

Simulation d'écoulements Navier-Stokes incompressibles avec un demi milliard de degrés de liberté sur un seul GPU.

A. Chemin¹, J. Lambrechts², J.-F. Remacle¹

¹ Université catholique de Louvain, Institute of Mechanics, Materials and Civil Engineering (iMMC),
Bâtiment Euler, Avenue Georges Lemaître 4, 1348 Louvain-la-Neuve, Belgium,
{alexandre.chemin,jonathan.lambrechts,jean-francois.remacle}@uclouvain.be

Résumé — Ces travaux présentent une méthode développée afin de résoudre de manière efficace les équations incompressibles de Navier-Stokes sur GPU pour des maillages hexaédriques complets. La résolution est basée sur une séparation des équations de Navier-Stokes incompressibles menant à la résolution de quatre équations scalaires de Poisson. Une méthode éléments finis spectraux est utilisée pour discrétiser ces équations. La résolution est effectuée à l'aide d'un gradient conjugué préconditionné sans assembler les opérateurs concernés. Le préconditionneur utilisé est un préconditionneur Schwartz additif à deux échelles. Cette méthode permet de résoudre des problèmes d'un demi milliard de degrés de liberté sur un seul GPU.

Mots clés — Navier-Stokes incompressible, calcul GPU, méthode des éléments finis spectraux haut ordre, maillages hexaédriques.

1 Introduction

La méthode de résolution mise en place permet de résoudre efficacement les équations de Navier-Stokes incompressibles sur GPU pour des maillages hexaédriques complets. Les maillages hexaédriques présentent de nombreux avantages sur les maillages tétraédriques, notamment pour le traitement des couches limites en CFD [1]. De nombreux efforts ont été faits pour développer des méthodes rapides et efficaces pour la génération de maillages hex-dominants [3, 4] et il est maintenant possible de générer des maillages volumiques contenant plus de 80% d'éléments hexaédriques (par exemple avec GMSH [2]).

Une méthode de résolution des équations de Poisson basée sur une discrétisation éléments finis spectraux haut ordre sur GPU a été présentée dans [5]. Cette méthode est utilisée pour résoudre les quatre équations de Poisson provenant de la méthode de séparation des équation de Navier-Stokes incompressible utilisée, décrite dans [6]. L'utilisation de GPU pour la résolution de problèmes éléments finis fait l'objet de nombreux travaux. Sans entrer dans les détails, on peut citer [7] pour la résolution de problèmes éléments finis sur cluster GPU, [8] pour l'utilisation de méthodes multigrilles sur cluster GPU ainsi que [9, 10] pour l'efficacité de l'assemblage de matrices de rigidité sur GPU.

Nous allons rapidement présenter le principe de résolution mis en place, la méthode de séparation des équations de Navier-Stokes incompressibles, la méthode des éléments finis spectraux et la résolution des équations de Poisson sur GPU utilisée.

2 Principe de résolution

Nous nous intéressons à la résolution sur un domaine tridimensionnel Ω des équations suivantes :

$$\frac{\partial \mathbf{v}}{\partial t} = -\nabla p + \nu \mathbf{L}(\mathbf{v}) + \mathbf{N}(\mathbf{v}) \quad \text{in } \Omega,$$

avec

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega,$$

et

$$\begin{cases} \mathbf{L}(\mathbf{v}) \equiv \nabla^2 \mathbf{v} = \nabla(\nabla \cdot \mathbf{v}) - \nabla \times (\nabla \times \mathbf{v}) \\ \mathbf{N}(\mathbf{v}) \equiv -\frac{1}{2}[\mathbf{v} \cdot \nabla \mathbf{v} + \nabla(\mathbf{v} \cdot \mathbf{v})]. \end{cases}$$

Pour cela, nous utiliserons la méthode de séparation des équations de Navier-Stokes en quatre équations de Poisson décrite dans [6]. Ces quatre équations seront résolues à l'aide du solveur mis en place dans [5], basé sur un gradient conjugué préconditionné et une discrétisation éléments finis spectraux de celles-ci.

2.1 Méthode de séparation des équations de Navier-Stokes

Afin de résoudre les équations de Navier-Stokes présentées précédemment, il est proposé dans [6] de décomposer la résolution à un instant $t = t_n$ en trois sous étapes.

On pose tout d'abord :

$$\int_{t_n}^{t_{n+1}} \mathbf{N}(\mathbf{v}) dt = \Delta t \sum_{q=0}^{J_e-1} \beta_q \mathbf{N}(\mathbf{v}^{n-q})$$

un schéma d'intégration explicite Adams-Bashforth d'ordre J_e , et :

$$\int_{t_n}^{t_{n+1}} \mathbf{L}(\mathbf{v}) dt = \Delta t \sum_{q=0}^{J_i-1} \gamma_q \mathbf{L}(\mathbf{v}^{n+1-q})$$

un schéma d'intégration implicite Adams-Moulton d'ordre J_i .

Les trois sous étapes de résolution proposée sont :

$$\begin{cases} \frac{\hat{\mathbf{v}} - \mathbf{v}^n}{\Delta t} = \sum_{q=0}^{J_e-1} \beta_q \mathbf{N}(\mathbf{v}^{n-q}) & \text{in } \Omega \\ \frac{\hat{\mathbf{v}} - \hat{\mathbf{v}}}{\Delta t} = -\nabla \bar{p}^{n+1} & \text{in } \Omega \\ \frac{\mathbf{v}^{n+1} - \hat{\mathbf{v}}}{\Delta t} = \nu \sum_{q=0}^{J_i-1} \gamma_q \mathbf{L}(\mathbf{v}^{n+1-q}) & \text{in } \Omega \end{cases}$$

L'équation de pression peut être ramenée à la résolution de l'équation de Poisson suivante :

$$\begin{cases} \nabla^2 \bar{p}^{n+1} = \nabla \cdot \left(\frac{\hat{\mathbf{v}}}{\Delta t} \right) & \text{in } \Omega \\ \frac{\partial \bar{p}^{n+1}}{\partial n} = -\frac{\vec{v}_0 \cdot \mathbf{n} - \hat{\mathbf{v}} \cdot \mathbf{n}}{\Delta t} & \text{on } \partial\Omega \\ \mathbf{v}^{n+1} = \vec{v}_0 & \text{on } \partial\Omega. \end{cases}$$

La résolution du problème pour la détermination des champs de pression et vitesse à un instant t_n se ramène donc ici à la résolution de quatre équations de Poisson : une pour l'inconnue en pression et trois pour les composantes de vitesse. Cette méthode nous permet d'utiliser le solveur GPU pour équations de Poisson développé dans [5]. Ce solveur est basé sur une discrétisation éléments finis spectraux haut ordre en espace.

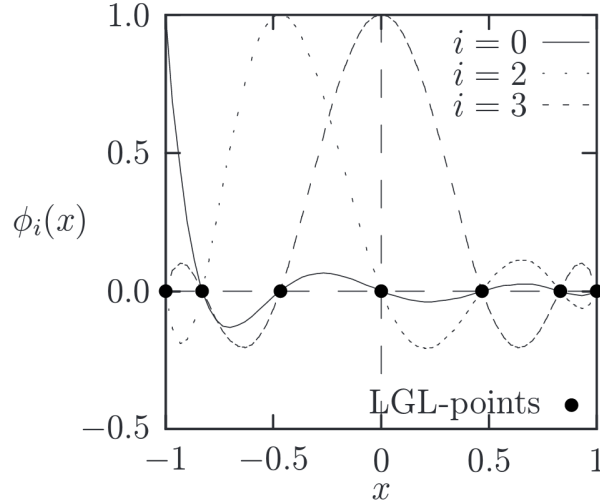
2.2 Méthode des éléments finis spectraux

La méthode de discrétisation spatiale utilisée ici est une discrétisation de type éléments finis pour laquelle les fonctions de forme polynomiales sont basées sur les points de Gauss-Lobatto-Legendre. Les

maillages considérés ici étant des maillages hexaédriques complets, les fonctions de forme tridimensionnelles sont obtenues par produit tensoriel des fonctions de forme unidimensionnelles. Pour un champ inconnu u , l'espace de recherche est le suivant :

$$u^e(\xi, \eta, \zeta) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n u_{ijk;e} \phi_i(\xi) \phi_j(\eta) \phi_k(\zeta)$$

où les ϕ_i sont les fonctions de forme éléments finis spectrales unidimensionnelles. La figure suivante représente ces fonctions pour un degré polynomial 6.



L'utilisation de la méthode de éléments finis spectraux haut ordre sur maillage hexaédrique permet de profiter pleinement de la parallélisation sur deux niveaux qu'apporte une architecture GPU.

2.3 Résolution des équations de Poisson sur GPU

Nous allons très brièvement présenter ici le solveur développé afin de résoudre le problème matriciel associé à une équation de Poisson. Ce système matriciel est résolu à l'aide d'un gradient conjugué préconditionné. L'opérateur associé à cette équation n'est jamais assemblé ou stocké pleinement sur le GPU, la mémoire disponible sur cette unité de calcul étant un point limitant. Tous les termes nécessaires à la résolution sont donc calculés chaque fois que cela est nécessaire.

Le préconditionneur additif Schwartz avec overlapping sur deux niveaux couple un préconditionnement local fin (haut ordre) effectué sur chaque élément, avec un préconditionnement grossier (bas ordre) sur l'ensemble de la structure. Le préconditionnement grossier est effectué sur le CPU de manière simultanée au préconditionnement fin qui est effectué sur le GPU.

Pour de plus amples détails on pourra se référer à [5].

3 Performances

Nous présenterons ici les performances obtenues sur un seul cas test. Le problème qui nous intéresse est la résolution du tourbillon de Taylor-Green. Il s'agit d'un problème défini par la solution analytique initiale :

$$\begin{aligned} u &= V_0 \sin\left(\frac{x}{L}\right) \cos\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right) \\ v &= -V_0 \cos\left(\frac{x}{L}\right) \sin\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right) \\ w &= 0 \\ p &= p_0 + \frac{\rho_0 V_0^2}{16} \left(\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right) \right) \left(\cos\left(\frac{2z}{L}\right) + 2 \right) \end{aligned}$$

Le problème est résolu dans l'enceinte d'un cube périodique tel que $-\pi L \leq x, y, z \leq \pi L$.

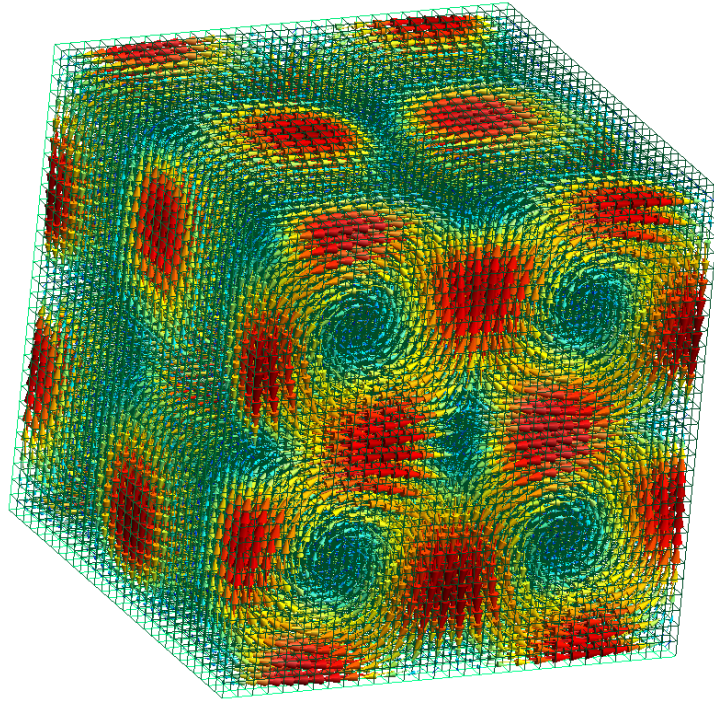


FIGURE 1 – Champ de vitesse initial.

En effectuant la résolution sur une carte graphique NVIDIA GeForce GTX 980M avec un total de $125 \cdot 10^6$ degrés de liberté (soit environ $31,25 \cdot 10^6$ degré de liberté pour chaque champs), les performances obtenues en fonction de l'ordre des fonctions éléments finis spectrales sont les suivantes :

Ordre	2	3	4	5	6	7
Gflops	33	60	69	82	99	99

La méthode mise en place montre des performances augmentant avec le degré des fonctions éléments finis spectrales. De plus l'implémentation utilisée permet, à quantité de mémoire constante, en augmentant le degré des fonctions de forme utilisées, d'augmenter le nombre de degrés de liberté maximal de la simulation.

Nous sommes par exemple parvenus à effectuer cette simulation pour un total de $0,5 \cdot 10^9$ degrés de liberté, soit $125 \cdot 10^6$ degrés de liberté par champs avec des fonctions de forme de degré 7 sur la carte NVIDIA Titan X.

Références

- [1] MA Puso and J Solberg. *A stabilized nodally integrated tetrahedral*, International Journal for Numerical Methods in Engineering, 67(6) :841-867, 2006.
- [2] C. Geuzaine and J.-F. Remacle. *Gmsh : A 3-D finite element mesh generator with built-in pre-and post-processing facilities*, International Journal for Numerical Methods in Engineering, 79(11) :1309-1331, 2009
- [3] T. C. Baudouin, J.-F. Remacle, E. Marchandise, F. Henrotte, and C. Geuzaine. *A frontal approach to hex-dominant mesh generation*, Advanced Modeling and Simulation in Engineering Sciences, 1(1) :1-30, 2014.
- [4] S. Yamakawa and K. Shimada. *Fully-automated hex-dominant mesh generation with directionality control via packing rectangular solid cells*, International journal for numerical methods in engineering, 57(15) :2099-2129, 2003.
- [5] J.-F. Remacle, R. Gandham, and T. Warburton. *GPU accelerated spectral finite elements on all-hex meshes*, arXiv preprint arXiv :1506.05996, 2015.
- [6] G. E. Karniadakis, M. Israeli, and S. A. Orszag. *High-order splitting methods for the incompressible Navier-Stokes equations*, Journal of computational physics 97.2 : 414-443, 1991.

- [7] D. Göddeke, R. Strzodka, J. Mohd-Yusof, P. McCormick, S. HM Buijssen, M. Grajewski, and S. Turek. *Exploring weak scalability for fem calculations on a gpu-enhanced cluster*, *Parallel Computing*, 33(10) :685-699, 2007.
- [8] D. Göddeke. *Fast and accurate finite-element multigrid solvers for PDE simulations on GPU clusters*, Logos Verlag Berlin GmbH, 2011.
- [9] C. Cecka, A. J. Lew, and E. Darve. *Assembly of finite element methods on graphics processors*, *International journal for numerical methods in engineering*, 85(5) :640-669, 2011.
- [10] G.R. Markall, A. Slemmer, D.A. Ham, P.H.J. Kelly, C.D. Cantwell, and S.J. Sherwin. *Finite element assembly strategies on multi-core and many-core architectures*, *International Journal for Numerical Methods in Fluids*, 71(1) :80697, 2013.