# Cours d'analyse numérique de licence de mathématiques

Roland Masson

16 novembre 2011

# Analyse numérique: objectifs

- Analyse numérique: conçoit et analyse mathématiquement les algorithmes à la base des simulations numériques de la physique

- Objectifs du cours
  - Introduction à quelques algorithmes de bases en calcul scientifique
  - Fondements mathématiques (complexité, stabilité, convergence, consistance, ...)
  - Exemples d'applications et mise en oeuvre informatique sous scilab (TDs et TPs)

# Plan du cours

- Résolution des systèmes linéaires $Ax = b$, $A \in \mathcal{M}_n$, $b, x \in \mathbb{R}^n$
  - Méthodes directes: méthode d'élimination de Gauss, factorisation LU, factorisation de Choleski
  - Méthodes itératives: méthodes de Richardon, de Jacobi, de Gauss Seidel

- Résolution des systèmes non linéaires $f(x) = 0$, $f : \mathbb{R}^n \to \mathbb{R}^n$
  - Méthodes de Newton

- Algorithmes d'optimisation: $x = \text{argmin}_{y \in \mathbb{R}^n} f(y)$, $f : \mathbb{R}^n \to \mathbb{R}$
  - Méthodes de descente selon le gradient

- Résolution des équations différentielles ordinaires (EDO): $y' = f(y, t)$, $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$
  - Schémas d'Euler explicite et implicite

# Références

Cours d'analyse numérique de Raphaèle Herbin (Université de Provence): http://www.cmi.univ-mrs.fr/∼herbin/anamat.html

Livre de Quateroni et al: "Méthodes numériques, algorithmes, analyse et applications", Springer, 2007.

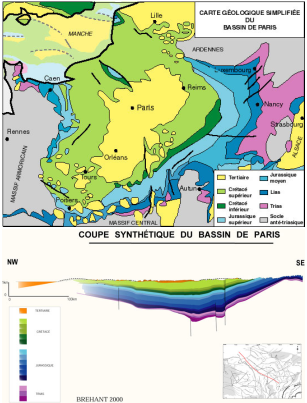
# Domaines d’applications du calcul scientifique

- Énergie
  - Nucléaire
  - Pétrole
  - Fusion nucléaire
  - Eolien, hydrolien, solaire, ...
- Transport
  - Aéronautique
  - Spatial
  - Automobile
- Environnement
  - Météorologie
  - Hydrologie
  - Géophysique
  - Climatologie
- Finance, Biologie, Santé, Télécommunications, Chimie, matériaux, ...

# Exemple de la simulation des réservoirs pétroliers

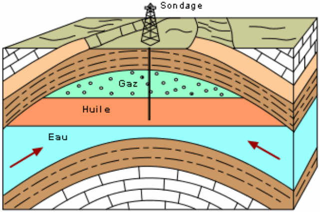
- Pétrole = huile de pierre

- Bassin de paris

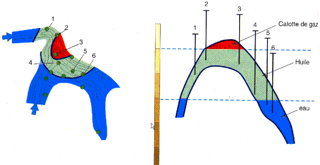


# Exemple de la simulation des réservoirs pétroliers

- Réservoir: piège géologique rempli d'hydrocarbures

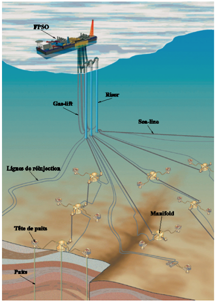


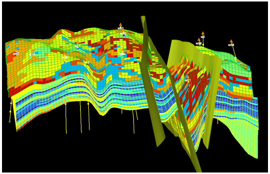

Piège structural: anticlinal




Gisement vu du dessus
et implantation des puits

Coupe transversale



# Exemple de la simulation des réservoirs pétroliers

- Enjeux de la simulation
  - Prédiction de la production
  - Optimisation de la production (ou du rendement économique)
  - Intégration des données
  - Evaluation des incertitudes sur la production

# Débouchés

- Compétences
  - Analyse numérique
  - Modélisation
  - Informatique

- Métiers
  - Développements de codes de calculs scientifiques
  - Etudes en modélisation numérique
  - Ingénieur de recherches en calcul scientifique
  - Chercheur académique en mathématiques appliquées

- Employeurs
  - SSII en calcul scientifique
  - EPIC: CEA, ONERA, IFPEN, BRGM, IFREMER, INRA, CERFACS, ...
  - Industrie: EDF, EADS, Dassault, Michelin, Areva, Total, CGGVeritas, Thales, Safran, Veolia, Rhodia, ...
  - Académique: Universités, CNRS, INRIA, Ecoles d'ingénieurs, ...

# Calendrier du cours

- Cours en amphi biologie le mercredi de 17h à 18h30: semaines 1,2,3,5,6,7,8,10,11,12,13,14,15

- TD et TP en M31/PV212-213 le jeudi de 15h à 18h15: semaines 2,3,5,7,8,10,11,12,13,14,15

- Un seul groupe en TD
  - salle M31
- A priori deux groupes en TP
  - salles PV212 et PV213
  - deuxième groupe avec AUDRIC DROGOUL

# Evaluation

- Un examen partiel semaine 11 en cours: note $P$
- Contrôle continu: note $C = (C1 + C2)/2$
  - une interrogation écrite en TD semaine 7: note $C1$
  - une interrogation écrite en TP semaine 13: note $C2$
- Un examen final: note $F$

Note finale: $0.4F + 0.3P + 0.3C$

# Espaces vectoriels

- Définition d'un e.v. sur $K = \mathbb{R}$ ou $\mathbb{C}$: ensemble E muni d'une loi de composition interne notée $+$ et d'une loi d'action de $K$ sur $E$ notée . tels que:
  - $(E, +)$ est un groupe commutatif
  - $1.\mathbf{x} = \mathbf{x}$, $(\alpha\beta).\mathbf{x} = \alpha.(\beta.\mathbf{x})$ (associativité)
  - $(\alpha + \beta).\mathbf{x} = \alpha.\mathbf{x} + \beta.\mathbf{x}$, $\alpha.(\mathbf{x} + \mathbf{y}) = \alpha.\mathbf{x} + \alpha.\mathbf{y}$ (distributivité)
- Exemple: $\mathbb{R}^n$ e.v. sur $\mathbb{R}$ ($\mathbb{C}^n$ e.v. sur $\mathbb{C}$):
  - $\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{x} + \mathbf{y} = \begin{pmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{pmatrix}, \quad \lambda.\mathbf{x} = \begin{pmatrix} \lambda x_1 \\ \vdots \\ \lambda x_n \end{pmatrix}$

# Familles libres, génératrices, base, dimension

- Famille libre de $m$ vecteurs $\mathbf{v}_1, \cdots, \mathbf{v}_m$ de $E$:
  - $\sum_{i=1}^{m} \lambda_i \mathbf{v}_i = 0 \Rightarrow \lambda_i = 0 \; \forall \, i = 1, \cdots, m$
- Famille génératrice de $m$ vecteurs $\mathbf{v}_1, \cdots, \mathbf{v}_m$ de $E$:
  - $E = \text{Vect}\{\mathbf{v}_1, \cdots, \mathbf{v}_m\}$
- Base: famille libre et génératrice
  - Dimension (supposée finie): toutes les bases ont même dimension appelée dimension de l'espace vectoriel $E$ notée $n$
  - Une famille libre de $n$ vecteurs est génératrice, c'est une base
  - Une famille génératrice de $n$ vecteurs est libre, c'est une base

# Espaces vectoriels normés

- Définition: e.v. muni d'une norme, ie une application de $E \to \mathbb{R}^+$, notée $\mathbf{x} \to \|\mathbf{x}\|$ satisfaisant les propriétés suivantes
  - $\|\mathbf{x}\| = 0 \Rightarrow \mathbf{x} = 0$
  - $\|\lambda.\mathbf{x}\| = |\lambda| \|\mathbf{x}\|$
  - $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

- Une norme définit sur $E$ une topologie d'espace métrique avec $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$
  - Limite de suite de vecteurs: $\lim_{k \to +\infty} \mathbf{v}_k = \mathbf{v} \Leftrightarrow \lim_{k \to +\infty} \|\mathbf{v}_k - \mathbf{v}\| = 0$

- Exemples de normes sur $\mathbb{R}^n$
  - $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$, $\|\mathbf{x}\|_2 = \left(\sum_{i=1}^n |x_i|^2\right)^{1/2}$, $\|\mathbf{x}\|_\infty = \max_{i=1,\cdots,n} |x_i|$.
- En dimension finie toutes les normes sont équivalentes ie il existe $c, C > 0$ telles que $c\|\mathbf{x}\| \leq \|\mathbf{x}\|_\star \leq C\|\mathbf{x}\|$ (attention $c$ et $C$ dépendent de $n$).

# Espaces vectoriels euclidiens

- e.v. muni d'un produit scalaire ie une forme bilinéaire symétrique définie positive notée $\langle ., . \rangle$
- Sur $\mathbb{R}^n$ le produit scalaire canonique est $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i$
- $\|\mathbf{x}\| = \langle \mathbf{x}, \mathbf{x} \rangle^{1/2}$ est une norme appelée norme euclidienne

# Applications linéaires

- $f : E \to F$, $f(\lambda.\mathbf{x}) = \lambda.f(\mathbf{x})$, $f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$
- $\mathcal{L}(E,F)$ espace vectoriel des applications linéaires de $E$ dans $F$
- $\mathcal{L}(E)$ espace vectoriel des applications linéaires de $E$ dans $E$ ou endomorphismes de $E$
- $\Big(\mathcal{L}(E), +, ., \circ\Big)$ anneau unitaire non commutatif munie de la loi de composition des applications $f \circ g(\mathbf{x}) = f(g(\mathbf{x}))$
- Noyau de $f$, $\mathrm{Ker}(f) = \{\mathbf{x} \in E$ tels que $f(\mathbf{x}) = 0\}$ (sous e.v. de $E$)
- Image de $f$, $\mathrm{Im}(f) = \{f(\mathbf{x}), \mathbf{x} \in E\}$ (sous e.v. de $F$)
- Endomorphismes de $E$ inversibles:
  - Application bijective ssi il existe $f^{-1} \in \mathcal{L}(E)$ telle que $f \circ f^{-1} = f^{-1} \circ f = Id$
  - $f$ bijective $\Leftrightarrow$ $f$ injective: $\mathrm{Ker}(f) = \{0\}$
  - $f$ bijective $\Leftrightarrow$ $f$ surjective: $\mathrm{Im}(f) = E$

# Matrice d'une application linéaire

- Bases $\left(\mathbf{e}_j, j = 1, \cdots, n\right)$ de $E$ et $\left(\mathbf{f}_i, i = 1, \cdots, m\right)$ de $F$
- $f \in \mathcal{L}(E, F)$ telle que $f(\mathbf{e}_j) = \sum_{i=1}^m A_{i,j}\mathbf{f}_i$
- $\mathbf{x} = \sum_{j=1}^n x_j\mathbf{e}_j \in E$
- $\mathbf{y} = f(\mathbf{x}) = \sum_{i=1}^m \left(\sum_{j=1}^n A_{i,j}x_j\right)\mathbf{f}_i$
- $X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n, \quad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m, \; Y = AX$
- Retenir que les $n$ colonnes $j$ de $A$ sont données par les images $f(\mathbf{e}_j)$
- Espace vectoriel des matrices de dimension $m, n$: $\mathcal{M}_{m,n}$ (à coefficients dans $K = \mathbb{R}$ ou $\mathbb{C}$)
- Matrices remarquables: diagonale, symétrique, triangulaires inférieure ou supérieure

# Exercice: produit de matrices versus composition d'applications linéaires

- Soient $E, F, G$ des e.v de dimensions resp. $n$,$m$,$p$, $f \in \mathcal{L}(E, F)$ et $g \in \mathcal{L}(F, G)$
- Des bases étant données, $f$ a pour matrice $A \in \mathcal{M}_{m,n}$ et $g$ a pour matrice $B \in \mathcal{M}_{p,m}$
- $g \circ f$ a pour matrice le produit $BA \in \mathcal{M}_{p,n}$ tel que

$$(BA)_{i,j} = \sum_{k=1}^{m} B_{i,k} A_{k,j}$$

- Produit de matrices: $\mathcal{M}_{p,m} \times \mathcal{M}_{m,n} \to \mathcal{M}_{p,n}$
  - produit matrice vecteur: $\mathcal{M}_{m,n} \times \mathcal{M}_{n,1} \to \mathcal{M}_{m,1}$
  - produit scalaire de deux vecteurs: ligne . colonne $\mathcal{M}_{1,n} \times \mathcal{M}_{n,1} \to \mathcal{M}_{1,1}$
  - produit tensoriel de deux vecteurs: colonne. ligne $\mathcal{M}_{n,1} \times \mathcal{M}_{1,n} \to \mathcal{M}_{n,n}$

# Exercice: changements de base pour les vecteurs et les matrices

- $P$: matrice de passage d'une base dans une autre $\tilde{e}_j = \sum_{k=1}^{n} P_{k,j} \mathbf{e}_k$ (colonnes de la nouvelle base dans l'ancienne)
- Changement de base pour les coordonnées des vecteurs: $X = P\tilde{X}$.
- Changement de base pour les matrices des applications linéaires: $X = P\tilde{X}$, $Y = Q\tilde{Y}$ et $\tilde{Y} = \tilde{A}\tilde{X}$, $Y = AX$ implique que

$$\tilde{A} = Q^{-1}AP.$$

# Matrices carrés inversibles

- $A \in \mathcal{M}_{n,n} = \mathcal{M}_n$ est inversible ssi l'une des propriétés suivantes est vérifiée
  - Il existe $A^{-1} \in \mathcal{M}_{n,n}$ tel que $AA^{-1} = A^{-1}A = I$
  - $A$ est injective ie $AX = 0 \Rightarrow X = 0$
  - $A$ est surjective ie $\mathrm{Im}(A) = \{AX,\, X \in \mathbb{R}^n\} = \mathbb{R}^n$

- $A, B \in \mathcal{M}_n$ inversibles
$$(AB)^{-1} = B^{-1}A^{-1}$$

# Transposition de matrices

- $A \in \mathcal{M}_{m,n}$, on définit $A^t \in \mathcal{M}_{n,m}$ par

$$(A^t)_{i,j} = A_{j,i} \text{ pour tous } i = 1, \cdots, n, \; j = 1, \cdots, m$$

- Produit scalaire canonique de deux vecteurs (colonnes) $X, Y \in \mathbb{R}^n$:

$$X^t Y = \sum_{i=1}^{n} X_i Y_i$$

- Matrice carrée $A \in \mathcal{M}_n$ est symétrique ssi

$$A^t = A$$

# Diagonalisation d'une matrice carrée symétrique $A \in \mathcal{M}_n$

- Les valeurs propres sur $\mathbb{C}$ d'une matrice réelle symétrique $A$ sont réelles et il existe une base orthonormée de vecteurs propres $F^i \in \mathbb{R}^n$, $i = 1, \cdots, n$ telle que
$$AF^i = \lambda_i F^i \text{ et } (F^i)^t F^j = \delta_{i,j} \text{ pour tous } i, j = 1, \cdots, n$$
- Si $P$ est la matrice de passage de la base canonique dans la base $F^i$, $i = 1, \ldots, n$, alors on a
$$P^{-1} = P^t$$
et
$$P^t A P = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$$

# Déterminants de $n$ vecteurs dans un e.v. $E$ de dimension $n$ pour une base donnée

- Unique forme n-linéaire alternée sur $E$ valant 1 sur la base
  - $\text{Det}\left(\mathbf{v}_1, \cdots, \mathbf{v}, \cdots, \mathbf{v}, \cdots, \mathbf{v}_n\right) = 0$ (alternée)
  - Antisymétrie:

$$\text{Det}\left(\mathbf{v}_1, \cdots, \mathbf{v}_i, \cdots, \mathbf{v}_j, \cdots, \mathbf{v}_n\right) = -\text{Det}\left(\mathbf{v}_1, \cdots, \mathbf{v}_j, \cdots, \mathbf{v}_i, \cdots, \mathbf{v}_n\right)$$

  - On a donc aussi pour toute permutation $\sigma$ de $\{1, \cdots, n\}$,

$$\text{Det}\left(\mathbf{v}_1, \cdots, \mathbf{v}_n\right) = \text{sign}(\sigma)\text{Det}\left(\mathbf{v}_{\sigma(1)}, \cdots, \mathbf{v}_{\sigma(n)}\right)$$

- Déterminant d'une matrice carrée $A$ = déterminant des vecteurs colonnes

$$\text{Det}\left(A\right) = \text{Det}\left(A_{.,1}, \cdots, A_{.,n}\right) = \sum_{\sigma \in \Sigma_n} \prod_{i=1}^{n} \text{sign}(\sigma) A_{\sigma(i),i}$$

# Propriétés du déterminant

- Les vecteurs colonnes de $A$ sont libres ssi $\mathrm{Det}(A) \neq 0$
- Donc $A$ est inversible ssi $\mathrm{Det}(A) \neq 0$
- $\mathrm{Det}(AB) = \mathrm{Det}(A)\mathrm{Det}(B) = \mathrm{Det}(BA)$
- $\mathrm{Det}(A^t) = \mathrm{Det}(A)$
- Développement par rapport aux lignes ou aux colonnes

$$\mathrm{Det}(A) = \sum_{i=1}^{n} (-1)^{i+j} \mathrm{Det}(A^{(i,j)}) = \sum_{j=1}^{n} (-1)^{i+j} \mathrm{Det}(A^{(i,j)})$$

# Normes matricielles

- Une norme matricielle sur l'e.v. $\mathcal{M}_n$ est une norme telle que

$$\|AB\| \leq \|A\|\|B\|$$

- Une norme matricielle induite par une norme $\|.\|$ sur $\mathbb{R}^n$ est la norme matricielle définie par

$$\|A\| = \sup_{X \neq 0} \frac{\|AX\|}{\|X\|}$$

- On a pour une norme matricielle induite: $\|AX\| \leq \|A\|\|X\|$ pour tout $X \in \mathbb{R}^n$

# Exercice: exemples de normes induites

- $\|A\|_\infty = \mathrm{Sup}_{X\neq 0}\frac{\|AX\|_\infty}{\|X\|_\infty} = \max_{i=1,\cdots,n}\sum_{j=1}^{n}|A_{i,j}|$
- $\|A\|_1 = \mathrm{Sup}_{X\neq 0}\frac{\|AX\|_1}{\|X\|_1} = \max_{j=1,\cdots,n}\sum_{i=1}^{n}|A_{i,j}|$
- $\|A\|_2 = \mathrm{Sup}_{X\neq 0}\frac{\|AX\|_2}{\|X\|_2} = \rho({}^tAA)^{1/2}$

# Convergence de la suite $A^k$ pour $A \in \mathcal{M}_n$

- Rayon spectral $\rho(A)$, $A \in \mathcal{M}_n$ est le module de la valeur propre maximale de $A$ dans $\mathbb{C}$.
- On admettra le lemme suivant:
  - $\rho(A) < 1$ ssi $\lim_{k \to +\infty} A^k = 0$ quel que soit la norme sur $\mathcal{M}_n$
  - $\rho(A) = \lim_{k \to +\infty} \|A^k\|^{1/k}$ quel que soit la norme sur $\mathcal{M}_n$
  - $\rho(A) \leq \|A\|$ quel que soit la norme matricielle sur $\mathcal{M}_n$

# Matrices de la forme $I + A$ ou $I - A$

- Si $\rho(A) < 1$ alors les matrices $I + A$ et $I - A$ sont inversibles
- La série de terme général $A^k$ converge (vers $(I - A)^{-1}$ ssi $\rho(A) < 1$
  - Preuve: $\sum_{k=0}^{N} A^k(I - A) = I - A^{N+1}$ et utiliser le lemme précédent
- Si $\|A\| < 1$ pour une norme matricielle, alors $I - A$ est inversible et on a $\|(I - A)^{-1}\| \leq \frac{1}{1-\|A\|}$ (idem pour $I + A$)

# Méthode d'élimination de Gauss: exemple

$$AX = \begin{pmatrix} 1 & 3 & 2 \\ -1 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix} X = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} = b \quad \det(A) = 5$$

Descente: élimination sur la première colonne ($x_1$)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 & 2 \\ -1 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix} X = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} = b$$

$$\Rightarrow \begin{pmatrix} 1 & 3 & 2 \\ 0 & 5 & 3 \\ 0 & -5 & -2 \end{pmatrix} X = \begin{pmatrix} 1 \\ 3 \\ -1 \end{pmatrix}$$

Descente: élimination sur la deuxième colonne ($x_2$)

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 & 2 \\ 0 & 5 & 3 \\ 0 & -5 & -2 \end{pmatrix} X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ -1 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} 1 & 3 & 2 \\ 0 & 5 & 3 \\ 0 & 0 & 1 \end{pmatrix} X = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} \quad \text{d'où par remontée } X = \begin{pmatrix} -6/5 \\ -3/5 \\ 2 \end{pmatrix}$$

# factorisation de Gauss: exemple

D'où $UX = b'$ avec

$$b = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}^{-1} b' = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} b'$$

$$b = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix} b'$$

D'où la factorisation de Gauss

$$A = LU$$

avec

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix} \text{ et } U = \begin{pmatrix} 1 & 3 & 2 \\ 0 & 5 & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

# Généralisation à $A \in \mathcal{M}_n$ inversible: $AX = b$

$$A^{(1)} = A, \;\; b^{(1)} = b$$

$$A^{(2)} = L^{(1)} A^{(1)} = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ -\frac{a_{2,1}^{(1)}}{a_{1,1}^{(1)}} & 1 & 0 & \dots & \dots & 0 \\ \vdots & & & & & \\ -\frac{a_{k,1}^{(1)}}{a_{1,1}^{(1)}} & 0 & \dots & 1 & \dots & 0 \\ \vdots & & & & & \\ -\frac{a_{n,1}^{(1)}}{a_{1,1}^{(1)}} & 0 & \dots & \dots & \dots & 1 \end{pmatrix} \begin{pmatrix} a_{1,1}^{(1)} & \dots & \dots & \dots & \dots & a_{1,n}^{(1)} \\ a_{2,1}^{(1)} & \dots & \dots & \dots & \dots & a_{2,n}^{(1)} \\ \vdots & & & & & \vdots \\ a_{k,1}^{(1)} & \dots & \dots & \dots & \dots & a_{k,n}^{(1)} \\ \vdots & & & & & \vdots \\ a_{n,1}^{(1)} & \dots & \dots & \dots & \dots & a_{n,n}^{(1)} \end{pmatrix} \text{ et } b^{(2)} = L^{(1)} b^{(1)}$$

$$A^{(2)} = \begin{pmatrix} a_{1,1}^{(1)} & \dots & \dots & \dots & \dots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & \dots & \dots & \dots & a_{2,n}^{(2)} \\ \vdots & \vdots & & & & \vdots \\ 0 & a_{k,2}^{(2)} & \dots & \dots & \dots & a_{k,n}^{(2)} \\ \vdots & \vdots & & & & \vdots \\ 0 & a_{n,2}^{(2)} & \dots & \dots & \dots & a_{n,n}^{(2)} \end{pmatrix} \text{ avec } a_{i,j}^{(2)} = a_{i,j}^{(1)} - \frac{a_{i,1}^{(1)} a_{1,j}^{(1)}}{a_{1,1}^{(1)}}, \;\; i,j = 2, \cdots, n$$

# Généralisation à $A \in \mathcal{M}_n$ inversible: $AX = b$

$$A^{(k+1)} = L^{(k)} A^{(k)} = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & \dots & \vdots \\ \vdots & 0 & 1 & \dots & \dots & 0 \\ 0 & 0 & -\frac{a^{(k)}_{k+1,k}}{a^{(k)}_{k,k}} & 1 & 0 & \vdots \\ 0 & 0 & \vdots & 0 & 1 & 0 \\ 0 & 0 & -\frac{a^{(k)}_{n,k}}{a^{(k)}_{k,k}} & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} a^{(1)}_{1,1} & \dots & \dots & \dots & \dots & a^{(1)}_{1,n} \\ 0 & a^{(2)}_{2,2} & \dots & \dots & \dots & a^{(2)}_{2,n} \\ \vdots & 0 & a^{(k)}_{k,k} & \dots & \dots & a^{(k)}_{k,n} \\ 0 & \dots & a^{(k)}_{k+1,k} & \dots & \dots & a^{(k)}_{k+1,n} \\ \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & a^{(k)}_{n,k} & \dots & \dots & a^{(k)}_{n,n} \end{pmatrix}$$

et $b^{(k+1)} = L^{(k)} b^{(k)}$

$$A^{(k+1)} = \begin{pmatrix} a^{(1)}_{1,1} & \dots & \dots & \dots & \dots & a^{(1)}_{1,n} \\ 0 & a^{(2)}_{2,2} & \dots & \dots & \dots & a^{(2)}_{2,n} \\ \vdots & 0 & a^{(k)}_{k,k} & a^{(k)}_{k,k+1} & \dots & a^{(k)}_{k,n} \\ 0 & \dots & 0 & a^{(k+1)}_{k+1,k+1} & \dots & a^{(k+1)}_{k+1,n} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & a^{(k+1)}_{n,k+1} & \dots & a^{(k+1)}_{n,n} \end{pmatrix} \text{ avec } a^{(k+1)}_{i,j} = a^{(k)}_{i,j} - \frac{a^{(k)}_{i,k} a^{(k)}_{k,j}}{a^{(k)}_{k,k}},\ i, j = k+1, \cdots, n$$

# Généralisation à $A \in \mathcal{M}_n$ inversible: $AX = b$

$$A = LU$$

avec

$$U = A^{(n)}$$

$$U = \begin{pmatrix} a_{1,1}^{(1)} & \dots & \dots & \dots & \dots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & \dots & \dots & \dots & a_{2,n}^{(2)} \\ \vdots & 0 & a_{k,k}^{(k)} & a_{k,k+1}^{(k)} & \dots & a_{k,n}^{(k)} \\ 0 & \dots & 0 & a_{k+1,k+1}^{(k+1)} & \dots & a_{k+1,n}^{(k+1)} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & a_{n,n}^{(n)} \end{pmatrix}$$

# Généralisation à $A \in \mathcal{M}_n$ inversible: $AX = b$

$$A = LU$$

avec

$$U = A^{(n)}$$

$$L = \left(L^{(n-1)} \dots L^{(k)} \dots L^{(1)}\right)^{-1} = (L^{(1)})^{-1} \dots (L^{(k)})^{-1} \dots (L^{(n-1)})^{-1}$$

$$(L^{(k)})^{-1} = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & \dots & \vdots \\ \vdots & 0 & 1 & \dots & \dots & 0 \\ 0 & 0 & \frac{a^{(k)}_{k+1,k}}{a^{(k)}_{k,k}} & 1 & 0 & \vdots \\ 0 & 0 & \vdots & 0 & 1 & 0 \\ 0 & 0 & \frac{a^{(k)}_{n,k}}{a^{(k)}_{k,k}} & \dots & 0 & 1 \end{pmatrix}$$

# Généralisation à $A \in \mathcal{M}_n$ inversible: $AX = b$

$$(L^{(k)})^{-1}(L^{(k+1)})^{-1} = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & \dots & \vdots \\ \vdots & 0 & 1 & \dots & \dots & 0 \\ 0 & 0 & \frac{a^{(k)}_{k+1,k}}{a^{(k)}_{k,k}} & 1 & 0 & \vdots \\ 0 & 0 & \vdots & \frac{a^{(k+1)}_{k+2,k+1}}{a^{(k+1)}_{k+1,k+1}} & 1 & 0 \\ 0 & 0 & \frac{a^{(k)}_{n,k}}{a^{(k)}_{k,k}} & \frac{a^{(k+1)}_{n,k+1}}{a^{(k+1)}_{k+1,k+1}} & 0 & 1 \end{pmatrix}$$

# Généralisation à $A \in \mathcal{M}_n$ inversible: $AX = b$

$$
L = \begin{pmatrix}
1 & 0 & \cdots & \cdots & \cdots & 0 \\
\frac{a^{(1)}_{2,1}}{a^{(1)}_{1,1}} & 1 & 0 & \cdots & \cdots & \vdots \\
\vdots & \ddots & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \frac{a^{(k)}_{k+1,k}}{a^{(k)}_{k,k}} & 1 & 0 & \vdots \\
\vdots & \vdots & \vdots & \frac{a^{(k+1)}_{k+2,k+1}}{a^{(k+1)}_{k+1,k+1}} & 1 & 0 \\
\frac{a^{(1)}_{n,1}}{a^{(1)}_{1,1}} & \vdots & \frac{a^{(k)}_{n,k}}{a^{(k)}_{k,k}} & \frac{a^{(k+1)}_{n,k+1}}{a^{(k+1)}_{k+1,k+1}} & \frac{a^{(n-1)}_{n,k}}{a^{(n-1)}_{n-1,n-1}} & 1
\end{pmatrix}
$$

# Existence et unicité de la factorisation $A = LU$

- Soit $A \in \mathcal{M}_n$, on suppose que les sous matrices diagonales de dimension $k$

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,k} \\ \vdots & \cdots & \vdots \\ a_{k,1} & \cdots & a_{k,k} \end{pmatrix}$$

  sont inversibles pour tous $k = 1, \cdots n$.
- Alors la factorisation $A = LU$ avec $L_{i,i} = 1$, $i = 1, \cdots, n$ existe et est unique.

Preuve:

- Existence établie précédemment car on montre que le pivot $a_{k,k}^k \neq 0$
- Unicité: $A = L_1 U_1 = L_2 U_2$ implique $L_2^{-1} L_1 = U_2 U_1^{-1} = I$

# Algorithme: factorisation $LU$ de $A \in \mathcal{M}_n$ inversible

Initialisation: $U = A$, $L = I$

- For $k = 1, \cdots, n-1$ (boucle sur les pivots)
  - For $i, j = k+1, \cdots, n$
    - $U_{i,j} \leftarrow U_{i,j} - \dfrac{U_{i,k} U_{k,j}}{U_{k,k}}$ (on suppose le pivot $U_{k,k}$ non nul)
  - End For
  - For $i = k+1, \cdots, n$
    - $L_{i,k} = \dfrac{U_{i,k}}{U_{k,k}}$
  - End For
- End For
- $U \leftarrow triu(U)$

Remarque 1: on a supposé que $U_{k,k} \neq 0$
Remarque 2: on peut tout stocker dans $A$ au cours de l'algorithme

# Résolution de $LUX = b$

Descente: $LY = b$

- For $i = 1, \cdots, n$
  - $Y_i = b_i - \sum_{j=1}^{i-1} L_{i,j} Y_j$
- End For

Remontée: $UX = Y$

- For $i = n, \cdots, 1; -1$
  - $X_i = \dfrac{Y_i - \sum_{j=i+1}^{n} U_{i,j} X_j}{U_{i,i}}$
- End For

# Complexité de l'algorithme

On compte le nombre d'additions et de multiplications et de divisions (opérations flottantes)

- Factorisation: $2/3n^3 + \mathcal{O}(n^2)$
- Descente remontée: $2n^2 + \mathcal{O}(n)$

# Conservation de la largeur de bande

$$q = \max_{i,j=1,\cdots,n} \{|j-i| \text{ tel que } A_{i,j} \neq 0\}$$

La factorisation $A = LU$ précédente (sans pivotage) conserve la largeur de bande $q$ pour $U$ et $L$
Preuve: propriété vérifiée pour toutes les matrices $A^{(k)}$ à chaque étape $k = 1, \cdots, n$

Complexité:

- Factorisation: $2nq^2 + \mathcal{O}(nq)$
- Descente remontée: $2nq + \mathcal{O}(n)$

# Algorithme: factorisation $LU$ pour une matrice bande de largeur de bande $q$

Initialisation: $U = A$, $L = I$

- For $k = 1, \cdots, n-1$ (boucle sur les pivots)
  - For $i, j = k+1, \cdots, \max(k+q, n)$
    - $U_{i,j} \leftarrow U_{i,j} - \dfrac{U_{i,k}U_{k,j}}{U_{k,k}}$ (on suppose le pivot $U_{k,k}$ non nul)
  - End For
  - For $i = k+1, \cdots, \max(k+q, n)$
    - $L_{i,k} = \dfrac{U_{i,k}}{U_{k,k}}$
  - End For
- End For
- $U \leftarrow triu(U)$

# Résolution de $LUX = b$ pour une matrice bande de largeur de bande $q$

Descente: $LY = b$

- For $i = 1, \cdots, n$
  - $Y_i = b_i - \sum_{j=\max(i-q,1)}^{i-1} L_{i,j} Y_j$
- End For

Remontée: $UX = Y$

- For $i = n, \cdots, 1; -1$
  - $X_i = \dfrac{Y_i - \sum_{j=i+1}^{\min(i+q,n)} U_{i,j} X_j}{U_{i,i}}$
- End For

# Matrices de permutation

Bijection de $\{1, \cdots, n\}$ dans $\{1, \cdots, n\}$
Première représentation:

$$P = \left(j_1, \cdots, j_n\right)$$

avec $j_i \in \{1, \cdots, n\}$ et $j_i \neq j_l$ pour $i \neq l$.

Action sur les vecteurs $b \in \mathbb{R}^n$: $(Pb)_i = b_{P(i)}$ pour $i = 1, \cdots, n$

D'où la représentation matricielle: $P_{i,j} = 1$ si $j = P(i)$, sinon 0. On a sur les matrices $A \in \mathcal{M}_n$:

$$(PA)_{i,j} = A_{P(i),j}.$$

Exemple

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, Pb = \begin{pmatrix} b_2 \\ b_1 \\ b_4 \\ b_3 \end{pmatrix}$$

# Transpositions

$$\tau = (i_1, i_2)$$

est la permutation telle que

$$\tau(i_1) = i_2, \tau(i_2) = i_1, \tau(i) = i \ \forall \, i \neq i_1, i_2.$$

On a

$$\tau^2 = I,$$

donc

$$\tau^{-1} = \tau.$$

# factorisation avec pivotage $PA = LU$

On a $A^{(1)} = A$ et pour $k = 1, \cdots, n-1$

$$A^{(k+1)} = L^{(k)} \tau^{(k)} A^{(k)}$$

avec $\tau^{(k)} = (k, i_k)$ pour $i_k \geq k$ tel que $|A^{(k)}_{i_k,k}| = max_{i=k,\cdots,n} |A^{(k)}_{i,k}|$.
D'où par récurrence

$$A^{(n)}X = UX = L^{(n-1)}\tau^{(n-1)}L^{(n-2)}\tau^{(n-2)}\cdots\tau^{(k+1)}L^{(k)}\tau^{(k)}\cdots\tau^{(2)}L^{(1)}\tau^{(1)}b$$

$$UX = L^{(n-1)}\left(\tau^{(n-1)}L^{(n-2)}\tau^{(n-1)}\right)\cdots\left(\tau^{(n-1)}\cdots\tau^{(2)}L^{(1)}\tau^{(2)}\cdots\tau^{(n-1)}\right)\left(\tau^{(n-1)}\cdots\tau^{(1)}\right)b$$

d'où

$$L = \left(\tau^{(n-1)}\cdots\tau^{(2)}(L^{(1)})^{-1}\tau^{(2)}\cdots\tau^{(n-1)}\right)\cdots\left(\tau^{(n-1)}(L^{(n-2)})^{-1}\tau^{(n-1)}\right)(L^{(n-1)})^{-1}$$

et

$$P = \left(\tau^{(n-1)}\cdots\tau^{(2)}\tau^{(1)}\right)$$

# Algorithme avec pivotage partiel $PA = LU$

Initialisation: $U = A$, $L = I$, $P = (1, \cdots, n)$

- For $k = 1, \cdots, n-1$ (boucle sur les pivots)
  - $i_k = \mathrm{argmax}_{i=k,\cdots,n}|U_{i,k}|$ (choix du pivot) transposition: $\tau = (k, i_k)$, $i_k \geq k$
  - $U \leftarrow \tau U$ permutation des lignes de $U$
  - $L \leftarrow \tau L \tau$ permutation des lignes de $L$ hors diagonale
  - $P \leftarrow \tau P$ mise à jour de la permutation $P$ pour $b$
  - For $i, j = k+1, \cdots, n$
    - $U_{i,j} \leftarrow U_{i,j} - \dfrac{U_{i,k}U_{k,j}}{U_{k,k}}$
  - End For
  - For $i = k+1, \cdots, n$
    - $L_{i,k} = \dfrac{U_{i,k}}{U_{k,k}}$
  - End For
- End For
- $U \leftarrow triu(U)$

# Algorithme avec pivotage partiel $PA = LU$ et avec stockage de $L$ (sans la diagonale) et de $U$ dans la matrice $A$

Initialisation: $P = (1, \cdots, n)$

- For $k = 1, \cdots, n-1$ (boucle sur les pivots)
  - $i_k = \text{argmax}_{i=k,\cdots,n} |A_{i,k}|$ (choix du pivot) transposition: $\tau = (k, i_k)$, $i_k \geq k$
  - $A \leftarrow \tau A$ permutation des lignes $k$ et $i_k$
  - $P \leftarrow \tau P$ mise à jour de la permutation
  - For $i = k+1, \cdots, n$
    - $A_{i,k} \leftarrow \dfrac{A_{i,k}}{A_{k,k}}$
  - End For
  - For $i, j = k+1, \cdots, n$
    - $A_{i,j} \leftarrow A_{i,j} - A_{i,k} A_{k,j}$
  - End For
- End For

$L$ est la partie triangulaire inférieure stricte de $A$ plus la diagonale unité.
$U$ est la partie triangulaire supérieure de $A$ (avec la diagonale).

# Résolution de $PAX = LUX = Pb$ pour la factorisation avec pivotage $P$

Descente: $LY = Pb$

- For $i = 1, \cdots, n$
  - $Y_i = b_{P(i)} - \sum_{j=1}^{i-1} L_{i,j} Y_j$
- End For

Remontée: $UX = Y$

- For $i = n, \cdots, 1; -1$
  - $X_i = \dfrac{Y_i - \sum_{j=i+1}^{n} U_{i,j} X_j}{U_{i,i}}$
- End For

# Conditionnement

Soit $\|\|$ la norme induite dans $\mathcal{M}_n$ par une norme $\|\|$ sur $\mathbb{R}^n$

$$\text{Conditionnement de } A \text{ inversible} \ : \ \text{Cond}(A) = \|A\|\|A^{-1}\|$$

$$\left\{ \begin{array}{l} \text{Cond}(A) \geq 1 \\ \text{Cond}(\alpha A) = \text{Cond}(A) \\ \text{Cond}(AB) \leq \text{Cond}(A)\text{Cond}(B) \end{array} \right.$$

Soit $A$ inversible et $\sigma_1$, $\sigma_n$ les vp min et max de $A^t A$, on a pour la norme $\|\|_2$

$$\text{Cond}_2(A) = \left(\frac{\sigma_n}{\sigma_1}\right)^{1/2}$$

On en déduit que $\text{Cond}_2(A) = 1$ ssi $A = \alpha Q$ où $Q$ matrice orthogonale

Pour $A$ SDP de vp min et max $\lambda_1$ et $\lambda_n$, on a pour la norme $\|\|_2$

$$\text{Cond}_2(A) = \frac{\lambda_n}{\lambda_1}$$

# Erreur d'arrondi

Soit $A$ une matrice inversible, on cherche à estimer l'influence sur la solution d'une erreur d'arrondi sur le second membre $b$

$$\left\{ \begin{array}{rcl} Ax & = & b \\ A(x+\delta x) & = & b+\delta b \end{array} \right.$$

implique

$$\frac{\|\delta x\|}{\|x\|} \leq \|A\| \|A^{-1}\| \frac{\|\delta b\|}{\|b\|}$$

# Méthodes itératives: motivations

- Matrice creuse: $\mathcal{O}(n)$ termes non nuls
- Méthodes itératives: calcul d'une suite $x^n$ faisant intervenir que des produits matrice - vecteur
- Pour une matrice creuse, une itération coûte $\mathcal{O}(n)$ opérations flottantes
- Le problème à résoudre est la maîtrise de la convergence de la suite $x^n$ vers $x$ et du nombre d'itérations
- Coût pour une matrice creuse et une convergence en *nit* itérations

$$\mathcal{O}(n.nit)$$

# matrices creuses: exemple du Laplacien 2D sur maillage Cartésien

Maillage cartésien uniforme $(n+1) \times (n+1)$ du carré $\Omega = (0,1) \times (0,1)$ de pas $\Delta x = \frac{1}{n+1}$ Laplacien avec conditions limites homogènes:

$$\left\{ \begin{array}{r} -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f \text{ sur } \Omega \\ u(x) = 0 \text{ sur } \partial\Omega \end{array} \right.$$

Discrétisation:

$$\left\{ \begin{array}{l} \frac{1}{(\Delta x)^2}(4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} = f_{i,j} \text{ pour } i,j = 1, \cdots, n \\ u_{i,j} = 0 \text{ pour } i = 0, n+1, j = 0, \cdots, n+1 \text{ et } j = 0, n+1, i = 0, \cdots, n+1 \end{array} \right.$$

Système linéaire: numérotation des inconnues $k = i + (j-1)n$ de 1 à $N = n^2$

$$AU = F \text{ avec } A \text{ matrice pentadiagonale de largeur de bande } q = n$$

Coût d'une méthode directe $LU$: $2Nn^2 = 2N^2$
Coût d'une méthode itérative convergente en *nit* itérations: $10N.nit$

# Méthode de Richardson

Soit $A \in \mathcal{M}_n$ inversible et $b \in \mathbb{R}^n$. Soit $x \in \mathbb{R}^n$ solution de

$$Ax = b.$$

Méthode de Richardson: soit $\alpha \in \mathbb{R}$, on construit une suite de solution $x^k$ de la forme

$$x^{k+1} = x^k + \alpha(b - Ax^k).$$

On a donc

$$(x^{k+1} - x) = (I - \alpha A)(x^k - x),$$

$$(x^{k+1} - x) = (I - \alpha A)^k (x^1 - x),$$

$$B = (I - \alpha A)$$

- Convergence: ssi $\rho(B) < 1$
- Taux de convergence: $\|B^k\|^{1/k} \to \rho(I - \alpha A)$

# Méthode de Richardon pour les matrices SDP

Soit $A \in \mathcal{M}_n$ SDP (Symétrique Définie Positive) et $\lambda_i > 0, i = 1, \cdots, n$ ses valeurs propres par ordre croissant

$$\rho(I - \alpha A) = max\Big(|1 - \alpha\lambda_1|, |1 - \alpha\lambda_n|\Big)$$

$$\alpha_{opt} = argmin_{\alpha\in\mathbb{R}} max\Big(|1 - \alpha\lambda_1|, |1 - \alpha\lambda_n|\Big)$$

$$\alpha_{opt} = \frac{2}{\lambda_1 + \lambda_n}, \quad \rho(I - \alpha_{opt}A) = \frac{\lambda_n - \lambda_1}{\lambda_1 + \lambda_n} = \frac{\kappa - 1}{\kappa + 1} < 1$$

Problème: on ne connait pas les valeurs propres de $A$
Voir Exercice: Méthode de Richardson à pas variable

# Méthode de Richardon pour les matrices SDP

Nombre d'itérations pour une précision fixée:

$$\|x^{k+1} - x\|_2 \leq \Big(\rho(I - \alpha_{opt} A)\Big)^k \|x^1 - x\|_2,$$

$$\|x^{k+1} - x\|_2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|x^1 - x\|_2,$$

On cherche le nb d'itération $k$ pour atteindre une précision $\epsilon$, ie

$$\left(\frac{\kappa - 1}{\kappa + 1}\right)^k \leq \epsilon,$$

$$k \geq \frac{\ln(\frac{1}{\epsilon})}{\ln(\frac{1+\frac{1}{\kappa}}{1-\frac{1}{\kappa}})}$$

Pour $\kappa$ grand:

$$k \gtrsim \frac{\kappa}{2} \ln(\frac{1}{\epsilon})$$

# Méthode de Richardon à pas variable pour les matrices SDP

Soit $A \in \mathcal{M}_n$ SDP (Symétrique Définie Positive).

$$\text{On pose } e^k = x - x^k,\ \ r^k = Ae^k = b - Ax^k,$$

et on considère l'algorithme itératif: $x^1$ donné et pour $k = 1, \cdots$

$$\begin{cases} \alpha^k &= \dfrac{(r^k, r^k)}{(Ar^k, r^k)}, \\ x^{k+1} &= x^k + \alpha^k \ r^k. \end{cases}$$

On montre que

$$\alpha^k = \text{Argmin}_{\alpha\in\mathbb{R}}(Ae^{k+1}, e^{k+1}) = \alpha^2(Ar^k, r^k) - 2\alpha(r^k, r^k) + (Ae^k, e^k),$$

et

$$(Ae^{k+1}, e^{k+1}) = \Big(1 - \frac{(r^k, r^k)^2}{(Ar^k, r^k)(A^{-1}r^k, r^k)}\Big)(Ae^k, e^k),$$

d'où

$$(Ae^{k+1}, e^{k+1}) \leq \Big(1 - \frac{1}{\text{Cond}_2(A)}\Big)(Ae^k, e^k)$$

# Méthode de Richardon à pas variable pour les matrices SDP: algorithme

$$Ax = b \text{ avec } A \text{ matrice SDP}$$

- Choix de la précision $\epsilon$ sur le résidu relatif
- Initialisation: $x^1, r^1 = b - Ax^1, nr = nr^0 = \|r^1\|$
- Itérer tant que $\frac{nr}{nr^0} \geq \epsilon$
  - $p^k = Ar^k$
  - $\alpha^k = \frac{(r^k, r^k)}{(p^k, r^k)}$
  - $x^{k+1} = x^k + \alpha^k r^k$
  - $r^{k+1} = r^k - \alpha^k p^k$
  - $nr = \|r^{k+1}\|$

# Méthode de Richardon préconditionnée

Préconditionnement: matrice $C \in \mathcal{M}_n$ inversible

$$x^{k+1} = x^k + \alpha C^{-1}(b - Ax^k)$$

$$(x^{k+1} - x) = (I - \alpha C^{-1}A)(x^k - x)$$

$$B = (I - \alpha C^{-1}A)$$

On cherche un préconditionnement $C$ tel que

- $C \sim \alpha\ A$ ie $\rho(I - \alpha C^{-1}A) << 1$
- le système $Cy = r$ est peu coûteux à résoudre

# Exemple des matrices et préconditionnements SDP

$A, C \in \mathcal{M}_n$ symétriques définies positives.

Soient $y = C^{1/2}x$, $y^k = C^{1/2}x^k$, $c = C^{-1/2}b$ on a

$$\left(C^{-1/2}AC^{-1/2}\right)y = c,$$

La matrice $C^{-1/2}AC^{-1/2}$ est SDP, et

$$y^{k+1} = y^k + \alpha\left(c - C^{-1/2}AC^{-1/2}y^k\right)$$

Convergence ssi $\rho(I - \alpha C^{-1/2}AC^{-1/2}) < 1$

$$\alpha_{opt} = \frac{2}{\lambda_{min}(C^{-1/2}AC^{-1/2}) + \lambda_{max}(C^{-1/2}AC^{-1/2})}$$

$$\rho(I - \alpha_{opt}C^{-1/2}AC^{-1/2}) = \frac{\lambda_{max}(C^{-1/2}AC^{-1/2}) - \lambda_{min}(C^{-1/2}AC^{-1/2})}{\lambda_{min}(C^{-1/2}AC^{-1/2}) + \lambda_{max}(C^{-1/2}AC^{-1/2})}$$

# Exemple des matrices et préconditionnements SDP

$A, C \in \mathcal{M}_n$ symétriques définies positives.

Soient $y = C^{1/2}x$, $y^k = C^{1/2}x^k$, $c = C^{-1/2}b$ on a

$$\left(C^{-1/2}AC^{-1/2}\right)y = c,$$

La matrice $C^{-1/2}AC^{-1/2}$ est SDP, et

$$y^{k+1} = y^k + \alpha\left(c - C^{-1/2}AC^{-1/2}y^k\right)$$

Convergence ssi $\rho(I - \alpha C^{-1/2}AC^{-1/2}) < 1$

$$\alpha_{opt} = \frac{2}{\lambda_{min}(C^{-1/2}AC^{-1/2}) + \lambda_{max}(C^{-1/2}AC^{-1/2})}$$

$$\rho(I - \alpha_{opt}C^{-1/2}AC^{-1/2}) = \frac{\lambda_{max}(C^{-1/2}AC^{-1/2}) - \lambda_{min}(C^{-1/2}AC^{-1/2})}{\lambda_{min}(C^{-1/2}AC^{-1/2}) + \lambda_{max}(C^{-1/2}AC^{-1/2})}$$

# Méthode de Richardon préconditionnée à pas variable pour les matrices et préconditionnements SDP: algorithme

Soient $A$ et $C$ SDP et le système $Ax = b$.
On applique l'algorithme de Richardon à pas variable au système

$$C^{-1/2}AC^{-1/2}y = C^{-1/2}b.$$

Il se formule comme précédemment avec la matrice $\widetilde{A} = C^{-1/2}AC^{-1/2}$, le second membre $c = C^{-1/2}b$, les itérés $y^k = C^{1/2}x^k$ et les résidus $\tilde{r}^k = C^{-1/2}r^k$.
En repassant à $A$, $x$, $r$ on obtient:

- Choix de la précision $\epsilon$ sur le résidu relatif
- Initialisation: $x^1, r^1 = b - Ax^1, nr = nr^0 = \|r^1\|$
- Itérer tant que $\frac{nr}{nr^0} \geq \epsilon$
  - $q^k = C^{-1}r^k$
  - $p^k = Aq^k$
  - $\alpha^k = \frac{(q^k, r^k)}{(p^k, q^k)}$
  - $x^{k+1} = x^k + \alpha^k q^k$
  - $r^{k+1} = r^k - \alpha^k p^k$
  - $nr = \|r^{k+1}\|$

# Exemples de préconditionnements

$$A = D - E - F$$

avec $D$ diagonale de $A$ (supposée inversible), $D - E = \text{tril}(A)$, $D - F = \text{triu}(A)$

- Jacobi:

$$C = D$$

- Gauss Seidel

$$C = D - E \text{ ou } C = D - F$$

- SOR (Successive over relaxation) $\omega \in (0, 2)$

$$C = \frac{D}{\omega} - E$$

- SSOR (Symmetric Successive over relaxation) $\omega \in (0, 2)$

$$C = \left(\frac{D}{\omega} - E\right)\left(\frac{D}{\omega} - F\right)$$

# Jacobi

$C = D$ et $\alpha = 1$

$$x^{k+1} = x^k + D^{-1}(b - Ax^k)$$

$$Dx^{k+1} = b - (A - D)x^k$$

Pour $i = 1, \cdots, n$:

$$A_{i,i}x_i^{k+1} = b_i - \sum_{j \neq i} A_{i,j}x_j^k$$

# Gauss Seidel $A = D - E - F$

$C = D - E$ et $\alpha = 1$

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

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

Pour $i = 1, \cdots, n$:

$$A_{i,i}x_i^{k+1} = b_i - \sum_{j<i} A_{i,j}x_j^{k+1} - \sum_{j>i} A_{i,j}x_j^k$$

# SOR $A = D - E - F$

Pour $i = 1, \cdots, n$:

$$\left\{ \begin{array}{rl} A_{i,i}\tilde{x}_i^{k+1} &= b_i - \sum_{j<i} A_{i,j}x_j^{k+1} - \sum_{j>i} A_{i,j}x_j^k \\ x_i^{k+1} &= \omega\tilde{x}_i^{k+1} + (1-\omega)x_i^k \end{array} \right.$$

Vérification de $C = \frac{D}{\omega} - E$.

$$A_{i,i}\Big(\frac{x_i^{k+1}}{\omega} - \frac{(1-\omega)}{\omega}x_i^k\Big) = b_i - \sum_{j<i} A_{i,j}x_j^{k+1} - \sum_{j>i} A_{i,j}x_j^k$$

$$(\frac{D}{\omega} - E)x^{k+1} = b - \Big(-F - \frac{(1-\omega)}{\omega}D\Big)x^k = (\frac{D}{\omega} - E)x^k + (b - Ax^k)$$

# Convergence de Gauss Seidel

Si $A$ est une matrice SDP, alors $\rho\Big(I-(D-E)^{-1}A\Big)<1$ et la méthode de Gauss Seidel converge

On va montrer que pour tous $A$ SDP et $M$ inversible telle que $\Big(M^t+M-A\Big)$ SDP alors

$$\rho(I-M^{-1}A)<1$$

Preuve: on considère la norme sur $\mathbb{R}^n$ $\|x\|_\star^2=(Ax,x)$ et on va montrer que

$$\|I-M^{-1}A\|_\star^2=\sup_{x\neq 0}\frac{\|(I-M^{-1}A)x\|_\star^2}{\|x\|_\star^2}<1$$

Soit $x\neq 0$, on définit $y\neq 0$ tel que $Ax=My$

$$\begin{aligned}\|(I-M^{-1}A)x\|_\star^2 &= (A(x-y),(x-y))\\ &= \|x\|_\star^2+(Ay,y)-2(Ax,y)\\ &= \|x\|_\star^2+(Ay,y)-2(My,y)\\ &= \|x\|_\star^2-((M^t+M-A)y,y)<\|x\|_\star^2\end{aligned}$$

On conclut pour Gauss Seidel avec
$M^t+M-A=D-E+D-F-D+E+F=D>0$

# Solveurs non linéaires: plan

- Rappels de calculs différentiels pour des fonctions vectorielles de variable vectorielle.
- Algorithme de Newton pour résoudre $f(x) = 0$ avec $f \in C^1(U, \mathbb{R}^n)$ avec $U$ ouvert de $\mathbb{R}^n$.
- Convergence quadratique de l'algorithme de Newton pour $f \in C^2(U, \mathbb{R}^n)$.

# Rappels sur les fonctions vectorielles: différentielles

- Application linéaire tangente: soit $U$ un ouvert de $\mathbb{R}^n$ et $f : U \to \mathbb{R}^m$, on dit que $f$ est différentiable au point $x \in U$ ssi il existe une application linéaire notée $f'(x) \in \mathcal{L}\left(\mathbb{R}^n, \mathbb{R}^m\right)$ telle que

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

- Si $f$ est différentiable au point $x \in U$ alors $f$ est continue en $x$.
- Différentielle: si $f$ est différentiable pour tout $x \in U$, on note $x \to f'(x)$ l'application de $U$ dans $\mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ appelée différentielle de $f$

# Rappels sur les fonctions vectorielles: différentielles

Exemples:

- pour $n = 1$ on retrouve la dérivée au sens classique $f'(x) \in \mathbb{R}^m$
- pour $m = 1$, $f'(x)$ est une forme linéaire de $\mathcal{L}(\mathbb{R}^n, \mathbb{R})$
  - Gradient $\nabla f(x) \in \mathbb{R}^n$: grace à la structure euclidienne de $\mathbb{R}^n$, il existe un unique vecteur $\nabla f(x)$ appelé gradient de $f$ au point $x$ tel que $(\nabla f(x), v) = f'(x)(v)$ pour tout $v \in \mathbb{R}^n$ et dont la définition dépend du produit scalaire.

# Méthodes de Newton pour résoudre $f(x) = 0$

Soit

$$f : U \subset \mathbb{R}^n \to \mathbb{R}^n,$$

telle qu'il existe $\bar{x} \in U$ avec $f(\bar{x}) = 0$.

Etant donné $x^1 \in U$, la méthode de Newton calcule une suite $x^k$, $k = 2, \cdots$, par linéarisations successives de $f$ ie on approche l'équation $f(x^{k+1}) = 0$ par sa linéarisation au voisinage de $x^k$:

$$f'(x^k)(x^{k+1} - x^k) = -f(x^k).$$

- A chaque itération il faudra donc calculer la dérivée $f'(x^k)$ et résoudre un système linéaire.
- L'analyse de la méthode de Newton doit donner des conditions suffisantes sur $f$ et sur $x^1$ pour que $f'(x^k)$ soit inversible pour tout $k = 1, \cdots$, et pour que la suite $x^k$ converge vers $\bar{x}$.

# Rappels sur les fonctions vectorielles $f : U \subset \mathbb{R}^n \to \mathbb{R}^m$

Dérivées partielles: $\frac{\partial f}{\partial x_j}(x)$ est la dérivée (si elle existe) de $f$ selon la direction $e_j$ au point $x$ ie

$$\frac{\partial f}{\partial x_j}(x) = lim_{h_j \to 0} \frac{f(x + h_j e_j) - f(x)}{h_j}.$$

- Si $f$ est différentiable au point $x$ alors elle admet des dérivées partielles au point $x$ et

$$f'(x)(h) = \sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x) h_j \text{ pour tout } h \in \mathbb{R}^n.$$

Matrice représentant l'application linéaire $f'(x)$:

$$J(x) \in \mathcal{M}_{m,n} \text{ telle que } J_{i,j}(x) = \frac{\partial f_i}{\partial x_j}(x), i = 1, \cdots, m; j = 1, \cdots, n$$

est appelée la matrice Jacobienne de $f$ au point $x$.

# Rappels sur les fonctions vectorielles $f : U \subset \mathbb{R}^n \to \mathbb{R}^m$

- La réciproque n'est pas vraie: si $f$ admet des dérivées partielles en $x$, elle n'est pas nécéssairement différentiable au point $x$.
  - Exemple: $f(x_1, x_2) = \frac{x_1 x_2}{\sqrt{x_1^2 + x_2^2}}$ a des dérivées partielles nulles en 0 mais n'est pas différentiable en 0.
- Si $f$ admet des dérivées partielles continues au point $x$ pour tout $i = 1 \cdots, n$ alors $f$ est différentiable au point $x$
- $f$ est continuement différentiable sur $U$ (ie $f'$ existe et est continue sur $U$) ssi $f$ admet des dérivées partielles continues sur $U$. On dit que $f$ est $C^1\left(U, \mathbb{R}^n\right)$.

# Rappels sur les fonctions vectorielles: formule des accroissements finis

- Rappel dans le cas $n = m = 1$ (théorème de Rolle): $f : [a, b] \subset \mathbb{R} \to \mathbb{R}^m$: si $f$ est continue sur $[a, b]$ et différentiable sur $(a, b)$ alors il existe $c \in (a, b)$ tel que
$$f(b) - f(a) = f'(c)(b - a).$$
- Extension au cas $m = 1$, $n \geq 1$
$[a, b] = \{(1 - t)a + tb, t \in [0, 1]\} \subset U \subset \mathbb{R}^n$,
$(a, b) = \{(1 - t)a + tb, t \in (0, 1)\}$.
Si $f$ est différentiable sur $U$, alors il existe $c \in (a, b)$ tel que
$$f(b) - f(a) = f'(c)(b - a) = (\nabla f(c), b - a) \text{ aussi noté } \nabla f(c) \cdot (b - a)$$
Preuve: on applique le théorème de Rolle à $\varphi(t) = f((1 - t)a + tb)$

# Rappels sur les fonctions vectorielles: formule des accroissements finis

- Cas général: $n \geq 1$, $m \geq 1$: $f : U \subset \mathbb{R}^n \to \mathbb{R}^m$ différentiable sur $U$ et $[a, b] \subset U$, alors

$$\|f(b) - f(a)\| \leq \sup_{x \in (a,b)} \|f'(x)\| \|b - a\|.$$

- Preuve:
  - Soit $\phi(t) = f((1-t)a + tb)$, on a $\phi'(t) = f'((1-t)a + tb)(b-a)$ et

$$\|\phi'(t)\| \leq \sup_{x \in (a,b)} \|f'(x)\| \|b - a\|,$$

on conclut par $f(b) - f(a) = \int_0^1 \phi'(t)dt$, d'où

$$\|f(b) - f(a)\| \leq \int_0^1 \|\phi'(t)\| dt \leq \sup_{x \in (a,b)} \|f'(x)\| \|b - a\|.$$

# Rappels sur les fonctions vectorielles: différentielle d'ordre 2

Soit $f \in C^1(U, \mathbb{R}^m)$ avec $U$ ouvert de $\mathbb{R}^n$. On a $f' \in C^0\Big(U, \mathcal{L}(\mathbb{R}^n; \mathbb{R}^m)\Big)$. Si $f'$ est continuement différentiable sur $U$ on dit que $f \in C^2(U, \mathbb{R}^m)$ et on note $f''$ sa différentielle appelée différentielle seconde de $f$. La différentielle seconde $f''(x)$ est un élément de $\mathcal{L}\Big(\mathbb{R}^n; \mathcal{L}(\mathbb{R}^n; \mathbb{R}^m)\Big)$ isomorphe à l'ensemble des applications bilinéaires $\mathcal{L}\Big(\mathbb{R}^n, \mathbb{R}^n; \mathbb{R}^m\Big)$.
Dérivées partielles d'ordre 2: si $f$ est différentiable d'ordre 2 en $x$ alors elle admet des dérivées partielles d'ordre 2 au point $x$ notées $\dfrac{\partial^2 f(x)}{\partial x_i \partial x_j}$ avec

$$f''(x)(h, k) = \sum_{i,j=1}^{m} \frac{\partial^2 f(x)}{\partial x_i \partial x_j} h_i k_j \text{ pour tous } h, k \in \mathbb{R}^n$$

On a alors le théorème de Schwarz: $f''(x)$ est symétrique au sens où

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i} \text{ pour tous } i, j = 1, \cdots, n.$$

Dans le cas $m = 1$, on appelle $H(x) \in \mathcal{M}_{n,n}$ la matrice dite Hessienne représentant la forme bilinéaire $f''(x)$ dans la base canonique.

# Rappels sur les fonctions vectorielles: formule de Taylor à l'ordre 2 dans le cas $f \in C^2\Big(U, \mathbb{R}^m\Big)$

Soit $f \in C^2(U, \mathbb{R}^m)$ avec $U$ ouvert de $\mathbb{R}^n$ et $a, b \in U$ tels que $[a, b] \subset U$. Alors on a

$$\|f(b) - f(a) - f'(a)(b-a)\| \leq \frac{1}{2} \sup_{x \in (a,b)} \|f''(x)\| \|b-a\|^2,$$

avec

$$\|f''(x)\| = \sup_{u,v \neq 0 \in \mathbb{R}^n} \frac{\|f''(x)(u,v)\|}{\|u\|\|v\|}.$$

## Rappels sur les fonctions vectorielles: formule de Taylor à l'ordre 2 dans le cas $f \in C^2\Big(U, \mathbb{R}^m\Big)$

**Preuve**: soit

$$\varphi(t) = f(x + t(y-x)) - f(x) - t\ f'(x)(y-x),$$

on a $\varphi'(t) = \Big(f'(x + t(y-x)) - f'(x)\Big)(y-x)$ et donc

$$\varphi(1) = \varphi(1) - \varphi(0) = \int_0^1 \Big(f'(x + t(y-x)) - f'(x)\Big)(y-x)dt$$

$$\|\varphi(1)\| = \|f(y) - f(x) - f'(x)(y-x)\| \leq \int_0^1 \|f'(x+t(y-x)) - f'(x)\|\|y-x\|dt$$

on conclut par la formule des accroissements finis sur $f' \in C^1(U, \mathbb{R}^m)$:

$$\|f'(x + t(y-x)) - f'(x)\| \leq \sup_{z \in (x+t(y-x),y)} \|f''(z)\|\ \|y-x\|\ t$$

# Algorithme de Newton

Soit

$$f \in C^1(U, \mathbb{R}^n), \ \ U \text{ ouvert de } \mathbb{R}^n$$

telle qu'il existe $\bar{x} \in U$ avec $f(\bar{x}) = 0$.

Etant donné $x^1 \in U$, la méthode de Newton calcule une suite $x^k$, $k = 2, \cdots$, par linéarisations successives de $f$ ie on approche l'équation $f(x^{k+1}) = 0$ par sa linéarisation au voisinage de $x^k$:

$$f'(x^k)(x^{k+1} - x^k) = -f(x^k).$$

- A chaque itération il faudra donc calculer la dérivée $f'(x^k)$ et résoudre un système linéaire.
- L'analyse de la méthode de Newton doit donner des conditions suffisantes sur $f$ et sur $x^1$ pour que $f'(x^k)$ soit inversible pour tout $k = 1, \cdots$, et pour que la suite $x^k$ converge vers $\bar{x}$.

# Convergence quadratique de l'algorithme de Newton

Soit $f \in C^2(U, \mathbb{R}^n)$ avec $U$ ouvert de $\mathbb{R}^n$ et $\bar{x} \in U$ tel que $f(\bar{x}) = 0$. On suppose que $f'(\bar{x})$ est inversible. Alors il existe $\alpha > 0$ et $\beta > 0$ tels que

- $B(\bar{x}, \alpha) = \{x \mid \|x - \bar{x}\| < \alpha\} \subset U$
- Si $x^1 \in B(\bar{x}, \alpha)$ alors la suite $x^k, k \in \mathbb{N}$ est bien définie et $x^k \in B(\bar{x}, \alpha)$ pour tout $k \in \mathbb{N}$
- Si $x^1 \in B(\bar{x}, \alpha)$ alors la suite $x^k, k \in \mathbb{N}$ converge vers $\bar{x}$ et

$$\|x^{k+1} - \bar{x}\| \leq \beta \|x^k - \bar{x}\|^2 \text{ (convergence quadratique)}$$

# Convergence quadratique de l'algorithme de Newton: Preuve

On commence par montrer le lemme suivant:
Soit $f \in C^2(U, \mathbb{R}^n)$ avec $U$ ouvert de $\mathbb{R}^n$ et $\bar{x} \in U$ tel que $f'(\bar{x})$ est inversible. Alors il existe $\gamma > 0$, $C_1 > 0$, $C_2 > 0$ tels que $B(\bar{x}, \gamma) \subset U$ et

- $f'(x)$ inversible et $\|(f'(x))^{-1}\| \leq C_1$ pour tous $x \in B(\bar{x}, \gamma)$
- $\|f(y) - f(x) - f'(x)(y-x)\| \leq C_2 \|y - x\|^2$ pour tous $(x, y) \in B(\bar{x}, \gamma)$

**Preuve**: le point 1 est une application de $\|(I-A)^{-1}\| \leq \frac{1}{(1-\|A\|)}$ pour $\|A\| < 1$ et le point 2 résulte directement de la formule de Taylor d'ordre 2.

# Convergence quadratique de l'algorithme de Newton: Preuve suite

Soit $\alpha = \min\left(\gamma, \frac{1}{C_1 C_2}\right)$. On suppose que $x^k \in B(\bar{x}, \alpha)$ (et donc $f'(x^k)$ inversible), on va montrer que ceci implique $x^{k+1} \in B(\bar{x}, \alpha)$.

Comme $f'(x^k)(x^{k+1} - x^k) + f(x^k) = 0$, on a

$$f'(x^k)(x^{k+1} - \bar{x}) = f(\bar{x}) - f(x^k) - f'(x^k)(\bar{x} - x^k)$$

d'où

$$\|x^{k+1} - \bar{x}\| \leq \|(f'(x^k))^{-1}\| C_2 \|x^k - \bar{x}\|^2 \leq C_1 C_2 \|x^k - \bar{x}\|^2 \leq \alpha.$$

On a donc $x^{k+1} \in B(\bar{x}, \alpha)$ et on donc a montré par récurrence que c'est vrai pour tout $k \in \mathbb{N}$ si $x^1 \in B(\bar{x}, \alpha)$. On a ensuite

$$\|x^{k+1} - \bar{x}\| \leq (C_1 C_2)^{2k-1} \|x^1 - \bar{x}\|^{2k}$$

Par ailleurs comme $x^1 \in B(\bar{x}, \alpha)$, on a $\|x^1 - \bar{x}\| < \frac{1}{C_1 C_2}$, d'où la convergence de la suite vers $\bar{x}$. Elle est quadratique avec $\beta = C_1 C_2$.

# Variantes de l'algorithme de Newton: Inexact Newton

Si le système linéaire est résolu avec une méthode itérative on veut ajuster le critère d'arrêt du solveur linéaire pour préserver la convergence quadratique de l'algorithme de Newton à moindre coût.
Ceci revient à résoudre le système linéaire de façon approchée avec un résidu $r^k$

$$f'(x^k)(x^{k+1} - x^k) = -f(x^k) + r^k,$$

tel que

$$\|r^k\| \leq \eta_k \|f(x^k)\|.$$

Différentes stratégies existent pour ajuster $\eta_k$ de façon à préserver la convergence quadratique sans trop résoudre le système linéaire: par exemple

$$\eta_k = \min\left(\eta_{max}, \frac{|\|f(x^k)\| - \|f(x^{k-1}) + f'(x^{k-1})(x^k - x^{k-1})\||}{\|f(x^{k-1})\|}\right),$$

avec $\eta_{max} = 0.1$. Ce choix prend en compte la fiabilité de l'approximation tangentielle de $f$.

# Variantes de l'algorithme de Newton: Quasi Newton

On n'a pas toujours en pratique accès au calcul exact de la Jacobienne de $f$. Il existe des méthodes itératives pour l'approcher comme par exemple l'algorithme de Broyden suivant:

- Initialisation: $x^0$, $x^1 \in U$, $B^0 \in \mathcal{M}_n$
- Itérations
  - On pose $\delta^k = x^k - x^{k-1}$ et $y^k = f(x^k) - f(x^{k-1})$
  - Mise à jour de rang 1 de la Jacobienne approchée:

$$B^k = B^{k-1} + \left(\frac{y^k - B^{k-1}\delta^k}{(\delta^k)^t\delta^k}\right)(\delta^k)^t$$

  - On résoud le système linéaire $B^k\left(x^{k+1} - x^k\right) = -f(x^k)$

La correction de rang 1 de la Jacobienne approchée est construite pour vérifier la condition dite de la sécante:

$$B^k(x^k - x^{k-1}) = f(x^k) - f(x^{k-1}).$$