

Utilisation de méthodes de type Lanczos dans la bibliothèque MODULAD

J L Vaudescal

I N R I A

Domaine de Voluceau

78153 le Chesnay CEDEX France

email octobre@goofy.inria.fr

Résumé :

Cet article se propose d'indiquer aux utilisateurs de la bibliothèque MODULAD, l'existence des méthodes de calcul spectral de type Lanczos.

Ces méthodes sont très efficaces quand une faible partie du spectre est recherchée, ce qui est le cas dans beaucoup d'applications en analyse des données. Des comparaisons avec l'algorithme QR sont effectuées.

Mots Clés :

Problème aux valeurs propres, Méthodes de Lanczos, Méthode QR.

1 Introduction

L'analyse de nombreux problèmes statistiques nécessite la connaissance d'une partie du spectre de matrices. Nous nous proposons ici d'attirer l'attention des utilisateurs de la bibliothèque MODULAD, sur l'existence de méthodes numériques de calcul de valeurs et de vecteurs propres, adaptées aux problèmes où seul un faible pourcentage du spectre est recherché (grossièrement moins de 25 %).

Les méthodes de Lanczos, sous diverses formes, ont fait l'objet de travaux importants et de nombreuses publications traitent de ces algorithmes. Leur emploi est déjà très répandu dans le calcul numérique intensif dans des domaines tels : la mécanique vibratoire, la physique des plasmas, la neutronique (physique nucléaire) entre autres.

Leur utilisation dans les méthodes de traitement statistique, notamment en analyse des données, semble moins répandue, la méthode QR ou celle d'Housolder-Jacobi étant très utilisées et préférées.

La raison de cette préférence semble venir du fait que juste qu'à récemment, les tailles des matrices rencontrées étaient de tailles raisonnables (quelques centaines) et que dans ce cas, les méthodes de type QR sont assez performantes.

Cependant, le coût de cette méthode QR est proportionnel à n^3 (n est la taille de la matrice à diagonaliser), ce qui peut devenir prohibitif pour des matrices ne serait-ce de dimension 1000, sur les ordinateurs moyennement puissants.

Il semblait donc intéressant de donner quelques informations sur des méthodes connues pour être beaucoup plus rapide que la méthode QR dans certains créneaux d'utilisation.

Nous rappelons donc brièvement le principe de la forme la plus simple d'algorithme de Lanczos généralisée, connue sous le nom de méthode d'Arnoldi, utilisable dans une grande variété de problèmes. Nous indiquons ensuite, une forme particulière de l'algorithme quand on l'applique aux matrices symétriques, aboutissant à l'algorithme de Lanczos simple.

Enfin, nous présentons des exemples numériques montrant des voies d'utilisation de ces méthodes, dans des cas où elles apparaissent plus performantes que la méthode QR.

2 Présentation de la méthode

Les méthodes de Lanczos généralisées consistent toutes à vouloir projeter la matrice A de l'opérateur à diagonaliser, sur un sous-espace de dimension réduite m noté K_m .

Ce sous-espace K_m est en général un sous-espace de Krylov (voir Parlett [6] par exemple) engendré par la suite des vecteurs $v_1, A \cdot v_1, A^2 \cdot v_1, \dots, A^{m-1} \cdot v_1$, où v_1 est un vecteur initial à choisir. L'étude théorique de ce procédé est présenté dans Saad ([8]).

L'algorithme le plus simple permettant d'effectuer cette projection est l'algorithme d'Arnoldi dont nous présentons la forme générale

Soit v_1 un vecteur initial et m un entier choisis, A matrice $N \times N$.

choisir un vecteur v^1 tel que $\|v^1\| = 1$

(ici, $\|\cdot\|$ désigne la norme l^2 discrète.)

choisir la dimension m du sous-espace de Krylov.

```

pour j=1 à m faire :
    * Faire  $Aw = v^j$ 
    pour i=1 à j faire :
        calculer  $h(i, j) = (w, v^j)$ 
    calculer  $w = w - h(i, j) \cdot v^i$ 
    fin pour i
    * Calculer  $h(j+1, j) = (w, w)$ 
    * Calculer  $v^{j+1} = \frac{w}{h(j+1, j)}$ 
    fin pour j
    
```

Les propriétés de convergence de cet algorithme et ses variantes sont données dans [8],[9] ou [10], on en rappelle simplement les principaux résultats :

Théorème 2.1 *Le système $\{v_1, v_2, \dots, v_m\}$ obtenu à partir de l'algorithme ci-dessus forme une base orthonormée de K_m engendrée par $\{v_1, Av_1, \dots, (A)^{m-1}v_1\}$.*

Soit V_m la matrice $n \times m$ (n dimension de A) formée à partir des vecteurs colonnes v_1, v_2, \dots, v_m . Alors la matrice $H_m = V_m^t A V_m$ est une matrice de Hessenberg Supérieure $m \times m$ dont les coefficients sont les $h(i, j)$ obtenus grâce à l'algorithme précédent

Corollaire 2.1 *Les valeurs de Ritz de A dans K_m sont les valeurs propres λ_i^m de H_m et les vecteurs de Ritz sont les vecteurs $\phi^m = V_m y_i^m$ où y_i^m sont les vecteurs propres de H_m associés à λ_i^m*

On peut mesurer la précision à laquelle on a estimé les valeurs propres à l'aide de l'estimateur de la norme résiduelle suivant :

$$\|(A - \lambda^m I) \cdot \phi^m\| = h(m+1, m) \cdot |\epsilon_m^t y^m| \text{ avec } \epsilon_m = (0, 0, \dots, 0, 1)^t.$$

Le corollaire indique donc que pour calculer les valeurs propres approchées des valeurs propres de A , il suffit de calculer les valeurs propres de la matrice H_m . Celle-ci étant de petite taille (en général), on applique l'algorithme QR à cette matrice de Hessenberg.

Nous insistons sur le fait que cette méthode est généralement utilisée pour des problèmes à matrices creuses de grandes tailles (plusieurs centaines de milliers d'inconnues) et l'outil de base est une procédure de produit par la matrice A par un vecteur. Plus cette partie du programme sera optimisée, plus l'algorithme sera efficace.

Il convient de se convaincre qu'il est préférable d'utiliser cette méthode lorsqu'une faible partie du spectre est recherchée ; lorsqu'une partie trop importante (voir la totalité) de ce spectre est recherchée, l'algorithme QR reste plus rapide.

Mentionnons également que lorsqu'une seule valeur propre est recherchée (généralement la valeur propre extrême) et que celle-ci est mal séparée de ses voisines, la méthode traditionnellement employée est la méthode des puissances et celle-ci connaît alors des ralentissements importants de convergence ([11] par exemple).

Des études ont montré la supériorité de la méthode d'Arnoldi ([5]) dans ce cas de figure.

La forme présentée de l'algorithme ci-dessus est utilisable quand l'information recherchée est essentiellement le haut du spectre, mais il fournit également un peu d'information dans l'autre extrémité du spectre, la convergence de la méthode se faisant d'abord par les extrémités du spectre de A (voir Saad [8]).

L'utilisateur désireux d'obtenir de l'information d'abord dans la partie basse du spectre effectuera les itérations sur la matrice A^{-1} (ie faire $w = A^{-1}.v_j$) ou avec une matrice $(A - \mu.I)$ si l'information est recherchée autour de la valeur μ .

Enfin des problèmes aux valeurs propres généralisés de type $Ax = \lambda.Bx$ peuvent être appréhendés en itérant sur des matrices équivalentes à $A^{-1}B$ ou à $B^{-1}A$ (ou sous d'autres formes si les matrices sont factorisées) selon la nature du problème considéré.

Mentionnons enfin qu'une version itérative de l'algorithme d'Arnoldi peut être utilisée.

Si à la fin des m itérations, l'information obtenue est insuffisante, on peut redémarrer l'algorithme avec un nouveau vecteur initial formé d'une combinaison linéaire des approximations des vecteurs propres insuffisamment convergés

Cette forme est particulièrement adaptée quand des problèmes de limitations de mémoire surgissent.

Cependant, on notera qu'en général, la convergence de l'algorithme utilisé p fois avec m itérations est plus faible que si on effectue un seul passage avec mp itérations.

Diverses autres formes de cet algorithme sont présentées dans [8], [3] ou dans [9] et l'utilisation des méthodes accélératrices (de type accélérations de Tchebychev) est possible (voir [1] ou [10]).

3 Le cas symétrique

Dans le cas où la matrice A à diagonaliser est symétrique, le version de l'algorithme de projection sur le sous-espace K_m correspond à l'algorithme de Lanczos simple. Une vaste bibliographie à ce sujet existe, nous citerons par exemple [3], [11], [6] et [4] pour une présentation assez complète de l'algorithme sous ses diverses formes. L'algorithme sous sa forme la plus simple s'écrit sous la forme suivante :

fournir u et m

$$\text{faire } v_1 = \frac{u}{\|u\|}$$

$$\alpha_1 = v_1^t . A . v_1$$

$$\beta_1 = 0$$

Pour $j = 1$ à m faire

$$w = A . v_j - \alpha_j . v_j - \beta_j . v_{j-1}$$

$$\beta_{j+1} = \|w\|$$

si $\beta_{j+1} > 0$ alors $v_{j+1} = \frac{w}{\beta_{j+1}}$ sinon stop

$$\alpha_{j+1} = v_{j+1}^t . A . v_{j+1}$$

fin pour j

Notons que cet algorithme correspond à l'algorithme d'Arnoldi quand A est symétrique et que dans ce cas, le vecteur $A.v_j$ est orthogonal aux vecteurs v_k pour $k \leq j - 2$.

Ceci explique pourquoi une relation seulement tri-réursive est effectuée dans le cas symétrique. Cette propriété est vraie en arithmétique exacte mais elle n'est plus assurée en arithmétique finie sur ordinateur et ceci parfois dès la deuxième itération.

L'étude et la compréhension de ce phénomène est présentée dans [6] (et les références s'y trouvant) ; c'est précisément cette perte d'orthogonalité des vecteurs successifs qui est à l'origine d'un engouement moindre pour la méthode de Lanczos.

Une réorthogonalisation des vecteurs v_j produits par l'algorithme doit être effectuée au cours des itérations pour pouvoir continuer d'utiliser la relation à trois termes. Différentes stratégies

de réorthogonalisation ont été étudiées à savoir, réorthogonalisation totale, partielle [7] ou inexistante [4] ou [12] : chacune aboutissant à une mise en oeuvre spécifique.

La version écrite utilise une réorthogonalisation totale avec l'algorithme de "Gramm-Schmidt" ([6] ou [11]), ce qui nécessite de conserver en mémoire tous les vecteurs v_j successifs

Cet algorithme construit une matrice T_m tridiagonale symétrique dont les termes diagonaux sont les α_i , ($i = 1, m$) et les termes surdiagonaux les β_i , ($i = 2, m$) obtenus par l'algorithme précédent.

Les résultats annoncés pour la méthode d'Arnoldi restent valables ici

On notera qu'utilisés sous cette forme simple, ces algorithmes ne permettent pas de détecter des valeurs propres multiples (à moins bien entendu d'itérer sur l'espace tout entier...) (voir [8] ou [4] pour une démonstration).

Des versions blocs de ces algorithmes ont été étudiées, permettant l'obtention de valeurs propres multiples, de multiplicité au maximum la taille des blocs.

Nous renvoyons le lecteur à [4] ou à [2] pour plus de détails sur ce sujet

On remarquera qu'une des premières applications de cet algorithme proposé par Lanczos (1950), était d'effectuer la tridiagonalisation d'une matrice hermitienne.

La perte d'orthogonalité des vecteurs mentionnée ci-dessus (et ses remèdes) ont rendu cette méthode inintéressante par rapport à la méthode d'Householder beaucoup plus stable et de ce fait, préférée par les numériciens

Le regain d'utilisation de la méthode de Lanczos est dû au fait de sa capacité de permettre le calcul d'une petite partie du spectre sans avoir à calculer tout le spectre (contrairement à la méthode QR) ce qui, quand les matrices traitées sont déjà (seulement) de l'ordre du millier d'inconnues, est un avantage conséquent.

Du point de vue pratique, on utilise l'algorithme ci-dessus pour obtenir une matrice tridiagonale T_m , on en calcule ensuite les valeurs propres par un algorithme de type QL (par exemple).

Plus m sera grand, plus les valeurs calculées seront proches des valeurs propres de la matrice A , la qualité d'approximation étant généralement meilleure aux extrémités du spectre ou/et près des valeurs propres bien séparées à l'intérieur du spectre.

C'est cette propriété qui semble intéressante au moins dans les cas présentés dans le paragraphe suivant

4 Quelques applications possibles

Nous présentons ici des cas numériques pour lesquels les méthodes de type Lanczos semblent être, en général, plus performantes que les méthodes QR et d'Householder-Jacobi.

La première application peut se présenter sous la forme suivante :

On dispose d'une matrice carrée (tableau de distances) dont on cherche à savoir si elle est proche d'une métrique euclidienne. Pour ce faire, on cherche à savoir si des valeurs propres négatives existent et quelle proportion de valeurs strictement positives sont relevées, afin de connaître une dimension de sous-espace qui serait considéré comme euclidien.

Les cas présentés englobent les cas de figure suivants :

- peu de valeurs propres non nulles sont présentes (cas 1).
- peu de valeurs propres sont nulles mais certaines valeurs propres sont négatives (cas 2 et cas 3)
- la taille de la matrice traitée devient relativement grande (cas 4).

Le cas 4 est le plus intéressant car le plus proche des situations de calculs où le volume des données traité est important. Typiquement, la matrice correspondante, de dimension 525×525 doit être diagonalisée car on recherche les valeurs propres de plus grand module.

Cette situation est celle de l'A.C.P ou de l'analyse factorielle sous ses diverses formes, méthodes qui ne demandent finalement que le calcul des premières plus grandes valeurs propres et des vecteurs propres associés.

Dans le cas 4, la difficulté est accrue par le fait que le spectre de cette matrice est très large et aucun caractère de positivité n'est alors exploitable.

On conçoit ainsi fort bien l'intérêt de la méthode en analyse factorielle.

Les graphiques suivants montrent comment, en fonction du problème traité, et en fonction de la taille du sous-espace de projection, les estimations des valeurs propres fournies par l'algorithme, se rapprochent des estimations fournies par ailleurs par l'algorithme QR.

Le tableau de temps CPU indique les performances de la méthode en fonction de la taille de sous-espace choisie, comparativement à l'algorithme QR.

Les temps ont été relevés sur station de travail APOLLO série 5550/35.

Ces résultats illustrent essentiellement les remarques des sections précédentes sur le comportement et l'utilisation possible de cet algorithme.

Ainsi, le cas 1 est très favorable pour les méthodes de Lanczos, il suffit d'une dimension de sous-espace de 10 pour "capter" correctement toutes les valeurs propres non nulles, la valeur propre nulle étant multiple, celle-ci est détectée une fois comme les autres, la multiplicité apparente de la valeur propre nulle est due aux erreurs d'arrondis (cette affirmation n'est pas illustrée ici).

L'intérêt de la méthode s'atténue quelque peu pour les cas 2 et 3, la raison essentielle est que l'on traite des matrices de tailles insuffisantes par rapport au volume d'information demandée, on illustre ici le fait que cet algorithme devient moins intéressant que la méthode QR quand environ plus de 25% du spectre est souhaité.

Dans le cas 4, on notera le gain important en temps calcul, le spectre approché est très rapidement suffisant quand quelques valeurs propres sont souhaitées, ceci est dû à la relative bonne séparation des valeurs propres de plus grand module.

Dans le cas 4, on notera le gain important en temps calcul. le spectre approché est très rapidement suffisant quand quelques valeurs propres sont souhaitées, ceci est dû à la relative bonne séparation des valeurs propres de plus grand module.

Contrairement aux autres cas, par soucis de lisibilité, nous indiquons les valeurs calculées pour les différentes valeurs de dimension de sous-espace, ainsi que les valeurs calculées par l'algorithme QR (qui font office de référence). On note bien la bonne convergence de l'algorithme pour les valeurs propres de plus grands modules, la convergence des valeurs propres internes au spectre augmentant avec la taille du sous-espace.

m=10	m=15	m=20	m=25	m=30	QR
-2731.823	-2731.824	-2731.824	-2731.824	-2731.823	-2731.823
-1067.832	-1067.832	-1067.832	-1067.833	-1067.832	-1067.832
-797.7861	-797.7858	-797.7861	-797.7862	-797.7863	-797.7870
-412.0186	-412.0432	-412.0431	-412.0430	-412.0432	-412.0432
-243.6162	-250.3029	-250.3031	-250.3031	-250.3032	-250.3033
-220.1582	-228.9462	-228.9463	-228.9463	-228.9462	-228.9462
-128.4179	-132.1026	-132.7116	-132.7116	-132.7114	-132.7116
-46.39637	-129.3141	-129.6212	-129.6212	-129.6212	-129.6214
-14.28421	-101.1184	-102.4787	-102.4787	-102.4787	-102.4789
7756.058	-70.11836	-88.55044	-88.55334	-88.55333	-88.55350
	-43.00409	-70.96017	-70.96509	-70.96495	-70.96535
	-27.08489	-50.24157	-68.65553	-68.98966	-68.98974
	-13.56328	-44.42218	-54.06242	-55.43913	-55.44155
	-3.640239	-35.97620	-48.28256	-52.27536	-52.28609
	7756.058	-25.37877	-44.77990	-47.76767	-47.79875
		-17.55693	-38.33160	-44.92157	-45.14880
		-10.03498	-31.97774	-39.27994	-41.20641
		-4.483261	-24.12108	-37.57647	-38.51022
		-1.199528	-18.72959	-32.66642	-33.05197
		7756.056	-13.45545	-26.97902	-32.56234
			-8.528644	-22.59573	*
			-4.873158	-18.80737	*
			-0.3036596	-10.88276	*
			7756.059	-7.322281	*
				-4.682716	*
				-2.734231	*
				-1.035441	*
				7.5025618E-02	7.2416127E-02
				7756.056	7756.049

On donne maintenant les temps CPU relevés dans les différents cas.

m	5	10	15	20	25	30	QR
cas 1	0.18	0.39	0.69	1.05	1.53	2.22	1.62
cas 2	0.1	0.25	0.46	0.82	1.2	1.7	0.6
cas 3	0.38	0.82	1.3	1.93	2.65	3.52	4.88
cas 4	*	10.5	15.9	21.4	27.2	34.3	240

Ces quelques résultats numériques n'ont bien entendu qu'une valeur illustratrice, chaque problème aux valeurs propres ayant sa spécificité. Nous insistons également sur le fait que les exemples traités restent de tailles modestes et que les possesseurs de matrices beaucoup plus importantes doivent pouvoir trouver quelque intérêt à tester les méthodes de Lanczos. Signalons enfin que l'algorithme présenté est utilisé sans redémarrage.

References

- [1] D HO F. CHATELIN M. BENNAMI. - Arnoldi-Tchebychev procedure for large scale non symmetric matrices. *Mathematical Modeling and Numerical analysis*, vol. 24, n° 1, 1990.
- [2] B. VITAL. - - Thèse de Doctorat, Université Rennes, 1989. Thèse
- [3] F. CHATELIN - *Valeurs propres de matrices*. - Masson, 1988
- [4] J.K CULLUM ET R A WILLOUGHBY. - *Lanczos Algorithms for Large Symmetric Eigenvalue Computations* - Progress in Scientific Computing Series, 1985 volume 1.
- [5] J. JAFFRE ET J.L VAUDESCAL. - *Etude de la résolution du problème de la diffusion neutronique à deux groupes par la méthode d'Arnoldi*. - Rapport de Recherche, INRIA, to appear.
- [6] B. N. PARLETT - *The Symmetric Eigenvalue Problem*. - Prentice Hall, 1980
- [7] B. N. PARLETT ET D S SCOTT. - The Lanczos Algorithm with Selective Orthogonalization. *Mathematics of Computation*, vol. 33, n° 145, 1979, pp 217-238.
- [8] Y. SAAD. - Variation on Arnoldi's Method for computing eigenvalues of large unsymmetric matrices. *Linear Algebra and its Applications*, vol. 34, 1980, pp. 269-295.
- [9] Y. SAAD. - Chebyshev Acceleration Techniques for solving Nonsymmetric Eigenvalue Problems. *Mathematics of Computation*, vol. 42, n° 166, 1984, pp. 576-588
- [10] Y. SAAD. - Iterative techniques for solving large sparse linear systems and eigenvalue problems. In: *Cycle de formation sur les supercalculateurs C.E.R.F.A.C.S.*
- [11] J. H. WILKINSON. - *The Algebraic Eigenvalue Problem* - Oxford University Press, 1965.
- [12] W. KERNER. - Large-Scale Complex Eigenvalue Problems. *Journal of Computational Physics*, no85, 1989, pp. 1-85.

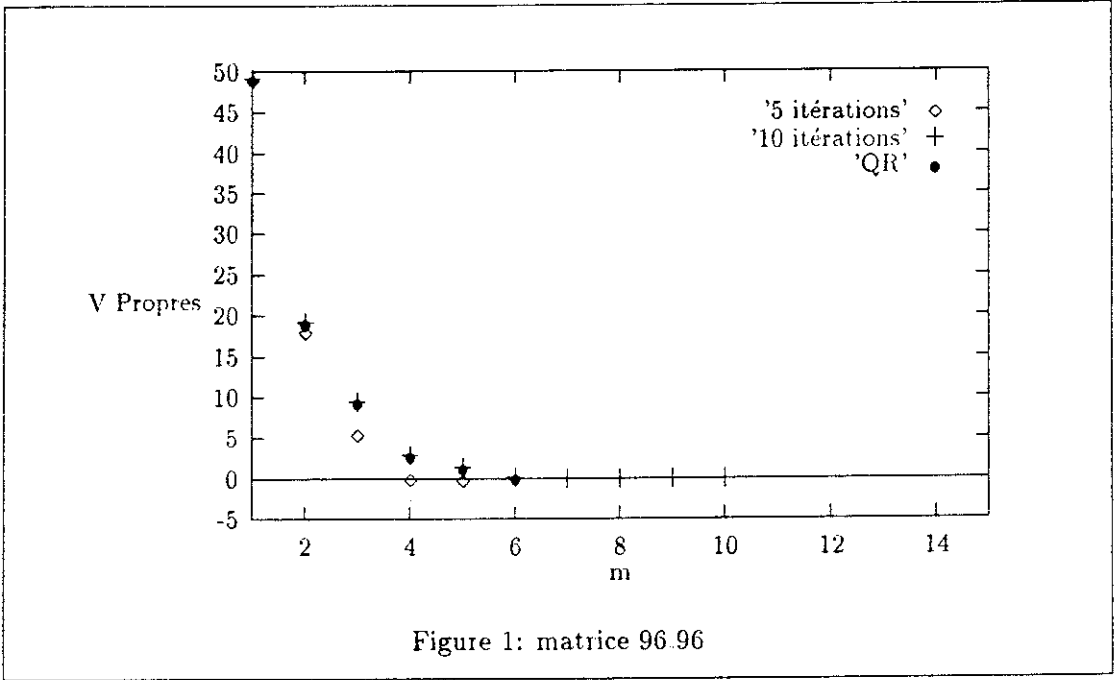


Figure 1: matrice 96.96

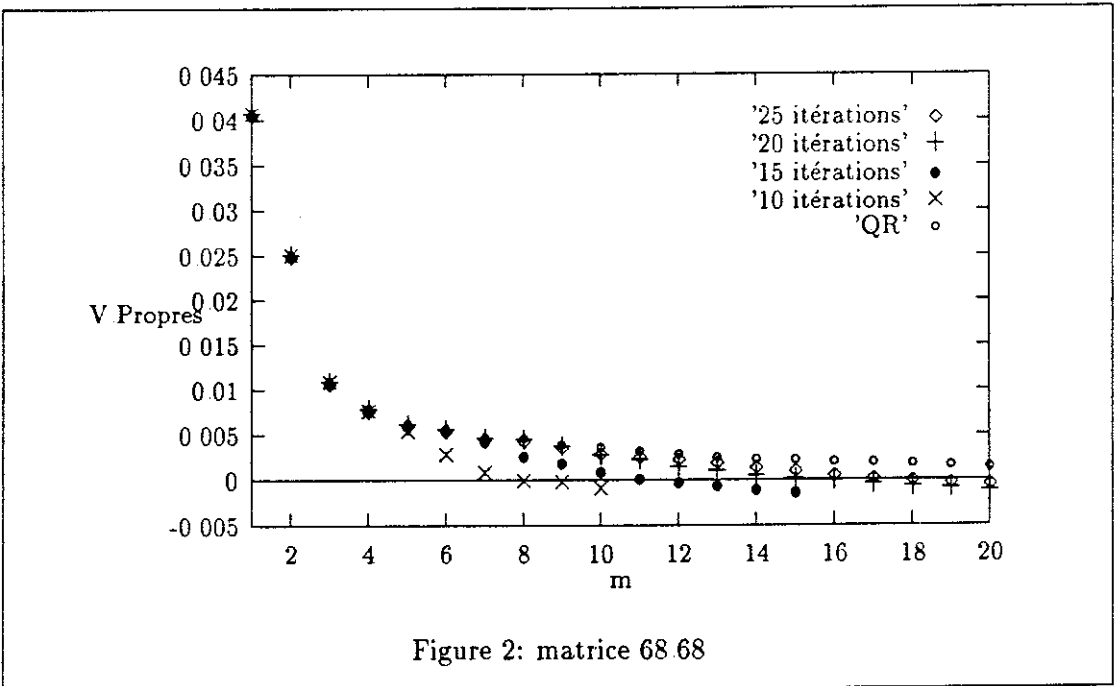


Figure 2: matrice 68.68

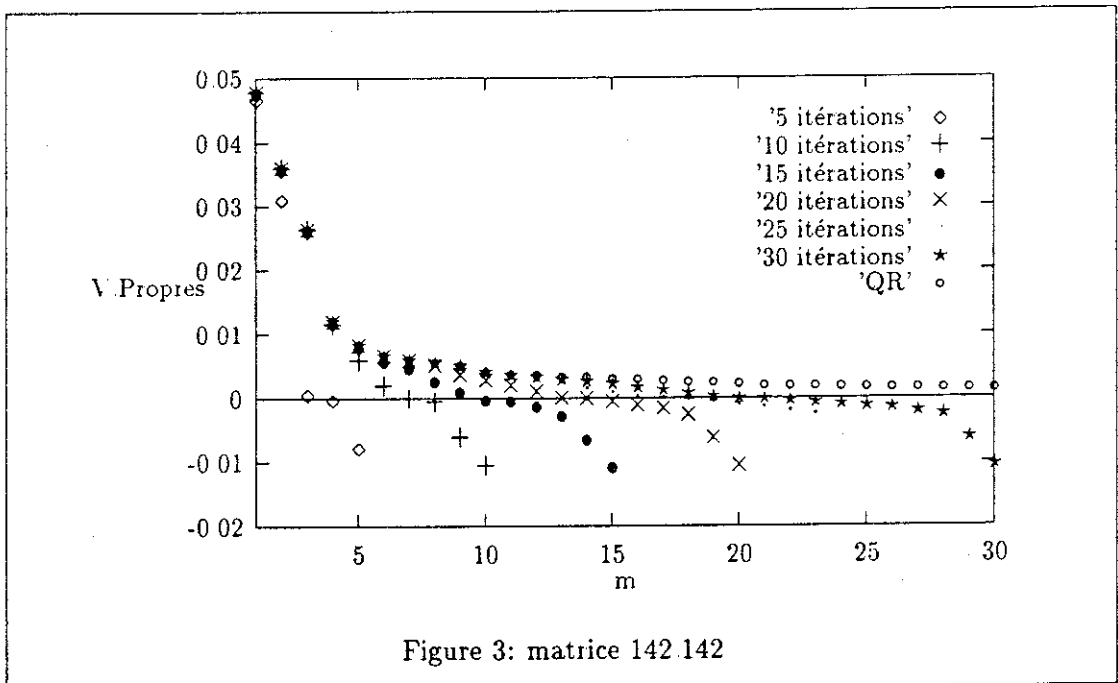


Figure 3: matrice 142.142