Méthodes itératives

2022–2023

Introduction

On cherche une suite $x^{(p)}$ de $\mathbb{R}^n$ convergeant vers la solution x^* de $A \cdot x = b$

$$\forall x^{(0)} \in \mathbb{R}^n \qquad x^{(p+1)} = H(x^{(p)})$$

- La matrice A n'est jamais modifiée
- Problème du suivi de la convergence et du choix du test d'arrêt
- La solution obtenue n'est pas exacte
- La matrice doit vérifier des conditions de convergence
- La vitesse de convergence dépend de la valeur des coefficients de la matrice

Familles de méthodes

Algorithmes itératifs de relaxation

Algorithmes associés à une décomposition de A sous la forme $M - N$: Gauss-Seidel, Jacobi, SOR

Méthodes de gradient

Plus grande pente, directions conjuguées, gradient conjugué

Première partie I

Méthodes itératives de relaxation

Algorithme itératif de relaxation associé à une décomposition de A sous une forme $M - N$

Soit $A = M - N$ avec M inversible

$$
\begin{aligned}
A \cdot x^* = b &\iff (M - N) \cdot x^* = b \\
&\iff M \cdot x^* = N \cdot x^* + b \\
&\iff x^* = M^{-1} \cdot N \cdot x^* + M^{-1} \cdot b
\end{aligned}
$$

Algorithme itératif de relaxation associé :

$$
\begin{cases}
x^{(0)} \in \mathbb{R}^n \\
x^{(p+1)} = M^{-1} \cdot N \cdot x^{(p)} + M^{-1} \cdot b \\
\phantom{x^{(p+1)}} = M^{-1} \cdot (M - A) \cdot x^{(p)} + M^{-1} \cdot b \\
\phantom{x^{(p+1)}} = x^{(p)} + M^{-1} \cdot r^{(p)}
\end{cases}
$$

avec $r^{(p)} = b - A \cdot x^{(p)}$ le vecteur résidu

Coût d'une itération (en dehors du calcul du résidu) : résolution d'un système de matrice M

Soit la décomposition de $A = D - E - F$ avec :

$$D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & a_{nn} \end{pmatrix} \qquad E = \begin{pmatrix} 0 & \dots & \dots & 0 \\ -a_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ -a_{n1} & \dots & -a_{nn-1} & 0 \end{pmatrix}$$

$$F = \begin{pmatrix} 0 & -a_{12} & \dots & -a_{1\,n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -a_{n-1\,n} \\ 0 & \dots & \dots & 0 \end{pmatrix}$$

Deux variantes

Algorithme de Jacobi : $M = D$ et $N = E + F$

L'itération p consiste à résoudre l'équation :
$D \cdot x^{(p+1)} - (E + F) \cdot x^{(p)} = b$
C.N. d'application : D inversible $i = 1, \dots, n$ $a_{ii} \neq 0$

Algorithme de Gauss-Seidel : $M = D - E$ et $N = F$

L'itération p consiste à résoudre l'équation :
$(D - E) \cdot x^{(p+1)} - F \cdot x^{(p)} = b$
C.N. d'application : $(D - E)$ inversible $\iff$ $i = 1, \dots, n$ $a_{ii} \neq 0$

Condition d'arrêt

Mise en œuvre du test d'arrêt (à la fin de chaque itération) Choix d'une norme vectorielle $|\cdot|$, d'une précision ε puis d'un test d'arrêt :

- $\left\|x^{(p+1)} - x^{(p)}\right\| < \varepsilon$: simple mais numériquement dangereux (risque d'arrêt prématuré loin de la solution).
- $\left\|b - A \cdot x^{(p+1)}\right\| / \|b\| < \varepsilon$: numériquement plus sûr.

La mise en œuvre de ce test d'arrêt n'entraîne pas de calculs supplémentaires

Pour Jacobi :	$x^{(p+1)} = D^{-1} \cdot \left[b - A.x^{(p)}\right] + x^{(p)}$
Pour Gauss-Seidel :	$x^{(p+1)} = (D - E)^{-1} \cdot \left[b - A.x^{(p)}\right] + x^{(p)}$

Convergence d'un algorithme issu d'une décomposition de $A = M - N$

Soit $x^{(p)} = x^* + \varepsilon^{(p)}$

$$\left.\begin{array}{r} x^{(p+1)} = M^{-1} \cdot N \cdot x^{(p)} + M^{-1} \cdot b \\ x^* = M^{-1} \cdot N \cdot x^* + M^{-1} \cdot b \end{array}\right\} \implies \varepsilon^{(p+1)} = M^{-1} \cdot N \cdot \varepsilon^{(p)}$$

$$\implies \varepsilon^{(p)} = \left(M^{-1} \cdot N\right)^p \cdot \varepsilon^{(0)}$$

$$\begin{aligned}
\text{Convergence} &\iff \forall x^{(0)} \in \mathbb{R}^n \ \lim_{p \to +\infty} x^{(p)} = x^* \\
&\iff \forall \varepsilon^{(0)} \in \mathbb{R}^n \ \lim_{p \to +\infty} \varepsilon^{(p)} = 0 \\
&\iff \forall \varepsilon^{(0)} \in \mathbb{R}^n \ \lim_{p \to +\infty} \left(M^{-1} \cdot N\right)^p \cdot \varepsilon(0) = 0 \\
&\iff \lim_{p \to +\infty} \left(M^{-1} \cdot N\right)^p = 0 \\
&\iff \rho\left(M^{-1} \cdot N\right) < 1 \ \ \text{(propriété admise)}
\end{aligned}$$

CNS de convergence

CNS de convergence (admise)

$$\lim_{p \to +\infty} \left(M^{-1} \cdot N\right)^p = 0 \iff \rho\left(M^{-1} \cdot N\right) < 1$$

ρ désignant le rayon spectral d'une matrice : soient $\lambda_i \in \mathbb{C}, i = 1, \ldots, n$ les valeurs propres d'une matrice A :

$$\rho(A) = \max_{i=1,\ldots,n} |\lambda_i|$$

Autre version de la CNS de convergence :

$$\forall \lambda \in \mathbb{C} \text{ valeur propre de } M^{-1} \cdot N \ : \ |\lambda| < 1$$

Première possibilité pour vérifier la convergence : utiliser la CNS
Jacobi : $\rho(D^{-1} \cdot (E + F)) < 1$
Gauss-Seidel : $\rho((D - E)^{-1} \cdot F) < 1$

Exercice 1

Soit le système $A \cdot x = b$ avec $A \in \mathcal{M}_n(\mathbb{R})$ inversible.
On considère une décomposition de A sous la forme $M - N$.
On supposera (pour simplifier) que toutes les valeurs propres de la matrice $M^{-1} \cdot N$ sont réelles et que tous les vecteurs propres associés appartiennent à $\mathbb{R}^n$.
Montrer la propriété suivante (réciproque de la propriété admise en cours) :

La méthode itérative de relaxation associée à la décomposition de A converge vers x quelque soit le vecteur initial $\implies \rho(M^{-1} \cdot N) < 1$

CS de convergence

Deuxième possibilité pour vérifier la convergence : utiliser des CS spécifiques (déduites de la CNS)

A à diagonale dominante (stricte) : $i = 1, \dots, n \qquad |a_{ii}| \underset{>}{\geqslant} \sum_{j=1, j \neq i}^{n} |a_{ij}|$

Théorème 1 : Si A est à diagonale dominante stricte, alors les algorithmes de Jacobi et de Gauss-Seidel convergent quelque soit le vecteur de départ.

Théorème 2 : Si A est à diagonale dominante et si :

$$\exists k \in \{1, \dots, n\} \; |a_{kk}| > \sum_{j=1, j \neq k}^{n} |a_{kj}|$$

alors l'algorithme de Gauss-Seidel converge quelque soit le vecteur de départ.

Théorème 3 : Si A est symétrique définie positive, alors l'algorithme de Gauss-Seidel converge quelque soit le vecteur de départ.

Méthode SOR (successive over-relaxation)

Technique d'accélération de Gauss-Seidel

$$x_i^{(p+1)} = \omega(\text{Gauss-Seidel}) + (1-\omega)x_i^{(p)}$$

$$x_i^{(p+1)} = \frac{\omega}{a_{ii}}\Big(b_i - \sum_{j>i} a_{ij}x_j^{(p)} - \sum_{j<i} a_{ij}x_j^{(p+1)}\Big) + (1-\omega)x_i^{(p)}$$

C'est un algorithme itératif de relaxation issu d'une décomposition de A sous la forme $M_\omega - N_\omega$ avec $M_\omega = \frac{D}{\omega} - E$ et $N_\omega = \left(\frac{1}{\omega} - 1\right) D + F$

Conditions sur ω pour assurer la convergence ?

Quelques résultats pour des cas particuliers :

- A symétrique définie positive et tridiagonale par blocs, un seul paramètre optimal ω : $\omega = \frac{2}{1+\sqrt{1-\rho(D^{-1}\cdot(E+F))^2}}$

Deuxième partie II

Méthodes de gradient

Principes

Rappel : si $F : \mathbb{R}^n \rightarrow \mathbb{R}$ et $x \in \mathbb{R}^n$, $grad(F)(x) = \begin{pmatrix} \frac{\partial F}{\partial x_1}(x) \\ \vdots \\ \frac{\partial F}{\partial x_n}(x) \end{pmatrix}$

<u>Propriété 1</u>

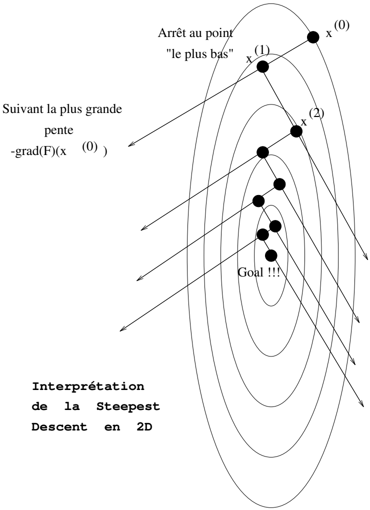
$$\begin{aligned} F : \mathbb{R}^n &\rightarrow \mathbb{R} \\ x &\mapsto F(x) = \frac{1}{2} x^T \cdot A \cdot x - x^T \cdot b \end{aligned}$$

Si A symétrique, $A \cdot x - b = grad(F)(x)$ ou $b - A \cdot x = -grad(F)(x)$

<u>Propriété 2</u>

Si A symétrique définie positive (supposé vrai pour toute méthode de gradient) : $A \cdot x^* = b \iff \forall x' \neq x^* \qquad F(x') > F(x^*)$

Interprétation de la “steepest descent” en 2D



Méthode de la Steepest Descent

A symétrique définie positive, $x^{(0)} \in \mathbb{R}^n$ quelconque
Au cours de l'itération p, $x^{(p+1)}$ se déduire de $x^{(p)}$ par :

$$x^{(p+1)} = x^{(p)} + \lambda_p r^{(p)}$$

- $r^{(p)}$: direction de descente (vecteur)
- λ_p : progression dans la direction de descente (scalaire)

Choix des paramètres de la descente :

- Direction : suivant la plus grande pente i.e. direction opposée au gradient en $x^{(p)}$: $r^{(p)} = -grad(F)(x^{(p)}) = b - A \cdot x^{(p)}$
- Progression : recherche du point "le plus bas" dans la direction $r^{(p)}$ obtenu en minimisant :

$$f(\lambda) = F(x^{(p)} + \lambda r^{(p)}) - F(x^{(p)}) = \frac{1}{2}\lambda^2 (r^{(p)})^T \cdot A \cdot r^{(p)} - \lambda (r^{(p)})^T \cdot r^{(p)}$$

Le minimum de cette fonction de λ_p est atteint en $\lambda_p = \frac{(r^{(p)})^T \cdot r^{(p)}}{(r^{(p)})^T \cdot A \cdot r^{(p)}}$

Algorithme

D'où l'algorithme :
$x^{(0)} = \dots$
$r^{(0)} = b - A \cdot x^{(0)}$
$p = 1$
Tant que $\frac{\|r^{(p-1)}\|}{\|b\|} > \varepsilon$ faire

$$\lambda_{p-1} = \frac{(r^{(p-1)})^T \cdot r^{(p-1)}}{(r^{(p-1)})^T \cdot A \cdot r^{(p-1)}}$$

$$x^{(p)} = x^{(p-1)} + \lambda_{p-1} r^{(p-1)}$$

$$r^{(p)} = b - A \cdot x^{(p)}$$

$$p = p + 1$$

Coût d'une itération : 2 produits matrice-vecteur mais
$r^{(p)} = b - A \cdot x^{(p)} = b - A \cdot x^{(p-1)} - \lambda_{p-1} A \cdot r^{(p-1)}$
$r^{(p)} = r^{(p-1)} - \lambda_{p-1} A \cdot r^{(p-1)}$

Propriétés de la Steepest Descent

- Deux directions de descente successives $r^{(p+1)}$ et $r^{(p)}$ sont orthogonales.
- Soit u vecteur propre de A. Si $x^{(0)}$ vérifie : $\exists \beta \in \mathbb{R}, \beta \neq 0$ t.q. $x^* - x^{(0)} = \beta u$, la SD atteint la solution exacte en une itération.
- Si toutes les valeurs propres de la matrice A sont égales (une seule valeur propre), alors, $\forall x^{(0)}$, la SD atteint la solution exacte en une itération.

- Soit u vecteur propre de A. Si $x^{(0)}$ vérifie : $\exists \beta \in \mathbb{R}, \beta \neq 0$ t.q. $x^* - x^{(0)} = \beta u = \varepsilon^{(0)}$, la SD atteint la solution exacte en une itération.

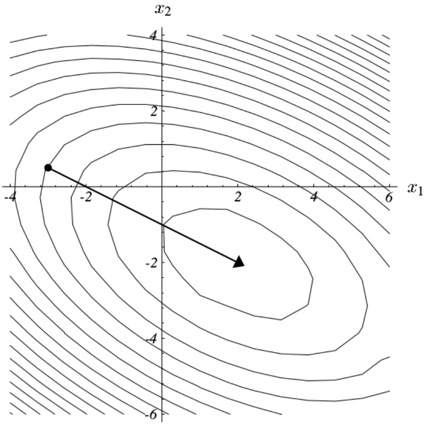


Figure 14: Steepest Descent converges to the exact solution on the first iteration if the error term is an eigenvector.

Propriétés de la Steepest Descent

- Si toutes les valeurs propres de la matrice A sont égales (une seule valeur propre), alors, $\forall x^{(0)}$, la SD atteint la solution exacte en une itération.

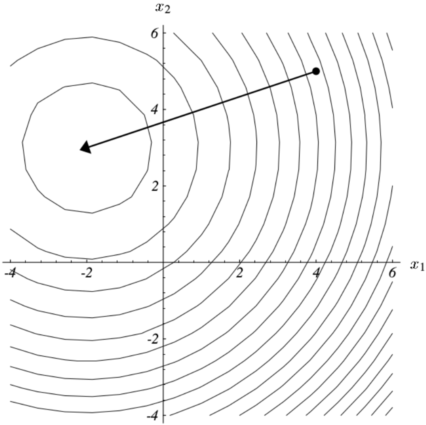
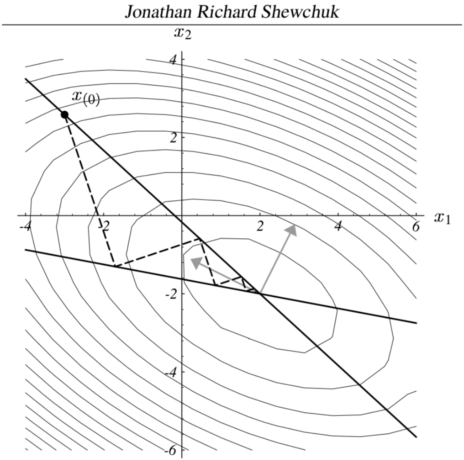


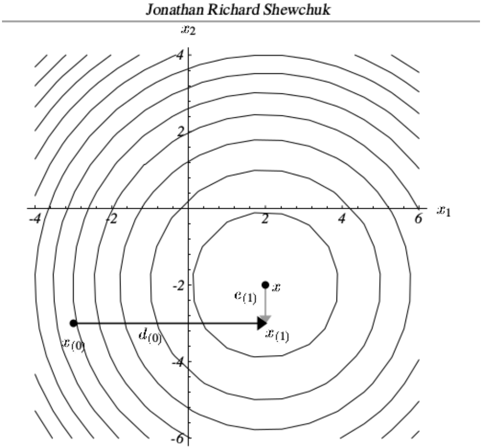
Figure 15: Steepest Descent converges to the exact solution on the first iteration if the eigenvalues are all equal.

cas défavorable de la "steepest descent" en 2D



Référence : *An Introduction to the Conjugate Gradient Method without the Agonizing Pain*, **Jonathan Richard Shewchuk**

Idée : ne considérer une direction qu'une fois



Problème : pour savoir où s'arrêter dans une direction, il faut connaître la suivante

Solution : considérer des directions A-conjuguées

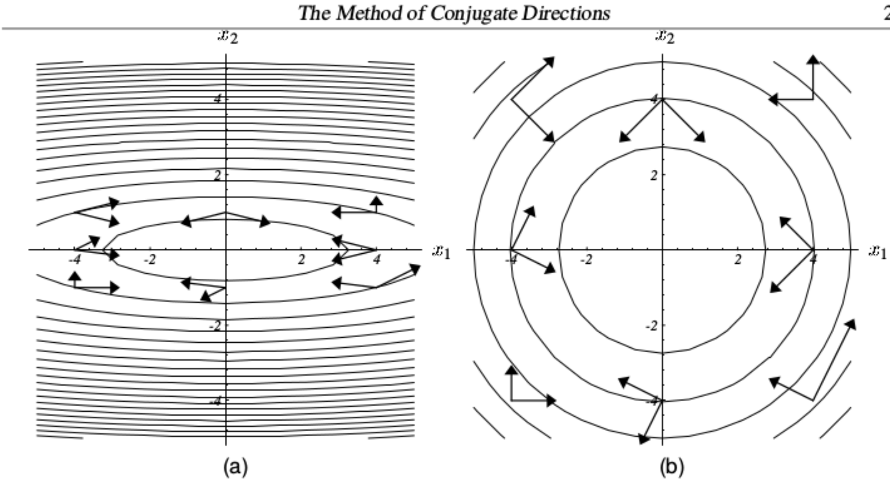


Figure 22: These pairs of vectors are A-orthogonal … because these pairs of vectors are orthogonal.

A-orthogonalisation de Gram-Schmidt

Soit u_i $i = 0, \ldots, n-1$ une base quelconque de $\mathbb{R}^n$ et $A \in \mathcal{M}_n(\mathbb{R})$ symétrique définie positive.

On considère le produit scalaire suivant (A-produit scalaire) :

$$\forall (x, y) \in (\mathbb{R}^n)^2, (x|y)_A = (x|A \cdot y)$$

On veut construire une nouvelle base d_i $i = 0, \ldots, n-1$ qui soit A-orthogonale i.e. $\forall i \neq j, (d_i|d_j)_A = 0$

$$\begin{aligned}
d_0 &= u_0 \\
d_1 &= u_1 + \beta_{10} d_0 \\
d_2 &= u_2 + \beta_{20} d_0 + \beta_{21} d_1 \\
d_{p+1} &= u_{p+1} + \sum_{j=0}^{p} \beta_{p+1j} d_j \\
&\ldots \\
d_{n-1} &= u_{n-1} + \sum_{j=0}^{n-2} \beta_{nj} d_j
\end{aligned}$$

A-orthogonalisation de Gram-Schmidt – suite

$$d_{p+1} = u_{p+1} + \sum_{j=0}^{p} \beta_{p+1j} d_j$$

Calcul de d_{p+1} en supposant $d_i, i = 0, \dots, p$ déjà calculés et $\forall j \neq i, (d_j|d_i)_A = 0$:

$$i = 0, \dots, p \qquad \begin{aligned} (d_{p+1}|d_i)_A &= (u_{p+1}|d_i)_A + \sum_{j=0}^{p} \beta_{p+1j}(d_j|d_i)_A \\ 0 &= (u_{p+1}|d_i)_A + \beta_{p+1i}(d_i|d_i)_A \end{aligned}$$

D'où

$$\beta_{p+1i} = -\frac{(u_{p+1}|d_i)_A}{(d_i|d_i)_A} = -\frac{(u_{p+1}|A \cdot d_i)}{(d_i|A \cdot d_i)}$$

Exercice 2 (avec nouvelles notations)

Soit à résoudre $A \cdot x = b$ avec $A \in \mathcal{M}_n(\mathbb{R})$ symétrique définie positive.
On note $u_i, \ i = 0, \ldots, n-1$ les différentes colonnes de A.
Le procédé de A-orthogonalisation de Gram-Schmidt est appliqué à $u_i, \ i = 0, \ldots, n-1$ pour construire une base $d_i, \ i = 0, \ldots, n-1$ A-orthogonale de $\mathbb{R}^n$.

1. Exprimer la solution du système $A \cdot x^* = b$ dans la base $d_i, \ i = 0, \ldots, n-1$.
2. Montrer que $\forall y \in \mathbb{R}^n, x^* = y + \displaystyle\sum_{i=0}^{n-1} \frac{(d_i | b - A \cdot y)}{(d_i | A \cdot d_i)} d_i$
3. On considère l'algorithme itératif suivant :

$$\forall x^{(0)} \in \mathbb{R}^n, \ x^{(p+1)} = x^{(p)} + \alpha_p d_p \text{ avec } \alpha_p = \frac{(d_p | b - A \cdot x^{(p)})}{(d_p | A \cdot d_p)}$$

   3.1 Donner la relation entre $x^{(p)}$ et $x^{(0)}$
   3.2 Montrer que pour $p = 1, 2, 3, \ldots (d_p | A \cdot x^{(p)}) = (d_p | A.x^{(0)})$
4. Montrer que l'algorithme de la question ci-dessus atteint exactement la solution du système $A \cdot x^* = b$ en n itérations.

Quelques propriétés

Propriété 0 (question 3 de l'exercice)

$$p = 1, 2, 3, \ldots, \ (d_p | A \cdot x^{(p)}) = (d_p | A \cdot x^{(0)}) \implies (d_p | r^{(p)}) = (d_p | r^{(0)})$$

Propriété 1

$$j = 0, \ldots, p-1, \ (r^{(p)} | d_j) = 0$$

Propriété 2

$$j = 0, \ldots, p-1, \ (r^{(p)} | u_j) = 0$$

Propriété 3

$$(r^{(p)} | d_p) = (r^{(p)} | u_p)$$

Mise en forme d'une itération de l'algorithme :

$x^{(0)} = ?,\ r^{(0)} = b - Ax^{(0)}$
$\{u_i\}_{i=0,\dots,n-1}$ colonnes de A
$d_0 = u_0$
Pour $p = 0, n-1$

$\alpha_p = \frac{(d_p|r^{(p)})}{(d_p|A \cdot d_p)}$

$x^{(p+1)} = x^{(p)} + \alpha_p d_p$

$r^{(p+1)} = r^{(p)} - \alpha_p A \cdot d_p$

pour $j = 0, \dots, p,\ \beta_{p+1j} = -\frac{(u_{p+1}|A \cdot d_j)}{(d_j|A \cdot d_j)}$

$d_{p+1} = u_{p+1} + \sum_{j=0}^{p} \beta_{p+1j} d_j$

Fin Pour

Coût : mémoire, calcul ?

Cas particulier du Gradient Conjugué

Soit $u_i = r^{(i)},\ i = 0, \ldots, n-1$

- 1ère conséquence :

$$\alpha_p = \frac{(d_p|r^{(p)})}{(d_p|A \cdot d_p)} = \frac{(r^{(p)}|r^{(p)})}{(d_p|A \cdot d_p)}$$

- 2ème conséquence : $j = 0, \ldots, p$

$$\beta_{p+1j} = -\frac{(u_{p+1}|A \cdot d_j)}{(d_j|A \cdot d_j)} = -\frac{(r^{(p+1)}|A \cdot d_j)}{(d_j|A \cdot d_j)}$$

D'autre part $r^{(j+1)} = r^{(j)} - \alpha_j A \cdot d_j$

$$(r^{(p+1)}|r^{(j+1)}) = (r^{(p+1)}|r^{(j)}) - \alpha_j (r^{(p+1)}|A \cdot d_j)$$

et donc

$$\alpha_j (r^{(p+1)}|A \cdot d_j) = (r^{(p+1)}|r^{(j)}) - (r^{(p+1)}|r^{(j+1)})$$

Cas particulier du gradient conjugué – suite

$$\alpha_j(r^{(p+1)}|A\cdot d_j) = (r^{(p+1)}|r^{(j)}) - (r^{(p+1)}|r^{(j+1)})$$

$j = p$

$$\begin{aligned}
\alpha_p(r^{(p+1)}|A\cdot d_p) &= (r^{(p+1)}|r^{(p)}) - (r^{(p+1)}|r^{(p+1)})\\
&= 0 - (r^{(p+1)}|r^{(p+1)}) \quad \text{(prop. 2)}\\
(r^{(p+1)}|A\cdot d_p) &= -\frac{1}{\alpha_p}(r^{(p+1)}|r^{(p+1)})\\
(r^{(p+1)}|A\cdot d_p) &= -\frac{(d_p|A\cdot d_p)}{(r^p|r^{(p)})}(r^{(p+1)}|r^{(p+1)})\\
-\frac{(r^{(p+1)}|A\cdot d_p)}{(d_p|A.d_p)} &= \frac{(r^{(p+1)}|r^{(p+1)})}{(r^p|r^{(p)})}\\
\Longrightarrow \beta_{p+1p} &= \frac{(r^{(p+1)}|r^{(p+1)})}{(r^p|r^{(p)})}
\end{aligned}$$

$j = 0, \ldots, p-1$

$$\begin{aligned}
\alpha_j(r^{(p+1)}|A\cdot d_j) &= (r^{(p+1)}|r^{(j)}) - (r^{(p+1)}|r^{(j+1)})\\
&= 0 - 0 \quad \text{(prop. 2)}\\
\Longrightarrow \beta_{p+1j} &= \frac{(r^{(p+1)}|A\cdot d_j)}{(d_j|A\cdot d_j)} = 0
\end{aligned}$$

d'où l'expression de d_{p+1} :

$$d_{p+1} = r^{(p+1)} + \beta_{p+1p}d_p \quad \text{avec} \quad \beta_{p+1p} = \frac{\left(r^{(p+1)}|r^{(p+1)}\right)}{\left(r^{(p)}|r^{(p)}\right)}$$

Algorithme final du Gradient Conjugué

$x^{(0)} =?$, $d^{(0)} = b - Ax^{(0)}$, $r^{(0)} = d^{(0)}$, $p = 0$

Boucler

$$\alpha_p = \frac{\left(r^{(p)}|r^{(p)}\right)}{(d_p|A \cdot d_p)}$$

$$x^{(p+1)} = x^{(p)} + \alpha_p d_p$$

$$r^{(p+1)} = r^{(p)} - \alpha_p A \cdot d_p$$

$$\beta_p = \beta_{p+1p} = \frac{\left(r^{(p+1)}|r^{(p+1)}\right)}{\left(r^{(p)}|r^{(p)}\right)}$$

$$d_{p+1} = r^{(p+1)} + \beta_p d_p$$

$$p = p + 1$$

Jusqu'à Convergence

Coût : mémoire, calcul ?

Convergence ?