Méthodes itératives de Krylov

Gérard MEURANT

Novembre 2014

On veut résoudre

$$Ax = b$$

où A est carrée d'ordre n et non singulière

Lorsque la dimension du système linéaire est petite, il est préférable d'utiliser une méthode directe

Lorsque la matrice est de grande dimension et creuse, il vaut mieux utiliser une méthode itérative

Le choix de la méthode dépend de la nature du problème et du prix que l'on est prêt à payer en termes de CPU et de stockage

Aleksei N. Krylov, 1863–1945

Soit x_0 donné (en général le vecteur nul)

Résidu initial $r_0 = b - Ax_0$

Espace de Krylov $\mathcal{K}_k$ d'ordre k basé sur A et r_0

$$span\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$$

On cherche les itérés

$$x_k \in x_0 + \mathcal{K}_k$$

Si V_k est une matrice dont les colonnes sont les vecteurs de base v_j, $j = 1, \ldots, k$ de l'espace de Krylov

$$x_k = x_0 + V_k y_k$$

Note: $A(x - x_0) = r_0$ et
$x - x_0 = A^{-1}r_0 = (A^{n-1} + \alpha_{n-1}A^{n-2} + \cdots + \alpha_0 I)r_0$

Il existe 2 types de base de méthodes de Krylov :

1) Méthodes de résidus orthogonaux (OR) pour lesquelles

$$(r_k)^T V_k = 0$$

2) Méthodes de résidus minimum (MR) minimisent la norme l_2 du résidu $r^k = b - Ax^k$

$$(r_k)^T AV_k = 0$$

ou résolvent un problème de moindres carrés

Exemples de méthodes OR : CG pour les matrices symétriques définies positive (SPD) et FOM (Full Orthogonal Method) pour les matrices non symétriques

Examples de méthodes MR : MINRES (MINimum RESidual) pour les matrices symétriques indéfinies et GMRES (Generalized Minimum RESidual) pour les matrices non symétriques

Il existe des relations étroites entre les méthodes OR et MR

Les méthodes OR peuvent dégénérer (breakdown) pour les matrices non SPD

Pb : la base “naturelle” est mal conditionnée

$A^i v \rightarrow \mu q_n$, q_n vecteur propre de la valeur propre de plus grand module

Numériquement on perd l'indépendance linéaire des vecteurs de base

Pour des raisons de stabilité on préfère utiliser une base orthogonale de l'espace de Krylov : le procédé d'Arnoldi (1951)

Les vecteurs de base sont calculés récursivement en utilisant Gram-Schmidt en partant de $v_1 = r_0/\|r_0\|$

L'algorithme pour calculer la colonne $j+1$ de V est

$$h_{i,j} = (Av_j, v_i), \quad i = 1, \ldots, j$$

$$\tilde{v}_j = Av_j - \sum_{i=1}^{j} h_{i,j} v_i$$

$$h_{j+1,j} = \|\tilde{v}_j\|$$

$$v_{j+1} = \frac{\tilde{v}_j}{h_{j+1,j}}$$

En général on utilise plutôt Gram–Schmidt modifié

$$w_j = Av_j$$

et pour $i = 1, \ldots, j$

$$h_{i,j} = (w_j, v_i), \quad w_j = w_j - h_{i,j} v_i$$

Les 2 algorithmes sont identiques en arithmétique exacte, mais MGS est plus stable (mais moins parallèle) en arithmétique flottante

On peut également utiliser les transformations de Householder

$$AV_k = V_k H_k + h_{k+1,k} v_{k+1} (e_k)^T$$

H_k est une matrice de Hessenberg supérieure d'éléments $h_{i,j}$

$$AV_k = V_{k+1} \underline{H}_k$$

avec

$$\underline{H}_k = \begin{pmatrix} H_k \\ h_{k+1,k} (e_k)^T \end{pmatrix}$$

On a également

$$H_k = V_k^T A V_k$$

Si l'algorithme ne s'arrête pas avant l'étape n, on a

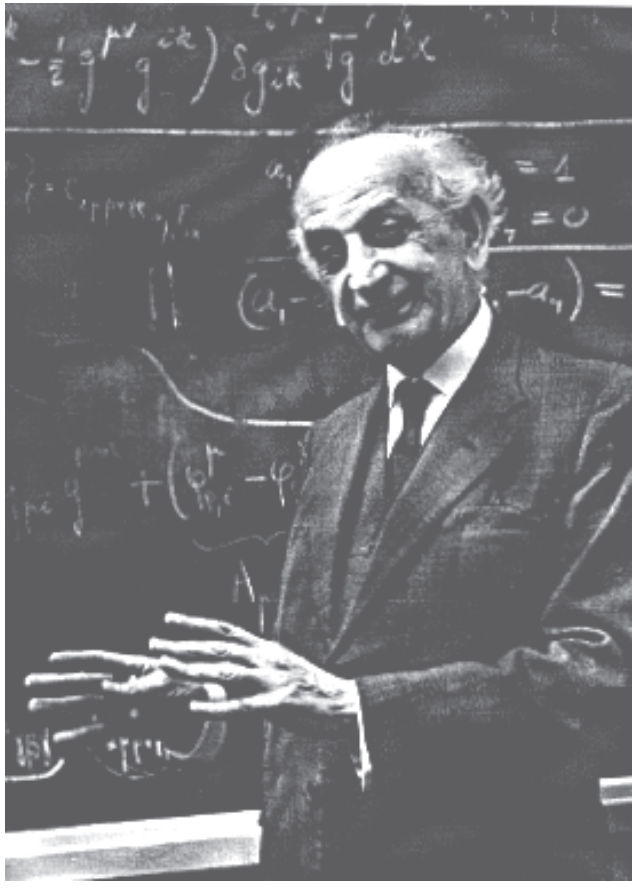
$$AV = VH$$

Exemple : matrice de Hessenberg d'ordre 6

$$H = \begin{pmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \end{pmatrix}$$

Si la matrice A est symétrique, $H_k = T_k$ est tridiagonale

C'est la méthode de Lanczos qui a été proposée avant la méthode d'Arnoldi



Cornelius Lanczos, 1893–1974

$$v_1 = v/\|v\|, \quad \alpha_1 = (v_1)^T A v_1, \quad \tilde{v}_2 = A v_1 - \alpha_1 v_1$$

for $k = 2, 3, \ldots$

$$\eta_k = \|\tilde{v}_k\|$$

$$v_k = \tilde{v}_k/\eta_k$$

$$u_k = A v_k - \eta_k v_{k-1}$$

$$\alpha_k = (v_k)^T u_k$$

$$\tilde{v}_{k+1} = u_k - \alpha_k v_k$$

end

Dans la méthode de Lanczos on n'a besoin d'orthogonaliser que par rapport aux deux vecteurs précédents

$$T_k = \begin{pmatrix} \alpha_1 & \eta_2 & & & \\ \eta_2 & \alpha_2 & \eta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \eta_{k-1} & \alpha_{k-1} & \eta_k \\ & & & \eta_k & \alpha_k \end{pmatrix}$$

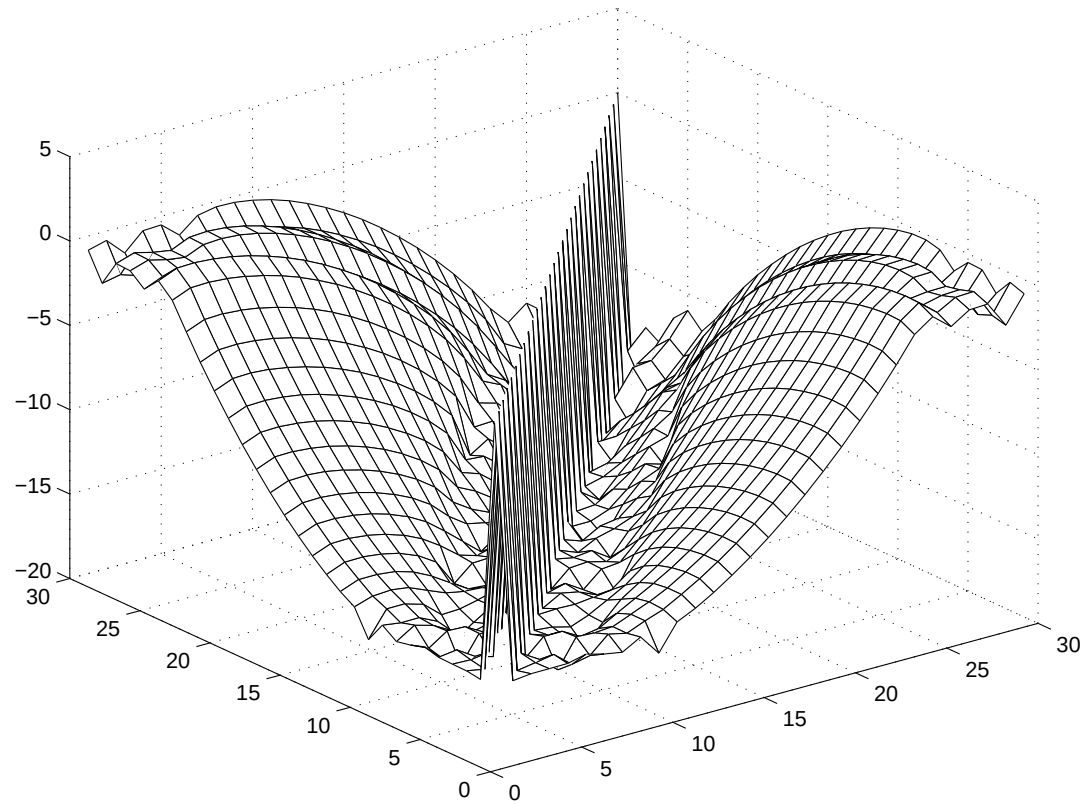
En théorie $V_k V_k^T = I$ et $V_k^T A V_k = T_k$

Si $AV_m = V_m T_m,\ m \leq n$ les valeurs propres de T_m sont des valeurs propres de A

Déjà pour $k \ll n$, les valeurs propres de T_k approchent certaines valeurs propres de A

Lanczos en précision finie

- En théorie $(v_i, v_j) = 0, \ i \neq j$
- En arithmétique flottante c'est faux (erreurs d'arrondi)
- Matrice Strakos30, $\log_{10} |V_{30}^T V_{30}|$



Erreurs d'arrondi

- Convergence d'une valeur propre $\Rightarrow$ perte d'orthogonalité
- $\Rightarrow$ Retard dans la convergence des autres valeurs propres
- $\Rightarrow$ Apparition de copies multiples des valeurs propres déjà trouvées

Lanczos $\Rightarrow$ CG

Supposons A symétrique définie positive (SPD)

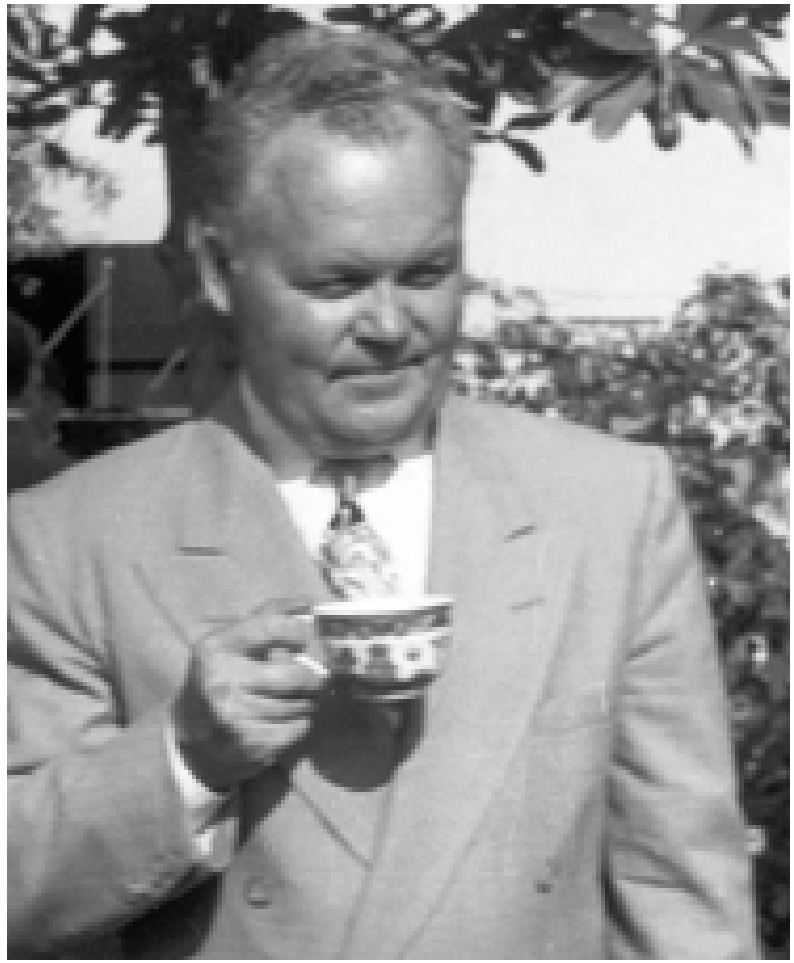
$$AV_k = V_k T_k + G_k, \quad G_k = \begin{pmatrix} 0 & \eta_{k+1} v_{k+1} \end{pmatrix}$$

$r_k = b - Ax_k$, on demande $(r_k)^T V_k = 0$

$$x_k = x_0 + V_k y_k, \quad T_k y_k = \|r_0\| e_1$$

En utilisant la décomposition LU de $T_k = L_k \Omega_k^{-1} U_k$ et un changement de variable $P_k = V_k U_k^{-1}$ on obtient l'algorithme du Gradient Conjugué

CG a été développé indépendamment par Magnus Hestenes et Eduard Stiefel aux US et en Suisse au début des années 50



Magnus Hestenes, 1906–1991

Eduard Stiefel, 1909–1978

Gradient Conjugué (CG)

A SPD, x_0 donné et $r_0 = b - Ax_0$:
for $k = 0, 1, \ldots$ jusqu'à convergence

$$\beta_k = \frac{(r_k, r_k)}{(r_{k-1}, r_{k-1})}, \ \beta_0 = 0$$

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

$$\gamma_k = \frac{(r_k, r_k)}{(Ap_k, p_k)}$$

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

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

Relations avec les coefficients de Lanczos

$$\alpha_k = \frac{1}{\gamma_{k-1}} + \frac{\beta_{k-1}}{\gamma_{k-2}}, \quad \beta_0 = 0, \quad \gamma_{-1} = 1$$

$$\eta_{k+1} = \frac{\sqrt{\beta_k}}{\gamma_{k-1}}$$

Propriétés d'orthogonalité

$$(r_i, r_j) = 0, \quad i \neq j$$

$$(p_i, Ap_j) = 0, \quad i \neq j$$

A–norme de l'erreur

$$\varepsilon_k = x - x_k, \quad A\varepsilon_k = r_k$$

ainsi,

$$\|\varepsilon_k\|_A^2 = (A\varepsilon_k, \varepsilon_k) = (A^{-1}r_k, r_k) = (r_k)^T A^{-1} r_k$$

L'algorithme de Lanczos (et CG) est (sont) étroitement lié aux formules de quadrature de Gauss

$$A = Q\Lambda Q^T$$

Q matrice des vecteurs propres

Soit $y = Q^T r_0/\|r_0\|$, on considère la mesure α constante par morceaux

$$\alpha(\lambda) = \begin{cases} 0 & \text{si } \lambda < a = \lambda_1 \\ \sum_{j=1}^{i} y_j^2 & \text{si } \lambda_i \leq \lambda < \lambda_{i+1} \\ \sum_{j=1}^{n} y_j^2 & \text{si } b = \lambda_n \leq \lambda \end{cases}$$

où les λ_i sont les valeurs propres de A

Les polynômes $p_1(\lambda), p_2(\lambda), \ldots$ qui sont orthonormaux par rapport à α sont les polynômes de Lanczos

$$v_k = p_k(A)v_1$$

$$(v_k, v_\ell) = \int_a^b p_k(\lambda)p_\ell(\lambda_l)\, d\alpha(\lambda)$$

$$\lambda p(\lambda) = T_k p(\lambda) + \eta_{k+1} p_{k+1}(\lambda) e_k$$

où

$$p(\lambda)^T = [p_1(\lambda)\ p_2(\lambda) \cdots\ p_k(\lambda)]$$

Les nœuds t_j de la quadrature de Gauss pour α sont les valeurs propres (valeurs de Ritz) $\theta_j^{(k)}$ de T_k et les poids sont les carrés des premiers éléments des vecteurs propres normalisés

On a

$$(r_0, A^{-1} r_0) = \|r_0\|^2 e_1^T T_n^{-1} e_1 = \int_a^b \frac{1}{\lambda} \, d\alpha(\lambda)$$

L'approximation par quadrature de Gauss de l'intégrale est

$$\|r_0\|^2 e_1^T T_k^{-1} e_1$$

Pour CG on a :

$$\|\varepsilon_k\|_A^2 = \|r_0\|^2[(T_n^{-1}e_1, e_1) - (T_k^{-1}e_1, e_1)]$$

et

$$\|\varepsilon^k\|_A^2 = \|r_0\|^2 \left[\sum_{j=1}^{n} \frac{[(z_{(n)}^j)_1]^2}{\lambda_j} - \sum_{j=1}^{k} \frac{[(z_{(k)}^j)_1]^2}{\theta_j^{(k)}}\right]$$

où $z_{(k)}^j$ est le $j$ème vecteur propre de T_k

Notons qu'à l'itération k on ne connaît pas T_n

On peut aussi montrer

$$\|\varepsilon_k\|_A^2 = \sum_{j=k}^{n-1} \gamma_j \|r_j\|^2$$

formule due à Hestenes et Stiefel (1952)

Cette formule est utile pour obtenir des bornes inférieures de la A-norme de l'erreur

$$\|r_k\|^2 = \sum_{j=1}^{n} \prod_{i=1}^{k} \left(1 - \frac{\lambda_j}{\theta_i^{(k)}}\right)^2 (\bar{r}_0)_j^2$$

$$\|\varepsilon_k\|^2 = \sum_{j=1}^{n} \prod_{i=1}^{k} \left(\frac{1}{\lambda_j} - \frac{1}{\theta_i^{(k)}}\right)^2 (\bar{r}_0)_j^2$$

$$\|\varepsilon_k\|_A^2 = \sum_{j=1}^{n} \prod_{i=1}^{k} \left(\frac{1}{\sqrt{\lambda_j}} - \frac{\sqrt{\lambda_j}}{\theta_i^{(k)}}\right)^2 (\bar{r}_0)_j^2$$

$\bar{r}_0 = Q^T r_0$, Q matrice des vecteurs propres de A

Optimalité de CG

Considérons toutes les méthode itératives qui peuvent s'écrire

$$\overline{x}_{k+1} = \overline{x}_0 + Q_k(A)\overline{r}_0, \quad \overline{x}_0 = x_0, \quad \overline{r}_0 = b - A\overline{x}_0$$

où Q_k est un polynôme de degré k

Parmi toutes ces méthodes, CG est celle qui minimise $\|\varepsilon_k\|_A$ à chaque itération

$$\|\varepsilon_k\|_A^2 \leq \max_{1 \leq i \leq n} (R_k(\lambda_i))^2 \|\varepsilon_0\|_A^2$$

pour tous les polynômes R_k de degré k tels que $R_k(0) = 1$

On choisit

$$R_k(\lambda) = \frac{C_k(\frac{\lambda_1+\lambda_n-2\lambda}{\lambda_n-\lambda_1})}{C_k(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1})}$$

C_k polynôme de Tchebycheff et on obtient

$$\|\varepsilon_k\|_A^2 \leq 4\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2k} \|\varepsilon_0\|_A^2$$

où $\kappa = \frac{\lambda_n}{\lambda_1} = \|A\| \, \|A^{-1}\|$ est le conditionnement de A

Mais nous avons vu que la convergence dépend en fait de la distribution des valeurs propres de A

Dans de nombreux cas cette borne ne reflète pas le comportement de $\|\varepsilon_k\|_A$

Estimations de la A–norme de l'erreur

A l'itération k de CG on ne connaît pas $(T_n^{-1})_{1,1}$

Soit d un entier donné, on obtient une approximation de la A–norme de l'erreur à l'itération $k-d$ par

$$\|\varepsilon_{k-d}\|_A^2 \approx \|r_0\|^2((T_k^{-1})_{(1,1)} - (T_{k-d}^{-1})_{(1,1)})$$

On peut comprendre cette approximation en écrivant

$$\|\varepsilon_{k-d}\|_A^2 - \|\varepsilon_k\|_A^2 = \|r_0\|^2((T_k^{-1})_{(1,1)} - (T_{k-d}^{-1})_{(1,1)})$$

et en supposant que $\|\varepsilon_k\|_A$ est négligeable devant $\|\varepsilon_{k-d}\|_A$

Une autre interprétation est qu'utilisant une quadrature de Gauss avec $k-d$ nœuds à l'itération $k-d$, on se sert d'une formule de quadrature plus précise avec k nœuds pour estimer l'erreur de la première règle

Il faut être prudent pour calculer $(T_k^{-1})_{(1,1)} - (T_{k-d}^{-1})_{(1,1)}$
Soit $t_k = T_k^{-1} e_k$ la dernière colonne de l'inverse de T_k; on utilise la formule de Sherman–Morrison

$$(T_{k+1}^{-1})_{1,1} = (T_k^{-1})_{1,1} + \frac{\eta_{k+1}^2 (t_k t_k^T)_{1,1}}{\alpha_{k+1} - \eta_{k+1}^2 (t_k)_k}$$

et la factorisation de Cholesky de T_k, les éléments diagonaux étant $\delta_1 = \alpha_1$ et

$$\delta_i = \alpha_i - \frac{\eta_i^2}{\delta_{i-1}}, \quad i = 2, \ldots, k$$

On a

$$(t_k)_1 = (-1)^{k-1} \frac{\eta_2 \cdots \eta_k}{\delta_1 \cdots \delta_k}, \quad (t_k)_k = \frac{1}{\delta_k}$$

Soit $b_k = (T_k^{-1})_{1,1}$

$$b_k = b_{k-1} + f_k, \quad f_k = \frac{\eta_k^2 c_{k-1}^2}{\delta_{k-1}(\alpha_k \delta_{k-1} - \eta_k^2)} = \frac{c_k^2}{\delta_k}$$

où

$$c_k = \frac{\eta_2 \cdots \eta_{k-1}}{\delta_1 \cdots \delta_{k-2}} \frac{\eta_k}{\delta_{k-1}} = c_{k-1} \frac{\eta_k}{\delta_{k-1}}$$

Puisque T_k est définie positive, $f_k > 0$

De plus

$$c_k = \frac{\eta_2 \cdots \eta_k}{\delta_1 \cdots \delta_{k-1}} = \frac{\|r_{k-1}\|}{\|r_0\|}$$

et $\gamma_{k-1} = 1/\delta_k$ où γ_{k-1} est un paramètre de CG $(=(r_{k-1}, r_{k-1})/(p_{k-1}, Ap_{k-1}))$

Ainsi

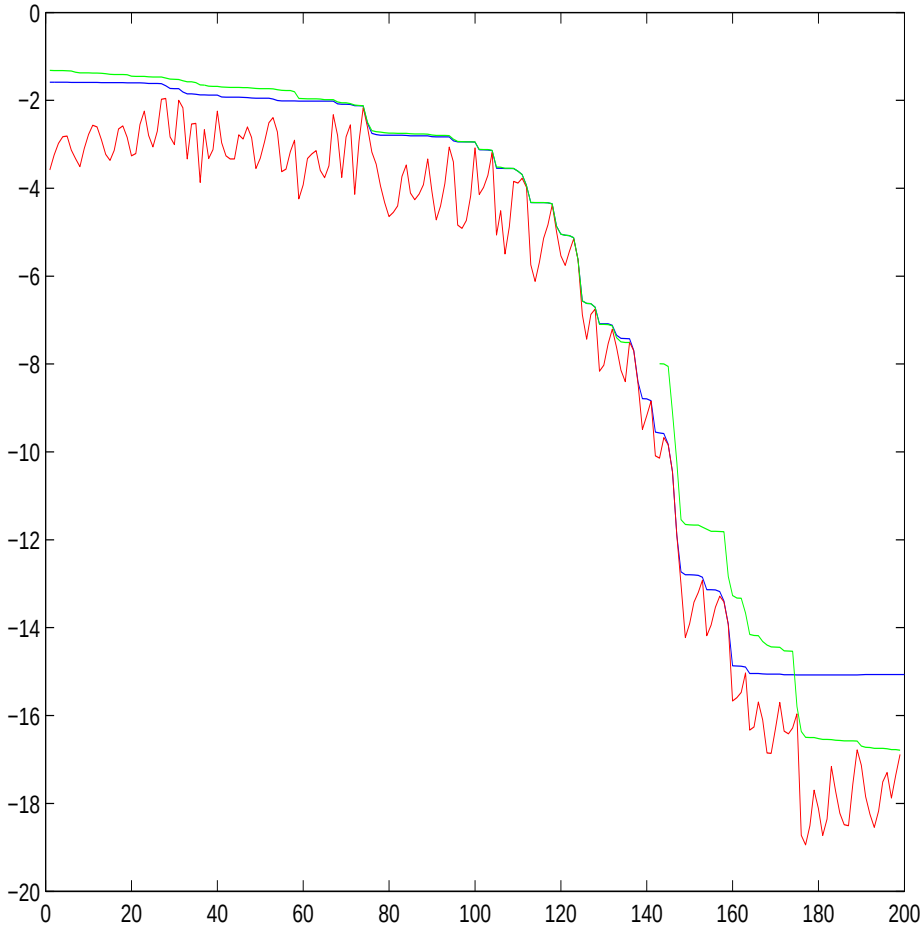
$$\|\varepsilon_{k-d}\|_A^2 \approx \sum_{j=k-d}^{k-1} \gamma_j \|r_j\|^2$$

On obtient une borne inférieure de la A-norme de l'erreur à l'iteration $k - d$

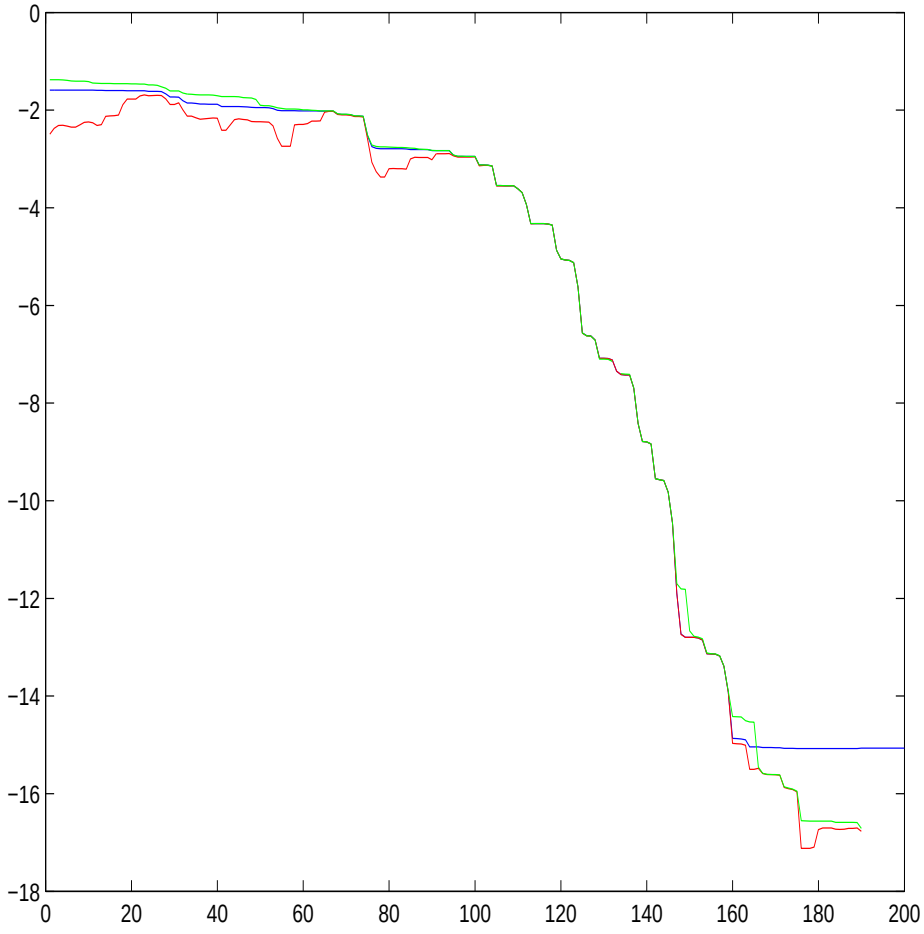
(voir la formule de Hestenes et Stiefel)

D'autre bornes sont obtenues en utilisant les formules de quadrature de Gauss–Radau et Gauss–Lobatto en étendant la matrice de Jacobi pour avoir les nœuds qui sont prescrits

Gauss–Radau donne une borne supérieure, mais il faut connaître une estimation de la plus petite valeur propre de A

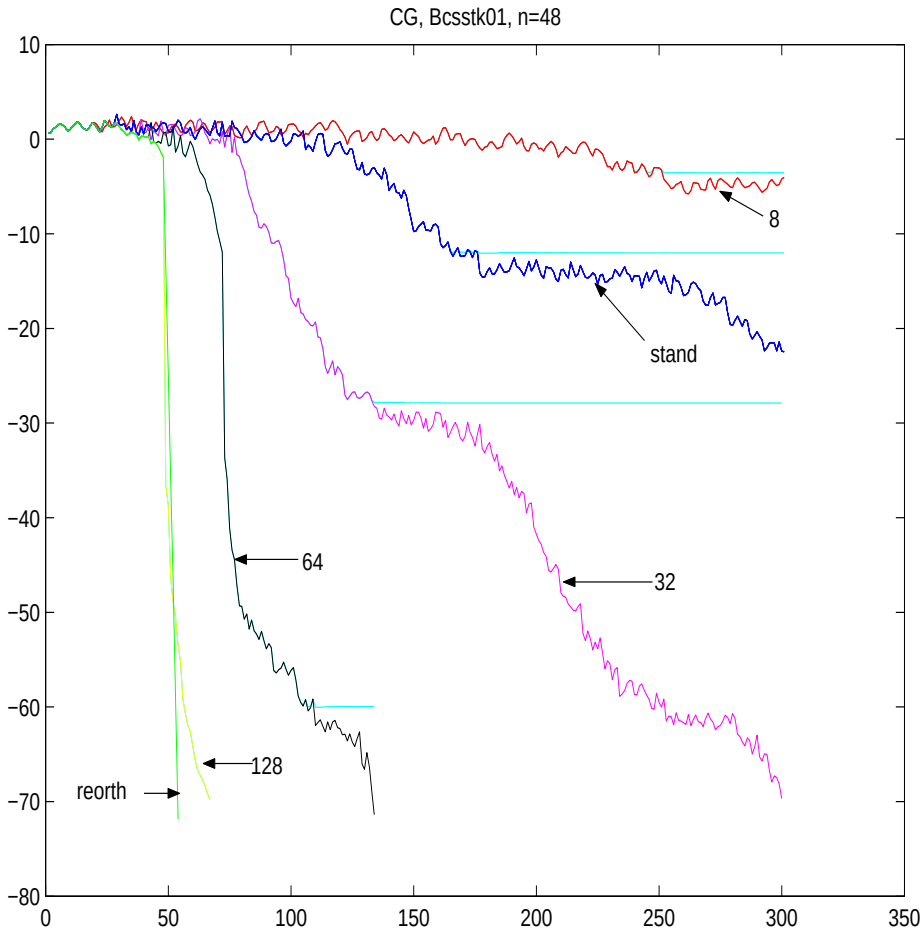


Bcsstk01: $\log_{10}$ de la A–norme de l'erreur (bleu), bornes données par Gauss (rouge), Gauss–Radau (vert), $d = 1$



Bcsstk01: $\log_{10}$ de la A–norme de l'erreur (bleu), bornes données par Gauss (rouge), Gauss–Radau (vert), $d = 10$

CG en précision finie



Bcsstk01: norme du résidu, différentes précisions

CG préconditionné

En pratique on utilise CG avec un préconditionnement M (SPD)
But : améliorer la distribution des valeurs propres
x_0 donné et $r_0 = b - Ax_0$, $Mz_0 = r_0$:
pour $k = 0, 1, \ldots$ jusqu'à convergence

$$\beta_k = \frac{(z_k, r_k)}{(z_{k-1}, r_{k-1})}, \ \beta_0 = 0$$

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

$$\gamma_k = \frac{(z_k, r_k)}{(Ap_k, p_k)}$$

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

$$r_{k+1} = r_k - \gamma_k Ap_k$$

$$Mz_{k+1} = r_{k+1}$$

CG préconditionné

Pour CG préconditionné la formule est

$$\|\varepsilon_k\|_A^2 = (z_0, r_0)((T_n^{-1})_{1,1} - (T_k^{-1})_{1,1})$$

où $Mz_0 = r_0$, M étant le préconditioneur, matrice symétrique définie positive

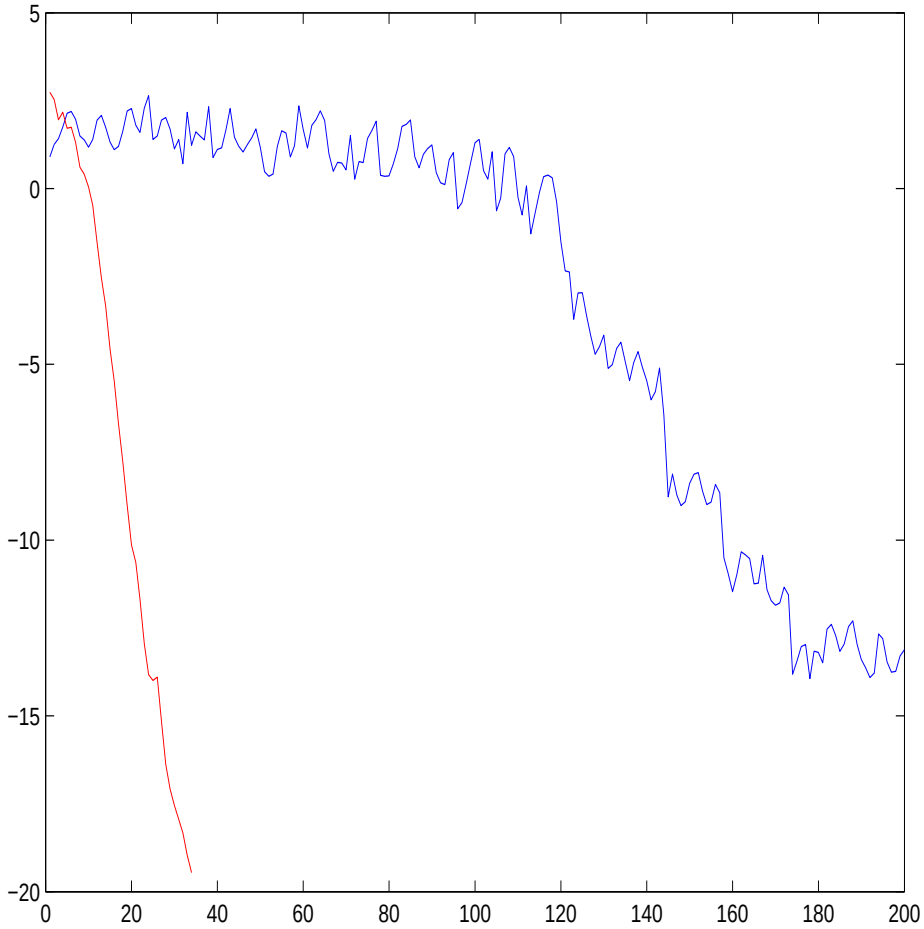
L'estimation donnée par Gauss est

$$\|\varepsilon_{k-d}\|_A^2 \approx \sum_{j=k-d}^{k-1} \gamma_j(z_j, r_j)$$

avec

$$Mz_j = r_j$$

Exemple de préconditionnement



Bcsstk01 préconditionnée: norme du résidu, sans (bleu) et IC (rouge)

Critères d'arrêt

La plupart des codes arrêtent les itérations avec

$$\frac{\|r_k\|}{\|r_0\|} \leq \epsilon$$

ou

$$\frac{\|r_k\|}{\|b\|} \leq \epsilon$$

C'est identique si $x_0 = 0$

Autre possibilité : utiliser l'erreur inverse

$$\epsilon_B^k = \frac{\|r_k\|}{\|A\| \, \|x_k\| + \|b\|} \leq \epsilon$$

En général on ne connaît pas $\|A\|$ mais on peut utiliser $\|A\|_\infty$

Pb : la norme du résidu peut être très différente de la norme de l'erreur $\varepsilon_k = x - x_k$
On a $A\varepsilon_k = r_k$ et

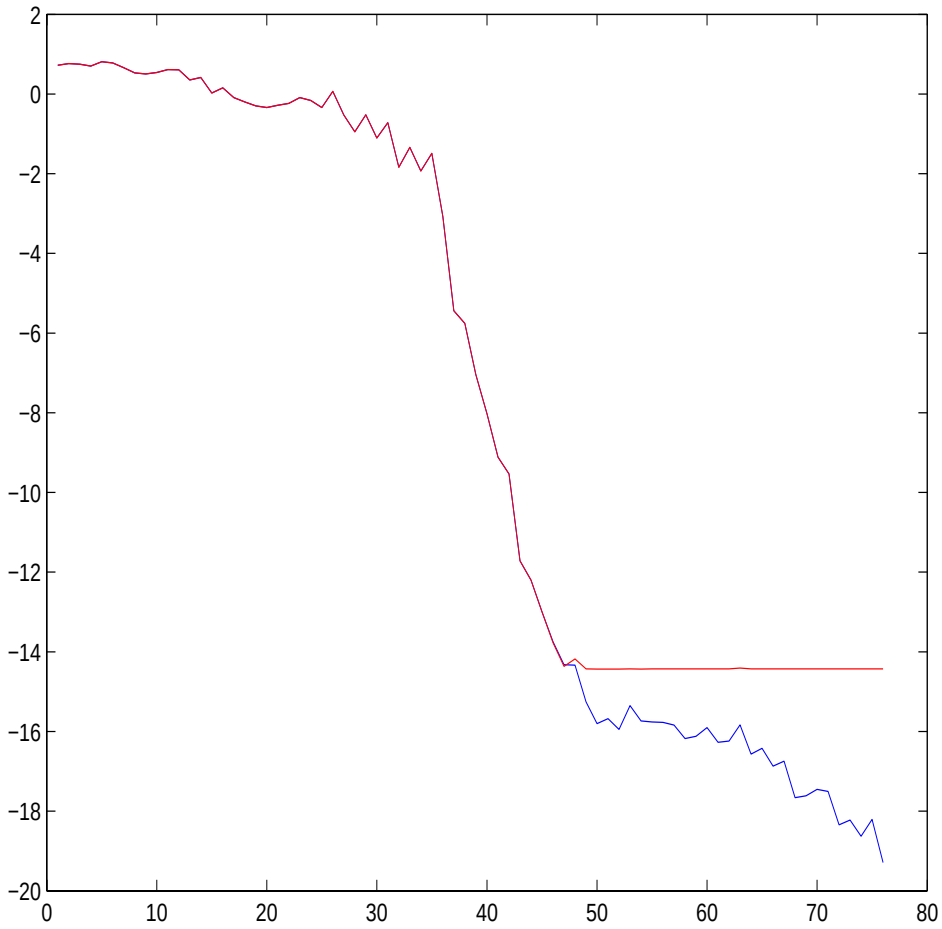
$$\frac{1}{\|A\|}\|\varepsilon_k\| \leq \|r_k\| \leq \|A\| \, \|r_k\|$$

Dans CG on peut estimer $\|\varepsilon_{k-d}\|$ avec d "petit" et utiliser l'estimation comme critère d'arrêt

Autre pb : la norme du résidu itératif r_k peut être différente de celle de $b - Ax_k$
$\|r_k\| \longrightarrow 0$ mais $\|b - Ax_k\|$ est bornée inférieurement

Exemple 1

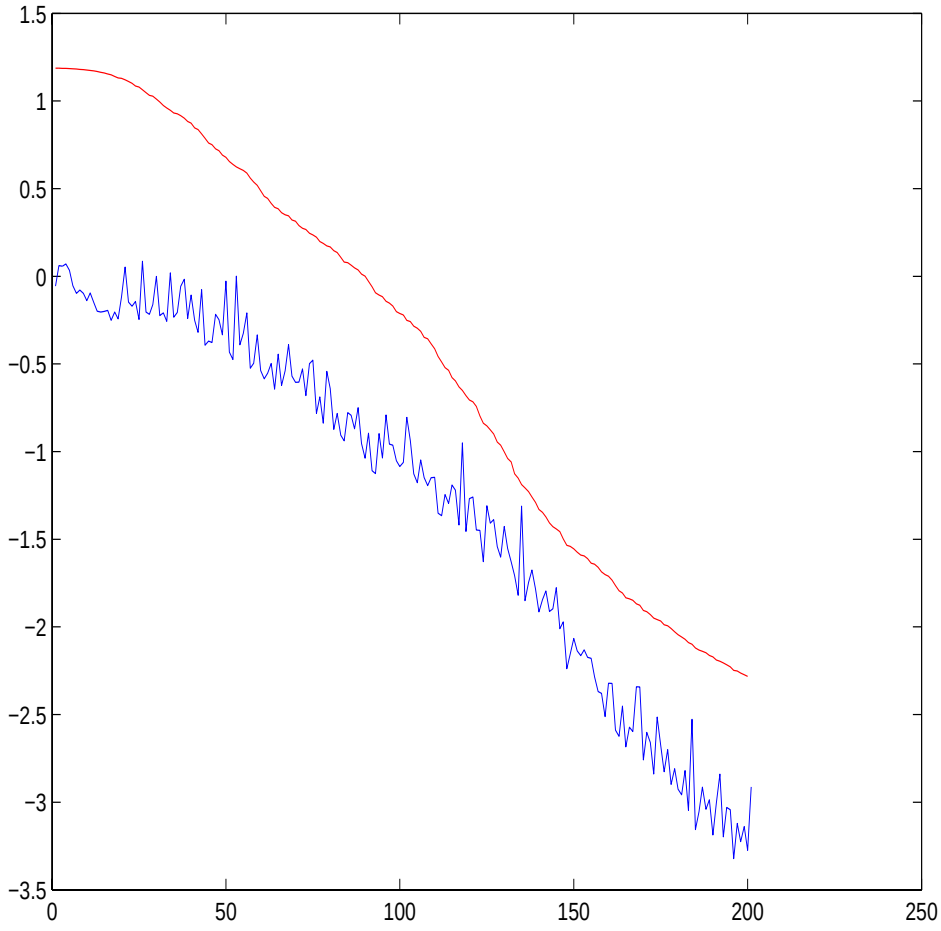
Strakos, n = 30



Strakos30: $\|r_k\|$ (bleu) et $\|b - Ax_k\|$ (rouge)

Exemple 2

Sin-sin, n = 900



Sin-sin: $\|r_k\|$ (bleu) et $\|\varepsilon_k\|$ (rouge)