

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE  
ROUEN

INSA DE ROUEN

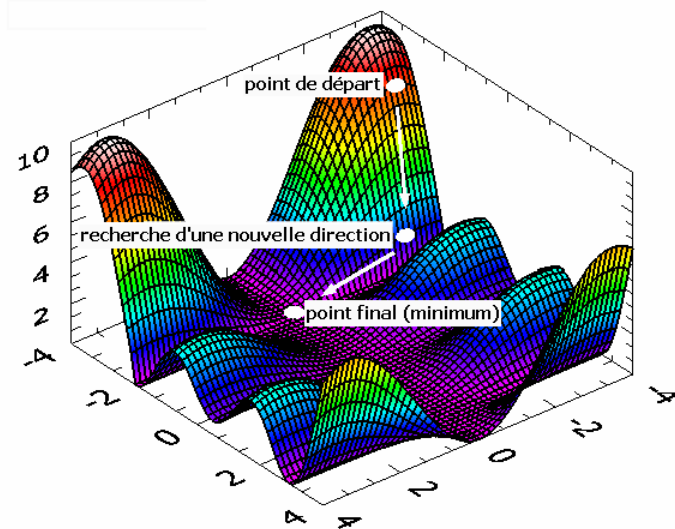


PROJET MMSN GM3 - VAGUE 3 - SUJET 4

---

# Résolution de système linéaire par la méthode du gradient conjugué

---



Auteurs :  
Thibaut ANDRÉ-GALLIS  
thibaut.andregallis@insa-rouen.fr  
Kévin GATEL  
kevin.gatel@insa-rouen.fr

Enseignant :  
Bernard GLEYSE  
bernard.gleyse@insa-rouen.fr

4 Janvier 2021

# Table des matières

|  |           |
|--|-----------|
| <b>Introduction</b>  | <b>3</b>  |
| <b>1 Présentation du problème</b>  | <b>4</b>  |
| 1.1 Principe . . . . .   | 4         |
| 1.2 Résolution mathématique . . . . .                                      | 4         |
| 1.2.1 Choix de la fonctionnelle à minimiser . . . . .                      | 4         |
| 1.2.2 Choix optimal de $\alpha_k$ dans une direction fixée $p_k$ . . . . . | 5         |
| 1.2.3 Méthode du gradient conjugué . . . . .                               | 5         |
| <b>2 Résolution numérique</b>  | <b>7</b>  |
| 2.1 Algorithmes et langage utilisés . . . . .                              | 7         |
| 2.2 Résultats obtenus . . . . .  | 7         |
| 2.2.1 Matrices sans perturbation . . . . .                                 | 7         |
| 2.2.2 Matrices avec perturbation . . . . .                                 | 13        |
| 2.2.3 Matrices dif . . . . .   | 13        |
| 2.2.4 Matrice elec . . . . .   | 14        |
| 2.2.5 Matrice elecmodif . . . . .  | 14        |
| 2.2.6 Matrices de Hilberts . . . . .                                       | 15        |
| 2.2.7 Matrice lap . . . . .  | 16        |
| 2.2.8 Matrices $tri_{alpha}$ . . . . .                                     | 16        |
| 2.2.9 Matrice de Wilson . . . . .  | 17        |
| <b>Conclusion</b>  | <b>18</b> |
| <b>Annexes</b>   | <b>19</b> |
| <b>Bibliographie</b>   | <b>20</b> |

# Table des figures

|  |           |
|--|-----------|
| <b>Résolution numérique</b>  | <b>7</b>  |
| 2.1 Algorithme du gradient conjugué muni du test d'arrêt <b>t1</b>   | 7         |
| 2.2 Algorithme du gradient conjugué muni du test d'arrêt <b>t2</b>   | 7         |
| 2.3 Exemple de fichier de résultat avec une matrice dif_4 sans perturbation                                      | 8         |
| 2.4 Vecteur $x_k$ à la 25 <sup>ème</sup> itération pour $dif_{50}$   | 8         |
| 2.5 Vecteur $r_k$ à la 26 <sup>ème</sup> itération pour $dif_{50}$   | 8         |
| 2.6 Vecteur $x_k$ à la 3 <sup>ème</sup> itération pour $elec$  | 9         |
| 2.7 Vecteurs $r_k$ jusqu'à la 4 <sup>ème</sup> itération pour $elec$   | 9         |
| 2.8 Produits scalaires entre les $r_k$ pour $elec$   | 9         |
| 2.9 Produits scalaires entre les $r_k$ pour $elecmodif$  | 10        |
| 2.10 Convergence des suites $x_k$ et $r_k$ pour $H_6$  | 10        |
| 2.11 Fichier résultat pour $lap_3$   | 11        |
| 2.12 Convergence parfaite de la suite $x_k$ à la 5 <sup>ème</sup> itération pour les matrices $tri_{alpha_{10}}$ | 11        |
| 2.13 Produits scalaires entre les résidus pour $W$   | 12        |
| 2.14 Suite des résidus $r_k$ pour $W$  | 12        |
| 2.15 Conditionnement de la matrice de Wilson   | 12        |
| 2.16 Vecteur solution $x_k$ à partir de la 4 <sup>ème</sup> itération pour $W$                                   | 12        |
| 2.17 Exemple de fichier de résultat avec une matrice dif_6 avec perturbation                                     | 13        |
| 2.18 Résultats obtenus pour la matrice dif en dimension 50   | 13        |
| 2.19 Résultats obtenus pour la matrice elec  | 14        |
| 2.20 Résultats obtenus pour la matrice elecmodif   | 15        |
| 2.21 Résultats obtenus pour la matrice de Hilbert en dimension 6   | 16        |
| 2.22 Résultats obtenus pour la matrice lap   | 16        |
| 2.23 Résultats obtenus pour la matrice $tri_5$   | 17        |
| 2.24 Résultats obtenus pour la matrice de Wilson   | 17        |
| <b>Annexes</b>   | <b>19</b> |
| Matrice elec   | 19        |
| Matrice elecmodif  | 19        |
| Matrice dif de dim 8   | 19        |
| Matrice de Hilbert de dim 5  | 19        |
| Matrice Laplacienne_3 (de dim 3 <sup>2</sup> )   | 19        |
| Matrice tri_α de dim 10 avec α = 5   | 19        |
| Matrice de Wilson  | 19        |

# Introduction

La méthode présentée dans ce rapport est celle du gradient conjugué. Il ne s'agit non seulement d'une des techniques les plus utiles pour résoudre des grands systèmes linéaires, mais elle peut même être adaptée de telle manière à ce qu'elle résout des problèmes d'optimisation non-linéaires. Ces deux variantes, reposant sur la même idée de base, sont respectivement appelées méthodes du gradient conjugué linéaire et non-linéaire. Dans la suite nous nous intéresserons uniquement à la méthode du gradient conjugué linéaire.

La méthode a été trouvée dans les années 50 par Magnus Hestenes et Eduard Stiefel, deux mathématiciens. Cette dernière se base sur la recherche de directions successives permettant d'atteindre la solution exacte d'un système linéaire de matrice symétrique et définie positive et représente une alternative à l'algorithme d'élimination de Gauss. Elle est même souvent préférée à cette dernière lorsque les systèmes d'équations sont de grandes tailles.

Les résultats obtenus ont été calculés avec deux machines différentes afin d'observer et de discuter des éventuelles différences. Les caractéristiques des deux ordinateurs sont indiquées dans le fichier "README".

# 1. Présentation du problème

## 1.1 Principe

La méthode du gradient conjugué linéaire est une méthode qui résout deux problèmes équivalents possédant la même solution unique. Ces problèmes sont le système d'équations linéaires

$$Ax = b$$

et le problème de minimisation suivant :

$$J(x) = (Ax, x) - 2(b, x)$$

où  $A$  est une matrice carrée symétrique définie positive de taille  $n$ ,  $x$  et  $b$  deux vecteurs de taille  $n$  et  $(.,.)$  représente le produit scalaire dans  $\mathbb{R}^n$ .

## 1.2 Résolution mathématique

### 1.2.1 Choix de la fonctionnelle à minimiser

La solution  $\bar{x}$  du problème  $Ax = b$  est le vecteur pour lequel  $J(x)$  atteint son minimum. On a l'expression :

$$J(\bar{x}) = -(b, A^{-1}b).$$

Posons

$$g(x) = 2(Ax - b) = -2r(x)$$

où  $r(x) = b - Ax = A\bar{x} - Ax$  est le vecteur résidu du système  $Ax = b$ .

Si on pose  $\bar{x} - x = e(x)$ , on a :

$$E(x) = (Ae(x), e(x))$$

Il est équivalent de minimiser  $J$  ou  $E$  comme définies ci-dessus.

Puisque  $A$  est symétrique et définie positive, alors  $(Ax, y)$  est un produit scalaire et  $E(x) = \|e(x)\|_A^2$ , avec  $\|e\|_A^2 = (Ae, e)^{\frac{1}{2}}$  norme associée à ce produit scalaire. Le minimum de  $E$  est nul et est atteint en  $\bar{x}$ .

$E(x)$  peut aussi s'exprimer en fonction du résidu  $r(x) = A\bar{x} - Ax$  :

$$E(x) = (r(x), A^{-1}r(x)).$$

Pour minimiser la fonctionnelle  $E$ , les méthodes de descente comme celle du gradient conjugué donnent  $x_{k+1}$  à partir de  $x_k$  en choisissant à la  $(k+1)^{me}$  itération une direction de descente  $p_k \neq 0$  (un vecteur de  $\mathbb{R}^n$ ) et un scalaire  $\alpha_k$  avec

$$x_{k+1} = x_k + \alpha_k p_k$$

de manière à ce que  $E(x_{k+1}) < E(x_k)$ .

### 1.2.2 Choix optimal de $\alpha_k$ dans une direction fixée $p_k$

On suppose la direction  $p_k$  fixée.

Le choix local optimal de  $\alpha_k$  est obtenu lorsqu'à chaque itération, on minimise  $E(x_{k+1})$ . dans la direction  $p_k$  :

$$E(x_k + \alpha_k p_k) = \min_{\alpha \in \mathbb{R}} E(x_k + \alpha p_k)$$

Son minimum est atteint pour

$$\alpha_k = \frac{(r_k, p_k)}{(Ap_k, p_k)}.$$

**Lemme 1.1.**  $\forall p_k \neq 0$ , pour  $\alpha_k$  optimal local, on a la relation suivante valable pour  $k \geq 0$  :

$$\frac{(r_k, p_k)^2}{(Ap_k, p_k)(A^{-1}r_k, r_k)} \geq \frac{1}{\text{cond}(A)} \left( \frac{r_k}{\|r_k\|_2}, \frac{p_k}{\|p_k\|_2} \right)^2$$

Ce lemme permet notamment le choix des directions de descente.

**Théorème 1.2.** Pour  $\alpha_k$  optimal local, toute direction  $p_k$  qui vérifie  $\forall k \geq 0$  :

$$\left( \frac{r_k}{\|r_k\|_2}, \frac{p_k}{\|p_k\|_2} \right)^2 > 0$$

Ce théorème implique que la suite  $(x_k)_{k \geq 0}$  converge vers la solution  $\bar{x}$  qui minimise  $E(x)$ .

### 1.2.3 Méthode du gradient conjugué

#### Introduction

Recopier en gros le cours d'andré draux à partir de son introduction 2.3.1 page 44, toutes les propriétés sans démonstrations ni définitions + dire la définition 2.3.5 + rappeler l'inégalité du conditionnement + rappeler la formule de l'erreur relative + rappeler la complexité de l'algorithme

On prend les  $\alpha_k$  = minimum local, alors  $(p_{k-1}, r_k) = 0$ . On cherche les  $p_k$  dans le plan  $(r_k, p_{k-1})$ .

on pose  $p_k = r_k + \beta_k p_{k-1}$

$\beta_k$  sera déterminé de telle façon que le facteur de réduction de l'erreur soit maximal. On a  $E(x_{k+1}) = E(x_k) * (1 - \gamma_k)$ . On choisit  $\beta_k$  pour que  $\gamma_k$  soit maximum.

comme  $(r_k, p_k) = (r_k, r_k) + \beta_k (r_k, p_{k-1}) = \|r_k\|_2^2$  (on prend  $p_0 = r_0$  ( $\beta_0 = 0$ )) pour que la relation soit vraie  $\forall k \geq 0$ ,  $\gamma_k$  sera maximum, si  $(Ap_k, p_k)$  est minimum.

$$\begin{aligned} (Ap_k, p_k) &= (A(r_k + \beta_k p_{k-1}), r_k + \beta_k p_{k-1}) \\ &= \beta_k^2 (Ap_{k-1}, p_{k-1}) + 2\beta_k (Ap_{k-1}, r_k) + (Ar_k, r_k). \end{aligned}$$

le trinôme est donc minimal pour  $\beta_k = -\frac{(Ap_{k-1}, r_k)}{(Ap_{k-1}, p_{k-1})}$ .

Cette valeur de  $\beta_k$  correspond aussi au point d'annulation de la dérivée. On obtient donc :

$$\beta_k (Ap_{k-1}, p_{k-1}) + (Ap_{k-1}, r_k) = (Ap_{k-1}, r_k + \beta_k p_{k-1}) = (Ap_{k-1}, p_k) = 0$$

Comme A est symétrique définie positive,  $(u, v)$  est un produit scalaire  $(u, v)_A$ . Par conséquent deux vecteurs A-conjugués sont orthogonaux pour ce produit scalaire.

**Propriété 1.2.1.** Si  $r_i \neq 0$  pour  $i=0, \dots, k$  alors :

i)  $(r_{k+1}, r_k) = 0$  pour  $k \geq 0$

ii)  $\beta_0 = 0, \beta_k = \frac{\|r_k\|^2}{\|r_{k-1}\|^2}$  pour  $k \geq 1$

**Remarque 1.2.1.** comme  $2r_k = -g_k$ , où  $-g_k$  est le gradient de la fonctionnelle, alors :

$$\beta_k = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$$

### L'algorithme

$$\boxed{\begin{array}{l} \text{On initialise : } \left\{ \begin{array}{l} x_0 \\ p_0 = r_0 = b - Ax_0 \end{array} \right. \quad \text{P} \\ \text{pour } k=0,1,\dots \left\{ \begin{array}{l} \alpha_k = \frac{\|r_k\|^2}{(Ap_k, p_k)} \\ x_{k+1} = x_k + \alpha_k p_k \\ r_{k+1} = r_k - \alpha_k Ap_k \\ \beta_{k+1} = \frac{\|r_{k+1}\|^2}{\|r_k\|^2} \\ p_{k+1} = r_{k+1} + \beta_{k+1} p_k \end{array} \right. \end{array}}$$

Complexité : Si on prend N le nombre d'itérations on évaluera la complexité de cet algorithme à environ  $2N^3$ . Ce qui reste important en comparaison l'algorithme de la méthode de Choleski possède une complexité de  $\frac{N^3}{3}$  ce qui est nettement inférieur pour le même type de matrice.

Une solution à cela serait de préconditionner la matrice afin de réduire considérablement le nombres d'itérations et de cette façon on aurait alors l'une des méthodes les mieux adapté à la résolution de systèmes linéaires dont la matrice est symétrique définie positive.

**Définition 1.2.1.** L'espace vectoriel  $k_k = V(r_0, \dots, A^{k-1}r_0)$  est appelé espace de Krylov

Les  $r_i = 0, \dots, k-1$  forment une base orthogonale de cet espace.

**Théorème 1.3.** L'inégalité du conditionnement :

$$\frac{\|\delta X\|_q}{\|X\|_q} \leq \text{cond}_q(A) * \frac{\|\delta A\|_q}{\|A\|_q}$$

Cette inégalité nous dit que l'erreur relative sur la solution X doit être inférieure au conditionnement de la matrice A multiplié par la norme de la perturbation  $\delta A$  sur la norme de A. Dans notre cas on prendra la norme infini pour q. Cette inégalité n'est présente et analysée uniquement dans le cas de résolution de systèmes avec perturbations

## 2. Résolution numérique

### 2.1 Algorithmes et langage utilisés

L'algorithme du gradient conjugué a été implémenté en Fortran. Le choix du langage a été influencé par la facilité d'écrire certaines opérations vectorielles mathématiques en Fortran. L'ensemble des variables sont déclarées en double précision afin d'avoir le plus de chiffres significatifs (précision autour de  $10^{-16}$ ). Un algorithme avec et sans perturbation de la matrice  $A$  a été implémenté avec un test d'arrêt différent chacun.

```

 $\mathbf{x}$  =  $\mathbf{x}^0$ 
 $\mathbf{r}$  =  $A\mathbf{x} - \mathbf{b}$ 
 $\mathbf{p}$  =  $\mathbf{r}$ 
tant que  $\|\mathbf{r}\| \geq \varepsilon \|\mathbf{r}^0\|$  faire
     $\alpha = \frac{(\mathbf{r}, \mathbf{r})}{(A\mathbf{p}, \mathbf{p})}$ 
     $\mathbf{x} = \mathbf{x} - \alpha \mathbf{p}$ 
     $\mathbf{r} = \mathbf{r} - \alpha A\mathbf{p}$ 
     $\beta = \frac{(\mathbf{r}, \mathbf{r})}{\alpha(A\mathbf{p}, \mathbf{p})}$ 
     $\mathbf{p} = \mathbf{r} + \beta \mathbf{p}$ 
fin

```

FIGURE 2.1 – Algorithme du gradient conjugué muni du test d'arrêt **t1**

```

 $\mathbf{x}$  =  $\mathbf{x}^0$ 
 $\mathbf{r}$  =  $A\mathbf{x} - \mathbf{b}$ 
 $\mathbf{p}$  =  $\mathbf{r}$ 
tant que  $\|\mathbf{r}\| \geq \varepsilon$  faire
     $\alpha = \frac{(\mathbf{r}, \mathbf{r})}{(A\mathbf{p}, \mathbf{p})}$ 
     $\mathbf{x} = \mathbf{x} - \alpha \mathbf{p}$ 
     $\mathbf{r} = \mathbf{r} - \alpha A\mathbf{p}$ 
     $\beta = \frac{(\mathbf{r}, \mathbf{r})}{\alpha(A\mathbf{p}, \mathbf{p})}$ 
     $\mathbf{p} = \mathbf{r} + \beta \mathbf{p}$ 
fin

```

FIGURE 2.2 – Algorithme du gradient conjugué muni du test d'arrêt **t2**

Le vecteur initial  $x_0$  a été initialisé au vecteur nul, la tolérance  $\varepsilon$  a été fixée à  $10^{-10}$  et le vecteur  $\mathbf{b}$  est choisi tel que la solution du problème  $Ax = b$  soit un vecteur contenant uniquement des 1. En d'autres termes,  $b_i$  est la résultante des colonnes de la ligne  $i$  de la matrice  $A$ .

L'ensemble des matrices testées (à une dimension près) se trouvent en Annexes.

### 2.2 Résultats obtenus

Convergence des  $x_n$ , convergence des résidus, p.s. des résidus qui forment bien une base, inégalité du conditionnement...

#### 2.2.1 Matrices sans perturbation

Les fichiers de résultats sont obtenus en entrant les données dans un fichier de données puis en donnant en entrée de l'exécutable ce fichier de donnée comme ci-dessous :



FIGURE 2.3 – Exemple de fichier de résultat avec une matrice dif 4 sans perturbation

## Matrices dif

En grandes dimensions, nous observons davantage de résultats. En dimension 50, on observe une convergence de la suite  $x_k$  plus lente vers la solution.

FIGURE 2.4 – Vecteur  $x_k$  à la 25<sup>eme</sup> itération pour  $dif_{50}$

FIGURE 2.5 – Vecteur  $r_k$  à la 26<sup>eme</sup> itération pour  $dif_{50}$ 

L'erreur relative est de l'ordre de  $10^{-15}$  ce qui est très faible, et donc positif pour notre cas.

Même en grande dimension, pour cette matrice les convergences sont bien respectées malgré la précision machine.

### Matrice elec

La matrice étant de petite taille les  $x_k$  convergent très rapidement vers la solution.

```
3  0.99999999999999978      0.99999999999999978      0.99999999999999978
```

FIGURE 2.6 – Vecteur  $x_k$  à la 3<sup>eme</sup> itération pour *elec*

De même pour la suite  $r_k$  qui convergent vers le vecteur nul :

```
0  4.0019317313140970E-322  1.3235777704702403  -1.3235777704702398
1 -2.0000000000000000      2.2204460492503131E-016  0.0000000000000000
2 -2.3439490445859872     -2.4904458598726116  1.0254777070063703
3  1.3235777704702403     -1.3235777704702398  -0.18908253863860636
4  2.2204460492503131E-016  0.0000000000000000  -3.3306690738754696E-016
```

FIGURE 2.7 – Vecteurs  $r_k$  jusqu'à la 4<sup>eme</sup> itération pour *elec*

En revanche certains produits scalaires sont éloignés de zéro ne formant pas une base orthogonale.

```
produit scalaire entre les rn :
(r      0 .r      1 ) =  2.9389330313161827E-016
(r      0 .r      2 ) = -4.6535982757934571
(r      0 .r      3 ) = -1.5015926695568318
(r      0 .r      4 ) =  4.4083995469742726E-016
(r      1 .r      2 ) =  4.6878980891719735
(r      1 .r      3 ) = -2.6471555409404810
(r      1 .r      4 ) = -4.4408920985006262E-016
(r      2 .r      3 ) = -1.6653345369377348E-015
(r      2 .r      4 ) = -8.6201392803698488E-016
(r      3 .r      4 ) =  3.5687043951696522E-016
estimation de l'erreur relative :
2.2204460492503136E-016
```

FIGURE 2.8 – Produits scalaires entre les  $r_k$  pour *elec*

Cette observation est due aux vecteurs résidus  $r_2$  et  $r_3$  qui posent des problèmes d'orthogonalité.

On remarque que l'erreur relative est très basse malgré tout.

### Matrice elecmodif

Cette matrice n'étant pas définie positive il est inutile de l'étudier en profondeur afin de montrer la cohérence avec la partie mathématique attendue numériquement.

Cependant ses propriétés sont étonnantes malgré sa non définie-positivité. En effet ses  $x_k$  convergent vers la solution, ses vecteurs résidus  $r_k$  convergent vers le vecteur nul.

On retrouve les mêmes produits scalaires anormaux que la matrice *elec* causés par les vecteurs  $r_2$  et  $r_3$  :

```

produit scalaire entre les rn :
(r      0 .r      1 ) = 1.0603405902531558E-015
(r      0 .r      2 ) = -14.144747700160016
(r      0 .r      3 ) = -0.64509701535473418
(r      0 .r      4 ) = 1.1712940538058922E-015
(r      1 .r      2 ) = -188.57327818737690
(r      1 .r      3 ) = -4.0118045221231409
(r      1 .r      4 ) = 4.0500935938325721E-013
(r      2 .r      3 ) = -4.1211478674085811E-013
(r      2 .r      4 ) = -3.1284918910098789E-013
(r      3 .r      4 ) = -7.5602677515395840E-016
estimation de l'erreur relative :
1.1102230246251569E-015

```

FIGURE 2.9 – Produits scalaires entre les  $r_k$  pour *elecmodif*

On remarque que l'erreur relative est également très basse.

Même si la matrice n'est pas définie positive, on observe les mêmes propriétés que sa matrice d'origine elec numériquement.

## Matrices de Hilbert

En dimension 6 la suite des  $x_k$  et la suite des résidus convergent beaucoup moins rapidement que les matrices précédentes.

```

suite de vecteurs x et le n'd'itération :
0 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000
1 1.5210871468871527 0.98892837829689528 0.75610891703865757 0.61814182888562785 0.52501404438233279 0.45728474656175461
2 0.93489743892528543 1.1699804214389617 1.0947766113800910 0.99240407447023804 0.89838393852941589 0.81736193343418462
3 1.0037085607344767 0.97162950690639938 1.0271946795309368 1.0292165772692474 1.0020215865016213 0.96223451189827158
4 0.999896121045222396 1.0016605060834245 0.99453198148151856 1.0035929039545122 1.0048074940229736 0.99541943762847696
5 0.99992551625939163 1.0010278853413876 0.99663521332752059 1.0020441617657465 1.0032623647250449 0.99701952514430747
6 1.0000011581990764 0.99996689396291449 1.0002237793324107 0.99941847047305743 1.0006413536266310 0.99974747670367237
7 1.0000011606244197 0.99996689313915810 1.0002237778062124 0.99941846916395971 1.0006413517907162 0.99974747536908670

suite de vecteurs r :
r 0 1.5069002198158020E-321 -1.5369050243800593E-005 -9.0324062131132285E-006 -6.6275384443997366E-006 -5.1868184528469504E-006 -4.3086303128609755E-006
r 1 -2.4499999999999997 -5.8112640307500518E-010 1.9604419696232951E-010 3.7036643133914074E-010 3.1034021953955460E-010 4.3659581235701235E-010
r 2 0.15334003223998582 6.7177246484060040E-012 -5.8795426730387274E-012 2.0355805934944045E-011 -6.7232164579370554E-011 6.5257147561399808E-011
r 3 -1.1820179993962676E-003 3.2566650697245861E-003 1.1372478250830653E-003 -5.6371249307769662E-004 -1.6548969082078452E-003 -2.3294915443311109E-003
r 4 2.4209653369403148E-006 -1.8703972592999749E-005 2.0041148098081480E-005 1.5817874204312240E-005 -1.6050791141813611E-006 -2.0280365341271249E-005
r 5 -1.0212550176866748E-009 2.1662456459224324E-008 -7.1611362412287154E-008 5.2430003776128889E-008 5.2480022909865512E-008 -5.4127413426814246E-008
r 6 -1.5369050243800593E-005 -9.0324062131132285E-006 -6.6275384443997366E-006 -5.1868184528469504E-006 -4.3086303128609755E-006 -3.7582297935742183E-006
r 7 -5.8112640307500518E-010 1.9604419696232951E-010 3.7036643133914074E-010 3.1034021953955460E-010 4.3659581235701235E-010 3.2333947615648212E-010
r 8 6.7177246484060040E-012 -5.8795426730387274E-012 2.0355805934944045E-011 -6.7232164579370554E-011 6.5257147561399808E-011 -3.7582297935742183E-006

```

FIGURE 2.10 – Convergence des suites  $x_k$  et  $r_k$  pour  $H_6$

En effet, pour les résidus nous sommes aux alentours de  $10^{-12}$  au lieu de  $10^{-16}$  dans les exemples précédents. La précision tourne autour de  $10^{-5}$  pour le vecteur  $x_k$  à partir de la 7<sup>me</sup> itération.

Cependant les produits scalaires satisfont bien l'orthogonalité et l'erreur relative est environ  $10^{-5}$  ce qui est plutôt élevé par rapport aux exemples précédents.

En dimension plus élevée, nous observons les mêmes propriétés. Une convergence des  $x_k$  et des  $r_k$  assez lente, des produits scalaires au voisinage de zéro et une erreur relative du même ordre.

## Matrice lap

```

suite de vecteurs x et le n°d'itération :
0 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000
0.0000000000000000 0.0000000000000000
1 1.0714285714285714 0.5357142857142857 0.5357142857142857 0.5357142857142857 0.5357142857142857 0.5357142857142857
0.5357142857142857 1.0714285714285714
2 1.0249999999999999 1.0000000000000000 1.0000000000000000 1.0000000000000000 1.0000000000000000 1.0000000000000000
1.0000000000000000 1.0249999999999999
3 0.9999999999999999 1.0000000000000000 0.7500000000000000 1.0000000000000000 1.2500000000000000 1.0000000000000000
1.0000000000000000 0.9999999999999999
suite de vecteurs r :
0 3.2460112931769898E-321 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000
0.0000000000000000 0.0000000000000000
r 1 -2.0000000000000000 -1.0000000000000000 -1.0000000000000000 -1.0000000000000000 -1.0000000000000000 -1.0000000000000000
-1.0000000000000000 -2.0000000000000000
r 2 1.2142857142857144 -1.0000000000000000 7.1428571428571397E-002 -1.0000000000000000 -1.0000000000000000 -1.0000000000000000
-1.0000000000000000 1.2142857142857144
r 3 9.999999999999999E-002 0.3000000000000000 0.3000000000000000 -1.0000000000000000 0.3000000000000000 -0.3000000000000000
0.3000000000000000 9.999999999999999E-002
r 4 1.3877787807814457E-017 5.5511151231257827E-017 5.5511151231257827E-017 0.0000000000000000 0.0000000000000000 5.5511151231257827E-017
0.0000000000000000 -8.326672684686741E-017
produit scalaire entre les rni :
(r 0 .r 1 ) = -6.4920225863539796E-321
(r 0 .r 2 ) = 3.9426438538131474E-321
(r 0 .r 3 ) = 3.260833262552272E-322
(r 0 .r 4 ) = 0.0000000000000000
(r 1 .r 2 ) = -4.440892098500626E-016
(r 1 .r 3 ) = 0.0000000000000000
(r 1 .r 4 ) = -8.326672684686741E-017
(r 2 .r 3 ) = -8.0817641970012523E-016
(r 2 .r 4 ) = -1.8735013540549517E-016
(r 3 .r 4 ) = -6.9388939839071945E-018
estimation de l'erreur relative :
0.2000000000000000

```

FIGURE 2.11 – Fichier résultat pour  $lap_3$

La suite des résidus  $r_k$  convergent très rapidement vers la limite théorique. Cependant, pour la suite des  $x_k$ , certaines composantes convergent moins rapidement que d'autres ce qui explique une erreur relative bien au dessus de celles vu précédemment pour les autres matrices.

Les produits scalaires sont tous vérifiés et égaux à 0 à la précision machine près. L'espace de Krylov engendré par la base orthogonale des résidus  $r_k$  pour cette matrice est bien vérifié numériquement

## Matrices $tri_{alpha}$

Pour ces types de matrice, nous avons décidé de prendre des matrices de dimension 10 avec des  $\alpha$  égaux à 2, 5 et 20.

Pour les 3 matrices confondues, la convergence est parfaite pour le vecteur solution  $x_k$  l'erreur relative.

```

5 1.0000000000000000 1.0000000000000000 1.0000000000000000 1.0000000000000000 1.0000000000000000 1.0000000000000000 1.0000000000000000
1.0000000000000000 1.0000000000000000 1.0000000000000000

```

FIGURE 2.12 – Convergence parfaite de la suite  $x_k$  à la  $5^{me}$  itération pour les matrices  $tri_{alpha_{10}}$

L'erreur relative est donc nulle pour l'ensemble de ces matrices.

Nous observons également une convergence très rapide des résidus et des produits scalaires tous respectés.

```

(r      0 .r      1 ) = -1.6549716938744235E-319
(r      0 .r      2 ) = -3.0977915994246158E-321
(r      0 .r      3 ) = 0.0000000000000000
(r      0 .r      4 ) = 0.0000000000000000
(r      0 .r      5 ) = 0.0000000000000000
(r      0 .r      6 ) = 0.0000000000000000
(r      1 .r      2 ) = 1.2931877790833823E-012
(r      1 .r      3 ) = -4.7073456244106637E-014
(r      1 .r      4 ) = 1.0189874118105013E-015
(r      1 .r      5 ) = -2.8829189876273739E-017
(r      1 .r      6 ) = 1.5362043941141139E-018
(r      2 .r      3 ) = 4.1524943206194820E-017
(r      2 .r      4 ) = -3.1405334706229287E-018
(r      2 .r      5 ) = 1.5547653189334323E-019
(r      2 .r      6 ) = -5.7553470233283642E-021
(r      3 .r      4 ) = 7.7530540215344237E-020
(r      3 .r      5 ) = -1.9295594602799997E-021
(r      3 .r      6 ) = 2.4974844145580805E-022
(r      4 .r      5 ) = 1.4936241102263203E-022
(r      4 .r      6 ) = -2.9668407185213662E-024
(r      5 .r      6 ) = 8.0499081584687750E-026

```

FIGURE 2.13 – Produits scalaires entre les résidus pour  $W$

La matrice étant très bien conditionnée, le choix du  $\alpha$  importe peu. Il doit simplement être supérieur à 1 de manière à ce que la matrice soit définie positive.

### Matrice de Wilson

Pour cette matrice, certains résidus ont du mal à converger vers zéro. En particulier celui à la deuxième itération :

```

r      0  7.1639518646980749E-322  1.4010923440210227E-003 -2.0823028461624377E-003  5.3544930608207669E-005
r      1 -32.000000000000000000 -7.3619282957120002E-010 -5.2962965263469997E-010 -7.687297422238219E-010
r      2 -0.23747079037800489 -0.15210996563573786 0.11623367697594489 0.23425429553264721
r      3 -1.5724095509345171E-003 -9.7546299276141113E-004 8.0486077895987052E-003 -6.2210097611014037E-003
r      4  1.4010923440210227E-003 -2.0823028461624377E-003 5.3544930608207669E-005 4.1646057185144858E-005
r      5 -7.3619282957120002E-010 -5.2962965263469997E-010 -7.687297422238219E-010 -7.2549848901648072E-010

```

FIGURE 2.14 – Suite des résidus  $r_k$  pour  $W$

Ce manque de convergence pour certains de ces vecteurs impliquent des produits scalaires très éloignés de 0 et donc des résultats numériques ne permettant pas de vérifier les résultats mathématiques théoriques avec précision.

Ce manque de convergence et ce manque de précision sont dus au conditionnement élevé de la matrice carrée de dimension 4.

```

cond =
4488.000000000000

```

FIGURE 2.15 – Conditionnement de la matrice de Wilson

Cependant la suite  $x_k$  converge assez rapidement vers la solution malgré le conditionnement :

```

4  0.99999999997569455      0.99999999998249978      0.99999999997462652      0.99999999997605027

```

FIGURE 2.16 – Vecteur solution  $x_k$  à partir de la 4<sup>me</sup> itération pour  $W$

Enfin l'estimation de l'erreur relative est  $10^{-11}$  ce qui est assez faible.

### 2.2.2 Matrices avec perturbation

A toi de jouer beau goss

De même Les fichiers de résultats sont obtenus en entrant les données dans un fichier de données puis en donnant en entrée de l'exécutable ce fichier de donnée qui contient en plus la matrice de perturbation  $\delta A$  comme ci-dessous :

```

nbw@uthbawt-ubuntu: /Documents/GNS/Projets S1/projet_amsh/Gradient/Avec_Perturbation$ ./gradientper_t2 < dif_0_-10_1_0.d
A =
2.0000000000000000 -1.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000
-1.0000000000000000 2.0000000000000000 -1.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000
0.0000000000000000 -1.0000000000000000 2.0000000000000000 -1.0000000000000000 0.0000000000000000 0.0000000000000000
0.0000000000000000 -1.0000000000000000 2.0000000000000000 -1.0000000000000000 -1.0000000000000000 0.0000000000000000
0.0000000000000000 0.0000000000000000 0.0000000000000000 -1.0000000000000000 2.0000000000000000 -1.0000000000000000
0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 -1.0000000000000000 2.0000000000000000
delta A =
-2.0000000000000000E-016 -0.0000000000000000 0.0000000000000000 0.0000000000000000 -0.0000000000000000 -0.0000000000000000
-0.0000000000000000 2.0000000000000000E-013 -0.0000000000000000 -0.0000000000000000 -0.0000000000000000 -0.0000000000000000
0.0000000000000000 -0.0000000000000000 0.0000000000000000 1.0000000000000000E-015 0.0000000000000000 0.0000000000000000
0.0000000000000000 0.0000000000000000 1.0000000000000000E-015 0.0000000000000000 1.0000000000000000E-013 0.0000000000000000
-0.0000000000000000 -0.0000000000000000 0.0000000000000000 1.0000000000000000E-013 -2.0000000000000000E-014 1.0000000000000000E-014
-0.0000000000000000 -0.0000000000000000 0.0000000000000000 0.0000000000000000 1.0000000000000000E-014 2.0000000000000000E-014
b =
1.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 1.0000000000000000
Nmax = 100
tol = 1.0000000000000000E-010
cond = 24.0000000000000000
x0 = 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000
suite de vecteurs x et le n'd'itération :
1 0.49999999999999756 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.49999999999999756
2 0.6666666666665342 0.3333333333331006 0.0000000000000000 0.0000000000000000 0.3333333333330672 0.6666666666664676
3 0.99999999999980915 0.9999999999994796 0.9999999999995677 0.99999999999948674 0.9999999999996879 0.99999999999976885
estimation de l'erreur relative :
5.1325610428430781E-013
norme infini de delta A sur la norme infini de A :
1.0000000000000000E-013
second membre de l'inégalité du conditionnement :
2.0000000000000000E-012
l'inégalité du conditionnement est respectée

```

FIGURE 2.17 – Exemple de fichier de résultat avec une matrice dif 6 avec perturbation

Les propriétés des fichiers de résultat entre les deux tests étant similaires, nous décidons de travailler avec le **test d'arrêt 2** dans cette partie.

## Matrices dif

En petites dimensions la suite des  $x_k$  converge très rapidement vers la solution malgré les perturbations. On notera que pour ces matrices le conditionnement est très faible en petite dimension. Plus intéressant observons les résultats obtenus en dimension 50 :

```

138      25      0.99999999999987166      0.99999999999985198      0.999999999999962164      0.999999999999961064      0.999999999999928835      0.999999999999916566      0.999999999999896061
139      0.99999999999985617      0.999999999999875788      0.999999999999884459      0.999999999999894329      0.999999999999922995      0.99999999999983863      0.999999999999902411
140      0.999999999999870548      0.999999999999880385      0.999999999999837885      0.999999999999827638      0.999999999999794265      0.999999999999803846      0.999999999999758682
141      0.99999999999980542      0.999999999999777039      0.999999999999811651      0.999999999999810030      0.999999999999839684      0.999999999999817923      0.999999999999838629
142      0.999999999999818923      0.999999999999843470      0.999999999999813649      0.999999999999827294      0.999999999999805511      0.999999999999836964      0.999999999999834133
143      0.999999999999883515      0.999999999999881684      0.999999999999890994      0.999999999999888733      0.999999999999896438      0.999999999999877542      0.999999999999878320
144      0.999999999999858757      0.999999999999859823      0.999999999999843303      0.999999999999837497      0.999999999999820721      0.999999999999845846      0.999999999999879341
145      0.999999999999844567
146      estimation de l'erreur relative :
147      2.413108766325500E-012
148      norme infnft de delta A sur la norme infnft de A :
149      1.0000000000000000E-013
150      second membre de l'inégalité du conditionnement :
151      1.3000000000000000E-010
152      l'inégalité du conditionnement est respectée

```

FIGURE 2.18 – Résultats obtenus pour la matrice dif en dimension 50

On observe que la suite des  $x_k$  converge au bout de la 25<sup>eme</sup> itération. L'erreur relative reste très faible et l'inégalité du conditionnement est respectée.

On voit donc que même en grande dimension on converge bien et tout de même rapidement vers la solution.



## Matrice elec

la matrice est petite de dimension et son conditionnement assez faible. La suite des  $x_k$  converge en seulement 3 itérations.

```
| A =
  13.000000000000000    -8.000000000000000    -3.000000000000000
 -8.000000000000000    10.000000000000000    -1.000000000000000
 -3.000000000000000    -1.000000000000000    11.000000000000000
delta_A =
 -1.3000000000000001E-014  -8.000000000000006E-015  -3.000000000000003E-013
 -8.000000000000006E-015  -1.000000000000002E-014  -0.000000000000000
 -3.000000000000003E-013  -0.000000000000000    1.100000000000000E-013
b =
  2.000000000000000    1.000000000000000    7.000000000000000
Nmax =
    100
tol =
  1.000000000000000E-010
cond =
  10.936000000000000
x0 =
  0.000000000000000    0.000000000000000    0.000000000000000
suite de vecteurs x et le n°d'itération :
   1  0.22929936305732637    0.11464968152866319    0.80254777070064232
   2  1.0289378493917170    0.88885235120025841    0.98059848733972221
   3  1.00000000000000768    1.00000000000000677    1.00000000000000442
estimation de l'erreur relative :
  7.6827433304054926E-014
norme infini de delta A sur la norme infini de A :
  2.3076923076923079E-014
second membre de l'inégalité du conditionnement :
  2.5236923076923078E-013
l'inégalité du conditionnement est respectée
```

FIGURE 2.19 – Résultats obtenus pour la matrice elec

L'erreur relative est toujours très faible et l'inégalité du conditionnement est respectée, cette matrice ne pose donc pas de problème malgré les perturbations.

## Matrice elecmodif

Cette matrice reste la même que la précédente mais on a mis un signe moins sur le premier élément de la matrice. Elle reste donc symétrique mais elle n'est plus définie positive car son premier mineur est négatif.

```

A =
-13.000000000000000      -8.000000000000000      -3.000000000000000
-8.000000000000000      10.000000000000000     -1.000000000000000
-3.000000000000000     -1.000000000000000      11.000000000000000
delta_A =
 1.300000000000000E-014   8.000000000000000E-014  -3.000000000000002E-015
 8.000000000000000E-014   0.000000000000000      1.000000000000001E-015
-3.000000000000002E-015   1.000000000000001E-015   0.000000000000000
b =
-24.000000000000000      1.000000000000000      7.000000000000000
Nmax =
    100
tol =
 1.000000000000000E-010
cond =
 2.698500000000000
x0 =
 0.000000000000000      0.000000000000000      0.000000000000000
suite de vecteurs x et le n°d'itération :
    1   2.7016723610861373   -0.11256968171192239   -0.78798777198345671
    2   0.98891837250978543    1.0611889996189439    0.94056922739215487
    3   1.0000000000000087    0.9999999999999822    1.0000000000000020
estimation de l'erreur relative :
 8.6597395920761453E-015
norme infini de delta A sur la norme infini de A :
 6.1538461538461540E-015
second membre de l'inégalité du conditionnement :
 1.6606153846153847E-014
l'inégalité du conditionnement est respectée

```

FIGURE 2.20 – Résultats obtenus pour la matrice elecmodif

Malgré cela on observe de bon résultats notamment avec l'erreur relative faible et la suite des  $x_k$  qui converge aussi vite. L'inégalité du conditionnement est elle aussi respectée.

### Matrices de Hilberts

Ces Matrices sont déjà plus compliquées à utiliser car elles possèdent un grand conditionnement. De plus en y ajoutant des perturbations cela n'arrange pas les choses. En dimension 4 cela se passe encore plutôt bien mais au-delà on observe des complications.



```

| A =
1.0000000000000000 0.5000000000000000 0.3333333333333331 0.2500000000000000 0.2000000000000001 0.1666666666666666
0.5000000000000000 0.3333333333333331 0.2500000000000000 0.2000000000000001 0.1666666666666666 0.14285714285714285
0.3333333333333331 0.2500000000000000 0.2000000000000001 0.1666666666666666 0.14285714285714285 0.1250000000000000
0.2500000000000000 0.2000000000000001 0.1666666666666666 0.14285714285714285 0.1250000000000000 0.1111111111111111
0.2000000000000001 0.1666666666666666 0.14285714285714285 0.1250000000000000 0.1111111111111111 0.1000000000000001
0.1666666666666666 0.14285714285714285 0.1250000000000000 0.1111111111111111 0.1000000000000001 9.09090909090912E-002
delta_A =
-1.000000000000000E-014 -5.000000000000002E-014 0.000000000000000 0.000000000000002E-016 0.000000000000000 -1.666666666666664E-017
-5.000000000000002E-014 3.333333333333329E-017 2.499999999999999E-017 2.000000000000002E-015 -1.666666666666667E-014 -1.4285714285714284E-017
0.000000000000000 2.499999999999999E-017 0.000000000000000 -1.666666666666667E-014 -1.4285714285714284E-017 -1.250000000000001E-016
-2.500000000000002E-016 2.000000000000002E-015 -1.666666666666667E-014 -1.4285714285714284E-014 0.000000000000000 0.000000000000000
0.000000000000000 -1.666666666666667E-014 -1.4285714285714284E-017 0.000000000000000 -1.111111111111112E-016 -1.000000000000001E-015
-1.666666666666664E-017 -1.4285714285714284E-017 -1.250000000000001E-016 0.000000000000000 -1.000000000000001E-015 9.09090909090923E-017
b =
2.449999999999999 1.5928571428571427 1.2178571428571427 0.99563492063492065 0.84563492063492063 0.73654401154401161
Nmax =
100
tol =
1.000000000000000E-010
cond =
29070278.999740001
x0 =
0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
suite de vecteurs x et le n° d'itération :
1 1.5210871468871943 0.98892837829692226 0.75610891703867822 0.61814182888564484 0.52501404438234722 0.45728474656176715
2 0.93489743892529809 1.1699804214391549 1.0947766113800248 0.99240407447023415 0.89838393852941323 0.81736193343412600
3 1.0037085607339182 0.97162950690780159 1.0271946795300804 1.0292165772696154 1.0020215865015498 0.96223451189710729
4 0.99989612156088414 1.0016605064021324 0.99453198167278134 0.9945929041463219 1.0048074941785961 0.99541943774642183
5 0.99989667666632553 1.0015845747150165 0.99477752352895299 1.0034102001373364 1.0046253795213433 0.99560648191018986
6 1.0000011601291872 0.9996689361097657 1.000223770747548 0.99941847018778440 1.0006413528577713 0.99974747488076987
7 1.0000011607221204 0.9996689342650569 1.000223776589799 0.99941847014994789 1.0006413521025999 0.99974747467269731
estimation de l'erreur relative :
6.4094103372023799E-004
norme infini de delta A sur la norme infini de A :
5.000000000000002E-014
second membre de l'inégalité du conditionnement :
1.453139499870001E-006
l'inégalité du conditionnement n'est pas respectée

```

FIGURE 2.21 – Résultats obtenus pour la matrice de Hilbert en dimension 6

On observe bien une lente convergences des  $x_k$  et une erreur relative élevée. La même chose se produit en dimension supérieur mais en dimension 6 on se rend compte que l'inégalité du conditionnement n'est pas respectée étonnamment. En effet l'erreur relative est supérieure au membre de droite de l'inégalité. La différence est à  $10^{-2}$  près.

Cela est dû au conditionnement qui est plutôt élevé et qui rend compliqué les calculs en flottants.

## Matrice lap

On prendra la matrice lap en dimension 9 :

```

suite de vecteurs x et le n° d'itération :
1 1.0714285714285983 0.53571428571429913 0.53571428571429913 0.53571428571429913 0.53571428571429913 0.53571428571429913 0.53571428571429913
0.53571428571429913 1.0714285714285983 0.675000000000003912 1.00000000000000788 1.00000000000001439 1.00000000000000822 0.675000000000006122
2 1.0250000000000079 1.00000000000000365 0.675000000000003912 1.00000000000000788 1.00000000000001439 1.00000000000000822 0.675000000000006122
1.00000000000000919 1.02500000000000365 0.675000000000003912 1.00000000000000788 1.00000000000001439 1.00000000000000822 0.675000000000006122
3 1.00000000000000409 1.00000000000000859 0.750000000000004952 1.00000000000000806 1.25000000000002067 1.00000000000000793 0.750000000000008193
1.00000000000000961 1.00000000000000415 0.750000000000004952 1.00000000000000806 1.25000000000002067 1.00000000000000793 0.750000000000008193
estimation de l'erreur relative :
0.20000000000001229
norme infini de delta A sur la norme infini de A :
1.000000000000000E-013
second membre de l'inégalité du conditionnement :
9.000000000000000E-013
l'inégalité du conditionnement n'est pas respectée

```

FIGURE 2.22 – Résultats obtenus pour la matrice lap

La solution converge très vite mais certaines coordonnées des  $x_k$  convergent mal ce qui entraîne une grosse erreur relative. De ce fait l'inégalité du conditionnement n'est une nouvelle fois pas respectée.

## Matrices $tri_{\alpha}$

Comme précédemment nous avons pris cette matrice pour des valeurs de  $\alpha$  telles que 2,5 et 20 en dimension 10.

Pour les 3 matrices les résultats sont excellent malgré les perturbations.

Regardons plus en détail pour  $\alpha=5$

```

suite de vecteurs x et le n°d'itération :
  1 0.93023537976219639 1.0148022324678505 1.0148022324678505 1.0148022324678505 1.0148022324678505 1.0148022324678505 1.0148022324678505
1.0148022324678505 1.0148022324678505 0.93023537976219639 0.99818999980547463 0.99818999980547374 0.99818999980546586 0.99818999980546463
  2 0.99871868935658181 1.0065611187590731 0.99871868935648156 0.99818999980547463 0.99818999980547374 0.99818999980546586 0.99818999980546463
0.99818999980547463 1.0065611187590731 0.99871868935648156 0.99818999980547463 0.99818999980547374 0.99818999980546586 0.99818999980546463
  3 0.99998236395623297 1.0001192005270914 0.99998236395613427 1.0002496695110190 1.0002496695110210 1.0002496695110132 1.0002496695110119
0.99998236395623297 1.0001192005270914 0.99998236395613427 1.0002496695110190 1.0002496695110210 1.0002496695110132 1.0002496695110119
  4 0.99999981775656266 1.0000013838306174 0.99999981775646352 1.0000478946208942 0.99999981775646352 0.99999981775646352 0.99999981775646352
0.99999981775656266 1.0000013838306174 0.99999981775646352 1.0000478946208942 0.99999981775646352 0.99999981775646352 0.99999981775646352
  5 0.9999999999999999 1.0000000000000000 0.9999999999999999 0.9999999999999999 0.9999999999999999 0.9999999999999999 0.9999999999999999
0.9999999999999999 1.0000000000000000 0.9999999999999999 0.9999999999999999 0.9999999999999999 0.9999999999999999 0.9999999999999999
estimation de l'erreur relative :
1.001357259847850E-013
norme infini de delta A sur la norme infini de A :
1.000000000000000E-013
second membre de l'inégalité du conditionnement :
1.500000000000000E-013
l'inégalité du conditionnement est respectée

```

FIGURE 2.23 – Résultats obtenus pour la matrice  $tri_5$

On voit que la suite converge en 5 itérations, que l'erreur relative est très faible et que l'inégalité du conditionnement est respectée.

On pouvait s'attendre à de tels résultats en observant la valeur du conditionnement qui vaut 1,5

### Matrice de Wilson

Le conditionnement de cette matrice reste assez élevé mais malgré tout on obtient des résultats plutôt bon :

```

A =
10.000000000000000 7.000000000000000 8.000000000000000 7.000000000000000
7.000000000000000 5.000000000000000 6.000000000000000 5.000000000000000
8.000000000000000 6.000000000000000 10.000000000000000 9.000000000000000
7.000000000000000 5.000000000000000 9.000000000000000 10.000000000000000
delta_A =
-1.000000000000000E-015 -7.000000000000000E-013 -8.000000000000000E-014 7.000000000000000E-014
-7.000000000000000E-013 5.000000000000000E-015 0.000000000000000 0.000000000000000
-8.000000000000000E-014 0.000000000000000 0.000000000000000 -8.999999999999999E-014
7.000000000000000E-014 0.000000000000000 -8.999999999999999E-014 0.000000000000000
b =
32.000000000000000 23.000000000000000 33.000000000000000 31.000000000000000
Nmax =
100
tol =
1.000000000000000E-010
cond =
4488.0000000000000
x0 =
0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
suite de vecteurs x et le n°d'itération :
  1 1.0565498281787062 0.75939518900344516 1.0895670103092907 1.0235326460481218
  2 1.1184958803925191 0.79910087804648344 1.0597648435988547 0.96309198448691369
  3 1.1206872982560552 0.80046481710810169 1.0495528583061984 0.97069307479687206
  4 0.99999999995275235 0.9999999998809541 0.99999999995629207 0.99999999996468247
  5 0.9999999999083278 1.0000000000154914 0.9999999999605982 1.0000000000022156
estimation de l'erreur relative :
1.5491385951564677E-011
norme infini de delta A sur la norme infini de A :
7.000000000000000E-014
second membre de l'inégalité du conditionnement :
3.141600000000000E-010
l'inégalité du conditionnement est respectée

```

FIGURE 2.24 – Résultats obtenus pour la matrice de Wilson

On peut voir que malgré tout la suite converge assez vite et que l'inégalité du conditionnement est respectée.

# Conclusion

Dans la conclusion, vous devez commenter les résultats numériques par rapport à ce que l'on pouvait espérer au vu des résultats théoriques.

Dire qu'avec une complexité comme celle ci (je crois que c'est  $O(n)$  ) le gradient conjugué est très apprécié dans des problèmes d'optimisation de grande taille.

# Annexes

```
13.0 -8.0 -3.0
-8.0 10.0 -1.0
-3.0 -1.0 11.0
```

Matrice elec

```
-13.0 -8.0 -3.0
-8.0 10.0 -1.0
-3.0 -1.0 11.0
```

Matrice elecmodif

```
2.0 -1.0 0.0 0.0 0.0 0.0 0.0 0.0
-1.0 2.0 -1.0 0.0 0.0 0.0 0.0 0.0
0.0 -1.0 2.0 -1.0 0.0 0.0 0.0 0.0
0.0 0.0 -1.0 2.0 -1.0 0.0 0.0 0.0
0.0 0.0 0.0 -1.0 2.0 -1.0 0.0 0.0
0.0 0.0 0.0 0.0 -1.0 2.0 -1.0 0.0
0.0 0.0 0.0 0.0 0.0 -1.0 2.0 -1.0
0.0 0.0 0.0 0.0 0.0 0.0 -1.0 2.0
```

Matrice dif de dim 8

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix}.$$

Matrice de Hilbert de dim 5

```
4 -1 0 -1 0 0 0 0 0
-1 4 -1 0 -1 0 0 0 0
0 -1 4 0 0 -1 0 0 0
-1 0 0 4 -1 0 -1 0 0
0 -1 0 -1 4 -1 0 -1 0
0 0 -1 0 -1 4 0 0 -1
0 0 0 -1 0 0 4 -1 0
0 0 0 0 -1 0 -1 4 -1
0 0 0 0 0 -1 0 -1 4
```

Matrice Laplacienne\_3 (de dim 3<sup>2</sup>)

```
10 1 0 0 0 0 0 0 0 0
1 10 1 0 0 0 0 0 0 0
0 1 10 1 0 0 0 0 0 0
0 0 1 10 1 0 0 0 0 0
0 0 0 1 10 1 0 0 0 0
0 0 0 0 1 10 1 0 0 0
0 0 0 0 0 1 10 1 0 0
0 0 0 0 0 0 1 10 1 0
0 0 0 0 0 0 0 1 10 1
0 0 0 0 0 0 0 0 1 10
```

Matrice tri\_α de dim 10 avec α = 5

```
10 7 8 7
7 5 6 5
8 6 10 9
7 5 9 10
```

Matrice de Wilson

# Bibliographie

- [1] André Draux *Analyse numérique*, poly, chapitre 2 *Les méthodes de descente*.
- [2] Maria Kazakova *GM3 Analyse numérique I*, Année 2020-2021, section 1.2.4
- [3] Daniel Kauth *Les méthodes de Krylov Optimisation numérique Méthodes du gradient conjugué linéaire*, chapitre 5.1, 5 novembre 2009.