

Thème 5: résolution numérique des équations aux dérivées partielles(EDP) Problèmes dynamiques

Université Pierre et Marie CURIE
Méthodes numériques et informatiques 2

29 mars 2011

Sommaire

1 Introduction

- Exemples d'EDP
- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

Sommaire

1 Introduction

- Exemples d'EDP
- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique : résolution numérique de l'équation de Laplace

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

Sommaire

1 Introduction

- Exemples d'EDP
- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique : résolution numérique de 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–Nicholson

Sommaire

1 Introduction

- Exemples d'EDP
- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique : résolution numérique de 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–Nicholson

4 Équation des ondes

Sommaire

1 Introduction

- Exemples d'EDP
- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique : résolution numérique de 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–Nicholson

4 Équation des ondes

Contexte

Les équations aux dérivées partielles (EDP) interviennent dans la description de très nombreux problèmes de physique, chimie, sciences de la Terre, biologie : mécanique des fluides, propagation des ondes, électromagnétisme, phénomènes de diffusion ...

Nous nous limitons dans tout ce qui suit à des EDP relatives à un champ scalaire $u(\vec{r}, t)$. Les généralisations vectorielles sont bien sûr possibles, et techniquement analogues.

Les exemples d'EDP scalaires les plus connues :

- ① L'équation des ondes
- ② L'équation de Schrödinger
- ③ L'équation de diffusion
- ④ Les équations de Poisson et de Laplace

Équation des ondes

L'équation des ondes ou « de d'Alembert » :

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

où $u = u(\vec{r}, t)$ est un champ scalaire fonction de la position \vec{r} et du temps t , comme la pression d'un gaz dans le cas du son.

La constante c la vitesse de propagation des ondes et Δ est le Laplacien scalaire :

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad \text{en 2 dimensions} \quad (2)$$

Équation de Schrödinger

L'équation de Schrödinger :

$$i \frac{\partial u}{\partial t} = -\frac{1}{2} \Delta u + V u \quad (3)$$

qui décrit en mécanique quantique la fonction d'onde d'une particule massive dans un potentiel V .

Elle représente aussi la propagation d'une onde harmonique à la limite paraxiale, dans un milieu de permittivité $\epsilon = n^2 = V + 1$.

Équation de la diffusion

L'équation de diffusion

$$\frac{\partial u}{\partial t} + \operatorname{div} \vec{J} = 0 \quad \text{avec} \quad \vec{J} = -D \overrightarrow{\operatorname{grad}}(u)$$

Ici \vec{J} est le flux (de u) par unité de surface et D le coefficient de diffusion, ce qui donne, dans le cas où D est homogène :

$$\frac{\partial u}{\partial t} = D \Delta u \tag{4}$$

Cette équation décrit :

- les transferts de chaleur par conduction dans un milieu continu, elle est alors appelée *équation de la chaleur* et $u \equiv T$ et $D \equiv \lambda/C_p$.
- la diffusion des particules d'un fluide dans un autre (équation de Fick).

Équations de Poisson et de Laplace

Des cas particuliers sont les équations de Poisson et de Laplace, qui peuvent être vues comme des cas limites lorsque la dépendance en temps est supprimée, ou remplacée par une dépendance harmonique.

L'équation de Poisson

$$\Delta u = \rho$$

est en électrostatique celle qui régit le potentiel $\Phi(\vec{r})$ créé par la densité de charges $\rho(\vec{r})$.

L'équation de Laplace est un cas particulier où la charge est nulle :

$$\Delta u = 0 \tag{5}$$

On va notamment la retrouver comme solution stationnaire de l'équation de la chaleur.

Équations elliptiques/paraboliques/hyperboliques

L'étude théorique des EDP est un vaste domaine en mathématiques qui est hors du cadre de notre étude.

Les EDP linéaires évoquées plus haut sont classées en EDP elliptiques/paraboliques/hyperboliques selon la forme des coefficients figurant devant les dérivées partielles :

- Équation hyperbolique : équation de d'Alembert 1-D
- Équation parabolique : diffusion ou Schrödinger
- Équations elliptiques : équations de Laplace, Poisson, Helmholtz

Elles ont des propriétés mathématiques différentes, et leur résolution approchée par des techniques numériques requièrent aussi parfois des méthodes distinctes.

Dans la pratique, nous nous concentrerons sur une méthode de résolution approchée par des méthodes numériques, dans le cas principalement de l'équation de la chaleur.

Conditions aux limites

On doit se donner des « conditions aux limites » (CL). L'équation étant vérifiée dans un domaine \mathcal{D} de l'espace (ou espace-temps), on distingue des conditions de 2 types :

- **Conditions de Dirichlet** : on impose la valeur de u sur la bordure de \mathcal{D} . Dans le cas où on étudie un problème dépendant du temps, cela inclut des conditions « initiales »
- **Conditions de Neumann** : c'est la valeur de la dérivée normale $\partial u / \partial n = \overrightarrow{\text{grad}} u \cdot \vec{n}$ que l'on impose.
- **Dans le cas général**, on a une combinaison des deux selon différentes parties de la bordures.

À titre d'exemple pour l'équation de la chaleur :

- Des CL de Dirichlet correspondent à des parois isothermes, qui imposent leur température.
- Des CL de Neumann avec $\overrightarrow{\text{grad}} u \cdot \vec{n} = 0$ annulent le transfert de chaleur : parois adiabatiques.

Problèmes statiques et dynamiques

Du point de vue de leur résolution numérique, deux classes de problèmes doivent être distingués :

- **Problèmes dynamiques** : il s'agit de déterminer $u(\vec{r}, t)$ dans un domaine \mathcal{D} , sur un certain intervalle de temps, en fonction des CL spatio-temporelles :
$$\text{CI} : u(\vec{r}, t = 0) = \dots + \text{CL} : u(\vec{r} = \cdot, t) = \dots$$
- **Problèmes statiques** : le temps n'intervient pas, il s'agit de déterminer $u(\vec{r})$ en fonctions des seules CL ; cela correspond en général à un régime stationnaire atteint après un temps d'évolution assez long.

Régime permanent

Nous nous intéresserons en premier lieu aux **problèmes statiques**.

Il s'agit de problèmes où l'on cherche la solution d'une EDP où le temps n'intervient pas. C'est par exemple la solution d'équilibre de l'équation de diffusion qui sera asymptotiquement atteinte quand $t \rightarrow \infty$, et qui doit donc vérifier l'équation de Laplace.

Dans ce cas u représente un champ de température $T(x, y)$ (avec une origine arbitraire) et on a à 2-D :

$$\Delta T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 . \quad (6)$$

Problèmes dynamiques

Puis, dans une deuxième partie, nous considérerons les **problèmes dynamiques**.

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

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} \quad (7)$$

avec diverses conditions initiales.

Sommaire

1 Introduction

- Exemples d'EDP
- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique : résolution numérique de 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–Nicholson

4 Équation des ondes

Résolution de l'équation de Laplace

Nous cherchons une solution numérique approchée de $\Delta T = 0$ à l'intérieur d'un domaine \mathcal{D} que pour simplifier nous supposons rectangulaire, de côtés Δx et Δy (voir figure 1).

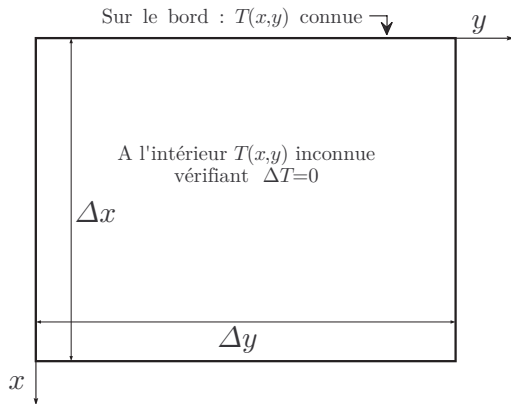


FIGURE 1: Géométrie du problème de l'équation de Laplace : noter le choix inhabituel de l'orientation des axes x et y .

Exemple de carte d'isothermes

Nous supposons en outre que les parois ont des températures imposées de l'extérieur (CL de Dirichlet).

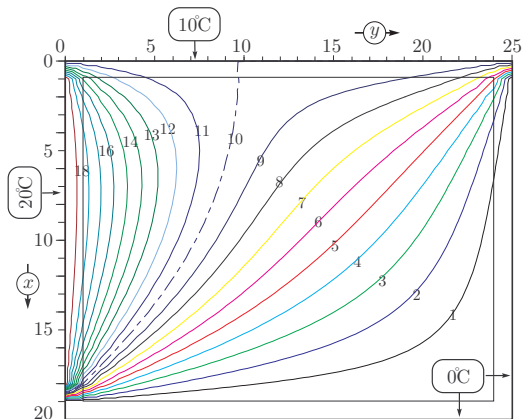


FIGURE 2: Exemple de solution dans le cas où une paroi est maintenue à 20°C , une autre adjacente est à 10°C et les deux autres parois sont à $T = 0^{\circ}\text{C}$. Les lignes indexées de 1 à 18 représentent les isothermes.

Discrétisation du domaine

Nous procédons à une discrétisation du domaine \mathcal{D} : la température sera calculée en un nombre fini de points. Pour simplifier, le maillage choisi est une grille de pas $dx = dy = p$ comme représenté sur les figures 3 et 4.

On a $nl = \Delta x/p - 1$ lignes et $nc = \Delta y/p - 1$ colonnes.

Maillage du domaine \mathcal{D}

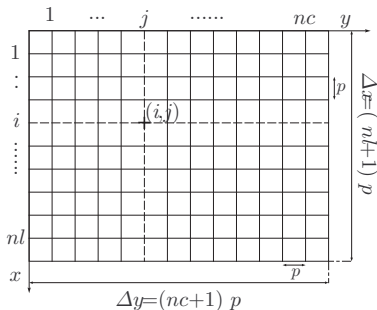


FIGURE 3: Chaque point de la grille est repéré par un couple (i, j) où i repère la ligne et j la colonne. L'intérieur du domaine est numéroté de $i = 1$ à nl et de $j = 1$ à nc .

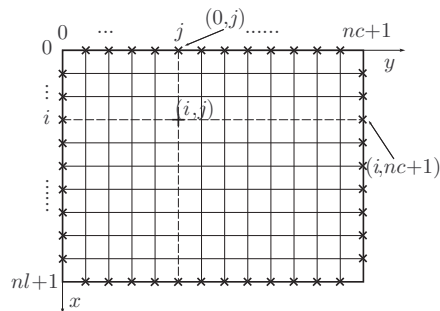


FIGURE 4: Les bords du domaine portant les conditions aux limites sont numérotés : $i = 0$ pour le haut, $i = nl + 1$ pour le bas, $j = 0$ pour la gauche, $j = nc + 1$ pour la droite.

Indexation du domaine

On a à l'intérieur de la grille :

- $nl = \Delta x / p - 1$ 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 $1 \leq i \leq nl, 1 \leq j \leq nc$ qui seront déduites des valeurs de $T_{i,j}$ aux bords (pour $i = 0$ ou $i = nl + 1, j = 0$ ou $j = nc + 1$).

Différences finies

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

$$T(x \pm p, y) = T(x, y) \pm p \frac{\partial T}{\partial x} + \frac{p^2}{2} \frac{\partial^2 T}{\partial x^2} \pm \frac{p^3}{3!} \frac{\partial^3 T}{\partial x^3} + O(p^4) \quad (8)$$

montre que :

$$T(x + p, y) + T(x - p, y) = 2T(x, y) + p^2 \frac{\partial^2 T}{\partial x^2}(x, y) + O(p^4) \quad (9)$$

d'où :

$$\frac{\partial^2 T}{\partial x^2}(x, y) = \frac{T(x + p, y) + T(x - p, y) - 2T(x, y)}{p^2} + O(p^2) \quad (10)$$

ce qui conduit à :

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

dont la précision est du deuxième ordre en p .

Différences finies (suite)

On note que cette approximation peut s'obtenir de façon alternative à partir de :

$$\frac{\partial T}{\partial x}(x \pm \frac{p}{2}, y) \approx \pm \frac{T(x \pm p, y) - T(x, y)}{p} \quad (12)$$

qui conduit à :

$$\begin{aligned} \frac{\partial^2 T}{\partial x^2}(x_i, y_j) &\approx \frac{1}{p} \left[\frac{\partial T}{\partial x} \left(x_i + \frac{p}{2}, y_j \right) - \frac{\partial T}{\partial x} \left(x_i - \frac{p}{2}, y_j \right) \right] \\ &= \frac{T_{i+1,j} + T_{i-1,j} - 2 T_{i,j}}{p^2} + O(p^2) \end{aligned} \quad (13)$$

Différences finies (suite)

On obtient ainsi le Laplacien discrétisé :

$$\Delta T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4 T_{i,j}}{p^2} + O(p^2) \quad (14)$$

L'équation de Laplace se traduit alors par un système linéaire de N équations reliant les valeurs de $T_{i,j}$ au 4 points voisins :

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4 T_{i,j} = 0 \quad (15)$$

où $2 \times (nl + nc)$ températures sont connues (celles sur les points aux bords indiqués par des croix sur la figure 4) et $N = nl \times nc$ sont à déterminer.

Réindexation 1-D (matrice aplatie)

On ré-indexe des points à l'intérieur de la grille par un indice k : (cf fig. 5)

$$k = (i - 1)nc + j \quad (16)$$

qui fait correspondre au tableau 2-D des $T_{i,j}$ à calculer, le vecteur à $N = nl \times nc$ composantes Z_k .

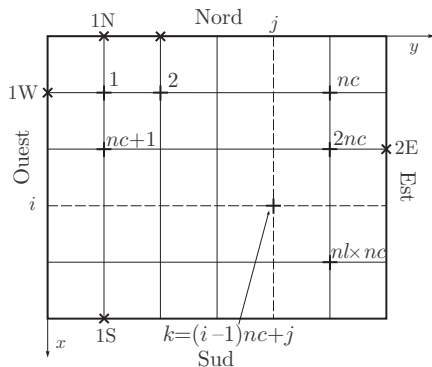


FIGURE 5: Renumérotation des points de grille pour représenter la solution sous la forme d'un vecteur : le domaine est ré-indexé ligne par ligne de la gauche vers la droite et de bas en haut.

Formulation vectorielle

Pour un point intérieur au domaine, l'équation (15) s'écrit :

$$Z_{k-nc} + Z_{k-1} - 4Z_k + Z_{k+1} + Z_{k+nc} = 0 \quad (17)$$

En l'absence de bords ce système serait un système homogène, conduisant à une température uniforme.

Avec des bords de température différentes, certaines des équations (17) font apparaître un terme de bord, comme celle du Nord par exemple si $i = 1$. En plaçant à droite du signe égal les températures imposées sur les bords, ce système d'équations prend alors la forme :

$$A Z = B \quad (18)$$

où A est une matrice carrée $N \times N$ comportant $nl \times nl$ blocs carrés de dimension $nc \times nc$, du type :

Matrice L_{2D}

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

nl blocs
de taille
 $nc \times nc$

$\leftarrow nc \rightarrow \quad \leftarrow nc \rightarrow \quad \leftarrow nc \rightarrow$

Chaque ligne et chaque colonne contient 5 termes non-nuls comme (17) le prévoit, sauf celles de numéro nc et $nc - 1$, où apparaissent les $\color{red}{0}$.

Sens de parcours et influence des bords

Le vecteur B résultant des CL

B est un vecteur à N composantes formé à partir des températures aux bords. On peut le décomposer sous la forme :

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

alias $b_{k=(i-1)nc+j} = -(\delta_{i,1} T_{i-1,j} + \delta_{i,nc} T_{i+1,j} + \delta_{j,1} T_{i,j-1} + \delta_{j,nl} T_{i,j+1}) \quad (19)$

Solution du système linéaire

Plusieurs méthodes permettent de résoudre le système linéaire (18), 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 ».
- 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 de 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 linéaire

Les méthodes générales, comme Gauss-Jordan ou LU demandent un nombre d'opérations en N^3 pour résoudre un système de N équations².

Dans le cas présent, avec au plus 5 coefficients non nuls par ligne, on devrait exploiter le caractère « creux » de la matrice, et descendre à un nombre d'opérations proportionnel à N .

En TE, nous pourrons utiliser la bibliothèque Blas/Lapack optimisée, et on pourra donc explorer des régimes de tailles relativement intéressants, de l'ordre de 1000 à 10000 points.

On pourra commencer avec la fonction Lapack xGESV pour les matrices générales, et comparer ensuite à ce que l'on obtient avec la procédure dédiée aux matrices symétriques xSYSV ou aux matrices bandes xSBSV. Notons que cette dernière requiert un stockage spécifique, par diagonales.

2. Pour une matrice symétrique positive la factorisation de Choleski permet une résolution en $\sim N^2$

Sommaire

1 Introduction

- Exemples d'EDP
- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique : résolution numérique de 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–Nicholson

4 Équation des ondes

Introduction aux problèmes dynamiques

Comme indiqué dans l'introduction générale, les problèmes dynamiques sont ceux où l'on suit l'évolution temporelle à partir d'un état « hors d'équilibre ».

Il sera donc caractérisé par des « conditions initiales », et des « conditions aux limites » susceptibles de dépendre du temps. Dans ce dernier cas, on pourra étudier un système « forcé », qui, comme pour les systèmes linéaires décrits par une EDO, donnent lieu à un transitoire suivi d'un régime permanent forcé à la fréquence d'excitation.

Comme précédemment, nous allons nous concentrer sur le problème de l'équation de la chaleur, et pour simplifier, réduire la dimensionnalité de l'espace **à une dimension**.

Équation de diffusion 1-D

On cherche la solution numérique $u(x, t)$ de l'équation de diffusion à 1-D :

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} \quad (20)$$

dans le domaine spatial $x \in [0, L]$, et temporel $t \in [0, T]$, avec des condition initiale (CI) :

$$u(x, 0) = u_0(x) \quad \forall x \quad (21)$$

et des conditions aux limites (CL) :

$$u(0, t) = u_g(t) \quad \text{et} \quad u(L, t) = u_d(t) \quad \forall t \quad (22)$$

Le coefficient de diffusion D , nécessairement positif, est supposé uniforme et constant.

Discrétisation du domaine

Il faut discrétiser les deux dimensions :

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

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

On note u_j^n , la valeur numérique de $u(x, t)$ au point $x = j\delta x$ et à l'instant $t = n\delta t$.

- Les CL fixent donc les $u_0^n \equiv u_g^n$ et les $u_{n_x+1}^n \equiv u_d^n$ pour tout n .
- Les CI fixe quant à elles les u_j^0 pour $1 \leq j \leq n_x$.

La méthode de résolution sera fondamentalement itérative, chaque pas de temps nécessitant de calculer les n_x valeurs de u_j^{n+1} **intérieures** à l'aide de u_j^n et des conditions aux limites u_g^{n+1} et u_d^{n+1} .

Les méthodes vont différer selon l'ordre d'évaluation de la dérivée temporelle, et le caractère **explicite** ou **implicite**.

Différences finies

L'algorithme le plus simple et le plus naturel consiste à remplacer l'équation différentielle (20) par l'équation aux différences finies :

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

où le second membre est celui que nous avons utilisé pour les problèmes statiques (du 2^e ordre en δx) et le membre de droite est une expression **explicite**, au 1^{er} ordre de la dérivée temporelle.

Du point de vue temporel, cette approche peut être vue comme une extension de la méthode d'Euler explicite (progressive) étudiée sur les EDO.

Système d'équations linéaires

Il est utile d'introduire le paramètre sans dimension :

$$\alpha = \frac{D \delta t}{\delta x^2} \quad (24)$$

ce qui permet d'écrire :

$$u_j^{n+1} = \alpha u_{j+1}^n + (1 - 2\alpha) u_j^n + \alpha u_{j-1}^n \quad \text{pour} \quad 1 \leq j \leq n_x \quad (25)$$

qui met clairement en évidence le caractère **explicite** puisque les u_j^{n+1} s'obtiennent directement en fonction des valeurs des u_k^n à l'instant **initial** du pas considéré.

Lorsque l'un des u_{j-1}^n ou αu_{j+1}^n n'existe pas, c'est à dire que $j = 1$ ou $j = n_x$ on retrouve une condition de bord, et $u_0^n \equiv u_g^n$ et $u_{n_x+1}^n \equiv u_d^n$ apparaîtront au second membre, affectés d'un signe moins.

Formulation matricielle

En considérant le vecteur $U^n = (u_1^n, \dots, u_{n_x}^n)$ des solutions à $t = n\delta t$ pour les n_x points intérieurs, le système d'équations linéaire prend la forme :

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

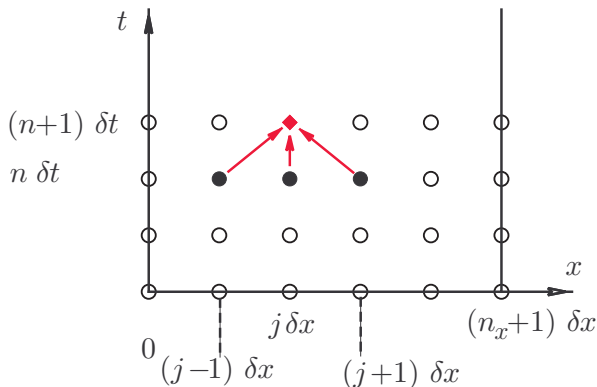
où $V^n = (u_g^n, 0, \dots, 0, u_d^n)$ est le vecteur des conditions aux limites, et la matrice est $M(-\alpha) = \mathbb{1} + \alpha L_{1D}$ où L_{1D} est la matrice tridiagonale représentant la dérivée seconde à *une* dimension :

$$L_{1D} = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix} \longrightarrow M(\alpha) = \begin{pmatrix} 1+2\alpha & -\alpha & 0 & \cdots & 0 \\ -\alpha & 1+2\alpha & -\alpha & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1+2\alpha & -\alpha \\ 0 & \cdots & 0 & -\alpha & 1+2\alpha \end{pmatrix} \quad (26)$$

Schéma de progression explicite

Il est utile de représenter cette situation par un schéma de progression :

FIGURE 6: Schéma illustrant la progression d'un pas de temps avec l'algorithme explicite du premier ordre en δt .



Analyse de stabilité de von Neumann

Cette méthode à l'avantage de la simplicité, mais souffre **d'instabilité**.
 Considérons l'exemple d'une onde harmonique stationnaire :

$$u(x, t) = A(t) \exp(ikx) \quad \Rightarrow \quad u_j^n = A^{(n)} \exp(i k j \delta x) , \quad (27)$$

qui est une solution exacte pour les condition de Dirichlet $u_g = u_d = 0$ si $k = p\pi/L$ et que $A(t)$ vérifie l'EDO :

$$dA/dt = -Dk^2 A(t) . \quad (28)$$

Or la version discrétisée (23) donne, après division par u_j^n :

$$\frac{A^{(n+1)} - A^{(n)}}{A^{(n)}} = D\delta t \frac{e^{ik\delta x} - 2 + e^{-ik\delta x}}{\delta x^2} = -4\alpha \sin^2\left(\frac{k\delta x}{2}\right) .$$

soit une suite géométrique de raison θ , $A^{(n)} = A_0 \theta^n$, dont la stabilité exige :

$$\theta = 1 - 4\alpha \sin^2\left(\frac{k\delta x}{2}\right) \in [-1, 1] \quad \Rightarrow \quad -1 \leq \theta \quad \Rightarrow \quad \alpha \leq \frac{1}{2} . \quad (29)$$

Stabilité : Lien avec le spectre de L_{1D}

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

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

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

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

La condition de stabilité traduit donc simplement le fait que ces valeurs propres sont de module inférieur à 1.

Exemple : Diffusion d'une distribution en « marche »

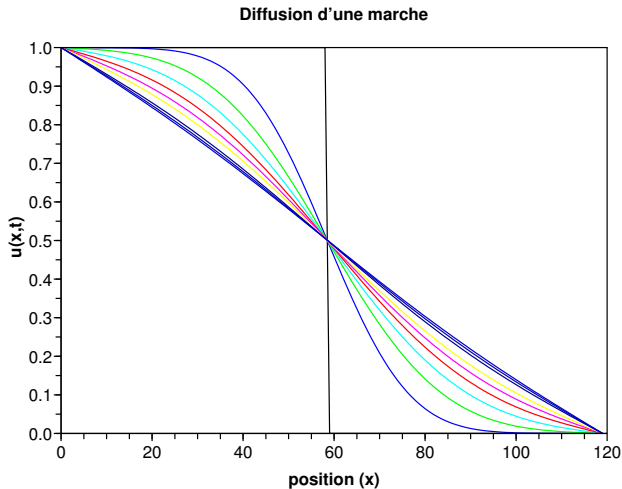


FIGURE 7: Diffusion d'une marche

Exemple : Diffusion d'une superposition de 2 modes

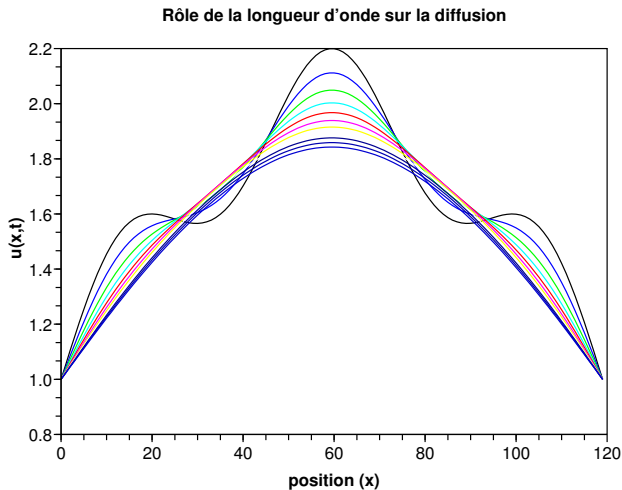


FIGURE 8: Amortissement de deux modes

Lien avec la méthode de relaxation (*déplacer dans Laplace*)

Pour résoudre l'équation de Laplace on aurait pu opter pour la méthode de Jacobi :

On sépare les parties diagonale D et non-diagonale R du laplacien comme ci-contre. On doit avoir :

$$L_{1D}Z = B \Leftrightarrow DZ = B - RZ \Leftrightarrow Z = -\frac{1}{2}(B - RZ)$$

que l'on résout

de façon itérative en faisant (avec $Z^{(0)} = B$:

$$Z^{(0)} = B, \quad Z^{(n+1)} = -\frac{1}{2}(B - RZ^{(n)})$$

qui, vu la structure de R s'écrit aussi :

$$Z_i^{(n+1)} - Z_i^{(n)} = \frac{1}{2} \left\{ (Z_{i+1}^{(n)} + Z_{i-1}^{(n)} - 2Z_i^{(n)}) - B \right\}$$

qui est exactement ce que nous avons écrit à l'équation (23) ou (25), avec $\alpha = \frac{1}{2}$.

Cette approche est une méthode alternative de résolution de l'équation de Laplace, facile à mettre en oeuvre, mais nettement plus lente car itérative. Elle revient à résoudre une équation dépendant du temps, en propageant les conditions de bords par *relaxation*.

Pour simplifier prenons ici L_{1D} :

$$L_{1D} = \underbrace{-2\mathbb{1}}_D + \underbrace{\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}}_R$$

Solution au problème de stabilité : méthode rétrograde

L'algorithme explicite est simple mais la condition de stabilité oblige à faire de très petits pas...

On a vu, avec les EDO, que ces problèmes peuvent être résolus en utilisant un algorithme **implicite**. C'est ce que nous faisons ici en introduisant les nouvelles équations :

$$\frac{u_j^{n+1} - u_j^n}{\delta t} = D \left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\delta x)^2} \right] \quad (30)$$

Les u_j^{n+1} ($j \in [1, n_x]$) s'obtiennent alors en fonction des u_j^n ($j \in [0, n_x + 1]$) en résolvant le système de n_x équations linéaires :

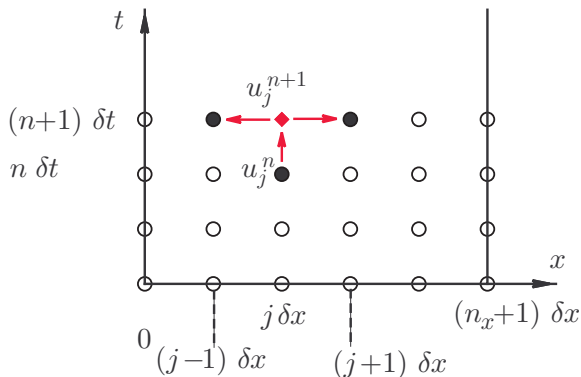
$$-\alpha u_{j-1}^{n+1} + (1 + 2\alpha) u_j^{n+1} - \alpha u_{j+1}^{n+1} = u_j^n \quad \text{pour } j \in [1, n_x] \quad (31)$$

On peut cette fois faire un parallèle avec la méthode d'Euler implicite (rétrograde).

Schéma de progression implicite

Cette méthode met en œuvre le schéma de progression suivant :

FIGURE 9: Schéma illustrant la progression d'un pas de temps avec l'algorithme implicite du premier ordre en δt .



Formulation matricielle

Ce système peut être écrit de façon matricielle sous la forme :

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

avec les mêmes, notations que précédemment, et notamment :

$$M(\alpha) = \mathbf{1} + \alpha L_{1D} = \begin{pmatrix} 1+2\alpha & -\alpha & 0 & \cdots & 0 \\ -\alpha & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha \\ 0 & \cdots & 0 & -\alpha & 1+2\alpha \end{pmatrix} \quad (32)$$

Notons que si M est tridiagonale, son inverse est au contraire une matrice dense. Ainsi u_j^{n+1} dépend de tous les autres éléments, mais d'autant moins qu'ils sont plus éloignés. Bien sûr, lorsque $\alpha \rightarrow 0$, $M(\alpha)^{-1} \rightarrow M(-\alpha)$ ce qui signifie que cette dépendance tend à disparaître et on retrouve alors l'algorithme explicite.

Stabilité de l'algorithme implicite

On cherche à nouveau les modes propres définis par (27).

Si on reporte leur expression dans la nouvelle équation aux différences finies (30), on obtient la raison de la suite géométrique :

$$\theta = \frac{1}{1 + 4 \alpha \sin^2 \left(\frac{k \delta x}{2} \right)} \quad (33)$$

qui montre que l'algorithme implicite est **toujours stable** quel que soit le choix des pas δx et δt . On obtient plus simplement le même résultat à partir des valeurs propres de $M(-\alpha)^{-1}$.

On obtient donc toujours la bonne solution d'équilibre.

Toutefois, cet algorithme étant basé sur une approximation du premier ordre en δt seulement, une description précise de la solution transitoire requiert le choix d'un δt assez petit.

Principe de l'algorithme

Il s'agit d'améliorer l'approximation en différences finies du terme de dérivée partielle temporelle. Plutôt que de modifier celui-ci directement, on utilise la même stratégie que pour la « méthode d'Euler modifiée » pour les EDO, qui est du 2nd ordre en temps.

On fait donc la moyenne des seconds membres des deux algorithmes précédents. On obtient les équations :

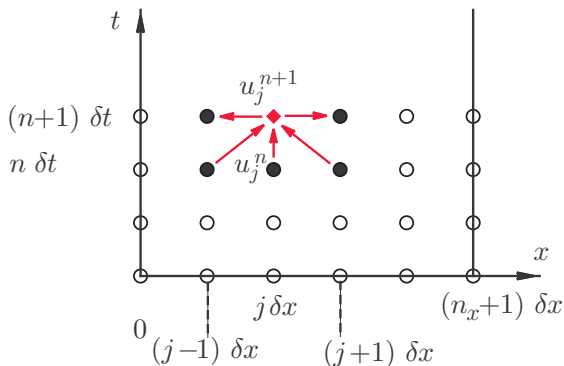
$$\frac{u_j^{n+1} - u_j^n}{\delta t} = \frac{D}{2} \left[\frac{(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) + (u_{j+1}^n - 2u_j^n + u_{j-1}^n)}{(\delta x)^2} \right], \quad (34)$$

dont on montre qu'elles sont du deuxième ordre en δt comme en δx .

Schéma de progression de Crank–Nicholson

Le schéma de progression est désormais le suivant :

FIGURE 10: Schéma illustrant la progression d'un pas de temps avec l'algorithme de Crank–Nicholson.



Stabilité de l'algorithme de Crank–Nicholson

L'étude de stabilité de von Neumann peut être reprise pour ce nouvel algorithme et conduit à une décroissance géométrique des modes avec une raison :

$$\theta = \frac{1 - 2 \alpha \sin^2 \left(\frac{k \delta x}{2} \right)}{1 + 2 \alpha \sin^2 \left(\frac{k \delta x}{2} \right)} \quad (35)$$

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

Formulation matricielle

Les u_j^{n+1} s'obtiennent désormais en fonction des u_j^n en résolvant le système d'équations linéaires (pour $j \in [1, n_x]$) :

$$-\alpha u_{j-1}^{n+1} + (1 + 2\alpha)u_j^{n+1} - \alpha u_{j+1}^{n+1} = \alpha u_{j-1}^n + (1 - 2\alpha)u_j^n + \alpha u_{j+1}^n \quad (36)$$

Ce système peut être écrit de façon matricielle sous la forme :

$$M \left(\frac{\alpha}{2} \right) U^{n+1} = M \left(-\frac{\alpha}{2} \right) U^n + \frac{\alpha}{2} [V^n + V^{n+1}] \quad (37)$$

avec les mêmes notations que pour les deux autres algorithmes.

Sommaire

1 Introduction

- Exemples d'EDP
- Généralités
- Conditions aux limites
- Problèmes statiques et dynamiques

2 Exemple de problème statique : résolution numérique de 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–Nicholson

4 Équation des ondes

Équation des ondes : différences finies

Évoquons brièvement la résolution de l'équation de d'Alembert :

- On discrétise le domaine comme pour l'équation de la chaleur ;
- On se donne des conditions initiales arbitraires $u(x, 0)$;
- On se fixe pour simplifier des conditions aux limites de Dirichlet, ou de Neumann homogènes ;
- On essaye le schéma de différences finies :

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad \Rightarrow \quad \frac{u_j^{n+1} + u_j^{n-1} - 2u_j^n}{(\delta t)^2} = c^2 \frac{u_{j+1}^n + u_{j-1}^n - 2u_j^n}{(\delta x)^2} \quad (38)$$

qui est du deuxième ordre en x et en t .

Algorithme explicite

En posant $\alpha = c\delta t/\delta x$, on obtient les équations **explicites** :

$$u_j^{n+1} = \alpha^2(u_{j+1}^n + u_{j-1}^n) + 2(1 - \alpha^2)u_j^n - u_j^{n-1} . \quad (39)$$

soit, sous forme matricielle :

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

où \mathbf{V} représente à nouveau les conditions aux limites sur les bords.

Ce schéma explicite est potentiellement instable, selon la valeur de α .

Critère CFL

Le critère de Courant–Friedrichs–Lewy donne une prescription sur la valeur de α qui doit être « assez petit ».

La borne précise dépend de l'équation et aussi de la dimension de l'espace.

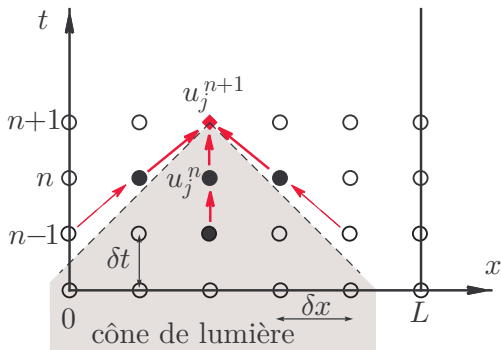
L'idée générale est néanmoins toujours la même :

Le pas de temps δt doit être inférieur au temps mis par l'onde (à la vitesse c) pour traverser un pas d'espace δx .

ou de façon équivalente : Le pas d'espace doit être supérieur à la distance parcourue par l'onde durant le pas de temps. Dans le cas contraire, la méthode est nécessairement instable.

Le schéma de progression est désormais le suivant :

FIGURE 11: Schéma illustrant la progression d'un pas de temps dans l'algorithme explicite pour l'équation de d'Alembert



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 ».