Algorithmique numérique (LU3IN028)

Cours n°6 : Méthodes itératives pour la résolution de systèmes linéaires

Stef Graillat

Sorbonne Université

Résumé du cours précédent

Tranformée de Fourier discrète :

- Algorithme FFT : version récursive et itérative
- Complexité en $\mathcal{O}(n \log n)$
- Applications à la multiplication de polynômes, au traitement d'image et au traitement du signal

Objectifs

1. Résolution de systèmes linéaires par des méthodes itératives (Jacobi, Gauss-Seidel, SOR)
2. Méthode du gradient conjugué
3. Méthode de Krylov

Bibliographie

- An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, J. Shewchuk, 1994
- Scientific Computing, An Introductory Survey, Michael .T. Heath, McGraw-Hill, 2002
- Scientific Computing with Case Studies, Dianne P. O'Leary, SIAM, 2009
- Iterative methods for sparse linear systems, Y. Saad, SIAM, 2007
- Linear and Nonlinear Programming, Luenberger, Ye, 3e édition, Springer, 2010
- Algèbre linéaire numérique : Cours et exercices, Allaire, Kaber Sidi, Ellipses, 2002

Pourquoi utiliser les méthodes itératives

On cherche à résoudre une équation de la forme :

$$Ax = b$$

Les méthodes directes fournissent la solution x^* en un nombre fini d'opérations.

Mais

- la complexité est en $\mathcal{O}(n^3)$
- ne prennent pas en compte le caractère creux de la matrice (beaucoup de coefficients sont nuls).

Méthodes itératives

- On construit une suite de vecteurs (x_k), $k = 0, 1, \ldots$ qui tend vers x^*.
- Le point de départ est une approximation x_0 de x^*
- Pour construire cette suite, on utilise la linéarité pour décomposer la matrice A en une partie facilement inversible et un reste.

Principe Général

On décompose la matrice A en $A = M - N$, de telle façon que M soit facilement inversible.

Alors,

$$Ax = b \quad \Leftrightarrow \quad Mx = Nx + b$$

On calcule la suite de vecteurs (x_i) à partir d'un vecteur x_0 choisi arbitrairement et de la relation :

$$M\, x_{k+1} = N\, x_k + b \quad \Leftrightarrow \quad x_{k+1} = M^{-1}N\, x_k + M^{-1}b$$

C'est-à-dire

$$\begin{cases} x_0 & \text{donné} \\ x_{k+1} & = \quad M^{-1}N\, x_k + M^{-1}b \end{cases}$$

Posons le problème

Posons $C = M^{-1}N$, et $d = M^{-1}b$. Nous devons donc étudier la suite récurrente

$$\begin{cases} x^0 & \text{donné} \\ x_{k+1} & = Cx_k + d \end{cases}$$

Avec :

- x^* est point fixe de la fonction linéaire

$$x \mapsto C\,x + d$$

Question :

- Sous quelles conditions la suite va-t-elle converger ?

Normes matricielles

On appelle norme matricielle une norme définie sur $\mathcal{M}_n(\mathbb{C})$ qui est compatible avec la multiplication de matrice, c'est-à-dire :

$$\begin{array}{rccc} \|\cdot\| & : & \mathcal{M}_n(\mathbb{C}) & \to & \mathbb{R}^+ \\ & & A & \mapsto & \|A\| \end{array}$$

telle que $\forall A, B \in \mathcal{M}_n(\mathbb{C})$ et $\forall \lambda \in \mathbb{C}$:

- Séparation : $\|A\| = 0 \Leftrightarrow A = 0$,
- Homogénéité : $\|\lambda A\| = |\lambda| \cdot \|A\|$,
- Inégalité triangulaire : $\|A + B\| \leq \|A\| + \|B\|$,
- $\|A \cdot B\| \leq \|A\| \cdot \|B\|$.

Exemple de normes

Norme de Frobenius :
$\forall A \in \mathcal{M}_n(\mathbb{C})$

$$\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\mathrm{Tr}\left(A^T A\right)}$$

Normes induites (ou subordonnées) :
Soit $\|\cdot\|_v$ une norme vectorielle définie sur $\mathbb{C}^n$ la fonction qui $\forall A \in \mathcal{M}_n(\mathbb{C})$ associe

$$\|A\| = \max_{x\in\mathbb{C}^n, x\neq 0} \frac{\|Ax\|_v}{\|x\|_v}$$

est une norme matricielle dite norme matricielle induite ou subordonnée

Exemple de norme induite

Soit la norme vectorielle $\|\cdot\|_\infty$:
$\forall x \in \mathbb{C}^n$

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

Cette norme induit la norme matricielle $\|\cdot\|_\infty$ sur $\mathcal{M}_n(\mathbb{C})$:
$\forall A \in \mathcal{M}_n(\mathbb{C})$

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

Norme matricielle et norme vectorielle

Définition 1

On dit qu'une norme matricielle $\|\cdot\|$ est compatible avec une norme vectorielle $\|\cdot\|_v$ si $\forall x$

$$\|Ax\|_v \quad \leq \quad \|A\| \, \|x\|_v$$

Propriétés :

- Pour toute norme matricielle $\|\cdot\|$, il existe une norme vectorielle avec laquelle elle est compatible.
- Toute norme matricielle induite est compatible avec sa norme vectorielle.

Rayon spectral

Définition 2

Soit $A \in \mathcal{M}_n(\mathbb{C})$, on appelle rayon spectral de la matrice A le nombre réel

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

où λ_i sont les valeurs propres de A

Remarque : $\rho(\cdot)$ n'est pas une norme.

Propriétés :

- Pour toute norme matricielle $\|\cdot\|$ et pour toute matrice :

$$\rho(A) \leq \|A\|$$

- $\forall A \in \mathcal{M}_n(\mathbb{C})$, $\forall \varepsilon \in \mathbb{R}^+$, il existe une norme matricielle induite $\|\cdot\|_*$ telle que :

$$\rho(A) \leq \|A\|_* \leq \rho(A) + \varepsilon$$

Convergence

Revenons sur la convergence de la suite :

$$\begin{cases} x_0 & \text{donné} \\ x_{k+1} & = \quad Cx_k + d \end{cases}$$

Théorème 1

$\forall C \in \mathcal{M}_n(\mathbb{C})$, s'il existe une norme matricielle induite $\|\cdot\|$ telle que

$$\|C\| \quad < \quad 1$$

alors

1. *L'équation $x = Cx + d$ admet une solution unique x^*.*
2. *La suite $x_k \to x^*$ quelle que soit x_0.*

Preuve

Existence de la solution :

$$\rho(C) \quad \leq \quad \|C\| \quad < \quad 1$$

Donc les valeurs propres λ de C sont telles que $|\lambda| < 1$.

Cela signifie que la matrice $I - C$ est inversible

Donc il existe une solution unique à l'équation

$$x \quad = \quad Cx + d$$

On appelle x^* cette solution

Preuve (suite)

Convergence :

Soit $e_k = x_k - x^*$ On peut déduire une relation entre e_k et e_{k-1}. En effet

$$\begin{aligned} Ce_{k-1} &= C(x_{k-1} - x^*) \\ &= C(x_{k-1}) - C(x^*) \\ &= C(x_{k-1}) + d - x^* \end{aligned}$$

Par conséquent $e_k = C\, e_{k-1}$ pour $k = 1, 2, \ldots$
Nous avons alors

$$e_k = C^k e_0$$

Soit la norme matricielle induite $\|\cdot\|$ et sa norme vectorielle $\|\cdot\|_v$ telle que $\|C\| < 1$:

$$\|e_k\|_v \leq \|C\|^k \|e_0\|_v$$

Donc $e_k \to 0$ et $x_k \to x^*$

Autres conditions

Connaissant C, comment savoir si la suite va converger ?

Cas général :

Théorème 2

Il y a équivalence entre les propositions suivantes :

- *C est une matrice convergente (i.e. C^k tend vers 0)*
- $\rho(C) < 1$
- *Il existe une norme matricielle induite telle que* $\|C\| < 1$

Méthode de Jacobi

La méthode de Jacobi correspond au découpage

$$A = D - U - L$$

avec

$$D = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{pmatrix}$$

$$-L = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{pmatrix} \qquad -U = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{n1} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Méthode de Jacobi (suite)

La méthode de Jacobi correspond au découpage

$$A = D - U - L$$

avec

$$M = D \quad \text{et} \quad N = L + U$$

L'itération $Mx^{(k+1)} = Nx^{(k)} + b$ s'écrit

$$Dx^{(k+1)} = (L + U)x^{(k)} + b$$

c'est-à-dire

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

Méthode de Jacobi (suite)

À partir d'un vecteur $x^{(0)}$, on construit la suite $(x^{(k)})_{k\geq 0}$ de la manière suivante

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

Cette méthode n'est définie que si tous les a_{ii} sont non nuls

```
function x=Jacobi(A,y,xo,itmax)
n = length(y);
x = xo;
xold = x;
for it=1:itmax
   for i=1:n
      x(i) = (y(i)-A(i,[1:i-1,i+1:n])*xold([1:i-1,i+1:n]))/A(i,i);
   end
   xold=x;
end
```

Méthode de Jacobi (suite)

Carl Gustav Jakob Jacobi
Mathématicien allemand
10 décembre 1804 à Potsdam - 18 février 1851 à Berlin

Méthode de Gauss-Seidel

La méthode de Gauss-Seidel correspond au découpage

$$A = D - U - L$$

avec

$$D = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{pmatrix}$$

$$-L = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{pmatrix} \qquad -U = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{n1} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Méthode de Gauss-Seidel (suite)

La méthode de Jacobi correspond au découpage

$$A = D - U - L$$

avec

$$M = D - L \quad \text{et} \quad N = U$$

L'itération $Mx^{(k+1)} = Nx^{(k)} + b$ s'écrit

$$Dx^{(k+1)} = Lx^{(k+1)} + Ux^{(k)} + b$$

c'est-à-dire

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

Méthode de Gauss-Seidel (suite)

À partir d'un vecteur $x^{(0)}$, on construit la suite $(x^{(k)})_{k\geq 0}$ de la manière suivante

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

Cette méthode n'est définie que si tous les a_{ii} sont non nuls

```
function x=GS(A,y,xo,itmax)
n = length(y);
x = xo;
for it=1:itmax
  for i=1:n
     x(i)=(y(i)-A(i,1:i-1)*x(1:i-1)-A(i,i+1:n)*x(i+1:n))/A(i,i);
  end
end
```

Méthode de Gauss-Seidel (suite)

Johann Carl Friedrich Gauss
Mathématicien allemand
30 avril 1777 à Brunswick - 23 février 1855 à Göttingen

Philipp Ludwig Seidel
Physicien allemand
24 octobre 1821 à Zweibrücken - 13 août 1896 à Munich

Comparaison entre Jacobi et Gauss-Seidel

Jacobi : seules les composantes de $x^{(k)}$ sont utilisées pour calculer les composantes de $x^{(k+1)}$

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

Gauss-Seidel : on utilise dès que possible les nouvelles composantes de $x^{(k+1)}$

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

Méthodes de relaxation SOR (Successive Overrrelation)

Étant donné $\omega \in]0, 2[$, la méthode SOR correspond au découpage

$$A = D - U - L$$

avec

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

Étant donné x_0, on construit la suite $(x^{(k)})_{k \geq 0}$ de la manière suivante

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

Retour sur les méthodes de gradient

On suppose à partir de maintenant que la matrice A est symétrique définie positive ($A = A^T$ et pour tout $x \neq 0, x^T Ax > 0$)

- On a vu en TD qu'un point critique de la fonction

$$f(x) = \frac{1}{2} x^T Ax - b^T x$$

était une solution de l'équation $Ax = b$. En effet $\nabla f(x) = Ax - b$

- On peut donc chercher à résoudre le système $Ax = b$ en résolvant le problème

$$\min_x f(x)$$

Dans la suite, on note $r(x) = b - Ax$ le résidu. On remarque que $r(x) = -\nabla f(x)$

Algorithme du gradient

$x_0 =$ approximation initiale
pour $k = 0, 1, 2, \ldots$
    evaluer $p_k = -\nabla f(x_k) = r_k$
    $x_{k+1} = x_k + \alpha_k p_k$ ou α_k est la solution du probleme $\min_\alpha f(x_k + \alpha p_k)$
fin pour

On peut trouver une expression explicite pour α_k. En effet

$$\begin{aligned} f(x_k + \alpha p_k) &= \frac{1}{2}(x_k + \alpha p_k)^T A(x_k + \alpha p_k) - b^T(x_k + \alpha p_k) \\ &= \frac{1}{2}\alpha^2 p_k^T A p_k + \alpha p_k^T A x_k - \alpha b^T p_k + \text{constante} \end{aligned}$$

Le minimum de f par rapport à α s'obtient pour

$$p_k^T A x_k + \alpha p_k^T A p_k - b^T p_k = 0$$

Algorithme du gradient (suite)

On peut trouver une expression explicite pour α_k. En effet

$$\begin{aligned} f(x_k + \alpha p_k) &= \frac{1}{2}(x_k + \alpha p_k)^T A(x_k + \alpha p_k) - b^T(x_k + \alpha p_k) \\ &= \frac{1}{2}\alpha^2 p_k^T A p_k + \alpha p_k^T A x_k - \alpha b^T p_k + \text{constante} \end{aligned}$$

Le minimum de f par rapport à α s'obtient pour

$$p_k^T A x_k + \alpha p_k^T A p_k - b^T p_k = 0$$

soit

$$\alpha = -\frac{p_k^T(Ax_k - b)}{p_k^T A p_k} = \frac{p_k^T r_k}{p_k^T A p_k}$$

Vitesse de convergence

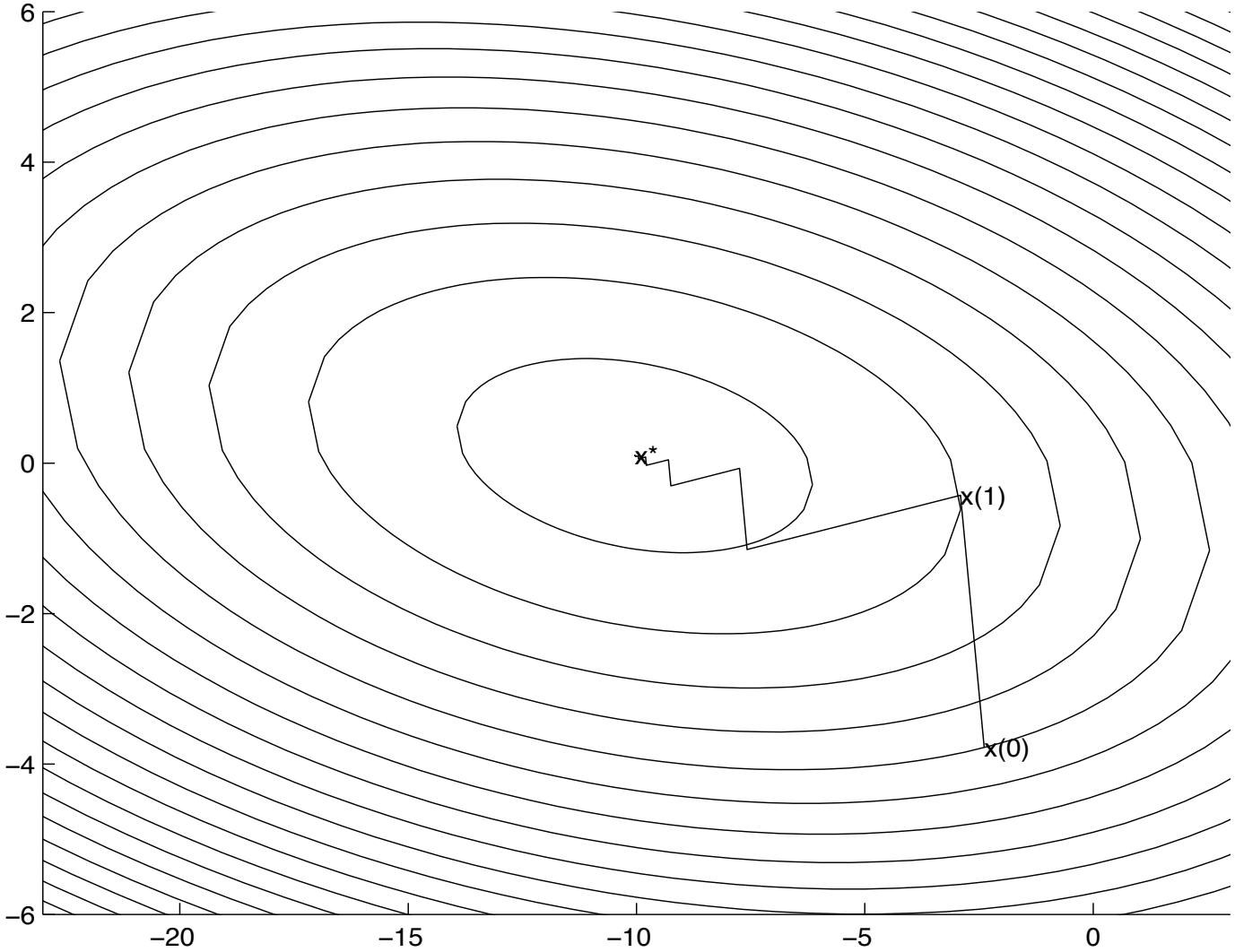
Soit x^* la solution du système $Ax = b$. On note

$$E(x) = \frac{1}{2}(x - x^*)^T A(x - x^*)$$

Cette fonction est minimale quand $x = x^*$ et est un moyen pour mesurer la convergence. On peut montrer que la méthode du gradient a le taux de convergence

$$E(x_k) \leq \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}\right)^{2k} E(x_0)$$

Exemple



Après 20 itérations, l'erreur a été réduite d'un facteur 10^{-5}

Directions A-conjugées

- Comme on l'a vu, l'algorithme du gradient peut être lent
- On va développer un algorithme de gradient modifié qui converge en n étapes au plus (n étant la taille de la matrice)
- L'idée est basée sur l'existence de n vecteurs linéairement indépendants p_k, $k = 0, \ldots, n-1$ qui sont A-conjugués,

$$p_k^T A p_j = 0, \quad k \neq j$$

- Puisqu'il s'agit de n vecteurs linéairement indépendants, il forme une base et donc

$$x^* - x_0 = \sum_{j=0}^{n-1} \alpha_j p_j$$

Directions A-conjuguées (suite)

- On a

$$x^* - x_0 = \sum_{j=0}^{n-1} \alpha_j p_j$$

- En multipliant à gauche par $p_k^T A$, on obtient

$$p_k^T A(x^* - x_0) = p_k^T (b - Ax_0) = p_k^T r_0$$

- En multipliant à droite par $p_k^T A$, on obtient

$$p_k^T A \sum_{j=0}^{n-1} \alpha_j p_j = \alpha_k p_k^T A p_k$$

- Par conséquent

$$\alpha_k = \frac{p_k^T r_0}{p_k^T A p_k}$$

Algorithme du gradient conjugué

On peut donc résoudre l'équation $Ax^* = b$ par l'algorithme suivant :

On choisit $x_0 =$ et p_k, $k = 0, \ldots, n-1$
des directions A-conjuguees
pour $k = 0, 1, \ldots, n-1$
Calculer $\alpha_k = \frac{p_k^T r_0}{p_k^T A p_k}$
Poser $x_{k+1} = x_k + \alpha_k p_k$
fin pour

À la fin, $x_n = x^*$. De plus comme $p_k^T r_0 = p_k^T r_k$ (à cause de la A-conjugaison), on obtient pour α_k la même formule que pour l'algorithme du gradient classique

Il reste néanmoins le problème du calcul de n directions A-conjuguées

Algorithme de Gram-Schmidt

Étant donnés n vecteurs linéairement indépendants v_k, $k = 0, \ldots, n-1$, on peut calculer n vecteurs A-conjugués engendrant le même espace

On choisit $p_0 = v_0$

pour $k = 0, 1, \ldots, n-2$

Calculer $$p_{k+1} = v_{k+1} - \sum_{j=0}^{k} \frac{p_j^T A v_{k+1}}{p_j^T A p_j} p_j$$

fin pour

Algorithme du gradient conjugué

L'algorithme du gradient conjugué interclasse le calcul du nouveau x et du nouveau p. Les v_k utilisés sont en fait les résidus r_k.

On choisit un x_0, $r_0 = b - Ax_0$ et $p_0 = r_0$
pour $k = 0, 1, \ldots, n-1$
    Calculer $\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k A p_k$
    On calcule p_{k+1} par Gram-Schmidt sur r_{k+1} et les p_j, $j \leq k$ afin de rendre p_{k+1} A-conjugues aux directions precedentes
fin pour

Algorithme du gradient conjugué (suite)

Il apparaît que $p_j^T A r_{k+1} = 0$ pour $j < k$. Par conséquent l'étape de Gram-Schmidt se résume à

$$p_{k+1} = r_{k+1} - \frac{p_k^T A r_{k+1}}{p_k^T A p_k} p_k$$

On choisit un x_0, $r_0 = b - Ax_0$ et $p_0 = r_0$

pour $k = 0, 1, \ldots, n-1$

Calculer $\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$ (equivalent a $\frac{r_k^T r_k}{p_k^T A p_k}$)

$$x_{k+1} = x_k + \alpha_k p_k$$
$$r_{k+1} = r_k - \alpha_k A p_k$$

Calculer la nouvelle direction

$$\beta_k = -\frac{p_k^T A r_{k+1}}{p_k^T A p_k} \quad \text{(equivalent a } \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}\text{)}$$
$$p_{k+1} = r_{k+1} + \beta_k p_k$$

fin pour

Algorithme du gradient conjugué (suite)

- Après K étapes, $K \leq n$, on aura $r_K = 0$ et donc $x_K = x^*$.
- On peut montrer que l'on a

$$E(x_k) \leq \left(\frac{1 - \sqrt{\kappa^{-1}}}{1 + \sqrt{\kappa^{-1}}}\right)^{2k} E(x_0)$$

où κ est le conditionnement de la matrice A, $\kappa = \lambda_{\max}/\lambda_{\min}$

Espaces de Krylov et méthodes de Krylov

Définition 3

Étant donnés une matrice $A \in \mathbb{R}^{n\times n}$, un vecteur $r \in \mathbb{R}^n$, on appelle sous-espace de Krylov d'ordre k associé à A et r, noté $\mathcal{K}_k(A, r)$, le sous-espace vectoriel engendré par les vecteurs

$$r, Ar, A^2r, \ldots, A^{k-1}r.$$

Les méthodes de Krylov consistent à chercher l'itéré x_k dans l'espace $x_0 + \mathcal{K}_k(A, r_0)$ où $r_0 = b - Ax_0$.

Méthode de Krylov et gradient conjugué

- On peut montrer que l'algorithme du gradient conjugué revient à chercher $x_k \in x_0 + \mathcal{K}_k(A, r_0)$ vérifiant $r_k = b - Ax_k \perp \mathcal{K}_k(A, r_0)$
- On peut aussi montrer que l'algorithme du gradient conjugué revient à chercher x_k minimisant la fonction $f(x) = \frac{1}{2}x^T Ax - b^T x$ sur l'espace $x_0 + \mathcal{K}_k(A, r_0)$

Méthodes de Krylov

Alexei Nikolaevich Krylov
Mathématicien russe
15 août 1863 à Simbirsk Gubernia - 26 octobre 1945 à Saint Petersburg

Conclusion

- Des algorithmes efficaces en pratique, surtout pour l'algorithme du gradient conjugué
- Des algorithmes surtout utilisés pour des matrices creuses