



TECHNIQUES
DE L'INGÉNIEUR

Réf. : **AF488 V1**

Date de publication :
10 avril 2007

Méthodes de Krylov pour la résolution des systèmes linéaires

Cet article est issu de : **Sciences fondamentales | Mathématiques**

par **Gérard MEURANT**

Résumé Les méthodes de Krylov pour la résolution des systèmes linéaires sont généralement utilisées en liaison avec un préconditionneur qui permet d'accélérer la convergence. Elles ne requièrent que des multiplications de la matrice du système par un vecteur, des produits scalaires et des additions de vecteurs. Cet article propose une explication des méthodes et de leur principal but. Les méthodes de Krylov sont ensuite analysées en profondeur : construction de la base, méthodes GMRES et FOM, gradient conjugué, méthodes BiCG et BiCGstab ou méthode QMR. Un exemple de méthodes clôture l'article.

Abstract Krylov methods for solving linear systems are generally used with a preconditioner which accelerates the convergence. They only require matrix multiplication by a vector, scalar products and vector additions. This article explains these methods and their different aims. An in-depth analysis of the Krylov methods is then provided: construction of the basis, GMRES and FOM methods, conjugate gradient, BiCG and BiCGstab or QMR methods. An example of methods concludes this article.

Pour toute question :
Service Relation clientèle
Techniques de l'Ingénieur
Immeuble Pleyad 1
39, boulevard Ornano
93288 Saint-Denis Cedex

Par mail :
infos.clients@teching.com
Par téléphone :
00 33 (0)1 53 35 20 20

Document téléchargé le : **04/05/2017**

Pour le compte : **7200043660 - centralesupelec // 138.195.79.110**

© Techniques de l'Ingénieur | tous droits réservés

Méthodes de Krylov pour la résolution des systèmes linéaires

par Gérard MEURANT
CEA/DIF (Bruyères le Chatel)

1.	But des méthodes	AF 488 - 2
2.	Méthodes de Krylov	— 2
2.1	Construction de la base.....	— 2
2.2	Méthodes GMRES et FOM.....	— 3
2.3	Gradient conjugué.....	— 4
2.4	Méthodes BiCG et BiCGstab.....	— 4
2.5	Méthode QMR.....	— 5
3.	Exemple	— 6
	Références bibliographiques	— 7

Ce dossier expose l'état de l'art pour résoudre des grands systèmes linéaires creux avec **des méthodes itératives de Krylov**. Ces méthodes ne requièrent que des multiplications de la matrice du système par un vecteur, des produits scalaires et des additions de vecteurs. Elles sont généralement utilisées en liaison avec un préconditionneur qui permet d'accélérer la convergence.

Parution : avril 2007 - Ce document a été délivré pour le compte de 7200043660 - centralesupelec // 138.195.79.110

1. But des méthodes

On s'intéresse à la résolution de systèmes linéaires $Ax = b$ avec des matrices A non singulières creuses (c'est-à-dire comportant beaucoup de zéros) de grande dimension. On doit résoudre de tels systèmes, par exemple, lorsque l'on discrétise des (systèmes d') équations aux dérivées partielles par des méthodes de différences finies ou d'éléments finis. On obtient des systèmes linéaires dont la matrice comporte peu d'éléments non nuls par ligne, pour lesquels il est utile d'utiliser des techniques particulières qui permettent de ne stocker que les éléments non nuls de la matrice et des pointeurs qui permettent de retrouver facilement les indices de ligne et de colonne des éléments et de parcourir les lignes et/ou les colonnes (cf. [7] [9]).

On considère ici des méthodes itératives modernes pour résoudre des systèmes $Ax = b$ où la matrice A (d'ordre n) et le second membre sont donnés. Les matrices considérées possèdent des éléments réels mais la plupart des méthodes exposées s'étendent facilement à des matrices ayant des éléments complexes.

En partant d'un vecteur initial donné x^0 , on construit une suite de vecteurs x^k , en faisant en sorte que x^k converge vers la solution x du système linéaire lorsque $k \rightarrow \infty$. La plupart des méthodes en usage aujourd'hui appartiennent à une classe appelée « **méthodes de Krylov** ». Elles sont basées sur des principes d'orthogonalisation ou de minimisation. De nombreuses méthodes ont été proposées durant les vingt-cinq dernières années. La plupart ne sont que des variantes des méthodes de base. Dans la suite, nous allons décrire les méthodes les plus utilisées. Pour être réellement efficaces, ces méthodes sont employées en liaison avec un préconditionnement. La méthode itérative choisie est alors appliquée à un système équivalent :

$$M^{-1}Ax = M^{-1}b \text{ (ou bien } AM^{-1}y = b \text{ avec } y = Mx)$$

où la matrice non singulière M est le préconditionnement considéré.

La matrice M est choisie (le plus souvent de manière heuristique) de telle sorte que la matrice $M^{-1}A$ ait de meilleures propriétés que la matrice A en ce qui concerne la convergence de la méthode.

On souhaite que $M^{-1}A$ soit « proche » de l'identité ou bien possède une distribution des valeurs propres favorable à la convergence.

Une difficulté supplémentaire qui se présente aujourd'hui lorsque l'on veut résoudre de très grands systèmes (de plusieurs dizaines ou centaines de millions d'inconnues) est que, non seulement il faut avoir une bonne vitesse de convergence mais, de plus, il faut que la méthode choisie soit utilisable efficacement sur des calculateurs parallèles comportant plusieurs centaines ou milliers de processeurs. Malheureusement, les méthodes les plus efficaces ne sont, le plus souvent, que peu naturellement parallélisables. Il faut donc modifier ces méthodes pour les rendre plus parallèles en essayant de ne pas détruire leurs bonnes propriétés de convergence.

2. Méthodes de Krylov

2.1 Construction de la base

Toutes ces méthodes démarrent d'un vecteur initial donné x^0 ayant n composantes. En général, on choisit $x^0 = 0$ ou bien un vecteur ayant des composantes aléatoires. Le résidu initial r^0 est défini par $r^0 = b - Ax^0$. L'espace de Krylov d'ordre k construit sur A et r^0 et noté $\mathcal{K}_k(A, r^0)$ est défini comme l'espace engendré par les vecteurs :

$$r^0, Ar^0, \dots, A^{k-1}r^0$$

Aleksei N. Krylov était un mathématicien et ingénieur russe qui a vécu de 1863 à 1945. Les espaces qu'il a utilisés pour des calculs de valeurs propres sont associés à son nom, mais les méthodes « dites de Krylov », en particulier pour les systèmes non symétriques, ont été découvertes beaucoup plus tard.

On cherche les itérés x^k dans l'espace $x^0 + \mathcal{K}_k(A, r^0)$.

Si V_k est une matrice dont les colonnes sont des vecteurs v^j ($j = 1, \dots, k$) qui constituent une base de l'espace de Krylov \mathcal{K}_k , on peut écrire :

$$x^k = x^0 + V_k z^k \quad (1)$$

et le problème se réduit à construire les vecteurs v^j de façon incrémentale et à savoir comment calculer le vecteur z^k comportant k composantes définissant la combinaison linéaire des vecteurs de base.

Il existe deux types de méthodes de Krylov.

■ Le premier type est composé de **méthodes de résidu minimal**. On parle de méthodes de type MR (*Minimum Residual*) ; elles minimisent la norme ℓ_2 du résidu $r^k = b - Ax^k$. En utilisant la définition (1)

de x^k , on est conduit à la condition :

$$(r^k)^T A V_k = 0$$

ce qui est équivalent, comme on le verra, à résoudre un problème de moindres carrés. Des exemples de ces méthodes sont MINRES (*MINimum RESidual*) pour les matrices symétriques indéfinies (ayant des valeurs propres positives et négatives) et GMRES (*Generalized MINimum RESidual*) pour les matrices quelconques.

■ Pour le deuxième type de méthodes, on parle de **méthodes OR** (*Orthogonal Residual*) dans lesquelles on demande que les résidus soient orthogonaux à l'espace de Krylov, ce qui donne la condition :

$$(r^k)^T V_k = 0$$

Des exemples de ces méthodes sont le **gradient conjugué** (CG pour *Conjugate Gradient*) pour les matrices symétriques définies positives (ayant toutes leurs valeurs propres strictement positives) et **FOM** (*Full Orthogonal Method*) pour les matrices quelconques. Les conditions précédentes vont permettre de déterminer le vecteur des coordonnées z^k à chaque itération.

Il existe plusieurs ouvrages récents décrivant et expliquant le fonctionnement des méthodes de Krylov pour la résolution des systèmes linéaires (B. Fischer [2], A. Greenbaum [5], G. Meurant [7], Y. Saad [9] et H.A. Van der Vorst [12]).

Pour développer des méthodes de Krylov, la première question qui se pose est de savoir comment choisir la base de l'espace $\mathcal{K}_k(A, v)$ pour un vecteur v donné. Idéalement, tant que k est inférieur ou égal à l'ordre du polynôme minimal de v par rapport à A , le sous-espace $\mathcal{K}_k(A, v)$ est de dimension k et les vecteurs $A^j v$ ($j = 0, \dots, k-1$) sont linéairement indépendants. On pourrait donc choisir ces vecteurs comme base de l'espace de Krylov.

Cependant, même si ce choix est mathématiquement licite, ce n'est pas celui qui est fait dans la pratique. En effet, lorsque j croît, les vecteurs $A^j v$ tendent à devenir parallèles au vecteur propre correspondant à la valeur propre de module maximal. Numériquement, ces vecteurs peuvent devenir dépendants avant que l'on atteigne le degré du polynôme minimal. On choisit donc, pour des raisons de stabilité, de construire de façon incrémentale une base orthogonale de l'espace de Krylov. Cela est fait par le procédé dit d'Arnoldi qui n'est pas autre chose que la méthode de Gram-Schmidt (cf. [4]) appliquée à l'espace de Krylov. Lorsque la matrice est symétrique, on obtient l'algorithme de Lanczos.

Les vecteurs de base orthonormaux v^j de $\mathcal{H}(A, v)$ sont construits de la façon suivante :

$$v^1 = \frac{v}{\|v\|}$$

L'algorithme pour calculer la colonne $j+1$ de V_k et donc le vecteur v^{j+1} est :

$$h_{ij} = (Av^j, v^i), \quad i = 1, \dots, j$$

$$\bar{v}^j = Av^j - \sum_{i=1}^j h_{ij} v^i$$

$$h_{j+1,j} = \|\bar{v}^j\|, \quad v^{j+1} = \frac{\bar{v}^j}{h_{j+1,j}}$$

Généralement, pour des raisons de stabilité, on préfère utiliser l'algorithme de Gram-Schmidt modifié. Cela revient à calculer h_{ij} et v^{j+1} comme :

$$w^j = Av^j$$

et, pour $i = 1, \dots, j$:

$$h_{ij} = (w^j, v^i), \quad w^j = w^j - h_{ij} v^i$$

avec

$$\bar{v}^j = w^j$$

Les relations précédentes peuvent s'écrire de façon matricielle jusqu'à l'étape $k+1$:

$$AV_k = V_k H_k + h_{k+1,k} v^{k+1} (e^k)^T \quad (2)$$

avec H_k matrice de Hessenberg supérieure (ce qui signifie que les éléments de la partie triangulaire supérieure sont non nuls ainsi que la sous-diagonale adjacente à la diagonale principale) d'éléments h_{ij} ,

e^k k ème colonne de la matrice identité.

L'équation matricielle peut s'écrire également :

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

avec

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

une matrice rectangulaire de dimension $(k+1) \times k$.

En multipliant la relation (2) par V_k^T , on obtient :

$$V_k^T AV_k = H_k$$

Remarquons que, lorsque A est une matrice symétrique, la matrice H_k est tridiagonale (seules les trois diagonales principales sont non nulles) et on la note par T_k . Cela entraîne que les vecteurs de base v^j peuvent être calculés par une récurrence à trois termes peu coûteuse en termes d'opérations et de stockage.

2.2 Méthodes GMRES et FOM

■ Dans la méthode GMRES [10], à chaque itération, on minimise la norme ℓ_2 du résidu $r^k = b - Ax^k$.

En considérant que $r^0 = \|r^0\| V_{k+1} e^1$, on peut écrire :

$$\begin{aligned} \|r^k\| &= \|b - Ax^k\| \\ &= \|r^0 - AV_k z^k\| \\ &= \|r^0\| \|V_{k+1} e^1 - V_{k+1} \tilde{H}_k z^k\| \\ &= \|r^0\| \|e^1 - \tilde{H}_k z^k\| \end{aligned}$$

La dernière relation est due au fait que la norme ℓ_2 est invariante pour les transformations orthogonales, représentées ici par la matrice V_{k+1} . Ainsi, le problème de moindres carrés avec A d'ordre n se ramène à un problème avec \tilde{H}_k de dimension $(k+1) \times k$ qui, de plus, possède une structure particulière de Hessenberg. Pour résoudre ce problème de moindres carrés, on calcule de façon incrémentale une factorisation QR de la matrice telle que :

$$Q_k^T \tilde{H}_k = R_k$$

avec Q_k orthogonale et R_k triangulaire supérieure.

Le problème de minimiser $\|r^0\| \|e^1 - \tilde{H}_k z^k\|$ se réduit alors à :

$$\|r^0\| \|Q_k^T e^1 - R_k z^k\|$$

et la solution est obtenue en résolvant un système triangulaire avec la matrice R_k .

On utilise des rotations de Givens pour annihiler les éléments sous-diagonaux de \tilde{H}_k . Supposons que l'on ait déjà traité les $k-1$ premières colonnes dans les étapes précédentes. On a alors une matrice ayant la structure suivante :

$$\begin{pmatrix} x & x & \dots & x \\ & \ddots & & \vdots \\ & & \ddots & \\ & & & x & x \\ & & & 0 & r \\ & & & 0 & h \end{pmatrix}$$

Pour annihiler l'élément en position $(k+1, k)$, on multiplie par une matrice (orthogonale) de rotation :

$$\begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & c & -s \\ & & & s & c \end{pmatrix}$$

$$c = \frac{r}{\sqrt{r^2 + h^2}}, \quad s = -\frac{h}{\sqrt{r^2 + h^2}}$$

On obtient ainsi :

$$Q_k^T \tilde{H}_k = R_k$$

avec Q_k^T formé du produit de toutes les matrices de rotation.

Si on initialise $f = \|r^0\| e^1$, à l'itération k après avoir construit le nouveau vecteur de base v^{k+1} , on applique les rotations précédentes au vecteur $(h_{1,k}, \dots, h_{k+1,k})^T$, on calcule la rotation $R_{k+1,k}$ qui élimine $h_{k+1,k}$ et on obtient le nouveau second membre $f = R_{k+1,k} f$.

La solution z^k du problème de moindres carrés est obtenue en résolvant le système linéaire de matrice R_k et de second membre le vecteur des k premières composantes de f . En utilisant la relation (2), on voit que la norme du résidu est égale à $\|h_{k+1,k} z^k\|$. On peut donc tester la norme du résidu sans avoir à le calculer par la formule $b - Ax^k$.

La mise en œuvre de cette méthode n'étant pas tout à fait triviale, on donne ci-dessous une implantation de GMRES(nd) sous forme de pseudo-code dans lequel nd est le nombre de vecteurs de

base que l'on accepte de stocker. Bien entendu, il faut ajouter un test d'arrêt à ce code pour stopper les itérations lorsque le résidu est suffisamment petit.

```
nd=k
% produit matrice vecteur
avk=a*v(:,k)
w=avk
% Gram-Schmidt modifié
for l=1:k
    hl=v(:,l)'*w
    h(l,k)=hl
    w=w-hl*v(:,l)
end %for
vk=w
hk=sqrt(vk'*vk)
% normalisation
vk1=vk/hk
% nouveau vecteur de la base
v(:,k+1)=vk1
hk1=hk
% On applique les rotations précédentes à la dernière
colonne
for l=1:k-1
    h1=h(l,k)
    h2=h(l+1,k)
    h(l,k)=rot(1,l)*h1-rot(2,l)*h2
    h(l+1,k)=rot(2,l)*h1+rot(1,l)*h2
end %for
% On calcule, stocke et applique la rotation
hk=h(k,k)
cs=sqrt(hk^2+hk^2)
rot(1,k)=hk/cs
rot(2,k)=-hk1/cs
h(k,k)=cs
c=rhs(k)/cs
rhs(k)=hk*c
rhs(k+1)=-hk1*c
% Résolution du système triangulaire donnant
% la solution des moindres carrés
z(nd)=rhs(nd)/h(nd,nd)
for l=nd-1:-1:1
    z(l)=(rhs(l)-h(l,l+1:nd)*z(l+1:nd))/h(l,l)
end %for
% Solution à l'itération courante
x=x0+v(:,1:nd)*z(1:nd)
```

■ Dans la méthode OR correspondante qui s'appelle FOM, on impose la relation $V_k^T r^k = 0$. Cela conduit à :

$$\begin{aligned} V_k^T r^k &= V_k^T (b - Ax^k) \\ &= V_k^T (r^0 - AV_k z^k) \\ &= \|r^0\| e^1 - V_k^T AV_k z^k \\ &= \|r^0\| e^1 - H_k z^k \end{aligned}$$

Ici, pour calculer z^k , on résout un système linéaire :

$$H_k z^k = \|r^0\| e^1$$

La solution peut aussi être obtenue à l'aide de la factorisation QR utilisée dans GMRES.

■ Un des inconvénients majeurs des méthodes GMRES et FOM est que le stockage mémoire et le nombre d'opérations par itération croissent avec le nombre d'itérations puisque, à chaque itération, on ajoute un vecteur de base. La parade qui a été imaginée est d'utiliser GMRES avec redémarrage. Toutes les m itérations, les vecteurs de base déjà calculés sont éliminés et les itérations sont redémarrées à partir de l'itéré courant. On appelle cette méthode **GMRES(m)**.

Pour des résultats de convergence, se reporter aux références [7] [9]. Si la partie symétrique de A ($= (A + A^T)/2$) est définie positive, alors GMRES(m) converge. Si ce n'est pas le cas, il existe des exemples pour lesquels GMRES(m) ne converge pas ou bien si lentement que la méthode n'est pas utilisable.

Dans la pratique, GMRES et FOM sont utilisées avec un préconditionnement M . Pour GMRES, cela revient à calculer r^0 par $Mr^0 = b - Ax^0$ et à initialiser le vecteur w du procédé de Gram-Schmidt modifié par $Mw = Av^k$; le reste de l'algorithme est inchangé.

2.3 Gradient conjugué

Lorsque la matrice A est symétrique, H_k est tridiagonale et FOM se réduit à la méthode de Lanczos pour la résolution de systèmes linéaires. Si, de plus, A est définie positive, en se servant de la factorisation de Cholesky de $T_k = H_k$ et en utilisant un changement de variables astucieux, on obtient la méthode du gradient conjugué [6] qui est l'algorithme itératif le plus populaire et le plus efficace pour résoudre les grands systèmes symétriques définis positifs. L'algorithme avec un préconditionnement M symétrique et défini positif est le suivant :

soit x^0 donné, $r^0 = b - Ax^0$. Pour $k = 0, 1, \dots$

$$Mz^k = r^k,$$

$$\beta_k = \frac{(z^k, Mz^k)}{(z^{k-1}, Mz^{k-1})}, \beta_0 = 0,$$

$$p^k = z^k + \beta_k p^{k-1},$$

$$\gamma_k = \frac{(z^k, Mz^k)}{(p^k, Ap^k)},$$

$$x^{k+1} = x^k + \gamma_k p^k,$$

$$r^{k+1} = r^k - \gamma_k A p^k.$$

Cette définition mathématique est particulièrement facile à coder. Le gradient conjugué est une méthode extrêmement intéressante car, en sus d'être une méthode d'orthogonalisation, elle minimise la A -norme de l'erreur à chaque itération. De plus, les différents vecteurs sont fournis par des récurrences courtes ; le stockage est donc minimal au contraire d'une méthode comme GMRES. Toutefois, les propriétés d'orthogonalité ne sont pas toujours vérifiées lorsque l'on calcule en précision finie et la convergence peut en être ralentie (cf. référence [8]).

2.4 Méthodes BiCG et BiCGstab

Malheureusement, lorsque les matrices ne sont pas symétriques, il n'est pas possible (sauf dans des circonstances très particulières) d'obtenir l'orthogonalité des résidus avec des récurrences courtes. On a donc recours à d'autres méthodes pour obtenir un stockage mémoire peu important. On construit deux bases, la première avec A en partant d'un vecteur v^1 , la seconde avec A^T en partant d'un vecteur \tilde{v}^1 tel que $(v^1, \tilde{v}^1) \neq 0$. De plus, on demande que les deux suites de vecteurs soient biorthogonales, c'est-à-dire :

$$(v^i, \tilde{v}^j) = 0, i \neq j$$

Cette construction est faite à l'aide de l'algorithme de Lanczos non symétrique qui s'écrit pour $k = 0, 1, \dots$

$$\begin{aligned} z^k &= A v^k - \delta_k v^k - \eta_k v^{k-1}, \quad w^k = A^T \tilde{v}^k - \delta_k \tilde{v}^k - \tilde{\eta}_k \tilde{v}^{k-1} \\ \delta_k &= (\tilde{v}^k, A v^k) \\ \eta_{k+1} \tilde{\eta}_{k+1} &= (z^k, w^k) \\ v^{k+1} &= \frac{z^k}{\tilde{\eta}_{k+1}}, \quad \tilde{v}^{k+1} = \frac{w^k}{\eta_{k+1}} \end{aligned}$$

Les scalaires η_k et $\tilde{\eta}_k$ peuvent être choisis de différentes façons. Si l'on note :

$$T_k = \begin{pmatrix} \delta_1 & \eta_2 & & & \\ \tilde{\eta}_2 & \delta_2 & \eta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \tilde{\eta}_{k-1} & \delta_{k-1} & \eta_k \\ & & & \tilde{\eta}_k & \delta_k \end{pmatrix}$$

$$V_k = [v^1 \dots v^k], \quad \tilde{V}_k = [\tilde{v}^1 \dots \tilde{v}^k]$$

on peut écrire matriciellement :

$$\begin{aligned} A V_k &= V_k T_k + \tilde{\eta}_{k+1} v^{k+1} (e^k)^T \\ A^T \tilde{V}_k &= \tilde{V}_k T_k^T + \eta_{k+1} \tilde{v}^{k+1} (e^k)^T \end{aligned}$$

On note la parenté avec l'algorithme de Lanczos symétrique. En utilisant ces vecteurs de base, on peut construire une méthode itérative de la même façon que l'on construit le gradient conjugué à partir de l'algorithme de Lanczos symétrique. Cette méthode s'appelle le gradient biconjugué (BiCG, pour *Bi-Conjugate Gradient*) ; elle s'écrit :

$$\begin{aligned} \alpha_k &= \frac{(\tilde{r}^k, r^k)}{(\tilde{p}^k, A p^k)} \\ x^{k+1} &= x^k + \alpha_k p^k \\ r^{k+1} &= r^k - \alpha_k A p^k \\ \tilde{r}^{k+1} &= \tilde{r}^k - \alpha_k A^T \tilde{p}^k \\ \beta_{k+1} &= \frac{(\tilde{r}^{k+1}, r^{k+1})}{(\tilde{r}^k, r^k)} \\ p^{k+1} &= r^k + \beta_{k+1} p^k \\ \tilde{p}^{k+1} &= \tilde{r}^k + \beta_{k+1} \tilde{p}^k \end{aligned}$$

On a les relations de biorthogonalité :

$$(\tilde{r}^k, r^\ell) = 0, \quad (\tilde{p}^k, A p^\ell) = 0, \quad k \neq \ell$$

Cependant, BiCG ne possède pas toutes les qualités remarquables du gradient conjugué. En effet, on peut avoir des cas de dégénérescence avec $(\tilde{r}^k, r^k) = 0$ mais $\tilde{r}^k \neq 0$, $r^k \neq 0$. On peut également avoir $(\tilde{p}^k, A p^k) = 0$. Ces ennuis peuvent être traités avec des techniques de « *look-ahead* » qui sont de mise en œuvre délicate. De plus, pour certains problèmes, BiCG peut avoir un comportement assez erratique. On observe fréquemment des oscillations importantes de la norme du résidu qui nuisent à la convergence et qui peuvent même aller jusqu'à complètement polluer la solution. Pour remédier à ces problèmes, H. Van der Vorst a développé BiCGstab [11], une variante plus stable de BiCG. La méthode est construite pour avoir :

$$r^k = \psi_k(A) \phi_k(A) r^0$$

où ϕ est le polynôme de BiCG tel que $r^k = \phi(A) r^0$.

Pour ψ , on choisit le produit de polynômes de degré 1 de résidu minimal :

$$\psi_{k+1}(t) = (1 - \omega_k t) \dots (1 - \omega_1 t)$$

Le polynôme ψ_k peut être calculé par récurrence :

$$\psi_{k+1}(t) = (1 - \omega_k t) \psi_k(t)$$

et les paramètres ω_k sont choisis pour minimiser la norme du résidu r^{k+1} . Finalement, la méthode s'écrit (avec un préconditionnement M) :

soit x^0 donné, $r^0 = b - A x^0$, $p^0 = r^0$, \tilde{r}^0 arbitraire, pour $k = 0, 1, \dots$

$$\hat{p}^k = M^{-1} p^k,$$

$$\tilde{p}^k = A \hat{p}^k,$$

$$\alpha_k = \frac{(r^k, \tilde{r}^0)}{(\tilde{p}^k, \tilde{r}^0)},$$

$$s^k = r^k - \alpha_k \tilde{p}^k,$$

$$\hat{s}^k = M^{-1} s^k,$$

$$\tilde{s}^k = A \hat{s}^k,$$

$$\omega_k = \frac{(\tilde{s}^k, s^k)}{(\tilde{s}^k, \tilde{s}^k)},$$

$$x^{k+1} = x^k + \alpha_k \hat{p}^k + \omega_k \hat{s}^k,$$

$$r^{k+1} = \hat{s}^k - \omega_k \tilde{s}^k,$$

$$\beta_{k+1} = \frac{(r^{k+1}, \tilde{r}^0)}{(r^k, \tilde{r}^0)} \frac{\alpha_k}{\omega_k},$$

$$p^{k+1} = r^{k+1} + \beta_{k+1} (p^k - \omega_k \tilde{p}^k).$$

Cet algorithme présente l'avantage supplémentaire de ne pas nécessiter de produits par la matrice A^T .

2.5 Méthode QMR

Dans la méthode de Lanczos non symétrique, on construit deux bases biorthogonales ; la matrice V_k n'est donc pas orthogonale. Ainsi, on ne peut pas, avec cette base, faire la même chose que dans GMRES. Nous avons :

$$A V_k = V_{k+1} \tilde{T}_k$$

La norme du résidu est :

$$\begin{aligned} \|r^k\| &= \|b - A x^k\| \\ &= \|r^0 - A V_k z^k\| \\ &= \|r^0\| V_{k+1} e^1 - V_{k+1} \tilde{T}_k z^k \| \end{aligned}$$

Malheureusement, la matrice V_{k+1} n'étant pas orthogonale, on ne peut pas éliminer de l'expression de la norme. Freund et Nachtigal [3] ont suggéré de calculer z^k en minimisant :

$$\| \|r^0\| e^1 - \tilde{T}_k z \|^2$$

Cette méthode s'appelle QMR (*Quasi Minimum Residual*). Numériquement, le problème de moindres carrés est résolu de la même façon que dans GMRES, mais les choses sont plus simples du fait de la structure tridiagonale de \tilde{T}_k . Des rotations sont calculées à chaque itération et appliquées à la matrice et au second membre. L'algorithme QMR donne souvent une convergence plus régulière que BiCGstab mais généralement pas plus rapide.

Les méthodes les plus fréquemment employées sont GMRES(m) et BiCGstab.

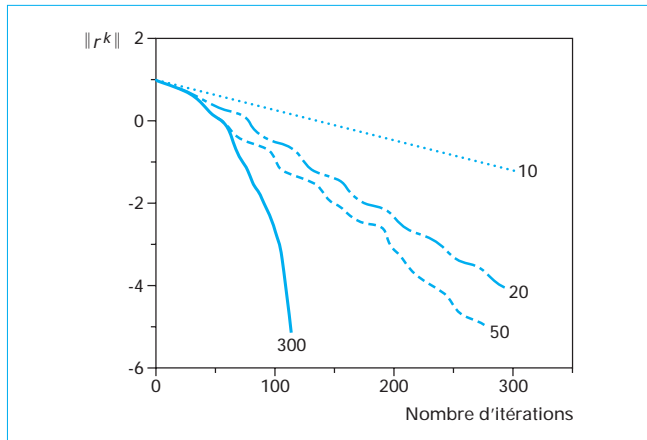


Figure 1 – Méthode GMRES(m). Norme ℓ_2 du résidu en fonction du nombre d'itérations ($m = 300, 50, 20, 10$)

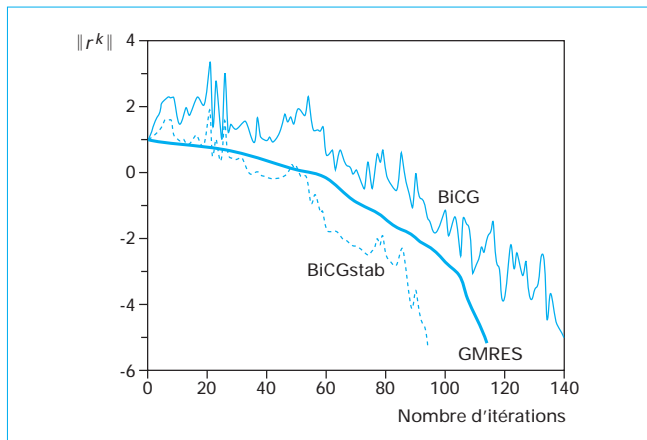


Figure 2 – Norme ℓ_2 du résidu en fonction du nombre d'itérations suivant les méthodes GMRES, BiCG et BiCGstab

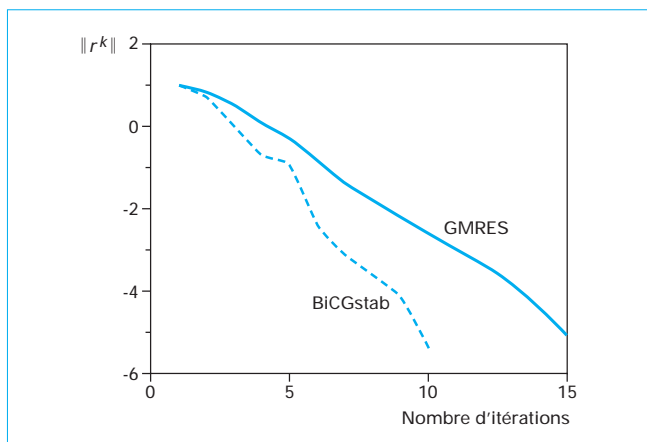


Figure 3 – Norme ℓ_2 du résidu en fonction du nombre d'itérations selon les méthodes GMRES et BiCGstab avec préconditionnement ILU(0)

3. Exemple

La matrice A non symétrique d'ordre $3m^2$ est pentadiagonale par blocs et dépend de cinq paramètres $\alpha, \delta, \mu, \omega$ et σ . Les blocs diagonaux sont des matrices 3×3 :

$$\begin{pmatrix} 4\alpha + \delta + \mu & -\mu & 0 \\ -\mu & 4\alpha + \mu + \omega & -\sigma \\ 0 & -\omega & 4\alpha + \sigma \end{pmatrix}$$

Les blocs sous-diagonaux sont égaux à :

$$\begin{pmatrix} -\alpha - \delta & & \\ & -\alpha & \\ & & -\alpha \end{pmatrix}$$

Les autres blocs non nuls sont égaux à $-\alpha I$. Si l'on considère les indices des blocs, pour le bloc J les blocs non nuls (lorsqu'ils existent) sont situés aux positions $J-m, J-1, J, J+1, J+m$.

Les paramètres choisis sont $m = 10$, ce qui donne un système d'ordre 300 et $\alpha = 1, \mu = 0,1, \omega = 0,2, \sigma = 0,3, \delta = 10$. Le vecteur de départ est le vecteur nul et le second membre composé de nombres aléatoires. Le critère d'arrêt est :

$$\|r^k\| \leq 10^{-6} \|r^0\|$$

La figure 1 montre la convergence de GMRES(m) en fonction du nombre d'itérations pour $m = 300, 50, 20, 10$. GMRES(300) correspond, dans ce cas, à GMRES sans redémarrage. Il faut cependant noter que, bien que le nombre d'itérations soit plus élevé pour GMRES(50), etc., le coût par itération est inférieur lorsque m décroît, ainsi il est possible que le coût de ces méthodes soit plus faible que pour GMRES(300).

Sur la figure 2, on voit la norme du résidu en fonction du nombre d'itérations de GMRES, BiCG et BiCGstab. BiCG oscille énormément et il convient de noter que BiCG demande deux produits matrice-vecteur par itération alors que GMRES ne requiert qu'un seul produit. BiCGstab présente beaucoup moins d'oscillations que BiCG.

Sur la figure 3, on utilise un préconditionnement $ILU(0)$ (cf. référence [7]), pour GMRES et BiCGstab. Les nombres d'itérations sont beaucoup plus faibles et la convergence de BiCGstab beaucoup plus régulière.

Pour une présentation générale de l'algèbre linéaire numérique, on pourra consulter le dossier paru dans cette base documentaire des Techniques de l'Ingénieur de Claude Brezinski [13].

Par ailleurs, il existe sur Internet des logiciels (gratuits) implémentant les méthodes que l'on a présentées ci-dessus.

On pourra aussi consulter l'ouvrage [1] qui se trouve aussi sur Internet à l'adresse :

<http://www.netlib.org/templates/Templates.html>

et les codes correspondants disponibles sur Netlib.

Il existe également des logiciels écrits par Y. Saad qui peuvent être obtenus à l'adresse :

<http://www.users.cs.umn.edu/~saad/software/SPARSKIT/sparskit.html>

Enfin, d'autres logiciels de qualité sont disponibles à : <http://www.cerfacs.fr/algors/Softs/index.html>

Références bibliographiques

- [1] BARRETT (R.), BERRY (M.), CHAN (T.F.), DEMMEL (J.), DONATO (J.), DONGARRA (J.), EIJKHOUT (V.), POZO (R.), ROMINE (C.) et VAN DER VORST (H.V.). – *Templates for the solution of linear systems : building blocks for iterative methods*. SIAM (1994).
- [2] FISCHER (B.). – *Polynomial based iteration methods for symmetric linear systems*. Wiley Teubner (1996).
- [3] FREUND (R.W.) et NACHTIGAL (N.M.). – *QMR : a quasi-minimal residual method for non Hermitian linear systems*. Numer. Math., vol. 60, p. 315-339 (1991).
- [4] GOLUB (G.H.) et VAN LOAN (C.). – *Matrix computations*. Johns Hopkins University Press (1989).
- [5] GREENBAUM (A.). – *Iterative methods for solving linear equations*. SIAM (1997).
- [6] HESTENES (M.R.) et STIEFEL (E.). – *Methods of conjugate gradients for solving linear systems*. J. Nat. Bur. Stand., vol. 49, n° 6, p. 409-436 (1952).
- [7] MEURANT (G.). – *Computer solution of large linear systems*. North-Holland (1999). gerard.meurant@cea.fr
- [8] MEURANT (G.). – *The Lanczos and conjugate gradient algorithms, from theory to finite precision computations*. SIAM (2006).
- [9] SAAD (Y.). – *Iterative methods for sparse linear systems*. PWS Publishing Company (1996).
- [10] SAAD (Y.) et SCHULTZ (M.H.). – *GMRES : a generalized minimal residual algorithm for solving nonsymmetric linear systems*. SIAM J. Sci. Comput., vol. 7, p. 856-869 (1986).
- [11] VAN DER VORST (H.A.). – *BiCGSTAB : a fast and smoothly converging variant of BiCG for the solution of nonsymmetric linear systems*. SIAM J. Sci. Comput., vol. 13, p. 631-644 (1992).
- [12] VAN DER VORST (H.A.). – *Iterative Krylov methods for large linear systems*. Cambridge University Press (2003).
- [13] BREZINSKI (C.). – *Méthodes numériques de base. Algèbre numérique (AF 1 221)*. Base documentaire « Mathématiques » (2006).

Aux Éditions T.I.
Dans les Techniques de l'Ingénieur