Sorbonne-Université, Faculté des Sciences et Ingénierie
Licence de mécanique, UE LU3ME005, 2019/2020

Méthodes numériques pour la mécanique
Cours $(20h)$

Bibliographie

G. H. Golub, G.A. Meurant : "Résolution numérique des grands systèmes linéaires", Edition Eyrolles, 1983.
P. Lascaux, R. Théodor : "Analyse numérique matricielle appliquée à l'art de l'ingénieur", tomes 1 et 2, Edition Masson, 1986.
R. Théodor : "Initiation à l'analyse numérique", Edition Masson, 1989.
J. P. Nougier : "Méthodes de calcul numérique", Edition Masson, 1989.
M. Crouzeix, A.L. Mignot : "Analyse numérique des équations différentielles", Edition Masson, 1989.
P. G. Ciarlet : "Introduction à l'analyse numérique matricielle et à l'optimisation", Edition Masson, 1990.
J. P. Demailly : "Analyse numérique et équations différentielles", Edition Presses Universitaires de Grenoble, 1991.
GH Golub, CF Van Loan, "Matrix Computation", 3rd Edition, books.google.com, 1996
F. Jedrzejewski : "Introduction aux méthodes numériques", Edition Springer, 2001.
Y. Saad, "Iterative Methods for Sparce Linear Systems", 2nd edition, SIAM , 2003

1 Racines d'équations

1.1 Problème :

Soit une fonction réelle d'une variable $x, x \in \mathbb{I}$ un intervalle de $\mathbb{R}$. On suppose que f est C^2 : elle admet une dérivée seconde continue dans $\mathbb{I}$ (souvent f est C^∞). On cherche les racines de l'équation $f(x) = 0$.

Exemple :
Si $f(x)$ est un polynôme

1. il est facile de trouver les racines pour les degrés 1 et 2.
2. pour les degrés 3 et 4, on doit utiliser des simplifications et des formules de transformation.
3. pour les polynômes de $d^o \geq 5$, il n'existe plus de solution sous forme algébrique simple.

1.2 Localisation grossière des racines

En général, on cherche des solutions de f sur $\mathbb{I}$ avec une approximation donnée à l'avance. On décompose $\mathbb{I}$ en intervalles partiels dans lesquels on sait qu'il y a soit 0, soit 1 racine.

1. Si f est monotone et ne change pas de signe $\rightarrow$ pas de racine.
2. Si f prend aux extrémités de cet intervalle des valeurs de signes opposés, il y a alors 1 racine.

Exemple :

$$f(x) = \frac{1}{x-1} + \frac{2}{x-3} + \frac{3}{x-5} - 1, \qquad f'(x) = \frac{-1}{(x-1)^2} - \frac{2}{(x-3)^2} - \frac{3}{(x-5)^2} < 0$$

Donc f est décroissante dans chacun des intervalles $]1,3[,]3,5[,]5,\infty[$.

$f(1+\epsilon) = +\infty \qquad f(3-\epsilon) = -\infty \quad \rightarrow 1$ racine dans $]1,3[$
$f(3+\epsilon) = +\infty \qquad f(5-\epsilon) = -\infty \quad \rightarrow 1$ racine dans $]3,5[$
$f(5+\epsilon) = +\infty \qquad f(\infty) = -1 \qquad \rightarrow 1$ racine dans $]5,\infty[$

L'étude de la fonction est nécessaire pour localiser les sous-intervalles où se situe chaque racine. Les racines sont proches de $4/3, 11/3, 10$.

1.3 Méthode de la bissection ou dichotomie

Supposons que $\mathbb{I} =]a,b[$, que f est strictement monotone dans $\mathbb{I}$, et que $f(a)\, f(b) < 0$. Il existe alors 1 racine unique $r \in]a,b[$ de l'équation $f(x) = 0$.
Par exemple, soit $f(a) < 0$ et $f(b) > 0$, $f(x)$ croissante dans $\mathbb{I}$.
On pose $x_0 = a$, $x_1 = b$, et $x_2 = (a+b)/2$.
Si $f(x_2) < 0$, alors $r \in]x_2, b[$. On pose alors $x_0 = x_2$, et on recoupe l'intervalle en 2.
Si $f(x_2) > 0$, alors $r \in]a, x_2[$. On pose alors $x_1 = x_2$, et on recoupe l'intervalle en 2.
Ainsi la longueur de l'intervalle où se trouve r est réduite de moitié : $(b-a)/2$. En répétant cette opération n fois, on a un intervalle de longueur $(b-a)/2^n \rightarrow 0 (n \rightarrow \infty)$ permettant d'avoir r avec l'approximation voulue.

1.4 Méthode de la sécante

Supposons $\mathbb{I} = [a, b]$, $f'(x) \neq 0, \forall\, x \in]a, b[$ et $f(a)\, f(b) < 0$. On définit les points A $(a, f(a))$ et $B(b, f(b))$. Pour obtenir une valeur approchée de la racine $r \in]a, b[$ de l'équation $f(x) = 0$, on trace la droite AB qui coupe l'axe des x en un point plus proche de r que ne le sont a et b.
On pose à nouveau $x_0 = a$, $x_1 = b$, et on note x_2 l'intersection de la sécante AB et de l'axe des x. On a :
$(f(x_1) - f(x_0))/(x_1 - x_0) = (f(x_1) - 0)/(x_1 - x_2)$,
et donc $x_2 = x_1 - f(x_1)\,(x_1 - x_0)/[f(x_1) - f(x_0)]$. On construit alors la suite

$$x_0 = a, \qquad x_1 = b, \qquad x_{n+1} = x_n - f(x_n)\,(x_n - x_{n-1})/[f(x_n) - f(x_{n-1})], \quad n = 1, 2, ...$$

Le critère d'arrêt des itérations est $|x_{n+1} - x_n| < \epsilon$ où ϵ est donné (petit). Par exemple, $\epsilon = 10^{-10}$. Si $f'(x)$ et $f''(x)$ ne changent pas de signe dans l'intervalle $]a, b[$, c'est-à-dire $f(x)$ monotone et sans point d'inflexion dans $]a, b[$, alors la suite x_n converge vers la racine r.

1.5 Méthode du point fixe

1.5.1 Principe de la méthode

L'équation $f(x) = 0$ dont on cherche les racines dans $[a, b] \subset \mathbb{R}$ est mise sous la forme $\phi(x) - x = 0$, avec par exemple $\phi(x) = x + f(x)$. Les racines sont alors les points fixes $\in [a, b]$ de la fonction $\phi(x)$.
Pour tout point initial $x_0 \in [a, b]$, on construit la suite itérée x_n définie par $x_{n+1} = \phi(x_n)$, $n \geq 0$. Le critère d'arrêt des itérations est $|x_{n+1} - x_n| < \epsilon$ où ϵ est donné (petit).

1.5.2 Convergence de la méthode

Déf. 1 : Soit $\phi : E \to E$ une application continue. On dit que $a \in E$ est un point fixe de ϕ si $\phi(a) = a$.
Déf. 2 : L'application ϕ est k lipschitzienne si $\forall\, x, y \in E$, $|\phi(x) - \phi(y)| \leq k\,|x - y|$.
Déf. 3 : L'application ϕ est dite strictement contractante si elle est lipschitzienne de rapport $k < 1$.

Théorème 1 : Si $\phi([a, b]) \subset [a, b]$ et si ϕ est continue sur $[a, b]$, alors ϕ possède au moins un point fixe dans $[a, b] \Longleftrightarrow \phi(x) - x = 0$ possède au moins une racine.
Preuve : $\phi([a, b]) \subset [a, b] \Longrightarrow a < \phi(a) < b$ et $a < \phi(b) < b \Longrightarrow \phi(a) - a > 0$ et $\phi(b) - b < 0$.
D'après le théorème des valeurs intermédiaires, $\exists\, \xi \in [a, b]$ tel que $\phi(\xi) - \xi = 0 \Longrightarrow \phi(\xi) = \xi$.

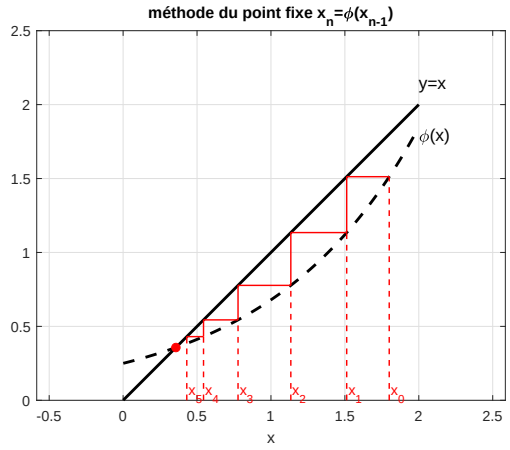
Théorème 2 : Si $\phi([a, b]) \subset [a, b]$ et si $|\phi'(x)| \leq k < 1\,, \forall\, x \in [a, b]$, alors il existe un seul point fixe pour ϕ dans $[a, b]$.
Preuve : Soient $x_1, x_2 \in [a, b]$ tel que $\phi(x_1) = x_1$ et $\phi(x_2) = x_2$.
$\Longrightarrow |x_2 - x_1| = |\phi(x_2) - \phi(x_1)| = |\phi'(\xi)|\,|x_2 - x_1| \leq k\,|x_2 - x_1| < |x_2 - x_1|$.
Ce qui est absurde.

Théorème 3 : Si $\phi\,([a, b]) \subset [a, b]$ et ϕ strictement contractante sur $[a, b]$, alors la suite $x_{n+1} = \phi(x_n)$ converge vers $r \in [a, b]$, $\forall\, x_0 \in [a, b]$ et on a :

$$|x_n - r| < \frac{k^n}{(1 - k)} |x_1 - x_0|$$

où r est le point fixe de ϕ.
Preuve : On a $x_n = \phi(x_{n-1})$ et $r = \phi(r)$ existe et est unique (th. 2).
$\Longrightarrow x_n - r = \phi(x_{n-1}) - \phi(r) = \phi'(\xi)\,(x_{n-1} - r) \quad \xi \in [a, b]$.
$$\Longrightarrow \begin{cases} |x_n - r| & \leq k\,|x_{n-1} - r| \\ & \leq k^2\,|x_{n-2} - r| ..... \leq k^n\,|x_0 - r| \to 0_{n\to\infty} \end{cases}$$
Par ailleurs : $|x_{n+1} - x_n| = |\phi(x_n) - \phi(x_{n-1})| = |\phi'(\xi_n)|\,|x_n - x_{n-1}| \leq k\,|x_n - x_{n-1}|$
$\Longrightarrow |x_{n+1} - x_n| \leq k\,|x_n - x_{n-1}| \leq ..... \leq k^n\,|x_1 - x_0|$
et $|x_{n+2} - x_{n+1}| \leq k^{n+1}\,|x_1 - x_0|$
et $|x_{n+p} - x_{n+p-1}| \leq k^{n+p-1}\,|x_1 - x_0|$
$$\Longrightarrow \begin{cases} |x_{n+p} - x_n| & \leq |x_{n+p} - x_{n+p-1}| + ...|x_{n+1} - x_n| \\ & \leq (k^{n+p-1} + k^{n+p-2} + ...k^n)\,|x_1 - x_0| \\ & = k^n\,(1 + k + k^2 + ...k^{p-1})|x_1 - x_0| \\ & = k^n \frac{(1-k^p)}{(1-k)}|x_1 - x_0| \end{cases}$$
donc pour $p \to \infty$: $x_{n+p} \to r \Longrightarrow |x_n - r| \leq \frac{k^n}{(1-k)}\,|x_1 - x_0|$, car $k < 1$

Théorème 4 : Si $\phi'(x)$ est continue dans un intervalle ouvert contenant un point fixe r de ϕ et si $|\phi'(r)| < 1$, alors il existe $\epsilon > 0$ tel que la suite $x_{n+1} = \phi(x_n)$ soit convergente (vers r) pour $|x_0 - r| < \epsilon$.
Preuve : La continuité de ϕ' et $|\phi'(r)| < 1$ entrainent que : $\exists\ \epsilon > 0$ tel que $|x - r| \leq \epsilon \Longrightarrow |\phi'(x)| \leq k < 1$.
Alors, $\forall x \in \mathbb{I}_\epsilon = [r - \epsilon, r + \epsilon]$, $\exists\, \eta \in]x, r[$ ou $]r, x[$ tel que $|\phi(x) - r| = |\phi(x) - \phi(r)| = \phi'(\eta)\,|x - r| \leq k\,\epsilon < \epsilon \Longrightarrow \phi(\mathbb{I}_\epsilon) \subset \mathbb{I}_\epsilon \Longrightarrow$ d'après le théorème 3, il y a convergence.



1.5.3 Classification des point fixes

On suppose $\phi : \mathbb{I} \to \mathbb{I}$, I fermé $\subset \mathbb{R}$, et ϕ de classe C^1.

a) $|\phi'(r)| < 1$: r est un point fixe attractif. La méthode du point fixe converge dans un voisinage de r (théorème 4).

b) $|\phi'(r)| > 1$: r est un point fixe répulsif. La méthode du point fixe ne converge pas (ou pas vers r) dans un voisinage de r. On considère alors l'application réciproque ϕ^{-1} de ϕ. Comme on a $\phi(r) = r$, alors $\phi^{-1}(r) = r$.
Comme $(\phi^{-1})'(r) = 1/\phi'(\phi^{-1}(r)) = 1/\phi'(r)$, ϕ^{-1} est contractante, et r est un point fixe attractif pour ϕ^{-1}. Donc, il faut employer ϕ^{-1} pour construire la suite.

c) $|\phi'(r)| = 1$: cas indéfini (répulsif ou attractif).

1.5.4 Application de la méthode, choix de ϕ

En pratique, choisit $\phi(x)$ de façon à avoir un point fixe attractif, et une convergence la plus rapide possible.
On peut "relaxer" une méthode afin d'accélér sa convergence en choisissant un paramètre θ en utilisant $\phi_\theta(x) = (1-\theta)x + \theta\phi(x)$. On cherche alors la valeur optimale de θ en adoptant l'une des deux stratégies suivantes :

a) On choisit une valeur du critère de convergence ϵ relativement "grande" ($\epsilon = 10^{-2}$ par exemple) et on cherche la valeur de θ pour laquelle la convergence s'effectue en un minimum d'itérations (avec la même condition initiale x_0).
b) On choisit un nombre maximal d'itérations relativement "petit" ($n_{max} = 100$ par exemple) et on cherche la valeur de θ pour laquelle l'écart $|x_{n+1} - x_n|$ obtenu à l'issue des n_{max} itérations est minimum (avec la même condition initiale x_0).

On peut également prendre : $\phi(x) = x + \Phi(x)\, f(x)$ (voir TD).

1.6 Méthode de Newton

1.6.1 Construction de la méthode

Connaissant une valeur x_0 au voisinage d'une racine, on construit la tangente en x_0 à la courbe $x = f(x)$: $y - f(x_0) = f'(x_0)\,(x - x_0)$. L'intersection de cette tangente avec l'axe des x nous donne x_1 qui s'obtient en remplacant y par 0, soit : $x_1 = x_0 - f(x_0)/f'(x_0)$ avec $f'(x_0) \neq 0$. On répète l'opération en remplacant x_0 par x_1.
A la $(k+1)^{eme}$ itération, on a : $x_{k+1} = x_k - f(x_k)/f'(x_k)$ avec $f'(x_k) \neq 0\ \forall\, k = 0, 1, ...$ Il s'agit d'une méthode de point fixe, avec $\phi(x) = x - f(x)/f'(x)$.

1.6.2 Convergence de la méthode

On a $\phi'(x) = f(x)f''(x)/(f'(x))^2$, donc $\phi'(r) = 0$. Alors, en utilisant le théorème 4, on montre qu'il existe un voisinage $\mathbb{I}$ de r tel que si $x_0 \in \mathbb{I}$, la méthode de Newton converge.

De plus, on peut montrer que la convergence est quadratique : Si ϕ est de classe C^2, $\phi'(r) = 0$ et $|\phi''| < M$ sur $\mathbb{I}$, on peut écrire le développement de Taylor-Lagrange à l'ordre 2 de $\phi(x)$ en $x = r$:
$\phi(x) = \phi(r + x - r) = \phi(r) + \phi'(r)(x-r) + (1/2)\,(x-r)^2\,\phi''(\xi),\ \xi \in]r, x[$
$\Longrightarrow \phi(x) = \phi(r) + (1/2)\,(x-r)^2\,\phi''(\xi) \Longrightarrow |\phi(x) - \phi(r)| = |\phi(x) - r| \leq (M/2)\,|x-r|^2$
En considérant la suite x_0 donné, et $x_{n+1} = \phi(x_n)$, on a donc :
$|x_n - r| = |\phi(x_{n-1}) - r| \leq (M/2)\,|x_{n-1} - r|^2 \leq ... \leq (M/2)^n\,|x_0 - r|^{2n}$. La convergence est quadratique.
On peut constater graphiquement qu'on a intérêt à prendre x_0 tel que $f(x_0)\,f''(x_0) > 0$.

$x_0 = a < r,\ f(a)\,f''(a) < 0 \longrightarrow x_1 > r$
$x_0 = b > r,\ f(b)\,f''(b) < 0 \longrightarrow x_1 < r$

Théorème 5 : Si f est de classe C^2 sur $[a, b]$, f' et f'' gardent des signes constants sur $[a, b]$ et $f(a)\,f(b) < 0$, la suite de Newton converge vers la racine unique $r \in [a, b]$ de $f(x) = 0$ si le point initial vérifie : $f(x_0)\,f''(x_0) > 0$.
Preuve : Prenons par exemple $f(a) < 0$, $f(b) > 0$, $f'(x) > 0$, $f''(x) > 0$, $\forall x \in [a, b]$. Alors $f(b)\,f''(b) > 0$ et on peut choisir $x_0 = b$ (ou x_0 entre r et b : $x_0 > r$).
Par récurrence supposons $x_1 > r$, $x_2 > r$, ...$x_n > r$, alors

$f(r) = 0 = f(x_n) + f'(x_n)\,(r - x_n) + 1/2\, f''(\xi_n)\,(r - x_n)^2$
puisque $f''(x) > 0,\ \forall\, x \in [a, b] \Longrightarrow f(x_n) + f'(x_n)\,(r - x_n) < 0.$
mais $f'(x) > 0,\ \forall\, x \in [a, b]$ d'où $r + f(x_n)/f'(x_n) - x_n < 0 \longrightarrow x_{n+1} > r$
donc $x_n > r\, \forall\, n \in \mathbb{N}$.
Mais $x_{n+1} = x_n - f(x_n)/f'(x_n)$ montre que $x_{n+1} < x_n$ car $f(x_n)/f'(x_n) > 0\, \forall\, n$. Il s'agit donc d'une suite décroissante minorée par r. Elle est convergente vers une limite l qui doit vérifier $l = l - f(l)/f'(l) \Longrightarrow f(l) = 0$. Comme la racine de $f(x) = 0$ est unique sur $[a, b]$, on en déduit que $l \equiv r$. On peut mener une démonstration analogue dans les autres cas.

1.6.3 Avantages et inconvénients

Pour utiliser la méthode de Newton, il faut connaître $f'(x)$ sous forme analytique, et connaître la forme de la courbe $f(x)$. D'autre part, il faut avoir bien localisé la racine r pour pouvoir initialiser le calcul avec x_0 suffisamment proche de r.

Si x_0 est suffisamment proche de r pour que la méthode soit convergente, alors la convergence est quadratique.

1.7 Méthode de Newton-Raphson de recherche des extréma

Soit à résoudre le système de 2 équations à 2 inconnues :

$$\left\{ \begin{array}{rcl} f(x, y) = & 0 \\ g(x, y) = & 0 \end{array} \right.$$

On se donne (x_0, y_0) aussi proches que possible de la solution cherchée.
On pose $x = x_0 + \Delta x,\ y = y_0 + \Delta y$

$$\Longrightarrow \left\{ \begin{array}{rcl} f(x, y) = & f(x_0, y_0) + \Delta x\, f'_{x_0} + \Delta y\, f'_{y_0} = & 0 \\ g(x, y) = & g(x_0, y_0) + \Delta x\, g'_{x_0} + \Delta y\, g'_{y_0} = & 0 \end{array} \right.$$

C'est un système linéaire par rapport à Δx et Δy. On pose :
$f_0 = f(x_0, y_0),\ f'_{x_0} = f'_x(x_0, y_0),\ f'_{y_0} = f'_y(x_0, y_0),$
$g_0 = g(x_0, y_0),\ g'_{x_0} = g'_x(x_0, y_0),\ g'_{y_0} = g'_y(x_0, y_0).$

Par inversion du système, on a : $\left\{ \begin{array}{l} \Delta x = (g_0\, f'_{y_0} - f_0\, g'_{y_0})/(f'_{x_0}\, g'_{y_0} - g'_{x_0}\, f'_{y_0}) \\ \Delta y = (f_0\, g'_{x_0} - g_0\, f'_{x_0})/(f'_{x_0}\, g'_{y_0} - g'_{x_0}\, f'_{y_0}) \end{array} \right.$

On obtient $x_1 = x_0 + \Delta x,\ y_1 = y_0 + \Delta y$.
On répète ce processus jusqu'à la convergence : $|x_p - x_{p-1}| < \epsilon$ et $|y_p - y_{p-1}| < \epsilon$.

Ce processus se généralise dans le cas d'un système à n équations et n inconnues :

$$\left\{ \begin{array}{rcl} f_1(x_1, ..., x_n) = & 0 \\ ... \\ ... \\ ... \\ f_n(x_1, ..., x_n) = & 0 \end{array} \right.$$

qu'on met sous la forme vectorielle $F(X) = 0$ où X et F sont données par leurs composantes : $X = \{x_1, ..., x_n\}^t,\ F = \{f_1(x), ..., f_n(x)\}^t$.

On a alors $F(X) = F(X_0 + \Delta X) = F(X_0) + [J(X_0)]\,\Delta X = 0$

$$[J(X)] = \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 & ..... & \partial f_1/\partial x_n \\ ..... & ..... & ..... & ..... \\ ..... & ..... & ..... & ..... \\ ..... & ..... & ..... & ..... \\ \partial f_n/\partial x_1 & \partial f_n/\partial x_2 & ..... & \partial f_n/\partial x_n \end{bmatrix}.$$

$[J(X_0)]$ est la matrice jacobienne $[J(X)]$ du système en $X = X_0$.

Le système linéaire à n équations et n inconnues à résoudre est :

$$F(X_0) + [J(X_0)]\ \Delta X = 0$$

où les inconnues sont ΔX qu'on obtient par

$$\Delta X = -\left[J(X_0)\right]^{-1}\ F(X_0) \quad \Longrightarrow \quad X_1 = X_0 + \Delta X$$

On continue le processus jusqu'à la convergence qui est obtenue lorsque X_p et X_{p-1} sont suffisamment proches.

2 Méthodes directes

2.1 Inversion de systèmes linéaires

On considère un système de n équations linéaires à n inconnues $x_1, x_2, ..., x_n$.

$$\begin{aligned} a_{1,1}\,x_1 + a_{1,2}\,x_2 + ... + a_{1,n}\,x_n &= b_1 \\ a_{2,1}\,x_1 + a_{2,2}\,x_2 + ... + a_{2,n}\,x_n &= b_2 \\ &. \\ &. \\ a_{n,1}\,x_1 + a_{n,2}\,x_2 + ... + a_{n,n}\,x_n &= b_n \end{aligned}$$

Ce système s'écrit matriciellement sous la forme $A\,x = b$ où A est une matrice (n,n), de coefficients $a_{ij}, (i = 1, ..., n\,;\ j = 1, ..., n)$ où i est l'indice des lignes et j l'indice des colonnes, $b^t = (b_1, ..., b_n)$, $x^t = (x_1, ..., x_n)$.
On suppose que la matrice A est régulière (A carrée, $det(A) \neq 0$). Donc il existe une solution unique qui peut s'écrire matriciellement sous la forme : $x = A^{-1}\,b$.
En utilisant les formules de Cramer : $x_i = \Delta_i/\Delta,\ \ i = 1, 2, ..., n$, où Δ est le déterminant de A, Δ_i est le déterminant de la matrice A où la i^{eme} colonne est remplacée par les composantes de b. La solution $x_i = \Delta_i/\Delta$ n'est pas utilisable numériquement pour $n > 10$: Estimons le nombre d'opérations élémentaires nécessaires pour résoudre ce système : on doit calculer $(n+1)$ déterminants (n, n) puis effectuer n divisions. Le calcul d'un déterminant nécessite $n!$ multiplications et $(n-1)!$ additions. Le nombre d'opérations grandit factoriellement avec n.
Exemple : $n = 5 \rightarrow 6! = 720, n = 10 \rightarrow 11 = !4 \times 10^7, n = 50 \rightarrow 51! = 1.5 \times 10^{66}, n = 100 \rightarrow 101! = 10^{160}$.

En considérant une performance moyenne de 100 GFlops (nombre d'opérations en virgule flottante par seconde), soit 10^{11} opérations par seconde (ce qui équivaut à 3×10^{18} opérations par an, car 1 an=3×10^{7} s) au delà de $n > 20$, le calcul est irréalisable. Il est donc impossible d'utiliser les formules de Cramer pour résoudre le système si n est grand.

Définition : Une méthode directe est une méthode permettant de résoudre à l'erreur machine près, en un nombre fini d'opérations, le système $Ax = b$. Ces méthodes sont basées sur des décompositions de la matrice A.

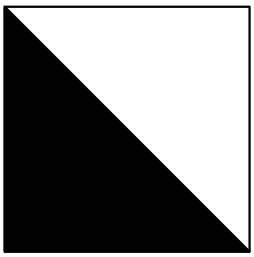
2.2 Matrices triangulaires

2.2.1 Triangulaire inférieure : algorithme de descente

On s'intéresse au système $Lx = b$, où L est une matrice triangulaire inférieure (Lower). Le système s'écrit sous la forme :

$$\begin{aligned} l_{1,1}\,x_1 &= b_1 \\ l_{2,1}\,x_1 + l_{2,2}\,x_2 &= b_2 \\ &\vdots \\ l_{n,1}\,x_1 + l_{n,2}\,x_2 ..... + l_{n,n}\,x_n &= b_n \end{aligned}$$

Un tel système se résout de proche en proche, dans l'ordre, en calculant x_1, puis x_2, ..., jusqu'à x_n.



$$\Longrightarrow \begin{cases} x_1 = & b_1/l_{11} & i = 1 \\ x_i = & (b_i - \sum_{j=1}^{i-1} l_{ij}\,x_j)/l_{ii} & 2 \le i \le n \end{cases}$$

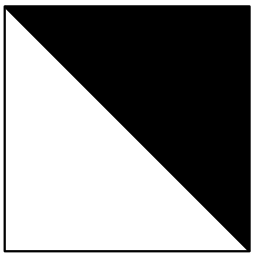
Nombre d'opérations pour la résolution :
Le nombre d'opérations pour la résolution est : n divisions, $1+2+...+(n-1) = n\,(n-1)/2$ multiplications, $n\,(n-1)/2$ additions, soit au total n^2 opérations élémentaires.

2.2.2 Triangulaire supérieure : Algorithme de remontée

On s'intéresse au système $Ux = b$, où U est une matrice triangulaire supérieure (Upper). Le système s'écrit sous la forme :

$$\begin{aligned} u_{1,1}\,x_1 + u_{1,2}\,x_2 + ... + u_{1,n}\,x_n &= b_1 \\ u_{2,2}\,x_2 + ... + u_{2,n}\,x_n &= b_2 \\ &\vdots \\ u_{n,n}\,x_n &= b_n \end{aligned}$$

Un tel système se résout de proche en proche, dans l'ordre, en calculant x_n, puis x_{n-1}, ..., jusqu'à x_1.



$$\Longrightarrow \begin{cases} x_n = & b_n/u_{nn} & i = n \\ x_i = & (b_i - \sum_{j=i+1}^{n} u_{ij}\, x_j)/u_{ii} & n-1 \geq i \geq 1 \end{cases}$$

Nombre d'opérations pour la résolution :
Ici aussi la résolution nécessite n^2 opérations élémentaires.

2.3 Méthode de Gauss (sans pivotage)

Elle s'applique aux matrices carrées. La méthode de Gauss transforme, en n étapes, le système $A\,x = b$ en un système triangulaire $A^{(n)}\,x = b^{(n)}$ où $A^{(n)} = \left(a_{i,j}^{(n)}\right)$ est une matrice triangulaire supérieure. Les différentes étapes de la méthode de Gauss sont :

Etape 1 : $A^{(1)} = A$, $b^{(1)} = b$
On pose $A^{(1)} = A, b^{(1)} = b \Longrightarrow A\,x = b \Longleftrightarrow A^{(1)}\,x = b^{(1)}$.
L'élément $a_{1,1}^{(1)}$ s'appelle premier pivot de l'élimination. Le système $A^{(1)}\,x = b^{(1)}$ s'écrit :

$$\begin{aligned} a_{1,1}^{(1)}\,x_1 + a_{1,2}^{(1)}\,x_2 + ... + a_{1,n}^{(1)}\,x_n &= b_1^{(1)} \\ . & \\ a_{i,1}^{(1)}\,x_1 + a_{i,2}^{(1)}\,x_2 + ... + a_{i,n}^{(1)}\,x_n &= b_i^{(1)} \\ . & \\ a_{n,1}^{(1)}\,x_1 + a_{n,2}^{(1)}\,x_2 + ... + a_{n,n}^{(1)}\,x_n &= b_n^{(1)} \end{aligned}$$

Etape 2 : $A^{(2)} = G^{(1)}\,A^{(1)}$, $b^{(2)} = G^{(1)}\,b^{(1)}$
Le système$^{(1)}$ est ensuite transformé en un système$^{(2)}$ défini par $A^{(2)}\,x = b^{(2)}$ obtenu en multipliant pour $i = 2, 3, ..., n$ la 1^{ere} équation du système$^{(1)}$ par $g_{i,1} = a_{i,1}^{(1)}/a_{1,1}^{(1)}$ et en retranchant l'équation obtenue de la i^{eme} équation du système$^{(1)}$. On a alors :

$$\begin{cases} a_{1,j}^{(2)} = & a_{1,j}^{(1)} \quad (j = 1, ..., n) \\ a_{i,1}^{(2)} = & 0 \quad (i = 2, ..., n) \\ a_{i,j}^{(2)} = & a_{i,j}^{(1)} - g_{i,1}\,a_{1,j}^{(1)} \quad (i = 2, ..., n,\ j = 2, ..., n) \end{cases}$$

$$\begin{cases} b_1^{(2)} = & b_1^{(1)} \\ b_1^{(2)} = & b_i^{(1)} - g_{i,1}\,b_1^{(1)} \quad (i = 2, ..., n) \end{cases}$$

Les matrices $G^{(1)}$ et $A^{(2)}$ s'écrivent :

$$G^{(1)} = \begin{bmatrix} 1 & & & \\ -g_{2,1} & 1 & & \\ . & & & \\ . & & & \\ . & & & \\ -g_{n,1} & & & 1 \end{bmatrix}, \quad A^{(2)} = \begin{bmatrix} a_{1,1}^{(2)} & a_{1,2}^{(2)} & & a_{1,n}^{(2)} \\ 0 & a_{2,2}^{(2)} & & a_{2,n}^{(2)} \\ . & & & \\ . & & & \\ . & & & \\ 0 & a_{n,2}^{(2)} & .... & a_{n,n}^{(2)} \end{bmatrix}$$

Etape 3 : $A^{(3)} = G^{(2)}\, A^{(2)}$, $b^{(3)} = G^{(2)}\, b^{(2)}$
On pose $g_{i,2} = a_{i,2}/a_{2,2}^{(2)} \quad (i = 3, 4, ..., n)$, puis on retranche la 2^{eme} équation multipliée par $g_{i,2}$ de la i^{eme} équation pour $(i = 3, ..., n)$. Les matrices $G^{(2)}$ et $A^{(3)}$ s'écrivent :

$$G^{(2)} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & -g_{3,2} & 1 & & \\ & . & & & \\ & . & & & \\ & -g_{n,2} & & & 1 \end{bmatrix}, \quad A^{(3)} = \begin{bmatrix} a_{1,1}^{(3)} & a_{1,2}^{(3)} & a_{1,3}^{(3)} & . & . & a_{1,n}^{(3)} \\ 0 & a_{2,2}^{(3)} & a_{2,3}^{(3)} & . & . & a_{2,n}^{(3)} \\ & 0 & a_{3,3}^{(3)} & . & . & a_{3,n}^{(3)} \\ . & & & & & \\ . & & & & & \\ 0 & 0 & a_{n,3}^{(3)} & . & . & a_{n,n}^{(3)} \end{bmatrix}$$

On obtient le système $A^{(3)}\, x = b^{(3)}$.

Etape k+1 : $A^{(k+1)} = G^{(k)}\, A^{(k)}$, $b^{(k+1)} = G^{(k)}\, b^{(k)}$
Ayant ainsi construit $A^{(k)}$ et $b^{(k)}$, on déduit $A^{(k+1)}$ et $b^{(k+1)}$ en posant $g_{i,k} = a_{i,k}^{(k)}/a_{k,k}^{(k)} \quad (i = k+1, ..., n)$ où on suppose que le k^{eme} pivot $a_{k,k}^{(k)} \neq 0$. On retranche la k^{eme} ligne multipliée par $g_{i,k}$ de la i^{eme} ligne pour $(i = k+1, ..., n)$. On obtient la matrice $A^{(k+1)}$ et le becteur $b^{(k+1)}$:

$$\begin{cases} a_{i,j}^{(k+1)} = a_{i,j}^{(k)} & (i = 1, ..., k, j = 1, ..., n) \\ a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - g_{i,k}\, a_{k,j}^{(k)} & (i = k+1, ..., n,\ j = k+1, ..., n) \end{cases}$$

$$\begin{cases} b_i^{(k+1)} = b_i^{(k)} & (i = 1, ..., k) \\ b_i^{(k+1)} = b_i^{(k)} - g_{i,k}\, b_k^{(k)} & (i = k+1, ..., n) \end{cases}$$

Etape n : $A^{(n)} = G^{(n-1)}\, A^{(n-1)}$, $b^{(n)} = G^{(n-1)}\, b^{(n-1)}$
On continue l'opération jusqu'à l'ordre n. A la fin de la n^{eme} étape, on trouve $A^{(n)}\, x = b^{(n)}$ où la matrice $A^{(n)}$ est triangulaire supérieure. Le nouveau système s'écrit :

$$\begin{aligned} a_{1,1}^{(n)}\, x_1 + a_{1,2}^{(n)}\, x_2 + ... + a_{1,n}^{(n)}\, x_n &= b_1^{(n)} \\ + a_{2,2}^{(n)}\, x_2 + ... + a_{2,n}^{(n)}\, x_n &= b_2^{(n)} \\ . & \\ . & \\ a_{n,n}^{(n)}\, x_n &= b_n^{(n)} \end{aligned}$$

Résolution de $A^{(n)}\, x = b^{(n)}$:
La résolution numérique par remontée est immédiate.

Nombre total d'opérations élémentaires effectuées :
Pour passer de $A^{(k)}, b^{(k)}$ à $A^{(k+1)}, b^{(k+1)}$, on effectue :
$(n-k)$ divisions, $(n-k)(n-k+1)$ multiplications et $(n-k)(n-k+1)$ soustractions.
Donc l'élimination nécessite au total :
$\sum_{k=1}^{n-1}(n-k) = \sum_{k=1}^{n-1} k = n(n-1)/2$ divisions,
$\sum_{k=1}^{n-1}(n-k)\,(n-k+1) = \sum_{k=1}^{n-1}(k^2+k) = n(n-1)(2n-1)/6 + n(n-1)/2 = n(n^2-1)/3$ multiplications, et autant de soustractions, soit $n(n^2-1)/3$ soustractions.
Au total, compte tenu du nombre d'opérations nécessaires à la résolution du système triangulaire, la méthode de Gauss utilise :
divisions : $n(n-1)/2 + n = n(n+1)/2$,
multiplications : $n(n^2-1)/3 + n(n-1)/2 = (2n^3+3n^2-5n)/6 = n(2n-1)(n+5)/6$,
additions ou soustractions : $n(2n-1)(n+5)/6$.
Soit au total : $(4n^3+9n^2-7n)/6$ opérations élémentaires.
exemple : $n=5 \to 115, n=10 \to 805, n=50 \to 87025, n=100 \to 681550$.

2.4 Méthode de Gauss avec pivotage partiel

On effectue n transformation de la matrice A. A chaque transformation on effectue, si nécessaire, une permutation de deux lignes afin que le pivot soit de module maximal. La permutation entre les lignes p et q s'effectue par utilisation de la matrice de permutation P. La matrice P est obtenue à partir de la matrice identité dans laquelle on a permuté les lignes p et q. La matrice P est symétrique et égale à son inverse. Le produit matriciel $A' = P\,A$ permet d'obtenir la matrice A' qui est identique à la matrice A avec permutation des lignes p et q de A.
La matrice de permutation $P(p,q) \in \mathbb{R}^{n,n}$ $(1 \le p < q \le n)$ tel que $P(p,q) = P^t(p,q)$ est définie par :

$$\begin{bmatrix}
1 & & & & & & & & & & & & \\
& \ddots & & & & & & & & & & & \\
& & \ddots & & & & & & & & & & \\
& & & 1 & & & & & & & & & \\
& & & & 0 & & & & & 1 & & & \\
& & & & & 1 & & & & & & & \\
& & & & & & \ddots & & & & & & \\
& & & & & & & \ddots & & & & & \\
& & & & & & & & 1 & & & & \\
& & & & 1 & & & & & 0 & & & \\
& & & & & & & & & & 1 & & \\
& & & & & & & & & & & \ddots & \\
& & & & & & & & & & & & \ddots \\
& & & & & & & & & & & & & 1
\end{bmatrix}$$

Etape k+1 : $A^{(k+1)} = G^{(k)}\,P^{(k)}\,A^{(k)}$, $b^{(k+1)} = G^{(k)}\,P^{(k)}\,b^{(k)}$
Pour des raisons de stabilité, on a intérêt à ce que le pivot $a_{k,k}^{(k)}$ soit le plus grand possible. On procède de la manière suivante : à la k^{eme} étape de l'élimination, on choisit comme k^{eme} pivot l'élément de module maximum parmi les $(n-k+1)$ dernières composantes de la k^{eme} colonne de $A^{(k)}$ et on permute la k^{eme} ligne avec la ligne où se trouve le pivot de plus grand module.

Etape n : $A^{(n)} = G^{(n-1)}\, P^{(n-1)}\, A^{(n-1)}$, $b^{(n)} = G^{(n-1)}\, P^{(n-1)}\, b^{(n-1)}$
De même que dans le cas sans pivotage, après n transformations, on aboutit au système linéaire suivant $A^{(n)}\, x = b^{(n)}$. La matrice $A^{(n)}$ est triangulaire supérieure. Le système est résolu par un algorithme classique de remontée.

2.5 Factorisation des matrices par points

2.5.1 factorisation $A = LU$, L à diagonale 1

Soit $A \in \mathbb{R}^{n,n}$ une matrice régulière, il existe deux matrices :
$L \in \mathbb{R}^{n,n}$ triangulaire inférieure à diagonale unité,
$U \in \mathbb{R}^{n,n}$ triangulaire supérieure,
telle que $A = L\,U$. La factorisation est unique. On dispose de (n^2) équations, les inconnues sont au nombre de $n(n-1)/2$ pour L, $n(n+1)/2$ pour U, soit n^2 inconnues au total.

Pour construire l'algorithme ou faire la factorisation à la main sur une matrice "petite" : développer le produit LU et identifier terme à terme, dans l'ordre (colonne par colonne ou ligne à ligne) avec la matrice A.

algorithme $1 \le j \le n$

$l_{jj} = 1.$
$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}\, u_{kj} \qquad 1 \le i \le j$
$l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik}\, u_{kj})/u_{jj} \qquad j+1 \le i \le n$

En fait, comme la décomposition s'effectue de façon ordonnée, le stockage des matrices L et U peut s'effectuer directement dans la matrice A, les valeurs de L et de U "écrasant" progressivement celles de la matrice A.
La résolution du système $A\,x = b$ s'effectue en deux étapes :
(a) $L\,y = b$ se résout par descente $\rightarrow y$,
(b) $U\,x = y$ se résout par remontée $\rightarrow x$.

On peut aussi choisir de faire la décomposition (unique) $A = L\,U$, avec U à diagonale unité. L'algorithme de décomposition est différent, mais la méthode de résolution complète (factorisation + descente + remontée) est équivalente en terme de nombre d'opérations élémentaires.

2.5.2 factorisation $A = LDU$

Soit $A \in \mathbb{R}^{n,n}$ une matrice régulière, il existe trois matrices :
$L \in \mathbb{R}^{n,n}$ triangulaire inférieure à diagonale unité,
$D \in \mathbb{R}^{n,n}$ diagonale,
$U \in \mathbb{R}^{n,n}$ triangulaire supérieure à diagonale unité,
telle que $A = L\,D\,U$. Cette factorisation est unique. On dispose de (n^2) équations, les inconnues sont au nombre de $n(n-1)/2$ pour L, n pour D, $n(n-1)/2$ pour U. Soit $2n(n-1)/2 + n = n^2$.

algorithme $1 \le j \le n$

$u_{ij} = (a_{ij} - \sum_{k=1}^{i-1} l_{ik}\, d_{kk}\, u_{kj})/d_{ii} \qquad 1 \le i \le j-1$
$d_{jj} = (a_{jj} - \sum_{k=1}^{j-1} l_{jk}\, d_{kk}\, u_{kj}) \qquad i = j$
$l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik}\, d_{kk}\, u_{kj})/d_{jj} \qquad j+1 \le i \le n$

La résolution du système $A\,x = b$ s'effectue en trois étapes :

(a) $L\,z = b$ se résout par descente $\rightarrow z$,
(b) $D\,y = z$ se résout par descente $\rightarrow y$,
(c) $U\,x = y$ se résout par remontée $\rightarrow x$.
Cette méthode de factorisation est équivalente à la précédente ($A = LU$).

2.5.3 factorisation de Cholesky $A = LL^t$

Cette décomposition s'applique aux matrices symétriques définies positives (On rappelle qu'une matrice A est définie positive ssi $x^t Ax \geq 0$, $\forall x \in \mathbb{R}^n$, et $x^t Ax = 0 \Leftrightarrow x = 0$).
On peut décomposer $A = L\,L^t$ où L est triangulaire inférieure à diagonale strictement positive. La décomposition s'effectue par identification ordonnée du produit $L\,L^t$ avec la matrice A.
algorithme $1 \leq j \leq n$

$l_{jj} = (a_{jj} - \sum_{k=1}^{j-1} |l_{jk}|^2)^{1/2}$
$l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik}\,l_{jk})/l_{jj} \qquad j+1 \leq i \leq n$

La résolution du système $A\,x = b$ s'effectue en deux étapes :
(a) $L\,y = b$ se résout par descente $\rightarrow y$,
(b) $L^t\,x = y$ se résout par remontée $\rightarrow x$.
On gagne sur le stockage (on ne stocke plus de matrice U, car on tient compte de la symétrie de la matrice A) ainsi que sur le nombre d'opérations nécessaires à la décomposition.

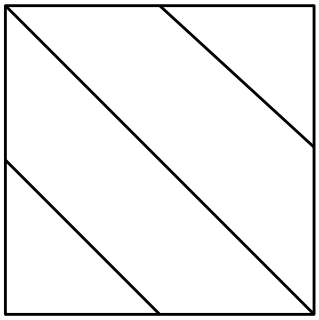
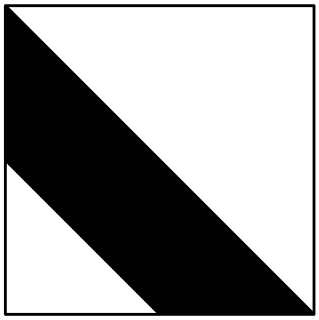
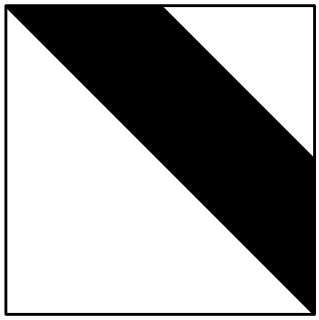
2.5.4 factorisation $A = QR$

Soit $A \in \mathbb{R}^{n,n}$ une matrice régulière, il existe une décomposition unique $A = Q\,R$ où $Q \in \mathbb{R}^{n,n}$ est orthogonale ($Q^{-1} = Q^t$) et $R \in \mathbb{R}^{n,n}$ est triangulaire supérieure à diagonale strictement positive.
On pose $M = A^t\,A \longrightarrow M$ est symétrique définie positive. En effet on a $M^t = (A^t\,A)^t = A^t(A^t)^t = M$, donc M est symétrique, et $x^t\,M\,x = x^t\,A^t\,A\,x = (Ax)^t\,(Ax) = \|Ax\|^2 \geq 0$, et $x^t\,M\,x = 0 \Leftrightarrow \|Ax\| = 0 \Leftrightarrow Ax = 0 \Leftrightarrow x = 0$.
La matrice M admet donc une décomposition de Cholesky, et se met sous la forme $M = R^t\,R$.
On pose $Q = A\,R^{-1}$. On a alors :
$Q^t\,Q = R^{-1^t}\,A^t\,A\,R^{-1} = R^{-1^t}\,M\,R^{-1} = (R^{-1})^t\,R^t\,R\,R^{-1} = (R\,R^{-1})^t = I$.
On conclut que Q est orthogonale et $A = Q\,R$.

2.5.5 factorisation des matrices bandes creuses

Les factorisations $L\,D\,U$, $L\,U$, LL^t conservent les largeurs de bandes d'où un gain en place mémoire et en temps de calcul. Le stockage se fait diagonale par diagonale. L'inconvénient de ces méthodes est qu'elles remplissent les bandes. Elles sont utilisées pour les systèmes d'ordre peu élevés ($n \leq 1000$).
exemple sur le stockage :
Soit A une matrice pentadiagonale creuse de taille (n, n), avec $n = 10^6$ ($n^2 = 10^{12}$). Si la demi-largeur de bande est $l = 100$, le stockage bande nécessite au plus $2nl = 2\,10^6\,10^2 = 2\,10^8$ termes, alors qu'il y a seulement $5\,10^6$ termes non nuls.

Un cas particulier est celui des matrices A tridiagonales. Dans ce cas, la matrice A est stockée sous forme de trois diagonales non nulles. L'algorithme utilisé est TDMA (tri-diagonal matrix

algorithm, ou algorithme de Thomas), basé sur une décomposition $A = LU$. Le stockage des matrices L et U s'effectue également par diagonale. Comme la décomposition s'effectue de façon ordonnée, le stockage des diagonales de L et U peut s'effectuer dans les diagonales de la matrice A, et les valeurs de L et de U écrasent progressivement celles de la matrice A. Cet algorithme direct est utilisé même pour des systèmes tridiagonaux de grande taille.
Exemple sur une matrice de taille (3,3) :

$$A = \left[\begin{array}{c|c|c|c} d_1 & c_1 & & \\ \hline a_2 & d_2 & c_2 & \\ \hline & a_3 & d_3 & c_3 \\ \hline & & a_4 & d_4 \end{array}\right] \qquad b = \left[\begin{array}{c} b_1 \\ \hline b_2 \\ \hline b_3 \\ \hline b_4 \end{array}\right]$$

$$L = \left[\begin{array}{c|c|c|c} l_1 & & & \\ \hline m_2 & l_2 & & \\ \hline & m_3 & l_3 & \\ \hline & & m_4 & l_4 \end{array}\right] \qquad U = \left[\begin{array}{c|c|c|c} 1 & u_1 & & \\ \hline & 1 & u_2 & \\ \hline & & 1 & u_3 \\ \hline & & & 1 \end{array}\right]$$

2.6 Normes, conditionnement

2.6.1 Normes de vecteurs

définitions :
La norme des vecteurs fournit une mesure de distance. La norme d'un vecteur sur $\mathbb{R}^n$ est une fonction $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ qui satisfait les propriétés suivantes :

$f(x) \geq 0 \qquad x \in \mathbb{R}^n \quad (f(x) = 0 \, ssi \, x = 0)$
$f(x+y) \leq f(x) + f(y) \qquad x, y \in \mathbb{R}^n$
$f(\alpha x) = |\alpha| f(x) \qquad \alpha \in \mathbb{R}, x \in \mathbb{R}^n.$

On note $f(x) = \|x\|$, et on définit la norme p par $\|x\|_p = (|x_1|^p + ... + |x_n|^p)^{1/p} \qquad p \geq 1.$

Les normes 1, 2 et ∞ sont les plus importantes.

$\|x\|_1 = |x_1| + ... + |x_n|$
$\|x\|_2 = (|x_1|^2 + ... + |x_n|^2)^{1/2} = (x^t \, x)^{1/2}$
$\|x\|_\infty = max_{1 \leq i \leq n} |x_i|.$

Un vecteur unitaire est défini par $\|x\| = 1$

propriétés :
Toutes les normes sur $\mathbb{R}^n$ sont équivalentes. Si $\|.\|_\alpha$ et $\|.\|_\beta$ sont des normes sur $\mathbb{R}^n$, il existe deux constantes C_1 et C_2 tel que, pour tout $x \in \mathbb{R}^n$:

$$C_1 \|x\|_\alpha \leq \|x\|_\beta \leq C_2 \|x\|_\alpha$$

Si une suite $\{x^{(k)}\}$ de vecteurs tend vers x pour une norme, alors elle converge vers x pour les autres normes : $lim_{k\to\infty}\|x^{(k)} - x\| = 0$.

relations

a) $\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\,\|x\|_2$
démonstration : $\|x\|_2^2 = \sum_{i=1}^n |x_i|^2 \leq \|x\|_1^2 = (\sum_{i=1}^n |x_i|)^2 \Longrightarrow \|x\|_2 \leq \|x\|_1$
$\|x\|_1^2 = (\sum_{i=1}^n |x_i|)^2 = \sum_{i=1}^n |x_i|^2 + 2\sum_{i<j} |x_i|\,|x_j| \leq n\sum_{i=1}^n |x_i^2|$
car $2\,|x_i|\,|x_j| \leq |x_i|^2 + |x_j|^2 \Longrightarrow \|x\|_1 \leq \sqrt{n}\,\|x\|_2$

b) $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\,\|x\|_\infty$
démonstration : $\|x\|_\infty^2 = (max_i\,|x_i|)^2 \leq (max_i\,|x_i|)^2 + \sum_{i=1,i\neq k}^n |x_i|^2 = \|x\|_2^2$
$\Longrightarrow \|x\|_\infty \leq \|x\|_2$
$\|x\|_2^2 = \sum_{i=1}^n |x_i|^2 \leq n\,max_i\,|x_i|^2 = n\,\|x\|_\infty^2 \Longrightarrow \|x\|_2 \leq \sqrt{n}\,\|x\|_\infty$

c) $\|x\|_\infty \leq \|x\|_1 \leq n\,\|x\|_\infty$
démonstration : $\|x\|_\infty = max_i\,|x_i| \leq \sum_{i=1}^n |x_i| = \|x\|_1$
$\|x\|_1 = \sum_{i=1}^n |x_i| \leq n\,\|x\|_\infty$

2.6.2 Normes des matrices

Définitions :
La définition de la norme matricielle est équivalente à la définition de la norme vectorielle. En particulier $f : \mathbb{R}^{m,n} \longrightarrow \mathbb{R}$ est une norme matricielle si :

$f(A) \geq 0 \qquad A \in \mathbb{R}^{m,n} \quad (f(A) = 0\, ssi\, A = 0)$
$f(A+B) \leq f(A) + f(B) \qquad A, B \in \mathbb{R}^{m,n}$
$f(\alpha A) = |\alpha| f(A) \qquad \alpha \in \mathbb{R}, x \in \mathbb{R}^{m,n}.$

On note $f(A) = \|A\|$.
On définit alors les normes suivantes :

a) normes p :

$$\|A\|_p = Sup_{x\neq 0}\frac{\|Ax\|_p}{\|x\|_p} = Sup_{x\neq 0}\left\|A\left(\frac{x}{\|x\|_p}\right)\right\|_p = max_{\|x\|_p=1}\|A\,x\|_p$$

Propriété (par définition) : Pour tout $A \in \mathbb{R}^{m,n}$ et $x \in \mathbb{R}^n$, on a $\|A\,x\|_p \leq \|A\|_p\|x\|_p$.
Plus généralement, pour toute norme vectorielle $\|.\|_\alpha$ sur $\mathbb{R}^n$ et $\|.\|_\beta$ sur $\mathbb{R}^n$, on a

$$\|A\,x\|_\beta \leq \|A\|_{\alpha,\beta}\,\|x\|_\alpha$$

où $\|A\|_{\alpha,\beta}$ est une norme matricielle définie par

$$\|A\,x\|_{\alpha,\beta} = Sup_{x\neq 0}\frac{\|A\,x\|_\beta}{\|x\|_\alpha}$$

On dit que $\|.\|_{\alpha,\beta}$ est subordonnée aux normes vectorielles $\|.\|_\alpha$ et $\|.\|_\beta$.
Ainsi $\|A\|_{\alpha,\beta} = max_{\|x\|_\alpha=1}\|A\,x\|_\beta = \|A\,x^*\|_\beta$ pour $x^* \in \mathbb{R}^n$ de norme α unité.
Pour $A \in \mathbb{R}^{m,n}$, on a ainsi les normes les plus utilisées :

$\|A\|_1 = max_{1\leq j\leq n} \sum_{i=1}^{m} |a_{ij}|$

$\|A\|_\infty = max_{1\leq i\leq m} \sum_{j=1}^{n} |a_{ij}|$

b) norme 2 :
On rappelle que le rayon spectral $\rho(A)$ d'une matrice A carrée est $\rho(A) = max\,|\lambda_i|$ où les λ_i sont les valeurs propres de A.
On rappelle également que les racines μ des valeurs propres de la matrice $A^t\,A$ sont appelées valeurs singulières de A.
Alors, la norme 2 de A est définie par : $\|A\|_2 = \rho^{1/2}(A^t\,A) = \mu_1$, la plus grande des valeurs singulières de A.
Si A est carrée et symétrique $\Longrightarrow \|A\|_2 = \rho(A)$.

Relations Pour $A \in \mathbb{R}^{m,n}$, on a :
$max_{i,j}\,|a_{ij}| \leq \|A\|_2 \leq \sqrt{mn}\; max_{i,j}\,|a_{ij}|$
$\frac{1}{\sqrt{n}}\;\|A\|_\infty \leq \|A\|_2 \leq \sqrt{m}\;\|A\|_\infty$
$\frac{1}{\sqrt{m}}\;\|A\|_1 \leq \|A\|_2 \leq \sqrt{n}\;\|A\|_1$
Les normes matricielles 1,2, et ∞ sont donc équivalentes.
Si on ne cherche qu'une estimation de $\|A\|_2$, on peut utiliser les relations de $\|A\|_2$ avec $\|A\|_1$ et $\|A\|_\infty$.
corollaire : Si $A \in \mathbb{R}^{m,n} \Longrightarrow \|A\|_2 \leq \sqrt{\|A\|_1\,\|A\|_\infty}$. En effet, si $z \neq 0$ tel que $A^t\,A\,z = \mu^2 z$ avec $\mu = \|A\|_2$, alors :

$$\mu^2\|z\|_1 = \|A^t\,A\,z\|_1 \leq \|A^t\|_1\,\|A\|_1\,\|z\|_1 = \|A\|_\infty\,\|A\|_1\,\|z\|_1.$$

Propriétés utiles par la suite :
i) On a $\|AB\| \leq \|A\|\|B\|$. En effet, pour tout x, on a $\|ABx\| \leq \|A\|\|Bx\| \leq \|A\|\|B\|\|x\|$ d'où $\frac{\|ABx\|}{\|x\|} \leq \|A\|\|B\|$ et donc $\|AB\| = Sup_{x\neq 0}\frac{\|ABx\|}{\|x\|} \leq \|A\|\|B\|$.
i) Pour les matrices carrées, on a $\rho(A) \leq \|A\|$: en effet, soit u un vecteur propre de A associé à la valeur propre λ, alors on a $\|Au\| = \|\lambda u\| = |\lambda|\|u\| \leq \|A\|\|u\|$, donc $|\lambda| \leq \|A\|$ pour toute valeur propre λ. C'est donc vrai pour la valeur propre de plus grand module, et donc $\rho(A) \leq \|A\|$.

2.6.3 Conditionnement des matrices

Lorsqu'on veut résoudre $Ax = b$, les éléments de A et de b sont connus avec une incertitude. La solution x sera affectée par ces incertitudes auxquelles s'ajoutent les erreurs d'arrondi. La solution obtenue est celle d'un système perturbé :
$(A + \Delta A)\,(x + \delta x) = (b + \delta b)$.

Afin d'estimer l'effet des perturbations ΔA et δb, on introduit le nombre de conditionnement de la matrice A. Le conditionnement n'est défini que pour les matrices inversibles. Pour une matrice régulière A, associée à une norme $\|.\|$, le conditionnement de A est le nombre $cond(A) = \|A\|\,\|A^{-1}\|$.
On a alors les relations suivantes :

a) cas où $\Delta A = 0$, on résout $A(x+\delta x) = (b+\delta b)$

$$\left\{\begin{array}{ll} A(x+\delta x) = & (b+\delta b) \\ A\,x = & b \end{array}\right. \Longrightarrow \left\{\begin{array}{ll} \delta x = & A^{-1}\,\delta b \\ b = & A\,x \end{array}\right. \Longrightarrow \left\{\begin{array}{ll} \|\delta x\| \leq & \|A^{-1}\|\,\|\delta b\| \\ \|b\| \leq & \|A\|\,\|x\| \end{array}\right.$$

$$\Longrightarrow \frac{\|\delta x\|}{\|x\|} \leq \{\|A\|\,\|A^{-1}\|\}\frac{\|\delta b\|}{\|b\|} \quad \text{ou} \quad \frac{\|\delta x\|}{\|x\|} \leq cond(A)\,\frac{\|\delta b\|}{\|b\|}$$

L'erreur relative sur le résultat est majorée par l'erreur relative sur les données multipliée par $cond(A)$.

b) cas où $\delta b = 0$, on résout $(A+\Delta A)(x+\delta x) = b$

$$\left\{\begin{array}{ll} (A+\Delta A)(x+\delta x) = & b \\ A\,x = & b \end{array}\right\}$$

$$\Longrightarrow A\,\delta x + \Delta A\,(x+\delta x) = 0 \quad \text{ou} \quad \delta x = -A^{-1}\,\Delta A\,(x+\delta x)$$

$$\Longrightarrow \|\delta x\| \leq \|A^{-1}\|\,\|\Delta A\|\|x+\delta x\| \Longrightarrow \frac{\|\delta x\|}{\|x+\delta x\|} \leq cond(A)\,\frac{\|\Delta A\|}{\|A\|}$$

c) cas où $\Delta A \neq 0$ et $\delta b \neq 0$: on a à résoudre $(A+\Delta A)(x+\delta x) = (b+\delta b)$
On démontre que :

$$\frac{\|\delta x\|}{\|x\|} \leq cond(A)\,\left[\frac{\|\Delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right]$$

Pour obtenir une estimation de $cond(A)$ (sans bien sûr calculer la matrice A^{-1}), on effectue alors les deux résolutions suivantes, en choisissant ΔA et δb : (1) $A\,x_1 = b$ et (2) $(A+\Delta A)\,x_2 = b+\delta b$
On calcule alors $\delta x = x_2 - x_1$, dont on peut calculer la norme (de son choix). On peut en déduire un minorant de $cond(A)$, dans la norme choisie :

$$\frac{\frac{\|\delta x\|}{\|x\|}}{\left[\frac{\|\Delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right]} \leq cond(A).$$

Ce n'est qu'un minorant, mais cela donne une idée. Le système le mieux conditionné est un système où $cond(A)$ est d'ordre 1 (matrice identité). Les systèmes mal conditionnés sont caractérisés par de très grandes valeurs de conditionnement.

Si l'on veut connaître exactement le conditionnement d'une matrice, on peut le calculer en norme 2 (voir chapitre sur les valeurs propres).

2.6.4 Equilibrage des matrices

Si $cond(A)$ est grand, on cherche deux matrices diagonales D_1 et D_2 tel que $B = D_1 A D_2$ soit mieux conditionnée que A. On résout $D_1\,A\,D_2 y = D_1 b$ puis on a immédiatement $x = D_2\,y$. On peut choisir $D_2 = I$ et D_1 de telle manière que $D_1\,A$ ait à peu près la même norme ∞ pour toutes les lignes.

3 Méthodes itératives

La résolution du système : $A\,x = b$, où $A \in \mathbb{R}^{n,n}$, $b \in \mathbb{R}^n$ par une méthode itérative revient à construire une suite de vecteurs $x^{(k)} \in \mathbb{R}^n$ convergente vers x : $lim_{k\to\infty} x^{(k)} = x = A^{-1}\,b$.

Définitions

1. Une méthode itérative est consistante si la limite de $x^{(k)}$, lorsqu'elle existe, est solution de $A\,x = b$.
2. Une méthode itérative est convergente si, pour toute valeur initiale $x^{(0)} \in \mathbb{R}^n$, on a : $lim_{k\to\infty} x^{(k)} = x = A^{-1}\,b$. Une méthode peut être consistante sans être convergente.

3.1 Algorithmes itératifs linéaires

Ils sont de la forme : $x^{(k+1)} = \Omega\,x^{(k)} + c$, $k \in \mathbb{N}$. (avec $\Omega \in \mathbb{R}^{n,n}$ et $c \in \mathbb{R}^n$). La matrice Ω est appelée matrice d'itération.

a) **consistance**
La méthode est consistante si et seulement si $(I - \Omega)$ inversible et $c = (I - \Omega)\,A^{-1}\,b$.
preuve : On a $x^{(k+1)} = \Omega\,x^{(k)} + c$, donc la limite x^* de $x^{(k)}$, si elle existe, vérifie : $x^* = \Omega\,x^* + c$, soit $(I - \Omega)x^* = c$. Pour que x^* soit égal à la solution $x = A^{-1}\,b$, il faut et il suffit que (a) la matrice $(I - \Omega)$ soit inversible, et (b) $(I - \Omega)A^{-1}\,b = c$.

b) **convergence**
Si la méthode est consistante, alors elle converge si et seulement si le rayon spectral de la matrice d'itération est strictement inférieur à 1, $\rho(\Omega) < 1$.
preuve : Si la méthode est consistante, on a $x^{(k+1)} = \Omega\,x^{(k)} + c$ et $x = \Omega\,x + c$. Si on note le vecteur erreur $e^{(k)} = x^{(k)} - x \in \mathbb{R}^n$, on a $e^{(k+1)} = \Omega\,e^{(k)}$, ce qui donne par récurrence $e^{(k)} = \Omega^k\,e^{(0)}$.
Donc la méthode est convergente si et seulement si $lim_{k\to\infty}\,\Omega^k\,e^{(0)} = 0,\ \forall\, e^{(0)} \in \mathbb{R}^n$.
Si la matrice Ω est diagonalisable, on note $(u_i)_{i=1,n}$ une base orthonormée de vecteurs propres, où chaque vecteur u_i est associé à la valeur propre λ_i.
On peut alors écrire, en décomposant le vecteur quelconque $e^{(0)}$ sur la base des vecteurs propres, $e^{(0)} = \sum_i \alpha_i u_i$, et $\lim_{k\to\infty} \Omega^k\,e^{(0)} = \lim_{k\to\infty} \sum_i \alpha_i {\lambda_i}^k u_i = 0 \Leftrightarrow |\lambda_i| < 1 \Leftrightarrow \rho(\Omega) < 1$. Ce résultat peut se généraliser aux matrices non diagonalisables.

3.2 Splitting régulier

3.2.1 Introduction

Définition :
Un splitting régulier de la matrice A, s'obtient en introduisant les matrices M et N, avec M inversible telles que $A = M - N$.

Principe de la résolution itérative associée :
Au lieu de résoudre $Ax = b$, on résout $Mx^{(k+1)} = Nx^{(k)} + b$ avec une solution initiale $x^{(0)}$.
La matrice M est choisie de telle façon que le système puisse se résoudre simplement à chaque itération (M triangulaire ou bien égale au produit de plusieurs matrices triangulaires).
Le critère d'arrêt des itérations est sur la norme du vecteur résidu $r^{(k)} = b - A\,x^{(k)}$, soit $\|r^{(k)}\| < \epsilon$, avec ϵ choisi, petit. La norme est une norme vectorielle au choix ($\|.\|_1$, $\|.\|_2$, ou $\|.\|_\infty$).
Formellement la suite x^{k+1} est solution de $x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b$.

La matrice d'itération est $\Omega = M^{-1}N$. La méthode itérative associée à un splitting converge ssi $\rho(\Omega) < 1$.

3.2.2 Décomposition de type sommation

On introduit la décomposition fixe : $A = D - E - F$, où $D \in \mathbb{R}^{n,n}$ est la diagonale de A (D est inversible), $E \in \mathbb{R}^{n,n}$ est triangulaire inférieure stricte, $F \in \mathbb{R}^{n,n}$ est triangulaire supérieure stricte.

Méthode de Jacobi
On choisit le splitting suivant :

$$M = D \Longrightarrow N = E + F \quad ; \quad \Omega_J = D^{-1}(E + F)$$

On résout par descente : $D\, x^{(k+1)} = (E + F)\, x^{(k)} + b$, car la matrice D est diagonale.
Si on note $x_i^{(k+1)}$ la i^{eme} composante de $x^{(k+1)}$, l'algorithme de résolution est, pour k_{max} et ϵ donnés, et pour un choix de norme vectorielle :

Initialisation : $x_i^{(0)}$ donnés pour $i = 1, 2, ..., n$
Pour $k = 0, 1, 2, ...k_{max}$

$$\text{pour } i = 1, 2, ..., n : \quad x_i^{(k+1)} = (b_i - \sum_{j=1, j\neq i}^{n} a_{i,j}\, x_j^{(k)})/a_{i,i}$$

$$\text{pour } i = 1, 2, ..., n : \quad r_i^{(k+1)} = (b_i - \sum_{j=1}^{n} a_{i,j}\, x_j^{(k+1)})$$

$$\text{si } \|r^{(k+1)}\| < \epsilon \text{ stop}$$

$$\text{pour } i = 1, 2, ..., n : \quad x_i^{(k)} = x_i^{(k+1)}$$

Fin de boucle

critères de convergence :
La convergence a lieu ssi $\rho(\Omega) < 1$, donc ssi $\rho(\Omega_J) = \rho(D^{-1}(E + F)) < 1$. On en déduit des conditions suffisantes de convergence :

Théorème de convergence :
Si A est inversible et à diagonale strictement dominante, alors la méthode de Jacobi est convergente.
preuve : On calcule que $\Omega_{i,i} = 0$ et $\Omega_{i,j} = -a_{i,j}/a_{i,i}$ pour $i \neq j$,
Prenons le cas où A est inversible et à diagonale strictement dominante selon les lignes, c'est-à-dire :
$|a_{i,i}| > \sum_{j=1, j\neq i}^{n} |a_{i,j}|,\ \forall\, i = 1, 2, ..., n.$
Alors, pour tout i, on a $\sum_{j=1}^{n} |\Omega_{i,j}| = (\sum_{j=1, j\neq i}^{n} |a_{i,j}|)/|a_{i,i}| < 1$, et donc $\|\Omega\|_\infty < 1$, et comme $\|\rho(\Omega) \leq \|\Omega\|$, on en déduit $\rho(\Omega) < 1$, la méthode est donc convergente.
De façon analogue, on montre que si A est inversible et à diagonale strictement dominante suivant les colonnes, alors $\|\Omega\|_1 < 1$, et comme $\|\rho(\Omega) \leq \|\Omega\|$, on en déduit $\rho(\Omega) < 1$, la méthode est donc convergente.

Théorème de convergence :
Si A et $2D-A$ sont symétriques définie positives, alors la méthode de Jacobi est convergente.

Méthode de Gauss-Seidel

$$M = D - E \Longrightarrow N = F \quad ; \quad \Omega_G = (D-E)^{-1}F$$

On résout par descente : $(D-E)\,x^{(k+1)} = F\,x^{(k)} + b$, car la matrice $D-E$ est triangulaire inférieure.

l'algorithme de résolution est, pour k_{max} et ϵ donnés, et pour un choix de norme vectorielle :

Initialisation : $x_i^{(0)}$ donnés pour $i = 1, 2, ..., n$
Pour $k = 0, 1, 2, ...k_{max}$

$$\text{pour } i = 1, 2, ..., n : \quad x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{i,j}\, x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j}\, x_j^{(k)})/a_{i,i}$$

$$\text{pour } i = 1, 2, ..., n : \quad r_i^{(k+1)} = (b_i - \sum_{j=1}^{n} a_{i,j}\, x_j^{(k+1)})$$

$$\text{si } \|r^{(k+1)}\| < \epsilon \text{ stop}$$

$$\text{pour } i = 1, 2, ..., n : \quad x_i^{(k)} = x_i^{(k+1)}$$

Fin de boucle

Théorème de convergence :
Si A est inversible et à diagonale strictement dominante, alors la méthode de Gauss-Seidel est convergente.
preuve : On fait l'hypothèse que A est inversible et à diagonale strictement dominante selon les lignes. Il faut montrer que $\rho(\Omega_G) < 1$.
Pour cela il suffit de montrer que $||\Omega_G||_\infty < 1$ puisque $\rho(\Omega_G) \leq ||\Omega_G||_\infty$.
On a $\Omega_G = (D-E)^{-1}\,F$. Soit $x \in \mathbb{R}^n - \{0\}$. Posons $y = \Omega_G\, x \Longrightarrow y = (D-E)^{-1}\,F\,x$
$\Longrightarrow (D-E)\,y = F\,x$ ou $D\,y = E\,y + F\,x$ ou $(D^{-1}\,E)\,y + (D^{-1}\,F)\,x$.

$$\Longrightarrow y_k = -\sum_{j=1}^{k-1} \frac{a_{k,j}}{a_{k,k}}\, y_j - \sum_{j=k+1}^{n} \frac{a_{k,j}}{a_{k,k}}\, x_j \quad \forall\, k = 1, 2, ..., n$$

Soit l tel que $|y_l| = Sup_k|y_k|$, alors $||y||_\infty = |y_l|$, on a

$$||y||_\infty = |y_l| \leq \sum_{j=1}^{l-1} \frac{|a_{l,j}|\,|y_j|}{|a_{l,l}|} + \sum_{j=l+1}^{n} \frac{|a_{l,j}|\,|x_j|}{|a_{l,l}|}$$

$$\Longrightarrow ||y||_\infty \leq \left(\sum_{j=1}^{l-1} \frac{|a_{l,j}|}{|a_{l,l}|}\right) ||y||_\infty + \left(\sum_{j=l+1}^{n} \frac{|a_{l,j}|}{|a_{l,l}|}\right) ||x||_\infty$$

ou $(1-s_l)||y||_\infty \leq r_l||x||_\infty$ avec $s_l = 1/|a_{l,l}|\ \sum_{j=1}^{l-1} |a_{l,j}|; r_l = 1/|a_{l,l}|\ \sum_{j=l+1}^{n} |a_{l,j}|$.
On remarque que $0 < s_l < 1$ et $0 < r_l < 1$ avec $s_l + r_l = 1/|a_{l,l}|\ \sum_{j=1,j\neq l}^{n} |a_{l,j}| < 1$ car A est

à diagonale dominante selon les lignes.
On a alors $||y||_\infty/||x||_\infty \leq r_l/(1-s_l) < 1$ car $r_l/(1-s_l) < 1 \Longleftrightarrow r_l + s_l < 1$. Ainsi il existe l, $1 \leq l \leq n$ tel que

$$\frac{||\Omega_G\, x||_\infty}{||x||_\infty} \leq \frac{r_l}{1-s_l} \leq Sup_{k=1,\ldots,n}(\frac{r_k}{1-s_k}) < 1 \;\forall\, x \in \mathbb{R}^n - \{0\}$$

$$\Longrightarrow ||\Omega_G||_\infty \leq Sup_{k=1,\ldots,n}(\frac{r_k}{1-s_k}) < 1$$

Par conséquent $||\Omega_G||_\infty < 1$ et la méthode de Gauss-Seidel est convergente.
On peut montrer de façon analogue à partir de la norme 1, que si A est inversible et à diagonale dominante selon les colonnes, alors la méthode de Gauss-Seidel est convergente.

Théorème de convergence :
Si A est symétrique définie positive, alors la méthode de Gauss-Seidel est convergente.
preuve : Remarquons que le produit scalaire $< x, y >_A =< A\,x, y >= y^t A x$ permet de définir sur $\mathbb{R}^n$ une norme $||.||_A$ telle que : $||x||_A = [< A\,x, x >]^{1/2} = (x^t A x)^{1/2}$ et la norme $||\Omega||_A = Sup_{x\neq 0}(||\Omega\, x||_A/||x||_A)$.
Montrons que $||\Omega_G||_A < 1$ où $\Omega_G = I - (D-E)^{-1}\,A = (D-E)^{-1}\,F$.
On a $A = A^t \Longrightarrow F = E^t$, d'où :

$$\begin{aligned} ||\Omega_G x||_A^2 = &< A\,\Omega_G\, x, \Omega_G\, x > \\ = &< A\,[I-(D-E)^{-1}\,A]\,x, [I-(D-E)^{-1}\,A]\,x > \\ = &< A\,x, x > + < A\,(D-E)^{-1}\,A\,x, (D-E)^{-1}\,A\,x > \\ &- < A\,(D-E)^{-1}\,A\,x, x > - < A\,x, (D-E)^{-1}\,A\,x > \end{aligned}$$

Si on pose $y = (D-E)^{-1}\,A\,x \Longleftrightarrow A\,x = (D-E)\,y$, on obtient, en développant les produits scalaires :
$||\Omega_G x||_A^2 = ||x||_A^2 + < A\,y, y > -(< A\,y, x > + < A\,x, y >)$.
Or, on a, en tenant compte de $A^t = A$, et $(D-E)^t = D - E^t$:
$< A\,y, x >=< x, A\,y >=< x, A[(D-E)(D-E)^{-1}]^t\,y >=< x, A(D-E)^{-1^t}\,(D-E^t)\,y >$
$=< (D-E)^{-1}A^t x, (D-E^t)\,y >=< y, (D-E^t)\,y >$.
Donc $< A\,y, x > + < A\,x, y >) =< y, (D-E^t)\,y > + < (D-E)\,y, y >$.
On a également : $< A\,y, y >=< (D-E-F)\,y, y >=< (D-E-E^t)\,y, y >$.
D'où :

$$\begin{aligned} < A\,y, y > -(< A\,y, x > + < A\,x, y >) = &< (D-E-E^t)\,y, y > \\ &- < (D-E^t)\,y, y > - < (D-E)\,y, y > \\ = &- < D\,y, y > < 0 \;\forall\, x \neq 0 \end{aligned}$$

$$\Longrightarrow \left(\frac{||\Omega_G\, x||_A}{||x||_A}\right)^2 - 1 = -\frac{< D\,y, y >}{||x||_A^2} < 0 \quad \forall\, x \neq 0 \Longrightarrow ||\Omega_G||_A < 1$$

Méthode SOR (successive over relaxation) On peut améliorer la convergence de la méthode de Gauss-Seidel en utilisant une matrice Ω_ω dépendant d'un paramètre ω choisi pour que le rayon spectral $\rho(\Omega_\omega)$ soit le plus petit possible.
On choisit alors :

$$M = \frac{1}{\omega}D - E \Longrightarrow N = (\frac{1}{\omega} - 1)D + F \quad ; \quad \Omega_\omega = (D - \omega E)^{-1}[(1-\omega)D + \omega F]$$

On résout par descente : $(\frac{1}{\omega}D - E)\, x^{(k+1)} = [(\frac{1}{\omega} - 1)D + F]\, x^{(k)} + b$, car la matrice $(\frac{1}{\omega}D - E)$ est triangulaire inférieure.
On remarque que pour $\omega = 1$, on retrouve la méthode de Gauss-Seidel.

l'algorithme de résolution est, pour ω, k_{max} et ϵ donnés, et pour un choix de norme vectorielle :

Initialisation : $x_i^{(0)}$ donnés pour $i = 1, 2, ..., n$
Pour $k = 0, 1, 2, ...k_{max}$

$$\text{pour } i = 1, 2, ..., n : \quad x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega(b_i - \sum_{j=1}^{i-1} a_{i,j}\, x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j}\, x_j^{(k)})/a_{i,i}$$

$$\text{pour } i = 1, 2, ..., n : \quad r_i^{(k+1)} = (b_i - \sum_{j=1}^{n} a_{i,j}\, x_j^{(k+1)})$$

$$\text{si } \|r^{(k+1)}\| < \epsilon \text{ stop}$$

$$\text{pour } i = 1, 2, ..., n : \quad x_i^{(k)} = x_i^{(k+1)}$$

Fin de boucle

proposition : Si $\omega \notin]0, 2[$, SOR diverge $\Longrightarrow \rho(\Omega_\omega) \geq 1$.
En effet, on a $\Omega_\omega = (D - \omega E)^{-1}[(1-\omega)D + \omega F] = [D(I - \omega D^{-1}E)]^{-1}D[(1-\omega)I + \omega D^{-1}F]$, d'où :
$\Omega_\omega = (I - \omega D^{-1}E)^{-1}[(1-\omega)I + \omega D^{-1}F]$.
On en déduit : $det(\Omega_\omega) = det([I - \omega D^{-1}E]^{-1})det[(1-\omega)I + \omega D^{-1}F]$, d'où : $det(\Omega_\omega) = det[(1-\omega)I + \omega D^{-1}F]/det[I - \omega D^{-1}E]$.
Or la matrice $(1-\omega)I + \omega D^{-1}F$ est triangulaire supérieure et tous ses termes diagonaux sont égaux à $1 - \omega$, donc son déterminant est égal au produit $(1-\omega)^n$.
De plus la matrice $(I - \omega D^{-1}E)$ est triangulaire inférieure à diagonale unité, donc son déterminant est égal à 1.
Finalement on a $det(\Omega_\omega) = (1-\omega)^n$. Or $det(\Omega_\omega) = \prod_{i=1}^n \lambda_i$, où les λ_i sont les valeurs propres de la matrice Ω_ω. Donc $\prod_{i=1}^n |\lambda_i| = |1-\omega|^n \leq \rho(\Omega_\omega)^n$ et donc $|1-\omega| \leq \rho(\Omega_\omega)$ Alors :
Si $\omega \leq 0 \Longrightarrow (1-\omega) \geq 1 \Longrightarrow \rho(\Omega_\omega) \geq 1$.
Si $\omega \geq 2 \Longrightarrow -1 \geq (1-\omega) \Longrightarrow (1-\omega)^n \geq 1 \Longrightarrow \rho(\Omega_\omega) \geq 1$.

Pour que S.O.R. converge, on doit prendre $0 < \omega < 2$ (c'est une condition nécessaire mais pas suffisante).

Théorème de convergence :
Si A est symétrique définie positive, alors la méthode SOR est convergente ssi $0 < \omega < 2$.

Calcul d'une valeur optimale ω_{opt}
On cherche alors une valeur optimale de ω en adoptant l'une des deux stratégies suivantes (la stratégie est analogue à celle choisie pour la méthode de relaxation du point fixe dans le cas du calcul des racines d'une équation) :

a) On choisit une valeur du critère de convergence ϵ relativement "grande" ($\epsilon = 10^{-3}$ par exemple) et on cherche la valeur de ω pour laquelle la convergence s'effectue en

un minimum d'itérations k_ϵ (avec la même condition initiale x_0).
En pratique, on fait varier $\omega \in]0,2[$ avec un pas $\Delta\omega$, et on construit la courbe $k_\epsilon = f(\omega)$. La valeur ω_{opt}, obtenue à $\pm\Delta\omega$ près, correspond au minimum $k_{\epsilon min}$.

b) On choisit un nombre maximal d'itérations relativement "petit" ($k_{max} = 100$ par exemple) et on cherche la valeur de ω pour laquelle la norme du résidu $\|r^{(k_{max})}\|$ obtenu à l'issue des k_{max} itérations est minimum (avec la même condition initiale x_0).
En pratique, on fait varier $\omega \in]0,2[$ avec un pas $\Delta\omega$, et on construit la courbe $\|r^{(k_{max})}\| = g(\omega)$. La valeur ω_{opt}, obtenue à $\pm\Delta\omega$ près, correspond au minimum $\|r^{(k_{max})}\|_{min}$.

3.2.3 Décomposition de type factorisation incomplète

Méthode S.I.P. (strongly implicit procedure) ou I.L.U.(incomplete LU) :
On utilise une décomposition fixe de la forme $A = L_0\,U_0 + R_0$ qui ne s'utilise que lorsque la matrice A est creuse, ou plus particulièrement si A est pentadiagonale creuse ou heptadiagonale creuse. La factorisation est dite incomplète, car les matrices L_0 et U_0 respectent la structure creuse de A (les diagonales non nulles sont celles de A). La matrice R_0 est dite matrice reste.
La factorisation incomplète est effectuée une seule fois avant le début du processus de calcul itératif, et elle permet de calculer les éléments non nuls de L_0 et de U_0, puis la matrice reste R_0.

On pose $M = \frac{1}{\alpha}L_0\,U_0 \Longrightarrow N = (\frac{1}{\alpha} - 1)L_0\,U_0 - R_0, \quad \alpha > 0 \in \mathbb{R}$.
On résout $\frac{1}{\alpha}L_0\,U_0\,x^{(k+1)} = [(\frac{1}{\alpha} - 1)L_0\,U_0 - R_0]\,x^{(k)} + b$ en effectuant une descente puis une remontée par points.

Méthode F.C.I. (factorisation de Cholesky incomplète) :
Si $A \in \mathbb{R}^{n,n}$ est creuse et symétrique définie positive, on utilise une décomposition incomplète $A = L_0\,L_0^t + R_0$ et on prend $M = \frac{1}{\alpha}L_0\,L_0^t \Longrightarrow N = (\frac{1}{\alpha} - 1)L_0\,L_0^t - R_0$.
On résout $\frac{1}{\alpha}L_0\,L_0^t\,x^{(k+1)} = [(\frac{1}{\alpha} - 1)L_0\,L_0^t - R_0]\,x^{(k)} + b$.

Méthode SSOR (symmetric successive over relaxation) :
Elle est basée sur un double splitting où les matrices M, N, P, Q, sont définies par :

$$M = \frac{1}{\omega}D - E \Longrightarrow N = (\frac{1}{\omega} - 1)D + F \quad \text{et} \quad P = \frac{1}{\omega}D - F \Longrightarrow Q = (\frac{1}{\omega} - 1)D + E$$

On résout par descente : $(\frac{1}{\omega}D - E)\,x^{(k+1/2)} = [(\frac{1}{\omega} - 1)D + F]\,x^{(k)} + b$
puis par remontée : $(\frac{1}{\omega}D - F)\,x^{(k+1)} = [(\frac{1}{\omega} - 1)D + E]\,x^{(k+1/2)} + b$

3.3 Méthode de Richardson stationnaire

Soit $M = \frac{1}{\alpha}I \rightarrow N = \frac{1}{\alpha}I - A$ et $\Omega_R = I - \alpha\,A$.

$$\text{on a} \quad \frac{1}{\alpha}x^{(k+1)} = (\frac{1}{\alpha}I - A)\,x^{(k)} + b$$

$$\text{ou} \quad x^{(k+1)} = x^{(k)} + \alpha\,(b - A\,x^{(k)}) \Longrightarrow x^{(k+1)} = x^{(k)} + \alpha\,r^{(k)} \quad \text{avec } r^{(k)} = b - A\,x^{(k)}.$$

3.4 Méthode de Richardson généralisée (préconditionnement des matrices)

Au lieu de résoudre $Ax = b$, on veut résoudre le système $C^{-1}\,Ax = C^{-1}b$, dont on espère qu'il sera mieux conditionné que le système initial. On choisit alors une matrice C dite matrice de préconditionnement, et "facilement inversible". Le "meilleur" choix de C est une matrice telle que $C^{-1}\,A$ soit proche de I, et donc C proche de A.
La méthode de Richardson stationnaire appliquée à ce nouveau système s'écrit :
$C\,x^{(k+1)} = C\,x^{(k)} + \alpha\,r^{(k)}$, ou encore $C\,(x^{(k+1)} - x^{(k)}) = \alpha\,r^{(k)}$, et la matrice d'itération est $\Omega_C = (I - \alpha\,C^{-1}\,A)$.
(On voit bien, pour $\alpha = 1$, que pour assurer $\rho(\Omega_C) < 1$ le plus petit possible, on a intérêt à choisir C proche de A.) L'algorithme de résolution est, pour α, k_{max} et ϵ donnés, et pour un choix de norme vectorielle :

Initialisation : $x^{(0)}$ donné, calcul de $r^{(0)} = b - A\,x^{(0)}$,
Pour $k = 0, 1, 2, ...k_{max}$
- Calcul de $r^{(k)} = b - A\,x^{(k)}$
- Test sur la norme du vecteur résidu : Si $\|r^{(k)}\| < \epsilon$ stop
- Résolution de $C\,\delta x^{(k+1)} = \alpha\,r^{(k)}$ pour obtenir $\delta x^{(k+1)}$
- Calcul de $x^{(k+1)} = x^{(k)} + \delta x^{(k+1)}$

Fin de boucle

En fait les méthodes classiques de splitting régulier se mettent toutes sous une forme Richardson généralisée, avec $C = \alpha M$.
En effet $M\,x^{(k+1)} = Nx^{(k)} + b = (M - A)\,x^{(k)} + b = M\,x^{(k)} + r^{(k)}$, si on pose $M = \frac{1}{\alpha}C$, est équivalent à écrire $C\,x^{(k+1)} = C\,x^{(k)} + \alpha\,r^{(k)}$. On peut ainsi retrouver les matrices de préconditionnement correspondant aux méthodes classiques, et évaluer si C est proche de A.

- Jacobi : $C = D$, $\alpha = 1$
- Gauss-Seidel : $C = D - E$, $\alpha = 1$
- S.O.R. : $C = D - \omega\,E$, $\alpha = \omega$ à optimiser
- S.I.P. : $C = L_0\,U_0$, α à optimiser
- F.C.I. : $C = L_0\,L_0^t$, α à optimiser
- S.S.O.R. : $C = \frac{1}{2-\omega}(D - \omega\,E)\,D^{-1}\,(D - \omega\,F)$, $\alpha = \omega$ à optimiser

3.5 Méthode du gradient ou de Richardson instationnaire

Introduction :
Soit $A \in \mathbb{R}^{n,n}$ symétrique définie positive. On peut montrer que résoudre $A\,x = b$ revient à minimiser la fonctionnelle $J(x) = \frac{1}{2}x^t\,A\,x - x^t\,b$ (voir cours d'analyse fonctionnelle, LU3ME003 par exemple).

Sans effectuer la démonstration complète, on peut remarquer que le gradient de $J(x)$ est $g(x) = -(A\,x - b) = -r(x)$ où $r(x) = b - A\,x$.
Ceci se montre par exemple en utilisant le calcul indiciel :
$\vec{grad}(J(x))|_k = (J(x))_{,k} = \frac{1}{2}(x_i A_{i,j} x_j)_{,k} - (x_i b_i)_{,k} = \frac{1}{2}(\delta_{ik}\,A_{i,j}x_j + x_i A_{i,j}\delta_{jk}) - \delta_{ik}b_i$, d'où

$\vec{grad}(J(x))|_k = \frac{1}{2}(A_{k,j}x_j + x_i A_{i,k}) - b_k = \frac{1}{2}(A_{k,j}x_j + A_{k,i}x_i) - b_k = A_{k,j}x_j - b_k = (A\,x - b)_k$.
Donc le minimum de $J(x)$ annule le gradient de $J(x)$ et donc annule le vecteur résidu.

Principe de la méthode :
On part donc de la méthode de Richardson stationnaire, mais cette fois, à chaque itération, on détermine le scalaire optimal α_k qui permet de calculer $x^{(k+1)}$ par

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

C'est la méthode de Richardson instationnaire, ou encore appelée méthode du gradient.

On suppose qu'on part d'un vecteur x. On cherche α optimal en minimisant $J(x + \alpha r)$:
$Q(\alpha) = J(x + \alpha r) = \frac{1}{2}(x^t + \alpha r^t)A(x + \alpha r) - (x^t + \alpha r^t)b$
$Q(\alpha) = \frac{1}{2}(x^t Ax + \alpha r^t Ax + \alpha x^t Ar + \alpha^2 r^t Ar) - x^t b - \alpha r^t b$. Or $r^t Ax$ est un scalaire et A est symétrique, et donc $r^t Ax = (r^t Ax)^t = x^t A^t r = x^t Ar$. On a donc :
$Q(\alpha) = \frac{1}{2}(r^t Ar)\alpha^2 + (r^t Ax - r^t b)\alpha + \frac{1}{2}r^t Ax - x^t b$.
Le coefficient de α^2 étant positif, le minimum est obtenu en α tel que $Q'(\alpha) = 0$, soit :
$Q'(\alpha) = (r^t Ar)\alpha + r^t(Ax - b) = 0$, et donc : $\alpha = \frac{r^t(b - Ax)}{r^t Ap} = \frac{<r,r>}{<Ar,r>}$.

D'après le calcul précédent, la valeur optimale de α_k est :

$$\alpha_k = \frac{< r^{(k)}, r^{(k)} >}{< A\,r^{(k)}, r^{(k)} >} = \frac{||r^{(k)}||_2^2}{< A\,r^{(k)}, r^{(k)} >}$$

et $\forall\, r^k \neq 0$, et α_k optimal, on a : $r^{(k+1)} = b - Ax^{(k+1)} = b - A(x^{(k)} + \alpha_k r^{(k)}) = r^{(k)} - \alpha_k\, A\, r(k)$.

Note : les notations équivalentes suivantes du produit scalaire de deux vecteurs u et v sont utilisées ici $< u, v >= u^t v$.

Algorithme :
L'algorithme correspondant est donc
Initialisation : $x^{(0)}$ donné, calcul de $r^{(0)} = b - A\,x^{(0)}$,
Pour $k = 0, 1, 2, ...k_{max}$
- Calcul de $\alpha_k = \frac{||r^{(k)}||_2^2}{<A\,r^{(k)}, r^{(k)}>}$
- Calcul de $x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)}$,
- Calcul de $r^{(k+1)} = r^{(k)} - \alpha_k\, A\, r^{(k)}$.
- Test sur la norme du vecteur résidu : Si $\|r^{(k+1)}\| < \epsilon \rightarrow$ stop

Fin de boucle

Convergence :
La rapidité de convergence dépend du rapport $[cond(A)-1]/[cond(A)+1]$ et est d'autant plus intéressante que $cond(A) \rightarrow 1$. Lorsque $cond(A)$ est grand, les valeurs propres extrêmes sont très différentes et la convergence suit un ellipsoide très aplati, alors que pour $cond(A) = 1$, la convergence est immédiate car toute tangente orientée vers l'intérieur de la sphère passe par le centre de cette sphère.

3.6 Méthode du gradient conjugué

Cette méthode ne s'applique que si A est symétrique, définie, positive.
Cette fois-ci, on cherche à déterminer à chaque itération une direction de descente optimale $p^{(k)}$, et on cherchera $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$.
On choisit α_k pour que $J(x^{(k)} + \alpha_k p^{(k)})$ soit un minimum local. On trouve, après des calculs similaires à ceux détaillés précédemment que $\alpha_k = \frac{<p^{(k)}, r^{(k)}>}{<A\,p^{(k)}, p^{(k)}>}$, et alors que $r^{(k+1)} = r^{(k)} - \alpha_k\, A\, p^{(k)}$. On montre que les directions p^k et r^{k+1} sont orthogonales :
En effet, on a $< p^{(k)}, r^{(k+1)} >=< p^{(k)}, r^{(k)} - \alpha_k\, A\, p^{(k)} >=< p^{(k)}, r^{(k)} > -\alpha_k < p^{(k)},\, A\, p^{(k)} >$ soit, $< p^{(k)}, r^{(k+1)} >=< p^{(k)}, r^{(k)} > -\frac{<p^{(k)}, r^{(k)}>}{<A\,p^{(k)}, p^{(k)}>} < p^{(k)}, A\, p^{(k)} >= 0$.
On cherche $p^{(k+1)}$ dans le plan formé par les deux directions orthogonales $r^{(k+1)}$ et $p^{(k)}$, et on pose $p^{(k+1)} = r^{(k+1)} + \beta_{k+1}\, p^{(k)}$. On est amené à chercher β_{k+1} tel que $< A\, p^{(k)}, p^{(k+1)} >= 0$.
Lorsque deux vecteurs u et v vérifient la relation $< A\, u, v >= 0$, on dit qu'ils sont A conjugués. Comme A est symétrique, définie, positive, alors $< A\, u, v >$ définit un produit scalaire. On cherche donc β_{k+1} tel que $p^{(k)}$ et $p^{(k+1)}$ soient A conjugués.

On en déduit $< A\, p^{(k)}, r^{(k+1)} > +\beta_{k+1}< A\, p^{(k)}, p^{(k)} > = 0$, et β_{k+1} optimal est donc donné par $\beta_{k+1} = -\frac{<A\,p^{(k)}, r^{(k+1)}>}{<A\,p^{(k)}, p^{(k)}>}$, ce qui se simplifie en $\beta_{(k+1)} = \frac{<r^{(k+1)}, r^{(k+1)}>}{<r^{(k)}, r^{(k)}>}$.

En effet, on a $\beta_{k+1} = -\frac{<A\,p^k, r^{k+1}>}{<A\,p^k, p^k>} = \frac{-<r^{(k)}, r^{(k+1)}>+<r^{(k+1)}, r^{(k+1)}>}{<r^{(k)}, p^{(k)}>-<r^{(k+1)}, p^{(k)}>} = \frac{<r^{k+1}, r^{k+1}>}{<r^k, r^k>}$

La rapidité de convergence dépend du rapport $\left[\sqrt{cond(A)} - 1\right] / \left[\sqrt{cond(A)} + 1\right]$.

algorithme :
La matrice $A \in \mathbb{R}^{n,n}$ est symétrique définie positive. Soit $x^{(0)}$ un vecteur donné, on calcule $r^{(0)} = b - A\, x^{(0)}$ et on pose $p^{(0)} = r^{(0)}$.

$$k = 0, 1, ..., n \quad \left| \begin{array}{rcl} \alpha_k = & \frac{<r^{(k)}, r^{(k)}>}{<A\,p^{(k)}, p^{(k)}>} & \\ x^{(k+1)} = & x^{(k)} + \alpha_k\, p^{(k)} & \\ r^{(k+1)} = & r^{(k)} - \alpha_k\, A\, p^{(k)} \rightarrow \text{test} & \text{deconvergence} \\ \beta_{k+1} = & \frac{<r^{(k+1)}, r^{(k+1)}>}{<r^{(k)}, r^{(k)}>} & \\ p^{(k+1)} = & r^{(k+1)} + \beta_{k+1}\, p^{(k)} & \end{array} \right.$$

Définition :
On appelle espace de Krylov, l'espace vectoriel $\mathcal{H}_k$ défini par $\mathcal{H}_k = \mathcal{E}(r^{(0)}, A\, r^{(0)}, ..., A^{k-1}\, r^{(0)})$.
Les $r^{(i)}$ $(0 \leq i \leq k-1)$ forment une base orthogonale de cet espace.

Théorème :
La solution $x^{(k)}$ obtenue à la k^{eme} itération de l'algorithme du gradient conjugué vérifie la relation :

$$J(x^{(k)}) < J(x) \qquad \forall\, x \in x^{(0)} + \mathcal{H}_k$$

Corollaire :
L'algorithme du gradient conjugué converge en au plus n itérations, pour une matrice (n, n) (symétrique définie positive).

3.7 Gradients Conjugués Préconditionnés

Soient A et C^{-1} deux matrices symétriques définies positives, le produit $C^{-1}\,A$ n'est pas nécessairement symétrique. L'algorithme du gradient conjugué préconditionné par C s'écrit :
soit $x^{(0)}$ un vecteur donné, on calcule $r^{(0)} = b - A\,x^{(0)}$, on résout $C\,p^{(0)} = r^{(0)}$, on pose $z^{(0)} = p^{(0)}$.

$$k = 0, 1, ..., n \quad \left| \begin{array}{rcl} \alpha_k = & \frac{<r^{(k)}, z^{(k)}>}{<A\,p^{(k)}, p^{(k)}>} \\ x^{(k+1)} = & x^{(k)} + \alpha_k\,p^{(k)} \\ r^{(k+1)} = & r^{(k)} - \alpha_k\,A\,p^{(k)} \rightarrow \text{test} \quad \text{C.V.} \\ C z^{(k+1)} = & r^{(k+1)} \rightarrow z^{(k+1)} \\ \beta_{k+1} = & \frac{<r^{(k+1)}, z^{(k+1)}>}{<r^{(k)}, z^{(k)}>} \\ p^{(k+1)} = & z^{(k+1)} + \beta_{k+1}\,p^{(k)} \end{array} \right.$$

A chaque itération, on résout en plus $C\,z = r$. Il faut prendre C de manière que $C\,z = r$ soit "facilement inversible".

3.8 Comparaison des performances de méthodes itératives

Dans le tableau suivant, on rapporte les performances de certaines méthodes itératives. Le système $A\,x = b$ à résoudre correspond à un problème de conduction stationnaire 3D. La matrice A, symétrique définie positive, est héptadiagonale creuse, de taille $(128^3, 128^3)$. La taille du vecteur inconnu est d'environ $2.1\,10^6$ inconnues. Le critère d'arrêt est $||r|| < 10^{-8}$. Les calculs sont effectués sur un PC de bureau.

Méthodes "stationnaires" .

methode	*omega*	*iterations*	*cpu*	*ram*
Jacobi	1.0	36953	$90m15s00$	$188M$
GS	1.0	18471	$48m07s49$	$188M$
SOR	1.93	644	$1m35s96$	$188M$

methode	*omega*	*iterations*	*cpu*	*ram*
SSOR	1.95	323	$1m14s03$	$188M$

Méthodes "instationnaires" .

methode	*omega*	*iterations*	*cpu*	*ram*
GC		379	$1m01s05$	$220M$
GCP(J)	1.0	379	$1m05s98$	$220M$
GCP(SSOR)	1.93	39	$11s21$	$220M$

discussion :
La matrice A, héptadiagonle creuse, est symétrique définie positive. Les méthodes dédiées à ce genre de matrices doivent être plus performantes. Pour la résolution du problème, la méthode SSOR nécessite 1m14s (188M) et la méthode GC 1m1s (220M). Si on prend les GC comme référence, SSOR nécessite 21 % de temps de calcul en plus et 15 % de stockage en moins. Si on conjugue ces deux méthodes en préconditionnant les GC par SSOR, temps cpu est divisé par 5.5 alors que le stockage reste le même que celui des GC.

Pour les matrices A non symétriques, les méthodes instationnaires de type gradients conjugués ne peuvent pas être utilisées. D'autres types de méthodes ont été développées (algorithme GMRES).

4 Valeurs propres de matrices carrées

Les valeurs propres de la matrice $A \in \mathbb{C}^{n,n}$ sont les n racines de son polynôme caractéristique $P(z) = det(z\,I - A)$. L'ensemble de ces racines est appelé spectre de A, noté $Sp(A)$ ou $\lambda(A)$. On a les propriétés suivantes :

1. si $\lambda(A) = \{\,\lambda_1, ..., \lambda_n\,\}$, alors $det(A) = \lambda_1\lambda_2.....\lambda_{n-1}\lambda_n$.
2. si on définit la trace de A par $trace(A) = \sum_{i=1}^{n} a_{ii}$
   $\Longrightarrow trace(A) = \lambda_1 + \lambda_2 + ... + \lambda_n$.
3. si $\lambda \in \lambda(A)$, les vecteurs $x \neq 0$ qui satisfont à $A\,z = \lambda\,z$ sont des vecteurs propres.

Il n'existe pas de méthodes directes ou itératives qui permettent le calcul des valeurs propres en un nombre fini d'opérations. Plusieurs méthodes itératives ($k \to \infty$) permettent le calcul des valeurs propres et éventuellement des vecteurs propres. Le choix d'une méthode s'effectue selon le but recherché : une ou plusieurs valeurs propres, calcul ou pas des vecteurs propres.

4.1 Méthode de la puissance itérée

4.1.1 calcul de λ_1

Supposons que $A \in \mathbb{C}^{n,n}$ soit diagonalisable tel que $X^{-1}\,A\,X = diag(\lambda_1, ..., \lambda_n)$ avec $|\lambda_1| > |\lambda_2| \geq ..... \geq |\lambda_n|$ et $X = [x_1, ..., x_n]$.

Décomposons un vecteur quelconque $v^{(0)}$ sur la base des vecteurs propres x_i. On écrit $v^0 = a_1x_1 + a_2x_2 + ... + a_nx_n$ avec $a_1 \neq 0$ si v^0 a une composante dans la direction du vecteur propre x_1. On a alors

$$A^k\,v^{(0)} = a_1\,\lambda_1^k \left[x_1 + \sum_{j=2}^{n} \frac{a_j}{a_1} (\frac{\lambda_j}{\lambda_1})^k\,x_j \right] \Longrightarrow a_1\,\lambda_1^k x_1$$

Le vecteur $A^k\,v^{(0)}$ devient colinéaire à x_1 lorsque $k \to \infty$.
Soit alors $v^0 \in \mathbb{C}^n$ un vecteur unitaire, la méthode de la puissance itérée se base sur l'analyse précédente pour construire une suite de vecteurs qui converge vers un vecteur unitaire colinéaire à x_1, vecteur propre associé à la valeur propre λ_1, valeur propre de plus grand module.
L'algorithme est le suivant :

Soit $v^{(0)}$ donné unitaire :
Pour k=1,...,kmax
$u^{(k)} = A\,v^{(k-1)}$
$\lambda^{(k)} =< v^{(k-1)}, u^{(k)} >$
Si $|\lambda^{(k)} - \lambda^{(k-1)}| < \epsilon$ donné $\to$ stop
sinon $v^{(k)} = \frac{u^{(k)}}{||u^{(k)}||_2}$
fin de boucle.

La méthode converge si λ_1 est dominante et si $v^{(0)}$ a une composante dans la direction du vecteur propre correspondant. La rapidité de convergence dépend du rapport $|\lambda_1|/|\lambda_2|$.
On peut aussi proposer d'utiliser la norme infinie, et alors l'algorithme est :

Soit $v^{(0)}$ donné :
Pour k=1,...,kmax
$u^{(k)} = A\, v^{(k-1)}$
$m_k = \|u^{(k)}\|_\infty$
Si $|m^{(k)} - m^{(k-1)}| < \epsilon$ donné $\rightarrow$ stop
sinon $v^{(k)} = \frac{u^{(k)}}{m_k}$
fin de boucle.

On a ainsi $lim_{k\rightarrow\infty}\, u^k = x_1$, et $lim_{k\rightarrow\infty}\, m_k = |\lambda_1|$. Si on veut un vecteur propre normé en norme 2, on prend $v/||v||_2$.

Définition :
On appelle espace de Krylov, l'espace vectoriel $\mathbb{K}$ défini par

$$\mathbb{K} = \mathbb{E}(v^{(0)}, v^{(1)} = Av^{(0)}, ..., v^{(k-1)} = A^{k-1}v^{(0)})$$

Les $v^{(i)}$ $(0 \leq i \leq k-1)$ forment une base orthogonale de cet espace.

4.1.2 calcul de λ_2

Les valeurs propres de la matrice A^t sont les mêmes que celles de A.
Soit alors y_1 le vecteur propre de A^t correspondant à λ_1, qu'on calcule en appliquant la méthode de la puissance itérée à la matrice A^t. On a $A\, x_1 = \lambda_1\, x_1$ et $A^t\, y_1 = \lambda_1\, y_1$. Si A est symétrique, on a $y_1 = x_1$.
Par déflation, on forme la matrice A_1 définie par

$$A_1 = A - \frac{\lambda_1\, x_1\, y_1^t}{y_1^t\, x_1}.$$

On a alors : $A_1 x_1 = Ax_1 - \frac{\lambda_1\, x_1\, y_1^t\, x_1}{y_1^t\, x_1} = Ax_1 - \lambda_1\, x_1 = 0$. Donc 0 est valeur propre de A_1 associée au vecteur propre x_1.
Pour les autres vecteurs propres, on s'appuie sur le fait que $y_1^t x_i = 0$, pour conclure que $A_1 x_i = Ax_i - 0 = \lambda_i\, x_i$.
On voit que les valeurs propres de la nouvelle matrice A_1 sont $\lambda_2, \lambda_3, ...\lambda_n$, associées aux vecteurs propres $x_2, x_3, ..., x_n$ de la matrice A, et la valeur propre de plus petit module est 0, associée à x_1.
On applique la méthode de la puissance itérée à A_1 pour obtenir λ_2 tel que $|\lambda_2| > |\lambda_3| \geq |\lambda_4|..... \geq |\lambda_n|$, et on a

$$A_1^k\, v^1 \approx a_2\, \lambda_2\, x_2 \Longrightarrow \lambda_2 \text{ et } x_2 \quad \text{avec} \quad |\lambda_2 - \lambda^{(k)}| = \mathcal{O}\left(|\frac{\lambda_3}{\lambda_2}|^k\right).$$

4.1.3 calcul de $\lambda_3, ..., \lambda_n$

On cherche y_2 puis $A_2 = A_1 - (\lambda_2\, x_2\, y_2^t)/(y_2^t\, x_2) \Longrightarrow x_3$ et λ_3, etc ... Les valeurs propres calculées sont entachées d'erreurs d'arrondi. On peut accélérer la convergence de cette méthode par le décalage des valeurs propres (méthode de Rayleigh). Soit a une constante, si $A\,x = \lambda\,x \Longrightarrow (A + a\,I)\,x = (\lambda + a)\,x$. On rajoute et on retranche un même nombre à chaque élément diagonal.

4.2 Méthode de la puissance itérée inverse

Les valeurs propres μ de la matrice A^{-1} sont les inverses de celles de A ($\mu = 1/\lambda$), associées aux mêmes vecteurs propres x.
On applique la méthode de la puissance itérée à A^{-1}, ce qui permet d'obtenir la valeur propre de A^{-1} ayant le plus grand module, et donc celle de A ayant le plus petit module et le vecteur propre associé car $\max_j |1/\lambda_j| = \min_j |\lambda_j|$. Dans la pratique, on ne calcule pas A^{-1} mais on met A sous la forme $A = L\,U$ (par exemple) et on résout successivement les systèmes $A\,u_{(k)} = v_{(k-1)}$.

L'algorithme est alors :
Soit $v^{(0)}$ donné unitaire :
Pour k=1,...,kmax
$Au^{(k)} = v^{(k-1)} \rightarrow u^{(k)}$ par résolution du système
$\mu^{(k)} =< v^{(k-1)}, u^{(k)} >$
Si $|\mu^{(k)} - \mu^{(k-1)}| < \epsilon$ donné $\rightarrow$ stop
sinon $v^{(k)} = \frac{u^{(k)}}{||u^{(k)}||_2}$
fin de boucle.
$\lambda_n = 1/\mu$

4.3 Méthode de Jacobi

Elle s'applique aux matrices symétriques réelles. Soit $\theta \in \mathbb{R}$, on définit une matrice de rotation $B(p, q, \theta) \in \mathbb{R}^{n,n}$ $(1 \leq p < q \leq n)$ tel que $B(p, q, -\theta) = B^t(p, q, \theta)$ où $B(p, q, \theta)$ est définie par :

$$\begin{bmatrix}
1 & & & & & & & & & & & & & & \\
& ... & & & & & & & & & & & & & \\
& & ... & & & & & & & & & & & & \\
& & & 1 & & & & & & & & & & & \\
& & & & cos(\theta) & & & & & & sin(\theta) & & & & \\
& & & & & 1 & & & & & & & & & \\
& & & & & & ... & & & & & & & & \\
& & & & & & & ... & & & & & & & \\
& & & & & & & & 1 & & & & & & \\
& & & & -sin(\theta) & & & & & & cos(\theta) & & & & \\
& & & & & & & & & & & 1 & & & \\
& & & & & & & & & & & & ... & & \\
& & & & & & & & & & & & & ... & \\
& & & & & & & & & & & & & & 1
\end{bmatrix}$$

On calcule $\tilde{A} = B^t(p,q,\theta)\, AB(p,q,\theta)$.
On choisit p et q tel que pour $1 \leq p \neq q \leq n, \quad |a_{pq}| = \max_{1\leq i\neq j\leq n} |a_{ij}|$,
On pose $tg(2\theta) = 2\, A_{pq}/(A_{pp} - A_{qq})$.
La matrice $\tilde{A}$ vérifie $\tilde{A}_{pq} = \tilde{A}_{qp} = 0 \quad (p \neq q)$ par construction.
On reproduit cette opération k fois : les "zéros" de la matrice $A^{(k)}$ ne sont pas conservés sur la matrice $A^{(k+1)}$, mais les termes hors diagonale deviennent progressivement petits.

algorithme :
A symétrique $\Longrightarrow A^{(0)} = A$
Pour $k = 0, 1, ..., k_{max}$
On identifie p, q tels que $|a_{pq}^{(k)}| = \max_{i\leq i\neq j\leq n} a_{ij}^{(k)} \quad (1 \leq p < q \leq n)$.
$\rightarrow$ on pose $tg(2\,\theta) = a_{pq}^{(k)}/(a_{pp}^{(k)} - a_{qq}^{(k)}) \qquad \theta < \pi/4$
Si $a_{pp}^{(k)} = a_{qq}^{(k)}$, on prend $\theta = \pi/4$
On calcule la matrice $A^{(k+1)} = B^t(p,q,\theta)\, A^{(k)} B(p,q,\theta)$ (Les opérations ne s'effectuent que sur les lignes p, q et sur les colonnes p, q.)
Test de convergence : soit $|a_{pp}^{(k)}| = min_{1\leq i\leq n} |a_{ii}^{(k)}| > 0 \quad ; \quad A_{ii}^{(k)} \neq 0$
si $a_{pq}^{(k)}/a_{pp}^{(k)} < \epsilon \rightarrow$ convergence.

remarque :
Si A est une matrice symétrique, non diagonale, alors la suite $A^{(k)}$ converge vers une matrice diagonale D semblable à A. On obtient :

$$A^{(k)} = D^{(k)} + E_{diag=0}^{(k)} \qquad \text{avec} \quad ||E_{ij}^{(k)}|| < \epsilon\, ||D_{ij}^{(k)}||$$

Si λ est une valeur propre de $A^{(k)}$, alors $A^{(k)}\, x = \lambda\, x, \quad ||x|| = 1$.
$\Longrightarrow ||D^{(k)}\, x - \lambda\, x|| < \epsilon\, ||D^{(k)}||$.
Si ϵ est très petit $\Longrightarrow D^{(k)}\, x \simeq \lambda\, x$ et $\lambda \cong$ valeur propre de A.

4.4 Méthode de Householder

La méthode de transformation de Householder permet de trouver une matrice B semblable à A. Si A n'est pas symétrique, B est de type Hessenberg. Le calcul des valeurs propres de B est plus facile que celui de A.
Si $u \in \mathbb{R}^n$, on prend $\tilde{u} \in \mathbb{R}^{n-r}$ un vecteur tel que $||\tilde{u}||_2 = 1$. On définit le vecteur u par :

$$u = \left|\begin{array}{c} 0 \\ \hline \tilde{u} \end{array}\right| \quad \text{On calcule} \quad u\, u^t = \left|\begin{array}{c} 0 \\ \hline \tilde{u} \end{array}\right| (0, \tilde{u}) = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \tilde{u}\, \tilde{u}^t \end{array}\right].$$

$$\text{Soit} \quad I_n = \left[\begin{array}{c|c} I_r & 0 \\ \hline 0 & I_{n-r} \end{array}\right] \quad \text{alors,} \quad I - 2\, \tilde{u}\, \tilde{u}^t = \left[\begin{array}{c|c} I_r & 0 \\ \hline 0 & I_{n-r} - 2\, \tilde{u}\, \tilde{u}^t \end{array}\right],$$

$$\text{ou} \quad H = \left[\begin{array}{c|c} I_r & 0 \\ \hline 0 & \widetilde{H} \end{array}\right]$$

On définit la matrice élémentaire de Householder d'ordre $(n-r)$ par H avec $\widetilde{H} \in \mathbb{R}^{n-r,n-r}$, et $\widetilde{H} = I - 2\, \tilde{u}\, \tilde{u}^t$. Le produit $H\, A\, H$ est une matrice semblable à A définie par :

$$H\, A\, H = \left[\begin{array}{c|c} I_r & 0 \\ \hline 0 & \widetilde{H} \end{array}\right] \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \left[\begin{array}{c|c} I_r & 0 \\ \hline 0 & \widetilde{H} \end{array}\right] = \left[\begin{array}{c|c} A_{11} & A_{12}\, \widetilde{H} \\ \hline \widetilde{H}\, A_{21} & \widetilde{H}\, A_{22}\, \widetilde{H} \end{array}\right]$$

Soient a et b deux vecteurs quelconques $\in \mathbb{R}^{n-r}$, non colinéaires avec $||b||_2 = 1$. On cherche $\tilde{u} \in \mathbb{R}^{n-r}$ tel que $||u||_2 = 1$ et

$$(I - 2\,\tilde{u}\,\tilde{u}^t)\,a = \alpha\,b$$

$$\Longrightarrow 2\,\tilde{u}\,(\tilde{u}^t\,a) = -\alpha\,b + a \Longrightarrow 2\,|\tilde{u}^t\,a|\,||\tilde{u}||_2 = ||\alpha\,b - a||_2 \;\text{ avec }\; ||\tilde{u}||_2 = 1.$$

$$\Longrightarrow |\tilde{u}^t\,a| = ||\alpha\,b - a||_2/2,$$

on choisit $\tilde{u}^t\,a = ||\alpha\,b - a||_2/2 \Longrightarrow \tilde{u} = (-\alpha\,b + a)/||\alpha\,b - a||_2$. Soit

$$\widetilde{H} = I - \frac{2\,(a - \alpha\,b)\,(a - \alpha\,b)^t}{2\,(\tilde{u}^t\,a)^2} = I - \frac{(a - \alpha\,b)\,(a - \alpha\,b)^t}{\alpha^2 - \alpha\,(a^t\,b)}$$

On a : $\alpha = \pm||a||$. On prend le signe qui donne la plus grande valeur de $[\alpha^2 - \alpha\,(a^t\,b)]$.
On pose $v = (a - \alpha\,b)$ défini par $v_r = a_r - \alpha$ et $v_i = a_i$ $(r + 1 \leq i \leq n)$.
On pose $\beta = \alpha^2 - \alpha\,a_r \Longrightarrow \widetilde{H} = I - (v\,v^t)/\beta$.

procédé de transformation :
$A^{(1)} = A$, on prend $r = 1 \Longrightarrow A^{(2)} = H^{(1)}\,A^{(1)}\,H^{(1)} \Longrightarrow A^{(3)} = H^{(2)}\,A^{(2)}\,H^{(2)}, ...$
$A^{(k+1)} = H^{(k)}\,A^{(k)}\,H^{(k)}, ..., A^{(n-2)} = H^{(n-3)}\,A^{(n-3)}\,H^{(n-3)},$

$$A^{(n-1)} = H^{(n-2)}\,A^{(n-2)}\,H^{(n-2)} = H^{(n-2)}\,H^{(n-3)}.....H^{(2)}\,H^{(1)}\,A\,H^{(1)}\,H^{(2)}.....H^{(n-3)}\,H^{(n-2)}$$

Soit $A^{(n-1)} = Q\,A\,Q^t$ où Q est une matrice orthogonale de dimension n.
Au bout de $(n-2)$ transformations, on obtient la matrice $A^{(n-1)}$ de type Hessenberg semblable à A. Si A est symétrique, alors $A^{(n-1)}$ est tridiagonale symétrique.
A l'étape de transformation r, la matrice $A^{(r)}$ est de la forme :

$$A^{(r)} = \left[\begin{array}{ccccccc|cccccc}
a_{1,1}^{(1)} & a_{1,2}^{(2)} & a_{1,3}^{(3)} & .... & .... & a_{1,r}^{(r-1)} & a_{1,r+1}^{(r)} & .... & .... & .... & .... & a_{1,n}^{(r)} \\
a_{2,1}^{(2)} & a_{2,2}^{(2)} & a_{2,3}^{(3)} & .... & .... & .... & a_{2,r}^{(-1r)} & a_{2,r+1}^{(r)} & .... & .... & .... & a_{2,n}^{(r)} \\
 & a_{3,2}^{(3)} & a_{3,3}^{(3)} & .... & .... & .... & a_{3,r}^{(r-1)} & a_{3,r+1}^{(r)} & .... & .... & .... & a_{3,n}^{(r)} \\
 & & .... & .... & .... & .... & .... & .... & .... & .... & .... & .... \\
 & & & .... & .... & .... & .... & .... & .... & .... & .... & .... \\
 & & & & a_{r-1,r-2}^{(r-1)} & a_{r-1,r-1}^{(r-1)} & a_{r-1,r}^{(r)} & a_{r-1,r+1}^{(r)} & .... & .... & .... & a_{r-1,n}^{(r)} \\
 & & & & & a_{r,r-1}^{(r)} & a_{r,r}^{(r)} & a_{r,r+1}^{(r)} & .... & .... & .... & a_{r,n}^{(r)} \\
\hline
 & & & & & & a_{r+1,r}^{(r)} & a_{r+1,r+1}^{(r)} & .... & .... & .... & a_{r+1,n}^{(r)} \\
 & & & & & & .... & .... & .... & .... & .... & .... \\
 & & & & & & .... & .... & .... & .... & .... & .... \\
 & & & & & & .... & .... & .... & .... & .... & .... \\
 & & & & & & a_{n,r}^{(r)} & a_{n,r+1}^{(r)} & .... & .... & .... & a_{n,n}^{(r)}
\end{array}\right]$$

A l'étape de transformation $(n-1)$, la matrice $A^{(n-1)}$ est de la forme :

$$A^{(n-1)} = \begin{bmatrix} a_{1,1}^{(1)} & a_{1,2}^{(2)} & .... & .... & .... & .... & .... & .... & .... & .... & a_{1,n-1}^{(n-1)} & a_{1,n}^{(n-1)} \\ a_{2,1}^{(2)} & a_{2,2}^{(2)} & .... & .... & .... & .... & .... & .... & .... & .... & a_{2,n-1}^{(n-1)} & a_{2,n}^{(n-1)} \\ & .... & .... & .... & .... & .... & .... & .... & .... & .... & .... & .... \\ & & .... & .... & .... & .... & .... & .... & .... & .... & .... & .... \\ & & & .... & .... & .... & .... & .... & .... & .... & .... & .... \\ & & & & .... & .... & .... & .... & .... & .... & .... & .... \\ & & & & & .... & .... & .... & .... & .... & .... & .... \\ & & & & & & .... & .... & .... & .... & .... & .... \\ & & & & & & & .... & .... & .... & .... & .... \\ & & & & & & & & .... & .... & .... & .... \\ & & & & & & & & & .... & a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ & & & & & & & & & & a_{n,n-1}^{(n-1)} & a_{n,n}^{(n-1)} \end{bmatrix}$$

4.5 Méthode $Q\,R$

Le point de départ de cette méthode est une matrice de Hessenberg A_1 obtenue après $(n-1)$ transformations de Householder.

On factorise alors $A_1 = Q_1\,R_1$, où Q_1 est orthogonale, R_1 est triangulaire supérieure à diagonale unité. La décomposition $A_1 = Q_1\,R_1$ est obtenue après $(n-1)$ transformations de type Householder.
On calcule $A_2 = R_1\,Q_1 = Q_1^t\,Q_1\,R_1\,Q_1 = Q_1^t\,A_1\,Q_1$, A_2 est semblable à A_1.
On décompose de nouveau $A_2 = Q_2\,R_2$, et on calcule $A_3 = R_2\,Q_2 = Q_2^t\,A_2\,Q_2$, etc ...

On peut montrer que la méthode revient à effectuer le produit des puissances de la matrice A par des sous-matrices de tailles de plus en plus petites. Les matrices A_k convergent vers une matrice triangulaire supérieure (les termes sous la diagonale deviennent négligeables), avec les valeurs propres ordonnées sur la diagonale de la plus grande à la plus petite (en $|.|$).

Les valeurs propres sont $\lambda_i \in Sp(A) = lim_{k\to\infty} a_{ii}^{(k+1)}$. Les matrices $A^{(k)}$ et $A^{(k+1)}$ sont semblables.

Avantage : utilisation de matrices Q très bien conditionnées
Inconvénient : calcul de Q, de R, des produits $Q\,R$ et $R\,Q$.

5 Interpolation polynômiale

On est souvent amené à réaliser une interpolation polynômiale lorsqu'une fonction est soit une fonction donnée analytiquement mais difficile à manipuler, soit une fonction tabulée connue seulement pour certaines valeurs de x (par exemple une fonction mesurée expérimentalement). Il existe quatre types de méthodes permettant de réaliser une interpolation polynômiale :

a) méthodes de collocation : la fonction interpolée $F(x)$ coïncide avec $f(x)$ aux points x_j où la fonction $f(x_j)$ est connue : $F(x_j) = f(x_j)$
b) polynômes osculatoires : en plus de la coincidence de $F(x_j)$ et de $f(x_j)$, il y a coïncidence en x_j de leurs m premières dérivées
c) moindres carrés : la fonction d'interpolation $G(x)$ ne passe pas par les points $[x_j, f(x_j)]$ mais entre ces points. Le critère est que $S = \sum_{j=1}^{n}[F(x_j) - f(x_j)]^2$ soit minimale.
d) mini-max : la fonction d'interpolation $M(x)$ passe entre les points $[x_j, f(x_j)]$. Le critère est que la distance à $M(x)$ du point le plus éloigné, soit la plus petite possible.

5.1 Méthodes d'interpolation par collocation

5.1.1 Forme polynômiale développée en puissances de x

Soit $f(x)$ connue aux points x_j par $f_j = f(x_j),\ (0 \leq j \leq n)$, on approxime $f(x)$ par le polynôme $P_n(x)$ de degré n, passant par les points (x_j, f_j). Ce polynôme est unique. Il peut s'écrire :

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$$

avec : $f_j = a_0 + a_1 x_j + a_2 x_j^2 + ... + a_n x_j^n,\ (0 \leq j \leq n)$. Les a_j sont solution d'un système $(n+1, n+1)$ qui utilise la matrice de Vandermonde d'ordre n.

$$\begin{bmatrix} 1 & x_0 & x_0^2 & .... & x_0^{n-1} & x_0^n \\ 1 & x_1 & x_1^2 & .... & x_1^{n-1} & x_1^n \\ .... & .... & .... & .... & .... & .... \\ .... & .... & .... & .... & .... & .... \\ .... & .... & .... & .... & .... & .... \\ .... & .... & .... & .... & .... & .... \\ 1 & x_n & x_n^2 & .... & x_n^{n-1} & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ .... \\ .... \\ .... \\ .... \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ .... \\ .... \\ .... \\ .... \\ f_n \end{bmatrix}$$

5.1.2 Forme polynômiale de Lagrange

Polynômes de degré n :
On connait $(n+1)$ points distincts $x_0, x_1, ..., x_n$ dans $[a, b]$ et les valeurs $f_j = f(x_j)$, $0 \leq j \leq n$. On cherche à construire le polynôme $P_n(x)$ de degré $\leq n$ tel que $P_n(x_j) = f_j$, $0 \leq j \leq n$. On définit les polynômes de Lagrange notés $L_i(x)$ de degré $\leq n$ tels que $L_i(x_j) = \delta_{ij}, 0 \leq i, j \leq n$:

$$L_i(x) = \frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)} = \prod_{j=0, j\neq i}^{n} \frac{(x-x_j)}{(x_i-x_j)}$$

Le polynôme $P_n(x)$ défini alors par :

$$P_n(x) = \sum_{i=0}^{n} f(x_i) L_i(x) \ = \sum_{i=0}^{n} \left[\prod_{j=0, j\neq i}^{n} \frac{(x-x_j)}{(x_i-x_j)} \right] f_i,$$

$P_n(x)$ est le polynôme d'interpolation de Lagrange de la fonction f.
Remarques :
— La formule de Lagrange contient explicitement les f_i.

— Pour $n = 1$, $P_1(x)$ passe par (x_0, f_0) et (x_1, f_1), d'où

$$P_1(x) = f(x_0)\frac{x - x_1}{x_0 - x_1} + f(x_1)\frac{x - x_0}{x_1 - x_0}$$

C'est la droite qui passe par les deux points (x_0, f_0) et (x_1, f_1).

— Pour $n = 2$, $P_2(x)$ est l'équation de la parabole qui passe par les trois points (x_0, f_0), (x_1, f_1) et (x_2, f_2) et on a :

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}, L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

$$P_2(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + f(x_2)L_2(x)$$

— Pour la programmation : On ne développe pas la forme analytique du polynôme $P(x)$. En pratique, on calcule pour chaque valeur de x souhaitée la valeur de $P(x)$ en calculant la somme des produits qui figurent dans l'expression.

Evaluation de l'erreur de la formule de Lagrange :
On a construit pour la fonction $f(x)$ le polynôme de Lagrange $P_n(x)$ qui prend en $x_0, x_1, ..., x_n$ les valeurs données $f_0 = f(x_0), ..., f_n = f(x_n)$. Quelle est la valeur du reste $R_n(x) = f(x) - P_n(x)$ pour les autres valeurs de x ?
Supposons que, dans $[a, b]$, qui contient les x_j, $f(x)$ possède des dérivées $f', f'', ..., f^{(n+1)}(x)$.
Posons $u(x) = f(x) - P_n(x) - k\pi_{n+1}(x)$, avec $\pi_{n+1}(x) = (x - x_0)(x - x_1).....(x - x_n)$.
Il est évident que $u(x)$ possède $(n+1)$ racines $x_0, x_1, ..., x_n$. Soit $\bar{x}$ arbitrairement choisi dans $[a, b]$, différent des x_i, et fixons k pour que $\bar{x}$ soit également racine de $u(x)$:
$u(\bar{x}) = 0 = f(\bar{x}) - P_n(\bar{x}) - k\,\pi_{n+1}(\bar{x}) \Longrightarrow k = \frac{f(\bar{x}) - P_n(\bar{x})}{\pi_{n+1}(\bar{x})}$.
Ainsi dans $[a, b]$, $u(x)$ s'annule en n points intérieurs + les 2 extrémités. Le théorème de Rolle entraine que $u'(x)$ possède au moins $(n + 1)$ racines sur le segment $[a, b]$. De même $u''(x)$ est nulle au moins n fois sur $[a, b]$, ... , la dérivée $u^{(n+1)}(x)$ possède au moins un zéro : $\exists \xi \in [a, b]$ tel que $u^{(n+1)}(\xi) = 0$.
Or $P_n^{(n+1)}(x) = 0$ et $\pi_{n+1}^{(n+1)}(x) = (n + 1)! \Longrightarrow u^{(n+1)}(x) = f^{(n+1)}(x) - k\,(n + 1)!$.
Donc $\exists \xi \in [a, b]$ tel que $f^{(n+1)}(\xi) - k\,(n + 1)! = 0$
$\Longrightarrow k = [f(\bar{x}) - P_n(\bar{x})]/\pi_{n+1}(\bar{x}) = f^{(n+1)}(\xi)/(n + 1)! \Longrightarrow f(\bar{x}) - P_n(\bar{x}) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(\bar{x})$.
Ce raisonnement pouvant être reproduit pour tout $\bar{x}$, on a :

$$R_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!}\pi_{n+1}(x)$$

5.2 Polynômes osculatoires

5.2.1 Interpolation d'Hermite

On souhaite construire un polynôme passant par les $n+1$ points $(x_j, f(x_j))$ et de dérivée coïncidant aux $n+1$ points x_j avec les dérivées $f'(x_j)$ imposées. On a donc $2n+2$ équations, et le polynôme recherché est de degré $2n + 1$.
On construit le polynôme d'Hermite en utilisant les polynômes de Lagrange et leurs propriétés. On obtient :

$$H(x) = \sum_{j=0}^{n} U_j(x)\, f_j + \sum_{j=0}^{n} V_j(x)\, f'_j$$

où f_j et f'_j sont les valeurs de la fonction donnée et de sa dérivée en x_j. Les fonctions $U_j(x)$ et $V_j(x)$ sont définis par :

$$U_j(x) = [1 - 2L'_j(x_j)(x - x_j)][L_j(x)]^2 \qquad ; \qquad V_j(x) = (x - x_j)[L_j(x)]^2$$

$$\text{l'erreur est :} \quad f(x) - P(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \left[\prod_{j=0}^{n}(x - x_j)\right]^2$$

5.2.2 Les splines cubiques

Il s'agit d'une méthode d'interpolation qui respecte la collocation, et permet d'obtenir une courbe "lissée". Supposons connus les $f_j = f(x_j)$ $(0 \leq j \leq n)$. On suppose qu'on ne connait pas f'_j et f''_j.
L'interpolation $g(x)$ est un polynôme d'ordre 3 par morceaux définis entre x_j et x_{j+1}, tel que : $g(x) = a_0^j + a_1^j\, x + a_2^j\, x^2 + a_3^j\, x^3$. Pour déterminer $a_0^j, a_1^j, a_2^j, a_3^j$, on impose les 4 conditions suivantes :
(i) collocation $g(x_j) = f(x_j)$, (ii) $g(x_{j+1}) = f(x_{j+1})$, (iii) continuité de dg/dx en x_j et (iv) continuité de d^2g/dx^2 en x_j.
Or d^2g/dx^2 vaut $2\, a_2^j + 6\, a_3^j x$, et est donc linéaire en x. Soit pour $x \in [x_j, x_{j+1}]$:

$$\frac{g''(x) - g''(x_j)}{(x - x_j)} = \frac{g''(x_{j+1}) - g''(x_j)}{(x_{j+1} - x_j)} \Longrightarrow g''(x) = g''(x_j) + \frac{x - x_j}{x_{j+1} - x_j}[g''(x_{j+1}) - g''(x_j)]$$

On intègre deux fois cette relation avec $\Delta x_j = (x_{j+1} - x_j)$ et $g''(x_j) = g''_j$. Soit :

$$g'(x) = (x - x_j)g''_j + \frac{(x - x_j)^2}{2\Delta x_j}(g''_{j+1} - g''_j) + A$$

$$g(x) = \frac{(x - x_j)^2}{2}g''_j + \frac{(x - x_j)^3}{6\Delta x_j}(g''_{j+1} - g''_j) + Ax + B$$

en imposant $g(x_j) = f(x_j) = f_j$ et $g(x_{j+1}) = f(x_{j+1}) = f_{j+1}$, ainsi que la continuité de $g'(x)$ en x_j, on obtient un système tridiagonal à résoudre par rapport à g''_j en prenant $g''_0 = g''_n = 0$.

$$\Delta x_{j-1}\, g''_{j-1} + 2(x_{j+1} - x_{j-1})\, g''_j + \Delta x_j\, g''_{j+1} = 6\left[\frac{f_{j+1} - f_j}{\Delta x_j} - \frac{f_j - f_{j-1}}{\Delta x_{j-1}}\right]$$

5.3 Méthode des moindres carrés

On se donne une fonction f dont on connait la valeur en $(n+1)$ points distincts $(0 \leq j \leq n)$.
On cherche une approximation de f par un polynôme $G_m(x)$ de degré m qui minimise $S = \sum_{j=0}^{n} [G_m(x_j) - f(x_j)]^2$.
Le polynôme G_m est défini par $G_m(x) = \sum_{i=0}^{m} a_i\, x^i \quad (a_i \in \mathbb{R},\, 0 \leq m \leq n)$.
On pose $S = \sum_{j=0}^{n} \left[\sum_{i=0}^{m} a_i\, x_j^i - f(x_j)\right]^2$. Le minimum est atteint pour les $a_j \quad (0 \leq j \leq n)$ qui verifient $\partial/\partial a_l = 0$, $(0 \leq l \leq m)$. Or on a :

$$\frac{\partial S}{\partial a_l} = 2\sum_{j=0}^{n}\left[\sum_{i=0}^{m} a_i\, x_j^i - f(x_j)\right] x_j^l = 0, \quad (0 \leq l \leq m)$$

$$\text{soit } \sum_{i=0}^{m} a_i \left(\sum_{j=0}^{n} x_j^{i+l} \right) = \sum_{j=0}^{n} f(x_j) x_j^l$$

On en déduit que pour trouver un polynôme de degré m qui approche au sens des moindres carrés une fonction connue en $(n+1)$ points distincts avec $m < n$ il faut résoudre un système linéaire de degré $(m+1)$ qui s'écrit :

$$\begin{bmatrix} n+1 & \sum x_j & \sum x_j^2 & .... & \sum x_j^{m-1} & \sum x_j^m \\ \sum x_j & \sum x_j^2 & \sum x_j^3 & .... & \sum x_j^m & \sum x_j^{m+1} \\ \sum x_j^2 & \sum x_j^3 & \sum x_j^4 & .... & \sum x_j^{m+1} & \sum x_j^{m+2} \\ .... & .... & .... & .... & .... & .... \\ .... & .... & .... & .... & .... & .... \\ .... & .... & .... & .... & .... & .... \\ .... & .... & .... & .... & .... & .... \\ \sum x_j^m & \sum x_j^{m+1} & \sum x_j^{m+2} & .... & \sum x_j^{2m-1} & \sum x_j^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ .... \\ .... \\ .... \\ .... \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{j=0}^n f_j \\ \sum_{j=0}^n x_j\, f_j \\ \sum_{j=0}^n x_j^2\, f_j \\ .... \\ .... \\ .... \\ .... \\ \sum_{j=0}^n x_j^m\, f_j \end{bmatrix}$$

La matrice A est inversible, mais a un mauvais conditionnement. Les résultats pour $m \geq 10$ ne sont pas significatifs. En général, on prend $m = 1$ (droite), ou $m = 2$ (parabole).

5.4 Polynômes mini-max

Soit $h_j = h(x_j) = M(x_j) - f(x_j)$ et soit H la plus grande de ces erreurs en valeur absolue. Le polynôme mini-max est le polynôme pour lequel H est la plus petite possible.
La méthode de "substitution" est un algorithme pour trouver $M(x)$ en s'appuyant sur la propriété d'égale erreur. On choisit un sous-ensemble initial de points (x_j, f_j). On trouve un polynôme d'égale erreur correspondant à ces données. Si l'erreur maximum portée par ce polynôme est l'erreur H, ce polynôme est le polynôme $M(x)$ cherché. Si ce n'est pas le cas, on remplace un point de l'ensemble par un point extérieur et on recommence le processus. On démontre la convergence vers $M(x)$.

5.4.1 Droite mini-Max

On cherche $M(x) = a + b\,x$ et on note $h_i = M(x_i) - f_i$. Pour déterminer une droite d'égale erreur, il faut trouver a, b, h tels que $M(x_i) - f(x_i) = \pm h$:

$$\begin{bmatrix} 1 & x_1 & -1 \\ 1 & x_2 & 1 \\ 1 & x_3 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ h \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

On calcule $h_j = M(x_j) - f_j$ dans les autres points $(0 \leq j \neq i \leq n)$. Si tous les $|h_j| \leq |h|$, alors ce polynôme est bien le polynôme mini-max de $f(x_j)$. Sinon, on choisit un autre point pour remplacer un des 3 points i, et on recommence....

5.4.2 Parabole mini-Max

Soit $M(x) = a + bx + cx^2$ et soit $h_i = M(x_i) - f(x_i)$ les erreurs en 4 points donnés. On démontre qu'il existe une parabole unique d'égale erreur tel que $h_1 = h$, $h_2 = -h$, $h_3 =$

h, $h_4 = -h$. A partir de la relation $M(x_i) - f(x_i) = \pm h$, on forme le système suivant qui permet de déterminer les coefficients a, b, c, et h :

$$\begin{bmatrix} 1 & x_1 & x_1^2 & -1 \\ 1 & x_2 & x_2^2 & 1 \\ 1 & x_3 & x_3^2 & -1 \\ 1 & x_4 & x_4^2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ h \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

On calcule $h_j = M(x_j) - f_j$ dans les autres points ($0 \leq j \neq i \leq n$). Si tous les $|h_j| \leq |h|$, alors ce polynôme est bien le polynôme mini-max de $f(x_j)$. Sinon, on choisit un autre point pour remplacer un des 4 points i, et on recommence....

5.4.3 Polynômes mini-Max de degré $m > 2$

De manière similaire aux cas de la droite et de la parabole, on prend $(m + 2)$ points et on détermine $M(x)$, on calcule l'erreur dans les autres points si $H > |h|$. On recommence jusqu'à trouver le bon polynôme $M(x)$ de degré m.

6 Dérivation numérique

Connaissant une fonction f en un ensemble fini de points x_j ($0 \leq j \leq n$), on construit un schéma permettant l'approximation de $f'(x)$ et de $f''(x)$.

6.1 Utilisation des polynômes de Lagrange

La fonction $f(x)$ est approchée par le polynôme de Lagrange $F(x)$ défini par
$F(x) = \sum_{j=0}^{n} f(x_j)\, L_j(x)$ avec $L_j(x) = \prod_{0 \leq i \leq n (i \neq j)} \frac{(x - x_i)}{(x_j - x_i)}$.
On a $f(x) = F(x) + E \Longrightarrow f'(x) = F'(x) + E'(x)$.

6.1.1 Dérivée centrée à 3 points

Dans ce cas $n = 2 \Longrightarrow f(x_0), f(x_1), f(x_2)$ connues.

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \quad L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}, \quad L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

$$F'(x) = f(x_0)\, \frac{2\,x - (x_1 + x_2)}{(x_0 - x_1)(x_0 - x_2)} + f(x_1)\, \frac{2\,x - (x_0 + x_2)}{(x_1 - x_0)(x_1 - x_2)} + f(x_2)\, \frac{2\,x - (x_0 + x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Si on prend $h > 0$, $x_0 = x_1 - h$, $x_2 = x_1 + h$, on a :

$$F'(x) = f(x_0)\, \frac{2\,(x - x_1) - h}{2\,h^2} - f(x_1)\, \frac{2\,(x - x_1)}{h^2} + f(x_2)\, \frac{2\,(x - x_1) + h}{2\,h^2}$$

$$\text{Soit} \quad F'(x_1) = \frac{f(x_2) - f(x_0)}{2\,h} \quad \text{ou} \quad f'(x) \cong \frac{f(x + h) - f(x - h)}{2\,h}$$

6.1.2 Dérivée seconde à 3 points

On dérive l'expression précédente de la dérivée du polynôme de Lagrange, pour 3 points équidistants :

$$F''(x) = f(x_0)\,\frac{2}{2h^2} - f(x_1)\,\frac{2}{h^2} + f(x_2)\,\frac{2}{2\,h^2}$$

$$\text{Soit} \quad F''(x_1) = \frac{f(x_0) - 2\,f(x_1) + f(x_2)}{h^2}$$

$$\Rightarrow f''(x) \cong \frac{f(x+h) - 2\,f(x) + f(x-h)}{h^2}$$

6.2 Utilisation des développements de Taylor

Si les points x_j où la fonction $f(x)$ est connue sont équidistants, on introduit le pas constant $h = x_j - x_{j-1}$.
Rappel des diverses formes utilisées de développements de Taylor de $f(x+h)$ au point x :
$f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + ... + \frac{h^{n-1}}{(n-1)!}f^{(n-1)}(x) + \frac{h^n}{n!}f^{(n)}(x) + ...,$
ou $f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + ... + \frac{h^{n-1}}{(n-1)!}f^{(n-1)}(x) + \frac{h^n}{n!}f^{(n)}(\xi)$ avec $\xi \in]x, x+h[$,
ou (forme la plus utilisée dans ce cours) :
$\mathbf{f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2}f''(\xi) + \frac{h^3}{6}f'''(x) + ... + \frac{h^{n-1}}{(n-1)!}f^{(n-1)}(x) + \mathcal{O}(h^n)}$,
On dit que le développement est effectué jusqu'à l'ordre n en h.
ou $f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2}f''(\xi) + \frac{h^3}{6}f'''(x) + ... + \frac{h^{n-1}}{(n-1)!}f^{(n-1)}(x) + o(h^{n-1})$.
D'une façon générale, on souhaite augmenter l'ordre d'approximation des schémas utilisés le plus possible.

6.2.1 Différences progressives (ou à droite ou avales)

Il s'agit d'exprimer les dérivées en x en utilisant les données de f en des points situés à droite de x.

a) Dérivées à l'ordre 1 en h

Calcul de f'_j
On écrit le développement de Taylor de $f(x+h)$ en x, et on en déduit une approximation de $f'(x)$:

$$f(x+h) = f(x) + h\,f'(x) + \mathcal{O}(h^2)$$

$$\Longrightarrow f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{1}{h}\mathcal{O}(h^2) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h)$$

$$\text{soit} \quad f'_j \simeq \frac{f_{j+1} - f_j}{h}$$

La formule est équivalente à prendre la dérivée au point x du polynôme de Lagrange passant par les 2 points x et $x+h$. (C'est une droite, et donc la dérivée est la pente de cette droite !).

Calcul de f''_j
Pour obtenir une dérivée d'ordre supérieur, on doit prendre en considération plus de points. On combine les développements de Taylor de $f(x+h)$ et de $f(x+2h)$ en x, et on en déduit une approximation de $f''(x)$:

$$(1) \qquad f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2} f''(x) + \mathcal{O}(h^3)$$

$$(2) \qquad f(x+2h) = f(x) + 2hf'(x) + \frac{(2h)^2}{2} f''(x) + \mathcal{O}(h^3)$$

En formant $(2) - 2 \times (1)$, on élimine le terme en $f'(x)$ et on obtient :

$$f''(x) = \frac{f(x) - 2f(x+h) + f(x+2h)}{h^2} + \mathcal{O}(h)$$

$$\text{soit} \quad f''_j \simeq \frac{f_j - 2\,f_{j+1} + f_{j+2}}{h^2}$$

Ici aussi, la formule est équivalente à prendre la dérivée seconde au point x du polynôme de Lagrange passant par les 3 points x, $x+h$ et $x+2h$. (C'est une parabole !).

b) Dérivées à l'ordre 2 en h
Pour augmenter l'ordre, il faut également prendre en compte un nombre supérieur de points.

Calcul de f'_j

$$(1) \qquad f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2} f''(x) + \mathcal{O}(h^3)$$

$$(2) \qquad f(x+2h) = f(x) + 2hf'(x) + \frac{(2h)^2}{2} f''(x) + \mathcal{O}(h^3)$$

On forme $(2) - 4 \times (1)$ pour éliminer le terme en $f''(x)$, et on obtient :

$$f'(x) = \frac{-3\,f(x) + 4\,f(x+h) - f(x+2\,h)}{2\,h} + \mathcal{O}(h^2)$$

$$\text{soit} \quad f'_j \simeq \frac{1}{2\,h}(-3\,f_j + 4\,f_{j+1} - f_{j+2})$$

Calcul de f''_j. En suivant le même principe, on trouve :

$$f''_j = \frac{1}{h^2}(2\,f_j - 5\,f_{j+1} + 4\,f_{j+2} - f_{j+3}) + \mathcal{O}(h^2)$$

6.2.2 Différences régressives (ou à gauche ou amont)

On peut refaire les développements de Taylor en utilisant les points à gauche de x : $x-h$, $x-2h$, ...ou bien on peut utiliser les formules décentrées progressives avec h négatif. On obtient, à l'ordre 1 en h :

$$f'_j = \frac{f_j - f_{j-1}}{h} + \mathcal{O}(h)$$

et

$$f''_j = \frac{f_j - 2\,f_{j-1} + f_{j-2}}{h^2} + \mathcal{O}(h)$$

6.2.3 Différences centrées

Cette fois, on utilise les points entourant x : $x-h$, $x+h$, ... On montre que les ordres en h obtenus à l'aide des formules centrées sont pairs.

$$(1) \qquad f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2} f''(x) + +\mathcal{O}(h^3)$$

$$(2) \qquad f(x-h) = f(x) - h\,f'(x) + \frac{h^2}{2}\,f''(x) + \mathcal{O}(h^3)$$

On forme (1) − (2), ce qui permet d'éliminer le terme en h^2, et on obtient :

$$f'(x) = \frac{f(x+h) - f(x-h)}{2\,h} + \mathcal{O}(h^2)$$

soit,

$$f'_j = \frac{f_{j+1} - f_{j-1}}{2\,h} + \mathcal{O}(h^2)$$

Pour obtenir la dérivée seconde, on écrit les développements jusqu'à l'ordre 4 :

$$(1) \qquad f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2} f''(x) + \frac{h^3}{6} f'''(x) + \mathcal{O}(h^4)$$

$$(2) \qquad f(x-h) = f(x) - h\,f'(x) + \frac{h^2}{2}\,f''(x) - \frac{h^3}{6} f'''(x) + \mathcal{O}(h^4)$$

On forme (1) + (2), et on obtient :

$$f''(x) = \frac{f(x-h) - 2\,f(x) + f(x+h)}{h^2} + \mathcal{O}(h^2) \Longrightarrow f''_j = \frac{f_{j-1} - 2\,f_j + f_{j+1}}{h^2} + \mathcal{O}(h^2)$$

6.3 Utilisation des différences divisées

On définit les opérateurs Δ_+ et Δ_- par :
$\Delta_+\,f(x) = f(x+h) - f(x)$ pour les différences progressives,
et $\Delta_-\,f(x) = f(x) - f(x-h)$ pour les différences régressives.
On introduit aussi les opérateurs A et R tels que $A\,f(x) = f(x+h)$ et $R\,f(x) = f(x-h)$, et l'opérateur Identité I, de telle sorte que l'on a :
$\Delta_+\,f(x) = f(x+h) - f(x) = Af(x) - f(x) = (A-I)f(x)$, et
$\Delta_-\,f(x) = f(x) - f(x-h) = f(x) - Rf(x) = (I-R)f(x)$.

Ceci permettra le calcul des dérivées successives de $f(x)$ plus simplement et de les écrire sous forme de tableaux jusqu'à la dérivée quatrième.

6.3.1 Différences progressives

Dérivées d'ordre 1
On a démontré que

$$f'_j = \frac{\Delta_+\,f_j}{h} + \mathcal{O}(h) \simeq \frac{(A-I)f_j}{h}$$

et on a alors

$$f''_j = \frac{\Delta_+ f'_j}{h} + \mathcal{O}(h) \simeq \frac{\Delta_+^2 f_j}{h^2} = \frac{(A-I)^2 f_j}{h^2} = \frac{(A^2-2A+I) f_j}{h^2} = \frac{A^2 f_j - 2A f_j + f_j}{h^2}$$

Or $A f_j = f_{j+1}$ et $A^2 f_j = f_{j+2}$. Donc on retrouve :

$$f''_j = \frac{f_{j+2} - 2 f_{j+1} + f_j}{h^2} + \mathcal{O}(h)$$

On peut alors généraliser le résultat pour le calcul de la dérivée nième, en utilisant le développement du binôme de Newton, avec $C_n^k = \frac{n!}{k!\,(n-k)!}$:

$$f_j^{(n)} = \frac{\Delta_+^n f_j}{h^n} + \mathcal{O}(h) \simeq \frac{(A-I)^n f_j}{h^2} = \frac{1}{h^n} \sum_{k=0}^{n} C_n^k\, A^k I^{n-k} (-1)^{n-k}\, f_j = \frac{(-1)^n}{h^n} \sum_{k=0}^{n} (-1)^k\, C_n^k\, f_{j+k}$$

car $A^k I^{n-k} f_j = A^k f_j = f_{j+k}$. On peut alors construire le tableau suivant :

	f_j	f_{j+1}	f_{j+2}	f_{j+3}	f_{j+4}
$h\, f'_j$	-1	1			
$h^2\, f''_j$	1	-2	1		
$h^3\, f'''_j$	-1	3	-3	1	
$h^4\, f_j^{(4)}$	1	-4	6	-4	1

6.3.2 Différences régressives

Dérivées d'ordre 1 De façon analogue, on a

$$\frac{\Delta_-^n f_j}{h^n} + \mathcal{O}(h) \simeq \frac{(I-R)^n f_j}{h^2} = \frac{(-R+I)^n f_j}{h^2} = \frac{1}{h^n} \sum_{k=0}^{n} C_n^k\, (-1)^k R^k I^{n-k} f_j = \frac{1}{h^n} \sum_{k=0}^{n} (-1)^k\, C_n^k\, f_{j-k}$$

On en déduit le tableau suivant :

	f_j	f_{j-1}	f_{j-2}	f_{j-3}	f_{j-4}
$h\, f'_j$	1	-1			
$h^2\, f''_j$	1	-2	1		
$h^3\, f'''_j$	1	-3	3	-1	
$h^4\, f_j^{(4)}$	1	-4	6	-4	1

6.3.3 Différences centrées

Dérivées d'ordre 2 :
On utilise les résultats suivants :
pour n pair ($n = 2\,p$)

$$f_j^{(2\,p)} = \frac{\Delta_-^{2\,p} f_{j+p} + \Delta_+^{2\,p} f_{j-p}}{2\, h^{2\,p}} + \mathcal{O}(h^2)$$

pour n impair ($n = 2\,p+1$)

$$f_j^{(2\,p+1)} = \frac{\Delta_-^{2\,p+1} f_{j+p} + \Delta_+^{2\,p+1} f_{j-p}}{2\, h^{2\,p+1}} + \mathcal{O}(h^2)$$

Ce qui permet de construire le tableau :

	f_{j-2}	f_{j-1}	f_j	f_{j+1}	f_{j+2}
$2\,h\,f'_j$		-1		1	
$h^2\,f''_j$		1	-2	1	
$2\,h^3\,f'''_j$	-1	2		-2	1
$h^4\,f^{(4)}_j$	1	-4	6	-4	1

7 Intégration numérique

Définition :
Soit l'intégrale $I(f) = \int_a^b f(x)dx$ avec $b > a$, on cherche une valeur approchée de cette intégrale au moyen de sommes finies. On appelle formule de quadrature à $(n+1)$ points une formule du type :

$$I_n(f) = \sum_{j=0}^{n} A_j^n\, f(x_j)$$

où les A_j^n ne dépendent pas de la fonction f.

7.1 Formules de quadrature du type interpolation

Soient $(n+1)$ points x_j $(0 \le j \le n)$ où la fonction f est connue. Le polynôme $L_n(x)$ interpolé de Lagrange de f est donné par

$$L_n(x) = \sum_{j=0}^{n} L_j(x)\, f(x_j) \;\text{ avec }\; L_j(x) = \prod_{i=0, i\neq j}^{n} \frac{(x - x_i)}{(x_j - x_i)}.$$

La formule de quadrature associée est

$$\int_a^b f(x)\,dx \cong \int_a^b L_n(x)\,dx = \sum_{j=0}^{n} \left(\int_a^b L_j(x)\,dx \right) f(x_j) \;\text{ et donc }\; A_j^n = \int_a^b L_j(x)\,dx.$$

Propriétés :
- l'intégration des polynômes est très simple
- en général, $f(x_j)$ est tabulée en certains points donnés, et on n'a pas le choix des x_j
- si $f(x)$ est une fonction compliquée mais connue analytiquement, on peut :
  - soit prendre des subdivisions régulières de $[a, b]$ (formules de Newton-Cotes)
  - soit choisir les x_j "au mieux", au sens de Gauss

Erreur de quadrature :
On définit l'erreur $R(f) = I(f) - I_n(f)$.
On peut alors montrer qu'il existe $\xi \in]a, b[$ tel que $R(f) = \int_a^b \frac{\nu(x)}{(n+1)!} f^{(n+1)}(\xi)dx$ avec $\nu(x) = \Pi_{j=0}^{n}(x - x_j)$.
Une formule de quadrature est dite exacte si $R(f) = 0$.

Théorème :
Une formule de quadrature à $n + 1$ points de type interpolation est exacte pour $f(x) = x^k$ $(0 \le k \le n)$ par construction).

Définition :
On dit qu'une formule de quadrature a un degré de précision n si la formule est exacte pour $f(x) = x^k$ $(0 \leq k \leq n)$, mais non exacte pour $f(x) = x^{n+1}$.

7.2 Formules de Newton-Cotes

Définition :
On prend des subdivisions régulières de $[a, b]$ en posant $h = (b-a)/n$
$\Longrightarrow x_j = a + jh$ avec $(x_0 = a, x_n = b)$.

$$\int_a^b f(x)\,dx \cong \sum_{j=0}^{n} \left(\int_a^b L_j(x)\,dx \right) f(x_j) \;=\; (b-a) \sum_{j=0}^{n} B_j^n\, f(a+jh)$$

avec $\quad B_j^n = \frac{1}{b-a} \int_a^b L_j(x)\,dx$

Calcul des B_j^n :
On pose $\quad y = \frac{x-a}{h} \Longrightarrow \prod_{i=0, i\neq j}^{n}(x - x_i) = h^n \prod_{i=0, i\neq j}^{n}(y - i)$
et $\prod_{i=0, i\neq j}^{n}(x_j - x_i) = h^n \prod_{i=0, i\neq j}^{n}(j - i)$
$= h^n \times j \times (j-1) \times \ldots \times 2 \times 1 \times (-1) \times (-2) \times \ldots \times (-n-1+j) \times (-n+j) = h^n\,(-1)^{n-j}\,j!\,(n-j)!$

$$\Longrightarrow B_j^n = \frac{1}{nh} \int_0^n \frac{h^n \prod_{i=0, i\neq j}^{n}(y-i)}{h^n \prod_{i=0, i\neq j}^{n}(j-i)}\, h dy = \frac{(-1)^{n-j}}{j!\,(n-j)!\,n} \int_0^n \prod_{i=0, i\neq j}^{n} (y-i)\,dy$$

Remarque : On a $B_j^n = B_{n-j}^n$. Il suffit de calculer B_j^n pour $j \leq n/2$
$n = 1 \Longrightarrow B_0^1 = -1 \int_0^1 (y-1)\,dy = (-1)(1/2 - 1) = 1/2, \quad B_1^1 = B_0^1$
$n = 2 \Longrightarrow B_0^2 = \frac{(-1)^2}{0!2!2} \int_0^2 (y-1)(y-2)\,dy = 1/6, \quad B_2^2 = B_0^2;$
$\qquad\qquad B_1^2 = \frac{(-1)^1}{1!1!2} \int_0^2 y(y-2)\,dy = 4/6$

7.2.1 Formule des trapèzes $(n = 1)$

$$\int_a^b f(x)\,dx \cong (b-a)\,\left[\frac{1}{2}f(a) + \frac{1}{2}f(b)\right]$$

calcul d'erreur :
$R(f) = \int_a^b f(x)\,dx - \frac{b-a}{2}[f(a) + f(b)]$.
On pose $b = a + h$ et on a alors $R(f) = \int_a^{a+h} f(x)dx - \frac{h}{2}[f(a) + f(a+h)]$. Si $f(x)$ est suffisamment dérivable, on fait des développements limités au voisinage de $h = 0$.
On a : $f(a+h) = f(a) + hf'(a) + \frac{h^2}{2}f''(a) + O(h^3)$.
Soit $F(x)$ une primitive de $f(x)$, on a $\int_a^{a+h} f(x)dx = F(a+h) - F(a)$, et
$F(a+h) = F(a) + hf(a) + \frac{h^2}{2}f'(a) + \frac{h^3}{6}f''(a) + O(h^4)$,
On trouve $R(f) = (\frac{1}{6} - \frac{1}{4})h^3 f''(a) + O(h^4) = -\frac{h^3}{12}f''(a) + O(h^4)$.

7.2.2 Formule de Simpson $(n = 2)$

$$\int_a^b f(x)\,dx \cong (b-a)\,\left[\frac{1}{6}f(a) + \frac{4}{6}f(\frac{a+b}{2}) + \frac{1}{6}f(b)\right]$$

calcul d'erreur :
$R(f) = \int_a^b f(x)\,dx - \frac{b-a}{6}\Big[f(a) + 4f(\frac{a+b}{2}) + f(b)\Big]$.
On pose $h = (b-a)/2$. Vue la symétrie de $R(f)$ par rapport à $\alpha = (a+b)/2$, on effectue le développement limité au voisinage du point α, soit :

$$R(f) = \int_{\alpha-h}^{\alpha+h} f(x)\,dx - \frac{h}{3}[f(\alpha-h) + 4f(\alpha) + f(\alpha+h)]$$

En faisant des développements de Taylor, on montre qu'il existe $\xi \in]\alpha-h, \alpha+h[$ tel que $R(f) = -\frac{h^5}{90} f^{(4)}(\xi)$

7.2.3 tableau de coefficients

Sur le tableau suivant, on donne les B_j^n pour $1 \leq n \leq 6$:

n	1	2	3	4	5	6
B_0^n	1/2	1/6	1/8	7/90	19/288	41/840
B_1^n		4/6	3/8	32/90	75/288	216/840
B_2^n				12/90	50/288	27/840
B_3^n						272/840

Théorème : Si le nombre de points d'intégration est $(n+1)$ l'erreur de quadrature des formules de Newton-Cotes est en h^{n+2} pour n impair ($R(f) = O(h^{n+2})$), et h^{n+3} pour n pair ($R(f) = O(h^{n+3})$).

Le degré de précision dans le cas des formules de Newton-Côtes à $(n+1)$ points est n pour n impair„ et $n+1$ pour n pair.

7.3 Les méthodes composites

Définition : On applique à des sous-intervalles de (a, b) une formule de Newton-Cotes de degré q petit, fixé. On pose $h = (b-a)/n$.

7.3.1 Méthode composite des trapèzes ($q = 1$)

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n-1} \int_{a+ih}^{a+(i+1)h} f(x)\,dx \cong \sum_{i=0}^{n-1} h\left[\frac{1}{2}f(a+ih) + \frac{1}{2}f(a+(i+1)h)\right]$$

$$\text{donc} \quad \int_a^b f(x)\,dx \cong h\left[\frac{1}{2}f(a) + f(a+h) + f(a+2h) + ... + f(a+(n-1)h) + \frac{1}{2}f(b)\right]$$

7.3.2 Méthode composite de Simpson ($q = 2$)

On choisit n pair et on considère les intervalles de longueur $2h$.

$$\int_a^{a+2h} f(x)\,dx \cong 2h\left[\frac{1}{6}f(a) + \frac{4}{6}f(a+h) + \frac{1}{6}f(a+2h)\right]$$

$$\int_{a+2h}^{a+4h} f(x)\,dx \cong 2h\left[\frac{1}{6}f(a+2h) + \frac{4}{6}f(a+3h) + \frac{1}{6}f(a+4h)\right]$$

$$\int_{a+(n-2)h}^{b} f(x)\,dx \cong 2h\left[\frac{1}{6}f(a+(n-2)h)+\frac{4}{6}f(a+(n-1)h)+\frac{1}{6}f(b)\right]$$

Soit en additionnant :

$$\begin{aligned}\int_a^b f(x)\,dx \;\cong\; & \tfrac{h}{3}\{f(a)+f(b)+2[f(a+2h)+f(a+4h)+.....+f(a+(n-2)h)]\\ & +4\,[f(a+h)+f(a+3h)+.....+f(a+(n-1)h)]\}\end{aligned}$$

7.4 Formules de quadrature du type Gauss

Pour obtenir des formules de quadrature à $(n+1)$ points possédant un degré de précision supérieur à celui obtenu par les formules de Newton-Côtes, on peut, au lieu de prendre des abscisses x_j régulièrement espacées, les choisir "au mieux".

7.4.1 Gauss-Legendre

Comme les inconnues sont à présent les $n+1$ coefficients A_j^n et les points $n+1$ points x_j (soient $2n+2$ inconnues), on peut espérer augmenter le degré de précision à $2n+1$. On cherche les A_j^n et les x_j tels que la formule de quadrature $I_\omega(f)=\sum_{j=0}^n A_j^n\, f(x_j)$ soit exacte pour tout polynôme de degré $\leq 2n+1$.

a) Formules par identification :

a) formule à 1 point ($n=0$) : On cherche $A_0^0 \in \mathbb{R}$, et $x_0 \in [a,b]$ tel que la quadrature soit de degré de précision le plus élevé possible.

$$\int_a^b f(x)\,dx \cong \left[A_0^0 f(x_0)\right]$$

On écrit que la quadrature est exacte pour $f(x)=1$ et $f(x)=x$. $f(x)=1 \to \int_a^b dx = b-a = A_0^0$
$f(x)=x \to \int_a^b x dx = \frac{(b^2-a^2)}{2} = A_0^0 x_0$
on trouve : $A_0^0 = b-a$ et $x_0 = \frac{(a+b)}{2}$.
On a ainsi la formule de Gauss à 1 point, de degré de précision 1, est :

$$\int_a^b f(x)\,dx = (b-a) f[\frac{(a+b)}{2}]$$

b) formule à 2 points ($n=1$) :
On effectue tout d'abord le calcul sur l'intervalle $[0,1]$, puis on effectue un changement de variable pour se ramener à l'intervalle $[a,b]$.

$$\int_0^1 f(x)\,dx \cong \left[A_0^1 f(x_0)+A_1^1 f(x_1)\right]$$

$f(x)=1 \to \int_0^1 dx = 1 = A_0^1 + A_1^1$
$f(x)=x \to \int_0^1 x dx = 1/2 = A_0^1 x_0 + A_1^1 x_1$
Soit le polynôme de degré 2 : $\pi(x)=(x-x_0)(x-x_1)=0=x^2-sx+p$ et soit le polynôme de degré 3 : $\pi_1(x)=x\,\pi(x)$. En écrivant que la quadrature est exacte pour $f(x)=\pi(x)$ et $f(x)=\pi_1(x)$, on a $\int_0^1 \pi(x)\,dx = 0$ et $\int_0^1 \pi_1(x)\,dx = 0$, et on trouve

$s = 1, p = 1/6$.
En résolvant $x^2 - sx + p = 0$, on a $x_0 = (1 - 1/\sqrt{3})/2, x_1 = (1 + 1/\sqrt{3})/2$.
A partir des relations $A_0^1 + A_1^1 = 1$ et $A_0^1\,(1 - 1/\sqrt{3})/2 + A_1^1\,(1 + 1/\sqrt{3})/2 = 1/2$, on trouve alors $A_0^1 = A_1^1 = 1/2$.
On obtient donc : $\int_0^1 f(x)\,dx = [f((1 - 1/\sqrt{3})/2) + f((1 + 1/\sqrt{3})/2)]/2$.
La formule de Gauss à 2 points, exacte pour les polynômes de degré 3 est :

$$\int_a^b f(x)\,dx \cong ((b-a)/2)[f[a + (b-a)(1 - 1/\sqrt{3})/2] + f[a + (b-a)(1 + 1/\sqrt{3})/2]]$$

.

b) Utilisation des polynômes de Legendre
Les polynômes orthogonaux de Legendre permettent de construire de façon systématique les quadratures de Gauss-Legendre.

Ces polynômes constituent une famille de polynômes dits orthogonaux, définis sur l'intervalle $[-1, 1]$. Les polynômes $X_0, X_1, ..., X_n$ constituent une base de l'espace des polynômes de degré inférieur ou égal à n, définis sur $[-1, 1]$.

Orthogonalité :
Les relations d'orthogonalité sont

$$\int_{-1}^{1} X_n(x)\,X_p(x)\,dx = 0 \quad \text{si} \quad n \neq p \quad \text{et} \quad \int_{-1}^{1} X_n^2(x)\,dx = \frac{2}{2n+1}$$

Récurrence :
Les polynômes orthogonaux de Legendre vérifient une relation de récurrence à trois termes
$(n + 1)\,X_{n+1} = (2n + 1)xX_n - nX_{n-1}$
On les calcule à partir de $X_0 = 1$ et $X_1 = x$.
On a donc $X_2 = \frac{3}{2}x^2 - \frac{1}{2}$, $X_3 = \frac{5}{2}x^3 - \frac{3}{2}x$, ...

Equation :
On montre que ces polynômes sont solutions de l'équation différentielle :

$$(x^2 - 1)\;y'' + 2\,x\;y' - n\,(n+1)\,y = 0$$

Quadrature sur [-1,1] :

Soit à calculer l'intégrale $I_1(f) = \int_{-1}^{1} f(\xi)d\xi$, on introduit les $(n + 1)$ racines ξ_j du polynôme de Legendre $X_{n+1}(x)$ (qui est un polynôme de degré $n + 1$). La quadrature est alors une quadrature de type interpolation construite à partir des ξ_j :

$$I_1(f) \cong I_{1n} \sum_{j=0}^{n} \omega_j^n\,f(\xi_j)$$

Les facteurs de pondération ω_j^n sont en général tabulés.
On peut retrouver leur valeur par :

$$\omega_j^n = \int_{-1}^{1} \left[\frac{X_{n+1}(x)}{(x-\xi_j)\, X'_{n+1}(\xi_j)} \right]^2 dx$$

soit,

$$\omega_j = \frac{2}{\left[(1-x_j^2)(X'_n(x_j))^2\right]}$$

Le tableau suivant donne les valeurs des ξ_j et des ω_j^n $(2 \leq n+1 \leq 6)$:

n+1	ξ_j	w_j^n
2	± 0.577350	1.0000000
3	0.000000	0.8888889
	± 0.774597	0.5555556
4	± 0.333333	0.6521450
	± 0.861136	0.3478548

n+1	ξ_j	w_j^n
5	0.000000	0.5688889
	± 0.538469	0.4786290
	± 0.906280	0.2369270
6	± 0.238619	0.4679140
	± 0.661209	0.3607616
	± 0.932469	0.1713245

Théorème : Les formules de Gauss à $n+1$ points sont exactes pour les polynômes de degré $2n+1$.

Preuve : Comme il s'agit d'une quadrature de type interpolation à $n+1$ points, la quadrature est exacte pour les polynômes de degré inférieur ou égal à n.
Soit alors un polynôme $P(x)$ défini sur $[-1,1]$ de degré $2n+1$, et $X_{n+1}(x) = (x-\xi_0)(x-\xi_1)....(x-\xi_n)$ le polynôme de Legendre de degré n, de racines ξ_j.
0n effectue la division polynômiale :
$P(x) = X_{n+1}(x)Q(x) + R(x)$, où le degré de $Q(x)$ est inférieur ou égal à n (le degré de $R(x)$ l'est également). On a alors :
$\int_{-1}^{1} P(x)dx = \int_{-1}^{1} X_{n+1}(x)Q(x)dx + \int_{-1}^{1} R(x)dx$
La propriété d'orthogonalité des polynômes de Legendre entraîne $\int_{-1}^{1} X_{n+1}(x)Q(x)dx = 0$.
La quadrature étant exacte pour $R(x)$, on a
$\int_{-1}^{1} P(x)dx = \int_{-1}^{1} R(x)dx = \sum_{j=0}^{n} \omega_j\, R(\xi_j)$
Or on a $R(\xi_j) = P(\xi_j) - X_{n+1}(\xi_j)Q(\xi_j) = P(\xi_j)$, et donc $\int_{-1}^{1} P(x)dx = \sum_{j=0}^{n} \omega_j\, P(\xi_j)$.
La quadrature est donc exacte pour $P(x)$.

Généralisation à la quadrature sur [a,b] :

Soit à calculer $I(f) = \int_a^b f(x)dx$.

On utilise un changement de variables pour se ramener sur l'intervalle $[-1,1]$, en posant :

$$x = \frac{b+a}{2} + \frac{b-a}{2}\xi$$

, soit

$$\xi = -\frac{b+a}{b-a} + \frac{2}{b-a}x.$$

On a alors

$$I(f) = \int_a^b f(x)dx = \frac{(b-a)}{2}\int_{-1}^{1} f(\frac{b+a}{2} + \frac{b-a}{2}\xi)d\xi \cong \frac{(b-a)}{2}\sum_{j=0}^{n} \omega_j\, f(\frac{b+a}{2} + \frac{b-a}{2}\xi_j),$$

où les ξ_j sont les $n+1$ racines du polynôme de Legendre $X_{n+1}(x)$, et les facteurs ω_j sont ceux déterminés par la quadrature de Gauss.

Remarque Supposons que l'on souhaite calculer l'intégrale suivante où $\omega(x)$ est une fonction poids, positive sur $]a, b[$.

$$I_\omega(f) = \int_a^b \omega(x) f(x) dx$$

Si $\omega = \frac{1}{\sqrt{1-x^2}}$, on utilise les polynômes de Chebyshev T_n, qui sont une autre base de polynômes orthogonaux, également définis sur $[-1, 1]$.

7.4.2 Gauss-Radau

On travaille de nouveau sur l'intervalle $[-1, 1]$. Supposons que dans la quadrature, l'extrémité -1 est assignée. Pour une quadrature à deux points, on cherche x_1 tel que

$$\int_{-1}^{1} f(x)\, dx = \left[A_0^1 f(-1) + A_1^1 f(x_1)\right] + R(f)$$

soit exacte ($R(f) = 0$) sur l'espace vectoriel des polynômes du degré le plus élevé possible.

On a 3 inconnues : $x_1,\ A_0^1\, A_1^1$. En écrivant les trois premières relations, on a :

$f(x) = 1 \to R(f) = 2 - A_0^1 - A_1^1 = 0$
$f(x) = x \to R(f) = 0 + A_0^1 - x_1 A_1^1 = 0$
$f(x) = x^2 \to R(f) = \frac{2}{3} - A_0^1 - x_1^2 A_1^1 = 0$

$\Longrightarrow x_1 = 1/3,\ A_0^1 = 1/2,\ A_1^1 = 3/2$

d'où $\int_{-1}^{1} f(x)dx = \frac{1}{2}[f(-1) + 3f(1/3)] + R(f)$. Pour $f(x) = x^3$, $R(f) \neq 0$.

7.4.3 Gauss-Lobatto

On travaille de nouveau sur l'intervalle $[-1, 1]$. Supposons que dans la quadrature, les deux extrémités -1 et 1 sont assignées. Par identification, on obtient les formules suivantes, exactes respectivement pour les polynômes de degré 3 et 5 :

$$\int_{-1}^{1} f(x)\, dx = \frac{1}{3}f(-1) + \frac{4}{3}f(0) + \frac{1}{3}f(1) + R(f)$$

$$\int_{-1}^{1} f(x)\, dx = \frac{1}{6}f(-1) + \frac{5}{6}f(-1/\sqrt{5}) + \frac{5}{6}f(1/\sqrt{5}) + \frac{1}{6}f(1) + R(f)$$

8 Equations différentielles ordinaires (EDO)

Introduction - Définitions
Dans ce chapitre, on cherche à résoudre numériquement une équation différentielle ordinaire vérifiant une condition initiale (problème de Cauchy). Avant d'effectuer une résolution numérique, il faut s'assurer que le problème admet une unique solution.

Pb de Cauchy, EDO du 1er ordre :
On se donne un intervalle $[a, b]$ de $\mathbb{R}$, une fonction $f : [a, b] \times \mathbb{R} \to \mathbb{R}$, et $\alpha \in \mathbb{R}$.
On souhaite déterminer la fonction $y : [a, b] \to \mathbb{R}$, vérifiant le problème de Cauchy ci-dessous :

$$\begin{cases} y'(x) &= f(x, y(x)) \qquad \forall\ x \in [a, b] \\ y(a) &= \alpha \end{cases}$$

On rappelle le théorème : Si f est continue dans $[a, b] \times \mathbb{R}$ et lipschitzienne par rapport à la seconde variable, alors le problème de Cauchy admet une solution unique.

Pb de Cauchy, systèmes d'EDOs du 1er ordre :
On se donne un intervalle $[a, b]$ de $\mathbb{R}$, une fonction $f : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$, et et $\alpha \in \mathbb{R}^n$.
On souhaite déterminer la fonction $y : [a, b] \to \mathbb{R}^n$, vérifiant le problème de Cauchy :

$$\begin{cases} y_1'(x) = f_1(x, y_1, y_2, ..., y_n) \\ y_2'(x) = f_2(x, y_1, y_2, ..., y_n) \\ ... \\ y_n'(x) = f_n(x, y_1, y_2, ..., y_n) \end{cases} \quad \text{et} \quad \begin{cases} y_1(a) = \alpha_1 \\ y_2(a) = \alpha_2 \\ ... \\ y_n(a) = \alpha_n \end{cases}$$

Le théorème d'existence et d'unicité est identique au cas précédent. Il suffit d'utiliser pour la condition de Lipschitz, une norme dans $\mathbb{R}^n$.

Systèmes d'EDOs d'ordre supérieur à 1 :
On peut se ramener à un système d'équations différentielles du premier ordre.

8.1 Méthodes d'intégration à un pas

8.1.1 Définition

On subdivise l'intervalle d'intégration $[a, b]$ en n points équidistants, en posant $h = (b - a)/n$, et $x_{i+1} - x_i = h$, de sorte que : $a = x_0, x_1, x_2, ..., x_n = b$.
On pose $y_0 = y(x_0) = y(a) = \alpha$, et on note y_i la valeur approchée de $y(x_i)$.

Une méthode à un pas permet de calculer y_{i+1} à partir de y_i.
A partir de la donnée de $y_0 = \alpha$, on calcule donc successivement les y_i $(y_1, y_2, ..., y_n)$.
On relie ensuite les points (x_i, y_i) par interpolation pour définir une fonction y_i sur $[a, b]$.
L'erreur de discrétisation $e_i = y(x_i) - y_i$ dépend de la valeur du pas h.

8.1.2 Schémas explicite/implicite

On distingue les schémas explicites, pour lesquels on peut calculer explicitement y_{i+1} en fonction de la donnée y_i, des schémas implicites, pour lesquels il faut résoudre une équation pour calculer y_{i+1} en fonction de la donnée y_i.

1. **Exemple 1**
   Supposons qu'on utilise un différence progressive d'ordre 1 pour approcher $y'(x_i)$. On obtient $y'(x_i) = f(x_i, y_i) \cong (y_{i+1} - y_i)/h$, soit : $y_{i+1} = y_i + h\, f(x_i, y_i)$. C'est un schéma explicite, car on peut calculer explicitement y_{i+1} en fonction de la donnée y_i.
2. **Exemple 2**
   Supposons qu'on utilise un différence régressive d'ordre 1 pour approcher $y'(x_i)$. On obtient $y'(x_i) = f(x_i, y_i) \cong (y_i - y_{i-1})/h$, soit : $y_i = y_{i-1} + h\, f(x_i, y_i)$, ou encore $y_{i+1} = y_i + h\, f(x_{i+1}, y_{i+1})$. C'est un schéma implicite, car pour calculer y_{i+1} en fonction de la donnée y_i, il faut résoudre une équation, qui peut être non-linéaire.
   On peut par exemple utiliser une méthode itérative de type point fixe :
   $y_{i+1}^{(k+1)} = y_i + h\, f(x_{i+1}, y_{i+1}^{(k)})$, avec $y_{i+1}^{(0)}$ calculé par un schéma explicite.
3. **Exemple 3**
   Supposons qu'on utilise une différence centrée d'ordre 2 pour approcher $y'(x_i)$. On pose $y'(x_i) = f(x_i, y_i) \cong (y_{i+1} - y_{i-1})/(2h)$. On obtient $y_{i+1} = y_{i-1} + 2\, h\, f(x_i, y_i)$. C'est un schéma explicite à deux pas qui nécessite la connaissance de deux conditions initiales y_0 et y_1. En général, on calcule y_1 en utilisant une méthode à un pas.

8.1.3 Généralités sur les méthodes à un pas explicites : Consistance - stabilité - convergence

Soit une méthode à un pas dont le schéma est donné par :

$$\begin{cases} y_0 & = \alpha \\ y_{i+1} & = y_i + h\, \phi(x_i, y_i, h) \quad (0 \leq i \leq n) \end{cases}$$

Les diverses méthodes se distinguent par le choix de la fonction $\phi(x, y, h)$.

1. **Consistance**
   La méthode $y_{i+1} = y_i + h\, \phi(x_i, y_i, h)$ est consistante avec l'équation différentielle si, pour toute solution $y(x)$ de $y'(x) = f(x, y)$ on a :

$$lim_{h \to 0} \left[max_i \left| \frac{1}{h}(y(x_{i+1}) - y(x_i)) - \phi(x_i, y(x_i), h) \right| \right] = 0$$

   Théorème (admis) : Pour qu'une méthode à un pas soit consistante, il faut et il suffit que $\phi(x, y, 0) = f(x, y)$.

2. **Ordre de l'erreur**
   L'erreur de discrétisation est définie par $e_{i+1} = y(x_{i+1}) - y_{i+1}$.

   On peut alors calculer l'erreur commise sur un pas, en supposant que $y_i = y(x_i)$:

   $e_{i+1} = y(x_{i+1}) - y_{i+1} = y(x_{i+1}) - y(x_i) + y(x_i) - y_{i+1}$
   $= y(x_{i+1}) - y(x_i) + y(x_i) - [y_i + h\phi(x_i, y_i, h)] = y(x_{i+1}) - y(x_i) - h\phi(x_i, y_i, h)$.

   Définition : On dit que la méthode est d'ordre $\geq p$ si

$$max_i \left| \frac{1}{h}(y(x_{i+1}) - y(x_i)) - \phi(x_i, y(x_i), h) \right| = \mathcal{O}(h^p)$$

En effet, l'erreur sur un pas vérifiera alors $e_{i+1} = \mathcal{O}(h^{p+1})$, et l'erreur globale sur l'intervalle $[a, b] = [x_0, x_n]$, donnée par $e = \sum_{i=0}^{n-1} e_{i+1}$, sera majorée par :
$|e| \leq n \times max_i\, |e_{i+1}| = n \times \mathcal{O}(h^{p+1}) = n \times h \times \mathcal{O}(h^p) = (b-a) \times \mathcal{O}(h^p)$, et donc la méthode est globalement d'ordre p.

Dire qu'une méthode est consistante revient à dire qu'elle est au moins d'ordre 1. En général on utilise des développements limités pour démontrer la consistance et calculer l'ordre de l'erreur.

3. **Stabilité**
   Soient y_i et z_i $(1 \leq i \leq n)$ les solutions respectives des systèmes :

$$\begin{cases} y_{i+1} & = y_i + h\,\phi(x_i, y_i, h) \\ y_0 & \text{donné} \end{cases} \qquad \begin{cases} z_{i+1} & = z_i + h\,[\phi(x_i, z_i, h) + \epsilon_i] \\ z_0 & \text{donné} \end{cases}$$

   La méthode est dite théoriquement stable s'il existe deux constantes M_1 et M_2 indépendantes de h telles que :

$$max_i\, |y_i - z_i| \leq M_1\, |y_0 - z_0| + M_2\, max_i\, |\epsilon_i|$$

   Signification : Une méthode est stable si une petite perturbation sur les données (α, ϕ) n'entraîne qu'une petite perturbation sur la solution, et ceci, indépendamment de h.

   Théorème : Si la fonction ϕ est Lipschitzienne par rapport à la seconde variable, alors la méthode est stable.

4. **Convergence**
   La méthode converge si $lim_{h\to 0} max_i |y(x_i) - y_i| = 0$, quelque soit la condition initiale α.
   Si une méthode à un pas est consistante et stable, elle est alors convergente.

5. **Interprétation en terme de quadrature.** En repartant de l'EDO, $y'(x) = f(x, y(x))$, si on l'intègre entre x_i et x_{i+1}, on obtient : $y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} f(x, y(x))\, dx$
   La méthode à un pas correspondant à $\phi(x, y, h)$ donnée, consiste donc à approcher l'intégrale par $h\phi(x_i, y_i, h)$.

8.2 Méthode d'Euler.

Définition. On choisit $\phi(x_i, y_i, h) = f(x_i, y_i)$. La méthode est donc la suivante :

$$\begin{cases} y_0 & = \alpha \\ y_{i+1} & = y_i + h\, f(x_i, y_i) \quad (0 \leq i \leq n-1) \end{cases}$$

Consistance et ordre. Ecrivons le développement de Taylor de la fonction $y(x)$ en x_i, en supposant qu'elle est suffisamment régulière :
$y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + \mathcal{O}(h^2) = y(x_i) + hf(x_i, y(x_i)) + \mathcal{O}(h^2)$, donc :
$y(x_{i+1}) = y(x_i) + h\phi(x_i, y(x_i), h) + \mathcal{O}(h^2)$ et on a : $\frac{1}{h}[y(x_{i+1}) - y(x_i)] - \phi(x_i, y(x_i), h) = \mathcal{O}(h)$.
La méthode d'Euler est d'ordre 1. Elle est donc consistante.

Stabilité. Si f est k lipschitzienne par rapport à la deuxième variable, $\forall x \in [a,b]$, alors ϕ est Lipschitzienne par rapport à la seconde variable, et donc la méthode d'Euler est stable.

Convergence. Si f est k lipschitzienne par rapport à la deuxième variable, $\forall x \in [a,b]$, alors la méthode d'Euler converge.

Interprétation en terme de quadrature. La méthode d'Euler consiste donc à approcher l'intégrale par $\int_{x_i}^{x_{i+1}} f(x, y(x))\, dx \cong h f(x_i, y_i)$.

8.3 Méthodes de Runge-Kutta

8.3.1 Méthodes RK2

On suppose que la fonction est suffisamment régulière, $f(x,y) \in C^2$, et on prend, avec $0 < \beta < 1$:

$$\phi(x,y,h) = (1-\beta) f(x,y) + \beta f\left[x + \frac{h}{2\beta}, y + \frac{h}{2\beta} f(x,y)\right]$$

Les deux méthodes RK2 les plus utilisées sont :

a) méthode d'Euler modifiée :

$$\beta = 1/2 \Longrightarrow y_{i+1} = y_i + \frac{h}{2}\left\{f(x_i,y_i) + f\left[x_i + h, y_i + h f(x_i,y_i)\right]\right\}$$

Elle est basée sur la quadrature des trapèzes. En pratique, pour la résolution du pas $[x_i, x_{i+1}]$, on procède en deux étapes :

$$\begin{cases} k_1 = f(x_i, y_i) & \Longrightarrow \tilde{y}_{i+1} = y_i + h\, k_1 \\ k_2 = \frac{1}{2}[k_1 + f(x_{i+1}, \tilde{y}_{i+1})] & \Longrightarrow y_{i+1} = y_i + h\, k_2 \end{cases}$$

On peut montrer que la méthode d'Euler modifié est une méthode d'ordre 2.

b) méthode de Heun d'ordre 2 :

$$\beta = 1 \Longrightarrow y_{i+1} = y_i + h f\left[x_i + \frac{h}{2}, y_i + \frac{h}{2} f(x_i, y_i)\right]$$

Elle est basée sur la méthode d'intégration du point milieu. En pratique, pour la résolution du pas $[x_i, x_{i+1}]$, on procède en deux étapes :

$$\begin{cases} k_1 = f(x_i, y_i) & \Longrightarrow y_{i+1/2} = y_i + \frac{h}{2}\, k_1 \\ k_2 = f(x_{i+1/2}, y_{i+1/2})] & \Longrightarrow y_{i+1} = y_i + h\, k_2 \end{cases}$$

On peut montrer que la méthode de Heun est d'ordre 2.

8.3.2 Méthode RK3

Elle est basée sur la méthode d'intégration de Gauss-Radau en deux points. En pratique, pour la résolution du pas $[x_i, x_{i+1}]$, on procède en trois étapes :

$$\begin{cases} k_1 = f(x_i, y_i) & \Longrightarrow y_{i+1/3} = y_i + \frac{h}{3}\, k_1 \\ k_2 = f(x_{i+1/3}, y_{i+1/3})] & \Longrightarrow y_{i+2/3} = y_i + \frac{2h}{3}\, k_2 \\ k_3 = f(x_{i+2/3}, y_{i+2/3})] & \Longrightarrow y_{i+1} = y_i + \frac{h}{4}\,[k_1 + 3\, k_3] \end{cases}$$

La méthode RK3 est d'ordre 3.

8.3.3 Méthode RK4

Elle est basée sur la méthode d'intégration de Simpson en trois points équidistants. En pratique, pour la résolution du pas $[x_i, x_{i+1}]$, on procède en quatre étapes :

$$\begin{cases} k_1 = f(x_i, y_i) & \Longrightarrow \tilde{y}_{i+1/2} = y_i + \frac{h}{2}\, k_1 \\ k_2 = f(x_{i+1/2}, \tilde{y}_{i+1/2})] & \Longrightarrow y_{i+1/2} = y_i + \frac{h}{2}\, k_2 \\ k_3 = f(x_{i+1/2}, y_{i+1/2}) & \Longrightarrow \tilde{y}_{i+1} = y_i + h\, k_3 \\ k_4 = f(x_{i+1}, \tilde{y}_{i+1}) & \Longrightarrow y_{i+1} = y_i + \frac{h}{6}\,[k_1 + 2\, k_2 + 2\, k_3 + k_4] \end{cases}$$

La méthode RK4 est d'ordre 4. Elle est la plus utilisée pour la résolution des équations différentielles du premier ordre.