Introduction

On veut résoudre $Ax = b$

- Méthodes directes: Gauss, Cholesky, etc ...
- Méthodes itératives:
  - Méthodes stationnaires: Jacobi, Gauss-Seidel, SOR, etc...
  - Méthodes non-stationnaires: Gradient conjugué et dérivés

Notation:

$\bar{x}$: solution exacte du système

$r_k = b - Ax_k$: résidu

$K_k(A, v) = Vect(v, Av, \cdots, A^{k-1}v)$: espace de Krylov
On note $K_k = K_k(A, r_0)$

Méthodes de descente

Cas où A est SPD.

La résolution de $Ax = b$ est remplacé par la minimisation de :

$$J(x) = (Ax, x) - 2(b, x) \text{ ou}$$

$$E(x) = (A(x - \bar{x}), x - \bar{x})$$

On cherche des itérés x_k sous la forme $\boxed{x_{k+1} = x_k + \alpha_k p_k}$ où α_k minimise E dans la direction p_k choisie.

Propriétés:

- $\alpha_k = \frac{(r_k, p_k)}{(Ap_k, p_k)}$
- $r_{k+1} = r_k + \alpha_k A p_k$
- $(p_k, r_{k+1}) = 0$

Choix de p_k: $(r_k, p_k) \neq 0$

Le gradient conjugué

On choisit:

$$p_k = r_k + \beta_k p_{k-1}$$

β_k est choisi pour que le facteur de réduction de l'erreur soit le plus petit possible.

$$\beta_k = \frac{\|r_k\|^2}{\|r_{k-1}\|^2}$$

Propriétés

- $(r_k, p_i) = 0 \;\; i \leq k-1$
- $(r_k, r_i) = 0 \;\; i \leq k-1$
- $(p_k, Ap_i) = 0 \;\; i \leq k-1$: relation de conjugaison

Algorithme du Gradient Conjugué

Initialisation

choix de x_0

$p_0 = r_0 = b - Ax_0$

Pour $k = 0, 1 \dots$

$$\alpha_k = \frac{\|r_k\|^2}{(Ap_k, p_k)}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

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

$$\beta_{k+1} = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}$$

$$p_{k+1} = r_{k+1} + \beta_{k+1} p_k$$

Les propriétés remarquables

- On minimise sur K_k et non pas seulement dans la direction p_k
- Convergence théorique en n étapes
- Récurrence à 3 termes: r_k ne dépend que de r_{k-1} et r_{k-2}

Problème: A doit être SPD.

Bi-CG

Idée:

Garder la récurrence à 3 termes

Supprimer l'idée de minimisation

$\Longrightarrow$ adaptation d'un algorithme de calcul des valeurs propres (dont on peut déduire GC) à la résolution de système

Les différences d'avec GC:

- Introduction de A^T
- Double suite de vecteurs à orthogonaliser: r_k et $\bar{r}_k$
  On orthogonalise r_k par rapport à
  $K_k(A^T, \bar{r}_0) = Vect(\bar{r}_0, A^T\bar{r}_0, \cdots, (A^T)^{k-1}\bar{r}_0)$
  et $\bar{r}_k$ par rapport à K_k

Bi-CG

Initialisation

choix de x_0

$r_0 = b - Ax_0$

choix de $\bar{r}_0$

(que l'on peut prendre égal à r_0)

$\rho_{-1} = 1$

$p_{-1} = \bar{p}_{-1} = 0$

Pour $k = 0, 1 \ldots$

$$\rho_k = (\bar{r}_k, r_k)$$

$$\beta_k = \frac{\rho_k}{\rho_{k-1}}$$

$$p_k = r_k + \beta_k p_{k-1}$$

$$\bar{p}_k = \bar{r}_k + \beta_k \bar{p}_{k-1}$$

$$\sigma_k = (\bar{p}_k, Ap_k)$$

$$\alpha_k = \frac{\rho_k}{\sigma_k}$$

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

$$\bar{r}_{k+1} = \bar{r_k} - \alpha_k A^T \bar{p}_k$$

$$x_{k+1} = x_k + \alpha_k p_k$$

CGS

Réécriture de Bi-CG sous forme polynomiale:

$$\left.\begin{aligned} r_k &= P_k(A)r_0 \\ \bar{r}_k &= P_k(A^T)\bar{r}_0 \end{aligned}\right\} \Longrightarrow (r_k, \bar{r}_k) = (P_k^2(A)r_0, \bar{r}_0)$$

$\longrightarrow$ Générer directement $P_k^2 \Longrightarrow r_k = P_k^2(A)r_0$

Avantages

- Supression de $\bar{r}_k$ et de A^T
- Convergence plus rapide

Inconvénients

- Mauvais comportement près de la solution
- Importance du choix de r_0
- Moindre assurance de la convergence

CGS

Initialisation

choix de x_0

$r_0 = b - Ax_0$

choix de $\bar{r}_0$

(que l'on peut prendre égal à r_0)

$q_0 = p_{-1} = 0$

$\rho_{-1} = 1$

Pour $k = 0, 1 \ldots$

$\rho_k = (\bar{r}_0, r_k)$

$\beta_k = \frac{\rho_k}{\rho_{k-1}}$

$u_k = r_k + \beta_k q_k$

$p_k = u_k + \beta_k(q_k + \beta_k p_{k-1})$

$v_k = Ap_k$

$\sigma_k = (\bar{r}_0, v_k)$

$\alpha_k = \frac{\rho_k}{\sigma_k}$

$q_{k+1} = u_n - \alpha_k v_k$

$r_{k-1} = r_k - \alpha_k A(u_k + q_{k+1})$

$x_{k+1} = x_k + \alpha_k(u_k + q_{k+1})$

Bi-CGStab

Idée:

- Repartir de Bi-CG
- Garder une formulation polynomiale (comme CGS)
- Rajouter une idée de minimisation

Bi-CG: $r_k = P_k(A)r_0 \perp K_i(A^T, \bar{r}_0)$ pour $i < k$

$\Longrightarrow P_k(A)r_0 \perp \bar{P}_i(A^T)\bar{r}_0$

$\Longrightarrow r_k = \bar{P}_k(A)P_k(A)r_0$

CGS: $\bar{P}_k = P_k$

Bi-CGStab: $\bar{P}_k = (1 - \omega_1 x) \cdots (1 - \omega_k x)$

avec ω_k qui minimise r_k

CGStab

Initialisation:

on choisit x_0

$r_0 = b - Ax_0$

choix de $\bar{r}_0$

on peut prendre $\bar{r}_0 = r_0$

$\rho_0 = \alpha = \omega_0 = 1$

$v_0 = p_0 = 0$

Pour $k = 1, 2, 3, \ldots$

$\rho_k = (\bar{r}_0, r_{k-1})$

$\beta = \frac{\rho_k}{\rho_{k-1}} \frac{\alpha}{\omega_{k-1}}$

$p_k = r_{k-1} + \beta(p_{k-1} - \omega_{k-1} v_{k-1})$

$v_k = Ap_k$

$\alpha = \frac{\rho_k}{(\bar{r}_0, v_k)}$

$s = r_{k-1} - \alpha v_k$

$t = As$

$\omega_k = \frac{(t,s)}{(t,t)}$

$x_k = x_{k-1} + \alpha p_k + \omega_k s$

Test d'arrêt

$r_k = s - \omega_k t$

GMRES

Idée:

- Supression de la récurrence courte
- Importance de la minimisation

Principe:

- $x_k = x_0 + d_k$
- $d_k \in K_k$
- r_k est minimum sur K_k
  $\Rightarrow \|b - A(x_0 + d_k)\| = \min_{d \in K_k} \|b - A(x_0 + d)\|$

$\longrightarrow$ Construction d'une base orthonormale de K_k

GMRES

Initialisation:

$r_0 = b - Ax_0$

$v_1 = \frac{r_0}{\|r_0\|}$

Algorithme d'Arnoldi

Pour $j = 1, \ldots, k$

$\hat{v}_{j+1} = Av_j$

Pour $i = 1, \ldots, j$

$\beta_{i,j} = (\hat{v}_{j+1}, v_i)$

$\hat{v}_{j+1} = \hat{v}_{j+1} - \sum_{i=1}^{j} \beta_{i,j} v_i$

$\beta_{j+1,j} = \|\hat{v}_{j+1}\|$

$v_{j+1} = \frac{\hat{v}_{j+1}}{\beta_{j+1,j}}$

Construction de x_k:

Calcul de $\min_{d \in K_k} \|b - A(x_0 + d)\|$

Ce qui revient à minimiser

$\| \|r_0\| e_1 - \bar{H}_k y\|$

où $\bar{H}_k$ est la matrice des $\beta_{i,j}$

Cette minimisation se fait par

une décomposition QR

$x_k = x_0 + V_k y_k$

où V_k est la matrice des v_j

Améliorations possibles

Problème de GMRES:

- Construction de la base orthogonale coûteuse
- Nécessité de garder tous les v_i en mémoire

2 solutions:

GMRES(m):

On rend la méthode itérative: après m étapes de l'algorithme précédent, on recommence avec $x_0 = x_m$

DQGMRES:

Algorithme d'Arnoldi local: on n'orthogonalise que sur les m derniers vecteurs calculés (gain du nombre de calculs)

$\longrightarrow$ la convergence n'est plus assurée

Les méthodes de Krylov

De façon plus générale, on construit x_k tel que:

- $x_k \in x_0 + K_k$ où $K_k = Vect(r_0, Ar_0, \cdots, A^{k-1})$
- $b - Ax_k = r_k \perp L_k$

Plusieurs choix de L_k

$L_k = K_k \longrightarrow$ Gradient conjugué, FOM, orthores

$L_k = K_k(A, Ar_0) \longrightarrow$ Erreur minimal, GMRES

$L_k = K_k(A^T, r_0) \longrightarrow$ Bi-CG, CGS, Bi-CGStab, QMR

Les préconditionnements

$Ax = b \implies K^{-1}Ax = K^{-1}b$

Préconditionnement diagonal (ou Jacobi): $K = D$

Préconditionnement SSOR: $K = (\Delta + L)\Delta^{-1}(\Delta + U)$

Un cas particulier: $K = \frac{1}{\omega(2-\omega)}(D + \omega L)D^{-1}(D + \omega L^T)$

Décomposition LU incomplète: $A = LU + R \Rightarrow K = LU$

Plusieurs stratégie pour déterminer R:

- On néglige certains éléments
- On détermine au préalable le remplissage qu'on s'autorise

Préconditionnements polynomiaux :

- Neumann: $K^{-1} = I + (I - A) + (I - A)^2 + \cdots + (I - A)^d$
- Chebyshev: $K^{-1} = P_d(A)$ où les coefficients de P dépendent des valeurs propres de A.