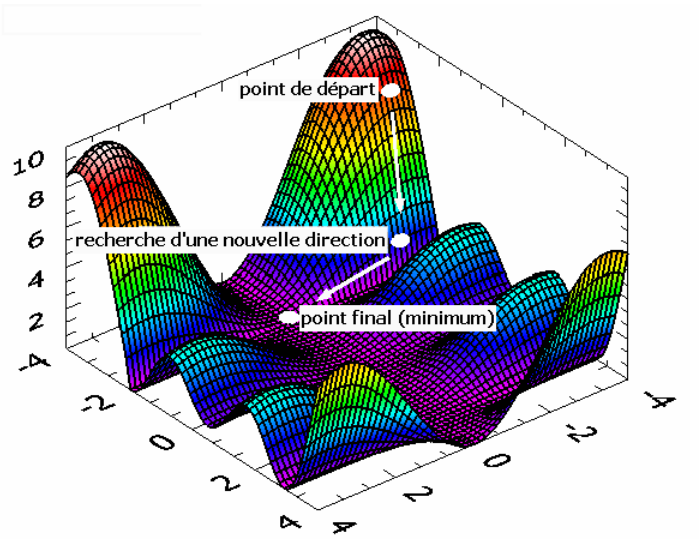
Institut national des sciences appliquées de Rouen

Projet MMSN GM3 - Vague 3 - Sujet 4

Etude des erreurs sur la méthode du Gradient Conjugué



Auteurs :
Thibaut André-Gallis
thibaut.andregallis@insa-rouen.fr
Kévin Gatel
kevin.gatel@insa- rouen.fr

Enseignants :
Bernard Gleyse
bernard.gleyse@insa-rouen.fr

6 Juin 2021

Table des matières

Introduction **2**

1 Présentation du problème **3**

2 Vecteur résidu r **4**
2.1 Etape 0 4
2.2 Etape 1 4
2.3 Etape 2 4
2.4 Etape 3 4
2.5 Etape 4 4

3 Vecteur solution x **5**
3.1 Etape 1 5
3.2 Etape 2 5
3.3 Etape 3 5
3.4 Etape 4 5

4 Analyse numérique du problème **6**

Conclusion **7**

Introduction

Pour commencer, nous avons longtemps pensé que les calculatrices et les ordinateurs étaient les références absolues en termes de calcul mathématique. Qu'ils ne se trompaient jamais pourvu qu'on leur donne le bon calcul. Cependant nous avons par la suite découvert que les réels n'existaient pas en machine et qu'on utilisait les flottants pour les représenter. Nous ne détaillerons pas la construction des flottants, ce n'est pas notre sujet mais il est important de connaître leur existence. De cela nous avons compris qu'il était impossible de représenter tous les réels par les flottants. Ce qui entrainera forcément des erreurs inévitables lors des calculs.

C'est là tout le sujet de notre projet, les erreurs. En effet nous allons nous intéresser aux erreurs de calculs survenus lors de la résolution d'un système linéaire sur machine. Plus particulièrement avec la méthode du gradient conjugué que nous avons déjà étudié auparavant lors d'un précédent projet. Nous allons donc à l'aide de la valeur binaire des nombres et de la connaissance de la construction des flottants pouvoir étudier et quantifier les erreurs survenues lors de la résolution du système linéaire.

Pour obtenir une étude plus large tous les calculs ont été réalisés sur deux machines différentes dont les caractéristiques ont été détaillés dans le fichier "README". Nous pourrons par conséquent les comparer pour remarquer des éventuelles différences d'une machine à l'autre.

1. Présentation du problème

L'objectif est donc d'étudier les erreurs que fait la machine en utilisant l'arithmétique flottante plutôt que l'ensemble théorique des réels.

Ces erreurs seront étudiées sur la solution du problème linéaire $Ax = b$ avec la méthode du gradient conjugué. En choisissant la matrice A de dimension 4 définie comme ci-dessous :

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix}$$

FIGURE 1.1 – Matrice de Hilbert de dimension 4

Le nombre d'étape pour trouver la solution sera en théorie inférieur ou égale à 4 (assuré par la méthode du gradient conjugué).

En notant $K_2(A)$ le conditionnement 2 de A tel que : [1]

$$K_2(a) = 1.5514 * 10^4$$

On a l'inégalité du conditionnement pour majorer l'erreur :

$$\frac{||\Delta x||_2}{||x||_2} \leq K_2(A) \frac{||\Delta b||_2}{||b||_2}$$

Le test d'arrêt est de la forme

$$tol^2 * (b, b) > (r, r)$$

avec $(\bullet, \bullet)$ le produit scalaire usuel et $tol = 10^{-10}$.

Enfin, le vecteur b est choisi comme ci-dessous :

$$b_i = \sum_{k=1}^{4} A_{ik}$$

de manière à avoir

$$x^T = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$$

1. conditionnement obtenu sur Matlab

2. Vecteur résidu r

2.1 Etape 0

2.2 Etape 1

2.3 Etape 2

2.4 Etape 3

2.5 Etape 4

3. Vecteur solution x

3.1 Etape 1

3.2 Etape 2

3.3 Etape 3

3.4 Etape 4

4. Analyse numérique du problème

Analysons maintenant le problème numériquement. On sait que la solution théorique est

$$x^T = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$$

Comparons maintenant ce résultat avec celui qu'on a obtenu numériquement au bout de la 4^{eme} étape :

$$\hat{x}^T = \begin{pmatrix} 0.9999999987685677105 & 0.9999999992953791939 & 0.9999999994982825546 & 0.99999999960549057491 \end{pmatrix}$$

On peut maintenant obtenir l'erreur absolue pour chaque composante afin d'obtenir le vecteur absolu : [1]

$$\varepsilon_{abs} = \begin{pmatrix} 1.2314322895 * 10^{-9} & 7.046208061 * 10^{-10} & 5.017174454 * 10^{-10} & 3.9450942509 * 10^{-10} \end{pmatrix}$$

On remarque que l'on obtient le même vecteur pour le vecteur erreur relative puisqu'on divise toutes les composantes par 1 :

$$\varepsilon_{rel} = \begin{pmatrix} 1.2314322895 * 10^{-9} & 7.046208061 * 10^{-10} & 5.017174454 * 10^{-10} & 3.9450942509 * 10^{-10} \end{pmatrix}$$

On observe des erreurs beaucoup plus élevées que celles obtenues localement. En effet pour une étude local on obtenait des erreurs d'ordre de grandeur d'environ 10^{-17} alors qu'ici il est d'environ 10^{-10}. Une différence de 10^7 qui n'est pas négligeable.

Cependant, on peut souligner l'efficacité de la méthode car en seulement 4 itérations l'erreur de la solution obtenue par rapport à celle théorique est de seulement 10^{-9}. Si l'on veut obtenir plus de précision il suffit de diminuer la tolérance et d'observer davantage d'étapes.

1. calcul effectué sur *wolframalpha.com*

Conclusion

Annexe