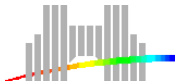


# Résoudre et certifier la solution d'un système linéaire

**NGUYEN Hong Diep**  
**Encadrants : Gilles Villard, Nathalie Revol**

EVA-Flo - Saclay 2008

8 avril 2008



- 1 Problème
- 2 Précisions utilisées
- 3 Résultats expérimentaux
- 4 Inversion d'une matrice mal conditionnée

- 1 Problème
- 2 Précisions utilisées
- 3 Résultats expérimentaux
- 4 Inversion d'une matrice mal conditionnée

## Objectif :

- ➊ Résoudre un système linéaire mal conditionné
  - $A \in \mathbb{R}^{n \times n}$
  - $b \in \mathbb{R}^n$
  - Trouver un vecteur  $\tilde{x} \in \mathbb{R}^n$  tel que :  $A \times \tilde{x} \approx b$
- ➋ Borner l'erreur de la solution trouvée en même temps en utilisant l'arithmétique par intervalle.
  - $\Delta x = x^* - \tilde{x}$
  - Trouver un intervalle petit  $[e]$  contenant  $\Delta x$ .

**Environnement :** Bibliothèques `mpfr` et `mpfi`.

**Notations** :  $x^*$  la solution exacte,  $\tilde{x}$  approximation flottante,  $\mathbf{x}$  un intervalle contenant la solution

- 1 Calculer une approximation  $R$  de la matrice inverse de  $A$ .
- 2 Approximation du résultat  $\tilde{x} = fl(R \times b)$
- 3 Utiliser l'arithmétique par intervalle pour calculer le résidu du système  
$$\mathbf{r} = b - A \times \tilde{x} \quad \Rightarrow \quad A \times \Delta x \in \mathbf{r}$$
- 4 Utiliser  $R$  comme une matrice de préconditionnement  
$$\mathbf{RA} = R * A \quad , \quad \mathbf{k} = R * \mathbf{r}$$
- 5 Soit  $\mathbf{e}$  un vecteur par intervalle tel que  
$$\mathbf{RA} \times \mathbf{e} = \mathbf{k} \quad \Rightarrow \quad \Delta x \in \mathbf{e}$$
- 6  $\mathbf{x} = \tilde{x} + \mathbf{e} \ni x^*$
- 7 Calculer une nouvelle approximation  $\tilde{x} = mid(\mathbf{x})$ .  
Retourner à l'étape 3

Soient  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , et  $\mathbf{b} \in \mathbb{R}^n$

$$\mathbf{A} \times \mathbf{x} = \mathbf{b}$$

**Hypothèse** :  $\mathbf{A}$  est une H-matrice.

Il existe un vecteur  $u > 0$  tel que  $v = \langle \mathbf{A} \rangle \times u > 0$ .

**Proposition [4, p. 121]** :  $\mathbf{A}^H \mathbf{b} \subseteq \|\mathbf{b}\|_v \times [-u, u]$

Soient :

- $\|\mathbf{b}\|_v = \max\{|\mathbf{b}_i|/u_i \mid i = 1, \dots, n\}$
- $\mathbf{A}^H$  une enveloppe convexe de l'inverse de la matrice  $\mathbf{A}$

Approximation initiale pour la solution :

$$\mathbf{x}_0 = \|\mathbf{b}\|_v \times [-u, u]$$

Méthodes :

- Krawczyk
- Gauss-Seidel
- Élimination de Gauss par intervalle

La méthode de Gauss-Seidel est prouvée converger plus rapidement dans le cas où la matrice  $\mathbf{A}$  est une H-matrice.

$$y_i := \left( \mathbf{b}_i - \sum_{k < i} \mathbf{A}_{ik} y_k - \sum_{k > i} \mathbf{A}_{ik} x_k \right) / \mathbf{A}_{ii} \cap x_i \quad (i = 1, \dots, n)$$

- 1 Problème
- 2 Précisions utilisées**
- 3 Résultats expérimentaux
- 4 Inversion d'une matrice mal conditionnée

- 1 Calculer une approximation  $R$  de la matrice inverse de  $A$ .
- 2 Approximation du résultat  $\tilde{x} = fl(R \times b)$
- 3 Utiliser l'arithmétique par intervalle pour calculer le résidu du système  
$$\mathbf{r} = b - A \times \tilde{x} \quad \Rightarrow \quad A \times \Delta \mathbf{x} \in \mathbf{r}$$
- 4 Utiliser  $R$  comme une matrice de préconditionnement  
$$\mathbf{R}\mathbf{A} = R * A \quad , \quad \mathbf{k} = R * \mathbf{r}$$
- 5 Soit  $\mathbf{e}$  un vecteur par intervalle tel que  
$$\mathbf{R}\mathbf{A} \times \mathbf{e} = \mathbf{k} \quad \Rightarrow \quad \Delta \mathbf{x} \in \mathbf{e}$$
- 6  $\mathbf{x} = \tilde{x} + \mathbf{e} \ni x^*$
- 7 Calculer une nouvelle approximation  $\tilde{x} = mid(\mathbf{x})$ .  
Retourner à l'étape 3

- 1 Calculer une approximation  $R$  de la matrice inverse de  $A$ .
- 2 Approximation du résultat  $\tilde{x} = fl(R \times b)$
- 3 **Utiliser l'arithmétique par intervalle pour calculer le résidu du système**  
$$r = b - A \times \tilde{x} \quad \Rightarrow \quad A \times \Delta x \in r$$
- 4 Utiliser  $R$  comme une matrice de préconditionnement  
$$RA = R * A \quad , \quad k = R * r$$
- 5 Soit  $e$  un vecteur par intervalle tel que  
$$RA \times e = k \quad \Rightarrow \quad \Delta x \in e$$
- 6  $x = \tilde{x} + e \ni x^*$
- 7 Calculer une nouvelle approximation  $\tilde{x} = mid(x)$ .  
Retourner à l'étape 3

$$\mathbf{r} = \mathbf{b} - \mathbf{A} * \tilde{\mathbf{x}}$$

Le résidu calculé est lié directement à la qualité du résultat :

$$\frac{\|\Delta \mathbf{e}\|}{\|\mathbf{e}\|} \leq \kappa(\mathbf{A}) \times \frac{\|\Delta \mathbf{r}\|}{\|\mathbf{r}\|}$$

**Historiquement** : Utiliser la double précision pour le calcul du résidu.

**Expérimental** : Utiliser une précision un peu plus grande :

$$p_{\text{residu}} = p_A + p_{\tilde{\mathbf{x}}} + \log_2 n + 16$$

- 1 Calculer une approximation  $R$  de la matrice inverse de  $A$ .
- 2 Approximation du résultat  $\tilde{x} = fl(R \times b)$
- 3 Utiliser l'arithmétique par intervalle pour calculer le résidu du système  
$$r = b - A \times \tilde{x} \quad \Rightarrow \quad A \times \Delta x \in r$$
- 4 **Utiliser  $R$  comme une matrice de préconditionnement**  
$$RA = R * A \quad , \quad k = R * r$$
- 5 Soit  $e$  un vecteur par intervalle tel que  
$$RA \times e = k \quad \Rightarrow \quad \Delta x \in e$$
- 6  $x = \tilde{x} + e \ni x^*$
- 7 Calculer une nouvelle approximation  $\tilde{x} = mid(x)$ .  
Retourner à l'étape 3

$$RA = R * A$$

**Conditionnement** :  $\kappa(A) > \mathbf{u}^{-1} \Rightarrow \kappa(R * A) \gg 1$ .

Higham :  $\|\Delta RA\|_p \leq \gamma_n \times \|A\|_p \times \|R\|_p = \frac{n\mathbf{u}}{1-n\mathbf{u}} \times \|A\|_p \times \|R\|_p$

**Nombre d'opérations** : entre  $2n^3$  et  $4n^3$ .

Soit  $p_{cond}$  la précision utilisée

- Si  $p_{cond} < p_A + p_R$ , nombre d'opérations  $\approx 4n^3$ .
- Si  $p_{cond} \gg p_A + p_R$ , nombre d'opérations  $\approx 2n^3$ .

**Expérimentaux** :  $p_{cond} = p_A + p_R + \log_2(n) + 16$ .

- 1 Calculer une approximation  $R$  de la matrice inverse de  $A$ .
- 2 Approximation du résultat  $\tilde{x} = fl(R \times b)$
- 3 Utiliser l'arithmétique par intervalle pour calculer le résidu du système  
$$\mathbf{r} = \mathbf{b} - A \times \tilde{x} \quad \Rightarrow \quad A \times \Delta \mathbf{x} \in \mathbf{r}$$
- 4 Utiliser  $R$  comme une matrice de préconditionnement  
$$\mathbf{RA} = R * A \quad , \quad \mathbf{k} = R * \mathbf{r}$$
- 5 **Soit  $\mathbf{e}$  un vecteur par intervalle tel que**  
$$\mathbf{RA} \times \mathbf{e} = \mathbf{k} \quad \Rightarrow \quad \Delta \mathbf{x} \in \mathbf{e}$$
- 6  $\mathbf{x} = \tilde{x} + \mathbf{e} \ni x^*$
- 7 Calculer une nouvelle approximation  $\tilde{x} = mid(\mathbf{x})$ .  
Retourner à l'étape 3

Si **RA** est une H-matrice, la méthode de Gauss-Seidel converge rapidement.  
⇒ Utiliser un petit nombre d'itérations (implémentation : **5**).

Après chaque raffinement par Gauss-Seidel :  $\text{rad}(\mathbf{x})$  diminue. D'où

$$\begin{aligned}\text{mid}(\mathbf{x}) &\rightarrow \mathbf{x}^* \\ b - A * \text{mid}(\mathbf{x}) &\rightarrow 0\end{aligned}$$

Il faut augmenter la précision pour le calcul du résidu.

**Expérimental** : Augmenter la précision du résidu par un mot (32 bits) après chaque itération extérieure.

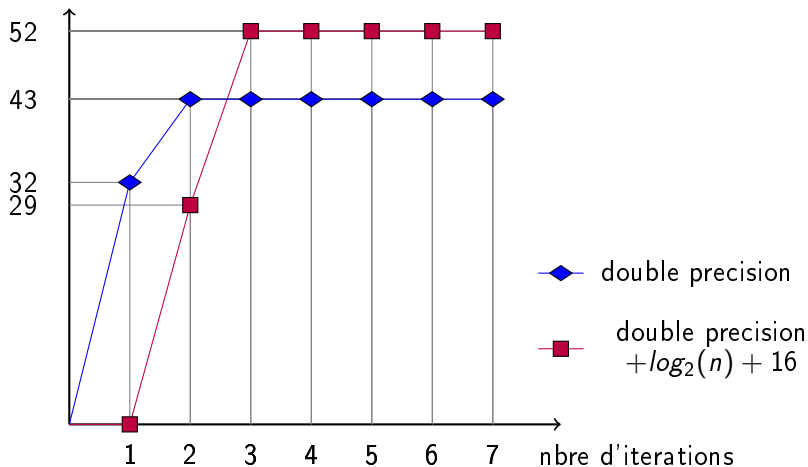
# Plan de l'exposé

- 1 Problème
- 2 Précisions utilisées
- 3 Résultats expérimentaux**
- 4 Inversion d'une matrice mal conditionnée

# Résultats expérimentaux

$A$  est une matrice d'Hilbert de taille  $200 \times 200$  :  $A(i,j) = 1/(i+j+1)$   
 $b$  est un vecteur de 200 éléments :  $b = A * [1, 2, \dots, 200]^T$

nbre de chiffres corrects



- 1 Problème
- 2 Précisions utilisées
- 3 Résultats expérimentaux
- 4 Inversion d'une matrice mal conditionnée

**But** : Trouver une approximation  $R$  de  $A$  telle que  $\mathbf{RA}$  est une H-matrice. Si  $A$  est une matrice mal conditionnée ( $\kappa(A) > u^{-1}$ ) il est difficile de trouver une bonne approximation.

$$R = (A + \Delta A)^{-1}$$
$$\frac{\|R - A^{-1}\|}{\|A^{-1}\|} \leq \kappa(A) \times \Delta A$$

La précision courante n'est pas suffisante  $\Rightarrow$  Il faut augmenter la précision pour calculer la matrice inverse.

**Question** : Quelle précision est suffisante ?

Le conditionnement de LU-factorisation :

$$\begin{aligned}A + \Delta A &= L \times U \\ |\Delta A| &\leq c_{LU} |L| \times |U|\end{aligned}$$

Donc

$$|R - A^{-1}| \leq c_{LU} |A^{-1}| \times |L| \times |U| \times |R|$$

## Expérimentaux :

- Doubler la précision de calcul jusqu'à ce qu'on puisse obtenir une bonne approximation (**RA** est une H-matrice).
- Utiliser la précision courante pour calculer une première approximation de  $A$ . Utiliser la précision de  $\log_2(\|A\| \times \|R_0\|) + p_A/2$  pour calculer la matrice inverse de  $A$ . Pas de preuve!

# Inversion de matrice : Méthode de Rump

Méthode de Rump : méthode itérative en utilisant la précision étendue (sans renormalisation).

## Algorithme 1 (Méthode de Rump)

```
 $\tilde{S}_0 = A + \Delta A$       % perturbation pour A  
 $X_0 = \text{inv}(\tilde{S}_0)$ ;  $R_1 = X_0$ ;  
for  $k = 1 : K$   
     $C = R_{k-1} * A$ ;  
     $\tilde{S}_k = C + \Delta C$       % perturbation pour C  
     $X_k = \text{inv}(\tilde{S}_k)$ ;  
     $R_{k+1} = I * R_k$ ;      % (k+1)-fold accuracy  
end
```




Décroissance du conditionnement : Si  $\kappa(R_k) \geq \mathbf{u}^{-1}$

$$\kappa(R_{k+1}) = \mathcal{O}(\sqrt{\mathbf{u}})\kappa(R_k) + \mathcal{O}(1)$$


Tester avec des matrices d'Hilbert

Dimension	Double précision $+ \log_2(n) + 16$	Méthode de Rump
$15 \times 15$	2.1144867e-7   86	3.280362093072142e-015   106
$50 \times 50$	1.3968208e-3   92	6.514514405350780e-013   106
$100 \times 100$	8.8455157e-2   93	1.743405295129286e-009   106
$200 \times 200$	7.6330220e-2   94	2.201670061541261e-010   106
$300 \times 300$	4.1865923e-2   97	1.389205833701723e-009   106

$\|I - \mathbf{RA}\|_1$  | nombre de bits

-  Takeshi Ogita, Siegfried M. Rump, and Shin'ichi Oishi.  
Accurate sum and dot product.  
*SIAM J. Sci. Comput.*, 26(6) :1955–1988, 2005.
-  Siegfried M. Rump,  
Computer-assisted Proofs and Self-validating Methods  
*MicroHandbook on Accuracy and Reliability in Scientific Computation*  
*Bo Einarsson*, (p.195-240).
-  Shin'ichi Oishi, Kuino Tanabe, Takeshi Ogita and Siegfried M. Rump.  
Convergence of Rump's Method for Inverting Arbitrarily Ill-Conditioned  
Matrices  
*J. Comput. Appl. Math.*, 205(1) :533–544, 2007.

 Arnold Neumaier  
Interval Methods for Systems of Equations  
*Cambridge University Press, 1990.*

 Nicholas J. Higham  
Accuracy and Stability of Numerical Algorithms  
*SIAM, 2002.*