

# HPC, introduction à OpenMP

19 janvier 2026

## 1 Schéma saute-mouton pour les ondes

1. Télécharger le fichier leapfrog.py. Exécuter ce code, ajouter les fonctions utiles pour mesurer le temps de calcul pour diverses finesse de maillage.
2. Convertir le code Python en code C++. Générer le code source kfrog.cpp.
3. Comparer les temps d'exécution du code pour diverses tailles de maillage (“time ./kfrog”). Vérifier les taux d'occupation du CPU avec la commande “htop”.
4. Ajouter une directive OpenMP dans kfrog.cpp pour paralléliser la boucle interne. Mesurer les temps de calcul. Déplacer la directive OpenMP avant la boucle sur i. Conclusions ?

## 2 Tiling

1. Modifier les boucles sur (i,j) en utilisant un algorithme de tiling. On paramétrera la taille de la tuile au moyen de la directive :  
`#define TILE_SIZE`
2. Essayer diverses tailles de tuilage et diverses directives OpenMP. Avec quelle combinaisons obtenez vous les calculs les plus rapides ? Pensez à utiliser des tailles de maillages suffisamment grandes pour que les données de la grille ne rentrent pas dans le cache du CPU...

## 3 Calcul 3D

1. Modifier le code C++ pour résoudre l'équation des ondes 3D avec un maillage en (i,j,k). Pour les tracés, on conservera un tracé 2D (en fixant par exemple une valeur de i, j ou k).
2. Optimiser ce code avec OpenMP.