– PHS3903 –
Projet de simulation

Méthode de la matrice 2D

Jérémie Villeneuve

Maksim Skorobogatiy

7 février 2023

Particularités de la méthode de la matrice 2D

La méthode de la matrice suit toujours les mêmes grandes étapes, peu importe le type d'équation considéré.

Méthode de la matrice

1. Écrire l'équation différentielle et définir la forme du domaine de simulation.
   Opérateur différentiel 2D
   Domaine 2D
2. Définir les conditions aux limites et les conditions initiales sur le domaine de simulation.
   Frontière passe de 2 à plusieurs points.
3. Discrétiser l'équation différentielle à l'intérieur du domaine.
4. Discrétiser l'équation différentielle aux limites du domaine.
5. Poser un système d'équations sous forme matricielle ($\mathbf{A}\vec{u} = \vec{b}$)
   Structure et stockage de la matrice **A**
6. Résoudre numériquement l'équation différentielle en résolvant le système d'équations construit à l'étape 5.

ÉDP linéaires
(dépendantes du temps ou non)

Nous allons encore une fois utiliser l'**équation de diffusion** comme exemple.

Équation de diffusion 2D

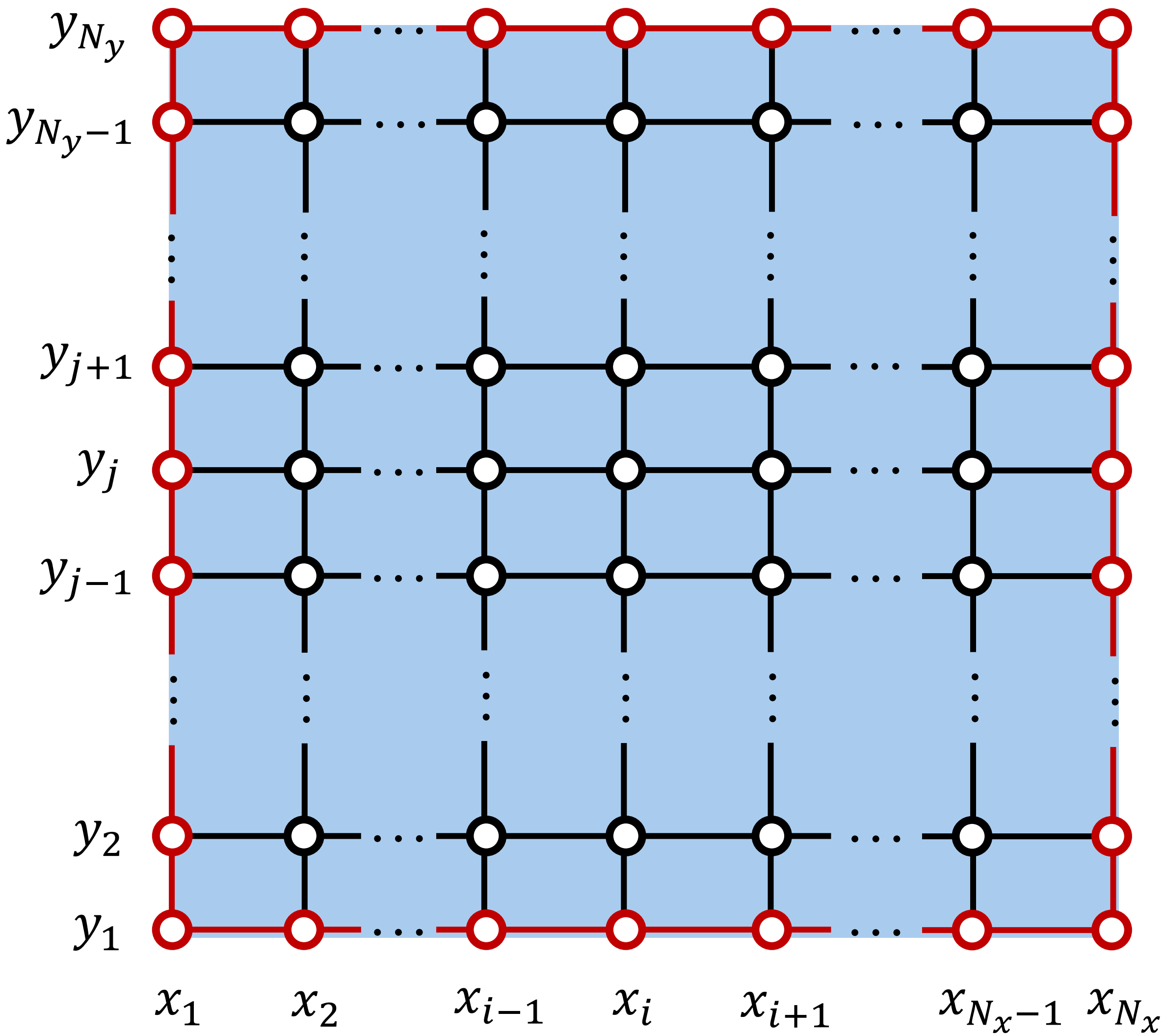
$$\alpha \frac{\partial u(x,y,t)}{\partial t} = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} - g(x,y,t)$$

L'équation de diffusion en régime stationnaire devient l'**équation de Poisson**.

Équation de Poisson 2D

$$\nabla^2 u(x,y) = \frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = g(x,y)$$

Domaine de simulation 2D



$$x_i = x_1 + (i-1)\Delta x,$$
$$i = 1, \ldots, N_x$$

$$y_i = y_1 + (j-1)\Delta y,$$
$$j = 1, \ldots, N_y$$

Si l'équation dépend du temps :

$$t_n = n\Delta t,$$
$$n = 0,1, \ldots$$

Fonction u au point (x_i, y_j) au temps t_n :
$$u_{i,j}^n$$

Plan du cours

- **Méthode de la matrice 2D**
  - Différentiation en 2D
    - **Laplacien sur un maillage uniforme**
    - Construction de la matrice $\mathbf{A}$ des coefficients
    - Application : chauffage d'une pièce (code sur Moodle)
  - Considérations de mémoire
    - Matrices pleines et matrices creuses

Différentiation en 2D

Laplacien 2D (coordonnées cartésiennes)
On veut discrétiser l'opérateur différentiel suivant :

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

La manière la plus directe est d'utiliser deux fois la formule aux différences finies centrée d'ordre 2 pour la dérivée seconde en 1D :

$$\nabla^2 u(x_i, y_j) = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2} + O(\Delta x^2) + O(\Delta y^2)$$

Si le pas de la grille est le même en x qu'en y ($h = \Delta x = \Delta y$), alors on obtient :

Laplacien cartésien 2D à 5 points

$$\boxed{\nabla^2 u(x_i, y_j) = \frac{u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1}}{h^2} + O(h^2)}$$

	1	
1	-4	1
	1	

Est-ce la seule façon de discrétiser le laplacien en 2D ?

Laplacien cartésien discrétisé en 2D

On pourrait obtenir une autre formule à 5 points qui utilise les points de la grille aux coins adjacents au nœud (i, j). Il faut alors utiliser le développement de Taylor de la fonction $u(x, y)$ au voisinage de (x_i, y_j).

On cherche les coefficients A, B, C, D et E tels que :

$$\nabla^2 u = \boxed{\begin{matrix} E & & B \\ & A & \\ D & & C \end{matrix}} + O(h^m)$$

Développement de Taylor multivariable

$$u(\vec{r}_0 + \Delta\vec{r}) = u(\vec{r}_0) + \sum_{k=1}^{\infty} \frac{1}{k!} \left[\left(\Delta\vec{r} \cdot \vec{\nabla} \right)^k u(\vec{r}) \right|_{\vec{r}=\vec{r}_0}$$

2D $$u(x_0 + \Delta x, y_0 + \Delta y) = u(x_0, y_0) + \sum_{k=1}^{\infty} \frac{1}{k!} \left[\left(\Delta x \frac{\partial}{\partial x} + \Delta y \frac{\partial}{\partial y} \right)^k u(x, y) \right|_{\substack{x=x_0 \\ y=y_0}}$$

Laplacien cartésien discrétisé en 2D

On écrit le développement pour chacun des quatre coins pour un pas $h = \Delta x = \Delta y$ constant dans les deux directions.

N.B. Les variables en exposant représentent des dérivées.

E		B
	A	
D		C

$$B = 1 \qquad u_{i+1,j+1} = u_{i,j} + h\left(u_{i,j}^{x} + u_{i,j}^{y}\right) + \frac{h^2}{2}\left(u_{i,j}^{xx} + 2u_{i,j}^{xy} + u_{i,j}^{yy}\right) + O\left(h^3\right)$$

$+$

$$C = 1 \qquad u_{i+1,j-1} = u_{i,j} + h\left(u_{i,j}^{x} - u_{i,j}^{y}\right) + \frac{h^2}{2}\left(u_{i,j}^{xx} - 2u_{i,j}^{xy} + u_{i,j}^{yy}\right) + O\left(h^3\right)$$

$+$

$$E = 1 \qquad u_{i-1,j+1} = u_{i,j} + h\left(-u_{i,j}^{x} + u_{i,j}^{y}\right) + \frac{h^2}{2}\left(u_{i,j}^{xx} - 2u_{i,j}^{xy} + u_{i,j}^{yy}\right) + O\left(h^3\right)$$

$+$

$$D = 1 \qquad u_{i-1,j-1} = u_{i,j} + h\left(-u_{i,j}^{x} - u_{i,j}^{y}\right) + \frac{h^2}{2}\left(u_{i,j}^{xx} + 2u_{i,j}^{xy} + u_{i,j}^{yy}\right) + O\left(h^3\right)$$

$$u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1} = 4u_{i,j} + 2h^2\left(u_{i,j}^{xx} + u_{i,j}^{yy}\right) + O\left(h^4\right)$$

(Les ordres impairs s'annulent.)

Laplacien cartésien discrétisé en 2D

$$u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1} = 4u_{i,j} + 2h^2\left(u_{i,j}^{xx} + u_{i,j}^{yy}\right) + O\left(h^4\right)$$

Le laplacien que l'on souhaite évaluer est le terme $\left(u_{i,j}^{xx} + u_{i,j}^{yy}\right)$.

Laplacien cartésien 2D à 5 points (x)

$$\nabla^2 u = \frac{0.5u_{i-1,j-1} + 0.5u_{i-1,j+1} - 2u_{i,j} + 0.5u_{i+1,j-1} + 0.5u_{i+1,j+1}}{h^2} + O\left(h^2\right)$$

0.5		0.5
	-2	
0.5		0.5

Laplacien cartésien 2D à 5 points (+)

$$\nabla^2 u = \frac{u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1}}{h^2} + O\left(h^2\right)$$

	1	
1	-4	1
	1	

Quelle est la différence entre ces deux laplaciens ?

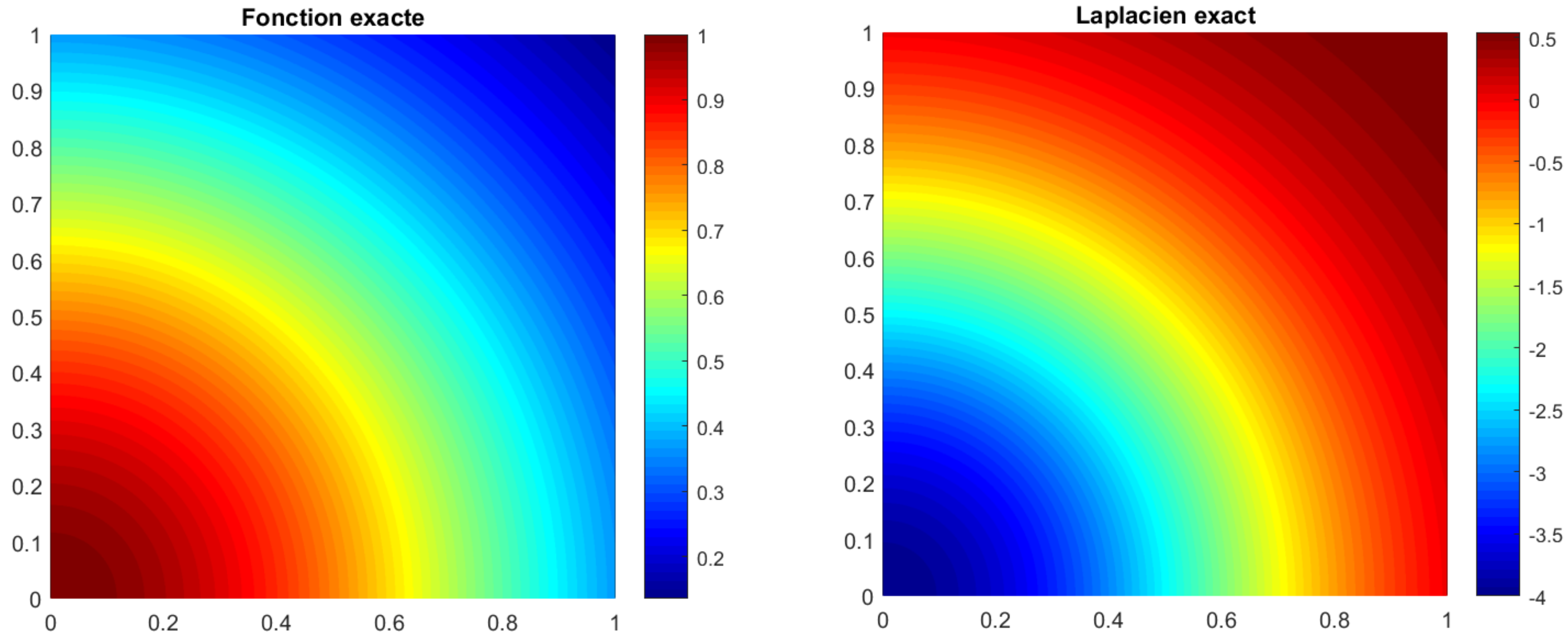
Laplaciens à 5 points

Pour observer les différences entre les deux laplaciens à 5 points, on utilise la fonction test avec une symétrie radiale

$$u(x,y) = e^{-r^2} = e^{-(x^2+y^2)}$$

dont le laplacien exact a aussi une symétrie radiale

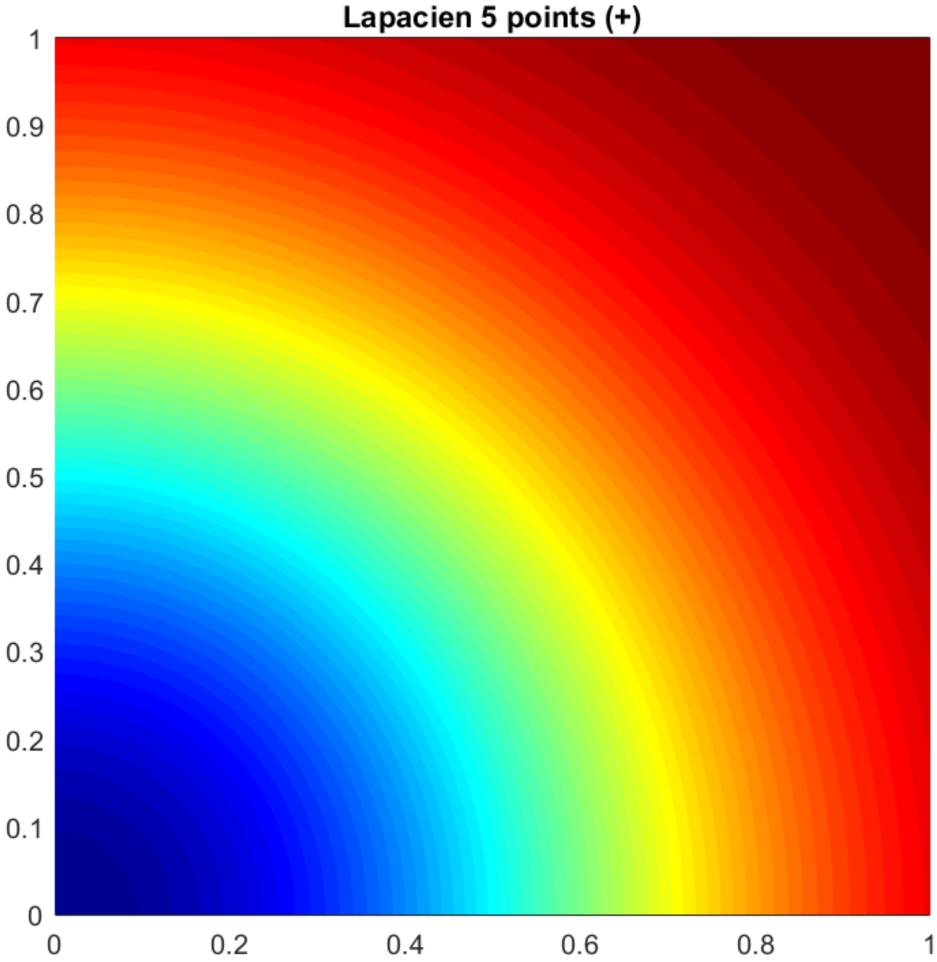
$$\nabla^2 u(x,y) = 4\left(x^2+y^2-1\right)e^{-(x^2+y^2)} = 4\left(r^2-1\right)e^{-r^2}.$$

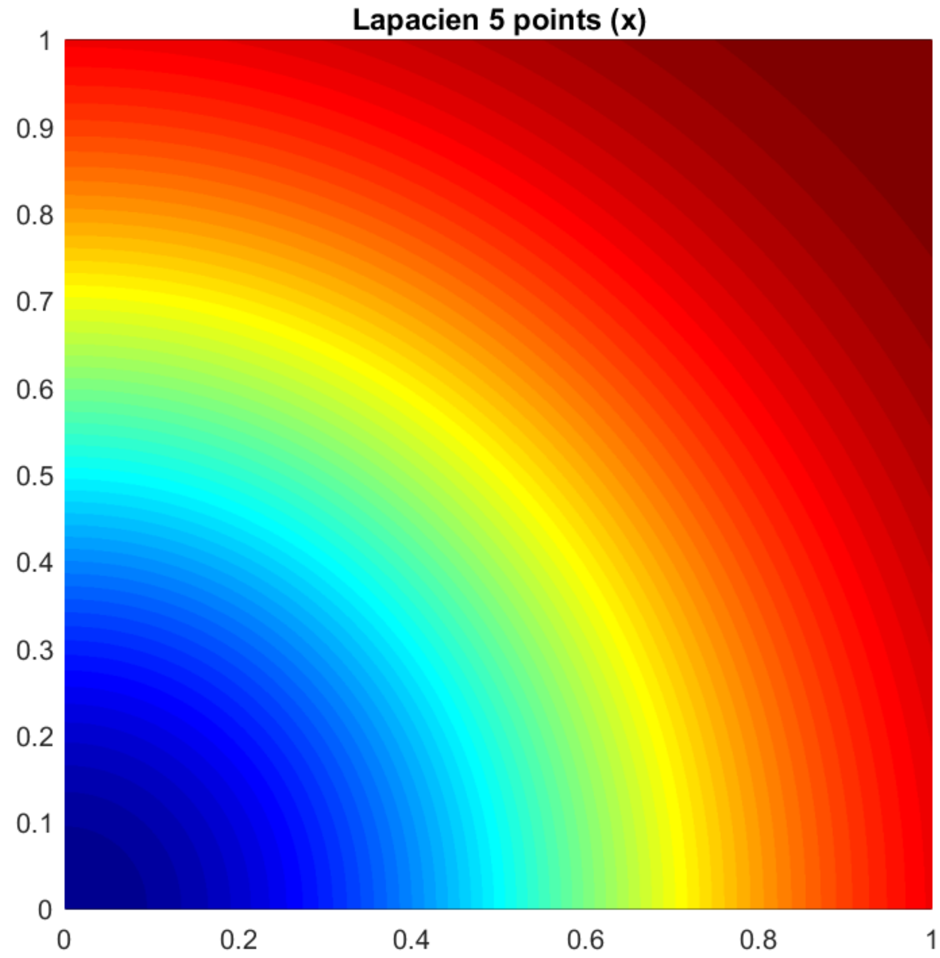


Comparaison des laplaciens à 5 points

	1	
1	-4	1
	1	

0.5		0.5
	-2	
0.5		0.5



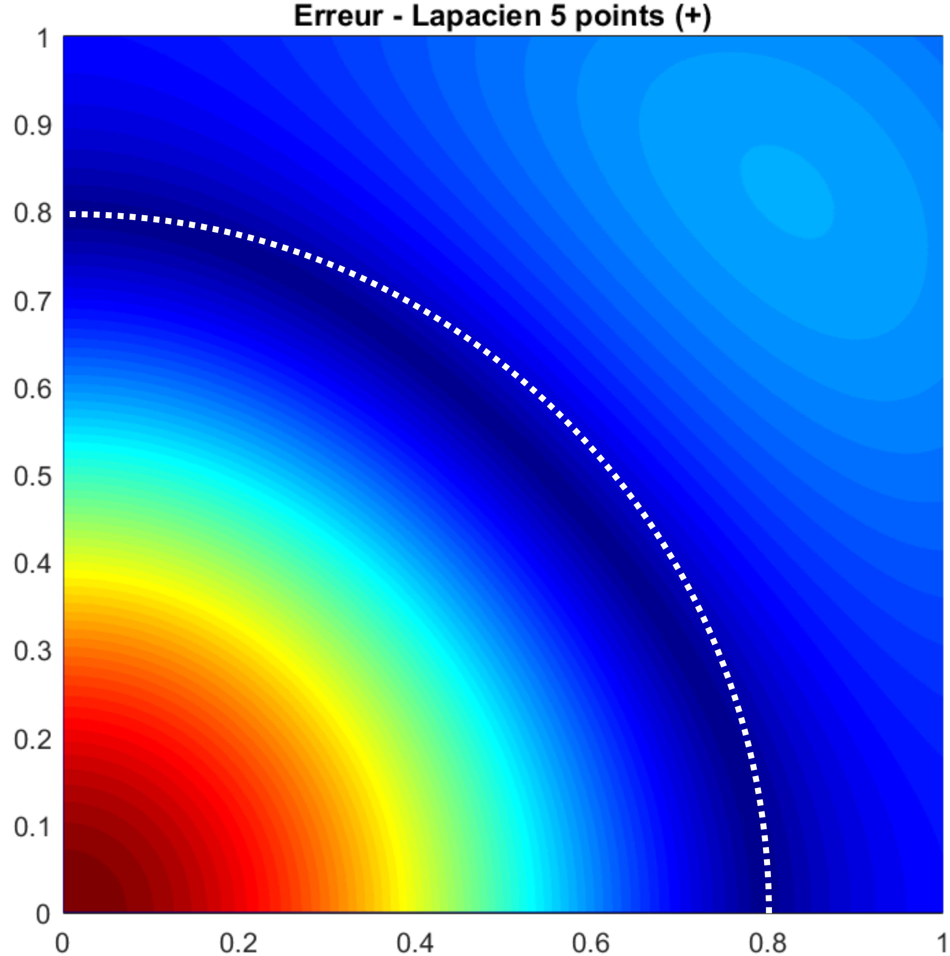


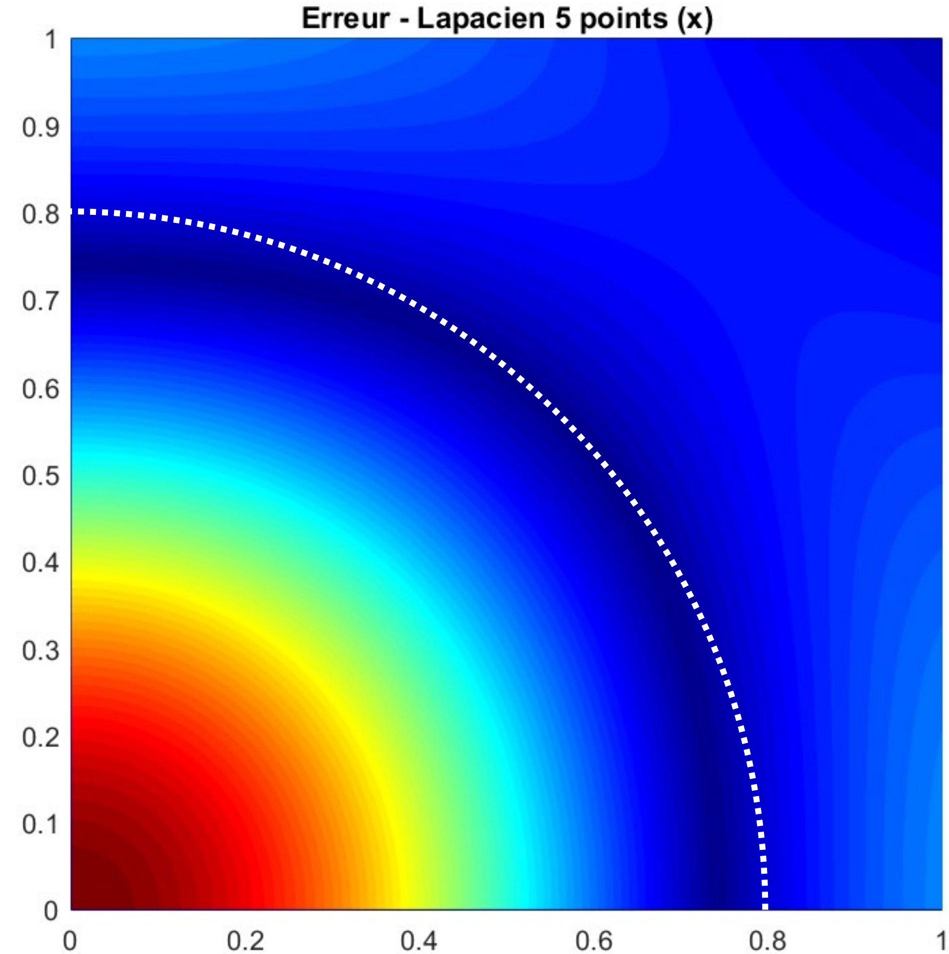
Comparaison des laplaciens à 5 points (erreur absolue)

	1	
1	-4	1
	1	

L'erreur n'a pas une symétrie radiale !

0.5		0.5
	-2	
0.5		0.5





Erreur du laplacien

Même si la fonction de test $u(x,y)$ possède une symétrie radiale, les deux laplaciens à 5 points ne reflètent pas cette propriété. Cela est dû à leur terme d'erreur à l'ordre h^2.

Laplacien cartésien 2D à 5 points (+)

$$\nabla^2 u = \frac{1}{h^2}\boxed{\begin{matrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{matrix}}\,u + \frac{h^2}{12}\left(\frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4}\right) + O\left(h^4\right)$$

Laplacien cartésien 2D à 5 points (x)

$$\nabla^2 u = \frac{1}{h^2}\boxed{\begin{matrix} 0.5 & & 0.5 \\ & -2 & \\ 0.5 & & 0.5 \end{matrix}}\,u + \frac{h^2}{12}\left(\frac{\partial^4 u}{\partial x^4} + 6\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4}\right) + O\left(h^4\right)$$

Si $u(x,y) = u(r)$ possède une symétrie radiale, pour que le terme d'erreur possède la symétrie radiale, il doit être multiple de :

$$\left(\frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4}\right) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)^2 u(x,y) = \left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right)\right)^2 u(r)$$

Laplacien à 9 points (symétrie radiale)

On arrive à obtenir la forme d'erreur voulue en prenant une combinaison linéaire particulière des laplaciens à 5 points.

$$\nabla^2 u \to \gamma \frac{1}{h^2} \boxed{\begin{matrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{matrix}} + (1-\gamma) \frac{1}{h^2} \boxed{\begin{matrix} 0.5 & & 0.5 \\ & -2 & \\ 0.5 & & 0.5 \end{matrix}} + O(h^2)$$

$$\gamma = \frac{2}{3}$$

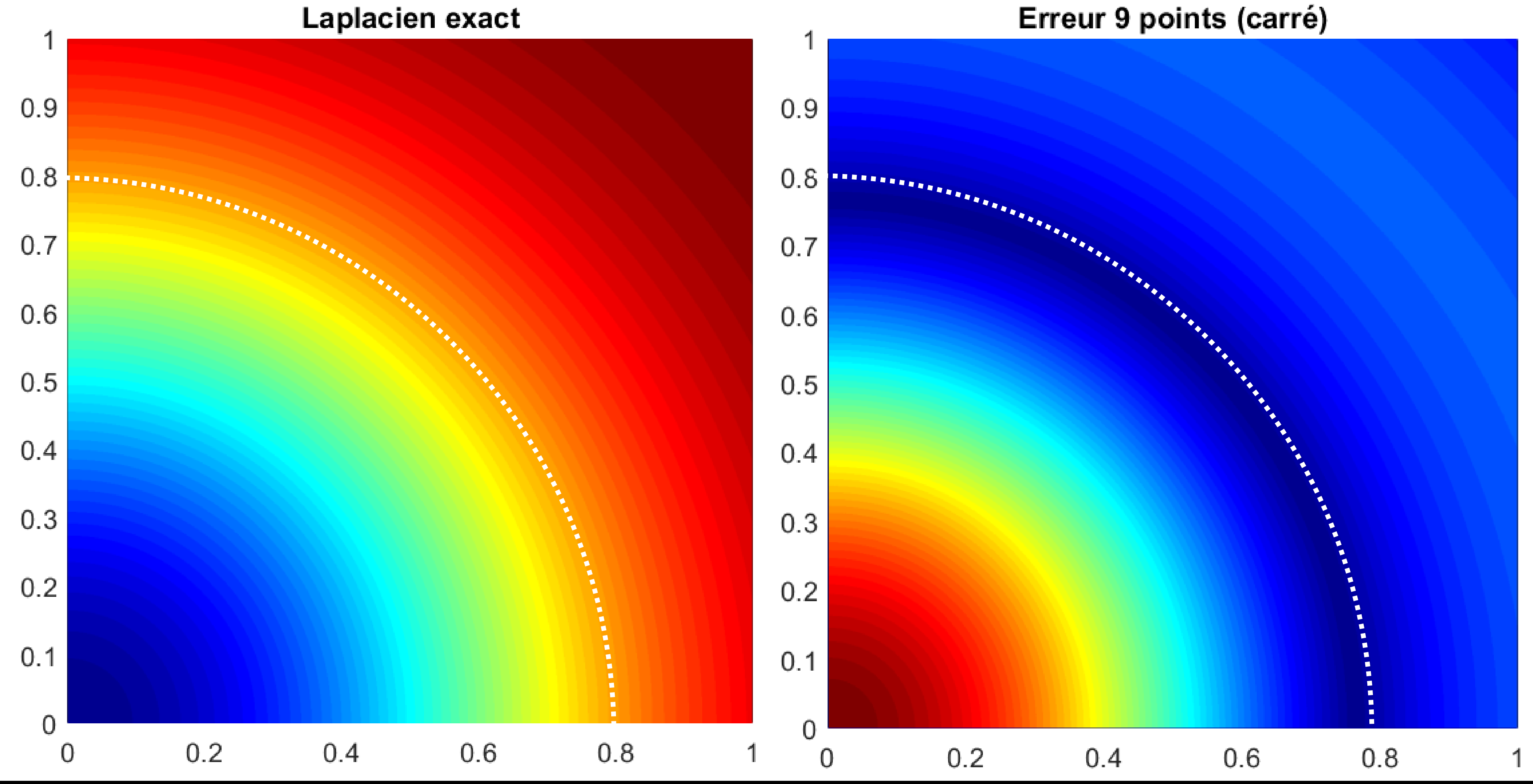
Laplacien cartésien 2D à 9 points (carré)

$$\nabla^2 u \to \frac{1}{6h^2} \boxed{\begin{matrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{matrix}} u + O(h^2)$$

Le laplacien à 9 points est toujours d'ordre $O(h^2)$ même si le nombre de points utilisés a augmenté.

Laplacien à 9 points (symétrie radiale)

L'erreur entre le laplacien à 9 points et le laplacien exact respecte maintenant la symétrie radiale de la fonction test.



Plan du cours

- **Méthode de la matrice 2D**
  - Différentiation en 2D
    - Laplacien sur un maillage uniforme
    - **Construction de la matrice $\mathbf{A}$ des coefficients**
    - **Application : chauffage d'une pièce (code sur Moodle)**
  - Considérations de mémoire
    - Matrices pleines et matrices creuses

Construction de la matrice des coefficients

On considère un maillage 5 x 5 de pas $h = \Delta x = \Delta y$ sur lequel on veut résoudre l'équation de diffusion de la chaleur en régime permanent.

$$\nabla^2 T(x,y) = -\frac{S(x,y)}{k(x,y)}$$

Température fixée à T_p

$$T_{i,1} = T_p \qquad T_{i,5} = T_p \qquad i = 1, \dots, 5$$

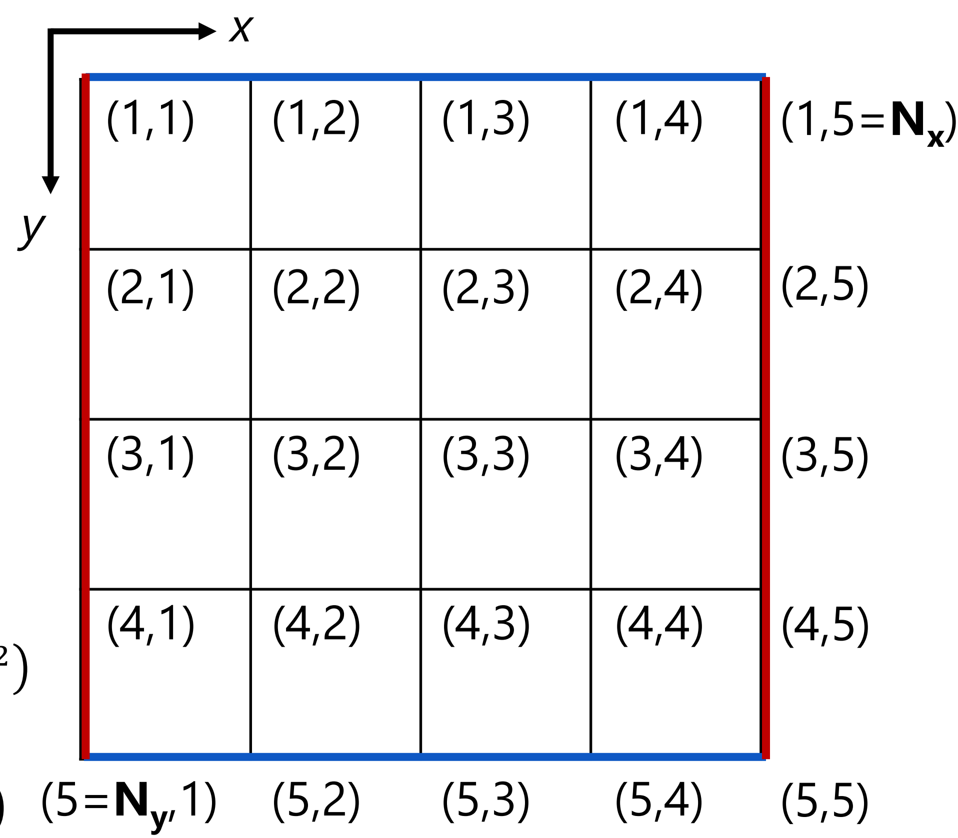
Condition de convection avec température ambiante T_a

$$-k\frac{\partial T}{\partial y} = \pm \hbar (T - T_a)$$

$$-k_{1,j}\frac{-3T_{1,j} + 4T_{2,j} - T_{3,j}}{2h} = -\hbar\left(T_{1,j} - T_a\right) + O\left(h^2\right)$$

$$-k_{5,j}\frac{T_{3,j} - 4T_{4,j} + 3T_{5,j}}{2h} = \hbar\left(T_{5,j} - T_a\right) + O\left(h^2\right)$$

$$j = 2, \dots, 4$$



Convention de Matlab pour les indices
$M_{i,j}$: i – ligne (y), j – colonne (x)

Construction de la matrice des coefficients

En utilisant le laplacien à 5 points (+), on a pour les points intérieurs du domaine :

$$T_{i,j-1} + T_{i,j+1} - 4T_{i,j} + T_{i-1,j} + T_{i+1,j} = -\frac{S_{i,j}}{k_{i,j}} h^2 + O\left(h^4\right) \qquad \begin{array}{l} i = 2,3,4 \\ j = 2,3,4 \end{array}$$

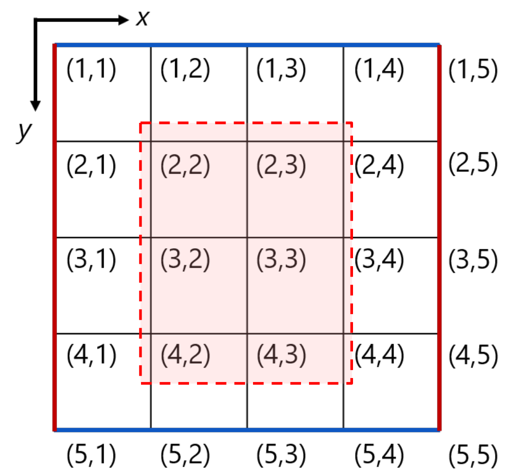
Nombre d'équations à résoudre

(disons $N_x = N_y = N$)

Intérieur : 9 équations ($(N-2)^2$ en général)

Frontières : 16 équations ($4(N-1)$ en général)

Total : 25 équations (N^2 en général)



Il y a encore une équation à résoudre par nœud de la grille.

La taille de la matrice des coefficients du système d'équations est donc N^2=25 x N^2 =25 ($N_x{\cdot}N_y \times N_x{\cdot}N_y$ en général).

Matrice des coefficients

Numérotation colonne par colonne.
Équation sur nœud #

Numérotation colonne par colonne.
Équation sur nœud #

$N_x \cdot N_y$

N_y

11	1																									T_{11}	T_p
21		1																								T_{21}	T_p
31			1																							T_{31}	T_p
41				1																						T_{41}	T_p
51					1																					T_{51}	T_p
12						α	-4	1																		T_{12}	βT_a
22		1				1	-4	1				1														T_{22}	$-\gamma S_{22}$
32			1				1	-4	1				1													T_{32}	$-\gamma S_{23}$
42				1				1	-4	1				1												T_{42}	$-\gamma S_{24}$
52								1	-4	α																T_{52}	βT_a
13											α	-4	1													T_{13}	βT_a
23							1				1	-4	1				1									T_{23}	$-\gamma S_{32}$
33								1				1	-4	1				1								T_{33}	$-\gamma S_{33}$
43									1				1	-4	1				1							T_{43}	$-\gamma S_{34}$
53													1	-4	α											T_{53}	βT_a
14																α	-4	1								T_{14}	βT_a
24												1				1	-4	1				1				T_{24}	$-\gamma S_{42}$
34													1				1	-4	1				1			T_{34}	$-\gamma S_{43}$
44														1				1	-4	1				1		T_{44}	$-\gamma S_{44}$
54																		1	-4	α						T_{54}	βT_a
15																					1					T_{15}	T_p
25																						1				T_{25}	T_p
35																							1			T_{35}	T_p
45																								1		T_{45}	T_p
55																									1	T_{55}	T_p

$N_x \cdot N_y$ $N_x \cdot N_y$

$$\mathbf{A}\vec{T} = \vec{b}$$

$$\mathbf{A}\vec{T} = \vec{b}$$

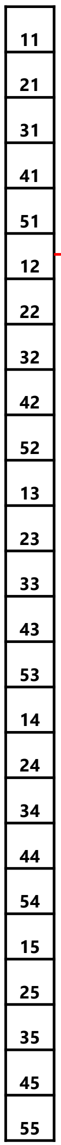
$$\alpha \equiv 3 + \frac{2\hbar}{k}h$$

$$\beta \equiv \frac{2\hbar}{k}h$$

$$\gamma \equiv \frac{h^2}{k}$$

Matrice des coefficients

Numérotation colonne par colonne.
Équation sur nœud #



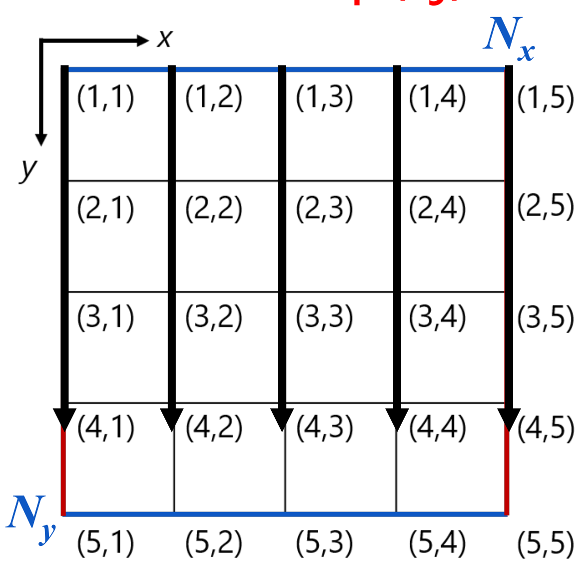
Ex. (1,2)

p(1,2)=1+(2-1)*5=6

noeud (i,j)

correspond à la ligne

$p(i,j)=i+(j-1)*N_y$ de la matrice



$$\mathbf{A}\vec{T} = \vec{b}$$

Sur le noeud (i,j) on code l'équation appropriée par le choix des coefficients sur la ligne p(i,j) de **A**
A(p(i,j) , …)

Matrice des coefficients

Numérotation colonne par colonne.
Équation sur nœud #

11
21
31
41
51
12
22
32
42
52
13
23
33
43
53
14
24
34
44
54
15
25
35
45
55

noeud (i,j)

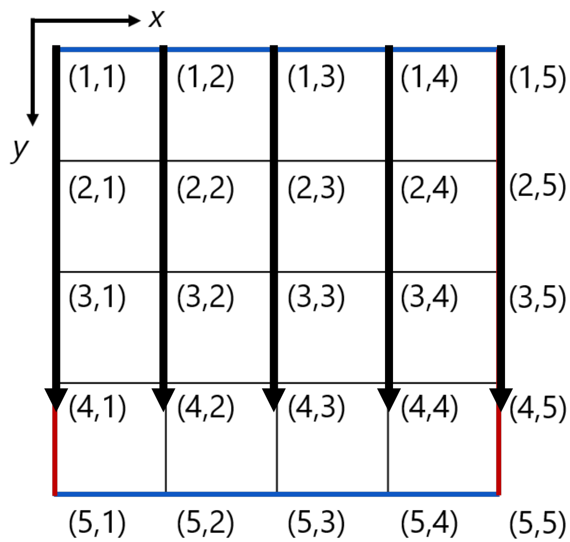
ligne $p(i,j)=i+(j-1)*N_y$
de la matrice A

$$\mathbf{A}\vec{T} = \vec{b}$$

?

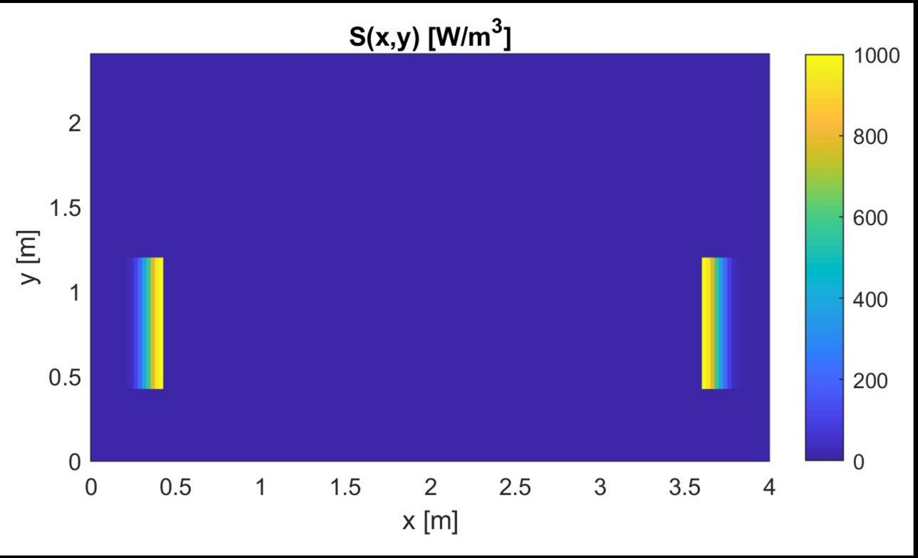
p(i,j)

$$T_{i,j-1} + T_{i,j+1} - 4T_{i,j} + T_{i-1,j} + T_{i+1,j} = -\frac{S_{i,j}}{k_{i,j}}h^2 + O(h^4)$$

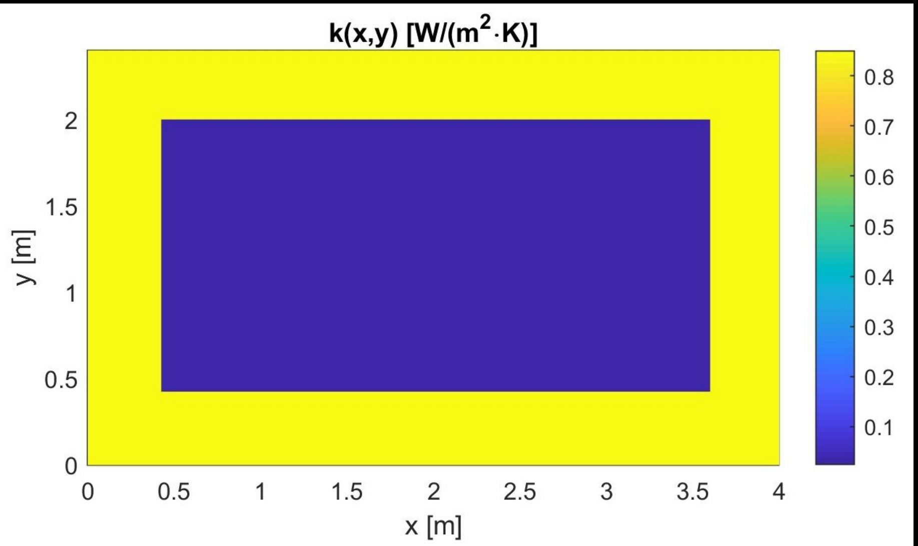


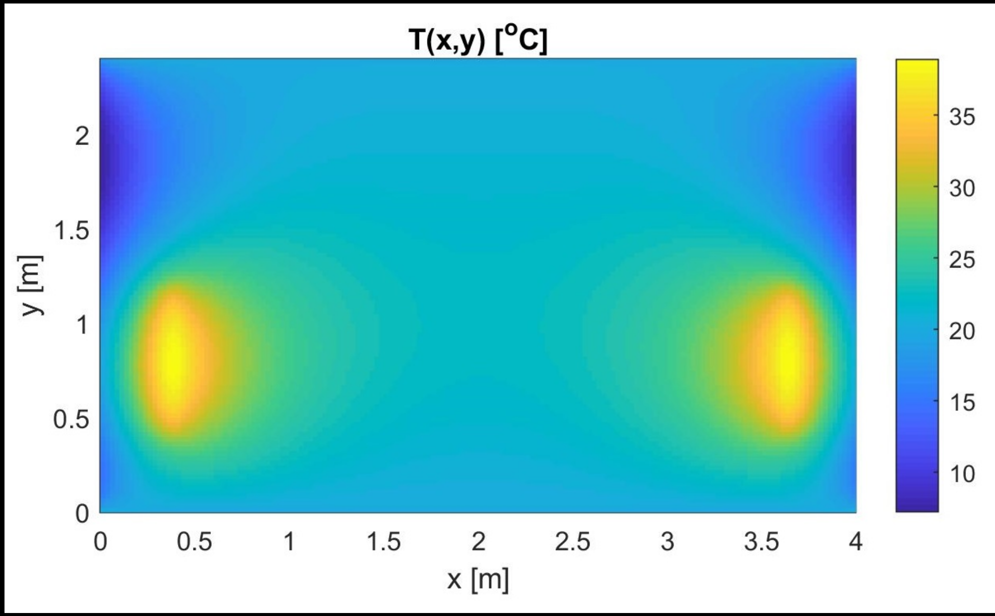
A(p(i,j) , p(i,j-1))=1
A(p(i,j) , p(i,j+1))=1
A(p(i,j) , p(i,j))=-4
A(p(i,j) , p(i-1,j))=1
A(p(i,j) , p(i+1,j))=1

Code Moodle – Chauffage d'une pièce



Equation_independante_du_temps_2D.m





Plan du cours

- **Méthode de la matrice 2D**
  - Différentiation en 2D
    - Laplacien sur un maillage uniforme
    - Construction de la matrice $\mathbf{A}$ des coefficients
    - Application : chauffage d'une pièce (code sur Moodle)
  - Considérations de mémoire
    - **Matrices pleines et matrices creuses**

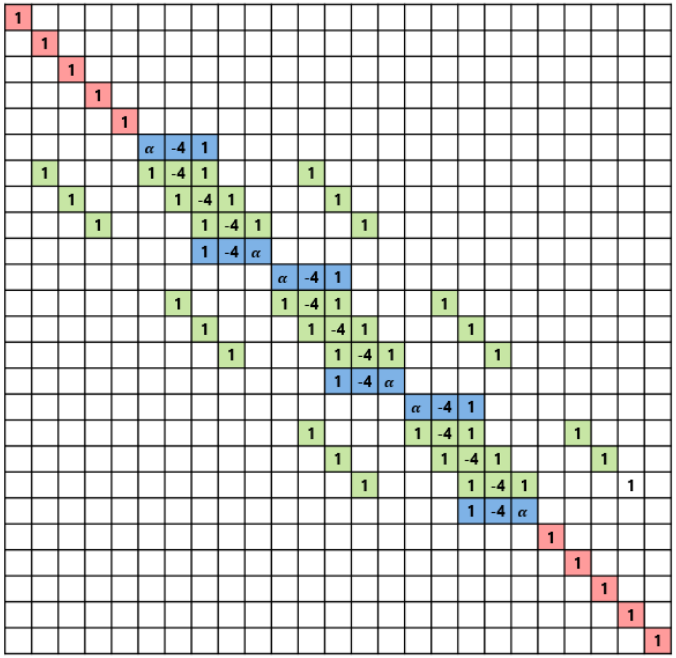
Structure de la matrice des coefficients

Par rapport aux problèmes 1D, les matrices de coefficients des problèmes en 2D ont une **largeur de bande** (*bandwidth*) plus élevée.

Matrice 1D

$$\begin{bmatrix} \begin{pmatrix} 2c_2\Delta x \\ -3c_1 \end{pmatrix} & 4c_1 & c_1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & -2 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & -2 & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & -2 & 1 & \cdot & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdot & \cdot & \cdot & \cdot & 1 & -2 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & -2 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & d_1 & -4d_1 & \begin{pmatrix} 2d_2\Delta x \\ +3d_1 \end{pmatrix} \end{bmatrix}$$

Matrice 2D



1D : largeur de bande = #Bandes-1 = 2

2D : largeur de bande = N_y = 5

La structure de bandes de la matrice dépend du schéma de discrétisation utilisé pour les opérateurs différentiels.

Matrices creuses

Une **matrice creuse** (*sparse matrix*) est une matrice qui contient un **grand nombre de zéros.**

Quel est le pourcentage d'éléments non nuls dans la matrice 2D ci-contre ?

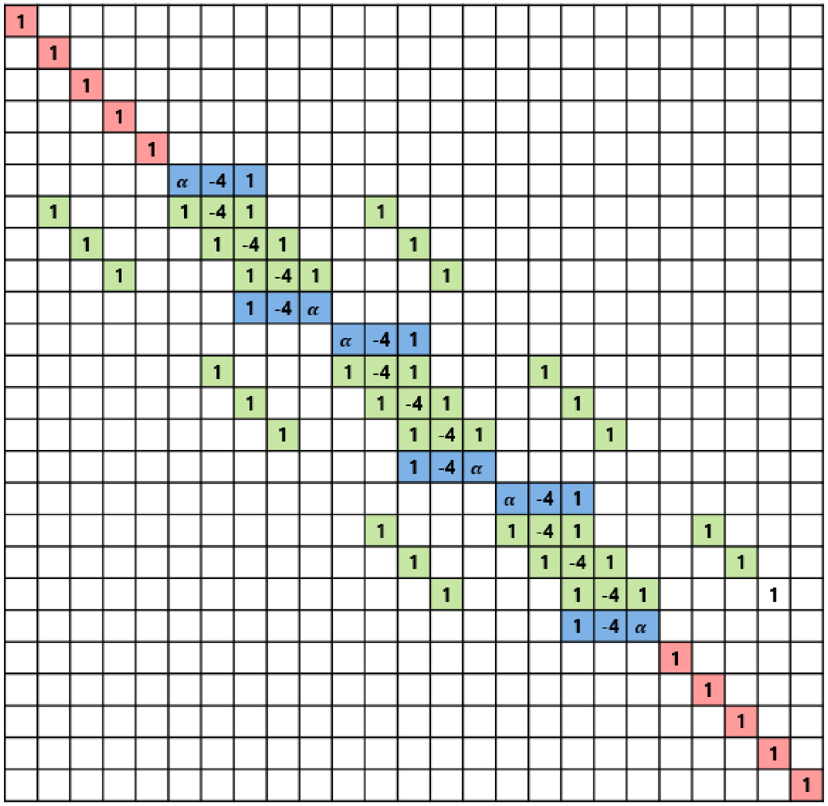
$$\eta = \frac{10 + 3 \times 6 + 5 \times 9}{25^2} = \frac{73}{625} \approx 12\ \%$$

Pour une grille N x N :

$$\eta = \frac{2N + 3 \times 2(N-2) + 5 \times (N-2)^2}{N^4}$$
$$= \frac{5N^2 - 12N + 9}{N^4} \sim \frac{5}{N^2}$$

Plus le nombre de points est élevé, plus la matrice est creuse !

Matrice 2D



Matrices creuses

Évidemment, il est inutile de stocker en mémoire tous les zéros d'une matrice creuse. C'est pourquoi **on stocke seulement les éléments non nuls** de la matrice.

*Quelle quantité de mémoire doit-on allouer pour stocker une **matrice pleine N² x N²** composée de nombres en **arithmétique flottante double précision** ?*

Chaque élément de la matrice est constitué de 64 bits, soit 8 octets (*bytes*).

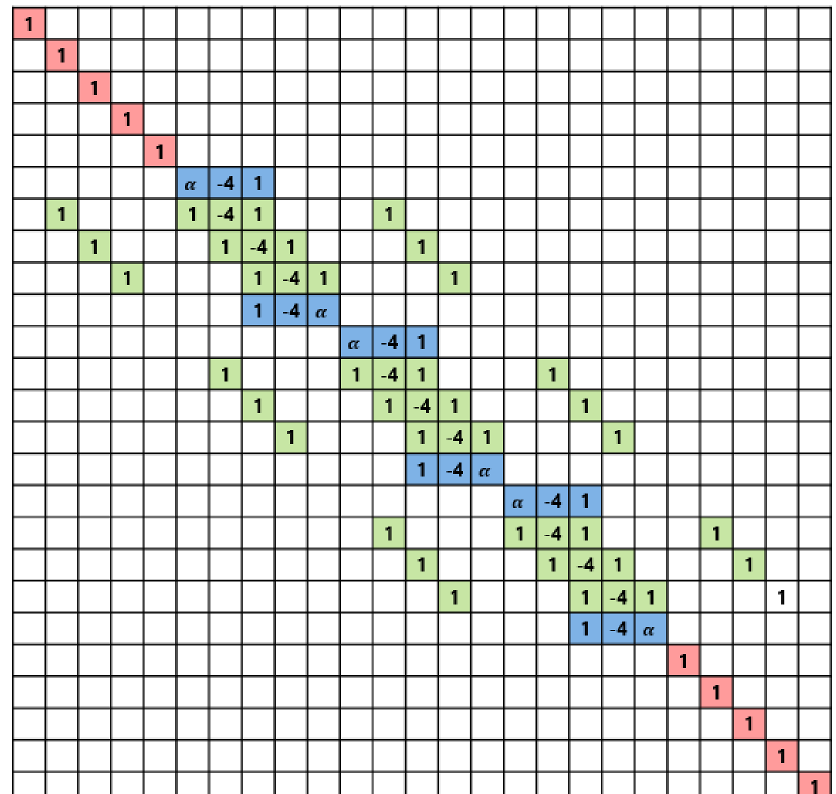
Il y a $N^2 \times N^2 = N^4$ éléments dans la matrice pleine.

Mémoire à allouer : $\sim 8N^4$ octets

Pour $N = 5$, on obtient ~ 5 ko.

Pour $N = 100$, on obtient ~ 800 Mo.

Matrice 2D



Matrices creuses

Évidemment, il est inutile de stocker en mémoire tous les zéros d'une matrice creuse. C'est pourquoi **on stocke seulement les éléments non nuls** de la matrice.

Quelle quantité de mémoire doit-on allouer pour stocker une ***matrice creuse N² x N²*** *composée de nombres en* ***arithmétique flottante double précision*** *?*

Matrice pleine : $\sim 8N^4$ octets

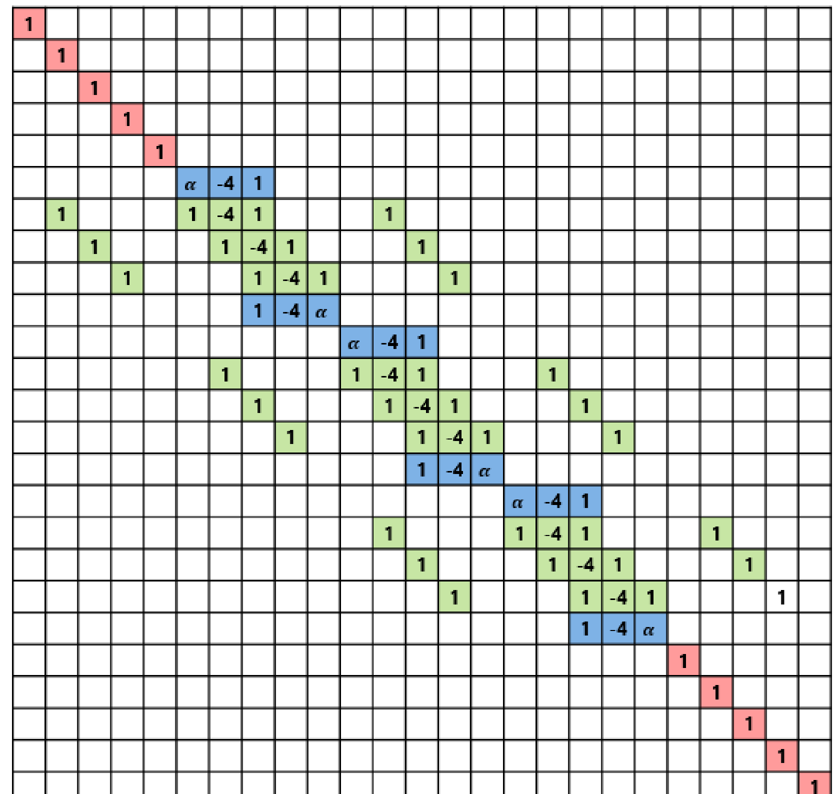
Fraction d'éléments non nuls : $\sim \frac{5}{N^2}$

Matrice creuse : $\sim 40N^2$ octets

Pour $N = 5$, on obtient ~ 1ko.

Pour $N = 100$, on obtient ~ 400 ko.

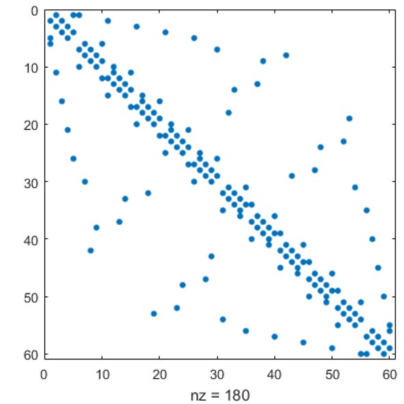
Matrice 2D



Matrices creuses dans MATLAB

Pour travailler avec des matrices creuses dans MATLAB, il faut utiliser certaines commandes spécifiques.

`S = sparse(A)`	Convertit une matrice pleine en matrice creuse
`S = sparse(m,n)`	Crée une matrice creuse (implicite) $m \times n$ remplie de zéros
`S = sparse(i,j,v)`	Crée une matrice creuse dans laquelle `S(i(k),j(k)) = v(k)`.
`S = speye(m,n)`	Crée une matrice creuse $m \times n$ diagonale remplie de 1
`cond = issparse(A)`	Retourne 1 si A est creuse et 0 si elle est pleine.
`spy(S)`	Trace un graphe de la structure de la matrice creuse
`nnz(S)`	Retourne le nombre d'éléments non nuls dans la matrice creuse



Site Web de MATLAB

Matrices creuses dans PYTHON

Pour travailler avec des matrices creuses dans PYTHON, il faut utiliser une bibliothèque des fonctions SCIPY et certaines commandes spécifiques.

```
S = scipy.sparse.csr_matrix((m,n), dtype=np.double);
```

Crée une matrice creuse (implicite) $m \times n$ remplie de zéros

```
S = scipy.sparse.csr_matrix((data,(row,col)), shape=(m,n), dtype=np.double);
```

Crée une matrice creuse dans laquelle

`S(row(k),col(k)) = data(k)`.

```
S = scipy.sparse.eye(m, n, k)
```

Crée une matrice creuse $m \times n$ avec diagonale k remplie de 1

```
cond = scipy.sparse.issparse(S)
```

Retourne 1 si S est creuse et 0 si elle est pleine.

`S.getnnz()` Retourne le nombre d'éléments non nuls dans la matrice creuse

Site Web de SCIPY https://docs.scipy.org/doc/scipy/