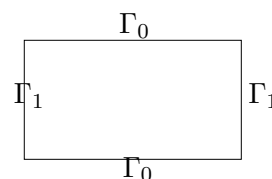


## PROJET de Programmation CHP

### Calcul Haute Performance

On se place dans le domaine  $[0, L_x] \times [0, L_y]$  de  $\mathbb{R}^2$  dans lequel on résoud l'équation de la chaleur :

- (1)  $\partial_t u(x, y, t) - D\Delta u(x, y, t) = f(x, y, t)$
- (2)  $u|_{\Gamma_0} = g(x, y, t)$
- (3)  $u|_{\Gamma_1} = h(x, y, t)$



#### 1. Analyse du problème

On se propose de résoudre numériquement cette équation par la méthode des différences finies.

- (a) Ecrire le schéma d'Euler implicite à l'aide de différences finies centrées du second ordre en espace.

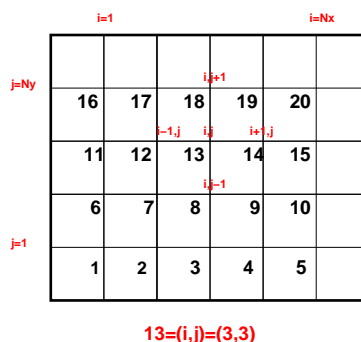


Figure 2 : Principe de la numérotation globale et locale

- (b) En utilisant la numérotation proposée figure 2, montrer que le schéma précédent se met sous la forme matricielle  $AU = F$ .
- (c) Décrire précisément la structure de la matrice  $A$ .
- (d) Enoncer les propriétés de la matrice  $A$ .

#### 2. Implémentation informatique

On utilisera les cas test suivants pour valider le travail :

$$L_x = L_y = 1.0, D = 1.0$$

La solution stationnaire résultant des conditions suivantes

$$(4) \quad f = 2(y - y^2 + x - x^2) \quad g = 0 \quad h = 0$$

Puis

$$(5) \quad f = \sin(x) + \cos(y) \quad g = \sin(x) + \cos(y) \quad h = \sin(x) + \cos(y)$$

La solution instationnaire périodique résultant des conditions suivantes

$$(6) \quad f = e^{-(x - \frac{L_x}{2})^2} e^{-\left(y - \frac{L_y}{2}\right)^2} \cos\left(\frac{\pi}{2}t\right) \quad g = 0 \quad h = 1$$

(a) **Ecrire le code séquentiel**

- i. Cahier des charges pour le nom des variables :  $Nx$  nombre de noeuds dans la direction  $x$ ,  $Ny$  nombre de noeuds dans la direction  $y$ ,  $U$  solution au temps  $n$ ,  $Uo$  solution au temps  $n - 1$ ,  $dx$ ,  $dy$ ,  $dt$  pas d'espace et de temps,  $Lx$ ,  $Ly$ ,  $D$
- ii.  $Nx$ ,  $Ny$ ,  $Lx$ ,  $Ly$ ,  $D$ ,  $dt$  seront lues dans un fichier de paramètres.
- iii. Construire un module contenant les trois fonctions :  $f$ ,  $g$  et  $h$ .
- iv. Construire un module contenant l'assemblage de la matrice  $A$  et le second membre.
- v.  $A$  est une matrice creuse, on ne stockera pas toute la matrice.
- vi. Construire un module contenant le gradient conjugué.

(b) **Ecrire le code parallèle**

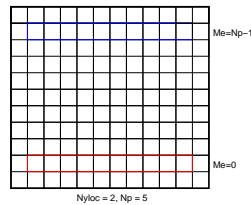


Figure 3 : Exemple de répartition des inconnues sur les processeurs

- i. Expliquer comment vous envisagez la répartition des inconnues entre les différents processeurs, vous pouvez vous inspirer de la figure 3.
- ii. Chaque processeur connaîtra une partie de  $U$  seulement,  $U(i1:iN)$ .
- iii. L'algorithme du gradient conjugué devra être parallélisé :
  - Listez, dans le rapport, les opérations nécessitant des communications.
  - Détaillez, dans le rapport, les communications pour chacun des processeurs.
  - Optimisez les communications.
- iv. Chaque processeur écrira sa partie de solution dans son fichier sol00Me.dat (on pourra utiliser la subroutine Rename proposée en annexe).

## Il faudra fournir les documents suivants :

### 1. Un rapport contenant

- L'analyse mathématique du problème :
  - L'écriture du schéma numérique proposé.
  - La mise en forme matricielle du problème.
  - La description détaillée de la structure de la matrice et de ses propriétés.
- L'explication précise du parallélisme mis en oeuvre dans votre code contenant :
  - La description de votre répartition des inconnues entre les différents processeurs.
  - Les différents points de l'algorithme nécessitant des communications entre les processeurs.
  - Une description détaillée des communications réalisées (taille des messages, émetteur, récepteur).
- Le processus de validation du code.
- Les courbes de speed-up et d'efficacité de votre code parallèle et autres courbes que vous jugerez utiles pour l'analyse de votre code.
- Votre analyse **détaillée** de vos résultats et vos conclusions.

### 2. Le code // documenté et commenté, avec les commandes de compilation.

## Annexe:

```
subroutine Rename(Me,name)
  implicit none
  integer :: Me
  character*13 :: name
  character*3 :: tn
  integer :: i1,i2,i3
  i1 = Me/100
  i2 =( Me - 100*i1)/10
  i3 = Me - 100*i1 -10*i2
  tn = char(i1+48)//char(i2+48)//char(i3+48)
  name='sol'//tn//'.dat'
end subroutine Rename
```