left(u \,|\, n_\Omega\right)_\mu = 0 \text{ sur } \partial\Omega, \tag{6.7}$$

où $\left(u \,|\, v\right)_\mu = \sum_{i=1}^3 \mu_i u_i v_i = u^t \mathrm{diag}(\mu_1, \mu_2, \mu_3) v$.

6.6 Viscosité et rapport d'aspect

Notons respectivement

$$d = d_1 \approx d_2 \qquad \text{et} \qquad h_0 = d_3$$

le diamètre horizontal et la profondeur du bassin W et $\varepsilon = \frac{h_0}{d}$ le rapport d'aspect.

Le vecteur de viscosité cinématique turbulente $\nu = (\nu_1, \nu_2, \nu_3)$ est déterminé à l'aide de l'approximation hydrostatique des équations de Navier–Stokes anisotropes [15] (voir aussi [4, 5]), obtenue en faisant tendre le paramètre ε vers 0. Pour l'établir, la valeur de ν_3 doit dépendre de ε de manière quadratique [15]. Ainsi, nous choisissons

$$\nu_1 \approx \nu_2,\ \nu_3 \approx \nu_1 \varepsilon^2, \tag{6.8}$$

afin de vérifier asymptotiquement l'approximation hydrostatique. Un modèle asymptotique des équations de Stokes est également traité dans [14].

Remarque 6.4 *Considérons les équations de Navier–Stokes renormalisées (6.5). En conséquence du choix de ν, dans le terme*

$$\Delta_\lambda u_i = \sum_{j=1}^3 \lambda_j \frac{\partial^2 u_i}{\partial x_j^2}$$

de l'équation (6.5a), les coefficients $\lambda_j = d_j^{-2}\nu_j$ *sont du même ordre de grandeur, car* $d_3 = h_0 = d \cdot \varepsilon$.

De plus, ν est le vecteur de viscosité cinématique *turbulente*. En considérant un maillage du bassin pour la résolution numérique, les turbulences locales (petits "tourbillons") à l'échelle d'une maille ne peuvent pas être simulées. Elles font apparaître des contraintes appelées contraintes de Reynolds et pour tenir compte de l'énergie qu'elles dissipent, nous augmentons la viscosité moléculaire du fluide (qui est alors appelée viscosité turbulente) [40].

D'après (6.8), il reste à calibrer ν_1. Ceci se fait de manière expérimentale de sorte à obtenir des courants de vitesse raisonnable lors de la résolution numérique.

CHAPITRE 7

Discrétisation en temps, méthodes de prédicteur–correcteur

La résolution numérique nécessite une discrétisation du problème en temps et en espace. Nous proposons une discrétisation spatiale par une méthode d'éléments finis mixtes au chapitre suivant. Ce chapitre traite la discrétisation temporelle des équations de Navier–Stokes par des méthodes de projections, voir par exemple [43, 56]. Deux modèles sont présentés dans les sections 7.2 et 7.5. Ils sont basés sur une technique à pas fractionnaire : un pas de temps est subdivisé en deux sous-étapes qui permettent de dissocier le calcul de la vitesse et de la pression, ainsi que les effets visqueux et l'incompressibilité [32]. La première est une étape de prédiction et la seconde une étape de correction (de pression [29] pour le modèle de la section 7.2 et de vitesse [33] pour celui de la section 7.5). Ces méthodes de projections sont ainsi aussi appelées méthodes de *prédicteur–correcteur*. Ce type de méthode est largement utilisé et leur fiabilité est démontrée, voir par exemple [32, 29, 33, 28, 30, 31, 48, 57].

Notre but dans cette deuxième partie est d'obtenir une simulation d'un écoulement océanique. Les méthodes utilisées sont présentées ; l'étude des estimations des erreurs des schémas sort du cadre de ce travail.

Tout d'abord, rappelons quelques notions classiques d'analyse numérique et fonctionnelle.

7.1 Espaces de Sobolev, notions de base

Les espaces de Sobolev sont étudiés dans [1], voir aussi [39, 45, 24]. Donnons ici quelques notions.

Soit U un ouvert de $\mathbb{R}^m$ (le cas qui nous intéresse est $m = 3$). Notons $L^2(U)$

l'espace de Hilbert des fonctions sur U à valeurs réelles de carré intégrable,

$$(\varphi \,|\, \psi) = \int_U \varphi(x)\psi(x)\mathrm{d}x,\ \varphi, \psi \in L^2(U)$$

le produit scalaire de $L^2(U)$ et $\|.\|$ la norme associée.
Considérons l'espace de Sobolev

$$H^1(U) = \left\{\varphi \in L^2(U)\ :\ \nabla\varphi \in L^2(U)^m\right\},$$

c'est–à–dire, une fonction φ de $L^2(U)$ est dans $H^1(U)$ lorsque, pour $1 \leqslant i \leqslant m$, la dérivée $\frac{\partial\varphi}{\partial x_i}$ est identifiable à une fonction de $L^2(U)$ au sens des distributions. C'est un espace de Hilbert pour le produit scalaire

$$\begin{aligned}(\varphi \,|\, \psi)_{1,U} &= (\varphi \,|\, \psi) + (\nabla\varphi \,|\, \nabla\psi)_{L^2(U)^m} \\ &= (\varphi \,|\, \psi) + \sum_{i=1}^{m} \left(\frac{\partial\varphi}{\partial x_i} \,|\, \frac{\partial\psi}{\partial x_i} \right),\ \varphi, \psi \in H^1(U)\ ;\end{aligned}$$

notons

$$\|\varphi\|_{1,U} = (\varphi \,|\, \varphi)^{1/2} = \left(\|\varphi\|^2 + \|\nabla\varphi\|^2_{L^2(U)^m} \right)^{1/2},\ \varphi \in H^1(U)$$

la norme associée à ce produit scalaire.
Sur $H^1(U)$, nous avons la semi–norme

$$|\varphi|_{1,U} = \|\nabla\varphi\|_{L^2(U)^m},\ \varphi \in H^1(U).$$

L'espace de Sobolev

$$H^1_0(U) = \overline{\mathcal{D}(U)}^{H^1(U)}$$

est un espace de Hilbert car il est fermé dans $H^1(U)$.

Lemme 7.1 (Inégalité de Poincaré–Friedrichs) *[45] Si U est un ouvert borné de $\mathbb{R}^m$, alors il existe une constante $C = C(U) > 0$ telle que*

$$\|\varphi\| \leqslant C \|\nabla\varphi\|_{L^2(U)^m}$$

pour tout $\varphi \in H^1_0(U)$.

D'après ce lemme, si U est borné, la semi–norme $|.|_{1,U}$ est une norme sur $H^1_0(U)$ équivalente à $\|.\|_{1,U}$ et $H^1_0(U)$ est un espace de Hilbert pour le produit scalaire

$$(\varphi \,|\, \psi)_{1,0,U} = (\nabla\varphi \,|\, \nabla\psi)_{L^2(U)^m},\ \varphi, \psi \in H^1_0(U).$$

Énonçons encore un lemme utile pour la suite :

Lemme 7.2 *[45] Si l'ouvert U est suffisamment régulier[1], alors l'espace*

$$\mathcal{D}(\overline{U}) = \{\varphi|_{\overline{U}}\ :\ \varphi \in \mathcal{D}(\mathbb{R}^m)\}$$

est dense dans $H^1(U)$.

[1] voir la référence pour les détails

7.1.1 Trace sur $H^1(U)$

La notion de trace sert à donner un sens à la restriction d'une application φ de $H^1(U)$ sur le bord ∂U de U, ce qui n'est *a priori* pas évident puisque les fonctions de $L^2(U)$ sont définies presque partout dans U et que ∂U est de mesure nulle.

Lemme 7.3 *[45] Avec les hypothèses du lemme 7.2, l'application linéaire* $\gamma_0 : \mathcal{D}(\overline{U}) \to L^2(\partial U)$, $\varphi \mapsto \varphi|_{\partial U}$ *se prolonge par continuité et densité en une application encore notée* $\gamma_0 \in L\left(H^1(U), L^2(\partial U)\right)$, *appelée* trace.

Remarque 7.1 *Nous supposerons U suffisamment régulier lorsque nous voulons utiliser la trace γ_0.*

Nous avons le résultat bien connu :

Théorème 7.4 (Formule de Green) *[45] Avec les hypothèses du lemme 7.2, nous avons, pour tout $\varphi, \psi \in H^1(U)$,*

$$\int_U \frac{\partial \varphi}{\partial x_i} \psi \, dx = -\int_U \varphi \frac{\partial \psi}{\partial x_i} \, dx + \int_{\partial U} \varphi \psi n_i \, d\sigma, \ 1 \leqslant i \leqslant m,$$

où n est le vecteur normal unité sortant de U, $d\sigma$ l'élément de surface de ∂U (et $\int_{\partial U} \varphi\psi n_i d\sigma = \int_{\partial U} \gamma_0(\varphi)\gamma_0(\psi) n_i d\sigma$).

La trace permet de caractériser l'espace $H_0^1(U)$:

Théorème 7.5 *[45] L'espace $H_0^1(U)$ est l'espace des fonctions de $H^1(U)$ qui sont nulles sur le bord de U :*

$$H_0^1(U) = \operatorname{Ker} \gamma_0 = \left\{ \varphi \in H^1(U) \ : \ \varphi = 0 \ \textit{sur} \ \partial U \right\}.$$

Grâce à la trace, si Γ_0 est une partie de ∂U, nous pouvons considérer l'espace

$$V(U, \Gamma_0) = \left\{ \varphi \in H^1(U) \ : \ \varphi = 0 \text{ sur } \Gamma_0 \right\} = \operatorname{Ker}(P\gamma_0),$$

où P est la restriction de $L^2(\partial U)$ à $L^2(\Gamma_0)$. Comme $V(U, \Gamma_0)$ est un sous-espace fermé de $H^1(U)$, c'est un espace de Hilbert pour le produit scalaire $(\,.\,|\,.\,)_{1,U}$.

7.1.2 L'espace $H^{1/2}(\partial U)$

Rappelons que si H est un espace de Hilbert et F un sous-espace fermé de H, alors l'espace quotient H/F, dont les éléments sont les classes d'équivalence de la relation sur H définie par

$$\varphi \sim \psi \iff \varphi - \psi \in F,$$

est un espace de Hilbert pour la norme

$$\|[\varphi]\|_{H/F} = \inf_{\varphi \in [\varphi]} \|\varphi\|_H, \ [\varphi] \in H/F$$

provenant du produit scalaire

$$(\,[\varphi]\,|\,[\psi]\,)_{H/F} = (\,\varphi_0\,|\,\psi_0\,)_H, \ [\varphi], [\psi] \in H/F,$$

où φ_0 (respectivement ψ_0) est l'unique élément de $[\varphi]$ (resp. $[\psi]$) orthogonal à F.

L'espace $H^1(U)/\operatorname{Ker}\gamma_0$ est obtenu en identifiant les fonctions de $H^1(U)$ qui ont même trace. L'espace de Sobolev $H^{1/2}(\partial U)$ est défini par

$$H^{1/2}(\partial U) = \gamma_0(H^1(U)).$$

Il est muni de la norme rendant la bijection linéaire $\overline{\gamma_0} : H^1(U)/\operatorname{Ker}\gamma_0 \to H^{1/2}(\partial U)$, $[\varphi] \mapsto \gamma_0\varphi$ isométrique, *i.e.*

$$\|f\|_{1/2,\partial U} = \left\|\overline{\gamma_0}^{-1} f\right\|_{H^1(U)/\operatorname{Ker}\gamma_0} = \inf_{\varphi\in H^1(U)\ :\ \gamma_0\varphi=f} \|\varphi\|_{1,U}\,,\ f \in H^{1/2}(\partial U).$$

L'espace $H^{1/2}(\partial U)$ est un espace de Hilbert pour le produit scalaire

$$(\,f\,|\,g\,)_{1/2,\partial U} = (\,\varphi_0\,|\,\psi_0\,)_{1,U}\,,\ f,g \in H^{1/2}(\partial U),$$

où φ_0 (respectivement ψ_0) est l'unique élément de $\{\varphi \in H^1(U)\ :\ \gamma_0\varphi = f\}$ (resp. $\{\psi \in H^1(U)\ :\ \gamma_0\psi = g\}$) orthogonal à $\operatorname{Ker}\gamma_0$.

Remarquons que si $\Gamma_0 \subset \partial U$, l'espace $H^{1/2}(\Gamma_0)$ se définit comme ci–dessus en remplaçant γ_0 par $P\gamma_0$, où P est la restriction de $L^2(\partial U)$ à $L^2(\Gamma_0)$.

7.1.3 Dualité

Si H est un espace de Hilbert réel, H' désigne l'espace dual de H, *i.e.* l'espace des formes linéaires continues de H dans $\mathbb{R}$. Pour $f \in H'$ et $\varphi \in H$, la dualité est notée $f(\varphi) = \langle f, \varphi\rangle_{H',H}$ ($= \langle f, \varphi\rangle$ lorsque le contexte est clair). L'espace H' est muni de la norme opérateur :

$$\|f\|_{H'} = \sup_{\varphi\in H\setminus\{0\}} \frac{\langle f,\varphi\rangle_{H',H}}{\|\varphi\|_H},\ f \in H'.$$

Comme H' est isométriquement isomorphe à H via l'application de Riesz $J : H \to H'$, $\varphi \mapsto (\,\varphi\,|\,.\,)_H$, c'est un espace de Hilbert.

Les notations suivantes sont utilisées pour les espaces duaux de $H_0^1(U)$ et $H^{1/2}(\partial U)$:

$$\begin{aligned} H^{-1}(U) &= \left(H_0^1(U)\right)', \\ H^{-1/2}(\partial U) &= \left(H^{1/2}(\partial U)\right)'. \end{aligned}$$

La norme de $H^{-1}(U)$ se note $\|.\|_{-1,U}$ et celle de $H^{-1/2}(\partial U)$ se note $\|.\|_{-1/2,\partial U}$.

7.1.4 L'espace quotient $H^1(U)/\mathbb{R}$

Supposons que l'ouvert U soit connexe et borné et identifions les fonctions de $H^1(U)$ égales à une constante additive près, c'est–à–dire quotientons $H^1(U)$ par

le sous-espace fermé $\langle 1\rangle = \left\{\varphi \in H^1(U) \ : \ \varphi = \text{cste p.p. dans } U\right\}$. Cet espace quotient, noté $H^1(U)/\mathbb{R}$, est un espace de Hilbert pour la norme

$$\|\dot{\varphi}\|_{H^1(U)/\mathbb{R}} = \inf_{\varphi\in\dot{\varphi}} \|\varphi\|_{1,U}\,,\ \dot{\varphi} \in H^1(U)/\mathbb{R}.$$

Remarquons que pour chaque classe $\dot{\varphi} \in H^1(U)/\mathbb{R}$, nous pouvons choisir le représentant $\varphi \in H^1(U)$ de moyenne nulle sur U et donc noter

$$H^1(U)/\mathbb{R} = \left\{\varphi \in H^1(U) \ : \ \int_U \varphi(x)\mathrm{d}x = 0\right\}. \tag{7.1}$$

Théorème 7.6 (Nečas) *[39] Lorsque le domaine U est suffisamment régulier,*

$$|\dot{\varphi}|_{H^1(U)/\mathbb{R}} = |\varphi|_{1,U}\,,\ \dot{\varphi} \in H^1(U)/\mathbb{R}$$

est une norme sur $H^1(U)/\mathbb{R}$ équivalente à la norme $\|.\|_{H^1(U)/\mathbb{R}}$ et donc $H^1(U)/\mathbb{R}$ est un espace de Hilbert pour le produit scalaire

$$\left(\dot{\varphi}\,|\,\dot{\psi}\right)_{1,0,H^1(U)/\mathbb{R}} = (\,\varphi\,|\,\psi\,)_{1,0,U}\,,\ \dot{\varphi},\dot{\psi} \in H^1(U)/\mathbb{R}.$$

7.1.5 Une décomposition de $(L^2(U))^m$ en somme orthogonale

Introduisons encore les espaces de Sobolev

$$H(\mathrm{div};U) = \left\{v \in \left(L^2(U)\right)^m \ : \ \mathrm{div}\, v \in L^2(U)\right\}$$

qui est un espace de Hilbert pour le produit scalaire

$$(\,u\,|\,v\,)_{H(\mathrm{div};U)} = (\,u\,|\,v\,)_{L^2(U)^m} + (\,\mathrm{div}\, u\,|\,\mathrm{div}\, v\,)\,,\ u,v \in H(\mathrm{div};U)$$

et

$$H_0(\mathrm{div};U) = \overline{(\mathcal{D}(U))^m}^{H(\mathrm{div};U)}.$$

Théorème 7.7 *[24] Si l'ouvert U est suffisamment régulier, $\left(\mathcal{D}(\overline{U})\right)^m$ est dense dans $H(\mathrm{div};U)$ et l'application linéaire $\gamma_n : \left(\mathcal{D}(\overline{U})\right)^m \to L^2(\partial U)$, $v \mapsto (\,v\,|\,n\,)\,|_{\partial U}$ se prolonge par continuité en une application encore notée $\gamma_n \in L\left(H(\mathrm{div};U), H^{-1/2}(\partial U)\right)$.*

En supposant que U soit suffisamment régulier, nous avons les résultats suivants (voir [24]).

Théorème 7.8 *[24] L'espace $H_0(\mathrm{div};U)$ est caractérisé par*

$$H_0(\mathrm{div};U) = \mathrm{Ker}\,\gamma_n = \{v \in H(\mathrm{div};U) \ : \ (\,u\,|\,n\,)\,|_{\partial U} = 0\}\,.$$

L'espace

$$H = \{v \in H_0(\mathrm{div};U) \ : \ \mathrm{div}\, u = 0\}$$

est un sous-espace fermé de $\left(L^2(U)\right)^m$, d'orthogonal

$$H^\perp = \left\{\nabla\varphi \ : \ \varphi \in H^1(U)\right\} = \nabla\left(H^1(U)/\mathbb{R}\right),$$

ce qui nous permet d'écrire

$$\left(L^2(U)\right)^m = H \oplus H^\perp = H \oplus \nabla\left(H^1(U)/\mathbb{R}\right).$$

7.1.6 Théorème de Lax–Milgram

Rappelons encore le résultat bien connu :

Théorème 7.9 (Lax–Milgram) *[45] Soit H un espace de Hilbert réel, soit $a : H \times H \to \mathbb{R}$ une application bilinéaire, continue,* i.e.

$$\exists M \ : \ |a(u,v)| \leqslant M \|u\| \|v\| \ \ \forall u, v \in H$$

et coercitive, i.e.

$$\exists \alpha > 0 \ : \ a(u,u) \geqslant \alpha \|u\|^2 \ \ \forall u \in H$$

et soit $l \in H'$ (une forme linéaire continue de H dans $\mathbb{R}$). Alors le problème

$$\textit{Trouver } u \in H \textit{ tel que } a(u,v) = \langle l, v \rangle \textit{ pour tout } v \in H$$

admet une unique solution; de plus, l'application qui à l fait correspondre la solution u est linéaire et continue de H' dans H et vérifie

$$\|u\| \leqslant \alpha^{-1} \|l\| .$$

7.2 Méthode de prédicteur–correcteur (I)

Dans cette section, une première méthode de prédicteur–correcteur est décrite (avec correction de pression) pour la discrétisation en temps des équations de Navier–Stokes.

Notons $u(t) = u(.,t)$ et $p(t) = p(.,t)$. Fixons un pas de temps $\delta t > 0$ et posons $t_k = k\delta t$ pour $k \geqslant 0$. Le but est de calculer des approximations u^k de $u(t_k)$ et p^k de $p(t_k)$, pour $k \geqslant 0$. Les mêmes notations sont utilisées pour les fonctions connues τ et Θ provenant des tractions.
La dérivée $\frac{\partial u}{\partial t}$ est approchée par la formule d'Euler implicite (d'autres cas sont considérés plus loin (section 7.4)) et les approximations u^{k+1} et p^{k+1} sont obtenues à partir de u^k et p^k en deux étapes de la manière suivante. La première étape consiste à calculer une prédiction $\tilde{u}^{k+1}$ de la vitesse au temps t_{k+1} vérifiant les conditions de bord mais à divergence libre. À la deuxième étape, la prédiction $\tilde{u}^{k+1}$ est projetée sur un espace de fonctions à divergence nulle. Pour cela l'approximation de la pression p^{k+1} en t_{k+1} est calculée et une correction de pression est apportée à $\tilde{u}^{k+1}$ pour obtenir la projection u^{k+1}. La vitesse u^{k+1} vérifie alors une condition de Dirichlet homogène sur le bord seulement pour sa composante normale, voir par exemple [29, 43].

7.2.1 Cas des équations de Navier–Stokes non renormalisées

Pour les équations de Navier–Stokes non renormalisées (6.3) : nous cherchons $\tilde{u}^{k+1}$, solution de

$$\frac{\tilde{u}^{k+1} - u^k}{\delta t} - \Delta_\nu \tilde{u}^{k+1} = -\left(u^k \,|\, \nabla\right) u^k - 2\omega \wedge u^k - \nabla p^k \text{ dans } W, \qquad (7.2a)$$

$$\tilde{u}_1^{k+1} = 0 \text{ sur } G_1, \ \tilde{u}_2^{k+1} = 0 \text{ sur } G_2, \ \tilde{u}_3^{k+1} = 0 \text{ sur } G_3, \qquad (7.2b)$$

$$\nu_3 \frac{\partial \tilde{u}_1^{k+1}}{\partial \xi_3} = \tau_1^{k+1}, \ \nu_3 \frac{\partial \tilde{u}_2^{k+1}}{\partial \xi_3} = \tau_2^{k+1}, \ \tilde{u}_3^{k+1} = 0 \text{ sur } G_s, \qquad (7.2c)$$

puis, u^{k+1} et p^{k+1}, solutions de

$$\frac{u^{k+1}-\tilde{u}^{k+1}}{\delta t}+\nabla\left(p^{k+1}-p^{k}\right)=0 \qquad \text{dans } W, \tag{7.3a}$$

$$\operatorname{div} u^{k+1}=0 \qquad \text{dans } W, \tag{7.3b}$$

$$\left(u^{k+1}\,|\,n_W\right)=0 \qquad \text{sur } \partial W, \tag{7.3c}$$

où n_W est le vecteur unité sortant normal à ∂W. Des équations (7.2a) et (7.3a), nous avons

$$\frac{u^{k+1}-u^{k}}{\delta t}-\Delta_\nu\tilde{u}^{k+1}=-\left(u^{k}\,|\,\nabla\right)u^{k}-2\omega\wedge u^{k}-\nabla p^{k+1} \text{ dans } W, \tag{7.4}$$

ce qui donne une approximation de l'équation (6.3a). Le système (7.2) consiste à calculer une *prédiction* $\tilde{u}^{k+1}$ à divergence libre de la vitesse $u(t_{k+1})$, alors que le système (7.3) donne la *correction* u^{k+1} à divergence nulle de $\tilde{u}^{k+1}$ ainsi que la pression p^{k+1}. Plus précisément, u^{k+1} est la projection orthogonale de $\tilde{u}^{k+1}$ sur

$$H=\{u\in H(\operatorname{div};W)\ :\ \operatorname{div} u=0,\ (u\,|\,n_W)\,|_{\partial W}=0\}$$

d'après la somme orthogonale

$$\left(L^2(W)\right)^3=H\oplus\nabla(H^1(W)/\mathbb{R})$$

(voir section 7.1.5). Cette méthode dissocie le calcul de la diffusion (système (7.2)) et la condition d'incompressibilité (caractérisée par la divergence nulle) (système (7.3)).

Remplaçons le système (7.3) par

$$-\Delta\left(p^{k+1}-p^{k}\right)=-\frac{1}{\delta t}\operatorname{div}\tilde{u}^{k+1} \qquad \text{dans } W, \tag{7.5a}$$

$$\frac{\partial\left(p^{k+1}-p^{k}\right)}{\partial n_W}=0 \qquad \text{sur } \partial W \tag{7.5b}$$

et

$$u^{k+1}=\tilde{u}^{k+1}-\delta t\nabla\left(p^{k+1}-p^{k}\right) \text{ dans } W, \tag{7.6}$$

où $\frac{\partial q}{\partial n_W}=(\nabla q\,|\,n_W)$ désigne la dérivée normale de q.

Montrons que le système (7.3) est équivalent à (7.5) et (7.6). Les équations (7.3a) et (7.6) sont les mêmes. L'équation (7.5a) est obtenue en prenant la divergence de (7.3a) et en utilisant (7.3b). D'après les hypothèses considérées à la section 6.2 sur les parties G_1, G_2 et G_3 du bord de W, la solution $\tilde{u}^{k+1}$ du système (7.2) vérifie (voir (6.2))

$$\left(\tilde{u}^{k+1}\,|\,n_W\right)=0 \text{ sur } \partial W. \tag{7.7}$$

Donc, en prenant le produit scalaire de (7.3a) contre n_W sur ∂W et avec (7.3c), nous obtenons (7.5b). Réciproquement, en prenant la divergence de (7.6) et avec (7.5a), nous déduisons (7.3b) ; en prenant le produit scalaire de (7.6) contre n_W sur ∂W et avec (7.5b) et (7.7), nous avons (7.3c).

Remarquons que le vecteur u^{k+1} satisfait la condition (6.3d) sur G_s :

$$\begin{aligned} u_3^{k+1} &= \left(u^{k+1} \,|\, n_W \right) = 0, \\ \nu_3 \frac{\partial u_i^{k+1}}{\partial \xi_3} &= \nu_3 \left(\frac{\partial \tilde{u}_i^{k+1}}{\partial \xi_3} - \delta t \frac{\partial}{\partial \xi_3} \frac{\partial (p^{k+1} - p^k)}{\partial \xi_i} \right) \\ &= \tau_i^{k+1} - \nu_3 \delta t \frac{\partial}{\partial \xi_i} \frac{\partial (p^{k+1} - p^k)}{\partial n_W} = \tau_i^{k+1}, \end{aligned}$$

alors que sur G_b, nous avons seulement $\left(u^{k+1} \,|\, n_W \right) = 0$, c'est–à–dire la composante normale à G_b nulle et la condition (6.3c) ($u_i^{k+1} = 0$ sur G_i, $i = 1, 2, 3$) n'est pas nécessairement satisfaite.

7.2.2 Cas des équations de Navier–Stokes renormalisées

Récrivons ces étapes pour les équations de Navier–Stokes renormalisées (6.5). L'étape de prédiction s'écrit

$$\frac{\tilde{u}^{k+1} - u^k}{\delta t} - \Delta_\lambda \tilde{u}^{k+1} = - \left(u^k \,|\, \nabla_\mu \right) u^k - 2\omega \wedge u^k - \nabla_\mu p^k \text{ dans } \Omega, \quad (7.8a)$$

$$\tilde{u}_1^{k+1} = 0 \text{ sur } \Gamma_1, \ \tilde{u}_2^{k+1} = 0 \text{ sur } \Gamma_2, \ \tilde{u}_3^{k+1} = 0 \text{ sur } \Gamma_3, \quad (7.8b)$$

$$\lambda_3 \frac{\partial \tilde{u}_1^{k+1}}{\partial x_3} = \Theta_1^{k+1}, \ \lambda_3 \frac{\partial \tilde{u}_2^{k+1}}{\partial x_3} = \Theta_2^{k+1}, \ \tilde{u}_3^{k+1} = 0 \text{ sur } \Gamma_s. \quad (7.8c)$$

L'étape de correction s'écrit

$$\frac{u^{k+1} - \tilde{u}^{k+1}}{\delta t} + \nabla_\mu \left(p^{k+1} - p^k \right) = 0 \qquad \text{dans } \Omega, \quad (7.9a)$$

$$\operatorname{div}_\mu u^{k+1} = 0 \qquad \text{dans } \Omega, \quad (7.9b)$$

$$\left(u^{k+1} \,|\, n_\Omega \right)_\mu = 0 \qquad \text{sur } \partial\Omega. \quad (7.9c)$$

Des équations (7.8a) et (7.9a), nous obtenons l'approximation

$$\frac{u^{k+1} - u^k}{\delta t} - \Delta_\lambda \tilde{u}^{k+1} = - \left(u^k \,|\, \nabla_\mu \right) u^k - 2\omega \wedge u^k - \nabla_\mu p^{k+1} \text{ dans } \Omega \quad (7.10)$$

de l'équation (6.5a).

Le système (7.9) est remplacé par

$$-\Delta_{\mu^2} \left(p^{k+1} - p^k \right) = -\frac{1}{\delta t} \operatorname{div}_\mu \tilde{u}^{k+1} \qquad \text{dans } \Omega, \quad (7.11a)$$

$$\frac{\partial \left(p^{k+1} - p^k \right)}{\partial_{\mu^2} n_\Omega} = 0 \qquad \text{sur } \partial\Omega \quad (7.11b)$$

et

$$u^{k+1} = \tilde{u}^{k+1} - \delta t \nabla_\mu \left(p^{k+1} - p^k \right) \text{ dans } \Omega, \quad (7.12)$$

où $\mu^2 = \left(\mu_1^2, \mu_2^2, \mu_3^2\right)$ et où la notation $\frac{\partial\varphi}{\partial_\alpha n_\Omega} = \left(\nabla_\alpha\varphi \,|\, n_\Omega\right)$ est utilisée pour une fonction φ et un vecteur α.

De manière similaire au cas des équations non renormalisées, la solution $\tilde{u}^{k+1}$ du système (7.8) vérifie (voir (6.7))

$$\left(\tilde{u}^{k+1} \,|\, n_\Omega\right)_\mu = 0 \text{ sur } \partial\Omega$$

et le système (7.9) est équivalent à (7.11) et (7.12). De même, u^{k+1} vérifie la condition (6.5d), alors que sur Γ_b nous avons $\left(u^{k+1} \,|\, n_\Omega\right)_\mu = 0$ sans que la condition (6.5c) soit nécessairement satisfaite.

7.2.3 Compléments sur les conditions de bord

Dans le système (7.2), la composante $\tilde{u}_i^{k+1}$ est libre sur la partie $G_b \setminus G_i$ et donc ne subit aucune traction sur cette partie, c'est–à–dire,

$$\frac{\partial\tilde{u}_i^{k+1}}{\partial_\nu n_W} = 0 \text{ sur } G_b \setminus G_i,\ i = 1,2,3.$$

Pour le système renormalisé (7.8), ceci se traduit par

$$\frac{\partial\tilde{u}_i^{k+1}}{\partial_\lambda n_\Omega} = 0 \text{ sur } \Gamma_b \setminus \Gamma_i,\ i = 1,2,3. \tag{7.13}$$

7.3 Formulations faibles

Dès maintenant, les équations renormalisées sont considérées et le vecteur unité normal sortant de Ω est noté $n = n_\Omega$.

Vitesse (prédiction)

Donnons la formulation faible pour chaque composante du système (7.8) avec la condition de bord supplémentaire (7.13). Notons

$$F^k = \frac{1}{\delta t}u^k - \left(u^k \,|\, \nabla_\mu\right) u^k - 2\omega \wedge u^k - \nabla_\mu p^k. \tag{7.14}$$

Pour les deux premières composantes ($i = 1,2$), nous avons affaire au problème elliptique avec condition de bord mixte (condition de Dirichlet homogène sur Γ_i et condition de Neumann sur $\Gamma_b \setminus \Gamma_i$ et Γ_s) :

$$\frac{1}{\delta t}\tilde{u}_i^{k+1} - \Delta_\lambda\tilde{u}_i^{k+1} = F_i^k \text{ dans } \Omega, \tag{7.15a}$$

$$\tilde{u}_i^{k+1} = 0 \text{ sur } \Gamma_i, \tag{7.15b}$$

$$\frac{\partial\tilde{u}_i^{k+1}}{\partial_\lambda n} = 0 \text{ sur } \Gamma_b \setminus \Gamma_i, \tag{7.15c}$$

$$\lambda_3\frac{\partial\tilde{u}_i^{k+1}}{\partial x_3} = \Theta_i^{k+1} \text{ sur } \Gamma_s. \tag{7.15d}$$

Remarquons que la dernière condition se récrit

$$\frac{\partial \tilde{u}_i^{k+1}}{\partial_\lambda n} = \Theta_i^{k+1} \text{ sur } \Gamma_s,$$

car $n = (0, 0, 1)$ sur Γ_s.
Considérons les espaces

$$V_i = V(\Omega, \Gamma_i) = \left\{ \varphi \in H^1(\Omega) \ : \ \varphi = 0 \text{ sur } \Gamma_i \right\}, \ i = 1, 2. \tag{7.16}$$

En multipliant (7.15a) par une fonction $\varphi \in V_i$ et en intégrant sur Ω, nous obtenons

$$\frac{1}{\delta t} \int_\Omega \tilde{u}_i^{k+1} \varphi \mathrm{d}x - \int_\Omega \Delta_\lambda \tilde{u}_i^{k+1} \varphi \mathrm{d}x = \int_\Omega F_i^k \varphi \mathrm{d}x. \tag{7.17}$$

En utilisant la formule de Green, nous avons

$$- \int_\Omega \Delta_\lambda \tilde{u}_i^{k+1} \varphi \mathrm{d}x = \int_\Omega \left(\nabla u_i^{k+1} \,|\, \nabla \varphi \right)_\lambda \mathrm{d}x - \int_{\partial\Omega} \frac{\partial \tilde{u}_i^{k+1}}{\partial_\lambda n} \varphi \mathrm{d}\sigma.$$

Avec le fait que $\varphi = 0$ sur Γ_i ($\varphi \in V_i$) et les conditions (7.15c) et (7.15d), nous déduisons

$$\int_{\partial\Omega} \frac{\partial \tilde{u}_i^{k+1}}{\partial_\lambda n} \varphi \mathrm{d}\sigma = \int_{\Gamma_b \setminus \Gamma_i} \frac{\partial \tilde{u}_i^{k+1}}{\partial_\lambda n} \varphi \mathrm{d}\sigma + \int_{\Gamma_s} \frac{\partial \tilde{u}_i^{k+1}}{\partial_\lambda n} \varphi \mathrm{d}\sigma = \int_{\Gamma_s} \Theta_i^{k+1} \varphi \mathrm{d}\sigma.$$

L'équation (7.17) devient donc

$$\frac{1}{\delta t} \int_\Omega \tilde{u}_i^{k+1} \varphi \mathrm{d}x + \int_\Omega \left(\nabla \tilde{u}_i^{k+1} \,|\, \nabla \varphi \right)_\lambda \mathrm{d}x = \int_\Omega F_i^k \varphi \mathrm{d}x + \int_{\Gamma_s} \Theta_i^{k+1} \varphi \mathrm{d}\sigma. \tag{7.18}$$

La formulation faible de (7.8) pour les composantes $i = 1, 2$ peut donc s'énoncer :

$$\boxed{\text{Trouver } \tilde{u}_i^{k+1} \in V_i \text{ vérifiant (7.18) pour tout } \varphi \in V_i.} \tag{7.19}$$

Pour la troisième composante, nous devons résoudre le problème elliptique avec condition de bord mixte (condition de Dirichlet homogène sur $\Gamma_3 \cup \Gamma_s$ et condition de Neumann homogène sur $\Gamma_b \setminus \Gamma_3$) :

$$\frac{1}{\delta t} \tilde{u}_3^{k+1} - \Delta_\lambda \tilde{u}_3^{k+1} = F_3^k \text{ dans } \Omega,$$

$$\tilde{u}_3^{k+1} = 0 \text{ sur } \Gamma_3 \cup \Gamma_s,$$

$$\frac{\partial \tilde{u}_3^{k+1}}{\partial_\lambda n} = 0 \text{ sur } \Gamma_b \setminus \Gamma_3.$$

Considérons l'espace

$$V_3 = V(\Omega, \Gamma_3 \cup \Gamma_s) = \left\{ \varphi \in H^1(\Omega) \ : \ \varphi = 0 \text{ sur } \Gamma_3 \cup \Gamma_s \right\}. \tag{7.20}$$

De la même manière que pour les deux premières composantes, pour $\varphi \in V_3$, nous avons

$$\frac{1}{\delta t} \int_\Omega \tilde{u}_3^{k+1} \varphi \mathrm{d}x + \int_\Omega \left(\nabla \tilde{u}_3^{k+1} \,|\, \nabla \varphi \right)_\lambda \mathrm{d}x = \int_\Omega F_3^k \varphi \mathrm{d}x \tag{7.21}$$

et la formulation faible :

$$\boxed{\text{Trouver } \tilde{u}_3^{k+1} \in V_3 \text{ vérifiant (7.21) pour tout } \varphi \in V_3.} \tag{7.22}$$

Avec

$$\begin{aligned} &a_i : V_i \times V_i \longrightarrow \mathbb{R}, \\ &\quad a_i(\varphi,\psi) = \frac{1}{\delta t}\int_\Omega \varphi\psi \mathrm{d}x + \int_\Omega (\,\nabla\varphi \,|\, \nabla\psi\,)_\lambda \,\mathrm{d}x,\ i=1,2,3, \end{aligned} \tag{7.23}$$

et

$$l_i : V_i \longrightarrow \mathbb{R},\quad \langle l_i,\varphi\rangle = \int_\Omega F_i^k\varphi \mathrm{d}x + \int_{\Gamma_s} \Theta_i^{k+1}\varphi \mathrm{d}\sigma,\ i=1,2, \tag{7.24}$$

$$l_3 : V_3 \longrightarrow \mathbb{R},\quad \langle l_3,\varphi\rangle = \int_\Omega F_3^k\varphi \mathrm{d}x, \tag{7.25}$$

les problèmes faibles du système (7.8) s'écrivent, pour $i=1,2,3$:

$$\text{Trouver } \tilde{u}_i^{k+1} \in V_i \text{ vérifiant } a_i(\tilde{u}_i^{k+1},\varphi) = \langle l_i,\varphi\rangle \text{ pour tout } \varphi \in V_i. \tag{7.26}$$

Les espaces V_i sont des espaces de Hilbert pour la norme $\|.\|_{1,\Omega}$. Les formes bilinéaires a_i sont continues :

$$|a_i(\varphi,\psi)| \leqslant \max\left(\frac{1}{\delta t},\lambda_1,\lambda_2,\lambda_3\right)\|\varphi\|_{1,\Omega}\,\|\psi\|_{1,\Omega}\ \ \forall\varphi,\psi\in V_i,$$

et coercitives :

$$a_i(\varphi,\varphi) \geqslant \min\left(\frac{1}{\delta t},\lambda_1,\lambda_2,\lambda_3\right)\|\varphi\|_{1,\Omega}^2\ \ \forall\varphi\in V_i.$$

Avec $F_i^k \in V_i'$ (en particulier avec $F_i^k \in L^2(\Omega)$) et $\Theta_i^{k+1} \in H^{-1/2}(\Gamma_s)$ (en particulier avec $\Theta_i^{k+1} \in L^2(\Gamma_s)$), les formes linéaires l_i sont continues :

$$\begin{aligned} |\langle l_i,\varphi\rangle| &\leqslant \left(\left\|F_i^k\right\|_{V_i'} + \left\|\Theta_i^{k+1}\right\|_{-1/2,\Gamma_s}\right)\|\varphi\|_{1,\Omega}\ \ \forall\varphi\in V_i,\ i=1,2, \\ |\langle l_3,\varphi\rangle| &\leqslant \left\|F_i^k\right\|_{V_i'}\|\varphi\|_{1,U}\ \ \forall\varphi\in V_3. \end{aligned}$$

Ainsi, par le théorème de Lax–Milgram, les problèmes faibles (7.26)(c'est–à–dire (7.19) et (7.22)) admettent chacun une unique solution.

Remarquons que dans l'intégrale $\int_\Omega F_i^k\varphi \mathrm{d}x$, apparaissant dans les formes linéaires (7.24) et (7.25), la contribution du dernier terme du vecteur F^k défini en (7.14) peut s'écrire, avec $\varphi \in V_i$,

$$-\int_\Omega \mu_i \frac{\partial p^k}{\partial x_i}\varphi \mathrm{d}x = \int_\Omega \mu_i p^k \frac{\partial\varphi}{\partial x_i}\mathrm{d}x,\ i=1,2,3, \tag{7.27}$$

ce qui permet d'éviter le calcul du gradient de la pression. En effet, en utilisant la formule de Green, il vient, pour $\varphi \in V_i$,

$$-\int_\Omega \mu_i \frac{\partial p^k}{\partial x_i}\varphi \mathrm{d}x = \int_\Omega \mu_i p^k \frac{\partial\varphi}{\partial x_i}\mathrm{d}x - \int_{\partial\Omega} p^k\varphi\mu_i n_i \mathrm{d}\sigma,\ i=1,2,3.$$

Comme $\varphi = 0$ sur Γ_i ($\varphi \in V_i$), nous avons

$$\int_{\partial\Omega} p^k \varphi n_i \mathrm{d}\sigma = \int_{\Gamma_s} p^k \varphi \mu_i n_i \mathrm{d}\sigma + \int_{\Gamma_b \setminus \Gamma_i} p^k \varphi \mu_i n_i \mathrm{d}\sigma,\ i = 1, 2, 3.$$

Sur Γ_s, nous avons

$$\int_{\Gamma_s} p^k \varphi \mu_i n_i \mathrm{d}\sigma = 0,\ i = 1, 2, 3,$$

car $n_i = 0$ pour $i = 1, 2$ et $\varphi = 0$ si $\varphi \in V_3$ (pour $i = 3$). Enfin, avant renormalisation du bassin, $G_b \setminus G_i$ est supposé contenu dans un plan parallèle à l'axe ξ_i (voir section 6.2), c'est–à–dire, le vecteur unité sortant de W vérifie $(n_W)_i = 0$; par conséquent, comme $\mu_i n_i = \mu_i (n_\Omega)_i$ est proportionnel à $(n_W)_i$ (voir (6.6)), nous avons $\mu_i n_i = 0$ sur $\Gamma_b \setminus \Gamma_i$ et donc

$$\int_{\Gamma_b \setminus \Gamma_i} p^k \varphi \mu_i n_i \mathrm{d}\sigma = 0,\ i = 1, 2, 3,$$

ce qui montre (7.27).

Pression

Donnons la formulation faible du système (7.11) pour la détermination de la pression p^{k+1}. Posons

$$q^{k+1} = p^{k+1} - p^k.$$

Le système (7.11) est un problème elliptique avec condition de Neumann. Par la formule de Green, nous avons, pour $\varphi \in H^1(\Omega)$,

$$\begin{aligned} -\int_\Omega \Delta_{\mu^2} q^{k+1} \varphi \mathrm{d}x &= \int_\Omega \left(\nabla_\mu q^{k+1} \,|\, \nabla_\mu \varphi \right) \mathrm{d}x - \int_{\partial\Omega} \frac{\partial q^{k+1}}{\partial_{\mu^2} n} \varphi \mathrm{d}\sigma \\ &= \int_\Omega \left(\nabla_\mu q^{k+1} \,|\, \nabla_\mu \varphi \right) \mathrm{d}x \end{aligned}$$

d'après la condition (7.11b). Nous obtenons ainsi, en multipliant l'équation (7.11a) par φ et en intégrant sur Ω,

$$\int_\Omega \left(\nabla_\mu q^{k+1} \,|\, \nabla_\mu \varphi \right) \mathrm{d}x = -\frac{1}{\delta t} \int_\Omega \mathrm{div}_\mu\, \tilde{u}^{k+1} \varphi \mathrm{d}x. \tag{7.28}$$

Comme $\tilde{u}^{k+1} \in V_1 \times V_2 \times V_3$, nous avons $\left(\tilde{u}^{k+1} \,|\, n \right)_\mu = 0$ sur $\partial\Omega$ (voir (6.7)) et, en utilisant la formule de Green, il suit

$$\begin{aligned} -\int_\Omega \mathrm{div}_\mu\, \tilde{u}^{k+1} \varphi \mathrm{d}x &= \int_\Omega \left(\tilde{u}^{k+1} \,|\, \nabla_\mu \varphi \right) \mathrm{d}x - \int_{\partial\Omega} \left(\tilde{u}^{k+1} \,|\, n \right)_\mu \varphi \mathrm{d}\sigma \\ &= \int_\Omega \left(\tilde{u}^{k+1} \,|\, \nabla_\mu \varphi \right) \mathrm{d}x. \end{aligned} \tag{7.29}$$

Nous pouvons alors écrire la formulation faible de (7.11) :

$$\boxed{\begin{array}{l} \text{Trouver } \dot{q}^{k+1} \in H^1(\Omega)/\mathbb{R} \text{ vérifiant} \\ \qquad a_p(\dot{q}^{k+1}, \dot{\varphi}) = \left\langle \dot{l}_p, \dot{\varphi} \right\rangle \text{ pour tout } \dot{\varphi} \in H^1(\Omega)/\mathbb{R}, \end{array}} \tag{7.30}$$

où

$$\dot{a}_p : H^1(\Omega)/\mathbb{R} \times H^1(\Omega)/\mathbb{R} \longrightarrow \mathbb{R},\ \dot{a}_p(\dot{\varphi},\dot{\psi}) = \int_\Omega (\,\nabla_\mu\varphi \,|\, \nabla_\mu\psi\,)\,\mathrm{d}x \tag{7.31}$$

et

$$\begin{array}{rcl} \dot{l}_p : H^1(\Omega)/\mathbb{R} \longrightarrow \mathbb{R},\ \left\langle \dot{l}_p, \dot{\varphi} \right\rangle & = & \dfrac{1}{\delta t}\displaystyle\int_\Omega (\,\tilde{u}^{k+1} \,|\, \nabla_\mu\varphi\,)\,\mathrm{d}x \\ & = & -\dfrac{1}{\delta t}\displaystyle\int_\Omega \mathrm{div}_\mu\, \tilde{u}^{k+1}\varphi \mathrm{d}x. \end{array} \tag{7.32}$$

En supposant Ω suffisamment régulier, nous pouvons munir $H^1(\Omega)/\mathbb{R}$ de la norme $|.|_{H^1(\Omega)/\mathbb{R}}$ (voir théorème 7.6) ; ainsi la forme bilinéaire $\dot{a}_p$ est continue :

$$|\dot{a}_p(\dot{\varphi},\dot{\psi})| \leqslant \max(\mu_1,\mu_2,\mu_3)^2 |\dot{\varphi}|_{H^1(\Omega)/\mathbb{R}} |\dot{\psi}|_{H^1(\Omega)/\mathbb{R}}\ \forall \dot{\varphi},\dot{\psi} \in H^1(\Omega)/\mathbb{R}$$

et coercitive :

$$\dot{a}_p(\dot{\varphi},\dot{\varphi}) \geqslant \min(\mu_1,\mu_2,\mu_3)^2 |\dot{\varphi}|^2_{H^1(\Omega)/\mathbb{R}}\ \forall \dot{\varphi} \in H^1(\Omega)/\mathbb{R}$$

et la forme linéaire $\dot{l}_p$ est continue :

$$\left|\left\langle \dot{l}_p, \dot{\varphi} \right\rangle\right| \leqslant \frac{1}{\delta t} \max(\mu_1,\mu_2,\mu_3)\, \left\|\tilde{u}^{k+1}\right\|_{(L^2(\Omega))^3} |\dot{\varphi}|_{H^1(\Omega)/\mathbb{R}}\ \forall \dot{\varphi} \in H^1(\Omega)/\mathbb{R}.$$

Par le théorème de Lax–Milgram, le problème faible (7.30) admet une unique solution (fonction de $H^1(\Omega)$ de moyenne nulle ($\int_\Omega q^{k+1}\mathrm{d}x = 0$), voir (7.1)).

Vitesse (correction)

Traitons enfin l'équation (7.12) composante à composante. Posons

$$w^{k+1} = \tilde{u}^{k+1} - u^{k+1}.$$

En multipliant la i–ème composante de (7.12) par une fonction $\varphi \in L^2(\Omega)$ et en intégrant sur Ω, nous obtenons

$$\int_\Omega w_i^{k+1}\varphi \mathrm{d}x = \delta t \int_\Omega \mu_i \frac{\partial q^{k+1}}{\partial x_i}\varphi \mathrm{d}x \tag{7.33}$$

et les formulations faibles, pour $i = 1,2,3$:

$$\boxed{\text{Trouver } w_i^{k+1} \in L^2(\Omega) \text{ vérifiant (7.33) pour tout } \varphi \in L^2(\Omega).} \tag{7.34}$$

Avec

$$a_c : L^2(\Omega) \times L^2(\Omega) \longrightarrow \mathbb{R},\ a_c(\varphi,\psi) = \int_\Omega \varphi\psi \mathrm{d}x \tag{7.35}$$

et

$$l_{c,i} : L^2(\Omega) \longrightarrow \mathbb{R},\ \langle l_c, \varphi\rangle = \delta t \int_\Omega \mu_i \frac{\partial q^{k+1}}{\partial x_i}\varphi \mathrm{d}x,\ i = 1,2,3, \tag{7.36}$$

ces problèmes faibles s'écrivent :

$$\begin{array}{l} \text{Trouver } w_i^{k+1} \in L^2(\Omega) \text{ vérifiant} \\ \quad a_c(w_i^{k+1},\varphi) = \langle l_{c,i}, \varphi\rangle \ \text{ pour tout } \varphi \in L^2(\Omega). \end{array} \tag{7.37}$$

et admettent clairement une unique solution par le théorème de Lax–Milgram.

7.3.1 Méthode de pénalisation pour le calcul de la pression

Pratiquement, le calcul de la pression est délicat. Une discrétisation spatiale par une méthode d'éléments finis est effectuée dans le chapitre suivant pour traiter les problèmes faibles, ce qui mène à la résolution de systèmes linéaires. La matrice du système obtenue pour le calcul de la pression, d'ordre N, est de rang $N-1$ car la pression est définie à une constante près. En supprimant une équation et une inconnue, nous avons alors affaire à une matrice SDP (voir sections 8.4 et 8.4.3) qui est très mal conditionnée (voir chapitre 1), ce qui rend la résolution avec une méthode du gradient conjugué (préconditionné) difficile. Pour obtenir plus facilement la pression (à chaque pas de temps), ajoutons un *terme de pénalisation* à l'équation (7.11a) : remplaçons cette dernière par

$$-\Delta_{\mu^2} q^{k+1} + \varepsilon_p q^{k+1} = -\frac{1}{\delta t}\,\mathrm{div}_\mu\,\tilde{u}^{k+1} + \varepsilon_p q^k \text{ dans } \Omega, \tag{7.38}$$

où $q^{k+1} = p^{k+1} - p^k$ et le *facteur de pénalisation* ε_p est positif et proche de 0. Dans le membre de gauche, nous avons ajouté $\varepsilon_p \mathcal{I} q^{k+1}$, où $\mathcal{I}$ est l'opérateur identité, dans le but d'augmenter les valeurs propres de la matrice obtenue ultérieurement (qui sera alors de rang maximal), voir section 9.3.1 pour une application numérique. Pour que ε_p ne soit pas en concurrence avec le rapport d'aspect ε (qui est un paramètre physique du modèle, voir section 6.6), nous imposons

$$0 < \varepsilon_p \ll \varepsilon.$$

L'équation (7.28) devient

$$\int_\Omega \left(\nabla_\mu q^{k+1} \,|\, \nabla_\mu \varphi \right) \mathrm{d}x + \varepsilon_p \int_\Omega q^{k+1}\varphi \mathrm{d}x = -\frac{1}{\delta t}\int_\Omega \mathrm{div}_\mu\,\tilde{u}^{k+1}\varphi \mathrm{d}x + \varepsilon_p \int_\Omega q^k \varphi \mathrm{d}x \tag{7.39}$$

et la formulation faible (7.30) est remplacée par

$$\boxed{\text{Trouver } q^{k+1} \in H^1(\Omega) \text{ vérifiant (7.39) pour tout } \varphi \in H^1(\Omega).} \tag{7.40}$$

La forme bilinéaire $\tilde{a}_p : H^1(\Omega) \to H^1(\Omega)$ définie par

$$\tilde{a}_p(\varphi, \psi) = \int_\Omega \left(\nabla_\mu \varphi \,|\, \nabla_\mu \psi \right) \mathrm{d}x + \varepsilon_p \int_\Omega \varphi\psi \mathrm{d}x \tag{7.41}$$

est continue :

$$|\tilde{a}_p(\varphi,\psi)| \leqslant \max\left(\mu_1^2, \mu_2^2, \mu_3^2, \varepsilon_p\right) \|\varphi\|_{1,\Omega} \|\psi\|_{1,\Omega} \;\; \forall \varphi, \psi \in H^1(\Omega)$$

et coercitive :

$$\tilde{a}_p(\varphi,\varphi) \geqslant \min\left(\mu_1^2, \mu_2^2, \mu_3^2, \varepsilon_p\right) \|\varphi\|_{1,\Omega}^2 \;\; \forall \varphi \in H^1(\Omega)$$

et la forme linéaire $\tilde{l}_p : H^1(\Omega) \to \mathbb{R}$ définie par

$$\begin{aligned} \left\langle \tilde{l}_p, \varphi \right\rangle &= \frac{1}{\delta t}\int_\Omega \left(\tilde{u}^{k+1} \,|\, \nabla_\mu \varphi \right) \mathrm{d}x + \varepsilon_p \int_\Omega q^k \varphi \mathrm{d}x \\ &\overset{(7.29)}{=} -\frac{1}{\delta t}\int_\Omega \mathrm{div}_\mu\,\tilde{u}^{k+1}\varphi \mathrm{d}x + \varepsilon_p \int_\Omega q^k \varphi \mathrm{d}x. \end{aligned} \tag{7.42}$$

est continue :

$$\left|\left\langle \tilde{l}_p, \varphi \right\rangle\right| \leqslant \max\left(\frac{\mu_1}{\delta t}, \frac{\mu_2}{\delta t}, \frac{\mu_3}{\delta t}, \varepsilon_p\right) \left\|(\tilde{u}^{k+1}, q^k)\right\|_{(L^2(\Omega))^4} \|\varphi\|_{1,\Omega} \quad \forall \varphi \in H^1(\Omega),$$

donc, par le théorème de Lax–Milgram, le problème faible (7.40) qui se récrit

$$\text{Trouver } q^{k+1} \in H^1(\Omega) \text{ vérifiant } \tilde{a}_p(q^{k+1}, \varphi) = \left\langle \tilde{l}_p, \varphi \right\rangle \text{ pour tout } \varphi \in H^1(\Omega), \tag{7.43}$$

admet une unique solution.

Remarque 7.2 *Il est aussi possible de ne pénaliser que le membre de gauche de l'équation (7.11a), c'est–à–dire de considérer les équations*

$$-\Delta_{\mu^2} q^{k+1} + \varepsilon_p q^{k+1} = -\frac{1}{\delta t} \operatorname{div}_\mu \tilde{u}^{k+1} \text{ dans } \Omega$$

et, avec $\varphi \in H^1(\Omega)$,

$$\int_\Omega \left(\nabla_\mu q^{k+1} \,|\, \nabla_\mu \varphi \right) dx + \varepsilon_p \int_\Omega q^{k+1} \varphi dx = -\frac{1}{\delta t} \int_\Omega \operatorname{div}_\mu \tilde{u}^{k+1} \varphi dx \tag{7.44}$$

et la formulation faible

$$\boxed{\text{Trouver } q^{k+1} \in H^1(\Omega) \text{ vérifiant (7.44) pour tout } \varphi \in H^1(\Omega).} \tag{7.45}$$

Remarque 7.3 *Ayant ajouté un terme de pénalisation, il n'est plus nécessaire de chercher une solution à moyenne nulle (*i.e. *de travailler avec le quotient* $H^1(\Omega)/\mathbb{R}$*).*

Remarque 7.4 *De l'équation (7.12) et avec (7.38), il suit*

$$\operatorname{div}_\mu u^{k+1} = \operatorname{div}_\mu \tilde{u}^{k+1} - \delta t \Delta_{\mu^2} q^{k+1} = \delta t \varepsilon_p \left(q^k - q^{k+1}\right) \text{ dans } \Omega,$$

c'est–à–dire, les vitesses obtenues (étape de correction) ne sont plus à divergence nulle et le fluide est faiblement compressible (le facteur de pénalisation étant "petit"). De la même manière pour la méthode de pénalisation de la remarque 7.2, nous obtenons $\operatorname{div}_\mu u^{k+1} = -\delta t \varepsilon_p q^{k+1}$ *dans* Ω.

7.4 Variante : méthode de différentiation rétrograde

Pour traiter la dérivée de la vitesse par rapport au temps dans les équations de Navier–Stokes, nous avons utilisé l'approximation

$$\frac{\partial u}{\partial t}(t_{k+1}) \approx \frac{1}{\delta t}(u^{k+1} - u^k) \tag{7.46}$$

(voir les équations (7.4) et (7.10)). Au lieu de prendre en compte uniquement la dernière approximation calculée de la vitesse, nous pouvons prendre les q précédentes, c'est–à–dire une approximation de la forme

$$\frac{\partial u}{\partial t}(t_{k+1}) \approx \frac{1}{\delta t} \left(\beta_0 u^{k+1} + \sum_{j=1}^{q} \beta_j u^{k+1-j} \right). \tag{7.47}$$

Les coefficients β_j sont choisis de sorte que l'erreur soit d'ordre δt^q et la formule (7.47) est appelée *formule de différentiation rétrograde (à q pas)* (*backward differentiation formula (with k steps)* en anglais). Ces formules sont données dans [34]. De plus, elles sont stables pour $1 \leqslant q \leqslant 6$, mais instables pour $q > 6$ [34].

Utilisons la formule de différentiation rétrograde à q pas dans les équations de Navier–Stokes renormalisées. L'équation (7.8a) est remplacée par

$$\begin{aligned}\frac{1}{\delta t}\left(\beta_0 \tilde{u}^{k+1} + \sum_{j=1}^{q} \beta_j u^{k+1-j}\right) - \Delta_\lambda \tilde{u}^{k+1} \\ = -\left(u^k \,|\, \nabla_\mu\right) u^k - 2\omega \wedge u^k - \nabla_\mu p^k \text{ dans } \Omega\end{aligned}$$

et l'équation (7.9a) par

$$\frac{\beta_0}{\delta t}(u^{k+1} - \tilde{u}^{k+1}) + \nabla_\mu \left(p^{k+1} - p^k\right) = 0 \text{ dans } \Omega.$$

Ces deux dernières équations donnent l'approximation

$$\begin{aligned}\frac{1}{\delta t}\left(\beta_0 u^{k+1} + \sum_{j=1}^{q} \beta_j u^{k+1-j}\right) - \Delta_\lambda \tilde{u}^{k+1} \\ = -\left(u^k \,|\, \nabla_\mu\right) u^k - 2\omega \wedge u^k - \nabla_\mu p^{k+1} \text{ dans } \Omega\end{aligned}$$

de (6.5a). L'équation (7.11a) devient

$$-\Delta_{\mu^2}\left(p^{k+1} - p^k\right) = -\frac{\beta_0}{\delta t} \operatorname{div}_\mu \tilde{u}^{k+1} \text{ dans } \Omega$$

et (7.12) devient

$$u^{k+1} = \tilde{u}^{k+1} - \frac{\delta t}{\beta_0} \nabla_\mu \left(p^{k+1} - p^k\right) \text{ dans } \Omega.$$

Donnons encore les changements dans les formulations faibles. Le vecteur défini en (7.14) est remplacé par

$$F^k = -\frac{1}{\delta t} \sum_{j=1}^{q} \beta_j u^{k+1-j} - \left(u^k \,|\, \nabla_\mu\right) u^k - 2\omega \wedge u^k - \nabla_\mu p^k.$$

Ensuite, chaque δt est remplacé par $\frac{\delta_t}{\beta_0}$.

7.4.1 Formule à deux pas

Donnons la formule de différentiation rétrograde pour $q = 2$. En supposant u suffisamment régulière, nous avons

$$u(t_{k-1}) = u(t_{k+1}) - 2\delta t \frac{\partial u}{\partial t}(t_{k+1}) + \frac{(2\delta t)^2}{2} \frac{\partial^2 u}{\partial t^2}(t_{k+1}) + \mathcal{O}(\delta t^3), \quad (7.48)$$

$$u(t_k) = u(t_{k+1}) - \delta t \frac{\partial u}{\partial t}(t_{k+1}) + \frac{\delta t^2}{2} \frac{\partial^2 u}{\partial t^2}(t_{k+1}) + \mathcal{O}(\delta t^3). \quad (7.49)$$

En multipliant (7.49) par 4 puis en soustrayant (7.48), nous obtenons

$$4u(t_k) - u(t_{k-1}) = 3u(t_{k+1}) - 2\delta t \frac{\partial u}{\partial t}(t_{k+1}) + \mathcal{O}(\delta t^3),$$

c'est–à–dire

$$\frac{\partial u}{\partial t}(t_{k+1}) = \frac{1}{\delta t}\left(\frac{3}{2}u(t_{k+1}) - 2u(t_k) + \frac{1}{2}u(t_{k-1})\right) + \mathcal{O}(\delta t^2) \tag{7.50}$$

et les coefficients de la formule (7.47) avec $q = 2$ sont

$$\beta_0 = \frac{3}{2}, \qquad \beta_1 = -2, \qquad \beta_2 = \frac{1}{2}. \tag{7.51}$$

Remarque 7.5 *La formule de différentiation rétrograde à un pas est simplement la formule d'Euler implicite donnée en (7.46).*

Remarque 7.6 *Avec une méthode de différentiation rétrograde à deux pas, le premier pas de temps (calcul de u^1 et p^1) se fait avec la formule d'Euler implicite. Plus simplement, si $u_0 = 0$, nous pouvons prendre $u^0 = u^1 = 0$ et $p^0 = p^1 = 0$ en supposant que les tractions sont nulles jusqu'au temps t_1.*

Remarque 7.7 *Expérimentalement, la méthode de prédicteur–correcteur (I) donne des vitesses irrégulières proche de la surface. Ceci est dû au fait qu'au bord, les vitesses obtenues ne satisfont une condition de Dirichlet homogène que pour la composante normale. Ainsi, nous utiliserons plutôt la méthode de prédicteur–correcteur (II) décrite à la section suivante, car les vitesses respectent alors les conditions de bord.*

7.5 Méthode de prédicteur–correcteur (II)

La méthode de prédicteur–correcteur présentée dans les sections précédentes consiste à donner, au temps t^{k+1}, une prédiction $\tilde{u}^{k+1}$ de la vitesse en tenant compte des conditions de bord du problème (système (7.8)), puis une pression p^{k+1} et une vitesse corrigée u^{k+1} à divergence nulle (système (7.9)). L'incompressibilité du fluide est ainsi respectée mais les conditions de bord ne sont pas garanties.

Inversons ici la prise en compte de l'incompressibilité du fluide et des conditions de bord. Nous procédons d'abord à une projection sur un espace de fonctions à divergence nulle (calcul de $\tilde{u}^{k+1}$ (prédiction) et p^{k+1}), puis effectuons une correction de vitesse pour obtenir u^{k+1} satisfaisant les conditions de bord [32, 33].

Utilisons ici la formule de différentiation rétrograde à deux pas pour l'approximation de la dérivée temporelle (voir (7.47) et (7.51)) et considérons seulement le cas des équations de Navier–Stokes renormalisées (6.5) (et notons $n = n_\Omega$).

Nous cherchons tout d'abord la vitesse $\tilde{u}^{k+1}$ (prédiction) et la pression p^{k+1}, solutions de

$$\begin{aligned}
\frac{1}{\delta t}\left(\frac{3}{2}\tilde{u}^{k+1} - 2u^k + \frac{1}{2}u^{k-1}\right) - \Delta_\lambda u^k &= -\left(u^k \,|\, \nabla_\mu\right) u^k - 2\omega \wedge u^k \\
&\quad -\nabla_\mu p^{k+1} \text{ dans } \Omega, \qquad (7.52a)\\
\operatorname{div}_\mu \tilde{u}^{k+1} &= 0 \text{ dans } \Omega, \qquad (7.52b)\\
\left(\tilde{u}^{k+1} \,|\, n\right)_\mu &= 0 \text{ sur } \partial\Omega, \qquad (7.52c)
\end{aligned}$$

puis la vitesse u^{k+1} (correction) telle que

$$\frac{1}{\delta t}\cdot\frac{3}{2}\left(u^{k+1} - \tilde{u}^{k+1}\right) - \Delta_\lambda\left(u^{k+1} - u^k\right) = 0 \text{ dans } \Omega, \qquad (7.53a)$$

$$u_1^{k+1} = 0 \text{ sur } \Gamma_1,\ u_2^{k+1} = 0 \text{ sur } \Gamma_2,\ u_3^{k+1} = 0 \text{ sur } \Gamma_3, \qquad (7.53b)$$

$$\lambda_3\frac{\partial u_1^{k+1}}{\partial x_3} = \Theta_1^{k+1},\ \lambda_3\frac{\partial u_2^{k+1}}{\partial x_3} = \Theta_2^{k+1},\ u_3^{k+1} = 0 \text{ sur } \Gamma_s. \qquad (7.53c)$$

Les équations (7.52a) et (7.53a) donnent l'approximation

$$\begin{aligned}
\frac{1}{\delta t}\left(\frac{3}{2}u^{k+1} - 2u^k + \frac{1}{2}u^{k-1}\right) - \Delta_\lambda u^{k+1} &= -\left(u^k \,|\, \nabla_\mu\right) u^k - 2\omega \wedge u^k \\
&\quad -\nabla_\mu p^{k+1} \text{ dans } \Omega \qquad (7.54)
\end{aligned}$$

de l'équation (6.5a).
Avec cette méthode, toutes les conditions de bord du problème (6.5) sont vérifiées, alors que le fluide n'est plus nécessairement incompressible ($\operatorname{div}_\mu u^{k+1}$ peut être non nulle).

En soustrayant à (7.52a) l'équation (7.52a) de l'étape précédente, *i.e.*

$$\begin{aligned}
\frac{1}{\delta t}\left(\frac{3}{2}\tilde{u}^{k} - 2u^{k-1} + \frac{1}{2}u^{k-2}\right) - \Delta_\lambda u^{k-1} &= -\left(u^{k-1} \,|\, \nabla_\mu\right) u^{k-1} - 2\omega \wedge u^{k-1} \\
&\quad -\nabla_\mu p^{k} \text{ dans } \Omega,
\end{aligned}$$

nous obtenons

$$\begin{aligned}
&\frac{1}{\delta t}\left(\frac{3}{2}\left(\tilde{u}^{k+1} - \tilde{u}^k\right) - 2u^k + \frac{5}{2}u^{k-1} - \frac{1}{2}u^{k-2}\right) - \Delta_\lambda\left(u^k - u^{k-1}\right) = \\
&\quad -\left(u^k \,|\, \nabla_\mu\right) u^k + \left(u^{k-1} \,|\, \nabla_\mu\right) u^{k-1} - 2\omega \wedge \left(u^k - u^{k-1}\right) - \nabla_\mu\left(p^{k+1} - p^k\right)
\end{aligned}$$

et avec (7.53a) de l'étape précédente, *i.e.* $\Delta_\lambda(u^k - u^{k-1}) = \frac{1}{\delta t}\cdot\frac{3}{2}(u^k - \tilde{u}^k)$, il suit

$$\begin{aligned}
&\frac{1}{\delta t}\left(\frac{3}{2}\tilde{u}^{k+1} - \frac{7}{2}u^k + \frac{5}{2}u^{k-1} - \frac{1}{2}u^{k-2}\right) = \\
&\quad -\left(u^k \,|\, \nabla_\mu\right) u^k + \left(u^{k-1} \,|\, \nabla_\mu\right) u^{k-1} - 2\omega \wedge \left(u^k - u^{k-1}\right) - \nabla_\mu\left(p^{k+1} - p^k\right). \qquad (7.55)
\end{aligned}$$

Notons

$$\begin{aligned} v^k &= -\frac{1}{\delta t}\left(-\frac{7}{2}u^k + \frac{5}{2}u^{k-1} - \frac{1}{2}u^{k-2}\right) \\ &\quad - \left(u^k \,|\, \nabla_\mu\right) u^k + \left(u^{k-1} \,|\, \nabla_\mu\right) u^{k-1} - 2\omega \wedge \left(u^k - u^{k-1}\right); \quad (7.56) \end{aligned}$$

comme $\operatorname{div}_\mu \tilde{u}^{k+1} = 0$ dans Ω et $\left(\tilde{u}^{k+1} \,|\, n\right)_\mu = 0$ sur $\partial\Omega$, en prenant la divergence (div_μ) de (7.55) et en faisant le produit scalaire de (7.55) contre n ($(.\,|\,n)_\mu$), nous obtenons respectivement les équations

$$-\Delta_{\mu^2}\left(p^{k+1} - p^k\right) = -\operatorname{div}_\mu v^k \text{ dans } \Omega, \quad (7.57a)$$

$$-\frac{\partial\left(p^{k+1} - p^k\right)}{\partial_{\mu^2} n} = -\left(v^k \,|\, n\right)_\mu \text{ sur } \partial\Omega, \quad (7.57b)$$

qui déterminent la pression p^{k+1}.

Ensuite, l'équation (7.54) avec les conditions de bord (7.53b) et (7.53c), c'est–à–dire le système

$$\begin{aligned} \frac{1}{\delta t}\left(\frac{3}{2}u^{k+1} - 2u^k + \frac{1}{2}u^{k-1}\right) - \Delta_\lambda u^{k+1} &= -\left(u^k \,|\, \nabla_\mu\right) u^k - 2\omega \wedge u^k \\ &\quad -\nabla_\mu p^{k+1} \text{ dans } \Omega \quad (7.58a) \end{aligned}$$

$$u_1^{k+1} = 0 \text{ sur } \Gamma_1,\ u_2^{k+1} = 0 \text{ sur } \Gamma_2,\ u_3^{k+1} = 0 \text{ sur } \Gamma_3, \quad (7.58b)$$

$$\lambda_3 \frac{\partial u_1^{k+1}}{\partial x_3} = \Theta_1^{k+1},\ \lambda_3 \frac{\partial u_2^{k+1}}{\partial x_3} = \Theta_2^{k+1},\ u_3^{k+1} = 0 \text{ sur } \Gamma_s, \quad (7.58c)$$

permet de calculer la vitesse u^{k+1}.

Remarque 7.8 *Les vitesses de l'étape de prédiction (*i.e. *les vecteurs $\tilde{u}^j$) n'apparaissent plus dans les systèmes (7.57) et (7.58) et ne sont donc pas calculées.*

7.6 Formulations faibles

Pression

Donnons la formulation faible du système (7.57). Notons comme précédemment

$$q^{k+1} = p^{k+1} - p^k.$$

Multiplions l'équation (7.57a) par $\varphi \in H^1(\Omega)$; en utilisant la formule de Green et (7.57b), nous obtenons

$$\int_\Omega \left(\nabla_\mu q^{k+1} \,|\, \nabla_\mu \varphi\right) \mathrm{d}x = \int_\Omega \left(v^k \,|\, \nabla_\mu \varphi\right) \mathrm{d}x. \quad (7.59)$$

et la formulation faible :

$$\boxed{\text{Trouver } q^{k+1} \in H^1(\Omega)/\mathbb{R} \text{ vérifiant (7.59) pour tout } \varphi \in H^1(\Omega)/\mathbb{R}.} \quad (7.60)$$

Comme la formulation faible (7.30), le problème (7.60) admet une unique solution (de moyenne nulle).

Remarque 7.9 *Le calcul de la pression peut se faire avec pénalisation en suivant la même méthodologie qu'à la section (7.3.1).*

Vitesse

La formulation faible du système (7.58) s'obtient de la même manière que celle du système (7.8), en particulier la condition de bord supplémentaire (7.13) est prise en compte. Notons

$$w^k = -\frac{1}{\delta t}\left(-2u^k + \frac{1}{2}u^{k-1}\right) - \left(u^k \,|\, \nabla_\mu\right) u^k - 2\omega \wedge u^k - \nabla_\mu p^{k+1}. \tag{7.61}$$

Reprenons les espaces V_1, V_2 et V_3 définis en (7.16) et (7.20). Pour les deux premières composantes de la vitesse, $i = 1, 2$, nous avons, avec $\varphi \in V_i$,

$$\frac{1}{\delta t} \cdot \frac{3}{2} \int_\Omega u_i^{k+1} \varphi \mathrm{d}x + \int_\Omega \left(\nabla u_i^{k+1} \,|\, \nabla \varphi\right)_\lambda \mathrm{d}x = \int_\Omega w_i^k \varphi \mathrm{d}x + \int_{\Gamma_s} \Theta_i^{k+1} \varphi \mathrm{d}\sigma. \tag{7.62}$$

et la formulation faible :

$$\boxed{\text{Trouver } u_i^{k+1} \in V_i \text{ vérifiant (7.62) pour tout } \varphi \in V_i.} \tag{7.63}$$

Pour la troisième composante de la vitesse et $\varphi \in V_3$, nous avons

$$\frac{1}{\delta t} \cdot \frac{3}{2} \int_\Omega u_3^{k+1} \varphi \mathrm{d}x + \int_\Omega \left(\nabla u_3^{k+1} \,|\, \nabla \varphi\right)_\lambda \mathrm{d}x = \int_\Omega w_3^k \varphi \mathrm{d}x \tag{7.64}$$

et la formulation faible :

$$\boxed{\text{Trouver } u_3^{k+1} \in V_3 \text{ vérifiant (7.64) pour tout } \varphi \in V_3.} \tag{7.65}$$

Comme avant, la relation (7.27) (avec $k+1$ au lieu de k) permet d'éviter le calcul du gradient de la pression dans l'intégrale $\int_\Omega w_i^k \varphi \mathrm{d}x$, $i = 1, 2, 3$.

Remarque 7.10 *L'initialisation peut se faire de manière suivante. Nous considérons les vitesses $u^0 = u^{-1} = u^{-2} = u_0$ et la pression p^0 de moyenne nulle satisfaisant l'équation (6.5a). Ainsi, nous pouvons amorcer la résolution par le calcul de p^1 avec (7.60), puis de u^1 avec (7.63) et (7.65).*

CHAPITRE 8

Discrétisation en espace, méthode des éléments finis

Une discrétisation spatiale avec des éléments finis permet d'obtenir des formulations approchées des problèmes faibles du chapitre précédent, qui consistent en la résolution de systèmes linéaires dont les matrices sont SDP et creuses. Des méthodes du gradient conjugué (préconditionné) sont alors utilisées pour les résoudre (voir première partie).

Dans ce chapitre, nous décrivons les éléments finis employés (de type Q_m), la méthodologie pour obtenir un problème faible approché et la construction de sa matrice et de son second membre. Les calculs pour les formulations faibles (7.60), (7.63) et (7.65), obtenues par la méthode de prédicteur–correcteur (II), sont présentés en détail.

8.1 Maillage

Considérons un maillage tridimensionnel τ_h, formés d'*éléments* hexaédriques de $\mathbb{R}^3$, approchant Ω et renotons

$$\Omega = \text{int}\left(\bigcup_{T\in\tau_h} T\right).$$

Un maillage τ_h de Ω a les propriétés suivantes :

- deux éléments distincts de τ_h sont disjoints ou ont une face, une arête ou un sommet en commun,
- $\overline{\Omega} = \bigcup_{T\in\tau_h} T$.

Un maillage τ_h de Ω induit sur la surface Γ_s de Ω un maillage bidimensionnel $\tau_{s,h}$. Dans la suite, la lettre T désigne un élément tridimensionnel de τ_h et la lettre K un élément bidimensionnel de $\tau_{s,h}$.

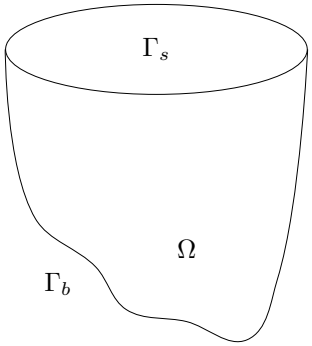

Figure 22: Bassin Ω "original".

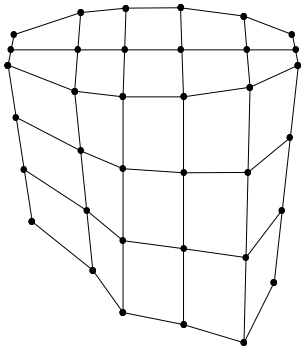

Figure 23: Maillage de Ω.

8.2 Éléments finis

Définissons précisément la notion d'élément fini.

Définition 8.1 *[21] Un élément fini dans $\mathbb{R}^d$ est un triplet (L, P, Σ) vérifiant les propriétés suivantes :*

1. *L est un compact de $\mathbb{R}^d$ d'intérieur non vide et de bord $\mathcal{C}^1$ par morceaux,*
2. *P est un espace vectoriel de fonctions de L dans $\mathbb{R}$,*
3. *Σ est un ensemble de formes linéaires sur P, $\Sigma = \{\phi_i\}_{1 \leqslant i \leqslant n}$, tel que*
   - *Σ est une famille linéairement indépendante,*
   - *Σ est unisolvant,* i.e. *pour tout $\theta_1, \ldots, \theta_n \in \mathbb{R}$, il existe un unique $p \in P$ satisfaisant $\phi_i(p) = \theta_i$, $1 \leqslant i \leqslant n$.*

Avec les notations de cette définition, la famille $\{\varphi_j\}_{1 \leqslant j \leqslant n}$ de P vérifiant

$$\phi_i(\varphi_j) = \delta_{ij},\ 1 \leqslant i, j \leqslant n$$

est une base de P dans laquelle tout $p \in P$ s'écrit

$$p = \sum_{j=1}^{n} \phi_j(p)\varphi_j.$$

L'élément fini (L, P, Σ) peut aussi être noté $P(L)$ sans expliciter Σ.

8.2.1 Élément fini tridimensionnel de type Q_m

Considérons l'élément de référence tridimensionnel

$$\widehat{T} = [-1, 1] \times [-1, 1] \times [-1, 1].$$

Pour $m \geqslant 1$, considérons l'espace vectoriel des fonctions de $\widehat{T}$ à valeurs dans $\mathbb{R}$

$$\begin{aligned} Q_m(\widehat{T}) \quad = \quad & \Big\{ \widehat{\varphi} : \widehat{T} \to \mathbb{R} \ :\ \widehat{\varphi} \text{ est polynômiale de degré} \\ & \text{inférieur ou égal à } m \text{ en chaque variable} \Big\}. \end{aligned}$$

La dimension de $Q_m(\widehat{T})$ est $(m+1)^3$. Considérons les $(m+1)^3$ points de $\widehat{T}$ donnés par

$$\widehat{S}_{ijk}^{(m)} = \widehat{S}_{ijk} = \left(\frac{2i}{m} - 1, \frac{2j}{m} - 1, \frac{2k}{m} - 1\right),\ 0 \leqslant i, j, k \leqslant m$$

et les $(m+1)^3$ formes linéaires sur $Q_m(\widehat{T})$,

$$\widehat{\phi}_{ijk}^{(m)} = \widehat{\phi}_{ijk} : Q_m(\widehat{T}) \longrightarrow \mathbb{R},\ \widehat{\phi}_{ijk}\left(\widehat{\varphi}\right) = \widehat{\varphi}\left(\widehat{S}_{ijk}\right),\ 0 \leqslant i, j, k \leqslant m,$$

i.e. les évaluations aux points $\widehat{S}_{ijk}$. Pour $1 \leqslant l \leqslant 3$ et $0 \leqslant i \leqslant m$, notons

$$\hat{x}_l^{(i)} = \frac{2i}{m} - 1,$$

ainsi $\widehat{S}_{ijk} = \left(\hat{x}_1^{(i)}, \hat{x}_2^{(j)}, \hat{x}_3^{(k)}\right)$. Pour $0 \leqslant i, j, k \leqslant m$, considérons les fonctions de $Q_m(\widehat{T})$

$$\widehat{g}_{ijk}(\hat{x}) = \prod_{l_1 \neq i} \left(\hat{x}_1 - \hat{x}_1^{(l_1)}\right) \cdot \prod_{l_2 \neq j} \left(\hat{x}_2 - \hat{x}_2^{(l_2)}\right) \cdot \prod_{l_3 \neq k} \left(\hat{x}_3 - \hat{x}_3^{(l_3)}\right)$$

(où les indices l_1, l_2, l_3 varient de 0 à m) et

$$\widehat{\varphi}_{ijk}^{(m)}(\hat{x}) = \widehat{\varphi}_{ijk}(\hat{x}) = \left(\widehat{g}_{ijk}\left(\widehat{S}_{ijk}\right)\right)^{-1} \cdot \widehat{g}_{ijk}(\hat{x}).$$

Ces fonctions vérifient

$$\phi_{i_1 j_1 k_1}\left(\widehat{\varphi}_{i_2 j_2 k_2}\right) = \widehat{\varphi}_{i_2 j_2 k_2}\left(\widehat{S}_{i_1 j_1 k_1}\right) = \delta_{i_1 i_2} \delta_{j_1 j_2} \delta_{k_1 k_2},$$

et la famille $\{\widehat{\varphi}_{ijk}\}_{0 \leqslant i,j,k \leqslant m}$ est une base de $Q_m(\widehat{T})$ dans laquelle toute $\widehat{\varphi} \in Q_m(\widehat{T})$ s'écrit

$$\widehat{\varphi} = \sum_{i,j,k=0}^{m} \widehat{\varphi}\left(\widehat{S}_{ijk}\right) \widehat{\varphi}_{ijk}.$$

Le triplet $\left(\widehat{T}, Q_m(\widehat{T}), \Sigma = \{\phi_{ijk}\}_{0 \leqslant i,j,k \leqslant m}\right)$ définit l'élément fini tridimensionnel de référence de type Q_m, noté simplement $Q_m(\widehat{T})$. Les points $\widehat{S}_{ijk}$, $0 \leqslant i, j, k \leqslant m$ sont appelés les ***nœuds*** de l'élément $\widehat{T}$.

Considérons $(m+1)^3$ points de Ω, $S_{ijk}^{(m)} = S_{ijk}$, $0 \leqslant i, j, k \leqslant m$ et l'application

$$F^{(m)} = F : \widehat{T} \longrightarrow \mathbb{R}^3,\ F_T(\hat{x}) = \sum_{i,j,k=0}^{m} S_{ijk} \widehat{\varphi}_{ijk}(\hat{x}). \tag{8.1}$$

Lorsque F est injective, son image $T = F(\widehat{T})$ définit un élément isoparamétrique [21] de type Q_m ; les points $(m+1)^3$ points $S_{ijk}^{(m)}$ sont alors appelés les nœuds de T et l'application F est notée $F = F_T^{(m)} = F_T$. Les nœuds de T sont donc les images par F_T des nœuds de l'élément de référence. Remarquons qu'un élément T de τ_h peut être un hexaèdre dont les arêtes et les faces sont "courbes".

En pratique, nous renumérotons les nœuds et les fonctions de base afin d'éviter les triples indices. Explicitons les cas $m = 1$ et $m = 2$.

8.2.2 Élément fini tridimensionnel de type Q_1

Les nœuds de $\widehat{T}$ sont ses sommets,

$$\begin{array}{llll} \widehat{S}_1 & = (-1,-1,-1), & \widehat{S}_5 & = (-1,-1,\ 1), \\ \widehat{S}_2 & = (\ 1,-1,-1), & \widehat{S}_6 & = (\ 1,-1,\ 1), \\ \widehat{S}_3 & = (\ 1,\ 1,-1), & \widehat{S}_7 & = (\ 1,\ 1,\ 1), \\ \widehat{S}_4 & = (-1,\ 1,-1), & \widehat{S}_8 & = (-1,\ 1,\ 1), \end{array} \tag{8.2}$$

et les fonctions de base de $Q_1(\widehat{T})$ (de dimension 8) sont

$$\begin{array}{ll} \widehat{\varphi}_1^{(1)}(\hat{x}) = \frac{1}{8}(1-\hat{x}_1)(1-\hat{x}_2)(1-\hat{x}_3), & \widehat{\varphi}_5^{(1)}(\hat{x}) = \frac{1}{8}(1-\hat{x}_1)(1-\hat{x}_2)(1+\hat{x}_3), \\ \widehat{\varphi}_2^{(1)}(\hat{x}) = \frac{1}{8}(1+\hat{x}_1)(1-\hat{x}_2)(1-\hat{x}_3), & \widehat{\varphi}_6^{(1)}(\hat{x}) = \frac{1}{8}(1+\hat{x}_1)(1-\hat{x}_2)(1+\hat{x}_3), \\ \widehat{\varphi}_3^{(1)}(\hat{x}) = \frac{1}{8}(1+\hat{x}_1)(1+\hat{x}_2)(1-\hat{x}_3), & \widehat{\varphi}_7^{(1)}(\hat{x}) = \frac{1}{8}(1+\hat{x}_1)(1+\hat{x}_2)(1+\hat{x}_3), \\ \widehat{\varphi}_4^{(1)}(\hat{x}) = \frac{1}{8}(1-\hat{x}_1)(1+\hat{x}_2)(1-\hat{x}_3), & \widehat{\varphi}_8^{(1)}(\hat{x}) = \frac{1}{8}(1-\hat{x}_1)(1+\hat{x}_2)(1+\hat{x}_3). \end{array} \tag{8.3}$$

Pour un élément T de nœuds (sommets) $S_1, \ldots, S_8$, nous avons

$$F_T^{(1)} : \widehat{T} \longrightarrow T,\ F_T^{(1)}(\hat{x}) = \sum_{i=1}^{8} S_i \widehat{\varphi}_i^{(1)}(\hat{x}).$$

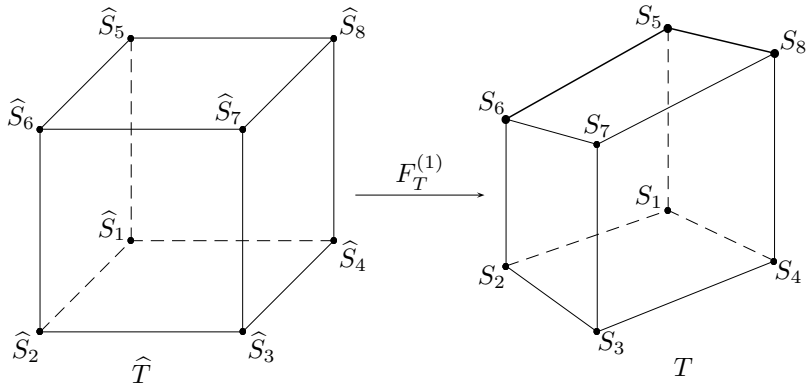

Figure 24.

Remarque 8.1 (Faces d'un élément tridimensionnel) *Si $S_1, \ldots, S_8$ désignent les sommets d'un élément T tridimensionnel (voir figure 24), les faces de T sont numérotées comme suit.*

Face	*Sommets*				*Plan*		
K_1	S_1	S_2	S_6	S_5	$\hat{x}_2$	$=$	-1
K_2	S_2	S_3	S_7	S_6	$\hat{x}_1$	$=$	1
K_3	S_3	S_4	S_8	S_7	$\hat{x}_2$	$=$	1
K_4	S_1	S_5	S_8	S_4	$\hat{x}_1$	$=$	-1
K_5	S_1	S_4	S_3	S_2	$\hat{x}_3$	$=$	-1
K_6	S_5	S_6	S_7	S_8	$\hat{x}_3$	$=$	1

Tableau 43.

Dans la dernière colonne du tableau 43 figure le plan contenant la face correspondante de $\widehat{T}$.

8.2.3 Élément fini tridimensionnel de type Q_2

Les nœuds de $\widehat{T}$ sont ses sommets (voir (8.2)),

$$\widehat{S}_i,\ 1 \leqslant i \leqslant 8,$$

les milieux de ses arêtes,

$$\begin{array}{llll}
\widehat{S}_9 = (\ 0,-1,-1) = \frac{1}{2}\left(\widehat{S}_1+\widehat{S}_2\right), & \widehat{S}_{13} = (\ 0,-1,\ 1) = \frac{1}{2}\left(\widehat{S}_5+\widehat{S}_6\right), \\
\widehat{S}_{10} = (\ 1,\ 0,-1) = \frac{1}{2}\left(\widehat{S}_2+\widehat{S}_3\right), & \widehat{S}_{14} = (\ 1,\ 0,\ 1) = \frac{1}{2}\left(\widehat{S}_6+\widehat{S}_7\right), \\
\widehat{S}_{11} = (\ 0,\ 1,-1) = \frac{1}{2}\left(\widehat{S}_3+\widehat{S}_4\right), & \widehat{S}_{15} = (\ 0,\ 1,\ 1) = \frac{1}{2}\left(\widehat{S}_7+\widehat{S}_8\right), \\
\widehat{S}_{12} = (-1,\ 0,-1) = \frac{1}{2}\left(\widehat{S}_4+\widehat{S}_1\right), & \widehat{S}_{16} = (-1,\ 0,\ 1) = \frac{1}{2}\left(\widehat{S}_8+\widehat{S}_5\right),
\end{array} \tag{8.4}$$

$$\begin{array}{ll}
\widehat{S}_{17} = (-1,-1,\ 0) = \frac{1}{2}\left(\widehat{S}_1+\widehat{S}_5\right), \\
\widehat{S}_{18} = (\ 1,-1,\ 0) = \frac{1}{2}\left(\widehat{S}_2+\widehat{S}_6\right), \\
\widehat{S}_{19} = (\ 1,\ 1,\ 0) = \frac{1}{2}\left(\widehat{S}_3+\widehat{S}_7\right), \\
\widehat{S}_{20} = (-1,\ 1,\ 0) = \frac{1}{2}\left(\widehat{S}_4+\widehat{S}_8\right),
\end{array} \tag{8.5}$$

les milieux de ses faces,

$$\begin{array}{ll}
\widehat{S}_{21} = (\ 0,\ 0,-1) = \frac{1}{4}\left(\widehat{S}_1+\widehat{S}_4+\widehat{S}_3+\widehat{S}_2\right), \\
\widehat{S}_{22} = (\ 1,\ 0,\ 0) = \frac{1}{4}\left(\widehat{S}_2+\widehat{S}_3+\widehat{S}_7+\widehat{S}_6\right), \\
\widehat{S}_{23} = (\ 0,\ 0,\ 1) = \frac{1}{4}\left(\widehat{S}_5+\widehat{S}_6+\widehat{S}_7+\widehat{S}_8\right), \\
\widehat{S}_{24} = (-1,\ 0,\ 0) = \frac{1}{4}\left(\widehat{S}_1+\widehat{S}_5+\widehat{S}_8+\widehat{S}_4\right), \\
\widehat{S}_{25} = (\ 0,-1,\ 0) = \frac{1}{4}\left(\widehat{S}_1+\widehat{S}_2+\widehat{S}_6+\widehat{S}_5\right), \\
\widehat{S}_{26} = (\ 0,\ 1,\ 0) = \frac{1}{4}\left(\widehat{S}_3+\widehat{S}_4+\widehat{S}_8+\widehat{S}_7\right),
\end{array} \tag{8.6}$$

et son centre

$$\widehat{S}_{27} = (0,0,0) = \frac{1}{8}\sum_{i=1}^{8}\widehat{S}_i. \tag{8.7}$$

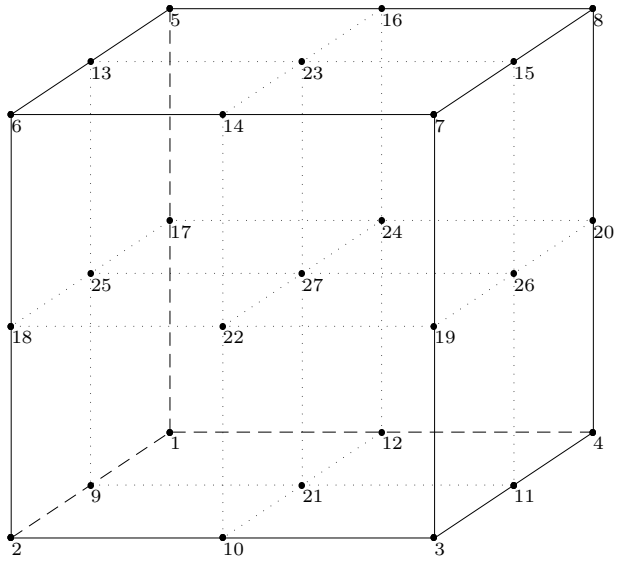

Figure 25: Numérotation des nœuds de $Q_2(\widehat{T})$.

Les fonctions

$$\alpha_1(t) = -\frac{1}{2}t(1-t),\ \alpha_2(t) = (1+t)(1-t),\ \alpha_3(t) = \frac{1}{2}(1+t)t, \tag{8.8}$$

vérifiant

$$\begin{array}{rclrclrcl} \alpha_1(-1) & = & 1, & \alpha_1(0) & = & 0, & \alpha_1(1) & = & 0, \\ \alpha_2(-1) & = & 0, & \alpha_2(0) & = & 1, & \alpha_2(1) & = & 0, \\ \alpha_3(-1) & = & 0, & \alpha_3(0) & = & 0, & \alpha_3(1) & = & 1, \end{array}$$

permettent d'exprimer facilement les fonctions de base de $Q_2(\widehat{T})$ (de dimension 27),

$$\begin{array}{lll} \widehat{\varphi}_1^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_1(\hat{x}_2)\alpha_1(\hat{x}_3), & \widehat{\varphi}_{10}^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_2(\hat{x}_2)\alpha_1(\hat{x}_3), & \widehat{\varphi}_{19}^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_3(\hat{x}_2)\alpha_2(\hat{x}_3), \\ \widehat{\varphi}_2^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_1(\hat{x}_2)\alpha_1(\hat{x}_3), & \widehat{\varphi}_{11}^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_3(\hat{x}_2)\alpha_1(\hat{x}_3), & \widehat{\varphi}_{20}^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_3(\hat{x}_2)\alpha_2(\hat{x}_3), \\ \widehat{\varphi}_3^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_3(\hat{x}_2)\alpha_1(\hat{x}_3), & \widehat{\varphi}_{12}^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_2(\hat{x}_2)\alpha_1(\hat{x}_3), & \widehat{\varphi}_{21}^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_2(\hat{x}_2)\alpha_1(\hat{x}_3), \\ \widehat{\varphi}_4^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_3(\hat{x}_2)\alpha_1(\hat{x}_3), & \widehat{\varphi}_{13}^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_1(\hat{x}_2)\alpha_3(\hat{x}_3), & \widehat{\varphi}_{22}^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_2(\hat{x}_2)\alpha_2(\hat{x}_3), \\ \widehat{\varphi}_5^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_1(\hat{x}_2)\alpha_3(\hat{x}_3), & \widehat{\varphi}_{14}^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_2(\hat{x}_2)\alpha_3(\hat{x}_3), & \widehat{\varphi}_{23}^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_2(\hat{x}_2)\alpha_3(\hat{x}_3), \\ \widehat{\varphi}_6^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_1(\hat{x}_2)\alpha_3(\hat{x}_3), & \widehat{\varphi}_{15}^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_3(\hat{x}_2)\alpha_3(\hat{x}_3), & \widehat{\varphi}_{24}^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_2(\hat{x}_2)\alpha_2(\hat{x}_3), \\ \widehat{\varphi}_7^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_3(\hat{x}_2)\alpha_3(\hat{x}_3), & \widehat{\varphi}_{16}^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_2(\hat{x}_2)\alpha_3(\hat{x}_3), & \widehat{\varphi}_{25}^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_1(\hat{x}_2)\alpha_2(\hat{x}_3), \\ \widehat{\varphi}_8^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_3(\hat{x}_2)\alpha_3(\hat{x}_3), & \widehat{\varphi}_{17}^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_1(\hat{x}_2)\alpha_2(\hat{x}_3), & \widehat{\varphi}_{26}^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_3(\hat{x}_2)\alpha_2(\hat{x}_3), \\ \widehat{\varphi}_9^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_1(\hat{x}_2)\alpha_1(\hat{x}_3), & \widehat{\varphi}_{18}^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_1(\hat{x}_2)\alpha_2(\hat{x}_3), & \widehat{\varphi}_{27}^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_2(\hat{x}_2)\alpha_2(\hat{x}_3). \end{array} \tag{8.9}$$

8.2.4 Passage du type Q_1 au type Q_2 (en 3D)

Soit T un élément de type Q_1 de sommets (nœuds) $S_1, \ldots, S_8$ et

$$F_T^{(1)} : \widehat{T} \longrightarrow T,\ F_T^{(1)}(\hat{x}) = \sum_{i=1}^{8} S_i \widehat{\varphi}_i^{(1)}(\hat{x})$$

l'application envoyant $\widehat{T}$ sur T où $\widehat{\varphi}_i^{(1)}$, $1 \leqslant i \leqslant 8$ sont les fonctions de base de l'espace $Q_1(\widehat{T})$ (voir (8.3)). Pour considérer T comme un élément de type Q_2, nous construisons les points

$$S_i = F_T^{(1)}(\widehat{S}_i) \in T,\ 9 \leqslant i \leqslant 27.$$

Les relations (8.4), (8.5), (8.6) et (8.7) sont alors aussi vérifiées en remplaçant les points de $\widehat{T}$ par les points de T correspondants et l'application

$$F_T^{(2)} : \widehat{T} \longrightarrow T,\ F_T^{(2)}(\hat{x}) = \sum_{i=1}^{27} S_i \widehat{\varphi}_i^{(2)}(\hat{x}),$$

où $\widehat{\varphi}_i^{(2)}$, $1 \leqslant i \leqslant 27$ sont les fonctions de base de l'espace $Q_2(\widehat{T})$ (voir (8.9)), est identique à $F_T^{(1)}$, *i.e.*

$$F_T^{(1)} = F_T^{(2)}.$$

De plus, l'espace $Q_1(\widehat{T})$ est inclus dans l'espace $Q_2(\widehat{T})$.

8.2.5 Élément fini bidimensionnel de type Q_m

En deux dimensions, l'élément de référence est défini par

$$\widehat{K} = [-1, 1] \times [-1, 1].$$

L'espace vectoriel considéré,

$$Q_m(\widehat{K}) = \left\{ \widehat{\varphi} : \widehat{K} \to \mathbb{R} \ :\ \widehat{\varphi} \text{ est polynômiale de degré inférieur ou égal à } m \text{ en chaque variable} \right\},$$

est de dimension $(m+1)^2$; les nœuds de $\widehat{K}$ sont les $(m+1)^2$ points

$$\widehat{C}_{ij}^{(m)} = \widehat{C}_{ij} = \left(\frac{2i}{m} - 1, \frac{2j}{m} - 1 \right),\ 0 \leqslant i, j \leqslant m,$$

et les $(m+1)^2$ fonctions de base

$$\widehat{\psi}_{ij}^{(m)}(\hat{x}) = \widehat{\psi}_{ij}(\hat{x}) = c_{ij} \cdot \prod_{l_1 \neq i} \left(\hat{x}_1 - \hat{x}_1^{(l_1)} \right) \cdot \prod_{l_2 \neq j} \left(\hat{x}_2 - \hat{x}_2^{(l_2)} \right),\ 0 \leqslant i, j \leqslant m,$$

où les c_{ij} sont les constantes réelles de "normalisation", c'est–à–dire les constantes qui donnent

$$\widehat{\psi}_{i_1 j_1} \left(\widehat{C}_{i_2 j_2} \right) = \delta_{i_1 i_2} \delta_{j_1 j_2}.$$

Considérons un élément T de τ_h de type Q_m de sommets S_{ijk}, $0 \leqslant i, j, k \leqslant m$. Lorsque la face supérieure (face numéro 6 d'après le tableau 43) de T est contenu dans la surface Γ_s de Ω, l'intersection $T \cap \Gamma_s$ est un élément K de $\tau_{s,h}$ de type Q_m de nœuds

$$C_{ij}^{(m)} = C_{ij} = S_{ijm},\ 0 \leqslant i, j \leqslant m. \tag{8.10}$$

L'élément K de type Q_m est l'image de l'application

$$F_K^{(m)} = F_K : \widehat{K} \longrightarrow K,\ F_K(\hat{x}) = \sum_{i,j=0}^{m} C_{ij}\widehat{\psi}_{ij}(\hat{x}),$$

qui envoie $\widehat{C}_{ij}$ sur C_{ij}, $0 \leqslant i,j \leqslant m$; nous avons

$$\widehat{\psi}_{ij}^{(m)} = \widehat{\varphi}_{ijm}^{(m)}\Big|_{\widehat{T}\cap\{\hat{x}_3=1\}},\ 0 \leqslant i,j \leqslant m \tag{8.11}$$

$$F_K^{(m)} = F_T^{(m)}\Big|_{\widehat{T}\cap\{\hat{x}_3=1\}}. \tag{8.12}$$

À nouveau, en pratique, les nœuds sont renumérotés pour éliminer le double indiçage. Explicitons les cas $m = 1$ et $m = 2$.

8.2.6 Élément fini bidimensionnel de type Q_1

Les nœuds de $\widehat{K}$ sont ses sommets,

$$\begin{array}{ll} \widehat{C}_1 & = (-1,-1), \\ \widehat{C}_2 & = (\ \ 1,-1), \\ \widehat{C}_3 & = (\ \ 1,\ \ 1), \\ \widehat{C}_4 & = (-1,\ \ 1) \end{array} \tag{8.13}$$

et les fonctions de base de $Q_1(\widehat{K})$ (de dimension 4) sont

$$\begin{array}{rcl} \widehat{\psi}_1^{(1)}(\hat{x}) & = & \frac{1}{4}(1-\hat{x}_1)(1-\hat{x}_2), \\ \widehat{\psi}_2^{(1)}(\hat{x}) & = & \frac{1}{4}(1+\hat{x}_1)(1-\hat{x}_2), \\ \widehat{\psi}_3^{(1)}(\hat{x}) & = & \frac{1}{4}(1+\hat{x}_1)(1+\hat{x}_2), \\ \widehat{\psi}_4^{(1)}(\hat{x}) & = & \frac{1}{4}(1-\hat{x}_1)(1+\hat{x}_2). \end{array}$$

Pour un élément $K \in \tau_{s,h}$ de nœuds (sommets) $C_1, \ldots, C_4$, nous avons

$$F_K^{(1)} : \widehat{K} \longrightarrow K,\ F_K^{(1)}(\hat{x}) = \sum_{i=1}^{4} C_i\widehat{\psi}_i^{(1)}(\hat{x}).$$

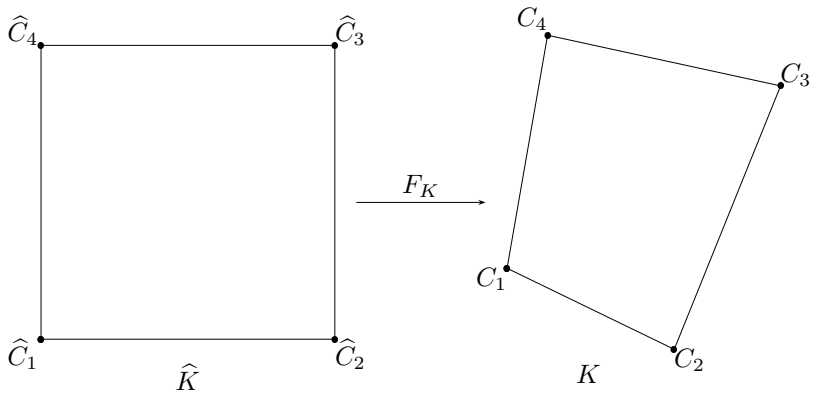

Figure 26.

Si $S_1, \ldots, S_8$ désignent les nœuds d'un élément T de τ_h de type Q_1 (voir figure 24 dont la face supérieure est contenue dans Γ_s, alors $K = T \cap \Gamma_s$ est un élément bidimensionnel de type Q_1 avec, d'après (8.10),

$$C_1 = S_5,\ C_2 = S_6,\ C_3 = S_7,\ C_4 = S_8.$$

8.2.7 Élément fini bidimensionnel de type Q_2

Les nœuds de $\widehat{K}$ sont ses sommets (voir (8.13)),

$$\widehat{C}_i,\ 1 \leqslant i \leqslant 4,$$

les milieux de ses arêtes,

$$\begin{aligned}
\widehat{C}_5 &= (\ 0,-1) = \tfrac{1}{2}\left(\widehat{C}_1 + \widehat{C}_2\right),\\
\widehat{C}_6 &= (\ 1,\ 0) = \tfrac{1}{2}\left(\widehat{C}_2 + \widehat{C}_3\right),\\
\widehat{C}_7 &= (\ 0,\ 1) = \tfrac{1}{2}\left(\widehat{C}_3 + \widehat{C}_4\right),\\
\widehat{C}_8 &= (-1,\ 0) = \tfrac{1}{2}\left(\widehat{C}_4 + \widehat{C}_1\right)
\end{aligned}$$

et son centre

$$\widehat{C}_9 = (0,0).$$

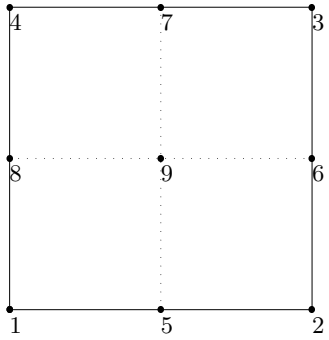

Figure 27: Numérotation des nœuds de $Q_2(\widehat{K})$.

Avec les fonctions $\alpha_1, \alpha_2, \alpha_3$ définies en (8.8), les fonctions de base de $Q_2(\widehat{K})$ (de dimension 9) s'expriment comme suit.

$$\begin{array}{lll}
\widehat{\psi}_1^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_1(\hat{x}_2), & \widehat{\psi}_4^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_3(\hat{x}_2), & \widehat{\psi}_7^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_3(\hat{x}_2),\\
\widehat{\psi}_2^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_1(\hat{x}_2), & \widehat{\psi}_5^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_1(\hat{x}_2), & \widehat{\psi}_8^{(2)}(\hat{x}) = \alpha_1(\hat{x}_1)\alpha_2(\hat{x}_2),\\
\widehat{\psi}_3^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_3(\hat{x}_2), & \widehat{\psi}_6^{(2)}(\hat{x}) = \alpha_3(\hat{x}_1)\alpha_2(\hat{x}_2), & \widehat{\psi}_9^{(2)}(\hat{x}) = \alpha_2(\hat{x}_1)\alpha_2(\hat{x}_2).
\end{array}$$

Comme dans la section précédente, si $S_1, \ldots, S_{27}$ désignent les nœuds d'un élément T de τ_h de type Q_2, dont la face supérieure est contenue dans Γ_s, alors $K = T \cap \Gamma_s$ est un élément bidimensionnel de type Q_2 avec, d'après (8.10),

$$\begin{array}{llllll}
C_1 = S_5, & C_2 = S_6, & C_3 = S_7,\\
C_4 = S_8, & C_5 = S_{13}, & C_6 = S_{14}, & (8.14)\\
C_7 = S_{15}, & C_8 = S_{16}, & C_9 = S_{23}
\end{array}$$

(voir la numérotation des figures 25 et 27).

8.3 Formulation faible approchée (cadre général)

Afin de traiter les problèmes faibles du chapitre précédent avec des méthodes d'éléments finis, considérons un cas plus général.

Soit $\widetilde{H}$ un espace de Hilbert de fonctions de Ω dans $\mathbb{R}$ et H le sous–espace fermé de $\widetilde{H}$ formé des fonctions de $\widetilde{H}$ valant 0 sur une partie Γ_0 de $\partial\Omega$; soit $a : \widetilde{H} \times \widetilde{H} \to \mathbb{R}$ une forme bilinéaire, continue et coercitive et $l \in \widetilde{H}'$ (*i.e.* $l : \widetilde{H} \to \mathbb{R}$ une forme linéaire continue). Considérons le problème faible

$$\text{Trouver } u \in H \text{ vérifiant } a(u,\varphi) = \langle l, \varphi \rangle \text{ pour tout } \varphi \in H. \tag{8.15}$$

Ce problème admet une unique solution par le théorème de Lax–Milgram (7.9). Montrons comment des éléments finis de type Q_m permettent de résoudre (8.15) de manière approchée. Pour un élément T du maillage τ_h et une fonction φ définie sur $\overline{\Omega}$, notons

$$\widehat{\varphi}_T^{(m)} = \widehat{\varphi}_T = \varphi \circ F_T : \widehat{T} \longrightarrow \mathbb{R},$$

où $F_T = F_T^{(m)}$ est définie en (8.1). Considérons l'espace

$$\widetilde{V}_{h,m} = \widetilde{V}_{h,m}(\Omega) = \left\{ \varphi \in \mathcal{C}(\overline{\Omega}) \ : \ \widehat{\varphi}_T \in Q_m(\widehat{T}),\ \forall T \in \tau_h \right\} \tag{8.16}$$

supposé inclus dans $\widetilde{H}$. Pour $T \in \tau_h$, notons les nœuds de T

$$S_{ijk}^{(m)}(T) = S_{ijk}(T) = F_T\left(\widehat{S}_{ijk}\right),\ 0 \leqslant i,j,k \leqslant m.$$

Si $\varphi \in \widetilde{V}_{h,m}$, alors, pour $T \in \tau_h$,

$$\widehat{\varphi}_T = \sum_{i,j,k=0}^{m} \widehat{\varphi}_T\left(\widehat{S}_{ijk}\right) \widehat{\varphi}_{ijk} = \sum_{i,j,k=0}^{m} \varphi\left(S_{ijk}(T)\right) \widehat{\varphi}_{ijk},$$

i.e. φ est entièrement déterminée par les valeurs prises aux nœuds du maillage τ_h. Considérons une numérotation des nœuds de τ_h,

$$\bigcup_{T \in \tau_h} \{S_{ijk}(T),\ 0 \leqslant i,j,k \leqslant m\} = \{R_1, \ldots, R_N\} \tag{8.17}$$

et, pour $1 \leqslant l \leqslant N$, notons φ_l la fonction de $\widetilde{V}_{h,m}$ vérifiant

$$\varphi_l(R_j) = \delta_{lj},\ 1 \leqslant j \leqslant N,$$

c'est–à–dire, pour $T \in \tau_h$,

$$\widehat{\varphi}_{lT} = \begin{cases} \widehat{\varphi}_{ijk} \text{ si } R_l = S_{ijk}(T), \\ 0 \quad \text{ si } R_l \notin T. \end{cases} \tag{8.18}$$

Ainsi, $\{\varphi_l\}_{l=1}^N$ est une base de $\widetilde{V}_{h,m}$ dans laquelle toute $\varphi \in \widetilde{V}_{h,m}$ s'écrit

$$\varphi = \sum_{l=1}^{N} \varphi(R_l)\varphi_l.$$

Supposons que la numérotation des nœuds (8.17) est telle que les nœuds $R_1, \ldots, R_L$ n'appartiennent pas à Γ_0 et les nœuds $R_{L+1}, \ldots, R_{L+M} = R_N$ sont sur Γ_0, la partie de $\partial\Omega$ où la valeur des fonctions de H est fixée à 0. Considérons le sous–espace

$$V_{h,m} = V_{h,m}(\Omega, \Gamma_0) = \left\{ \varphi \in \widetilde{V}_{h,m} \ : \ \varphi(R_{L+j}) = 0, \ 1 \leqslant j \leqslant M \right\} \tag{8.19}$$

de $\widetilde{V}_{h,m}$ et la base $\{\varphi_l\}_{l=1}^{L}$ de $V_{h,m}$. Supposons que $V_{h,m}$ soit un sous–espace de H et remplaçons le problème (8.15) par le problème faible approché

$$\text{Trouver } u_{h,m} \in V_{h,m} \text{ vérifiant } a(u_{h,m}, \varphi) = \langle l, \varphi \rangle \text{ pour tout } \varphi \in V_{h,m}. \tag{8.20}$$

La fonction inconnue s'écrit

$$u_{h,m} = \sum_{l=1}^{L} \xi_l \varphi_l$$

où $\xi_l = u_{h,m}(R_l)$ sont les valeurs à déterminer. Le problème (8.20) est équivalent à trouver $\xi_1, \ldots, \xi_L$ tels que

$$\sum_{l=1}^{L} \xi_l a(\varphi_l, \varphi_j) = \langle l, \varphi_j \rangle \, , \ 1 \leqslant j \leqslant L,$$

ce qui se traduit matriciellement par le système linéaire

$$A\xi = b$$

avec $\xi = (\xi_1, \ldots, \xi_L)^t \in \mathbb{R}^L$ et la matrice $A = (a_{jl})$ de dimension $L \times L$ et le vecteur $b = (b_1, \ldots, b_L)^t$ de $\mathbb{R}^L$ définis par

$$a_{jl} = a(\varphi_l, \varphi_j), \ 1 \leqslant j, l \leqslant L, \tag{8.21}$$

$$b_j = \langle l, \varphi_j \rangle \ 1 \leqslant j \leqslant L. \tag{8.22}$$

La matrice A est appelée *matrice de rigidité*. Comme la forme bilinéaire a est supposée coercitive, la matrice A est définie positive, *i.e.* $(A\xi \,|\, \xi) > 0$ si $\xi \neq 0$, donc A est régulière et le système $A\xi = b$ admet une unique solution. De plus, si la forme bilinéaire a est symétrique, alors la matrice A est SDP (c'est le cas pour les formulations faibles du chapitre précédent).

Remarque 8.2 *Si les conditions sur Γ_0 sont non homogènes, c'est–à–dire si le problème à résoudre est*

$$\textit{Trouver } v \in \widetilde{H} \textit{ vérifiant } a(v, \varphi) = \langle l, \varphi \rangle \textit{ pour tout } \varphi \in H,$$

avec $v = v_0$ sur Γ_0, où $v_0 : \Gamma_0 \to \mathbb{R}$ est donnée (non nulle), alors, pour se ramener à ce qui précède, il suffit de prolonger v_0 en une fonction $\tilde{v}_0$ de $\widetilde{H}$ et de poser $u = v - \tilde{v}_0 \in H$. Nous considérons alors le problème avec conditions homogènes sur Γ_0 suivant :

$$\textit{Trouver } u \in H \textit{ vérifiant } a(u, \varphi) = \langle l, \varphi \rangle - a(\tilde{v}_0, \varphi) \textit{ pour tout } \varphi \in H.$$

Nous considérons le sous–espace $\widetilde{V}_{h,m}$ *de* $\widetilde{H}$*, le sous–espace* $V_{h,m}$ *de* H *et la fonction*

$$\tilde{v}_{0_{h,m}} = \sum_{i=1}^{M} v_0(R_{L+i})\varphi_{L+i} \in \widetilde{V}_{h,m}.$$

Comme précédemment, nous résolvons matriciellement le problème faible approché

Trouver $u_{h,m} \in V_{h,m}$ *vérifiant* $a(u_{h,m}, \varphi) = \langle l, \varphi \rangle - a(\tilde{v}_{0_{h,m}}, \varphi)\ \forall \varphi \in V_{h,m}$.

La matrice du système est la même qu'avant (voir (8.21)) et le second membre est donné par

$$b_j = \langle l, \varphi_j \rangle - \sum_{i=1}^{M} v_0(R_{L+i}) a(\varphi_{L+i}, \varphi_j),\ 1 \leqslant j \leqslant L.$$

Nous obtenons alors la solution approchée

$$v_{h,m} = u_{h,m} + \tilde{v}_{0_{h,m}} = \sum_{l=1}^{L} \xi_l \varphi_l + \sum_{i=1}^{M} v_0(R_{L+i})\varphi_{L+i}.$$

8.3.1 Calcul de la matrice de rigidité et du second membre

D'après les formulations faibles du chapitre précédent, supposons que les applications a et l sont de la forme

$$\begin{aligned} a(\varphi, \psi) &= \int_\Omega g(\varphi, \nabla\varphi, \psi, \nabla\psi)\mathrm{d}x, \\ \langle l, \varphi \rangle &= \int_\Omega h(v, \nabla v, \varphi, \nabla\varphi)\mathrm{d}x + \int_{\Gamma_s} h_s(\Theta, \varphi)\mathrm{d}\sigma, \end{aligned}$$

où v, Θ sont des fonctions de x connues et $g, h : \mathbb{R} \times \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^3 \to \mathbb{R}$ et $h_s : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ sont des fonctions intégrables. De plus, lorsque dans le second terme de la forme l nous avons $h_s \neq 0$, nous supposons que la surface du bassin (renormalisé) Γ_s ne coupe pas Γ_0 (la partie de $\partial\Omega$ à valeur fixée). Les coefficients de la matrice de rigidité (8.21) et les composantes du second membre (8.22) s'écrivent

$$\begin{aligned} a_{jl} &= a(\varphi_l, \varphi_j) = \int_\Omega g(\varphi_l, \nabla\varphi_l, \varphi_j, \nabla\varphi_j)\mathrm{d}x \\ &= \sum_{T \in \tau_h} \int_T g(\varphi_l, \nabla\varphi_l, \varphi_j, \nabla\varphi_j)\mathrm{d}x, \qquad (8.23) \\ b_j &= \langle l, \varphi_j \rangle = \int_\Omega h(v, \nabla v, \varphi_j, \nabla\varphi_j)\mathrm{d}x + \int_{\Gamma_s} h_s(\Theta, \varphi_j)\mathrm{d}\sigma \\ &= \sum_{T \in \tau_h} \left(\int_T h(v, \nabla v, \varphi_j, \nabla\varphi_j)\mathrm{d}x + \int_{T \cap \Gamma_s} h_s(\Theta, \varphi_j)\mathrm{d}\sigma \right), \qquad (8.24) \end{aligned}$$

pour $1 \leqslant j, l \leqslant L$. Les calculs de la matrice A et du vecteur b se font avec les égalités ci–dessus, c'est–à–dire en sommant la contribution de chaque élément du maillage.

Travaillons avec des éléments de type Q_m et fixons, pour l'instant, un élément $T \in \tau_h$. Numérotons localement $S_1(T), \ldots, S_I(T)$ les nœuds de cet élément, avec $I = (m+1)^3$. Considérons l'application qui fait le lien entre cette numérotation locale et la numérotation globale des nœuds du maillage (8.17) :

$$\alpha_T : \{1, \ldots, I\} \longrightarrow \{1, \ldots, N\}$$

définie par

$$S_i(T) = R_{\alpha_T(i)},\ 1 \leqslant i \leqslant I.$$

La fonction de base de $\widetilde{V}_{h,m}$ valant 1 au nœud $S_i(T)$ est aussi notée

$$\varphi_{S_i(T)} = \varphi_{\alpha_T(i)},\ 1 \leqslant i \leqslant I.$$

La *matrice de rigidité élémentaire* A_T (de T), de dimension $I \times I$, est définie par

$$(A_T)_{ik} = \int_T g(\varphi_{S_k(T)}, \nabla\varphi_{S_k(T)}, \varphi_{S_i(T)}, \nabla\varphi_{S_i(T)})\mathrm{d}x,\ 1 \leqslant i, k \leqslant I. \qquad (8.25)$$

Pour le second membre, le premier terme de la parenthèse de (8.24) se traite de la même manière : le *premier second membre élémentaire* b_T (de T) dans $\mathbb{R}^I$, est défini par

$$(b_T)_i = \int_T h(v, \nabla v, \varphi_{S_i(T)}, \nabla\varphi_{S_i(T)})\mathrm{d}x,\ 1 \leqslant i \leqslant I. \qquad (8.26)$$

Pour le second terme de la parenthèse de (8.24), lorsque la face supérieure de T est contenu dans la surface Γ_s, $T \cap \Gamma_s = K(T)$ est un élément bidimensionnel du maillage $\tau_{s,h}$ de la surface Γ_s, induit par le maillage τ_h de Ω. Notons $K = K(T)$ pour alléger l'écriture et, comme avant, numérotons localement $C_1(K), \ldots, C_J(K)$, $J = (m+1)^2$, les nœuds de cet élément et considérons l'application faisant le lien entre cette numérotation locale et la numérotation globale des nœuds :

$$\alpha_K : \{1, \ldots, J\} \longrightarrow \{1, \ldots, N\},\ C_j(K) = R_{\alpha_K(j)},\ 1 \leqslant j \leqslant J.$$

Le *deuxième second membre élémentaire* b_K (de K) dans $\mathbb{R}^J$ est défini par

$$(b_K)_j = \int_K h_s(\Theta, \varphi_{C_j(K)})\mathrm{d}x,\ 1 \leqslant j \leqslant J, \qquad (8.27)$$

où $\varphi_{C_j(K)} = \varphi_{\alpha_K(j)},\ 1 \leqslant j \leqslant J$.

La matrice de rigidité A et le second membre b s'obtiennent alors avec l'algorithme 13.

Calcul de A et b

Initialisation : $A = 0$ et $b = 0$ ($a_{ij} = b_i = 0$, $1 \leqslant i, j \leqslant L$)
Pour $T \in \tau_h$, faire :
  Calcul de la matrice élémentaire A_T
  Pour $1 \leqslant i, k \leqslant I$, faire :
    Si $\alpha_T(i), \alpha_T(k) \leqslant L$:
      $a_{\alpha_T(i),\alpha_T(k)} = a_{\alpha_T(i),\alpha_T(k)} + (A_T)_{ik}$
    Fin si
  Fin pour i, k
  Calcul du premier second membre élémentaire b_T
  Pour $1 \leqslant i \leqslant I$, faire :
    Si $\alpha_K(i) \leqslant L$:
      $b_{\alpha_T(i)} = b_{\alpha_T(i)} + (b_T)_i$
    Fin si
  Fin pour i
  Si $T \cap \Gamma_s = K \neq \emptyset$:
    Calcul du deuxième second membre élémentaire b_K
    Pour $1 \leqslant j \leqslant J$, faire[1] :
      $b_{\alpha_K(j)} = b_{\alpha_K(j)} + (b_K)_j$
    Fin pour i
  Fin si
Fin pour T.

Algorithme 13.

Pour calculer les intégrales (8.25) et (8.26), effectuons le changement de variable $x = F_T(\hat{x})$ pour se ramener sur l'élément de référence tridimensionnel $\widehat{T}$. D'après (8.18) et avec la numérotation locale des nœuds de T, il suit, pour $1 \leqslant i \leqslant I$,

$$\varphi_{S_i(T)} \circ F_T = \widehat{\varphi_{S_i(T)}}_T = \widehat{\varphi}_i.$$

Par conséquent, nous avons

$$\nabla\widehat{\varphi}_i(\hat{x}) = \nabla(\varphi_{S_i(T)} \circ F_T)(\hat{x}) = (J_{F_T}(\hat{x}))^t \, \nabla\varphi_{S_i(T)}(F_T(\hat{x})),$$

c'est–à–dire,

$$\left(\nabla\varphi_{S_i(T)}\right) \circ F_T = (J_{F_T})^{-t} \, \nabla\widehat{\varphi}_i,$$

où J_{F_T} est la matrice jacobienne de la transformation F_T. Nous obtenons ainsi, en supposant que $\det J_{F_T} > 0$,

$$(A_T)_{ik} = \int_{\widehat{T}} g(\widehat{\varphi}_k, (J_{F_T})^{-t} \nabla\widehat{\varphi}_k, \widehat{\varphi}_i, (J_{F_T})^{-t} \nabla\widehat{\varphi}_i) \det J_{F_T} \mathrm{d}\hat{x} \tag{8.28}$$

pour $1 \leqslant i, k \leqslant I$ et

$$(b_T)_i = \int_{\widehat{T}} h(v \circ F_T, (\nabla v) \circ F_T, \widehat{\varphi}_i, (J_{F_T})^{-t} \nabla\widehat{\varphi}_i) \det J_{F_T} \mathrm{d}\hat{x} \tag{8.29}$$

pour $1 \leqslant i \leqslant I$.

[1] $\Gamma_s \cap \Gamma_0$ étant supposé vide, nous avons nécessairement $\alpha_K(j) \leqslant L$.

De manière similaire, pour calculer l'intégrale (8.27), le changement de variable $x = F_K^{(m)}(\hat{x}) = F_K(\hat{x})$ envoyant l'élément de référence bidimensionnel $\widehat{K}$ sur K est utilisé. Notons $\widehat{\psi}_r$ la fonction de base de $Q_m(\widehat{K})$ valant 1 au point $\widehat{C}_r$ de $\widehat{K}$ et $C_r(K) = F_K(\widehat{C}_r)$, $1 \leqslant r \leqslant J$. D'après (8.11) et (8.12), il suit

$$\varphi_{C_j(K)} \circ F_K = \varphi_{C_j(K)} \circ F_T|_{\widehat{T}\cap\{\hat{x}_3=1\}} = \widehat{\psi}_j,\ 1 \leqslant j \leqslant J.$$

En supposant que la matrice jacobienne J_{F_K} de la transformation F_K vérifie $\det J_{F_K} > 0$, (8.27) devient

$$(b_K)_j = \int_{\widehat{K}} h_s(\Theta \circ F_K, \widehat{\psi}_j) \det J_{F_K} \mathrm{d}\hat{x},\ 1 \leqslant j \leqslant J. \tag{8.30}$$

Les intégrales (8.28), (8.29) et (8.30) peuvent par exemple être estimées avec la formule de quadrature de Gauss–Legendre (voir annexe B). Cette formule avec n points est de la forme

$$\int_{-1}^{1} f(t)\mathrm{d}t \approx \sum_{k=1}^{n} a_k f(t_k),$$

où $t_k \in (-1,1)$ et $a_k > 0$, $1 \leqslant k \leqslant n$. Comme $\widehat{T} = [-1,1]\times[-1,1]\times[-1,1]$ et $\widehat{K} = [-1,1]\times[-1,1]$, nous obtenons

$$\int_{\widehat{T}} f(\hat{x}_1,\hat{x}_2,\hat{x}_3)\mathrm{d}\hat{x} \approx \sum_{i,j,k=1}^{n} a_i a_j a_k f(t_i,t_j,t_k)$$

et

$$\int_{\widehat{K}} f(\hat{x}_1,\hat{x}_2)\mathrm{d}\hat{x} \approx \sum_{i,j=1}^{n} a_i a_j f(t_i,t_j).$$

8.4 Formulations faibles approchées des équations de Navier–Stokes

Les formulations faibles (7.60), (7.63) et (7.65) (obtenues par la méthode de prédicteur–correcteur (II)) sont traitées ici en détails avec la méthode de la section précédente.

Considérons un maillage de Ω formé d'éléments finis de type Q_1. En suivant le procédé de la section 8.2.4, les éléments du maillage peuvent être considérés de type Q_2. Les vitesses horizontales (*i.e.* les deux premières composantes de la vitesse) sont calculées avec des éléments finis de type Q_2 et la vitesse verticale (*i.e.* la troisième composante de la vitesse) ainsi que la pression avec des éléments finis de type Q_1. Ces choix sont motivés à la section 8.5.

Pour le calcul de la pression, d'après (7.59), considérons la forme bilinéaire

$$a_p : H^1(\Omega)\times H^1(\Omega) \longrightarrow \mathbb{R},\ a_p(\varphi,\psi) = \int_\Omega (\nabla_\mu\varphi \,|\, \nabla_\mu\psi)\,\mathrm{d}x \tag{8.31}$$

et la forme linéaire

$$l_p : H^1(\Omega) \longrightarrow \mathbb{R},\ \langle l_p, \varphi\rangle = \int_\Omega \left(v^k \,|\, \nabla_\mu \varphi \right) \mathrm{d}x, \tag{8.32}$$

où v^k est défini en (7.56). Nous choisissons le sous–espace $\widetilde{V}_{h,1}(\Omega)$ (voir (8.16)) de $H^1(\Omega)$ et considérons le problème faible approché

$$\text{Trouver } q^{k+1} \in \widetilde{V}_{h,1}(\Omega) \text{ vérifiant } a_p(q^{k+1}, \varphi) = \langle l_p, \varphi\rangle \ \ \forall \varphi \in \widetilde{V}_{h,1}(\Omega). \tag{8.33}$$

N'ayant ici pas travaillé avec l'espace quotient $H^1(\Omega)/\mathbb{R}$, il faudra adapter la formulation trouvée pour garantir l'unicité de la solution, voir section 8.4.3. De plus, une méthode de pénalisation pour le calcul de la pression est présentée à la section 8.4.4.

Pour le calcul des vitesses, d'après (7.62) et (7.64), considérons les formes bilinéaires

$$\begin{aligned} &a_i : V_i \times V_i \longrightarrow \mathbb{R}, \\ &\quad a_i(\varphi, \psi) = \frac{3}{2\delta t} \int_\Omega \varphi\psi \mathrm{d}x + \int_\Omega \left(\nabla\varphi \,|\, \nabla\psi \right)_\lambda \mathrm{d}x,\ i = 1, 2, 3, \end{aligned} \tag{8.34}$$

et les formes linéaires

$$l_i : V_i \longrightarrow \mathbb{R},\ \ \langle l_i, \varphi\rangle = \int_\Omega w_i^k \varphi \mathrm{d}x + \int_{\Gamma_s} \Theta_i^{k+1} \varphi \mathrm{d}\sigma,\ i = 1, 2, \tag{8.35}$$

$$l_3 : V_3 \longrightarrow \mathbb{R},\ \ \langle l_3, \varphi\rangle = \int_\Omega w_3^k \varphi \mathrm{d}x, \tag{8.36}$$

où les espaces $V_i = V_i(\Omega, \Gamma_i)$, $i = 1, 2$, sont définis en (7.16), l'espace $V_3 = V_3(\Omega, \Gamma_3 \cup \Gamma_s)$ en (7.20) et le vecteur w^k en (7.61).
Pour $i = 1, 2$, nous choisissons le sous–espace $V_{h,2}(\Omega, \Gamma_i)$ (voir (8.19)) de $V_i = V(\Omega, \Gamma_i)$ et posons le problème faible approché

$$\text{Trouver } u_i^{k+1} \in V_{h,2}(\Omega, \Gamma_i) \text{ vérifiant } a_i(u_i^{k+1}, \varphi) = \langle l_i, \varphi\rangle \ \ \forall \varphi \in V_{h,2}(\Omega, \Gamma_i). \tag{8.37}$$

Pour $i = 3$, le sous–espace $V_{h,1}(\Omega, \Gamma_3 \cup \Gamma_s)$ (voir (8.19)) de $V_3 = V(\Omega, \Gamma_3 \cup \Gamma_s)$ et le problème faible approché

$$\begin{aligned} &\text{Trouver } \tilde{u}_3^{k+1} \in V_{h,1}(\Omega, \Gamma_3 \cup \Gamma_s) \text{ vérifiant} \\ &\quad a_3(\tilde{u}_3^{k+1}, \varphi) = \langle l_3, \varphi\rangle \ \ \forall \varphi \in V_{h,1}(\Omega, \Gamma_3 \cup \Gamma_s) \end{aligned} \tag{8.38}$$

sont considérés.

Pour $m = 1, 2$, notons $I_m = (m+1)^3$ (respectivement $J_m = (m+1)^2$) la dimension de $Q_m(\widehat{T})$ (resp. $Q_m(\widehat{K})$) et utilisons les notations des sections précédentes. D'après le passage du type Q_1 au type Q_2 présenté dans la section 8.2.4, nous avons $F_T = F_T^{(1)} = F_T^{(2)}$ et de (8.12), il suit $F_K = F_K^{(1)} = F_K^{(2)}$. Les nœuds d'un élément T sont notés $S_r(T) = F_T(\widehat{S}_r)$, $1 \leqslant r \leqslant 27$; si la face supérieure de T est contenue dans la surface Γ_s, les nœuds de $K = T \cap \Gamma_s$ sont aussi notés $C_r(T) = F_K(\widehat{C}_r)$, $1 \leqslant r \leqslant 9$, avec les relations (8.14).

Les intégrales obtenues pour les matrices de rigidité et seconds membres élémentaires présentés ci–dessous seront calculées avec la formule de quadrature de Gauss–Legendre à n points avec $n = 2$ lorsque des éléments finis de type Q_1 sont employés et avec $n = 3$ si les éléments finis sont de type Q_2 (voir section 8.5).

8.4.1 Matrices de rigidité élémentaires

Notons $A_T^{(p)}$ la matrice de rigidité élémentaire du problème faible approché pour la pression et, pour $i = 1, 2, 3$, $A_T^{(i)}$ celle du problème faible approché pour la i–ème composante de la vitesse. Leurs coefficients se calculent avec (8.28). C'est–à–dire, d'après la forme bilinéaire (8.31), nous avons

$$\left(A_T^{(p)}\right)_{jk} = \int_{\widehat{T}} \left((J_{F_T})^{-t} \nabla \widehat{\varphi}_k^{(1)} \,|\, (J_{F_T})^{-t} \nabla \widehat{\varphi}_j^{(1)} \right)_{\mu^2} \det J_{F_T} \mathrm{d}\hat{x}, \ 1 \leqslant j, k \leqslant 8 \tag{8.39}$$

et, d'après (8.34),

$$\begin{aligned} \left(A_T^{(i)}\right)_{jk} &= \frac{3}{2\delta t} \int_{\widehat{T}} \widehat{\varphi}_k^{(m)} \widehat{\varphi}_j^{(m)} \det J_{F_T} \mathrm{d}\hat{x} \\ &\quad + \int_{\widehat{T}} \left((J_{F_T})^{-t} \nabla \widehat{\varphi}_k^{(m)} \,|\, (J_{F_T})^{-t} \nabla \widehat{\varphi}_j^{(m)} \right)_{\lambda} \det J_{F_T} \mathrm{d}\hat{x} \end{aligned}$$

pour $1 \leqslant j, k \leqslant I_m$, avec $m = 2$ ($I_m = 27$) pour $i = 1, 2$ et $m = 1$ ($I_m = 8$) pour $i = 3$.

Remarque 8.3 *Pour la méthode de prédicteur–correcteur (I) avec la formule de différentiation rétrograde à deux pas, les matrices de rigidités (élémentaires) pour le calcul de la pression et des vitesses (à l'étape de prédiction) sont les mêmes que ci–dessus.*

8.4.2 Seconds membres élémentaires

Pression

Le vecteur $v^k = (v_1^k, v_2^k, v_3^k)^t$ apparaissant dans la forme linéaire l_p (8.32) est donné par (voir (7.56))

$$\begin{aligned} v_i^k &= -\frac{1}{2\delta t}(-7u_i^k + 5u_i^{k-1} - u_i^{k-2}) \\ &\quad - \left(u^k \,|\, \nabla_\mu u_i^k \right) + \left(u^{k-1} \,|\, \nabla_\mu u_i^{k-1} \right) + f_i^k - f_i^{k-1}, \ i = 1, 2, 3, \end{aligned}$$

avec, d'après (6.4),

$$\begin{aligned} f_1^l &= -2(\omega \wedge u^l)_1 = 2\omega_3(u_2^l \sin\alpha - u_3^l \cos\alpha), \\ f_2^l &= -2(\omega \wedge u^l)_2 = -2\omega_3 u_1^l \sin\alpha, \\ f_3^l &= -2(\omega \wedge u^l)_3 = 2\omega_3 u_1^l \cos\alpha, \end{aligned}$$

$\alpha = \alpha(x)$ étant la latitude du point x et ω_3 la vitesse de rotation de la Terre.

D'après (8.29), le second membre élémentaire, $b_T^{(p)} \in \mathbb{R}^8$, est donné par

$$\begin{aligned}\left(b_T^{(p)}\right)_j &= \int_{\widehat{T}} \left(v^k \circ F_T \,|\, (J_{F_T})^{-t} \nabla \widehat{\varphi}_j^{(1)} \right)_\mu \det J_{F_T} \mathrm{d}\hat{x} \qquad (8.40)\\ &= \sum_{i=1}^{3} \mu_i \left[-\frac{1}{2\delta t} \left(-7 \left(b_T^{(p,\mathrm{a},k,i)}\right)_j + 5 \left(b_T^{(p,\mathrm{a},k-1,i)}\right)_j - \left(b_T^{(p,\mathrm{a},k-2,i)}\right)_j \right) \right.\\ &\quad \left. - \left(b_T^{(p,\mathrm{b},k,i)}\right)_j + \left(b_T^{(p,\mathrm{b},k-1,i)}\right)_j + \left(b_T^{(p,\mathrm{c},k,i)}\right)_j - \left(b_T^{(p,\mathrm{c},k-1,i)}\right)_j \right],\end{aligned}$$

où

$$\begin{aligned}\left(b_T^{(p,\mathrm{a},l,i)}\right)_j &= \int_{\widehat{T}} \left(u_i^l \circ F_T\right) \Phi_i \det J_{F_T} \mathrm{d}\hat{x},\\ \left(b_T^{(p,\mathrm{b},l,i)}\right)_j &= \int_{\widehat{T}} \left(u^l \circ F_T \,|\, (\nabla u_i^l) \circ F_T \right)_\mu \Phi_i \det J_{F_T} \mathrm{d}\hat{x},\\ \left(b_T^{(p,\mathrm{c},l,i)}\right)_j &= \int_{\widehat{T}} \left(f_i^l \circ F_T\right) \Phi_i \det J_{F_T} \mathrm{d}\hat{x},\end{aligned}$$

avec

$$\Phi_i = \left[(J_{F_T})^{-t} \nabla \widehat{\varphi}_j^{(1)} \right]_i .$$

Comme $u_1^l \circ F_T \in Q_2(\widehat{T})$, $u_2^l \circ F_T \in Q_2(\widehat{T})$ et $u_3^l \circ F_T \in Q_1(\widehat{T}) \subset Q_2(\widehat{T})$, nous avons $u^l \circ F_T \in \left(Q_2(\widehat{T})\right)^3$,

$$\begin{aligned}u^l \circ F_T &= \sum_{r=1}^{27} u^l(S_r(T)) \widehat{\varphi}_r^{(2)},\\ \left(\nabla u_i^l\right) \circ F_T &= \sum_{r=1}^{27} u_i^l(S_r(T)) \, (J_{F_T})^{-t} \nabla \widehat{\varphi}_r^{(2)}, \ i = 1,2,3,\end{aligned}$$

Remarque 8.4 *Si y est une fonction telle que $y \circ F_T \in Q_1(\widehat{T})$, ses valeurs aux 8 sommets de l'élément sont connus. Nous pouvons écrire $y \circ F_T$ dans la base de $Q_2(\widehat{T})$,*

$$y \circ F_T = \sum_{r=1}^{27} y(S_r(T)) \widehat{\varphi}_r^{(2)},$$

en calculant au préalable

$$y(S_{r_2}(T)) = \sum_{r_1=1}^{8} y(S_{r_1}(T)) \widehat{\varphi}_{r_1}^{(1)}(S_{r_2}(T)), \ 9 \leqslant r_2 \leqslant 27.$$

La latitude étant presque constante sur un même élément T, nous supposons que $f_i^l \circ F_T \in Q_2(\widehat{T})$ et écrivons ainsi

$$f_i^l \circ F_T = \sum_{r=1}^{27} f_i^l(S_r(T)) \widehat{\varphi}_r^{(2)}, \ i = 1,2,3.$$

Nous obtenons alors

$$\begin{aligned}
\left(b_T^{(p,\mathrm{a},l,i)}\right)_j &= \sum_{r=1}^{27} u_i^l(S_r(T)) \int_{\widehat{T}} \widehat{\varphi}_r^{(2)} \Phi_i \det J_{F_T} \mathrm{d}\hat{x}, \\
\left(b_T^{(p,\mathrm{b},l,i)}\right)_j &= \sum_{r_1,r_2=1}^{27} u_i^l(S_{r_2}(T)) \cdot \\
&\quad \int_{\widehat{T}} \left(u^l(S_{r_1}(T)) \,|\, (J_{F_T})^{-t} \nabla \widehat{\varphi}_{r_2}^{(2)} \right)_\mu \widehat{\varphi}_{r_1}^{(2)} \Phi_i \det J_{F_T} \mathrm{d}\hat{x}, \\
\left(b_T^{(p,\mathrm{c},l,i)}\right)_j &= \sum_{r=1}^{27} f_i^l(S_r(T)) \int_{\widehat{T}} \widehat{\varphi}_r^{(2)} \Phi_i \det J_{F_T} \mathrm{d}\hat{x}.
\end{aligned}$$

Vitesse

En utilisant la relation (7.27), les formes linéaires l_i, $i = 1, 2$ (8.35) et l_3 (8.36) s'écrivent

$$\begin{aligned}
\langle l_i, \varphi \rangle &= \int_\Omega \tilde{w}_i^k \varphi \mathrm{d}x + \int_\Omega \mu_i p^{k+1} \frac{\partial \varphi}{\partial x_i} \mathrm{d}x + \int_{\Gamma_s} \Theta_i^{k+1} \varphi \mathrm{d}\sigma, \ i = 1, 2, \\
\langle l_3, \varphi \rangle &= \int_\Omega \tilde{w}_3^k \varphi \mathrm{d}x + \int_\Omega \mu_3 p^{k+1} \frac{\partial \varphi}{\partial x_3} \mathrm{d}x,
\end{aligned}$$

où le vecteur $\tilde{w}^k = w^k + \nabla_\mu p^{k+1}$ (voir (7.61)) est donné par

$$\tilde{w}_i^k = -\frac{1}{\delta t} \left(-2u_i^k + \frac{1}{2} u_i^{k-1} \right) - \left(u^k \,|\, \nabla_\mu u_i^k \right) + f_i^k, \ i = 1, 2, 3,$$

avec f_i^k défini comme pour la pression.

Notons $b_T^{(i)}$ (resp. $b_K^{(i)}$) le premier (resp. deuxième) second membre élémentaire pour la i–ème composante de la vitesse, $i = 1, 2, 3$ (resp. $i = 1, 2$). Nous avons $b_T^{(i)} \in \mathbb{R}^{27}$ et $b_K^{(i)} \in \mathbb{R}^9$ pour $i = 1, 2$ et $b_T^{(3)} \in \mathbb{R}^8$.

D'après (8.29), nous avons

$$b_T^{(i)} = -\frac{1}{\delta t} \left(-2b_T^{(i,\mathrm{a},k)} + \frac{1}{2} b_T^{(i,\mathrm{a},k-1)} \right) - b_T^{(i,\mathrm{b},k)} + b_T^{(i,\mathrm{c},k)} + b_T^{(i,\mathrm{d},k+1)},$$

où

$$\begin{aligned}
\left(b_T^{(i,\mathrm{a},l)}\right)_j &= \int_{\widehat{T}} \left(u_i^l \circ F_T \right) \widehat{\varphi}_j^{(m)} \det J_{F_T} \mathrm{d}\hat{x}, \\
\left(b_T^{(i,\mathrm{b},k)}\right)_j &= \int_{\widehat{T}} \left(u^k \circ F_T \,|\, (\nabla u_i^k) \circ F_T \right)_\mu \widehat{\varphi}_j^{(m)} \det J_{F_T} \mathrm{d}\hat{x}, \\
\left(b_T^{(i,\mathrm{c},k)}\right)_j &= \int_{\widehat{T}} \left(f_i^k \circ F_T \right) \widehat{\varphi}_j^{(m)} \det J_{F_T} \mathrm{d}\hat{x}, \\
\left(b_T^{(i,\mathrm{d},k+1)}\right)_j &= \int_{\widehat{T}} \mu_i \left(p^{k+1} \circ F_T \right) \left[(J_{F_T})^{-t} \nabla \widehat{\varphi}_j^{(m)} \right]_i \det J_{F_T} \mathrm{d}\hat{x},
\end{aligned}$$

avec $m = 2$ pour $i = 1, 2$ et $m = 1$ pour $i = 3$.

De la même manière que pour la pression, nous obtenons

$$\begin{aligned}
\left(b_T^{(i,\mathrm{a},l)}\right)_j &= \sum_{r=1}^{27} u_i^l(S_r(T)) \int_{\widehat{T}} \widehat{\varphi}_r^{(2)} \widehat{\varphi}_j^{(m)} \det J_{F_T} \mathrm{d}\hat{x}, \\
\left(b_T^{(i,\mathrm{b},k)}\right)_j &= \sum_{r_1,r_2=1}^{27} u_i^k(S_{r_2}(T)) \cdot \\
&\quad \int_{\widehat{T}} \left(u^k(S_{r_1}(T)) \,|\, (J_{F_T})^{-t} \nabla \widehat{\varphi}_{r_2}^{(2)} \right)_\mu \widehat{\varphi}_{r_1}^{(2)} \widehat{\varphi}_j^{(m)} \det J_{F_T} \mathrm{d}\hat{x}, \\
\left(b_T^{(i,\mathrm{c},k,i)}\right)_j &= \sum_{r=1}^{27} f_i^k(S_r(T)) \int_{\widehat{T}} \widehat{\varphi}_r^{(2)} \widehat{\varphi}_j^{(m)} \det J_{F_T} \mathrm{d}\hat{x}.
\end{aligned}$$

et, comme $p^{k+1} \circ F_T \in Q_1(\widehat{T}) \subset Q_2(\widehat{T})$,

$$\left(b_T^{(i,\mathrm{d},k+1)}\right)_j = \sum_{r=1}^{I_m} \mu_i p^{k+1}(S_r(T)) \int_{\widehat{T}} \widehat{\varphi}_r^{(m)} \left[(J_{F_T})^{-t} \nabla \widehat{\varphi}_j^{(m)} \right]_i \det J_{F_T} \mathrm{d}\hat{x},$$

avec $m = 2$ ($I_m = 27$) pour $i = 1, 2$ et $m = 1$ ($I_m = 8$) pour $i = 3$.

Traitons le deuxième second membre élémentaire pour $i = 1, 2$. Lorsque la face supérieure de T est contenue dans la surface Γ_s, les composantes du deuxième second membre élémentaire sont, d'après (8.30),

$$\left(b_K^{(i)}\right)_j = \int_{\widehat{K}} \left(\Theta_i^{k+1} \circ F_K\right) \widehat{\psi}_j^{(2)} \det J_{F_K} \mathrm{d}\hat{x},$$

pour $i = 1, 2$ avec $K = T \cap \Gamma_s$. En prenant en compte des valeurs de Θ^{k+1} aux nœuds $C_r(K)$, $1 \leqslant r \leqslant 9$ de l'élément de surface K, c'est–à–dire, en supposant que $\Theta^{k+1} \circ F_K \in \left(Q_2(\widehat{K})\right)^2$, nous avons

$$\Theta^{k+1} \circ F_K = \sum_{r=1}^{9} \Theta^{k+1}(C_r(K)) \widehat{\psi}_r^{(2)}.$$

Nous obtenons alors

$$\left(b_K^{(i)}\right)_j = \sum_{r=1}^{9} \Theta_i^{k+1}(C_r(K)) \int_{\widehat{K}} \widehat{\psi}_r^{(2)} \widehat{\psi}_j^{(2)} \det J_{F_K} \mathrm{d}\hat{x}, \ i = 1, 2.$$

8.4.3 Adaptation pour le calcul de la pression

Notons $A^{(p)}$ la matrice de rigidité et $b^{(p)}$ le second membre du problème faible approché (8.33) pour le calcul de la pression. La matrice $A^{(p)}$ est de dimension $N \times N$ où N est la dimension de $\widetilde{V}_{h,1}(\Omega)$ et est symétrique et semi–définie positive, *i.e.* $(A\xi \,|\, \xi) \geqslant 0$ pour tout $\xi \in \mathbb{R}^N$, car la forme bilinéaire a_p (8.31) l'est.

Le problème faible (7.60) possède une unique solution dans $H^1(\Omega)$ à constante additive près, ce qui se traduit sur la matrice $A^{(p)}$ par

$$\operatorname{Ker} A^{(p)} = \langle (1, 1, \ldots, 1)^t \rangle. \tag{8.41}$$

En particulier $A^{(p)}$ est de rang $N-1$. En traçant dans $A^{(p)}$ une ligne et une colonne de même indice, nous obtenons une matrice symétrique définie positive. En effet, notons

$$0 = \lambda_1 < \lambda_2 \leqslant \lambda_3 \leqslant \ldots \leqslant \lambda_N$$

les valeurs propres de $A^{(p)}$ et choisissons une base orthonormée de $\mathbb{R}^N$ formée de vecteurs propres, $\{v_1, \ldots, v_N\}$, avec $A^{(p)} v_i = \lambda_i v_i$, $1 \leqslant i \leqslant N$ et $v_1 = N^{-1/2}(1, \ldots, 1)^t$. Pour un vecteur $v = \sum_{i=1}^N \alpha_i v_i$ de $\mathbb{R}^N$, nous avons

$$\begin{aligned} \left(A^{(p)} v \,|\, v \right) = \sum_{i=1}^N \alpha_i^2 \lambda_i = 0 \quad &\Longleftrightarrow \quad \alpha_i = 0,\ i = 2, \ldots N \\ &\Longleftrightarrow \quad v = \alpha_1 v_1 \in \operatorname{Ker} A^{(p)}. \end{aligned} \tag{8.42}$$

Notons $A_j^{(p)}$ la matrice de dimension $(N-1) \times (N-1)$ obtenue en traçant dans $A^{(p)}$ la j–ème ligne et la j–ème colonne et montrons qu'elle est SDP. Pour $x = (x_1, \ldots, x_{N-1})^t \in \mathbb{R}^{N-1}$, posons $\tilde{x} = (x_1, \ldots, x_{j-1}, 0, x_j, \ldots, x_{N-1})^t \in \mathbb{R}^N$; nous avons alors, par (8.41) et (8.42),

$$\left(A_j^{(p)} x \,|\, x \right) = \left(A^{(p)} \tilde{x} \,|\, \tilde{x} \right) = 0 \iff \tilde{x} \in \operatorname{Ker} A^{(p)} \iff x = 0.$$

La matrice A étant symétrique et semi–définie positive, nous déduisons que $A_j^{(p)}$ est SDP.

Montrons que la somme des composantes du second membre $b^{(p)}$ est nulle. Soit $R_1, \ldots, R_N$ les nœuds du maillage de Ω en éléments de type Q_1 et $\{\varphi_l\}_{l=1}^N$ les fonctions de base de $\widetilde{V}_{h,1}(\Omega)$ où φ_l vaut 1 au nœud R_l et s'annule sur les autres nœuds, $1 \leqslant l \leqslant N$. La somme des fonctions de base,

$$\varphi = \sum_{l=1}^N \varphi_l,$$

est la fonction constante 1. En effet, l'unique fonction de $Q_1(\widehat{T})$ valant 1 en chaque nœud $\widehat{S}_i$ $(1 \leqslant i \leqslant 8)$ de $\widehat{T}$ est la fonction constante 1, donc, pour chaque $T \in \tau_h$,

$$\widehat{\varphi}_T = \varphi \circ F_T = \sum_{i=1}^8 \widehat{\varphi}_i^{(1)} \equiv 1.$$

Comme

$$\left(b^{(p)} \right)_l = \langle l_p, \varphi_l \rangle = \frac{1}{\delta t} \int_\Omega \left(v^k \,|\, \nabla_\mu \varphi_l \right) \mathrm{d}x,$$

il suit

$$\sum_{l=1}^N \left(b^{(p)} \right)_l = \langle l_p, \varphi \rangle = \frac{1}{\delta t} \int_\Omega \left(v^k \,|\, \nabla_\mu \varphi \right) \mathrm{d}x = 0. \tag{8.43}$$

Montrons alors comment procéder pour obtenir la solution du problème faible approché (8.33) à moyenne nulle. Notons $b_j^{(p)}$ le vecteur $b^{(p)}$ privé de sa j–ème composante. Premièrement, nous cherchons l'unique solution de

$$A_j^{(p)}(\zeta_1, \ldots, \zeta_{j-1}, \zeta_{j+1}, \ldots, \zeta_N)^t = b_j^{(p)}.$$

Ainsi, le vecteur $\zeta = (\zeta_1, \ldots, \zeta_{j-1}, 0, \zeta_{j+1}, \ldots, \zeta_N)^t$ vérifie

$$A^{(p)}\zeta = b^{(p)}.$$

Cette dernière égalité est évidente pour les composantes différentes de j. Comme $(1, 1, \ldots, 1)^t \in \operatorname{Ker} A^{(p)}$ et que $A^{(p)}$ est symétrique, la somme des coefficients d'une colonne de $A^{(p)}$ est nulle, *i.e.*

$$\sum_{l=1}^{N} \left(A^{(p)}\right)_{lk} = 0, \ 1 \leqslant k \leqslant N$$

et, avec (8.43), nous obtenons

$$\begin{aligned} \left(A^{(p)}\zeta\right)_j &= \sum_{k \neq j} \left(A^{(p)}\right)_{jk} \zeta_k = -\sum_{l \neq j} \sum_{k \neq j} \left(A^{(p)}\right)_{lk} \zeta_k \\ &= -\sum_{l \neq j} \left(b^{(p)}\right)_l = \left(b^{(p)}\right)_j. \end{aligned}$$

Ensuite, nous calculons la moyenne des composantes du vecteur ζ,

$$M = \frac{1}{N} \sum_{\substack{l=1 \\ l \neq j}}^{N} \zeta_l$$

et nous obtenons la solution ξ à moyenne nulle sur les nœuds du maillage en posant

$$\begin{aligned} \xi_l &= \zeta_l - M, \ 1 \leqslant l \leqslant N, \ l \neq j, \\ \xi_j &= -M. \end{aligned}$$

Remarquons que le choix $j = N$ simplifie le code informatique.

8.4.4 Calcul de la pression avec pénalisation

En utilisant une méthode de pénalisation (selon la méthodologie de la section 7.3.1) pour le calcul de la pression, la forme bilinéaire a_p (8.31) est remplacée par

$$\tilde{a}_p(\varphi, \psi) = a_p(\varphi, \psi) + \varepsilon_p \int_\Omega \varphi\psi \mathrm{d}x$$

et la forme linéaire l_p (8.32) par

$$\left\langle \tilde{l}_p, \varphi \right\rangle = \langle l_p, \varphi \rangle + \varepsilon_p \int_\Omega q^k \varphi \mathrm{d}x$$

dans le problème faible approchée (8.33), ε_p désignant le facteur de pénalisation.

Pour obtenir la matrice de rigidité élémentaire $\widetilde{A}_T^{(p)}$ et le second membre élémentaire $\tilde{b}_T^{(p)}$ dans ce cas, il suffit d'ajouter la contribution de la pénalisation à $A_T^{(p)}$ (donné en (8.39)) et à $b_T^{(p)}$ (donné en (8.40)), *i.e.*

$$\left(\widetilde{A}_T^{(p)}\right)_{jk} = \left(A_T^{(p)}\right)_{jk} + \varepsilon_p \int_{\widehat{T}} \widehat{\varphi}_k^{(1)} \widehat{\varphi}_j^{(1)} \det J_{F_T} \mathrm{d}\hat{x},\ 1 \leqslant j,k \leqslant 8$$

et, avec $q^k \circ F_T \in Q_1(\widehat{T})$,

$$\left(\widetilde{b}_T^{(p)}\right)_j = \left(b_T^{(p)}\right)_j + \varepsilon_p \sum_{r=1}^{8} q^k(S_r(T)) \int_{\widehat{T}} \widehat{\varphi}_r^{(1)} \widehat{\varphi}_j^{(1)} \det J_{F_T} \mathrm{d}\hat{x},\ 1 \leqslant j \leqslant 8.$$

Comme la forme bilinéaire $\tilde{a}_p$ est symétrique et coercitive, la matrice de rigidité $\widetilde{A}^{(p)}$ est SDP et aucune adaptation n'est alors nécessaire.

Remarque 8.5 *En ne pénalisant pas le second membre (voir remarque 7.2), la matrice de rigidité élémentaire se calcule comme ci–dessus et le second membre élémentaire reste* $\tilde{b}_T^{(p)} = b_T^{(p)}$.

8.5 Notes sur le choix des méthodes

L'étude de la convergence des solutions des problèmes faibles approchés vers les solutions des problèmes faibles originaux ne fait pas l'objet de ce travail. Cependant, motivons les choix des éléments finis utilisés.

Diverses méthodes d'éléments finis en mécanique des fluides sont traitées dans [42]. Considérons ici le problème de Stokes stationnaire pour un fluide incompressible isotrope (ν est ici un scalaire)

$$\begin{aligned} -\nu\Delta u + \nabla p &= f && \text{dans } \Omega, \\ \operatorname{div} u &= 0 && \text{dans } \Omega, \\ u &= 0 && \text{sur } \partial\Omega, \end{aligned}$$

dont les discrétisations spatiales peuvent s'appliquer aux équations de Navier–Stokes (voir remarque 6.3). La résolution de ce problème avec des méthodes d'éléments finis mixtes est analysée dans [12, 51, 50, 49]. L'existence et l'unicité de la solution des formulations faibles sont obtenues à l'aide de l'inégalité *inf–sup* (due à Brezzi et Babuška [6]), voir aussi [24]. Des estimations des erreurs sont données pour des éléments finis de type Q_2 pour la vitesse et de type Q_1 pour la pression (élément de Taylor–Hood (voir [17, 24])) ; dans [12], il est montré que l'erreur est d'ordre h^2 pour la vitesse et d'ordre h pour la pression, h étant le maximum des diamètres des éléments du maillage.

D'autre part, le vecteur de viscosité cinématique turbulente est choisi de sorte que les équations de Navier–Stokes vérifient asymptotiquement l'approximation hydrostatique (voir section 6.6) qui admet une solution faible où la vitesse verticale est moins régulière que les vitesses horizontales (seule la dérivée partielle

par rapport à la troisième variable est dans L^2 pour la vitesse verticale), voir [15].

Ainsi, par ce qui précède, il semble raisonnable d'utiliser des éléments finis de type Q_2 pour les vitesses horizontales et de type Q_1 pour la vitesse verticale et la pression (voir aussi [5] pour l'approximation hydrostatique).

De plus, si la convergence est d'ordre h^k, elle n'est pas altérée par l'utilisation de la formule de quadrature de Gauss–Legendre à $k+1$ points [21, chap. 4]. Ainsi, il est raisonnable d'utiliser la formule de Gauss–Legendre à 3 points (resp. 2 points) lorsque les éléments finis sont de type Q_2 (resp. Q_1).

CHAPITRE 9

Un écoulement tridimensionnel dans l'océan Atlantique nord

Dans ce chapitre, nous présentons un écoulement tridimensionnel dans l'océan Atlantique nord. Pour l'obtenir, les équations de Navier–Stokes (chapitre 6) sont résolues numériquement en utilisant la méthode de prédicteur–correcteur (II) avec pénalisation pour le calcul de la pression (chapitre 7) et des éléments finis de type Q_2 pour les vitesses horizontales et de type Q_1 pour la vitesse verticale et la pression (chapitre 8). Les systèmes linéaires obtenus sont résolus avec des méthodes du gradient conjugué (préconditionné) (première partie). Un logiciel parallèle développé sur les machines CRAY XT3 du CSCS (Swiss National Supercomputing Center, http ://www.cscs.ch) ont permis la réalisation des calculs. Le logiciel *AVS/Express* (http ://www.avs.com) a été utilisé pour obtenir les figures présentées dans ce chapitre.

La bathymétrie de l'océan Atlantique nord ainsi que les tractions moyennes en surface de 1993 ont été fournies par le projet de recherche français *Mercator Océan*, qui développe notamment un système d'océanographie opérationnelle basée sur l'assimilation de données, permettant d'obtenir des analyses et prédictions des océans (courants, température, salinité, etc.), voir le site web
http ://www.mercator-ocean.fr.

9.1 Bassin et bathymétrie

La partie de l'océan Atlantique considérée est délimitée à l'ouest par le continent américain, à l'est par l'Europe et l'Afrique, au nord par le parallèle à 70° de latitude et au sud par l'équateur. Les bords nord et sud sont artificiels à travers lesquels il n'y a aucun échange d'eau, c'est–à–dire, la deuxième composante de la vitesse (voir section 6.3) est fixée à zéro. En effet, à l'équateur, cette hypothèse est justifiée par la rotation de la Terre (voir remarque 6.2) et à 70° de latitude, elle est proche de la réalité car les courants de direction nord–sud y sont faibles,

négligeables dans la circulation globale de l'eau dans l'océan Atlantique dont les moteurs principaux sont les alizés (vents d'est en ouest proche de l'équateur). Si H_n et H_s sont respectivement les bords nord et sud, d'après les notations de la section 6.2, nous avons $G_1 = G_3 = G_b \setminus (H_n \cup H_s)$ et $G_2 = G_b$.

Remarque 9.1 *Notre modèle considère un fluide à densité constante (voir chapitre 6). Il ne tient compte ni de la salinité ni de la température. Nous négligeons le flux d'eau douce provenant du pôle nord.*

La bathymétrie de l'océan Atlantique nord est donnée sur la figure 28.

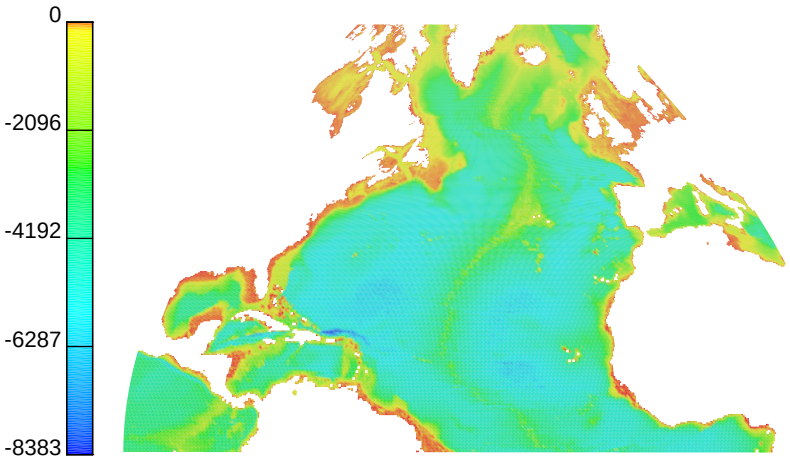

Figure 28: Bathymétrie de l'océan Atlantique nord [m].

9.2 Maillages et tractions

Tout d'abord, un maillage bidimensionnel de la surface (G_s) est construit, avec des éléments quadrilatères (de type Q_1) de côtés suivant les parallèles et les méridiens de la Terre (voir figure 29). Un maillage tridimensionnel d'éléments hexaédriques (de type Q_1) est construit à partir de celui de la surface de la manière suivante. Les segments verticaux (orthogonaux à G_s) reliant les nœuds du maillage de la surface au fond sont considérés. Chaque segment est subdivisé en c parties de même longueur, ce qui permet de construire naturellement un maillage tridimensionnel à c couches. Les faces externes du maillage tridimensionnel obtenu à partir du maillage de surface de la figure 29 en considérant 20 couches sont représentées sur la figure 30 avec une dilatation d'un facteur 200 pour les coordonnées verticales.

Les nombres de nœuds et d'éléments obtenus pour ces maillages sont donnés dans le tableau 44.

Maillage	Éléments	Nœuds	
		Q_1	Q_2
3D	119220	134106	1012741
2D (surface)	5961	6386	24701

Tableau 44.

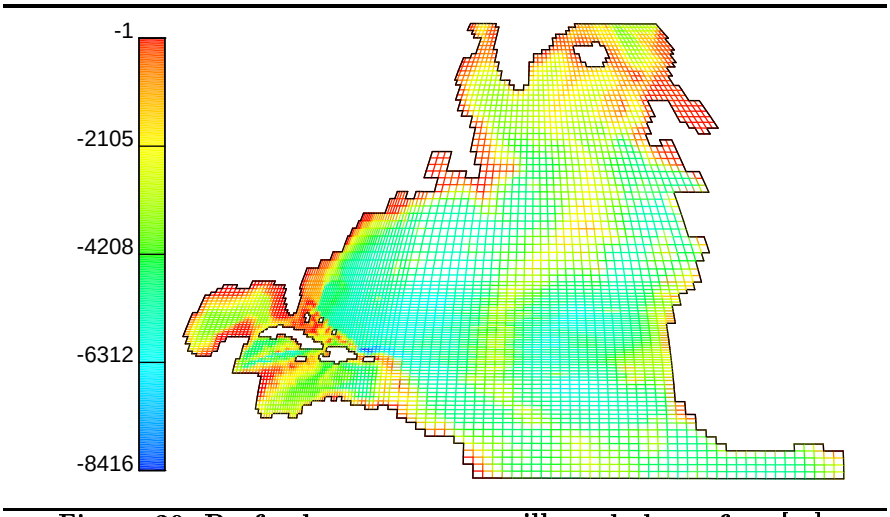

Figure 29: Profondeur pour un maillage de la surface [m].

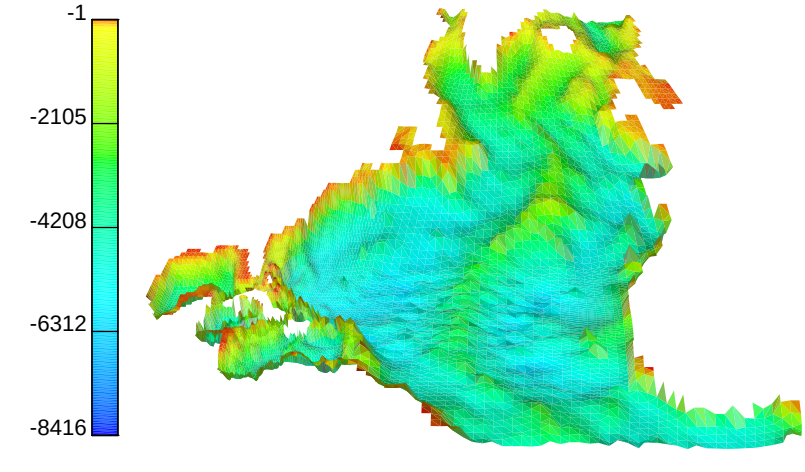

Figure 30: Profondeur pour un maillage 3D [m].

Les tractions de surface sont représentées sur la figure 31 avec la longueur des vecteurs normalisée.

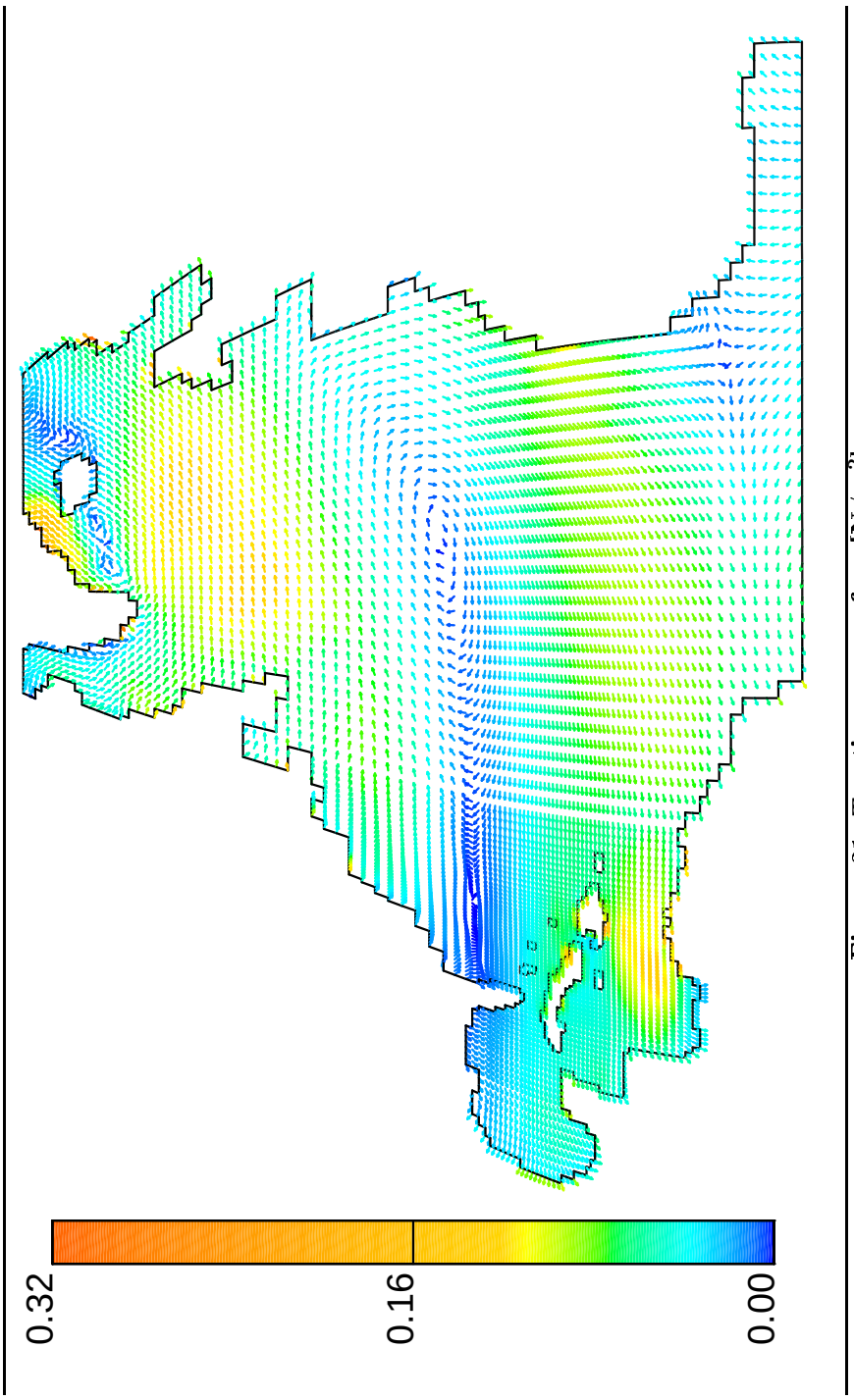

Figure 31: Tractions en surface [N/m^2].

9.3 Simulation numérique des courants

Le but est de calculer la circulation globale de l'eau dans l'océan Atlantique nord. Localement, les courants ne peuvent pas être calculés précisément avec les maillages considérés, par exemple le long de la côte française ou dans le golfe du Mexique.

9.3.1 Paramètres et méthode

Le vecteur de viscosité cinématique turbulente et le pas de temps considérés sont

$$\begin{aligned} \nu &= (10^9, 10^9, 2.5 \cdot 10^1), \\ \delta t &= 1 \text{ semaine } = 6.048 \cdot 10^5 \text{secondes.} \end{aligned}$$

De plus, les tractions sont multipliées par 1000.

De nombreux tests numériques ont été effectués avant de fixer ces valeurs. Expliquons brièvement comment elles ont été choisies.
Les courants dominants sont engendrés par les alizés, ils se situent peu au–dessus de l'équateur et sont dirigés d'est en ouest. Avec une vitesse moyenne de 1 m/s, la distance entre l'Afrique et l'Amérique centrale ($\approx 5'000$ km, longueur caractéristique de l'océan Atlantique nord) est parcourue en un peu plus de 8 semaines. Ainsi, il semble raisonnable de prendre un pas de temps de 1 semaine. Avec un diamètre horizontal de $d \approx 5'000$ km $= 5'000'000$ m et une profondeur de $h_0 \approx 5'000$ m, l'océan Atlantique nord a un rapport d'aspect $\varepsilon = h_0/d \approx 10^{-3}$ (voir section 6.1). Ainsi, d'après la section 6.6, le vecteur de viscosité devrait satisfaire la relation $\nu_3/\nu_1 \approx \varepsilon^2 \approx 10^{-6}$. Avec $\nu_1 = \nu_2 \in [10^6, 10^7]$ et $\nu_3 \in [1, 10]$, de premiers tests ont pu être effectués. Comme les plus grandes valeurs rendaient la résolution plus aisée, les valeurs de ν_i ont étés amplifiées ainsi que les tractions de surface pour garder des solutions avec des vitesses raisonnables. Finalement, la valeur de ν_3 a été réduite (nous avons $\nu_3 = 2.5 \cdot 10^1$, $\nu_1 = 10^9$, $\nu_3/\nu_1 = 2.5 \cdot 10^{-8} < \epsilon^2$) afin de rendre mieux compte du mouvement vertical du fluide.

Le modèle de prédicteur–correcteur (I) ne donnent pas de bons résultats : les courants obtenus sont irréguliers proche de la surface. Ainsi, pour garantir des courants corrects au bord du domaine, le modèle de prédicteur–correcteur (II) est choisi, avec la formule de différentiation rétrograde à 2 pas pour la discrétisation en temps, voir sections 7.5 et 7.6. Nous considérons la condition initiale $u_0 = 0$ et procédons selon la remarque 7.10 pour amorcer la résolution. Les maillages de la section 9.2 et les méthodes d'éléments finis présentées en détails au chapitre 8 sont employés.

Remarque 9.2 *Comme le cas non stationnaire des équations de Navier–Stokes a été considéré, le modèle utilisé permet d'estimer le temps nécessaire à l'obtention d'un écoulement stationnaire à partir de la condition initiale* $u_0 = 0$.

Le partie délicate de la résolution est le calcul de pression. Sans utiliser de méthode de pénalisation, la matrice de rigidité de la pression n'est pas de rang maximal. En traçant sa dernière ligne et sa dernière colonne (voir section 8.4.3),

nous obtenons une matrice SDP, d'ordre $134'105$ avec $1'742'654$ coefficients non nuls dans sa partie triangulaire supérieure (ou inf.). Cette dernière est très mal conditionnée (sa plus petite valeur propre n'a pas pu être estimée avec la méthode de la puissance inverse !) et, malgré l'emploi de préconditionneurs, la convergence pour le calcul de la pression n'a pas pu être obtenue de cette manière.
Par conséquent, une méthode de pénalisation est utilisée pour obtenir la pression. Seule la matrice est modifiée (voir section 7.3.1 et remarque 7.2) en utilisant un facteur de pénalisation $\varepsilon_p = 10^{-9}$. La matrice obtenue est alors SDP d'ordre $134'106$ avec $1'742'662$ coefficients non nuls dans sa partie triangulaire supérieure (ou inf.), dont la plus petite valeur propre vaut $\lambda_{min} \approx 2.34817751E-19$, la plus grande $\lambda_{max} \approx 4.22406008E-08$ (estimées avec la méthode de la puissance (inverse)) et la condition $\kappa \approx 1.79886745E+11$. Cette matrice reste mal conditionnée, mais la convergence des systèmes linéaires associés a pu être obtenue grâce à des préconditionneurs de type DIAG + GSC MC (OPT) BLOCS développés dans la première partie de ce travail (chapitre 3).

Remarque 9.3 *La méthode de prédicteur–correcteur (II) ne respecte pas la condition d'incompressibilité du fluide (voir section 7.5). Il est raisonnable d'employer cette méthode car cette condition est déjà violée par la pénalisation de la pression (voir remarque 7.4).*

Le préconditionneur diagonal (voir chapitre 2) est utilisé pour résoudre les systèmes linéaires donnant la vitesse.

Finalement, un logiciel parallèle développé sur les machines CRAY XT3 du CSCS (http ://www.cscs.ch) a permis d'obtenir des résultats.

9.3.2 Résultats

Un nombre de pas de temps suffisamment grand est considéré afin d'obtenir des vitesses (presque) stationnaires. Les courants obtenus après 400 ans sont donnés sur les figures 32–37 et 39–46 (avec une dilatation d'un facteur 200 sur l'axe vertical) ; les champs de vitesse dans des plans de coupe sont représentés. La profondeur (respectivement la latitude) est fixée sur les figures 32–37 (resp. 39–46). La longueur des vecteurs est normalisée sur les figures 32, 33 et 39–46.

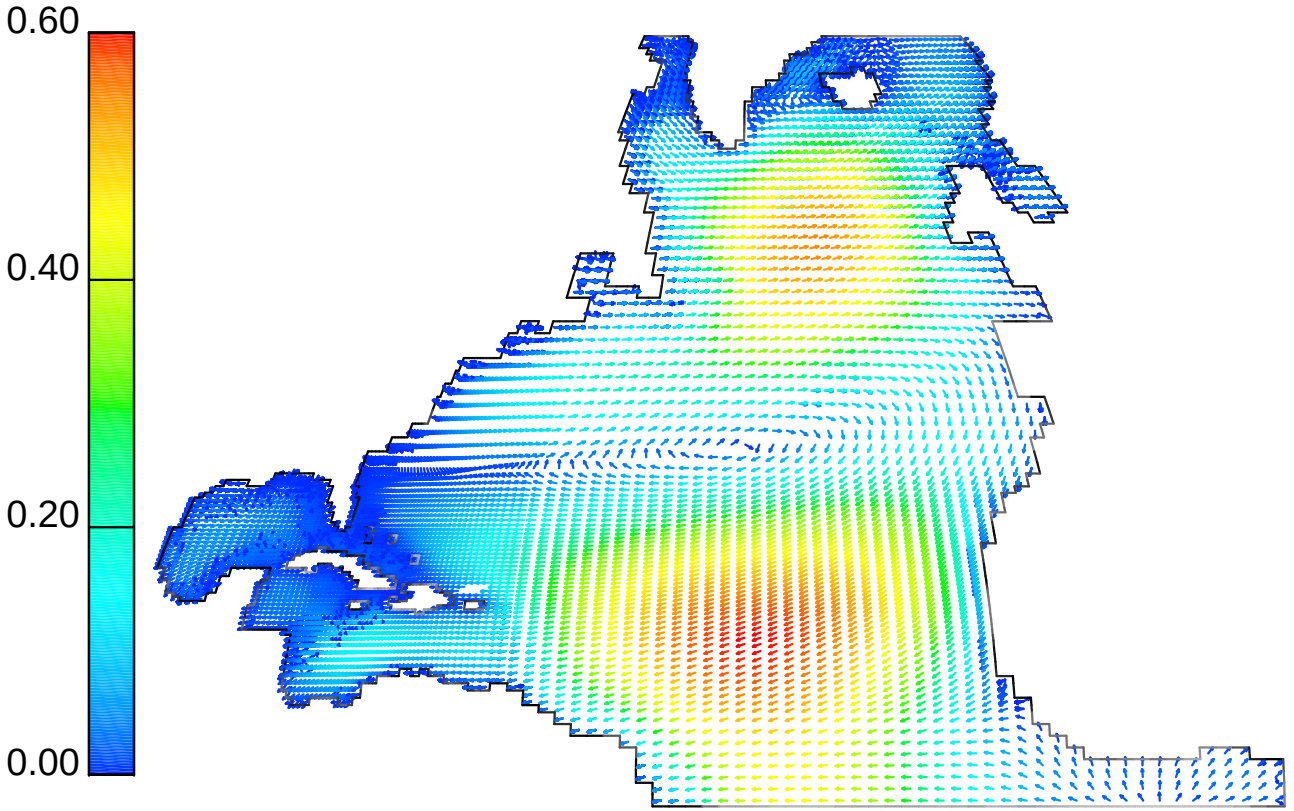

Figure 32: Champ des vitesses à la surface [m/s].

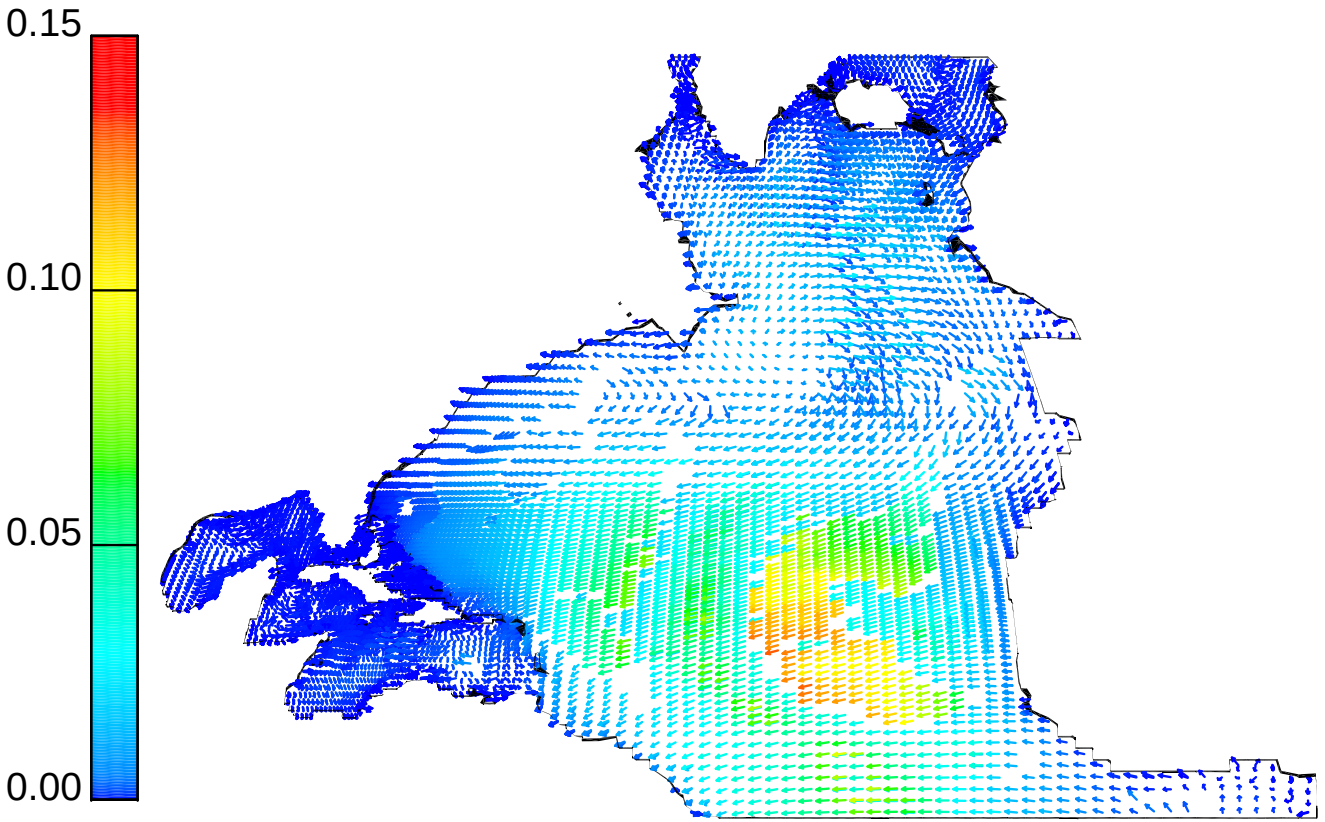

Figure 33: Champ des vitesses à 500 m de profondeur [m/s].

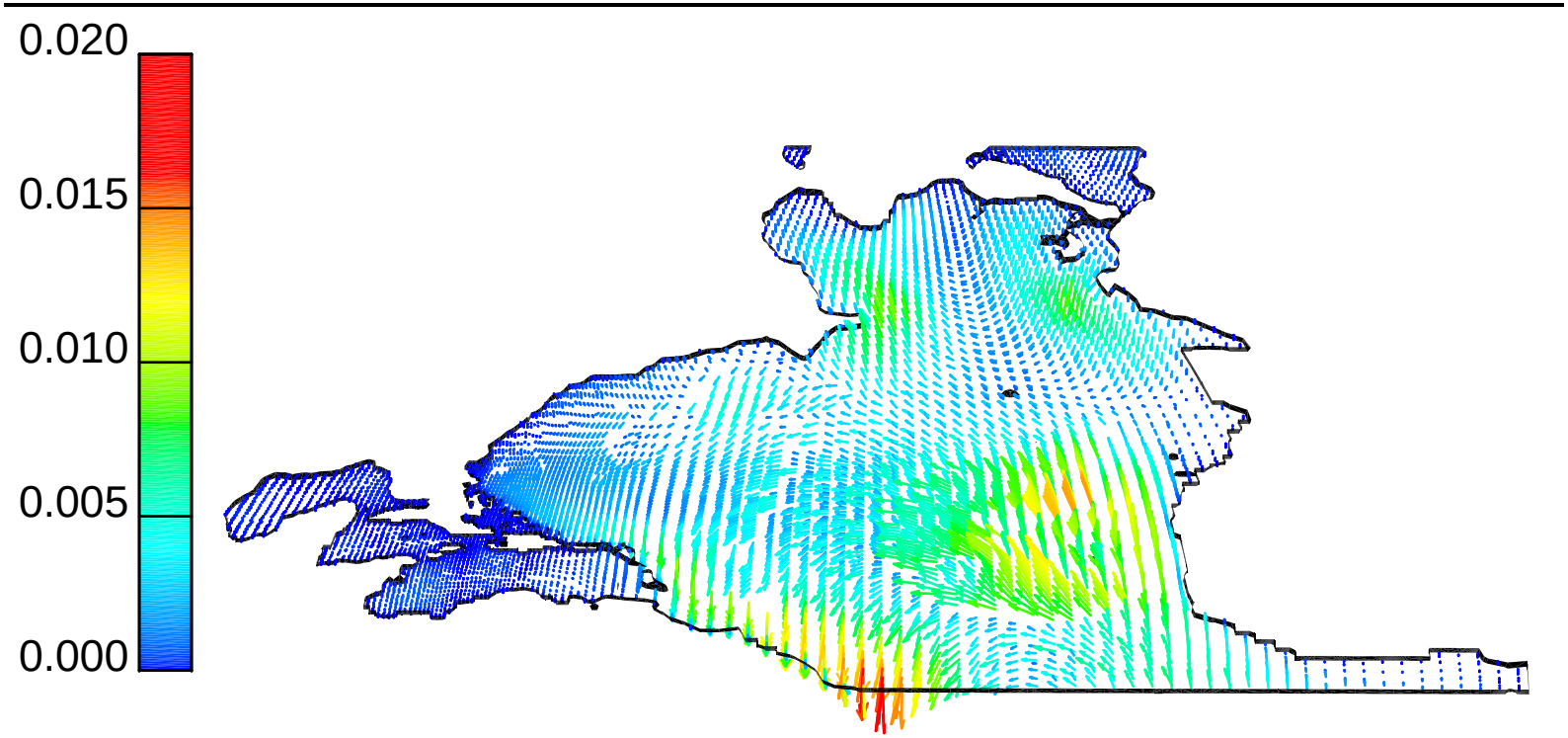

Figure 34: Champ des vitesses à 1000 m de profondeur [m/s].

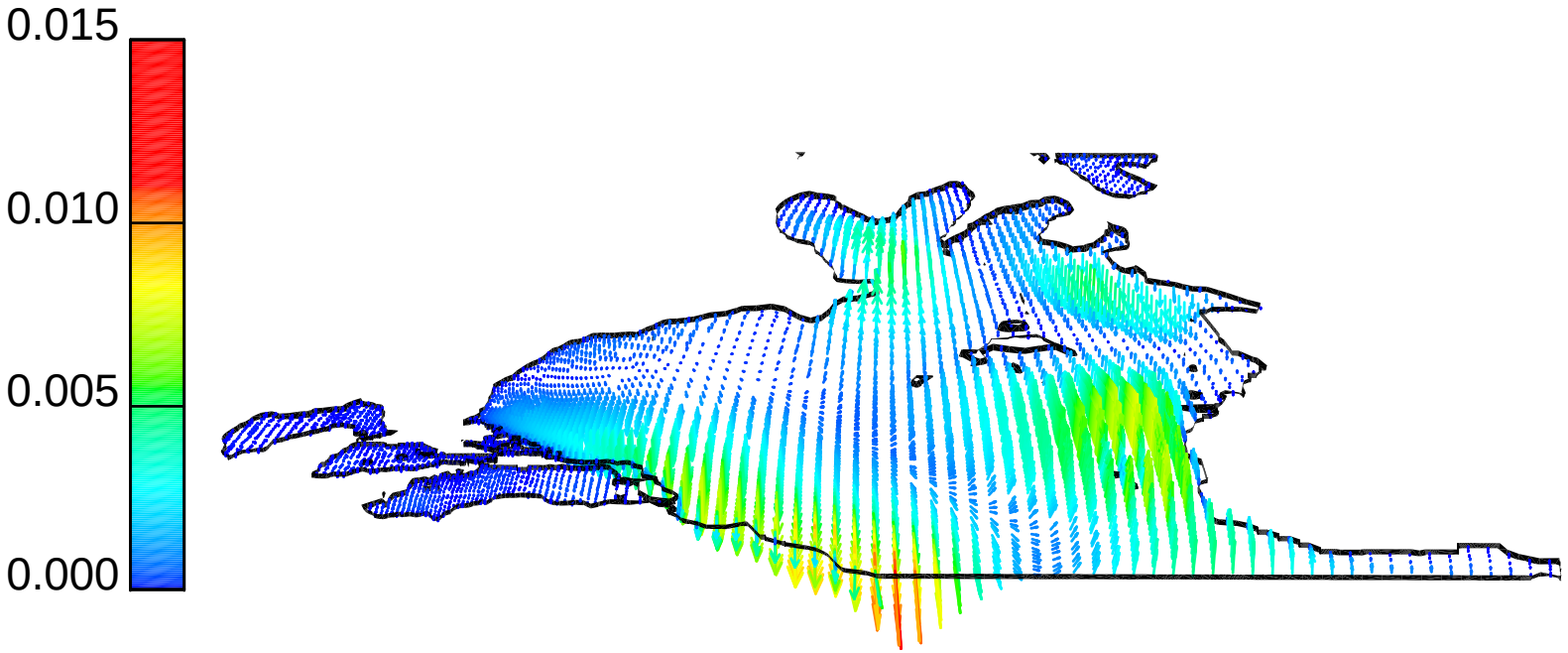

Figure 35: Champ des vitesses à 2000 m de profondeur [m/s].

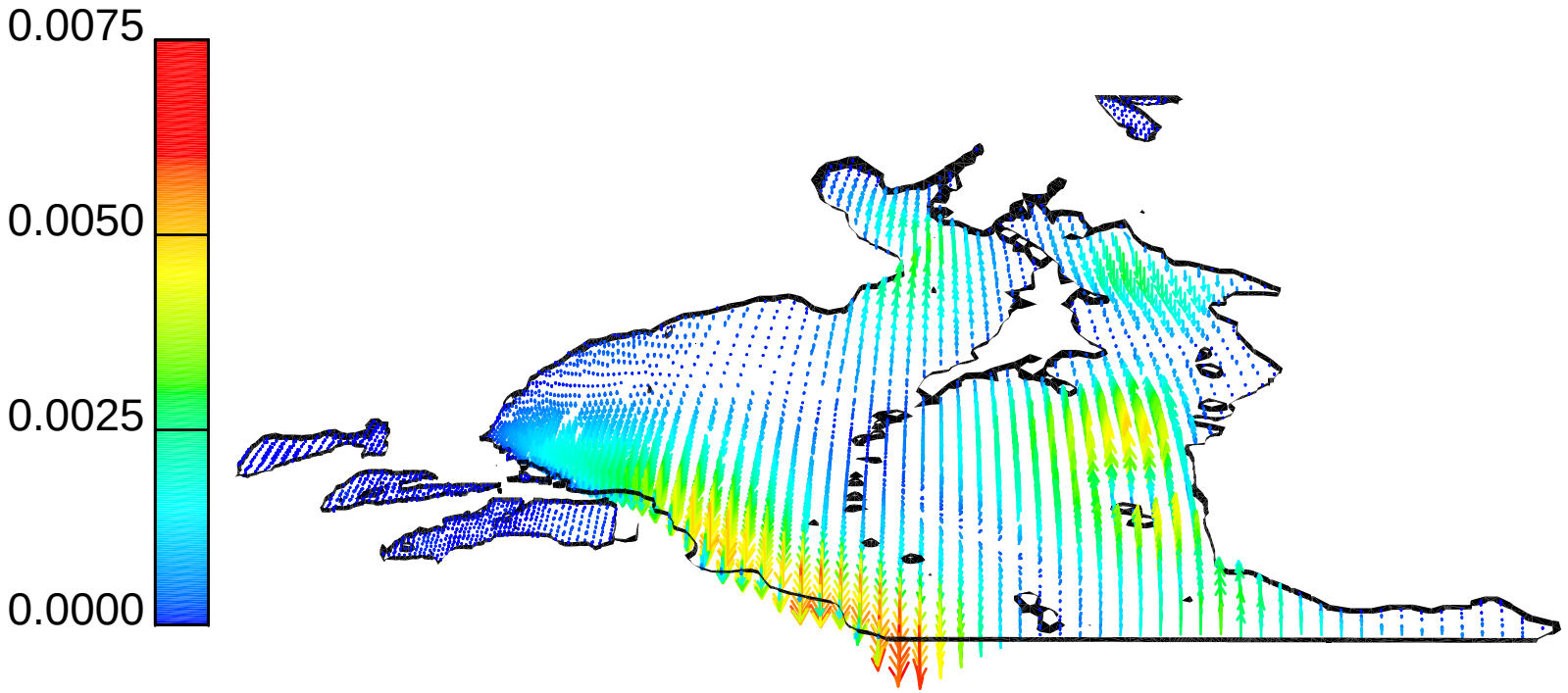

Figure 36: Champ des vitesses à 3000 m de profondeur [m/s].

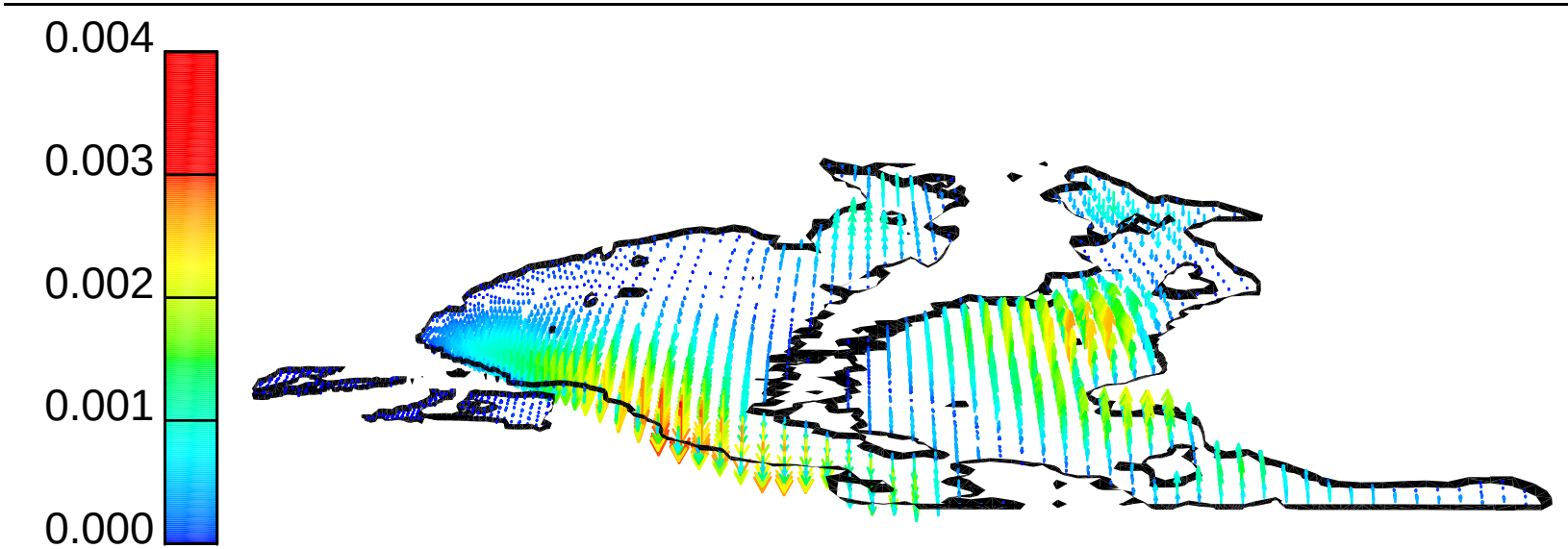

Figure 37: Champ des vitesses à 4000 m de profondeur [m/s].

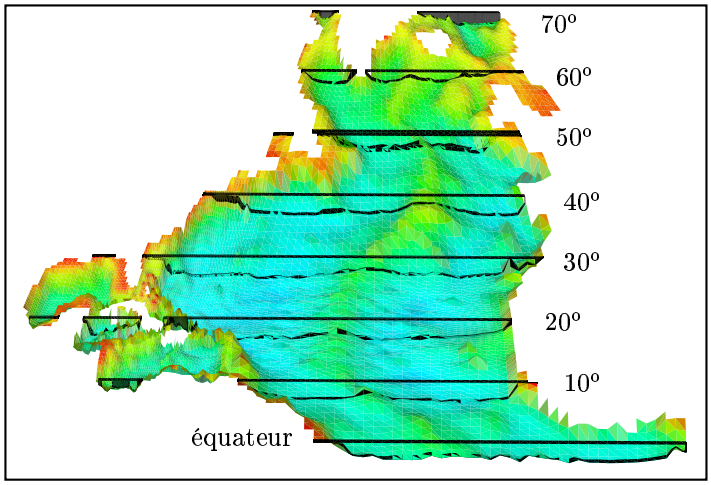

Figure 38: Coupes suivant les parallèles.

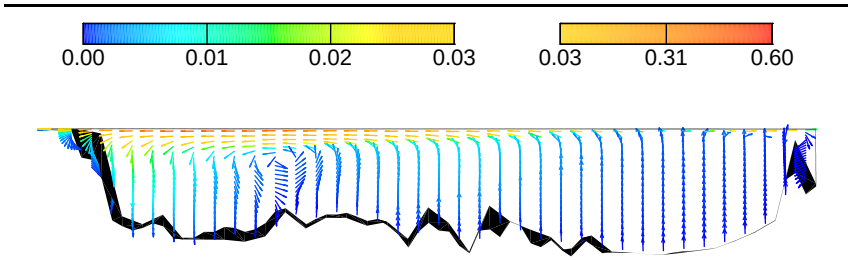

Figure 39: Champ des vitesses à l'équateur.

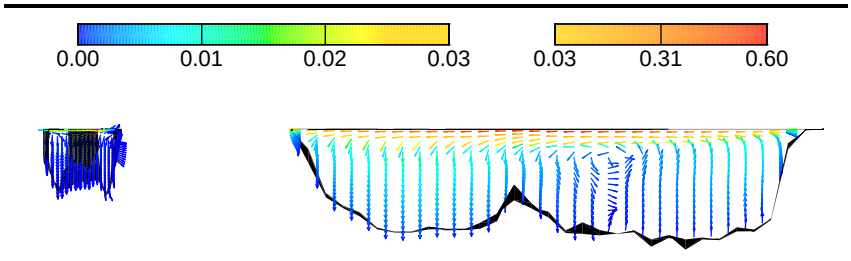

Figure 40: Champ des vitesses à 10° de latitude.

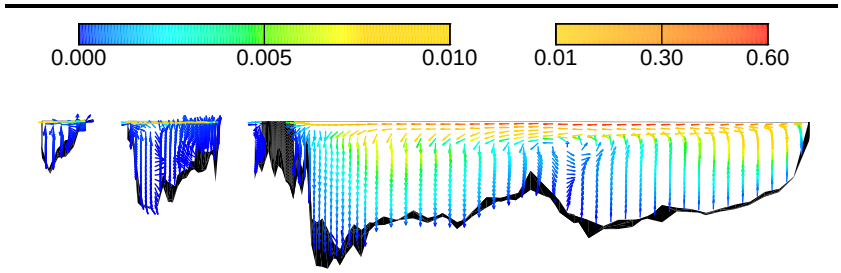

Figure 41: Champ des vitesses à 20° de latitude.

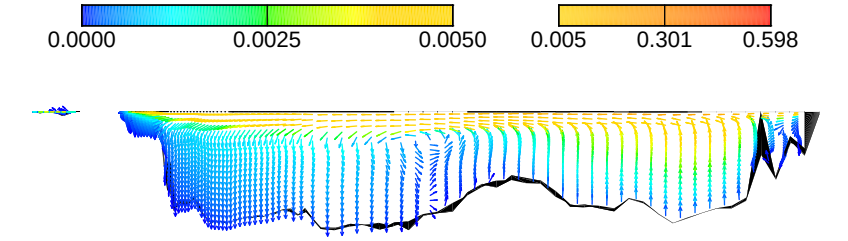

Figure 42: Champ des vitesses à 30° de latitude.

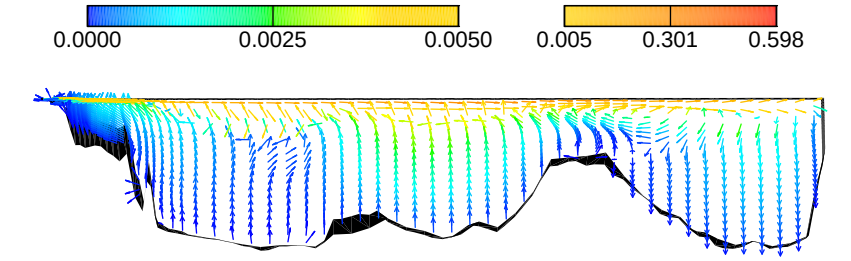

Figure 43: Champ des vitesses à 40° de latitude.

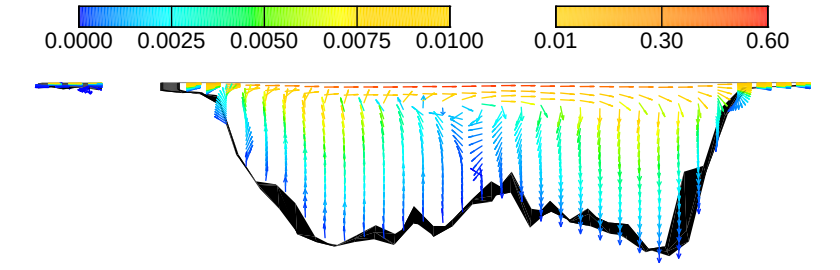

Figure 44: Champ des vitesses à 50° de latitude.

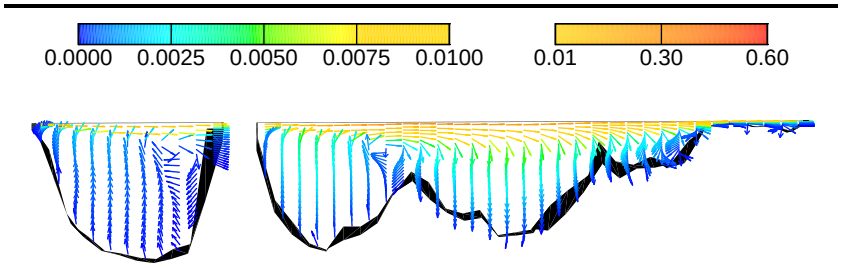

Figure 45: Champ des vitesses à 60° de latitude.

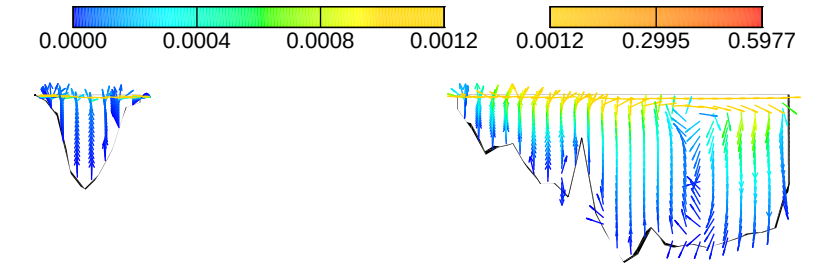

Figure 46: Champ des vitesses à 70° de latitude.

Ces différentes cartes de courants donnent une bonne idée de la circulation globale de l'eau dans l'océan Atlantique nord.
En surface (figure 32), nous pouvons observer le *Gulf Stream*, courant océanique allant de la Floride jusqu'en Europe, impliquant le climat tempéré de l'Europe occidentale ; le *courant du Labrador* provenant du nord et longeant le nord de la côte est des États–Unis est également mis en évidence, ce qui explique le climat froid de New York en comparaison avec celui de Naples par exemple, villes de même latitude ($\approx 41°$).
Ces cartes montrent que la grandeur de la vitesse diminue rapidement lorsque la profondeur augmente. À $4'000$ m de profondeur (figure 37) la vitesse maximale observée est de 4 mm/s. Cependant, vu la quantité d'eau déplacée, cela représente une énorme quantité d'énergie. De plus à des profondeurs élevées, l'intensité des courants est très faible dans la moitié nord du bassin par rapport à la moitié sud.
La dorsale Atlantique (chaîne de montagnes sous–marines, à la frontière de deux plaques tectoniques) s'étend de l'Islande à l'équateur, voir figures 28, 30 et 38. Nous pouvons observer son influence sur les courants sur les coupes verticales des figures 39–45.
Les forts courants de surface venant de l'est et se dirigeant vers l'Amérique centrale "plongent" vers le fond à proximité des îles Antillaises (voir figures 38 et 41). La prise en compte des effets dus à la température modifierait la circulation dans la région des Caraïbes (une couche d'eau chaude resterait en surface).

ANNEXE A

Détails sur la méthode du gradient conjugué

Présentons dans les détails comment s'obtiennent l'algorithme du gradient conjugué et ses principales propriétés [3, 19, 46].

A.1 Rappel de la méthode

La méthode du gradient conjugué est une méthode itérative pour résoudre le système (1.1) : $x_0 \in \mathbb{R}^n$ est fixé, puis pour $k \geqslant 0$, une solution approchée x_{k+1} est construite à partir de x_k ; le résidu associé à x_k est $r_k = b - Ax_k$.

Rappelons que la solution $\hat{x} = A^{-1}b$ de (1.1) est le minimum de la fonction

$$f(x) = \|x - \hat{x}\|_A^2 = \|b - Ax\|_{A^{-1}}^2 = (\, x \,|\, Ax \,) - 2\,(\, b \,|\, x \,) + (\, b \,|\, \hat{x} \,).$$

Nous obtenons x_{k+1} en cherchant depuis x_k et dans une direction d_k le minimum de cette application, *i.e.*

$$x_{k+1} = x_k + \alpha_k d_k,$$

où α_k réalise le minimum de la fonction quadratique en α, $f(x_k + \alpha d_k) = \alpha^2\,(\, d_k \,|\, Ad_k \,) + 2\alpha((\, Ax_k \,|\, d_k \,) - (\, b \,|\, d_k \,)) + (\, x_k \,|\, Ax_k \,) - 2\,(\, b \,|\, x_k \,) + (\, b \,|\, \hat{x} \,)$, *i.e.*

$$\alpha_k = \frac{(\, r_k \,|\, d_k \,)}{(\, d_k \,|\, Ad_k \,)}.$$

Reste à définir les directions de descente d_k. Nous choisissons $d_0 = r_0$ (parallèle à $\nabla f(x_0)$), la direction de plus grande pente de f au point x_0 et, pour $k \geqslant 0$, nous choisissons d_{k+1} sous la forme

$$d_{k+1} = r_{k+1} + \beta_k d_k,$$

de sorte que soit $(\, d_{k+1} \,|\, d_k \,)_A = 0$, c'est–à–dire, nous prenons

$$\beta_k = -\frac{(\, r_{k+1} \,|\, Ad_k \,)}{(\, d_k \,|\, Ad_k \,)}.$$

A.2 Premières propriétés

La proposition suivante nous donnera notamment les expressions de α_k et de β_k apparaissant dans l'algorithme GC.

Proposition A.1 *Pour $k \geqslant 0$, nous avons*

(i) $r_{k+1} = r_k - \alpha_k A d_k$,

(ii) $(\, r_{k+1} \,|\, d_k \,) = 0$,

(iii) $(\, r_k \,|\, d_k \,) = (\, r_k \,|\, r_k \,)$, *en particulier* $\alpha_k = \frac{(\, r_k \,|\, r_k \,)}{(\, d_k \,|\, A d_k \,)}$,

(iv) $(\, r_{k+1} \,|\, r_k \,) = 0$,

(v) $\beta_k = \frac{(\, r_{k+1} \,|\, r_{k+1} \,)}{(\, r_k \,|\, r_k \,)}$,

(vi) $d_k = (\, r_k \,|\, r_k \,) \sum_{i=0}^{k} \frac{r_i}{(\, r_i \,|\, r_i \,)}$.

Preuve.

(i) $r_{k+1} = b - A x_{k+1} = b - A(x_k + \alpha_k d_k) = r_k - \alpha_k A d_k$.

(ii) Comme $\alpha_k = \frac{(\, r_k \,|\, d_k \,)}{(\, d_k \,|\, A d_k \,)}$, avec (i), nous avons

$$(\, r_{k+1} \,|\, d_k \,) = (\, r_k \,|\, d_k \,) - \alpha_k \,(\, d_k \,|\, A d_k \,) = 0.$$

(iii) Pour $k = 0$, c'est évident ($d_0 = r_0$) et pour $k \geqslant 1$,

$$(\, r_k \,|\, d_k \,) = (\, r_k \,|\, r_k + \beta_{k-1} d_{k-1} \,) = (\, r_k \,|\, r_k \,) + \beta_{k-1} \,(\, r_k \,|\, d_{k-1} \,) \overset{(ii)}{=} (\, r_k \,|\, r_k \,).$$

(iv) Nous avons $(\, d_k \,|\, A d_k \,) = (\, r_k \,|\, A d_k \,)$. En effet, pour $k = 0$, c'est évident ($d_0 = r_0$) et pour $k \geqslant 1$,

$$(\, d_k \,|\, A d_k \,) = (\, r_k + \beta_{k-1} d_{k-1} \,|\, A d_k \,) = (\, r_k \,|\, A d_k \,) + \beta_{k-1} \,(\, d_{k-1} \,|\, A d_k \,) = (\, r_k \,|\, A d_k \,)$$

car $(\, d_k \,|\, d_{k-1} \,)_A = 0$ par construction de la méthode. Ainsi,

$$(\, r_{k+1} \,|\, r_k \,) \overset{(i)}{=} (\, r_k \,|\, r_k \,) - \alpha_k \,(\, A d_k \,|\, r_k \,) \overset{(iii)}{=} (\, r_k \,|\, r_k \,) \left(1 - \frac{(\, r_k \,|\, A d_k \,)}{(\, d_k \,|\, A d_k \,)} \right) = 0.$$

(v) De (i), nous avons $A d_k = \alpha_k^{-1} (r_k - r_{k+1})$; donc

$$\begin{aligned} \beta_k &= -\frac{(\, r_{k+1} \,|\, A d_k \,)}{(\, d_k \,|\, A d_k \,)} = \frac{(\, r_{k+1} \,|\, r_{k+1} - r_k \,)}{\alpha_k \,(\, d_k \,|\, A d_k \,)} \overset{(iv)}{=} \frac{(\, r_{k+1} \,|\, r_{k+1} \,)}{\alpha_k \,(\, d_k \,|\, A d_k \,)} \\ &\overset{(iii)}{=} \frac{(\, r_{k+1} \,|\, r_{k+1} \,)}{(\, r_k \,|\, r_k \,)}. \end{aligned}$$

(vi) C'est évident pour $k = 0$ ($d_0 = r_0$) ; supposons que l'égalité est vérifiée pour k et montrons qu'elle l'est encore pour $k+1$:

$$\begin{aligned} \mathrm{d}_{k+1} &= r_{k+1} + \beta_k d_k = r_{k+1} + \beta_k \,(\, r_k \,|\, r_k \,) \sum_{i=0}^{k} \frac{r_i}{(\, r_i \,|\, r_i \,)} \\ &\overset{(v)}{=} (\, r_{k+1} \,|\, r_{k+1} \,) \sum_{i=0}^{k+1} \frac{r_i}{(\, r_i \,|\, r_i \,)}. \end{aligned}$$

■

Remarque A.1 *Il n'y a pas de division par zéro ; en effet, si* $r_k \neq 0$*, alors, par (iii) de la proposition précédente,* $d_k \neq 0$*. (Bien sûr, si* $r_k = 0$*, alors* x_k *est la solution du système et nous nous arrêtons.)*

Proposition A.2

(i) Si $i > j$*, alors* $(\, r_i \,|\, d_j \,) = 0$*.*

(ii) Les résidus r_k *sont orthogonaux deux à deux,* i.e. $(\, r_i \,|\, r_j \,) = 0$ *si* $i \neq j$*.*

(iii) Les directions de descentes d_k *sont A–orthogonales deux à deux,* i.e. $(\, d_i \,|\, d_j \,)_A = 0$ *si* $i \neq j$*.*

Preuve. Prouvons les trois points simultanément par récurrence en montrant que, pour tout $k \geqslant 0$, nous avons

$$\begin{aligned} &\text{(i')} && (\, r_{k+1} \,|\, d_j \,) = 0, \\ &\text{(ii')} && (\, r_{k+1} \,|\, r_j \,) = 0, \\ &\text{(iii')} && (\, d_{k+1} \,|\, d_j \,)_A = 0, \end{aligned}$$

pour $0 \leqslant j \leqslant k$.

Nous avons $(\, r_1 \,|\, d_0 \,) = (\, r_1 \,|\, r_0 \,) = 0$ par la proposition A.1(ii) (ou (iv)) et $(\, d_1 \,|\, d_0 \,)_A = 0$ par construction, ce qui montre (i'), (ii') et (iii') pour $k = j = 0$. Supposons les égalités (i'),(ii') et (iii') vraies pour $k-1$ et $0 \leqslant j \leqslant k-1$, et montrons qu'elles le sont encore pour k et $0 \leqslant j \leqslant k$.

(i') Il faut montrer $(\, r_{k+1} \,|\, d_j \,) = 0$, pour $0 \leqslant j \leqslant k$. Pour $j = k$, c'est la proposition A.1(ii) et pour $0 \leqslant j \leqslant k-1$:

$$(\, r_{k+1} \,|\, d_j \,) = (\, r_k - \alpha_k A d_k \,|\, d_j \,) = (\, r_k \,|\, d_j \,) - \alpha_k \,(\, d_k \,|\, d_j \,)_A = 0$$

par hypothèse de récurrence sur (i') et (iii').

(ii') Il faut montrer $(\, r_{k+1} \,|\, r_j \,) = 0$, pour $0 \leqslant j \leqslant k$. Pour $j = k$, c'est la proposition A.1(iv). Pour $0 \leqslant j \leqslant k-1$, nous avons

$$(\, r_{k+1} \,|\, r_j \,) = (\, r_k \,|\, r_j \,) - \alpha_k \,(\, A d_k \,|\, r_j \,) = -\alpha_k \,(\, A d_k \,|\, r_j \,)$$

par hypothèse de récurrence sur (ii'). Si $j = 0$, $(\, A d_k \,|\, r_j \,) = (\, A d_k \,|\, d_0 \,) = 0$ par hypothèse de récurrence sur (iii'). Si $1 \leqslant j \leqslant k-1$,

$$(\, A d_k \,|\, r_j \,) = (\, A d_k \,|\, d_j - \beta_{j-1} d_{j-1} \,) = 0$$

par hypothèse de récurrence sur (iii').

(iii') Il faut montrer $(\, d_{k+1} \,|\, d_j \,)_A = 0$, pour $0 \leqslant j \leqslant k$. C'est vrai pour $j = k$ par construction. Pour $0 \leqslant j \leqslant k-1$, nous avons

$$(\, d_{k+1} \,|\, d_j \,)_A = (\, d_{k+1} \,|\, A d_j \,) = (\, r_{k+1} \,|\, A d_j \,) + \beta_k \,(\, d_k \,|\, A d_j \,) = (\, r_{k+1} \,|\, A d_j \,)$$

par hypothèse de récurrence sur (iii') et, comme $A d_j = \alpha_j^{-1}(r_j - r_{j+1})$,

$$(\, r_{k+1} \,|\, A d_j \,) = \alpha_j^{-1}((\, r_{k+1} \,|\, r_j \,) - (\, r_{k+1} \,|\, r_{j+1} \,)) = 0$$

par (ii') que nous venons de montrer.

■

La propriété (ii) implique que la méthode du gradient conjugué converge en au plus n itérations (où n est l'ordre de la matrice).
Le nom de la méthode est dû à la propriété (iii) : les directions de descente sont conjuguées.

A.3 Projections et sous-espaces de Krylov

Pour $m \geqslant 1$, considérons le sous-espace de Krylov

$$K_m = K_m(A, r_0) = \langle r_0, Ar_0, A^2 r_0, \ldots, A^{m-1} r_0 \rangle,$$

i.e. le sous-espace de $\mathbb{R}^n$ engendré par les vecteurs $r_0, Ar_0, A^2 r_0, \ldots, A^{m-1} r_0$.

Proposition A.3 *Lorsque $r_{m-1} \neq 0$ (ainsi que les résidus précédents), K_m est de dimension m, $\{r_j\}_{j=0}^{m-1}$ en est une base orthogonale et $\{d_j\}_{j=0}^{m-1}$ en est une base A–orthogonale.*

Preuve. Comme K_m est engendré par m vecteurs, sa dimension est inférieure ou égale à m. De la proposition A.2, nous savons que $\{r_j\}_{j=0}^{m-1}$ est une famille orthogonale de $\mathbb{R}^n$ et que $\{d_j\}_{j=0}^{m-1}$ est une famille A–orthogonale de $\mathbb{R}^n$. Pour démontrer la proposition, il suffit donc de montrer que $r_j, d_j \in K_m$ pour $0 \leqslant j \leqslant m-1$.

Montrons par récurrence sur m que $r_j \in K_m$ pour $0 \leqslant j \leqslant m-1$. Pour $m = 1$, c'est évident. Supposons que c'est vrai pour m et montrons que c'est encore le cas pour $m+1$. Comme $K_m,\ A(K_m) \subset K_{m+1}$ d'après la définition des sous-espaces K_m, nous avons $r_j \in K_m \subset K_{m+1}$, pour $0 \leqslant j \leqslant m-1$ et, par la proposition A.1,

$$r_m = r_{m-1} - \alpha_{m-1} A d_{m-1} = r_{m-1} - \alpha_{m-1} \, (\, r_{m-1} \,|\, r_{m-1} \,) \sum_{j=0}^{m-1} \frac{Ar_j}{(\, r_j \,|\, r_j \,)} \in K_{m+1}.$$

Enfin, par ce qui précède et la proposition A.1(vi), nous avons $d_j \in K_m$ pour $0 \leqslant j \leqslant m-1$. ■

Nous allons décrire dans la proposition suivante que la méthode du gradient conjugué peut être vue comme une méthode de projection :

Proposition A.4 *Dans l'algorithme du gradient conjugué, nous avons*

(i) $x_m \in x_0 + K_m$,

(ii) $\tilde{x}_m := x_m - x_0$ *est la projection A–orthogonale de $\hat{x} - x_0$ sur le sous-espace K_m, où $\hat{x} = A^{-1} b$ est la solution du système, c'est-à-dire*

$$\|x_m - \hat{x}\|_A = \min_{x \in x_0 + K_m} \|x - \hat{x}\|_A \,.$$

Preuve. Le point (i) est évident. En effet, $x_1 = x_0 + \alpha_0 d_0 = x_0 + \alpha_0 r_0 \in x_0 + K_1$ et si $x_m \in x_0 + K_m$, alors $x_{m+1} = x_m + \alpha_m d_m \in x_0 + K_{m+1}$, car $d_m \in K_{m+1}$ et $K_m \subset K_{m+1}$.

Montrons le point (ii). Le vecteur $r_m = b - Ax_m = A(\hat{x} - x_m)$ est orthogonal (pour le produit scalaire usuel) au sous-espace K_m, donc $\hat{x} - x_m$ est A–orthogonal à K_m. Ainsi, $\tilde{x}_m = x_m - x_0$ est la projection A–orthogonale de $\hat{x} - x_0$ sur K_m : $((\hat{x} - x_0) - \tilde{x}_m) = (\hat{x} - x_m) \perp_A K_m$. Par conséquent

$$\|x_m - \hat{x}\|_A = \|(\hat{x} - x_0) - \tilde{x}_m\|_A = \min_{\tilde{x} \in K_m} \|(\hat{x} - x_0) - \tilde{x}\|_A = \min_{x \in x_0 + K_m} \|x - \hat{x}\|_A$$

■

Corollaire A.5 *En notant $e_m = x_m - \hat{x}$, nous avons*

$$\|e_m\|_A = \min_{q \in \mathbb{P}_{m-1}} \|(1 - Aq(A))e_0\|_A \,,$$

où $\mathbb{P}_{m-1}$ désigne l'ensemble des polynômes de degré inférieur ou égal à $m - 1$.

Preuve. Comme

$$K_m = \langle r_0, Ar_0, \ldots, A^{m-1} r_0 \rangle = \{q(A) r_0 \mid q \in \mathbb{P}_{m-1}\}$$

et $r_0 = b - Ax_0 = A(\hat{x} - x_0) = -Ae_0$, le résultat découle immédiatement de la proposition précédente. ■

A.4 Polynômes de Chebychev

Les polynômes de Chebychev sont définis par récurrence comme suit :

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x), \ k \geqslant 1. \end{aligned}$$

Ils vérifient

$$T_k(\cos\Theta) = \cos(k\Theta), \tag{A.1}$$
$$T_k(\cosh\Theta) = \cosh(k\Theta), \tag{A.2}$$

pour tout $\Theta \in \mathbb{R}$ et tout $k \geqslant 0$. En effet : c'est évident pour $k = 0$ et $k = 1$; supposons que les égalités soient vérifiées pour $k - 1$ et k, alors, en utilisant les relations

$$\begin{aligned} \cos(\alpha + \beta) + \cos(\alpha - \beta) &= 2\cos\alpha\cos\beta, \\ \cosh(\alpha + \beta) + \cosh(\alpha - \beta) &= 2\cosh\alpha\cosh\beta \end{aligned}$$

avec $\alpha = k\Theta$ et $\beta = \Theta$, nous obtenons les égalités pour $k + 1$.

Proposition A.6 *Soit $\mathbb{P}_k^1$ l'ensemble des polynômes de degré inférieur ou égal à k prenant la valeur 1 en 0 et soit $0 < a < b$. Alors*

$$P_k(x) = \left(T_k\left(\frac{b+a}{b-a}\right)\right)^{-1} T_k\left(\frac{b+a-2x}{b-a}\right)$$

est un polynôme de $\mathbb{P}_k^1$ vérifiant

$$\left(T_k\left(\frac{b+a}{b-a}\right)\right)^{-1} = \max_{a\leqslant x\leqslant b} |P_k(x)| = \min_{P\in\mathbb{P}_k^1} \max_{a\leqslant x\leqslant b} |P(x)|.$$

Preuve. Comme $\frac{b+a}{b-a} > 1$, par la relation (A.2),

$$\zeta := T_k\left(\frac{b+a}{b-a}\right) \geqslant 1$$

et donc P_k est bien défini. Comme T_k est de degré k par construction, $P_k \in \mathbb{P}_k^1$. La relation (A.1) implique $\max_{-1\leqslant x\leqslant 1} |T_k(x)| = T_k(1) = 1$. Ainsi, comme $\{\frac{b+a-2x}{b-a} \mid a \leqslant x \leqslant b\} = [-1, 1]$, nous avons

$$\max_{a\leqslant x\leqslant b} |P_k(x)| = \frac{1}{\zeta}.$$

Il reste à montrer que $\max_{a\leqslant x\leqslant b} |P_k(x)| \leqslant \max_{a\leqslant x\leqslant b} |P(x)|$, pour tout $P \in \mathbb{P}_k^1$. Pour $j = 0, 1, \ldots, k$, considérons x_j défini par

$$\frac{b+a-2x_j}{b-a} = \cos\left(\frac{j\pi}{k}\right).$$

Nous avons $a = x_0 < x_1 < \ldots < x_k = b$ et, par la relation (A.1),

$$P_k(x_j) = \frac{1}{\zeta} T_k\left(\cos\left(\frac{j\pi}{k}\right)\right) = \frac{1}{\zeta}\cos(j\pi) = \frac{(-1)^j}{\zeta},\ 0 \leqslant j \leqslant k.$$

Soit $P \in \mathbb{P}_k^1$. Supposons par l'absurde que $\max_{a\leqslant x\leqslant b} |P_k(x)| > \max_{a\leqslant x\leqslant b} |P(x)|$. Alors, par ce qui précède, le polynôme $R(x) = P_k(x) - P(x)$ est non nul au points x_j, $0 \leqslant j \leqslant k$ et son signe alterne ($R(x_0) > 0 > R(x_1) < 0 < R(x_2) > 0 > \ldots$). Donc, R s'annule sur chaque intervalle $]x_j, x_{j+1}[$, $0 \leqslant j \leqslant k-1$. De plus, $R(0) = 1 - 1 = 0$, par conséquent, R possède au moins $k+1$ zéros. Comme R est de degré inférieur ou égal à k, il suit que R est le polynôme identiquement nul, *i.e.* $P = P_k$, d'où la contradiction. ■

A.5 Vitesse de convergence de la méthode du gradient conjugué

Dans cette section, le théorème 1.1 est démontré ; rappelons l'énoncé :

Théorème A.7 *Si $\kappa = \lambda_{max}/\lambda_{min}$ désigne la condition de la matrice A (λ_{min}, λ_{max} étant respectivement la plus petite et la plus grande valeur propre de A), alors*

$$\|e_m\|_A \leqslant 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m \|e_0\|_A .$$

Preuve. Si $\kappa = 1$, alors toutes les valeurs propres sont égales et la matrice A est un multiple de la matrice identité, $A = \lambda I$, $\lambda > 0$; dans ce cas, la méthode du gradient conjugué converge en au plus une itération (et $\|e_1\|_A = 0$) :

$$\begin{aligned} x_1 &= x_0 + \alpha_0 d_0 = x_0 + \frac{(\, r_0 \,|\, r_0 \,)}{(\, d_0 \,|\, Ad_0 \,)} d_0 \\ &= x_0 + \frac{(\, r_0 \,|\, r_0 \,)}{(\, r_0 \,|\, \lambda r_0 \,)} r_0 = x_0 + \lambda^{-1}(b - Ax_0) \\ &= \lambda^{-1} b = \hat{x}. \end{aligned}$$

Supposons $\kappa > 1$, *i.e.* $\lambda_{\min} < \lambda_{\max}$. D'après le corollaire A.5, nous avons

$$\|e_m\|_A = \min_{q \in \mathbb{P}_{m-1}} \|(1 - A(q(A))e_0\|_A = \min_{p \in \mathbb{P}^1_m} \|p(A)e_0\|_A .$$

Notons $0 < \lambda_{\min} = \lambda_1 \leqslant \ldots \leqslant \lambda_n = \lambda_{\max}$ les valeurs propres de A et considérons une base orthonormée $\{\xi_1, \ldots, \xi_n\}$ de $\mathbb{R}^n$ avec $A\xi_i = \lambda_i \xi_i$, $1 \leqslant i \leqslant n$.
Si $e_0 = \sum_{i=1}^n \gamma_i \xi_i$ est l'expression de e_0 dans cette base et $p \in \mathbb{P}^1_m$, alors

$$\begin{aligned} \|p(A)e_0\|_A^2 &= (\, p(A)e_0 \,|\, Ap(A)e_0 \,) = \sum_{i,j=1}^n \gamma_i \gamma_j \,(\, p(A)\xi_i \,|\, Ap(A)\xi_j \,) \\ &= \sum_{i,j=1}^n \gamma_i \gamma_j p(\lambda_i)\lambda_j p(\lambda_j)\,(\, \xi_i \,|\, \xi_j \,) = \sum_{i=1}^n \lambda_i p(\lambda_i)^2 \gamma_i^2 \\ &\leqslant \max_{i=1,\ldots,n} \left(p(\lambda_i)^2\right) \sum_{i=1}^n \lambda_i \gamma_i^2 = \max_{i=1,\ldots,n} \left(p(\lambda_i)^2\right) \|e_0\|_A^2 \\ &\leqslant \max_{\lambda_1 \leqslant \lambda \leqslant \lambda_n} \left(p(\lambda)^2\right) \|e_0\|_A^2 \end{aligned}$$

et

$$\|e_m\|_A = \min_{p \in \mathbb{P}^1_m} \|p(A)e_0\|_A \leqslant \min_{p \in \mathbb{P}^1_m} \max_{\lambda_1 \leqslant \lambda \leqslant \lambda_n} |p(\lambda)| \, \|e_0\|_A .$$

Par la proposition A.6, nous avons donc

$$\|e_m\|_A \leqslant \left(T_m\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)\right)^{-1} \|e_0\|_A .$$

Posons $\eta = \frac{\lambda_1}{\lambda_n - \lambda_1}$; nous avons $\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1} = 1 + 2\eta > 1$. Comme, par la relation (A.2),

$$T_m(x) = \cosh(m \operatorname{arccosh} x) = \frac{1}{2}\left(\left(x + \sqrt{x^2 - 1}\right)^m + \left(x + \sqrt{x^2 - 1}\right)^{-m}\right)$$

pour tout $x \geqslant 1$, nous avons

$$T_m(1 + 2\eta) \geqslant \frac{1}{2}\left(1 + 2\eta + \sqrt{(1 + 2\eta)^2 - 1}\right)^m = \frac{1}{2}\left(1 + 2\eta + 2\sqrt{\eta}\sqrt{\eta + 1}\right)^m .$$

En remplaçant η par sa valeur, nous obtenons

$$\begin{aligned} 1+2\eta+2\sqrt{\eta}\sqrt{\eta+1} &= \left(\sqrt{\eta}+\sqrt{\eta+1}\right)^2 = \frac{\left(\sqrt{\lambda_1}+\sqrt{\lambda_n}\right)^2}{\lambda_n-\lambda_1} \\ &= \frac{\sqrt{\lambda_n}+\sqrt{\lambda_1}}{\sqrt{\lambda_n}-\sqrt{\lambda_1}} = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \end{aligned}$$

et donc

$$\|e_m\|_A \leqslant \left(T_m(1+2\eta)\right)^{-1} \|e_0\|_A \leqslant 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m \|e_0\|_A .$$

■

ANNEXE B

Intégration numérique

Dans cette annexe, la formule de quadrature de Gauss–Legendre est présentée. Il s'agit d'une méthode d'intégration numérique sur un intervalle $[a, b]$; nous pouvons supposer que $a = -1$ et $b = 1$, quitte à effectuer le changement de variables $y = -1 + 2(x-a)/(b-a)$, qui envoie l'intervalle $[a, b]$ sur $[-1, 1]$. Pour une fonction continue f définie sur $[-1, 1]$ et à valeurs dans $\mathbb{R}$, cette formule est du type

$$J_n(f) = \sum_{k=1}^{n} a_k f(x_k)$$

avec des nombres réels a_k et $x_k \in [-1, 1]$ déterminés dans la suite. Le nombre $J_n(f)$ donne une approximation de l'intégrale de -1 à 1 de la fonction f et la valeur exacte de cette intégrale pour des polynômes de degré inférieur ou égal à $2n-1$. L'erreur commise est estimée dans la section B.4. Voir par exemple [44].

B.1 Polynômes de Legendre

Les polynômes de Legendre sont définis par

$$P_n(x) = \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left\{ \left(x^2 - 1\right)^n \right\}, \ n \geqslant 0.$$

Le polynôme P_n est clairement de degré n, par conséquent la famille $\{P_0, \ldots, P_n\}$ est une base de l'espace vectoriel $\mathbb{P}_n$ des polynômes réels de degré inférieur ou égal à n ; de plus, comme la dérivée d'une fonction paire (respectivement impaire) est impaire (resp. paire), P_n est de même parité que n.

Proposition B.1 *Les propriétés suivantes sont satisfaites :*

(i) Pour $n \geqslant 0$, le coefficient dominant de P_n est

$$\gamma_n = \frac{(2n)!}{2^n (n!)^2}$$

et $P_n(1) = 1$.

(ii) Les polynômes de Legendre sont orthogonaux deux à deux dans $L^2([-1,1])$, i.e. *pour* $m, n \geqslant 0$, $m \neq n$,

$$\int_{-1}^{1} P_n(x)P_m(x)dx = 0.$$

Ainsi P_n *est orthogonal à* $\mathbb{P}_{n-1}$.

(iii) Le carré de la norme de P_n *dans* $L^2([-1,1])$ *est donnée par*

$$\int_{-1}^{1} (P_n(x))^2 \, dx = \frac{2}{2n+1}, \ n \geqslant 0.$$

(iv) Les polynômes P_n *satisfont la relation de récurrence*

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x, \\ P_{n+1}(x) &= \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x), \ n \geqslant 1. \end{aligned}$$

(v) Pour $n \geqslant 0$, *le polynôme* P_n *admet exactement* n *racines distinctes dans l'intervalle* $]-1,1[$.

Preuve. (i). Le terme de plus haut degré de P_n est

$$\frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left\{x^{2n}\right\} = \frac{(2n)!}{2^n (n!)^2} x^n,$$

ce qui donne γ_n.
Nous avons

$$\frac{\mathrm{d}^n}{\mathrm{d}x^n} \left\{ \left(x^2-1\right)^n \right\} = \sum_{j=0}^{n} \binom{n}{j} \frac{\mathrm{d}^j}{\mathrm{d}x^j} \left\{(x-1)^n\right\} \frac{\mathrm{d}^{n-j}}{\mathrm{d}x^{n-j}} \left\{(x+1)^n\right\}. \tag{B.1}$$

Comme $x = 1$ est une racine de $(x-1)^n$ de multiplicité n, les dérivées jusqu'à l'ordre $n-1$ de $(x-1)^n$ s'annulent en $x = 1$. Ainsi, en évaluant l'égalité (B.1) en $x = 1$, nous obtenons $P_n(1) = 1$.

(ii). Soit $m > n \geqslant 0$. En intégrant $n+1$ fois par parties, nous obtenons

$$\begin{aligned} & 2^{m+n} m! n! \int_{-1}^{1} P_n(x)P_m(x)\mathrm{d}x \\ &= \sum_{j=0}^{n} \left((-1)^j \frac{\mathrm{d}^{n+j}}{\mathrm{d}x^{n+j}} \left\{ \left(x^2-1\right)^n \right\} \frac{\mathrm{d}^{m-1-j}}{\mathrm{d}x^{m-1-j}} \left\{ \left(x^2-1\right)^m \right\} \right) \Bigg|_{-1}^{1} \\ &\quad + (-1)^{n+1} \int_{-1}^{1} \frac{\mathrm{d}^{2n+1}}{\mathrm{d}x^{2n+1}} \left\{ \left(x^2-1\right)^n \right\} \frac{\mathrm{d}^{m-n-1}}{\mathrm{d}x^{m-n-1}} \left\{ \left(x^2-1\right)^m \right\} \mathrm{d}x \end{aligned}$$

Comme $x = \pm 1$ sont des zéros d'ordre m du polynôme $\left(x^2-1\right)^m$, les dérivées jusqu'à l'ordre $m-1$ de ce polynôme s'annulent en $x = \pm 1$ et les $n+1$ premiers termes du membre droit valent 0. Le polynôme $\left(x^2-1\right)^n$ étant de degré $2n$, sa dérivée d'ordre $2n+1$ est identiquement nulle et la propriété (ii) est démontrée.

(iii). Comme au point (ii), en intégrant n fois par parties, nous obtenons

$$\begin{aligned} 2^{2n}(n!)^2 \int_{-1}^{1} (P_n(x))^2 \, \mathrm{d}x &= (-1)^n \int_{-1}^{1} \frac{\mathrm{d}^{2n}}{\mathrm{d}x^{2n}} \left\{ \left(x^2-1\right)^n \right\} \left(x^2-1\right)^n \mathrm{d}x \\ &= (-1)^n (2n)! \int_{-1}^{1} \left(x^2-1\right)^n \mathrm{d}x. \end{aligned}$$

Posons

$$I_n = \int_{-1}^{1} \left(x^2-1\right)^n \mathrm{d}x, \ n \geqslant 0.$$

En intégrant par parties, nous obtenons, pour $n \geqslant 1$,

$$\begin{aligned} I_n &= x \left(x^2-1\right)^n \Big|_{-1}^{1} - 2n \int_{-1}^{1} x^2 \left(x^2-1\right)^{n-1} \mathrm{d}x \\ &= -2n \int_{-1}^{1} \left(x^2-1+1\right) \left(x^2-1\right)^{n-1} \mathrm{d}x \\ &= -2n \left(I_n + I_{n-1}\right), \end{aligned}$$

c'est–à–dire

$$I_n = -\frac{2n}{2n+1} I_{n-1}.$$

En itérant cette relation et avec $I_0 = 2$, nous obtenons

$$I_n = (-1)^n \frac{2^{2n+1}(n!)^2}{(2n+1)!}$$

et donc la propriété (iii).

(iv). Nous obtenons P_0 et P_1 par simple calcul. Par (i), pour $n \geqslant 0$, le polynôme

$$Q_n(x) = P_{n+1}(x) - \frac{2n+1}{n+1} x P_n(x)$$

appartient à $\mathbb{P}_n$ et s'écrit

$$Q_n = \sum_{j=0}^{n} \alpha_j P_j,$$

avec, par le point (ii),

$$\alpha_j = \frac{(\, Q_n \,|\, P_j \,)}{(\, P_j \,|\, P_j \,)}, \ 0 \leqslant j \leqslant n,$$

où $(\,.\,|\,.\,)$ désigne le produit scalaire de $L^2([-1,1])$.

Supposons $n \geqslant 1$. Il reste à montrer que $\alpha_{n-1} = -n/(n+1)$ et $\alpha_j = 0$, si $j \neq n-1$.

Soit $j < n-1$. Le polynôme $xP_j(x)$ est de degré inférieur ou égal à $n-1$ et donc par (ii)

$$(\,xP_n\,|\,P_j\,) = (\,P_n\,|\,xP_j\,) = 0.$$

Toujours par (ii), $(\,P_{n+1}\,|\,P_j\,) = 0$. Il suit $\alpha_j = 0$.

Comme $x(P_n(x))^2$ est une fonction impaire,

$$(\,xP_n\,|\,P_n\,) = \int_{-1}^{1} x\,(P_n(x))^2\,\mathrm{d}x = 0.$$

À nouveau, comme $(\,P_{n+1}\,|\,P_n\,) = 0$, il suit $\alpha_n = 0$.
Calculons α_{n-1}. Le polynôme xP_{n-1} appartient à $\mathbb{P}_n$ et s'écrit donc

$$xP_{n-1} = \sum_{j=0}^{n} \beta_j P_j.$$

En identifiant le terme de plus haut degré, nous obtenons

$$\beta_n = \frac{n}{2n-1}.$$

Avec (ii) et (iii), nous obtenons

$$(\,xP_n\,|\,P_{n-1}\,) = (\,P_n\,|\,xP_{n-1}\,) = (\,P_n\,|\,P_n\,)\,\beta_n$$

et

$$\alpha_{n-1} = -\frac{(2n+1)\,(\,xP_n\,|\,P_{n-1}\,)}{(n+1)\,(\,P_{n-1}\,|\,P_{n-1}\,)} = -\frac{n}{n+1}.$$

(v). Fixons $n \geqslant 1$. Notons $-1 < x_1 < x_2 < \ldots < x_m < 1$ les points où P_n change de signe. En particulier, ces points sont des racines de P_n, donc $0 \leqslant m \leqslant n$ (car P_n est de degré n). Considérons le polynôme de degré m

$$Q(x) = \prod_{j=1}^{m}(x - x_j) \text{ (ou } Q(x) = 1 \text{ si } m = 0).$$

Le polynôme $Q(x) \cdot P_n(x)$ conserve alors le même signe sur l'intervalle $[-1,1]$ et donc

$$(\,Q\,|\,P_n\,) = \int_{-1}^{1} Q(x)P_n(x)\mathrm{d}x \neq 0.$$

Comme P_n est orthogonal à $\mathbb{P}_{n-1}$, $Q \notin \mathbb{P}_{n-1}$, *i.e.* $m \geqslant n$. Au total, nous avons $m = n$ et la propriété (v). ■

Voici encore une propriété sur les racines des polynômes P_n :

Proposition B.2 *Les racines de deux polynômes de Legendre consécutifs alternent,* i.e. *si* $-1 < x_1^{(n)} < \ldots < x_n^{(n)} < 1$ *désignent les racines de* P_n*, alors, nous avons*

$$-1 < x_1^{(n+1)} < x_1^{(n)} < x_2^{(n+1)} < x_2^{(n)} < \ldots < x_n^{(n+1)} < x_n^{(n)} < x_{n+1}^{(n+1)} < 1. \tag{B.2}$$

Preuve. Nous avons $P_1(x) = x$ et $P_2(x) = (3x^2 - 1)/2$, donc la propriété est vérifiée pour $n = 1$:

$$-1 < x_1^{(2)} = -\sqrt{1/3} < x_1^{(1)} = 0 < x_2^{(2)} = \sqrt{1/3} < 1.$$

Supposons qu'elle le soit pour $n-1$, *i.e.*

$$-1 < x_1^{(n)} < x_1^{(n-1)} < x_2^{(n)} < x_2^{(n-1)} < \ldots < x_{n-1}^{(n)} < x_{n-1}^{(n-1)} < x_n^{(n)} < 1.$$

En utilisant la relation de récurrence (iii) de la proposition précédente,

$$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x),$$

et le fait que $P_l(1) = 1$ et que P_l est de même parité que l pour tout l, nous obtenons,

$$P_{n+1}(1) = 1 > 0,\ P_{n+1}(x_n^{(n)}) < 0,\ P_{n+1}(x_{n-1}^{(n)}) > 0, \ldots$$

jusqu'à

$$\begin{array}{lllllll} P_{n+1}(x_1^{(n)}) & < & 0, & P_{n+1}(-1) & = & 1 & > 0 \quad \text{si } n \text{ est impair,} \\ P_{n+1}(x_1^{(n)}) & > & 0, & P_{n+1}(-1) & = & -1 & < 0 \quad \text{si } n \text{ est pair.} \end{array}$$

Nous déduisons alors les inégalités (B.2) par continuité. ■

B.2 Égalité de Christoffel–Darboux

Pour $n \geqslant 0$, considérons P_n le polynôme de Legendre de degré n,

$$h_n = \int_{-1}^{1} (P_n(x))^2 \, \mathrm{d}x = \frac{2}{2n+1}$$

sa norme au carré et

$$\gamma_n = \frac{(2n)!}{2^n (n!)^2}$$

son coefficient dominant (voir proposition B.1).

Lemme B.3 (Égalité de Christoffel–Darboux) *Avec les notations introduites, nous avons l'égalité*

$$\sum_{j=0}^{n} \frac{1}{h_j} P_j(x) P_j(y) = \frac{\gamma_n}{\gamma_{n+1} h_n} \cdot \frac{P_{n+1}(x) P_n(y) - P_n(x) P_{n+1}(y)}{x - y} \tag{B.3}$$

pour $n \geqslant 0$.

Preuve. Par récurrence sur n. Pour $n = 0$, les deux expressions valent $1/2$. Supposons l'égalité vraie pour $n-1$ et montrons-la pour n. Nous avons

$$\sum_{j=0}^{n} \frac{1}{h_j} P_j(x)P_j(y) = \frac{\gamma_{n-1}}{\gamma_n h_{n-1}} \cdot \frac{P_n(x)P_{n-1}(y) - P_{n-1}(x)P_n(y)}{x-y} + \frac{1}{h_n} P_n(x)P_n(y).$$

De la proposition B.1 (iv), nous avons

$$P_{n-1}(x) = \frac{n+1}{n}\left(\frac{2n+1}{n+1} x P_n(x) - P_{n+1}(x)\right)$$

et donc

$$\begin{aligned} P_n(x)P_{n-1}(y) - P_{n-1}(x)P_n(y) &= -\frac{2n+1}{n}(x-y)P_n(x)P_n(y) \\ &\quad + \frac{n+1}{n}\left(P_{n+1}(x)P_n(y) - P_n(x)P_{n+1}(y)\right). \end{aligned}$$

Ainsi,

$$\begin{aligned} \sum_{j=0}^{n} \frac{1}{h_j} P_j(x)P_j(y) &= \left(-\frac{\gamma_{n-1}}{\gamma_n h_{n-1}} \cdot \frac{2n+1}{n} + \frac{1}{h_n}\right) P_n(x)P_n(y) \\ &\quad + \frac{\gamma_{n-1}}{\gamma_n h_{n-1}} \cdot \frac{n+1}{n} \cdot \frac{P_{n+1}(x)P_n(y) - P_n(x)P_{n+1}(y)}{x-y}. \end{aligned}$$

Nous avons

$$\frac{\gamma_{n-1}}{\gamma_n h_{n-1}} = \frac{(2n-2)!}{2^{n-1}((n-1)!)^2} \cdot \frac{2^n (n!)^2}{(2n)!} \cdot \frac{2n-1}{2} = \frac{n}{2},$$

donc

$$\begin{aligned} -\frac{\gamma_{n-1}}{\gamma_n h_{n-1}} \cdot \frac{2n+1}{n} + \frac{1}{h_n} &= 0, \\ \frac{\gamma_{n-1}}{\gamma_n h_{n-1}} \cdot \frac{n+1}{n} = \frac{n+1}{2} &= \frac{\gamma_n}{\gamma_{n+1} h_n} \end{aligned} \tag{B.4}$$

et l'égalité (B.3) suit. ■

B.3 Formule de quadrature de Gauss–Legendre

Fixons $n \geqslant 1$. Notons $x_1, \ldots, x_n$ les racines du polynôme de Legendre P_n. Considérons les polynômes élémentaires de Lagrange en ces points :

$$L_k(x) = \prod_{\substack{j=1 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j},\ 1 \leqslant k \leqslant n.$$

Ils vérifient $L_k(x_j) = \delta_{kj}$, $1 \leqslant k, j \leqslant n$. Pour une fonction continue $f : [-1, 1] \to \mathbb{R}$, notons

$$f_n(x) = \sum_{k=1}^{n} f(x_k) L_k(x)$$

le polynôme d'interpolation de Lagrange (de $\mathbb{P}_{n-1}$) qui coïncide avec f aux points $x_1, \ldots, x_n$.
Nous considérons alors l'estimation de $\int_{-1}^{1} f(x)\mathrm{d}x$ suivante :

$$J_n(f) = \int_{-1}^{1} f_n(x)\mathrm{d}x = \sum_{k=1}^{n} a_k f(x_k), \tag{B.5}$$

où

$$a_k = \int_{-1}^{1} L_k(x)\mathrm{d}x, \ 1 \leqslant k \leqslant n.$$

La formule (B.5) est appelée *formule de quadrature de Gauss–Legendre à n points* et les points $x_1, \ldots, x_n$ *points de Gauss.*

Les coefficients a_k, appelés aussi *poids*, s'expriment de manière plus simple :

Proposition B.4 *Nous avons, pour $1 \leqslant k \leqslant n$,*

$$a_k = -\frac{2}{(n+1)P_{n+1}(x_k)P_n'(x_k)} = \frac{2}{nP_{n-1}(x_k)P_n'(x_k)}.$$

Preuve. Remarquons que les racines de P_n sont simples et donc n'annulent pas la dérivée de P_n. De plus, de la proposition B.2, il suit que les racines de deux polynômes de Legendre consécutifs sont toutes distinctes, par conséquent, $P_{n+1}(x_k), P_{n-1}(x_k) \neq 0$.

Les polynômes

$$L_k(x) \text{ et } \frac{P_n(x)}{(x-x_k)P_n'(x_k)}$$

sont de degré $n-1$, s'annulent en $x = x_j$, $1 \leqslant j \leqslant n$, $j \neq k$ et valent 1 en $x = x_k$, donc ils sont égaux ; ainsi,

$$a_k = \frac{1}{P_n'(x_k)} \int_{-1}^{1} \frac{P_n(x)}{x - x_k}\mathrm{d}x. \tag{B.6}$$

En prenant $y = x_k$ dans l'égalité de Christoffel–Darboux (B.3), nous obtenons

$$\sum_{j=0}^{n-1} \frac{1}{h_j} P_j(x)P_j(x_k) = -\frac{\gamma_n}{\gamma_{n+1}h_n} \cdot \frac{P_n(x)P_{n+1}(x_k)}{x - x_k} \tag{B.7}$$

et, comme

$$\int_{-1}^{1} P_j(x)\mathrm{d}x = \left(\frac{\mathrm{d}^{j-1}}{\mathrm{d}x^{j-1}} \left\{ \left(x^2-1\right)^j \right\} \right) \Big|_{-1}^{1} = 0, \ j \geqslant 1,$$

nous avons, en intégrant (B.7),

$$-\frac{\gamma_n P_{n+1}(x_k)}{\gamma_{n+1}h_n} \int_{-1}^{1} \frac{P_n(x)}{x - x_k}\mathrm{d}x = \frac{P_0(x_k)}{h_0} \int_{-1}^{1} P_0(x)\mathrm{d}x = 1.$$

Ainsi, de (B.6) et avec (B.4),

$$a_k = -\frac{1}{P_n'(x_k)} \cdot \frac{\gamma_{n+1} h_n}{\gamma_n P_{n+1}(x_k)} = -\frac{2}{(n+1) P_{n+1}(x_k) P_n'(x_k)}.$$

Enfin, de la proposition B.1 (iv), nous avons

$$P_{n+1}(x_k) = \frac{2n+1}{n+1} x_k P_n(x_k) - \frac{n}{n+1} P_{n-1}(x_k) = -\frac{n}{n+1} P_{n-1}(x_k),$$

ce qui donne la deuxième égalité. ■

Remarquons que, comme le polynôme P_n est de même parité que n, les n racines de P_n vérifient

$$x_k = -x_{n+1-k},\ 1 \leqslant k \leqslant n,$$

et, d'après la proposition précédente, les poids satisfont

$$a_k = a_{n+1-k},\ 1 \leqslant k \leqslant n;$$

de plus, de $P_l(1) = 1$, $l \geqslant 1$ et de la proposition B.2, nous déduisons que

$$a_k > 0,\ 1 \leqslant k \leqslant n.$$

B.4 Estimation de l'erreur

Nous allons tout d'abord montrer que la formule de quadrature de Gauss–Legendre à n points (B.5) est exacte pour les polynômes de degré inférieur ou égal à $2n-1$, ensuite, nous allons utiliser l'interpolation d'Hermite pour estimer l'erreur dans un cadre plus général.

B.4.1 Cas particulier

Proposition B.5 *Avec les notations précédentes, si $f \in \mathbb{P}_{2n-1}$, alors*

$$J_n(f) = \int_{-1}^{1} f(x)dx$$

Preuve. Notons f_n le polynôme d'interpolation de Lagrange (de $\mathbb{P}_{n-1}$) qui coïncide avec f aux points de Gauss $x_1, \ldots, x_n$ et $r_n = f - f_n$. Supposons $f \in \mathbb{P}_{2n-1}$; alors r_n appartient à $\mathbb{P}_{2n-1}$ et s'annule en $x_1, \ldots, x_n$, il est donc divisible par P_n :

$$r_n(x) = P_n(x) s_n(x),$$

où $s_n \in \mathbb{P}_{n-1}$. Ainsi, par la proposition B.1(ii), nous avons

$$\int_{-1}^{1} r_n(x)\mathrm{d}x = 0,$$

ce qui termine la preuve. ■

B.4.2 Interpolation d'Hermite

Soit f une fonction dérivable sur l'intervalle $[-1,1]$ (ou plus généralement $[a,b]$) et $x_1 < \ldots < x_n$ n points de cet intervalle. Le *polynôme d'interpolation d'Hermite* de f est le polynôme $F_n \in \mathbb{P}_{2n-1}$ vérifiant

$$F_n(x_k) = f(x_k),\ F_n'(x_k) = f'(x_k),\ 1 \leqslant k \leqslant n. \tag{B.8}$$

Montrons l'unicité. Soit $F_{n,1}$, $F_{n,2}$ deux tels polynômes. Comme ils prennent les mêmes valeurs aux points x_j, nous avons

$$F_{n,1}(x) - F_{n,2}(x) = (x-x_1)\ldots(x-x_n)Q(x),$$

où $Q \in \mathbb{P}_{n-1}$. L'égalité $F_{n,1}'(x_k) - F_{n,2}'(x_k) = 0$ entraîne $Q(x_k) = 0$ pour $1 \leqslant k \leqslant n$. Par conséquent, Q est identiquement nul et $F_{n,1} = F_{n,2}$.

Donnons l'expression de ce polynôme et montrons ainsi l'existence. Cherchons F_n sous la forme

$$F_n(x) = \sum_{k=1}^{n} h_k(x) f(x_k) + \sum_{k=1}^{n} \tilde{h}_k(x) f'(x_k), \tag{B.9}$$

avec $h_k, \tilde{h}_k \in \mathbb{P}_{2n-1}$ vérifiant

$$h_k(x_j) = \delta_{kj} \quad \text{et} \quad h_k'(x_j) = 0, \tag{B.10}$$
$$\tilde{h}_k(x_j) = 0 \quad \text{et} \quad \tilde{h}_k'(x_j) = \delta_{kj}, \tag{B.11}$$

pour $1 \leqslant j,k \leqslant n$. Considérons les polynômes élémentaires de Lagrange

$$L_k(x) = \prod_{\substack{j=1 \\ j\neq k}}^{n} \frac{x-x_j}{x_k-x_j},\ 1 \leqslant k \leqslant n,$$

de degré $n-1$ et vérifiant $L_k(x_j) = \delta_{kj}$. Écrivons h_k et $\tilde{h}_k$ sous la forme

$$\begin{aligned} h_k(x) &= s_k(x)\left(L_k(x)\right)^2, \\ \tilde{h}_k(x) &= \tilde{s}_k(x)\left(L_k(x)\right)^2. \end{aligned}$$

Nous avons $h_k'(x) = s_k'(x)(L_k(x))^2 + 2s_k(x)L_k(x)L_k'(x)$. Il suffit que $s_k, \tilde{s}_k \in \mathbb{P}_1$ vérifient

$$\begin{aligned} s_k(x_k) = 1 \quad &\text{et} \quad s_k'(x_k) = -2L_k'(x_k), \\ \tilde{s}_k(x_k) = 0 \quad &\text{et} \quad \tilde{s}_k'(x_k) = 1, \end{aligned}$$

pour que (B.10) et (B.11) soient satisfaits. Ainsi, $s_k(x) = 1 - 2L_k'(x_k)(x-x_k)$ et $\tilde{s}_k(x) = x - x_k$ donnent

$$\begin{aligned} h_k(x) &= \left(1 - 2L_k'(x_k)(x-x_k)\right)\left(L_k(x)\right)^2, \\ \tilde{h}_k(x) &= (x-x_k)\left(L_k(x)\right)^2 \end{aligned}$$

et le polynôme (B.9) vérifiant (B.8).

Proposition B.6 *Avec les notations ci–dessus et en supposant f de classe $\mathcal{C}^{2n}$, l'erreur commise entre f et F_n en un point $x \in [-1,1]$ est*

$$E(x) = f(x) - F_n(x) = \frac{f^{(2n)}(\xi)}{(2n)!} \left(\pi_n(x)\right)^2,$$

où $\pi_n(x) = (x - x_1) \dots (x - x_n)$ et $\xi = \xi(x) \in [-1,1]$.

Preuve. Fixons $x_0 \in [-1,1]$ et supposons $x_0 \neq x_j$, $1 \leqslant j \leqslant n$ (sinon $E(x_0) = 0$ et il n'y a rien à faire). La fonction

$$G(x) = f(x) - F_n(x) - \frac{f(x_0) - F_n(x_0)}{\left(\pi_n(x_0)\right)^2} \left(\pi_n(x)\right)^2$$

s'annule en $x_0, x_1, \dots, x_n$ ($n+1$ points distincts), donc par le théorème de Rolle, G' s'annule en n points distincts et différents de $x_0, \dots, x_n$. D'autre part, par construction de F_n, G' s'annule aussi en $x_1, \dots, x_n$. Ainsi, G' s'annule en $2n$ points distincts et, en appliquant le théorème de Rolle $2n-1$ fois, nous trouvons un point ξ entre les points $x_0, \dots, x_n$ avec $G^{(2n)}(\xi) = 0$, c'est–à–dire,

$$f^{(2n)}(\xi) - \frac{f(x_0) - F_n(x_0)}{\left(\pi_n(x_0)\right)^2} \cdot (2n)! = 0.$$

Nous avons donc

$$E(x_0) = f(x_0) - F_n(x_0) = \frac{f^{(2n)}(\xi)}{(2n)!} \left(\pi_n(x_0)\right)^2.$$

■

De cette proposition, il suit immédiatement

$$\left| \int_{-1}^{1} f(x)\mathrm{d}x - \int_{-1}^{1} F_n(x)\mathrm{d}x \right| \leqslant \frac{1}{(2n)!} \max_{\xi \in [-1,1]} \left| f^{(2n)}(\xi) \right| \int_{-1}^{1} \left(\pi_n(x)\right)^2 \mathrm{d}x \quad \text{(B.12)}$$

pour une fonction f de classe $\mathcal{C}^{2n}$.

B.4.3 Erreur de la formule de quadrature de Gauss–Legendre

Proposition B.7 *Soit f une fonction définie sur l'intervalle $[-1,1]$ et $n \geqslant 1$. Supposons que f est de classe $\mathcal{C}^{2n}$ et considérons $J_n(f)$, la formule de quadrature de Gauss–Legendre à n points (B.5). Nous avons alors*

$$E_n = \left| \int_{-1}^{1} f(x)dx - J_n(f) \right| \leqslant \frac{2^{2n+1}(n!)^4}{(2n+1)\left((2n)!\right)^3} \max_{\xi \in [-1,1]} \left| f^{(2n)}(\xi) \right|.$$

Preuve. Le polynôme d'interpolation d'Hermite F_n (B.9) relativement aux points de Gauss $x_1, \dots, x_n$ (racines du polynôme de Legendre P_n) est de degré inférieur ou égal à $2n-1$. Ainsi, par la proposition B.5, la formule de quadrature de Gauss–Legendre à n points est exacte pour ce polynôme, *i.e.*

$$\int_{-1}^{1} F_n(x)\mathrm{d}x = J_n(F_n).$$

Par construction de F_n, F_n et f prennent les mêmes valeurs aux points x_k, $1 \leqslant k \leqslant n$ et donc $J_n(F_n) = J_n(f)$. Il suit, avec l'inégalité (B.12),

$$E_n = \left| \int_{-1}^{1} f(x)\mathrm{d}x - J_n(f) \right| \leqslant \frac{1}{(2n)!} \max_{\xi \in [-1,1]} \left| f^{(2n)}(\xi) \right| \int_{-1}^{1} (\pi_n(x))^2 \, \mathrm{d}x.$$

Comme $\pi_n(x) = (x-x_1)\ldots(x-x_n)$ est un polynôme de même degré et ayant les mêmes racines que le polynôme de Legendre P_n, nous avons $\pi_n(x) = \gamma_n^{-1} P_n(x)$, où γ_n est le coefficient dominant de P_n. Ainsi, par la proposition B.1 (i) et (iii),

$$\int_{-1}^{1} (\pi_n(x))^2 \, \mathrm{d}x = \frac{1}{\gamma_n^2} \int_{-1}^{1} (P_n(x))^2 \, \mathrm{d}x = \frac{2^{2n+1}(n!)^4}{(2n+1)\left((2n)!\right)^2},$$

ce qui termine la preuve. ■

B.5 Exemples

Donnons explicitement la formule de quadrature de Gauss–Legendre à $1, 2, 3$ et 4 points.

Les premiers polynômes de Legendre et leur dérivée sont

$$\begin{aligned}
P_0(x) &= 1, & P_0'(x) &= 0,\\
P_1(x) &= x, & P_1'(x) &= 1,\\
P_2(x) &= \tfrac{3}{2}x^2 - \tfrac{1}{2}, & P_2'(x) &= 3x,\\
P_3(x) &= \tfrac{5}{2}x^3 - \tfrac{3}{2}x, & P_3'(x) &= \tfrac{15}{2}x^2 - \tfrac{3}{2},\\
P_4(x) &= \tfrac{35}{8}x^4 - \tfrac{15}{4}x^2 + \tfrac{3}{8}, & P_4'(x) &= \tfrac{35}{2}x^3 - \tfrac{15}{2}x.
\end{aligned}$$

Les graphes de $P_0, \ldots, P_4$ sont représentés sur la figure 47.

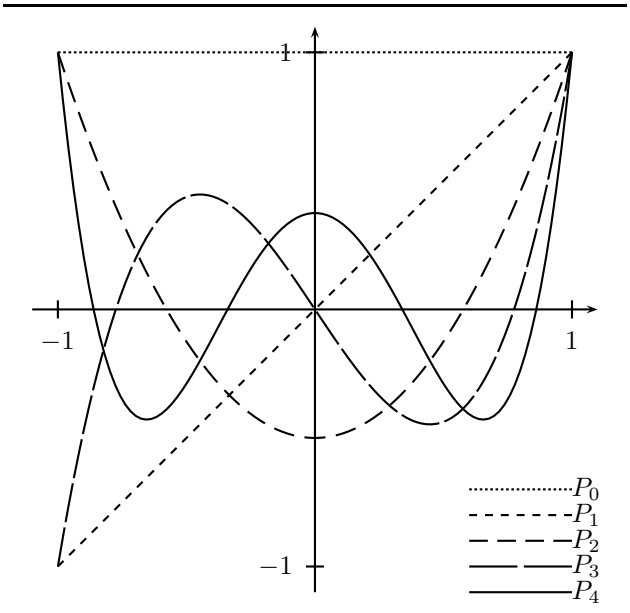

Figure 47.

Les points de Gauss, les poids correspondants et l'estimation des erreurs (avec $M_n = \max_{\xi\in[-1,1]} |f^{(2n)}(\xi)|$) sont donnés dans le tableau 45.

n	Points de Gauss					Poids					Erreur		
1	x_1	$=$	0			a_1	$=$	2			E	$\leqslant$	$1/3 \cdot M_1$
2	x_1	$=$	$\sqrt{1/3}$	$\approx$	0.57735	a_1	$=$	1			E	$\leqslant$	$1/135 \cdot M_2$
	x_2	$=$	$-\sqrt{1/3}$	$\approx$	-0.57735	a_2	$=$	1				$\approx$	$7.4074\mathrm{E}-03 \cdot M_2$
3	x_1	$=$	$\sqrt{3/5}$	$\approx$	0.77460	a_1	$=$	$5/9$	$\approx$	0.55556	E	$\leqslant$	$1/15750 \cdot M_3$
	x_2	$=$	0			a_2	$=$	$8/9$	$\approx$	0.88889		$\approx$	$6.3492\mathrm{E}-05 \cdot M_3$
	x_3	$=$	$-\sqrt{3/5}$	$\approx$	-0.77460	a_3	$=$	$5/9$	$\approx$	0.55556			
4	x_1			$\approx$	-0.86114	a_1			$\approx$	0.34786	E	$\lesssim$	$2.8795\mathrm{E}-07 \cdot M_4$
	x_2			$\approx$	-0.33998	a_2			$\approx$	0.65215			
	x_3			$\approx$	0.33998	a_3			$\approx$	0.65215			
	x_4			$\approx$	0.86114	a_4			$\approx$	0.34786			

Tableau 45.

Bibliographie

[1] R. A. Adams. *Sobolev Spaces.* Academic Press, 1975.

[2] H. Anton et C. Rorres. *Elementary Linear Algebra with Applications.* John Wiley & Sons, Inc., 1987.

[3] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, 1994.

[4] P. Azérad. Analyse et approximation du problème de Stokes dans un bassin peu profond. *C. R. Acad. Sci. Paris Sér. I Math.*, 318(1) :53–58, 1994.

[5] P. Azérad. *Analyse des équations de Navier–Stokes en bassin peu profond et de l'équation de transport.* Thèse de doctorat, Université de Neuchâtel, 1996.

[6] I. Babuška. Error–bounds for finite element method. *Numer. Math.*, 16 :322–333, 1971.

[7] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine et H. van der Vorst. *Templates for the Solution of Linear Systems : Building Blocks for Iterative Methods, 2nd Edition.* SIAM, Philadelphia, PA, 1994. Disponible sur ftp.netlib.org/templates/templates.ps.

[8] M. Benzi. Preconditioning techniques for large linear systems : a survey. *J. Comput. Phy.*, 182(2) :418–477, 2002.

[9] M. Benzi, J. K. Cullum et M. Tůma. Robust approximate inverse preconditioning for the conjugate gradient method. *SIAM J. Sci. Comput.*, 22(4) :1318–1332, 2000.

[10] M. Benzi, C. D. Meyer et M. Tůma. A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM J. Sci. Comput.*, 17(5) :1135–1149, 1996.

[11] M. Benzi et M. Tůma. A robust incomplete factorization preconditioner for positive definite matrices. *Numer. Linear Algebra Appl.*, 10(5–6) :385–400, 2003.

[12] M. Bercovier et O. Pironneau. Error estimates for finite element method solution of the Stokes problem in the primitive variables. *Numer. Math.*, 33(2) :211–224, 1979.

[13] O. Besson. Finite element solution of Navier–Stokes equations in shallow domains. *Ann. Math. Blaise Pascal*, 9(2) :161–180, 2002.

[14] O. Besson, M. Laydi et R. Touzani. Un modèle asymptotique en océanographie. *C. R. Acad. Sci. Paris Sér. I Math.*, 310(9) :661–665, 1990.

[15] O. Besson et M. R. Laydi. Some estimates for the anisotropic Navier–Stokes equations and for the hydrostatic approximation. *RAIRO Modél. Math. Anal. Numér.*, 26(7) :855–865, 1992.

[16] D. Bresch, T. Huck et M. Sy. Circulation thermohaline et équations planétaires géostrophiques : propriétés physiques, numériques et mathématiques. *Ann. Math. Blaise Pascal*, 9(2) :181–212, 2002.

[17] F. Brezzi et M. Fortin. *Mixed and Hybrid Finite Element Methods.* Springer–Verlag, 1991.

[18] P. G. J. T. Brummelhuis, A. W. Heemink et H. F. P. van den Boogaard. Identification of shallow sea models. *Internat. J. for Numer. Methods Fluids*, 17 :637–665, 1993.

[19] R. Bulirsch et J. Stoer. *Introduction to Numerical Analysis.* Springer–Verlag, 1980.

[20] A. J. Chorin et J. E. Marsden. *A Mathematical Introduction to Fluid Mechanics.* Springer–Verlag, 1979.

[21] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North–Holland, 1978.

[22] J. Dongarra, S. Huss-Lederman, S. Otto, M. Snir et D. Walker. *MPI - The Complete Reference : The MPI Core, second edition*, volume 1. The MIT Press, 1998.

[23] G. Duvaut. *Mécanique des milieux continus.* Masson, Paris, 1990.

[24] V. Girault et P.-A. Raviart. *Finite Element Methods for Navier–Stokes Equations.* Springer–Verlag, 1986.

[25] A. Grama, A. Gupta, G. Karypis et V.Kumar. *Introduction to Parallel Computing.* The Benjamin/Cummings Publishing Company, Inc., 1994.

[26] W. Gropp, S. Huss-Lederman, A. Lumsdaine, E. Lusk, B. Nitzberg, W. Saphir et M. Snir. *MPI - The Complete Reference : The MPI Extensions*, volume 2. The MIT Press, 1998.

[27] M. Grote et T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, 18(3) :838–853, 1997.

[28] J.-L. Guermond. Remarques sur les méthodes de projection pour l'approximation des équations de Navier–Stokes. *Numer. Math.*, 67(4) :465–473, 1994.

[29] J.-L. Guermond. Some implementations of projection methods for Navier–Stokes equations. *RAIRO Modél. Math. Anal. Numér.*, 30(5) :637–667, 1996.

[30] J.-L. Guermond. Un résultat de convergence d'ordre deux en temps pour l'approximation des équations de Navier–Stokes par une technique de projection incrémentale. *M2AN Math. Model. Numer. Anal.*, 33(1) :169–189, 1999. Aussi dans C. R. Acad. Paris Sér. I Math., 325(12) :1329–1332, 1997.

[31] J.-L. Guermond et L. Quartapelle. On the approximation of the unsteady Navier–Stokes equations by finite element projection methods. *Numer. Math.*, 80(2) :207–238, 1998.

[32] J.-L. Guermond et J. Shen. Quelques résultats nouveaux sur les méthodes de projection. *C. R. Acad. Sci. Paris Sér. I Math.*, 333(12) :1111–1116, 2001.

[33] J.-L. Guermond et J. Shen. Velocity–correction projection methods for incompressible flows. *SIAM J. Numer. Anal.*, 41(1) :112–134, 2003.

[34] E. Hairer, S. P. Nørsett et G. Wanner. *Solving Ordinary Differential Equations I.* Springer–Verlag, 1987.

[35] T. J. R. Hughes et J. E. Marsden. *A Short Course in Fluid Mechanics.* Publish or Perish, Inc., 1976.

[36] L. Yu. Kolotilina et A. Yu. Yeremin. Factorized sparse approximate inverse preconditionings I. Theory. *SIAM J. Matrix Anal. Appl.*, 14(1) :45–58, 1993.

[37] L. Landau et E. Lifchitz. *Mécanique des fluides.* Éditions Mir, Moscou, 1971.

[38] J. A. Meijerink et H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31(137) :148–162, 1977.

[39] J. Nečas. *Les méthodes directes en théorie des équations elliptiques.* Masson, 1967.

[40] J. Pedlosky. *Geophysical Fluid Dynamics.* Springer–Verlag, 1987.

[41] J. Pedlosky. *Ocean Circulation Theory.* Springer–Verlag, 1996.

[42] O. Pironneau. *Méthodes des éléments finis pour les fluides.* Masson, Paris, 1988.

[43] L. Quartapelle. *Numerical Solution of the Incompressible Navier–Stokes Equations.* Birkhäuser Verlag, 1993.

[44] P. Rabinowitz et A. Ralston. *A First Course in Numerical Analysis (Second Edition).* International Student Edition, McGraw–Hill International Book Company, 1978.

[45] P.-A. Raviart et J.-M. Thomas. *Introduction à l'analyse numérique des équations aux dérivées partielles.* Dunod, Paris, 1998.

[46] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, 1996.

[47] J. Serrin. Mathematical principles of classical fluid mechanics. *Handbuch der Physik*, 8/1 :125–263, 1959.

[48] J. Shen. On error estimates of the projection methods for the Navier–Stokes equations : second–order schemes. *Math. Comp.*, 65(215) :1039–1065, 1996.

[49] R. Stenberg. On some three–dimensional finite elements for incompressible media. *Comput. Methods Appl. Mech. Engrg.*, 63(3) :261–269, 1987.

[50] R. Stenberg. Error analysis of some finite element methods for the Stokes problem. *Math. Comp.*, 54(190) :495–508, 1990.

[51] R. Stenberg. A technique for analysing finite element methods for viscous incompressible flow. *Internat. J. Numer. Methods Fluids*, 11(6) :935–948, 1990.

[52] G. Strang. *Introduction to Linear Algebra (third edition).* Wellesley–Cambridge Press, 2003.

[53] J. Straubhaar. Parallel preconditioners for the conjugate gradient algorithm using gram–schmidt and least squares methods. Soumis à Parallel Comput., 2007.

[54] J. Straubhaar. Preconditioners for the conjugate gradient algorithm using gram–schmidt and least squares methods. *Internat. J. Comput. Math.*, 84(1) :89–108, 2007.

[55] M. Sy. *Effets des termes diffusifs sur des modèles diphasiques et un modèle géophysique.* Thèse de doctorat, Université Blaise Pascal, 2005.

[56] R. Temam. *Navier–Stokes Equations.* North–Holland, 1984.

[57] J. van Kan. A second–order accurate pressure–correction scheme for viscous incompressible flow. *SIAM J. Sci. Stat. Comput.*, 7(3) :870–891, 1986.

[58] R. S. Varga. *Matrix Iterative Analysis.* Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1962.

[59] C. B. Vreugdenhil. *Numerical Methods for Shallow–Water Flow.* Springer–Verlag, 1994.

Mai 2007

Julien Straubhaar
Institut de mathématiques, Université de Neuchâtel,
Rue Emile–Argand 11, CH–2009 Neuchâtel
e–mail : julien.straubhaar@unine.ch