Deuxième année
FICM

Département Énergie & Fluides

# Module EFS7AD

# Méthodes Numériques pour la Mécanique-Énergétique

# - Cours et TD -

**version 1.2.2**

Mathieu Jenny

Année universitaire 2017 – 2018

# Table des matières

Chapitre 1

# Introduction aux méthodes numériques

## 1.1 Objectifs

L'objectif de ce cours est d'acquérir les bases nécessaires au développement d'un code numérique permettant de résoudre un problème physique décrit par des équations aux dérivées partielles continues. Les problèmes de mécaniques des fluides et de thermiques sont décrits par de telles équations. Ces bases servent aussi, et ce sera d'ailleurs leur principal intérêt pour l'ingénieur, à faire les bons choix lors de l'utilisation d'un code commercial de simulation. Enfin, la connaissance des méthodes numériques permet de mieux cerner ce qu'on peut en attendre.

Évidemment, nous n'avons pas la prétention d'être exhaustif sur les méthodes numériques existantes, d'autant que cela constitue encore aujourd'hui un domaine de recherche actif.

## 1.2 Recherche numérique de zéros

Avant de s'attaquer à la résolution d'équations différentielles, nous allons nous pencher sur un problème plus simple, qui, la suite le montrera, se révèlera utile et même central pour la résolution des problèmes aux dérivées partielles.

Nous allons donc essayer de résoudre numériquement une équation algébrique de type $f(x) = 0$, où $f$ est une fonction connue et $x$ l'inconnue à déterminer. Évidemment, on suppose que $f$ admet au moins un zéros $\alpha$ dans son domaine de définition (les nombres réels par exemple). Deux cas sont à distinguer : soit une solution analytique et explicite peut être trouvée, soit cette solution est inaccessible analytiquement. L'utilisation d'une méthode numérique (pas d'une calculatrice) ne se justifie pleinement que dans le deuxième cas. En effet, la plupart du temps, la solution numérique n'est qu'une approximation de la solution exacte. Cela dit, on peut approcher cette dernière autant que l'on veut si l'algorithme numérique converge bien vers la solution. Ce dernier point est en général non garanti si l'on ne choisit pas le bon algorithme de résolution.

### 1.2.1 Ordre de convergence

Soit $x_n$ une suite qui admet une limite $\alpha$ quand $n \to +\infty$. Alors, s'il existe un nombre réel $q > 0$ tel que :

$$\lim_{n\to+\infty} \left( \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^q} \right) = \xi \tag{1.1}$$

avec $\xi \geq 0$, on dit que la vitesse de convergence de $x_n$ est d'ordre $q$.

Une version plus pratique de la définition de l'ordre de convergence est de dire que $\exists C > 0$ tel que $\forall n \in \mathbb{N}$,

$$\frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^q} \leq C \tag{1.2}$$

$C$ est appelé le facteur de convergence.

Cette notion d'odre de convergence est importante car elle caractérise l'efficacité d'un algorithme. En effet, en général, on arrête les calculs pour $N > 0$ tel que $|x_N - \alpha| < \varepsilon$ avec $\varepsilon$ arbitrairement petit. Plus l'ordre est élevé, moins le nombre d'itérations $N$ nécessaire à l'approximation de la solution $\alpha$ sera grand. Si $q = 1$, alors on dit que la convergence est linéaire, pour $q = 2$, on dit qu'elle est quadratique, *etc* ...

### 1.2.2 Méthode de dichotomie

La méthode de dichotomie est l'une des méthodes les plus basiques pour rechercher un zéro de fonction. C'est aussi l'une des plus robustes.

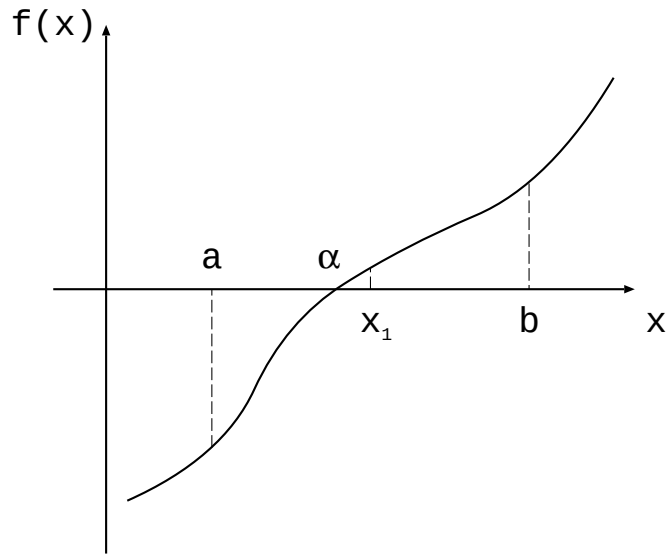


FIGURE 1.1 – Recherche de zéro par dichotomie.

**THÉORÈME 1.** *Soit $f$ une fonction continue sur un intervalle $I = [a,\ b]$, alors $f$ prend toutes les valeurs possibles entre $f(a)$ et $f(b)$ sur l'intervalle $I$.*

**Corollaire 1.** *Si $f(a)f(b) < 0$, alors, $\exists \alpha \in I$ tel que $f(\alpha) = 0$.*

C'est sur ce corollaire 1 que se base la méthode de dichotomie. On propose pour approcher $\alpha$, $x_1 = (a+b)/2$. L'erreur est majorée par $(b-a)/2$. On vérifie le signe de $f(x_1)$ et on recommence avec $I_1 = [a,\ x_1]$ (comme sur la figure 1.1) ou $I_1 = [x_1,\ b]$.

Par récurrence, on obtient une majoration de l'erreur absolue à l'itération $n$ :

$$|x_n - \alpha| \leq \frac{b-a}{2^n} \tag{1.3}$$

On a donc :

$$\frac{|x_{n+1} - \alpha|}{|x_n - \alpha|} \leq \frac{1}{2} \tag{1.4}$$

Cela permet de dire que la convergence est linéaire ($q = 1$). De plus, c'est une des rares méthodes où l'on peut prédire le nombre d'itérations $N$ nécessaires à l'obtention d'une précision $\varepsilon$ donnée. On a :

$$\boxed{2^N > \frac{b-a}{\varepsilon}} \tag{1.5}$$

### 1.2.3 Méthode de Newton

La méthode de Newton a été décrite par Isaac Newton dans *De analysi per aequationes numero terminorum infinitas* (Newton, 1669) et *De metodis fluxionum et serierum infinitarum* (Newton, 1671). Cependant, il n'applique sa méthode qu'à la recherche des racines d'un polynôme. La formulation moderne permet de trouver le zéro d'une multitude de fonctions. C'est une méthode très efficace et très largement utilisée (dans vos calculatrices par exemple) mais nous allons voir qu'elle n'est tout de même pas universelle et que la convergence vers un zéro de la fonction n'est pas toujours garantie même si ce dernier existe. Il faut prendre quelques précautions.

On considère une fonction $f$ de $\mathbb{R}^m \to \mathbb{R}^m$, $C^1$ et dérivable à l'ordre 2 sur un domaine $I$ de $\mathbb{R}^m$. On note $f'(x_0)$ sa dérivée (ou le Jacobien si $m > 1$) de $f$ au point $x_0 \in I$. Dans la suite, on considère le cas où $m = 1$ (on travaille avec des nombres réels). Cependant, les résultats peuvent s'étendre à $m > 1$.

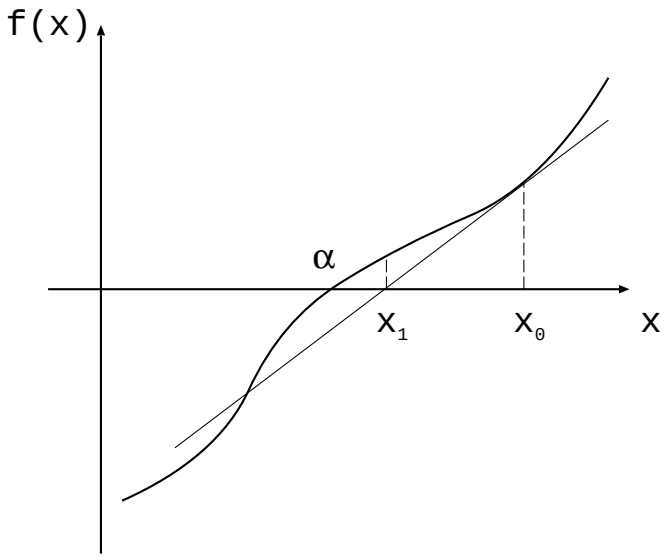


FIGURE 1.2 – Recherche de zéro par méthode de Newton.

La méthode repose sur la formule de Taylor qui peut alors s'appliquer :

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + f''(\zeta)\frac{(x - x_0)^2}{2} \tag{1.6}$$

avec $x \in I$ et $\zeta \in ]x_0, x[$ (ou $\zeta \in ]x, x_0[$). Si on néglige le terme d'ordre 2, ce qui se justifie si $x$ est proche de $x_0$, on obtient l'approximation suivante de la fonction $f$ au point $x$ :

$$f(x) \simeq f(x_0) + f'(x_0)(x - x_0) \tag{1.7}$$

Si on remplace $f$ par notre relation 1.7 dans l'équation « trouvez $x$ tel que : »

$$f(x) = 0 \tag{1.8}$$

on obtient une approximation $x_1$ de la solution $\alpha$ de l'équation 1.8 :

$$x_1 = x_0 - f(x_0)/f'(x_0) \tag{1.9}$$

si $f'(x_0) \neq 0$.

On peut recommencer la procédure suivant la relation de récurence suivante pour $n \in \mathbb{N}$ :

$$\boxed{x_{n+1} = x_n - f(x_n)/f'(x_n)} \tag{1.10}$$

en prenant garde à ce que $f'(x_n) \neq 0$.

A ce stade, on peut faire plusieurs remarques :

— Si $x_0 = \alpha$, alors $\forall n \in \mathbb{N}$, $x_n = \alpha$ (point fixe).

— Si $f$ est une fonction afine, alors on trouve la solution exacte dès la première itération ($x_1 = \alpha$).

— Dans les autres cas, il faut se demander si $x_{n+1}$ est plus proche de la solution $\alpha$ de l'équation que ne l'est $x_n$. Cela revient à se demander si la suite $x_n$ converge. C'est ce point que nous allons développer.

Prenons $I = [a,\ b]$, un intervalle autour de la solution $\alpha$ tel que $f'$ ne change pas de signe (donc ne s'annule pas). On suppose que $f'(x) > 0$ pour $x \in I$ et que $x_n \in I$. Pour savoir si la suite $x_n$ converge

vers $\alpha$, on va étudier l'erreur $e_n = |x_n - \alpha|$. En utilisant la formule de Taylor 1.6 et le fait que $f(\alpha) = 0$, on obtient :

$$f(x_n) = -f'(x_n)(\alpha - x_n) - \frac{f''(\zeta_n)(\alpha - x_n)^2}{2} \tag{1.11}$$

avec $\zeta_n \in ]\alpha,\ x_n[$ (ou $\zeta_n \in ]x_n,\ \alpha[$). On obtient alors la relation de récurrence pour $e_n$ :

$$e_{n+1} = e_n^2 \frac{|f''(\zeta_n)|}{2|f'(x_n)|} \tag{1.12}$$

Par définition, on a $e_n \geq 0$, $\forall n \in \mathbb{N}$. De plus, on pose :

$$M = \max_{x \in I}(|f''(x)|) \tag{1.13}$$

$$m = \min_{x \in I}(|f'(x)|) \tag{1.14}$$

Comme $m > 0$, on peut poser $K = M/2m$,

$$e_{n+1} \leq K e_n^2 \tag{1.15}$$

Ainsi, $e_{n+1}$ converge de façon quadratique vers le point fixe 0. Cela revient à dire qu'il existe un voisinage de $\alpha$ pour lequel la convergence de la méthode de Newton est quadratique si, dans ce voisinage $I$, la fonction $f$ est $C^1$, $f'(x) \neq 0$ et $f'$ est dérivable ($M$ est fini). Si de plus on a $f''.f > 0$ sur $I$ alors la suite $x_n$ est monotone et converge vers $\alpha$.

Si $f'(\alpha) = 0$, on peut montrer que la méthode est linéairement convergente dans un voisinage de $\alpha$ si $f'(x) \neq 0$ pour $x \neq \alpha$.

Ces critères de convergence sont locaux et la taille du voisinage n'est pas déterminée à priori. De plus, si $f$ admet plusieurs zéros, la méthode va converger vers l'un des zéros proche de $x_0$. On ne peut donc pas l'utiliser pour trouver toutes les racines d'un polynôme par exemple.

Dans les autres cas, la convergence n'est pas assurée. En pratique, les logiciels comme Mathematica ou Matlab utilisent plusieurs méthodes. Le code analyse un peu le comportement de $f$ avant de sélectionner automatiquement un algorithme, voir une palette de méthodes (méthodes hybrides). On peut notamment utiliser la méthode de dichotomie pour déterminer $x_0$, la valeur initiale de la méthode de Newton. Cette dernière converge beaucoup plus rapidement qu'une dichotomie pour obtenir un résulat précis.

### 1.2.4 Méthode des sécantes

La méthode des sécantes est très proche de la méthode de Newton. Elle consiste à remplacer $f'(x_n)$ par :

$$f'(x_n) \simeq \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \tag{1.16}$$

La relation de récurence devient alors :

$$\boxed{x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}} \tag{1.17}$$

L'intérêt de cette méthode est évident : on n'a pas besoin de connaitre $f'$ pour l'utiliser. Cependant, afin d'initialiser la relation de récurrence, il faut prendre deux points assez proches $x_0$ et $x_1$. Dans ce cas, l'ordre de convergence est en général $q = (1 + \sqrt{5})/2$ (le nombre d'or). La méthode est donc, a priori,

un peu moins efficace que la méthode de Newton. Cependant, en pratique, elle se révèle plus efficace. En effet, évaluer $f(x_n)$ ou $f'(x_n)$ prend le même temps. Ainsi, pour la méthode de Newton, il faut environ deux fois plus de temps (calculer $f(x_n)$ et $f'(x_n)$) pour effectuer une itération que dans la méthode de la sécante (calculer uniquement $f(x_n)$). On fait donc deux itérations avec la méthode de la sécante pendant qu'une seule itération est faite avec la méthode de Newton. L'ordre de convergence équivalent est donc $q^2 \simeq 1.62^2 \simeq 2.62$ ce qui est mieux qu'une convergence quadratique ! Enfin, la plupart du temps, on ne connait explicitement que $f$. La méthode de la sécante est donc à privilégier.

Malheureusement, comme avec la méthode de Newton, la convergence n'est pas garantie dans le cas général.

**Exercice : Démonstration pour l'ordre de convergence**

Montrer qu'au voisinage $I$ de $\alpha$, dans le cas où $f$ est $C^2$ sur $I$ et que $x_0$ et $x_1$ sont choisis suffisamment proches (dans $I$), on a $q = (1+\sqrt{5})/2$ pour la méthode de la sécante. Commencer par démontrer le lemme ci-dessous :

**Lemme 1.** *Soit $x_n$ une suite de réels positifs telle que $x_{n+1} \leq x_n x_{n-1}$. Alors, il existe $c > 0$ tel que $x_n \leq c x_0^{\phi^n}$ où $\phi$ est le nombre d'or.*

### 1.2.5 Racines d'un polynôme

À l'origine, la méthode de Newton a été utilisée pour chercher les racines d'un polynôme. Cependant, elle ne permet pas de les trouver toutes de façon systématique : la méthode fournit la racine proche de $x_0$. Afin de calculer la valeur des racines $\lambda_p$ d'un polynôme $P$ de degré $n$, on va introduire la notion de polynôme caractéristique d'une matrice $A$.

**Définition 1.** *Le polynôme caractéristique d'une matrice carré $A$ de dimension $n$ est défini par :*

$$P(X) = det(XI_n - A) \tag{1.18}$$

*où $I_n$ désigne la matrice identité de dimension $n$ et $det(.)$ le déterminant.*

Il est trivial de constater que les racines $\lambda_p$ de $P$ sont les valeurs propres de $A$. Ainsi, trouver les racines de $P$ revient à trouver les valeurs propres d'une matrice $A$. On peut résoudre numériquement ce type de problème avec des variantes des méthodes d'inversion que l'on verra dans le chapitre 4.

Reste à définir notre matrice $A$ sachant qu'on peut toujours se ramener à :

$$P(X) = X^n + \sum_{k=0}^{n-1} a_k X^k \tag{1.19}$$

quitte à diviser notre polynôme par le coefficient du terme de degré le plus élevé $a_n$. En effet, cela ne modifie par les racines $\lambda_p$ que nous cherchons. On peut alors écrire :

$$\lambda_p^n = -\sum_{k=0}^{n-1} a_k \lambda_p^k \tag{1.20}$$

En introduisant le vecteur de $\mathbb{R}^n$

$$V = \begin{pmatrix} 1 \\ \lambda_p \\ \vdots \\ \lambda_p^{n-1} \end{pmatrix} \tag{1.21}$$

D'après la relation 1.20, on a :

$$\lambda_p I_n V = AV \tag{1.22}$$

avec la matrice $A$ donnée par

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-2} & -a_{n-1} \end{pmatrix} \tag{1.23}$$

$A$ est appelée la matrice compagnon du polynôme $P$. Par construction, les valeurs propres de la matrice compagnon $A$ sont les racines de $P$. Comme $A$ est une matrice creuse, cela autorise l'emploi d'algorithmes rapides et efficaces pour la recherche des valeurs propres.

### 1.2.6 Bilan des méthodes

Il existe d'autres méthodes de recherche de zéro que celles qui sont présentées précédemment. Ce sont, soit des variantes de ces méthodes, soit la combinaison de plusieures méthodes (méthodes hybrides). **Cependant, la méthode universelle n'existe pas.** Si $f$ est trop irrégulière ou que la recherche n'est pas initialisée avec un point suffisemment proche de $\alpha$, les méthodes peuvent diverger ou converger vers un autre zéro (cela peut aussi être grave) ou encore vers une singularité de $f$.

En pratique, il faut donc toujours avoir une idée de la valeur de $\alpha$. La méthode de dichotomie permet de réduire l'intervalle de recherche et en général la méthode de Newton (ou sécante) permet une évaluation précise de $\alpha$.

## Bibliographie

NEWTON, ISAAC 1669 De analysi per aequationes numero terminorum infinitas. *Publié en 1711 par William Jones* .

NEWTON, ISAAC 1671 De metodis fluxionum et serierum infinitarum. *Traduit et publié comme méthode de fluxions en 1736 par John Colson* .

## 1.3 TD

### 1.3.1 Exercice : Recherche de zéro

On cherche à résoudre numériquement $f(x) = 0$ pour :

- $f(x) = exp(x) - 1$
- $f(x) = ln(x) - 1$
- $f(x) = x^2 - x - 1$
- $f(x) = sin(x)$
- $f(x) = exp(-x)$

1. Résoudre l'équation à la main.
2. Proposer un algorithme utilisant la méthode de dichotomie. Retrouver l'ordre de convergence.
3. Écrire le programme correspondant à l'algorithme sous Mathematica (ne pas utiliser la fonction intégrée de recherche de zéro !).
4. Donner le zéro avec une précision de $10^{-6}$. S'il en existe plusieurs ou en cas d'échec, discuter.
5. Tracer $|x_{n+1} - \alpha|$ en fonction $|x_n - \alpha|$ de avec $n$ le nombre d'itérations et $\alpha$ la solution analytique. Retrouver numériquement l'ordre de convergence.
6. Recommencer avec la méthode de Newton et des sécantes. Pour $f(x) = exp(-x)$, donner le résultat pour une précision de $10^{-3}$, $10^{-4}$, $10^{-5}$ et $10^{-6}$. Conclure sur ce cas.
7. Proposer une méthode hybride.
8. Pour le polynôme, écrire un programme utilisant la fonction Mathematica donnant les valeurs propres d'une matrice pour calculer ses racines.
9. Conclure sur l'efficacité des différentes méthodes et faites une remarque intelligente sur la dernière fonction.

### 1.3.2 Exercice : Recherche de minimum

On recherche le minimum des fonctions :

- $f(x) = x^2 + x - 1$
- $f(x) = sin(x)$

1. Proposer un algorithme de recherche de minimum.
2. Écrire le programme avec Mathematica.
3. Donner le minimum des fonctions avec une précision de $10^{-6}$ et une estimation numérique de l'ordre de convergence.

Chapitre 2

# Discrétisation des équations continues aux dérivées partielles

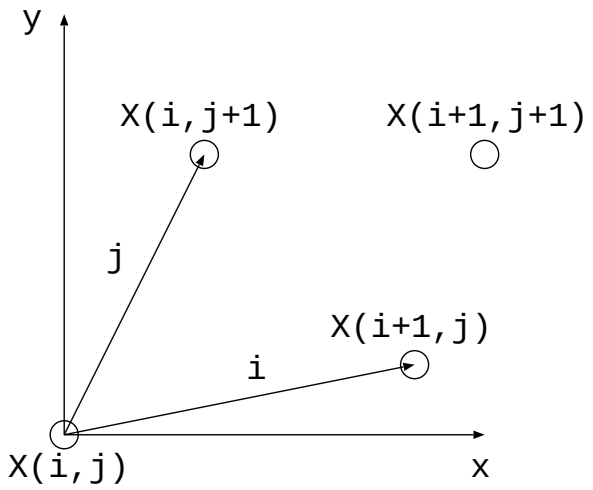


FIGURE 2.1 – Maillage irrégulier en différences finies.

Les méthodes numériques permettent la résolution des équations aux dérivées partielles (edp) que l'on ne sait pas résoudre à la main, c'est-à-dire leur trouver une solution analytique. Le problème général se pose de la façon suivante : trouver la fonction scalaire $u(x, y, z, t)$ vérifiant une équation différentielle dans un domaine de l'espace $D \subset \mathbb{R}^d$, $d = 1$, 2 ou 3 :

$$f(u, \cdots, \partial_{x^\alpha, y^\beta, z^\gamma, t^\delta} u) = 0 \tag{2.1}$$

et les conditions limites sur les bords $\delta D_i$ du domaine $D$ :

$$\forall (x, y, z) \in \delta D_i,\ C_i(u, \cdots, \partial_{x^\alpha, y^\beta, z^\gamma, t^\delta} u) = 0 \tag{2.2}$$

avec $f$ et les $C_i$ des fonctions scalaires données faisant intervenir les dérivées partielles de $u$. Évidemment on suppose que ce système (2.1-2.2) admet une solution unique $u$. Sinon, le problème est mal posé et il est illusoire d'essayer de le résoudre quelle que soit la méthode envisagée.

Par la suite, on se limitera au cas 1D ou 2D pour des raisons de simplicité. Cependant, les méthodes se généralisent aussi en 3D. La résolution de problèmes instationnaires sera plus spécifiquement traitée dans le chapitre 3.

## 2.1 Différences finies

### 2.1.1 Discrétisation spatiale

En différences finies, on discrétise le champ $u(x, y)$ défini dans le domaine $D$ en ne considérant que ses valeurs en des points discrets $X(i, j) = (x(i, j), y(i, j)) \in D$ avec $i$ et $j$ les indices identifiant le point $X$. Définir un maillage du domaine $D$ revient à définir la fonction suivante :

$$X : \{1 : m\} \times \{1 : n\} \rightarrow D \tag{2.3}$$

$$(i, j) \mapsto (x(i, j), y(i, j)) \tag{2.4}$$

avec, ici, $D \subset \mathbb{R}^2$ et $m \times n$ le nombre de points. Les directions $i$ et $j$ sont définies **localement** et ne correspondent pas forcement aux directions de $x$ et de $y$ (voir figure 2.1).

Ainsi, on approxime $u(x, y)$ par l'ensemble de ses valeurs $u(X(i, j))$ que l'on note $u_{i,j}$ pour simplifier. Le nombre de valeurs $u_{i,j}$ que nous considérons est appelé le nombre de degrés de liberté du système. Si on prend $m$ points dans la direction $i$ et $n$ points dans la direction $j$, le nombre de degrés de liberté est $N_{ddl} = mn$. Le champ $u$ discrétisé est alors décrit dans le sous-espace vectoriel $\mathbb{R}^{mn}$ de base canonique :

$$e_{i,j} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \quad \leftarrow (i-1)n + j \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{2.5}$$

On a donc une représentation vectorielle/matricielle de notre champ discret $U$ :

$$U = \sum_{i=1}^{m} \sum_{j=1}^{n} u_{i,j} e_{i,j} \tag{2.6}$$

C'est sous forme de tableau que le résultat d'un calcul est stocké en mémoire :

$$U = \begin{pmatrix} u_{1,1} \\ \vdots \\ u_{i,j} \quad \leftarrow (i-1)n + j \\ \vdots \\ u_{m,n} \end{pmatrix} = \begin{pmatrix} u_1 \\ \vdots \\ u_k \\ \vdots \\ u_{mn} \end{pmatrix} \tag{2.7}$$

En effet, l'indice $k = (i-1)n + j$ correspond alors à une adresse dans la mémoire où est stockée la valeur du champ au point $X(i, j)$. On peut remarquer que les ordinateurs ne traitent que des problèmes 1D, le couple $(i, j)$ étant transformé en un indice $k$ du tableau $U$. La correspondance entre les coordonées spatiales $X$, les indices $(i, j)$ et enfin l'indice $k$ est assurée par le mailleur.

La solution numérique approchée de $u$ est donc le champ discret $U$. La valeur de $u$ au point $X(i, j)$ est donnée par :

$$U^\top e_{i,j} = u_{i,j} \tag{2.8}$$

Cette dernière est une approximation à l'ordre 1 de $u(x, y)$.

**Preuve 1.** *Le développement de Taylor (on suppose u est $C^2$) donne :*

$$u(x, y) = u_{i,j} + (x - x_{i,j}) \left.\frac{\partial u}{\partial x}\right|_{i,j} + (y - y_{i,j}) \left.\frac{\partial u}{\partial y}\right|_{i,j} + \bigcirc((x - x_{i,j})(y - y_{i,j})) \tag{2.9}$$

L'ordre de cette approximation est une des limitations intrinsèques de la méthode des différences finies : entre les points du maillage, la solution numérique approche la solution exacte à l'ordre 1 seulement, quel que soit l'ordre du schéma numérique que l'on choisit pour discrétiser les opérateurs de dérivation (le laplacien, par exemple). Cependant, c'est une méthode efficace dans de nombreux cas et c'est surtout la méthode la plus simple à mettre en oeuvre.

### 2.1.2 Discrétisation des opérateurs de dérivation spatiale

La discrétisation des opérateurs de dérivation de l'équation différentielle 2.1 repose sur le développement en série de Taylor des valeurs de $u$ au voisinage du point $X(i,j)$. Pour la suite, on prend $n=1$ (cas 1D) pour raccourcir les expressions. En faisant un développement au point $i$ à l'ordre $p+q+1$ (le reste est d'ordre $p+q+1$), on a donc le système de $p+q+1$ équations :

$$u_{i-p} = \sum_{k=0}^{p+q} \frac{(x_{i-p}-x_i)^k}{k!} \left.\frac{\partial^k u}{\partial x}\right|_i \tag{2.10}$$

$$\vdots = \vdots \tag{2.11}$$

$$u_{i+q} = \sum_{k=0}^{p+q} \frac{(x_{i+p}-x_i)^k}{k!} \left.\frac{\partial^k u}{\partial x}\right|_i \tag{2.12}$$

qui s'écrit sous forme matricielle :

$$\begin{pmatrix} u_{i-p} \\ \vdots \\ u_{i+q} \end{pmatrix} = \begin{pmatrix} 1 & (x_{i-p}-x_i) & \cdots & \frac{(x_{i-p}-x_i)^{p+q}}{(p+q)!} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (x_{i+q}-x_i) & \cdots & \frac{(x_{i+q}-x_i)^{p+q}}{(p+q)!} \end{pmatrix} \begin{pmatrix} u_i \\ \vdots \\ u_i^{(p+q)} \end{pmatrix} \tag{2.13}$$

avec $u^{(k)} = \partial^k u/\partial x^k$. En posant

$$A_i = \begin{pmatrix} 1 & (x_{i-p}-x_i) & \cdots & \frac{(x_{i-p}-x_i)^{p+q}}{(p+q)!} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (x_{i+q}-x_i) & \cdots & \frac{(x_{i+q}-x_i)^{p+q}}{(p+q)!} \end{pmatrix} \tag{2.14}$$

on obtient le système

$$\begin{pmatrix} u_{i-p} \\ \vdots \\ u_{i+q} \end{pmatrix} = A_i \begin{pmatrix} u_i \\ \vdots \\ u_i^{(p+q)} \end{pmatrix} \tag{2.15}$$

avec $A_i$ une matrice carrée inversible de dimension $p+q+1$. Pour obtenir la dérivée d'ordre $k$ ($1 \leq k \leq p+q$) avec un schéma à l'ordre $p+q+1-k$ (au pire), il suffit de calculer $A_i^{-1}$. En posant $[A_i^{-1}]_{k,l} = d_k(i,l)$, on constate que $u^{(k)}$ est une combinaison linéaire des valeurs de $u_{i-p+l-1}$ :

$$u_i^{(k)} = \sum_{l=1}^{p+q+1} d_k(i,l) u_{i-p+l-1} \quad [+ \bigcirc (x_{i+q}-x_{i-p})^{p+q+1-k}] \tag{2.16}$$

Si on prend tous les $u_i^{(k)}$ de $D$ (*i.e.* pour $1 \leq i \leq m$), il y a un problème : on doit avoir $i-p \geq 1$ et $i+q \leq m$ car sinon on sort de $D$. Or, cela n'est pas garanti lorsque l'on se rapproche du bord du domaine $D$, c'est-à-dire lorsque $i < p+1$ ou $i > m-q$. On ne peut donc écrire la dérivée $k$ème que pour $m-p-q$ points intérieurs au domaine avec le schéma considéré et on obtient un système qui peut, encore une fois, s'écrire sous forme matricielle :

$$\begin{pmatrix} u_{p+1}^{(k)} \\ \vdots \\ u_{m-q}^{(k)} \end{pmatrix} = D_k \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} \tag{2.17}$$

avec la matrice $(m-p-q) \times m$ :

$$D_k = \begin{pmatrix} d_k(p+1,1) & \cdots & d_k(p+1,p+q+1) & 0 & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & \cdots & 0 & d_k(m-q,1) & \cdots & d_k(m-q,p+q+1) \end{pmatrix} \tag{2.18}$$

Pour conclure, il faut faire quelques remarques :

— En général, $A_i$ est de petite dimension car sa dimension est $r+k$ avec $r$ l'ordre du schéma qu'on veut avoir.

— L'ordre des schémas numériques utilisés pour calculer les dérivées $k$èmes doit être le même pour tout $k$ dans l'edp. $A_i^{-1}$ ne fournit donc qu'une seule dérivée. Il faut recommencer pour les autres en s'arrêtant au même ordre. Cependant, ce travail est fait une fois pour toute si le maillage est fixe.

— Si $A_i$ est facile à inverser, il peut être utile d'écrire explicitement $d_k(i,l)$ en fonction des $x_i$ c'est-à-dire du maillage.

— Les points du maillage ne doivent pas être numérotés n'importe comment car l'erreur est de l'ordre de $|x_{i+1} - x_i|^{p+q+1-k}$. Programmer un mailleur est une tâche fastidieuse. Il existe des mailleurs commerciaux ou libres qui peuvent être utilisés afin d'éviter ce travail.

— Si le maillage est régulier et ordonné, l'écriture du schéma de dérivation est très simplifiée mais ce n'est, en général, pas la meilleure solution d'un point de vue de l'efficacité numérique. Il faut mettre plus de points de maillage là où le gradient de $u$ est grand.

— Si le maillage est orthogonnal, la procédure est la même pour $x$, $y$ et $z$.

— Si $p = q$, on dit que le schéma est dit centré. Ce type de schéma est souvent utilisé pour des raisons de stabilité numérique, notamment dans les problèmes de diffusion. Lorsque qu'il y a une vitesse de propagation $v > 0$ dans la direction $x$ du champ $u$, alors il peut être utile de décentrer le schéma. En effet, la vitesse $v$ transporte le champ $u$ de $x_{i-1}$ vers $x_i$ (en supposant que $i$ et $x$ ont le même sens de variation) et on a intérêt à choisir $p < q$. C'est un schéma dit « upwind », l'inverse est dit « downwind ».

**Exemples de schémas**

On se propose de considérer la dérivée $\partial/\partial x$ et la dérivée seconde $\partial^2/\partial x^2$.

1. Donner les schémas centrés (p=q) à l'ordre 2 sans faire d'hypothèse sur la répartition des points $x_i$.
2. Donner les schémas centrés si $x_{i+1} - x_i = \Delta x$, $\forall i$ (pas constant).

### 2.1.3 Forme matricielle du système d'équations aux dérivées partielles

Le champ scalaire $u$ défini sur le domaine $D$ délimité par des bords $\delta D = \bigcup \delta D_l$ vérifie une équation aux dérivées partielles (edp) d'ordre $k$. On dispose de $N_{CL}$ conditions limites sur les bords du domaine.

$$\text{edp} \quad \left\{ f(u, \cdots, u^{(k)}) = 0, \quad \forall x \in D \right. \tag{2.19}$$

$$\text{CL} \quad \begin{cases} C_1(u, \cdots, u^{(k)}) = 0 \quad \forall x \in \delta D_1 \\ \vdots \\ C_{N_{CL}}(u, \cdots, u^{(k)}) = 0 \quad \forall x \in \delta D_{N_{CL}} \end{cases} \tag{2.20}$$

Dans le cas à une dimension (même principe à deux ou trois dimensions) la discrétisation du champ $u$ transforme le problème continu en un problème à $m$ inconnues scalaires $(u_1 \dots u_m)$, c'est-à-dire le tableau $U$. Le remplacement des dérivées par leurs expressions en fonction des valeurs discrètes $u_i$ dans l'edp fournit $m - p - q$ équations :

$$f(u, \cdots, u^{(k)}) = 0 \quad \underset{\text{discrétisation}}{\Longrightarrow} \quad \begin{array}{c} f(u_{p+1}, \cdots, u^{(k)}_{p+1}) = f_{p+1}(U) = 0 \\ \vdots \\ f(u_{m-q}, \cdots, u^{(k)}_{m-q}) = f_{m-q}(U) = 0 \end{array} \tag{2.21}$$

On dispose ainsi de $m - p - q$ équations pour $m$ inconnues. On ne peut donc pas résoudre le système en l'état. Il faut $p + q$ équations supplémentaires. Ces dernières sont données par les conditions limites que l'on discrétise sur les points appartenant aux bords du domaine $D$. Il y a donc une contrainte sur le nombre de conditions limites. Dans notre cas 1D, il y a deux points appartenant aux extrémités du domaine, $x_1$ et $x_m$ et donc la somme des conditions en ces points doit être équale à $p+q$. Si les conditions limites font intervenir $u^{(k)}$, il faut changer de schéma numérique pour faire intervenir les points internes au domaine. Lorsque le problème continu est bien posé, la fermeture du problème discret est possible en discrétisant correctement les dérivées et les conditions limites. Cependant, il n'y a pas de recette générale, la discrétisation doit être adaptée au type de problème à résoudre. On peut quand même dire que les dérivées aux limites doivent être discrétisées au même ordre que les dérivées internes pour éviter des problèmes de précision aux bords du domaine.

Ainsi, le problème continu est remplacé par un système de $m$ équations à $m$ inconnues :

$$\begin{pmatrix} f_1(U) \\ \vdots \\ f_{p+1}(U) \\ \vdots \\ f_{m-q}(U) \\ \vdots \\ f_m(U) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{2.22}$$

qui peut s'écrire sous forme compacte :

$$F(U) = 0 \tag{2.23}$$

$F$ est une fonction de $\mathbb{R}^m$ dans $\mathbb{R}^m$. Pour résoudre 2.23, on peut donc utiliser la méthode de Newton (voir 1.2.3) en introduisant la matrice jacobienne de $F$, $J_F$.

**Définition 2.** *Soit une fonction $F$ de $\mathbb{R}^m$ dans $\mathbb{R}^m$ définie par :*

$$F : \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} \mapsto \begin{pmatrix} f_1(u_1, \cdots, u_m) \\ \vdots \\ f_m(u_1, \cdots, u_m) \end{pmatrix} \tag{2.24}$$

*La matrice jacobienne de $F$ est donnée par :*

$$J_F = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial u_1} & \cdots & \frac{\partial f_m}{\partial u_m} \end{pmatrix} \tag{2.25}$$

Cela revient à dire que résoudre le problème 2.23 revient à résoudre $N$ fois le problème :

$$U_{n+1} = U_n - J_F^{-1} F(U_n) \tag{2.26}$$

jusqu'à ce que $\|U_{n+1} - U_n\| < \varepsilon$ ou $\|F(U_{n+1})\| < \varepsilon$, avec $\varepsilon > 0$. Le coût numérique est du au calcul de la valeur approchée de la matrice jacobienne à l'aide d'une généralisation de l'équation 1.16 utilisée par la méthode des sécantes :

$$\frac{\partial f_i}{\partial u_j} = \frac{f_i(U_n) - f_i(U_{n-1})}{u_j(n) - u_j(n-1)} \tag{2.27}$$

et à son inversion.

La méthode de Newton converge en peu d'itérations mais nous avons vu précédemment que le champ initial ne doit pas être trop éloigné de la solution et qu'une itération est très coûteuse. De plus, il faut que $F$ vérifie au moins le théorème d'inversion local :

**Théorème 2.** *Si $F$ est une fonction continue et dérivable, alors $F$ est inversible au voisinage d'un point $M$ si et seulement si le jacobien $\mathcal{J}_F = det(J_F)$ de $F$ est non-nul en $M$.*

Le théorème d'inversion global permet d'étendre le voisinage de recherche à un ouvert $A \subset E = \mathbb{R}^m$ :

**Théorème 3.** *On suppose que l'on travaille dans un espace $E = \mathbb{R}^m$. Soit $F$ une application :*

— *injective d'un ouvert $A$ de $E$ dans $E$ ;*

— *de classe $C^1$ sur $A$.*

*Alors $F$ est un $C^1$-difféomorphisme si et seulement si son jacobien ne s'annule pas sur $A$. L'injectivité est nécessaire : la non-nullité du jacobien seule ne l'implique pas.*

### 2.1.4 Cas d'une edp linéaire

Dans ce cas, l'edp est une combinaison linéaire d'opérations de dérivation et de l'identité. Or, les dérivées sont des opérateurs linéaires qui sont représentés par des matrices dans leur forme discrète. On peut donc représenter le système discret directement sous forme matricielle :

$$F(U) = AU - B = 0 \tag{2.28}$$

$F$ est donc une fonction affine de $U$ et la matrice $A$ représente un opérateur linéaire discrétisé. $B$ est le terme constant de l'edp. Pour que notre système admette une solution, il faut que $A$ soit inversible et alors on a :

$$U = A^{-1} B \tag{2.29}$$

On peut remarquer que la relation de récurrence 2.26 est équivalente à 2.29 car dans ce cas, $J_F = A$ et $J_F U_n - F(U_n) = B$. La solution est alors trouvée en un seul pas. Cependant, dans tous les cas, il faut inverser une matrice carrée de dimension $N_{ddl}(= m)$ pour trouver la solution numérique à notre problème.

Le calcul de la matrice jacobienne devant être effectué à chaque itération, le temps de calcul est donc beaucoup plus long si $F$ n'est pas affine. Cependant, on constate que dans tous les cas de figure, résoudre un problème numérique revient à inverser un problème linéaire de dimension $N_{ddl}$, une fois (équation linéaire), ou plusieurs fois (équation non linéaire).

### 2.1.5 Réduction de degré de l'edp

En général, on évite de discrétiser les dérivées d'ordre supérieur à 2. Si l'edp en fait intervenir on utilise une méthode de réduction de degré en introduisant des variables supplémentaires qui sont les dérivées des variables de départ. Ainsi, on écrit :

$$v = u' \tag{2.30}$$

$$f(u, v, \cdots, v^{(k-1)}) = 0 \tag{2.31}$$

pour réduire de 1 l'ordre de l'edp.

## 2.2 Éléments finis

La méthode des éléments finis diffère de la méthode des différences finies sur deux points essentiels :

— Le champ $u$ est interpolé par des fonctions continues sur le domaine $D$.

— L'équation aux dérivées partielles est écrite sous une forme intégrale dite faible.

### 2.2.1 Discrétisation d'un champ

On considère notre champ $u$ défini sur $D$ comme précédemment. Si on se place entre les points d'un maillage fait pour la méthode des différences finies, la valeur de $u$ n'est pas clairement définie. Maintenant, on veut améliorer cette situation et on va essayer d'interpoler $u$ entre les points d'un maillage afin d'en avoir une approximation $u_h$ définie sur tout le domaine $D$. Les points du maillage où l'on connait la valeur de $u$ s'appellent les points de collocation ou noeuds du maillage $x_i$. On se donne donc une base de fonctions $\varphi_i$ d'interpolation autour du point $i$ dans $D$ et cette base engendre un espace de Sobolev $W^{k,\alpha}(D)$ (espace vectoriel muni d'une norme $\|.\|_\alpha$ de l'espace de Lebesque $L^\alpha(D)$). La fonction interpolée s'écrit alors :

$$u_h(x) = \sum_i^m u_i \varphi_i(x) \tag{2.32}$$

Ici, $x$ peut s'entendre comme la coordonnée spatiale sur un maillage à une dimension ou comme le vecteur position d'un point du maillage dans les cas 2D et 3D. On veut que notre fonction $u_h$ vérifie :

$$u_h(x_i) = u_i \tag{2.33}$$

et que dans le domaine $D$, il existe $M \geq 0$ tel que :

$$\|u(x) - u_h(x)\|_\alpha < M, \forall x \in D \tag{2.34}$$

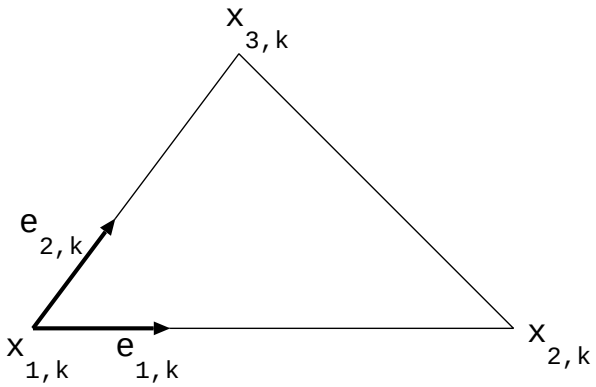


FIGURE 2.2 – Élément fini triangulaire de type P1.

Ces exigences sont minimales. Cela impose une des propriétés des fonctions de bases :

$$\varphi_i(x_j) = \delta_{ij} \tag{2.35}$$

avec $\delta_{ij}$ le symbole de Kronecker. En dehors de cette propriété 2.35, toutes les autres propriétés de la base $\{\varphi_i\}$ dépendent de la régularité de la fonction $u_h$ que l'on veut ($C^0$, $C^1$, *etc* ...).

### 2.2.2 Élément fini

Pour définir la fonction $\varphi_i$, on sait que

$$\varphi_i(x_i) = 1 \tag{2.36}$$
$$\varphi_i(x_{j\neq i}) = 0 \tag{2.37}$$

Le système d'équations (2.36-2.37) contient a priori $m$ équations pour définir la fonction $\varphi_i$. La fonction $\varphi_i$ ne peut donc pas appartenir à un sous-espace de fonctions de dimension supérieure à $m$. On se donne un sous-espace de fonctions générateur des fonctions $\varphi_i$ de dimension $1 \leq m_e \leq m$. Il faut donc $m_e$ points de collocation pour définir $\varphi_i$, c'est-à-dire $m_e$ équations (2.36-2.37). Le nœud $i$ est donc lié à $m_e - 1$ autres points du maillage. Cet ensemble de points définit un sous domaine de $D$ noté $D_e$. Si on se place dans ce sous domaine $D_e$, on peut renuméroter les nœuds de 1 à $m_e$. Toujours en restant dans $D_e$, on constate que l'on a $m_e$ fonctions d'interpolation *i.e.* une par nœud de $D_e$. De plus, si l'on prend le nœud 1 comme origine d'un repère local au sous-domaine $D_e$, il faut que l'ensemble des vecteurs $x_l - x_1$ ($l \in \{2, \cdots, m_e\}$) soit de la même dimension que $D$.

Exemple : $D$ est une surface (2D). Il faut donc au moins deux vecteurs non colinéaires pour former un ensemble de dimension 2. Le nombre de points minimum pour définir $D_e$ est donc de trois. Dans ce cas, on obtient un élément triangulaire dont les sommets sont les nœuds du maillage (figure 2.2).

**Définition 3.** *Le sous domaine $D_e$ contenant les $m_e$ nœuds $x_{i,e}$ associé à l'ensemble des $m_e$ fonctions d'interpolation $\varphi_{i,e}$ définies sur $D_e$ est ce qu'on appelle un élément fini.*

On doit aussi s'assurer que :

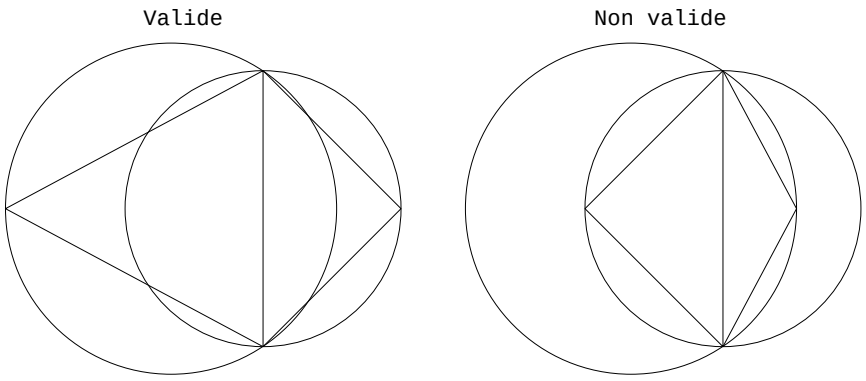


FIGURE 2.3 – Algorithme de Delaunay.

$$D = \bigcup_{i=1}^{N_e} D_i \tag{2.38}$$

$$\text{Si } i \neq j,\ D_i \cap D_j = \emptyset \tag{2.39}$$

avec $N_e$ le nombre d'éléments du maillage.

**Preuve 2.** *2.38 : $u_h$ est défini sur $\tilde{D} = \bigcup_{i=1}^{N_e} D_i$ et sur $D$. Donc $D \subset \tilde{D}$. Comme $\forall i$, $D_i \subset D$, on a $\tilde{D} \subset D$ et donc $D = \tilde{D}$.*

**Preuve 3.** *2.39 : Soit $E = D_i \cap D_j$ avec $i \neq j$. On a d'après la définition de l'interpolation de $u$ dans un élèment :*

$$\forall x \in E, \quad u_h(x) = \sum_{l=1}^{m_e} u_{l,i}\varphi_{l,i}(x) = \sum_{l=1}^{m_e} u_{l,j}\varphi_{l,j}(x) \tag{2.40}$$

*Comme les $\varphi_{l,i}$ et $\varphi_{l,j}$ sont des fonctions libres si $i \neq j$ (elles forment une base pour $u_h$ dans $D$), la dernière égalité de 2.40 implique que $u_h = 0$ ou que $i = j$. Ainsi donc, $E = \emptyset$.*

L'ensemble des éléments finis forme donc un pavage du domaine $D$. Le bord d'un élément délimité par deux (minimum en 2D) nœuds (nodes) s'appelle une face (edge). Les faces d'un élément fini sont soit sur le bord du domaine $D$, soit à l'interface avec un autre élément fini. Dans le premier cas, les fonctions d'interpolation doivent être compatibles avec les conditions limites. Cela veut dire que $u_h$ doit pouvoir vérifier les conditions limites. Dans le deuxième cas, on impose certaines conditions comme la continuité de $u_h$ entre les éléments. Dans la majorité des cas (c'est le cas pour les éléments de type P1 et P2, figure 2.2), les nœuds du maillage sont sur les interfaces entre éléments.

Le positionnement et l'organisation des nœuds sont du ressort du mailleur. Les informations que le mailleur doit fournir sont donc :

— La position spatiale des nœuds et leur numérotation.
— La définition des faces à partir des nœuds.
— La définition des éléments à partir des faces.

L'algorithme de Delaunay (1934) est couramment utilisé pour réaliser un maillage automatiquement. En partant d'un triangle de départ, l'algorithme vérifie qu'aucun autre point du maillage n'est à l'intérieur du cercle circonscrit à ce triangle (voir figure 2.3). Cela permet de générer le maillage d'une surface ou d'un volume en évitant les chevauchements entre éléments et en garantissant une bonne qualité de maillage.

Par qualité, on entend que les directions $e_{i,k}$ doivent être bien distinctes, c'est-à-dire que le triangle de la figure 2.2 ne doit pas être écrasé.

### 2.2.3 Formulation faible de l'edp

Si on écrit naïvement l'edp en remplaçant $u$ par $u_h$, on a :

$$f(u_h, \cdots, u_h^{(k)}) = 0, \quad \forall x \in D + CL \tag{2.41}$$

D'après la définition de $u_h$ donnée par 2.32, on a donc $m$ inconnues scalaires à résoudre. Il faut résoudre le système de $m$ équations, obtenu à partir de 2.41 :

$$\begin{cases} f(u_h(x_1), \cdots, u_h^{(k)}(x_1)) = 0 \\ \vdots \\ f(u_h(x_m), \cdots, u_h^{(k)}(x_m)) = 0 \end{cases} \tag{2.42}$$

car on connait les fonctions $\varphi_i$ et qu'on suppose qu'elles sont choisies de telle manière que les conditions limites puissent être respectées. On peut réécrire ce système d'équations 2.42 en posant :

$$U = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} \quad \text{et} \quad \phi_i^{(k)} = \begin{pmatrix} \varphi_1^{(k)}(x_i) \\ \vdots \\ \varphi_m^{(k)}(x_i) \end{pmatrix} \tag{2.43}$$

ce qui donne

$$F(U) = \begin{pmatrix} f(U^\top \phi_1, \cdots, U^\top \phi_1^{(k)}) \\ \vdots \\ f(U^\top \phi_m, \cdots, U^\top \phi_m^{(k)}) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \tag{2.44}$$

On reconnait une équation du type de 2.23. On peut donc envisager d'utiliser la méthode de Newton, par exemple. Cependant, si le système d'équations 2.44 admet une solution $U$, alors $u = u_h$ est une solution algébrique exacte, c'est-à-dire que la solution $u \in W(D)$ avec $W(D)$ l'espace de fonctions engendré par les $\{\varphi_i\}$. Malheureusement, $u \notin W(D)$ en général car cela reviendrait à savoir résoudre algébriquement l'edp. Dans ce cas, l'équation 2.41 n'a pas de solution. Il faut donc trouver une autre formulation que la formulation forte 2.41, qui soit moins contraignante et qui permette de minimiser la différence entre la solution cherchée $u$ et son approximation $u_h$. Pour cela, on passe à une vision globale ou intégrale de l'équation locale 2.41. C'est la formulation faible de l'edp 2.41.

**Théorème 4.** *Lax-Milgram*

*Soient :*

- *$\mathcal{H}$ un espace de Hilbert réel muni de son produit scalaire noté $\langle ., . \rangle$, de norme associée notée $\|.\|$.*
- *$a(.\,,.)$ une forme bilinéaire qui est*
  - *continue sur $\mathcal{H} \times \mathcal{H}$ : $\exists\, c > 0, \forall (u, v) \in \mathcal{H}^2\,,\ |a(u, v)| \leq c\|u\|\|v\|$*
  - *coercive sur $\mathcal{H}$ (certains auteurs disent plutôt $\mathcal{H}$-elliptique) : $\exists\, \alpha > 0, \forall u \in \mathcal{H}\,,\ a(u, u) \geq \alpha\|u\|^2$*
- *$l$ une forme linéaire continue sur $\mathcal{H}$.*

*Sous ces hypothèses il existe un unique $u$ de $\mathcal{H}$ tel que l'équation $a(u, v) = l(v)$ soit vérifiée pour tout $v$ de $\mathcal{H}$ :*

$$\exists!\ u \in \mathcal{H},\ \forall v \in \mathcal{H}, \quad a(u, v) = l(v) \tag{2.45}$$

*Si de plus la forme bilinéaire $a$ est symétrique, alors $u$ est l'unique élément de $\mathcal{H}$ qui minimise la fonctionnelle $J : \mathcal{H} \to \mathbb{R}$ définie par $J(v) = \frac{1}{2}a(v,v) - l(v)$ pour tout $v$ de $\mathcal{H}$, c'est-à-dire :*

$$\exists!\ u \in \mathcal{H}, \quad J(u) = \min_{v \in \mathcal{H}} J(v) \tag{2.46}$$

Ainsi, d'après le théorème de Lax-Milgram, on doit se ramener à un problème linéaire. On a vu que c'est possible avec la matrice jacobienne. On suppose donc que le problème à inverser s'écrit :

$$\mathcal{L}(u) = b \tag{2.47}$$

avec $\mathcal{L}$ un opérateur linéaire inversible et $b$ une fonction continue connue. On définit un produit scalaire $\langle ., . \rangle$. En outre, on choisit $a(.,.) = \langle \mathcal{L}(.), . \rangle$. D'après le théorème de Lax-Milgram, le problème :

$$\forall v_h \in \mathcal{W}(D),\ a(u_h, v_h) = \langle b, v_h \rangle \tag{2.48}$$

admet une solution unique $u_h \in W(D)$ avec $\mathcal{H}(D) = W(D)$ le sous-espace engendré par la base orthonormale $\{\varphi_i\}$. Ainsi, la formulation faible permet de chercher une solution approchée dans $W(D)$ même si la solution exacte $u$ du problème 2.47 n'est pas dans $W(D)$.

On va maintenant écrire 2.48 sous forme matricielle pour pouvoir résoudre numériquement notre problème. Pour cela, on montre qu'il y a équivalence entre $\forall v_h \in \mathcal{W}(D)$ et $\forall \varphi_i$ :

**Preuve 4.** *Si $\forall v_h \in \mathcal{W}(D)$, $a(u_h, v_h) = \langle b, v_h \rangle$, comme $\varphi_i \in \mathcal{W}(D)$ alors pour $v_h = \varphi_i$, $a(u_h, \varphi_i) = \langle b, \varphi_i \rangle$.*

*Si $\forall \varphi_i$, $i \in \{1, \cdots, m\}$, $a(u_h, \varphi_i) = \langle b, \varphi_i \rangle$ alors soit $v_h \in \mathcal{W}(D)$, on peut écrire :*

$$v_h = \sum_{i=1}^{m} v_i \varphi_i \tag{2.49}$$

*Comme $a(.,.)$ est une forme bilinéaire, on a :*

$$a(u_h, v_h) = \sum_{i=1}^{m} v_i a(u_h, \varphi_i) = \sum_{i=1}^{m} v_i \langle b, \varphi_i \rangle = \langle b, v_h \rangle \tag{2.50}$$

Notre équation 2.48 peut donc s'écrire comme le système d'équations :

$$\begin{cases} a(u_h, \varphi_1) = b_1 \\ \vdots \\ a(u_h, \varphi_m) = b_m \end{cases} \tag{2.51}$$

avec $b_i = \langle b, \varphi_i \rangle$. La décomposition de $u_h$ dans la base de fonctions $\{\varphi_i\}$ donne :

$$a(u_h, \varphi_i) = \sum_{j=1}^{m} u_j a(\varphi_j, \varphi_i) = A_i^\top U \tag{2.52}$$

avec

$$A_i = \begin{pmatrix} a(\varphi_1, \varphi_i) \\ \vdots \\ a(\varphi_m, \varphi_i) \end{pmatrix} \tag{2.53}$$

Ainsi, en posant $A = [A_1, \cdots, A_m]^\top$, c'est-à-dire $A_{ij} = a(\varphi_j, \varphi_i)$, à partir du système 2.51, on obtient l'équation matricielle :

$$AU = B \tag{2.54}$$

avec $B = [b_1, \cdots, b_m]^\top$.

Il faut donc, comme pour les différences finies, inverser un problème matriciel. Avant cela, il faut pouvoir générer la matrice $A$, c'est-à-dire calculer $A_{ij} = a(\varphi_j, \varphi_i)$. On doit donc choisir des fonctions d'interpolation $\{\varphi_i\}$ et le produit scalaire.

### 2.2.4 Produit scalaire

Le produit scalaire est une forme linéaire définie positive. Le produit scalaire de l'espace de Lebesgue $L^2(D, \mu)$ est défini par :

$$\langle u, v \rangle = \int_D uv\mu(x)dx \tag{2.55}$$

où $D$ est le domaine de mesure et $\mu$ le poids, *i.e.* $\forall x \in D$, $\mu > 0$. Bien souvent, $\mu = 1$. Cependant, il peut être utile de définir un poids $\mu$ différent pour assurer l'orthogonalité de deux fonctions par rapport au produit scalaire ainsi défini.

### 2.2.5 Fonctions d'interpolation

Afin de bien choisir la base $\{\varphi_i\}$, on peut remarquer que si $\mathcal{L} = Id$, on aimerait que la matrice $A$ soit la matrice identité. Ce n'est pas une obligation, mais cela simplifie grandement la forme de $A$ et surtout son inversion. On a alors :

$$A_{ij} = a(\varphi_j, \varphi_i) = \langle \varphi_j, \varphi_i \rangle = \delta_{ij} \tag{2.56}$$

En d'autre terme, la base $\{\varphi_i\}$ est orthonormée. Cela permet d'écrire :

$$u_i = \langle u_h, \varphi_i \rangle \tag{2.57}$$

Pour les éléments finis, $\varphi_i$ est en général un polynôme d'interpolation de Lagrange de degré 1 ou 2. S'il s'agit d'un polynôme de degré supérieur ou d'un autre type de fonction, on parle d'éléments spectraux (généralisation des éléments finis).

#### Polynômes de Lagrange

En analyse numérique, les polynômes de Lagrange, du nom de Joseph Louis Lagrange, permettent d'interpoler une série de points par un polynôme qui passe exactement par ces points appelés aussi nœuds. Cette technique d'interpolation polynomiale a été découverte par Edward Waring en 1779 et redécouverte plus tard par Leonhard Euler en 1783. On se donne $m_e$ points $(x_1, u_1), \ldots, (x_{m_e}, u_{m_e})$ (avec les $x_i$ distincts 2 à 2). On se propose de construire un polynôme de degré minimal qui prend les valeurs $u_i$ aux abscisses $x_i\,(i = 1, ..., m_e)$.

Les polynômes de Lagrange associés à ces points sont les polynômes définis par :

$$l_i(x) = \prod_{j=1, j\neq i}^{m_e} \frac{x - x_j}{x_i - x_j} = \frac{x - x_1}{x_i - x_1} \cdots \frac{x - x_{i-1}}{x_i - x_{i-1}} \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdots \frac{x - x_{m_e}}{x_i - x_{m_e}}. \tag{2.58}$$

On a en particulier deux propriétés :

1. $l_i$ est de degré $m_e - 1$ pour tout $i$.
2. $l_i(x_j) = \delta_{i,j}, 1 \leq i, j \leq m_e$.

Le polynôme défini par $L(x) = \sum_{i=1}^{m_e} u_i l_i(x)$ est l'unique polynôme de degré au plus $m_e - 1$ vérifiant $L(x_i) = u_i$ pour tout $i$.

**Preuve 5.** *En effet $L(x_i) = \sum_{j=1}^{m_e} u_j l_j(x_i) = u_i$ et $L$ est une combinaison linéaire de polynômes de degré au plus $m_e - 1$ donc est de degré $m_e - 1$ au plus.*

*Si un autre polynôme, $Q$, vérifie ces propriétés alors $L - Q$ est de degré $m_e - 1$ au plus, et s'annule en $m_e$ points (les $x_i$) donc est nul ce qui prouve l'unicité.*

### Polynômes de Tchebychev

Les polynômes de Tchebychev sont nommés d'après le mathématicien Pafnouti Tchebychev. Ils forment une famille de polynômes indexés par les entiers.

Polynômes de Tchebychev de 1re espèce

Le polynôme de Tchebychev de première espèce $T_n$ d'indice $n = 0, 1, 2, ...$ est uniquement défini par la propriété suivante : pour tout nombre réel $x$,

$$T_n(\cos(x)) = \cos(nx) \tag{2.59}$$

Les premiers polynômes de Tchebychev sont :

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1, \quad T_3(x) = 4x^3 - 3x \tag{2.60}$$

— On a :

$$T_n(x) = \frac{n}{2} \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \frac{(n-k-1)!}{k!(n-2k)!} (2x)^{n-2k}, \quad n \neq 0. \tag{2.61}$$

— $T_n$ forme une suite de polynômes orthogonaux avec le poids

$$\frac{1}{\sqrt{1-x^2}}, \tag{2.62}$$

sur l'intervalle $]-1, 1[$, c'est-à-dire :

$$\int_{-1}^{1} T_n(x) T_m(x) \frac{dx}{\sqrt{1-x^2}} = \begin{cases} 0 & : n \neq m \\ \pi & : n = m = 0 \\ \pi/2 & : n = m \neq 0 \end{cases} \tag{2.63}$$

— Les valeurs :

$$a_k^{(n)} = \cos\left(\frac{(2k-1)\pi}{2n}\right), \quad k \in \{1, ..., n\}, \quad n \neq 0, \tag{2.64}$$

sont les $n$ racines de $T_n$.

Polynômes de Tchebychev de 2e espèce

Il existe aussi des polynômes de Tchebychev de seconde espèce, $U_n$ définis par :

$$U_n(\cos(x)) = \frac{\sin((n+1)x)}{\sin x}, \quad n \in \mathbb{N}, \quad x \in \mathbb{R}. \tag{2.65}$$

— Les premiers polynômes sont :

$$U_0(x) = 1 \quad U_1(x) = 2x \quad U_2(x) = 4x^2 - 1. \tag{2.66}$$

— Pour tout réel $x$, on a :

$$U_n(x) = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \binom{n-k}{k} (2x)^{n-2k}, \quad n \neq 0. \tag{2.67}$$

— Les polynômes de Tchebychev de seconde espèce sont orthogonaux avec le poids

$$\sqrt{1-x^2} \tag{2.68}$$

sur l'intervalle $[-1, 1]$, c'est-à-dire :

$$\int_{-1}^{1} U_n(x) U_m(x) \sqrt{1-x^2}\, dx = \begin{cases} 0 & : n \neq m \\ \pi/2 & : n = m \end{cases} \tag{2.69}$$

— Pour tout $n$ entier

$$U_n(1) = n + 1 \tag{2.70}$$

— Les valeurs

$$a_k^{(n)} = \cos\left(\frac{k\pi}{n+1}\right), \quad k \in \{1, ..., n\}, \quad n \neq 0, \tag{2.71}$$

sont les $n$ racines de $U_n$.

Tchebychev a découvert ceux-ci en travaillant sur le problème de convergence des interpolations de Lagrange. On peut démontrer que pour minimiser l'erreur engendrée par l'interpolation (cf. phénomène de Runge), il faut choisir les racines des polynômes de Tchebychev comme points d'interpolation.

#### Coordonnées locales

Il est souvent commode de définir un système de coordonées local à l'élément (voir la figure 2.1). Les fonctions d'interpolation sont écrites dans ce système. Cela permet de se ramener à ce qui a été dit précédemment sur les polynômes d'interpolation.

### 2.2.6 Conditions limites

Afin que $\mathcal{L}$ soit inversible, c'est-à-dire que $A$ le soit, il faut intégrer les conditions limites.

#### Méthode directe

Si les conditions limites sont de type Dirichlet, c'est-à-dire que l'on impose une valeur à $u$ sur le bord du domaine $D$, $u|_{\delta D} = u_0$, on a pour $x_i \in \delta D$, $u_i = u_0$. La méthode la plus directe pour imposer cette condition consiste à introduire directement l'équation $u_i = u_0$ dans le système 2.51. Cela revient à modifier une ligne et une colonne de la matrice $A$ :

$$A = \begin{pmatrix} a(\varphi_1,\varphi_1) & \cdots & a(\varphi_{i-1},\varphi_1) & 0 & a(\varphi_{i+1},\varphi_1) & \cdots & a(\varphi_m,\varphi_1) \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a(\varphi_1,\varphi_{i-1}) & \cdots & a(\varphi_{i-1},\varphi_{i-1}) & 0 & a(\varphi_{i+1},\varphi_{i-1}) & \cdots & a(\varphi_m,\varphi_{i-1}) \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ a(\varphi_1,\varphi_{i+1}) & \cdots & a(\varphi_{i-1},\varphi_{i+1}) & 0 & a(\varphi_{i+1},\varphi_{i+1}) & \cdots & a(\varphi_m,\varphi_{i+1}) \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a(\varphi_1,\varphi_m) & \cdots & a(\varphi_{i-1},\varphi_m) & 0 & a(\varphi_{i+1},\varphi_m) & \cdots & a(\varphi_m,\varphi_m) \end{pmatrix} \tag{2.72}$$

Cette méthode augmente beaucoup la largeur de bande de $A$ et rend donc son inversion plus longue et plus difficile. Elle n'est donc pratiquement jamais utilisée.

### Méthode de pénalisation

Cette méthode consiste à approcher avec la précision que l'on souhaite la valeur imposée en $x_i$ au bord de $D$ en ne modifiant qu'un élément de $A$, ce qui permet de conserver la largeur de bande. On a au point $i$ l'équation :

$$\sum_{j=1}^{m} A_{ij}u_j = b_i \tag{2.73}$$

Si on ajoute à $A_{ii}$ un nombre $K$ très grand devant tous les $A_{ij}$, on peut écrire :

$$\sum_{j=1}^{m} A_{ij}u_j \simeq Ku_i \tag{2.74}$$

Écrire la condition limite revient maintenant à écrire :

$$Ku_i = Ku_0 \simeq Ku_0 + b_i \tag{2.75}$$

Ainsi donc, la méthode de pénalisation consiste à choisir $K \gg \max(|A_{ij}|)$ (par exemple, $K = 10^{30}$ dans Freefem++) et à modifier $A$ et $B$ en remplaçant :

$$A_{ii} \quad \text{par} \quad A_{ii} + K \tag{2.76}$$

$$\text{et } b_i \quad \text{par} \quad b_i + Ku_0 \tag{2.77}$$

Plus $K$ est grand, meilleure est l'approximation. La méthode de pénalisation est la méthode la plus utilisée car elle n'ajoute pas de variables supplémentaires et elle ne change pas la largeur de bande de $A$.

### Méthode Lagrangienne

Cette méthode consiste à ajouter des variables supplémentaires, le multiplicateur de Lagrange $\lambda$. Elle n'est valable que si $A$ est symétrique car elle permet de chercher le minimum de la fonctionnelle $J(U)$ (voir le théorème de Lax-Milgram) :

$$J(U) = \frac{1}{2}U^\top AU - B^\top U \tag{2.78}$$

Si les conditions limites peuvent s'écrire comme un système de $r$ équations linéaires, on peut les écrire sous forme matricielle :

$$RU = C \tag{2.79}$$

avec $R$ une matrice $(r \times m)$ et $C$ une matrice colonne de $r$ éléments. On introduit $\lambda$ le multiplicateur de Lagrange sous forme de matrice colonne de $r$ éléments. On doit minimiser la fonctionnelle :

$$F = \frac{1}{2}U^\top AU - B^\top U + \lambda^\top (RU - C) \tag{2.80}$$

c'est-à-dire annuler la variation de la fonctionnelle :

$$\forall(\delta U, \delta\lambda), \quad \delta F = \delta U^\top (AU - B + R^\top \lambda) + \delta\lambda^\top (RU - C) = 0 \tag{2.81}$$

Cela revient donc à résoudre le système :

$$\begin{pmatrix} A & R^\top \\ R & 0 \end{pmatrix} \begin{pmatrix} U \\ \lambda \end{pmatrix} = \begin{pmatrix} B \\ C \end{pmatrix} \tag{2.82}$$

La matrice définie par bloc est clairement inversible. Cette méthode est coûteuse par rapport à la méthode de pénalisation car on a une matrice $(m + r) \times (m + r)$ à inverser (ajout de $r$ variables).

## 2.3 Volumes finis

La méthode des éléments finis présente un défaut majeur : à moins de choisir des fonctions d'interpolation bien spécifiques (par exemple à divergence nulle), la conservation des flux de $u$ n'est pas garantie. D'un point de vue physique, cela veut dire que la conservation de la masse, de l'énergie, *etc* ... n'est pas exactement vérifiée. En fait, elle l'est aux erreurs numériques près. La vérification rigoureuse de ces bilans n'est pas toujours nécessaire mais, parfois, même un petit reste finit par s'accumuler, surtout lorsque le problème n'est pas stationnaire. Par exemple, lorsque le fluide est compressible, il a une capacité à accumuler de la matière en se comprimant. Ainsi, les éléments finis sont souvent utilisés pour les fluides incompressibles mais rarement pour les fluides compressibles.

Les volumes finis utilisent une approche numérique qui vise à **conserver rigoureusement** les flux numériques. En effet, cette méthode a été introduite à l'origine pour résoudre des lois de conservation.

### 2.3.1 Loi de conservation

La méthode a été mise en place pour résoudre les équations décrivant une loi de conservation d'une quantité $q$. Une telle équation est la traduction mathématique de : la variation temporelle de quantité $q$ dans un volume $V$ est égale à la somme du flux entrant $\phi_e$ de $q$ dans $V$ et à sa production $S_q$ dans $V$ (terme source) :

$$\frac{dq}{dt} = \phi_e(q) + S_q \tag{2.83}$$

En remplaçant $\phi_e$ par le flux sortant $\phi$, on a :

$$\frac{dq}{dt} + \phi(q) = S_q \tag{2.84}$$

**Définition 4.** *Le flux $\phi$ d'une quantité $q$ à travers une surface $S$ de normale sortante $\vec{n}$ est défini par :*

$$\phi(q) = \iint_S \vec{F}\vec{n}dS \tag{2.85}$$

*avec $\vec{F}$ la densité de flux de $q$.*

**Définition 5.** *La quantité $q$ contenue dans un volume $V$ est donnée par :*

$$q = \iiint_V u dV \tag{2.86}$$

*avec $u$ la densité volumique de $q$*

**Définition 6.** *On peut aussi définir une production volumique locale $s_q$ de $q$ par :*

$$S_q = \iiint_V s_q dV \tag{2.87}$$

Compte tenu des différentes définitions, on peut réécrire l'équation 2.84 :

$$\iiint_V \frac{\partial u}{\partial t} dV + \iint_S \vec{F}\vec{n} dS = \iiint_V s_q dV \tag{2.88}$$

**Théorème 5.** *Green-Ostrogradski*

*Soit $V$ un volume délimité par une surface $S$, on a :*

$$\iiint_V div(\vec{F}) dV = \iint_S \vec{F}\vec{n} dS \tag{2.89}$$

Ainsi, on obtient l'équation d'une loi de conservation d'une quantité $q$ sous forme globale ou intégrale dans un volume $V$ :

$$\iiint_V \left( \frac{\partial u}{\partial t} + div(\vec{F}) - s_q \right) dV = 0 \tag{2.90}$$

Si cette loi de conservation est valable pour tout sous-volume ou sous-domaine $V = D_i$ d'un domaine $D$, alors l'équation globale 2.90 fait apparaître une edp locale, valable en tout point de $D$ :

$$\frac{\partial u}{\partial t} + div(\vec{F}) - s_q = 0 \tag{2.91}$$

La forme locale 2.91 ne garantit pas la conservation des flux dans un volume $V$ à cause des erreurs numériques. Il s'agit donc de réécrire l'edp décrivant le système 2.1 à résoudre tel que nous l'avons vu dans les sections précédentes sous la forme d'une équation de conservation 2.84. On ne cherche pas à résoudre l'équation locale mais l'équation intégrale 2.90 sur un volume $D_i$ afin de garantir numériquement la conservation dans ce volume. On cherche bien la densité locale $u$ de $q$. La densité de flux $\vec{F}$ est une fonction de $u$ qui définit le flux via la relation 2.85. Par exemple, si $u$ est la masse volumique $\rho$, on a $\vec{F} = \rho\vec{v}$, avec $\vec{v}$ la vitesse de convection.

Comme précédemment, on suppose que l'on connait $u$ en des points $x_i$ du maillage et que sa valeur locale est $u_i$. Entre les points du maillage, on peut avoir une interpolation à partir des valeurs $u_i$ comme en éléments finis. On définit les volumes $D_i$ à partir d'un sous-ensemble de points $x_j$ du maillage. Ces volumes sont tout à fait comparables aux éléments des éléments finis. Cependant, si les points $x_j \in D_i$, il n'y a pas de règle générale pour leur positionnement dans $D_i$ : ils peuvent être à l'intérieur de $D_i$ ou sur sa surface externe (son enveloppe). Bien entendu, comme pour les éléments finis :

$$D = \bigcup_i D_i \text{ et } D_i \cap D_{j \neq i} = \emptyset \tag{2.92}$$

D'après l'équation 2.88, on a sur notre maillage un système d'équations qui fait intervenir tous les volumes (cellules) $D_i$ :

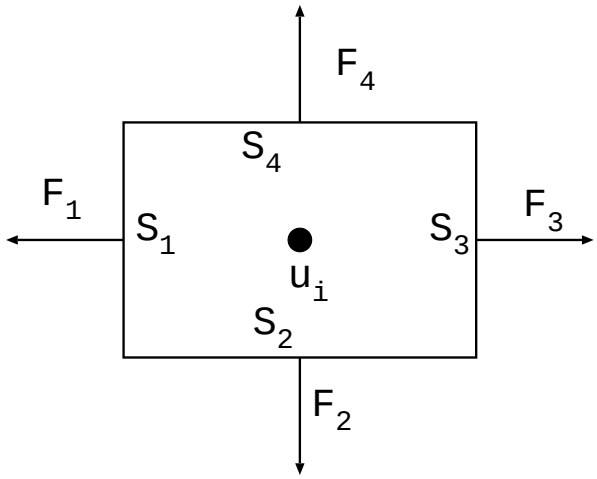


FIGURE 2.4 – Cellule centrée. La valeur moyenne du champ est donnée au centre de la cellule. Les nœuds du maillage sont au centre des cellules.

$$\begin{cases} \iiint_{D_1} \frac{\partial u}{\partial t} dV + \iint_{S_1} \vec{F}(u)\vec{n}dS = \iiint_{D_1} s_q dV \\ \vdots \\ \iiint_{D_N} \frac{\partial u}{\partial t} dV + \iint_{D_N} \vec{F}(u)\vec{n}dS = \iiint_{D_N} s_q dV \end{cases} \tag{2.93}$$

Pour résoudre le système 2.93, on introduit la moyenne d'un champ $u$ :

**Définition 7.** *Le champ moyen $\bar{u}_i$ dans un volume $D_i$ est défini par :*

$$\bar{u}_i = \frac{1}{V_i} \iiint_{D_i} u dV \tag{2.94}$$

*avec $u$ la définition locale du champ.*

On obtient donc :

$$\begin{cases} \frac{d\bar{u}_1}{dt} + \frac{1}{V_1}\phi_1 = \bar{s}_1 \\ \vdots \\ \frac{d\bar{u}_N}{dt} + \frac{1}{V_N}\phi_N = \bar{s}_N \end{cases} \tag{2.95}$$

Les $N$ relations entre les flux et le champ moyen dans la cellule, via la relation 2.85 permettent de fermer le problème :

$$\phi_i = \sum_j \vec{F}_j(\bar{u}_i)\vec{n}_j S_j \tag{2.96}$$

avec $\vec{n}_j$ le vecteur normal de la face $j$ de surface $S_j$ de la cellule $D_i$. Le flux $F_j(\bar{u}_i)$ est donné par l'équation de conservation de départ. La méthode des volumes finis ne donne donc pas les valeurs locales du champ mais ses valeurs moyennes dans les cellules $D_i$. En pratique, lorsque le volume $V_i$ est petit, on peut utiliser cette valeur moyenne comme une approximation de la valeur locale. La façon de définir la relation entre les valeurs de $u$ aux nœuds du maillage et $\bar{u}_i$ à la position des nœuds dans la cellule $D_i$ détermine la précision du schéma de discrétisation spatiale. Par exemple, dans les schémas 'cellule centrée', la valeur moyenne de $u$ est affectée au centre de la cellule $D_i$ (voir figure 2.4).

La dérivée par rapport au temps se discrétise en général avec un schéma aux différences finis. Les flux dépendent linéairement de $u$ dans de nombreuses équations de conservation. Sinon, on se ramène à

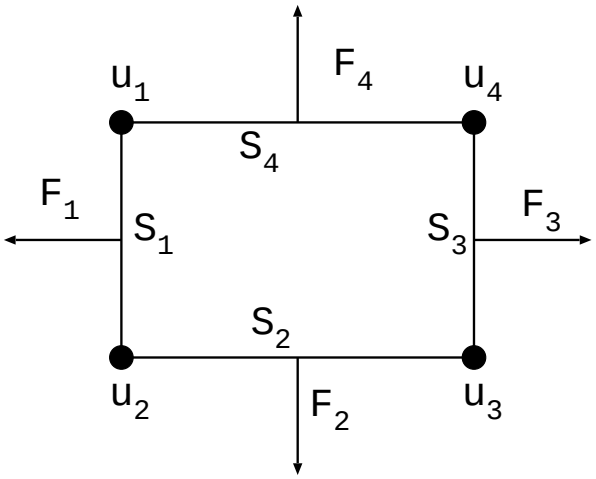


FIGURE 2.5 – Cellule centrée sur les sommets. La valeur moyenne du champ est donnée par la moyenne des valeurs périphériques. Les nœuds du maillage définissent les faces de la cellule.

un problème linéaire par linéarisation entre deux instants, $t$ et $t + \Delta t$, ou encore en utilisant la matrice jacobienne. Dans ce cas, on retrouve la forme matricielle du problème :

$$AU = B \tag{2.97}$$

avec

$$U = \begin{pmatrix} \bar{u}_1 \\ \vdots \\ \bar{u}_N \end{pmatrix} \tag{2.98}$$

Il ne faut pas oublier d'intégrer les conditions limites au système en utilisant les méthodes précedemment vues dans le cours. En fonction de la méthode volumes finis choisie, les valeurs moyennes sont prises au centre des cellules ou aux nœuds du maillage. En effet, on peut tout à fait considérer différentes façons de construire le maillage soit la valeur de $u$ est prise sur les sommets de la cellule (voir figure 2.5) soit au centre (voir figure 2.4).

Les volumes finis ne sont pas une méthode réellement distincte des deux précédentes (différences finies et éléments finis). Il s'agit plutôt d'une mise en forme particulière de l'edp à résoudre pour garantir la conservation. Contrairement à la méthode des éléments finis, on résoud les équations sous une formulation forte et contrairement aux différences finies, l'équation est écrite sous forme intégrale et non locale. L'équation doit être écrite sous forme d'une équation de conservation, c'est-à-dire qu'il faut définir un flux. Heureusement, beaucoup d'équations physiques sont de ce type (équation de la chaleur par exemple).

## Bibliographie

DELAUNAY, BORIS 1934 Sur la sphère vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk* **7**, 793–800.

## 2.4 Travaux Dirigés

### 2.4.1 Exercice : Différences finies

On veut résoudre une équation de convection-diffusion correspondant au transfert de chaleur entre deux points A et B par un fluide en écoulement, de masse volumique $\rho$, de chaleur spécifique massique $c$ et de conductivité thermique $\lambda$ :

$$\vec{v}.\vec{\nabla}T = a\nabla^2 T \tag{2.99}$$

avec $a = \lambda/\rho c$.

On se limite au cas 1D, c'est-à-dire que $\vec{v} = u\vec{e}_x$ et $T = T(x)$. La distance entre A et B est fixée à 1 (longueur de référence) et l'équation doit être comprise comme une équation sans dimension. Le fluide est incompressible donc l'équation de conservation impose que $u$ est une constante.

Par la suite, on prendra $u = 1$ et $a = 0.3$ (on pourra essayer d'autres valeurs en fonction du temps disponible).

1. Résoudre l'équation à la main avec des conditions limites de Dirichlet, *i.e.* $T(0) = T_A$ et $T(1) = T_B$. Faire de même avec une condition de Dirichlet $T(0) = T_A$ et une condition de Neumann $T'(0) = q$.
2. On commence avec un maillage homogène de $n$ points également espacés de $\Delta x$. Écrire l'équation avec un schéma centré d'ordre 2 en différences finies.
3. Écrire le problème sous forme matricielle et intégrer les conditions limites (Dirichlet et Neumann).
4. Sous Mathematica, écrire un programme pour résoudre le problème avec $T_A = 10$ et $T_B = 1$ puis $q = -1$. Avec la condition de Neumann, quelle est la température $T_B$ donnée par le calcul numérique avec $n = 10$ ?
5. Donner l'erreur par rapport à la solution analythique en fonction de $n$ avec les conditions limites de Dirichlet. Retrouver numériquement l'ordre du schéma. Combien faut-il de points pour avoir une erreur relative de moins de 0.5% ? (On pourra éventuellement faire de même avec la condition de Neumann.)
6. Maintenant, on va utiliser un maillage non homogène de $n$ points. Proposer un schéma à l'ordre 1 qui utilise les valeurs $T_{i-1}$, $T_i$ et $T_{i+1}$ aux points $x_{i-1}$, $x_i$ et $x_{i+1}$. Vérifier que le schéma correspond au schéma d'ordre 2 précédent quand $\forall i$, $x_i - x_{i-1} = \Delta x$.
7. Donner la loi de récurrence entre les points $x_{i+1}$, $x_i$ et $x_{i-1}$ pour avoir une variation constante de la température entre les points, *i.e.* $T_{i+1} - T_i = K$.
8. Proposer un algorithme pour générer un maillage à partir d'une solution du problème obtenue avec un maillage régulier.
9. Sous Mathematica, écrire un programme pour résoudre le problème.
10. Donner l'erreur par rapport à la solution analythique en fonction de $n$. Comparer le nombre de points nécessaires pour une précision de 0.5% avec un maillage homogène.
11. Méthode alternative : sans réduire l'ordre de l'équation différentielle en introduisant le champ $Y(x) = T'(x)$. Écrire les équations discrètes avec un schéma numérique centré sur le point $i + 1/2$ d'ordre 2 pour la dérivée.

### 2.4.2 Problème : Stabilité d'un écoulement de Taylor-Couette

L'écoulement de Taylor-Couette entre deux cylindres concentriques (figure 2.6) a fait l'objet de très nombreux travaux scientifiques, tant pour son intérêt académique que pratique (un arbre en rotation dans un carter cylindrique). Depuis les travaux de Taylor (1923), on sait que cet écoulement devient instable lorsque la vitesse du cylindre interne dépasse une valeur critique.

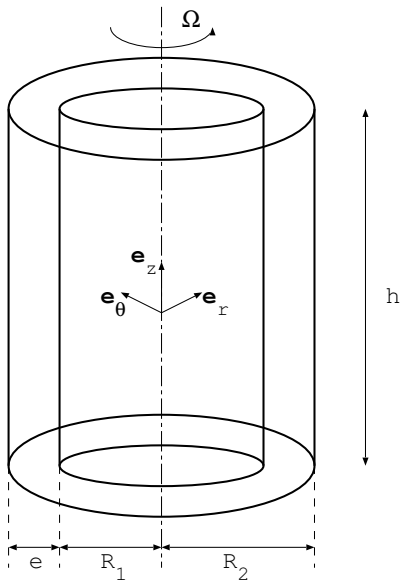


FIGURE 2.6 – Géométrie d'un dispositif de Taylor-Couette. Dans notre cas, seul le cylindre interne est en rotation.

Nous considèrerons ici le cas représenté par la figure (2.6) où seul le cylindre interne est en rotation avec la vitesse angulaire $\Omega$. On considère que l'écoulement est périodique suivant l'axe de rotation $\vec{e}_z$ entre la base $z = 0$ et $z = h$. Le fluide dans l'entrefer $e$ est Newtonien et incompressible, de masse volumique $\rho$ et de viscosité $\mu$. On note $\vec{u} = u\vec{e}_r + v\vec{e}_\theta + w\vec{e}_z$ la vitesse du fluide dans la base cylindrique.

Les équations du mouvement sont écrites en coordonnées cylindriques et sous forme adimensionnelles (voir ou revoir les polycopiés du cours de Plaut (2011*a*) pour les opérateurs en coordonnées cylindriques et Plaut (2011*b*) pour les équations de Navier-Stokes) :

$$\frac{1}{r}\frac{\partial}{\partial r}(ru) + \frac{1}{r}\frac{\partial v}{\partial \theta} + \frac{\partial w}{\partial z} = 0 \tag{2.100}$$

$$\begin{aligned}\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial r} + \frac{v}{r}\left(\frac{\partial u}{\partial \theta} - v\right) + w\frac{\partial u}{\partial z} &= -\frac{\partial p}{\partial r} \\ &+ \frac{1}{\sqrt{Ta}}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} + \frac{\partial^2 u}{\partial z^2} - \frac{2}{r^2}\frac{\partial v}{\partial \theta} - \frac{u}{r^2}\right]\end{aligned} \tag{2.101}$$

$$\begin{aligned}\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial r} + \frac{v}{r}\left(\frac{\partial v}{\partial \theta} + u\right) + w\frac{\partial v}{\partial z} &= -\frac{1}{r}\frac{\partial p}{\partial \theta} \\ &+ \frac{1}{\sqrt{Ta}}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 v}{\partial \theta^2} + \frac{\partial^2 v}{\partial z^2} + \frac{2}{r^2}\frac{\partial u}{\partial \theta} - \frac{v}{r^2}\right]\end{aligned} \tag{2.102}$$

$$\begin{aligned}\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial r} + \frac{v}{r}\frac{\partial w}{\partial \theta} + w\frac{\partial w}{\partial z} &= -\frac{\partial p}{\partial z} \\ &+ \frac{1}{\sqrt{Ta}}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial w}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 w}{\partial \theta^2} + \frac{\partial^2 w}{\partial z^2}\right]\end{aligned} \tag{2.103}$$

avec le nombre de Taylor $Ta$ définit de la même manière que Chandrasekhar (1961) :

$$Ta = 4\left(\frac{\rho\Omega}{\mu}\right)^2 \frac{R_1^2}{R_1+R_2} e^3 \tag{2.104}$$

Le nombre de Taylor est équivalent à un nombre de Reynolds. Pour obtenir ce nombre, on doit choisir l'entrefer $e$ comme longueur de référence et la vitesse de référence est alors définie par $V_{ref} = 2\Omega R_1\sqrt{e/(R_1+R_2)}$. Il reste à définir le rapport entre les rayons interne et externe $\eta = R_1/R_2$ pour que le problème soit totalement posé.

Afin de ne pas alourdir l'écriture, $R_1$ désigne le rayon sans dimension $(R_1/e)$. Il en va de même pour toutes les variables définies précédemment.

1. Donner $R_1$ et $R_2$ en fonction de $\eta$ sachant que $e = 1$ (longueur de référence).
2. Donner $\Omega$ en fonction de $\eta$ sachant que $V_{ref} = 1$ (vitesse de référence).
3. On cherche une solution de base sous la forme $\vec{u} = V(r)\vec{e}_\theta$. À partir des équations de Navier-Stokes, donner l'équation différentielle que vérifie $V(r)$.
4. On cherche une solution sous la forme $V(r) = Kr^\alpha$ avec $K$ une constante. Que vaut $\alpha$ ?
5. On déduit de la question précédente que $V(r) = Ar + B/r$ avec $A$ et $B$ deux constantes. Donner ces constantes pour vérifier les conditions limites sur les rayons interne et externe.
6. Pour étudier la stabilité de l'écoulement de base, on va s'intéresser à l'évolution d'une perturbation $\vec{v} = u_p\vec{e}_r + v_p\vec{e}_\theta + w_p\vec{e}_z$ de la vitesse de base. On a donc la vitesse totale de l'écoulement perturbé $\vec{u} = V(r)\vec{e}_\theta + \vec{v}$. On note $p$ la perturbation de la pression et $P_b$ la pression de l'écoulement de base. Linéariser les équations vérifiées par la perturbation de vitesse $\vec{v}$. Si les conditions limites sur $\vec{u}$ sont strictement respectées sur les rayons interne et externe ainsi qu'en $z = 0$ et $z = h$, donner les conditions sur $\vec{v}$.
7. On suppose que l'on peut écrire :

$$u_p = exp(\sigma t)exp(im\theta)exp(ikz)\tilde{u}(r) + cc \tag{2.105}$$
$$v_p = exp(\sigma t)exp(im\theta)exp(ikz)\tilde{v}(r) + cc \tag{2.106}$$
$$w_p = exp(\sigma t)exp(im\theta)exp(ikz)\tilde{w}(r) + cc \tag{2.107}$$
$$p = exp(\sigma t)exp(im\theta)exp(ikz)\tilde{p}(r) + cc \tag{2.108}$$

   avec $cc$ désignant le complexe conjugué. $\sigma$ est une valeur complexe, $m$ un entier et $k = 2\pi/h$. $\tilde{u}(r)$, $\tilde{v}(r)$, $\tilde{w}(r)$ et $\tilde{p}(r)$ sont des fonctions complexes de $r$. Écrire le système d'équations différentielles linéaires que vérifient ces fonctions de $r$.
8. On discrétise $\tilde{u}(r)$, $\tilde{v}(r)$ et $\tilde{w}(r)$ sur $N$ points $r_j$ également espacés entre $R_1$ et $R_2$ (indice $j$ de 1 à $N$). La pression $\tilde{p}(r)$ est prise au centre de deux points successifs de l'espace discret des vitesses $r_{j-1/2}$ (indice $j$ de 1 à $N+1$). Les dérivées première et seconde par rapport à $r$ sont calculées avec un schéma centré à l'ordre 2 en différences finies. On désigne par $U$ la matrice colonne qui contient les valeurs $\tilde{u}(r_j)$, $\tilde{v}(r_j)$ et $\tilde{w}(r_j)$. L'équation de conservation de la masse écrite aux points $j-1/2$ peut s'écrire sous forme matricielle $DU = 0$ avec $D$ une matrice $(N+1)\times 3N$ qui permet de calculer la divergence à l'ordre 2. Donner $D$.
9. De même, on peut écrire sous forme matricielle le gradient de $p$ aux points $j$ sous la forme $GP$ avec $G$ une matrice $3N \times (N+1)$. Écrire $G$.

10. La linéarisation des termes d'inertie aux points $j$ peut aussi s'écrire sous forme matricielle $LiU$, avec $Li$ une matrice $3N \times 3N$. Donner $Li$.
11. Procéder de même pour le terme de dissipation visqueuse aux points $j$ (Laplacien) en l'écrivant sous la forme $LU$ avec $L$ une matrice $3N \times 3N$. Donner $L$.
12. Montrer que le système linéarisé peut alors s'écrire :

$$DU = 0 \tag{2.109}$$
$$\sigma U = -GP + (L - Li)U \tag{2.110}$$

13. En appliquant la divergence $D$ sur la deuxième équation du système (2.110), montrer que l'on peut avoir la pression $P$ en fonction de la vitesse $U$ sachant que $DG$ est inversible. En déduire l'équation matricielle vérifiée par $U$. Montrer alors que la solution d'un tel système est le vecteur propre d'une matrice que vous donnerez et que $\sigma$ est sa valeur propre associée. L'étude de stabilité consiste à vérifier le signe de la partie réelle de $\sigma$ : si cette dernière est positive, l'écoulement est linéairement instable.
14. Réaliser un programme Mathematica permettant de calculer la matrice de l'opérateur linéaire, de rechercher ses vecteurs propres et ses valeurs propres et de vérifier si l'une de ces dernières à une partie réelle positive. Faite des essais avec $m = 0$, $k = 3.12$ et $\eta = 0.99$ pour différentes valeurs du nombre de Taylor. La valeur critique attendue est $Ta \simeq 3413$.
15. Toujours avec Mathématica, reconstituer le champ de vitesse correspondant au mode le moins stable (dont la partie réelle de la valeur propre est maximal). Tracer le contour des isovaleurs de vitesse dans le plan $(r, z)$. On pourra aussi tracer le contour des isovaleurs du rotationnel de la vitesse. On voit alors la structure des tourbillons de Taylor.

## Bibliographie

CHANDRASEKHAR, S. 1961 Hydrodynamic and hydromagnetic stability. *Dover Publications Inc. New York* .

PLAUT, EMMANUEL 2011*a* Mécanique des milieux continus solides et fluides, Tome 0, Le calcul tensoriel : outil mathématiques pour la physique des milieux continus. École des Mines, FICM 1A.

PLAUT, EMMANUEL 2011*b* Mécanique des milieux continus solides et fluides, Tome 2, Fluides - Analyse dimensionnelle et similitude. École des Mines, FICM 1A.

TAYLOR, G.I. 1923 Stability of a viscous liquid contained between two rotating cylinders. *Phil. Trans. R. Soc. Lond. A* **223**, 289.

### 2.4.3 Exercice : Éléments finis

On veut résoudre une équation de convection-diffusion correspondant au transfert de chaleur entre deux points A et B par un fluide en écoulement, de masse volumique $\rho$, de chaleur spécifique massique $c$ et de conductivité thermique $\lambda$ :

$$\vec{v}.\vec{\nabla}T = a\nabla^2 T \tag{2.111}$$

avec $a = \lambda/\rho c$.

On se limite au cas 1D, c'est-à-dire que $\vec{v} = u\vec{e}_x$ et $T = T(x)$. La distance $L$ entre A et B est fixée à 1 (longueur de référence) et l'équation doit être comprise comme une équation sans dimension. Le fluide est incompressible donc l'équation de conservation impose que $u$ est une constante.

Par la suite, on prendra $u = 1$ et $a = 0.3$ (on pourra essayer d'autres valeurs en fonction du temps disponible).

1. Résoudre l'équation 2.111 à la main avec des conditions limites :
   1. de Dirichlet, *i.e.* $T(0) = T_A$ et $T(1) = T_B$ ;
   2. de Dirichlet $T(0) = T_A$ et une condition de Neumann $T'(L) = q_B$ ;
   3. de Dirichlet $T(0) = T_A$ et une condition de Neumann $T'(0) = q_A$.
2. On décide de résoudre l'edp (2.111) avec des éléments finis. Écrire la formulation faible de l'edp (2.111) sur un segment de longueur $L$. On notera $v$ la fonction test.
3. On choisit des éléments de type P1 (polynômes d'interpolation d'ordre 1). Le maillage est réguliers. Donner le pas $\Delta x$ en fonction du nombre de points $n$. Faire un schéma des éléments $i$ et $i+1$ entre les points $x_{i-1}$, $x_i$ et $x_{i+1}$ avec les fonctions d'interpolation $\varphi_{i-1}$, $\varphi_i$ et $\varphi_{i+1}$.
4. Discrétiser la forme faible de l'edp (2.111) sur les éléments finis, c'est-à-dire sur les fonctions d'interpolation $\varphi_i$. On se ramènera à une intégration entre 0 et $\Delta x$ et on utilisera les variables locales $\xi_i$ pour chaque élément $i$. Expliciter les fonctions d'interpolation $\varphi_i$ à l'aide des variables locales $\xi_i$. Calculer les intégrales en fonction de $\Delta x$ et des valeurs $T_i$ aux points $x_i$ pour donner la forme discrète finale de la formulation faible de l'edp.
5. Écrire le système discret d'équations sous forme matricielle.
6. Intégrer les conditions limites de Dirichlet par la méthode de pénalisation. On choisit un coefficient de pénalisation $K = 10^{30}$.
7. Sous Mathematica, écrire un programme pour résoudre le système matriciel obtenu avec des éléments finis P1 de l'edp (2.111) et le couple 1 de conditions limites.
8. Intégrer le deuxième couple de conditions limites est assez facile. Par contre, le troisième couple de conditions limites interdit l'utilisation de la méthode de pénalisation. Réécrire le problème matriciel pour résoudre le problème avec ce couple de conditions limites. Indication : retirer $T_1$ ($T_A$) des inconnues.
9. Sous Mathematica, écrire un programme pour résoudre le problème avec les conditions limites 2 et 3.
10. Sous Mathematica, écrire un programme pour retrouver l'ordre du schéma numérique à partir de l'erreur par rapport à la solution analytique avec les conditions limites 1.

### 2.4.4 Exercice : Volumes finis

On veut résoudre une équation de convection-diffusion correspondant au transfert de chaleur entre deux points A et B par un fluide en écoulement, de masse volumique $\rho$, de chaleur spécifique massique $c$ et de conductivité thermique $\lambda$ :

$$\vec{v}.\vec{\nabla}T = a\nabla^2 T \tag{2.112}$$

avec $a = \lambda/\rho c$.

On se limite au cas 1D, c'est-à-dire que $\vec{v} = u\vec{e}_x$ et $T = T(x)$. La distance $L$ entre A et B est fixée à 1 (longueur de référence) et l'équation doit être comprise comme une équation sans dimension. Le fluide est incompressible donc l'équation de conservation impose que $u$ est une constante.

Par la suite, on prendra $u = 1$ et $a = 0.3$ (on pourra essayer d'autres valeurs en fonction du temps disponible).

1. Résoudre l'équation 2.112 à la main avec des conditions limites (voir TD03) :
   1. de Dirichlet, *i.e.* $T(0) = T_A$ et $T(1) = T_B$ ;
   2. de Dirichlet $T(0) = T_A$ et une condition de Neumann $T'(L) = q_B$ ;
   3. de Dirichlet $T(0) = T_A$ et une condition de Neumann $T'(0) = q_A$.
2. On décide de résoudre l'edp (2.112) avec des volumes finis. Pour cela, on décide d'utiliser une discrétisation spatiale de type différence finie, c'est-à-dire que l'on considère les valeurs $T_i = T(x_i)$. Le pas spatial $\Delta x$ est constant. Le volume fini est défini comme étant le segment de longueur $\Delta x$ de centre $x_i$ (voir figure 2.7). Intégrez l'équation aux dérivées partielles 2.112 sur le volume fini $i$

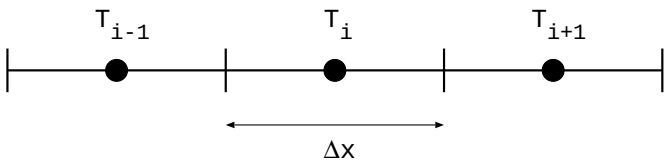


FIGURE 2.7 – Domaine de calcul et définition des volumes finis.

   centré sur $x_i$. On pourra utiliser des indices en demi-entier pour les bords du volume, *i. e.* $i - 1/2$ et $i + 1/2$ respectivement.
3. Proposer un schéma centré à l'ordre 2 pour calculer les valeurs de $T_{i-1/2}$, $T_{i+1/2}$, $T'_{i-1/2}$ et $T'_{i+1/2}$.
4. Écrire l'équation obtenue sur la cellule $i$ avec $2 \leq i \leq n - 1$.
5. Intégrer les conditions limites 1 et écrire le problème à résoudre sous forme matricielle. On reconnaitra un système déjà vu. Cela justifie que les volumes finis ne constituent pas une méthode à proprement parler mais une mise en forme particulière de l'équation 2.112 qui assure la conservation des flux.
6. Écrire le système à résoudre avec les conditions limites 2. Indication : écrire les intégrales sur la cellule $n$ et utiliser les schémas d'ordre 2 de la question 3 avec un bord virtuel en $n + 1/2$.
7. Écrire le système à résoudre avec les conditions limites 3.
8. Question facultative : Sous Mathematica, écrire un programme avec les conditions limites 1, 2 et 3. Retrouver l'ordre du schéma numérique à partir de l'erreur par rapport à la solution analytique avec les conditions limites 1.

Chapitre 3

# Méthodes numériques de résolution d'un problème

On s'intéresse maintenant aux problèmes instationnaires :

$$\frac{\partial u}{\partial t} + f(u) = 0 \tag{3.1}$$

L'opérateur de dérivée temporelle est quasiment toujours discrétisé par la méthode des différences finies. On note $u^{n+1}$ la valeur du champ $u$ à l'instant $t+\Delta t$ et $u^n$ sa valeur à l'instant $t$. $\Delta t$ est donc le pas de temps. On veut donc calculer $u^{n+1}$, connaissant les valeurs de $u$ aux instants antérieurs. L'équation 3.1 s'écrit de manière rigoureuse à l'intant $t + \Delta t$ :

$$\frac{\partial u^{n+1}}{\partial t} + f(u^{n+1}) = 0 \tag{3.2}$$

La discrétisation de la dérivée temporelle en différences finies donne à l'ordre 1 :

$$\frac{\partial u^{n+1}}{\partial t} \simeq \frac{u^{n+1} - u^n}{\Delta t} \tag{3.3}$$

On pourrait utiliser un schéma à un ordre plus élevé, mais le sens de déroulement du temps impose de n'utiliser que des $u^k$ avec $k \leq n + 1$.

## 3.1 Schéma explicite

Comme on connait $u$ aux instants antérieurs à $n + 1$, on peut écrire :

$$f(u^{n+1}) \simeq f(u^n) \tag{3.4}$$

à l'ordre 1 et on obtient le schéma explicite :

$$\frac{\partial u^{n+1}}{\partial t} + f(u^n) = 0 \tag{3.5}$$

En utilisant la discrétisation temporelle 3.3, on obtient :

$$\boxed{u^{n+1} = u^n - \Delta t f(u^n)} \tag{3.6}$$

Comme on connait $u^k$, $k \leq n$, ce schéma permet un calcul explicite de $u^{n+1}$. Malheureusement, c'est un schéma au mieux conditionnellement stable : il faut utiliser un pas de temps suffisamment petit pour assurer la stabilité numérique du schéma (dans certains cas, le schéma est même instable $\forall \Delta t$). On peut s'en convaincre en prenant un exemple très simple pour $f$ :

$$f(u) = au \tag{3.7}$$

avec $a$ un nombre réel. On a alors la relation de récurrence :

$$u^{n+1} = (1 - a\Delta t)u^n \tag{3.8}$$

qui s'écrit de façon explicite, avec $u^0$ la valeur initiale de $u$ :

$$u^{n+1} = (1 - a\Delta t)^{n+1}u^0 \tag{3.9}$$

Il est clair que cette suite diverge si :

$$|1 - a\Delta t| > 1 \tag{3.10}$$

ce qui donne une condition sur $\Delta t$ :

$$\Delta t < \begin{cases} 0 & \text{si } a \text{ est négatif. } \Rightarrow \text{ instable.} \\ 2/a & \text{si } a \text{ est positif. } \Rightarrow \text{ conditionnellement stable.} \end{cases} \tag{3.11}$$

Malgré sa simplicité, cet exemple peut se généraliser lorsque $f$ est linéaire. Dans ce cas, $a = \|f\|$, c'est-à-dire, $a = \|A\|$ avec $A$ la matrice représentant $f$.

Pour conclure, les schémas explicites sont les plus simples. Mais, leur stabilité n'est pas acquise et la condition de stabilité numérique sur $\Delta t$ peut imposer un pas de temps très petit et donc rendre la simulation pendant un temps donné très longue.

## 3.2 Schéma implicite

Si on veut être plus rigoureux, on garde la forme 3.2 de l'équation qui est la forme implicite. En introduisant la discrétisation temporelle, il vient :

$$\boxed{u^{n+1} + \Delta t f(u^{n+1}) = u^n} \tag{3.12}$$

Si on introduit l'identité $Id$, on a :

$$[Id + \Delta t f](u^{n+1}) = u^n \tag{3.13}$$

Comme $u^n$ est connu, cela veut dire qu'il faut inverser l'opérateur $[Id + \Delta t f]$, ce qui peut être compliqué et couteux. Pour cela, on utilise les mêmes méthodes que pour les équations stationnaires que nous avons vues précedemment, sauf que l'on répète l'opération à chaque pas de temps.

Si on reprend notre exemple simple, on a :

$$u^{n+1} = \frac{1}{1 + a\Delta t} u^n \tag{3.14}$$

c'est-à-dire que :

$$u^{n+1} = \frac{1}{(1 + a\Delta t)^{n+1}} u^0 \tag{3.15}$$

Le schéma est stable si la suite $u^{n+1}$ est bornée, c'est-à-dire :

$$\Delta t > \begin{cases} 2/|a| & \text{si } a \text{ est négatif. } \Rightarrow \text{ conditionnellement stable.} \\ 0 & \text{si } a \text{ est positif. } \Rightarrow \text{ inconditionnellement stable.} \end{cases} \tag{3.16}$$

On voit donc qu'en terme de stabilité, on a beaucoup arrangé les choses. Cependant, il reste un problème : la précision du schéma. En effet, il faut que les deux schémas explicite 3.8 et implicite 3.14 donnent un peu près le même résultat numérique. Or on constate que si on fait un développement en série de Taylor du schéma implicite, on a

$$\frac{1}{1 + a\Delta t} = 1 - a\Delta t + \bigcirc\left((a\Delta t)^2\right) \tag{3.17}$$

Ainsi donc, il faut que $|a\Delta t| \ll 1$ pour que ce que l'on fasse ait un sens. Ainsi, le choix de $\Delta t$ n'est pas guidé par des problèmes de stabilité numérique avec un schéma implicite, mais par des problèmes de précision numérique.

## 3.3 Schéma de Crank-Nicolson

Les schémas explicites et implicites que nous avons vus précédemment ont une résolution temporelle d'ordre 1. On peut améliorer les choses en prenant un schéma centré à l'ordre 2 pour la dérivée temporelle à l'instant $t + \Delta t/2$, soit $n + 1/2$ :

$$\frac{\partial u^{n+1/2}}{\partial t} \simeq \frac{u^{n+1} - u^n}{\Delta t} \tag{3.18}$$

L'équation 3.2 s'écrit :

$$\frac{\partial u^{n+1/2}}{\partial t} + f(u^{n+1/2}) = 0 \tag{3.19}$$

On veut donc une valeur approchée de $f(u^{n+1/2})$ qui fasse intervenir $u^{n+1}$ et $u^n$. Or, on montre que :

$$f(u^{n+1/2}) = \frac{1}{2}\left(f(u^{n+1}) + f(u^n)\right) + \bigcirc\left(\Delta t^2\right) \tag{3.20}$$

On obtient ainsi le schéma de Crank-Nicolson à l'ordre 2 :

$$\frac{u^{n+1} - u^n}{\Delta t} + \frac{1}{2}\left(f(u^{n+1}) + f(u^n)\right) = 0 \tag{3.21}$$

En séparant ce qu'on connait de ce qu'on cherche on a :

$$\boxed{u^{n+1} + \frac{\Delta t}{2} f(u^{n+1}) = u^n - \frac{\Delta t}{2} f(u^n)} \tag{3.22}$$

Le schéma de Crank-Nicolson est un schéma semi-implicite car il fait intervenir $f(u^{n+1})$ et $f(u^n)$. Il est souvent utilisé car il permet une meilleure précision que le schéma implicite et une meilleure stabilité que le schéma explicite.

Si on reprend encore une fois notre exemple simple, on trouve :

$$u^{n+1} = \left(\frac{1 - a\Delta t/2}{1 + a\Delta t/2}\right)^{n+1} u^0 \tag{3.23}$$

Pour assurer la convergence de la suite $u^{n+1}$, on montre que :

$$\begin{cases} \Delta t < 0 & \text{si } a \text{ est négatif. } \Rightarrow \text{ instable.} \\ \Delta t > 0 & \text{si } a \text{ est positif. } \Rightarrow \text{ inconditionnellement stable.} \end{cases} \tag{3.24}$$

On peut généraliser le schéma de Crank-Nicolson en prenant un temps intermédiaire non-centré, ce qui amème à des schémas du type :

$$u^{n+1} + \frac{\alpha\Delta t}{\alpha + \beta} f(u^{n+1}) = u^n - \frac{\beta\Delta t}{\alpha + \beta} f(u^n) \tag{3.25}$$

L'ordre de tels schémas est compris entre 1 et 2. Cela permet de passer de manière continue d'un schéma explicite ($\alpha = 0$) à un schéma implicite ($\beta = 0$).

Les schémas faisant intervenir $u^k$, $k < n$ sont rares car extrèmement coûteux en espace mémoire et en temps de calcul sans pour autant gagner beaucoup sur le pas de temps $\Delta t$.

## 3.4 Formulation générale d'un problème numérique

On pose $F$, la forme discrétisée de $f$ et $U$ le vecteur colonne contenant les $N_{ddl}$ valeurs décrivant le champ $u$ discrétisé. En reprenant le schéma général 3.25, la forme discrète de l'équation s'écrit :

$$U^{n+1} + \frac{\alpha \Delta t}{\alpha + \beta} F(U^{n+1}) = U^n - \frac{\beta \Delta t}{\alpha + \beta} F(U^n) \tag{3.26}$$

On peut réécrire 3.26 :

$$\tilde{F}(U^{n+1}) = C^n \tag{3.27}$$

en posant :

$$\tilde{F}(U^{n+1}) = U^{n+1} + \frac{\alpha \Delta t}{\alpha + \beta} F(U^{n+1}) \tag{3.28}$$

et

$$C^n = U^n - \frac{\beta \Delta t}{\alpha + \beta} F(U^n) \tag{3.29}$$

Pour résoudre numériquement 3.27, on utilise une méthode de Newton qui consiste à incrémenter la relation sur un pseudo-temps $k$ :

$$U^{k+1} = U^k - J_{\tilde{F}}^{-1} \left( \tilde{F}(U^k) - C^n \right) \tag{3.30}$$

Lorsque $\|U^{k+1} - U^k\| < \varepsilon$ (ou $\|\tilde{F}(U^{k+1}) - C^n\| < \varepsilon$) avec $\varepsilon$ le critère de convergence on pose $U^{n+1} = U^{k+1}$. Pour initialiser la méthode de Newton, on prend $U^{k=0} = U^n$. Si le pas de temps est suffisamment petit, la convergence de la méthode de Newton est immédiate (en un seul pas) car $U^{n+1}$ et $U^n$ sont suffisamment proches pour que l'approximation suivante puisse être directement utilisée :

$$\tilde{F}(U^{n+1}) \simeq \tilde{F}(U^n) + J_{\tilde{F}}(U^{n+1} - U^n) \tag{3.31}$$

Dans ce cas, il suffit de réécrire la relation 3.30 en remplaçant directement $k$ par $n$.

Le travail du numéricien est de calculer $J_{\tilde{F}}$ et son inverse $J_{\tilde{F}}^{-1}$. En condensant un maximum l'écriture, on doit donc résoudre, quelle que soit la méthode utilisée, un système matriciel de la forme :

$$AU^{k+1} = B^k \tag{3.32}$$

en posant :

$$A = J_{\tilde{F}} \tag{3.33}$$

et

$$B = J_{\tilde{F}} U^k - \tilde{F}(U^k) + C^n \tag{3.34}$$

En appelant $U$ l'inconnue à trouver et en laissant de côté les indices, on constate que la forme générique de tout problème numérique consiste à résoudre un problème linéaire qui s'écrit sous forme matricielle :

$$AU = B \tag{3.35}$$

Un travail essentiel reste à faire : trouver des méthodes numériques efficaces pour inverser une matrice $A$, surtout quand sa taille est importante. C'est la dernière partie du cours de méthodes numériques.

## 3.5 TD

### 3.5.1 Exercice : Résolution d'une edp instationnaire 1D

On veut résoudre une équation de convection-diffusion correspondant au transfert de chaleur entre deux points A et B par un fluide en écoulement, de masse volumique $\rho$, de chaleur spécifique massique $c$ et de conductivité thermique $\lambda$ :

$$\frac{\partial T}{\partial t} + \vec{v}.\vec{\nabla}T = a\nabla^2 T \tag{3.36}$$

avec $a = \lambda/\rho c$.

On se limite au cas 1D, c'est-à-dire que $\vec{v} = u\vec{e}_x$ et $T = T(t, x)$. La distance $L$ entre A et B est fixée à 1 (longueur de référence) et l'équation doit être comprise comme une équation sans dimension. Le fluide est incompressible donc l'équation de conservation impose que $u$ est une constante.

Par la suite, on prendra $u = 0$ (conduction pure, on développera tout de même les calculs en conservant le terme convectif) et $a = 0.3$ (on pourra essayer d'autres valeurs en fonction du temps disponible).

On veut résoudre l'équation 3.36 avec des conditions limites :

1. de Dirichlet, *i.e.* $T(0) = T_A sin(\omega t)$ et $T(1) = T_B$ ;
2. de Dirichlet $T(0) = T_A sin(\omega t)$ et une condition de Neumann $T'(L) = q_B$ ;
3. de Dirichlet $T(0) = T_A sin(\omega t)$ et une condition de Neumann $T'(0) = q_A$.

On prendra $\omega = 2\pi$ (si $t$ est en jours, par exemple, cela représente les oscillations journalières de température sur un mur), $T_A = 10$ et $T_B = 5$. Pour les conditions 2 et 3, on fixe $q_A = 0$ et $q_B = 0$.

1. Écrire la discrétisation temporelle de $\frac{\partial T}{\partial t}(t + \alpha\Delta t)$ qui ne fait intervenir que $T(t)$ et $T(t + \Delta t)$ avec $0 \leq \alpha \leq 1$. Donner l'ordre du schéma temporel en fonction de $\alpha$.

2. Proposer une interpolation de $T(t + \alpha\Delta t)$ en utilisant les valeurs $T(t)$ et $T(t + \Delta t)$. Donner l'ordre de l'interpolation en fonction de $\alpha$.

3. Écrire l'équation 3.36 à l'instant $t + \alpha\Delta t$ en utilisant le schéma de discrétisation temporelle pour la dérivée défini précédemment ainsi que l'interpolation utilisant les valeurs aux instants $t$ et $t + \Delta t$. Préciser l'ordre de discrétisation temporelle pour l'équation en fonction de la valeur de $\alpha$. Donner la valeur optimale de $\alpha$ d'un point de vue de l'ordre de discrétisation.

4. Pour la discrétisation spatiale, on utilise le schéma centré d'ordre 2 de l'exercice précédent (volumes finis ou différences finis) avec $m$ points entre 0 et $L$. Écrire le système à résoudre entre deux instants $t$ et $t + \Delta t$ pour les points intérieurs (sans les conditions limites). Pour simplifier l'écriture, on note $T(t_n, x_i) = T_i^{(n)}$ avec $t_{n+1} = t_n + \Delta t$.

5. Écrire le système d'équations sous forme matricielle entre deux instants $t$ et $t+\Delta t$ avec les conditions limites 1, 2 et 3. L'inconnue est le champ de température à l'instant $t + \Delta t$.

6. Donner une condition nécessaire de convergence du schéma temporel. Préciser cette condition pour $\alpha = 0$ (schéma explicite).

7. Sous Mathematica, écrire un programme pour résoudre le système avec les conditions limites 1 (2 et 3 si on a le temps) pour une durée de $t_f - t_0 = 1$. Faites les simulations avec $\alpha = 0$ (schéma explicite), $\alpha = 1$ (schéma implicite) et $\alpha = 1/2$ (schéma de Crank-Nicolson) avec le même pas spatial $m = 31$ et le même pas de temps $\Delta t = \Delta t_{critique}/2$ avec le pas de temps critique determiné à la question précédante pour un schéma explicite.

8. Essayer plusieurs pas de temps et comparer la méthode explicite et Crank-Nicolson (se baser sur l'évolution d'un point historique). Quelle est la limite supérieure pour le pas de temps, indépendamment du schéma ?
9. Conclure sur les avantages/inconvénients des différents schémas temporels.

Chapitre 4

# Algorithmes d'inversion de matrice

Nous avons vu que le problème continu aux dérivées partielles est transformé après discrétisation en un problème linéaire qui s'écrit sous forme matricielle :

$$AU = B \tag{4.1}$$

où $A$ est une matrice de taille $m \times m$ inversible. Nous allons décrire différentes méthodes numériques pour résoudre ce système 4.1. En général, $m = N_{ddl}$, ce qui correspond aux nombres de points de collocation (ou nœuds) du maillage, sauf quand les conditions limites sont intégrées avec une méthode Lagrangienne.

## 4.1 Préconditionneur

Il arrive parfois que le conditionnement $\kappa(A)$ de la matrice soit beaucoup trop élevé (valeurs propres mal réparties). En effet, le nombre de conditionnement $\kappa(A)$ est défini par :

$$\kappa(A) = \|A\|.\|A^{-1}\| \tag{4.2}$$

Si on prend comme norme :

$$\|A\| = \max(\{|\lambda_i|;\, i = 1, \cdots, m\}) \tag{4.3}$$

avec $\lambda_i$ les valeurs propres de $A$, on a :

$$\kappa(A) = \frac{\max(\{|\lambda_i|;\, i = 1, \cdots, m\})}{\min(\{|\lambda_i|;\, i = 1, \cdots, m\})} \tag{4.4}$$

Dans le cas où $\kappa(A)$ est grand, les méthodes d'inversion convergent (très) lentement. Pour améliorer l'efficacité des méthodes, on se ramène à un problème plus facile à inverser, c'est le rôle du préconditionneur. La technique du préconditionnement consiste à introduire une matrice $C \in \mathcal{M}_m(\mathbb{R})$ (inversible). Le préconditionneur $C$ de la matrice $A$ est une matrice telle que $C^{-1}A$ (ou $AC^{-1}$) ait un nombre de conditionnement $\kappa(C^{-1}A)$ plus petit que celui de $A$, $\kappa(A)$. On peut se ramèner soit au système conditionné à gauche :

$$C^{-1}AU = C^{-1}B \tag{4.5}$$

en résolvant :

$$\tilde{B} = C^{-1}B, \qquad (C^{-1}A)U = \tilde{B} \tag{4.6}$$

soit au système préconditionné à droite :

$$AC^{-1}CU = B \tag{4.7}$$

en résolvant

$$(AC^{-1})V = B, \qquad U = C^{-1}V \tag{4.8}$$

Ces systèmes sont bien équivalents au système d'origine tant que le préconditionneur $C$ est inversible. Il y a plusieurs choix possibles pour $C$. Comme l'opérateur $C^{-1}$ doit être appliqué à chaque itération du solveur linéaire, le préconditionneur doit être très simple à inverser afin de réduire le coût en temps de calcul pour avoir $C^{-1}$. Le préconditionneur le plus efficace serait alors :

$$C = I, \quad \text{puisque} \quad C^{-1} = I \tag{4.9}$$

Malheureusement, ce préconditionneur ne change rien au système d'origine ! L'autre cas limite consiste à prendre :

$$C = A, \quad \text{puisque} \quad C^{-1}A = AC^{-1} = I \tag{4.10}$$

ce qui permet d'atteindre le nombre de conditionnement optimal de 1, qui ne requiert alors que d'une seule itération pour converger. Cependant, dans ce cas :

$$C^{-1} = A^{-1} \tag{4.11}$$

Le précontionneur est donc aussi difficile à résoudre que le système d'origine. On choisit donc $C$ entre ces deux cas extrèmes, en essayant de minimiser le nombre d'itérations linéaires nécessaires tout en préservant autant que possible la simplicité de l'opérateur $C^{-1}$. Quelques exemples de préconditionnement sont décrits dans ce qui suit. On remarque que la matrice préconditionnée $C^{-1}A$ (resp.$AC^{-1}$) n'est jamais explicitement calculée. En utilisant une méthode itérative, seule l'application de $C^{-1}$ à un vecteur doit être calculée. On remarque aussi que pour une matrice $A$ symétrique, l'effet souhaité par l'application d'un préconditionneur est de rendre la forme quadratique représentée par l'opérateur $C^{-1}A$ presque sphérique.

Deux stratégies courantes consistent à séparer $A$ en trois éléments :

$$A = L + D + \tilde{U} \tag{4.12}$$

avec $L$ la partie triangulaire inférieure de $A$, $D$ sa diagonale et $\tilde{U}$ sa partie triangulaire supérieure.

### 4.1.1 Le préconditionneur de Jacobi

Le préconditionneur de Jacobi est l'un des plus simples. Il consiste à prendre pour préconditionneur la matrice diagonale de $A$, $D$ :

$$C = D \tag{4.13}$$

L'avantage principal d'un tel préconditionneur est sans aucun doute la facilité de son implémentation et le peu d'espace mémoire qu'il occupe. Cependant, on peut tout de même trouver des préconditionneurs apportant une meilleure amélioration de la résolution du système linéaire. C'est le cas du préconditionnement SOR et SSOR.

### 4.1.2 Préconditionneur SOR et SSOR

Le préconditionneur SOR (Successive Over Relaxation) prend pour matrice $C$ :

$$C = \left(\frac{D}{\omega} + L\right) \frac{\omega}{2-\omega} D^{-1} \left(\frac{D}{\omega} + \tilde{U}\right) \tag{4.14}$$

avec $\omega$ un paramètre de relaxation choisi entre 0 et 2.

La version de SOR pour les matrices symétriques $A$, telles que $\tilde{U} = L^\top$ est le précontionneur SSOR (Symmetric Successive Over Relaxation) pour lequel :

$$C = \left(\frac{D}{\omega} + L\right) \frac{\omega}{2-\omega} D^{-1} \left(\frac{D}{\omega} + L^\top\right). \tag{4.15}$$

## 4.2 Méthodes matricielles

Les méthodes d'inversion se décomposent en deux grandes familles : les méthodes de décomposition matricielle et les méthodes itératives. Les méthodes matricielles consistent à décomposer la matrice $A$ de telle façon à ce que l'on puisse inverser le système avec une méthode de pivot de Gauss.

Les méthodes d'inversion matricielles sont en générales couteuses en terme de consommation mémoire car il faut générer et stocker toutes les matrices. Cependant, elles sont précises et rigoureuses.

### 4.2.1 Méthode du pivot de Gauss

On suppose que la matrice $A$ du système linéaire est échelonnée, c'est-à-dire triangulaire inférieure ou supérieure (sinon, on la décompose suivant une des méthodes présentée dans la suite), c'est-à-dire que l'on peut écrire par exemple $A = L$ (matrice triangulaire inférieure, ici $L$ intègre les éléments diagonaux). On doit alors résoudre le système triangulaire :

$$LU = B \tag{4.16}$$

On utilise alors un algorithme de descente pour le système 4.16 :

$$\begin{cases} u_1 = \frac{b_1}{l_{11}}; \\ u_i = \frac{1}{l_{ii}}(b_i - \sum_{j=1}^{i-1} l_{ij}u_j), \quad 2 \leq i \leq m. \end{cases} \tag{4.17}$$

avec $l_{ij}$ les éléments de $L$ (sachant que pour $i < j$, $l_{ij} = 0$).

Si $A = \tilde{U}$ (triangulaire supérieure), on utilise un algorithme de remontée qui consiste à partir du dernier élément.

### 4.2.2 Méthode de Cholesky

La factorisation de Cholesky, consiste, pour une matrice symétrique définie positive $A$, à déterminer une matrice triangulaire inférieure $L$ tel que $A = LL^T$. La matrice $L$ est en quelque sorte une « racine carrée » de $A$. Cette décomposition permet notamment de calculer la matrice inverse $A^{-1}$ et de calculer le déterminant de $A$ (égal au carré du produit des éléments diagonaux de $L$).

**Théorème 6.** *Factorisation de Cholesky d'une matrice :*

*Si $A$ est une matrice symétrique définie positive, il existe au moins une matrice réelle triangulaire inférieure $L$ telle que $A = LL^T$. On peut également imposer que les éléments diagonaux de la matrice $L$ soient tous positifs, et la factorisation correspondante est alors unique.*

#### Algorithme

On cherche la matrice :

$$L = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{m1} & l_{m2} & \cdots & l_{mm} \end{pmatrix} \tag{4.18}$$

De l'égalité $A = LL^T$ on déduit :

$$a_{ij} = \left(LL^T\right)_{ij} = \sum_{k=1}^{m} l_{ik} l_{jk} = \sum_{k=1}^{\min\{i,j\}} l_{ik} l_{jk},\ 1 \leq i, j \leq m \tag{4.19}$$

puisque $l_{ij} = 0$ si $1 \leq i < j \leq m$.

La matrice $A$ étant symétrique, il suffit que les relations ci-dessus soient vérifiées pour $i \leq j$, c'est-à-dire que les éléments $l_{ij}$ de la matrice $L$ doivent satisfaire :

$$a_{ij} = \sum_{k=1}^{i} l_{ik} l_{jk},\ 1 \leq i, j \leq m \tag{4.20}$$

Pour $j = 1$, on détermine la première colonne de $L$ :

$$\begin{cases} (i=1) & a_{11} = l_{11} l_{11} \text{ d'où } l_{11} = \sqrt{a_{11}} \\ (i=2) & a_{12} = l_{11} l_{21} \text{ d'où } l_{21} = \frac{a_{12}}{l_{11}} \\ & \vdots \\ (i=m) & a_{1m} = l_{11} l_{m1} \text{ d'où } l_{m1} = \frac{a_{1m}}{l_{11}} \end{cases} \tag{4.21}$$

On détermine la j-ème colonne de $L$, après avoir calculé les $(j-1)$ premières colonnes :

$$\begin{cases} (i=j) & a_{ii} = l_{i1} l_{i1} + \ldots + l_{ii} l_{ii} \text{ d'où } l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2} \\ (i=j+1) & a_{i,i+1} = l_{i1} l_{i+1,1} + \ldots + l_{ii} l_{i+1,i} \text{ d'où } l_{i+1,i} = \frac{1}{l_{ii}} \left(a_{i,i+1} - \sum_{k=1}^{i-1} l_{ik} l_{i+1,k}\right) \\ & \vdots \\ (i=m) & a_{i,m} = l_{i1} l_{m1} + \ldots + l_{ii} l_{mi} \text{ d'où } l_{mi} = \frac{1}{l_{ii}} \left(a_{im} - \sum_{k=1}^{i-1} l_{ik} l_{mk}\right) \end{cases} \tag{4.22}$$

**Résolution de système**

Pour la résolution de système linéaire de la forme $AU = B$, le système devient :

$$LL^\top U = B \Leftrightarrow \begin{cases} LV = B & (1), \\ L^\top U = V & (2). \end{cases} \tag{4.23}$$

On résout le système (1) pour trouver le vecteur $V$, puis le système (2) pour trouver le vecteur $U$. Comme les matrices sont triangulaires, la résolution des deux systèmes utilise la méthode du pivot de Gauss.

**Calcul de déterminant**

La méthode de Cholesky permet aussi de calculer le déterminant de $A$, qui est égal au carré du produit des éléments diagonaux de la matrice $L$, puisque :

$$det(A) = det(L) \times det(L^\top) = det(L)^2 \tag{4.24}$$

### 4.2.3 Méthode de décomposition LU

Soit $A$ une matrice de taille $m \times m$. La factorisation LU, consiste, pour une matrice $A$, à déterminer une matrice triangulaire inférieure $L$ à diagonale unité et une matrice triangulaire supérieure $\tilde{U}$ telle que $A = L\tilde{U}$ avec :

$$L = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{m1} & l_{m2} & \cdots & 1 \end{pmatrix} \tag{4.25}$$

et

$$\tilde{U} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ & u_{22} & \cdots & u_{2m} \\ & & \ddots & u_{m-1,m} \\ & & & u_{mm} \end{pmatrix} \tag{4.26}$$

Si $A$ est une matrice symétrique définie positive, cette méthode est équivalente à la méthode de Cholesky. La méthode LU est donc une généralisation de cette dernière utilisable avec des matrices non symétriques. Cependant, on doit éviter de l'utiliser à la place de la méthode de Cholesky car l'algorithme est environ deux fois plus lent (génération de $L$ et de $\tilde{U}$ au lieu de seulement $L$).

### Résolution de système

Pour la résolution de système linéaire de la forme $AU = B$, le système devient :

$$L\tilde{U}U = B \Leftrightarrow \begin{cases} LY = B & (1), \\ \tilde{U}U = Y & (2). \end{cases} \tag{4.27}$$

On résout le système (1) pour trouver le vecteur $Y$, puis le système (2) pour trouver le vecteur $U$. Comme les matrices sont triangulaires, la résolution des deux systèmes utilise la méthode du pivot de Gauss (analogue à la méthode de Cholesky).

**Théorème 7.** *Conditions d'application de la décomposition LU.*

- *Si $A$ admet une décomposition LU, alors celle-ci est unique.*
- *$A$ admet une décomposition LU si, et seulement si, ses mineurs principaux sont non nuls (le mineur principal d'ordre $k$ de $A$ désigne le déterminant de la matrice obtenue à partir de $A$ en extrayant les $k$ premières lignes et colonnes).*
- *Si $A$ est simplement supposée inversible, alors $A$ peut s'écrire $A = P\tilde{LU}$, où $P$ est une matrice de permutation.*

### Algorithme général de la décomposition LU

On suppose que $A$ admet une décomposition LU, on a alors l'algorithme de décomposition LU suivant :

$$k = 1, \ldots, m-1 \begin{cases} l_{ik} = 0 & | \; i = 1, \ldots, k-1 \\ l_{kk} = 1 & \\ l_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}} & | \; i = k+1, \ldots, m \\ a_{ij}^{(k+1)} = a_{ij}^{(k)} & | \; i = 1, \ldots, k \quad j = 1, \ldots, m \\ a_{ij}^{(k+1)} = 0 & | \; i = k+1, \ldots, m \quad j = 1, \ldots, k \\ a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)} & | \; i = k+1, \ldots, m \quad j = k+1, \ldots, m \end{cases} \tag{4.28}$$

$$\begin{array}{lll} l_{im} = 0 & i = 1, \ldots, m-1 & l_{mm} = 1 \\ \tilde{U} = (a_{ij}^{(m)})_{1 \leq i,j \leq m} & L = (l_{ij})_{1 \leq i,j \leq m} & \end{array} \tag{4.29}$$

**Calcul de déterminant**

La décomposition LU permet aussi de calculer le déterminant de $A$, qui est égal au produit des éléments diagonaux de la matrice $\tilde{U}$ si $A$ admet une décomposition LU

$$det(A) = det(L) \times det(\tilde{U}) = 1 \times det(\tilde{U}) = det(\tilde{U}) \tag{4.30}$$

## 4.3 Méthodes itératives

Les méthodes itératives permettent de ne pas générer et stocker les matrices à inverser. Elles demandent donc une quantité de mémoire beaucoup plus faible que les méthodes matricielles. L'inversion rigoureuse du système est obtenue en général après un nombre d'itérations égal à la dimension du système. Cependant, cela représenterait des temps de calculs trop importants. On se contente donc de vérifier un critère de convergence permettant ainsi une réduction très importante du nombre d'itérations. Les méthodes itératives fournissent alors une solution approchée du système dont la précision est contrôlée par ce critère de convergence.

### 4.3.1 Principe de construction

Une méthode itérative de résolution de système linéaire de la forme $AU = B$ consiste à construire une suite $U_k$ qui converge vers un point fixe $U_\star$, solution du système d'équations linéaires. On cherche à construire l'algorithme pour $U_0$ donné, la suite $U_{k+1} = F(U_k)$ avec $k \in \mathbb{N}$.

Tout d'abord, on décompose $A$ :

$$A = M - N \tag{4.31}$$

où $M$ est une matrice inversible. C'est le choix de $M$ qui définit la méthode. On a alors :

$$AU = B \quad \Leftrightarrow \quad MU = NU + B \quad \Leftrightarrow \quad U = M^{-1}NU + M^{-1}B \tag{4.32}$$

On pose alors

$$F(U) = M^{-1}NU + M^{-1}B \tag{4.33}$$

$F$ est une fonction affine. On a d'après 4.32 :

$$\left\{ \begin{array}{c} U_0 \text{ donné} \\ U_{k+1} = M^{-1}NU_k + M^{-1}B \quad \text{pour } k \geq 0. \end{array} \right. , \tag{4.34}$$

Il faut un critère d'arrêt, *i.e.* un critère de convergence. De plus, il faut s'assurer que la suite ainsi définie converge bien vers $U_\star$. On définit donc $E_k$ le vecteur erreur :

$$E_{k+1} = U_{k+1} - U_k = M^{-1}N(U_k - U_{k-1}) = M^{-1}NE_k \tag{4.35}$$

On pose $H = M^{-1}N$, ce qui donne :

$$E_{k+1} = HE_k = H^{k+1}E_0. \tag{4.36}$$

L'algorithme converge si :

$$\lim_{k\to\infty} \|E_k\| = 0 \Leftrightarrow \lim_{k\to\infty} \|H^k\| = 0 \tag{4.37}$$

**Théorème 8.** *Une condition nécessaire et suffisante pour que* $\lim_{k\to\infty} \|H^k\| = 0$ *est que le rayon spectral* $\rho$ *de* $H$ *vérifie* $\rho(H) < 1$. *On rappelle que* $\rho(H) = \max_{i=1,\dots,m} |\lambda_i|$ *où* $\{\lambda_1, \dots, \lambda_m\}$ *sont les valeurs propres de* $H$.

### 4.3.2 Méthode de Jacobi

On décompose la matrice $A$ de la façon suivante :

$$A = D - E - F \tag{4.38}$$

avec

* $D$ la diagonale.
* $-E$ la partie en dessous de la diagonale.
* $-F$ la partie au dessus.

Dans la méthode de Jacobi, on choisit $M = D$ et $N = E + F$ (dans la méthode de Gauss-Seidel, $M = D - E$ et $N = F$).

**Théorème 9.** *Si* $A$ *est à diagonale strictement dominante, i.e. :*

$$|a_{ii}| > \sum_{i\neq j} |a_{ij}|, \forall i = 1, \dots, n \tag{4.39}$$

*alors pour tout* $U_0$ *la méthode de Jacobi converge vers la solution* $U$ *du système* $AU = B$.

On obtient d'après 4.32 :

$$U_{k+1} = D^{-1}(E + F)U_k + D^{-1}B \tag{4.40}$$

avec pour la ligne $i$ de $D^{-1}(E + F)$ :

$$-\left(\frac{a_{i,1}}{a_{i,i}}, \cdots, \frac{a_{i,i-1}}{a_{i,i}}, 0, \frac{a_{i,i+1}}{a_{i,i}}, \cdots, \frac{a_{i,m}}{a_{i,i}}\right) \tag{4.41}$$

On a alors :

$$u_{i,k+1} = -\frac{1}{a_{ii}} \sum_{j=1,j\neq i}^{m} a_{ij}u_{j,k} + \frac{b_i}{a_{ii}} \tag{4.42}$$

En général, on ne sait pas calculer l'erreur $E_k$. Aussi, pour le test d'arrêt, on définit $R_k = b - AU_k$ le vecteur résidu (voir aussi la section sur le gradient conjugué 4.3.4). On peut alors écrire $u_{i,k+1} = \frac{r_{i,k}}{a_{ii}} + u_{i,k}$ avec $r_{i,k}$ que l'on calcule de la manière suivante :

$$r_{i,k+1} = -\sum_{j=1,j\neq i}^{m} a_{ij}\frac{r_{i,k}}{a_{jj}} \tag{4.43}$$

Pour le test d'arrêt, on utilise le vecteur résidu $R_k$, ce qui donne, pour une précision donnée $\varepsilon$ :

$$\frac{\|R_k\|}{\|B\|} = \frac{\|B - AU_k\|}{\|B\|} < \varepsilon \tag{4.44}$$

### 4.3.3 Méthode de Gauss-Seidel

La méthode de Gauss-Seidel est une méthode itérative de résolution de système linéaire de la forme $AU = B$. On décompose la matrice $A$ de la façon suivante :

$$A = D - E - F \tag{4.45}$$

avec

— $D$ la diagonale.

— $-E$ la partie en dessous de la diagonale.

— $-F$ la partie au dessus.

Dans la méthode de Gauss-Seidel, on choisit $M = D - E$ et $N = F$ (dans la méthode de Jacobi, $M = D$ et $N = E + F$).

**Théorème 10.** *Si $A$ est à diagonale strictement dominante, alors pour tout $U_0$ la méthode de Gauss-Seidel converge vers la solution $U$ du système $AU = B$.*

On a d'après ce qui précède :

$$U_{k+1} = (D - E)^{-1} F U_k + (D - E)^{-1} B \tag{4.46}$$

On a alors :

$$u_{i,k+1} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} u_{j,k+1} - \sum_{j=i+1}^{m} a_{ij} u_{j,k}}{a_{ii}} \tag{4.47}$$

Pour le test d'arrêt, on utilise le vecteur résidu, comme dans la méthode de Jacobi (voir 4.44).

### 4.3.4 Méthode du gradient conjugué

Considérons le système 4.1 où $A$ est une matrice de taille $m \times m$ symétrique définie positive ($A^\top = A$ et $U^\top AU > 0$, pour tout vecteur $U \in \mathbb{R}^m$ non nul). Soit $U_\star$ la solution exacte de ce système.

#### Directions conjuguées

Comme la matrice $A$ est symétrique définie positive, on peut définir le produit scalaire suivant sur $\mathbb{R}^m$ : $\langle U, V \rangle_A = U^\top AV$. Deux éléments $U, V \in \mathbb{R}^m$ sont dit $A$-conjugués si :

$$U^\top AV = 0 \tag{4.48}$$

La méthode du gradient conjugué consiste à construire une suite $\{V_k\}_{k \in \mathbb{N}^*}$ de $m$ vecteurs $A$-conjugués. Dès lors, la suite $\{V_1, V_2, \ldots, V_m\}$ forme une base de $\mathbb{R}^m$. La solution exacte $U_\star$ peut donc se décomposer comme suit :

$$U_\star = \sum_{k=1}^{m} \alpha_k V_k \tag{4.49}$$

où

$$\alpha_k = \frac{V_k^\top B}{V_k^\top A V_k}. \tag{4.50}$$

Il faut donc construire cette base $\{V_1, V_2, \ldots, V_m\}$.

### Construction des directions conjuguées

La solution exacte $U_\star$ peut être également vue comme l'unique minimisant de la fonctionnelle (déjà vu pour la méthode Lagrangienne 2.2.6) :

$$J(U) = \frac{1}{2} U^\top A U - B^\top U, \quad U \in \mathbb{R}^m \tag{4.51}$$

On a donc clairement $\nabla J(U) = AU - B, \quad U \in \mathbb{R}^m$ d'où $\nabla J(U_\star) = 0$. On définit le résidu du système d'équation comme suit :

$$R_k = B - AU_k = -\nabla J(U_k). \tag{4.52}$$

$R_k$ représente donc la direction du gradient de la fonctionnelle $J$ en $U_k$, l'approximation à l'itération $k$ de $U_\star$ (à un signe près). La nouvelle direction de descente $V_{k+1}$ suit donc celle du résidu modulo, sa $A$-conjugaison avec $V_k$, on a alors :

$$V_{k+1} = R_k - \frac{V_k^\top A R_k}{V_k^\top A V_k} V_k \tag{4.53}$$

C'est le choix du coefficient $\frac{V_k^\top A R_k}{V_k^\top A V_k}$ qui assure la $A$-conjugaison des directions $V_k$. Pour vous en assurer calculer $\langle V_k, V_{k+1} \rangle_A$, cette quantité est nulle.

Pour calculer $U_\star$ avec une méthode directe, il faut choisir une base $A$-conjuguée, ici, par exemple, on choisit un vecteur $V_1$ arbitraire. On définit d'après les relations 4.49 et 4.50 :

$$U_k = \sum_{i=1}^{k} \alpha_i V_i \tag{4.54}$$

On a alors $U_\star = U_m$. Malheureusement, cette méthode directe est très coûteuse : il faut évaluer les expressions $m$ fois, or $m$ peut être très grand. Cependant, si la base $\{V_i\}$ est bien choisie, l'approximation $U_k$ converge rapidement vers $U_\star$ et les termes supplémentaires sont donc très petits. C'est l'objet de la méthode itérative que l'on utilise en général.

### Algorithme itératif du gradient conjugué

On choisit un champ initial $U_0$ quelconque et on prend la première direction de descente suivant le gradient, c'est-à-dire $V_0 = R_0$. On suppose que $U_k$ et $V_k$ sont connus. $V_k$ est la direction de descente, c'est-à-dire que partant de $U_k$, on cherche à s'approcher de $U_\star$ suivant la direction $V_k$ :

$$U_{k+1} = U_k + a_k V_k \tag{4.55}$$

Reste à définir $a_k$ et $V_k$. Pour cela, on impose que la fonctionnelle $J(U_{k+1}(a_k))$ soit minimale, c'est-à-dire que :

$$\frac{dJ(U_{k+1}(a_k)}{da_k} = 0 \tag{4.56}$$

Cela revient à écrire :

$$-R_{k+1}^\top V_k = 0 \tag{4.57}$$

Par définition du reste $R_{k+1}$, on trouve la relation de récurrence :

$$R_{k+1} = R_k - a_k A V_k \tag{4.58}$$

d'où il vient :

$$a_k = \frac{R_k^\top V_k}{V_k^\top A V_k} \tag{4.59}$$

Pour construire notre base $A$-conjuguée, on prend la forme générique :

$$V_{k+1} = R_k - \sum_{i \leq k} \frac{V_i^\top A R_k}{V_i^\top A V_i} V_i \tag{4.60}$$

On montre que l'algorithme du gradient conjugué qui en résulte est donné par :
Pour $k = 1, 2, \ldots,$
$a_k = \frac{R_k^\top R_k}{V_k^\top A V_k}$
$U_{k+1} = U_k + a_k V_k$
$R_{k+1} = R_k - a_k A V_k$
Si $\|R_{k+1}\| < \varepsilon$ on sort de la boucle FinSi
$v_k = \frac{R_{k+1}^\top R_{k+1}}{R_k^\top R_k}$
$V_{k+1} = R_{k+1} + v_k V_k$
FinPour

Cet algorithme converge au bout de $m$ itérations dans le pire des cas si $A$ est symétrique définie positive. Si $U_0$ est proche de $U_\star$, la convergence peut être très rapide.

### Méthode du gradient conjugué préconditionné

Soit $U_0 \in \mathbb{R}^m$ un vecteur initial quelconque, l'algorithme du gradient conjugué préconditionné est le suivant :
$R_0 = B - AU_0$
$Q_0 = C^{-1} R_0$
$V_0 = W_0$
Pour $k = 1, 2, \ldots,$
$a_k = \frac{Q_k^\top R_k}{V_k^\top A V_k}$
$U_{k+1} = U_k + a_k V_k$
$R_{k+1} = R_k - a_k A V_k$
Si $\|R_{k+1}\| < \varepsilon$ on sort de la boucle FinSi
$Q_{k+1} = C^{-1} R_{k+1}$
$v_{k+1} = \frac{Q_{k+1}^\top R_{k+1}}{Q_k^\top R_k}$
$V_{k+1} = Q_{k+1} + v_{k+1} V_k$
FinPour

Plusieures méthodes sont envisageables pour construire le précondisionneur $C$ (voir section 4.1).

### 4.3.5 Méthode GMRES

La méthode généralisée des résidus minimaux (generalized minimal residual method, abrégée classiquement par GMRES) est une méthode itérative pour résoudre un système linéaire d'équation $AU = B$. La méthode approche la solution par un vecteur d'un espace de Krylov avec un résidu minimal. Les itérations d'Arnoldi sont utilisées pour trouver ce vecteur.

La méthode GMRES a été développée par Yousef Saad et Martin H. Schultz en 1986 (Saad & Schultz, 1986). On peut la voir comme une généralisation de la méthode des gradients conjugués applicable à des matrices $A$ non symétriques.

#### Algorithme

La matrice $A$ est supposée inversible de dimension $m$. De plus, on suppose que $B$ a été normalisé, *i.e.*, $\|B\| = 1$. Le sous-espace de Krylov de dimension $k$ pour ce problème est :

$$K_k = \text{Vect}\{B, AB, A^2B, \ldots, A^{k-1}B\}. \tag{4.61}$$

La méthode GMRES approche la solution exacte $U_\star$ de $AU = B$ par le vecteur $U_k \in K_k$ qui minimise le résidu $R_k = AU_k - B$. Les vecteurs $B$, $AB$, ..., $A^{k-1}B$ sont presque linéairement dépendants. On utilise donc l'itération d'Arnoldi (décrite dans la suite) pour générer une base orthonormale de vecteurs $\{Q_1, Q_2, \ldots, Q_k\}$ qui soit une base de $K_k$. Ainsi, $U_k$ peut s'écrire :

$$U_k = Q(k)Y_k \tag{4.62}$$

avec $Y_k \in \mathbb{R}^k$ et $Q(k)$ une matrice $m \times k$ constituée par les vecteurs $Q_i$. L'algorithme produit aussi la matrice supérieure de Hessenberg $\tilde{H}(k)$ de dimension $(k+1) \times k$ avec :

$$AQ(k) = Q(k+1)\tilde{H}(k) \tag{4.63}$$

Comme $Q(k)$ est orthogonal, nous avons :

$$\|AU_k - B\| = \|\tilde{H}(k)Y_k - \beta\tilde{E}_1\| \tag{4.64}$$

avec

$$\tilde{E}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{4.65}$$

qui est le premier vecteur de la base canonique de $\mathbb{R}^{k+1}$ et

$$\beta = \|B - AU_0\| \tag{4.66}$$

$U_0$ est le vecteur initial (souvent nul, alors $\beta = 1$). Ainsi, $U_k$ peut être calculé par minimisation de la norme du résidu :

$$\|R_k\| = \|\tilde{H}(k)Y_k - \beta\tilde{E}_1\| \tag{4.67}$$

L'équation 4.67 est un problème de moindre carré de dimension $k$. À chaque pas d'itération, les opérations sont :

1. Effectuer un pas de la méthode d'Arnoldi.
2. Trouver $Y_k$ qui minimise $\|R_k\|$.
3. Calculer $U_k = Q(k)Y_k$.
4. Recommencer si le résidu est supérieur au critère de convergence, *i.e.* si $\|R_k\| > \varepsilon$.

À chaque itération, le produit matriciel $AQ(k)$ doit être calculé. Le coût est d'environ $2m^2$ opérations à virgule flottante (floating-point) pour une matrice pleine de dimension $m$, mais ce coût diminiue en $\bigcirc(m)$ pour les matrices creuses. En plus du produit matriciel, $\bigcirc(km)$ opérations à virgule flottante doivent être effectuées à la $k$ème itération.

### Convergence

La $k$ème itération minimise le résidu dans le sous-espace de Krylov $K_k$. Comme tout sous-espace $K_k \subset K_{k+1}$, le résidu décroit de façon monotone. Après $m$ itérations, $m$ étant la dimension de la matrice $A$, l'espace de Krylov $K_m = \mathbb{R}^m$ et donc la méthode GMRES atteint la solution exacte $U_\star$. Cependant, l'idée, c'est d'atteindre une bonne approximation $U_n$ de $U_\star$ après un nombre d'itérations $n$ petit devant $m$.

En général, ce n'est malheureusement pas toujours le cas. En effet, le théorème de Greenbaum, Pták and Strakoš établit que pour toute suite monotone décroissante $x_1, ..., x_{m-1}, x_m = 0$, on peut trouver une matrice $A$ tel que $\|R_k\| = x_k$, $\forall k$ avec $R_k$ le résidu précèdemment défini. En particulier, il est possible de trouver une matrice pour laquelle le résidu reste constant pour $m-1$ itérations et passe à zéro seulement à la dernière itération.

En pratique, cependant, la méthode GMRES fonctionne bien. Cela peut être démontré dans des situations particulières. Si $A$ est définie positive alors :

$$\|R_k\| \leq \left(1 - \frac{\lambda_{\min}(A^\top + A)}{2\lambda_{\max}(A^\top + A)}\right)^{k/2} \|R_0\| \tag{4.68}$$

où $\lambda_{min}(M)$ et $\lambda_{max}(M)$ sont respectivement la plus petite et la plus grande valeur propre de $M$.

Si $A$ est symétrique définie positive, alors on a même :

$$\|R_k\| \leq \left(\frac{\kappa_2^2(A) - 1}{\kappa_2^2(A)}\right)^{k/2} \|R_0\| \tag{4.69}$$

où $\kappa_2(A)$ est le nombre de conditionnement de $A$ calculé avec la norme Euclidienne :

$$\kappa_2(A) = \|A\|_2 \cdot \|A^{-1}\|_2 \tag{4.70}$$

Dans le cas général, lorsque $A$ n'est pas définie positive, on a :

$$\|R_k\| \leq \inf_{p \in P_k} \|p(A)\| \leq \kappa_2(V) \inf_{p \in P_k} \max_{\lambda \in \sigma(A)} |p(\lambda)| \tag{4.71}$$

avec $P_k$ qui désigne l'ensemble des polynômes de degré inférieur ou égal à $k$ tel que $p(0) = 1$. $V$ est la matrice des vecteurs propres apparaissant dans la décomposition spectrale de $A$ et $\sigma(A)$ est le spectre

de $A$. Autrement dit, la convergence est rapide quand les valeurs propres de $A$ sont éloignées de zéro et que $A$ n'est pas trop éloignée d'une matrice normale (i.e. $\|A\| \sim 1$). On voit bien tout l'intérêt du préconditionnement.

Ces inégalités concernent le résidu au lieu de l'erreur, c'est-à-dire de la distance entre $U_k$ et $U_\star$.

### Les itérations d'Arnoldi

Les itérations d'Arnoldi utilisent le processus de Gram-Schmidt stabilisé pour générer une séquence de vecteurs orthogonaux $\{Q_1, Q_2, Q_3, ..., Q_k\}$ appelés vecteurs d'Arnoldi, tel que cet ensemble de vecteurs génère le sous-espace de Krylov $K_k$. L'algorithme s'écrit de façon explicite :

- Prendre un vecteur de norme arbitraire $Q_1$. Évidemment, on prend $Q_1 = B$.
- Répéter pour $k = 2, 3, ...$
  - $Q_k = AQ_{k-1}$
  - pour $j$ de 1 à $k-1$
    - $h_{j,k-1} = Q_j^\top Q_k$
    - $Q_k = Q_k - h_{j,k-1}Q_j$
  - $h_{k,k-1} = \|Q_k\|$
  - $Q_k = \frac{Q_k}{h_{k,k-1}}$

La boucle sur $j$ projète $AQ_{k-1}$ sur $Q_1, \ldots, Q_{k-1}$ ce qui assure l'orthogonalité des vecteurs générés. L'algorithme s'arrête quand $Q_k$ est un vecteur nul. Cela arrive quand le polynôme minimal de $A$ est de degré $k$. Dans la plupart des applications de l'itération d'Arnoldi, notamment la méthode GMRES, l'algorithme a alors convergé. Chaque étape de la boucle sur $k$ coûte le prix d'un produit matriciel et approximativement $4km$ opérations.

### Résolution du problème aux moindres carrés

Une étape de la méthode GMRES consiste à trouver le vecteur $Y_k$ qui minimise :

$$\|\tilde{H}(k)Y_k - \beta\tilde{E}_1\| \tag{4.72}$$

Cela peut être effectué en faisant une décomposition $QR$, *i.e.* trouver une matrice $(k+1) \times (k+1)$ orthogonale $\Omega(k)$ et une matrice $(k+1) \times (k)$ triangulaire supèrieure $\tilde{R}(k)$ telles que :

$$\Omega(k)\tilde{H}(k) = \tilde{R}(k) \tag{4.73}$$

La matrice triangulaire $\tilde{R}(k)$ a une ligne de plus qu'elle n'a de colonnes. Sa dernière ligne est donc nulle. Ainsi, on peut décomposer $\tilde{R}(k)$ suivant :

$$\tilde{R}(k) = \begin{pmatrix} R(k) \\ 0 \end{pmatrix} \tag{4.74}$$

avec $R(k)$ une matrice triangulaire supérieure $k \times k$. La décomposition $QR$ peut être facilement mise à jour d'une itération à la suivante, parce que la matrice de Hessenberg diffère seulement d'une ligne de zéros et d'une colonne :

$$\tilde{H}(k+1) = \begin{pmatrix} \tilde{H}(k) & H_k \\ 0 & h_{k+1,k} \end{pmatrix} \tag{4.75}$$

avec

$$H_k = \begin{pmatrix} h_{1,k} \\ \vdots \\ h_{k,k} \end{pmatrix} \tag{4.76}$$

Cela implique que la multiplication de la matrice de Hessenberg $\tilde{H}(k+1)$ avec $\Omega(k)$, à qui on ajoute des zéros et 1 sur le dernier élément de la diagonale produit une matrice presque triangulaire :

$$\begin{pmatrix} \Omega(k) & 0 \\ 0 & 1 \end{pmatrix} \tilde{H}(k+1) = \begin{pmatrix} R(k) & R_k \\ 0 & \rho \\ 0 & \sigma \end{pmatrix} \tag{4.77}$$

Cette matrice serait triangulaire si $\sigma$ était nul. Pour y remédier, on se donne une rotation :

$$G(k) = \begin{pmatrix} I(k-1) & 0 & 0 \\ 0 & c_k & s_k \\ 0 & -s_k & c_k \end{pmatrix} \tag{4.78}$$

avec

$$c_k = \frac{\rho}{\sqrt{\rho^2+\sigma^2}} \quad \text{et} \quad s_k = \frac{\sigma}{\sqrt{\rho^2+\sigma^2}}. \tag{4.79}$$

Avec cette rotation, on a :

$$\Omega(k+1) = G(k) \begin{pmatrix} \Omega(k) & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.80}$$

En effet,

$$\Omega(k+1)\tilde{H}(k+1) = \begin{pmatrix} R(k) & r_k \\ 0 & r_{kk} \\ 0 & 0 \end{pmatrix} \quad \text{avec} \quad r_{nn} = \sqrt{\rho^2+\sigma^2} \tag{4.81}$$

est une matrice triangulaire.

Soit la décomposition QR, le problème de minimisation est facilement résolu en constatant que :

$$\|\tilde{H}(k)Y_k - \beta\tilde{E}_1\| = \|\Omega(k)(\tilde{H}(k)Y_k - \beta\tilde{E}_1)\| = \|\tilde{R}(k)Y_k - \beta\Omega(k)\tilde{E}_1\|. \tag{4.82}$$

En notant le vecteur :

$$\tilde{G}_k = \begin{pmatrix} G_k \\ g_k \end{pmatrix} = \beta\Omega(k)\tilde{E}_1 \tag{4.83}$$

avec $G_k \in \mathbb{R}^k$ et $g_k \in \mathbb{R}$, on obtient :

$$\|\tilde{H}(k)Y_k - \beta\tilde{E}_1\| = \|\tilde{R}(k)Y_k - \beta\Omega(k)\beta\tilde{E}_1\| = \left\| \begin{pmatrix} R_k \\ 0 \end{pmatrix} Y - \begin{pmatrix} G_k \\ g_k \end{pmatrix} \right\|. \tag{4.84}$$

Le vecteur $Y_k$ qui minimise cette expression est donné par :

$$Y_k = R(k)^{-1}G_k. \tag{4.85}$$

Le vecteur $G_k$ est donné par la relation 4.83.

### Extension de la méthode GMRES

De la même façon que pour les autres méthodes, il est souvent utile de préconditionner la matrice $A$. Le coût des itérations croit comme $n^2$, avec $n$ le nombre d'itérations. Cependant, la méthode est parfois redémarrée après un nombre $k$ d'itérations avec $U_k$ comme vecteur initial (à la place de $U_0$). La méthode résultante s'appelle GMRES(k).

## Bibliographie

Saad, Y. & Schultz, M.H. 1986 GMRES : A generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comput.* **7**, 856–869.

## 4.4 TD

### 4.4.1 Exercice : Inversion d'un système linéaire

On veut résoudre une équation de convection-diffusion correspondant au transfert de chaleur entre deux points A et B par un fluide en écoulement, de masse volumique $\rho$, de chaleur spécifique massique $c$ et de conductivité thermique $\lambda$ :

$$\vec{v}.\vec{\nabla}T = a\nabla^2 T \tag{4.86}$$

avec $a = \lambda/\rho c$.

On se limite au cas 1D, c'est-à-dire que $\vec{v} = u\vec{e}_x$ et $T = T(x)$. La distance $L$ entre A et B est fixée à 1 (longueur de référence) et l'équation doit être comprise comme une équation sans dimension. Le fluide est incompressible donc l'équation de conservation impose que $u$ est une constante.

On se limite aux conditions de Dirichlet, *i.e.* $T(0) = T_A$ et $T(1) = T_B$. Par la suite, on prendra $a = 0.3$ (on pourra essayer d'autres valeurs en fonction du temps disponible).

On décide de résoudre l'edp (4.86) avec des éléments finis. La formulation faible de l'edp et l'écriture sous forme matricielle du problème a été faite dans le TD03.

$$AT = B \tag{4.87}$$

avec

$$A = \begin{pmatrix} -u/2 + a/\Delta x + K & u/2 - a/\Delta x & 0 & \cdots & 0 \\ -(u/2 + a/\Delta x) & 2a/\Delta x & u/2 - a/\Delta x & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -(u/2 + a/\Delta x) & 2a/\Delta x & u/2 - a/\Delta x \\ 0 & \cdots & 0 & -(u/2 + a/\Delta x) & u/2 + a/\Delta x + K \end{pmatrix} \tag{4.88}$$

et

$$B = \begin{pmatrix} KT_A \\ 0 \\ \vdots \\ 0 \\ KT_B \end{pmatrix}, \quad K = 10^{30} \tag{4.89}$$

Le but de ce TD est de s'affranchir de la fonction *Inverse[]* de Mathematica.

1. Justifier la nécessité d'utiliser un préconditionneur lorsque l'on intègre les conditions limites avec la méthode de pénalisation. Proposer une matrice de préconditionnement $C$.
2. On résoud le système avec $u = 1$. Justifier l'utilisation de la méthode de décomposition $LU$ dans ce cas.
3. Écrire le système à résoudre en intégrant le préconditionnement. On notera $\tilde{A} = C^{-1}A$ et $\tilde{B} = C^{-1}B$.
4. Écrire l'algorithme pour calculer les matrices $L$ et $U$ de la décomposition $LU$. On pose $\tilde{A} = [a_{ij}]$, $L = [l_{ij}]$ et $U = [u_{ij}]$.

5. Écrire l'algorithme de résolution du système matriciel à partir de la décomposition $LU$.
6. Reprendre le programme du TD03 de résolution du problème avec la méthode des éléments finis et ajouter la décomposition $LU$ pour inverser le problème sans utiliser la fonction d'inversion intégrée à Mathematica.
7. On considère le problème de conduction pure, *i. e.* $u = 0$. Quelle est la conséquence sur la matrice $A$ ?
8. Justifier l'utilisation de la méthode itérative du gradient conjugué préconditionné.
9. Écrire l'algorithme du gradient conjugué préconditionné.
10. Reprendre le programme du TD03 de résolution du problème avec la méthode des éléments finis et résoudre le problème avec un résidu inférieur à $10^{-6}$. Si l'on prend comme condition initiale un champ nul, combien d'itérations faut-il à la méthode du gradient conjugué pour converger ? Que se passe-t-il si on impose la solution analytique comme condition initiale ?
11. Donner l'erreur par rapport à la solution analytique avec $u = 0$ en utilisant 11 points avec la méthode $LU$ et la méthode du gradient conjugué. Dans ce cas, la méthode $LU$ est équivalente à une autre méthode. Laquelle ?