

Approximation polynômiale par moindres carrés

Gérard MEURANT

6 février 2008

- 1 Introduction
- 2 Solution du problème de moindres carrés
- 3 Construction de la matrice de Jacobi
- 4 Augmentation
- 5 Diminution
- 6 Calcul du vecteur propre
- 7 Exemple : fenêtre glissante
- 8 Conclusions

Introduction

- ▶ Etant donnés des nœuds t_j , des poids w_j^2 et des valeurs y_j , $j = 1, \dots, m$ trouver un polynôme q de degré $n \leq m$ qui minimise

$$\sum_{j=1}^m (y_j - q(t_j))^2 w_j^2$$

Introduction

- ▶ Etant donnés des nœuds t_j , des poids w_j^2 et des valeurs y_j , $j = 1, \dots, m$ trouver un polynôme q de degré $n \leq m$ qui minimise

$$\sum_{j=1}^m (y_j - q(t_j))^2 w_j^2$$

- ▶ Les poids et les nœuds définissent un produit scalaire

$$\langle f, g \rangle_m = \sum_{j=1}^m f(t_j)g(t_j)w_j^2$$

Introduction

- ▶ Etant donnés des nœuds t_j , des poids w_j^2 et des valeurs y_j , $j = 1, \dots, m$ trouver un polynôme q de degré $n \leq m$ qui minimise

$$\sum_{j=1}^m (y_j - q(t_j))^2 w_j^2$$

- ▶ Les poids et les nœuds définissent un produit scalaire

$$\langle f, g \rangle_m = \sum_{j=1}^m f(t_j)g(t_j)w_j^2$$

- ▶ La solution q peut être obtenue en utilisant les polynômes orthogonaux p_k associés à ce produit scalaire

Introduction (2)

- ▶ Les polynômes p_k vérifient une relation de récurrence à 3 termes

$$\beta_{k+1}p_{k+1}(\lambda) = (\lambda - \alpha_k)p_k(\lambda) - \beta_k p_{k-1}(\lambda), \quad k = 0, 1, \dots$$

Introduction (2)

- ▶ Les polynômes p_k vérifient une relation de récurrence à 3 termes

$$\beta_{k+1}p_{k+1}(\lambda) = (\lambda - \alpha_k)p_k(\lambda) - \beta_k p_{k-1}(\lambda), \quad k = 0, 1, \dots$$

- ▶ Ceci nous donne des matrices, dites de (Jacobi), J_k avec pour coefficients α_i, β_i

Introduction (2)

- ▶ Les polynômes p_k vérifient une relation de récurrence à 3 termes

$$\beta_{k+1}p_{k+1}(\lambda) = (\lambda - \alpha_k)p_k(\lambda) - \beta_k p_{k-1}(\lambda), \quad k = 0, 1, \dots$$

- ▶ Ceci nous donne des matrices, dites de (Jacobi), J_k avec pour coefficients α_i, β_i
- ▶ On montre que les nœuds t_j sont les valeurs propres de J_m et les w_j sont les premiers éléments des vecteurs propres normalisés

Problème inverse

- ▶ Ici, on connaît t_j et w_j et l'on veut calculer la matrice J_m à partir des données spectrales

Problème inverse

- ▶ Ici, on connaît t_j et w_j et l'on veut calculer la matrice J_m à partir des données spectrales
- ▶ Il s'agit d'un problème inverse difficile

Solution du problème de moindres carrés

- ▶ Notons les valeurs propres de J_m : $\theta_j^{(m)} = t_j$ (valeurs de Ritz)

Solution du problème de moindres carrés

- ▶ Notons les valeurs propres de J_m : $\theta_j^{(m)} = t_j$ (valeurs de Ritz)
- ▶ $p^m(\lambda) = (p_0(\lambda) \ \cdots \ p_{m-1}(\lambda))^T$
- ▶ On a $\lambda p^m(\lambda) = J_m p^m(\lambda) + \beta_{m-1} p_m(\lambda) e^m$

Solution du problème de moindres carrés

- ▶ Notons les valeurs propres de J_m : $\theta_j^{(m)} = t_j$ (valeurs de Ritz)
- ▶ $p^m(\lambda) = (p_0(\lambda) \ \cdots \ p_{m-1}(\lambda))^T$
- ▶ On a $\lambda p^m(\lambda) = J_m p^m(\lambda) + \beta_{m-1} p_m(\lambda) e^m$
- ▶ Les zéros de p_m sont les valeurs propres de J_m

Solution du problème de moindres carrés

- ▶ Notons les valeurs propres de J_m : $\theta_j^{(m)} = t_j$ (valeurs de Ritz)
- ▶ $p^m(\lambda) = (p_0(\lambda) \ \cdots \ p_{m-1}(\lambda))^T$
- ▶ On a $\lambda p^m(\lambda) = J_m p^m(\lambda) + \beta_{m-1} p_m(\lambda) e^m$
- ▶ Les zéros de p_m sont les valeurs propres de J_m
- ▶ $p^m(\theta_i^{(m)})$ est un vecteur propre non normalisé de J_m

Solution du problème de moindres carrés

- ▶ Notons les valeurs propres de J_m : $\theta_j^{(m)} = t_j$ (valeurs de Ritz)
- ▶ $p^m(\lambda) = (p_0(\lambda) \ \cdots \ p_{m-1}(\lambda))^T$
- ▶ On a $\lambda p^m(\lambda) = J_m p^m(\lambda) + \beta_{m-1} p_m(\lambda) e^m$
- ▶ Les zéros de p_m sont les valeurs propres de J_m
- ▶ $p^m(\theta_i^{(m)})$ est un vecteur propre non normalisé de J_m
- ▶ $P_m = (p^m(\theta_1^{(m)}) \ \cdots \ p^m(\theta_m^{(m)}))$

Solution du problème de moindres carrés

- ▶ Notons les valeurs propres de J_m : $\theta_j^{(m)} = t_j$ (valeurs de Ritz)
- ▶ $p^m(\lambda) = (p_0(\lambda) \ \cdots \ p_{m-1}(\lambda))^T$
- ▶ On a $\lambda p^m(\lambda) = J_m p^m(\lambda) + \beta_{m-1} p_m(\lambda) e^m$
- ▶ Les zéros de p_m sont les valeurs propres de J_m
- ▶ $p^m(\theta_i^{(m)})$ est un vecteur propre non normalisé de J_m
- ▶ $P_m = (p^m(\theta_1^{(m)}) \ \cdots \ p^m(\theta_m^{(m)}))$
- ▶ $J_m P_m = P_m \Theta_m$, Θ_m matrice diagonale des valeurs propres et $P_m^T P_m = D_m^{-2}$ matrice diagonale

Solution du problème de moindres carrés (2)

- ▶ La solution q s'écrit $q(t^m) = P_m^T d^m, t^m = (t_j)_{j=1, \dots, m}$

Solution du problème de moindres carrés (2)

- ▶ La solution q s'écrit $q(t^m) = P_m^T d^m$, $t^m = (t_j)_{j=1, \dots, m}$
- ▶ et $\sum_{j=1}^m (y_j - q(t_j))^2 w_j^2 = \|P_m D_m^2 y^m - d^m\|^2$

Solution du problème de moindres carrés (2)

- ▶ La solution q s'écrit $q(t^m) = P_m^T d^m$, $t^m = (t_j)_{j=1,\dots,m}$
- ▶ et $\sum_{j=1}^m (y_j - q(t_j))^2 w_j^2 = \|P_m D_m^2 y^m - d^m\|^2$
- ▶ donc $d^m = P_m D_m^2 y^m$ (interpolation) et

$$d^n = P_{n,m} D_m^2 y^m, \quad n < m$$

Augmentation et diminution

- ▶ Supposant que l'on connaisse la solution jusqu'à l'indice m

Augmentation et diminution

- ▶ Supposant que l'on connaisse la solution jusqu'à l'indice m
- ▶ **Augmentation** : ajouter un nouveau triplet de données $\{t_{m+1}, w_{m+1}, y_{m+1}\}$ et calculer la solution sans repartir de zéro

Augmentation et diminution

- ▶ Supposant que l'on connaisse la solution jusqu'à l'indice m
- ▶ **Augmentation** : ajouter un nouveau triplet de données $\{t_{m+1}, w_{m+1}, y_{m+1}\}$ et calculer la solution sans repartir de zéro
- ▶ **Diminution** : supprimer un triplet de données

Augmentation et diminution

- ▶ Supposant que l'on connaisse la solution jusqu'à l'indice m
- ▶ **Augmentation** : ajouter un nouveau triplet de données $\{t_{m+1}, w_{m+1}, y_{m+1}\}$ et calculer la solution sans repartir de zéro
- ▶ **Diminution** : supprimer un triplet de données
- ▶ Problème : il faut calculer les polynômes orthogonaux connaissant t_j and w_j

Augmentation et diminution

- ▶ Supposant que l'on connaisse la solution jusqu'à l'indice m
- ▶ **Augmentation** : ajouter un nouveau triplet de données $\{t_{m+1}, w_{m+1}, y_{m+1}\}$ et calculer la solution sans repartir de zéro
- ▶ **Diminution** : supprimer un triplet de données
- ▶ Problème : il faut calculer les polynômes orthogonaux connaissant t_j and w_j
- ▶ Il faut augmenter ou diminuer la matrice de **Jacobi** donnant les coefficients de la récurrence à 3 termes

Construction de la matrice de Jacobi J_m

- ▶ On peut le faire avec les algorithmes de **Stieltjes** ou de **Lanczos** pour une matrice diagonale (nœuds) et un vecteur de départ (poids), pb : stabilité?

Construction de la matrice de Jacobi J_m

- ▶ On peut le faire avec les algorithmes de **Stieltjes** ou de **Lanczos** pour une matrice diagonale (nœuds) et un vecteur de départ (poids), pb : stabilité?
- ▶ Algorithmes de reconstruction de J_m depuis les données spectrales :

Construction de la matrice de Jacobi J_m

- ▶ On peut le faire avec les algorithmes de **Stieltjes** ou de **Lanczos** pour une matrice diagonale (nœuds) et un vecteur de départ (poids), pb : stabilité?
- ▶ Algorithmes de reconstruction de J_m depuis les données spectrales :
 - ▶ **De Boor and Golub**, 1978

Construction de la matrice de Jacobi J_m

- ▶ On peut le faire avec les algorithmes de **Stieltjes** ou de **Lanczos** pour une matrice diagonale (nœuds) et un vecteur de départ (poids), pb : stabilité?
- ▶ Algorithmes de reconstruction de J_m depuis les données spectrales :
 - ▶ De Boor and Golub, 1978
 - ▶ Gragg and Harrod, 1984

Construction de la matrice de Jacobi J_m

- ▶ On peut le faire avec les algorithmes de **Stieltjes** ou de **Lanczos** pour une matrice diagonale (nœuds) et un vecteur de départ (poids), pb : stabilité?
- ▶ Algorithmes de reconstruction de J_m depuis les données spectrales :
 - ▶ De Boor and Golub, 1978
 - ▶ Gragg and Harrod, 1984
 - ▶ Gautschi

Construction de la matrice de Jacobi J_m

- ▶ On peut le faire avec les algorithmes de **Stieltjes** ou de **Lanczos** pour une matrice diagonale (nœuds) et un vecteur de départ (poids), pb : stabilité?
- ▶ Algorithmes de reconstruction de J_m depuis les données spectrales :
 - ▶ De Boor and Golub, 1978
 - ▶ Gragg and Harrod, 1984
 - ▶ Gautschi
 - ▶ Reichel, 1991 1993

Construction de la matrice de Jacobi J_m

- ▶ On peut le faire avec les algorithmes de **Stieltjes** ou de **Lanczos** pour une matrice diagonale (nœuds) et un vecteur de départ (poids), pb : stabilité?
- ▶ Algorithmes de reconstruction de J_m depuis les données spectrales :
 - ▶ De Boor and Golub, 1978
 - ▶ Gragg and Harrod, 1984
 - ▶ Gautschi
 - ▶ Reichel, 1991 1993
 - ▶ Laurie, 2000

Augmentation par rotations

Considérée par Elhay, Golub and Kautsky (1991)

Theorem

Soit $\sigma_m = (w_1^2 + \dots + w_m^2)^{1/2}$. La solution est donnée par

$$J_{m+1} = Q \begin{pmatrix} J_m & 0 \\ 0 & t_{m+1} \end{pmatrix} Q^T$$

$$\sigma_{m+1} = (\sigma_m^2 + w_{m+1}^2)^{1/2}$$

$$d^{m+1} = Q \begin{pmatrix} d^m \\ w_{m+1} y_{m+1} \end{pmatrix}$$

Q matrice orthogonale déterminée de façon unique en demandant que J_{m+1} soit tridiagonale et Q telle que

$$Q(\sigma_m e^1 + w_{m+1} e^{m+1}) = \sigma_{m+1} e^1$$

Augmentation par rotations (2)

- ▶ $Q = R_m R_{m-1} \cdots R_1$, R_j : rotation entre les lignes j et $m + 1$

Augmentation par rotations (2)

- ▶ $Q = R_m R_{m-1} \cdots R_1$, R_j : rotation entre les lignes j et $m+1$
- ▶ R_1 est calculée pour avoir $Q(\sigma_m e^1 + w_{m+1} e^{m+1}) = \sigma_{m+1} e^1$

Augmentation par rotations (2)

- ▶ $Q = R_m R_{m-1} \cdots R_1$, R_j : rotation entre les lignes j et $m+1$
- ▶ R_1 est calculée pour avoir $Q(\sigma_m e^1 + w_{m+1} e^{m+1}) = \sigma_{m+1} e^1$
- ▶ Ceci crée des éléments non nuls dans la dernière ligne (et la dernière colonne) que l'on chasse par d'autres rotations R_2, \dots, R_m pour avoir J_{m+1} tridiagonale

Diminution

- ▶ C'est similaire à annuler un poids. Ca peut être fait en utilisant des rotations hyperboliques

$$\begin{pmatrix} c & is \\ -is & c \end{pmatrix},$$

avec $c^2 - s^2 = 1$

Diminution

- ▶ C'est similaire à annuler un poids. Ca peut être fait en utilisant des rotations hyperboliques

$$\begin{pmatrix} c & is \\ -is & c \end{pmatrix},$$

avec $c^2 - s^2 = 1$

- ▶ Ca ne marche pas très bien... (voir exemple)

Example

$$t_j = -1 + 2(j-1)/(N-1), w_j = 1/\sqrt{N}, y_j = 1.5 + \sin(4t_j), j = 1, \dots, N$$

Exemple

$$t_j = -1 + 2(j-1)/(N-1), w_j = 1/\sqrt{N}, y_j = 1.5 + \sin(4t_j), j = 1, \dots, N$$

Partant de t_1, w_1 on construit récursivement les matrices de **Jacobi** en ajoutant un nœud à chaque étape jusqu'à N

Exemple

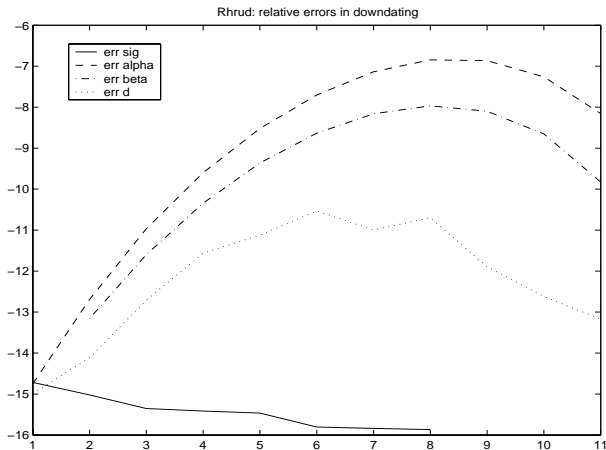
$$t_j = -1 + 2(j-1)/(N-1), w_j = 1/\sqrt{N}, y_j = 1.5 + \sin(4t_j), j = 1, \dots, N$$

Partant de t_1, w_1 on construit récursivement les matrices de **Jacobi** en ajoutant un nœud à chaque étape jusqu'à N

Ensuite, on diminue en supprimant un nœud à chaque étape et on compare les coefficients $(\alpha, \beta, \sigma, d)$ avec ce que l'on a obtenu lors de la construction par augmentation

La figure suivante se lit de droite à gauche !

Augmentation avec rot. – Diminution avec rot. hyp.



Exemple 1 : \log_{10} des erreurs relatives lors de la diminution,
 $N = 12$

Rotations inverses

Theorem (Elhay, Golub, Kautsky)

Etant donnés σ_{m+1} , J_{m+1} et d^{m+1} , la solution pour supprimer $\{t_{m+1}, w_{m+1}, y_{m+1}\}$ est

$$\begin{pmatrix} \tilde{J}_m & 0 \\ 0 & t_{m+1} \end{pmatrix} = QJ_{m+1}Q^T$$

$$\tilde{\sigma}_m = (\sigma_m^2 - w_{m+1}^2)^{1/2}, \quad \begin{pmatrix} \tilde{d}^m \\ w_{m+1}y_{m+1} \end{pmatrix} = Qd^{m+1}$$

où la matrice orthogonale Q est déterminée de façon unique en demandant que \tilde{J}_m soit tridiagonale et telle que

$$Q(\sigma_{m+1}e^1 - w_{m+1}q) = \tilde{\sigma}_m e^1$$

q est un vecteur propre normalisé de J_{m+1} correspondant à t_{m+1} normalisé pour avoir le premier élément positif

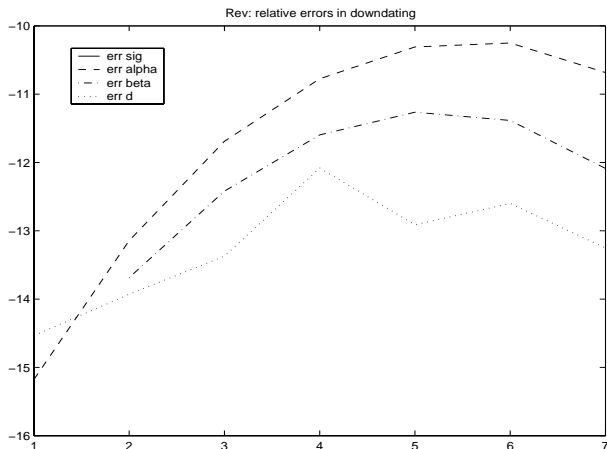
- ▶ Ca semble intéressant... Cependant, il faut connaître le vecteur propre correspondant à la valeur propre (noeud) à supprimer

- ▶ Ca semble intéressant... Cependant, il faut connaître le vecteur propre correspondant à la valeur propre (noeud) à supprimer
- ▶ Elhay, Golub and Kautsky utilisent un vecteur propre q calculé en résolvant un système linéaire avec J_{m+1}

- ▶ Ca semble intéressant... Cependant, il faut connaître le vecteur propre correspondant à la valeur propre (noeud) à supprimer
- ▶ Elhay, Golub and Kautsky utilisent un vecteur propre q calculé en résolvant un système linéaire avec J_{m+1}
- ▶ Ca ne marche pas bien...

- ▶ Ca semble intéressant... Cependant, il faut connaître le vecteur propre correspondant à la valeur propre (noeud) à supprimer
- ▶ Elhay, Golub and Kautsky utilisent un vecteur propre q calculé en résolvant un système linéaire avec J_{m+1}
- ▶ Ca ne marche pas bien...
- ▶ Tous ces calculs s'arrêtent (sqrt de nombres négatifs) même avec n petit

Augmentation avec rot. – Diminution avec REV.



Exemple 1 : \log_{10} des erreurs relatives lors de la diminution, $N = 8$

Calcul du vecteur propre

- ▶ Le vecteur propre calculée par **EGK** n'est pas assez précis

Calcul du vecteur propre

- ▶ Le vecteur propre calculée par **EGK** n'est pas assez précis
- ▶ Quand on ajoute un nœud à chaque étape et qu'ensuite on les supprime, on peut calculer les vecteurs propres de façon incrémentale en utilisant les rotations **Q**

Calcul du vecteur propre

- ▶ Le vecteur propre calculée par **EGK** n'est pas assez précis
- ▶ Quand on ajoute un nœud à chaque étape et qu'ensuite on les supprime, on peut calculer les vecteurs propres de façon incrémentale en utilisant les rotations **Q**
- ▶ Les résultats sont bien meilleurs...

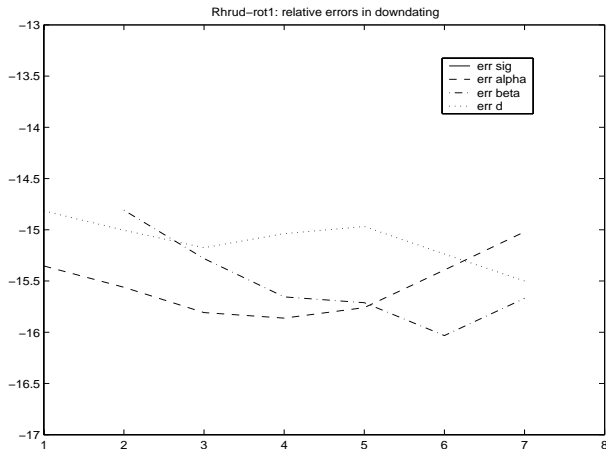
Calcul du vecteur propre

- ▶ Le vecteur propre calculée par **EGK** n'est pas assez précis
- ▶ Quand on ajoute un nœud à chaque étape et qu'ensuite on les supprime, on peut calculer les vecteurs propres de façon incrémentale en utilisant les rotations **Q**
- ▶ Les résultats sont bien meilleurs...
- ▶ On peut utiliser de grands ensembles de données sans problème

Calcul du vecteur propre

- ▶ Le vecteur propre calculée par **EGK** n'est pas assez précis
- ▶ Quand on ajoute un nœud à chaque étape et qu'ensuite on les supprime, on peut calculer les vecteurs propres de façon incrémentale en utilisant les rotations **Q**
- ▶ Les résultats sont bien meilleurs...
- ▶ On peut utiliser de grands ensembles de données sans problème
- ▶ Mais, c'est trop cher...

Augmentation avec rot. – Diminution avec v.p.



Exemple 1 : \log_{10} des erreurs relatives lors de la diminution, $N = 8$

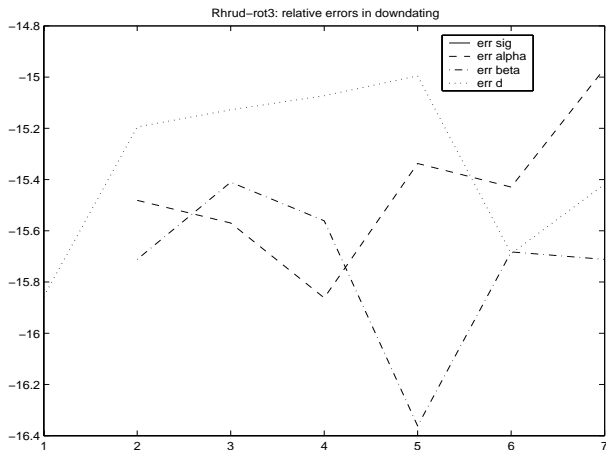
Calcul du vecteur propre (2)

- ▶ Une autre possibilité est d'utiliser ce qui est fait dans MR^3 ([Dhillon-Parlett](#)) pour calculer un vecteur propre précis

Calcul du vecteur propre (2)

- ▶ Une autre possibilité est d'utiliser ce qui est fait dans MR^3 ([Dhillon-Parlett](#)) pour calculer un vecteur propre précis
- ▶ Les résultats sont encore meilleurs...

Augmentation avec rot. – Diminution avec MR^3



Exemple 1 : \log_{10} des erreurs relatives lors de la diminution, $N = 8$

Exemple : fenêtre glissante

- ▶ Supposons que l'on connaisse les données pour $j = 1, \dots, N$

Exemple : fenêtre glissante

- ▶ Supposons que l'on connaisse les données pour $j = 1, \dots, N$
- ▶ Soit $Y_k = \{t_j, w_j, y_j\}_{j=k}^{k+M-1}$, $k = 1, \dots, N - M + 1$

Exemple : fenêtre glissante

- ▶ Supposons que l'on connaisse les données pour $j = 1, \dots, N$
- ▶ Soit $Y_k = \{t_j, w_j, y_j\}_{j=k}^{k+M-1}$, $k = 1, \dots, N - M + 1$
- ▶ On calcule une solution moindres carrés de dimension $n \leq M$ pour Y_1

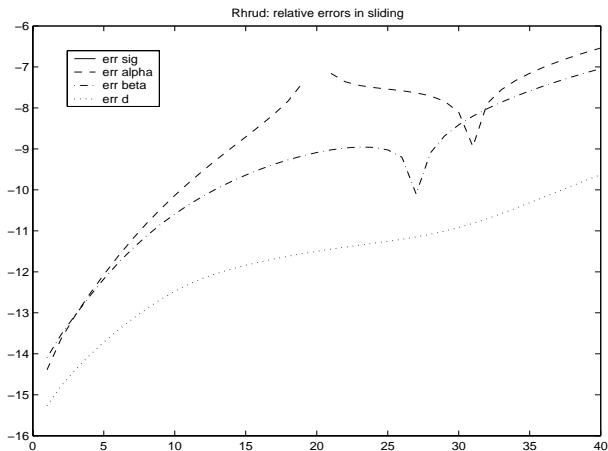
Exemple : fenêtre glissante

- ▶ Supposons que l'on connaisse les données pour $j = 1, \dots, N$
- ▶ Soit $Y_k = \{t_j, w_j, y_j\}_{j=k}^{k+M-1}$, $k = 1, \dots, N - M + 1$
- ▶ On calcule une solution moindres carrés de dimension $n \leq M$ pour Y_1
- ▶ Pour $k = 2, \dots, N - M + 1$ on ajoute à droite $\{t_{k+M}, w_{k+M}, y_{k+M}\}$ et on supprime à gauche $\{t_{k-1}, w_{k-1}, y_{k-1}\}$

Exemple : fenêtre glissante

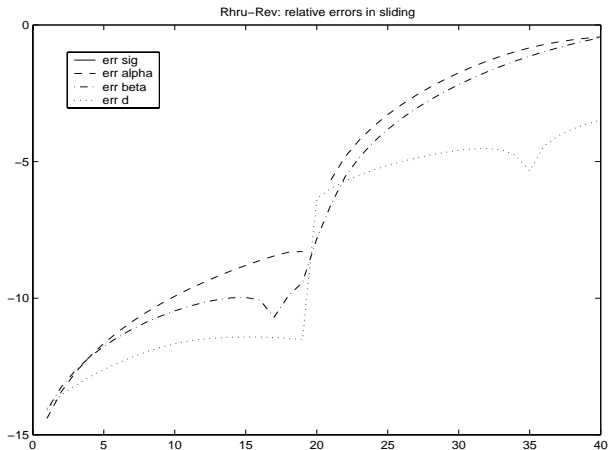
- ▶ Supposons que l'on connaisse les données pour $j = 1, \dots, N$
- ▶ Soit $Y_k = \{t_j, w_j, y_j\}_{j=k}^{k+M-1}$, $k = 1, \dots, N - M + 1$
- ▶ On calcule une solution moindres carrés de dimension $n \leq M$ pour Y_1
- ▶ Pour $k = 2, \dots, N - M + 1$ on ajoute à droite $\{t_{k+M}, w_{k+M}, y_{k+M}\}$ et on supprime à gauche $\{t_{k-1}, w_{k-1}, y_{k-1}\}$
- ▶ On compare à la solution calculée directement (avec RHR) pour Y_k

Augmentation – Diminution avec RHR



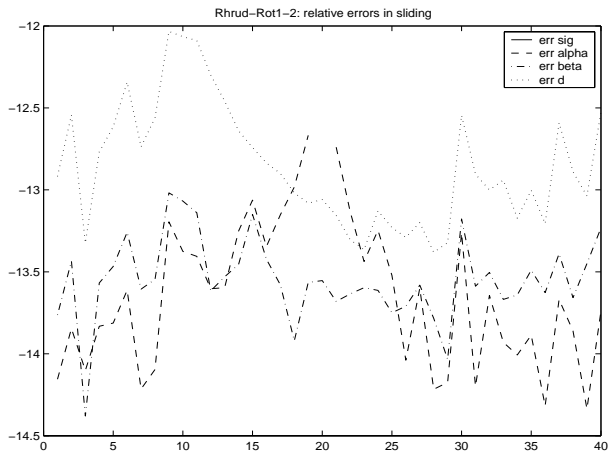
Exemple 1 : \log_{10} des erreurs relatives avec la fenêtre glissante,
 $N = 50, M = 10, n = 5$

Augmentation avec rot. – Diminution avec REV



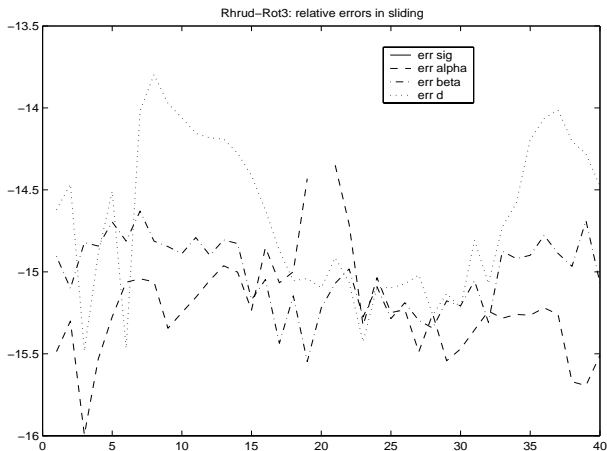
Exemple 1 : \log_{10} des erreurs relatives avec la fenêtre glissante,
 $N = 50, M = 10, n = 5$

Augmentation avec rot. – Diminution avec vecteurs propres



Exemple 1 : \log_{10} des erreurs relatives avec la fenêtre glissante,
 $N = 50, M = 10, n = 5$

Augmentation avec rot. – Diminution avec MR^3



Exemple 1 : \log_{10} des erreurs relatives avec la fenêtre glissante,
 $N = 50, M = 10, n = 5$

Conclusions

- ▶ Les méthodes d'augmentation par rotations fonctionnent bien

Conclusions

- ▶ Les méthodes d'augmentation par rotations fonctionnent bien
- ▶ Pour la diminution, il faut être attentif au calcul du vecteur propre

Conclusions

- ▶ Les méthodes d'augmentation par rotations fonctionnent bien
- ▶ Pour la diminution, il faut être attentif au calcul du vecteur propre
- ▶ En utilisant les modifications proposées, de grands ensembles de données peuvent être traités de façon incrémentale

Gene H. Golub (1932-2007)

