

# 1 Interpolation

$n + 1$  points

## 1.1 Polynôme d'interpolation de Lagrange avec 3+1 points

$$l_0(x) = \lambda_0(x - x_1)(x - x_2)(x - x_3)$$

$$\lambda_0 = \frac{1}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)}$$

$$l_1(x) = \lambda_1(x - x_0)(x - x_2)(x - x_3)$$

$$\lambda_1 = \frac{1}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)}$$

$$l_2(x) = \lambda_2(x - x_0)(x - x_1)(x - x_3)$$

$$\lambda_2 = \frac{1}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)}$$

$$l_3(x) = \lambda_3(x - x_0)(x - x_1)(x - x_2)$$

$$\lambda_3 = \frac{1}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)}$$

### Forme 1

$$p_3(x) = y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x) + y_3 l_3(x)$$

### Forme 2

$$p_3(x) = (x - x_0)(x - x_1)(x - x_2)(x - x_3) \cdot (y_0 \mu_0 + y_1 \mu_1 + y_2 \mu_2 + y_3 \mu_3)$$

$$\mu_i = \frac{\lambda_i}{x - x_i}$$

Forme barycentrique :

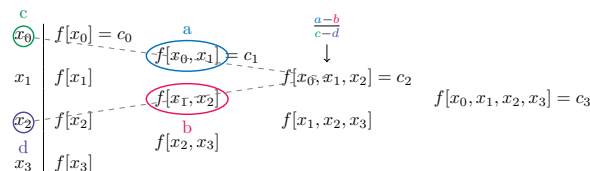
$$p_3(x) = \frac{y_0 \mu_0 + y_1 \mu_1 + y_2 \mu_2 + y_3 \mu_3}{\mu_0 + \mu_1 + \mu_2 + \mu_3}$$

### Forme 3 (un peu plus simple à écrire)

$$p_2(x) = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

## 1.2 Polynôme d'interpolation de Newton

$$p_3(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)(x - x_2)$$



Possible de rajouter une ligne en dessous du tableau

## 1.3 Évaluation

On utilise le schéma de Horner

	$x^3$	$x^2$	$x$	$x^0$	
$x = 2$	5	-3	0	2	
	+	10	14	20	
	5	7	14	30	$p_3(2) = 30$
	+	10	34		
	5	17	48		$p'_3(2) = 48$

Après la première dérivée, il faudra appliquer un facteur

$$\frac{P_n^{(k)}(x_i)}{k!}$$

## 1.4 Erreur

Ordre 1 (2 points)

$$|f(x) - p_1(x)| \leq \frac{1}{8} M_2 h^2$$

Ordre 2 (3 points)

$$|f(x) - p_2(x)| \leq \frac{\sqrt{3}}{27} M_3 h^3$$

Ordre 3 (4 points)

$$\begin{cases} |f(x) - p_3(x)| \leq \frac{3}{128} M_4 h^4 & x \in [x_1, x_2] \\ |f(x) - p_3(x)| \leq \frac{1}{24} M_4 h^4 & x \in [x_0, x_1] \cup [x_2, x_3] \end{cases}$$

Avec

$$M_k = \max_{\zeta \in [a, b]} |f^{(k)}(\zeta)|$$

L'utilisation des abscisses de Chebychev permettent de minimiser l'erreur avec un nombre de points donnés sur un intervalle donné

# 2 Splines

## 2.1 Polynômes de degré 1 (linéaire)

Entre  $(x_0, y_0)$  et  $(x_1, y_1)$ :

$$f(x) = m(x - x_0) + y_0 \quad m = \frac{y_0 - y_1}{x_0 - x_1}$$

Bonne approximation mais points anguleux

## 2.2 Polynômes de degré 2 (quadratique)

$$\begin{aligned} s_0(x_1) &= s_1(x_1) = y_1 & s'_0(x_1) &= s'_1(x_1) \\ s_1(x_2) &= s_2(x_2) = y_2 & s'_1(x_2) &= s'_2(x_2) \end{aligned}$$

$$s_0(x_0) = y_0 \quad s_n(x_{n+1}) = y_{n+1}$$

$3n - 1$  conditions pour  $3n$  inconnues. Un degré de liberté :  $s'_0(x_0)$

## 2.3 Polynômes de degré 3 (cubique)

$$\begin{aligned} s_0(x_1) &= s_1(x_1) = y_1 & s'_0(x_1) &= s'_1(x_1) & s''_0(x_1) &= s''_1(x_1) \\ s_1(x_2) &= s_2(x_2) = y_2 & s'_1(x_2) &= s'_2(x_2) & s''_1(x_2) &= s''_2(x_2) \end{aligned}$$

$$s_0(x_0) = y_0 \quad s_n(x_{n+1}) = y_{n+1}$$

$4n - 2$  conditions pour  $4n$  inconnues.

$$s''_0(x_0) = s''_{n-1}(x_n) = 0 \longrightarrow \text{splines naturelles}$$

Chaque segment est donné par

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

On trouve ensuite les coefficients pour chacun avec

$$\begin{cases} a_i &= \frac{1}{6h} (y''_{i+1} - y''_i) \\ b_i &= \frac{1}{2} y''_i \\ c_i &= \frac{1}{h} (y_{i+1} - y_i) - \frac{1}{6} h (y''_{i+1} + 2y''_i) \\ d_i &= y_i \end{cases}$$

$$\begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{pmatrix}_{n-1 \times n-1} \begin{pmatrix} y''_1 \\ y''_2 \\ y''_3 \\ y''_4 \end{pmatrix} = \frac{6}{h^2} \begin{pmatrix} y_2 - 2y_1 + y_0 \\ y_3 - 2y_2 + y_1 \\ y_4 - 2y_3 + y_2 \\ y_5 - 2y_4 + y_3 \end{pmatrix} - \begin{pmatrix} y''_0 \\ 0 \\ 0 \\ y''_5 \end{pmatrix}$$

Note :

$$\begin{pmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{pmatrix}^I = \frac{1}{56} \begin{pmatrix} 15 & -4 & 1 \\ -4 & 16 & -4 \\ 1 & -4 & 15 \end{pmatrix}$$

### 2.3.1 Erreur

$f(x)$  doit être 4 fois continûment dérivable et les points équi-distants

$$\begin{aligned} |f(x) - s(x)| &\leq C_0 h^4 \max_{\zeta \in I} |f^{(4)}(\zeta)| \\ |f'(x) - s'(x)| &\leq C_1 h^3 \max_{\zeta \in I} |f^{(4)}(\zeta)| \\ |f''(x) - s''(x)| &\leq C_2 h^2 \max_{\zeta \in I} |f^{(4)}(\zeta)| \\ |f'''(x) - s'''(x)| &\leq C_3 h \max_{\zeta \in I} |f^{(4)}(\zeta)| \quad x \neq x_0, x_1, \dots \end{aligned}$$

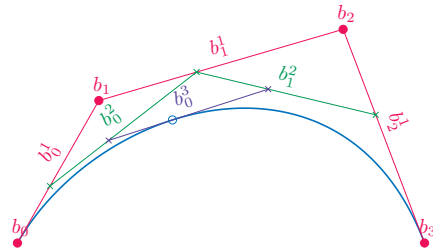
### 2.3.2 Courbes paramétrique

Faire une spline pour chaque coordonnée ( $x$ ,  $y$ , etc...) en fonction de  $t$ . Possibilité de faire une approximation

$$t_{i+1} - t_i \approx \text{distance entre } (x_i, y_i) \text{ et } (x_{i+1}, y_{i+1})$$

## 3 Courbes et surfaces

### 3.1 Courbes de Bézier (De Casteljau)



ordre  $n \longleftrightarrow n + 1$  points

$b_{\text{départ}}^{\text{degré}}$

Exemples :

$$\begin{aligned} b_0^1(t) &= (1-t)b_0 + tb_1 & b_0 &\rightarrow b_1 \\ b_1^1(t) &= (1-t)b_1 + tb_2 & b_1 &\rightarrow b_2 \\ b_0^2(t) &= (1-t)b_0^1(t) + tb_1^1(t) & b_0 &\rightarrow b_2 \end{aligned}$$

La courbe de Bézier est comprise dans le polygone de contrôle (points de contrôles reliés).

#### 3.1.1 Schéma triangulaire

On représente les points de base sur la première colonne

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 2 \\ 2.5 \end{bmatrix} \quad \begin{bmatrix} 4/3 \\ 3/2 \end{bmatrix} \quad \begin{bmatrix} 4 \\ 1.5 \end{bmatrix} \quad \begin{bmatrix} 8/3 \\ 13/6 \end{bmatrix} \quad \begin{bmatrix} 16/9 \\ 31/18 \end{bmatrix}$$

On prend la valeur du haut multipliée par  $(1-t)$  et celle du bas par  $t$

#### 3.1.2 Généralisation

$$b_0^3(t) = (1-t)b_0^2 + tb_1^2(t)$$

### 3.2 Polynômes de Bernstein

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad i = 0, 1, \dots, n \quad t \in [0, 1]$$

$n$	$i = 0$	$i = 1$	$i = 2$	$i = 3$
0	$B_0^0(t) = 1$			
1	$B_0^1(t) = 1 - t$	$B_1^1(t) = t$		
2	$B_0^2(t) = (1-t)^2$	$B_1^2(t) = 2t(1-t)$	$B_2^2(t) = t^2$	
3	$B_0^3(t) = (1-t)^3$	$B_1^3(t) = 3t(1-t)^2$	$B_2^3(t) = 3t^2(1-t)$	$B_3^3(t) = t^3$

$$\binom{n}{i} = C_i^n = \frac{n!}{i!(n-i)!}$$

La somme des polynômes donne 1

$$\sum_{i=0}^n B_i^n(t)$$

- $t = 0$  est un zéro de multiplicité  $i$
- $t = 1$  est un zéro de multiplicité  $n - i$
- $B_i^n(t) = B_{n-i}^n(1-t)$
- $B_i^n(t) = t B_{i-1}^{n-1}(t) + (1-t) B_i^{n-1}(t) \quad i = 1, 2, \dots, n-1$

Exemples :

$$\begin{aligned} b_0^1(t) &= b_0 B_0^1(t) + b_1 B_1^1(t) \\ b_1^1(t) &= b_1 B_0^1(t) + b_2 B_1^1(t) \\ b_0^2(t) &= b_0 B_0^2(t) + b_1 B_1^2(t) + b_2 B_2^2(t) \end{aligned}$$

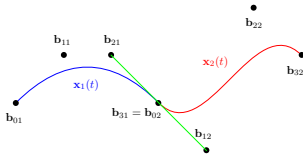
$$x(t) = b_0 B_0^3(t) + b_1 B_1^3(t) + b_2 B_2^3(t) + b_3 B_3^3(t)$$

### 3.3 Bernstein et Bézier

$$b_i^r(t) = \sum_{j=0}^r b_{i+j} B_j^r(t)$$

### 3.4 Courbes composées

1. Deux extrémités égales (points superposés)
2. Points autour de l'extrémité alignés



## 4 Intégration numérique

### 4.1 Formule du trapèze

$$\int_a^b f(x)dx \approx \frac{b-a}{2} (f(a) + f(b))$$

### 4.2 Formule composite du trapèze

Formule du trapèze avec sous-division ( $n$  intervalles)

$$h = \frac{b-a}{n}$$

$$\int_a^b f(x)dx \approx h \left( \frac{1}{2}f(a) + \sum_{j=1}^{n-1} f(x_j) + \frac{1}{2}f(b) \right)$$

### 4.3 Formule du point milieu

$$M = h(f(x_{0.5}) + f(x_{1.5}) + \dots + f(x_{n-0.5}))$$

$$T\left(\frac{h}{2}\right) = \frac{1}{2} (T(h) + M(h))$$

#### 4.3.1 Algorithme pour $n = 4$

1.  $h = b - a$
2.  $T(h) = \frac{h}{2} (f(x_0) + f(x_4))$
3.  $M(h) = hf(x_2)$

$$4. T\left(\frac{h}{2}\right) = \frac{1}{2} (M(h) + T(h))$$

$$5. M\left(\frac{h}{2}\right) = \frac{h}{2} (f(x_1) + f(x_3))$$

$$6. T\left(\frac{h}{4}\right) = \frac{1}{2} \left( M\left(\frac{h}{2}\right) + T\left(\frac{h}{2}\right) \right)$$

#### 4.3.2 Erreur

$$\left| \int_a^b f(x)dx - T(h) \right| \leq \frac{h^2(b-a)}{12} \max_{a \leq x \leq b} |f''(x)|$$

Optimal si

1. La fonction est **périodique**
2. La fonction est **infiniment dérivable**
3. On intègre sur une période

### 4.4 Méthode de Simpson

$$\int_a^b f(x)dx \approx \frac{h}{3} \left( f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right)$$

$$h = \frac{b-a}{2}$$

#### 4.4.1 Erreur

$$\left| \int_a^b f(x)dx - S \right| \leq \frac{h^5}{90} \max_{a \leq x \leq b} |f^{(4)}(x)|$$

### 4.5 Formule de Newton-Cotes

Avec  $n = 3$  (3/8 de Simpson)

$$\int_{x_0}^{x_3} f(x)dx \approx \frac{3h}{8} (f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3))$$

$$\int_{x_0}^{x_4} f(x)dx \approx \frac{2h}{45} (7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4))$$

Non composite  $\rightarrow$  utiliser les formules tel quel (pas de séparation en sous-intervalles).

$n$  pair : polynômes jusqu'à  $n + 1$ .  $n$  impair : polynômes jusqu'à  $n$

### 4.6 Formule de composition de Simpson

Cas général avec  $2n$  sous intervalles ( $n$  polynômes)

$$S_c = \frac{h}{3} \left( f(a) + 4f(x_1) + f(b) + 2 \sum_{k=1}^{n-1} (f(x_{2k}) + 2f(x_{2k+1})) \right)$$

$$h = \frac{b-a}{2n}$$

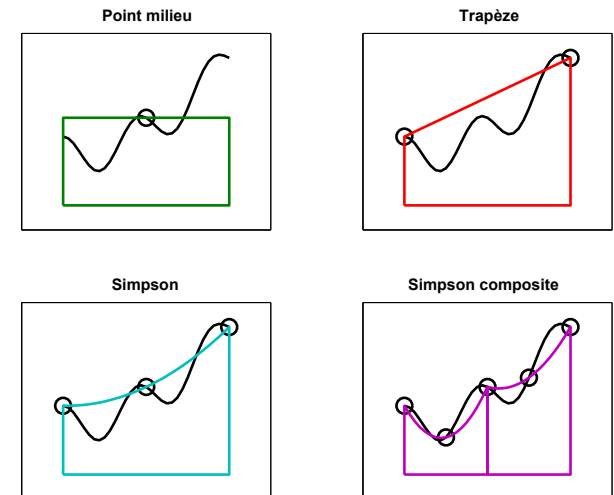
#### 4.6.1 Erreur

$$\left| \int_a^b f(x)dx - S_c \right| \leq \frac{h^4(b-a)}{180} \max_{a \leq x \leq b} |f^{(4)}(x)|$$

### 4.7 Formule de Simpson adaptative

Intervalles non uniformes

### 4.8 Récapitulatif



### 4.9 Romberg

$n = 4 \rightarrow$  5 lignes et 5 colonnes

$$h/2 \begin{matrix} \downarrow \\ \begin{matrix} T_{0,0} & & & & & \\ T_{1,0} & T_{1,1} & & & & \\ T_{2,0} & T_{2,1} & T_{2,2} & & & \\ T_{3,0} & T_{3,1} & T_{3,2} & T_{3,3} & & \\ T_{4,0} & T_{4,1} & T_{4,2} & T_{4,3} & T_{4,4} & \\ T_{5,0} & T_{5,1} & T_{5,2} & T_{5,3} & T_{5,4} & T_{5,5} \end{matrix} \end{matrix}$$

$$T_{n,c} = \frac{4^c T_{n,c-1} - T_{n-1,c-1}}{4^c - 1}$$

$$T_{0,0} = T(h)$$

$$T_{1,0} = T(h/2)$$

$$T_{2,0} = T(h/4)$$

$$T_{n,0} = T(h/2^n)$$

1. Facile
2. Coûteuse pour une grande précision
3. Il faut que la fonction soit  $2k + 2$  fois continûment dérivable pour aller jusqu'à la colonne  $k$

#### 4.10 Choix de la méthode

1. Périodique + infiniment dérivable : Trapèze
2. Polynôme cubique : Simpson non-composite (pas d'erreur)
3. Infiniment dérivable non périodique : Gauss ou Simpson adaptative (Romberg plus coûteuse)

#### 4.11 Polynômes de Legendre

$$P_n(x) = \frac{1}{2^n \cdot n!} \frac{d^n}{dx^n} ((x^2 - 1)^n)$$

$$P_0(x) = 1 \quad P_3(x) = \frac{1}{2}(5x^3 - 3x)$$

$$P_1(x) = x \quad P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3)$$

$$P_2(x) = \frac{1}{2}(3x^2 - 1) \quad P_5(x) = \frac{1}{8}(63x^5 - 70x^3 + 15x)$$

## 5 Équations non linéaires

### 5.1 Existence d'une solution

Solution  $f(r) = 0$  entre  $a$  et  $b$  pour  $f$  continue

$$f(a) \cdot f(b) < 0$$

### 5.2 Bissection

1.  $a_0 = a, b_0 = b, x_0 = \frac{a+b}{2}$
2. Répéter jusqu'à ce que  $|a_k - b_k| \geq tol \cdot |b_k|$ 
  - (a) Si  $f(x_k)f(a_k) < 0 : a_{k+1} = a_k, b_{k+1} = x_k$
  - (b) Si  $f(x_k)f(a_k) > 0 : a_{k+1} = x_k, b_{k+1} = b_k$
  - (c)  $x_k = \frac{a_k + b_k}{2}$

L'erreur converge avec

$$|e_k| < \frac{b-a}{2^{k+1}}$$

1. Robuste
2. Intervalles qui contiennent la solution
3. Majorant connu de l'erreur
4. Lente

### 5.3 Regula falsi

A l'exception du premier terme, on doit décider si on utilise  $x_{k-1}$  avec  $x_{k-2}$  ou  $x_{k-3}$

$$x_k = \begin{cases} x_{k-2} - y_{k-2} \frac{x_{k-1} - x_{k-2}}{y_{k-1} - y_{k-2}} & y_{k-1} \cdot y_{k-2} < 0 \\ x_{k-3} - y_{k-3} \frac{x_{k-1} - x_{k-3}}{y_{k-1} - y_{k-3}} & y_{k-1} \cdot y_{k-3} < 0 \end{cases}$$

### 5.4 Sécante

$$x_{k+1} = x_k - f(x_k) \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

Ordre de convergence de

$$p = \frac{1 + \sqrt{5}}{2} = \phi$$

1. Pas de connaissance de la dérivée
2. Une seule évaluation de  $f$
3. Plus efficace que Newton (plus facile à calculer)
4. Convergence non garantie

### 5.5 Newton

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$|e_{k+1}| \approx \left| \frac{f''(r)}{f'(r)} \right| \cdot |e_k|^2$$

1. Très rapide
2. Peut ne pas converger
3. Utilisation de la dérivée

### 5.6 Point fixe

$$x = F(x)$$

$$x_{k+1} = F(x_k)$$

Résultat global mais difficile de trouver un domaine  $D$ ... Il existe une version locale

$$\rho(\mathbf{J}(\mathbf{r})) < 1 \longrightarrow \text{converge dans un disque } D \text{ vers } \mathbf{r}$$

Avec  $\rho$  le rayon spectral (maximum des modules des valeurs propres de la matrice).

#### 5.6.1 Théorème de Banach

$$L = \max_{x \in I} |F'(x)| < 1 \longrightarrow \text{contraction sur } I$$

$$|F(x_1) - F(x_2)| \leq L|x_1 - x_2|$$

$$0 < L < 1 \quad \forall x_1, x_2 \in I$$

Alors il existe une seule solution  $r$  dans  $I$

1. Fonction Lipschitzienne : qui satisfait la condition de Lipschitz avec n'importe quel  $L$
2. Fonction contractante : qui satisfait la condition de Lipschitz et  $0 < L < 1$

$$L_{\text{optimal}} = \max_{x \in I} |F'(x)|$$

### 5.6.2 Estimation

1. A priori :

$$\|\mathbf{r} - \mathbf{x}_k\| \leq \frac{L^k}{1-L} \|\mathbf{x}_1 - \mathbf{x}_0\|$$

2. A posteriori :

$$\|\mathbf{r} - \mathbf{x}_k\| \leq \frac{L}{1-L} \|\mathbf{x}_k - \mathbf{x}_{k-1}\|$$

## 6 Systèmes d'équations non-linéaires

### 6.1 Point fixe

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} F(x, y) \\ G(x, y) \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$$

$\mathbf{F}$  est une contraction si

$$\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{x}')\| \leq L \|\mathbf{x} - \mathbf{x}'\| \quad \|\mathbf{x}\| = \sqrt{x^2 + y^2}$$

Et donc  $\mathbf{F}$  possède un seul point fixe (et y converge de toute façon).

$$\max_{(x,y) \in D} \|\mathbf{J}(x, y)\| < 1 \longrightarrow \text{contraction sur } D$$

$$\|\mathbf{J}(x, y)\| = \sqrt{F_x(x, y)^2 + F_y(x, y)^2 + G_x(x, y)^2 + G_y(x, y)^2}$$

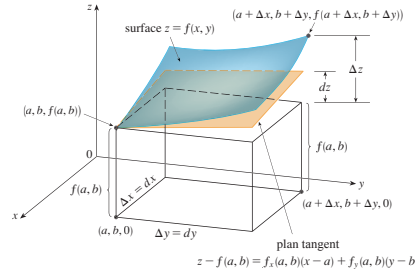
Matrice de Jacobi :

$$\mathbf{J}(x, y) = \begin{pmatrix} F_x(x, y) & F_y(x, y) \\ G_x(x, y) & G_y(x, y) \end{pmatrix}$$

### 6.2 Méthode de Newton

$$x_{k+1} = x_k + \frac{g(x_k, y_k)f_y(x_k, y_k) - f(x_k, y_k)g_y(x_k, y_k)}{f_x(x_k, y_k)g_y(x_k, y_k) - f_y(x_k, y_k)g_x(x_k, y_k)}$$

$$y_{k+1} = y_k + \frac{f(x_k, y_k)g_x(x_k, y_k) - g(x_k, y_k)f_x(x_k, y_k)}{f_x(x_k, y_k)g_y(x_k, y_k) - f_y(x_k, y_k)g_x(x_k, y_k)}$$



La matrice de Jacobi donne une information sur la convergence (lente si déterminant égal à 0)

#### 6.2.1 Ordre

Convergence linéaire

$$\|\mathbf{e}_{k+1}\| \lesssim \|\mathbf{J}(r, s)\| \cdot \|\mathbf{e}_k\|$$

Si  $\mathbf{J}(r, s)$  est nulle on a un ordre de convergence quadratique.

## 7 Résolution numérique de systèmes linéaires

Système mal conditionné  $\kappa_\infty(A)$  très grand

$$a_{11}x_1 + \dots + a_{1n}x_n = b_1$$

$\vdots$

$$a_{n1}x_1 + \dots + a_{nn}x_n = b_n$$

$$\underbrace{\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}}_A \underbrace{\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}}_{\vec{x}} = \underbrace{\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}}_{\vec{b}}$$

Si peu d'éléments sont non-nuls alors  $A$  est dite **creuse**

### 7.1 Condition d'arrêt

$$\|\vec{r}_k\| = \|\vec{b} - A\vec{x}_k\|$$

Tolérance fixe  $\tau$  (par exemple  $10^{-5}$ )

$$\|\vec{r}_k\| \leq \tau \|\vec{b}\|$$

Erreur :

$$\vec{e}_k = \vec{x} - \vec{x}_k$$

On peut aussi utiliser une condition d'arrêt sur l'erreur  $\vec{e}_k$  au lieu du résidu  $\vec{r}_k = \vec{b} - A\vec{x}_k$

#### 7.1.1 Lien entre résidu et erreur

$$\frac{\|\vec{x} - \vec{x}_k\|}{\|\vec{x}_k\|_p} \leq \underbrace{\|A\|_p \|A^{-1}\|_p}_{\kappa_p(A)} \frac{\|\vec{b} - A\vec{x}_k\|_p}{\|\vec{b}\|_p}$$

Autant de digits valides dans la mantisse que

$$N_{\text{digits}} = |\log_{10}(\varepsilon)| - \log_{10}(\kappa(A)_p)$$

Avec  $\varepsilon$  la précision machine (1e-16 en général)

#### 7.1.2 Perturbation

Perturbation sur  $A$

$$\frac{\|\delta \vec{x}_A\|}{\|\vec{x} + \delta \vec{x}_A\|} \leq \|\vec{A}\| \cdot \|\vec{A}^{-1}\| \cdot \frac{\|\delta A\|}{\|A\|}$$

Perturbation sur  $A$  et  $\vec{b}$ :

$$\frac{\|\delta \vec{x}\|}{\|\vec{x} + \delta \vec{x}\|} \leq \frac{\|\vec{A}\| \cdot \|\vec{A}^{-1}\|}{1 - \|A\| \cdot \|A^{-1}\| \frac{\|\delta A\|}{\|A\|}} \cdot \left( \frac{\|\delta A\|}{\|A\|} + \frac{\|\delta \vec{b}\|}{\|\vec{b}\|} \right)$$

## 7.2 Normes

$$\|\vec{v}\|_p = \left( \sum_{i=1}^n |v_i|^p \right)^{\frac{1}{p}}$$

### 1. Vecteurs

- (a) 1-norme : somme des composantes
- (b) 2-norme : norme euclidienne
- (c) max-norme :  $p \rightarrow \infty$  max des valeurs absolues

### 2. Matrices

- (a) 1-norme :  $\max(\sum |A|)$  (la somme est colonne par colonne).
- (b) 2-norme : ou max des valeurs propres de  $A^T A$
- (c) max-norme :  $\max(\sum |A|)$  (la somme est ligne par ligne).

1.  $\|\vec{v}\| = 0 \iff \vec{v} = \vec{0}$
2.  $\|\lambda \vec{v}\| = |\lambda| \cdot \|\vec{v}\|$
3.  $\|\vec{v} + \vec{u}\| \leq \|\vec{v}\| + \|\vec{u}\|$

## 7.3 Méthodes directes

### 7.3.1 Élimination de Gauss sans pivot

Effectuer des combinaisons linéaires des lignes pour obtenir une matrice triangulaire supérieure. Matrice augmentée :

$$\left( \begin{array}{ccc|c} 10 & -7 & 0 & 7 \\ 0 & -0.1 & 6 & 6.1 \\ 0 & 2.5 & 5 & 2.5 \end{array} \right)$$

1. Commencer par la colonne de gauche
2. Si nécessaire, permuter les lignes pour avoir un non-nul comme premier élément
3. Faire les combinaisons linéaires des lignes pour annuler les éléments inférieurs
4. Passer à la colonne suivante

### 7.3.2 Élimination de Gauss avec pivot

On fait le premier pas normalement (0 en dessous du premier élément). Mais avant d'effectuer le prochain pas, on échange les lignes pour avoir le plus grand élément en diagonale. A la fin on se retrouve avec, par exemple :

$$A = \begin{pmatrix} 10 & x & x \\ 0 & 6 & x \\ 0 & 0 & 0.01 \end{pmatrix}$$

## 7.4 LU

$$A\vec{x} = \vec{b} \longrightarrow L \underbrace{U\vec{x}}_{\vec{y}} = \vec{b}$$

L'avantage est que si on change  $b$ , il n'y a pas besoin de tout recommencer.

**Avec pivotement** Si on applique le pivotement, on a une matrice de permutation  $P$

$$LU = PA$$

### 7.4.1 Doolittle

Méthode équivalente à l'élimination de Gauss

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

$$\begin{pmatrix} \textcircled{1} a_{11} & a_{12} & a_{13} \\ a_{21} & \textcircled{2} a_{22} & a_{23} \\ a_{31} & a_{32} & \textcircled{3} a_{33} \end{pmatrix}$$

$$\begin{aligned} (1) \begin{cases} u_{11} &= a_{11} \\ u_{12} &= a_{12} \\ u_{13} &= a_{13} \end{cases} \quad (2) \begin{cases} l_{21} &= \frac{a_{21}}{u_{11}} \\ l_{31} &= \frac{a_{31}}{u_{11}} \end{cases} \\ (3) \begin{cases} u_{22} &= a_{22} - l_{21}u_{12} \\ u_{23} &= a_{23} - l_{21}u_{13} \end{cases} \quad (4) \begin{cases} l_{32} &= \frac{a_{32} - l_{31}u_{12}}{u_{22}} \end{cases} \\ (5) \begin{cases} u_{33} &= a_{33} - l_{31}u_{13} - l_{32}u_{23} \end{cases} \end{aligned}$$

Formules générales (au cas ou matrice plus petite ou plus grande) :

$$u_{km} = a_{km} - \sum_{j=1}^{k-1} l_{kj}u_{jm} \quad m = k, k+1, k+2, \dots, n$$

$$l_{ik} = \frac{1}{u_{kk}} \left( a_{ik} - \sum_{j=1}^{k-1} l_{ij}u_{jk} \right) \quad i = k+1, k+2, \dots$$

### 7.4.2 Cholesky

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} = \begin{pmatrix} \hat{l}_{11} & 0 & 0 \\ \hat{l}_{21} & \hat{l}_{22} & 0 \\ \hat{l}_{31} & \hat{l}_{32} & \hat{l}_{33} \end{pmatrix} \cdot \begin{pmatrix} \hat{l}_{11} & \hat{l}_{21} & \hat{l}_{31} \\ 0 & \hat{l}_{22} & \hat{l}_{32} \\ 0 & 0 & \hat{l}_{33} \end{pmatrix}$$

$$\begin{aligned} \hat{l}_{11} &= \sqrt{a_{11}} & \hat{l}_{21} &= \frac{a_{12}}{\sqrt{a_{11}}} \\ \hat{l}_{31} &= \frac{a_{13}}{\sqrt{a_{11}}} & \hat{l}_{22} &= \sqrt{a_{22} - \frac{a_{12}^2}{a_{11}}} \\ \hat{l}_{32} &= \frac{a_{23} - \frac{a_{13}a_{12}}{a_{11}}}{\sqrt{a_{22} - \frac{a_{12}^2}{a_{11}}}} & \hat{l}_{33} &= \sqrt{a_{33} - \hat{l}_{31}^2 - \hat{l}_{32}^2} \end{aligned}$$

### 7.4.3 Permutations

Attention, si on utilise des permutations, alors

$$PA\vec{x} = P\vec{b}$$

#### 7.4.4 Méthode QR

$R$  est triangulaire supérieur (en dessous-gauche de la diagonale : que des 0)

$$A = QR \longrightarrow R\vec{x} = Q^{-1}\vec{b} \longrightarrow R\vec{x} = \underbrace{Q^T\vec{b}}_{\vec{c}}$$

$$\begin{pmatrix} \vec{a}_1 & \cdots & \vec{a}_n \end{pmatrix} = \begin{pmatrix} \vec{q}_1 & \cdots & \vec{q}_m \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Avec les propriétés de  $Q$  :

- $Q^T Q = I$   $Q^{-1} = Q^T$
- $\det(Q) = \pm 1$

On commence avec une matrice  $A_{n \times n}$

1. Prendre le premier vecteur colonne de  $A$  :  $\vec{x}_1$
2. Déterminer  $\vec{y}$

$$y = -\rho \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = -\text{signe}(a_{ii})|x_i| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \left\{ \text{taille de } x \right.$$

3. Déterminer  $\vec{v}$

$$\vec{v} = \vec{x} - \vec{y}$$

$$\gamma = \frac{\|\vec{v}\|^2}{2}$$

4. Faire la décomposition pour obtenir  $H_1$

$$\tilde{H} = I - \frac{2}{\|\vec{v}\|^2} \vec{v} \vec{v}^T$$

$$H = \begin{pmatrix} I & 0 \\ 0 & \tilde{H} \end{pmatrix} \quad (\text{si nécessaire})$$

5. Construire  $H_1 A$  puis prendre  $v_2$  (première colonne de  $H_1 A$  **sans la première ligne et sans la première colonne**)
6. Recommencer jusqu'à avoir terminé
7. à la fin, déterminer  $R$  et  $Q$

$$H_n H_{n-1} \cdots H_2 H_1 A = R \longrightarrow A = \underbrace{H_1 H_2 \cdots H_{n-1} H_n^T}_{Q} R$$

#### 7.4.5 Système creux

: Réseaux hydrauliques, thermiques, électriques, etc... Possibilité de faire du "fill-in" si  $n$  ou  $b$  sont grands mais le coup sera inutilement grand.

#### 7.4.6 Valeur $\kappa$

$$\kappa(A)_p = \|A\|_p \|A^{-1}\|_p$$

En général on utilisera  $\kappa_2$  si ce n'est pas précisé (mais des fois on calcule  $\kappa_\infty$ )

#### 7.5 Résolution au sens des moindres carrés

On résout le système

$$A^T A \vec{x} = A^T \vec{b}$$

Si  $A$  est mal conditionné,  $A^T A$  est doublement mal conditionné

$$\kappa(A^T A) \approx \kappa^2(A)$$

Pour obtenir le nombre de digits corrects, on fait :

$$N_{\text{digits}} = |\log_{10}(\epsilon)| - \log_{10}(\kappa)$$

Avec  $QR$ , on a une méthode plus stable

$$A_{m \times n} = Q_{m \times m} \begin{pmatrix} R^* \\ 0 \end{pmatrix}_{m \times n}$$

#### 7.5.1 Minimisation

$$\|\vec{b} - A\vec{x}\| = \|Q Q^T \vec{b} - Q R \vec{x}\| = \left\| \underbrace{Q^T \vec{b}}_{\vec{c}} - R \vec{x} \right\|$$

1. Trouver  $Q$  et  $R$  avec la méthode au dessus
2. Calculer  $\vec{c}$

$$\vec{c} = Q^T \vec{b}$$

3. Trouver  $R^*$  (partie "supérieure" de  $R$ , donc sans les 0 en bas)

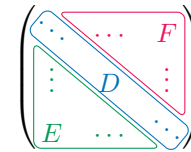
4. Trouver  $c^*$  (partie "supérieure" de  $\vec{c}$ , en ignorant la valeur du bas pour avoir la même hauteur que  $R^*$ )

5. Poser le système

$$R^* \vec{x} = \vec{c}^*$$

6. Résoudre à la main, en inversant, c'est égal

#### 7.6 Méthodes itératives



##### 7.6.1 Méthode Jacobi

On commence avec un  $\vec{x}(0) = \vec{0}$

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

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}^{(k+1)} = \begin{pmatrix} \frac{1}{a_{11}} \cdot (b_1 - a_{12}x_2^{(k)} \cdots - a_{1n}x_n^{(k)}) \\ \frac{1}{a_{22}} \cdot (b_2 - a_{21}x_1^{(k)} \cdots - a_{2n}x_n^{(k)}) \\ \vdots \\ \frac{1}{a_{nn}} \cdot (b_n - a_{n1}x_1^{(k)} \cdots - a_{nn-1}x_{n-1}^{(k)}) \end{pmatrix}$$

Alternativement :

$$\vec{x}^{(k)} = D^{-1} \left( \vec{b} - (A - D) \vec{x}^{(k-1)} \right)$$

$$N = D^{-1} = \text{diag}^{-1}(A)$$

Convergence assurée sur  $A$  est **strictement diagonalement dominante** 7.8

### 7.6.2 Méthode de Gauss-Seidel

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

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}^{(k+1)} = \begin{pmatrix} \frac{1}{a_{11}} \cdot (b_1 - a_{12}x_2^{(k)} - \dots - a_{1n}x_n^{(k)}) \\ \frac{1}{a_{22}} \cdot (b_2 - a_{21}x_1^{(k+1)} - \dots - a_{2n}x_n^{(k)}) \\ \vdots \\ \frac{1}{a_{nn}} \cdot (b_n - a_{n1}x_1^{(k+1)} - \dots - a_{nn-1}x_{n-1}^{(k+1)}) \end{pmatrix}$$

Alternativement :

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

$$N = (D + E)^{-1}$$

Gauss-Seidel est bien meilleur que la méthode de Jacobi.

Convergence assurée sur  $A$  est **strictement diagonalement dominante** 7.8

### 7.6.3 Méthode SOR

$$\vec{x}^{(k+1)} = (D + \omega E)^{-1} \cdot (\omega \vec{b} - (\omega F + (\omega - 1)D)\vec{x}^{(k)})$$

### 7.6.4 Itération simple

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + N (\vec{b} - A\vec{x}^{(k)})$$

$$\max_i |\lambda_i| < 1 \longrightarrow \text{convergence assurée pour tout } \vec{b}$$

## 7.7 Convergence

De manière générale on a une expression de la forme

$$\vec{x}^{(k)} = \vec{x}^{(k-1)} + N (\vec{b} - A\vec{x}^{(k-1)})$$

Avec un  $N$  qui se rapproche de  $A^{-1}$  pour que le système soit rapide (mais sans être trop dur à calculer).

$$\rho(I - NA) = \max(\text{valeurs propres}(I - NA))$$

$$\rho < 1 \longrightarrow \text{convergence garantie}$$

### 7.7.1 Erreur

$$e^{(k)} = \underbrace{(I - NA)}_{\text{matrice d'itération}} e^{(k-1)}$$

$$\|\vec{e}^{(k)}\| \leq \rho^k \|\vec{e}^{(0)}\|$$

Relation entre le nombre de décimales souhaitées  $D$  et le nombre d'itérations  $n$

$$n \geq \frac{\log_{10}(10^{-D})}{\log_{10}(\rho)} = -\frac{D}{\log_{10}(\rho)}$$

## 7.8 Matrice strictement diagonalement dominante

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

$a_{ii}$  est la plus grande valeur sur chaque ligne

## 7.9 Méthodes de minimisation $A\vec{x} - \vec{b}$

$$\vec{r} = \vec{b} - A\vec{x}$$

**Gradient**

$$\nabla G(\vec{x}) = \begin{pmatrix} G_x \\ G_y \\ G_z \end{pmatrix}$$

Si on multiplie par la transposée de  $\vec{n}$  on obtient la dérivée directionnelle.

$$\frac{\partial G}{\partial \vec{n}} = \vec{n}^T \nabla G(\vec{x})$$

Il existe deux méthodes de résolutions et deux algorithmes ( $F$  fonctionne toujours et  $G$  fonctionne si  $A$  est symétrique et semi-définie positive, voir définition plus bas)

### 7.9.1 Méthode 1 : $F(x) = \frac{1}{2} \|b - A\vec{x}\|_2^2$

On cherche à minimiser le résidu  $\vec{r}$  (fonctionne seulement si  $A\vec{x} = \vec{b}$  possède au moins une solution)

$$\vec{d} = A^T (\vec{b} - A\vec{x})$$

## Algorithme de la plus grande pente

$$\vec{d}_k = -\nabla F = A^T (\vec{b} - A\vec{x}_k)$$

$$\vec{x}_{k+1} = \vec{x}_k + \vec{d}_k$$

**Algorithme des gradients conjugués** (pas sur que cet algorithme peut être utilisé avec cette méthode) On utilise le résidu précédent et le résidu actuel pour optimiser encore plus l'itération

$$\vec{d}_{k+1} = \vec{r}_{k+1} + \beta_k \vec{d}_k$$

$$\beta = \frac{\vec{r}_{k+1}^T \vec{r}_{k+1}}{\vec{r}_k^T \vec{r}_k}$$

$$\vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{d}_k$$

### 7.9.2 Méthode 2 : $G(x) = \frac{1}{2} \vec{x}^T A \vec{x} - \vec{b}^T \vec{x}$

Deux notations possibles :

$$G(x) = \frac{1}{2} \vec{x}^T A \vec{x} - \vec{x}^T \vec{b} = \frac{1}{2} \vec{x}^T A \vec{x} - \vec{b}^T \vec{x}$$

$A$  doit être semi-définie positive (valeurs propres de  $A$  plus grandes ou égales à 0)

$$\vec{x}^T A \vec{x} \geq 0$$

$$\vec{d} = \vec{r} = \vec{b} - A\vec{x}$$

## Algorithme de la plus grande pente (zig-zag)

$$\vec{d}_k = -\nabla F = A^T (\vec{b} - A\vec{x}_k)$$

$$\vec{x}_{k+1} = \vec{x}_k + \vec{d}_k$$

Si on utilise  $\alpha$  optimal, alors toutes les itérations sont à angle droit. Calcul de  $\alpha$  optimal :

$$\alpha_k = \frac{\vec{d}_k^T \vec{r}_k}{\vec{d}_k^T A \vec{d}_k}$$



**Algorithme des gradients conjuguées** (réponse après  $n$  itérations, avec  $n$  la largeur de la matrice  $A$ )  
On utilise le résidu précédent et le résidu actuel pour optimiser encore plus l'itération

$$\vec{d}_{k+1} = \vec{r}_{k+1} + \beta_k \vec{d}_k$$

$$\beta = \frac{\vec{r}_{k+1}^T \vec{r}_{k+1}}{\vec{r}_k^T \vec{r}_k}$$

$$\vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{d}_k$$

## 8 Résolution EDO

$$y' = f(t, y(t))$$

### 8.1 Méthodes explicites

#### 8.1.1 Méthode d'Euler explicite

$$y_{k+1} = y_k + hf(t_k, y_k)$$

$$||\vec{y}_n - \vec{y}(T)|| \leq Ch$$

- 1 palier
- ordre 2
- Énergie totale non conservée
- $h$  suffisamment petit si on veut être stable

#### 8.1.2 Méthode de Heun

$$\vec{u}_{k+1} = \vec{u}_k + \frac{h}{2} (\vec{s}_1 + \vec{s}_2)$$

$$\vec{s}_1 = \vec{f}(t_k, \vec{u}_k) \quad \vec{s}_2 = \vec{f}(t_k + h, \vec{u}_k + h\vec{s}_1)$$

- 2 paliers
- Ordre 2
- $h$  suffisamment petit si on veut être stable

$$\begin{array}{c|c} 0 & \frac{1}{2} \\ \hline \frac{1}{2} & 0 \end{array} \quad \begin{array}{c|c} \frac{1}{2} & 1 \\ \hline 0 & 1 \end{array}$$

#### 8.1.3 Méthode optimale

$$\vec{u}_{k+1} = \vec{u}_k + h \left( \frac{1}{4} \vec{s}_1 + \frac{3}{4} \vec{s}_2 \right)$$

$$\vec{s}_1 = f(t_k, \vec{u}_k)$$

$$\vec{s}_2 = f \left( t_k + \frac{2}{3}h, \vec{u}_k + \frac{2}{3}h\vec{s}_1 \right)$$

$$\begin{array}{c|c} 0 & \frac{2}{3} \\ \hline \frac{2}{3} & \frac{1}{4} \end{array} \quad \begin{array}{c|c} \frac{2}{3} & \frac{3}{4} \\ \hline \frac{1}{4} & \frac{3}{4} \end{array}$$

#### 8.1.4 Crank-Nicolson

$$\begin{array}{c|cc} 0 & & \\ \hline 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}$$

#### 8.1.5 Runge-Kutta 2

$$\vec{u}_{k+1} = \vec{u}_k + h (w_1 \vec{s}_1 + w_2 \vec{s}_2)$$

$$\vec{s}_1 = \vec{f}(t_k, \vec{u}_k) \quad \vec{s}_2 = \vec{f}(t_k + c_2 h, \vec{u}_k + a_{21} h \vec{s}_1)$$

$$\begin{array}{c|c} 0 & \\ \hline c_2 & a_{21} \\ \hline & w_1 \quad w_2 \end{array}$$

1. Méthode à 2 paliers
2. Ordre 2
- 3.

#### 8.1.6 Runge-Kutta 3

$$\vec{u}_{k+1} = \vec{u}_k + h (w_1 \vec{s}_1 + w_2 \vec{s}_2 + w_3 \vec{s}_3)$$

$$\vec{s}_1 = \vec{f}(t_k, \vec{u}_k)$$

$$\vec{s}_2 = \vec{f}(t_k + c_2 h, \vec{u}_k + a_{21} h \vec{s}_1)$$

$$\vec{s}_3 = \vec{f}(t_k + c_3 h, \vec{u}_k + a_{31} h \vec{s}_1 + a_{32} h \vec{s}_2)$$

$$\begin{array}{c|ccc} 0 & & & \\ \hline c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & w_1 & w_2 & w_3 \end{array}$$

#### 8.1.7 Runge-Kutta 4

$$\vec{u}_{k+1} = \vec{u}_k + \frac{h}{6} (\vec{s}_1 + 2\vec{s}_2 + 2\vec{s}_3 + \vec{s}_4)$$

$$\vec{s}_1 = \vec{f}(t_k, \vec{u}_k)$$

$$\vec{s}_2 = \vec{f} \left( t_k + \frac{h}{2}, \vec{u}_k + \frac{h}{2} \vec{s}_1 \right)$$

$$\vec{s}_3 = \vec{f} \left( t_k + \frac{h}{2}, \vec{u}_k + \frac{h}{2} \vec{s}_2 \right)$$

$$\vec{s}_4 = \vec{f}(t_k + h, \vec{u}_k + h\vec{s}_3)$$

$$\begin{array}{c|ccc} 0 & & & \\ \hline \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ \hline 1 & 0 & 0 & 1 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

1. Méthode à 4 paliers
2. Ordre 4
3. Conditionnellement stable

### 8.1.8 Stabilité

Pour que le système soit stable (pour un système linéaire à une équation), il faut que

$$h < \frac{2}{\lambda}$$

Avec

$$\frac{du}{dt} = -\lambda u(t)$$

Pour un système de la forme

$$\frac{d\vec{u}}{dt} = -A\vec{u}$$

On va chercher le max des valeurs propres

$$h < \frac{2}{|\lambda_{\max}(A)|}$$

Pour un problème non-linéaire, on peut faire une linéarisation autour de  $u_0$  et  $t_0$

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

## 8.2 Méthodes implicites

### 8.2.1 Méthode de Euler implicite

Sois un système de la forme :

$$\frac{du}{dt} = f(t, u(t))$$

$$u(t_0) = u_0$$

La méthode d'Euler implicite est la suivante :

$$\left. \frac{du}{dt} \right|_{t_k} = f(t_k, u_k)$$

$$\left. \frac{du}{dt} \right|_{t_k} \approx \frac{\vec{u}_k - \vec{u}_{k-1}}{h}$$

$$\vec{u}_{k+1} = \vec{u}_k + hf(t_{k+1}, \vec{u}_{k+1})$$

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

1. Nécessité de résoudre un système linéaire à chaque fois
2. Ordre 1
3. Problèmes de conservation d'énergie
4. Inconditionnellement stable

### 8.2.2 Méthode de Crank-Nicolson

$$u_{k+1} = u_k + \int_{t_k}^{t_{k+1}} f(t, u(t)) dt$$

On peut utiliser la méthode des trapèzes pour approximer l'intégrale

$$\approx \frac{h}{2} (f(t_k, u_k) + f(t_{k+1}, u_{k+1}))$$

$$\vec{u}_{k+1} = \vec{u}_k + \frac{h}{2} (\vec{f}(t_k, \vec{u}_k) + \vec{f}(t_{k+1}, \vec{u}_{k+1}))$$

On peut dire que Crank-Nicolson est la moyenne entre la méthode d'Euler explicite et implicite.

$$\begin{array}{c|cc} 0 & & \\ \hline 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}$$

1. On doit résoudre un système linéaire à chaque fois
2. Ordre 2
3. Inconditionnellement stable

### 8.2.3 Méthode d'Euler symplectique

Utilisé pour la mécanique

$$\begin{pmatrix} v_{k+1} \\ x_{k+1} \end{pmatrix} = \begin{pmatrix} v_k + hF(t_k, x_k, v_k) \\ x_k + hv_{k+1} \end{pmatrix}$$

## 8.3 Réduction d'ordre

$$y(t) = y_1 \quad y'(t) = y_2 \quad \dots \quad y^{(n+1)} = y_n$$

$$\vec{u} = \frac{d}{dt} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad \vec{u}_0 = \begin{pmatrix} y_1(0) \\ y_2(0) \\ \vdots \\ y_n(0) \end{pmatrix}$$

## 8.4 Tableau de Butcher

$$\begin{array}{c|ccc} c_1 & & & \\ c_2 & a_{21} & & \\ \vdots & \vdots & \ddots & \\ c_n & a_{n1} & \dots & a_{n, n-1} \\ \hline & w_1 & \dots & w_{n-1} & w_n \end{array}$$

$$\updownarrow$$

$$\vec{u}_{k+1} = \vec{u}_k + h(w_1 \vec{s}_1 + \dots + w_n \vec{s}_n)$$

$$\vec{s}_1 = f(t_k + c_1 h, \vec{u}_k)$$

$$\vec{s}_2 = f(t_k + c_2 h, \vec{u}_k + h(a_{21} \vec{s}_1))$$

$$\vec{s}_n = f(t_k + c_n h, \vec{u}_k + h(a_{n1} \vec{s}_1 + \dots + a_{n, n-1} h \vec{s}_{n-1}))$$

On doit avoir

$$w_1 + w_2 + \dots + w_n = 1$$

$$a_{n1} + a_{n2} + \dots + a_{nk} = c_n$$

### 8.4.1 Erreur

1. L'erreur globale (après toutes les itérations) est d'ordre égal à la taille du tableau

$$|u_{\text{approché}}(T) - u_{\text{réel}}(T)| \leq Ch^n$$

2. L'erreur locale (commise après une itération) est d'ordre égal à la taille du tableau + 1

$$|u_{\text{approché}}(T) - u_{\text{réel}}(T - h)| \leq Ch^{n+1}$$

8.5 Problèmes mal posés

Les problèmes mal posés ont une énorme perturbation de  $u(t)$  pour une petite perturbation de  $u_0$ . Typiquement des soucis si on a une équation de la forme

$$\cdots (u_0 - 1) \cdots$$

Avec  $u_0 \approx 1$

9 Autres

9.1 Triangle de Pascal

$$\begin{array}{ccccccc} & & & & 1 & & \\ & & & 1 & 1 & & \\ & & 1 & 2 & 1 & & \\ & 1 & 3 & 3 & 1 & & \\ 1 & 4 & 6 & 4 & 1 & & \\ 1 & 5 & 10 & 10 & 5 & 1 & \end{array}$$

$$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$(a + b)^n = \sum_{k=0}^n \binom{n}{k} x^k y^{n-k}$$

$$\binom{n}{k} = C_k^n = \frac{n!}{k!(n-k)!}$$

9.2 Matrices

9.2.1 Inverses

$$\begin{pmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{a} & -\frac{b}{ad} & \frac{be-cd}{adf} \\ 0 & \frac{1}{d} & -\frac{e}{fd} \\ 0 & 0 & \frac{1}{f} \end{pmatrix}$$

Même principe si on renverse

$$(M^T)^{-1} = (M^{-1})^T$$

$$\begin{pmatrix} a & 0 & 0 \\ b & d & 0 \\ c & e & f \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{a} & 0 & 0 \\ -\frac{b}{ad} & \frac{1}{d} & 0 \\ \frac{be-cd}{adf} & -\frac{e}{fd} & \frac{1}{f} \end{pmatrix}$$

Pour une matrice  $2 \times 2$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$