MACS3 : projet calcul parallèle

1) Introduction

L'objectif de ce projet de programmation est d'évaluer les connaissances acquises au cours des 6 séances de 3h45 réalisées en novembre et décembre 2020. Les outils nécessaires (compilateur, librairie MPi, OpenMP,…) sont les mêmes que ceux que vous avez utilisé pendant ces séances.

2) Problème physique continu

Pour évaluer vos acquis concernant les techniques de programmation HPC, je vous demande de programmer la résolution d'un problème de convection/diffusion de la chaleur en 2 dimensions.

En **2D**, le problème continu s'écrit :

$$\frac{\partial T}{\partial t} + U\frac{\partial T}{\partial x} + V\frac{\partial T}{\partial y} = k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) \qquad \text{pour (x,y) in [0, 10]*[0, 10]}$$

La vitesse d'advection suivant les axes x et y est définie par le couple (U,V). Ce dernier est indépendant du temps et de l'espace. On choisit U=V =1m/s.

Le coefficient de diffusion, k, est lui aussi indépendant du temps et de l'espace : k=0.0005 m^2s^{-1})

Des conditions aux limites de périodicités sont appliquées dans les 2 directions :

- T(x=0,y,t) = T(x=10,y,t)
- T(x=10,y,t) = T(x=0,y,t)
- T(x, y=0,t) = T(x, y=10,t)
- T(x, y=10,t) = T(x, y=0,t)

La solution initiale du champ de température est définie comme suit :

$T(x,y,t=0) = 300 + 10\exp(-\frac{r}{0.02})$ où $r = \sqrt{(x-5)^2 + (y-5)^2}$

3) Problème discret

La résolution de l'edp est réalisée par une methode de type volumes finis sur un maillage structuré où l'inconnue est localisée au centre du volume de contrôle (methode cell centered).

- **dérivée temporelle: schéma de Heun explicite précis au 2ème ordre**

  - le problème continu peut s'écrire comme f'(**x**,t) = G(f(**x**,t))

  - f(**x**,t=t^n) connu = f^n

    $\tilde{f} = f^n + dt\, G(f^n)$

$$f^{n+1} = f^n + dt/2*(G(f^n) + G(f^{\sim}))$$

- **dérivée seconde sur maillage structuré: schéma centré précis au 2ème ordre**

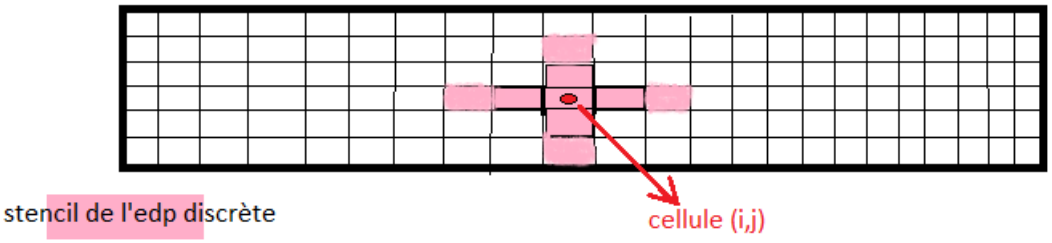
$$\partial^2 f \partial x^2 (i,j) = (f(i+1,j) - 2\, f(i,j) + f(i-1,j)) / \Delta x^2 + O(\Delta x^2)$$

$$\partial^2 f \partial y^2 (i,j) = (f(i,j+1) - 2\, f(i,j) + f(i,j-1)) / \Delta y^2 + O(\Delta y^2)$$

- **dérivée première sur maillage structuré: schéma centré précis au 4ème ordre**

$$\partial f \partial x (i,j) = \big(8 (f(i+1,j) - f(i-1,j)) - (f(i+2,j) - f(i-2,j)) \big) / 12\Delta x + O(\Delta x^4)$$
$$\partial f \partial y (i,j) = \big(8 (f(i,j+1) - f(i,j-1)) - (f(i,j+2) - f(i,j-2)) \big) / 12\Delta y + O(\Delta y^4)$$

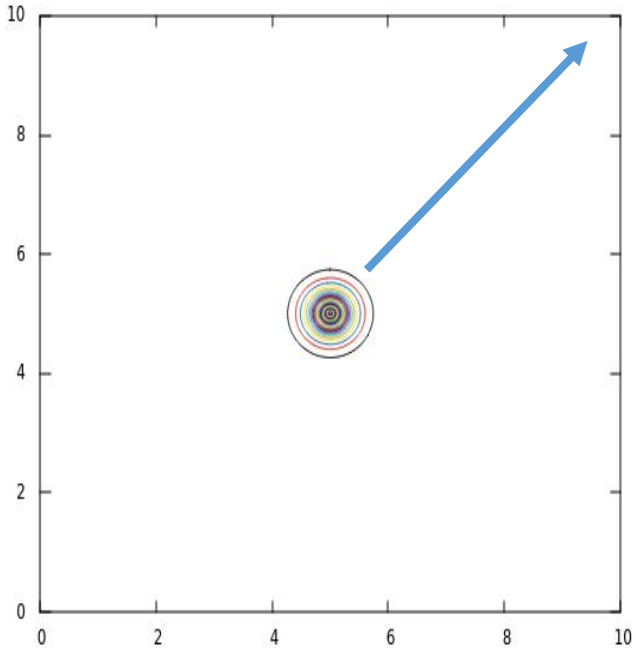
Le stencil de cette discrétisation, illustré sur la figure ci-dessous, comporte 5 points par direction à cause de la dérivée première précise au 4eme ordre. Il nécessite donc 2 cellules fantômes en x=0, x=10, y=0 et y=10.



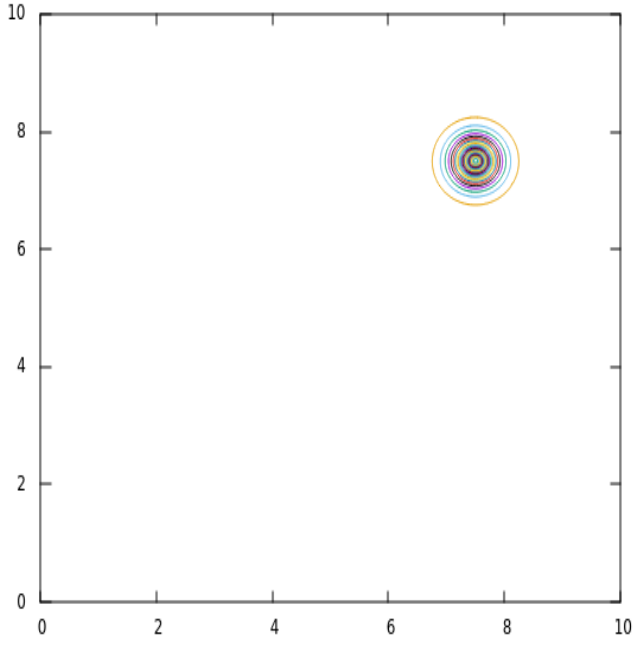
4) Prise en main du programme de calcul séquentiel

Un programme squellette, advectiondiffusion.cpp, existe pour résoudre le problème bidimensionnel en séquentiel :

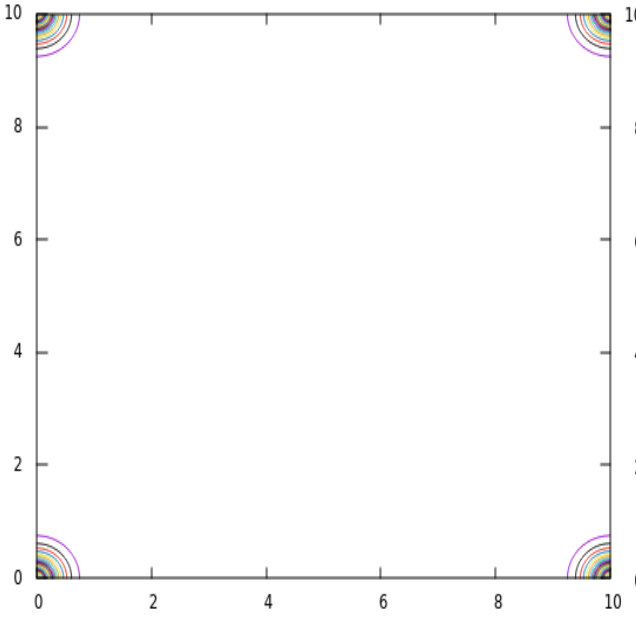
- Compiler ce programme (make –f Makefile) en adaptant le fichier Makefile à votre environnement.
- Executez le pour vous fournir une solution de réference (en terme de solution et de temps CPU). La solution obtenue après 0s et 7.5s de l'advection/diffusion du spot de température, doit être comme sur les figures ci-dessous où la flèche bleue indique la direction de l'advection:



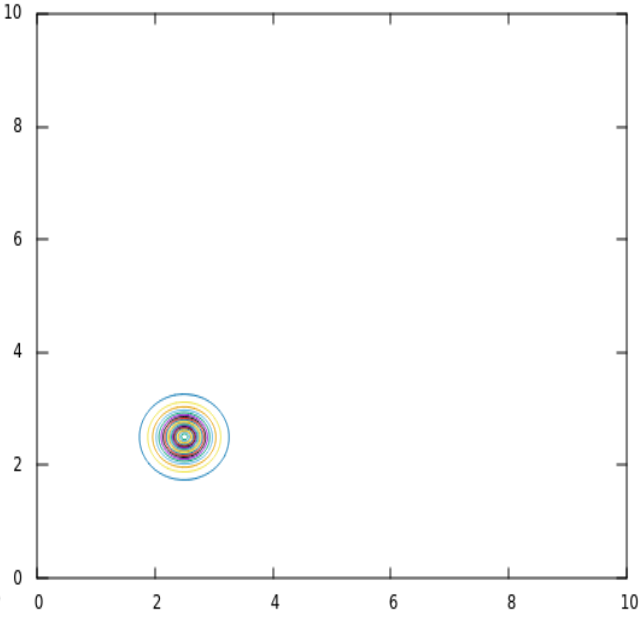
T(x,y, t=0s)



T(x,y, t=2.5s)



T(x,y, t=5s)



T(x,y, t=7.5s)

- Pour analyser les résultats, utilisez gnuplot et le script script.conf en tapant les commandes suivantes :

  ./postresultat1fichier Sortie.txt
  gnuplot script.conf –persist

- Analyser le programme pour comprendre son fonctionnement.
- Soit (Ni,Nj) le nombre de volume de contôle dans les direction(x,y), les tableaux sont stockés en 1D. Pour accéder à la Température T de la cellule (i,j), il faut pointer sur la case mémoire l suivante :
  - T(l) = T(i + j*Ni)

- **Deux** rangées de cellules fantôme sont ajoutées à chaque extrémité du domaine (4 dans la direction X et 4 dans la direction Y) pour imposer les conditions aux limites et faciliter l'implémentation de l'opérateur Laplacien et gradient sur les cellules jouxtant le bord du domaine de calcul

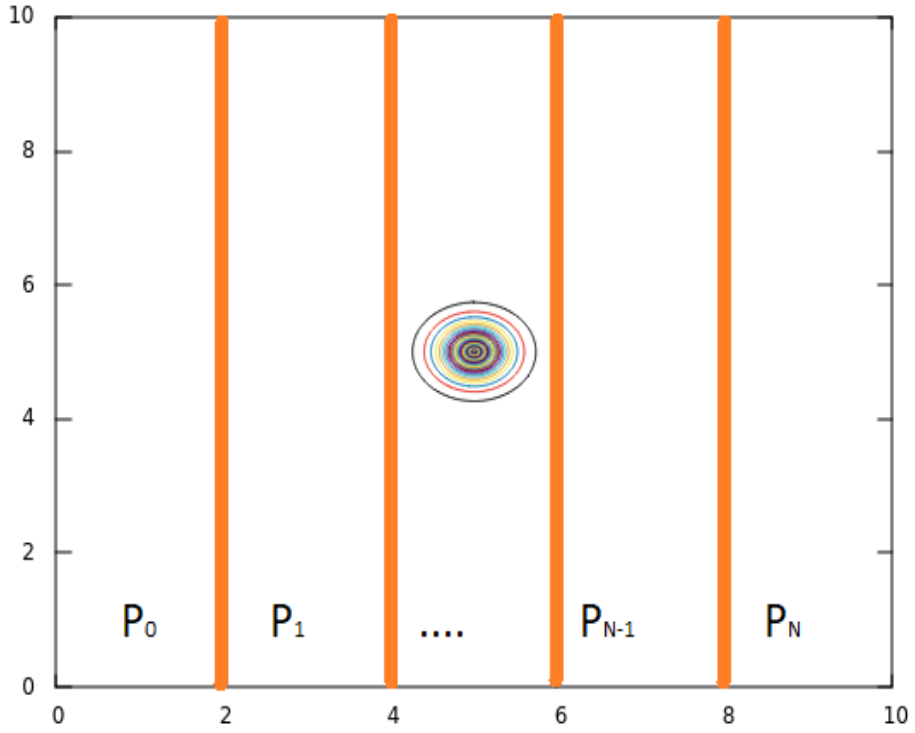
5) Exercices de parallélisation

5.1) OpenMP « coarse grain »

Ecrire une version parallèle du programme advection-diffusion à l'aide de la librairie OpenMP. Adoptez l'approche « Coarse grain » qui se base sur un partage du travail au niveau des boucles for.

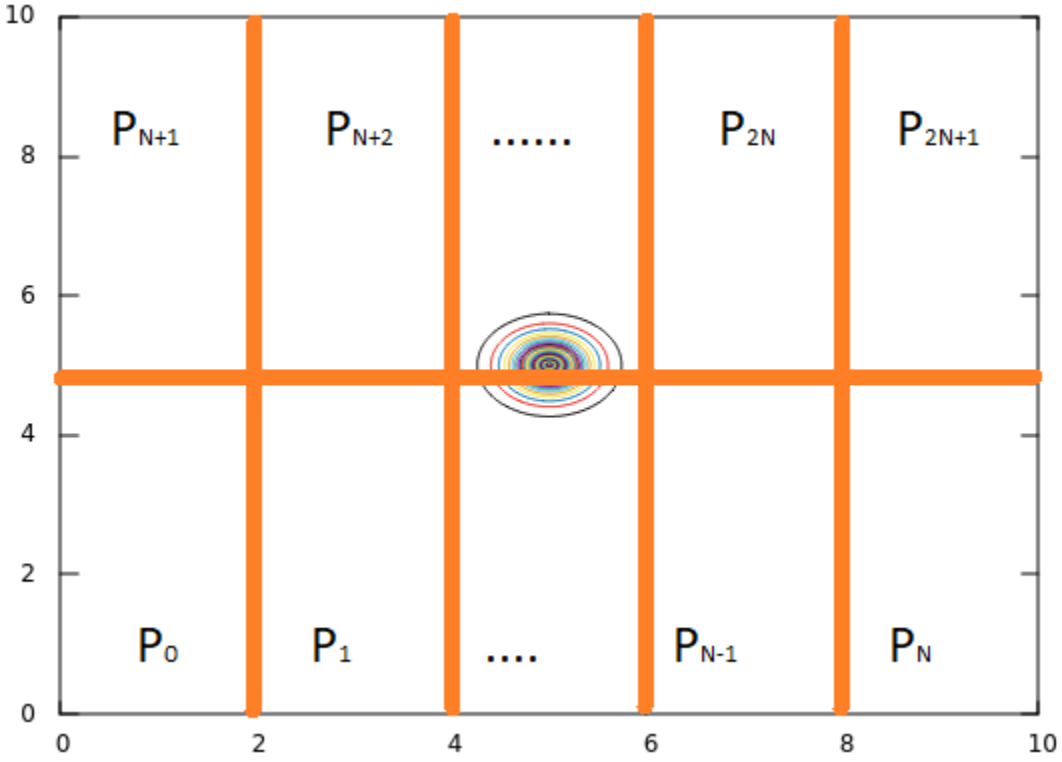
5.2) MPI version 1

Ecrire une version parallèle du programme advection-diffusion à l'aide de la librairie MPI. Le mode de communication est libre, mais il est conseillé d'employer des communications point à point non bloquante (MPI_Irecv) pour les receptions afin d'éviter les deadlocks. Pour le partage du travail, découpez l'espace dans la direction **X uniquement** comme indiquée sur la figure ci-dessous. Le programme doit être capable de calculer sur un nombre quelconque de processus MPI.



5.3) MPI version 2

Ecrire une version parallèle du programme advection-diffusion à l'aide de la librairie MPI. Pour le partage du travail, découpez l'espace dans la direction **X et Y** comme indiquée sur la figure ci-dessous. Le programme doit être capable de calculer sur un nombre quelconque de processus MPI.


10
8
6
4
2
0
0
2
4
6
8
10
P_{N+1}
P_{N+2}
.....
P_{2N}
P_{2N+1}
P_0
P_1
....
P_{N-1}
P_N