# Différences finies et analyse numérique matricielle

**Article** · September 2010

CITATION
1

READS
472

**1 author:**

Nicolas Champagnat
National Institute for Research in Computer Science and Control
**96** PUBLICATIONS **2,290** CITATIONS

SEE PROFILE

# Différences finies et analyse numérique matricielle

Nicolas Champagnat

# Différences finies et analyse numérique matricielle : cours d'harmonisation en IMAFA

Nicolas Champagnat

15 octobre 2010

## Table des matières

Version provisoire. Merci de me signaler les erreurs !

# 1 Introduction

Le but de ce cours est de présenter les méthodes d'approximation de solutions d'équations aux dérivées partielles (EDP) par *différences finies* en se basant sur le cas particulier de l'équation de la chaleur en dimension 1, puis de présenter les méthodes classiques *d'analyse numérique matricielle* pour la résolution des systèmes linéaires. Une bibliographie succinte se trouve en fin de document.

L'objet de la présente section est de motiver l'utilisation de ces méthodes, et d'introduire les notations et les propriétés de base utilisées.

## 1.1 EDP : exemples et rappels

### 1.1.1 Quelques exemples d'EDP

Les EDP apparaissent dans tous les domaines des sciences et de l'ingénierie. La plupart d'entre elles proviennent de la physique, mais beaucoup interviennent aussi en finance et en assurance. Voici quelques exemples.

**Déformation d'un fil élastique** Cette EDP s'appelle également *problème aux limites*. La position d'un fil élastique maintenu à ses extrémités en $x = 0$ et $x = 1$ à la hauteur 0, soumis à une charge transversale $f(x) \in \mathbb{R}$ et avec rigidité $c(x) \geq 0$ en $x \in [0, 1]$, est donnée par la solution $x \in [0, 1] \mapsto u(x)$ au problème

$$\begin{cases} -\frac{d^2u}{dx^2}(x) + c(x)u(x) = f(x), & \text{pour } x \in ]0, 1[, \\ u(0) = u(1) = 0. \end{cases}$$

$f$ s'appelle le *terme source* de l'équation, et les valeurs en $x = 0$ et 1 sont appelées *conditions au bord*.

Soit $0 < a < b$. En finance, on considère un marché de taux d'intérêt sans risque $r$ constant, et on note $t \in \mathbb{R}_+ \mapsto S_t$ la dynamique temporelle d'un actif risqué de volatilité $\sigma > 0$. Le prix d'arbitrage d'une option délivrant le flux $g_a$ si $S_t$ atteint $a$ avant $b$, et $g_b$ si $S_t$ atteint $b$ avant $a$, est donné dans le modèle de Black-Scholes par $v(y)$ si $S_0 = y$, où

$$\begin{cases} -\frac{\sigma^2}{2}y^2\frac{d^2v}{dy^2}(y) - ry\frac{dv}{dy}(y) + rv(y) = 0, & \text{pour } y \in ]a, b[, \\ v(a) = g_a, \\ v(b) = g_b. \end{cases}$$

On remarque que le changement de variable $y = \exp(x)$ ramène cette équation à la première (avec une condition au bord différente et avec un terme de dérivée du premier ordre en plus).

**Déformation d'un fil inélastique** Avec les mêmes paramètres, la position d'un fil inélastique est la solution $x \in [0,1] \mapsto u(x)$ au problème

$$\begin{cases} \frac{d^4u}{dx^4}(x) + c(x)u(x) = f(x), & \text{pour } x \in ]0,1[, \\ u(0) = u(1) = u'(0) = u'(1) = 0. \end{cases}$$

**Déformation d'une membrane élastique** Si l'on considère maintenant une membrane élastique fixée au bord d'un domaine $D$ ouvert de $\mathbb{R}^2$ (par exemple un disque dans le cas d'un film fermant un pot de confiture), on obtient l'EDP suivante.

$$\begin{cases} -\Delta u(x) + c(x)u(x) = f(x), & \text{pour } x \in D, \\ u(x) = 0, & \text{pour } x \in \partial D, \end{cases}$$

où $\partial D$ désigne la frontière (ou le bord) du domaine $D$ (dans le cas où $D$ est un disque, il s'agit du cercle au bord du disque). $\Delta$ désigne le *laplacien*, défini par

$$\Delta u(x) = \frac{\partial^2 u}{\partial x_1^2}(x) + \frac{\partial^2 u}{\partial x_2^2}(x), \qquad \text{où } x = (x_1, x_2).$$

Lorsque $c = 0$, cette EDP s'appelle *équation de Poisson*.

**L'équation de la chaleur en dimension 1** Il s'agit d'un *problème d'évolution* (c.-à-d. un problème dépendant du temps) décrivant la diffusion de la chaleur dans un fil métallique tendu sur l'intervalle $[0,1]$, soumis à une source de chaleur extérieure $f(t,x)$ dépendant du temps et de la position le long du fil, dont on connait la température initiale $u_0(x)$, la température $g_o(t)$ à l'extrémité $x = 0$, et la température $g_1(t)$ à l'autre extrémité $x = 1$.

$$\begin{cases} \frac{\partial u}{\partial t}(t,x) - \frac{\partial^2 u}{\partial x^2}(t,x) = f(t,x), & \text{pour } t \in ]0,+\infty[ \text{ et } x \in ]0,1[, \\ u(0,x) = u_0(x), & \text{pour } x \in ]0,1[, \\ u(t,0) = g_0(t) \text{ et } u(t,1) = g_1(t), & \text{pour } t \geq 0. \end{cases}$$

$f$ s'appelle le *terme source* ; $u_0$ s'appelle la *condition initiale* ; $g_0$ et $g_1$ s'appellent les *conditions au bord*.

Dans le marché financier décrit plus haut, le prix d'arbitrage d'une option européenne d'échéance $T$ et de flux $h(S_T)$ est donné par $v(t,y)$ à la date d'achat $t \leq T$ lorsque $S_t = y$, où

$$\begin{cases} \frac{\partial v}{\partial t}(t,y) + \frac{\sigma^2}{2}y^2\frac{\partial^2 v}{\partial y^2}(t,y) + ry\frac{\partial v}{\partial y}(t,y) - rv(t,y) = 0, & \text{pour } t \in [0,T[ \text{ et } y \in \mathbb{R}_+^*, \\ v(T,y) = h(y), & \text{pour } y \in ]0,+\infty[. \end{cases}$$

On remarque que $h$ est une *condition terminale*, et que ce problème n'a pas de condition au bord. Ceci est dû au fait que l'intervalle des valeurs possibles

de $y$ n'est pas majoré, et que la valeur minimale 0 ne peut pas être atteinte par l'actif $S_t$.

On remarque qu'en choisissant convenablement $\alpha$ et $\beta$, le changement de fonction inconnue $u(\alpha t, x - \beta t) = v(T - t, \exp(x))$ ramène cette équation à l'équation de la chaleur dans tout l'espace (c.-à-d. $x \in \mathbb{R}$).

**L'équation des ondes** On s'intéresse maintenant à la vibration d'une corde tendue entre $x = 0$ et $x = 1$ et soumise à une force $f(t, x)$ dépendant du temps et de la position le long de la corde. L'EDP correspondante est

$$\begin{cases} \frac{\partial^2 u}{\partial t^2}(t,x) - c^2 \frac{\partial^2 u}{\partial x^2}(t,x) = f(t,x), & \text{pour } t \in ]0, +\infty[ \text{ et } x \in ]0,1[, \\ u(0,x) = u_0(x) \text{ et } \frac{\partial u}{\partial t}(0,x) = u_1(x) & \text{pour } x \in ]0,1[, \\ u(t,0) = u(t,1) = 0, & \text{pour } t \geq 0. \end{cases}$$

Dans le cas d'une membrane vibrante tendue sur la frontière d'un domaine $D \subset \mathbb{R}^2$, l'équation devient

$$\begin{cases} \frac{\partial^2 u}{\partial t^2}(t,x) - c^2 \Delta u(t,x) = f(t,x), & \text{pour } t \in ]0, +\infty[ \text{ et } x \in D, \\ u(0,x) = u_0(x) \text{ et } \frac{\partial u}{\partial t}(0,x) = u_1(x) & \text{pour } x \in ]0,1[, \\ u(t,x) = 0, & \text{pour } x \in \partial D \text{ et } t \geq 0, \end{cases}$$

où le laplacien est calculé seulement par rapport à la variable $x$.

### 1.1.2 Classification des EDP

On a vu qu'il existe des EDP de nature très différentes. La terminologie suivante permet de les distinguer.

**Ordre de l'EDP** Il s'agit du plus grand ordre des dérivées intervenant dans l'équation. Par exemple, l'quation de la chaleur et l'quation des ondes sont d'ordre 2, celle de la déformation du fil inélastique est d'ordre 4.

**EDP linéaire ou non linéaire** On dit que l'EDP est *linéaire* si elle se met sous la form $Lu = f$, où $u \mapsto Lu$ est une application linéaire par rapport à $u$. Sinon, elle est *non linéaire*. Toutes les EDP décrites plus haut sont linéaires. L'étude et la discrétisation des EDP non-linéaires sont beaucoup plus délicates. On rencontre fréquemment des EDP non linéaires dans tous les domaines scientifiques. Par exemple, le prix $v(t, y)$ d'une options américaine de flux $h(S_\tau)$, où le temps d'exercie $\tau$ est choisi par le détenteur de l'option, est donné par

$$\begin{cases} \max \Big\{ \frac{\partial v}{\partial t}(t,y) + \frac{\sigma^2}{2} y^2 \frac{\partial^2 v}{\partial y^2}(t,y) \\ \qquad + ry \frac{\partial v}{\partial y}(t,y) - rv(t,y) \ ; \ h(y) - v(t,y) \Big\} = 0, & \text{dans } [0,T[ \times \mathbb{R}_+^*, \\ v(T,y) = h(y), & \text{pour } y \in \mathbb{R}_+^*. \end{cases}$$

**EDP elliptiques, paraboliques et hyperboliques** En dimension 2, les EDP linéaires d'ordre 2 s'écrivent

$$\sum_{i,j=1}^{2} a_{ij} \frac{\partial^2 u}{\partial x_i x_j}(x) + \sum_{i=1}^{2} b_i \frac{\partial u}{\partial x_i}(x) + cu(x) = f(x).$$

Si l'équation

$$\sum_{i,j=1}^{2} a_{ij} x_i x_j + \sum_{i=1}^{2} b_i x_i + c = 0$$

est celle d'une ellipse (ou parabole, ou hyperbole), on dit que l'EDP est *elliptique* (ou *parabolique*, ou *hyperbolique*). Dans les exemples précédents, l'équation de la membrane élastique est elliptique, l'équation de la chaleur est parabolique, et l'équation des ondes est hyperbolique.

**Conditions au bord** On distingue les conditions au bord de type *Dirichlet* lorsqu'elles sont de la forme $u = g$, et de type *Neumann* lorsqu'elles sont de la forme $\frac{\partial u}{\partial z} = g$. Il existe aussi des conditions au bord *mixtes* qui mélangent les deux.

En finance et en assurance, les EDP linéaires typiques sont presque exclusivement *d'ordre 2, paraboliques ou elliptiques, avec condition au bord de type Dirichlet*. De plus, dans le cas du modèle de Black-Scholes, elles se ramènent par changement de variable exponentiel à l'équation de la la chaleur dans le cas parabolique, et à l'équation de la corde ou membrane vibrante dans le cas elliptique. D'un point de vue numérique, il est plus commode de considérer ces dernières équations, et c'est pourquoi nous nous restreindrons à leur étude.

### 1.1.3 A propos des méthodes de discrétisation d'EDP

Les problèmes de type EDP sont infini-dimensionnels (il s'agit de déterminer toute une fonction). Si l'on veut calculer la solution numériquement, il faut se ramener à un problème en dimension finie. Deux grandes familles de méthodes sont utilisées.

**Approximation par différences finies** Il s'agit d'approcher la solution de l'EDP en un nombre fini de points seulement. On approche alors les dérivées par des accroissements de la fonction entre deux points, aussi appelés *différences finies*. Ces méthodes peuvent s'appliquer à tout type d'EDP. Dans la suite, on se limitera à l'étude de ces méthodes.

**Aproximation par éléments finis** (ou *méthode de Galerkin*, ou *approximation variationnelle*). Il s'agit ici de chercher une fonction "proche" de la solution de l'EDP dans un espace de fonctions de dimension finie. Par exemple, on peut chercher une fonction continues, et affine par morceaux sur un nombre fini de petits intervalles. La difficulté est alors de définir une notion correcte de "solution approchée" à l'EDP. Ces méthodes sont naturellement adaptées aux EDP elliptiques ou hyperboliques. Elles peuvent aussi être employées sur des EDP paraboliques en couplant différences finies et éléments finis.

Afin de s'assurer que ces méthodes convergent, il est souvent nécessaire de savoir que la solution de l'EDP est régulière (typiquement $C^4$). Il est en général difficile de garantir cette régularité, et il existe de nombreux cas où ce n'est pas vrai (particulièrement pour les EDP non linéaires). L'étude de cette question est l'objet de toute une théorie mathématique que nous n'aborderons pas. Nous supposerons toujours dans la suite que la solution est régulière, mais il faut garder à l'esprit que ce n'est pas automatique.

## 1.2 Analyse numérique matricielle : motivations et rappels

Le principal problème en analyse numérique matricielle est celui de la résolution d'un système d'équations linéaires de la forme

$$Ax = b,$$

où $A = (a_{ij})_{1\leq i,j\leq n}$ est une matrice carré $n \times n$ à coefficients réels, et $b = (b_i)_{1\leq i\leq n}$ et $x = (x_i)_{1\leq i\leq n}$ sont des vecteurs colonne de $\mathbb{R}^n$. $x$ est l'inconnue du système.

### 1.2.1 Motivation et classification des méthodes

Les systèmes linéaires de la forme $Ax = b$ interviennent dans un grand nombre de problèmes numériques, en autres

- *l'approximation des solutions des EDP*, que ce soit par différences finies (voir section 2) ou par éléments finis ;
- les problèmes *d'interpolation*, par exemple lorsque l'on cherche à construire une fonction $C^1$ passant par $n$ points $(x_1, y_1), \ldots, (x_n, y_n)$ avec $x_1 < x_2 < \ldots < x_n$, qui soit polynomiale de degré 3 sur chaque intervalle $]x_i, x_{i+1}[$ pour $0 \leq i \leq n$, avec la convention $x_0 = -\infty$ et $x_{n+1} = +\infty$.
- les problèmes *d'optimisation*, par exemple le problème des moindre carrés, où l'on cherche à trouver $v = (v_1, v_2, \ldots, v_n)$ qui minimise

$$\sum_{i=1}^{m} \Big| \sum_{j=1}^{n} v_j w_j(x_i) - c_i \Big|^2,$$

où les fonctions $w_1, \ldots, w_n$ et les vecteurs $x = (x_1, \ldots, x_m)$ et $c = (c_1, \ldots, c_m)$ sont donnés. Il s'agit de trouver une combinaison linéaire des fonctions $w_j$ qui passe aussi près que possible des $n$ points $(x_i, c_i)$, pour $1 \leq i \leq n$. Ce problème revient à résoudre ${}^tBBv = {}^tBc$, avec $B = (w_j(x_i))_{1 \leq i \leq n,\ 1 \leq j \leq m}$.

On distingue trois grandes familles de méthodes de résolution du problème $Ax = b$ :

- les méthodes **directes**, qui donneraient la solution en un nombre fini d'opérations si l'ordinateur faisait des calculs exacts ;
- les méthodes **itératives**, qui consistent à construire une suite $(x^{(k)})_{k \geq 0}$ qui converge vers la solution $x$ ;
- les méthodes **de gradient**, qui font en fait partie des méthodes itératives, et qui consistent à minimiser une fonction en suivant son gradient.

### 1.2.2 Rappels sur les matrices

La matrice $A$ est **symétrique** si $a_{ij} = a_{ji}$ pour tout $1 \leq i, j \leq n$. Une matrice symétrique est diagonalisable et ses valeurs propres sont réelles.

La matrice $A$ est **symétrique définie positive** si elle est symétrique et pour tout $z \in \mathbb{R}^n$, $z \neq 0$,

$$ {}^tzAz = \sum_{1 \leq i,j \leq n} a_{ij} z_i z_j > 0. $$

La matrice $A$ est **diagonale** si tous ses coefficients diagonaux sont nuls : $a_{ij} = 0$ si $i \neq j$. Elle est **triangulaire supérieure** si $a_{ij} = 0$ pour tout $j < i$. Elle est **triangulaire inférieure** si $a_{ij} = 0$ pour tout $i < j$. Elle est **tridiagonale** si $a_{ij} = 0$ pour tout $i, j$ tels que $|i - j| \geq 2$, c.-à-d. si les seuls coefficients non nuls sont les coefficients diagonaux, ceux juste au-dessus de la diagonale, et ceux juste en-dessous de la diagonale :

$$ A = \begin{pmatrix} a_1 & c_1 & & 0 \\ b_2 & a_2 & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ 0 & & b_n & a_n \end{pmatrix}. $$

Les mêmes définitions existent aussi **par blocs**, lorsque chaque coeficient de la matrice est remplacé par une matrice, avec la contrainte que les matrices sur la diagonale sont carrées. Par exemple, une matrice **tridiagonale par blocs** a la forme

$$ A = \begin{pmatrix} A_1 & C_1 & & 0 \\ B_2 & A_2 & \ddots & \\ & \ddots & \ddots & C_{n-1} \\ 0 & & B_n & A_n \end{pmatrix}, $$

où $A_1, \ldots, A_n$ sont des matrices carrées, $B_2, \ldots, B_n$, $C_1, \ldots, C_{n-1}$ sont des matrices (pas nécessairement carrées), et tous les autres "blocs" de la matrice $A$ sont nuls.

Lorsque la matrice $A$ a beaucoup de coefficients nuls (par exemple si elle est tridiagonale), on dit que la matrice est **creuse**. Sinon, elle est **pleine**.

On rappelle égalament la définition des normes $l^1$, $l^2$ et $l^\infty$ dans $\mathbb{R}^n$ : si $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$,

$$\|x\|_1 = |x_1| + \ldots + |x_n|, \qquad \|x\|_2 = \sqrt{x_1^2 + \ldots + x_n^2}, \qquad \|x\|_\infty = \max_{1 \le i \le n} |x_i|.$$

A partir de ces normes, on définit les trois **normes matricielles subordonnées** $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_\infty$ : si $A$ est une matrice $n \times n$,

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \sup_{\|x\|_p = 1} \|Ax\|_p,$$

où $p = 1$, 2, ou $\infty$. Le **rayon spectral** de la matrice $A$ est défini par

$$\rho(A) = \max_{1 \le i \le n} |\lambda_i|,$$

où les $\lambda_i$ sont les valeurs propres (réelles ou complexes) de la matrice $A$. On peut montrer les formules suivantes

**Proposition 1.1** *Pour chacune de ces trois normes, on a pour toutes matrices $n \times n$ $A$ et $B$ et pour tout $x \in \mathbb{R}^n$*

$$\|AB\| \le \|A\| \, \|B\| \qquad et \qquad \|Ax\| \le \|A\| \, \|x\|,$$

*où $\|\cdot\| = \|\cdot\|_1$, ou $\|\cdot\|_2$, ou $\|\cdot\|_\infty$.*

*Si $A = (a_{ij})_{1 \le i,j \le n}$ est une matrice $n \times n$,*

$$\|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|, \quad \|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| \quad et \quad \|A\|_2 = \sqrt{\rho({}^tAA)}.$$

*Si la matrice $A$ est symétrique, on a*

$$\|A\|_2 = \rho(A).$$

# 2 Approximation de la solution de l'équation de la chaleur en dimension 1 par différences finies

La méthode d'approximation de la solution d'une EDP par différences finies consiste à approcher la valeur de la solution en un nombre fini de points, appelés *points de discrétisation du maillage*. Nous allons d'abord décrire la méthode dans un cas simple en dimension 1 avant de nous intéresser aux schémas explicites et implicites pour l'équation de la chaleur.

## 2.1 Principe de la méthode : le problème aux limites d'ordre 2 en dimension 1

On considère le problème aux limites

$$\begin{cases} -\frac{d^2u}{dx^2} + c(x)u(x) = f(x), & \text{pour } x \in ]0,1[, \\ u(0) = g_0, \quad u(1) = g_1, \end{cases} \tag{1}$$

où $f$ et $c$ sont des fonctions données sur $[0,1]$ avec $c \geq 0$.

Les deux ingrédients principaux d'une approximation par différences finies sont le schéma d'approximation des dérivées et la grille de discrétisation.

### 2.1.1 Approximations des dérivées d'une fonction régulière

Plaçons-nous en dimension 1 pour simplifier. L'idée fondamentale consiste à approcher les dérivées (ou les dérivées partielles en dimension plus grande) de la fonction $u$ en un point $x$ en écrivant

$$\frac{du}{dt}(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h} \approx \frac{u(x+h) - u(x)}{h}$$

pour un $h$ petit (mais non nul) fixé. Il suffira ensuite de disposer les points de discrétisation régulièrement espacés de $h$ pour pouvoir employer cette approximation en tout point de discrétisation.

La qualité de l'approximation précédente dépend fortement de la régularité de la fonction $u$. Si la fonction $u$ est $C^2$ sur l'intervalle $[0,1]$, on déduit aisément du développement de Taylor

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(\theta)$$

où $\theta \in ]x, x+h[$, l'inégalité

$$\left| \frac{u(x+h) - u(x)}{h} - u'(x) \right| \leq Ch,$$

où $C = \sup_{y \in [0,1]} |u''(y)|$, pour tout $0 \leq x \leq 1-h$.

On dit alors que l'approximation de $u'(x)$ par $[u(x+h) - u(x)]/h$ est **consistante d'ordre 1**. Si l'erreur est majorée par $Ch^p$ pour $p > 0$ fixé, on dit plus généralement que l'approximation est **consistante d'ordre** $p$.

On vérifie facilement que l'approximation de $u'(x)$ par $\dfrac{u(x) - u(x-h)}{h}$ est également consistante d'ordre 1.

Il est possible d'obtenir une meilleure approximation de la dérivée première d'une fonction en utilisant un quotient différentiel *centré* : si la fonction

$u$ est $C^3$, la formule de Taylor à l'ordre 3 donne

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u'''(\theta_1),$$
$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u'''(\theta_2),$$

où $\theta_1 \in ]x, x+h[$ et $\theta_2 \in ]x-h, x[$. On obtient alors

$$\left|\frac{u(x+h)-u(x-h)}{2h} - u'(x)\right| = \left|\frac{h^2}{6}(u'''(\theta_1)+u''(\theta_2))\right| \leq C'h^2,$$

où $C' = \sup_{y\in[0,1]} |u'''(y)|$. L'approximation

$$u'(x) \approx \frac{u(x+h)-u(x-h)}{2h}$$

est donc **consistante d'ordre 2**.

En ce qui concerne les dérivées secondes, on démontre facilement (exercice !) en utilisant la formule de Taylor à l'ordre 4, que si $u$ est $C^4$ au voisinage de $x$, l'approximation

$$u''(x) \approx \frac{u(x+h)-2u(x)+u(x-h)}{h^2} \tag{2}$$

est **consistante d'ordre 2**. Plus précisément,

$$\left|\frac{u(x+h)-2u(x)+u(x-h)}{h^2} - u''(x)\right| \leq \frac{h^2}{12}\sup_{y\in[0,1]} |u^{(4)}(y)|, \tag{3}$$

pour tout $x \in [h, 1-h]$.

### 2.1.2 Approximation de (1) par différences finies

Soit $N \in \mathbb{N}$ fixé. On définit les points de discrétisation du maillage par

$$x_i = ih, \quad i \in \{0, 1, \ldots, N+1\}, \quad \text{où } h = \frac{1}{N+1}.$$

Les points $x_0 = 0$ et $x_{N+1} = 1$ constituent le *bord du domaine* (les extrémités de l'intervalle de définition de $u$), et les points $x_1, \ldots, x_N$ sont appelés *points internes du maillage.*

On cherhce en chacun de ces points une valeur approchée, notée $u_i$, de $u(x_i)$. On prend naturellement $u_0 = u(0) = g_0$ et $u_{N+1} = u(1) = g_1$. Pour les sommets internes, on utilise l'approximation (2) de la dérivée seconde décrite plus haut :

$$\begin{cases} -\dfrac{u_{i-1}-2u_i+u_{i+1}}{h^2} + c(x_i)u_i = f(x_i), & \text{pour } i \in \{1, \ldots, N\}, \\ u_0 = g_0, \quad u_{N+1} = g_1. \end{cases} \tag{4}$$

On observe qu'on obtient $N$ équations servant à déterminer les $N$ inconnues $u_1, \ldots, u_N$. On dit usuellement qu'on a discrétisé le problème par une méthode de différences finies utilisant le *schéma à trois points* de la dérivée seconde. On note que la connaissance des conditions au bord $u_0$ et $u_{N+1}$ est nécessaires à la résolution du système, puisqu'elles apparaissent dans (4) lorsque $i = 1$ et $i = N$.

Matriciellement, le problème s'écrit :

$$A_h u_h = b_h, \tag{5}$$

où

$$u_h = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{pmatrix}, \qquad b_h = \begin{pmatrix} f(x_1) + \frac{g_0}{h^2} \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \\ f(x_N) + \frac{g_1}{h^2} \end{pmatrix},$$

$$A_h = A_h^{(0)} + \begin{pmatrix} c(x_1) & 0 & \ldots & 0 \\ 0 & c(x_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & c(x_N) \end{pmatrix},$$

$$A_h^{(0)} = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \ldots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \ldots & 0 & -1 & 2 \end{pmatrix}. \tag{6}$$

On note que le système ne porte que sur les inconnues $u_1, \ldots, u_N$. En particulier, les conditions au bord $u_0 = g_0$ et $u_{N+1} = g_1$ n'apparaissent que dans le vecteur $b_h$. On note également que les matrices $A_h$ et $A_h^{(0)}$ sont tridiagonales. Pour déterminer la solution discrète $u_h$, il suffit donc de résoudre le système linéaire tridiagonal (5). La matrice $A_h$ est inversible puisqu'on a :

**Proposition 2.1** *Supposons $c \geq 0$. La matrice $A_h$ est symétrique définie positive.*

**Démonstration** La matrice $A_h$ est clairement symétrique. Soit $z$ un vecteur de $\mathbb{R}^N$, de composantes $z_1, \ldots, z_N$. On a

$$\begin{aligned} {}^t z A_h z &= {}^t z A_h^{(0)} z + \sum_{i=1}^{N} c(x_i) z_i^2 \geq {}^t z A_h^{(0)} z \\ &= \frac{z_1^2 + (z_2 - z_1)^2 + \ldots + (z_{N-1} - z_N)^2 + z_N^2}{h^2}. \end{aligned}$$

Cette quantité est positive, et non nulle si $z \neq 0$ (car au moins l'un des termes au numérateur est non nul). □

### 2.1.3 Convergence de la méthode

Afin d'étudier la convergence de la solution approchée $u_h$ vers la solution exacte $u$ lorsque $h \to 0$, on commence par étudier *l'erreur de consistance* :

**Définition 2.2** *On appelle* ***erreur de consistance*** *du schéma (4) $A_h u_h = b_h$, le vecteur $\varepsilon_h(u)$ de $\mathbb{R}^N$ défini par*

$$\varepsilon_h(u) = A_h(\pi_h(u)) - b_h, \quad \text{où } \pi_h(u) = \begin{pmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_N) \end{pmatrix}.$$

*$\pi_h(u)$ représente la projection de la solution exacte sur le maillage. On dit que le schéma est* ***consistant*** *pour la norme $\|\cdot\|$ de $\mathbb{R}^N$ si $\lim_{h\to 0} \|\varepsilon_h(u)\| = 0$. Si de plus il existe $C$ indépendante de $h$ telle que*

$$\|\varepsilon_h(u)\| \leq Ch^p$$

*pour $p > 0$, on dit que* ***le schéma est d'ordre*** *$p$ pour la norme $\|\cdot\|$.*

Usuellement, on utilise les normes $\|\cdot\| = \|\cdot\|_1$, $\|\cdot\|_2$ ou $\|\cdot\|_\infty$.

En utilisant (3), on obtient immédiatement

$$\|\varepsilon_h(u)\|_\infty \leq \frac{h^2}{12} \sup_{y\in[0,1]} |u^{(4)}(y)|,$$

et donc

**Proposition 2.3** *Supposons que la solution $u$ du problème (1) est $C^4$ sur $[0,1]$. Alors le schéma (4) est consistant d'ordre 2 pour la norme $\|\cdot\|_\infty$.*

L'erreur de convergence est l'erreur entre la solution approchée $u_h$ est la solution exacte aux points du maillage $\pi_h(u)$. Afin de majorer cette erreur, il suffit d'observer que, par définition de l'erreur de consistance et puique $A_h u_h = b_h$,

$$u_h - \pi_h(u) = -(A_h)^{-1}\varepsilon_h(u).$$

On peut montrer que $\|(A_h)^{-1}\|_\infty \leq 1/8$ (admis), ce qui donne le résultat de convergence :

**Théorème 2.4** *On suppose $c \geq 0$. Si la solution $u$ du problème (1) est $C^4$ sur $[0,1]$, alors le schéma (4) est convergent d'ordre 2 pour la norme $\|\cdot\|_\infty$. Plus précisément,*

$$\|u_h - \pi_h(u)\|_\infty \leq \frac{h^2}{96} \sup_{y\in[0,1]} |u^{(4)}(y)|.$$

On notera que, dans cet exemple, la consistance du schéma permet immédiatement d'obtenir la convergence. Ceci est particulier à cet exemple simple. En général, un autre ingrédient est nécessaire à la convergence : *la stabilité* du schéma (voir section 2.3).

## 2.2 L'équation de la chaleur en dimension 1 : les schémas implicites et explicites

On s'intéresse au problème

$$\begin{cases} \frac{\partial u}{\partial t}(t,x) - \frac{\partial^2 u}{\partial x^2}(t,x) = f(t,x), & \text{pour } t \in ]0,T[ \text{ et } x \in ]0,1[, \\ u(0,x) = u_0(x), & \text{pour } x \in ]0,1[, \\ u(t,0) = 0 \text{ et } u(t,1) = 0, & \text{pour } t \geq 0, \end{cases} \tag{7}$$

où l'on a pris pour simplifier des conditions au bord nulles. Il s'agit de l'équation de la chaleur en dimension 1 (d'espace), qui est un problème parabolique en dimension 2 (temps et espace). Cet exemple est typique de la situation générale des problèmes paraboliques. On distingue deux grandes familles d'approximations par différences finies : les schémas *explicites* et les schémas *implicites*.

### 2.2.1 Le maillage, les conditions initiales et les conditions au bord

Dans toute la suite, on supposera pour simplifier que la condition initiale est compatible avec les conditions au bord : $u_0(0) = u_0(1) = 0$.

On va chercher à calculer une solution approchée en un nombre fini de points $(t_j, x_i)$ du domaine espace-temps $[0,T] \times [0,1]$. On va se limiter au cas le plus simple du *maillage régulier* : soient $N, M$ deux entiers fixés. On pose

$$x_i = ih, \quad \forall i \in \{0,1,\ldots,N+1\}, \quad \text{où } h = \frac{1}{N+1},$$
$$t_j = j\Delta t, \quad \forall j \in \{0,1,\ldots,M+1\}, \quad \text{où } \Delta t = \frac{T}{M+1}.$$

En particulier, $x_0 = 0$, $x_{N+1} = 1$, $t_0 = 0$ et $t_{M+1} = T$. Les points $(t_j, x_i)$ sont alors les points d'intersection d'une "grille" régulière en espace-temps.

L'approximation par différences finies consiste alors à chercher une approximation, notée $u_i^{(j)}$, de $u(t_j, x_i)$ (notez que l'indice en temps est en exposant, et l'indice en espace en indice, comme précédemment).

Les valeurs approchées aux points de maillage au bord du domaine et en $t = 0$ sont données par la valeur exacte — donnée — de la fonction $u$ :

$$u_0^{(j)} = u_{N+1}^{(j)} = 0, \qquad \forall j \in \{0,\ldots,M+1\} \tag{8}$$
$$u_i^{(0)} = u_o(x_i), \qquad \forall i \in \{0,\ldots,N+1\}. \tag{9}$$

Ceci laisse $N(M+1)$ inconnues à déterminer — les $u_i^{(j)}$ pour $1 \leq i \leq N$ et $1 \leq j \leq M+1$. Les équations correspondantes sont obtenues en approchant les dérivées partielles dans l'EDP par des quotients différentiels. On a déjà vu que le terme de dérivée seconde pouvait être approché avec

$$\frac{\partial^2 u}{\partial x^2}(t_j, x_i) \approx \frac{u(t_j, x_{i+1}) - 2u(t_j, x_i) + u(t_j, x_{i-1})}{h^2} \approx \frac{u_{i+1}^{(j)} - 2u_i^{(j)} + u_{i-1}^{(j)}}{h^2}$$

(schéma à trois points pour la dérivée seconde). Comme on l'a vu dans la section 2.1.1, plusieurs choix sont possibles pour l'approximation de la dérivée en temps. Chacun de ces choix conduit à une famille de schémas distincte.

### 2.2.2 Le schéma explicite, à trois points pour la dérivée seconde

La première possibilité est d'utiliser *l'approximation décentrée à droite*

$$\frac{\partial u}{\partial t}(t_j, x_i) \approx \frac{u(t_{j+1}, x_i) - u(t_j, x_i)}{\Delta t} \approx \frac{u_i^{(j+1)} - u_i^{(j)}}{\Delta t}.$$

On obtient alors le schéma suivant :

$$\frac{u_i^{(j+1)} - u_i^{(j)}}{\Delta t} - \frac{u_{i+1}^{(j)} - 2u_i^{(j)} + u_{i-1}^{(j)}}{h^2} = f(t_j, x_i), \qquad \forall i \in \{1, \ldots, N\}, \quad j \in \{0, \ldots, M\},$$

soit $N(M+1)$ équations pour $N(M+1)$ inconnues. On note que les conditions aux limites (8) et (9) doivent être connues pour résoudre ces équations, et que l'indice de temps $j$ doit varier entre 0 et $M$.

Il est commode de réécrire ce système vectoriellement : on introduit la notation

$$U^{(j)} = \begin{pmatrix} u_1^{(j)} \\ \vdots \\ u_N^{(j)} \end{pmatrix}, \qquad \forall j \in \{0, \ldots, M+1\}.$$

Le vecteur $U^{(0)}$ est donné par les conditions initiales, et le schéma précédent s'écrit :

$$\frac{U^{(j+1)} - U^{(j)}}{\Delta t} + A_h^{(0)} U^{(j)} = C^{(j)}, \qquad \forall j \in \{0, \ldots, M\}, \tag{10}$$

où la matrice $A_h^{(0)}$ a été définie dans (6), et

$$C^{(j)} = \begin{pmatrix} f(t_j, x_1) \\ \vdots \\ f(t_j, x_N) \end{pmatrix}, \qquad \forall j \in \{0, \ldots, M+1\}.$$

Ce système peut également se réécrire sous la forme

$$U^{(j+1)} = (\mathrm{Id} - \Delta t\, A_h^{(0)})U^{(j)} + \Delta t\, C^{(j)}, \qquad \forall j \in \{0, \dots, M\}, \tag{11}$$

où on utilise la notation Id pour la matrice identité.

Cette équation justifie le nom *explicite* pour ce schéma, puiqu'il permet de calculer la valeur approchée au temps $t_{j+1}$ par simple produit de matrice avec la valeur approchée au temps $t_j$. En particulier, aucune inversion de matrice ou résolution de système linéaire n'est nécessaire pour le calcul.

### 2.2.3 Le scéma implicite, à trois points pour la dérivée seconde

On aurait pu approcher la dérivée partielle en temps par *l'approximation décentrée à gauche*

$$\frac{\partial u}{\partial t}(t_j, x_i) \approx \frac{u(t_j, x_i) - u(t_{j-1}, x_i)}{\Delta t} \approx \frac{u_i^{(j)} - u_i^{(j-1)}}{\Delta t}.$$

On obtient alors le schéma suivant :

$$\frac{u_i^{(j)} - u_i^{(j-1)}}{\Delta t} - \frac{u_{i+1}^{(j)} - 2u_i^{(j)} + u_{i-1}^{(j)}}{h^2} = f(t_j, x_i), \qquad \forall i \in \{1, \dots, N\}, \quad j \in \{1, \dots, M+1\},$$

couplé aux mêmes conditions initiales que le schéma explicite (noter que cette fois l'indice $j$ varie de 1 à $M+1$). Vectoriellement, on obtient

$$\frac{U^{(j)} - U^{(j-1)}}{\Delta t} + A_h^{(0)} U^{(j)} = C^{(j)}, \qquad \forall j \in \{1, \dots, M+1\},$$

où les $C^{(j)}$ sont définis comme plus haut, ce qui se réécrit

$$U^{(j)} = (\mathrm{Id} + \Delta t\, A_h^{(0)})^{-1} U^{(j-1)} + \Delta t\, (\mathrm{Id} + \Delta t\, A_h^{(0)})^{-1} C^{(j)}, \qquad \forall j \in \{1, \dots, M+1\}. \tag{12}$$

On remarque que la matrice $(\mathrm{Id} + \Delta t\, A_h^{(0)})$ est symétrique définie positive — et donc inversible — puisque $A_h^{(0)}$ est symétrique définie positive.

Ce schéma est dit *implicite* puisque, contrairement au schéma explicite, sa résolution nécessite la résolution d'un système linéaire à chaque pas de temps (ou le calcul initial de l'inverse de la matrice $(\mathrm{Id} + \Delta t\, A_h^{(0)})$, utilisé ensuite à chaque pas de temps). La résolution du schéma implicite est donc plus coûteuse que le schéma explicite. Cependant, comme on le verra plus loin, ce coût en temps de calcul est largement compensé par la meilleure stabilité du schéma.

### 2.2.4 Un autre schéma

Tous les schémas imaginables ne sont pas nécessairement bons, comme le montre l'exemple suivant : on pourrait chercher à améliorer la précision en temps en utilisant l'approximation d'ordre 2 (voir section 2.1.1) suivante de la dérivée en temps :

$$\frac{\partial u}{\partial t}(t_j, x_i) \approx \frac{u(t_{j+1}, x_i) - u(t_{j-1}, x_i)}{2\Delta t} \approx \frac{u_i^{(j+1)} - u_i^{(j-1)}}{2\Delta t}.$$

On obtiendrait alors le schéma

$$\frac{U^{(j+1)} - U^{(j-1)}}{2\Delta t} + A_h^{(0)} U^{(j)} = C^{(j)}, \qquad \forall j \in \{2, \dots, M+1\}, \tag{13}$$

qui est explicite en temps, puisque $U^{(j+1)}$ s'exprime directement en fonction de $U^{(j)}$ et $U^{(j-1)}$. On note qu'il est nécessaire pour ce schéma de connaître $U^{(0)}$ **et** $U^{(1)}$. Le précalcul de $U^{(1)}$ peut par exemple être fait en utilisant un pas en temps de l'une des méthodes précédentes.

Ce schéma, appellé *schéma saute-mouton* ou schéma de Richardson, est à la fois explicite et *a priori* d'ordre 2 en temps. Cependant, comme on le verra plus loin, il est toujours instable, et donc numériquement inutilisable.

## 2.3 Consistence, stabilité et convergence

Les trois schémas précédents peuvent s'écrire sous la forme générale

$$B_1 U^{(j+1)} + B_0 U^{(j)} + B_{-1} U^{(j-1)} = C^{(j)}, \tag{14}$$

avec $B_1$ inversible ou $B_1 = 0$ et $B_0$ inversible. Pour le schéma explicite, on a

$$B_1 = \frac{1}{\Delta t}\,\mathrm{Id}, \qquad B_0 = -\frac{1}{\Delta t}\,\mathrm{Id} + A_h^{(0)}, \qquad B_{-1} = 0.$$

Pour le schéma implicite, on a

$$B_1 = 0, \qquad B_0 = \frac{1}{\Delta t}\,\mathrm{Id} + A_h^{(0)}, \qquad B_{-1} = -\frac{1}{\Delta t}\,\mathrm{Id}.$$

Pour le schéma suate-mouton, on a

$$B_1 = \frac{1}{\Delta t}\,\mathrm{Id}, \qquad B_0 = A_h^{(0)}, \qquad B_{-1} = -\frac{1}{\Delta t}\,\mathrm{Id}.$$

L'étude de la convergence de ces schémas est basée sur les propriétés de consistance et stabilité, définies ci-dessous pour le schéma général (14).

On pourrait bien sûr imaginer des schémas faisant intervenir des indices en temps plus grands que $j+1$ et plus petits que $j-1$. Les définitions suivantes s'étendraient sans difficulté à ces schémas.

### 2.3.1 Consistance

La définition suivante étend celle de la section 2.1.3.

**Définition 2.5** *On appelle* ***erreur de consistance à l'instant*** $t_j$ *du schéma (14) le vecteur* $\varepsilon_h(u)^{(j)}$ *de* $\mathbb{R}^N$ *défini par*

$$\varepsilon_h(u)^{(j)} = B_1(\pi_h(u))(t_{j+1}) + B_0(\pi_h(u))(t_j) + B_{-1}(\pi_h(u))(t_{j-1}) - C^{(j)},$$

*où*

$$\pi_h(u)(t) = \begin{pmatrix} u(t,x_1) \\ \vdots \\ u(t,x_N) \end{pmatrix}.$$

*On dit que le schéma est* ***consistant*** *pour la norme* $\|\cdot\|$ *de* $\mathbb{R}^N$ *si*

$$\sup_{0\le j\le M+1} \|\varepsilon_h(u)^{(j)}\| \to 0, \quad \textit{quand} \quad \Delta t \to 0 \textit{ et } h \to 0.$$

*Si de plus il existe* $C > 0$, $p > 0$ *et* $q > 0$ *indépendants de* $\Delta t$ *et* $h$ *tels que*

$$\sup_{0\le j\le M+1} \|\varepsilon_h(u)^{(j)}\| \le C[(\Delta t)^p + h^q],$$

*on dit que* ***le schéma est consistant d'ordre*** $p$ ***en temps et*** $q$ ***en espace*** *pour la norme* $\|\cdot\|$.

Notons que la limite $\delta t \to 0$ correspond à la limite $M \to +\infty$, et la limite $h \to 0$ à $N \to +\infty$.

On peut alors montrer la :

**Proposition 2.6** *Supposons que la solution* $u$ *au problème (7) est* $C^2$ *par rapport à la variable* $t$ *et* $C^4$ *par rapport à la variable* $x$. *Alors les schémas explicites et implicites sont consistants d'ordre 1 en temps et 2 en espace.*

*Si de plus* $u$ *est* $C^3$ *par rapport à la variable* $t$, *alors le schéma saute-mouton est consistant d'ordre 2 en temps et en espace.*

**Démonstration** Les trois résultats se démontrent de façon similaires. On ne détaille la preuve que pour le schéma explicite. En utilisant le fait que

$$f(t_j,x_i) = \frac{\partial u}{\partial t}(t_j,x_i) - \frac{\partial^2 u}{\partial x^2}(t_j,x_i),$$

il découle de la définition de l'erreur de consistance que

$$\varepsilon(u)_i^{(j)} = E_i - F_i,$$

où l'on a posé

$$E_i = \frac{u(t_{j+1},x_i) - u(t_j,x_i)}{\Delta t} - \frac{\partial u}{\partial t}(t_j,x_i),$$
$$F_i = \frac{u(t_j,x_{i+1}) - 2u(t_j,x_i) + u(t_{j-1},x_i}{h^2} - \frac{\partial^2 u}{\partial x^2}(t_j,x_i).$$

Par un développement de Taylor par rapport à la variable de temps ($x_i$ étant fixé), on obtient :

$$u(t_{j+1}, x_i) = u(t_j, x_i) + \Delta t \, \frac{\partial u}{\partial t}(t_j, x_i) + \frac{\Delta t^2}{2} \, \frac{\partial^2 u}{\partial t^2}(\theta, x_i),$$

où $\theta \in ]t_j, t_{j+1}[$, de sorte que

$$E_i = \frac{\Delta t}{2} \, \frac{\partial^2 u}{\partial t^2}(x_i, \theta).$$

Concernant $F_i$, on montre de même que

$$F_i = \frac{h^2}{24} \left( \frac{\partial^4 u}{\partial x^4}(t_j, \xi_1) + \frac{\partial^4 u}{\partial x^4}(t_j, \xi_2) \right)$$

où $\xi_1 \in ]x_{i-1}, x_i[$ et $\xi_2 \in ]x_i, x_{i+1}[$. Compte-tenu des hypothèses de régularité, on en déduit facilement la majoration de l'erreur de consistance. □

### 2.3.2 Stabilité et convergence

Du point de vue numérique, la propriété fondamentale d'un schéma est celle de convergence :

**Définition 2.7** *On considère le schéma (14) et on suppose que les données initiales vérifient : si $B_1 = 0$ ou $B_{-1} = 0$ (par exemple pour les schémas explicite et implicite), supposons*

$$\|U^{(0)} - (\pi_h u)(0)\| \to 0 \quad \textit{quand} \quad \Delta t, \ h \to 0;$$

*si $B_1 \neq 0$ et $B_{-1} \neq 0$ (par exemple pour le schéma saute-mouton), supposons*

$$\|U^{(0)} - (\pi_h u)(0)\| + \|U^{(1)} - (\pi_h u)(t_1)\| \to 0 \quad \textit{quand} \quad \Delta t, \ h \to 0.$$

*On dit alors que le schéma (14) est* ***convergent*** *(pour la norme $\|\cdot\|$ en espace) si*

$$\sup_{0 \leq j \leq M+1} \|U^{(j)} - (\pi_h u)(t_j)\| \to 0, \quad \textit{quand} \quad \Delta t, \ h \to 0.$$

**Remarque 2.8** ***Attention !*** *Il se peut que la convergence ci-dessus ait lieu pour $\delta t$ et $h$ tendant vers 0, mais pas nécessairement indépendamment l'un de l'autre. Voir la propostion 2.11 plus loin.*

En plus de la consistance, la propriété de *stabilité* joue un rôle fondamental dans l'étude de la convergence des schémas d'approximations des EDP linéaires.

**Définition 2.9** *On dit que le schéma (14) est* ***stable pour la norme*** $\|\cdot\|$ *dans* $\mathbb{R}^N$ *s'il existe deux constantes positives* $C_1(T)$ *et* $C_2(T)$ *indépendantes de* $\Delta t$ *et* $h$ *telles que*

$$\max_{0\leq j\leq M+1}\|U^{(j)}\| \leq C_1(T)\|U^{(0)}\| + C_2(T)\max_{0\leq j\leq M+1}\|C^{(j)}\|$$

*si* $B_1 = 0$ *ou* $B_{-1} = 0$*, ou bien*

$$\max_{0\leq j\leq M+1}\|U^{(j)}\| \leq C_1(T)(\|U^{(0)}\| + \|U^{(1)}\|) + C_2(T)\max_{0\leq j\leq M+1}\|C^{(j)}\|$$

*si* $B_1 \neq 0$ *et* $B_{-1} \neq 0$*, et ceci quelles que soient les données initiales* $U^{(0)}$ *(et* $U^{(1)}$*) et les termes sources* $C^{(j)}$*.*

Le théorème fondamental suivant donne le lien entre convergence, consistance et stabilité.

**Théorème 2.10 (Théorème de Lax)** *Le schéma (14) est convergent si et seulement s'il est consistant et stable.*

**Démonstration** Nous n'allons démontrer que l'implication utile en pratique de ce théorème : supposons que le schéma est consistant et stable, et montrons qu'il est convergent. En faisant la différence entre la définition de l'erreur de consistance et l'équation (14), on obtient

$$B_1 e^{(j+1)} + B_0 e^{(j)} + B_{-1} e^{(j-1)} = -\varepsilon(u)^{(j)},$$

où on a noté $e^{(j)} = U^{(j)} - (\pi_h u)(t_j)$ l'erreur à l'instant $t_j$. Le schéma étant stable, on a

$$\max_{0\leq j\leq M+1}\|e^{(j)}\| \leq C_1(T)\|e^{(0)}\| + C_2(T)\max_{0\leq j\leq M+1}\|\varepsilon(u)^{(j)}\|$$

si $B_1 = 0$ ou $B_{-1} = 0$ (et similairement dans le cas contraire). Lorsque $\Delta t,\ h \to 0$, par hypothèse, $\|e^{(0)}\| \to 0$, et par consistance du schéma, $\max_{0\leq j\leq M+1}\|\varepsilon(u)^{(j)}\| \to 0$, ce qui montre la convergence du schéma. □

Grâce à ce résultat, il est maintenant facile d'étudier la convergence pour la norme $\|\cdot\|_\infty$ du schéma explicite :

**Proposition 2.11** *Sous la condition*

$$\frac{\Delta t}{h^2} \leq \frac{1}{2}, \tag{15}$$

*le schéma explicite (10) est convergent en norme* $\|\cdot\|_\infty$*.*

La condition (15) s'appelle **condition de CFL** (pour Courant, Friedrichs, Lewy, 1928). On peut en fait montrer que c'est une condition nécessaire et suffisante pour la convergence du schéma, ce qui montre que la convergence d'un schéma ne peut pas nécessairement être assurée quelle que

soit la manière dont $\Delta t$ et $h$ tendent vers 0. C'est une condition très contraignante d'un point de vue pratique, puisque si l'on souhaite avoir une bonne précision en espace, il est nécessaire de choisir $h$ petit, ce qui impose un choix de $\Delta t$ *nettement plus petit*. Par exemple, le choix $h = 10^{-3}$ impose $\Delta t$ de l'ordre de $10^{-6}$. Le coût en temps de calcul par rapport à un code où l'on peut choisir $\Delta t = 10^{-3}$ (par exemple, le schéma implicite, voir plus loin) est 1000 fois supérieur.

**Démonstration** D'après le théorème de Lax et la Propositon 2.6, il suffit de vérifier la stabilité pour la norme $\|\cdot\|_\infty$ du schéma. D'après l'équation (11), on a

$$u_i^{(j+1)} = (1-2r)u_i^{(j)} + ru_{i+1}^{(j)} + ru_{i-1}^{(j)} + \Delta t\, f(t_j, x_i),$$

où l'on a posé $r = \frac{\Delta t}{h^2}$. Le point clé de la preuve est que, sous l'hypothèse (15), le membre de droite de cette équation est une *combinaison convexe*[1] de $u_{i-1}^{(j)}$, $u_i^{(j)}$ et $u_{i+1}^{(j)}$ (plus un terme indépendant des $u_i^{(j)}$). On en déduit

$$\begin{aligned} |u_i^{(j+1)}| &\leq (1-2r)|u_i^{(j)}| + r|u_{i-1}^{(j)}| + r|u_{i+1}^{(j)}| + \Delta t\, |f(t_j, x_i)| \\ &\leq (1-2r+r+r)\|U^{(j)}\|_\infty + \Delta t\, \|C^{(j)}\|_\infty, \end{aligned}$$

d'où on déduit

$$\|U^{(j+1)}\|_\infty \leq \|U^{(j)}\|_\infty + \Delta t \max_{0\leq k\leq M+1} \|C^{(k)}\|.$$

Une récurrence évidente donne alors

$$\|U^{(j)}\|_\infty \leq \|U^{(0)}\|_\infty + j\Delta t \max_{0\leq k\leq M+1} \|C^{(k)}\| \leq \|U^{(0)}\|_\infty + T \max_{0\leq k\leq M+1} \|C^{(k)}\|$$

pour tout $j \in \{0, \ldots, M+1\}$, ce qui achève la démonstration de la stabilité du schéma explicite. □

### 2.3.3 Stabilité en norme $\|\cdot\|_h$

**La norme $\|\cdot\|_h$** L'étude de la stabilité des schémas implicite et saute-mouton est plus facile en norme $\|\cdot\|_2$. Puisque les vecteurs $U^{(j)}$ appartiennent à $\mathbb{R}^N$, il convient de normaliser convenablement cette norme afin de pouvoir obtenir des majorations indépendantes du choix de $N$ (et donc de $h$). La normalisation correcte est la suivante : on définit la norme $\|\cdot\|_h$ par

$$\|\cdot\|_h = \sqrt{h}\|\cdot\|_2, \quad \text{c.-à-d.} \quad \|V\|_h = \sqrt{h\sum_{i=1}^{N} v_i^2}, \quad \text{où} \quad V = (v_i)_{1\leq i\leq N}.$$

[1] C.-à-d. une somme de la forme $\alpha u_{i-1}^{(j)} + \beta u_i^{(j)} + \gamma u_{i+1}^{(j)}$ avec $\alpha$, $\beta$ et $\gamma$ positifs ou nuls et $\alpha + \beta + \gamma = 1$.

Pour s'en convaincre, il suffit de considérer une fonction $v(x)$ continue sur $[0,1]$, et d'écrire l'approximation d'intégrale par sommes de Riemann suivante :

$$\int_0^1 v^2(x) = \lim_{N\to+\infty} \frac{1}{N}\sum_{i=1}^{N} v(x_i)^2 = \lim_{h\to 0} \|\pi_h v\|_h^2.$$

On remarque que, pour tout vecteur $V = (v_i)_{1\le i\le N}$ de $\mathbb{R}^N$,

$$\|V\|_h^2 \le hN\|V\|_\infty^2 \le \|V\|_\infty^2.$$

Il est alors immédiat de déduire de la proposition 2.6 la

**Proposition 2.12** *Les trois schémas explicite, implicite et saute-mouton sont consistants pour la norme $\|\cdot\|_h$.*

**Critère de stabilité** Afin d'étudier la stabilité en norme $\|\cdot\|_h$ de ces trois schémas, on va considérer un schéma général de la forme

$$U^{(j+1)} = BU^{(j)} + \Delta t D^{(j)}, \qquad U^{(0)} \text{ donné}, \tag{16}$$

où la matrice $B$ les vecteurs $D^{(j)}$ peuvent dépendre de $h$ et $\Delta t$. On suppose de plus que la matrice $B$ est *symétrique* et que les vecteurs $D^{(j)}$ sont contrôlés par le terme source du schéma, au sens où il existe une constante $C > 0$ telle que

$$\max_{0\le j\le M+1} \|D^{(j)}\| \le C \max_{0\le j\le M+1} \|C^{(j)}\|.$$

On déduit aisément de (11) et (12) que les schémas explicites et implicites peuvent se mettre sous cette forme et satisfont ces hypothèses. On a alors le résultat suivant

**Théorème 2.13** *Un schéma de la forme (16) est stable pour la norme matricielle subordonnée $\|\cdot\|_h$ si et seulement s'il existe une constante $C_0 > 0$ indépendante de $\Delta t$ et $h$ telle que*

$$\rho(B) \le 1 + C_0\Delta t,$$

*où $\rho(B)$ est le* rayon spectral de $B$ *(voir section 1.2.2).*

Remarquons que ce critère porte uniquement sur la matrice $B$ et est indépendant des termes source $D^{(j)}$. Notons également que le plus souvent, il est suffisant en pratique de vérifier la condition plus restrictive $\rho(B) \le 1$ pour garantir la convergence de la plupart des schémas.

**Démonstration** On va commencer par démontrer que le schéma est convergent si et seulement si

$$\max_{0\leq j\leq M+1} \|B^j\|_2 \leq C(T) \tag{17}$$

pour une constante $C(T)$ indépendante de $\Delta t$ et $h$. Supposons d'abord que le schéma est stable. Il existe alors deux constantes $C_1(T)$ et $C_2(T)$ telles que

$$\max_{0\leq j\leq M+1} \|U^{(j)}\|_h \leq C_1(T)\|U^{(0)}\|_h + C_2(T) \max_{0\leq j\leq M+1} \|C^{(j)}\|_h,$$

quels que soient $U^{(0)}$ et les termes sources $C^{(j)}$. En particulier, pour $C^{(j)} = 0$, on obtient $U^{(j)} = B^j U^{(0)}$ et

$$\max_{0\leq j\leq M+1} \|B^j U^{(0)}\|_h \leq C_1(T)\|U^{(0)}\|_h.$$

Remarquons que, puisque $\|\cdot\|_h$ est proportionnelle à $\|\cdot\|_2$ sur $\mathbb{R}^N$, la norme matricielle subordonnée à $\|\cdot\|_h$ est exactement la norme $\|\cdot\|_2$ (voir section 1.2.2). On déduit donc de l'inégalité précédente, valable pour tout $U^{(0)} \in \mathbb{R}^N$, que

$$\max_{0\leq j\leq M+1} \|B^j\|_2 \leq C_1(T),$$

ce qui montre que (17) est vérifiée avec $C(T) = C_1(T)$.

Réciproquement, supposons (17). On montre par récurrence que

$$U^{(j)} = B^j U^{(0)} + \Delta t[D^{(j-1)} + BD^{(j-2)} + \ldots + B^{j-1}D^{(0)}].$$

On en déduit, d'après notre hypothèse sur $D^{(j)}$, que

$$\begin{aligned}
\|U^{(j)}\|_h &\leq \|B^j\|_2 \, \|U^{(0)}\|_h + C\Delta t \sum_{k=0}^{j-1} \|B^{j-k-1}\|_2 \, \|D^{(j)}\|_h \\
&\leq C(t)\|U^{(0)}\|_h + Cj\Delta t C(T) \max_{0\leq k\leq M+1} \|C^{(k)}\|_h \\
&\leq C(T)\|U^{(0)}\|_h + TC(T) \max_{0\leq k\leq M+1} \|C^{(k)}\|_h,
\end{aligned}$$

où l'on a utilisé le fait que $j\Delta t \leq T$ pour $0 \leq j \leq M+1$. Ceci démontre que le schéma est stable.

Montrons maintenant que la conditon (17) est équivalente à $\rho(B) \leq 1 + C_0\Delta t$. D'après la propsition 1.1, la matrice $B$ étant symétrique, on a

$$\|B^j\|_2 = \rho(B^j) = \rho(B)^j.$$

Supposons d'abord que $\rho(B) \leq 1 + C_0\Delta t$. Pour tout $j \geq 0$ tel que $j\Delta t \leq T$, on en déduit

$$\|B^j\|_2 \leq [1 + C_0\Delta t]^j \leq \exp(C_0\Delta t) \leq \exp(C_0 T),$$

en utilisant l'inégalité $\ln(1+x) \leq x$ pour tout $x > 0$. On a donc démontré (17) avec $C(T) = \exp(C_0 T)$, et le schéma est stable.

Réciproquement, supposons (17). Remarquons que $C(T) \geq 1$ car $\|B^0\|_2 = \|\mathrm{Id}\|_2 = 1$. On a alors $\rho(B)^j \leq C(T)$ pour tout $j \leq M+1$, donc en particulier pour $j = M+1$, ce qui donne

$$\rho(B) \leq C(T)^{\frac{1}{M+1}} = C(T)^{\frac{\Delta t}{T}}.$$

La fonction $x \mapsto C(T)^{\frac{x}{T}}$ est convexe sur l'intervalle $[0,T]$, donc

$$C(T)^{\frac{x}{T}} \leq \frac{C(T)-1}{T}x, \qquad \forall x \in [0,T].$$

En appliquant cette inégalité avec $x = \Delta t$, on obtient $\rho(B) \leq 1 + C_1\Delta t$ avec $C_1 = [C(T)-1]/T$, ce qui achève la preuve du théorème 2.13. □

**Stabilité su schéma explicite** L'application du critère précédent nécessite de connaître les valeurs propres de $B$, et donc les valeurs propres de $A_h^{(0)}$. On vérfie facilement à la main que la matrice $A_h^{(0}$ admet les valeurs propres

$$\lambda_k = \frac{4}{h^2}\,\sin^2\left(\frac{k\pi}{2(N+1)}\right), \qquad \forall k \in \{1,\dots,N\},$$

associées aux vecteurs propres $V^{(k)} = (v_j^{(k)})_{1\leq j\leq N}$ où

$$v_j^{(k)} = \sin\left(j\,\frac{k\pi}{N+1}\right).$$

Dans le cas du schéma explicite, la matrice $B$ est donnée par

$$B = \mathrm{Id} - \Delta t A_h^{(0)}.$$

Ses valeurs propres sont donc définies par

$$\lambda_k(B) = 1 - \frac{4\Delta t}{h^2}\,\sin^2\left(\frac{k\pi}{2(N+1)}\right), \qquad \forall k \in \{1,\dots,N\}.$$

On a $\lambda_k(B) \leq 1$ pour tout $k$, mais peut-on avoir $\lambda_k(B) \leq -1$ ? La plus petite des valeurs propres de $B$ est $\lambda_N(B)$, donc le schéma est convergent si

$$\frac{\Delta t}{h^2}\sin^2\left(\frac{N\pi}{2(N+1)}\right) \leq \frac{1}{2}$$

lorsque $\Delta t$ et $h$ tendent vers 0. En particulier, lorsque $h \to 0$, le sinus converge vers 1, et on a donc la

**Proposition 2.14** *Le schéma explicite est convergent pour la norme $\|\cdot\|_h$ si*

$$\frac{\Delta t}{h^2} \leq \frac{1}{2}. \tag{18}$$

*Plus précisément, si la solution $u$ du problème (7) est $C^2$ par rapport à la variable $t$ et $C^4$ par rapport à la variable $x$, sous la condition (18), le schéma est convergent d'ordre 1 en temps et 2 en espace.*

Ce résultat est consistent avec la proposition 2.11.

**Stabilité su schéma implicite** Dans le cas du schéma implicite, la matrice $B$ est donnée par

$$B = (\mathrm{Id} + \Delta t A_h^{(0)})^{-1}.$$

Ses valeurs propres sont donc données par

$$\lambda_k(B) = \frac{1}{1 + \frac{4\Delta t}{h^2} \sin^2\left(\frac{k\pi}{2(N+1)}\right)}, \qquad \forall k \in \{1, \dots, N\}.$$

Chacune de ces valeurs propres appartient à l'intervalle $[0, 1]$, donc

**Proposition 2.15** *Le schéma implicite est convergent pour la norme $\|\cdot\|_h$ quelle que soit la manière dont $\Delta t$ et $h$ tendent vers 0. Plus préciément, si la solution $u$ du problème (7) est $C^2$ par rapport à la variable $t$ et $C^4$ par rapport à la variable $x$, le schéma est convergent d'ordre 1 en temps et 2 en espace.*

On dit que le schéma implicite est *inconditionnellement stable*. Le schéma implicite reclame un travail numérique supplémentaire par rapport au schéma explicite (l'inversion de la matrice $\mathrm{Id} + \Delta t A_h^{(0)}$). En contrepartie, il n'impose aucune condition sur le choix de $\Delta t$ et $h$. Ce bénéfice est tellement important que l'approximation par schéma de différences finies implicite est presque toujours préféré à l'approximation par différences finies explicite dans les cas pratiques.

**Le schéma saute-mouton** Le sch'ema saute-mouton se ramène à la forme (16) en doublant la dimension du problème : pour tout $j \in \{1, \dots, M+1\}$, soit

$$V^{(j)} = \begin{pmatrix} U^{(j)} \\ U^{(j+1)} \end{pmatrix},$$

avec $U^{(j)}$ donné par le schéma (13). On a alors $V^{(j+1)} = BV^{(j)} + \Delta t D^{(j)}$, avec

$$B = \begin{pmatrix} -2\Delta t A_h^{(0)} & \mathrm{Id} \\ \mathrm{Id} & 0 \end{pmatrix} \qquad \text{et} \qquad D^{(j)} = \begin{pmatrix} C^{(j)} \\ 0 \end{pmatrix}.$$

L'étude de la stabilité du schéma saute-mouton est donc également donnée par le critère du théorème 2.13.

On vérifie aisément (exercice !) que les valeurs propres de la matrice $B$ sont données par les racines de $\lambda - \frac{1}{\lambda} = -2\Delta t \lambda_k$, pour $k \in \{1, \dots, M\}$. Les valeurs propres de $B$ sont donc données par

$$\lambda_k^{\pm}(B) = \frac{-\mu_k \pm \sqrt{\mu_k^2 + 4}}{2}, \text{ avec } \mu_k = \frac{8\Delta t}{h^2} \sin^2\left(\frac{k\pi}{2(N+1)}\right), \ \forall k \in \{1, \dots, N\}.$$

Pour $k = N$, on a

$$\mu_N = \frac{8\Delta t}{h^2} \sin^2\left(\frac{N\pi}{2(N+1)}\right) \geq \frac{4\Delta t}{h^2}$$

pour $N$ suffisamment grand. On a donc

$$\rho(B) \geq \lambda_N^+(B) = \frac{\mu_N + \sqrt{\mu_N^2 + 4}}{2} \geq \frac{\mu_N + 2}{2} \geq 1 + \frac{2\Delta t}{h^2}.$$

D'après le théorème 2.13, on a ainsi montré la

**Proposition 2.16** *Le schéma (13) est instable pour la norme $\|\cdot\|_h$.*

Même si le schéma saute-mouton est explicite, donc facile à programmer, et a une erreur de consistance d'ordre 2 en temps et en espace, il est totalement instable ! Un tel schéma doit être rejeté car il n'es pas convergent.

## 2.4 Quelques prolongements

Jusqu'ici, nous avons examiné en détail les propriétés des approximations par différences finies de l'équation de chaleur en dimension 1. La méthode se généralise aisément à tout type d'EDP (même si l'analyse de la méthode est en général délicate). L'objet de cette section est de décrire quelques aspects de cette généralisation.

### 2.4.1 Cas des conditions au bord non nulles

On peut rajouter des conditions au bord non nulles à l'équation de la chaleur (7) :

$$u(t,0) = g_0(t), \qquad u(t,1) = g_1(t), \qquad \forall t \in [0,T].$$

Comme dans l'exemple en dimension 1 (voir section 2.1.2), on se convainc facilement que ces conditions au bord apparaissent dans les schémas d'approximation par différences finies dans les termes source. On obtient dans notre cas

$$C^{(j)} = \begin{pmatrix} f(t_j, x_1) + \frac{g_0(t_j)}{h^2} \\ f(t_j, x_2) \\ \vdots \\ f(t_j, x_{N-1}) \\ f(t_j, x_N) + \frac{g_1(t_j)}{h^2} \end{pmatrix}. \tag{19}$$

L'analyse des schémas peut se faire exactement comme plus haut. En particulier, la définition et les résultats sur la consistance sont les mêmes (voir section 2.3.1). La convergence est définie de la même manière, avec l'hypothèse supplémentaire que les conditions au bord approchées convergent vers les conditions au bord exactes (voir la définition 2.7). Concernant la stabilité, la définition 2.9 doit être modifiée comme suit :

**Définition 2.17** *On dit que le schéma (14) avec termes source (19) est* ***stable pour la norme*** $\|\cdot\|$ *dans* $\mathbb{R}^N$ *s'il existe deux constantes positives* $C_1(T)$ *et* $C_2(T)$ *indépendantes de* $\Delta t$ *et* $h$ *telles que*

$$\max_{0\leq j\leq M+1}\|U^{(j)}\| \leq C_1(T)\Big[\|U^{(0)}\| + \max_{0\leq j\leq M+1}(|g_0(t_j)|+|g_1(t_j)|)\Big] + C_2(T)\max_{0\leq j\leq M+1}\|C^{(j)}\|$$

*si* $B_1 = 0$ *ou* $B_{-1} = 0$*, ou bien*

$$\max_{0\leq j\leq M+1}\|U^{(j)}\| \leq C_1(T)\Big[(\|U^{(0)}\| + \|U^{(1)}\| + \max_{0\leq j\leq M+1}(|g_0(t_j)|+|g_1(t_j)|)\Big] + C_2(T)\max_{0\leq j\leq M+1}\|C^{(j)}\|$$

*si* $B_1 \neq 0$ *et* $B_{-1} \neq 0$*, et ceci quelles que soient les données initiales* $U^{(0)}$ *(et* $U^{(1)}$*) et les termes sources* $C^{(j)}$*.*

Avec cette définition, il est facile d'étendre les résultats de stabilité et convergence des sections 2.3.2 et 2.3.3 au cas avec conditions au bord non nulles.

### 2.4.2 Cas d'un terme d'ordre 1 supplémentaire en espace

La méthode des différences finies s'étend aisément à des EDP plus générales. A titre d'exemple, nous détaillons le cas de l'EDP de la chaleur avec convection :

$$\begin{cases} \frac{\partial u}{\partial t}(t,x) - \frac{\partial^2 u}{\partial x^2}(t,x) - b\frac{\partial u}{\partial x}(t,x) = f(t,x), & \text{pour } t\in]0,T[ \text{ et } x\in]0,1[,\\ u(0,x) = u_0(x), & \text{pour } x\in]0,1[,\\ u(t,0) = 0 \text{ et } u(t,1) = 0, & \text{pour } t\geq 0, \end{cases} \tag{20}$$

où $b\in\mathbb{R}$.

De nouveau, il est nécessaire de choisir une discrétisation du terme $\frac{\partial u}{\partial x}$ : centrée, décentrée à droite ou décentrée à gauche. Le choix convenable est dicté par l'étude du cas de la discrétisation explicite de l'EDP (20) **sans terme de dérivée seconde** (il s'agit alors d'une *équation de transport*) :

$$\begin{cases} \frac{\partial u}{\partial t}(t,x) - b\frac{\partial u}{\partial x}(t,x) = f(t,x), & \text{pour } t\in]0,T[ \text{ et } x\in]0,1[,\\ u(0,x) = u_0(x), & \text{pour } x\in]0,1[,\\ u(t,0) = 0 \text{ et } u(t,1) = 0, & \text{pour } t\geq 0, \end{cases}$$

Dans ce cas, on vérifie facilement que le schéma explicite s'écrit sous la forme habituelle

$$U^{(j+1)} = BU^{(j)} + \Delta t C^{(j)},$$

où la matrice $B$ est donnée par : dans la cas du schéma décentré à droite

$$B = \mathrm{Id} + b\,\frac{\Delta t}{h}\begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \dots & \dots & 0 & -1 \end{pmatrix} ;$$

dans le cas du schéma décentré à droite

$$B = \mathrm{Id} + b\,\frac{\Delta t}{h}\begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ -1 & 1 & \ddots & & \vdots \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix} ;$$

dans le cas du schéma centré

$$B = \mathrm{Id} + b\,\frac{\Delta t}{2h}\begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \ddots & \vdots \\ 0 & -1 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \dots & 0 & -1 & 0 \end{pmatrix} .$$

En examinant la preuve de la proposition 2.11, il est facile de voir que la méthode utilisée pour montrer la stabilité du schéma pour la norme $\|\cdot\|_\infty$ ne s'applique que sous la condition que **tous les coefficients de la matrice $B$ sont positifs ou nuls**[2]. Il s'agit en fait d'un **principe général** : toute approximation *explicite* par différences finies d'une *EDP linéaire parabolique* se met sous la forme

$$U^{(j+1)} = BU^{(j)} + \Delta t C^{(j)},$$

et le schéma est stable pour la norme $\|\cdot\|_\infty$ *si et seulement si* la matrice $B$ a tous ses coefficients positifs ou nuls. La condition qui en résulte, portant sur les paramètres de discrétisation $\Delta t$ et $h$, s'appelle *condition de CFL*.

On voit alors que, suivant le signe de $b$, un seul choix de discrétisation est possible : si $b > 0$, il faut utiliser le schéma décentré à droite ; si $b < 0$, il faut utiliser le schéma décentré à gauche ; dans tous les cas, le schéma

[2]On observe que la somme des coefficients de la matrice $B$ est automatiquement égale à 1, ce qui permet d'utiliser l'argument de *combinaison convexe* utilisé dans la preuve de la proposition 2.11.

centré est instable pour la norme $\|\cdot\|_\infty$ (sauf si $b = 0$). Dans les deux cas, la condition de CFL obtenue est

$$\frac{\Delta t}{h} \leq \frac{1}{|b|}.$$

Dans le cas plus général de l'équation de la chaleur avec convection (20), le schéma n'est pas nécessairement instable si on choisit une autre discrétisation du terme de dérivée première. Cependant, on préfère employer la discrétisation décentrée à droite lorsque $b > 0$ et la discrétisation décentrée à gauche lorsque $b < 0$, parce que ce choix n'impose *qu'une seule condition de CFL*. On montre (exercice !) que la condition de CFL correspondant à l'approximation explicite par différences finies de (20) est

$$2\frac{\Delta t}{h^2} + |b|\frac{\Delta t}{h} \leq 1.$$

### 2.4.3 Cas de la dimension 2 dans un domaine rectangulaire

A titre de second exemple d'extension de la méthode, examinons ce qui se passe pour l'équation de la chaleur en dimension 2 dans un domaine carré (le cas d'un domaine rectangulaire est une extension évidente de ce qui suit) :

$$\begin{cases} \frac{\partial u}{\partial t}(t,x,y) - \Delta u(t,x,y) = f(t,x,y), & \text{pour } t \in ]0,T[ \text{ et } (x,y) \in ]0,1[^2, \\ u(0,x,y) = u_0(x,y), & \text{pour } (x,y) \in ]0,1[^2, \\ u(t,x,0) = u(t,x,1) = 0, & \text{pour } t \in [0,T], \\ u(t,0,y) = u(t,1,y) = 0, & \text{pour } t \in [0,T], \end{cases} \tag{21}$$

où le laplacien porte sur les variables $(x, y)$.

Il est alors naturel de considérer le maillage carré suivant : soit $N, M \in \mathbb{N}^*$ ; on pose

$$x_i = ih \quad \text{et} \quad y_j = jh, \quad \forall i,j \in \{0,\dots,N+1\}, \quad \text{où} \quad h = \frac{1}{N+1},$$
$$t_k = k\Delta t, \quad \forall k \in \{0,\dots,M+1\}, \quad \text{où} \quad \Delta t = \frac{T}{P+1}.$$

L'approximation *à 5 points* du laplacien par différences finies est alors donnée par

$$\begin{aligned} \Delta u(x,y) &= \frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) \\ &\approx \frac{u(x-h,y) - 2u(x,y) + u(x+h,y)}{h^2} + \frac{u(x,y-h) - 2u(x,y) + u(x,y+h)}{h^2} \\ &= \frac{u(x+h,y) + u(x-h,y) + u(x,y+h) + u(x,y-h) - 4u(x,y)}{h^2}. \end{aligned}$$

On note alors $u_{i,j}^{(k)}$ la valeur approchée de $u(t_k, x_i, y_j)$, et les conditions aux bords permettent de déterminer

$$u_{i,j}^{(0)} = u_0(x_i, y_j), \qquad \text{et} \qquad u_{0,j}^{(k)} = u_{N+1,j}^{(k)} = u_{i,0}^{(k)} = u_{i,N+1}^{(k)} = 0.$$

Afin d'écrire vectoriellement les schémas implicites et explicites obtenus, il faut d'abord choisir comment énumérer les points du maillage correspondant aux inconnues $u_{i,j}^{(k)}$ avec $1 \leq i, j \leq N$. Le choix le plus simple est de les énumérer ligne par ligne de la façon suivante : $x_{1,1}$, $x_{2,1}, \ldots$, $x_{N-1,1}$, $x_{N,1}$, $x_{1,2}$, $x_{2,2}, \ldots$, $x_{N-1,2}$, $x_{N,2}, \ldots$, $x_{1,N}$, $x_{2,N}, \ldots$, $x_{N-1,N}$, $x_{N,N}$. On peut ainsi définir le vecteur $U^{(k)}$ d'approximation au temps $t_k$ en énumérant les $u_{i,j}^{(k)}$ dans l'ordre correspondant.

Il est alors facile de se convaincre que le schéma explicite a la forme

$$\frac{U^{(k+1)} - U^{(k)}}{\Delta t} + A'_h U^{(k)} = C^{(k)}, \qquad \forall j \in \{0, \ldots, M\},$$

où $C^{(k)}$ est le vecteur des $f(t_k, x_i, y_j)$, et la matrice $A'_h$ est symétrique et tridiagonale par blocs, de la forme

$$A'_h = \frac{1}{h^2} \begin{pmatrix} A & -\mathrm{Id} & 0 & \ldots & 0 \\ -\mathrm{Id} & A & -\mathrm{Id} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\mathrm{Id} & A & -\mathrm{Id} \\ 0 & \ldots & 0 & -\mathrm{Id} & A \end{pmatrix},$$

où les matrices $A$, Id et 0 sont toutes de taille $n \times n$, et

$$A = \begin{pmatrix} 4 & -1 & 0 & \ldots & 0 \\ -1 & 4 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 4 & -1 \\ 0 & \ldots & 0 & -1 & 4 \end{pmatrix}.$$

De même, le schéma implicite a la forme

$$\frac{U^{(k)} - U^{(k-1)}}{\Delta t} + A'_h U^{(k)} = C^{(k)}, \qquad \forall j \in \{1, \ldots, M+1\},$$

et demande donc d'inverser la matrice $\mathrm{Id} + \Delta t A'_h$.

On verra dans la prochaine partie que certaines méthodes d'analyse numérique matricielles sont particulièrement bien adaptées à ce type de matrices (symétriques, tridiagonales ou tridiagonales par blocs), qu'on rencontre très souvent dans l'approximation implicite par différences finies d'EDP linéaires paraboliques.

# 3 Analyse numérique matricielle

Afin d'appliquer à des cas pratiques le schéma implicite d'approximation par différences finies d'EDP d'évolution, il est nécessaire de mettre en œuvre un algorithm de résolution de systèmes linéaires. L'objet de cette section est de décrire les différentes méthodes de résolution d'un système linéaire de la forme

$$Ax = b,$$

où $A = (a_{ij})_{1\leq i,j\leq n}$ est une matrice carré $n \times n$ à coefficients réels, supposée **inversible**, et $b = (b_i)_{1\leq i\leq n}$ et $x = (x_i)_{1\leq i\leq n}$ sont des vecteurs colonne de $\mathbb{R}^n$. $x$ est l'inconnue du système.

Théoriquement, si $A$ est inversible, $\det(A) \neq 0$, et on a existence et unicité de la solution $x$, donnée par la formule explicite

$$x_i = \frac{\det(A_i)}{\det(A)}, \qquad \forall i \in \{1, \dots, n\},$$

où $A_i$ est la matrice obtenues en remplaçant la $i$-ième colonne de $A$ par le vecteur $b$. Cependant, l'application de cette formule est inacceptable pour la résolution pratique des systèmes, car son coût est de l'ordre de $(n+1)! = (n+1)n(n-1)\dots 1$ *floating-point operations (flops)*. En fait, le calcul de chaque déterminant par la formuel

$$\det(A) = \sum_{\sigma} (-1)^{\varepsilon(\sigma)} \prod_{i=1}^{n} a_{i,\sigma(i)},$$

(où la somme porte sur toutes les permutations $\sigma$ sur $n$ objets) requiert $n!$ flops. Sur un ordinateur usuel, la résolution d'un système de 50 équation par cette méthode nécessiterait un temps plus long que l'âge présumé de l'univers !

Les sections suivantes détaillent des algorithmes alternatifs avec un coût raisonnable. On distingue les méthodes *directes*, qui donneraient un résultat exact en un nombre fini d'opérations si l'ordinateur faisait des calculs exacts, et les méthodes *itératives* qui consistent à construire une suite $(x_k)_{k\geq 0}$ convergeant vers la solution $x$.

## 3.1 Méthodes directes

### 3.1.1 Généralités

**Systèmes linéaires et inversion de matrices** Les méthodes présentées ici ont pour but de résoudre le système linéaire $Ax = b$. Notons qu'il est inutile pour cela de calculer effectivement l'inverse de la matrice $A$ (cela conduirait à un coût numérique supplémentaire inutile).

Il peut être parfois utile de calculer l'inverse de la matrice $A$, par exemple pour l'approximation implicite par différences finies de l'équation de la chaleur. Cette méthode demande de résoudre un grand nombre de fois un système $Ax = b$ (une fois pour chaque pas de temps), avec un second membre $b$ différent à chaque fois. Dans ce type de situation, il vaut mieux calculer $A^{-1}$ une fois pour toute avant de commencer la résolution du schéma implicite.

Pour cela, notons $v^{(1)}, \ldots, v^{(n)}$ les colonnes de la matrice $A^{-1}$, c.-à-d. $A^{-1} = (v^{(1)}, \ldots, v^{(n)})$. La relation $AA^{-1} = \mathrm{Id}$ se traduit par les $n$ systèmes

$$Av^{(k)} = e^{(k)}, \qquad \forall k \in \{1, \ldots, n\},$$

où $e^{(k)}$ est le vecteur colonne ayant tous ses éléments nuls sauf celui de la ligne $k$, qui vaut 1 :

$$e^{(k)} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} 1 \\ \vdots \\ k. \\ \vdots \\ n \end{matrix}$$

Il est donc nécessaire de résoudre $n$ systèmes linéaires, ce qui multiplie par $n$ le coût des algorithmes décrits ci-dessous.

Certains algorithmes, comme par exemple l'algorithme de Gauss, permettent de calculer $A^{-1}$ plus rapidement : une fois connue la décomposition $LU$ de la matrice $A$, les $n$ systèmes précédents se résolvent comme $2n$ systèmes triangulaires, ce qui conduit à un coût moindre que le calcul effectif des matrices $L$ et $U$ (voir section 3.1.3).

**A propos de la précision des méthodes directes** On suppose dans ce paragraphe que la matrice $A$ est symétrique définie positive. On commence par définir le conditionnement de la matrice $A$ :

**Définition 3.1** *On appelle* ***conditionnement*** $K(A)$ *de la matrice $A$ symétrique définie positive comme le rapport entre la valeur propre maximale et la valeur propre minimale de $A$ :*

$$K(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}.$$

La précision des méthodes directes de résolution de systèmes linéaires est intimement relié au conditionnement de la matrice $A$ : par exemple, si on résout le système $Ax = b$ avec un ordinateur, à cause des erreurs d'arrondi, on ne trouve pas la solution exacte $x$, mais une solution approchée $\hat{x}$. Alors $\hat{x}$ n'est pas nécessairement proche de $x$ : l'erreur relative entre $x$ et $\hat{x}$ est

reliée au conditionnement de la matrice $A$ par l'inégalité

$$\frac{\|x - \hat{x}\|_2}{\|x\|_2} \leq K(A) \frac{\|r\|_2}{\|b\|_2},$$

où $r$ est le résidu $r = b - A\hat{x}$. On peut même montrer que cette inégalité est en fait une égalité pour certains choix de $b$ et $x - \hat{x}$. En particulier, si le conditionnement de $A$ est grand, l'erreur $\|x - \hat{x}\|$ peut être grande même si le résidu est petit. Cette inégalité montre que c'est également le cas s'il y des erreurs d'arrondis sur le vecteur $b$, même si le calcul de la solution du système est exact.

### 3.1.2 Systèmes triangulaires

On dit que le système $Ax = b$ est *triangulaire supérieur* si la matrice $A$ est triangulaire supérieure. Le système est *triangulaire inférieur* si la matrice $A$ est triangulaire inférieure. Ces deux cas sont parmi les systèmes les plus simples à résoudre (après les systèmes diagonaux, bien sûr), et nous allons décrire les algorithmes correspondants.

Remarquons d'abord que, dans les deux cas, $\det(A) = \prod_{i=1}^{n} a_{ii}$. La matrice $A$ est donc inversible si et seulement si $a_{ii} \neq 0$ pour tout $i \in \{1, \ldots, n\}$.

Si $A$ est triangulaire inférieure, on a

$$x_1 = \frac{b_1}{a_{11}},$$

et pour $i = 2, 3, \ldots, n$,

$$x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j \right).$$

Cet algorithme s'appelle *méthode de descente*.

Si $A$ est triangulaire supérieure, on a

$$x_n = \frac{b_n}{a_{nn}},$$

et pour $i \in= n-1, n-2, \ldots, 2, 1$,

$$x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right).$$

Cet algorithme s'appelle *méthode de remontée*.

Le nombre de multliplications et divisions nécessaires dans ces algorithmes est $n(n+1)/2$ et la nombre d'additions et soustractions $n(n-1)/2$, soit un total de $n^2$ flops.

### 3.1.3 Méthode d'élimination de Gauss et décomposition LU

L'*algorithme d'élimination de Gauss* consiste à calculer une suite de matrices $A^{(k)} = (a_{ij}^{(k)})$ pour $k = 1, 2, \ldots, n$ de manière récursive de la façon suivante : on pose $A^{(1)} = A$, et pour $k = 1, 2, \ldots, n$, ayant calculé $A^{(k)}$, on calcule $A^{(k+1)}$ comme suit :

$$\begin{aligned} l_{ik} &= \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \qquad i = k+1, \ldots, n \ ; \\ a_{ij}^{(k+1)} &= a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)}, \qquad i = k+1, \ldots, n, \qquad j = k, \ldots, n. \end{aligned}$$

Autrement dit, connaissant $A^{(k)}$ et ayant calculé $l_{k+1,k}, l_{k+2,k}, \ldots, l_{nk}$, on note $L_i$ la $i$-ième ligne de la matrice $A^{(k)}$. L'algorithme revient alors à retrancher aux lignes $L_{k+1}, L_{k+2}, \ldots, L_n$ les lignes $l_{k+1,k}L_k, l_{k+2,k}L_k, \ldots, l_{nk}L_k$. Ce faisant, on a annulé les coefficients de la colonne $k$ de la matrice $A^{(k)}$ entre les lignes $k+1$ et $n$.

La matrice $A^{(k+1)}$ a donc la forme

$$A^{(k+1)} = \begin{pmatrix} T & M_1 \\ 0 & M_2 \end{pmatrix},$$

où $T$ est une matrice $k \times k$ *triangulaire supérieure*, et on a la relation

$$A^{(k+1)} = E_k A^{(k)}$$

où

$$E_k = \begin{pmatrix} 1 & & 0 & & & \\ & \ddots & & & 0 & \\ 0 & & 1 & & & \\ & & -l_{k+1,k} & 1 & & 0 \\ & 0 & \vdots & & \ddots & \\ & & -l_{n,k} & 0 & & 1 \end{pmatrix} \begin{matrix} 1 \\ \vdots \\ k \\ k+1 \\ \vdots \\ n \end{matrix}$$

En particulier, la matrice $A^{(n)}$ est triangulaire supérieure, et on a $A^{(n)} = E_{n-1} \ldots E_1 A^{(1)}$. Autrement dit,

$$A = LU, \qquad \text{où} \quad L = (E_{n-1} \ldots E_1)^{-1} \quad \text{et} \quad U = A^{(n)}.$$

La matrice $L$ est triangulaire inférieure ("L" pour *Lower traingular*), et la matrice $U$ est l'inverse d'une matrice triangulaire inférieure, donc est également triangulaire inférieure ("U" pour *Upper triangular*).

A la fin de l'algorithme de Gauss, on a calculé la matrice $U$ et la matrice $L^{-1}$. Il reste ensuite à calculer le produit matrice vecteur $y = L^{-1}b$, puis à résoudre le système triangulaire supérieur par la méthode de remontée.

Au total, cet algorithme nécessite de l'ordre de $2n^3/3$ flops. En effet, en passant de $A^{(k)}$ à $A^{(k+1)}$, on ne modifie que les éléments qui ne sont pas

sur les $k$ premières lignes et les $k$ premières colonnes. Pour chacun de ces éléments, on doit faire une multiplication et une soustraction, soit un total de $2(n-k)^2$ flops. Au total, pour obtenir $A^{(n)}$, il faut donc

$$\sum_{k=1}^{n-1} 2(n-k)^2 = 2\sum_{j=1}^{n-1} j^2 = \frac{(n-1)n(2n-1)}{3} \sim \frac{2n^3}{3}$$

flops. Le produit matrice-vecteur et la résolution du système triangulaire nécessitent seulement de l'ordre de $n^2$ flops.

**Remarque 3.2** *Pour que l'algorithme de Gauss puisse se terminer, il faut que tous les coefficients $a_{kk}^{(k)}$, qui correspondent aux termes diagonaux de la matrice $U$ et qu'on appelle* ***pivots****, soient non nuls.*

Même si la matrice $A$ n'a que des coefficients non nuls, il se peut que l'un des pivots soit nul. Par exemple, on vérifie (exercice !) que si

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{pmatrix}, \qquad \text{on obteint} \qquad A^{(2)} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & -1 \\ 0 & -6 & -12 \end{pmatrix}.$$

Cependant, le résultat suivant permet de caractériser les cas où l'algorithme de Gauss fonctionne.

**Théorème 3.3** *La factorisation $LU$ de la matrice $A$ par la méthode de Gauss existe si et seulement si les sous-matrices principales*

$$A_k = \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix},$$

*obtenues en retirant de $A$ les lignes $k+1, \dots, n$ et les colonnes $k+1, \dots, n$, sont toutes* inversibles. *C'est en particulier le cas si*

(i) *$A$ est symétrique définie positive ;*

(ii) *$A$ est à diagonale dominante par lignes, c.-à-d. pour tout $i \in \{1, \dots, n\}$,*

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|.$$

(ii) *$A$ est à diagonale dominante par colonnes, c.-à-d. pour tout $j \in \{1, \dots, n\}$,*

$$|a_{jj}| > \sum_{i \neq j} |a_{ij}|.$$

*De plus, si $A$ est inversible, sa décomposition $LU$ est unique si l'on impose que les coefficients diagonaux de la matrice $L$ sont tous égaux à 1 (ce qui est le cas de la matrice donnée par l'algorithme de Gauss).*

Notons enfin que, grâce à la factorisation $LU$, on peut calculer le déterminant d'une matrice $n \times n$ avec environ $2n^3/3$ opérations, puisque

$$\det(A) = \det(L)\det(U) = \det(U) = \prod_{k=1}^{n} u_{kk} = \prod_{k=1}^{n} a_{kk}^{(k)}.$$

### 3.1.4 Algorithme de Gauss avec recherche de pivot

Si au cours de l'algorithme de Gauss on trouve un pivot nul à l'étape $k$, le procédé s'arrête. Il est alors possible de permuter la ligne $k$ avec une ligne $r > k$ dont le $k$-ième élément est non nul, avant de poursuivre l'algorithme. Cette opération revient à multiplier la matrice $A^{(k)}$ à droite par la matrice

$$P^{(k,r)} = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 0 & \cdots & \cdots & \cdots & 1 & & \\ & & \vdots & 1 & & & \vdots & & \\ & & \vdots & & \ddots & & \vdots & & \\ & & \vdots & & & 1 & \vdots & & \\ & & 1 & \cdots & \cdots & \cdots & 0 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{pmatrix} \begin{matrix} 1 \\ k-1 \\ k \\ k+1 \\ \vdots \\ r-1 \\ r \\ r+1 \\ n \end{matrix}$$

On obtient alors une factorisation du type

$$U = E_{n-1}P_{n-1}E_{n-2}P_{n-2}\dots E_1P_1A,$$

où les $P_i$ sont soit la matrice identité, soit l'une des matrices $P^{(k,r)}$. La manière de résoudre le système $Ax = b$ demande alors d'effectuer les même opérations sur les lignes et les même échanges de lignes sur le vecteur $b$ que sur la matrice $A$. On obtient alors une suite de vecteurs $b^{(k)}$ telle que

$$E_{k-1}P_{k-1}\dots E_1P_1Ax = b^{(k)}, \qquad \forall k \in \{1, \dots, n\}.$$

Pour $k = n$, on obtient $Ux = b^{(n)}$, qui se résout simplement avec la méthode de remontée.

Il est également possible de choisir une permutation des lignes de la matrice $A$ avant de commencer l'algorithme de Gauss, afin de satisfaire le critère du Théorème 3.3 : on permute d'abord la première ligne avec une autre afin que $\det(A_1) = a_{11} \neq 0$ si nécessaire, puis on permute la seconde ligne avec une des autres qui suivent afin que $\det(A_2) \neq 0$ si nécessaire, et ainsi de suite. On obtient ainsi une matrice $PA$ où est un produit de matrices de la forme $P^{(k,r)}$, à laquelle l'algorithme de Gauss s'applique. La factorisation obtenue est alors

$$LU = PA.$$

Dans ce cas, la résolution du système $Ax = b$ revient simplement à résoudre $LU = Pb$ comme expliqué dans la section 3.1.3.

En pratique, afin d'éviter des erreurs d'arrondi trop importantes, on choisit le plus grand pivot possible (en valeur absolue) à chaque étape de l'algorithme : on choisit comme pivot à l'étape $k$ le nombre $a_{rk}^{(k)}$ avec $r \geq k$ tel que

$$|a_{rk}^{(k)}| = \max_{s \geq k} |a_{sk}^{(k)}|,$$

c.-à-d. qu'on échange les lignes $k$ et $r$ de la matrice $A^{(k)}$.

Si l'on applique pas cette règle, l'existence de pivots très petits augmente énormément les erreurs d'arrondi, comme le montre l'exemple suivant : considèrons le système

$$\begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 2 \end{pmatrix}.$$

avec $\varepsilon \neq 0$ très petit. L'algorithme de Gauss sans recherche de pivot consiste à éliminre la variable $x$ dans la seconde équation. On obtient

$$\begin{pmatrix} \varepsilon & 1 \\ 0 & 1 - \frac{1}{\varepsilon} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 2 - \frac{1}{\varepsilon} \end{pmatrix},$$

soit $y = \frac{1-2\varepsilon}{1-\varepsilon} \approx 1$ en tenant compte des erreurs d'arrondi de l'ordinateur. La valeur de $x$ obtenue par substitution devient $x = \frac{1-y}{\varepsilon} \approx 0$.

Lorsque l'on échange les lignes du système, ce qui revient à choisir le plus grand prmier pivot, on obtient

$$\begin{pmatrix} 1 & 1 \\ \varepsilon & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 & 1 \end{pmatrix}.$$

L'algorithme de Gauss donne alors

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 - \varepsilon \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 & 1 - 2\varepsilon \end{pmatrix},$$

A cause des erreurs d'arrondi de l'ordinateur, on obtient $y \approx 1$, et $x \approx 1$, solution tout à fait acceptable cette fois-ci.

### 3.1.5 Algorithme de Thomas pour les matrices tridiagonales

Dans le cas où la matrice $A$ est tridiagonale (ce qui est le cas pour la discrétisation implicite de l'équation de la chaleur de la section 2.2.3), l'algorithme de Gauss se simplifie notablement. Dans ce cas, l'algorithme s'appelle "algorithme de Thomas".

On suppose que

$$A = \begin{pmatrix} a_1 & c_1 & 0 & \dots & 0 \\ b_2 & a_2 & c_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & b_{n-1} & a_{n-1} & c_{n-1} \\ 0 & \dots & 0 & b_n & a_n \end{pmatrix}$$

Dans ce cas, les matrices $L$ et $U$ de la factorisation $LU$ ont la forme

$$L = \begin{pmatrix} 1 & 0 & \dots & 0 \\ \beta_2 & 1 & \ddots & \vdots \\ & \ddots & \ddots & 0 \\ 0 & & \beta_n & 1 \end{pmatrix} \qquad \text{et} \qquad U = \begin{pmatrix} \alpha_1 & c_1 & & 0 \\ 0 & \alpha_2 & \ddots & \\ \vdots & \ddots & \ddots & c_{n-1} \\ 0 & \dots & 0 & \alpha_n \end{pmatrix},$$

où les inconnues à déterminer sont les $\alpha_i$ et les $\beta_i$. On peut les déterminer en écrivant les coefficients de l'égalité $LU = A$ dans l'ordre $(1,1)$, $(2,1)$, $(2,2)$,...,$(n, n-1)$, $(n,n)$ :

$$\alpha_1 = a_1, \quad \beta_2\alpha_1 = b_2 \Rightarrow \beta_2 = \frac{b_2}{\alpha_1}, \quad \beta_2 c_1 + \alpha_2 = a_2 \Rightarrow \alpha_2 = a_2 - \beta_2 c_1, \quad \dots$$

On obtient finalement

$$\alpha_1 = a_1, \quad \beta_i = \frac{b_i}{\alpha_{i-1}}, \quad \alpha_i = a_i - \beta_i c_{i-1}, \quad \forall i = 2, \dots, n.$$

Avec cet algorithme, le système $Ax = b$ se ramène à résoudre deux systèmes bidiagonaux $Ly = b$ et $Ux = y$ avec

$$\begin{aligned} y_1 &= b_1, \qquad & y_i &= f_i - \beta_i y_{i-1}, \qquad & \forall i = 2, \dots, n, \\ x_n &= \frac{y_n}{\alpha_n}, \qquad & x_i &= \frac{y_i - c_i x_{i+1}}{\alpha_i}, \qquad & \forall i = n-1, \dots, 1. \end{aligned}$$

Le coût total de cet algorithme est de l'ordre de $9n$ flops.

### 3.1.6 Méthode de Choleski

Lorsque la matrice $A$ est symétrique définie positive (comme par exemple pour la discrétisation implicite de l'équation de la chaleur en dimension 1 ou 2, cf. sections 2.2.3 et 2.4.3), on peut améliorer l'algorithme de Gauss avec "l'algorithme de Choleski", basé sur une factorisation plus simple de la matrice $A$, appelée "factorisation de Choleski".

**Théorème 3.4 (factorisation de Choleski)** *Si la matrice $A$ est symétrique définie positive, il existe (au moins) une matrice triangulaire inférieure $B$ telle que*

$$A = B^t B.$$

*Si l'on impose de plus que les coeeficients diagonaux de la matrice $B$ sont positifs, la factorisation $A = B^t B$ est alors unique.*

**Démonstration** La matrice $A$ étant symétrique définie positive, c'est aussi le cas des matrices $A_i$ du théorème 3.3. En particulier, il existe donc unique factorisation $LU$ de la matrice $A$ telle que les éléments diagonaux de $L$ sont tous égaux à 1. De plus, tous les $u_{ii}$ sont strictement positifs puisque

$$\prod_{i=1}^{k} u_{ii} = \det(A_k) > 0, \qquad \forall k \in \{1, \ldots, n\}.$$

Soit $D$ la matrice diagonale dont les éléments diagonaux sont $\sqrt{u_{11}}$, $\sqrt{u_{22}}$, …, $\sqrt{u_{nn}}$. En posant $B = LD$ et $C = D^{-1}U$, on a

$$A = BC = \begin{pmatrix} \sqrt{u_1 1} & 0 & \ldots & 0 \\ \times & \sqrt{u_2 2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \times & \ldots & \times & \sqrt{u_{nn}} \end{pmatrix} \begin{pmatrix} \sqrt{u_{11}} & \times & \ldots & \times \\ 0 & \sqrt{u_{22}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \times \\ 0 & \ldots & 0 & \sqrt{u_{nn}} \end{pmatrix},$$

où les $\times$ représentent des nombres dont la valeur importe peu.

Puisque $A$ est symétrique, on a $BC = {}^tC{}^tC$ et donc $C({}^tB)^{-1} = B^{-1}{}^tC$. Or, la première matrice est triangulaire supérieure avec des 1 sur la diagonale, et la seconde matrice est triangulaire inférieure avec des 1 sur la diagonales. Elles ne peuvent être égales que si elles sont toutes les deux la matrice identité. Ceci implique que $C = {}^tB$ et termine la preuve. □

La mise en place pratique de l'algorithme de Choleski consiste à poser *a priori*

$$B = \begin{pmatrix} b_{11} & 0 & \ldots & 0 \\ b_{21} & b_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ b_{n1} & b_{n2} & \ldots & b_{nn} \end{pmatrix}$$

et à écrire les coeeficients de la relation $A = B{}^tB$ dans l'ordre $(1,1)$, $(1,2)$,…,$(1,n)$, $(2,2)$, $(2,3)$,…,$(2,n)$,…,$(n-1,n-1)$, $(n-1,n)$, $(n,n)$ (la matrice $A$ étant symétrique, il n'est pas nécessaire d'écrire les équations des coefficients $(i,j)$ pour $j < i$) : pour la ligne $i = 1$, on obtient

$$a_{11} = b_{11}^2 \Rightarrow b_{11} = \sqrt{a_{11}}, \quad a_{12} = b_{11}b_{21} \Rightarrow b_{21} = \frac{a_{12}}{b_{11}}, \ldots$$
$$\ldots \quad a_{1n} = b_{11}b_{n1} \Rightarrow b_{n1} = \frac{a_{1n}}{b_{11}}.$$

On a donc calculé la première colonne de la matrice $B$. Par récurrence, si l'on connait les $(i-1)$ premières colonnes de la matrice $B$, on calcule la

$i$-ième colonne en écrivant la $i$-ième ligne de $A = B^t B$ :

$$a_{ii} = b_{i1}^2 + \ldots + b_{ii}^2 \quad \Rightarrow \quad b_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} b_{ik}^2},$$

$$a_{i,i+1} = b_{i1}b_{i+1,1} + \ldots + b_{ii}b_{i+1,i} \quad \Rightarrow \quad b_{i+1,i} = \frac{a_{i,i+1} - \sum_{k=1}^{i-1} b_{ik}b_{i+1,k}}{b_{ii}}, \ldots$$

$$\ldots \quad a_{i,n} = b_{i1}b_{n,1} + \ldots + b_{ii}b_{n,i} \quad \Rightarrow \quad b_{n,i} = \frac{a_{i,n} - \sum_{k=1}^{i-1} b_{ik}b_{n,k}}{b_{ii}}.$$

D'après le théorème 3.4, on peut choisir la matrice $B$ telle que tous ses coefficients diagonaux sont positifs. Ceci assure que toutes les racines carrées dans l'algorithme précédent sont bien définies, ce qui n'était nullement évident *a priori*.

L'algorithme de Choleski consiste à calculer la matrice $B$ par l'algorithme précédent, puis à résoudre les deux systèmes triangulaires $By = b$ et ${}^tBx = y$. On vérifie qu'il nécessite $\frac{n^3}{6}$ flops (plus des termes d'ordre $n^2$ et $n$), ce qui donne un coût deux fois moindre que l'algorithme de Gauss.

On notera enfin que l'algorithme de Choleski permet de calculer le déterminant de $A$ grâce à la formule

$$\det(A) = (b_{11}b_{22}\ldots b_{nn})^2.$$

## 3.2 Méthodes itératives

On cherche maintenant à résoudrte le système $Ax = b$ en construisant une suite $(x^{(k)})_{k\geq 0}$ telle que

$$x = \lim_{k\to+\infty} x^{(k)}.$$

Les méthodes itératives sont souvent plus coûteuses en temps de calcul que les méthodes directes, mais elles ont l'avantage d'être plus faciles à programmer et nécessitent moins de place mémoire (ce qui peut être crucial lorsque $n$ est très grand).

### 3.2.1 Généralités

La stratégies habituelle consiste à construire la suite $(x^{(k)})$ par une relation de récurrence de la forme

$$x^{(k+1)} = Bx^{(k)} + c, \tag{22}$$

où $B$ est la **matrice d'itération** de la méthode itérative, construite à partir de $A$, et $c$ est un vecteur dépendant de $b$. En passant à la limite $k \to +\infty$, on voit que la solution $x$ du système doit satisfaire

$$x = Bx + c.$$

Puisque $x = A^{-1}b$, on voit que nécessairement, $c = (\mathrm{Id} - B)A^{-1}b$, donc la méthode itérative est en fait complètement définie par la matrice $B$.

**Critère de convergence** En définissant l'**erreur au pas** $k$ comme

$$e^{(k)} = x - x^{(k)},$$

il suit de la relation de récurrence que

$$e^{(k+1)} = Be^{(k)}, \qquad \text{soit} \qquad e^{(k)} = B^k e^{(0)}, \qquad \forall k \in \mathbb{N}.$$

On voit donc que $x^{(k)} \to x$ pour tout $x^{(0)}$ ssi $e^{(k)} \to 0$ pour tout $e^{(0)}$ ssi $B^k \to 0$ lorsque $k \to +\infty$. En utilisant le critère classique de convergence des puissances de matrices, on obtient le

**Théorème 3.5** *La méthode itérative (22) converge pour tout choix de $x^{(0)}$ ssi $\rho(B) < 1$, où $\rho(B)$ est le rayon spectral de $B$, défini dans la section 1.2.2. De plus, on a*

$$\lim_{k \to +\infty} \Big[ \sup_{x^{(0)} \ t.q. \ \|x^{(0)}\| \leq 1} \|e^{(k)}\|^{1/k} \Big] = \rho(B).$$

L'équation précédente signifie que la vitesse de convergence de la méthode se comporte en $\rho(B)^k$. C'est donc une convergence *géométrique*, d'autant plus rapide que $\rho(B)$ est petit. Le problème revient donc à choisir une matrice $B$ facile à calculer à partir de $A$, telle que $\rho(B)$ soit aussi petit que possible (la méthode la plus efficace serait bien sûr de choisir $B = 0$ et $c = x$, mais le problème est justement de calculer l'inconnue $x$).

**Test d'arrêt de l'algorithme** L'algorithme n'étant pas exact, il faut définir un critère d'arrêt d'itération en fonction de la prècision souhaitée.

Une première méthode consiste à calculer à chaque itération le **résidu**

$$r^{(k)} = b - Ax^{(k)},$$

et à arrêter l'algorithme au premier pas $k$ tel que

$$\frac{\|r^{(k)}\|_2}{\|b\|_2} \leq \varepsilon,$$

où $\varepsilon > 0$ est un seuil donné.

Dans le cas où la matrice $A$ est symétrique défini positive, ce critère permet d'assurer que l'erreur relative de la méthode est majorée par le produit du seuil $\varepsilon$ et du conditionnement de la matrice $K(A)$. En effet, des inégalités

$$\|e^{(k)}\|_2 = \|A^{-1} r^{(k)}\|_2 \leq \|A^{-1}\|_2 \, \|r^{(k)}\|_2$$

et

$$\|b\|_2 = \|Ax\|_2 \leq \|A\|_2 \, \|x\|_2,$$

on déduit que

$$\frac{\|r^{(k)}\|_2}{\|b\|_2} \leq \varepsilon \quad \Rightarrow \quad \frac{\|e^{(k)}\|_2}{\|x\|_2} \leq K(A)\varepsilon.$$

En pratique, ce critère peut être coûteux, puisqu'il nécessite de calculer à chaque pas d'itération le nouveau produit matrice-vecteur $Ax^{(k)}$. On peut alors utiliser le critère d'arrêt suivant :

$$\frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k)}\|} \leq \varepsilon.$$

En d'autre termes, on arrête l'algorithme lorsque l'écart relatif entre deux pas d'itération est petit. Si ce critère ne nécessite pas de calculer $r^{(k)}$ à chaque pas de temps, il ne garantit pas un contrôle de l'erreur relative de la méthode. Il est donc moins coûteux, mais moins fiable.

**Le splitting** Une technique générale pour construire la matrice $B$ est basée sur une décomposition (*splitting*) de la matrice $A$ sous la forme

$$A = P - N,$$

avec $P$ inversible et *facile à inverser* (par exemple diagonale ou triangulaire). La matrice $P$ est appelée *matrice de conditionnement*. L'algorithme itératif s'écrit alors

$$Px^{(k+1)} = Nx^{(k)} + b, \qquad \text{soit} \qquad x^{(k+1)} = P^{-1}Nx^{(k)} + P^{-1}b.$$

La matrice d'itération de la méthode est donc

$$B = P^{-1}N = \mathrm{Id} - P^{-1}A.$$

En pratique, les *splittings* les plus classiques sont basés sur l'écriture

$$\begin{array}{lcl} A & = & D - E - F, \quad \text{où} \\ D & : & \text{diagonale de } A, \\ E & : & \text{triangulaire inférieure avec des 0 sur la diagonale,} \\ F & : & \text{triangulaire supérieure avec des 0 sur la diagonale.} \end{array}$$

### 3.2.2 Les méthodes de Jacobi, Gauss-Seidel et SOR

**La méthode de Jacobi** Elle consiste à choisir le splitting "le plus simple" avec $P = D$ et $N = E + F$. On obtient

$$x^{(k+1)} = D^{-1}(E + F)x^{(k)} + D^{-1}b.$$

On appelle *matrice de Jacobi* la matrice d'itération de la méthode

$$J = D^{-1}(E + F) = \mathrm{Id} - D^{-1}A.$$

La mise en place de l'algorithme correspondant consiste à calculer

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij} x_j^{(0)} \right), \qquad \forall i = 1, 2, \ldots, n.$$

La mise en œuvre de l'algorithme nécessite de une place mémoire de $2n$ nombre flotants (en plus de la matrice $A$ et du vecteur $b$), puisqu'il nécessite de stocker les deux vecteurs $x^{(k)}$ et $x^{(k+1)}$.

**La méthode de Gauss-Seidel** Elle consiste à prendre le splitting $P = D - E$ et $N = F$. La matrice d'itération associée est alors donnée par

$$B_{GS} = (D - E)^{-1}F.$$

La matrice $P$ est triangulaire inférieure, donc facile à inverser. Toutefois, il n'est en fait pas nécessaire de calculer explicitement son inverse pour mettre en place l'algorithme, puisqu'on vérifie que le vecteur $x^{(k+1)}$ est donné par les formules suivantes :

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_i^{(k)} \right), \qquad \forall i = 1, 2, \ldots, n.$$

En comparant avec la méthode de Jacobi, on voit que cette méthode revient à se servir des coordonnées de $x^{(k+1)}$ déjà calculées afin d'obtenir la coordonnée suivante.

En terme de place mémoire, il n'est nécessaire que de stocker un seul vecteur de flottants, $x^{(k)}$ étant remplacé par $x^{(k+1)}$ au cours de l'itération.

En général, cette méthode converge plus rapidement que celle de Jacobi (voir section 3.2.3). Elle est donc préférable.

**La méthode de relaxation, ou SOR (Successive Over Relaxation)** On cherche ici à améliorer la méthode de Gauss-Seidel lorsqu'elle converge, en introduisant un poids sur la diagonale de $A$ dans la splitting. On introduit un paramètre réel $\omega \neq 0$, et on considère le splitting

$$A = \left( \frac{D}{\omega} - E \right) - \left( \frac{1 - \omega}{\omega} D + F \right).$$

La matrice d'itération correspondante s'appelle *matrice de relaxation*

$$B_\omega = \left( \frac{D}{\omega} - E \right)^{-1} \left( \frac{1 - \omega}{\omega} D + F \right) = (D - \omega E)^{-1}[(1 - \omega)D + \omega F].$$

Elle coincide avec la méthode de Gauss-Seidel lorsque $\omega = 1$. L'idée de cette définition est de choisir $\omega > 1$ (*sur-relaxation*) ou $\omega < 1$ (*sous-relaxation*) afin de diminuer $\rho(B_\omega)$ et donc d'accélérer la méthode. Dans les bons cas (par exemple le cas tridiagonal, voir section 3.2.3), il est même possible de déterminer le paramètre $\omega_0$ optimal.

De nouveau, il est inutile de calculer explicitement la matrice $B_\omega$ pour mettre en œuver l'algorithme : on vérifie que le vecteur $x^{(k+1)}$ est donné par les formules suivantes

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij} x_j^{(k)} \right), \qquad \forall i = 1, 2, \ldots, n.$$

**Remarque 3.6** *Lorsque la matrice $A$ a une décomposition par blocs agréable (par exemple si elle est tridiagonale par blocs), il est possible d'étendre les méthodes précédentes en choisissant pour $D$ la matrice diagonale par blocs constituée des blocs diagonaux de la matrice $A$. Dans ce cas, les méthodes de Jacobi, Gauss-Seidel et SOR par blocs sont définies par les mémes formules de récurrence, où $a_{ij}$ est remplacé par le bloc $(i, j)$ de la matrice $A$. On s'attend alors à une amélioration de la vitesse de convergence. Cependant, ces méthodes nécessitent de calculer les inverses des $n$ blocs diagonaux de $A$. La méthode est intéressante si le coût de ces calculs est suffisamment compensé par l'accélération de la convergence.*

### 3.2.3 Convergence des méthodes de Jacobi, Gauss-Seidel et SOR

Nous commençons par deux résultats généraux avant de donner le paramètre de relaxation optimal dans les cas des matrices tridiagonales par blocs.

#### Cas des matrices à diagonale dominantes par lignes

**Théorème 3.7** *Si la matrice $A$ est à diagonale dominante par lignes, alors les méthodes de Jacobi et de Gauss-Seidel sont convergentes.*

**Démonstration** Nous ne prouverons ici que le cas de la méthode de Jacobi. On a

$$x^{(k+1)} - x = \frac{-1}{a_{ii}} \sum_{j \neq i} a_{ij} (x_j^{(n)} - x_j),$$

donc

$$\begin{aligned} \max_{1 \leq i \leq n} |x_i^{(k+1)} - x_i| &\leq \frac{1}{a_{ii}} \sum_{j \neq i} |a_{ij}| \max_{1 \leq l \leq n} |x_l^{(k)} - x_l| \\ &\leq \alpha \max_{1 \leq l \leq n} |x_l^{(k)} - x_l|, \end{aligned}$$

où la constante $\alpha \in ]0,1[$ est donnée par l'hypothèse de diagonale-dominance de $A$. On en déduite que

$$\|x^{(k+1)} - x\|_\infty \leq \alpha \|x^{(k)} - x\|_\infty, \qquad \text{soit} \qquad \|x^{(k)} - x\|_\infty \leq \alpha^k \|x^{(0)} - x\|_\infty.$$

Puisque $\alpha < 1$, on a prouvé la convergence de la méthode. □

**Cas des matrices symétriques définies positives** Le résultat suivant est admis.

**Théorème 3.8** *Si la matrice $A$ est symétrique définie positive, alors les méthodes de Gauss-Seidel et SOR pour $0 < \omega < 2$ convergent (la méthode de Jacobi, pas forcément).*

**Cas des matrices tridiagonales et tridiagonales par blocs** De nouveau, les résultats suivants seront admis sans démonstration.

**Théorème 3.9** *Si la matrice $A$ est tridiagonale, alors*

$$\rho(B_{GS}) = \rho(J)^2.$$

*Si la matrice $A$ est tridiagonale par blocs, on a la même relation entre les matrices d'itérations des méthodes de Gauss-Seidel et de Jacobi* par blocs.

Ce résultat montre que, lorsque la matrice $A$ est tridiagonale ou tridiagonale par blocs et que la méthode de Jacobi converge, alors la méthode de Gauss-Seidel convegre toujours plus vite, puisque $\rho(J)^2 < \rho(J)$ dès que $\rho(J) < 1$.

**Théorème 3.10** *Si la matrice $A$ est tridiagonale et symétrique définie positive, il existe un unique paramètre de relaxation optimal*

$$\omega_0 = \frac{2}{1 + \sqrt{1 - \rho(J)^2}} \in ]1,2[,$$

*tel que*

$$\rho(B_{\omega_0}) = \inf_{0<\omega<2} \rho(B_\omega) = \omega_0 - 1 < \rho(B_1) = \rho(J)^2 < \rho(J).$$

*Si la matrice $A$ est tridiagonale par blocs et symétrique définie positive, on a le même résultat pour les méthodes de Jacobi, Gauss-Seidel et SOR* par blocs.

On a donc une caractérisation explicite de paramètre de relaxation optimal dans le cas des matrices tridiagonales (par blocs ou non) et symétriques définies positives.

**Remarque 3.11** *Ce cas apparaît naturellement dans les discrétisations implicites d'EDP d'évolution par différences finies, par exemple dans le cas de l'équation de la chaleur en dimensions 1 ou plus (cf. sections 2.2.3 et 2.4.3).*

## 3.3 Méthodes de gradient

On s'intéresse maintenant à une autre classe de méthodes itératives, appelées *méthodes de descente* ou *de gradient*. Elles supposent que la matrice $A$ est *symétrique définie positive*. On le supposera dans toute cette section. Ces méthodes sont généralement particulièrement bien adaptées aux cas où la matrice $A$ est *creuse* (voir section 1.2.2).

### 3.3.1 Généralités

Ces méthodes sont basées sur des itérations de la forme

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)},$$

où $p^{(k)}$ est la *direction de descente* choisie à l'étape $k$, et $\alpha_k$ et le *pas de descente* choisi à l'étape $k$.

Ces méthodes s'appliquent à des problèmes plus généraux que la résolution de systèmes linéaire. Elles sont par exemple utiulisées pour chercher le minimum d'une fonctionnelle $F : \mathbb{R}^n \to \mathbb{R}$. Le vecteur $p^{(k)}$ est alors souvent choisi comme le *gradient* de $F$ en $x^{(k)}$ (d'où le nom de *méthode de gradient*) et le pas $\alpha_k$ est souvent choisi afin de minimiser $F$ dans la direction $p^{(k)}$ (on dit alors que la méthode est *à pas optimal*).

Dans le cas d'un système linéaire symétrique défini positif, trouver la solution revient à minimiser la fonctionnelle

$$F(y) = {}^t yAy - 2b \cdot y,$$

où $\cdot$ représente le produit scalaire.

En effet, cette fonctionnelle est strictement convexe puisque la matrice $A$ est définie positive, et donc son unique minimum est obtenue en annulant son gradient. Or

$$\nabla F(y) = 2(Ay - b),$$

s'annule lorsque $y$ est la solution du système $Ax = b$. On va donc décrire les méthodes de descentes appliquées à cette fonctionnelle.

Remarquons que minimiser $F$ est équivalent à minimiser la fonctionnelle

$$E(y) = {}^t(y - x)A(y - x),$$

puisque

$$E(y) = {}^y Ay - 2{}^t yAx + {}^x Ax = F(y) + x \cdot b,$$

et le nombre $x \cdot b$ est une constante indépendante de $y$.

Signalons enfin qu'on utilise classiquement pour ces algorithmes les mêmes critères d'arrêt que pour les autres méthodes itératives (voir section 3.2.1).

### 3.3.2 Méthode du gradient à pas optimal

La méthode du gradient à pas optimal consiste à choisir $p^{(k)} = \nabla F(x^{(k)}) = \nabla E(x^{(k)})$, c.-à-d.

$$p^{(k)} = -2r^{(k)},$$

où $r^{(k)} = b - Ax^{(k)}$ est le *résidu* à l'étape $k$. Ensuite, il s'agit de choisir le pas $\alpha_k$ optimal, c.-à-d. celui minimisant la fonctionnelle $E$ dans la direction $p^{(k)}$. Or on a

$$E(x^{(k)} + \alpha p^{(k)}) = E(x^{(k)}) - 2\alpha r^{(k)} \cdot p^{(k)} + \alpha^{2t} p^{(k)} A p^{(k)},$$

qui est un trinôme du second degré en $\alpha$. Le minimum est donc donné par

$$\alpha_k = \frac{r^{(k)} \cdot p^{(k)}}{{}^t p^{(k)} A p^{(k)}}.$$

On a églament les propriétés (triviales à vérifier, exercice !)

$$r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)} \qquad \text{et} \qquad p^{(k)} \cdot r^{(k+1)} = 0.$$

L'algorithme finalement obtenu est le suivant (après division de $p^{(k)}$ par $-2$ et multiplication de $\alpha_k$ par $-2$) :

$$x^{(k+1)} = x^{(k)} + \beta_k r^{(k)}, \qquad \text{avec} \qquad \beta_k = \frac{\|r^{(k)}\|_2^2}{{}^t r^{(k)} A r^{(k)}}. \tag{23}$$

La vitesse de convergence de la méthode est donnée par le

**Théorème 3.12** *La suite $(x^{(k)})_{k\geq 0}$ définie par la méthode du gradient à pas optimal (23) vérifie*

$$\|x^{(k)} - x\|_2 \leq K(A) \left( \frac{K(A) - 1}{K(A) + 1} \right)^2 \|x^{(0)} - x\|_2, \qquad \forall k \geq 0.$$

Ce résultat signifie que plus $K(A)$ est proche de 1, plus vite la méthode converge.

### 3.3.3 Méthode du gradient conjugué

La méthode du gradient conjugué consiste à rechercher également à chaque étape à optimiser le choix de la direction de descente $p^{(k)}$. Le calcul précédent a montré que, quel que soit le choix de $p^{(k)}$, choisir le pas optimal $\alpha_k$ dans la direction $p^{(k)}$ conduit à la relation

$$p^{(k)} \cdot r^{(k+1)} = 0.$$

On peut alors rechercher $p^{(k+1)}$ dans le plan formé par les deux directions orthogonales $p^{(k)}$ et $r^{(k+1)}$.

Le calcul donne alors l'algorithme

Soit $x_0$ quelconque, par exemple $x_0 = 0$, et $p^{(0)} = r^{(0)} = b - Ax^{(0)}$,

Pour $k = 0, 1, \ldots,$

$$\alpha_k = \frac{\|r^{(k)}\|_2^2}{{}^t p^{(k)} A p^{(k)}}, \qquad x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \qquad r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}$$

$$\beta_{k+1} = \frac{\|r^{(k+1)}\|_2^2}{\|r^{(k)}\|_2^2}, \qquad p^{(k+1)} = r^{(k+1)} + \beta_{k+1} p^{(k)}.$$

On a le résultat suivant sur la vitesse de convergence

**Théorème 3.13** *La méthode du gradient conjugué est exacte au bout de $n$ itérations. De plus, on a*

$$\|x^{(k)} - x\|_2 \leq 2K(A) \left( \frac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1} \right)^k \|x^{(0)} - x\|_2, \qquad \forall k \in \{1, \ldots, n\}.$$

En d'autres termes, la méthode du gradient conjugué converge toujours plus vite (au sens de la vitesse de convegrence géométrique) que la méthode du gradient avec pas optimal. En effet, on a toujours $\sqrt{K(A)} < K(A)$ puisque $K(A) > 1$.

### 3.3.4 Méthodes de gradient préconditionnées

Les deux méthodes que l'on vient de présenter sont d'autant plus efficaces que le conditionnement de la matrice $A$ est proche de 1. C'est pourquoi il est souvent intéressant de *pré-conditionner* la matrice avant de lancer l'algorithme, c.-à-d. de remplacer $A$ par la matrice $P^{-1}A$ de telle sorte que $K(P^{-1}A)$ soit plus petit que $K(A)$. On obtient alors le système $P^{-1}Ax = P^{-1}b$. Le problème est que la matrice $P^{-1}A$ n'est pas nécessairement symétrique. Pour résoudre ce problème, on suppose que $P$ est *symétrique définie positive*, et on considère la matrice symétrique définie positive $P^{1/2}$ telle que $P^{1/2}P^{1/2} = P$ (si $P$ est diagonale, il suffit de prendre la matrice diagonale des racines carrées des élements diagonaux de $P$). Or la matrice

$$P^{1/2}(P^{-1}A)P^{-1/2} = P^{-1/2}AP^{-1/2} = \tilde{A}$$

est symétrique définie positive. Le système $Ax = b$ est équivalent au système

$$\tilde{A}P^{1/2}x = P^{-1/2}b.$$

En posant $\tilde{x} = P^{1/2}x$ et $\tilde{b} = P^{-1/2}b$, on doit alors résoudre le système

$$\tilde{A}\tilde{x} = \tilde{b}, \tag{24}$$

pour lequel on peut utiliser la méthode du gradient avec pas optimal ou la méthode du gradient conjugué.

**La méthode du gradient préconditionné avec pas optimal** On vérifie que la méthode du gradient conjugué appliquée au système (24) revient à calculer $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$, où

$$p^{(k)} = P^{-1} r^{(k)} \qquad \text{et} \qquad \alpha_k = \frac{r^{(k)} \cdot p^{(k)}}{{}^t p^{(k)} A p^{(k)}}. \tag{25}$$

On obtient alors la vitesse de convergence suivante (à comparer avec le théorème 3.12) :

$$\|x^{(k+1)} - x\|_2 \leq K(A) \left( \frac{K(P^{-1}A) - 1}{K(P^{-1}A) + 1} \right)^k \|x^{(0)} - x\|_2,$$

ce qui donne bien un gain lorsque $K(P^{-1}A) < K(A)$.

**La méthode du gradient conjugué préconditionné** La méthode du gradient conjugué appliquée au système (24) revient à applique l'algorithme suivant :

Soit $x^{(0)}$ quelconque, par exemple $x^{(0)} = 0$, $r^{(0)} = b - Ax^{(0)}$,
$z^{(0)} = P^{-1} r^{(0)}$, $p^{(0)} = z^{(0)}$,

et pour $k \geq 0$, on applique les formules

$$\alpha_k = \frac{r^{(k)} \cdot p^{(k)}}{{}^t p^{(k)} A p^{(k)}}, \qquad x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \qquad r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)},$$

$$z^{(k+1)} = P^{-1} r^{(k+1)}, \qquad \beta_{k+1} = \frac{{}^t p^{(k)} A z^{(k+1)}}{{}^t p^{(k)} A p^{(k)}}, \qquad p^{(k+1)} = z^{(k+1)} - \beta_k p^{(k)}.$$

Dans ce cas, on obtient la vitesse de convergence suivante (à comparer avec le théorème 3.13) :

$$\|x^{(k+1)} - x\|_2 \leq 2K(A) \left( \frac{\sqrt{K(P^{-1}A)} - 1}{\sqrt{K(P^{-1}A)} + 1} \right)^k \|x^{(0)} - x\|_2.$$

**Quelques exemples de préconditionnement** On peut obtenir facilement des préconditionnements avec la méthode de *splitting* décrite dans la section 3.2.1. Si $A = P - N$ avec $P$ symétrique définie positive (par exemple si $P = D$, la diagonale de $A$), la méthode itérative correspondante (Jacobi si $P = D$) s'écrit

$$Px^{(k+1)} = Nx^{(k)} + b, \qquad \text{soit} \qquad P(x^{(k+1)} - x^{(k)}) = r^{(k)}.$$

La méthode du gradient préconditionné avec pas optimale revient alors à généraliser cette méthode comme

$$P(x^{(k+1)} - x^{(k)}) = \alpha_k r^{(k)}.$$

Si $\alpha_k$ est choisi comme dans (25), on retrouve la méthode du gradient préconditionné à pas optimal. En particulier, la méthode du gradient à pas optimal avec matrice de préconditionnement $P = D$ converge plus vite que la méthode de Jacobi.

Un autre préconditionnement souvent utilisé est *le préconditionnement SSOR d'Evans*. En utilisant la décomposition habituelle

$$A = D - E - {}^tE,$$

on fixe un paramètre $\omega \in ]0, 2[$ et on choisit

$$P = \frac{1}{\omega(2-\omega)}(D - \omega E)D^{-1t}(D - \omega E).$$

On constate que $P$ est obtenue directement en fonction de $A$ : aucun calcul ni stockage n'est nécessaire.

A titre d'exemple, lorsque la matrice $A$ est celle obtenue par discrétisation implicite de l'équation de la chaleur en dimension 2 dans un carré (voir section 2.4.3), le choix optimal du paramètre $\omega$ conduirait à un conditionnement $K(P^{-1}A)$ de l'ordre de $1/h$, alors que $K(A)$ est de l'ordre de $1/h^2$. En pratique, le choix $\omega = 1$ permet généralement un gain important en terme de vitesse de convergence.

# 4 Exercices

## 4.1 Exercice 1

On considère le problème

$$\begin{cases} \frac{\partial u}{\partial t}(t,x) - \frac{\partial^2 u}{\partial x^2}(t,x) = 0, & \text{pour } t \in ]0,T[ \text{ et } x \in ]0,1[, \\ u(0,x) = u_0(x), & \text{pour } x \in ]0,1[, \\ u(t,0) = 0 \text{ et } u(t,1) = 0, & \text{pour } t \geq 0. \end{cases} \tag{26}$$

1. Chercher sous quelles conditions sur $u_0$ on a une solution à (26) de la forme $u(t,x) = f(t)g(x)$, et calculer $u(t,x)$.
   *Indication* : on se souviendra que les fonctions sin et cos sont solutions de l'équation différentielle $y'' + y = 0$.
   On choisira dans la suite l'une de ces fonctions $u_0$ pour tester la convergence des schémas de différences finies.
2. Implémenter le schéma d'approximation explicite par différences finies de l'EDP (26) avec $N = 10$ et $M = 100$, 200, 400. On veillera à mobiliser un minimum de mémoire. Comparer les résultats avec la solution exacte et commenter.
   *Indications* : l'algorithme consiste simplement à calculer des produits matrice-vecteur de la forme $U^{(j+1)} = AU^{(j)}$ avec $A$ et $U^{(0)}$ à déterminer ; il est inutile de stocker les vecteurs $U^{(j)}$ dans un tableau ; on se souviendra de la condition de CFL.

3. Implémenter le schéma d'approximation implicite par différences finies de l'EDP (26) avec $N = 10$ et $M = 10$, 100, 200, 400. On calculera l'inverse de la matrice par l'algorithme de Thomas, puis par l'algorithme de Gauss-Seidel. Comparer les résultats avec la solution exacte et ceux obtenus à la question 2.
   *Indications* : de nouveau, l'algorithme consiste à calculer des produits matrice-vecteur de la forme $U^{(j+1)} = AU^{(j)}$ où $A$ est l'inverse d'une matrice, à calculer avec l'algorithme de Thomas et la méthode de Gauss-Seidel.

### 4.2 Exercice 2

On considère maintenant le problème

$$\begin{cases} \frac{\partial u}{\partial t}(t,x) - \frac{\partial^2 u}{\partial x^2}(t,x) - \frac{\partial u}{\partial x}(t,x) = 1, & \text{pour } t \in ]0,T[ \text{ et } x \in ]0,1[, \\ u(0,x) = u_0(x), & \text{pour } x \in ]0,1[, \\ u(t,0) = 1 \text{ et } u(t,1) = 1, & \text{pour } t \geq 0. \end{cases} \tag{27}$$

1. Chercher une solution particulière polynômiale en $x$ à cette EDP, puis déduire de la question 1 de l'exercice 1 une famille de solutions de (27). Choisir l'une de ces solutions pour les tests numériques des questions suivantes.
2. Répondre aux questions 2 et 3 de l'exercice précédent dans le cas de l'EDP (27).
   *Indications* : il faudra faire un choix pour la discrétisation de $\frac{\partial u}{\partial x}$, se reporter à la section 2.4.2 ; pour traiter les conditions aux bords non nulles, se reporter à la section 2.4.1.

## Références


[1] P.G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. Masson (1982).

[2] P. Lascaux et R. Théodor. *Analyse numérique matricielle appliquée à l'art de l'ingénieur*. Tome 1. Masson (1986).

[3] P. Lascaux et R. Théodor. *Analyse numérique matricielle appliquée à l'art de l'ingénieur*. Tome 2. Masson (1987).

[4] B. Lucquin. *Equations aux dérivées partielles et leurs approximations*. Ellipse (2004).