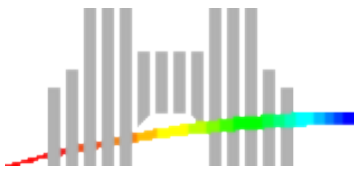
Résoudre et certifier la solution d'un système linéaire

Nguyễn Hồng Diệp

Journées Nationales de Calcul Formel (JNCF)
CIRM, Luminy, Marseille

23 octobre 2008


Problème

Objectif :

1. Résoudre un système linéaire
   - $A \in \mathbb{F}^{n \times n}$
   - $b \in \mathbb{F}^n$
   - Trouver un vecteur $\widetilde{x} \in \mathbb{F}^n$ qui approche x^* : $A * x^* = b$
2. Borner l'erreur de la solution trouvée en même temps en utilisant l'arithmétique par intervalle
   - $\Delta x = x^* - \widetilde{x}$
   - Trouver un petit intervalle **e** contenant Δx.

Méthode classique de raffinement itératif

Wilkinson, Li & Demmel, Higham, ...

Algorithme 1 (Méthode classique de raffinement itératif)

Entrées : $A \in \mathbb{F}^{n \times n}, b \in \mathbb{F}^n$

$\widetilde{x} = A \setminus b$;

while(test d'arrêt)

$\widetilde{r} = b - A * \widetilde{x}$;

$\widetilde{e} = A \setminus \widetilde{r}$;

$\widetilde{x} = \widetilde{x} + \widetilde{e}$;

end

Passer à l'arithmétique par intervalle

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

Algorithme 2 (Méthode de raffinement itératif par intervalle)

Entrées : $A \in \mathbb{F}^{n\times n}, b \in \mathbb{F}^n$

$\widetilde{x} = A \setminus b$;

while(test d'arrêt)

$\quad \boldsymbol{r} = [b - A * \widetilde{x}]$; % $A * (x^* - \widetilde{x}) \in \boldsymbol{r}$

$\quad \boldsymbol{e} = A \setminus \boldsymbol{r}$; % $x^* - \widetilde{x} \in \boldsymbol{e} \quad \rightarrow \quad x^* \in \widetilde{x} + \boldsymbol{e}$

$\quad \widetilde{x} = \widetilde{x} + mid(\boldsymbol{e})$;

end

Système résidu

$$A * \mathbf{e} = \mathbf{r}$$

$\mathbf{r}$ doit être calculé avec une précision étendue (précision doublée)

Méthode : Raffinement itératif par intervalle

- ne chercher pas à résoudre directement un système par intervalle.
- considérer ce système comme des contraintes pour raffiner la borne d'erreur

$$\begin{aligned} \mathbf{e'} &= \mathbf{e} \cap (A \setminus \mathbf{r}) \\ &= \{\widetilde{e} \in \mathbf{e} \quad | \quad \exists \widetilde{r} \in \mathbf{r} : \quad A * \widetilde{e} = \widetilde{r}\} \end{aligned}$$

Méthode détaillée I

Entrées : $A \in \mathbb{F}^{n \times n}, b \in \mathbb{F}^n$

1. Calculer la décomposition LU de A
2. Calculer une approximation flottante $\widetilde{x}$ du système
3. Calculer une matrice R telle que $L * R * U \simeq I$
4. Préconditionner la matrice des coefficients par R : $\mathbf{K} = [R * A]$
5. Tester si $\mathbf{K}$ est une H-matrice
   Si $\mathbf{K}$ n'est pas une H-matrice, alors échec de certification de la solution

Méthode détaillée II

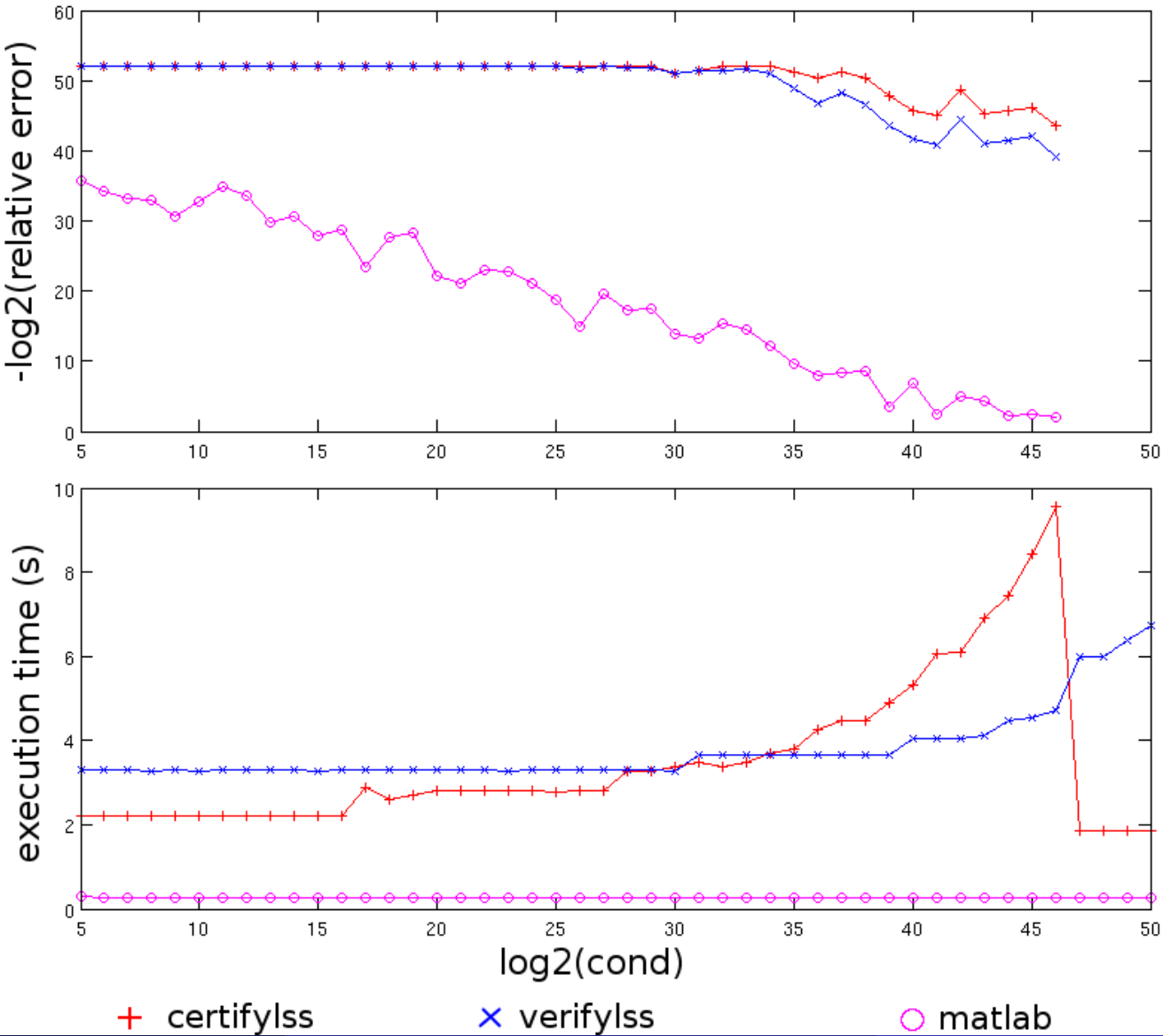
6. Calculer le résidu : $\mathbf{r} = [b - A * \widetilde{x}]$ avec une précision doublée
7. Préconditionner le résidu par R : $\mathbf{z} = R * \mathbf{r}$
8. Calculer une approximation initiale de l'erreur **e0**
9. while (not convergence)
   - Appliquer au maximum 10 itérations de Gauss-Seidel sur **K**, **z**, **e0**
   - Mettre à jour l'approximation flottante et la borne d'erreur :

$$\widetilde{x} = \widetilde{x} + \mathtt{mid}(\mathbf{e0}) \qquad \mathbf{e0} = \mathbf{e0} - \mathtt{mid}(\mathbf{e0})$$

   - Recalculer le résidu et **z**.

Résultats expérimentaux

$A = \texttt{gallery}('\texttt{randsvd}', 1000, cond)$ $\quad b = [1, \ldots, 1]^T$



Plan de l'exposé

Plan de l'exposé

Solution initiale (1)

Principe :

Si la matrice A a de bonnes propriétés : elle est facilement inversible
On peut la préconditionner par C et obtenir une matrice proche de l'Identité : diagonale positive et dominante par exemple.

$\longrightarrow$ alors on peut trouver un encadrement de la solution.

Solution initiale (2)

Proposition (simplifiée) de Neumaier [1] : Soient $A \in \mathbb{R}^{n \times n}$ et $C \in \mathbb{R}^{n \times n}$. Si $\langle [CA] \rangle u \geq v > 0$ pour $u \geq 0$ alors

$$A^{-1}\mathbf{b} \subseteq \|C\mathbf{b}\|_v * [-u, u].$$

Soient :

- $\|C\mathbf{b}\|_v = \max\{|C\mathbf{b}_i|/v_i \quad | \quad i = 1, \ldots, n\}$
- $\langle [CA] \rangle$ la matrice de comparaison de $[CA]$

$$\begin{aligned} \langle [CA] \rangle_{i,i} &= min(|[CA]_{i,i}|) \\ \langle [CA] \rangle_{i,j \neq i} &= -max(|[CA]_{i,j}|) \end{aligned}$$

1. page 121, Arnold Neumaier, *Interval Methods for Systems of Equations*

Solution initiale (3)

Utiliser comme la valeur de $C : R \approx inv(A) \Rightarrow CA = RA \quad (\approx \mathtt{I})$

$u = [1, \ldots, 1]^T \Rightarrow \langle [RA] \rangle * u \geq v > 0.$

Hypothèse : $[RA]$ est une H-matrice.

Approximation initiale pour la borne d'erreur :

$$\begin{aligned} \mathbf{e} &\subseteq A^{-1}\mathbf{r} \\ &\subseteq \underbrace{\|R * \mathbf{r}\|_v * [-u, u]}_{\mathbf{e0}} \end{aligned}$$

Pré-conditionner : $\mathbf{K} = [R * A] \qquad \mathbf{z} = R * \mathbf{r}$

$$\begin{aligned} A * \mathbf{e} &= \mathbf{r} \\ \Rightarrow \mathbf{K} * \mathbf{e} &= \mathbf{z} \end{aligned}$$

Plan de l'exposé

Raffinement par intervalle

Méthodes :

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

La méthode de Gauss-Seidel est prouvée converger plus rapidement dans le cas où la matrice K est une H-matrice (cf. Théorème de Neumaier [2])

2. page 138, Arnold Neumaier, *Interval Methods for Systems of Equations*

Itérations par intervalle de Gauss-Seidel

$$(\mathbf{K} \setminus \mathbf{z}) \cap \mathbf{e} = \{\widetilde{e} \in \mathbf{e} \quad | \quad \exists \widetilde{K} \in \mathbf{K}, \exists \widetilde{z} \in \mathbf{z} : \quad \widetilde{K} * \widetilde{e} = \widetilde{z}\}$$

$$\widetilde{K}_{i,1}\widetilde{e}_1 + \ldots + \widetilde{K}_{i,i-1}\widetilde{e}_{i-1} + \widetilde{K}_{i,i}\widetilde{e}_i + \widetilde{K}_{i,i+1}\widetilde{e}_{i+1} + \ldots + \widetilde{K}_{i,n}\widetilde{e}_n = z_i$$

$$\widetilde{e}_i = \left(\widetilde{z}_i - \sum_{j=1}^{i-1} \widetilde{K}_{i,j}\widetilde{e}_j - \sum_{j=i+1}^{n} \widetilde{K}_{i,j}\widetilde{e}_j\right) \Big/ \widetilde{K}_{i,i}$$

$$\widetilde{e}_i \in \left(z_i - \sum_{j=1}^{i-1} K_{i,j}e'_j - \sum_{j=i+1}^{n} K_{i,j}e_j\right) \Big/ K_{i,i} \cap e_i = e'_i$$

Itérations par intervalle de Gauss-Seidel

$$(\mathbf{K} \setminus \mathbf{z}) \cap \mathbf{e} = \{\widetilde{e} \in \mathbf{e} \quad | \quad \exists \widetilde{K} \in \mathbf{K}, \exists \widetilde{z} \in \mathbf{z} : \quad \widetilde{K} * \widetilde{e} = \widetilde{z}\}$$

$$\widetilde{K}_{i,1}\widetilde{e}_1 + \ldots + \widetilde{K}_{i,i-1}\widetilde{e}_{i-1} + \widetilde{K}_{i,i}\widetilde{e}_i + \widetilde{K}_{i,i+1}\widetilde{e}_{i+1} + \ldots + \widetilde{K}_{i,n}\widetilde{e}_n = z_i$$

$$\widetilde{e}_i = \left(\widetilde{z}_i - \sum_{j=1}^{i-1} \widetilde{K}_{i,j}\widetilde{e}_j - \sum_{j=i+1}^{n} \widetilde{K}_{i,j}\widetilde{e}_j\right) \Big/ \widetilde{K}_{i,i}$$

$$\widetilde{e}_i \in \left(\mathbf{z}_i - \sum_{j=1}^{i-1} \mathbf{K}_{i,j}\mathbf{e}'_j - \sum_{j=i+1}^{n} \mathbf{K}_{i,j}\mathbf{e}_j\right) \Big/ \mathbf{K}_{i,i} \quad \cap \mathbf{e}_i \quad = \mathbf{e}'_i$$

Itérations par intervalle de Gauss-Seidel

$$(\mathbf{K} \setminus \mathbf{z}) \cap \mathbf{e} = \{\widetilde{e} \in \mathbf{e} \quad | \quad \exists \widetilde{K} \in \mathbf{K}, \exists \widetilde{z} \in \mathbf{z} : \quad \widetilde{K} * \widetilde{e} = \widetilde{z}\}$$

$$\widetilde{K}_{i,1}\widetilde{e}_1 + \ldots + \widetilde{K}_{i,i-1}\widetilde{e}_{i-1} + \widetilde{K}_{i,i}\widetilde{e}_i + \widetilde{K}_{i,i+1}\widetilde{e}_{i+1} + \ldots + \widetilde{K}_{i,n}\widetilde{e}_n = z_i$$

$$\widetilde{e}_i = \left(\widetilde{z}_i - \sum_{j=1}^{i-1} \widetilde{K}_{i,j}\widetilde{e}_j - \sum_{j=i+1}^{n} \widetilde{K}_{i,j}\widetilde{e}_j\right) \Big/ \widetilde{K}_{i,i}$$

$$\widetilde{e}_i \in \left(\mathbf{z}_i - \sum_{j=1}^{i-1} \mathbf{K}_{i,j}\mathbf{e}'_j - \sum_{j=i+1}^{n} \mathbf{K}_{i,j}\mathbf{e}_j\right) \Big/ \mathbf{K}_{i,i} \cap \mathbf{e}_i = \mathbf{e'}_i$$

Itérations par intervalle de Gauss-Seidel

$$(\mathbf{K} \setminus \mathbf{z}) \cap \mathbf{e} = \{\widetilde{e} \in \mathbf{e} \quad | \quad \exists \widetilde{K} \in \mathbf{K}, \exists \widetilde{z} \in \mathbf{z} : \quad \widetilde{K} * \widetilde{e} = \widetilde{z}\}$$

$$\widetilde{K}_{i,1}\widetilde{e}_1 + \ldots + \widetilde{K}_{i,i-1}\widetilde{e}_{i-1} + \widetilde{K}_{i,i}\widetilde{e}_i + \widetilde{K}_{i,i+1}\widetilde{e}_{i+1} + \ldots + \widetilde{K}_{i,n}\widetilde{e}_n = z_i$$

$$\widetilde{e}_i = \left(\widetilde{z}_i - \sum_{j=1}^{i-1} \widetilde{K}_{i,j}\widetilde{e}_j - \sum_{j=i+1}^{n} \widetilde{K}_{i,j}\widetilde{e}_j\right) \Big/ \widetilde{K}_{i,i}$$

$$\widetilde{e}_i \in \left(\mathbf{z}_i - \sum_{j=1}^{i-1} \mathbf{K}_{i,j}\mathbf{e}'_j - \sum_{j=i+1}^{n} \mathbf{K}_{i,j}\mathbf{e}_j\right) \Big/ \mathbf{K}_{i,i} \cap \mathbf{e}_i = \mathbf{e'}_i$$

Algorithme (rappel)

6. Calculer le résidu : $\mathbf{r} = [b - A * \widetilde{x}]$ avec une précision doublée
7. Préconditionner le résidu par R : $\mathbf{z} = R * \mathbf{r}$
8. Calculer une approximation initiale de l'erreur **e0**
9. while (not convergence)
   - Appliquer au maximum 10 itérations de Gauss-Seidel sur **K**, **z**, **e0**
   - Mettre à jour l'approximation flottante et la borne d'erreur :

$$\widetilde{x} = \widetilde{x} + \mathtt{mid}(\mathbf{e0}) \qquad \mathbf{e0} = \mathbf{e0} - \mathtt{mid}(\mathbf{e0})$$

   - Recalculer le résidu et **z**.

Plan de l'exposé

Test d'arrêt

Hypothèse (rappel) : le problème de départ est de calculer
$x|Ax = b \quad A \in \mathbb{F}^{n\times n}, b \in \mathbb{F}^n \leftarrow$ système sans coefficient intervalle.

1. Nombre de bits corrects demandés :

$$nb\_bits = -log2(max(diam(\mathbf{e}_j)/abs(\widetilde{x}_j))) \qquad (j = 1, \ldots, n)$$

2. $diam(\mathbf{e}^0), diam(\mathbf{e}^1), \ldots$ est une suite positive décroissante

$$diam(\mathbf{e}^i) - diam(\mathbf{e}^{i+1}) < \epsilon * diam(\mathbf{e}^i)$$

$\Rightarrow$ test d'arrêt : stop quand $nb\_bits \geq nb\_bits\_demandes$ ou $\Delta diam(e^i)$ assez petit

Plan de l'exposé

Premier test

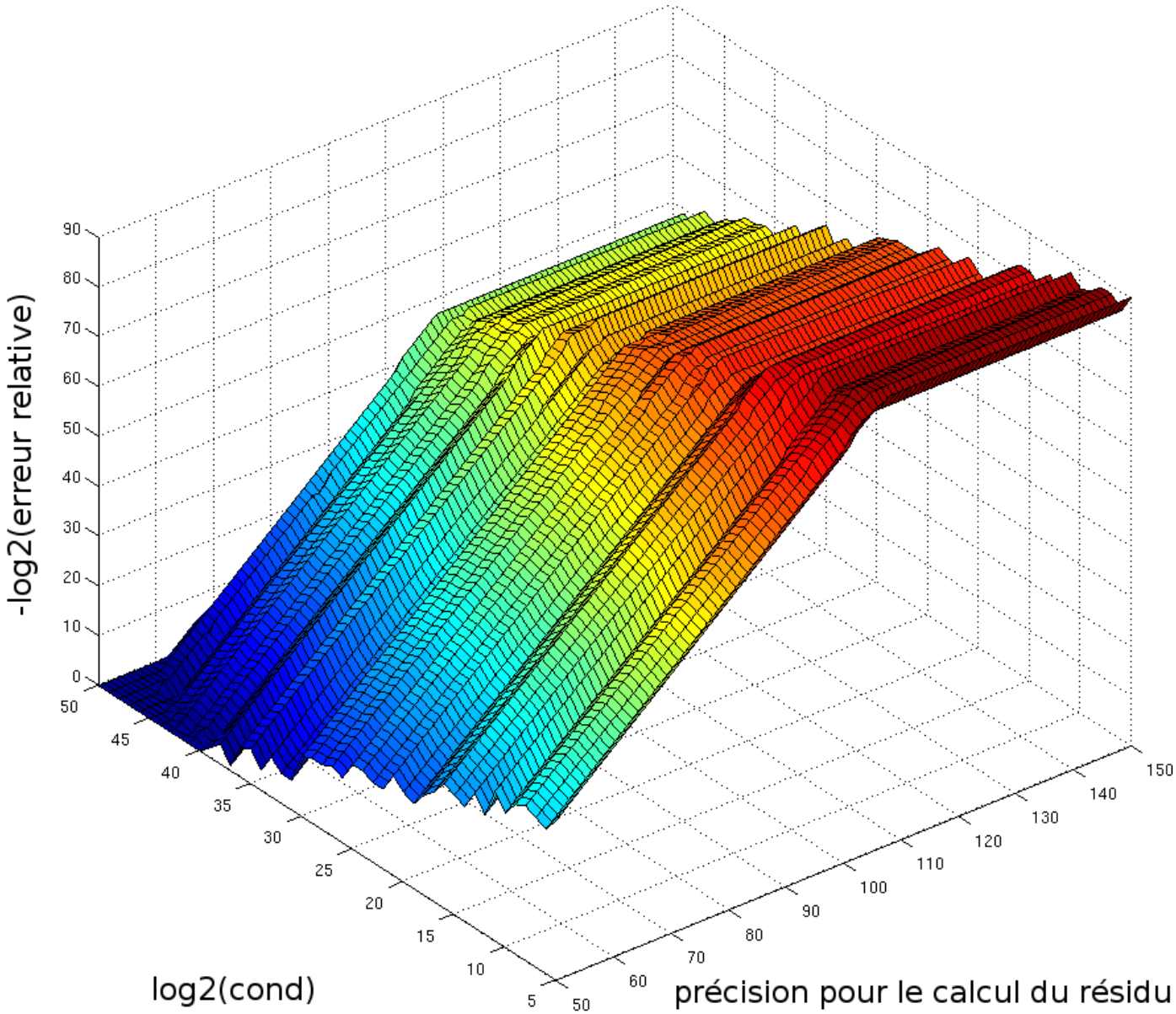
Précisions utilisées :

- Fixer la précision pour stocker toutes les variables à la précision courante.
- Fixer la précision de tous les calculs à la précision courante.
- Varier seulement la précision utilisée pour le calcul du résidu

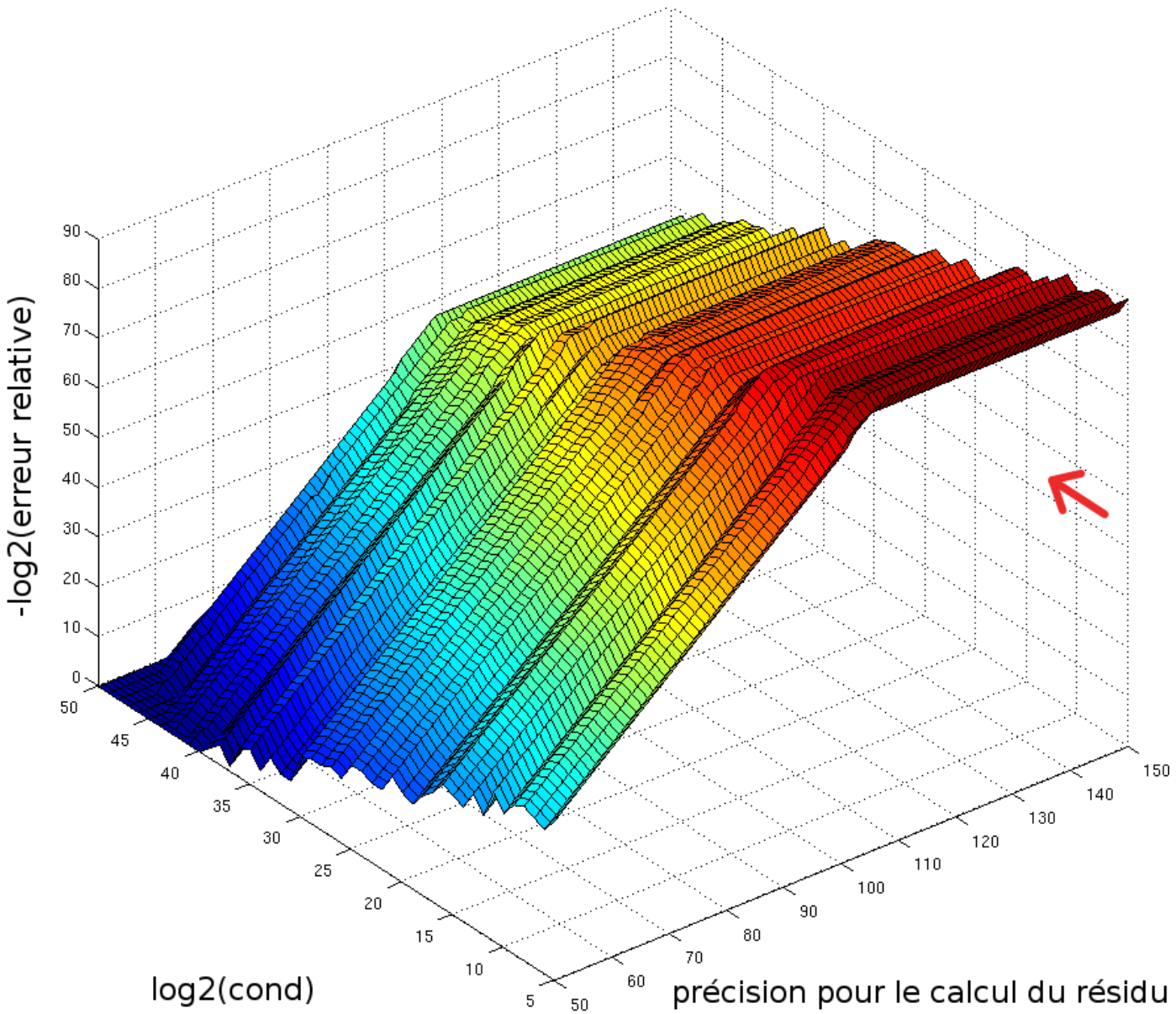
Données de test :

- Des matrices sont générées par la fonction `gallery('randsvd')` de Matlab.
- Le conditionnement varie entre 2^5 et 2^{50}.

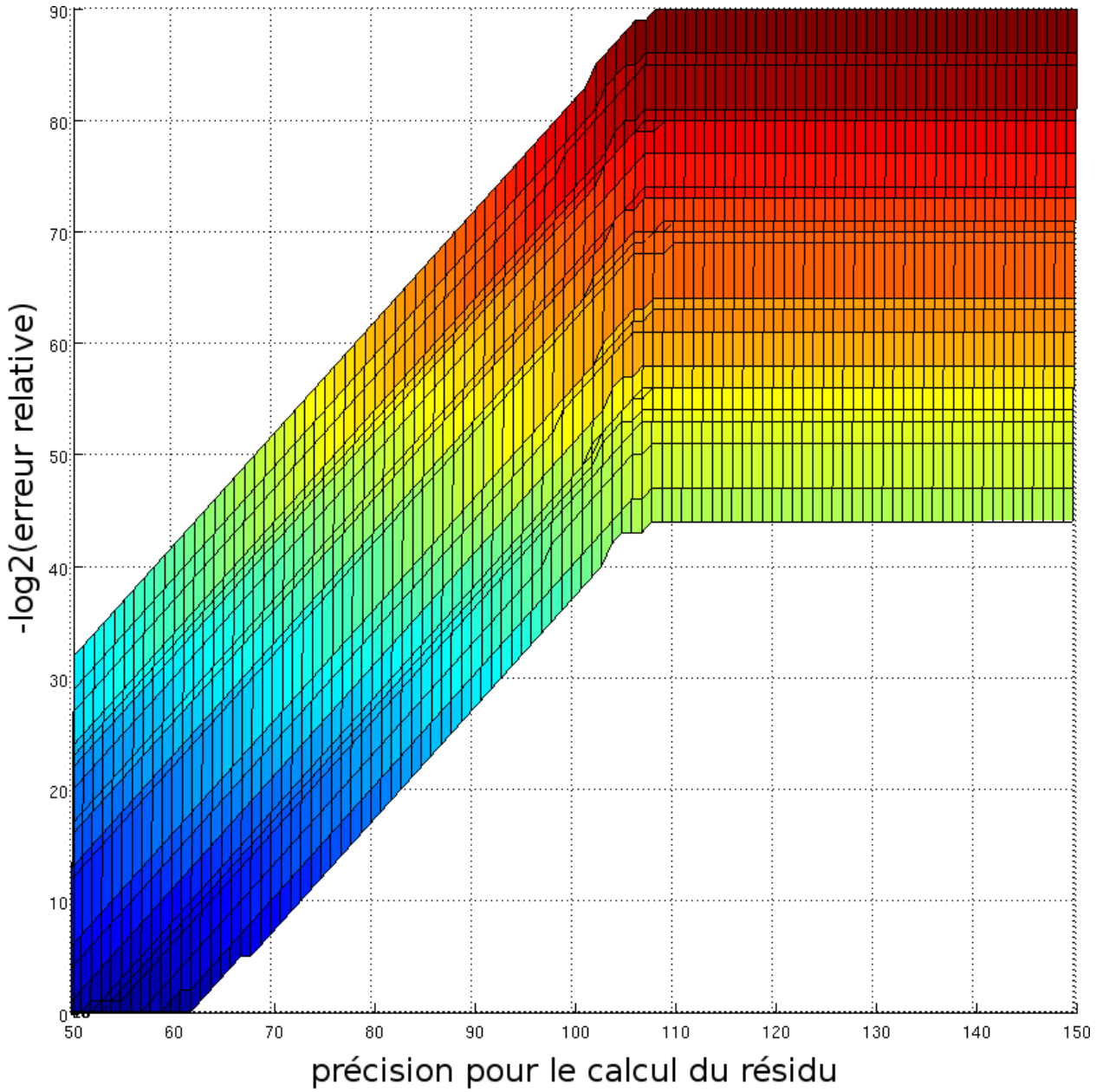
Premier test : Résultat



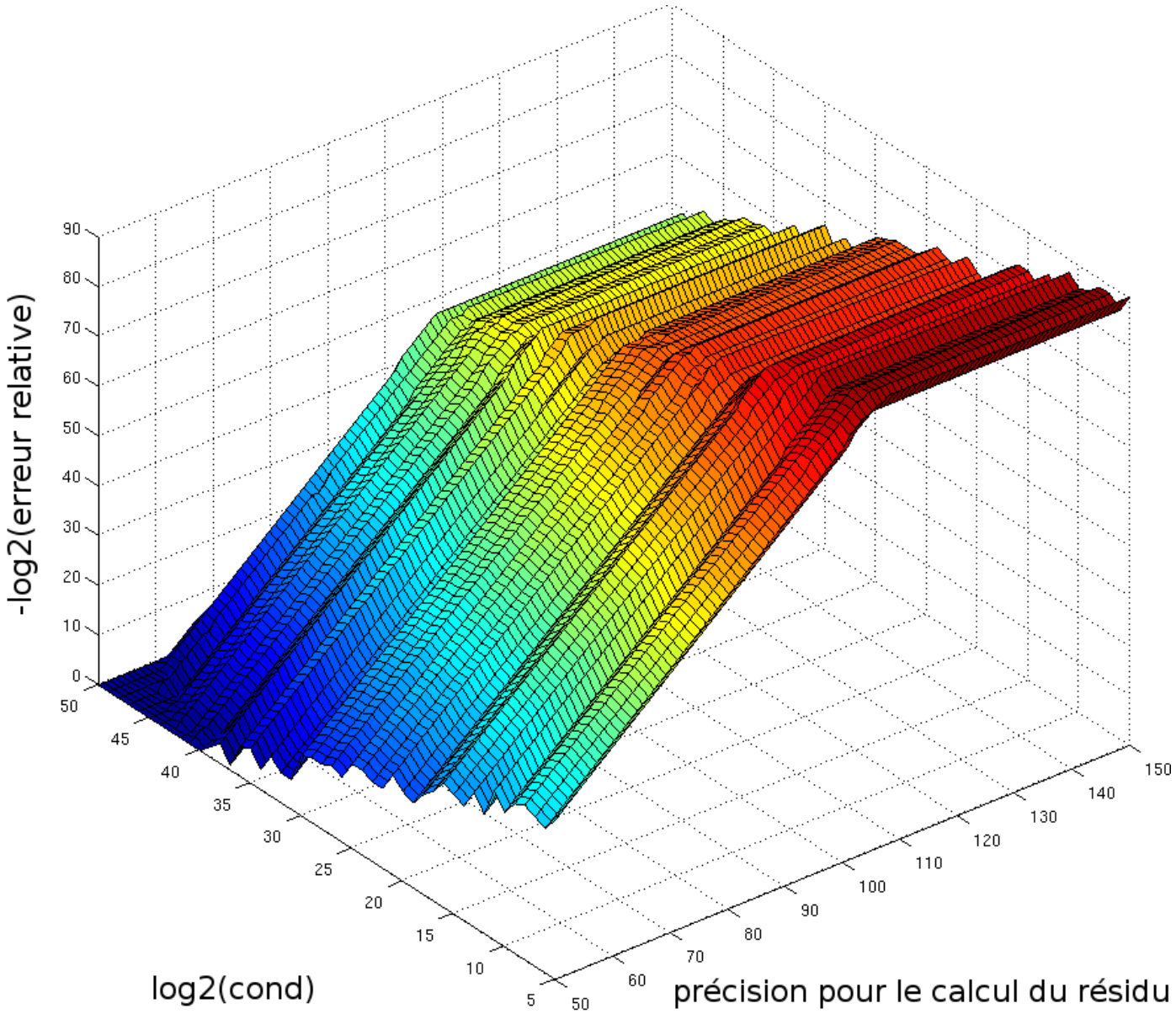
Premier test : Résultat



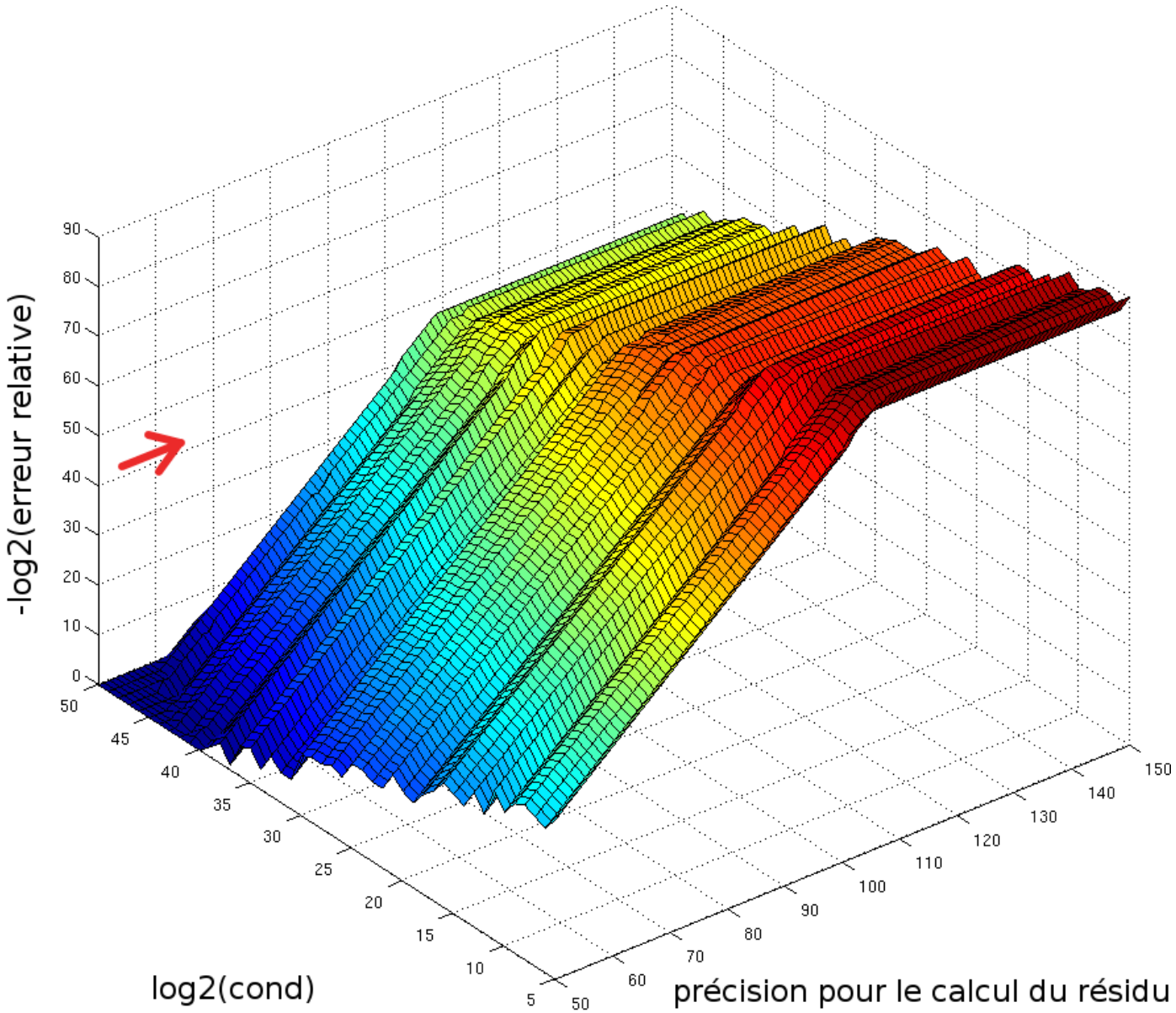
Premier test : Résultat



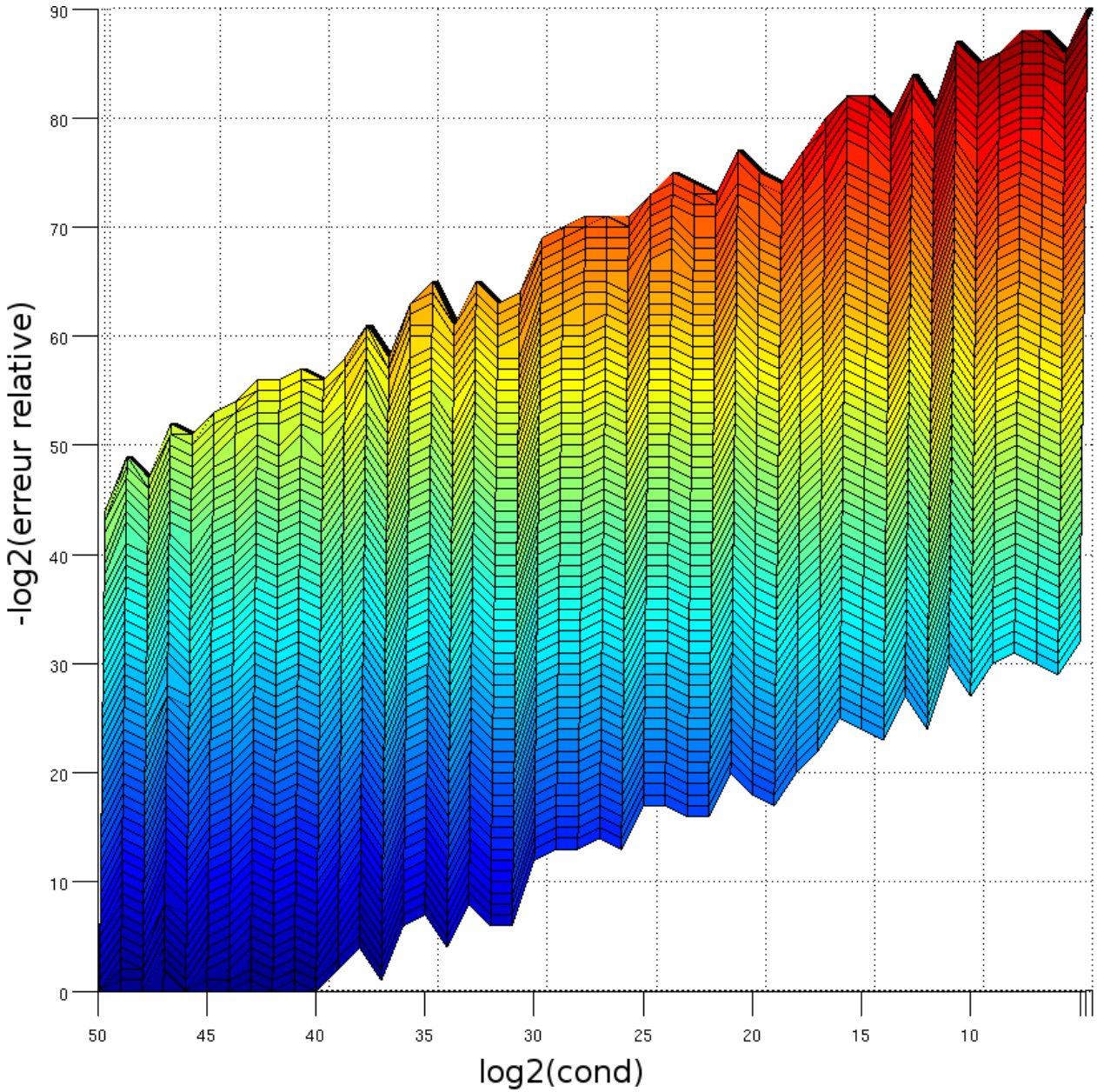
Premier test : Résultat



Premier test : Résultat



Premier test : Résultat



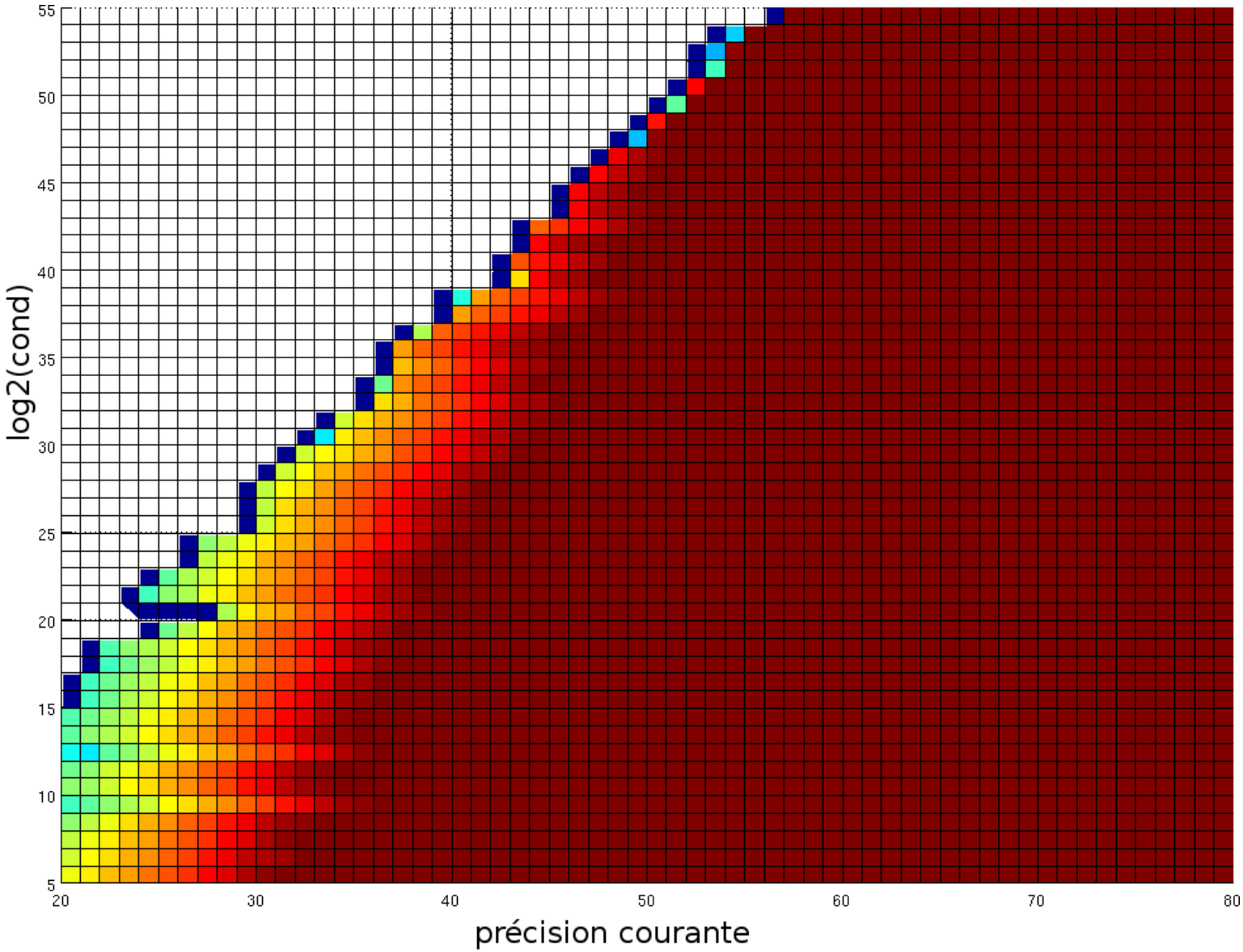
Deuxième test

Précisions utilisées :

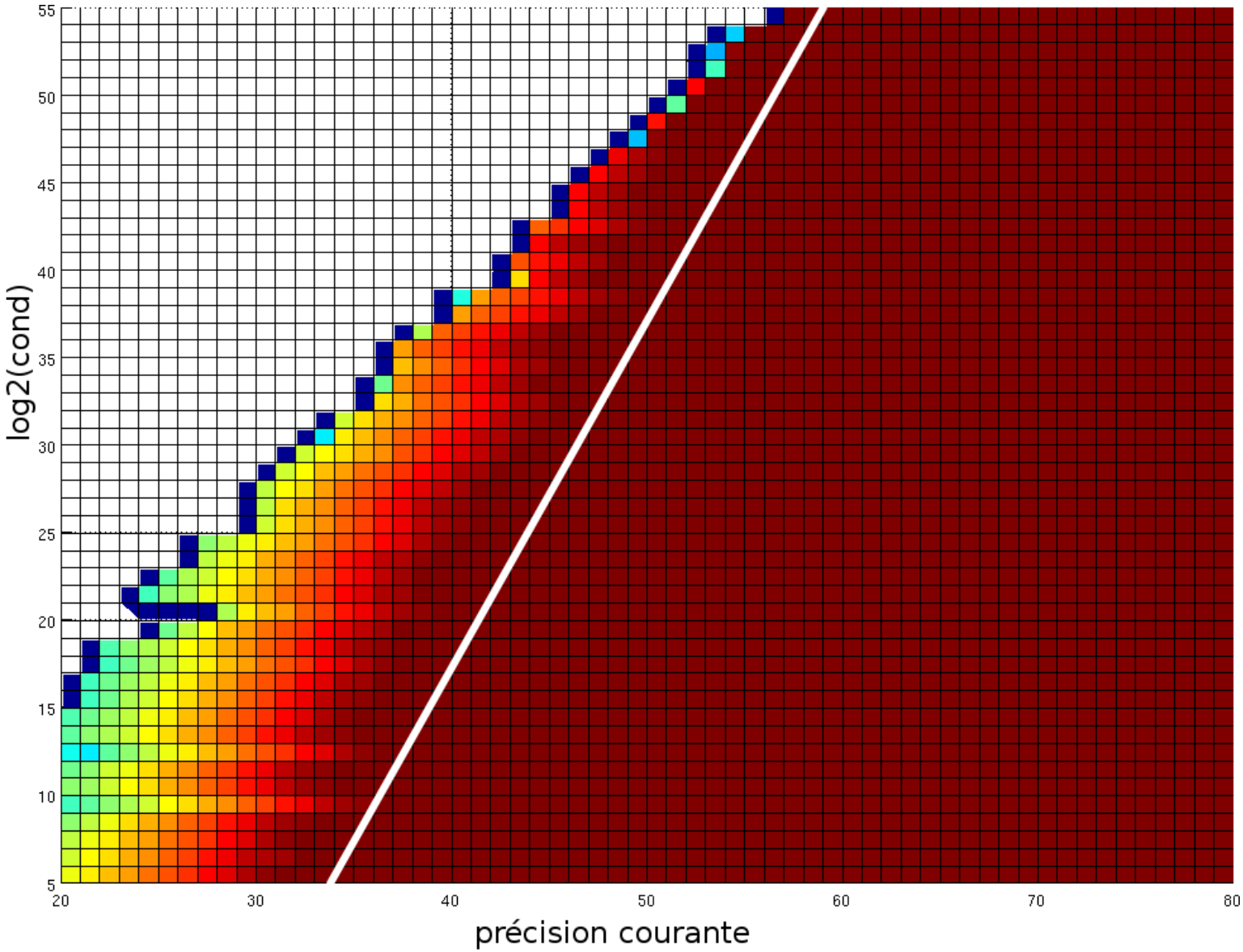
- Fixer la précision pour le calcul du résidu à deux fois la précision courante
- Varier la précision courante

$\Rightarrow$ effet de la précision courante sur la qualité du résultat

Deuxième test : Résultat



Deuxième test : Résultat



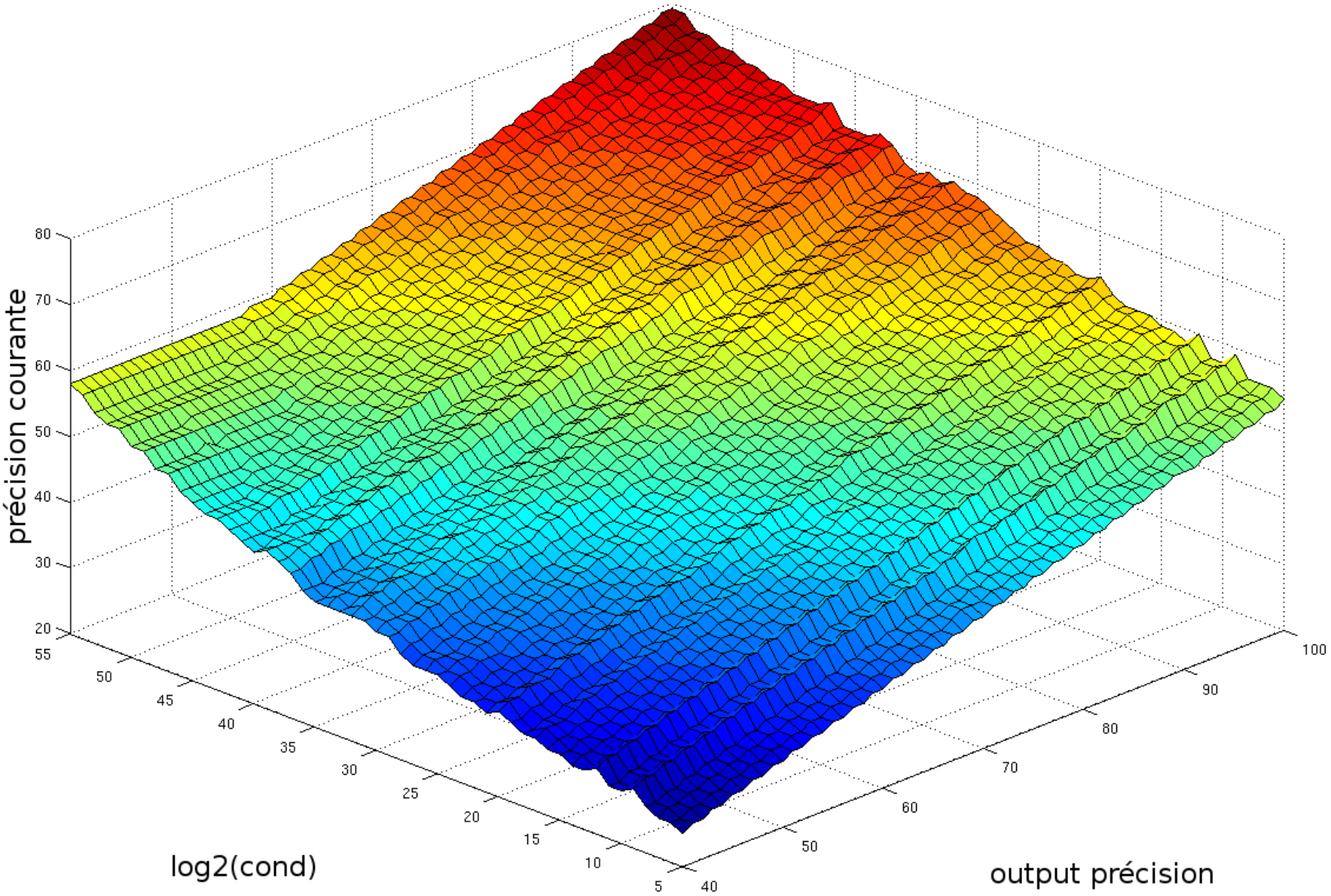
Troisième test

Précisions utilisées :

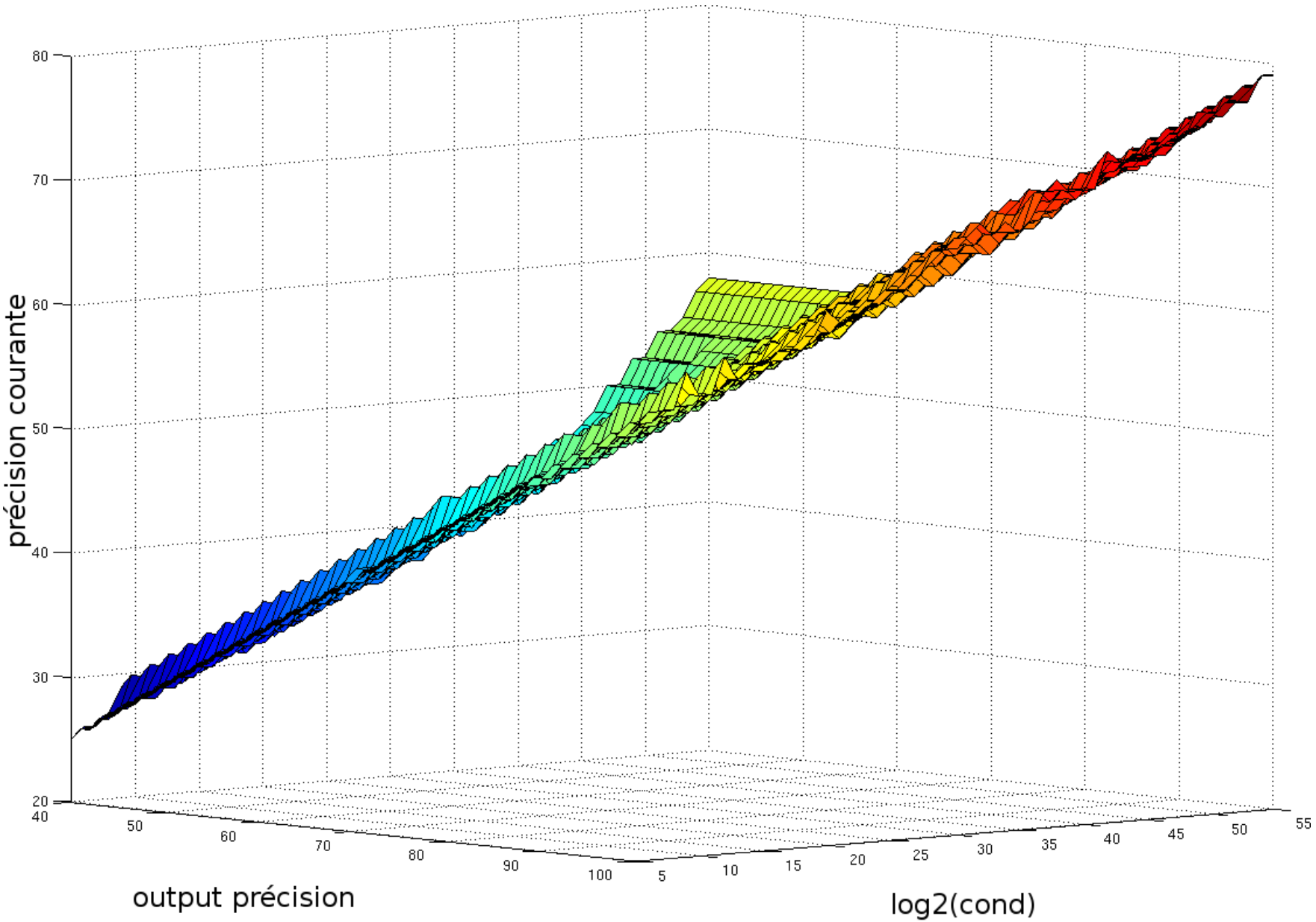
- Varier la précision courante la précision demandée pour le résultat
- trouver la précision minimale des calculs pour obtenir la précision demandée, en fixant la précision pour le calcul du résidu à deux fois la précision courante

$\Rightarrow$ relation entre la précision demandée et la précision minimale nécessaire pour les calculs

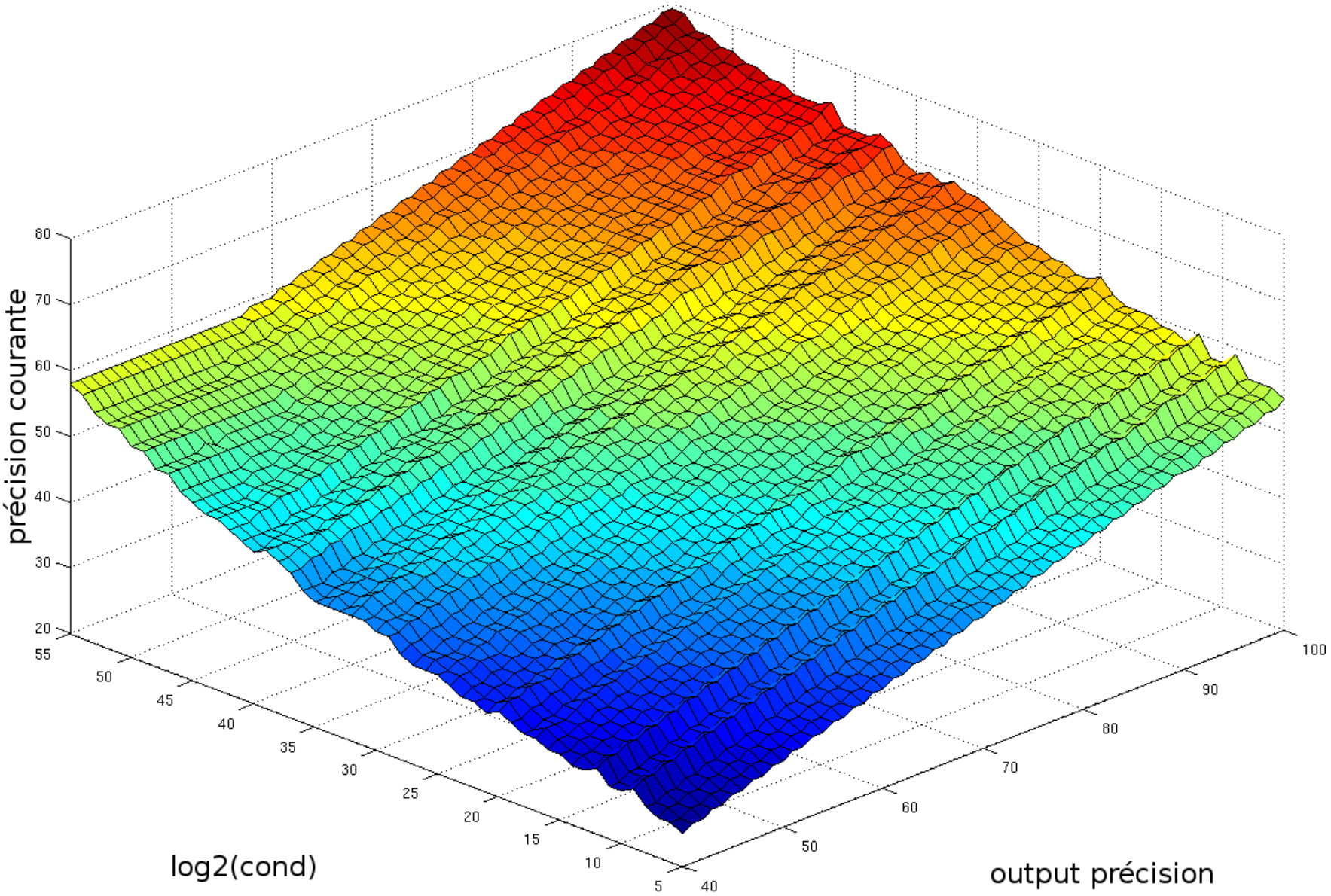
Troisième test : Résultat



Troisième test : Résultat



Troisième test : Résultat



Conclusions

- Notre méthode utilise deux niveaux de raffinement :
  1. Un raffinement flottant pour l'approximation flottante de la solution,
  2. Un raffinement par intervalle pour la borne d'erreur sur cette approximation.
- Notre méthode donne des résultats très fins quand le conditionnement de la matrice des coefficients est plus petit que 2^p (p est la précision courante).
- La dépendance entre la qualité du résultat avec la précision utilisée pour le calcul du résidu et avec le conditionnement du système sont quasiment linéaires.
- Quand la précision utilisée pour le calcul du résidu est plus grande que la précision doublée, il n'y a plus d'influence sur la qualité du résultat.

Perspectives

À faire : élargir à d'autres problèmes

- Systèmes linéaires d'inéquations
- Systèmes non linéaires
- Systèmes de contraintes numériques, en appliquant les techniques de propagation de contraintes

Références

[1] J.H. Wilkinson ; *Rounding Errors in Algebraic Processes*, Notes on Applied Science no 32, 1963.

[2] N.J. Higham ; *Accuracy and Stability of Numerical Algorithms*, 2nd edition, SIAM Press, 2002. Chapter 12 : Iterative Refinement.

[3] J. Demmel , Y. Hida, W. Kahan, X.S.Li, S. Mukherjee, E.J. Riedy ; *Error Bounds from Extra Precise Iterative Refinement*, Report No. UCB/CSD-04-1344, Computer Science Division (EECS), University of California, 2004.

[4] A. Neumaier ; *Interval methods for systems of equations*, Cambridge University Press, 1990. Chapter 4 : The Solution on Square Linear Systems of Equations.

[5] E. R. Hansen ; Bounding the Solution of Interval Linear Equations, SIAM Journal on Numerical Analysis, SIAM, 1992, pp 1493-1503.

[6] A. Neumaier ; *A simple derivation of the Hansen-Blek-Rohn-Ning-Kerfott enclosure for linear interval equations*, Reliable computing, 1999, pp 131–136.