

Thème 4: résolution numérique des EDP linéaires

Problèmes statiques et dynamiques

Université Pierre et Marie CURIE
MNCS
Méthodes numériques et calcul scientifique

2 et 3 mars 2017

1 Introduction

- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

1 Introduction

- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique linéaire : l'équation de Laplace

- Discrétisation du problème
- Reformulation vectorielle de la solution
- Solution du problème

1 Introduction

- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique linéaire : l'équation de Laplace

- Discrétisation du problème
- Reformulation vectorielle de la solution
- Solution du problème

3 Problèmes dynamiques

- Discrétisation de l'espace et du temps
- Algorithme explicite du premier ordre en temps
- Algorithme implicite du premier ordre en temps
- Algorithme de Crank–Nicolson

1 Introduction

- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique linéaire : l'équation de Laplace

- Discrétisation du problème
- Reformulation vectorielle de la solution
- Solution du problème

3 Problèmes dynamiques

- Discrétisation de l'espace et du temps
- Algorithme explicite du premier ordre en temps
- Algorithme implicite du premier ordre en temps
- Algorithme de Crank–Nicolson

4 Équation des ondes

Sommaire

1 Introduction

- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique linéaire : l'équation de Laplace

- Discrétisation du problème
- Reformulation vectorielle de la solution
- Solution du problème

3 Problèmes dynamiques

- Discrétisation de l'espace et du temps
- Algorithme explicite du premier ordre en temps
- Algorithme implicite du premier ordre en temps
- Algorithme de Crank–Nicolson

4 Équation des ondes

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 »

$$\frac{\partial^2 u}{\partial t^2} - c^2 \Delta u = 0 \quad (1)$$

- ② L'équation de Schrödinger

$$i \frac{\partial u}{\partial t} = -\frac{1}{2} \Delta u + V u \quad (2)$$

- ③ L'équation de diffusion

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

- ④ Les équations de Poisson et de Laplace

$$\Delta u = \rho \quad \Delta u = 0 \quad (4)$$

⚠ 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 :

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

- $\beta^2 - 4\alpha\gamma > 0$ 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

→ méthodes numériques spécifiques.

Conditions aux limites

L'équation est vérifiée dans un domaine \mathcal{D} de l'espace.

Problème « bien posé » \rightarrow conditions aux limites (CL).

Plusieurs types de conditions :

- **Dirichlet** : u imposé sur le bord $\partial\mathcal{D}$.
- **Neumann** : $\overrightarrow{\text{grad}} u \cdot \vec{n}$ (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 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** : 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 statiques

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

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

Équation de Laplace

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

Problèmes dynamiques

2 - Problèmes dynamiques :

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

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} \quad (5)$$

avec diverses conditions initiales.

Sommaire

1 Introduction

- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique linéaire : l'équation de Laplace

- Discrétisation du problème
- Reformulation vectorielle de la solution
- Solution du problème

3 Problèmes dynamiques

- Discrétisation de l'espace et du temps
- Algorithme explicite du premier ordre en temps
- Algorithme implicite du premier ordre en temps
- Algorithme de Crank–Nicolson

4 Équation des ondes

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 .

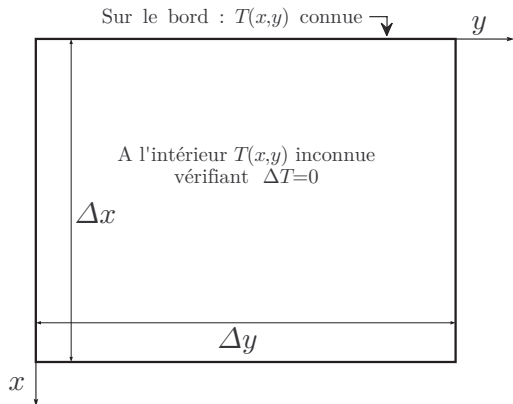


FIGURE 1 – Géométrie du problème de l'équation de Laplace : noter le choix inhabituel de l'orientation des axes x et y .

Exemple de carte d'isothermes

CL Dirichlet : températures imposées aux bords

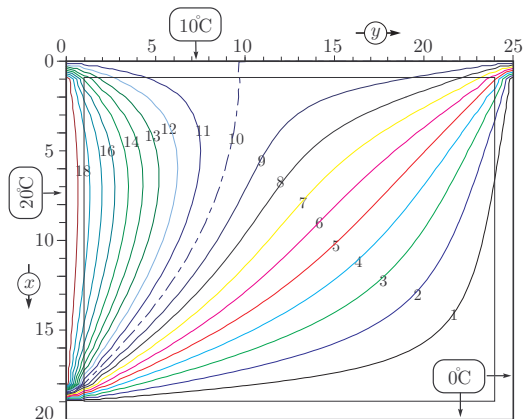


FIGURE 2 – Exemple de solution de l'équation $\Delta T = 0$, dans le cas où une paroi est maintenue à 20°C , une autre adjacente est à 10°C et les deux autres parois sont à $T = 0^{\circ}\text{C}$.

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 $dx = dy = p$

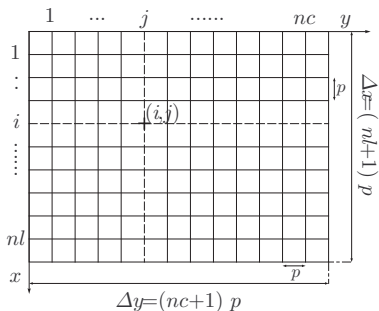


FIGURE 3 – Point de grille $\leftrightarrow (i, j)$.
Intérieur du domaine : $i = 1 \rightarrow nl$,
 $j = 1 \rightarrow nc$.

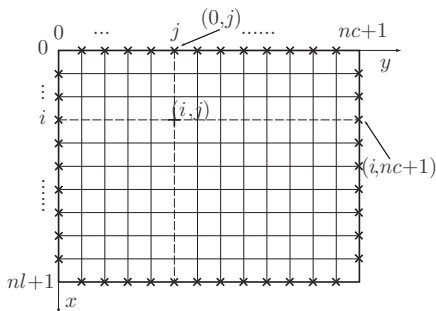


FIGURE 4 – Les **bords** du domaine (C.L.) sont numérotés : $i = 0$ pour le haut, $i = nl + 1$ pour le bas, $j = 0$ pour la gauche, $j = nc + 1$ pour la droite.

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 $1 \leq i \leq nl$, $1 \leq j \leq nc$.

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 :

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

montre que :

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

d'où :

$$\frac{\partial^2 T}{\partial x^2}(x) = \frac{T(x + p) + T(x - p) - 2T(x)}{p^2} + O(p^2) \quad (8)$$

ce qui conduit à :

$$\frac{\partial^2 T}{\partial x^2}(x_i, y_j) = \frac{T_{i+1,j} + T_{i-1,j} - 2T_{i,j}}{p^2} + O(p^2) \quad (9)$$

dont la précision est du deuxième ordre en p .

Différences finies (suite)

Obtention alternative :

$$\frac{\partial T}{\partial x} \left(x \pm \frac{p}{2} \right) \approx \pm \frac{T(x \pm p) - T(x)}{p} \quad (10)$$

qui conduit à :

$$\frac{\partial^2 T}{\partial x^2} (x_i, y_j) \approx \frac{1}{p} \left[\frac{\partial T}{\partial x} \left(x_i + \frac{p}{2}, y_j \right) - \frac{\partial T}{\partial x} \left(x_i - \frac{p}{2}, y_j \right) \right] \quad (11)$$

soit encore :

$$\frac{\partial^2 T}{\partial x^2} (x_i, y_j) = \frac{T_{i+1,j} + T_{i-1,j} - 2T_{i,j}}{p^2} + O(p^2) \quad (12)$$

Différences finies : Laplacien 2D

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

$$\Delta T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j}}{p^2} + O(p^2) \quad (13)$$

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

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = 0 \quad (14)$$

$N = nc \times nl$ 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 que 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 .

$$T_{i,j} \iff Z_k \quad 1 \leq k \leq nl \times nc$$

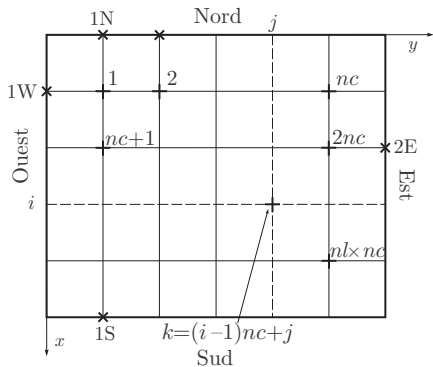


FIGURE 5 – Renumérotation des points de grille pour représenter la solution sous la forme d'un vecteur : le domaine est ré-indexé ligne par ligne de la gauche vers la droite, puis de haut en bas (comme une image numérique) donc en **RowMajor**.

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 (14) s'écrit :

$$Z_{k-nc} + Z_{k-1} - 4Z_k + Z_{k+1} + Z_{k+nc} = 0 \quad (15)$$

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 $i = 1$.

$$T_{2N} + Z_1 - 4Z_2 + Z_3 + Z_{2+nc} = 0 \quad \text{pour } j = 2$$

⇒ 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 $Z_{k'}$.

Matrice L_{2D}

Réécrire le système d'équations sous la forme :

$$A Z = B \quad (16)$$

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 de type L_{1D} avec -4 au lieu de -2 sur la 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).

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

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 :

$$B = \begin{array}{c} \text{Nord} \\ \begin{bmatrix} T_{1N} \\ T_{2N} \\ \vdots \\ T_{nc N} \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \\ \text{Sud} \\ \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ T_{1S} \\ T_{2S} \\ \vdots \\ T_{nc S} \end{bmatrix} \\ \text{Ouest} \\ \begin{bmatrix} T_{1W} \\ 0 \\ \vdots \\ 0 \\ T_{2W} \\ 0 \\ \vdots \\ 0 \\ T_{nl W} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ \text{Est} \\ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T_{1E} \\ 0 \\ \vdots \\ 0 \\ T_{2E} \\ 0 \\ \vdots \\ 0 \\ T_{nl E} \end{bmatrix} \end{array}$$

Solution du système linéaire

Plusieurs méthodes permettent de résoudre le système linéaire (16), $L_{2D}Z = B$, 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 »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. La factorisation « LU » exprime la matrice comme produit d'une matrice triangulaire inférieure (**L**ower) par une matrice triangulaire supérieure (**U**pper).

Résolution du système par méthode de relaxation

Méthode de relaxation : méthode **itérative**, basée sur la décomposition en parties diagonale/non-diagonale : $L_{2D} = D + N$
 $L_{2D}Z = B$ équivaut à $DZ = -(NZ - B)$.

Solution = **point fixe** de l'application $Z \mapsto -D^{-1}(NZ - B)$

Calcul par **récurrance** en partant d'une valeur quelconque Z^0 :

$$Z^{n+1} = -D^{-1}(NZ^n - B) = (NZ^n - B)/4$$

Convergence : $Z^{n+1} - Z = \frac{N}{4}(Z^n - Z)$ avec les valeurs absolues des valeurs propres de N toutes inférieures à 4 (rayon spectral de $N < 4$).

Méthode aisée à coder et bien adaptée aux géométries complexes.

Résolution du système linéaire à l'aide de Lapack

Méthodes générales (Gauss-Jordan, LU) $\propto N^3$ opérations.

Ici : au plus 5 coefficients non nuls par ligne

\implies exploitation du caractère «creux» de la matrice : N opérations.

utilisation de Blas/Lapack :

- Matrices générales : xGESV
- $-L_{2D}$ définie positive : xPOSV
- Symétrique : xSYSV
- Matrice bande : xSBSV (amélioration significative)
- Toutes ces propriétés : xPBSV

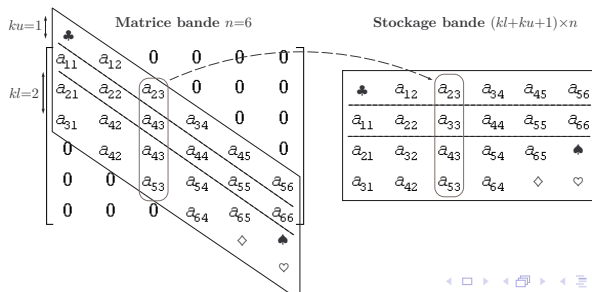
Stockage bande

Fonctions pour matrices bandes \implies stockage spécifique de la matrice : «stockage bande».

Ne fournir que les sur- et sous-diagonales significatives à la procédure.

Deux possibilités :

- Écriture directe de la matrice contenant les diagonales
- Écriture de la matrice complète puis « conversion ».
(en C : procédure `matfull2band` fournie)



Sommaire

- 1 Introduction
 - Généralités
 - Conditions aux limites
 - Problèmes statiques et dynamiques
- 2 Exemple de problème statique linéaire : l'équation de Laplace
 - Discrétisation du problème
 - Reformulation vectorielle de la solution
 - Solution du problème
- 3 Problèmes dynamiques
 - Discrétisation de l'espace et du temps
 - Algorithme explicite du premier ordre en temps
 - Algorithme implicite du premier ordre en temps
 - Algorithme de Crank–Nicolson
- 4 Équation des ondes

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)
⇒ 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 $D = \text{cste} > 0$:

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} \quad (17)$$

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) : $u(0, t) = u_g(t) \quad \text{et} \quad u(L, t) = u_d(t) \quad \forall t$

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

- CL/CI
- Degré différent des dérivées partielles

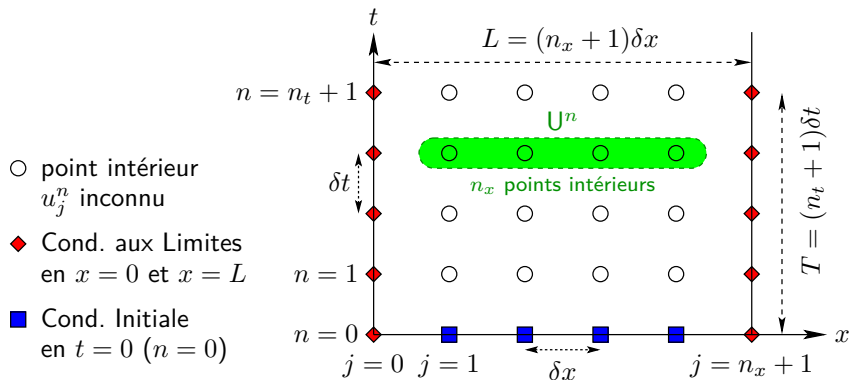
Discrétisation du domaine $[0, L] \times [0, T]$ 

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

NB : visualisation ultérieure avec les 2 bords
 \Rightarrow à chaque 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 $\implies u_0^n \equiv u_g^n$ et $u_{n_x+1}^n \equiv u_d^n$ pour $1 \leq n \leq n_t + 1$
- CI $\implies u_j^0$ pour $1 \leq j \leq n_x$.

Méthode de résolution fondamentalement **itérative** sur le temps :

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

Discrétisation des opérateurs : différences finies

Laplacien 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 ordre 1 par rapport au temps — au choix comme pour les EDO :

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

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

Algorithme explicite du premier ordre en temps

Approche simpliste : dérivée vers l'avant

$$\frac{u_j^{n+1} - u_j^n}{\delta t} = D \left[\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\delta x)^2} \right] \quad \text{pour } 1 \leq j \leq n_x \quad (18)$$

Semblable à méthode d'Euler explicite (progressive).

Paramètre caractéristique sans dimension :

$$\alpha = \frac{D \delta t}{\delta x^2} \quad (19)$$

Système d'équations linéaires (algorithme explicite)

On peut alors écrire :

$$u_j^{n+1} = \alpha u_{j+1}^n + (1 - 2\alpha) u_j^n + \alpha u_{j-1}^n \quad \text{pour } 1 \leq j \leq n_x \quad (20)$$

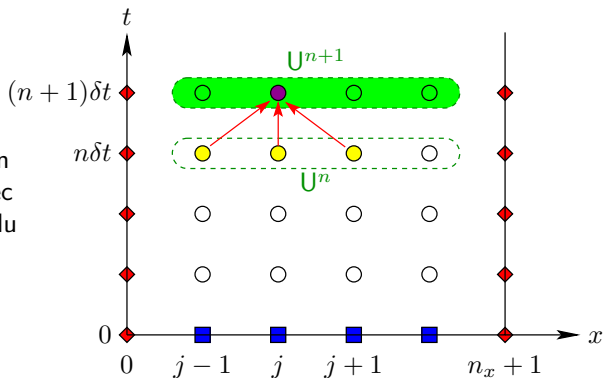
Méthode **explicite** : u_j^{n+1} obtenus directement en fonctions des u_k^n .

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

Note : équivalent à méthode de relaxation si $\alpha = 1/2$.

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 $\mathbf{U}^n = (u_1^n, \dots, u_{n_x}^n)$ des solutions à $t = n\delta t$
pour les n_x **points intérieurs**

Le système d'équations linéaires prend la forme :

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

$\mathbf{V}^n = (u_g^n, 0, \dots, 0, u_d^n)$: vecteur des conditions aux limites,

$$\mathbf{M}(-\alpha) = \mathbf{1} + \alpha \mathbf{L}_{1D}$$

où \mathbf{L}_{1D} = matrice tridiagonale représentant la dérivée seconde 1-D

$$\mathbf{L}_{1D} = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix} \rightarrow \mathbf{M}(-\alpha) = \begin{pmatrix} 1-2\alpha & \alpha & 0 & \cdots & 0 \\ \alpha & 1-2\alpha & \alpha & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1-2\alpha & \alpha \\ 0 & \cdots & 0 & \alpha & 1-2\alpha \end{pmatrix} \quad (21)$$

Explicite : stabilité

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

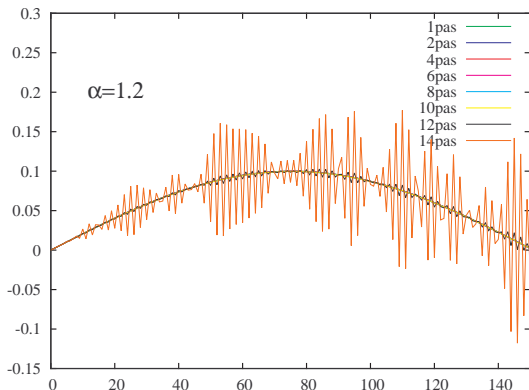


FIGURE 8 – Résolution explicite avec $\alpha = 1.2 > 0.5$.

Explicite : analyse de stabilité de Fourier–von Neumann

Stabilité des modes propres :

$$u(x, t) = A(t) \exp(ikx) \quad \Rightarrow \quad u_j^n = A^{(n)} e^{ikj \delta x}, \quad (22)$$

solution si $dA/dt = -Dk^2 A(t)$ soit $A(t) = A(0) \exp(-Dk^2 t)$

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

$$\begin{aligned} \frac{A^{(n+1)}}{A^{(n)}} &= 1 + \alpha \left(e^{ik \delta x} - 2 + e^{-ik \delta x} \right) \\ &= 1 + \alpha \left(e^{ik \delta x/2} - e^{-ik \delta x/2} \right)^2 \\ &= 1 - 4\alpha \sin^2 \left(\frac{k \delta x}{2} \right). \end{aligned} \quad (23)$$

→ 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 $|\theta| \leq 1$ pour tous les modes, soit

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

Stabilité de l'explicite : lien avec le spectre de L_{1D}

La p -ième valeur propre de la matrice L_{1D} , en dimension n , s'écrit en fonction de $\beta = \pi/2(n+1) = \pi\delta x/2L$:

valeur propre : $-4 \sin^2(p\beta)$; vecteur propre : $u_m^{(p)} = \sin(m p \beta)$

On en déduit celles de la matrice $M(-\alpha)$:

$$M(-\alpha) = \mathbf{1} + \alpha L_{1D} \quad \longrightarrow \quad 1 - 4\alpha \sin^2\left(\frac{p\pi\delta x}{2L}\right).$$

Stabilité \iff valeurs propres de M de module inférieur à 1 : $\rho(M) < 1$.

La relation de dispersion, $i\omega = Dk^2$ déduite de l'équation de la diffusion est remplacée par :

$$2i \sin\left(\frac{\omega\delta t}{2}\right) \exp\left(i\frac{\omega\delta t}{2}\right) = 4\alpha \sin^2\left(\frac{k\delta x}{2}\right) \quad (24)$$

qui n'a de solution ω à partie imaginaire négative (stable) que si $4\alpha \leq 2$.

Avant d'aller plus loin dans l'analyse numérique, voici quelques exemples.

Exemple : Diffusion d'une distribution en « marche »

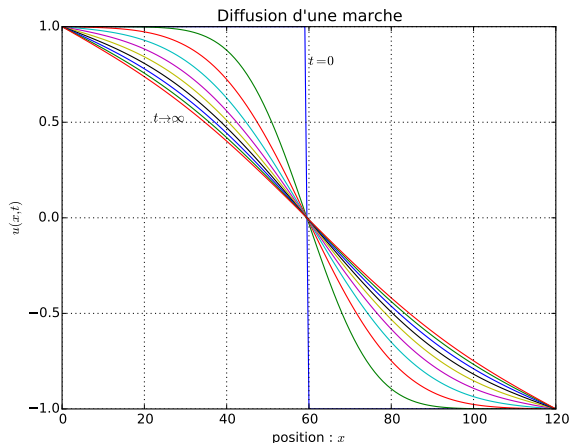


FIGURE 9 – Diffusion d'une marche avec des CL fixes +1 à gauche et -1 à droite : $u(x,t)$ tend vers une droite si $t \rightarrow \infty$

Exemple : Diffusion d'une superposition de 2 modes

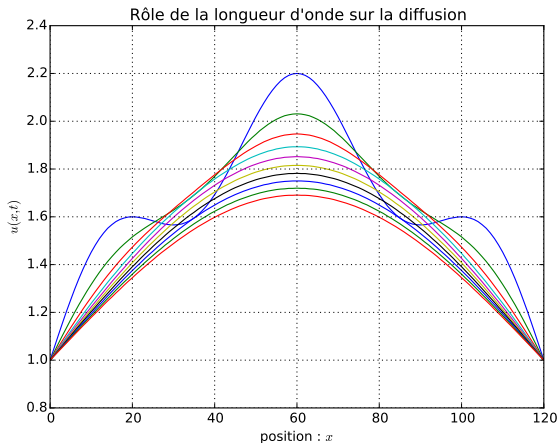


FIGURE 10 – Amortissement du mode 5 plus rapide que celui du mode 1 mais pour $t \rightarrow \infty$, on obtiendrait une droite.

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 ».

$$\frac{u_j^{n+1} - u_j^n}{\delta t} = D \left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\delta x)^2} \right] \quad (25)$$

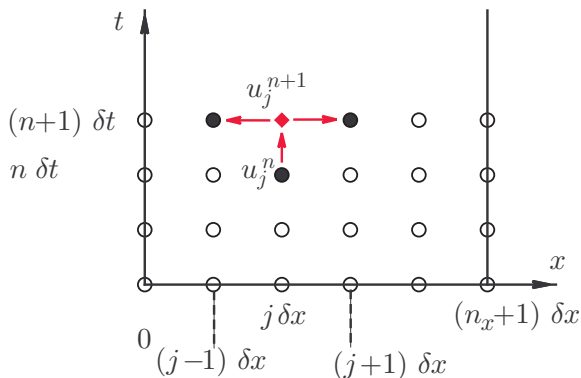
⇒ système de n_x équations linéaires à **inverser** :

$$-\alpha u_{j-1}^{n+1} + (1 + 2\alpha) u_j^{n+1} - \alpha u_{j+1}^{n+1} = u_j^n \quad \text{pour } j \in [1, n_x] \quad (26)$$

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 : $M(\alpha) U^{n+1} = U^n + \alpha V^{n+1}$

$$M(\alpha) = \mathbb{1} - \alpha L_{1D} = \begin{pmatrix} 1+2\alpha & -\alpha & 0 & \cdots & 0 \\ -\alpha & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha \\ 0 & \cdots & 0 & -\alpha & 1+2\alpha \end{pmatrix} \quad (27)$$

M tridiagonale mais M^{-1} **matrice dense** $\implies u_j^{n+1}$ dépend de tous les autres éléments, mais d'autant moins lorsqu'ils sont éloignés.

Lorsque $\alpha \rightarrow 0$, $M(\alpha)^{-1} \rightarrow M(-\alpha)$: 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 :

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

On obtient plus simplement ce résultat à partir des valeurs propres de $M(\alpha)^{-1} = (\mathbf{1} - \alpha L_{1D})^{-1}$, 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 \implies améliorer l'approximation en différences finies de $\frac{\partial}{\partial t}$.

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

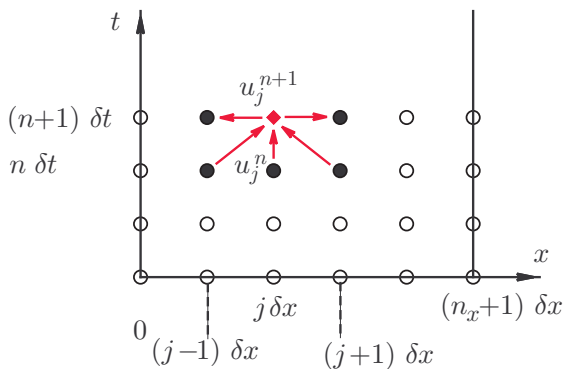
$$\frac{u_j^{n+1} - u_j^n}{\delta t} = \frac{D}{2} \left[\frac{(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) + (u_{j+1}^n - 2u_j^n + u_{j-1}^n)}{(\delta x)^2} \right]$$

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 :

$$\theta = \frac{1 - 2 \alpha \sin^2 \left(\frac{k \delta x}{2} \right)}{1 + 2 \alpha \sin^2 \left(\frac{k \delta x}{2} \right)} \quad (28)$$

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

$$\theta_{\text{CN}}(\alpha) = \theta_{\text{explicite}}\left(\frac{\alpha}{2}\right) \times \theta_{\text{implicite}}\left(\frac{\alpha}{2}\right)$$

Formulation matricielle

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

$$-\frac{\alpha}{2}u_{j-1}^{n+1} + (1 + \alpha)u_j^{n+1} - \frac{\alpha}{2}u_{j+1}^{n+1} = \frac{\alpha}{2}u_{j-1}^n + (1 - \alpha)u_j^n + \frac{\alpha}{2}u_{j+1}^n \quad (29)$$

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

De façon matricielle :

$$\mathbf{M} \left(\frac{\alpha}{2} \right) \mathbf{U}^{n+1} = \mathbf{M} \left(\frac{-\alpha}{2} \right) \mathbf{U}^n + \frac{\alpha}{2} [\mathbf{V}^n + \mathbf{V}^{n+1}] \quad (30)$$

où $\mathbf{M}(\alpha) = \mathbf{1} - \alpha \mathbf{L}_{1D}$

Sommaire

- 1 Introduction
 - Généralités
 - Conditions aux limites
 - Problèmes statiques et dynamiques
- 2 Exemple de problème statique linéaire : l'équation de Laplace
 - Discrétisation du problème
 - Reformulation vectorielle de la solution
 - Solution du problème
- 3 Problèmes dynamiques
 - Discrétisation de l'espace et du temps
 - Algorithme explicite du premier ordre en temps
 - Algorithme implicite du premier ordre en temps
 - Algorithme de Crank–Nicolson
- 4 Équation des ondes

Équation des ondes : différences finies

Résolution numérique de l'équation de d'Alembert :

- Discrétisation du domaine comme pour l'équation de la chaleur ;
- CI arbitraires $u(x, 0)$;
- CL Dirichlet ou Neumann.
- schéma de différences finies «naturel» :

$$\boxed{\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}} \quad \Rightarrow \quad \frac{u_j^{n+1} + u_j^{n-1} - 2u_j^n}{(\delta t)^2} = c^2 \frac{u_{j+1}^n + u_{j-1}^n - 2u_j^n}{(\delta x)^2} \quad (31)$$

(deuxième ordre en temps et espace).

Note : symétrie entre temps et espace.

Algorithme explicite

Poser $\alpha = c \frac{\delta t}{\delta x} \Rightarrow$ équations **explicites** à trois niveaux de temps :

$$u_j^{n+1} = \alpha^2(u_{j+1}^n + u_{j-1}^n) + 2(1 - \alpha^2)u_j^n - u_j^{n-1}. \quad (32)$$

soit, sous forme matricielle :

$$U^{n+1} = (2\mathbf{1} + \alpha^2 L_{1D}) U^n - U^{n-1} + \alpha^2 L_{1D} V^n$$

où V représente à nouveau les conditions aux limites sur les bords.

Ce schéma est appelé “**leap-frog**” en anglais, singulièrement traduit en « saute-mouton ».

Ordre deux en temps \Rightarrow 2 CI : $u(x, 0)$ et $\partial u / \partial t(x, 0)$,
ou 2 temps : $u(x, 0)$ et $u(x, \delta t)$.

Critère CFL

Relation de dispersion associée au schéma : $\sin^2\left(\frac{\omega \delta t}{2}\right) = \alpha^2 \sin^2\left(\frac{k \delta x}{2}\right)$

$\alpha \leq 1$: solutions réelles (ie non divergentes).

$\alpha = 1$: relation de dispersion exacte (et w_j^n disparaît).

⇒ Critère de **Courant–Friedrichs–Lewy** (CFL) :
 $\alpha \leq 1 \iff$ méthode stable.

Valeur précise dépend de la méthode de discrétisation, dimension de l'espace, éventuellement direction de propagation.

3 formulations équivalentes :

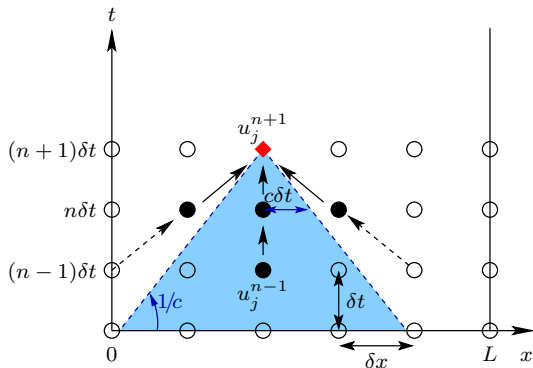
- pas de temps $\delta t < \delta x/c$ temps de propagation de l'onde ;
- pas d'espace $\delta x > c \delta t$ distance parcourue par l'onde ;
- vitesse «diffusion information» $\delta x/\delta t > c$ célérité de l'onde.

Schéma de progression et critère CFL

Le schéma explicite « saute-mouton » est désormais le suivant :

FIGURE 13 – Progression d'un pas de temps dans l'algorithme explicite du second ordre en temps et espace pour l'équation des ondes de d'Alembert

Critère CFL : $c\delta t < \delta x$



Le critère CFL se traduit (dans ce cas simple 1D) par le fait que des points intervenant dans la progression doivent être **hors** du cône de lumière, pour laisser à l'onde « le temps de se propager ».