Oil & Gas Science and Technology – Rev. IFP Energies nouvelles (2017) **72**, E1

DOI: 10.2516/ogst/2017003

Dossier

SimRace 2015: Numerical Methods and High Performance Computing for Industrial Fluid Flows
SimRace 2015 : Méthodes numériques et calcul haute performance pour la simulation d'écoulements complexes

Editorial

IFP ENERGIES NOUVELLES CONFERENCE SIMRACE 2015: NUMERICAL METHODS AND HIGH PERFORMANCE COMPUTING FOR INDUSTRIAL FLUID FLOWS

Stéphane de Chaisemartin[1,*] and Grégoire Allaire[2]

[1] *IFP Energies nouvelles, 1-4 avenue de Bois-Préau, 92852 Rueil-Malmaison Cedex - France*
[2] *Member of scientific board of IFP Energies nouvelles, Ecole Polytechnique, Centre de Mathématiques Appliquées, 91128 Palaiseau Cedex - France*
e-mail stephane.de-chaisemartin@ifpen.fr

* Corresponding author

Numerical simulation faces several challenges today. The different topics it relies on (scientific modelling, applied mathematics, High Performance Computing (HPC) and computer science) are continuously evolving, and not necessarily in compatible ways. The novelties emerging in the different fields involved in scientific computing, for instance graphics accelerators or many-core processors, programming languages, Domain Specific Languages (DSL; high level languages dedicated to an application domain), HPC libraries, mesh refinement, error estimates, high order numerical methods, ever finer models, ... are often not so easy to combine.

One part of the challenge of building the 21st century's ultimate numerical platform will be to create connections and joint works between these spheres so that they can feed each other and develop benefiting synergies.

Following this trend, *IFP Energies nouvelles* organized, at the end of 2015, an international conference entitled SimRace, devoted to numerical methods and high performance computing. The aim of SimRace (Simulation Race) was to cover multi-disciplinary topics relating to scientific computing applied to the simulation of industrial fluid flows. Our objective was to bring together people from the spheres of computer science, applied mathematics and applications. Over these three days we tackled issues ranging from the upcoming OpenMp 5.0 norm to various industrial applications in chemical engineering, geosciences and automotive engines, including topics such as a multiscale discontinuous Galerkin method, virtual element or volume formulations, directive-based automatic code generation and many others.

Among the topics raised by the conference, we have identified four main themes developed in the articles selected in this special issue of OGST.

HIGH-PERFORMANCE COMPUTATIONAL AND PROGRAMMING MODELS FOR EMERGING ARCHITECTURES

We are currently facing on-going architectural changes in HPC systems. The number of nodes continues to grow, intra-node concurrency is greatly increasing and we are moving towards heterogeneous systems. Within this context, it is important to ensure performance portability of HPC applications, though it is difficult to predict performance in a dynamic execution environment.

One of the answers provided in the conference is the need for a layered programming approach, from applications to hardware. In these layers, we find today's portable parallel programming API: MPI, OpenMP, PGAS, Charm++. These API are being adapted to follow the hardware evolution.

Another level coming over these parallel API is the Domain Specific Language (DSL) level. The paper by **J.-M. Gratien** [1] tackles this DSL issue. It presents an embedded language (*i.e.* embedded in a host standard language, here C++) dedicated to the implementation of numerical methods written in a variational form. The idea of this DSL is to provide a high level expressive language and to internally handle (thanks to statically generated code) performance and hardware compliance issues.

On top of such DSL comes another original issue presented in the paper of **B. Lelandais** and **M.-P. Oudot** [2], concerning software engineering of simulation codes. Starting from the increasing complexity and size of simulation applications, they present Modane, an application based on Eclipse EMF technology, designed to help developers (often physicists or mathematicians) in designing their simulator's architecture and its evolution. By allowing the user to graphically describe (using, for instance, class diagrams similar to the Unified Modelling Language (UML) standard) the components of the code and their relationships, Modane provides an overview of the overall simulator architecture, facilitates a rapid integration of new collaborators and allows following of the architecture's evolution over time.

SCALABLE LINEAR SOLVERS

Still raising subjects from a bottom-up point of view, we now face the first numerical item, the basis of several simulators: linear and nonlinear solvers.

The link of this numerical component with the underlying hardware is underlined in the contribution of **A. Roussel**, **J.-M. Gratien** and **T. Gautier** [3] where solutions to efficiently implement preconditioners on heterogeneous architectures are presented. Different preconditioners (ILU0 and Multi-Level Domain Decomposition, DDML) are implemented using task-based paradigms, and performance is assessed with hybrid parallelization. Some new numerical techniques derived to increase performance in a parallel context are presented in the contribution of **A. Anciaux-Sedrakian**, **L. Grigori**, **S. Moufawad** and **S. Yousef** [4]. In order to increase the scalability of linear solvers, a communication-avoiding Krylov subspace method algorithm has been developed for the BiCGStab (Bi-Conjugated Gradient Stabilized) solver, aiming at reducing significantly the need to communicate in parallel. The work is focused on increasing the numerical stability for systems arising in geoscience porous medial flow applications.

The importance of having efficient implementations is illustrated in two contributions dealing with solvers in geoscience applications. In these applications, up to 80% of the simulation time can be spent in the solver. The contributions present different techniques used to reduce this problem. The paper written by **J.-M. Gratien**, **O. Ricois** and **S. Yousef** [5] presents the use of mesh refinement and coarsening techniques along with original *a posteriori* error estimates to improve the performance of a compositional reservoir simulation dedicated to enhanced oil recovery. The mesh refinement allows a significant reduction in the load of data to solve. The error estimators prevent oversolving phenomena without introducing significant extra cost. In the paper written by **V. Seignole**, **J.-F. Thebault** and **R. Nabil** [6], the parallelization of Induced Dimension Reduction methods (IDR(s) solver), used in a simulator dedicated to gas-storage applications, is studied. A benchmark on several parallel strategies, multi-threaded, distributed-memory and GPU programming, is detailed.

NUMERICAL SCHEMES ON GENERAL MESHES FOR COMPLEX FLOWS

Getting further into the numerical field, we now deal with numerical scheme derivations, and again we propose to link it with HPC issues. In the contribution of **F. De Vuyst**, **T. Gasc**, **R. Motte**,

M. Peybernes and **R. Poncet** [7] the Lagrange-Flux schemes are reformulated in a "HPC-compatible" way. This paper evaluates the algorithmic aspects of the scheme with a performance model. Given information on the targeted architecture (bandwidth and peak) the model tries to predict the efficiency of the algorithms, for data and CPU usage. Following the results of this analysis, the algorithm has been rewritten to minimize memory transfers and maximize vectorization compatibility. The new algorithm shows a good scalability using AVX (Advanced Vector eXtensions for x86 processors).

MULTI-SCALE METHODS AND MODEL COUPLING

Finally, modelling issues are tackled in the contribution of **M. Essadki**, **S. de Chaisemartin**, **M. Massot**, **F. Laurent**, **A. Larat** and **S. Jay** [8]. This contribution details recent advances on high order moment methods for spray to simulate fuel injection. If modelling issues are the central focus of the paper, HPC concerns must be addressed. Indeed high order moment methods have been introduced to decrease the number of variables in Eulerian spray simulation. Moreover mesh refinement is used to decrease the CPU and memory cost, maintaining precision. To efficiently handle mesh refinement, the simulator used has been built upon the AMR library p4est, dedicated to massively parallel computations (up to several thousands of cores).

CONCLUSION

This collection of papers is of course not exhaustive but we hope it illustrates what we believe is a key issue for scientific computing today: the need to deeply link computer science, high performance computing, applied mathematics and scientific modelling.

Indeed the articles gathered in this special issue show that while developing linear solvers, numerical schemes or physical models, the question of targeted architectures, data access and CPU intensity must be addressed in one way or another. Furthermore they highlight the need to build multi-layered simulator codes, automatically handling the fast-moving underlying architectures and offering to the end-user developer a high-level expressive language, through DSL or parallel API and possibly software architecture design tools.

REFERENCES

1 Gratien J.-M. (2017) ArcFVDSL, a DSEL combined to HARTS, a runtime system layer to implement efficient numerical methods to solve diffusive problems on new heterogeneous hardware architecture, *Oil Gas Sci. Technol.*

2 Lelandis B., Oudot M.-P. (2016) Modane: a design support tool for numerical simulation codes, *Oil Gas Sci. Technol.* **71**, 57.

3 Roussel A., Gratien J.-M., Gautier T. (2016) Using runtime systems tools to implement efficient preconditioners for heterogeneous architectures, *Oil Gas Sci. Technol.* **71**, 65.

4 Anciaux-Sedrakian A., Grigori L., Moufawad S., Yousef S. (2016) *S*-step BiCGStab algorithms for geoscience dynamic simulations, *Oil Gas Sci. Technol.* **71**, 66.

5 Gratien J.-M., Ricois O., Yousef S. (2016) Reservoir simulator runtime enhancement based on *a posteriori* error estimation techniques, *Oil Gas Sci. Technol.* **71**, 59.

6 Seignole V., Thebault J.-F., Nabil R. (2016) Analysis of IDR(*s*) family of solvers for reservoir simulations on different parallel architectures, *Oil Gas Sci. Technol.* **71**, 63.

7 De Vuyst F., Gasc T., Motte R., Peybernes M., Poncet R. (2016) Lagrange-flux schemes: reformulating second-order accurate Lagrange-remap schemes for better node-based HPC performance, *Oil Gas Sci. Technol.* **71**, 64.

8 Essadki M., de Chaisemartin S., Massot M., Laurent F., Larat A., Jay S. (2016) Adaptive mesh refinement and high order geometrical moment method for the simulation of polydisperse evaporating sprays, *Oil Gas Sci. Technol.* **71**, 61.

Éditorial

LES RENCONTRES SCIENTIFIQUES D'IFP ENERGIES NOUVELLES : MÉTHODES NUMÉRIQUES ET CALCUL HAUTE PERFORMANCE POUR LA SIMULATION D'ÉCOULEMENTS COMPLEXES [SIMRACE 2015]

Stéphane de Chaisemartin[1,*] and Grégoire Allaire[2]

[1] IFP Energies nouvelles, 1-4 avenue de Bois-Préau, 92852 Rueil-Malmaison Cedex - France
[2] Membre du conseil scientifique IFP Energies nouvelles, Ecole Polytechnique, Centre de Mathématiques Appliquées, 91128 Palaiseau Cedex - France
e-mail stephane.de-chaisemartin@ifpen.fr

*Coordinateur du dossier

La simulation numérique doit faire face aujourd'hui à de nombreux défis. En effet les disciplines sur lesquelles elle est bâtie : la modélisation physique, les mathématiques appliquées, le calcul haute performance et l'informatique, sont en constante évolution, et dans des directions pas toujours compatibles. Émergence d'accélérateurs graphiques ou de processeurs multi-cœurs, nouveaux langages de programmation, langages spécifiques au domaine (DSL) et bibliothèques parallèles, utilisation du raffinement de maillage, des estimateurs d'erreur et des méthodes d'ordre élevé, modèles physiques de plus en plus fins, etc., autant de nouveautés qui sont souvent difficiles à combiner. La plateforme de simulation du 21^{e} siècle devra créer des connections fortes entre ces disciplines de sorte que leurs développements futurs se fassent dans une meilleure synergie.

Dans l'idée de contribuer à ce rapprochement, *IFP Energies nouvelles* a organisé la première conférence du cycle Rencontres Scientifiques entièrement consacrée aux méthodes numériques et au calcul haute performance (HPC) pour la simulation d'écoulements dans des configurations industrielles. Baptisée SimRace (pour *Simulation Race*), cette conférence avait pour ambition de couvrir des sujets pluridisciplinaires relatifs au calcul scientifique pour la simulation d'écoulements complexes. L'objectif était de rassembler des représentants du monde de l'informatique, des mathématiques appliquées et des domaines applicatifs. Durant ces trois journées, nous avons traité de sujets allant de la future norme OpenMP 5.0 à différentes applications industrielles dans les domaines du génie chimique, des géosciences et des moteurs automobiles. Nous avons également passé en revue des avancées récentes autour des méthodes de Galerkine discontinues pour le multi-échelle, des formulations éléments ou volumes virtuels, de la génération automatique de code à base de directives et autour de bien d'autres thèmes encore.

Aujourd'hui, de nombreux domaines d'applications mettant en jeu des écoulements complexes ont un besoin croissant en simulations toujours plus rapides et précises. Cette tendance s'observe par exemple dans la simulation des bassins ou des réservoirs pétroliers, des moteurs automobiles ou aérospatiaux, des processus de raffinage, etc.

SimRace proposait de créer une synergie entre les chercheurs et ingénieurs des mondes du HPC et de l'analyse numérique, et de constituer un creuset où se mêlent les idées et les points de vue de chercheurs, développeurs et utilisateurs.

Parmi les sujets couverts par la conférence, quatre thèmes se dégagent des articles retenus pour ce numéro spécial d'OGST.

MODÈLES DE PROGRAMMATION HPC POUR LES ARCHITECTURES ÉMERGEANTES

Nous sommes actuellement confrontés à des changements d'architecture des systèmes HPC. Le nombre de nœuds ne cesse de croître, le parallélisme intra-nœuds augmente considérablement et nous évoluons vers des systèmes hétérogènes. Dans ce contexte, il importe d'assurer la portabilité des performances des applications HPC, bien qu'il soit difficile de prédire des performances dans un environnement d'exécution dynamique.

Il apparaît alors nécessaire de tendre vers une approche de programmation stratifiée, séparant les couches applicatives de celles en charge du matériel. Parmi ces couches, nous retrouvons les API (interface de programmation applicative) actuelles de programmation parallèle : MPI, OpenMP, PGAS, Charm++. Ces API sont en cours d'adaptation pour suivre les évolutions du matériel.

Les langages spécifiques au domaine (*Domain Specific Language* ou DSL) vont également dans le sens de cette approche stratifiée, au-dessus de ces API. L'article de **J.-M. Gratien** [1] aborde la question de ces DSL. Il présente un DSL écrit en langage C++, dédié à l'implémentation de méthodes numériques écrites sous forme variationnelle. L'idée de ce DSL est de proposer un langage expressif de haut niveau et de gérer de manière transparente les problématiques bas-niveau de performance et d'adaptation au support d'exécution, grâce à du code généré statiquement.

Bâtir une application de simulation mettant en œuvre ces différentes strates informatiques n'est pas chose aisée. Un outil original d'aide à l'architecture des codes de simulation est présenté dans l'article de **B. Lelandais** et **M.-P. Oudot** [2]. Fort du constat de la complexité croissante des codes de calcul, les auteurs présentent Modane, une application basée sur la technologie Eclipse EMF, conçue pour aider les développeurs (souvent des physiciens ou des mathématiciens) à penser et à concevoir l'architecture de leur simulateur et à maîtriser son évolution. En permettant aux utilisateurs de décrire graphiquement (en utilisant par exemple des diagrammes de classe du standard UML - *Unified Modelling Language*) les composants de leur code et leurs relations, Modane améliore la vue d'ensemble de l'architecture du calculateur, facilite par exemple l'intégration de nouveaux développeurs, et permet de suivre l'évolution de l'architecture dans le temps.

SOLVEURS LINÉAIRES ET LEUR PASSAGE À L'ÉCHELLE

Une fois posées les premières strates en charge du matériel et de la performance, il convient de s'attaquer aux couches numériques. Nous commençons naturellement par les solveurs linéaires et non linéaires, socle numérique de très nombreux codes de calcul.

Le lien fort entre ce composant numérique et l'architecture de calcul sous-jacente est souligné dans la contribution de **A. Roussel**, **J.-M. Gratien** et **T. Gautier** [3] où des solutions pour implémenter efficacement des préconditionneurs sur des architectures hétérogènes sont présentées. Ces implémentations, parallélisées par tâches, sont testées dans un contexte de parallélisme hybride pour différents préconditionneurs (ILU0 et Décomposition de Domaines Multi-Niveaux, DDMN). Par ailleurs de nouvelles techniques numériques, conçues pour augmenter la performance en parallèle, sont présentées dans la contribution de **A. Anciaux-Sedrakian**, **L. Grigori**, **S. Moufawad** et **S. Yousef** [4]. Afin d'augmenter la scalabilité (passage à l'échelle) du solveur linéaire, un algorithme S-STEP est développé pour le solveur BiCGStab (*Bi-Conjugated Gradient Stabilized*), dans le but de réduire significativement le besoin de communiquer en parallèle. Cet algorithme est testé afin de stabiliser des systèmes provenant de simulations d'écoulement en milieu poreux, pour des applications de géosciences.

Finalement, deux illustrations applicatives nous montrent comment le coût d'un simulateur peut être lié à celui des solveurs. En effet dans ces applications, le solveur peut monopoliser jusqu'à 80 % du temps de simulation. Les contributions proposées présentent différentes techniques pour

réduire cette empreinte. L'article proposé par **J.-M. Gratien**, **O. Ricois** et **S. Yousef** [5] illustre comment, dans une simulation de réservoir pour une application de récupération assistée des hydrocarbures (EOR), le recours à l'adaptation de maillage, couplée avec une estimation *a posteriori* de l'erreur, améliore la performance du calcul. L'utilisation du raffinement de maillage permet de réduire significativement le nombre de degrés de liberté du problème, ainsi que la consommation mémoire du calcul. L'emploi des estimateurs d'erreur doit permettre d'éviter les phénomènes de sur-résolution (au-delà de la précision nécessaire) sans ajouter de surcoût significatif. Une autre direction est investiguée dans l'article proposé par **V. Seignole**, **J.-F. Thebault** et **R. Nabil** [6], où la parallélisation d'une méthode IDR (*Induced Dimension Reduction*, réduction des dimensions induites) utilisée dans un simulateur dédié aux applications de stockage de gaz, est étudiée. Différentes stratégies de parallélisation, à mémoire partagée, distribuée ou utilisant des accélérateurs graphiques, sont évaluées et comparées.

SCHÉMAS NUMÉRIQUES POUR ÉCOULEMENTS COMPLEXES SUR MAILLAGES GÉNÉRAUX

Nous poursuivons l'exploration des strates numériques en nous intéressant maintenant à la dérivation de schémas, sans pour autant perdre le lien avec les problématiques de calcul intensif. Dans l'article de **F. De Vuyst**, **T. Gasc**, **R. Motte**, **M. Peybernes** et **R. Poncet** [7] les schémas Lagrange-flux sont reformulés pour une meilleure efficacité computationnelle. Cette contribution évalue les algorithmes de ces schémas grâce à un modèle de performance en charge de prédire la performance en temps de calcul de l'algorithme et sa consommation mémoire à partir d'informations sur l'architecture cible (bande passante et puissance crête). Les résultats de cette analyse permettent d'orienter la réécriture de l'algorithme pour réduire les transferts de mémoire et maximiser sa compatibilité avec des instructions vectorisées. L'efficacité du nouvel algorithme dans un contexte SIMD, implémenté avec un jeu d'instructions AVX (*Advanced Vector eXtensions*), est ensuite analysée.

MÉTHODES MULTI-ÉCHELLES ET COUPLAGE DE MODÈLES

L'écriture des nouveaux modèles ne pourra pas non plus éluder la question du coût calcul. L'article de **M. Essadki**, **S. de Chaisemartin**, **M. Massot**, **F. Laurent**, **A. Larat** et **S. Jay** [8] détaille des avancées récentes sur les méthodes de moments d'ordre élevé pour le spray (aérosol) dans des applications de calcul d'injection de carburant. Bien que les questions de modélisation soient au cœur de l'article, les préoccupations de calcul haute performance ne sont pas bien loin. En effet les méthodes de moments d'ordre élevé ont été introduites pour diminuer le nombre de degré de liberté dans la simulation eulérienne des sprays. De plus le raffinement de maillage adaptatif (AMR) est utilisé pour diminuer le coût de calcul et la consommation mémoire, à précision égale. Afin de gérer efficacement le raffinement de maillage, le calculateur développé utilise p4est, une bibliothèque dédiée à l'AMR et efficace en parallèle jusqu'à plusieurs dizaines de milliers de cœurs.

CONCLUSION

Bien sûr cet ensemble d'articles n'est pas exhaustif, mais nous espérons qu'il illustre ce que nous pensons être un point clé pour le calcul scientifique aujourd'hui : la nécessité de coupler en profondeur les disciplines de l'informatique, du calcul haute performance, des mathématiques appliquées et de la modélisation scientifique.

En effet les articles rassemblés pour ce numéro spécial montrent que pour construire des solveurs linéaires, des schémas numériques ou des modèles physiques efficaces pour le calcul, les questions de l'architecture cible, de l'accès aux données et de la consommation CPU doivent

être impérativement abordées. De plus ces articles illustrent la nécessité de construire aujourd'hui des plateformes de calcul à plusieurs niveaux, capables de prendre en compte l'évolution rapide des architectures cibles, et proposant à l'utilisateur développeur un langage haut niveau, via des DSL ou des API parallèles, ainsi que des outils d'aide à la conception et à la maintenance des codes.

RÉFÉRENCES

1 Gratien J.-M. (2017) ArcFVDSL, a DSEL combined to HARTS, a runtime system layer to implement efficient numerical methods to solve diffusive problems on new heterogeneous hardware architecture, *Oil Gas Sci. Technol.*

2 Lelandis B., Oudot M.-P. (2016) Modane: a design support tool for numerical simulation codes, *Oil Gas Sci. Technol.* **71**, 57.

3 Roussel A., Gratien J.-M., Gautier T. (2016) Using runtime systems tools to implement efficient preconditioners for heterogeneous architectures, *Oil Gas Sci. Technol.* **71**, 65.

4 Anciaux-Sedrakian A., Grigori L., Moufawad S., Yousef S. (2016) *S*-step BiCGStab algorithms for geoscience dynamic simulations, *Oil Gas Sci. Technol.* **71**, 66.

5 Gratien J.-M., Ricois O., Yousef S. (2016) Reservoir simulator runtime enhancement based on *a posteriori* error estimation techniques, *Oil Gas Sci. Technol.* **71**, 59.

6 Seignole V., Thebault J.-F., Nabil R. (2016) Analysis of IDR(*s*) family of solvers for reservoir simulations on different parallel architectures, *Oil Gas Sci. Technol.* **71**, 63.

7 De Vuyst F., Gasc T., Motte R., Peybernes M., Poncet R. (2016) Lagrange-flux schemes: reformulating second-order accurate Lagrange-remap schemes for better node-based HPC performance, *Oil Gas Sci. Technol.* **71**, 64.

8 Essadki M., de Chaisemartin S., Massot M., Laurent F., Larat A., Jay S. (2016) Adaptive mesh refinement and high order geometrical moment method for the simulation of polydisperse evaporating sprays, *Oil Gas Sci. Technol.* **71**, 61.