

EXTRAPOLATION VECTORIELLE ET APPLICATIONS AUX ÉQUATIONS AUX DÉRIVÉES PARTIELLES

Sébastien DUMINIL

L.M.P.A
Université du Littoral - Calais



06 Juillet 2012
Soutenance de Thèse

- 1 MÉTHODES D'EXTRAPOLATION VECTORIELLES
- 2 APPLICATIONS À LA RÉOLUTION DE SYSTÈMES LINÉAIRES
 - Application aux méthodes multigrilles
 - Méthodes d'extrapolation et Méthodes de sous-espaces de Krylov
 - Méthode CMRH
- 3 APPLICATIONS À LA RÉOLUTION DE SYSTÈMES NON LINÉAIRES
 - Théorème de convergence
 - Application aux équations de Navier-Stokes
 - Application à l'équation de Schrödinger
- 4 CONCLUSION ET PERSPECTIVES

DÉFINITION GÉNÉRALE

Soit (s_k) une suite de vecteurs de \mathbb{R}^n . Posons $\Delta s_k = s_{k+1} - s_k$ et $\Delta^2 s_k = \Delta s_{k+1} - \Delta s_k$. À partir de la suite (s_k) , les méthodes polynômiales d'extrapolation construisent une nouvelle approximation t_k de la limite ou de l'anti-limite de (s_k) définie par

DÉFINITION

$$t_k = \sum_{j=0}^k \gamma_j s_j$$

avec

$$\sum_{j=0}^k \gamma_j = 1, \quad \sum_{j=0}^k \eta_{i,j} \gamma_j = 0, \quad i = 0, 1, \dots, k-1, \quad (1)$$

où les scalaires $\eta_{i,j}$ définissent la méthode.

η_{ij} est défini par

$$\eta_{ij} := (y_i, \Delta s_j)$$

avec

- $y_i := \Delta s_i$ pour la méthode Minimal Polynomial Extrapolation (MPE) (Cabay et Jackson, '76),
- $y_i := \Delta^2 s_i$ pour la méthode Reduced Rank Extrapolation (RRE) (Eddy et Mesina, '77),
- $y_i := y_i$ pour la méthode Modified Minimal Polynomial Extrapolation (MMPE) (Brezinski, '75, Pugatchev, '78).

DÉFINITION GÉNÉRALE

Posons

$$Y_k = [y_0, \dots, y_{k-1}]$$

et

$$\Delta^i S_k = [\Delta^i s_0, \dots, \Delta^i s_{k-1}], \quad i = 1, 2.$$

Alors

En utilisant la formule de Schur, les méthodes d'extrapolation sont définies par l'expression suivante :

$$t_k = s_0 - \Delta S_k (Y_k^T \Delta^2 S_k)^{-1} Y_k^T \Delta s_0.$$

avec

$$\begin{array}{lll} Y_k & := & \Delta S_k & \text{pour MPE,} \\ Y_k & := & \Delta^2 S_k & \text{pour RRE,} \\ Y_k & := & Y_k & \text{pour MMPE.} \end{array}$$

On introduit les scalaires β_i , $i = 0, \dots, k$ définis par $\beta_i = \frac{\gamma_i}{\gamma_k}$.
 Le système (1) peut s'écrire sous la forme suivante

$$(Y_k^T \Delta S_k) \beta = -Y_k^T \Delta s_k. \quad (2)$$

On introduit ensuite α défini par

$$\alpha_0 = 1 - \gamma_0, \quad \alpha_j = \alpha_{j-1} - \gamma_j \quad 1 \leq j \leq k-1.$$

On obtient

$$t_k = s_0 + \sum_{j=0}^{k-1} \alpha_j \Delta s_j = s_0 + \Delta S_k \alpha.$$

Supposons que ΔS_{k+1} est de rang maximal. Alors il existe :

- une factorisation QR telle que $\Delta S_{k+1} = Q_{k+1} R_{k+1}$
- une factorisation LU telle que $P_{k+1} \Delta S_{k+1} = L_{k+1} U_{k+1}$.

Alors β est solution de

- MPE : $R_k[\beta_0, \dots, \beta_{k-1}]^T = -r_k$ et $\beta_k = 1$
- RRE : $R_{k+1}^T R_{k+1} \beta = [1, \dots, 1]^T$
- MMPE : $U_k[\beta_0, \dots, \beta_{k-1}]^T = -u_k$ et $\beta_k = 1$

et

- MPE et RRE : $t_k = s_0 + Q_k R_k \alpha$
- MMPE : $t_k = s_0 + P_k^T L_k' U_k \alpha$

ALGORITHME DE LA MÉTHODE RRE

- Entrée : s_0, \dots, s_{k+1}
- Poser $y = s_1 - s_0$, $r_{0,0} = \|y\|$ et $q_0 = (1/r_{0,0})y$.
- Pour $i = 1, \dots, k$
 - Poser $y = s_{i+1} - s_i$
 - Pour $j = 0, \dots, i - 1$
 - $r_{j,i} = q_j^T y$ et $y = y - r_{j,i}q_j$
 - Fin Pour
- Fin Pour
- Résoudre le système linéaire
- $R_{k+1}^T R_{k+1} d = e$.
- Poser $\lambda = \sum_{i=0}^k d_i$.
- Poser $\gamma_i = (1/\lambda) d_i$ pour $i = 0, \dots, k$.
- Calculer $\alpha = (\alpha_0, \dots, \alpha_{k-1})^T$ via
- $\alpha_0 = 1 - \gamma_0$; $\alpha_j = \alpha_{j-1} - \gamma_j$, $j = 1, \dots, k - 1$.
- Calculer t_k via
- $t_k = s_0 + Q_k(R_k \alpha)$.

On suppose que $k \ll n$.

Méthodes	MPE	RRE	MMPE
Multiplications et Additions	$2nk^2$	$2nk^2$	nk^2
Stockage	$(k+1)n$	$(k+1)n$	$(k+1)n$

TABLE : Coût des différentes méthodes d'extrapolation

Lorsque k augmente :

- Coût en calculs augmente de façon quadratique
- Coût de stockage augmente de façon linéaire

Pour réduire ces coûts, on préférera utiliser une méthode cyclique. Pour cela, on fixe un entier q et l'on redémarre l'algorithme en posant $s_0 = t_q$.

Posons

$$\tilde{t}_k = \sum_{j=0}^k \gamma_j \mathbf{s}_{j+1}.$$

DÉFINITION

Le résidu généralisé de t_k est défini de la façon suivante :

$$\tilde{r}(t_k) = \tilde{t}_k - t_k.$$

Ce résidu peut être exprimé sous la forme suivante :

$$\tilde{r}(t_k) = \Delta \mathbf{s}_0 - \Delta^2 \mathbf{S}_k (\mathbf{Y}_k^T \Delta^2 \mathbf{S}_k)^{-1} \mathbf{Y}_k^T \Delta \mathbf{s}_0.$$

$\tilde{r}(t_k)$ vérifie

$$\tilde{r}(t_k) - \Delta \mathbf{s}_0 \in \text{sev}\{\Delta^2 \mathbf{s}_0, \dots, \Delta^2 \mathbf{s}_{k-1}\}$$

$$\tilde{r}(t_k) \perp \text{sev}\{\mathbf{y}_0, \dots, \mathbf{y}_{k-1}\}.$$

- 1 MÉTHODES D'EXTRAPOLATION VECTORIELLES
- 2 APPLICATIONS À LA RÉOLUTION DE SYSTÈMES LINÉAIRES
 - Application aux méthodes multigrilles
 - Méthodes d'extrapolation et Méthodes de sous-espaces de Krylov
 - Méthode CMRH
- 3 APPLICATIONS À LA RÉOLUTION DE SYSTÈMES NON LINÉAIRES
 - Théorème de convergence
 - Application aux équations de Navier-Stokes
 - Application à l'équation de Schrödinger
- 4 CONCLUSION ET PERSPECTIVES

Considérons le système d'équations linéaires suivant :

$$Ax = b \quad (3)$$

où $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$. Notons s la solution.

Étant donnée une approximation initiale s_0 , on construit la suite (s_k) en posant

$$s_{k+1} = Bs_k + d, \quad k = 0, 1, \dots \quad (4)$$

Si la suite (s_k) est convergente, alors sa limite vaut s .

On appliquera les méthodes d'extrapolation à la suite (s_k) formée par (4).

APPLICATION AUX MÉTHODES MULTIGRILLES :

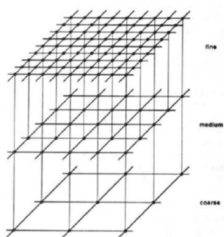
PRINCIPE

Soient s_k , une approximation de s et $e_k = s - s_k$, l'erreur correspondante.

Le résidu est défini par $r_k = b - As_k$. On obtient

$$Ae_k = r_k.$$

La résolution de ce système est aussi difficile que le problème original. On cherche donc une approximation v_k de e_k . Dans ce cas, $v_k + s_k$ sera probablement une meilleure approximation de s . Le principe des méthodes MG est de calculer v_k de façon plus "grossière".



Différentes étapes : Soit s_k l'approximation à l'itération k

- 1 Lissage : $\bar{s}_k = Ls_k$
- 2 Calcul du résidu : $\bar{r}_k = b - A\bar{s}_k$
- 3 Restriction : $\tilde{r}_k = R\bar{r}_k$
- 4 Résolution : $A_C \tilde{e}_k = \tilde{r}_k$
- 5 Prolongation : $\bar{v}_k = P\tilde{e}_k$
- 6 Calcul : $\bar{s}_{k+1} = \bar{v}_k + \bar{s}_k$
- 7 Lissage : $s_{k+1} = L\bar{s}_{k+1}$

L'étape 4 se fait de façon récursive.

- Lisseur : Jacobi, Jacobi relaxée, Gauss-Seidel, Red-Black ordering, ILU, Gradient conjugué
- Relaxation et Prolongation : Linéaire, Bilinéaire, Full Weighting, Partition Fine/Coarse avec une matrice d'influence
- Approximation grille grossière : même discrétisation que pour la grille fine, Approximation de Galerkin ($A_C = RAP$).

La méthode MG génère une suite (s_k) en posant

$$s_{k+1} = Bs_k + d$$

où B est la matrice itérative du processus définie par

$$B = L(I - PA_c^{-1}RA)L$$

$$d = LPA_c^{-1}Rb$$

RÉSULTATS NUMÉRIQUES (MÉTHODE GÉOMÉTRIQUE)

- Discrétisation de l'équation de Helmholtz avec une méthode d'éléments finis sur le carré unité avec des conditions aux bords de Dirichlet

$$\begin{aligned} -\nabla^2 u - \kappa^2 u &= f \text{ dans } \Omega \\ u &= 0 \text{ sur } \partial\Omega \end{aligned}$$

- Grille régulière
- $n = 3969$
- Test avec MGLab1.0 sur Matlab 7.1
- Critère d'arrêt : $\|r_k\| \leq 10^{-10} \|r_0\|$
- b est un vecteur aléatoire
- Vecteur initial $s_0 = 0$
- Lisseur : méthode de Gauss-Seidel
- Restriction : Full Weighting
- Interpolation : Linéaire
- Approximation grille grossière : Méthode d'éléments finis

MÉTHODE MULTIGRILLE GÉOMÉTRIQUE

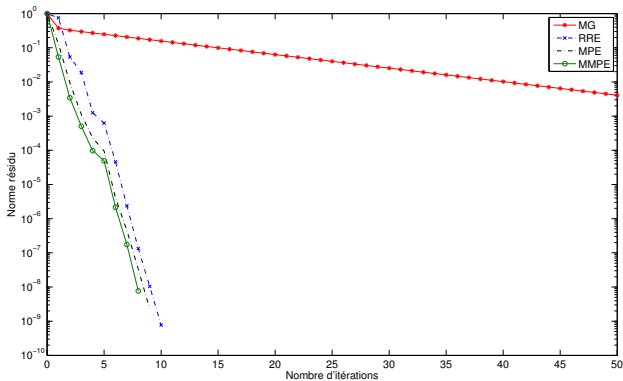
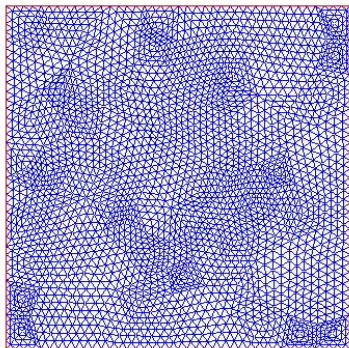


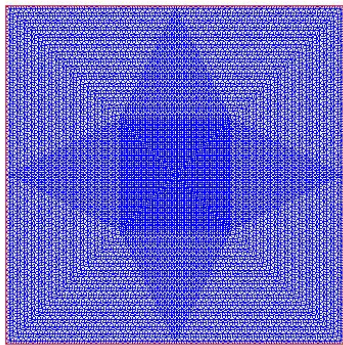
FIGURE : Equation de Helmholtz avec $\kappa = 100$

- Discrétisation de l'équation de Poisson sur le carré unité avec des conditions aux bords de Dirichlet avec une méthode d'éléments finis
- Grille non régulière
- Test avec PDE tool sur Matlab 7.1
- Critère d'arrêt : $\|r_k\| \leq 10^{-10} \|r_0\|$
- b est un vecteur aléatoire
- Vecteur initial $s_0 = 0$
- Lisseur : méthode de Gauss-Seidel symétrique
- Paramètres standards

MÉTHODE MULTIGRILLE ALGÈBRIQUE



(a) $n = 2577$



(b) $n = 8321$

	$n = 2577$		$n = 8321$	
	AMG	RRE	AMG	RRE
Itérations	121	22	195	30
Temps CPU (s)	2.09	0.38	19.21	2.65

Étant donné une approximation s_0 de s et $r_0 = b - Ax_0$ le résidu correspondant. Les méthodes de sous-espace de Krylov construisent une suite d'approximations (s_k) telle que

$$s_k \in s_0 + K_k(A, r_0)$$

$$r_k \perp V_k$$

où

- $V_k = K_k(A, r_0)$ pour la méthode d'Arnoldi
- $V_k = K_k(A, Ar_0)$ pour la méthode GMRES
- $V_k = Y_k$ pour la méthode de Hessenberg

Revenons à la suite formée par

$$s_{k+1} = Bs_k + d.$$

Posons $B = I - A$ et $d = b$.

Le résidu à l'itération k vérifie :

$$r_k = \Delta s_k$$

et

$$\tilde{r}(t_k) = r(t_k).$$

De plus

$$\Delta^2 S_k = -A \Delta S_k.$$

LIEN ENTRE EXTRAPOLATION ET SOUS-ESPACES DE KRYLOV

$r(t_k)$ vérifie les relations suivantes :

- $r(t_k) \in W_k = AV_k$
- $r(t_k) \perp L_k$

où

$$V_k = \text{sev}\{\Delta s_0, \dots, \Delta s_{k-1}\} = K_k(A, r_0)$$

et

$$\begin{array}{lll} L_k & := & V_k \quad \text{pour MPE,} \\ L_k & := & W_k \quad \text{pour RRE,} \\ L_k & := & Y_k \quad \text{pour MMPE.} \end{array}$$

Cela prouve que

MPE est mathématiquement équivalente à Arnoldi

RRE est mathématiquement équivalente à GMRES

MMPE est mathématiquement équivalente à Hessenberg

Étant donnés A , s_0 et r_0 . La méthode de Hessenberg crée une base $\{l_1, \dots, l_k\}$ de $K_k(A, r_0)$. L'algorithme de Hessenberg construit L_k une matrice trapézoïdale inférieure, H_k une matrice de Hessenberg et P une matrice de permutation vérifiant :

- $l_1 = \frac{r_0}{(r_0)_1}$
- $h_{i+1,i} l_{i+1} = A l_i - \sum_{j=1}^i h_{j,i} l_j$ pour $i = 1, \dots, k$

où les $h_{i,k}$ sont calculés de telle sorte que $l_{k+1} \perp e_{\rho(1)}, \dots, e_{\rho(k)}$ et $(l_{k+1})_{\rho(k+1)} = 1$.

La méthode CMRH construit une suite de solutions (s_k) de la forme :

$$s_k = s_0 + L_k d_k$$

où d_k est la solution du problème de moindres carrés :

$$\min_{d \in \mathbb{R}^{k+1}} \|\beta e_1 - H_k d\|$$

PROPOSITION

L'algorithme CMRH donne la solution du système avec le même nombre d'itérations que la méthode GMRES.

THÉORÈME

Soient r_k^G, r_k^C les k^e vecteurs résidu des algorithmes de GMRES et CMRH, respectivement et m le degré du polynôme minimal de la matrice A pour le vecteur r_0^C . Alors, l'itération s_k de la méthode CMRH est bien définie pour $k = 1, \dots, m$ et s_m est la solution exacte. De plus, si $r_0^C = r_0^G$, alors

$$\|r_k^C\| \leq \chi(L_{k+1}) \|r_k^G\|,$$

où $\chi(L_{k+1}) = \|L_{k+1}^+\| \|L_{k+1}\|$.

AVANTAGES DE LA MÉTHODE CMRH

- Coût de la méthode
 - Hessenberg : $\frac{nk^2}{2} - \frac{k^3}{6} + \sum_{i=1}^k n(n-i)$ multiplications
 - Arnoldi : $nk^2 + kn^2$ multiplications
- Coût de stockage :
 - On peut stocker L_k et H_k dans la matrice A : À l'itération m , on a

$$\left(\begin{array}{c|c} H_m & \\ \hline L_m & A_{:,m+1:n} \end{array} \right)$$

ALGORITHME : MÉTHODE DE HESSENBERG AVEC STOCKAGE

- $p = [1, 2, \dots, n]^t$
- On détermine i_0 tel que $|v_{i_0}| = \|v\|_\infty$
- $\beta = v_{i_0}$; $l_1 = v/\beta$; $p_1 \leftrightarrow p_{i_0}$
- $A_{i_0,:} \leftrightarrow A_{1,:}$; $A_{:,i_0} \leftrightarrow A_{:,1}$
- Pour $i = 1, \dots, k$
 - $u = A_{:,i:n} v_{i:n}$; $A_{i+1:n,i} = v_{i+1:n}$
 - Pour $j = 1, \dots, i$
 - $A_{j,i} = u_j$; $u_j = 0$; $u_{j+1:n} = u_{j+1:n} - A_{j,i} A_{j+1:n,j}$
 - Fin Pour
 - On détermine i_0 tel que $|u_{i_0}| = \|u\|_\infty$
 - $(h)_{k+1} = u_{i_0}$; $v = \frac{u}{h_{k+1}}$
 - $p_{i+1} \leftrightarrow p_{i_0}$; $v_{i+1} \leftrightarrow v_{i_0}$
 - $A_{i_0,:} \leftrightarrow A_{i+1,:}$; $A_{:,i_0} \leftrightarrow A_{:,i+1}$
- Fin Pour

- On applique l'algorithme de Hessenberg avec (A, r_0)
- On utilise les rotations de Givens pour éliminer la sous diagonale de H .
- On résout $\bar{H}_k d_k = g_k$ ($\bar{H}_k = \text{triu}(A_{1:k,1:k})$)
- $s_k = s_0 + L_k d_k$, ($L_k = \text{diag}(\text{ones}(k, 1)) + \text{tril}(A_{:,1:k}, -1)$);
- $(s_f)_{p_i} = (s_k)_i$ pour $i = 1, \dots, n$

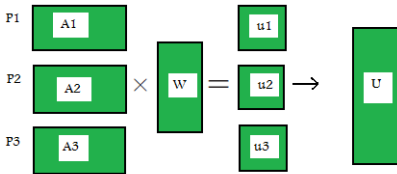
- On utilise la bibliothèque MPI
- On choisit de diviser la matrice en lignes pour chaque processus.

$$\begin{pmatrix} A \end{pmatrix} = \begin{pmatrix} \frac{A_1}{A_{nprocs}} \\ \vdots \end{pmatrix}$$

- Deux étapes importantes dans le programme demandant un temps important :
 - Calcul Matrice-Vecteur
 - Permutations des lignes et des colonnes de A

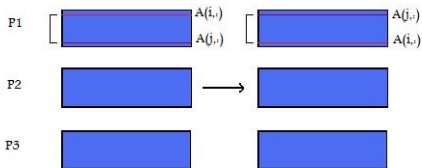
PRODUIT MATRICE-VECTEUR $u = A * w$

Sur chaque processus la matrice est de taille $n_l \times n$. Le vecteur w doit être de taille $n \times 1$. On obtient donc un vecteur u de taille $n_l \times 1$ sur chaque processus. On doit ensuite se ramener à un vecteur de taille $n \times 1$ donc on doit faire des échanges entre processus.



2 cas possibles :

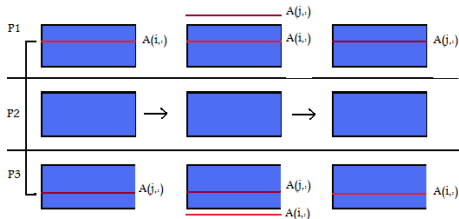
- Si permutations entre deux lignes d'un même processus



PERMUTATIONS

2 cas possibles :

- Si permutations entre deux lignes d'un même processus
- Si permutations entre deux lignes de processus différents



On choisit une matrice de taille $n \times n$ telle que :

$$A_{i,j} = \begin{cases} -\log |z_i - z_j|, & i \neq j, \\ -\log |w_i| & \end{cases} \quad \text{où les } z_i \text{ sont } n \text{ points choisis}$$

aléatoirement sur le carré unité centré en zéro et w_i est un nombre entre $(0, d_i]$ avec d_i la distance entre z_i et son voisin le plus proche.

Cette matrice correspond aux fonctions de Green à une constante $-\frac{1}{2\pi}$ près pour le Laplacien en dimension 2.

On choisit s le vecteur solution tel que :

$$s_i = n - i + 1, \quad \text{pour } i = 1, \dots, n$$

On choisit $n = 32000$.

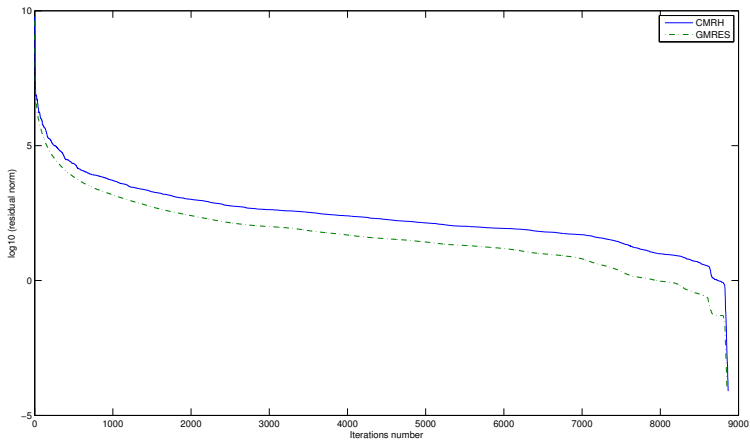


FIGURE : Convergence de la norme du résidu pour la matrice A

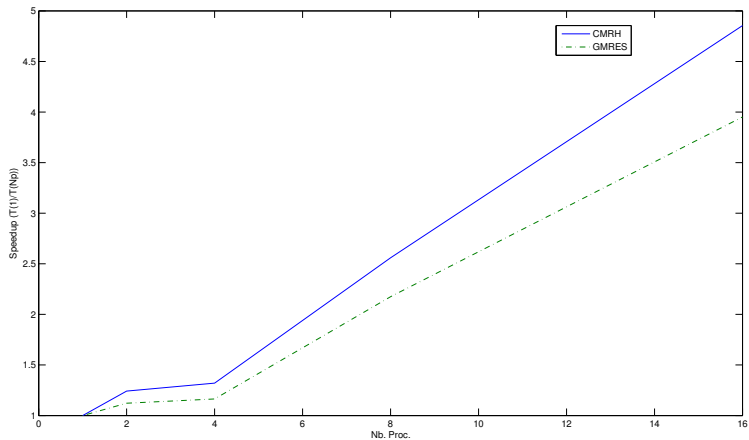


FIGURE : Accélération du temps pour la matrice A

On choisit la matrice B provenant de la formulation intégrale de l'équation de Helmholtz. Cette matrice est non-hermitienne et dense.

On choisit s le vecteur solution tel que :

$$s_i = 1, \text{ pour } i = 1, \dots, n$$

On choisit $n = 20800$ et $\kappa = 1000$.

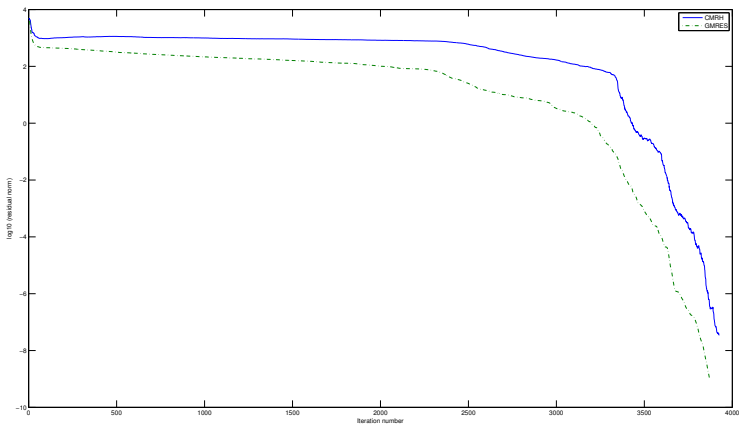


FIGURE : Convergence de la norme du résidu pour la matrice B

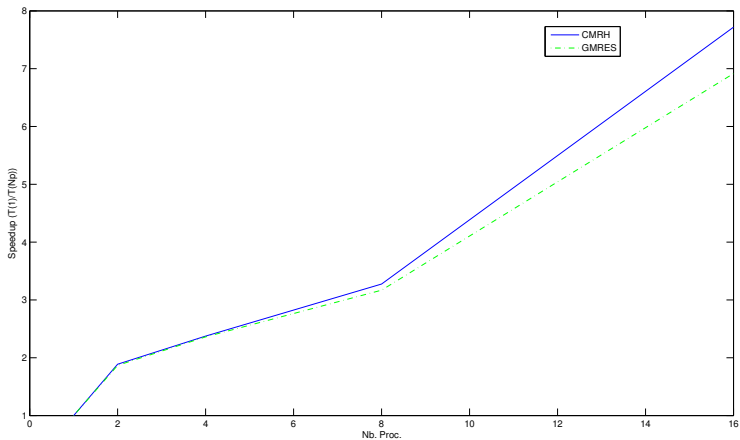


FIGURE : Accélération du temps pour la matrice B

- 1 MÉTHODES D'EXTRAPOLATION VECTORIELLES
- 2 APPLICATIONS À LA RÉOLUTION DE SYSTÈMES LINÉAIRES
 - Application aux méthodes multigrilles
 - Méthodes d'extrapolation et Méthodes de sous-espaces de Krylov
 - Méthode CMRH
- 3 APPLICATIONS À LA RÉOLUTION DE SYSTÈMES NON LINÉAIRES
 - Théorème de convergence
 - Application aux équations de Navier-Stokes
 - Application à l'équation de Schrödinger
- 4 CONCLUSION ET PERSPECTIVES

On considère le système d'équations non linéaires suivant

$$G(x) = x \quad (5)$$

où $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Notons s une solution de (5). Pour un vecteur $x \in \mathbb{R}^n$, le résidu est défini par

$$r(x) = G(x) - x.$$

À partir d'un vecteur initial s_0 , la suite (s_k) est générée comme suit

$$s_{k+1} = G(s_k), \quad k = 0, 1, \dots$$

On remarque

$$r(s_k) = \Delta s_k, \quad k = 0, 1, \dots$$

ALGORITHME DE LA MÉTHODE CYCLIQUE

- 1 $k = 0$, choisir s_0 et un entier q .
- 2 $x_k = s_0$
- 3 Construire $s_{j+1} = G(s_j)$, $j = 0, \dots, q$
- 4 Calculer t_q par RRE, MPE ou MMPE.
- 5 Poser $s_0 = t_q$, $x_{k+1} = t_q$, $k = k + 1$ et retour à l'étape 2.

On pose

$$F(x) = G(x) - x,$$

$$\Phi(x) = F(x) - F'(s)(x - s),$$

$$H_k(x) = \left(\frac{G(x) - x}{\|G(x) - x\|}, \dots, \frac{G^k(x) - G^{k-1}(x)}{\|G^k(x) - G^{k-1}(x)\|} \right),$$

$$C_k(x) = \left(\frac{\phi(G(x)) - \phi(x)}{\|G(x) - x\|}, \dots, \frac{\phi(G^k(x)) - \phi(G^{k-1}(x))}{\|G^k(x) - G^{k-1}(x)\|} \right)$$

et

$$\alpha_k(x) = \sqrt{\det(H_k^*(x)H_k(x))}$$

THÉORÈME DE CONVERGENCE

CONVERGENCE QUADRATIQUE DE LA MÉTHODE RRE

Soit $J = G'(s)$. Supposons que la matrice $J - I$ soit régulière. Supposons que la dérivée de Fréchet G' satisfasse les conditions de Lipschitz

$$\|G'(x) - G'(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{D}, \quad (6)$$

où \mathcal{D} est un sous-espace ouvert et convexe de \mathbb{C}^n . Si

$$\exists \alpha > 0, \exists K, \forall k \geq K : \alpha_{l_k}(x_k) > \alpha, \quad (7)$$

alors il existe un voisinage U de s tel que, pour tout $s_0 \in U$,

$$\|x_{k+1} - s\| \leq C\|x_k - s\|^2. \quad (8)$$

Dans ce théorème, l_k vérifie

$$\|\tilde{r}(t_{l_k}^{RRE})\| = 0.$$

Pour $i = 0, \dots, l_k$, on a

$$\Delta s_i = G^{i+1}(x_k) - G^i(x_k), \quad \Delta^2 s_i = F(G^{i+1}(x_k)) - F(G^i(x_k)).$$

Posons

$$\begin{aligned} B_{l_k} &= F'(s)^{-1}(\Delta^2 S_{l_k} - F'(s)\Delta S_{l_k})\Delta S_{l_k}^+ \\ &= F'(s)^{-1}C_{l_k}(x_k)H_{l_k}^+(x_k) \end{aligned}$$

On montre que

$$\|B_{l_k}\| \leq \frac{\sqrt{l_k}}{\alpha_{l_k}(x_k)} C_1 \max_{0 \leq i \leq l_k} \|G^i(x_k) - s\| \leq \frac{C_2}{\alpha} \|x_k - s\|.$$

Il existe donc un voisinage de s tel que $\|B_{l_k}\| \leq 1$. On a donc $I - B_{l_k}$ régulière. On pose $D_{l_k} = I - (I - B_{l_k})^{-1}$.

Revenons à l'algorithme :

$$\begin{aligned} x_{k+1} &= x_k - \Delta S_{l_k} \Delta^2 S_{l_k}^+ \Delta s_0 \\ &= x_k - (I - D_{l_k}) F'(s)^{-1} \Delta^2 S_{l_k} \Delta^2 S_{l_k}^+ \Delta s_0 \end{aligned}$$

L'hypothèse $\|\tilde{r}(t_{l_k}^{RRE})\| = 0$ permet d'obtenir

$$\begin{aligned} x_{k+1} &= x_k - (I - D_{l_k}) F'(s) \Delta s_0 \\ &= x_k - (I - D_{l_k}) F'(s) F(x_k) \end{aligned}$$

D'où

$$x_{k+1} - s = (D_{l_k} - I) F'(s)^{-1} \phi(x_k) + D_{l_k} (x_k - s)$$

En majorant chaque membre, on obtient

$$\|x_{k+1} - s\| \leq C \|x_k - s\|^2.$$

On considère les équations de Navier-Stokes stationnaires modélisant des fluides incompressibles :

$$\begin{cases} u \cdot \nabla u - \nu \nabla^2 u + \nabla p = f & \text{dans } \Omega \subset \mathbb{R}^2 \\ \nabla \cdot u = 0 \end{cases} \quad (9)$$

où $\nu \ll 1$ est une constante donnée.

CONDITIONS AUX BORDS

$$u = w \text{ on } \partial\Omega_D, \nu \frac{\partial u}{\partial n} - \vec{n}p = 0 \text{ on } \partial\Omega_N$$

Soient

$$H_E^1 := \left\{ u \in \mathcal{H}^1(\Omega)^2 \mid u = w \text{ sur } \partial\Omega_D \right\},$$

$$H_{E_0}^1 := \left\{ v \in \mathcal{H}^1(\Omega)^2 \mid v = 0 \text{ sur } \partial\Omega_D \right\}.$$

FORMULATION FAIBLE

$$\nu \int_{\Omega} \nabla u \cdot \nabla v + \int_{\Omega} (u \cdot \nabla u) \cdot v - \int_{\Omega} p \cdot \nabla v = \int_{\Omega} f \cdot v \quad \forall v \in H_{E_0}^1$$

$$\int_{\Omega} q \cdot \nabla u = 0 \quad \forall q \in L_2(\Omega).$$

LINÉARISATION

Etant donnée une solution initiale $(u_0, p_0) \in H_E^1 \times L_2(\Omega)$. On cherche $(\delta u_k, \delta p_k) \in H_{E_0}^1 \times L_2(\Omega)$ satisfaisant

$$D(u_k, \delta u_k, v) + \nu \int_{\Omega} \nabla \delta u_k : \nabla v - \int_{\Omega} \delta p_k \cdot \nabla v = R_k(v)$$
$$\int_{\Omega} q \cdot \nabla \delta u_k = r_k(q)$$

et on pose $u_{k+1} = u_k + \delta u_k$ et $p_{k+1} = p_k + \delta p_k$. Le couple $(R_k(v), r_k(q))$ représente le résidu à l'étape k défini par

$$R_k(v) = \int_{\Omega} f \cdot v - \int_{\Omega} (u_k \cdot \nabla u_k) \cdot v - \nu \int_{\Omega} \nabla u_k : \nabla v + \int_{\Omega} p_k \cdot \nabla v,$$

$$r_k(q) = - \int_{\Omega} q \cdot \nabla u_k$$

$D(u_k, \delta u_k, v)$ représente la différence des termes non linéaires

$$D(u_k, \delta u_k, v) = \int_{\Omega} (\delta u_k \cdot \nabla \delta u_k) \cdot v + \int_{\Omega} (\delta u_k \cdot \nabla u_k) \cdot v + \int_{\Omega} (u_k \cdot \nabla \delta u_k) \cdot v.$$

$$D(u_k, \delta u_k, v) = \int_{\Omega} (\delta u_k \cdot \nabla \delta u_k) \cdot v + \int_{\Omega} (\delta u_k \cdot \nabla u_k) \cdot v + \int_{\Omega} (u_k \cdot \nabla \delta u_k) \cdot v.$$

MÉTHODE DE NEWTON

Étant donné u^0 , on calcule u^1, u^2, \dots, u^k via

$$\begin{cases} u^{k+1} \cdot \nabla u^k - u^k \cdot \nabla u^k + u^k \cdot \nabla u^{k+1} - \nu \nabla^2 u^{k+1} + \nabla p^{k+1} = f \\ \nabla \cdot u^{k+1} = 0 \end{cases}$$

$$D(u_k, \delta u_k, v) = \int_{\Omega} (\delta u_k \cdot \nabla \delta u_k) \cdot v + \int_{\Omega} (\delta u_k \cdot \nabla u_k) \cdot v + \int_{\Omega} (u_k \cdot \nabla \delta u_k) \cdot v$$

MÉTHODE DE PICARD

Étant donné u^0 , on calcule u^1, u^2, \dots, u^k via

$$\begin{cases} u^k \cdot \nabla u^{k+1} - \nu \nabla^2 u^{k+1} + \nabla p^{k+1} = f \\ \nabla \cdot u^{k+1} = 0 \end{cases}$$

FORMULATION MATRICIELLE :

Picard :

$$\begin{pmatrix} \nu A + N_k & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \delta u_k \\ \delta p_k \end{pmatrix} = \begin{pmatrix} R_k \\ r_k \end{pmatrix}$$

Newton :

$$\begin{pmatrix} \nu A + N_k + W_k & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \delta u_k \\ \delta p_k \end{pmatrix} = \begin{pmatrix} R_k \\ r_k \end{pmatrix}$$

où

- $A_{ij} = (\nabla \Phi_i, \nabla \Phi_j)$ diffusion
- $B_{ij} = -(\nabla \Phi_j, \Psi_i)$ divergence
- $N_{ij} = (u_h \cdot \nabla \Phi_i, \Phi_j)$ convection
- $W_{ij} = (\Phi_i \cdot \nabla u_h, \Phi_j)$ Newton

RÉSULTATS NUMÉRIQUES

On considère le cas d'un fluide s'écoulant dans un tunnel contenant un obstacle.

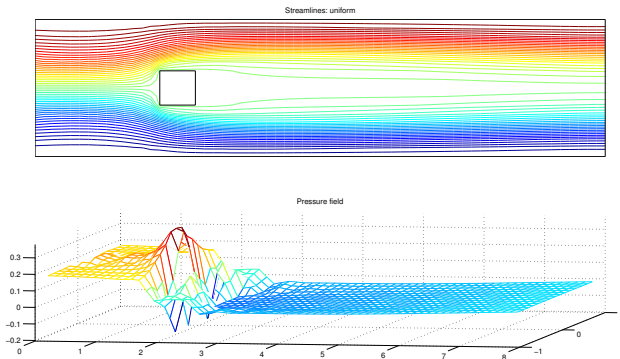
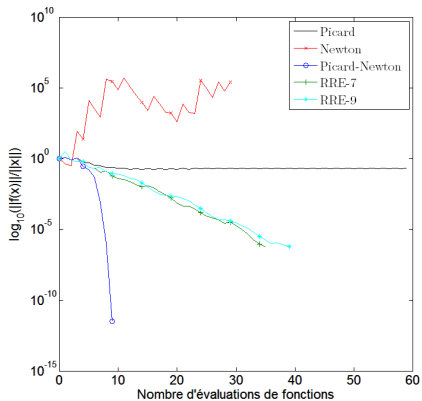
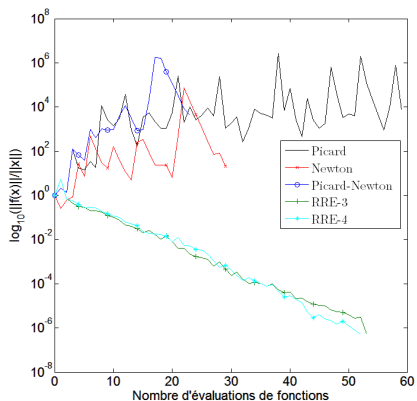


FIGURE : Solution et rendu 3D de la pression

RÉSULTATS NUMÉRIQUES



(a) $\nu = 1/600$



(b) $\nu = 1/800$

EQUATION DE SCHRÖDINGER

On considère la formulation Kohn-Sham de l'équation de Schrödinger :

$$H(\rho(\mathbf{x}))\psi(\mathbf{x}) = \left[\frac{-1}{2} \nabla^2 + V_{tot}[\rho(\mathbf{x}), \mathbf{x}] \right] \psi(\mathbf{x}) = E\psi(\mathbf{x}),$$

où

- ρ la densité de charge
- V_{tot} le potentiel total
- E l'énergie totale
- ψ une fonction d'onde

⇒ Problème aux valeurs propres non linéaire

MÉTHODE SCF (SELF CONSISTENT FIELD)

PRINCIPE DE LA MÉTHODE SCF

- Approximation initiale $V_{in} = V(\rho)$
- Résolution de $H(\rho)\psi = \left[\frac{-1}{2}\nabla^2 + V_{in}\right]\psi = E\psi$
- Calcul de $\rho_{new} = \mathit{diag}(\psi\psi^T)$
- Calcul de $V_{out} = V(\rho_{new})$

MÉTHODES DE RÉOLUTION

- Point fixe : $V_{k+1} = V_k - \omega(V_k - F(V_k))$
- Méthode de Broyden généralisée
- Méthode d'Anderson

MÉTHODE DE BROYDEN GÉNÉRALISÉE (FANG ET SAAD, '07)

On suppose connues $s_{k-m}, \dots, s_k \in \mathbb{R}^n$, $f_{k-m}, \dots, f_k \in \mathbb{R}^n$ et J_{k-m} l'approximation de la Jacobienne de F au point s_{k-m} . On pose

$$\Delta S_k = [\Delta s_{k-m}, \dots, \Delta s_{k-1}]$$

$$\Delta F_k = [\Delta f_{k-m}, \dots, \Delta f_{k-1}].$$

On cherche J_k une approximation de la Jacobienne de F au point s_k satisfaisant

$$J_k \Delta S_k = \Delta F_k$$

et

$$(J_{k-m} - J_k)q = 0 \quad \forall q \perp \text{Im}(\Delta S_k).$$

Le nouvel itéré est calculé en posant

$$s_{k+1} = s_k + J_{k-m} f_k + (\Delta F_k - J_{k-m} \Delta S_k) \Delta S_k^+ f_k.$$

MÉTHODE DE ANDERSON (ANDERSON, '65)

On suppose connues

$$s_{k-m}, \dots, s_k \in \mathbb{R}^n$$

et

$$f_{k-m}, \dots, f_k \in \mathbb{R}^n$$

On pose

$$\Delta S_k = [\Delta s_{k-m}, \dots, \Delta s_{k-1}]$$

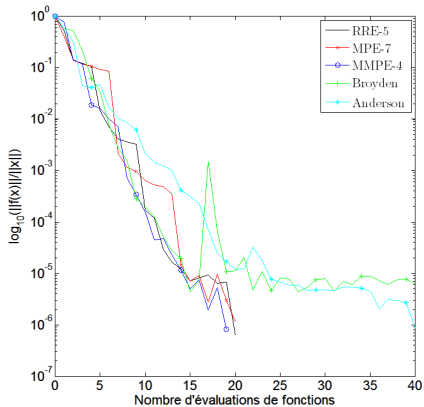
et

$$\Delta F_k = [\Delta f_{k-m}, \dots, \Delta f_{k-1}].$$

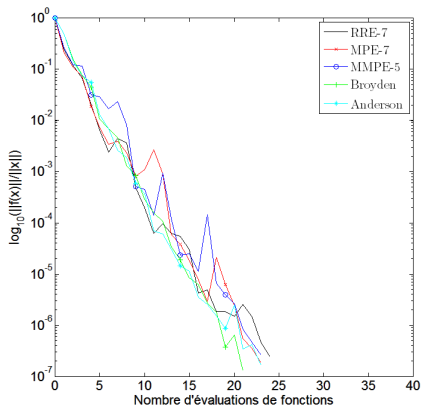
Le nouvel itéré est calculé en posant

$$s_{k+1} = s_k + \beta f_k - (\Delta S_k + \beta \Delta F_k)(\Delta F_k^T \Delta F_k)^{-1} \Delta F_k^T f_k$$

RÉSULTATS NUMÉRIQUES (PROCESSUS CYCLIQUE)



(a) C $\omega = 0.5$



(b) SiNa $\omega = 0.3$

- 1 MÉTHODES D'EXTRAPOLATION VECTORIELLES
- 2 APPLICATIONS À LA RÉOLUTION DE SYSTÈMES LINÉAIRES
 - Application aux méthodes multigrilles
 - Méthodes d'extrapolation et Méthodes de sous-espaces de Krylov
 - Méthode CMRH
- 3 APPLICATIONS À LA RÉOLUTION DE SYSTÈMES NON LINÉAIRES
 - Théorème de convergence
 - Application aux équations de Navier-Stokes
 - Application à l'équation de Schrödinger
- 4 CONCLUSION ET PERSPECTIVES

● Conclusion

- Méthodes d'extrapolation permettent d'accélérer la convergence dans le cas de problèmes linéaires et non linéaires
- Application à l'équation de Navier-Stokes pour des cas avec forte viscosité
- Application à l'équation de Schrödinger et amélioration des résultats en comparaison avec d'autres méthodes d'accélération
- Application aux méthodes MG et nette accélération de la convergence dans les cas étudiés
- Etude la méthode CMRH en parallèle et comparaison avec la méthode GMRES

● Perspectives

- Nouveaux résultats de convergence
- Stabilité dans le cas linéaire
- Application dans de nombreux autres domaines
- Application de la méthode CMRH parallèle avec plus de processeurs ou sur GPU.

Merci de votre attention