

Préconditionnement des grands systèmes : Pbs linéaires

J-P. CHEHAB¹

¹LAMFA, UMR 6140, Université de Picardie Jules Verne, Amiens, France and SIMPAF Project,
INRIA Nord-Europe (jean-paul.chehab@u-picardie.fr)

Amiens, GTAAAAA, 29 novembre 2010

C'est un sujet encore central en algèbre linéaire numérique

- On dispose de méthodes robustes pour résoudre de grands systèmes non symétriques (GMRES, ...)
- Le préconditionnement est l'approche la plus générale pour en accélérer la résolution
- pas de méthode universelle : problème difficile
- International Conference On Preconditioning Techniques For Scientific And Industrial Applications
 - Preconditioning 1999, The University of Minnesota, Minneapolis, June 10-12 1999.
 - Preconditioning 2001, The Granlibakken Conference Center, Tahoe City, April 29 - May 1, 2001.
 - Preconditioning 2003, Embassy Suites Napa Valley, Napa, October 27-29, 2003.
 - Preconditioning 2005, Emory University, Atlanta, May 19-21, 2005.
 - Preconditioning 2007, Météopole, Toulouse, France, July 9-12, 2007.
 - Preconditioning 2009, Hong-Kong, Hong Kong Baptist University
 - Preconditioning 2011 INRIA Bordeaux, (16-18 mai)
<http://precond11.bordeaux.inria.fr/>

Plan

- 1 Motivation : conditionnement d'un problème
 - Cas général
 - Système linéaire
 - Exemple
- 2 Méthodes itératives classiques
 - Liens avec la méthode de Newton
 - Les méthodes classiques
- 3 Méthodes de projection ou de descente 1
- 4 Méthodes de gradient, de projection
- 5 Construction des préconditionneurs (approche algébrique)
 - Préconditionneurs directs
 - Factorisations incomplètes
 - Avec LU par blocs (complément de Schur)
 - Préconditionneurs inverses
 - Préconditionneurs incrémentaux
- 6 Conclusion (provisoire)

Soit à résoudre $f(x) - a = 0$. Comment varie x lorsque a varie (légèrement) ?

Soient $a' = a + \delta a$ et $\xi' = \xi + \delta \xi$ vérifiant $f(\xi') - a' = 0$. On a

$$\xi' - \xi = f^{-1}(a') - f^{-1}(a) \simeq (a' - a)(f^{-1})'(a)$$

Ainsi

$$\xi' - \xi \simeq \frac{\delta a}{f'(\xi)}$$

Les conditionnements absolus et relatifs du problèmes sont respectivement les nombres

$$K_{abs}(a) = \frac{1}{|f'(\xi)|}, \quad K_{rel} = \frac{a}{|\xi| |f'(\xi)|}$$

Le conditionnement (introduit par J. von Neumann en 1948) d'une matrice en norme d'opérateur p est

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

avec $\max_{\|u\|=1} \|Au\|_p$ où $\|\cdot\|_p$ est la norme ℓ^p vectorielle.

On a toujours $cond_p(Id) = 1$.

Soit $A \in \mathcal{M}_n(\mathbb{R})$ inversible. On considère le problème $Ax = b$. Effets des perturbations (INEVITABLES) des données sur la solution ?

sur papier $Ax = b$ mais sur ordinateur $(A + \delta A)(x + \delta x) = b + \delta b$.

Evaluation de l'erreur relative $\frac{\|\delta x\|}{\|x\|}$

Si $\delta A = 0$ alors $\frac{\|\delta x\|}{\|x\|} \leq \kappa(A) \frac{\|\delta b\|}{\|b\|}$.

Si $\|\delta A\| \leq \frac{1}{\|A^{-1}\|}$ alors

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\|\delta A\|/\|A\|} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right).$$

La quantité $\kappa(A) = \|A\|\|A^{-1}\|$ est le **conditionnement** de A relativement à la norme matricielle $\|\cdot\|$. $\kappa(A) \geq 1$ et A est dite mal conditionnée si $\kappa(A) \gg 1$.

- (Schatzman)

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1.0001 \end{pmatrix} \text{ avec } b = (2, 2)^T, \text{ et } b + \delta b = (2, 2.0001)^T$$

On trouve $x = (2, 0)^T$ et $x + \delta x = (1, 1)^T$, $\kappa_2(A) = 40004.0001$

- (Wilkinson) $A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}$ Avec $b = (32, 23, 33, 31)^t$ et

$$\delta b = (0.1, -0.1, 0.1, -0.1)^t$$

On trouve $x = (1, 1, 1, 1)$ et $x + \delta x = (9.2, -12.6, 4.5, 4.1)^t$,

$$\kappa_2(A) = 2.9841e + 003$$

- (Matrice de Hilbert) $H_{ij}^{(n)} = \frac{1}{i+j-1}, i, j = 1, \dots, n+1, \kappa_2(H^{(n)}) \sim e^{\frac{7n}{2}}$

Préconditionnement pour $Ax = b$: consiste à résoudre un problème équivalent de matrice M ($Au = b \iff Mu = b'$)

- $LAx = Lb$ préconditionneur à gauche
- $ARy = b$ préconditionneur à droite
- $LARy = Lb$ préconditionneur distribué
- $S^T ASy = S^T b$ préconditionneur symétrique

avec le "cahier des charges" suivant

- $Cond(M) \ll Cond(A)$
- M "facile et peu coûteuse" à construire
- Méthodologie robuste (résolution de très gros systèmes)

Soit à résoudre $F(x) = Ax - b = 0$. La méthode de Newton

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

Or $F'(x) = A$, $\forall x \in \mathbb{R}^n$. Donc

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

Convergence en une itération mais pour cela il faut résoudre le problème de départ!!!

Idée : remplacer $F'(x^{(k)}) = A$ par une approximation de cette matrice qui soit facile à inverser et modifier les itérations en

$$x^{(k+1)} = x^{(k)} - \alpha_k B F(x^{(k)})$$

avec α_k , B (ou B^{-1}) connus.

Remarque : On peut remplacer B par $B_k \rightarrow A^{-1}$.

Soit A inversible,

$$A = \begin{pmatrix} \ddots & & & & \\ & \ddots & & & \\ & & D & -F & \\ & -E & & \ddots & \\ & & & & \ddots \end{pmatrix} \quad \text{avec} \quad \begin{array}{ll} D & \text{part. diagonale de } A \\ -E & \text{" triang. inf. stricte de } A \\ -F & \text{" triang. sup. stricte de } A \end{array}$$

- $M = D$: méthode de Jacobi (1846)
(système d'ordre 7)
- $M = D - E$: méthode Gauss-Seidel (Gauss 1823, Seidel 1874)
- $M = I - P^{-1}A$ avec $P = (D - \omega E)D^{-1}(D - \omega F)$: SSOR (Varga 1960)
(résolution d'un système 20 000 inconnues (1960) et à 100 000 inconnues en 1965 (pb phys nucléaire).
- $M = \frac{1}{\alpha} Id$: méthode de Richardson (Richardson 1910)

Elles peuvent toutes se réécrire sous la forme

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

- $B = D$: Jacobi
- $B = (D - E)$: Gauss-Seidel
- $B = \frac{1}{\alpha}Id$: Richardson
- Etc

Bien sûr, on ne calcule jamais l'inverse de la matrice, mais on résout à chaque fois un système linéaire auxiliaire

$$By^{(k)} = r^{(k)} = b - Ax^{(k)} \text{ puis } x^{(k+1)} = x^{(k)} + y^{(k)}.$$

Méthode de projection 1

Soit A SDP. La méthode de Richardson (1910) est obtenue pour $M = \frac{1}{\alpha} Id$

$$u^{(k+1)} = u^{(k)} + \alpha(b - Au^{(k)}) = u^{(k)} + \alpha r^{(k)} \rightarrow r^{(k)} = (Id - \alpha A)^k r^{(0)}$$

La méthode converge SSI $0 < \alpha < \frac{2}{\lambda_{\max}}$ et $\rho(I - \alpha A)$ est minimisé pour $\alpha^* = \frac{2}{\lambda_{\min} + \lambda_{\max}}$.

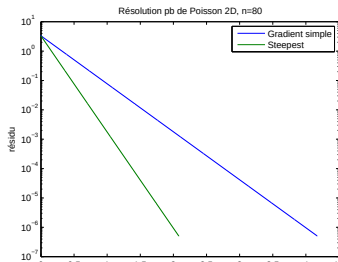
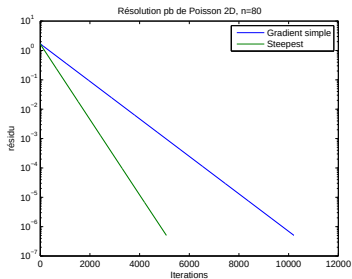
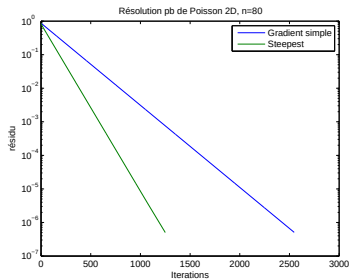
Généralisations pour α variable : $u^{(k+1)} = u^{(k)} + \alpha_k r^{(k)}$ (D. Young, 1954). On a

$$r^{(k+1)} = \left(\prod_{k=0}^k (Id - \alpha_k A) \right) r^{(0)} = P_{k+1}(A) r^{(0)}.$$

La méthode de la plus profonde descente due à L-A Cauchy (1847) consiste à prendre α_k de sorte à minimiser $\|r^{(k+1)}\|$.

Le problème de ces méthodes est qu'elles convergent lentement

-2.cm



Pour la méthode de plus profonde descente, on a

$$\|e_k\|_A \leq \left(\frac{\text{cond}_2(A) - 1}{\text{cond}_2(A) + 1} \right)^k \|e_0\|_A$$

où $e_k = u_k - u$ et $\|e\|_A^2 = \langle e, Ae \rangle$.

Pour accélérer la convergence, on réécrit là encore les itérations sous la forme

$$By^{(k)} = r^{(k)} \text{ puis } u^{(k+1)} = u^{(k)} + \alpha_k y^{(k)}$$

Finalement, on modifie l'algorithme original en insérant une étape de préconditionnement et on obtient l'algorithme préconditionné.

Méthode du gradient conjugué (Stiefel 1952)

Gradient Conjugué

```

poser  $r^0 = b - Au^0$ ,  $p^0 = r^0$ 
pour  $k = 0, \dots$ 
  poser  $z^k = Ap^k$ 
  poser  $\alpha_k = \frac{\langle p^k, r^k \rangle}{\langle p^k, z^k \rangle}$ 
  poser  $u^{k+1} = u^k + \alpha_k p^k$ 
  poser  $r^{k+1} = r^k - \alpha_k z^k$ 
  poser  $\beta_{k+1} = \frac{\langle z^k, r^{k+1} \rangle}{\langle p^k, z^k \rangle}$ 
  poser  $p^{k+1} = r^{k+1} - \beta_{k+1} p^k$ 
    
```

$$\|e^k\|_A \leq \frac{2c^k}{1 + 2c^{2k}} \|e^0\|_A, \text{ avec } c = \frac{\sqrt{K_2(A)} - 1}{\sqrt{K_2(A)} + 1}.$$

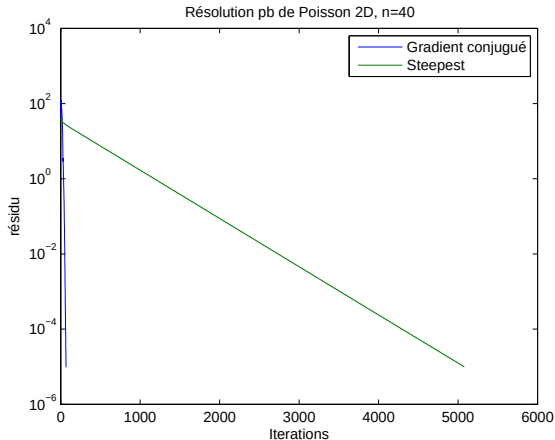


Fig.: Gradient conjugué et plus profonde descente

Gradient Conjugué préconditionné

poser $r^0 = b - Au^0$, $Bz^0 = r^0$, $d_0 = z_0$

pour $k = 0, \dots$

poser $\alpha_k = \frac{\langle z^k, r^k \rangle}{\langle d^k, Ad^k \rangle}$

poser $u^{k+1} = u^k + \alpha_k d^k$

poser $r^{k+1} = r^k - \alpha_k Ad^k$

résoudre $Bz^{k+1} = r^{k+1}$

poser $\beta_{k+1} = \frac{\langle z^k, r^{k+1} \rangle}{\langle z^k, r^k \rangle}$

poser $d^{k+1} = z^{k+1} + \beta_{k+1} d^k$

Remarque si $B = SS^T$ alors CGP $\iff$ CG appliqué à $S^{-1}AS^{-T}$ (exercice)

De la même façon, on obtient les versions préconditionnées des méthodes de Krylov.

Etant donné une matrice A , il s'agit de construire

- $K \simeq A$ (**préconditionneur direct**, facile à inverser)

On s'inspire des factorisations LU ou Cholesky qu'on effectue de manière incomplète

ou

- $K \simeq A^{-1}$ (**préconditionneur inverse**)

On approche l'inverse par minimisation avec contraintes de profil

Dans les deux cas, on cherche à ne garder que les éléments de module au delà d'un certain seuil pour obtenir un préconditionneur creux (la plupart des coefficients sont nuls). On stocke seulement les éléments non nuls (gain de place mémoire). Différentes techniques : stockage diagonal (matrices bandes), stockage morse ...

L'étape de préconditionnement a un coût en termes de temps de calcul et d'espace mémoire. On observe dans de nombreux cas pratiques (EDP) que les coefficients des facteurs ont une grande variation d'amplitude : pour n'avoir à stocker que les coefficients prépondérants Factorisation LU Dans certains cas (Pb EDP)

D'où l'idée de sélectionner les termes suivants les critères suivants

- placés à certains endroits prédéfinis (on remplit L et U dans A)

$$a_{i,j} = \begin{cases} a_{i,j} - \frac{a_{i,k}a_{k,j}}{a_{k,k}} & \text{si } (i,j) \in \mathcal{S} \\ 0 & \text{sinon} \end{cases}$$

- au dessus d'un certain seuil fixé

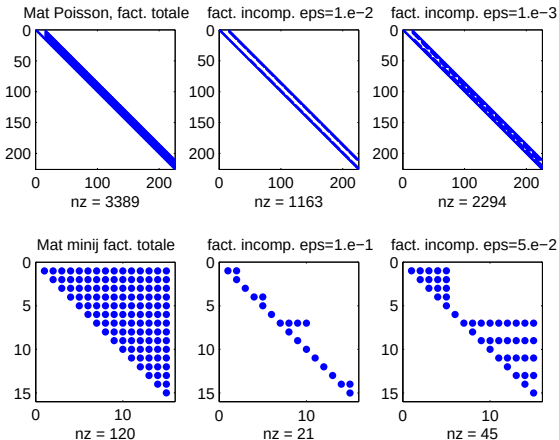
$$a_{i,j} = \begin{cases} a_{i,j} & \text{si } |a_{i,j}| > \epsilon \in \mathcal{S} \\ 0 & \text{sinon} \end{cases}$$

- en contrôlant le remplissage (*fill-in* en anglais)

$$Lev_{i,j} = \begin{cases} 0 & \text{si } a_{i,j} \neq 0 \text{ ou } i = j \\ \infty & \text{sinon} \end{cases}$$

puis $lev_{i,j} = \min(lev_{i,j}, le_{i,k} + lev_{k,j} + 1)$. $IUL(\ell)$ néglige tous les coefficients de niveau supérieur à ℓ .

Les 3 critères peuvent aussi être combinés. On procède de même pour Cholesky



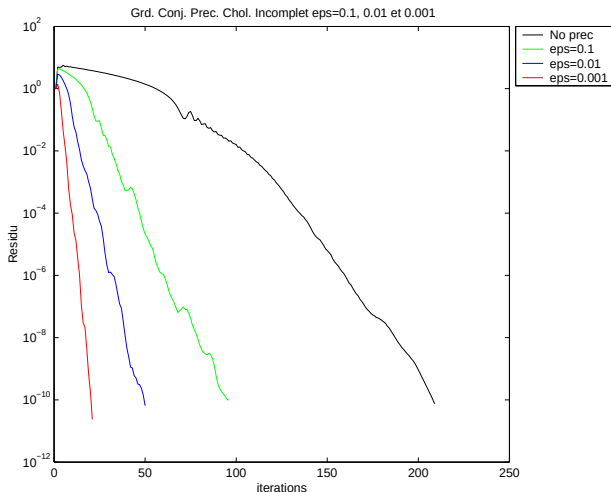


Fig.: Gradient conjugué préconditionné par Cholesky incomplet pour plusieurs valeurs de seuillage

$$-\Delta u + \gamma \left(e^{xy} \frac{\partial u}{\partial x} + e^{-xy} \frac{\partial u}{\partial y} \right) + \alpha u = f$$

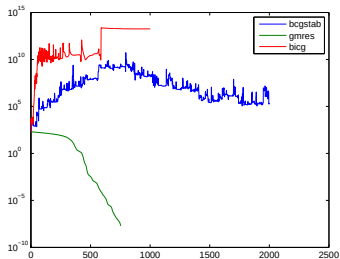
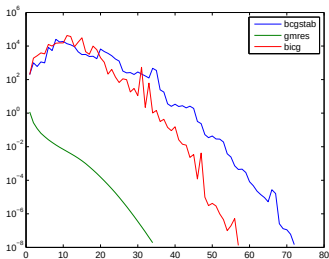


Fig.: Comparaison gmres, bcgstab, bicg, $n=200$, système $40\,000 \times 40\,000$ préconditionné par LU incomplet $\text{seuil}=0.1$

Factorisation LU et compléments de Schur

$$\begin{pmatrix} B & F \\ E & C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

se réécrit comme

$$\begin{pmatrix} L & 0 \\ EU^{-1} & L \end{pmatrix} \times \begin{pmatrix} U & L^{-1}F \\ 0 & S \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

où $S = C_E B^{-1} F$ est le complément de Schur et $B = LU$

Applications

- Factorisation incomplète par approximations récursives du complément de Schur (ARMS)
- Résolution directe ($\neq$ Uzawa) des problèmes de point selle (ex, Stokes)
- Interprétation matricielle des préconditionneurs hiérarchiques

Algebraic Recursive Multilevel Solver (ARMS)

Idée (Saad- Shumochel 2001)

- 1 Effectuer cette décomposition récursivement sur S
- 2 Simplifier en prenant L et U appochées (creuses)
- 3 Approcher S par une matrice creuse et recommencer.

Algorithme

$$A_k = \begin{pmatrix} B_k & F_k \\ E_k & C_k \end{pmatrix} \simeq \begin{pmatrix} L_k & 0 \\ E_k U_k^{-1} & I \end{pmatrix} \times \begin{pmatrix} I & 0 \\ 0 & A_{k+1} \end{pmatrix} \times \begin{pmatrix} U_k & L_k^{-1} F_k \\ 0 & I \end{pmatrix}$$

Préconditionneurs très robustes pour des problèmes de grandes échelles
 (plusieurs dizaines de millions d'inconnues)

Une interpretation des préconditionneurs hiérarchiques

$\mathcal{G}_{2h} \subset \mathcal{G}_h$ ou bien $V_{2h} \subset V_h$. On distingue les coordonnées de la grille grossière (c : coarse) de celles de la grille complémentaire (f : fine)

$$\begin{pmatrix} A_{cc} & A_{cf} \\ A_{fc} & A_{ff} \end{pmatrix} \begin{pmatrix} \vec{x}_c \\ \vec{x}_f \end{pmatrix} = \begin{pmatrix} \vec{b}_c \\ \vec{b}_f \end{pmatrix}. \quad (1)$$

$$A = \begin{pmatrix} I & 0 \\ A_{fc}A_{cc}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{cc} & 0 \\ 0 & \hat{S} \end{pmatrix} \begin{pmatrix} I & A_{cc}^{-1}A_{cf} \\ 0 & I \end{pmatrix}, \quad (2)$$

Avec $\hat{S} = A_{ff} - A_{fc}A_{cc}^{-1}A_{cf}$

On considère une approximation creuse R de $A_{fc}A_{cc}^{-1}$, soit $S = \begin{pmatrix} I & -R^T \\ 0 & I \end{pmatrix}$.

On a $S^T A S \simeq \begin{pmatrix} A_{cc} & 0 \\ 0 & \hat{S} \end{pmatrix}$ et on recommence avec A_{cc}

Pb de Stokes

$$\begin{cases} -\frac{1}{Re} \Delta u + \nabla p = f & \text{dans } \Omega \\ \nabla \cdot u = 0 & \text{dans } \Omega \\ u = 0 & \text{sur } \partial\Omega \end{cases}$$

Après discrétisation stable (cond inf-sup), on obtient le système

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

La méthode d'Uzawa a été abondamment utilisée. Depuis, Whaten, Sylvester, Loghin , ... ont bâti des solveurs basés sur une décomposition LU incomplète de la matrice du système couplé.

Il s'agit de construire directement une approximation de A^{-1} .

- Par factorisation et inversion des facteurs : si $A = LU$, $K_L \sim L$, $K_U \sim U$
alors $A^{-1} \sim K_U^{-1} K_L^{-1}$.
- Minimisation en norme de Frobenius

$$M = \operatorname{argmin}_{X \in \mathcal{S}} \|Id - AX\|_F$$

- optimisation quadratique $\rightarrow$ méthodes de descente
- parallélisable :

$$\|I - AX\|_F^2 = \sum_{j=1}^n \|e_j - Ax_j\|_2^2$$

Enrichissement de préconditionneurs Résolution efficace de suites de problèmes linéaires

$$A_k x_k = b_k, k = 0, \dots$$

où les matrices A_k varient légèrement d'une itération à l'autre.

Une factorisation (LU, Cholesky), même précise, de A_0 peut ne plus être efficace pour A_k , $k \gg 1$. Il faut donc actualiser cette factorisation, sans avoir à tout recalculer !

- Schémas (semi)implicites pour les problèmes d'évolution
- Résolution de systèmes d'équations non linéaires par des méthodes de type Newton (actualisation de la factorisation de la jacobienne)

Avons-nous fait le tour de la question ? Le conditionnement ne permet en fait que d'expliquer pourquoi une méthode converge lentement et le préconditionnement d'y remédier. *A contrario*, une méthode de descente appliquée à une matrice très mal conditionnée peut converger rapidement : cela dépend de la répartition des valeurs propres (mais aussi des composantes spectrale du second membre).

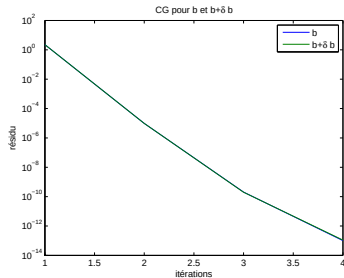
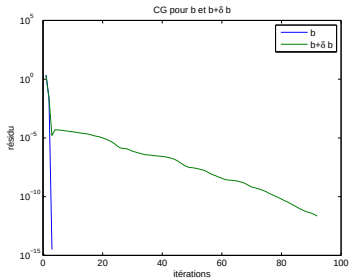


Fig.: CG et PCG (Cholesky incomplet) avec second membre et second membre perturbé, $\frac{\|\delta b\|}{\|b\|} = 8.3067e - 006$

Généralement

- On dispose de méthodes générales efficaces et robustes (GMRES) : le conditionnement est un problème central
- Problème difficile
 - Techniquement : allie méthodes directes, analyse spectrale
 - Conceptuellement : comment jouer directement sur la distribution des valeurs propres de la matrice préconditionnée ? (difficile et coûteux)
- avenir : préconditionneurs variables (GDT du 6 /12/ 2010)

Cas non-linéaire

- Pb pas vraiment réglé, le point de départ reste la méthode de Newton et ses dérivées. Les algorithmes non linéaires ne sont tous préconditionnables (Polac-Ribière)
- Préconditionnements implicites.