UNIVERSITÉ Clermont Auvergne

POLYTECH CLERMONT-FERRAND

TRONC COMMUN

Introduction à l'analyse numérique

Rachid Touzani

Mars 2021

Ces notes constituent l'essentiel du cours dispensé à l'École *Polytech Clermont-Ferrand* en première année. Le cours est enseigné en Tronc Commun ; il est donc destiné aux élèves de tous les départements. Il a pour but de faire connaissance avec les notions élémentaires de l'Analyse Numérique dans le but de former des utilisateurs "avertis" des méthodes d'approximation numérique dans les sciences de l'ingénieur.
Ce cours suppose la connaissance de notions élémentaires de l'algèbre linéaire sur les matrices et les vecteurs ainsi que les notions simples de l'analyse et de l'algèbre linéaire : matrices et vecteurs, fonctions, dérivées, formule de Taylor, ...
Ces notes de cours ne contiennent que peu d'exemples ou de figures. Plusieurs exemples et figures seront présentés lors du cours magistral.

R. TOUZANI

Table des matières

1

Méthodes directes de résolution des systèmes linéaires

Soit A une matrice carrée d'ordre n à coefficients réels inversible et soit b un n–vecteur ($b \in \mathbb{R}^n$). On cherche un vecteur $x \in \mathbb{R}^n$ tel que

$$Ax = b.$$

On suppose que la matrice A, de coefficients (a_{ij}) est inversible, *i.e.*, qu'il existe une matrice notée A^{-1}, dite *matrice inverse de* A, telle que

$$A\,A^{-1} = A^{-1}A = I, \qquad I : \text{Matrice identité.}$$

De plus, on sait que la matrice A est inversible si et seulement la seule solution du système $Ax = 0$ est $x = 0$.

1.1 Méthode d'élimination de Gauss

Nous allons illustrer cette méthode sur un exemple particulier. Soit le système linéaire :

$$2\,x_1 + \ 4\,x_2 + \ 6\,x_3 = \ 2, \tag{1.1}$$
$$3\,x_1 + \ 8\,x_2 + 13\,x_3 = \ 5, \tag{1.2}$$
$$2\,x_1 + \ 9\,x_2 + 18\,x_3 = 11. \tag{1.3}$$

On cherche à éliminer successivement les inconnues x_1, x_2, x_3. En divisant l'équation (1.1) par 2, on obtient :

$$x_1 + 2x_2 + 3x_3 = 1 \tag{1.4}$$

En multipliant (1.4) par 3 (resp. 2) et en retranchant le résultat de (1.2) (resp. (1.3)) on obtient successivement :

$$2\,x_2 + \ 4\,x_3 = \ 2, \tag{1.5}$$
$$5\,x_2 + 12\,x_3 = \ 9. \tag{1.6}$$

On a ainsi, lors de cette première étape, éliminé la variable x_1. Continuons ainsi ; nous obtenons :

$$2\,x_3 = 4;\ \text{soit } x_3 = 2.$$

Pour calculer x_2, on a l'équation $x_2 + 2\,x_3 = 1$; ce qui nous donne $x_2 = -3$. La valeur de x_1 peut être calculée dans (1.4). Ainsi

$$x_1 = 1 - 2\,x_2 - 3\,x_3 = 1.$$

Nous disons alors que la résolution se fait par *remontée*. Les différentes étapes de transformation du système linéaire précédent peuvent être illustrées comme suit :

$$A^{(1)}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 6 \\ 3 & 8 & 13 \\ 2 & 9 & 18 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 11 \end{pmatrix} = b^{(1)}$$

$$A^{(2)}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 2 & 4 \\ 0 & 5 & 12 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 9 \end{pmatrix} = b^{(2)}$$

$$A^{(3)}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 4 \end{pmatrix} = b^{(3)}$$

Cette procédure peut être étendue au cas général (n quelconque). La méthode de Gauss peut ainsi être résumée en trois étapes :

(i) Élimination successive des inconnues ; ce qui équivaut à trouver une matrice inversible M telle que la matrice MA soit triangulaire supérieure ;

(ii) calcul simultané du vecteur Mb ;

(iii) résolution du système linéaire $MAx = Mb$.

Détaillons l'élimination de Gauss dans le cas d'une $n \times n$–matrice inversible quelconque :
On note tout d'abord $a_{ij}^{(1)} := a_{ij}$ les coefficients de A.
1^{er} *pas* : Le coefficient $a_{11}^{(1)}$ est appelé *pivot* au premier pas. On suppose $a_{11}^{(1)} \neq 0$ et on pose $p^{(1)} := 1/a_{11}^{(1)}$. On obtient au premier pas le système linéaire transformé :

$$\begin{pmatrix} 1 & a_{12}^{(2)} & a_{13}^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n-1,2}^{(2)} & \dots & a_{n-1,n-1}^{(2)} & a_{n-1,n}^{(2)} \\ 0 & a_{n,2}^{(2)} & \dots & \dots & a_{n,n}^{(2)} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(2)} \\ b_2^{(2)} \\ \vdots \\ \vdots \\ b_n^{(2)} \end{pmatrix}$$

avec

$$
\begin{aligned}
&a_{1j}^{(2)} = p^{(1)} a_{1j}^{(1)}, && j = 1, \ldots, n, \\
&b_1^{(2)} = p^{(1)} b_1^{(1)}, \\
&a_{kj}^{(2)} = a_{kj}^{(1)} - a_{k1}^{(1)} a_{1j}^{(2)}, && j = 2, \ldots, n, \\
&b_k^{(2)} = b_k^{(1)} - a_{k1}^{(1)} b_1^{(2)}, &&
\end{aligned}
\Bigg\} \quad k = 2, \ldots, n.
$$

On obtient ainsi après la première étape le système $A^{(2)}x = b^{(2)}$. Dans ce système, l'inconnue x_1 a été éliminée. Ainsi, à la k^{e} étape nous aurons une matrice $A^{(k)}$ de la forme :

$$
A^{(k)} = \begin{pmatrix}
\times & \times & \times & \times & \ldots & \ldots & \times \\
0 & \times & \ldots & \times & \ldots & \ldots & \vdots \\
0 & 0 & \times & \ldots & & \ldots & \vdots \\
0 & \ldots & 0 & \times & \ldots & \ldots & \vdots \\
\vdots & \ldots & \vdots & \vdots & \ddots & \ldots & \vdots \\
0 & \ldots & 0 & \times & & \ddots & \vdots \\
0 & \ldots & 0 & \times & \ldots & \ldots & \times
\end{pmatrix} \leftarrow k^{\text{e}} \text{ ligne}
$$

L'algorithme d'élimination s'écrit :

```
Pour  k = 2 à n faire
      Pour i = k à n faire
          s := a(k-1)_{i,k-1} / a(k-1)_{k-1,k-1};
          Pour j = k à n faire
              a(k)_{ij} := a(k-1)_{ij} - s * a(k-1)_{k-1,j};
          fin j;
          b(k)_i := b(k-1)_i - s * b(k)_{k-1};
      fin i;
fin k.
```

Nous pouvons faire les remarques suivantes :

1. Pour que la méthode de Gauss ait un sens, il faut que les pivots successifs $a_{kk}^{(k)}$ ne soient pas nuls. Autrement, on adopte une stratégie de *pivotage*, *i.e.*, on peut échanger des lignes de la matrice.
2. Du point de vue programmation, un examen minutieux de la méthode montre que le coefficient $a_{ij}^{(k)}$ peut occuper le même espace mémoire que $a_{ij}^{(k-1)}$. Il ne serait donc pas raisonnable de stocker en mémoire les différents coefficients $a_{ij}^{(k)}$, $k = 1, 2, \ldots$ étapes de l'élimination.

1.2 Factorisation LU d'une matrice

Supposons que tous les pivots sont non nuls. L'élimination de Gauss permet de construire une suite de matrices

$$A^{(1)} = A,\ A^{(2)} = E^{(2)}A^{(1)},\ A^{(3)} = E^{(3)}A^{(2)},\ \ldots,\ A^{(n)} = E^{(n)}A^{(n-1)},$$

où les matrices $E^{(k)}$ sont triangulaires inférieures. Dans l'exemple du système (1.1)–(1.3), on a

$$E^{(2)} = \begin{pmatrix} -\frac{1}{2} & 0 & 0 \\ -\frac{3}{2} & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}, \quad E^{(3)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{5}{2} & 1 \end{pmatrix}.$$

Ainsi

$$A^{(n)} = E^{(n)}E^{(n-1)} \ldots E^{(2)}A^{(1)}.$$

Donc, la matrice $L = (E^{(n)} \ldots E^{(2)})^{-1}$ est triangulaire inférieure. On dit alors qu'on a effectué une *factorisation* (ou *décomposition*) LU. Le système linéaire $Ax = b$ peut s'écrire $LUx = b$. En notant $y = Ux$, on se ramène à la résolution du système linéaire à matrice triangulaire inférieure $Ly = b$ (résolution par *descente*). La solution x est obtenue en résolvant le système linéaire à matrice triangulaire supérieure $Ux = y$ (résolution par *remontée*).

Définition 1.2.1 *Pour tout $k = 1, 2, \ldots, n$, la matrice*

$$\Delta_k = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1k} \\ a_{21} & a_{22} & \ldots & a_{2k} \\ \vdots & & \ddots & \vdots \\ a_{k1} & a_{k2} & \ldots & a_{kk} \end{pmatrix}$$

est appelée sous–matrice principale de A d'ordre k.

Théorème 1.2.1 *Soit A une $n \times n$–matrice telle que les sous–matrices principales soient inversibles. Alors, il existe une matrice triangulaire inférieure $L = (l_{ij})$ avec $l_{ii} = 1$, $1 \leq i \leq n$ et une matrice triangulaire supérieure U telles que*

$$A = LU.$$

De plus, une telle factorisation est unique.

Démonstration. Le résultat du théorème équivaut à dire que l'élimination de Gauss est possible.
Clairement, le premier pivot est a_{11} qui est supposé non nul puisque Δ_1 est inversible. Calculons encore le deuxième pivot :

$$\begin{aligned} a_{22}^{(2)} &= a_{22}^{(1)} - a_{21}^{(1)}a_{12}^{(2)} \\ &= a_{22} - a_{21}\frac{a_{12}}{a_{11}} \\ &= \frac{1}{a_{11}}(a_{22}a_{11} - a_{12}a_{21}) \\ &= \frac{1}{a_{11}} \det \Delta_2. \end{aligned}$$

Puisque Δ_2 est inversible on a $a_{22}^{(2)} \neq 0$.
Supposons les pivots $a_{11}^{(1)}, a_{22}^{(2)}, \ldots, a_{k-1,k-1}^{(k-1)}$ tous non nuls et montrons que $a_{kk}^{(k)}$ est non nul. On a

$$\begin{aligned}
\det \Delta_k &= \det \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1k} \\ a_{21} & a_{22} & \ldots & a_{2k} \\ \vdots & & \ddots & \vdots \\ a_{k1} & a_{k2} & \ldots & a_{kk} \end{pmatrix} \\
&= a_{11}^{(1)} \det \begin{pmatrix} 1 & a_{12}^{(2)} & \ldots & a_{1k}^{(2)} \\ 0 & a_{22}^{(2)} & \ldots & a_{2k}^{(2)} \\ \vdots & & \ddots & \vdots \\ 0 & a_{k2}^{(2)} & \ldots & a_{kk}^{(2)} \end{pmatrix} \\
&= a_{11}^{(1)} a_{22}^{(2)} \det \begin{pmatrix} 1 & a_{12}^{(3)} & \ldots & a_{1k}^{(3)} \\ 0 & a_{22}^{(3)} & \ldots & a_{2k}^{(3)} \\ \vdots & & \ddots & \vdots \\ 0 & a_{k2}^{(3)} & \ldots & a_{kk}^{(3)} \end{pmatrix} \\
&= \ldots = a_{11}^{(1)} \, a_{22}^{(2)} \ldots a_{kk}^{(k)}.
\end{aligned}$$

Puisque $\det \Delta_k \neq 0$, on a $a_{kk}^{(k)} \neq 0$. □

Comptons le nombre d'opérations pour l'élimination de Gauss.
Nous ne nous intéressons qu'aux opérations effectuées sur la matrice. Une étape k de l'élimination s'écrit :

$p := 1/a_{kk}^{(k)}$;
`Pour` $i = k, \ldots, n$ `faire`
  $s := a_{ik}^{(k)} * p$;
  `Pour` $j = k, \ldots, n$ `faire`
    $a_{ij}^{(k+1)} := a_{ij}^{(k)} - s * a_{kj}^{(k)}$;
  `fin` j;
`fin` i.

On a donc
- 1 division ;
- $n - k + 1$ multiplications ;
- $(n - k + 1)^2$ additions et $(n - k + 1)^2$ multiplications.

En supposant que tous les types d'opérations consomment le même temps de calcul, on déduit qu'on a en tout $2(n - k + 1)^2 + n - k + 2$ opérations élémentaires. On somme pour $k = 1, \ldots, n - 1$ en rappelant les formules :

$$\sum_{k=1}^{m} k = \frac{m(m+1)}{2}, \quad \sum_{k=1}^{m} k^2 = \frac{m(m+1)(2m+1)}{6}.$$

On déduit alors que le nombre total d'opérations élémentaires est en $\mathcal{O}(n^3/3)$.

1.3 Matrices symétriques définies positives : factorisation de Cholesky

Rappelons qu'une matrice carrée symétrique A est dite *symétrique définie positive* si on a

$$x^T Ax > 0 \qquad \text{pour tout } x \neq 0.$$

On montre, en outre, qu'une matrice symétrique A est définie positive si et seulement si toutes les sous-matrices principales ont des déterminants strictement positifs.
Nous donnons maintenant une nouvelle caractérisation de ces matrices.

Théorème 1.3.1 *Une condition nécessaire et suffisante pour qu'une matrice A soit symétrique et définie positive est qu'il existe une matrice L inversible triangulaire inférieure telle que*

$$A = LL^T.$$

Si les éléments diagonaux l_{ii} de L sont choisis strictement positifs, la décomposition est unique.

Démonstration.
i– *Condition suffisante :* Soit L inversible, triangulaire inférieure. La matrice $A = LL^T$ est symétrique. De plus, on a pour tout $x \in \mathbb{R}^n$, $x \neq 0$:

$$x^T Ax = x^T LL^T x = (L^T x)^T (L^T x) > 0.$$

ii– *Condition nécessaire :* Puisque A est symétrique et définie positive, tous les déterminants $\delta_k = \det \Delta_k$ sont positifs. On peut donc écrire (de façon unique) $A = LU$ où L est triangulaire inférieure avec $l_{ii} = 1$ et U est triangulaire supérieure. D'autre part, on a

$$0 < \delta_i = a_{11}^{(1)} a_{22}^{(2)} \dots a_{ii}^{(i)} \qquad 1 \leq i \leq n.$$

On en déduit que $u_{jj} = a_{jj}^{(j)} > 0$, $1 \leq j \leq n$. Soit la matrice diagonale D de coefficients $d_{ii} = \sqrt{u_{ii}}$. On a

$$A = (LD)(D^{-1}U).$$

Posons $B = LD$, $C = D^{-1}U$. La matrice B est triangulaire inférieure et on a $b_{ii} = \sqrt{u_{ii}}$, $1 \leq i \leq n$. De même, la matrice C est triangulaire supérieure et on a $c_{ii} = u_{ii}/\sqrt{u_{ii}} = \sqrt{u_{ii}}$, $1 \leq i \leq n$. Montrons que $B = C^T$. On a

$$A = A^T = BC = C^T B^T,$$

ou encore que $(C^{-1})^T B = B^T C^{-1}$.

La matrice $(C^{-1})^T B$ (resp. $B^T C^{-1}$) est triangulaire inférieure (resp. triangulaire supérieure) ; donc les deux matrices $(C^{-1})^T B$ et $B^T C^{-1}$ sont diagonales. D'autre part, les éléments diagonaux de $B^T C^{-1}$ sont égaux à 1. On a donc $(C^{-1})^T B = B^T C^{-1} = I$. D'où $C^T = B$.
Dans cette construction, les coefficients de la matrice D ont été choisis positifs. Montrons que c'est la seule factorisation possédant cette propriété. Soient B_1 et B_2 deux matrices triangulaires inférieures avec $(B_1)_{ii} > 0$, $(B_2)_{ii} > 0$ et $B_1 B_1^T = B_2 B_2^T = A$. Soient D_1 et D_2 les matrices diagonales d'éléments respectifs $(D_1)_{ii} = (B_1)_{ii}$ et $(D_2)_{ii} = (B_2)_{ii}$. On pose

$$L_1 = B_1 D_1^{-1},\ L_2 = B_2 D_2^{-1},\ U_1 = D_1 B_1^T,\ U_2 = D_2 B_2^T.$$

On a $A_1 = L_1 U_1 = L_2 U_2$, $(L_1)_{ii} = (L_2)_{ii} = 1$. La décomposition LU étant unique, on obtient

$$B_1 D_1^{-1} = B_2 D_2^{-1},\ D_1 B_1^T = D_2 B_2^T.$$

L'égalité des éléments diagonaux des matrices $D_1 B_1^T$ et $D_2 B_2^T$ implique

$$(B_1)_{ii}^2 = (B_2)_{ii}^2 \qquad 1 \le i \le n.$$

Donc $D_1 = D_2$ car $(B_1)_{ii} = (B_2)_{ii} > 0$. D'où $B_1 = B_2$. □

Écrivons maintenant l'algorithme de Cholesky. On a en utilisant le fait que L est triangulaire inférieure :

$$a_{ij} = \sum_{k=1}^{n} l_{ik} l_{jk} = \sum_{k=1}^{\min(i,j)} l_{ik} l_{jk} \qquad 1 \le i, j \le n.$$

Comme A est symétrique, il suffit d'écrire cette relation pour $1 \le i \le j \le n$ par exemple. On a :

$$a_{ii} = \sum_{k=1}^{i} l_{ik}^2 = l_{ii}^2 + \sum_{k=1}^{i-1} l_{ik}^2, \qquad i = 1, \ldots, n;$$

d'où

$$l_{ii} = \pm \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}.$$

De même, pour $1 \le i < j \le n$:

$$a_{ij} = \sum_{k=1}^{i} l_{ik} l_{jk} = \sum_{k=1}^{i-1} l_{ik} l_{jk} + l_{ii} l_{ji}.$$

D'où

$$l_{ji} = \frac{1}{l_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} \right).$$

Ces deux relations ont un sens d'après le théorème précédent. De plus, la condition $l_{ii} > 0$ permet de choisir la valeur positive de la racine.
Notons que le calcul des éléments de L se fait de la manière suivante :

$$
\begin{aligned}
& l_{11} = \sqrt{a_{11}}, \\
& l_{21} = \frac{a_{12}}{l_{11}},\ l_{22} = \sqrt{a_{22} - l_{21}^2}, \\
& \dots
\end{aligned}
$$

i.e., ligne par ligne. L'algorithme informatique est le suivant :

```
Pour i = 1,...,n faire
    s := 0;
    Pour k = 1,...,i-1 faire
        s := s + a_ik^2
    fin k;
    a_ii := sqrt(a_ii - s);
    Pour j = i+1,...,n faire
        s := 0;
        Pour k = 1,...,i-1 faire
            s := s + a_ik * a_jk;
        fin k;
        a_ji := (a_ji - s)/a_ii;
    fin j;
fini.
```

Comptons le nombre d'opérations. Pour chaque ligne i on a :

$$
\begin{aligned}
& (i-1) && \text{(additions + multiplications)}, \\
& +1 && \text{addition}, \\
& +1 && \text{extraction de racine carrée}, \\
& +(n-i)(i-1) && \text{(additions + multiplications)}, \\
& +(n-i) && \text{(additions + divisions)}.
\end{aligned}
$$

Sommons pour $i = 1, \dots, n$, nous obtenons le terme dominant :

$$
\sum_{i=1}^{n} (ni - i^2 + i - n) = \left(\frac{n^3}{6} + O(n^2) \right) \text{(additions + multiplications)}.
$$

Soit, au total $O(n^3/6)$ opérations élémentaires. Il est donc recommandé d'utiliser la méthode de Cholesky si la matrice est symétrique définie positive.

2

Méthodes itératives de résolution des systèmes linéaires

Les méthodes directes sont très coûteuses en termes de nombre d'opérations, donc de temps de calcul lorsque la taille du système linéaire est assez élevée. En effet, nous avons vu que la résolution d'un système d'ordre n requiert $O(n^3)$ opérations. On peut alors avoir recours à des méthodes où la solution est obtenue par itérations successives et où chaque itération consiste à résoudre un système linéaire moins coûteux.
On peut décrire de manière générique une classe de méthodes itératives de la manière suivante : Considérons le système linéaire

$$Ax = b$$

où A est une $n \times n$–matrice inversible, et soit une décomposition de A sous la forme

$$A = M - N$$

où M est une matrice inversible, *facile à inverser*. En se donnant un vecteur $x^0 \in \mathbb{R}^n$, on construit une suite de vecteurs $(x^{(k)})$ par les itérations

$$Mx^{(k+1)} = Nx^{(k)} + b, \qquad k = 0, 1, \ldots. \tag{2.1}$$

Notons que si cette méthode converge vers un vecteur $x \in \mathbb{R}^n$ alors on a nécessairement

$$Mx = Nx + b, \quad \text{donc } Ax = b.$$

On dit alors que la méthode itérative est *consistante*.

2.1 Convergence des méthodes itératives

Soit $x \in \mathbb{R}^n$; on définit la norme :

$$\|x\| := \Big(\sum_{i=1}^{n} x_i^2\Big)^{\frac{1}{2}}.$$

On dira alors que la méthode itérative (2.1) converge s'il existe un vecteur $x \in \mathbb{R}^n$ tel que

$$\lim_{k\to\infty} \|x^{(k)} - x\| = 0$$

pour tout choix de vecteur initial $x^{(0)} \in \mathbb{R}^n$.

Définition 2.1.1 *Soit A une matrice carrée d'ordre n. On appelle* rayon spectral *de A (on note $\rho(A)$) la plus grande valeur propre en module,* i.e.

$$\rho(A) := \max\{|\lambda_i|;\ \lambda_i \text{ valeur propre de } A,\ 1 \le i \le n\}.$$

On a alors le théorème fondamental suivant :

Théorème 2.1.1 *La méthode itérative* (2.1) *converge si et seulement si*

$$\rho(M^{-1}N) < 1.$$

Démonstration. Soit x la solution du système linéaire $Ax = b$. On en déduit $Mx = Nx + b$. Donc

$$M(x^{(k+1)} - x) = N(x^{(k)} - x).$$

Soit $e^{(k)} = x^{(k)} - x$. On a

$$(x^{(k)} - x) = M^{-1}N\,(x^{(k-1)} - x) = \ldots = (M^{-1}N)^k\,(x^{(0)} - x),$$

ou encore

$$e^{(k)} = B^k\,e^{(0)} \quad \text{où} \quad B = M^{-1}N.$$

Considérons la décomposition de Jordan de la matrice B :

$$B = P\Lambda P^{-1}.$$

On en déduit $B^k = P\Lambda^k P^{-1}$ où Λ est une matrice diagonale par blocs, chaque bloc Λ_i étant de la forme

$$\Lambda_i = \begin{pmatrix} \lambda_i & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_i \end{pmatrix}$$

On montre alors que si $|\lambda_i| < 1$ alors $\Lambda_i^k \to 0$ si $k \to \infty$. On en déduit que Λ^k, et donc B^k tendent vers la matrice 0.
La propriété inverse se montre facilement. □

2.2 Quelques méthodes itératives

Écrivons A sous la forme

$$A = L + D + U$$

où

$$d_{ij} = \begin{cases} a_{ii} & \text{si } i = j \\ 0 & \text{sinon} \end{cases}, \qquad l_{ij} = \begin{cases} a_{ij} & \text{si } i > j \\ 0 & \text{sinon} \end{cases}, \qquad u_{ij} = \begin{cases} a_{ij} & \text{si } i < j \\ 0 & \text{sinon} \end{cases}.$$

On suppose en outre que la matrice diagonale D est inversible, *i.e.* $a_{ii} \neq 0$ pour $1 \leq i \leq n$.

2.2.1 Méthode de Jacobi

Cette méthode s'écrit

$$D\, x^{(k+1)} := -(L+U)\, x^{(k)} + b, \qquad k = 0, 1, \ldots$$

ou encore

$$\begin{cases} x_i^{(k+1)} := \dfrac{1}{a_{ii}} \left(b_i - \displaystyle\sum_{j \neq i} a_{ij} x_j^{(k)} \right) & 1 \leq i \leq n,\ k = 0, 1, \ldots \\ x_i^{(0)} \quad \text{donné} & 1 \leq i \leq n. \end{cases}$$

En d'autres termes, nous avons choisi

$$M = D,\ N = -(L+U).$$

Chaque itération consiste donc à résoudre un système linéaire diagonal.

2.2.2 Méthodes de Gauss–Seidel et de relaxation

La présentation de la méthode de Jacobi montre que, pour calculer les itérés $x_i^{(k+1)}$, les valeurs $x_j^{(k+1)}$, $1 \leq j < i$ sont disponibles. On peut donc modifier cette méthode en écrivant :

$$\begin{cases} x_i^{(k+1)} := \dfrac{1}{a_{ii}} \left(b_i - \displaystyle\sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^{(k)} \right) & 1 \leq j \leq n,\ k = 0, 1, \ldots \\ x_i^{(0)} \quad \text{donné} & 1 \leq i \leq n. \end{cases}$$

Cette méthode peut encore être écrite sous la forme :

$$(D+L)\, x^{(k+1)} = -U\, x^{(k)} + b, \qquad k = 0, 1, \ldots.$$

La méthode de relaxation consiste à écrire

$$\left(\frac{1}{\omega} D + L \right) x^{(k+1)} = \left(\frac{1-\omega}{\omega}\, D - U \right) x^{(k)} + b, \qquad k = 0, 1, \ldots$$

où $\omega > 0$ est un réel donné. Nous avons ainsi le choix

$$M = \frac{1}{\omega}\, D + L, \quad N = \frac{1-\omega}{\omega}\, D - U.$$

Pour $\omega = 1$, nous retrouvons la méthode de Gauss Seidel. Pour $\omega < 1$ on appelle cette méthode *méthode de sous–relaxation* et pour $\omega > 1$ *méthode de sur–relaxation*.

2.3 Matrices symétriques définies positives

Nous allons maintenant examiner la convergence de ces méthodes itératives dans le cas d'une matrice symétrique définie positive.

Théorème 2.3.1 *Soit A une matrice symétrique définie positive et soit $A = M-N$ une décomposition de A où M est inversible. On suppose que la matrice $M^T + N$ est symétrique définie positive. Alors, la méthode itérative $Mx^{(k+1)} = Nx^{(k)} + b$ converge.*

On peut déduire de ce résultat le corollaire suivant :

Corollaire 2.3.1 *Si A est symétrique définie positive, la méthode de relaxation avec $0 < \omega < 2$ converge.*

Démonstration. Pour la méthode de relaxation, on a

$$M = \frac{1}{\omega}D + L, \quad N = \frac{1-\omega}{\omega}D - U.$$

Comme A est symétrique, on a $L = U^T$. La diagonale de M est celle de D. On en déduit que M est inversible puisque les éléments d_{ii} sont positifs. Soit

$$M^T + N = \frac{1}{\omega}D + L^T + \frac{1-\omega}{\omega}D - L^T = \frac{2-\omega}{\omega}D.$$

Cette matrice diagonale est définie positive si et seulement si $0 < \omega < 2$. □

Corollaire 2.3.2 *Soit A une matrice symétrique définie positive telle que $2D-A$ soit définie positive. Alors, la méthode de Jacobi converge.*

Démonstration. On a $M^T + N = 2D - A$. □

2.4 Matrices à diagonale dominante

Une autre catégorie de matrices pour lesquelles l'utilisation des méthodes de décomposition est intéressante est celle des matrices à diagonale dominante.
On dit qu'une matrice A est à *diagonale dominante* si on a

$$\sum_{j \neq i} |a_{ij}| \leq |a_{ii}| \qquad \forall\, i = 1, \ldots, n.$$

On dit qu'elle est à diagonale strictement dominante si l'inégalité ci-dessus est stricte.

Théorème 2.4.1 *Soit A une matrice à* diagonale strictement dominante*; alors A est inversible. De plus, les méthodes de Jacobi et de Gauss-Seidel convergent.*

Démonstration.
– Montrons d'abord que A est inversible : Soit $x \in \mathbb{R}^n$ avec $x \neq 0$ et $Ax = 0$. Soit i tel que $|x_i| \geq |x_j|$ pour tout $j = 1, \ldots, n$. On a

$$a_{ii}x_i = -\sum_{j \neq i} a_{ij}x_j.$$

Donc

$$|a_{ii}||x_i| = \Big|\sum_{j \neq i} a_{ij}x_j\Big| \leq \sum_{j \neq i} |a_{ij}|\,|x_j| \leq |x_i| \sum_{j \neq i} |a_{ij}|.$$

Ceci est impossible puisque $x \neq 0$. La matrice A est donc inversible.
– Montrons maintenant la convergence : Posons $B = M^{-1}N$ et soit λ une valeur propre et x vecteur propre de B, *i.e.*

$$Bx = \lambda x, \quad x \neq 0.$$

Puisque l'on s'intéresse à la plus grande valeur propre en module, on suppose $\lambda \neq 0$. On a ainsi

$$\left(M - \frac{1}{\lambda}\,N\right)x = 0.$$

Pour la méthode de Jacobi, cela devient

$$\left(D + \frac{1}{\lambda}\,L + \frac{1}{\lambda}\,U\right)x = 0.$$

Notons $C = D + \frac{1}{\lambda}\,L + \frac{1}{\lambda}\,U$. Supposons, par contradiction, que $|\lambda| \geq 1$, on a dans ce cas

$$|c_{ii}| = |a_{ii}| > \sum_{j \neq i} |a_{ij}| \geq \frac{1}{|\lambda|} \sum_{j \neq i} |a_{ij}| = \sum_{j \neq i} |c_{ij}|.$$

On en déduit que C est à diagonale strictement dominante, donc inversible. Donc $x = 0$. Ceci est une contradiction avec le fait que x est vecteur propre.
– Pour la méthode de Gauss-Seidel, on pose

$$C = D + L + \frac{1}{\lambda}\,U.$$

Ici encore, par contradiction, si $|\lambda| \geq 1$, on déduit

$$\begin{aligned} |c_{ii}| &= |a_{ii}| \\ &> \sum_{j \neq i} |a_{ij}| \\ &\geq \sum_{j<i} |a_{ij}| + \frac{1}{|\lambda|} \sum_{j>i} |a_{ij}| \\ &= \sum_{j \neq i} |c_{ij}|. \end{aligned}$$

On obtient ainsi une contradiction. □

2.5 Remarques et conclusions

L'utilisation des méthodes itératives est généralement plus avantageuse lorsque celles-ci convergent. Les méthodes de Jacobi, de Gauss-Seidel et de relaxation ne convergent le plus souvent que dans les cas que nous avons décrits dans ce chapitre. Il existe des méthodes itératives beaucoup plus élaborées et qui convergent dans des cas plus généraux que ceux décrits ici. Dans le cas de matrices creuses, *i.e.*, les matrices contenant un nombre significatif de coefficients nuls, ces méthodes sont beaucoup plus économiques puisqu'il n'est pas nécessaire de stocker en mémoire les coefficients nuls ni d'effectuer les opérations les utilisant.
La méthode de Gauss-Seidel converge généralement plus vite et plus souvent que la méthode de Jacobi. La méthode de relaxation nécessite la connaissance d'une valeur optimale du paramètre de relaxation ω.
La convergence des méthodes itératives se teste généralement en utilisant un paramètre de *tolérance* ϵ. On arrête ainsi les calculs dès que l'erreur relative est assez petite :

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|\boldsymbol{x}^{(k)}\|} < \epsilon.$$

Une autre alternative est de tester le résidu :

$$\|Ax^{(k+1)} - b\| < \epsilon.$$

3

Interpolation et approximation de fonctions

Nous nous intéressons, dans ce chapitre, à l'approximation de fonctions soit par interpolation soit par moindres carrés.

3.1 Interpolation de Lagrange

Considérons un ensemble de points dans le plan

$$(x_0, y_0), (x_1, y_1), \ldots, (x_k, y_k), \ldots$$

Ces points peuvent constituer par exemple des résultats de mesures, des données d'expériences ou des résultats de calculs numériques. On cherche à déterminer une fonction "simple" passant par ces points. On parle d'*interpolation*. La connaissance d'une telle fonction permettrait non seulement de donner une approximation des valeurs en dehors de ces points mais aussi par exemple de calculer facilement la dérivée ou une intégrale. L'exemple le plus simple de fonction que l'on pourrait déterminer est celui d'un polynôme. En effet, un polynôme est une fonction facile à évaluer et indéfiniment dérivable.
On se donne donc une collection de $n+1$ points

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

de $\mathbb{R}^2$, d'abscisses distincts deux à deux. On cherche un polynôme de degré inférieur ou égal à n :

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

vérifiant les relations

$$p(x_i) = y_i \qquad 0 \leq i \leq n.$$

Notons que ceci équivaut à rechercher les $n+1$ coefficients (a_i) du polynôme p.

Remarque 3.1.1 *Ce problème peut s'écrire sous la forme d'un système linéaire :*

$$\sum_{j=0}^{n} x_i^j a_j = y_i \qquad 0 \leq j \leq n.$$

Il suffirait donc de résoudre ce système pour trouver les coefficients du polynôme p. Cette procédure peut se révéler cependant coûteuse et poser quelques problèmes numériques que nous n'aborderons pas ici.

3.1.1 Base de Lagrange

On définit, pour tout $i = 0, 1, \ldots, n$, le polynôme dit de *Lagrange* :

$$L_i(x) = \frac{(x - x_0)\,(x - x_1)\,\ldots\,(x - x_{i-1})\,(x - x_{i+1})\,\ldots\,(x - x_n)}{(x_i - x_0)\,(x_i - x_1)\,\ldots\,(x_i - x_{i-1})\,(x_i - x_{i+1})\,\ldots\,(x_i - x_n)} = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}.$$

Clairement L_i est un polynôme de degré n vérifiant

$$L_i(x_i) = 1 \quad \text{et} \quad L_i(x_j) = 0 \quad \text{si } i \neq j.$$

Les polynômes L_i sont linéairement indépendants. En effet, si $b_0, b_1, \ldots, b_n$ sont $(n+1)$ réels tels que

$$\sum_{j=0}^{n} b_j L_j(x) = 0 \qquad \text{pour tout } x \in \mathbb{R},$$

nous en déduisons pour $x = x_i$,

$$0 = \sum_{j=0}^{n} b_j L_j(x_i) = b_i.$$

On a donc $b_i = 0$ pour tout $i = 0, \ldots, n$. Notons par $\mathbb{P}_n$ l'espace des polynômes de degré inférieur ou égal à n. Nous avons ainsi montré que la famille $(L_i)_{i=0}^n$ forme une base de l'espace $\mathbb{P}_n$, appelée *Base de Lagrange*.
Considérons maintenant le polynôme de $\mathbb{P}_n$,

$$p(x) = \sum_{j=0}^{n} y_j L_j(x) \qquad x \in \mathbb{R}.$$

Ce polynôme vérifie les relations $p(x_i) = y_i$, $i = 0, \ldots, n$. Nous avons donc montré qu'il existe un polynôme de degré inférieur ou égal à n vérifiant $p(x_i) = y_i$. Montrons qu'un tel polynôme est unique. Supposons qu'il existe un autre polynôme q de $\mathbb{P}_n$ vérifiant le même type d'identité. Le polynôme $r(x) = p(x) - q(x)$ est donc un polynôme de $\mathbb{P}_n$ vérifiant $r(x_i) = 0$ pour $i = 0, \ldots, n$. Puisque les polynômes L_j sont linéairement indépendants, on trouve $r = 0$. D'où $p = q$.

3.1.2 Erreur d'interpolation

On veut maintenant évaluer l'erreur entre une fonction donnée f et le polynôme de Lagrange interpolant f. Supposons que la fonction f est de classe C^{n+1}, *i.e.* $(n+1)$ fois continûment dérivable. Soit $x \neq x_i$ et soit π_n le polynôme

$$\pi_n(x) = (x - x_0)\,(x - x_1)\,\ldots\,(x - x_n).$$

On définit le polynôme $q \in \mathbb{P}_{n+1}$ par

$$q(t) = p(t) + \frac{f(x) - p(x)}{\pi_n(x)}\,\pi_n(t).$$

On a $q(x_i) = p(x_i) = f(x_i)$ pour tout $i = 0, \ldots, n$. De plus

$$q(x) = p(x) + f(x) - p(x) = f(x).$$

La fonction q est donc le polynôme d'interpolation de Lagrange de f aux points $x, x_0, \ldots, x_n$. Soit $F = f - q$. Cette fonction admet aux points x_i un zéro de multiplicité ≥ 1 et de même au point x. Le nombre de zéros de F, multiplicités comprises, est donc $\geq n + 2$. Par le théorème de Rolle, F' s'annule entre deux zéros. Donc F' a au moins $n + 1$ zéros. On en déduit par récurrence que $F^{(n+1)}$ a au moins un zéro, noté ξ_x. Donc

$$0 = F^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_x) - q^{(n+1)}(\xi_x).$$

Le polynôme q est de degré $n + 1$. Son terme de plus haut degré est

$$\frac{f(x) - p(x)}{\pi_n(x)}\,t^{n+1}.$$

D'où

$$q^{(n+1)}(t) = \frac{f(x) - p(x)}{\pi_n(x)}\,(n+1)!$$

On en déduit ainsi le théorème suivant :

Théorème 3.1.1 *Soit f une fonction de classe C^{n+1} et soit p le polynôme d'interpolation de Lagrange de f, aux points $x_0, x_1, \ldots, x_n$ (polynôme de degré inférieur ou égal à n). Alors on a l'inégalité*

$$|f(x) - p(x)| \leq \frac{M_{n+1}}{(n+1)!}\,|\pi_n(x)| \qquad \textit{pour tout } x \in \mathbb{R},$$

où

$$M_{n+1} := \sup_{x \in \mathbb{R}} |f^{(n+1)}(x)|,$$
$$\pi_n(x) = (x - x_0)\,(x - x_1)\,\ldots\,(x - x_n).$$

3.1.3 Calcul du polynôme de Lagrange

En fait, le calcul des polynômes de Lagrange s'avère assez coûteux. En particulier, si l'on veut augmenter le degré du polynôme d'interpolation, tout le calcul est à refaire. Pour remédier à cela, nous allons présenter l'algorithme des *différences divisées de Newton*.
Notons par p_k le polynôme d'interpolation de Lagrange aux points $x_0, x_1, \ldots, x_k$ (de degré k) pour $0 \leq k \leq n$. On a

$$p_0(x) = f(x_0).$$

Pour $k \geq 1$, on a $p_k - p_{k-1} \in \mathbb{P}_k$ et de plus

$$p_k(x_i) - p_{k-1}(x_i) = 0 \qquad i = 0, \ldots, k-1.$$

On en déduit que ce polynôme est de la forme :

$$p_k(x) - p_{k-1}(x) = f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\ldots(x - x_{k-1}),$$

où $f[x_0, x_1, \ldots, x_k]$ est le coefficient de x^k dans $p_k(x)$. On en déduit alors par récurrence :

$$p_n(x) = f(x_0) + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\ldots(x - x_{k-1}).$$

Lemme 3.1.1 *Pour $k \geq 1$, on a*

$$f[x_0, x_1, \ldots, x_k] = \frac{f[x_1, \ldots, x_k] - f[x_0, x_1, \ldots, x_{k-1}]}{x_k - x_0},$$

et $f[x_i] = f(x_i)$.

Démonstration. Soit $k \geq 1$ et soit $\widetilde{p}_{k-1} \in \mathbb{P}_{k-1}$ le polynôme d'interpolation de f aux points $x_1, \ldots, x_k$. Le coefficient du terme x^{k-1} dans ce polynôme est $f[x_1, \ldots, x_k]$. En outre, le polynôme $q_k \in \mathbb{P}_k$ défini par

$$q_k(x) = \frac{(x - x_0)\widetilde{p}_{k-1}(x) - (x - x_k)p_{k-1}(x)}{x_k - x_0},$$

coïncide avec f aux points $x_0, \ldots, x_k$. Donc $q_k = p_k$ et on obtient le résultat désiré en identifiant les coefficients de x^k dans les deux membres.

On peut ainsi schématiser l'algorithme dit de *Newton* :

$$\begin{array}{llll}
f[x_0] & & & \\
 & f[x_0, x_1] & & \\
f[x_1] & & f[x_0, x_1, x_2] & \\
 & f[x_1, x_2] & & f[x_0, x_1, x_2, x_3] \\
f[x_2] & & f[x_1, x_2, x_3] & \\
 & f[x_2, x_3] & \cdot & \cdot \\
f[x_3] & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot
\end{array}$$

L'intérêt principal de cet algorithme est qu'il ne nécessite pas de recalculer tous les coefficients du polynôme de Lagrange lorsqu'on ajoute un point d'interpolation.

3.2 Approximation au sens des moindres carrés

On se donne m points distincts $x_1, \ldots, x_m$ de $\mathbb{R}$ et m valeurs numériques associées $y_1, \ldots, y_m$. Au lieu de chercher une fonction qui soit égale à y_i en x_i, on veut construire une courbe qui passe *aussi près que possible* des valeurs y_i.
L'exemple typique d'utilisation de ce type d'approximation est celui d'un modèle mathématique dépendant d'un certain nombre de paramètres que l'on souhaite ajuster à des mesures par une fonction

$$\phi = \sum_{j=1}^{n} a_j w_j \tag{3.1}$$

où a_i sont les paramètres à déterminer.
Soit $(w_j)_{j=1}^n$ un ensemble de n fonctions réelles linéairement indépendantes définies sur un intervalle contenant les points x_i. On cherche une fonction U de la forme (3.1) telle que les égalités soient approchées au « mieux ». Par exemple, on cherche une fonction U qui rend minimum le nombre

$$\sum_{i=1}^{m} \Big(\sum_{j=1}^{n} a_j w_j(x_i) - y_i \Big)^2 \qquad \text{lorsque } (a_j)_{j=1}^n \in \mathbb{R}^n.$$

On peut également écrire ceci sous la forme

$$\begin{cases} \text{Trouver } \bar{a} = (\bar{a}_i) \in \mathbb{R}^n \text{ tel que :} \\ \displaystyle\sum_{i=1}^{m} \Big(\sum_{j=1}^{n} \bar{a}_j w_j(x_i) - y_i \Big)^2 = \inf_{a \in \mathbb{R}^n} \sum_{i=1}^{m} \Big(\sum_{j=1}^{n} a_j w_j(x_i) - y_i \Big)^2, \end{cases}$$

Notons

$$E(a) = \sum_{i=1}^{m} \Big(\sum_{j=1}^{n} a_j w_j(x_i) - y_i \Big)^2 = \|Ba - y\|^2,$$

où $\|\cdot\|$ est la norme euclidienne sur $\mathbb{R}^m$, B est la matrice de coefficients $b_{ij} = w_j(x_i)$ et y est le vecteur de coefficients y_i.

Théorème 3.2.1 *Soit B une $m \times n$–matrice avec $m > n$ et $y \in \mathbb{R}^m$. Une condition nécessaire et suffisante pour que $\bar{a} \in \mathbb{R}^n$ réalise le minimum de la fonction $E(a)$ est que*

$$B^{\mathsf{T}} B\, \bar{a} = B^{\mathsf{T}} y.$$

Ce système admet toujours au moins une solution. Si rang(A) = n, la solution est unique.

Démonstration.
(i) Soit $\bar{a} \in \mathbb{R}^n$ tel que $B^{\mathsf{T}} B\bar{a} = B^{\mathsf{T}} y$ et soit $a \in \mathbb{R}^n$. Notons par $(\cdot, \cdot)$ le produit scalaire dans $\mathbb{R}^n$, *i.e.*

$$(a, b) = \sum_{j=1}^{n} a_j b_j.$$

On a

$$\begin{aligned} E(\bar{a}+a) &= (B(\bar{a}+a)-y, B(\bar{a}+a)-y) \\ &= (B\bar{a}, B\bar{a}) + (Ba, Ba) + 2(B\bar{a}, Ba) - 2(B\bar{a}, y) - 2(Ba, y) + (y, y) \\ &= (B^{\mathsf{T}}B\bar{a}, \bar{a}) + (B^{\mathsf{T}}Ba, a) + 2(B^{\mathsf{T}}B\bar{a}, a) - 2(B^{\mathsf{T}}y, \bar{a}) - 2(B^{\mathsf{T}}y, a) + (y, y) \\ &= -(B^{\mathsf{T}}y, \bar{a}) + (B^{\mathsf{T}}Ba, a) + (y, y) \\ &\geq (y, y) - (B^{\mathsf{T}}y, \bar{a}). \end{aligned}$$

Or

$$\begin{aligned} E(a) &= (Ba-y, Ba-y) \\ &= (Ba, Ba) - 2(y, Ba) + (y, y) \\ &= (y, y) - 2(B^{\mathsf{T}}y, a) + (B^{T}Ba, a) \\ &= (y, y) - (B^{\mathsf{T}}y, a). \end{aligned}$$

D'où

$$E(\bar{a}+a) \geq E(\bar{a}) \qquad \text{pour tout } a \in \mathbb{R}^n.$$

(ii) Soit $\bar{a} \in \mathbb{R}^n$ tel que

$$E(\bar{a}) \leq E(a) \qquad \text{pour tout } a \in \mathbb{R}^n.$$

Le vecteur qui minimise E annule le gradient de E. On a pour tout $k = 1, \ldots, m$:

$$\begin{aligned} E(a) &= \sum_{i=1}^{m} \Big((Ba)_i - y_i \Big)^2 \\ &= \sum_{i=1}^{m} \Big(\sum_{j=1}^{n} b_{ij} a_j - y_i \Big)^2. \end{aligned}$$

Donc, pour tout $k = 1, \ldots, m$:

$$\begin{aligned} \frac{\partial E}{\partial a_k} &= 2 \sum_{i=1}^{n} \sum_{j=1}^{n} b_{ij} b_{ik} a_j - 2 \sum_{i=1}^{m} b_{ik} y_i \\ &= 2\,(B^{\mathsf{T}}Ba - B^{\mathsf{T}}y)_k. \end{aligned}$$

D'où le résultat.
(iii) Le second membre $B^{\mathsf{T}}y$ est orthogonal au noyau de la matrice $B^{\mathsf{T}}B$. Il existe donc toujours une solution. Si rang$(B) = n$, la matrice $B^{\mathsf{T}}B$ est inversible et la solution est donc unique. □

Ainsi, le problème de minimisation se ramène à la résolution du système linéaire

$$B^{\mathsf{T}}Ba = B^{\mathsf{T}}y.$$

On dit alors que u est solution du système surdéterminé $Ba = y$ *au sens des moindres carrés*.

Remarque 3.2.1 *Prenons le cas où la fonction w est un polynôme de degré $\leq n-1$. Ainsi, $w_j(x) = x^{j-1}$, $1 \leq j \leq n$. Il est clair alors que la matrice A est de rang n et on a donc unicité du polynôme d'approximation.*

4

Dérivation numérique

4.1 Introduction

Soit f une fonction de classe C^1 sur $\mathbb{R}$. La dérivée de f en un point $x \in \mathbb{R}$ est définie par

$$f'(x) := \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

On se propose de trouver des méthodes numériques pour approcher cette dérivée, le calcul effectif de la dérivée pouvant être soit impossible (dérivées intervenant dans des équations différentielles par exemple), soit trop onéreux.
L'idée la plus immédiate est de calculer le quotient différentiel ci-dessus avec une valeur de h "assez petite", *i.e.*, on pose

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}.$$

4.2 Erreur d'arrondi

Soit δ la précision relative de la machine (*ex. :* 7 chiffres significatifs $\Longrightarrow$ $\delta = 10^{-7}$). L'erreur absolue sur l'évaluation d'une fonction f en un point x est de l'ordre de $\delta|f(x)|$. Si h est donné exactement, ainsi que x et $x+h$, on obtient une borne supérieure de l'erreur sur le numérateur donnée par

$$\delta|f(x+h)| + \delta|f(x)| \approx 2\delta|f(x)|.$$

Ainsi l'erreur absolue sur le quotient différentiel sera donnée par

$$E_a \approx 2\delta \left| \frac{f(x)}{h} \right|.$$

En reprenant l'exemple précédent ($\delta = 10^{-3}$), on a

- Pour $h = 0{,}1$; $\quad E_a = 2 \times 10^{-3} \dfrac{49}{0{,}1} = 0{,}98 \approx 1.$
- Pour $h = 0{,}01$; $\quad E_a = 2 \times 10^{-3} \dfrac{49}{0{,}01} \approx 10.$

L'expression de E_a montre clairement que pour que l'erreur soit petite, il faut que δ soit petite — cela va de soi — et que $|h|$ soit "grand".

4.3 Erreur de troncature

Supposons f de classe C^2. L'erreur introduite en remplaçant la dérivée par le quotient différentiel peut être évaluée en utilisant la formule de Taylor. On écrit

$$f(x+h) = f(x) + f'(x)h + f''(\widetilde{x})\frac{h^2}{2}, \qquad \text{où } |x - \widetilde{x}| < |h|.$$

Ainsi

$$f'(x) = \frac{f(x+h) - f(x)}{h} - f''(\widetilde{x})\frac{h}{2}.$$

L'erreur de troncature pour des petites valeurs de $|h|$ est définie par

$$E_t = \frac{|h|}{2}|f''(\widetilde{x})|.$$

On en déduit qu'il faut que $|h|$ soit suffisamment petit pour que E_t soit petit et qu'il ne faut pas qu'il soit trop petit pour que E_a ne soit pas trop grand !!!

Théorème 4.3.1 *La quantité*

$$E = 2\delta \left| \frac{f(x)}{h} \right| + \frac{|h|}{2}|f''(\widetilde{x})|$$

est minimale pour

$$|h| = 2\sqrt{\delta \left| \frac{f(x)}{f''(\widetilde{x})} \right|}.$$

Démonstration. On pose

$$g(x) := 2\delta \frac{|f(x)|}{s} + \frac{s}{2}|f''(\widetilde{x})| = \frac{a}{s} + bs$$

où

$$a = 2\delta f(x), \quad b = \frac{1}{2}|f''(\widetilde{x})|.$$

On en déduit

$$g'(\overline{s}) = 0 \Longleftrightarrow -\frac{a}{\overline{s}^2} + b \Longleftrightarrow \overline{s} = \sqrt{\frac{a}{b}}.$$

Ainsi

$$|h| = 2\sqrt{\delta \left| \frac{f(x)}{f''(\widetilde{x})} \right|}. \quad \square$$

Exemple : On prend $f(x) = x^2$ et $\delta = 10^{-3}$. On en déduit

$$f''(x) = 2, \quad h = \sqrt{2\delta f(x)} = \sqrt{2\delta}\,|x|.$$

Si $x = 7$, on a $h = 10\sqrt{\delta} \approx 0{,}3$. □
Le quotient différentiel

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

est appelé *différence décentrée à droite*. De manière analogue, la formule de *différence décentrée à gauche* est donnée par :

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

On peut prendre une différence centrée autour de x, *i.e.*

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

Dans ce cas, l'erreur d'arrondi reste la même. Par contre, on modifie l'erreur de troncature.

Théorème 4.3.2 *Si f est de classe C^3 et de dérivées bornées jusqu'à l'ordre 3, on a pour tout $x \in \mathbb{R}$:*

$$\left| f'(x) - \frac{f(x+h) - f(x-h)}{2h} \right| \leq \frac{h^2}{6} \sup_{y \in \mathbb{R}} |f'''(y)|.$$

Démonstration. On se restreint au cas $h > 0$. On a par la formule de Taylor :

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \frac{h^3}{6} f'''(\xi), \qquad \text{où } \xi \in [x, x+h],$$
$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(x) - \frac{h^3}{6} f'''(\eta), \qquad \text{où } \eta \in [x-h, x].$$

Donc

$$f'(x) - \frac{f(x+h) - f(x-h)}{2h} = -\frac{h^2}{12} \left(f'''(\xi) + f'''(\eta) \right).$$

D'où le résultat. □

On s'intéresse maintenant à la dérivée seconde. On utilise pour cela la formule :

$$\begin{aligned} f''(x) &\approx \frac{f'(x + \frac{h}{2}) - f'(x - \frac{h}{2})}{h} \\ &\approx \frac{1}{h} \left(\frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h)}{h} \right) \\ &= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \end{aligned}$$

On montre alors par les mêmes méthodes le résultat suivant :

Théorème 4.3.3 *On suppose que f est de classe C^4 et de dérivées bornées jusqu'à l'ordre 4. Alors on a pour tout $x \in \mathbb{R}$, la majoration :*

$$\left| f''(x) - \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \right| \leq \frac{h^2}{12} \sup_{y \in \mathbb{R}} |f^{(4)}(y)|.$$

5

Intégration numérique

5.1 Généralités

Soit $f : [a, b] \to \mathbb{R}$ une fonction continue, nous nous intéressons au calcul de l'intégrale

$$\int_a^b f(x)\,dx.$$

Soit $a \leq x_0 < x_1 < \ldots < x_n \leq b$ $(n+1)$ points distincts pris dans l'intervalle $[a, b]$ et soit p un polynôme de degré inférieur ou égal à n, l'interpolant de Lagrange de f aux points $(x_i)_{i=0}^n$. On a donc

$$p(x) = \sum_{i=0}^{n} f(x_i)L_i(x), \qquad x \in [a, b]$$

où

$$L_i(x) := \frac{(x - x_0)\ldots(x - x_{i-1})\,(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})\,(x_i - x_{i+1})\ldots(x_i - x_n)}.$$

Notons

$$w_i := \int_a^b L_i(x)\,dx.$$

Le calcul des w_i est particulièrement aisé puisqu'il s'agit d'intégrales de polynômes. On peut ainsi « approcher » l'intégrale $\int_a^b f(x)\,dx$ par

$$\int_a^b p(x)\,dx = \sum_{i=0}^{n} w_i f(x_i).$$

De faÃ§on générale, si $(x_i)_{i=0}^n$ est une suite de $(n+1)$ points de l'intervalle $[a, b]$ appelés *points d'intégration numérique* et si $(w_i)_{i=0}^n$ est une suite de nombres réels appelés *poids de la formule d'intégration numérique*, nous dirons que l'expression

$$I(f) := \sum_{i=0}^{n} w_i f(x_i)$$

est une *formule d'intégration numérique* pour f sur l'intervalle $[a, b]$.

Remarque 5.1.1 *Si f est un polynôme de degré n, alors*

$$I(f) = \int_a^b f(x)\,dx.$$

Nous nous intéresserons donc aux formules d'intégration numérique qui sont exactes pour des polynômes.

Nous allons maintenant donner un théorème général de majoration de l'erreur. Pour cela, nous considérerons l'intégrale d'une fonction $f : [a,b] \to \mathbb{R}$, sur l'intervalle $[\alpha,\beta]$. Nous notons les points d'intégration numérique :

$$x_i = \alpha + \theta_i(\beta - \alpha) \qquad 1 \le i \le m,$$

où $0 \le \theta_i \le 1$.

Théorème 5.1.1 *On suppose que $f \in C^{n+1}[a,b]$ et soient $\alpha,\beta \in [a,b]$ tels que $a \le \alpha < \beta \le b$. On suppose que la formule d'intégration numérique :*

$$I(f) = (\beta - \alpha)\sum_{i=1}^{m} w_i f(\alpha + \theta_i(\beta - \alpha))$$

est exacte pour des polynômes de degré n, i.e.

$$I(g) = \int_\alpha^\beta g(x)\,dx \qquad \textit{pour tout } g \textit{ polynôme de degré } n.$$

Alors, on a l'inégalité :

$$\Big|\int_\alpha^\beta f(x)\,dx - I(f)\Big| \le \frac{(\beta-\alpha)^{n+1}}{n\,!}\Big(1 + \sum_{i=1}^{m}|w_i|\theta_i^n\Big)\int_\alpha^\beta |f^{(n+1)}(x)|\,dx.$$

Démonstration. Écrivons la formule de Taylor avec reste intégral pour $x \in [\alpha,\beta]$:

$$f(x) = \sum_{i=0}^{n}\frac{f^{(i)}(\alpha)}{i\,!}(x-\alpha)^i + \frac{1}{n\,!}\int_\alpha^x (x-y)^n f^{(n+1)}(y)\,dy.$$

Si

$$g(x) := \sum_{i=0}^{n}\frac{f^{(i)}(\alpha)}{i\,!}(x-\alpha)^i \quad \text{et} \quad r(x) := \frac{1}{n\,!}\int_\alpha^x (x-y)^n f^{(n+1)}(y)\,dy,$$

on a $f(x) = g(x) + r(x)$. Comme l'application $f \mapsto I(f)$ est linéaire on a :

$$\Big|\int_\alpha^\beta f(x)\,dx - I(f)\Big| \le \Big|\int_\alpha^\beta g(x)\,dx - I(g)\Big| + \Big|\int_\alpha^\beta r(x)\,dx - I(r)\Big|.$$

Puisque $g \in \mathbb{P}_n$, on a $\int_\alpha^\beta g(x)\,dx = I(g)$ et donc

$$\begin{aligned}\Big|\int_\alpha^\beta f(x)\,dx - I(f)\Big| &\le \Big|\int_\alpha^\beta r(x)\,dx - I(r)\Big| \\ &\le \Big|\int_\alpha^\beta r(x)\,dx\Big| + |I(r)| \\ &\le \frac{1}{n\,!}\int_\alpha^\beta \int_\alpha^x |x-y|^n |f^{(n+1)}(y)|\,dy + |I(r)| \\ &\le \frac{1}{n\,!}(\beta-\alpha)^{n+1}\int_\alpha^\beta |f^{(n+1)}(y)|\,dy + |I(r)|.\end{aligned}$$

Pour évaluer $I(f)$ on est conduit à majorer

$$\begin{aligned}|r(\alpha+\theta_i(\beta-\alpha))| &\le \frac{1}{n\,!}\int_\alpha^{\alpha+\theta_i(\beta-\alpha)} |\alpha-\theta_i(\beta-\alpha)|^n |f^{(n+1)}(y)|\,dy \\ &\le \frac{\theta_i^n}{n\,!}(\beta-\alpha)^n\int_\alpha^\beta |f^{(n+1)}(y)|\,dy.\end{aligned}$$

Ainsi

$$\begin{aligned}|I(r)| &= \Big|(\beta-\alpha)\sum_{i=1}^m w_i\, r(\alpha+\theta_i(\beta-\alpha))\Big| \\ &\le \frac{(\beta-\alpha)^{n+1}}{n\,!}\Big(\sum_{i=1}^m |w_i|\theta_i^n\Big)\int_\alpha^\beta |f^{(n+1)}(y)|\,dy. \qquad \square\end{aligned}$$

5.2 Méthodes d'intégration numérique par morceaux

Au lieu d'augmenter le nombre de points d'intégration numérique pour augmenter la précision, nous allons construire une subdivision de l'intervalle $[a,b]$:

$$a = x_0 < x_1 < \ldots < x_{k-1} < x_k = b,$$

en posant $h = \max_{0\le i\le k-1}(x_{i+1}-x_i)$ et utiliser une méthode d'ordre « peu élevé » sur chaque intervalle $[x_i, x_{i+1}]$. La précision de la méthode d'intégration numérique ainsi construite peut être déterminée en fonction de h.
Nous allons donner maintenant quelques exemples de formules d'intégration numérique composées.

5.2.1 Formules des rectangles

On choisit $s_i \in [x_i, x_{i+1}]$ et on remplace f sur $[x_i, x_{i+1}]$ par le polynôme de degré 0 : $p_0(x) = f(s_i)$. On a alors

$$\int_{x_i}^{x_{i+1}} f(x)\,dx \approx (x_{i+1}-x_i)f(s_i)$$

et donc

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{k-1}(x_{i+1}-x_i)f(s_i).$$

On peut ainsi choisir

$$\begin{aligned}
&s_i = x_i : && \text{formule des rectangles à gauche,}\\
&s_i = x_{i+1} : && \text{formule des rectangles à droite,}\\
&s_i = \frac{x_i + x_{i+1}}{2} : && \text{formule du point milieu.}
\end{aligned}$$

Théorème 5.2.1 *On suppose $f \in C^1[a,b]$ et on note*

$$\begin{aligned}
I_h^1(f) &= \sum_{i=0}^{k-1}(x_{i+1}-x_i)f(x_i),\\
I_h^2(f) &= \sum_{i=0}^{k-1}(x_{i+1}-x_i)f(x_{i+1}),\\
I_h^0(f) &= \sum_{i=0}^{k-1}(x_{i+1}-x_i)f\Big(\frac{x_i+x_{i+1}}{2}\Big).
\end{aligned}$$

Alors on a :

$$\begin{aligned}
\Big|\int_a^b f(x)\,dx - I_h^1(f)\Big| &\le h\int_a^b |f'(s)|\,ds,\\
\Big|\int_a^b f(x)\,dx - I_h^2(f)\Big| &\le 2h\int_a^b |f'(s)|\,ds.
\end{aligned}$$

De plus, si $f \in C^2[a,b]$, on a

$$\Big|\int_a^b f(x)\,dx - I_h^0(f)\Big| \le \frac{3}{2}h^2\int_a^b |f''(s)|\,ds.$$

Démonstration.
(i) Examinons d'abord le cas de I_h^1. Notons par g_h la fonction définie par

$$\begin{cases} g_h(x_i^+) = f(x_i) & 0 \le i \le k,\\ g_{h|[x_i,x_{i+1}]} = \text{Const.} & 0 \le i \le k-1. \end{cases}$$

On a clairement

$$I_h^1(f) = \int_a^b g_h(x)\,dx.$$

On écrit

$$\begin{aligned}\Big|\int_a^b f(x)\,dx - I_h^1(f)\Big| &= \Big|\int_a^b (f(x)-g_h(x))\,dx\Big| \\ &= \Big|\sum_{i=0}^{k-1}\int_{x_i}^{x_{i+1}} (f(x)-g_h(x))\,dx\Big| \\ &\le \sum_{i=0}^{k-1}\Big|\int_{x_i}^{x_{i+1}} (f(x)-g_h(x))\,dx\Big|.\end{aligned}$$

Posons $\alpha = x_i$, $\beta = x_{i+1}$ et $I_h(f) = (\beta-\alpha)f(\alpha)$. On a

$$\int_{x_i}^{x_{i+1}} g_h(x)\,dx = I_h(f)$$

qui est exact pour des polynômes de degré 0. Appliquons le théorème 5.1.1 avec $m = 1$, $w_1 = 1$, $\theta_1 = 0$ et $n = 0$. Nous obtenons

$$\begin{aligned}\Big|\int_{x_i}^{x_{i+1}} (f(x)-g_h(x))\,dx\Big| &\le (x_{i+1}-x_i)\int_{x_i}^{x_{i+1}} |f'(x)|\,dx \\ &\le h\int_{x_i}^{x_{i+1}} |f'(x)|\,dx.\end{aligned}$$

D'où le résultat.
(ii) Pour I_h^2, on prend $g_h(x_i^-) = f(x_i)$, $0 \le i \le k$. On peut alors appliquer le théorème 5.1.1 avec $\alpha = x_i$, $\beta = x_{i+1}$, $I_h(f) = (\beta-\alpha)f(\beta)$, $m = 1$, $w_1 = 1$, $\theta_1 = 1$ et $n = 0$.
(iii) Pour I_h^0, on prend

$$\begin{cases} g_h(x_i^-) = f(x_i), & 0 \le i \le k, \\ g_{h|[x_i,x_{i+1}]} \text{ est un polynôme de degré 1} & 0 \le i \le k-1. \end{cases}$$

On a alors

$$I_h^0(f) = \int_a^b g_h(x)\,dx.$$

On peut donc utiliser le théorème 5.1.1 avec $a = x_i$, $b = x_{i+1}$, $m = 1$, $w_1 = \frac{1}{2}$, $\theta_1 = \frac{1}{2}$ et $n = 1$. On obtient

$$\begin{aligned}\Big|\int_{x_i}^{x_{i+1}} (f(x)-g_h(x))\,dx\Big| &\le (x_{i+1}-x_i)^2 \times \frac{3}{2}\int_{x_i}^{x_{i+1}} |f''(x)|\,dx \\ &\le \frac{3}{2}h^2\int_{x_i}^{x_{i+1}} |f''(x)|\,dx.\end{aligned}$$

D'où le résultat. □

5.2.2 Formule des trapèzes

Sur $[x_i, x_{i+1}]$ on remplace f par le polynôme de degré 1 qui interpole f aux points x_i et x_{i+1}. Donc :

$$\int_{x_i}^{x_{i+1}} f(x)\,dx \approx \int_{x_i}^{x_{i+1}} p_1(x)\,dx = \frac{x_{i+1}-x_i}{2}(f(x_i)+f(x_{i+1})).$$

On en déduit

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{k-1} \frac{x_{i+1}-x_i}{2}(f(x_i)+f(x_{i+1})).$$

Théorème 5.2.2 *On suppose que $f \in C^2[a,b]$ et on note*

$$I_h(f) = \sum_{i=0}^{k-1} \frac{x_{i+1}-x_i}{2}(f(x_i)+f(x_{i+1})).$$

Alors on a

$$\left| \int_a^b f(x)\,dx - I_h(f) \right| \leq \frac{3}{2}h^2 \int_a^b |f''(x)|\,dx.$$

Démonstration. On prend la même fonction g_h que pour la formule du point milieu. Ainsi, on a

$$\int_{x_i}^{x_{i+1}} g_h(x)\,dx = \frac{x_{i+1}-x_i}{2}(f(x_i)+f(x_{i+1})).$$

Cette formule est exacte pour des polynômes de degré 1. On applique alors le théorème 5.1.1 avec $m=2$, $\alpha = x_i$, $\beta = x_{i+1}$, $w_1 = w_2 = \frac{1}{2}$, $\theta_1 = 0$, $\theta_2 = 1$ et $n = 1$. On obtient

$$\begin{aligned}\left| \int_{x_i}^{x_{i+1}} (f(x) - g_h(x))\,dx \right| &\leq \frac{3}{2}(x_{i+1}-x_i)^2 \int_{x_i}^{x_{i+1}} |f''(x)|\,dx \\ &\leq \frac{3}{2}h^2 \int_{x_i}^{x_{i+1}} |f''(x)|\,dx.\end{aligned}$$

D'où le résultat. □

5.2.3 Formule de Simpson

Cette formule est obtenue en faisant une moyenne pondérée de la formule des trapèzes avec celle des rectangles.

$$\int_a^b f(x)\,dx \approx \frac{1}{3}\sum_{i=0}^{k-1} \left(\frac{f(x_i)+f(x_{i+1})}{2} + 2f\left(\frac{x_i+x_{i+1}}{2}\right)\right)(x_{i+1}-x_i).$$

Si la subdivision $(x_i)_{i=0}^k$ est *uniforme, i.e.*, $x_{i+1} - x_i = h$ pour tout $i = 0, 1, \ldots, k-1$ on a la formule :

$$\int_a^b f(x)\,dx \approx \frac{h}{6}\left(f(x_0) + 4\sum_{i=0}^{k-1} f\left(\frac{x_i+x_{i+1}}{2}\right) + 2\sum_{i=1}^{k-1} f(x_i) + f(x_k)\right),$$

qui est connue sous le nom de la *formule de Simpson*.

Théorème 5.2.3 *On suppose que $f \in C^4[a,b]$ et on note*

$$I_h(f) = \frac{1}{3}\sum_{i=0}^{k-1}\left(\frac{f(x_i)+f(x_{i+1})}{2} + 2f\left(\frac{x_i+x_{i+1}}{2}\right)\right)(x_{i+1}-x_i).$$

Alors

$$\left|\int_a^b f(x)\,dx - I_h(f)\right| \leq \frac{5}{24}h^4\int_a^b |f^{(4)}(x)|\,dx.$$

Démonstration. On utilise la même technique que pour les théorèmes 5.2.1 et 5.2.2 en notant que sur $[x_i, x_{i+1}]$ cette formule est exacte pour les polynômes de degré ≤ 3. □

6

Résolution d'équations algébriques non-linéaires

6.1 Généralités

Soit $f : \mathbb{R}^n \to \mathbb{R}^n$ une application. On s'intéresse à la résolution de l'équation algébrique

$$f(x) = 0. \tag{6.1}$$

L'équation (6.1) est en fait un système d'équations algébriques non-linéaires. Pour résoudre le système (6.1), on a généralement recours à des méthodes itératives. On se donne donc une première approximation $x^{(0)}$ d'une solution de l'équation (6.1) et on construit une suite d'itérés $x^{(1)}, x^{(2)}, \ldots, x^{(k)}, \ldots$ qui converge vers un $\bar{x} \in \mathbb{R}^n$ vérifiant $f(\bar{x}) = 0$.
Désignons par $\|\cdot\|$ la norme euclidienne sur $\mathbb{R}^n$, *i.e.*

$$\|x\| := \Big(\sum_{i=1}^{n} x_i^2\Big)^{\frac{1}{2}}, \quad x \in \mathbb{R}^n.$$

Définition 6.1.1 *Supposons la méthode itérative convergente*, i.e., *qu'il existe $\bar{x} \in \mathbb{R}^n$ tel que*

$$\lim_{k\to\infty} \|x^{(k)} - \bar{x}\| = 0.$$

On dira que la méthode itérative est d'ordre p *($p \geq 1$) s'il existe une constante C, indépendante de k, tell que*

$$\|\bar{x} - x^{(k+1)}\| \leq C\,\|\bar{x} - x^{(k)}\|^p$$

(i) Si $p = 1$, on parle de convergence linéaire. *Dans ce cas, on a convergence si $C < 1$.*

(ii) Si $p = 1$, $C = C_k$, $\lim_{k\to\infty} C_k = 0$, on parle de convergence super-linéaire.

(iii) Si $p = 2$, on parle de convergence quadratique.

Notons que si une méthode itérative a une convergence quadratique alors cette convergence est super-linéaire et si une méthode a une convergence super-linéaire alors elle est nécessairement linéaire.
Une autre notion est nécessaire pour caractériser la convergence d'une méthode itérative.

Définition 6.1.2

(i) On dit que la convergence est globale *si pour tout choix de $x^{(0)} \in \mathbb{R}^n$, on a*

$$\lim_{k\to\infty} \|x^{(k)} - \bar{x}\| = 0. \tag{6.2}$$

(ii) On dit que la convergence est locale *s'il existe $R > 0$ tel que pour tout choix de $x^{(0)} \in \mathbb{R}^n$ avec $\|x^{(0)} - \bar{x}\| < R$, on a* (6.2).

Nous examinons maintenant quelques méthodes itératives.

6.2 Points fixes

Soit $f : \mathbb{R}^n \to \mathbb{R}^n$ et soit $\overline{x}$ tel que $f(\overline{x}) = 0$. On veut transformer ce problème en recherche de $\overline{x}$ tel que $g(\overline{x}) = \overline{x}$. Pour cela, on peut par exemple choisir

$$g(x) = f(x) - x,$$

ou

$$g(x) = x - \lambda f(x) \qquad \lambda \neq 0.$$

Définition 6.2.1 *On dit que $\overline{x}$ est un* point fixe *de g si on a $g(\overline{x}) = \overline{x}$.*

Cette formulation permet de construire la méthode itérative suivante :

$$\begin{cases} x^{(0)} \in \mathbb{R}^n & \text{donné,} \\ x^{(k+1)} = g(x^k), & k = 0, 1, \ldots \end{cases} \tag{6.3}$$

Supposons la fonction g continue. Alors si

$$\lim_{k\to\infty} \|x - \overline{x}\| = 0,$$

on obtient $g(\overline{x}) = \overline{x}$. D'où $f(\overline{x}) = 0$.
Donnons maintenant une condition nécessaire et suffisante pour la convergence de cette méthode :

Théorème 6.2.1 *Soit $A \subset \mathbb{R}^n \to \mathbb{R}^n$ où A est un ensemble fermé de $\mathbb{R}^n$. On suppose que :*

1. *Pour tout $x \in A$, on a $g(x) \in A$,*
2. *La fonction g est strictement contractante,* i.e. *il existe une constante $0 \leq C < 1$ telle que*

$$\|g(x) - g(y)\| \leq C \|x - y\| \qquad \textit{pour tous } x, y \in A.$$

Alors la fonction g admet un point fixe unique. De plus, la méthode itérative (6.3) *converge.*

Démonstration. Montrons d'abord l'unicité : On suppose pour cela l'existence de deux points fixes x et y. On a

$$\|x - y\| = \|g(x) - g(y)\| \leq C \|x - y\|.$$

Ceci est impossible car $C < 1$. D'où l'unicité du point fixe.
Considérons maintenant la suite de points $(x^{(k)})$ construite par la méthode itérative (6.3), *i.e.* telle que $x^{(k+1)} = g(x^k)$. On a

$$\begin{aligned} \|x^{(k+1)} - x^{(k)}\| &= \|g(x^{(k)}) - g(x^{(k-1)})\| \\ &\leq C \|x^{(k)} - x^{(k-1)}\| \\ &= C^k \|x^{(1)} - x^{(0)}\| \end{aligned}$$

Puisque $C < 1$, on obtient

$$\lim_{k \to \infty} \|x^{(k+1)} - x^{(k)}\| = 0.$$

Ceci implique

$$\lim_{k,\ell \to \infty} \|x^{(k)} - x^{(\ell)}\| = 0.$$

Ainsi $(x^{(k)}$ est une suite de Cauchy de $\mathbb{R}^n$. Elle converge donc vers un $\overline{x} \in \mathbb{R}^n$. On a donc, par continuité de g, $g(\overline{x}) = \overline{x}$. Nous avons ainsi montré l'existence du point fixe et la convergence de la méthode itérative (6.3) vers $\overline{x}$. □

6.3 Cas d'une équation non-linéaire

On s'intéresse maintenant au cas d'une seule équation ($n = 1$), *i.e.* où $f : \mathbb{R} \to \mathbb{R}$.

6.3.1 Méthode de la bisection ou dichotomie

Supposons disposer de deux approximations a et b d'une solution $\bar{x}$ de l'équation $f(x) = 0$. Supposons, en outre, que $f(a)\, f(b) < 0$, *i.e.* f change de signe entre a et b et posons $x^{(0)} = (a + b)/2$. L'algorithme est le suivant :

Pour $k = 0, 1, 2, \ldots$
- si $f(x^{(k)}) = 0$, stop $\quad (\bar{x} = x^{(k)})$
- si $f(a)\, f(x^{(k)}) < 0$, $\quad b := x^{(k)}$
- sinon $\quad a := x^{(k)}$
- $x^{(k+1)} := (a + b)/2$

Fin k

On appelle cet algorithme méthode de la bisection ou dichotomie.

Théorème 6.3.1 *Soit $a, b \in \mathbb{R}$ tels que $f(a)\, f(b) < 0$. Alors la méthode de la bisection converge. De plus, la convergence est linéaire.*

6.3.2 Méthode Regula Falsi ou "fausse position"

Au lieu de prendre le milieu de l'intervalle $[a, b]$ comme dans la méthode de bisection, on définit un point c par la relation

$$\frac{f(b)}{b-c} = \frac{f(a)}{a-c}.$$

D'où

$$c = \frac{f(a)b - f(b)a}{f(a) - f(b)}.$$

Théorème 6.3.2 *Soit $a, b \in \mathbb{R}$ tels que $f(a)\, f(b) < 0$. Alors la méthode de Regula Falsi. De plus, la convergence est super-linéaire.*

Notons que bien que cette méthode ait la même vitesse de convergence que la méthode de dichotomie, elle converge, en pratique plus vite.

6.3.3 Méthode de Newton

On suppose que la fonction $f : \mathbb{R} \to \mathbb{R}$ est deux fois dérivable et qu'il au moins un point x tel que $f(x) = 0$. La méthode de Newton est définie par les itérations :

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}, \qquad k = 0, 1, \ldots$$

En d'autres termes, pour k donné, $x^{(k)}$ est l'abscisse du point d'intersection de la tangente au point $(x^{(k)}, f(x^{(k)}))$ avec l'axe Ox.

Théorème 6.3.3 *On suppose que $x^{(0)}$ est choisi « assez proche » de $\bar{x}$ et que $f'(x) \neq 0$ pour x dans un voisinage de $\bar{x}$. Alors la méthode de Newton converge. De plus, la convergence est* quadratique.

Le théorème précédent montre que la convergence de la méthode de Newton est locale.

Remarque 6.3.1 *Il est possible d'obtenir une variante de cette méthode en considérant l'approximation de la dérivée par un quotient différentiel. On définit ainsi*

$$f'(x) \approx \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}.$$

On obtient la méthode itérative :

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f(x^{(k)}) - f(x^{(k-1)})}\,(x^{(k)} - x^{(k-1)}),$$

appelée méthode de la sécante. Notons que cette méthode a quelque analogie avec la méthode Regula Falsi.

Nous avons le résultat suivant :

Théorème 6.3.4 *On suppose que f est de classe C^0. Alors la méthode* Regula Falsi *converge. De plus, la convergence est linéaire.*

6.4 Cas d'un système d'équations

On considère maintenant d'un système de n équations algébriques. Considérons donc la cas d'une fonction $f : \mathbb{R}^n \to \mathbb{R}^n$. On se limitera ici à la description de la méthode de Newton. Celle-ci peut se généraliser de la manière suivante :
L'équation algébrique $f(x) = 0$ peut s'écrire sous la forme :

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0, \\ f_2(x_1, \ldots, x_n) &= 0, \\ &\vdots \\ f_n(x_1, \ldots, x_n) &= 0. \end{aligned}$$

On écrit la méthode de Newton pour une fonction $f_i : \mathbb{R}^n \to \mathbb{R}$:

$$\begin{aligned} &\frac{\partial f_i}{\partial x_j}(x^{(k)})\, \delta x^{(k)} = -f_i(x^{(k)}), \\ &x^{(k+1)} = x^{(k)} + \delta x^{(k)}, \qquad k = 0, 1, \ldots \end{aligned}$$

La méthode fait ainsi intervenir la matrice $\nabla f(x)$ définie par

$$(\nabla f(x))_{ij} = \frac{\partial f_i}{\partial x_j}(x) \qquad 1 \leq i, j \leq n.$$

On peut donc écrire la méthode de Newton pour chercher le zéro d'une fonction f sous la forme suivante :

$$\begin{cases} x^{(0)} \in \mathbb{R}^n \quad \text{donné}, \\ (\nabla f(x^{(k)}))\, \delta x^{(k)} = -f(x^{(k)}), \\ x^{(k+1)} := x^{(k)} + \delta x^{(k)}. \end{cases}$$

On doit donc résoudre, à chaque itération k, un système linéaire faisant intervenir une matrice dépendant de l'itération k.

Théorème 6.4.1 *Soit $f : \mathbb{R}^n \to \mathbb{R}^n$ et soit $\bar{x} \in \mathbb{R}^n$ tel que*

$$f(\bar{x}) = 0.$$

On suppose que la matrice $\nabla f(\bar{x})$ est inversible. Alors, il existe un réel $r > 0$ tel que si $\|\bar{x} - x^{(0)}\| < r$ alors la méthode de Newton converge vers $\bar{x}$. De plus, on a

$$\|\bar{x} - x^{(k)}\| \leq C \, \|\bar{x} - x^{(k-1)}\|^2 \qquad \forall\, k > 0.$$

Remarque 6.4.1 *Cet algorithme peut se révéler assez coûteux. Pour cela, on peut utiliser la méthode de* Newton modifiée *où la matrice $\nabla f(x^{(k)})$ est remplacée par la matrice $\nabla f(x^{(0)})$. Dans ce cas, la convergence est seulement* linéaire.

7

Schémas numériques pour les équations différentielles

7.1 Introduction

On se donne une fonction continue

$$f : \mathbb{R}^m \times [0,T] \to \mathbb{R}^m$$

et on cherche une fonction $y : t \in [0,T] \mapsto y(t) \in \mathbb{R}^m$ de classe $\mathcal{C}^1$ telle que

$$\begin{cases} y'(t) = f(y(t),t) & \text{pour tout } t \in]0,T], \\ y(t_0) = g, \end{cases} \tag{7.1}$$

où $(g,t_0) \in \mathbb{R} \times [0,T]$ est donné. Dans les applications, la variable t désigne souvent le temps et le problème ci-dessus décrit l'évolution d'un système (physique, mécanique, ...) au cours du temps. Nous adopterons ainsi le choix $t_0 = 0$.
On dit que l'équation (7.1) est du *premier ordre* parce qu'elle fait intervenir la première dérivée de la fonction inconnue. Notons qu'en fait l'équation (7.1) est un système différentiel qui peut s'écrire sous la forme :

$$\begin{aligned} &y_1'(t) = f_1(y_1(t),\dots,y_m(t),t), \\ &y_2'(t) = f_2(y_1(t),\dots,y_m(t),t), \\ &\dots, \\ &y_m'(t) = f_m(y_1(t),\dots,y_m(t),t). \end{aligned} \tag{7.2}$$

Ci-dessus, f_i désigne une fonction $f_i : \mathbb{R}^m \times [0,T] \to \mathbb{R}$.

Théorème 7.1.1 *On suppose qu'il existe une constante L telle que*

$$|f(y,t) - f(x,t)| \leq L\,|y-x| \qquad \textit{pour tout } x,y \in \mathbb{R}^m, \textit{ pour tout } t > 0.$$

Alors, le problème (7.1) *admet une solution unique.*

Dans le théorème précédent, $|\cdot|$ désigne une norme sur $\mathbb{R}^m$.
Exemple. Considérons l'exemple du système différentiel linéaire :

$$\begin{cases} y'(t) = A(t)\, y(t) & 0 < t \leq T, \\ y(t_0) = g, \end{cases}$$

où A est une matrice carrée d'ordre m dépendant continûment de t, et $g \in \mathbb{R}^m$. Clairement la fonction f est donnée par $f(y,t) = A\, y$. On montre ainsi aisément que cette fonction vérifie l'hypothèse du théorème 7.1.1.

Remarque 7.1.1 *Pour résoudre le problème* (7.1)*, on peut écrire*

$$y(t) = g + \int_0^t f(y(s), s)\, ds \qquad t \in]0, T]$$

et ramener le problème de l'approximation numérique à un problème d'intégration numérique.

Dans tout ce qui suit, on suppose que f satisfait les hypothèses du théorème 7.1.1. De plus, on considérera le cas d'une seule équation différentielle ($m = 1$).
Une méthode numérique consiste à choisir des points

$$0 = t_0 < t_1 < t_2 < \cdots < t_n < \ldots$$

dans l'intervalle $[0, T]$ et à se donner un schéma numérique permettant de calculer des valeurs $y_n \in \mathbb{R}$ qui sont des approximations des vecteurs $y(t_n)$, $n = 1, 2, \ldots$
Pour simplifier la présentation, on se donne une subdivision uniforme de l'intervalle $[0, T]$:

$$t_n = nh \text{ où } h = \frac{T}{N} \quad \text{pour tout } n = 0, 1, \ldots, N.$$

7.2 Méthodes d'Euler

Pour approcher la dérivée $y'(t)$ en $t = t_n$, nous posons

$$y'(t_n) \approx \frac{y_{n+1} - y_n}{t_{n+1} - t_n} = \frac{y_{n+1} - y_n}{h},$$

ou

$$y'(t_n) \approx \frac{y_n - y_{n-1}}{h}.$$

On est ainsi conduit aux deux schémas suivants :
- *Schéma d'Euler progressif* (*forward*) :

$$\begin{cases} \dfrac{y_{n+1} - y_n}{h} = f(y_n, t_n), \quad n = 0, 1, \ldots, N-1, \\ y^0 = g. \end{cases}$$

- *Schéma d'Euler rétrograde* (*backward*) :

$$\begin{cases} \dfrac{y_{n+1} - y_n}{h} = f(y_{n+1}, t_{n+1}), \quad n = 0, 1, \ldots, N-1, \\ y^0 = g. \end{cases}$$

Notons que le premier schéma est *explicite, i.e.*, il permet d'obtenir directement y_{n+1} à partir de y_n. Le deuxième schéma est dit *implicite*; il se ramène à la résolution d'un problème non-linéaire pour chaque n. Ceci nécessitera de choisir une méthode numérique pour résoudre une équation algébrique non-linéaire pour chaque n.

Définition 7.2.1 *On dit que les schémas d'Euler progressif et rétrograde sont des schémas à un pas puisqu'ils ne font intervenir les solutions approchées qu'aux points t_n et t_{n+1}. De plus, on appelle h le pas de temps.*

7.2.1 Le schéma d'Euler progressif

Nous allons introduire une nouvelle notion : *la consistance*. Nous voulons mesurer avec quelle précision la solution exacte vérifie le schéma numérique. Nous appellerons *erreur de troncature* en $t = t_n$ la quantité

$$\varepsilon_h^n = \frac{y(t_{n+1}) - y(t_n)}{h} - f(y(t_n), t_n).$$

On dira qu'un schéma est consistant si

$$\lim_{h\to 0} \max_{0\le n\le T/h} |\varepsilon_h^n| = 0.$$

Dans ce qui suit, on notera par e_n l'erreur en $t = t_n$, *i.e.* $e_n := y(t_n) - y_n$.

Lemme 7.2.1 *On a l'inégalité*

$$|e_{n+1}| \le (1 + Lh)|e_n| + |\varepsilon_h^n|h.$$

Démonstration. On a

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + hf(y(t_n), t_n) + \varepsilon_h^n h, \\ y_{n+1} &= y_n + hf(y_n, t_n). \end{aligned}$$

D'où par soustraction :

$$e_{n+1} = e_n + \varepsilon_h^n h + h\,(f(y(t_n, t_n) - f(y_n, t_n)).$$

Donc

$$|e_{n+1}| \le |e_n| + |\varepsilon_h^n|h + Lh\,|y(t_n) - y_n|. \qquad \square$$

Lemme 7.2.2 *On suppose que la solution y est de classe $\mathcal{C}^2$ (ou que f est de classe $\mathcal{C}^1$ en la première variable), alors on a*

$$|\varepsilon_h^n| \le \frac{h}{2} \sup_{0\le t\le T} |y''(t)|.$$

Démonstration. La formule de Taylor donne :

$$y(t_{n+1}) = y(t_n) + h\, y'(t_n) + \frac{h^2}{2}\, y''(t_n + s),$$

où $s \in [0, h]$. D'où

$$\varepsilon_h^n = y'(t_n) + \frac{h}{2}\, y''(t_n + s) - f(y(t_n), t_n) = \frac{h}{2}\, y''(t_n + s). \qquad \square$$

Lemme 7.2.3 *Soient θ_n et α_n deux suites de réels positifs vérifiant*

$$\theta_{n+1} \leq (1 + Lh)\theta_n + \alpha_n \qquad 0 \leq n \leq N - 1.$$

Alors on a

$$\theta_n \leq e^{Lnh}\theta_0 + \sum_{j=0}^{n-1} e^{(n-j-1)Lh}\alpha_j \qquad 0 \leq n \leq N - 1. \tag{7.3}$$

Démonstration. On va montrer ce résultat par récurrence. Supposons que la relation (7.3) soit vraie pour $n \leq k$. Pour $n = 1$, on a par hypothèse :

$$\theta_1 \leq (1 + Lh)\theta_0 + \alpha_0.$$

Puisque $1 + x \leq e^x$ pour tout $x \in \mathbb{R}$, on a

$$\theta_1 \leq e^{Lh}\theta_0 + \alpha_0,$$

qui est bien l'inégalité voulue pour $n = 1$.
Supposons que

$$\theta_n \leq e^{Lnh}\theta_0 + \sum_{j=0}^{n-1} e^{(n-j-1)Lh}\alpha_j \qquad 0 \leq n \leq k,$$

et montrons cette inégalité pour $n = k + 1$. On a par hypothèse :

$$\begin{aligned}
\theta_{k+1} &\leq (1 + Lh)\theta_k + \alpha_k \\
&\leq e^{Lh}\theta_k + \alpha_k \\
&\leq e^{Lh}\left(e^{Lkh}\theta_0 + \sum_{j=0}^{k-1} e^{(k-j-1)Lh}\alpha_j\right) + \alpha_k \\
&= e^{(k+1)Lh}\theta_0 + \sum_{j=0}^{k-1} e^{(k-j)Lh}\alpha_j + \alpha_k \\
&= e^{(k+1)Lh}\theta_0 + \sum_{j=0}^{k} e^{(k-j)Lh}\alpha_j. \qquad \square
\end{aligned}$$

Les lemmes précédents permettent d'obtenir le résultat suivant :

Théorème 7.2.1 *On suppose que la solution $y(t)$ est de classe $\mathcal{C}^2$. Alors, il existe une constante C indépendante de h telle que*

$$|y(t_n) - y_n| \leq Ch \qquad n = 1, \ldots, N.$$

Démonstration. Par les lemmes 7.2.1 et 7.2.2, on obtient

$$|e_{n+1}| \leq (1+\beta)\,|e_n| + Ch^2 \sup_{0<t\leq T} |y''(t)|.$$

Par ailleurs, en utilisant le lemme 7.2.3 avec $\theta_n = |e_n|$, $\alpha_n = Ch^2 \sup_{0<t\leq T} |y''(t)|$ et $\beta = Lh$, et en notant que $\theta_0 = 0$,

$$\begin{aligned} |e_n| &\leq Ch^2 \sum_{j=1}^{n-1} e^{(n-j-1)Lh} \sup_{0\leq t\leq T} |y''(t)| \\ &\leq C'h \sup_{0\leq t\leq T} |y''(t)|. \end{aligned}$$

Ceci achève la démonstration. □

On dit que la méthode d'Euler progressive est d'ordre 1. En effet, nous avons montré que cette méthode est convergente puisque

$$\lim_{h\to 0} \max_{0\leq n\leq T/h} |y(t_n) - y_n| = 0.$$

7.2.2 Schéma d'Euler rétrograde

Rappelons que ce schéma s'écrit :

$$\begin{cases} \dfrac{y_{n+1} - y_n}{h} = f(y_{n+1}, t_{n+1}) \qquad 0 \leq n \leq N-1, \\ y_0 = g, \end{cases}$$

où y_n est une approximation de $y(t_n)$. Nous obtenons donc un problème non-linéaire à résoudre pour chaque n. On peut alors montrer que ce schéma est aussi d'ordre 1.

7.2.3 Schéma de Crank-Nicolson

Ce schéma est donné par :

$$\begin{cases} \dfrac{y_{n+1} - y_n}{h} = \dfrac{1}{2}(f(y_n, t_n) + f(y_{n+1}, t_{n+1})), \qquad 0 \leq n \leq N-1, \\ y_0 = g, \end{cases}$$

Ici aussi, on a à résoudre un problème non-linéaire pour chaque $n = 0, \ldots, N-1$.
On montre alors (Voir exercices) que ce schéma, dit de *Crank-Nicolson* est d'ordre 2, *i.e.*,

$$\max_{0\leq n\leq T/h} |y(t_n) - y_n| \leq Ch^2,$$

si la fonction y est de classe $\mathcal{C}^3$.

7.3 Autres schémas numériques

Nous présentons maintenant quelques schémas numériques d'ordre supérieur.

7.3.1 Méthodes basées sur la formule de Taylor

Supposons la fonction f assez régulière, on peut écrire :

$$\begin{aligned} y(t_{n+1}) &= y(t_n + h) \\ &= y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + \cdots + \frac{1}{k!}h^k y^{(k)}(t_n) + \frac{1}{(k+1)!}h^{k+1}y^{(k+1)}(\xi_n), \end{aligned}$$

où $\xi_n \in [t_n, t_{n+1}]$. On peut donc écrire pour h suffisamment petit :

$$y(t_{n+1}) \approx y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + \cdots + \frac{1}{k!}h^k y^{(k)}(t_n).$$

Pour $k = 1$, on retrouve le schéma d'Euler progressif car $y'(t_n) = f(y_n, t_n)$.
Pour $k = 2$, on obtient

$$y(t_{n+1}) \approx y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n).$$

On a $y'(t_n) = f(y(t_n), t_n)$. Pour évaluer $y''(t_n)$, on dérive l'équation différentielle :

$$y''(t) = \frac{\partial f}{\partial t}(y(t), t) + \frac{\partial f}{\partial y}(y(t), t)y'(t).$$

Puisque $y'(t_n)$ est approché par $f(y_n, t_n)$, on peut écrire :

$$y''(t_n) \approx \frac{\partial f}{\partial t}(y_n, t_n) + \frac{\partial f}{\partial y}(y_n, t_n)f(y_n, t_n).$$

D'où le schéma :

$$y_{n+1} = y_n + hf(y_n, t_n) + \frac{h^2}{2}\left(\frac{\partial f}{\partial t}(y_n, t_n) + \frac{\partial f}{\partial y}(y_n, t_n)f(y_n, t_n)\right). \tag{7.4}$$

Théorème 7.3.1 *On suppose que la fonction f est de classe $\mathcal{C}^3$ (en y et en t). Alors, le schéma* (7.4) *vérifie :*

$$\max_{1 \le n \le T/h} |y(t_n) - y_n| \le Ch^2.$$

On appelle alors ce schéma : *schéma de Taylor d'ordre 2*.

7.3.2 Méthodes de Runge-Kutta

Il s'agit de schémas numériques de résolution d'équations différentielles où chaque pas de temps est décomposé en sous pas. À titre d'exemple, nous donnons ici deux schémas. Le premier est un schéma de type Runge-Kutta d'ordre 2, appelé aussi schéma de *Heun*. Il est donné par :

$$\begin{aligned}\tilde{y}_{n+1} &= y_n + hf(y_n, t_n),\\ y_{n+1} &= y_n + \frac{h}{2}\big(f(y_n, t_n) + f(\tilde{y}_{n+1}, t_{n+1})\big).\end{aligned}$$

Le second schéma est la méthode de Runge-Kutta classique (dite *RK4*) s'écrit :

$$\begin{aligned}y_{n,1} &= y_n + \frac{h}{2}f(y_n, t_n),\\ y_{n,2} &= y_n + \frac{h}{2}f(y_{n,1}, t_n + \frac{h}{2}),\\ y_{n,3} &= y_n + hf(y_{n,2}, t_n + \frac{h}{2}),\\ y_{n+1} &= y_n + \frac{h}{6}\Big(f(y_n, t_n) + 2f(y_{n,1}, t_n + \frac{h}{2}) + 2f(y_{n,2}, t_n + \frac{h}{2}) + f(y_{n,3}, t_{n+1})\Big).\end{aligned}$$

On montre que cette méthode est d'ordre 4.

7.4 Systèmes différentiels d'ordre 1

On s'intéresse maintenant au cas des systèmes différentiels, *i.e.*, où $m > 1$. Le schéma d'Euler s'écrit naturellement :

$$\begin{cases} y_i^{n+1} = y_i^n + hf_i(y_1^n, \dots, y_m, t_n) & 1 \le i \le m,\ 0 \le n \le N-1,\\ y_i^0 = \boldsymbol{g}_i & 1 \le i \le m.\end{cases}$$

La résolution est donc explicite. De même le schéma d'Euler rétrograde s'écrit :

$$\begin{cases} y_i^{n+1} = y_i^n + hf_i(y_1^{n+1}, \dots, y_m^{n+1}, t_{n+1}) & 1 \le i \le m,\ 0 \le n \le N-1,\\ y_i^0 = g_i & 1 \le i \le m.\end{cases}$$

Ici le schéma est implicite. En effet, on résout, pour chaque n, un système d'équations algébriques non-linéaires.

7.5 Équations différentielles d'ordre 2

Considérons, dans le cas $m = 1$, l'équation différentielle du second ordre suivante :

$$\begin{cases} y''(t) = f(y(t), y'(t), t), & 0 < t \leq T, \\ y(0) = a, \\ y'(0) = b. \end{cases}$$

Pour résoudre numériquement ce problème, deux approches sont possibles :

1. On approche les dérivées première et seconde par des quotients différentiels.
2. On se ramène à un système différentiel du premier ordre. Pour cela, introduisons la nouvelle inconnue $z(t) = y'(t)$. Notre équation différentielle devient :

$$\begin{cases} z'(t) = f(y(t), z(t), t), & 0 < t \leq T, \\ y'(t) = z(t), & 0 < t \leq T, \\ y(0) = a, \\ z(0) = b. \end{cases}$$

On a ainsi transformé une équation différentielle du second ordre en un système de deux équations différentielles du premier ordre.