

## Stage ingénieur de 3-6 mois :

### KOSPEL ( **Kokkos PseudoSpectral Library** ) Codage C++ en HPC : passage du CPU au GPU avec Kokkos

[Yannick Ponty](#) (Dir. Rech. CNRS)

Laboratoire J-L Lagrange, Observatoire de la Côte d'Azur (OCA) – Nice

#### Contexte Scientifique :

Depuis les années 2000, Yannick Ponty a construit un code de simulation en langage C++ (Cubby), pour résoudre les équations de la MHD dans un espace périodique, avec des méthodes pseudo-spectrales. C'était l'un des premiers codes parallèles distribués utilisant la librairie MPI et MPI-IO. De 2006 à 2015, Alain Miniussi ingénieur de recherche à l'OCA a progressivement adapté le code avec des méthodes de "Design Pattern" en C++ et introduit des techniques de génie logiciel pour rendre le code plus fiable et reproductible (suiveur de version et implémentation de tests automatiques unitaires). Il est maintenant parallélisé en [MPI](#) et Pthread. Ce code a permis de produire plus d'une quinzaine de publications et il a tourné sur la plupart des clusters locaux et nationaux.

Mais cette librairie (**High Performant Computing (HPC)**) ne permet pas de tourner sur des clusters GPU (cartes graphiques) qui deviennent l'architecture prédominante pour la recherche ou l'industrie. La problématique dans de nombreuses équipes est de faire migrer leurs codes HPC vers ces nouvelles architectures.

Bibliothèque Kokkos : ( <https://github.com/kokkos> )

Il y a une librairie de calcul C++ qui permet de faire cette transition, c'est Kokkos. Elle permet de choisir à la compilation l'architecture voulue (CPU, GPU, OpenMP, ..., ect ) Elle a été construite pour fonctionner sur les plus gros clusters mondiaux. Le projet est sous fondation linux et financé par le département de l'énergie américain (DOE) et le [CEA](#) français contribue aussi à fortement à sa construction et sa diffusion par la Maison de la simulation de Saclay (<https://mdls.fr/>). Il y a notamment un groupe de travail du CEA qui est dédié à l'[exascale](#) et qui travaille essentiellement avec Kokkos (<https://cexa-project.org/>).

Kokkos est une bibliothèque de programmation parallèle, développée principalement pour les applications HPC (High Performance Computing). Elle fournit un environnement de programmation en C++ pour exprimer des calculs parallèles et distribués de manière efficace sur une variété d'architectures matérielles, notamment les CPU multicœurs et les accélérateurs comme les GPU. Voici quelques-uns des avantages de la bibliothèque :

Abstraction matérielle : Kokkos fournit une abstraction matérielle qui permet aux développeurs d'exprimer des algorithmes parallèles de manière indépendante de l'architecture matérielle sous-jacente.

Performance : Kokkos est conçu pour offrir des performances élevées en exploitant efficacement les ressources matérielles disponibles, telles que les CPU multicœurs et les GPU.

Productivité des développeurs : En fournissant une interface conviviale en C++, Kokkos permet aux développeurs de créer et de maintenir des applications parallèles de manière efficace.

Avoir un code qui tourne à la fois en CPU, GPU, OPENMP ou autre architecture d'avenir devient indispensable. Les concepteurs de code n'ont plus à prévoir l'optimisation des boucles et des calculs de base. En fonction de la plateforme et de la bibliothèque parallèle souhaitée, l'utilisateur décide à la compilation et Kokkos se charge de tout. Il aura un code unique pour toutes ses différentes plateformes ou architectures matérielles.

#### Objectifs :

L'objectif est de construire une bibliothèque en C++ qui utilise kokkos pour des méthodes pseudo-spectrales à la fois en CPU et/ou GPU. Cette bibliothèque sera une première approche pour explorer les possibilités de calculer sur des clusters GPU par des méthodes pseudo-spectrales. Ce travail préliminaire est essentiel pour ensuite faire migrer le code Cubby vers les solutions trouvées pendant ce stage.

Dans un premier temps, il faudra que cela fonctionne sur une architecture ayant un seul nœud de GPU et dans un deuxième temps pouvoir être suffisamment efficace sur un cluster ayant plusieurs nœuds GPU connectés (intra nœud, adastra à 8 GPU) avec MPI, et tout en respectant une balance idéale entre calcul et les communications intrinsèques de la méthode (les transpositions du cube de calcul) et donc une bonne scalabilité.

## Description du besoin en support stagiaire ingénieur :(3- 6 mois )

### Travail attendu :

L'ingénieur stagiaire devra s'imprégner des méthodes pseudo-spectrales et faire une recherche de l'existant pour trouver la meilleure stratégie pour le développement de la solution. Il existe d'ailleurs une jeune bibliothèque qui gère déjà certaines FTT avec kokkos qui pourra être directement utilisée pour ce projet ( <https://kokkosfft.readthedocs.io/en/latest/index.html> ).

Pour ensuite construire un code en C++ 11-17 simple, mais versatile qui permettra d'explorer les différentes optimisations sur les différentes architectures GPU. Le responsable de stage sera impliqué dans toutes les étapes de la construction de cette bibliothèque, avec parfois l'appui d'ingénieur en génie logiciel de l'institut. Ce sera un travail d'équipe.

### Compétences de l'ingénieur souhaitées :

Le stagiaire devra être compétent en programmation en C++ (version récente C++11, C++17, C++20), en programmation objet et en génie logiciel. Si le calendrier le permettra, il pourra suivre une formation pour la « library Kokkos » et interagir avec le groupe du CEA/maison de la simulation Saclay qui devient de plus en plus expert en ce domaine (<https://cexa-project.org/>). Savoir travailler en équipe et utiliser des suiveurs de version comme Gitlab ou Github.

### Atout pour la carrière :

L'acquisition de ces compétences sera profitable pour intégrer des projets HPC/data dans l'industrie, pour la transformation ou la création de code vers les plateformes GPU qui sont l'avenir du calcul scientifique ou du traitement de données comme l'AI.

### Environnement d'accueil :

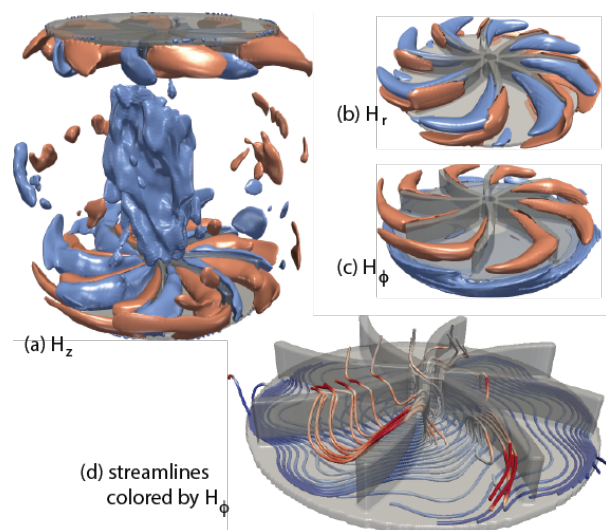
L'équipe de turbulence, fluide et plasmas du laboratoire J-L Lagrange est spécialisée dans les simulations numériques pour comprendre les processus turbulents et non linéaires dans divers fluides, notamment les hydrodynamiques, les magnétohydrodynamiques (MHD) et les plasmas. Ces recherches théoriques et numériques sont effectuées à l'aide de simulations haute résolution sur des clusters massivement parallèles de 1000 à 64000 processeurs, situés dans des centres nationaux, internationaux et des mésocentres azuréens. L'équipe a développé plusieurs codes pour fonctionner efficacement en parallèle sur ces grands clusters.

L'ingénieur rejoindra une équipe experte en calcul HPC et en simulations numériques. Il aura accès aux clusters de calcul locaux azuréens (<http://crimson.oca.eu>, <https://calculs.univ-cotedazur.fr/>) (Licallo et Azzurra), ainsi qu'aux clusters nationaux de GENCI (Adastra et Jean-Zay)(<http://www.gencl.fr/>).

Il aura un bureau qui est sur le [site](#) de l'observatoire de la Côte d'Azur au Mont-Gros (Nice), un desktop linux si c'est nécessaire. Il pourra profiter du cadre idyllique du parc et de sa fameuse cantine vue sur la ville de Nice et la baie des Anges.

### Contacts :

- Yannick Ponty : [yannick.ponty@oca.eu](mailto:yannick.ponty@oca.eu),
- +33 (0)4 92 00 30 57
- <https://www.oca.eu/yannick-ponty>
- **Laboratoire J-L Lagrange (UMR 7293)**  
Observatoire de la Côte d'Azur - Nice  
Boulevard de L'Observatoire  
CS 34229 06304 Nice Cedex 4 France



*Dynamo Enhancement and Mode Selection Triggered by High Magnetic Permeability*. S. Kreuzahler, Y. Ponty, N. Plihon, H. Homann, and R. Grauer  
Phys. Rev. Lett. 119, 234501 – Published 6 December 2017.  
[doi.org/10.1103/PhysRevLett.119.234501](https://doi.org/10.1103/PhysRevLett.119.234501)