Thème 3: résolution numérique des EDP linéaires
Problèmes statiques et dynamiques

MNCS – Albert Hertzog, Jacques Lefrère, Jean Hare, Noé Lahaye

Sorbonne Université
MNCS
Méthodes numériques et calcul scientifique

11 et 12 mars 2021

Sommaire

Sommaire

Sommaire

Sommaire

Sommaire

Contexte

- Équations différentielles ordinaires (EDO) : une variable indépendante
- **Équations aux dérivées partielles (EDP)** : plusieurs variables indépendantes

EDP omniprésentes en physique, chimie, sciences de la Terre, biologie : mécanique des fluides, propagation des ondes, électromagnétisme, phénomènes de diffusion...

Se limiter ici à des EDP linéaires régissant un champ scalaire $u(\vec{r}, t)$.
Généralisations vectorielles possibles, et techniquement analogues.

Résolution numérique par la méthode des différences finies avec un maillage régulier.
Mais il existe d'autres méthodes, éléments finis par exemple, mieux adaptées dans le cas de domaines à géométrie complexe.

Exemples d'EDP scalaires les plus connues

1. L'équation des ondes ou « de d'Alembert »

$$\frac{\partial^2 u}{\partial t^2} - c^2 \, \Delta \, u = 0 \tag{1}$$

2. L'équation de Schrödinger

$$i \frac{\partial u}{\partial t} = -\frac{1}{2} \Delta \, u + V \, u \tag{2}$$

3. L'équation de diffusion

$$\frac{\partial u}{\partial t} = \operatorname{div}[D \ \overrightarrow{\operatorname{grad}} \, u] = D \, \Delta \, u \quad \text{si } D \text{ est uniforme} \tag{3}$$

4. Les équations de Poisson et de Laplace

$$\Delta \, u = \rho \qquad\qquad \Delta \, u = 0 \tag{4}$$

⚠ Les équations de Navier-Stokes en mécanique des fluides sont en général non-linéaires à cause du terme d'advection.

Équations elliptiques/paraboliques/hyperboliques

Étude théorique des EDP : vaste domaine en mathématiques.

EDP linéaires (du second ordre) classées selon la forme des coefficients devant les dérivées partielles :

$$\alpha \frac{\partial^2 u}{\partial x^2} + \beta \frac{\partial^2 u}{\partial x \partial y} + \gamma \frac{\partial^2 u}{\partial y^2} = \ldots$$

- $\beta^2 - 4\alpha\gamma > 0$ hyperboliques : équation des ondes
- $\beta^2 - 4\alpha\gamma = 0$ paraboliques : diffusion ou Schrödinger
- $\beta^2 - 4\alpha\gamma < 0$ elliptiques : équations de Laplace, Poisson, Helmholtz

Propriétés mathématiques différentes

$\longrightarrow$ méthodes numériques spécifiques.

Conditions aux limites

L'équation est vérifiée dans un domaine $\mathcal{D}$ de l'espace.
Problème « bien posé » $\longrightarrow$ conditions aux limites (CL).
Plusieurs types de conditions :

- Dirichlet : u imposé sur le bord $\partial\mathcal{D}$.
- Neumann : $\overrightarrow{\mathrm{grad}}\, u.\vec{n}$ (flux) imposé sur $\partial\mathcal{D}$.
- Robin : plus général, combinaison linéaire des deux.
- Cas général : conditions mixtes, i.e. différentes sur différentes parties de $\partial\mathcal{D}$.

Exemple de l'équation de la chaleur.

- Dirichlet $\Leftrightarrow$ parois isothermes (imposent la température)
- Neumann $\Leftarrow$ parois adiabatiques (transfert de chaleur nul)

Problèmes statiques et dynamiques

Distinguer deux classes de problèmes selon la présence de la variable temps :

- Problèmes statiques : déterminer $u(\vec{r})$ en fonction des CL (pas de variable temps explicite).
  La solution statique peut correspondre à un régime stationnaire atteint après un temps d'évolution assez long.
- Problèmes dynamiques : déterminer $u(\vec{r}, t)$ dans $\mathcal{D}$ sur un intervalle de temps + CL spatio-temporelles :
  - CI (conditions initiales) : $u(\vec{r}, t = 0)$
  - CL (conditions aux limites) : $u(\vec{r} = \cdot, t)$

Problèmes statiques

1 - Problèmes statiques : solution d'une EDP où le temps n'intervient pas.

Exemple : solution d'équilibre de l'équation de diffusion lorsque $t \to \infty$.

Équation de Laplace 2D

$$\Delta T = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) T = 0$$

Problèmes dynamiques

2 - Problèmes dynamiques :

Deux méthodes simples seront exposées et analysées sur l'exemple de l'équation de diffusion à 1-D :

$$\frac{\partial u}{\partial t} = D \, \frac{\partial^2 u}{\partial x^2} \tag{5}$$

avec diverses conditions initiales.

Sommaire

Résolution de l'équation de Laplace dans un rectangle

Objectif : solution numérique approchée de $\Delta T = 0$ à l'intérieur de $\mathcal{D}$.

Simplification : $\mathcal{D}$ rectangulaire, de côtés Δx et Δy.

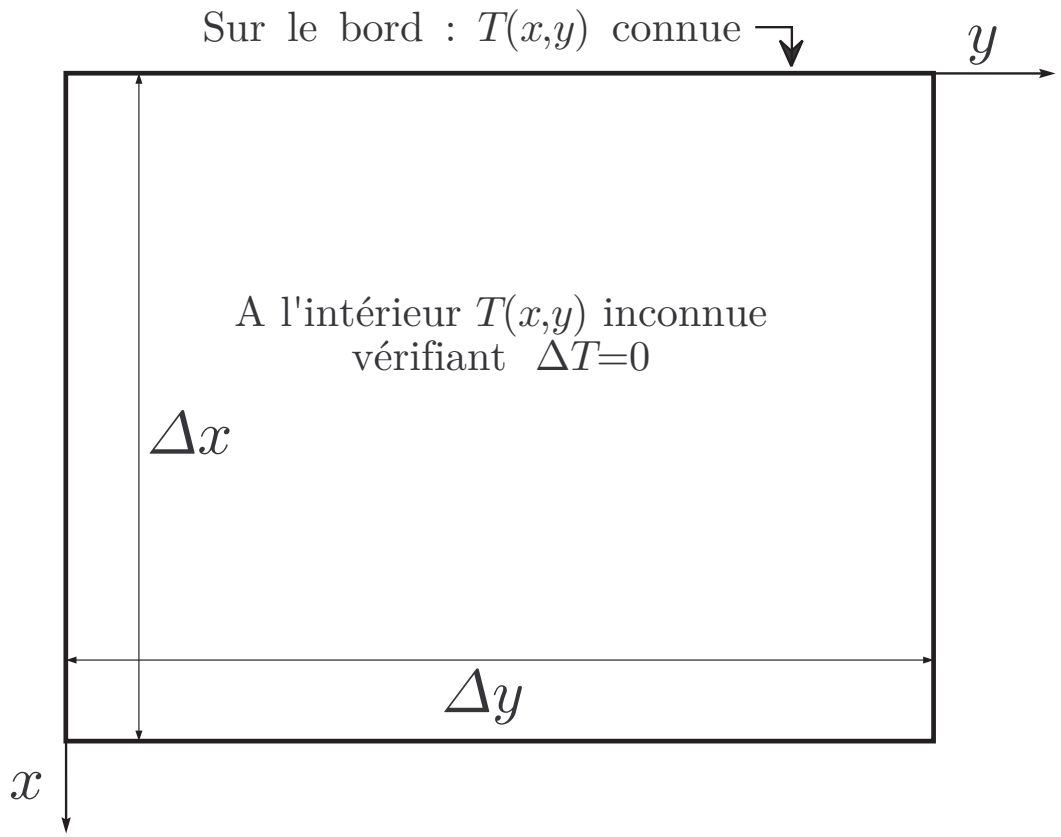


FIGURE 1 – Géométrie du problème de l'équation de Laplace : noter le choix inhabituel de l'orientation des axes x et y

Exemple de carte d'isothermes

CL Dirichlet : températures imposées aux bords

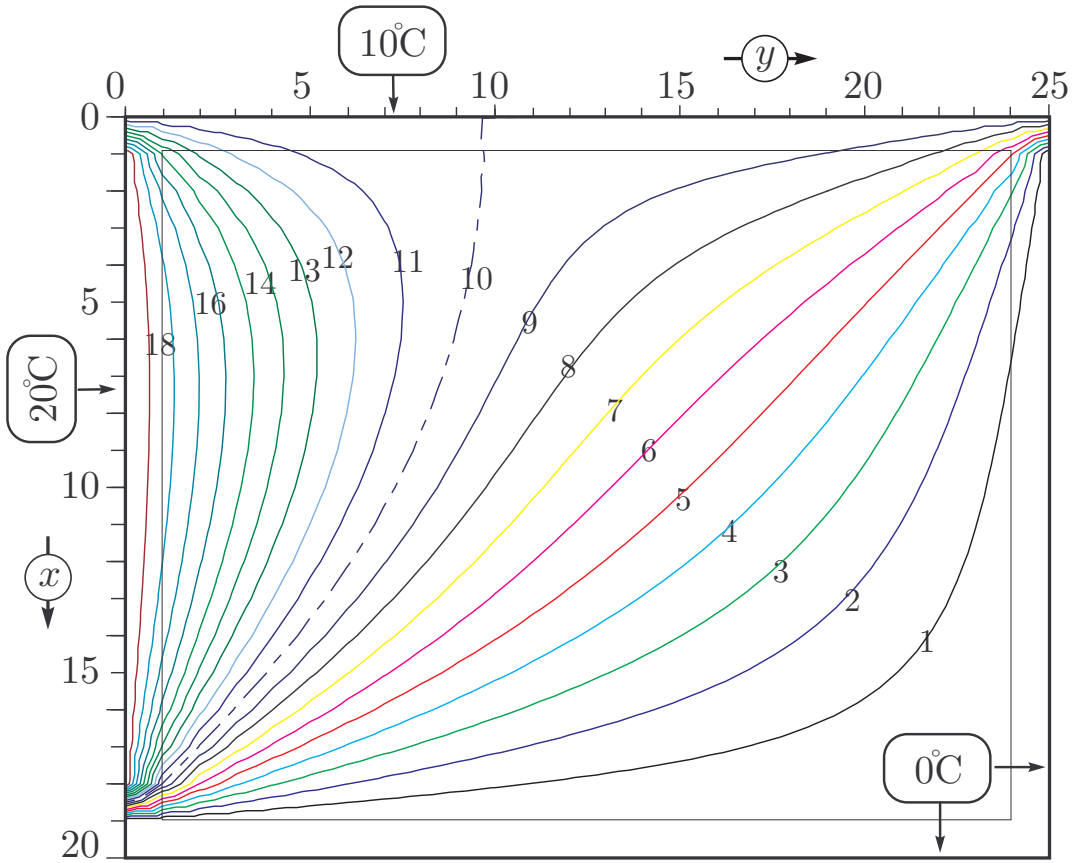


FIGURE 2 – Exemple de solution de l'équation $\Delta T = 0$, dans le cas où une paroi est maintenue à $20°\mathrm{C}$, une autre adjacente est à $10°\mathrm{C}$ et les deux autres parois sont à $T = 0°\mathrm{C}$.

Les lignes indexées de 1 à 18 représentent les isothermes.

Maillage du domaine $\mathcal{D}$

Discrétisation du domaine $\mathcal{D}$: température évaluée en un nombre fini de points. Le plus simple : grille de pas $dx = dy = p$

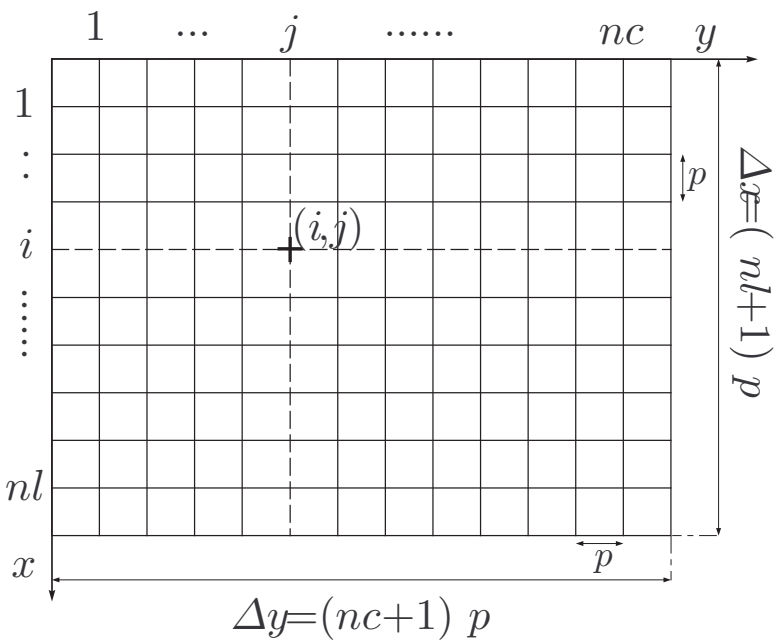


FIGURE 3 – Point de grille $\leftrightarrow (i, j)$. Intérieur du domaine : $i = 1 \rightarrow nl$, $j = 1 \rightarrow nc$.

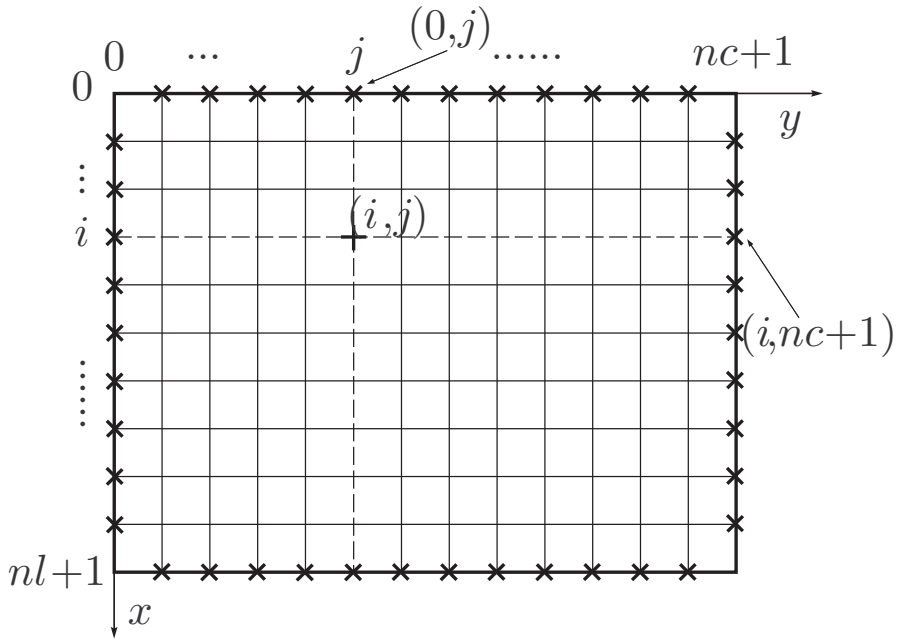


FIGURE 4 – Les bords du domaine (C.L.) sont numérotés : $i = 0$ pour le haut, $i = nl + 1$ pour le bas, $j = 0$ pour la gauche, $j = nc + 1$ pour la droite.

Indexation 2D du domaine

À l'intérieur de la grille, on a :

- $nl = \Delta x/p - 1$ lignes horizontales.
- $nc = \Delta y/p - 1$ colonnes verticales

Chaque point est repéré par un couple d'indices (i, j) où le premier indice note le numéro de la ligne et le second celui de la colonne.

On note $T_{i,j}$ la température en ce point, de coordonnées
$x_i = i \times p,\ y_j = j \times p.$

Les inconnues du problème sont les $n = nl \times nc$ valeurs de $T_{i,j}$
où $1 \leqslant i \leqslant nl,\ 1 \leqslant j \leqslant nc.$
Elles seront déduites des $2(nl + nc)$ valeurs de $T_{i,j}$ imposées aux bords

$i = 0$ (nord) ou $i = nl + 1$ (sud)
$j = 0$ (ouest) ou $j = nc + 1$ (est).

Différences finies

Les dérivées partielles sont approchées par des **différences finies**, basées sur des développements de Taylor. Le développement de Taylor,

$$T(x \pm p) = T(x) \pm p \frac{\partial T}{\partial x} + \frac{p^2}{2} \frac{\partial^2 T}{\partial x^2} \pm \frac{p^3}{3!} \frac{\partial^3 T}{\partial x^3} + O\left(p^4\right) \tag{6}$$

montre que :

$$T(x+p) + T(x-p) = 2T(x) + p^2 \frac{\partial^2 T}{\partial x^2}(x) + O\left(p^4\right) \tag{7}$$

d'où :

$$\boxed{\frac{\partial^2 T}{\partial x^2}(x) = \frac{T(x+p) + T(x-p) - 2\,T(x)}{p^2} + O\left(p^2\right)} \tag{8}$$

dont la précision est du deuxième ordre en p.

Différences finies (suite)

On obtient le même résultat en appliquant deux fois le schéma centré de pas $p/2$ à 2 termes de la dérivée :

$$\frac{\partial T}{\partial x}\left(x \pm \frac{p}{2}\right) \approx \pm \frac{T(x \pm p) - T(x)}{p} \tag{9}$$

Puis :

$$\frac{\partial^2 T}{\partial x^2}(x) \approx \frac{1}{p}\left[\frac{\partial T}{\partial x}\left(x + \frac{p}{2}\right) - \frac{\partial T}{\partial x}\left(x - \frac{p}{2}\right)\right] \tag{10}$$

qui redonne (8) :

$$\frac{\partial^2 T}{\partial x^2}(x) = \frac{T(x+p) + T(x-p) - 2\,T(x)}{p^2} + O\left(p^2\right)$$

Sur la grille, la dérivée seconde par rapport à x est alors approximée par :

$$\frac{\partial^2 T}{\partial x^2}(x_i, y_j) = \frac{T_{i+1,j} + T_{i-1,j} - 2\,T_{i,j}}{p^2} + O\left(p^2\right) \tag{11}$$

Différences finies : Laplacien 2D

On obtient ainsi le **Laplacien 2D discrétisé à l'ordre 2** :

$$\Delta T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4\,T_{i,j}}{p^2} + O\left(p^2\right) \tag{12}$$

Équation de Laplace $\Delta T = 0 \Longrightarrow$ système **linéaire** de N équations reliant $T_{i,j}$ aux 4 points voisins :

$$\boxed{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4\,T_{i,j} = 0} \tag{13}$$

$N = nc \times nl$ valeurs à déterminer
à partir de $2(nc + nl)$ températures connues.

Certains points intérieurs ont des voisins sur les parois avec des températures imposées qui vont constituer le second membre.
Noter qu'avec des conditions de Dirichlet, le pas p n'intervient plus.

Réindexation 1-D du domaine (en RowMajor)

Ré-indexation 1D des points intérieurs : $k = (i-1)nc + j \Rightarrow$ vecteur Z.

$$T_{i,j} \Longleftrightarrow \mathsf{Z}_k \quad 1 \leqslant k \leqslant nl \times nc$$

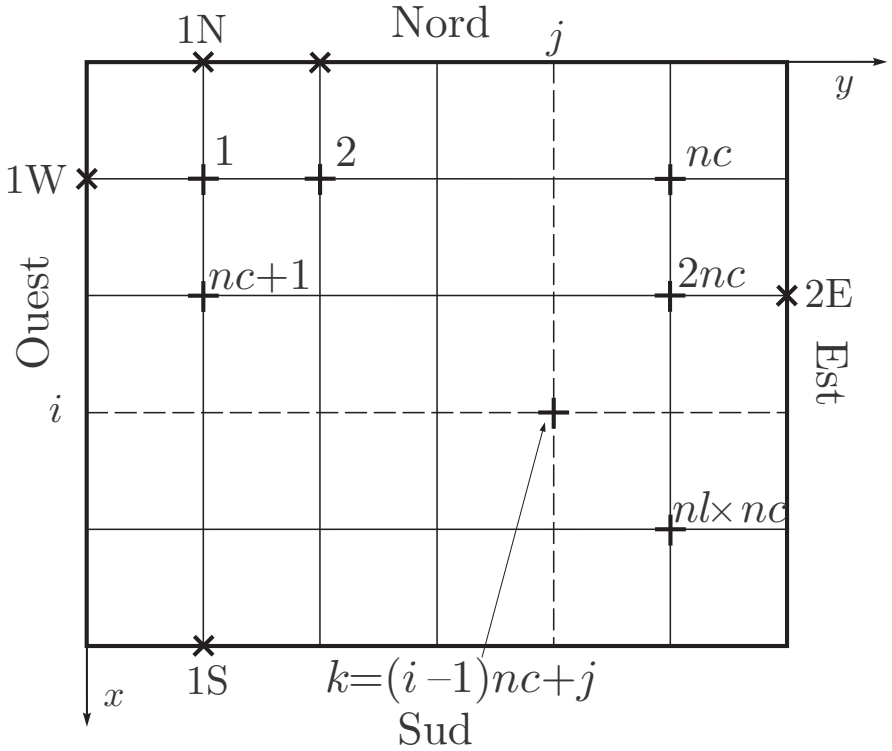


FIGURE 5 – Renumérotation des points de grille pour représenter la solution sous la forme d'un vecteur : le domaine est ré-indexé ligne par ligne de la gauche vers la droite, puis de haut en bas (comme une image numérique) donc en RowMajor.

Les voisins au Nord et au Sud sont alors espacés de nc dans Z.

Formulation vectorielle

Pour un point sans contact avec les bords du domaine, l'équation (13) s'écrit :

$$\boxed{\mathsf{Z}_{k-nc} + \mathsf{Z}_{k-1} - 4\mathsf{Z}_k + \mathsf{Z}_{k+1} + \mathsf{Z}_{k+nc} = 0} \tag{14}$$

⚠ Ne pas croire que le second membre du système linéaire est nul !

Pour un point en contact avec un bord, cette équation fait intervenir des températures imposées. Par exemple celles du bord Nord si $i = 1, j = 2$:

$$T_{2N} + \mathsf{Z}_1 - 4\mathsf{Z}_2 + \mathsf{Z}_3 + \mathsf{Z}_{2+nc} = 0$$

$\Longrightarrow$ passer ces termes dans le second membre du système

- Les CL de Dirichlet des bords créent le second membre
- Chaque changement de membre d'une température imposée annule un des coefficients des $\mathsf{Z}_{k'}$.

Matrice L_{2D}

Réécrire le système d'équations sous la forme :

$$A\ Z = B \tag{15}$$

où $A = L_{2D}$ est une matrice carrée $N \times N$ comportant $nl \times nl$ blocs carrés de dimension $nc \times nc$.

L_{2D} est symétrique tridiagonale par blocs :

1. blocs diagonaux avec -4 sur la diagonale, et 1 sur les sur-/sous-diagonale
2. blocs de type identité sur la sous-diagonale et la sur-diagonale.

Chaque ligne et chaque colonne contient 5 termes non-nuls,

- sauf celles de numéro $i \times nc$ et $i \times nc + 1$, où apparaissent les 0 des contacts avec les bords latéraux,
- et celles associées au premier et dernier bloc (bords haut et bas).

Matrice L_{2D} avec $nl = 3$ et $nc = 4$

$$L_{2D} = \left(\begin{array}{cccc|cccc|cccc}
-4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
1 & 0 & 0 & 0 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 0 & 0 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -4 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4
\end{array}\right)$$

$\longleftarrow nc \longrightarrow$ $\longleftarrow nc \longrightarrow$ $\longleftarrow nc \longrightarrow$

nl blocs de taille $nc \times nc$

$nl = 3$ et $nc = 4$

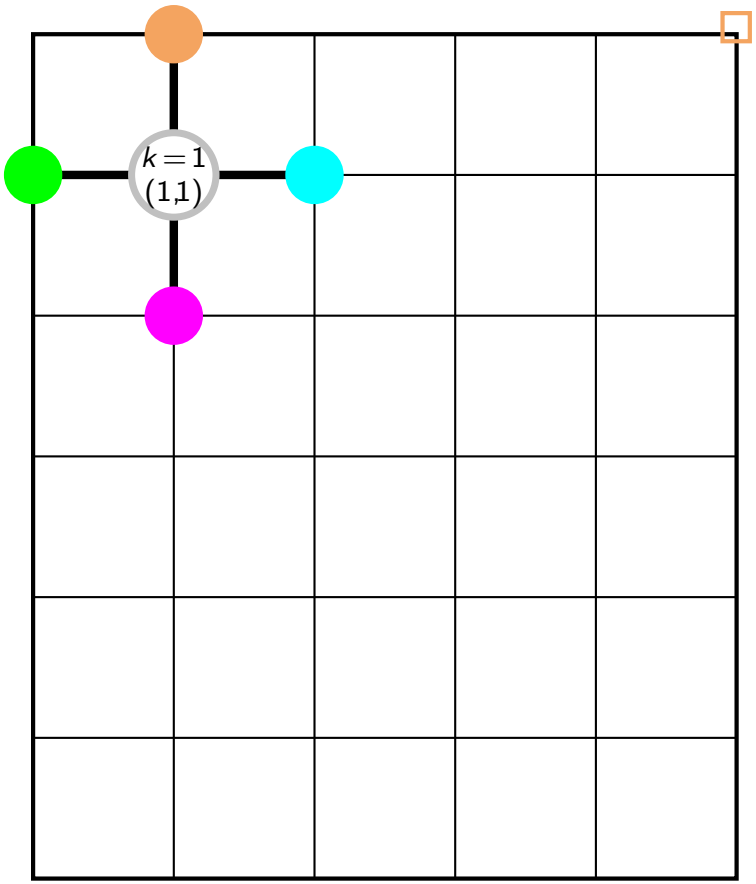
L_{2D} tridiagonale par blocs : exemple $nl = 5$ et $nc = 4$

$$L_{2D} = \left(\begin{array}{cccc|cccc|cccc|cccc|cccc}
-4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 0 & 0 & 0 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 0 & 0 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -4 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 \\
\hline
\multicolumn{4}{c}{\longleftarrow nc \longrightarrow} & \multicolumn{4}{c}{\longleftarrow nc \longrightarrow} & \multicolumn{4}{c}{\longleftarrow nc \longrightarrow} & \multicolumn{4}{c}{\longleftarrow nc \longrightarrow} & \multicolumn{4}{c}{\longleftarrow nc \longrightarrow}
\end{array}\right)$$

nl blocs de taille $nc \times nc$

$$nl = 5 \quad \text{et} \quad nc = 4$$

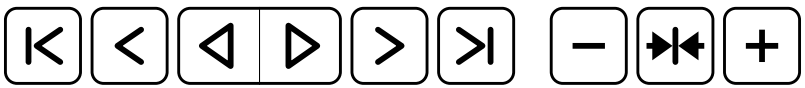
Sens de parcours et influence des bords ($nl = 5$ $nc = 4$)



-4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	-4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	-4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	-4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	-4	1	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	-4	1	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	-4	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	-4	1	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1	-4	1	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	1	-4	1	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	1	-4	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	-4	1	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	1	-4	1	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	1	-4	1	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	-4	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-4	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	-4	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	-4	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	-4

$\longleftarrow nc \longrightarrow$ $\longleftarrow nc \longrightarrow$ $\longleftarrow nc \longrightarrow$ $\longleftarrow nc \longrightarrow$ $\longleftarrow nc \longrightarrow$

nl blocs de taille $nc \times nc$



Le vecteur B résultant des CL

B est un vecteur creux à $N = nl \times nc$ composantes formé à partir des $2(nl + nc)$ températures aux bords. On peut le décomposer sous la forme :

$$
\mathsf{B} = -\begin{bmatrix} T_{1N} \\ T_{2N} \\ \vdots \\ T_{nc\,N} \end{bmatrix}
- \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}
- \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}
- \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}
\quad \text{Nord}
$$

$$
- \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}
- \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}
- \begin{bmatrix} T_{1S} \\ T_{2S} \\ \vdots \\ T_{nc\,S} \end{bmatrix}
\quad \text{Sud}
$$

$$
- \begin{bmatrix} T_{1W} \\ 0 \\ \vdots \\ 0 \end{bmatrix}
- \begin{bmatrix} T_{2W} \\ 0 \\ \vdots \\ 0 \end{bmatrix}
- \begin{bmatrix} T_{nl\,W} \\ 0 \\ \vdots \\ 0 \end{bmatrix}
\quad \text{Ouest}
$$

$$
- \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T_{1E} \end{bmatrix}
- \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T_{2E} \end{bmatrix}
- \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T_{nl\,E} \end{bmatrix}
\quad \text{Est}
$$

Solution du système linéaire

Plusieurs méthodes permettent de résoudre le système linéaire (15), $L_{2D}Z = B$, par exemple :

1. Les méthodes les plus générales de résolution numérique des systèmes linéaires, telles que la méthode d'élimination de Gauss-Jordan, la décomposition « LU » ou « QR » [1] ;
2. Des méthodes adaptées au cas où A est une matrice « creuse », bande, symétrique...
3. Des méthodes de relaxation qui s'apparentent, dans le cas d'une équation telle que l'équation de Laplace, à la résolution temporelle de l'équation de la chaleur.
4. L'utilisation des transformées de Fourier.

1. Ces méthodes permettent d'exprimier L_{2D} comme le produit de matrices triangulaire inférieure (**L**ower) et triangulaire supérieure (**U**pper) ou de matrices orthogonale (**Q**) et triangulaire supérieure (**R**).

Résolution du système linéaire à l'aide d'Eigen

Méthodes générales (Gauss-Jordan, LU, QR) $\propto N^3$ opérations $\Longrightarrow$ attention aux temps de calculs. Obligation d'utiliser des librairies (Eigen, Blas/LAPACK) dont les méthodes surpassent les méthodes naïves « à la main ».
Méthodes Eigen [2] :

- `mat.colPivHouseholderQr().solve(sm)` : décomposition QR d'une matrice quelconque et résolution du système linéaire pour un vecteur second membre `sm`
- `mat.ldlt().solve(sm)` : décomposition de Choleski (ou LDLT [3]) : L_{2D} définie négative $\Longrightarrow$ plus rapide, mais toujours en N^3
- caractère creux de L_{2D} (au plus 5 coefficients non nuls par colonne), permettant d'obtenir des méthodes $\propto N$. Utiliser les méthodes de Eigen/Sparse.

2. cf. doc Eigen Linear algebra and decompositions
3. factorisation sous la forme d'une matrice triangulaire inférieure (**L**ower), d'une matrice **D**iagonale et de la transposée de **L**

Sommaire

Initialisation avancées

Eigen définit plusieurs méthodes d'initialisation avancées [4], qui s'appliquent (notamment) sur des `MatrixXf` ou des `VectorXf` :

- Initialisation à 0 :
  ```
  MatrixXf mat = MatrixXf::Zero(n, m);
  ```
- Initialisation à une constante val :
  ```
  VectorXf vec = VectorXf::Constant(n, val);
  ```
- Initialisation à la matrice identité :
  ```
  MatrixXf matid = MatrixXf::Identity(n, n);
  ```

4. cf. doc Eigen Advanced initializations

Manipulation par blocs

Eigen permet de manipuler des vecteurs ou des matrices par blocs [5].
Pour les vecteurs :

- Accéder aux n premiers (resp. derniers) éléments
  `vec.head(n)` (resp. `vec.tail(n)`)
- Accéder aux n éléments à partir de la position i
  `vec.segment(i,n)`

Pour les matrices :

- Accéder aux colonnes (resp. lignes)
  `mat.col(j)` (resp. `mat.row(i)`)
- Accéder au bloc de taille (n, m) à partir de la position (i, j)
  `mat.block(i,j,n,m)`

N.B. : ces expressions peuvent être utilisées à droite (rvalue, lecture) ou à gauche (lvalue, affectation) du signe =

5. cf. doc Eigen Block Operations

Mesure du temps de calcul

Mesurer les performances d'une partie du code : par ex. temps de résolution de $\mathsf{L}_{2D}\ \mathsf{Z} = \mathsf{B}$ en fonction de la taille de L_{2D}.

```
#include <chrono>
using namespace std::chrono;
...
auto start = std::chrono::high_resolution_clock::now();
... // algorithme dont on souhaite mesurer la performance
auto stop = high_resolution_clock::now();
auto duration = duration_cast<milliseconds>(stop-start);
cout << duration << " ms" << endl;
```

Sommaire

Introduction aux problèmes dynamiques

Problèmes dynamiques :

- Évolution temporelle à partir d'un état d'équilibre
- Phénomènes de propagation

Caractéristiques :

- Conditions initiales
- Conditions aux limites (peuvent dépendre du temps)
  $\Longrightarrow$ système forcé : transitoire puis régime permanent $\nu \approx \nu_{\text{forcé}}$

Exemple : équation de la chaleur à une dimension spatiale pour simplifier.

Équation de diffusion 1-D

Équation de diffusion à 1-D avec $D = \text{cste} > 0$:

$$\boxed{\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}} \tag{16}$$

domaines spatial $x \in [0, L]$, et temporel $t \in [0, T]$

Conditions Initiales (CI) : $u(x, 0) = u_0(x) \quad \forall x$
Conditions aux Limites (CL) : $u(0, t) = u_g(t) \quad$ et $\quad u(L, t) = u_d(t) \quad \forall t$

N.B. : espace et temps ont un rôle dissymétrique :

- CL/CI
- Degré différent des dérivées partielles

Discrétisation du domaine $[0, L] \times [0, T]$

- ○ point intérieur
  u_j^n inconnu
- ◆ Cond. aux Limites
  en $x = 0$ et $x = L$
- ■ Cond. Initiale
  en $t = 0$ $(n = 0)$

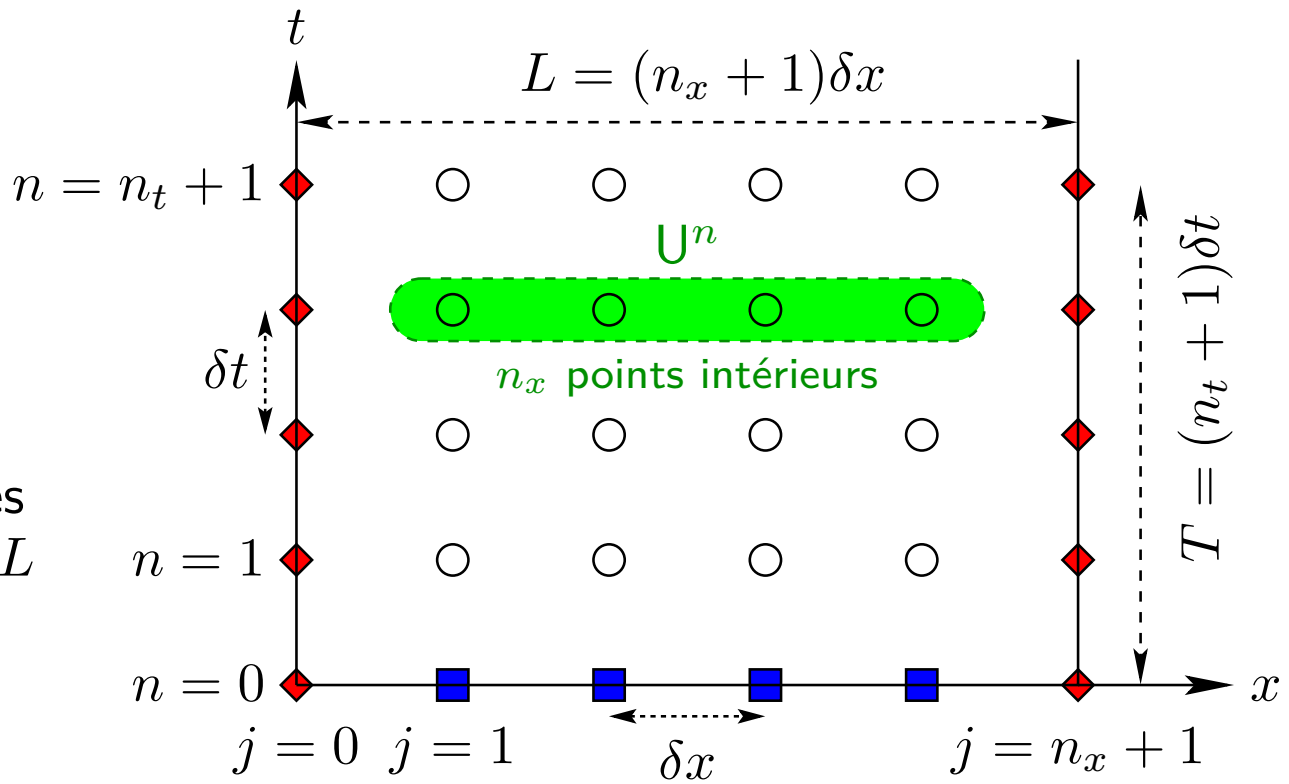


FIGURE 6 – Discrétisation du domaine $[0, L] \times [0, T]$

NB : visualisation ultérieure avec les 2 bords

$\Rightarrow$ à chaque pas de temps, écrire U^n (n_x points)

plus les CL (2 points) : $n_x + 2$ points

Discrétisation du domaine

Discrétiser les deux dimensions :

Espace : $[0, L]$ est divisé en $n_x + 1$ pas de largeur $\delta x = \frac{L}{n_x+1}$.
n_x points intérieurs + 2 bords

Temps : $[0, T]$ est subdivisé en $n_t + 1$ pas de largeur $\delta t = \frac{T}{n_t+1}$.

$u_j^n = u(x, t)$ pour $x = j\,\delta x$ et $t = n\,\delta t$

- CL $\Longrightarrow u_0^n \equiv u_g^n$ et $u_{n_x+1}^n \equiv u_d^n$ pour $1 \leqslant n \leqslant n_t + 1$
- CI $\Longrightarrow u_j^0$ pour $1 \leqslant j \leqslant n_x$.

Méthode de résolution fondamentalement itérative sur le temps :

$$u_j^n + \text{CL} \Longrightarrow u_j^{n+1} \text{ à l'intérieur du domaine.}$$

Discrétisation des opérateurs : différences finies

Laplacien 1D à l'ordre 2 (*cf* régimes stationnaires) :

$$\Delta u \approx \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\delta x)^2}$$

Dérivée par rapport au temps — au choix comme pour les EDO :

- vers l'avant : $\left(\frac{\partial u}{\partial t}\right)_j^n \approx \frac{u_j^{n+1} - u_j^n}{\delta t}$ (cf Euler forward)
- vers l'arrière : $\left(\frac{\partial u}{\partial t}\right)_j^n \approx \frac{u_j^n - u_j^{n-1}}{\delta t}$ (cf Euler backward)
- ou centrée : $\left(\frac{\partial u}{\partial t}\right)_j^n \approx \frac{u_j^{n+1} - u_j^{n-1}}{2\delta t}$

Ce choix détermine le caractère explicite ou implicite de la méthode.

Algorithme explicite du premier ordre en temps

Approche simpliste : dérivée vers l'avant

$$\frac{u_j^{n+1} - u_j^n}{\delta t} = D \left[\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\delta x)^2} \right] \quad \text{pour} \quad 1 \leqslant j \leqslant n_x \tag{17}$$

Semblable à méthode d'Euler explicite (progressive).

Paramètre caractéristique sans dimension :

$$\boxed{\alpha = \frac{D\,\delta t}{\delta x^2}} \tag{18}$$

Système d'équations linéaires (algorithme explicite)

On peut alors écrire :

$$u_j^{n+1} = \alpha\, u_{j-1}^n + (1-2\alpha)\, u_j^n + \alpha\, u_{j+1}^n \quad \text{pour} \quad 1 \leqslant j \leqslant n_x \tag{19}$$

Méthode explicite : u_j^{n+1} obtenus directement en fonctions des u_k^n.

Pour $j = 1$ ou n_x : u_{j-1}^n ou u_{j+1}^n sur les bords
$\Longrightarrow$ fixés par conditions aux limites

Schéma de progression explicite

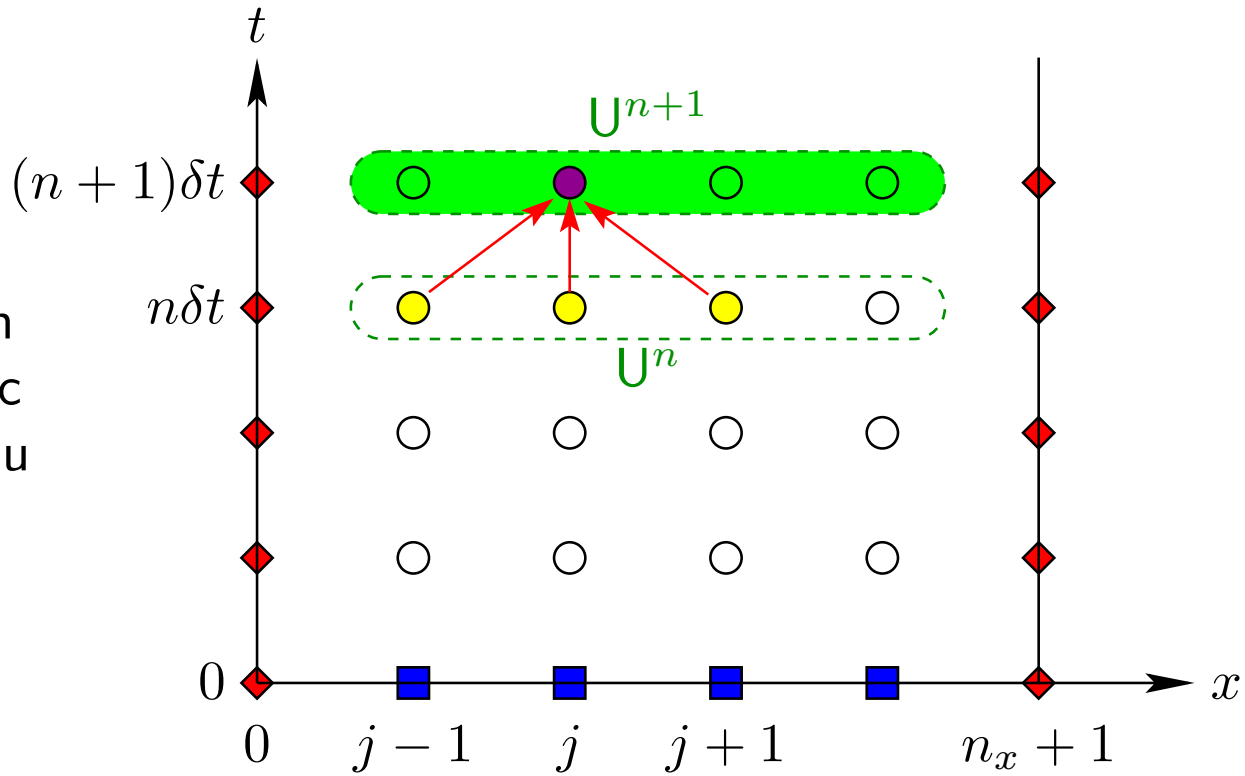


FIGURE 7 – Schéma illustrant la progression d'un pas de temps avec l'algorithme explicite du premier ordre en δt.

$$u_j^{n+1} = \alpha\, u_{j-1}^n + (1-2\alpha)\, u_j^n + \alpha\, u_{j+1}^n$$

Explicite : Formulation matricielle

Vecteur $\mathsf{U}^n = (u_1^n, \ldots, u_{n_x}^n)$ des solutions à $t = n\delta t$
pour les n_x points intérieurs
Le système d'équations linéaires prend la forme :

$$\mathsf{U}^{n+1} = \mathsf{M}(-\alpha)\, \mathsf{U}^n + \alpha \mathsf{V}^n \ ,$$

$\mathsf{V}^n = (u_g^n, 0, \ldots, 0, u_d^n)$: vecteur creux des 2 conditions aux limites,

$$\mathsf{M}(-\alpha) = \mathbb{1} + \alpha\, \mathsf{L}_{1D}$$

où L_{1D} = matrice tridiagonale représentant la dérivée seconde 1-D

$$\mathsf{L}_{1D} = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix} \quad \longrightarrow \quad \mathsf{M}(-\alpha) = \begin{pmatrix} 1{-}2\alpha & \alpha & 0 & \cdots & 0 \\ \alpha & 1{-}2\alpha & \alpha & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1{-}2\alpha & \alpha \\ 0 & \cdots & 0 & \alpha & 1{-}2\alpha \end{pmatrix} \tag{20}$$

Explicite : stabilité

Problème important : instabilité de la méthode explicite pour un pas de temps « un peu trop grand » ($\alpha > 0.5$) :

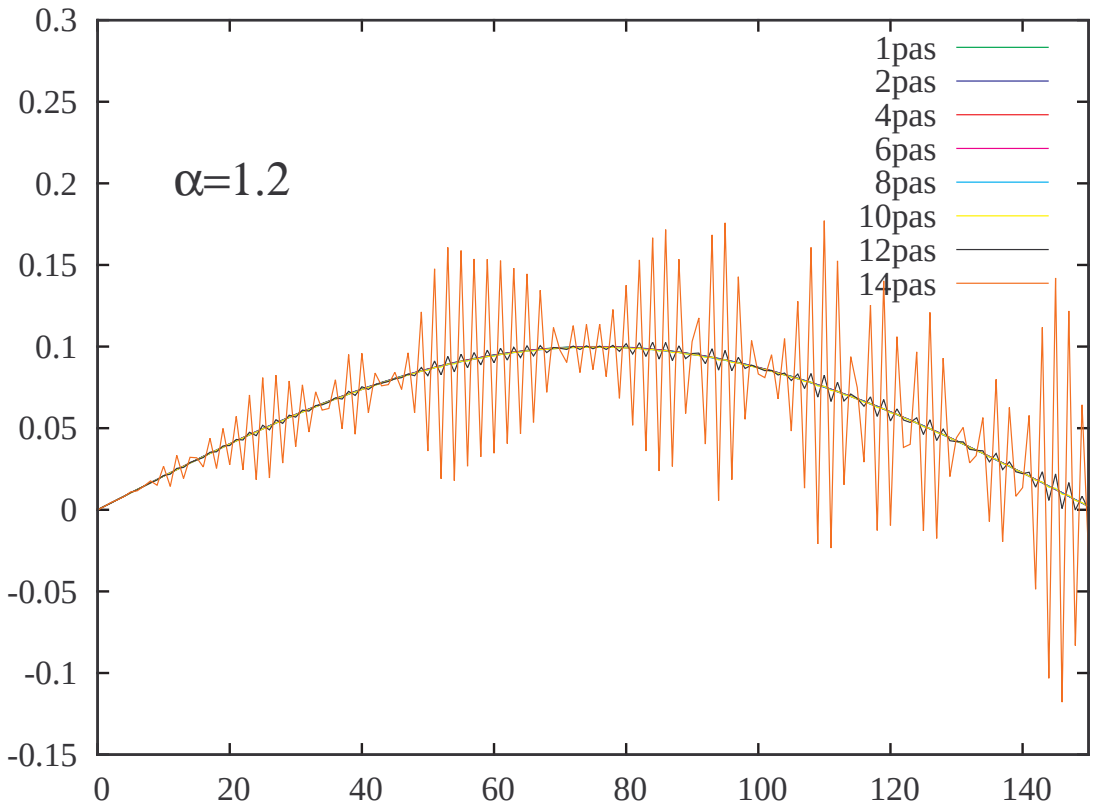


FIGURE 8 – Résolution explicite avec $\alpha = 1.2 > 0.5$.

Explicite : analyse de stabilité de Fourier–von Neumann

Stabilité des modes propres :

$$u(x,t) = A(t)\exp(ikx) \quad \Rightarrow \quad u_j^n = A^{(n)} e^{ik\,j\,\delta x}\ , \tag{21}$$

solution si $\mathrm{d}A/\,\mathrm{d}t = -Dk^2 A(t)$ soit $A(t) = A(0)\exp\left(-Dk^2 t\right)$

L'algorithme explicite donne la version discrète de $A(t)$:

$$\begin{aligned} \frac{A^{(n+1)}}{A^{(n)}} &= 1 + \alpha\left(e^{ik\,\delta x} - 2 + e^{-ik\,\delta x}\right) \\ &= 1 + \alpha\left(e^{ik\,\delta x/2} - e^{-ik\,\delta x/2}\right)^2 \\ &= 1 - 4\alpha\sin^2\left(\frac{k\,\delta x}{2}\right)\ . \end{aligned} \tag{22}$$

$\rightarrow$ suite géométrique $A^{(n)} = A^{(0)}\theta^n$, de raison réelle $\theta = 1 - 4\alpha\sin^2\left(\frac{k\,\delta x}{2}\right)$

L'algorithme explicite est stable si $|\theta| \leqslant 1$ pour tous les modes, soit

$$\boxed{0 \leqslant \alpha \leqslant 1/2} \quad \text{c-à-d} \quad \delta t \leqslant \delta x^2/2D$$

Exemple : Diffusion d'une distribution en « marche »

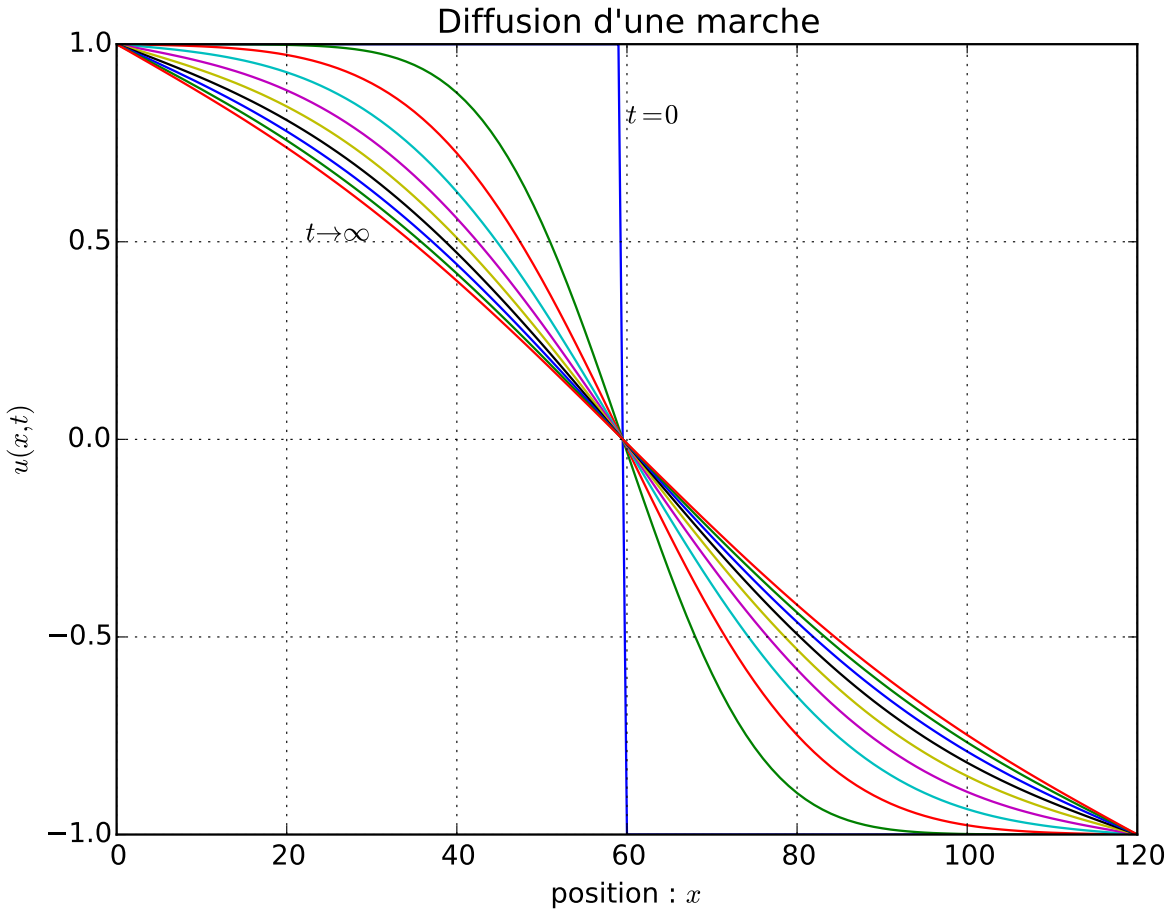


FIGURE 9 – Diffusion d'une marche avec des CL fixes $+1$ à gauche et -1 à droite : $u(x,t)$ tend vers une droite si $t \to \infty$

Exemple : Diffusion d'une superposition de 2 modes

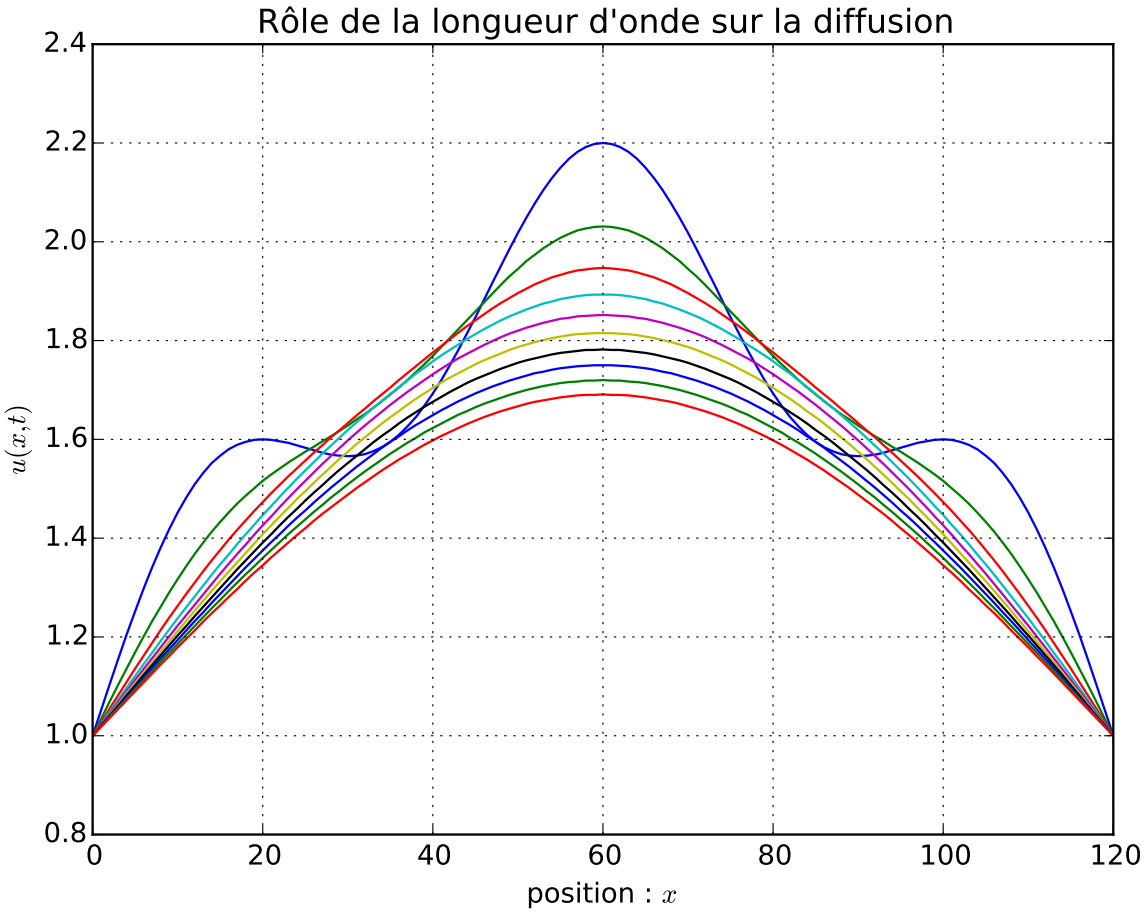


FIGURE 10 – Amortissement du mode 5 plus rapide que celui du mode 1 mais pour $t \to \infty$, on obtiendrait une droite.

Solution au problème de stabilité : méthode rétrograde

Algorithme explicite (progressif) simple, mais instable sauf très petits pas

Stabiliser (comme Euler en EDO) en passant en implicite (rétrograde).

Algorithme implicite : dérivée temporelle « vers l'arrière ».

$$\frac{u_j^{n+1} - u_j^n}{\delta t} = D \left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\delta x)^2} \right] \tag{23}$$

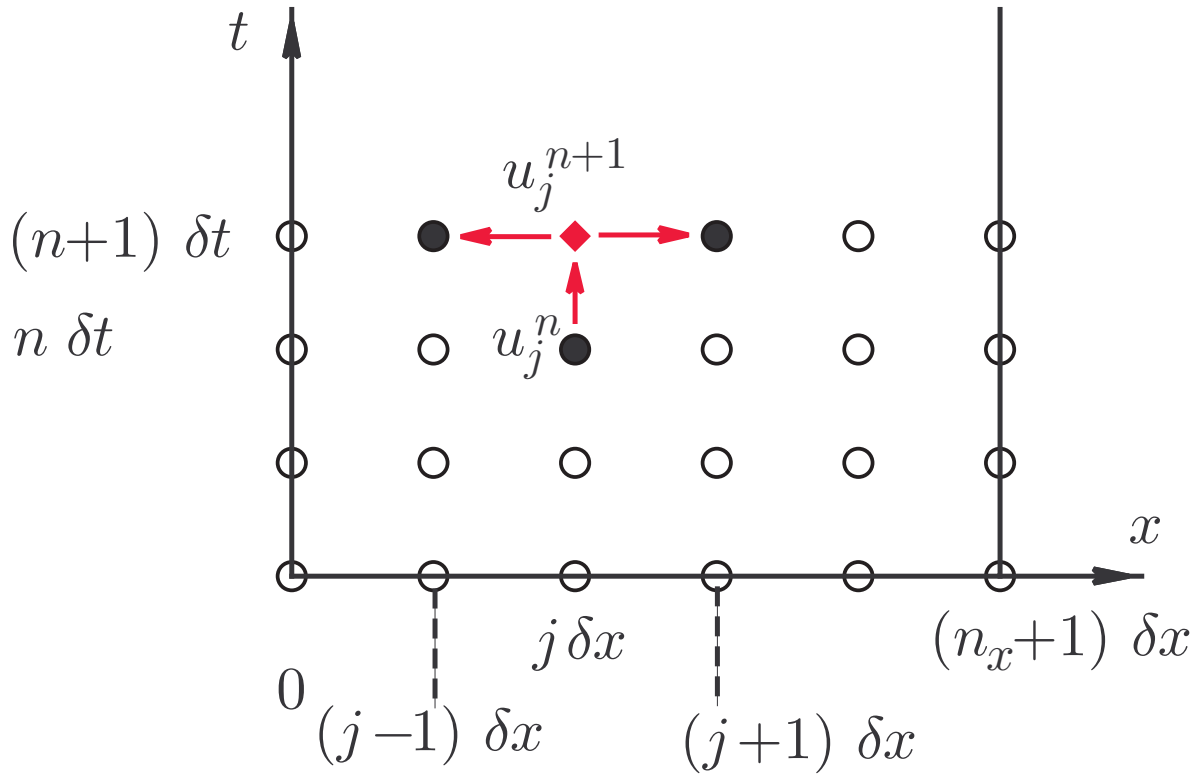
$\Longrightarrow$ système de n_x équations linéaires à inverser :

$$\boxed{-\alpha\, u_{j-1}^{n+1} + (1+2\alpha)\, u_j^{n+1} - \alpha\, u_{j+1}^{n+1} = u_j^n \quad \text{pour} \quad j \in [1, n_x]} \tag{24}$$

sauf pour $j = 1$ et $j = n_x$ où interviennent les CL.

Schéma de progression implicite

FIGURE 11 – Schéma illustrant la progression d'un pas de temps avec l'algorithme implicite du premier ordre en δt.



$$-\alpha\, u_{j-1}^{n+1} + (1+2\alpha)\, u_j^{n+1} - \alpha\, u_{j+1}^{n+1} = u_j^n$$

Formulation matricielle

De façon matricielle : $\mathsf{M}(\alpha)\,\mathsf{U}^{n+1} = \mathsf{U}^n + \alpha\,\mathsf{V}^{n+1}$

$$\mathsf{M}(\alpha) = \mathbb{1} - \alpha\,\mathsf{L}_{1D} = \begin{pmatrix} 1{+}2\alpha & -\alpha & 0 & \cdots & 0 \\ -\alpha & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha \\ 0 & \cdots & 0 & -\alpha & 1{+}2\alpha \end{pmatrix} \tag{25}$$

M tridiagonale mais M^{-1} matrice dense $\Longrightarrow u_j^{n+1}$ dépend de tous les autres éléments, mais d'autant moins lorsqu'ils sont éloignés.
Lorsque $\alpha \to 0$, $\mathsf{M}(\alpha)^{-1} \to \mathsf{M}(-\alpha)$: on retrouve l'algorithme explicite.

Stabilité de l'algorithme implicite

Analyse Fourier – Von Neumann : modes propres $\propto A(t)\sin(kx)$.
Suite géométrique de raison :

$$\theta = \frac{1}{1 + 4\,\alpha \sin^2\left(\frac{k\,\delta x}{2}\right)} \leqslant 1$$

On obtient plus simplement ce résultat à partir des valeurs propres de $\mathsf{M}(\alpha)^{-1} = (\mathbb{1} - \alpha \mathsf{L}_{1D})^{-1}$, inverses de $1 + 4\alpha \sin^2\left(\frac{p\,\pi\delta x}{2L}\right)$.

Algorithme implicite inconditionnellement stable quelque soit $\delta x, \delta t$.

Toutefois, approximation du premier ordre en δt : description précise de la solution transitoire nécessite un pas de temps petit.

Principe de l'algorithme de Crank–Nicolson

Objectif : accroître la précision $\Longrightarrow$ améliorer l'approximation en différences finies de $\frac{\partial}{\partial t}$.

Même stratégie que « Euler modifié » (cf. EDO) : moyenne des deux algorithmes.

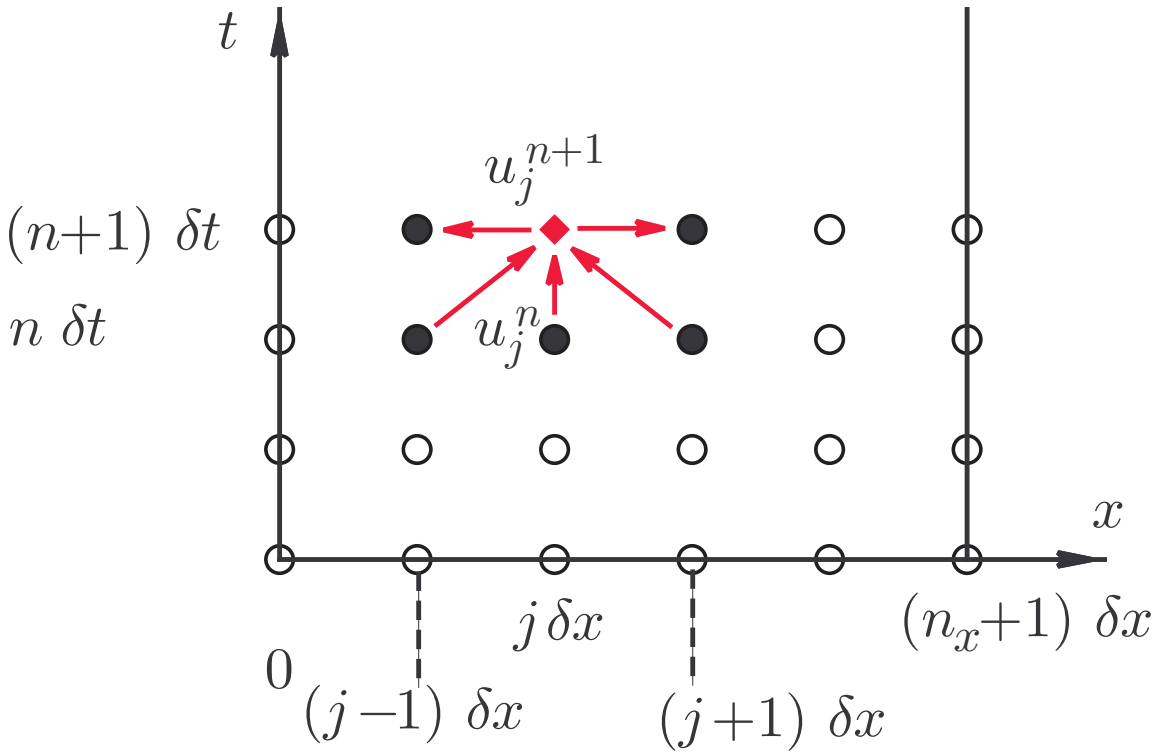
$$\frac{u_j^{n+1} - u_j^n}{\delta t} = \frac{D}{2} \left[\frac{(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) + (u_{j+1}^n - 2u_j^n + u_{j-1}^n)}{(\delta x)^2} \right]$$

NB : différence finie temporelle inchangée, mais vue comme schéma centré à l'instant milieu $(n + \frac{1}{2})\delta t$, grâce à l'évaluation de la moyenne des laplaciens discrets aux deux instants.

Algorithme de Crank–Nicolson du deuxième ordre en δt et δx.

Schéma de progression de Crank–Nicolson

FIGURE 12 – Schéma illustrant la progression d'un pas de temps avec l'algorithme de Crank–Nicolson.



$$-\frac{\alpha}{2}u_{j-1}^{n+1} + (1+\alpha)u_j^{n+1} - \frac{\alpha}{2}u_{j+1}^{n+1} = \frac{\alpha}{2}u_{j-1}^{n} + (1-\alpha)u_j^{n} + \frac{\alpha}{2}u_{j+1}^{n}$$

Stabilité de l'algorithme de Crank–Nicolson

Etude de stabilité Von Neumann : décroissance géométrique des modes :

$$\theta = \frac{1 - 2\,\alpha \sin^2\left(\frac{k\,\delta x}{2}\right)}{1 + 2\,\alpha \sin^2\left(\frac{k\,\delta x}{2}\right)} \tag{26}$$

ce qui permet de montrer que cet algorithme est lui aussi inconditionnellement stable (*i.e.* quel que soit le choix de δt).

$$\theta_{\mathsf{CN}}(\alpha) = \theta_{\mathsf{explicite}}(\tfrac{\alpha}{2}) \times \theta_{\mathsf{implicite}}(\tfrac{\alpha}{2})$$

Formulation matricielle

Système d'équations linéaires à résoudre ($j \in [1, n_x]$) :

$$-\frac{\alpha}{2}u_{j-1}^{n+1} + (1+\alpha)u_j^{n+1} - \frac{\alpha}{2}u_{j+1}^{n+1} = \frac{\alpha}{2}u_{j-1}^{n} + (1-\alpha)u_j^{n} + \frac{\alpha}{2}u_{j+1}^{n} \tag{27}$$

sauf pour $j = 1$ et $j = n_x$ où interviennent les CL.

De façon matricielle :

$$\mathsf{M}\left(\frac{\alpha}{2}\right)\ \mathsf{U}^{n+1} = \mathsf{M}\left(\frac{-\alpha}{2}\right)\ \mathsf{U}^{n} + \frac{\alpha}{2}\left[\mathsf{V}^{n} + \mathsf{V}^{n+1}\right] \tag{28}$$

où $\mathsf{M}(\alpha) = \mathbb{1} - \alpha\,\mathsf{L}_{1D}$