

Simulation plasma hybride Particle-In-Cell à raffinement de maillage adaptatif

Candidature :

- Contact : Nicolas Aunai, LPP/CNRS : nicolas.aunai@lpp.polytechnique.fr
- Merci d'envoyer CV et lettre de motivation
- Profil : niveau Master, bonnes connaissances HPC, maîtrise du C++ et python. connaissances du développement pour GPU, utilisation de git. Pas de connaissance en physique des plasma nécessaire.

Lieu : Laboratoire de Physique des Plasmas, Ecole Polytechnique, Palaiseau, France.

Note : Le stage pourra éventuellement déboucher sur un CDD de 12 mois dont le financement est déjà acquis.

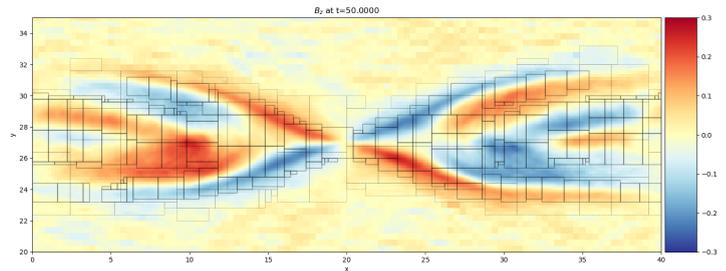
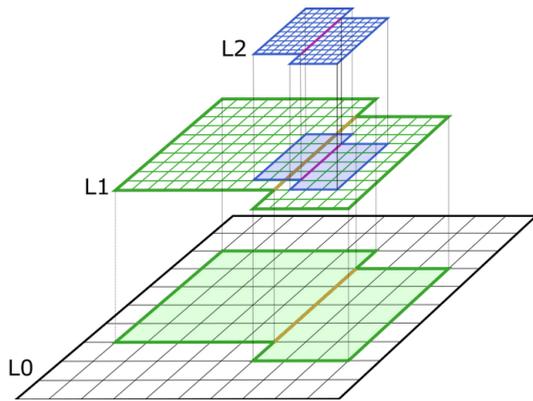
Mots clés:

- Particle-In-Cell, parallélisation GPU, RAJA, SYCL, décomposition de domaine, C++, open-source

Contexte du projet :

La Laboratoire de Physique des Plasmas (LPP), situé sur le campus de l'école polytechnique à Palaiseau, est fortement impliqué dans la simulation numérique des processus et environnements plasma. Ces codes sont utilisés de manière complémentaire aux expériences, de laboratoire (ex: mise au point de propulseurs plasmas), de grands instruments (ex: ITER) ou de mesures spatiales à bord de satellites (ex: mission Bepi-Colombo en vol vers Mercure). Dans la plupart de ces contextes, les processus plasmas étudiés sont faiblement collisionnels et leur dynamique s'établit sur une large gamme d'échelles spatiales et temporelles.

Depuis quelques années, le LPP, sous l'impulsion de la fédération de recherche *Plas@Par*, a entrepris le développement d'un nouveau code Particle-In-Cell hybride traitant les ions comme des particules et les électrons comme un fluide. Ce code, unique au monde, vise à résoudre les équations régissant l'évolution du plasma sur une hiérarchie de grille où à la fois la résolution spatiale et temporelle est incrémentalement raffinée, mais également le formalisme physique, afin de concentrer la haute résolution et la modélisation physique la plus fine aux régions critiques du système. Le code est écrit en C++17 et python, il est développé en open source sur github (<https://github.com/pharehub/phare>).



Gauche : Représentation schématique de la hiérarchie de grilles constituée de 3 niveaux de résolutions différentes, chaque niveau étant lui-même constitué d'une multitude des "patches" rectangulaires à maille uniforme cartésienne. A droite, une illustration 2D d'une simulation à 3 niveaux. Les rectangles en pointillé représentent les patches du niveau 1, ceux en trait plein les patches du niveau 2. La couleur représente l'une des composante du champ magnétique à un instant donné en fonction de la position (x,y) dans le domaine.

Objectifs du stage :

Ce stage à pour but de contribuer au développement de PHARE sur les aspects "performance" du projet.

- Le premier objectif consiste à évaluer la performance actuelle du code. Dans un premier temps, un profiling détaillé du code sera effectué et permettra de déterminer les points chauds et prioritaires à optimiser. Dans un second temps l'étude portera sur la scalabilité de la décomposition de domaine MPI et l'efficacité et limites de l'algorithme d'équilibrage de charge actuellement mis en place.
- Le second objectif consistera à établir une stratégie de parallélisation à l'intérieur de chaque processus MPI et notamment via l'utilisation de l'accélération GPU. Plusieurs pistes seront étudiées notamment l'utilisation de la bibliothèque RAJA et de SYCL.

Références :

- Référence du code PHARE : <https://arxiv.org/abs/2210.14580>
- Dépot du code : <https://github.com/pharehub/phare>