

Systèmes Linéaires en Grande Dimension

Cimino-Ferre

January 10, 2025

Abstract

Ce document présente différentes méthodes de résolution numérique des systèmes linéaires à grandes dimensions. Ces systèmes sont fréquemment rencontrés dans de nombreuses applications en sciences et ingénierie. Nous discuterons des problèmes associés à leur résolution, des méthodes itératives populaires comme celles de Jacobi, Gauss-Seidel (GS), Successive Over-Relaxation (SOR) et Symmetric Successive Over-Relaxation (SSOR), avant de comparer leur efficacité pour des matrices spécifiques telles que les matrices tridiagonales et de Poisson.

1 Introduction

La résolution de systèmes linéaires à grandes dimensions est un problème fondamental en mathématiques appliquées aux sciences et à l'ingénierie. La forme générale d'un tel système est $A\mathbf{x} = \mathbf{b}$, où A est une matrice carrée de taille $n \times n$, \mathbf{x} est le vecteur inconnu de taille n , et \mathbf{b} est le vecteur des termes constants. Les méthodes directes, comme la méthode de Gauss, peuvent être coûteuses en termes de mémoire et de temps de calcul lorsque n est très grand. C'est pourquoi des méthodes itératives sont souvent privilégiées pour de tels systèmes. Ces méthodes, qui génèrent une suite de vecteurs approchant la solution, sont particulièrement utiles lorsque la matrice A possède des propriétés spécifiques (comme les matrices symétriques définies positives par exemple (SDP)).

En particulier, les méthodes de Jacobi, GS, SOR et de SSOR sont largement utilisées. Ces méthodes sont adaptées aux systèmes linéaires avec des matrices creuses, comme les matrices tridiagonales et de Poisson, qui apparaissent fréquemment dans des applications en physique et en ingénierie.

2 Méthodes Itératives

Les méthodes itératives permettent d'approximer la solution d'un système linéaire de manière progressive. Les techniques suivantes sont parmi les plus utilisées dans la résolution de systèmes à grandes dimensions.

2.1 Méthode de Jacobi

La méthode de Jacobi est une méthode itérative simple qui consiste à isoler chaque variable dans l'équation $A\mathbf{x} = \mathbf{b}$ pour exprimer x_i en fonction des autres variables x_j :

$$x_i^{(k+1)} = \frac{b_i - \sum_{j \neq i} a_{ij} x_j^{(k)}}{a_{ii}}.$$

Ici, k représente le numéro de l'itération. La méthode de Jacobi est simple à mettre en œuvre, mais elle peut être lente à converger, notamment pour des systèmes de grande taille.

2.2 Méthode de Gauss-Seidel (GS)

La méthode de Gauss-Seidel améliore la méthode de Jacobi en utilisant les nouvelles valeurs de x_i dès qu'elles sont disponibles. L'itération de Gauss-Seidel est donnée par :

$$x_i^{(k+1)} = \frac{b_i - \sum_{j < i} a_{ij} x_j^{(k+1)} - \sum_{j > i} a_{ij} x_j^{(k)}}{a_{ii}}.$$

L'avantage principal de la méthode de Gauss-Seidel par rapport à Jacobi est qu'elle converge généralement plus rapidement, bien qu'elle soit également conditionnée par la structure de la matrice A .

2.3 Méthode de Successive Over-Relaxation (SOR)

La méthode SOR est une généralisation de la méthode de Gauss-Seidel, dans laquelle un facteur de relaxation ω est introduit pour accélérer la convergence. L'itération de SOR est donnée par :

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j < i} a_{ij} x_j^{(k+1)} - \sum_{j > i} a_{ij} x_j^{(k)} \right).$$

Le paramètre ω est choisi de manière à optimiser la vitesse de convergence de l'itération. Si $\omega = 1$, la méthode correspond à la méthode de Gauss-Seidel. Un choix approprié de ω peut considérablement améliorer la vitesse de convergence.

2.4 Méthode SSOR (Symmetric Successive Over-Relaxation)

La méthode SSOR combine deux passes de SOR : une première passe de SOR pour résoudre un système $A\mathbf{x} = \mathbf{b}$ et une deuxième passe de SOR pour résoudre le système symétrique associé. Cela peut améliorer la convergence, en particulier pour les systèmes avec une matrice A symétrique ou de Poisson. L'itération SSOR est donc une combinaison des étapes de relaxation successives avant et après.

3 Comparaison

Nous comparons maintenant les performances des méthodes de Jacobi, Gauss-Seidel, SOR et SSOR pour des matrices tridiagonales et de Poisson. Pour chaque méthode, nous analysons le nombre d'itérations nécessaires pour atteindre une tolérance donnée et le temps de calcul.

3.1 Comparaison de Jacobi avec Matrice Sparse et Matrice Dense

Que ce soit pour la matrice tridiagonale ou celle de poisson, toutes deux contiennent un grand nombre de zéros à n élevé. Il est donc potentiellement intéressant de stocker ces matrices sous forme de matrices creuses (sparse). Comparons donc les temps de convergence de la méthode de Jacobi (cf numéro Jacobi!!) pour une matrice très grande ($n = 10000$).

exemple pour $n=10000$ avec matrice tridiagonale
nombre d'itérations : 24
temps (dense) : 751.7574 secondes
temps (sparse): 0.0010 secondes

On observe une nette différence entre la matrice sparse et dense pour la vitesse de calcul dans la méthode de Jacobi. Nous continuerons donc de travailler avec des matrices encryptées.

3.2 Comparaison des formats de stockages CSR, CSC et COO

La différence de temps d'exécution entre les différents formats de stockage est présente mais minime, les temps de convergence suivent les mêmes tendances (cf Figure 2).

3.3 Matrices Tridiagonales

Les matrices tridiagonales sont des matrices carrées où les éléments non nuls se trouvent sur la diagonale principale et les deux diagonales adjacentes. Un exemple de matrice tridiagonale est :

$$A = \begin{bmatrix} d_1 & c_1 & 0 & \dots & 0 \\ a_2 & d_2 & c_2 & \dots & 0 \\ 0 & a_3 & d_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & d_n \end{bmatrix}.$$

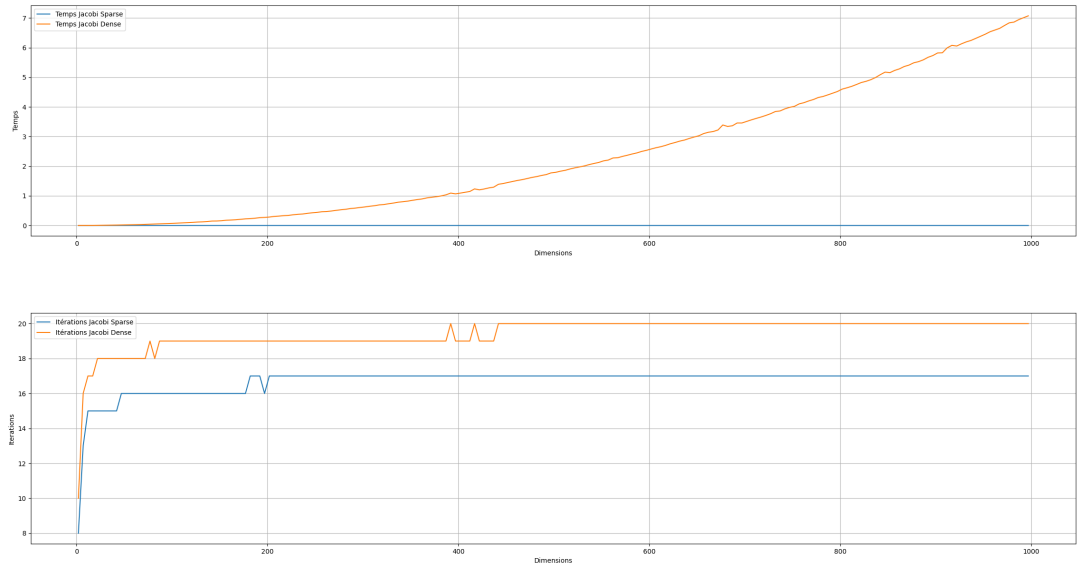


Figure 1: Comparaison de jacobi sparse et jacobi dense en fonction du temps et des itérations

3.3.1 Matrices Tridiagonales avec $h = \frac{1}{n+1}$

cf. Figure 3

cf. Figure 4

cf. Figure 5

3.3.2 Matrices Simple $D=5$ et $LU=1$

cf. Figure 6

cf. Figure 7

cf. Figure 8

On remarque que les méthodes sont très proches en termes d'itérations mêmes pour de très grands n

3.4 Matrices de Poisson

Les matrices de Poisson sont souvent utilisées dans la résolution de problèmes aux dérivées partielles, comme les équations de la chaleur et de la diffusion. Ces matrices sont de type tridiagonal par blocs.

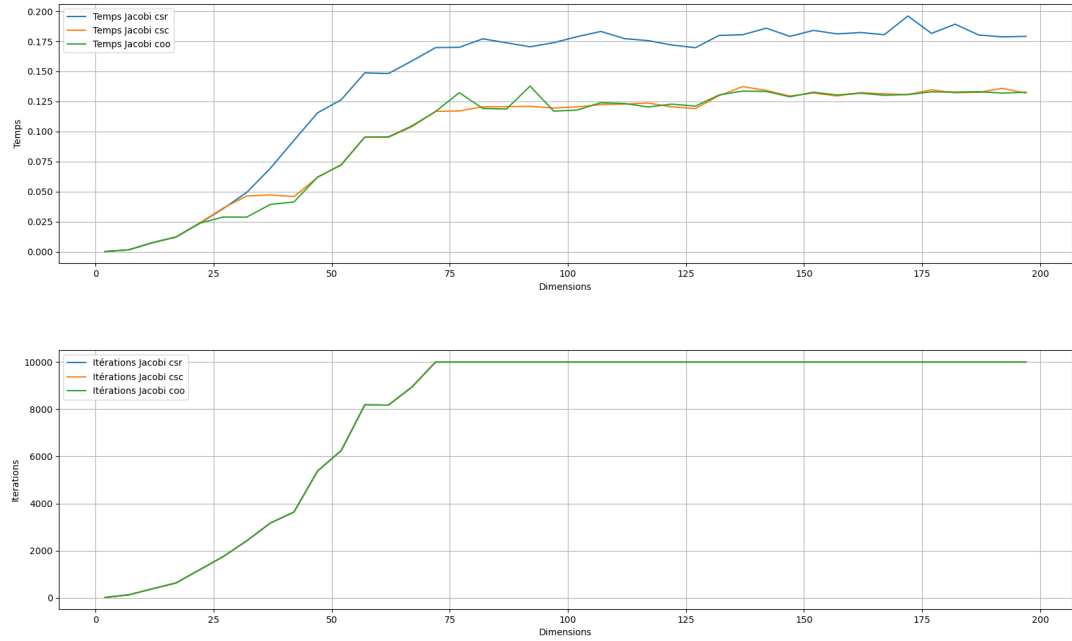


Figure 2: Comparaison csr, csc, coo en fonction du temps et des itérations

Le calcul pour la machine est beaucoup plus lourd que pour les matrices tester précédemment car la dimension vaut alors n^2

cf. Figure 9

cf. Figure 10

cf. Figure 11

3.5 Comparaison des Méthodes

Jacobi est la méthode qui demande le plus d'itération mais qui pour autant est la plus rapide des méthodes, à favoriser si on ne se préoccupe pas de la mémoire sinon à éviter. GS demande en général 2 fois moins d'itérations pour finir mais est extrêmement long par rapport aux autres méthodes. Si l'on veut prioriser le temps autant implémenter la méthode de Jacobi et sinon s'il est possible d'implémenter une autre méthode tel que la méthode Sor et Ssor alors Gauss-Seidel n'a aucun avantage. De plus, la méthode SOR est très rapide en terme de temps et demande très peu d'itération pour une erreur convenable. Finalement, la méthode Ssor demande le moins itérations parmi toutes nos méthodes en terme de temps elle est juste derrière la méthode sor

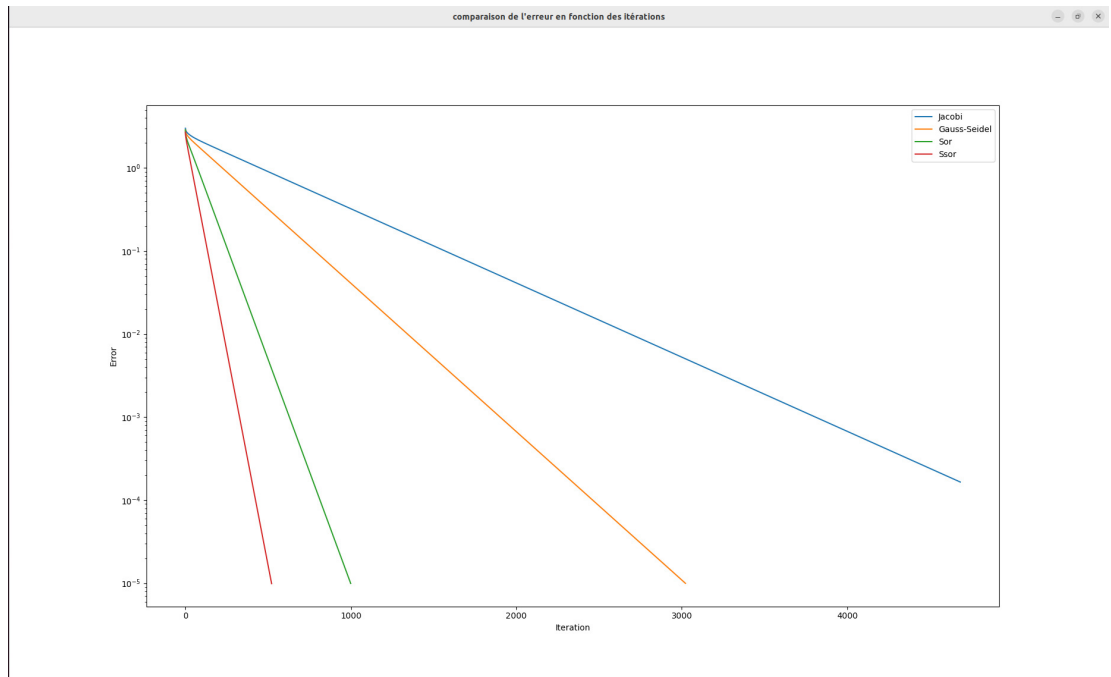


Figure 3: Comparaison de l'erreur en fonction de chaque méthode

Dans l'idéal s'il est possible d'implémenter la méthode SSOR ou SOR ce sera l'idéal à la fois en termes de temps et de mémoire

4 Etude du Coefficient de Relaxation pour la méthode SOR et SSOR

4.1 Recherche du meilleur coefficient de relaxation

Il est intéressant de chercher quel est le ω optimal pour la matrice considérée en fonction de n c'est ce que nous avons cherché à faire dans un dernier point. La méthode pour trouver les w ne trouve pas toujours les mêmes coefficients de relaxation pour autant on observe la même tendance pour n qui augmente autrement dit, pour n fixée le w trouvé est approximativement le même.

cf. Figure 12

cf. Figure 13

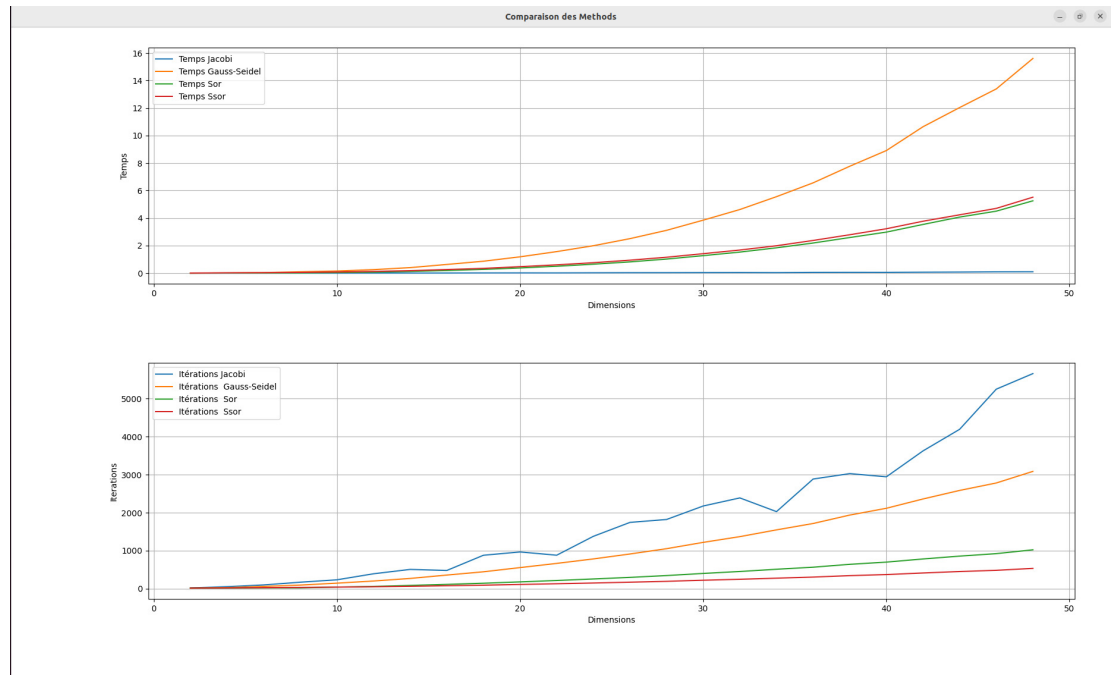


Figure 4: Comparaison du temps et du nombres d'itérations de chaque méthode en fonction de la dimension

5 Sources

Black, Noel Moore, Shirley. "Successive Overrelaxation Method". Math-World..

A. Hadjidimos, Successive overrelaxation (SOR) and related methods, Journal of Computational and Applied Mathematics 123 (2000), 177–199..

Correction de TP d'informatique de l'Ens Rennes.

Pour la méthode Ssor : Howard C. Eldman : Iterative Methods for Linear System. Department of Computer Science, University of Maryland).

Thierry Gallouët et Raphaële Herbin : Université Aix Marseille, Licence de mathématiques, Cours d'Analyse numérique).

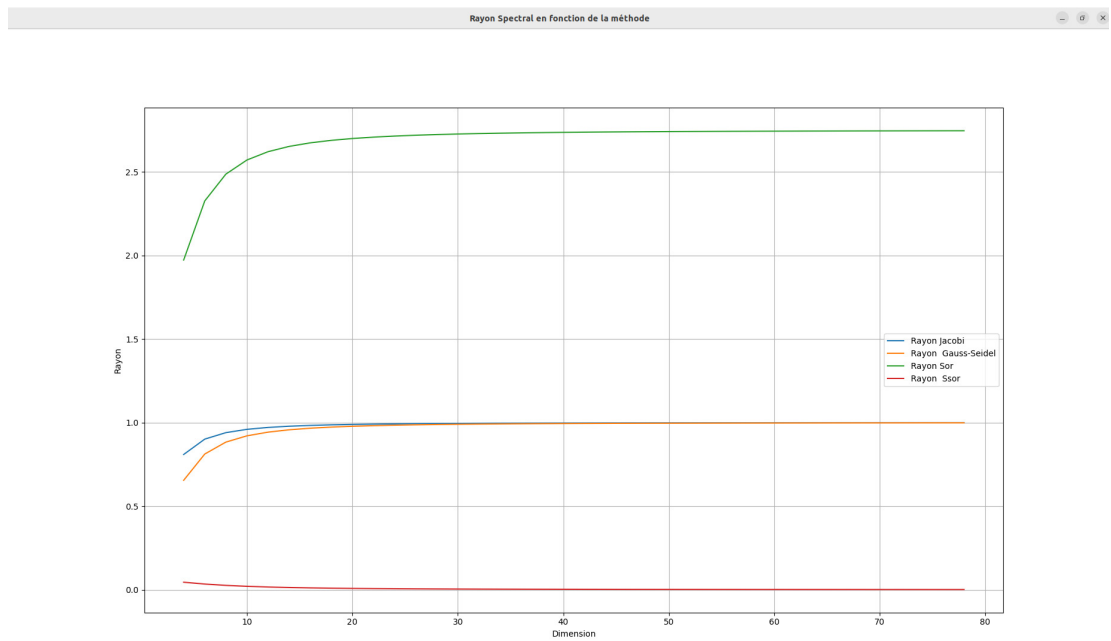


Figure 5: Comparaison des rayons spectraux en fonctions de la dimension

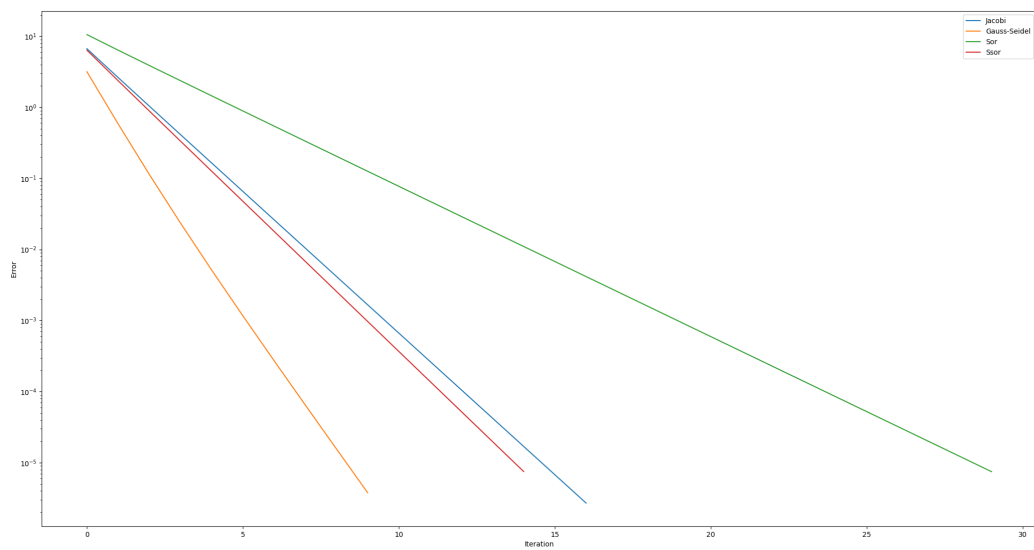


Figure 6: Comparaison de l'erreur en fonction des itérations

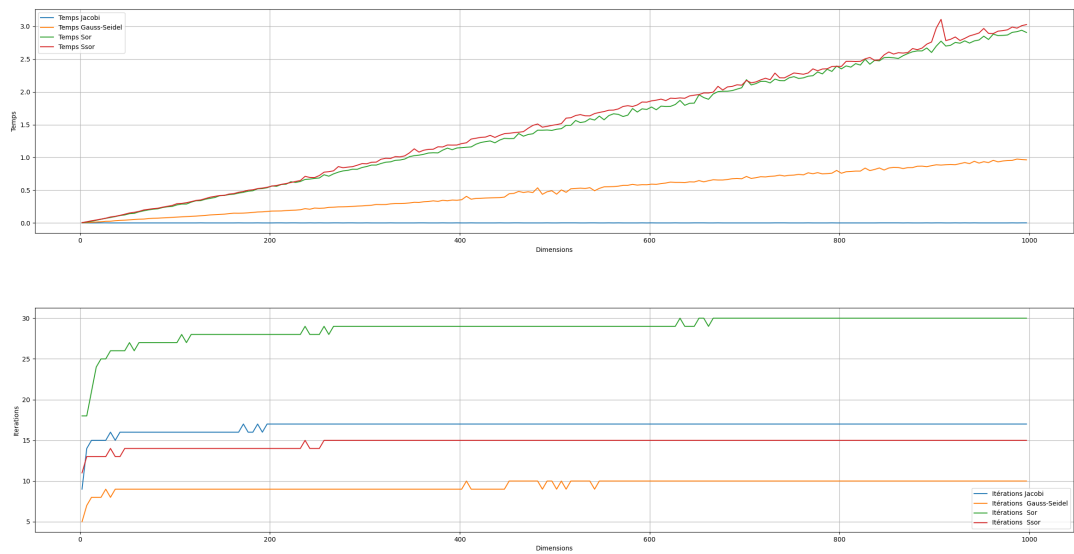


Figure 7: Comparaison des methods en fonctions du temps et des itérations

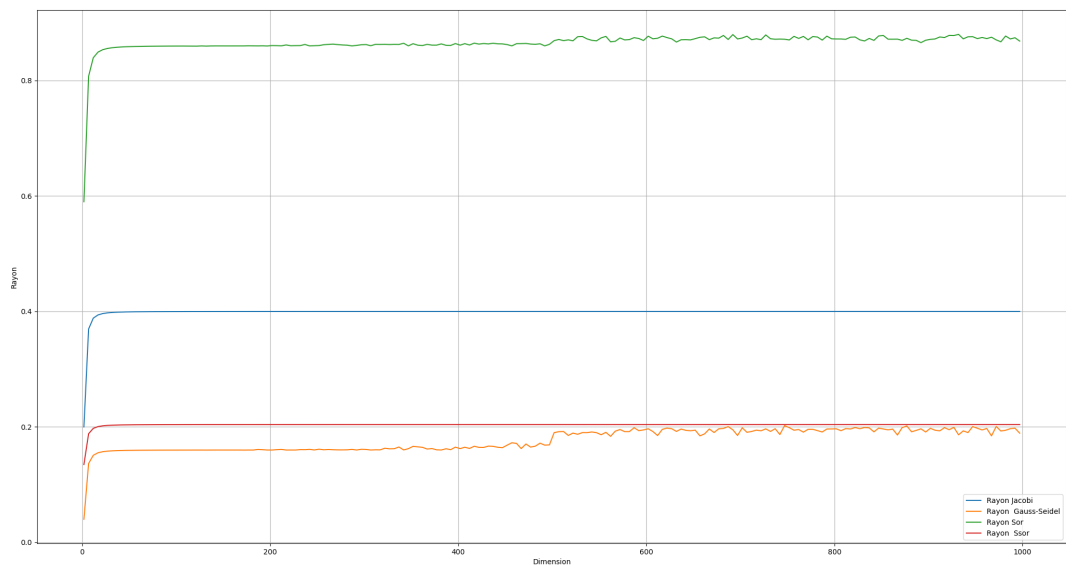


Figure 8: Comparaison des methods en fonctions du temps et des itérations

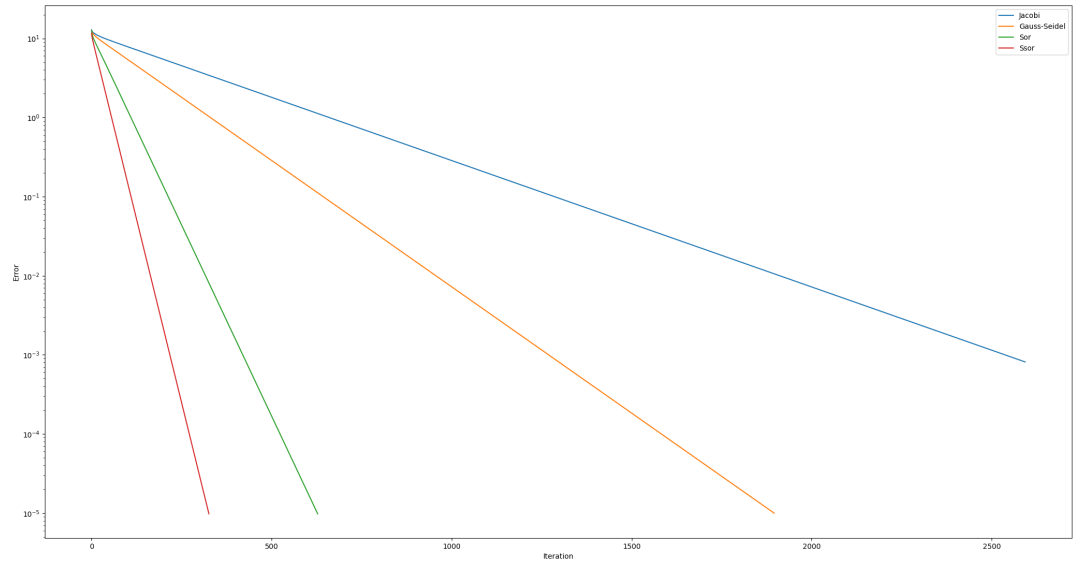


Figure 9: Comparaison des erreurs en fonction des itérations pour chaque méthode

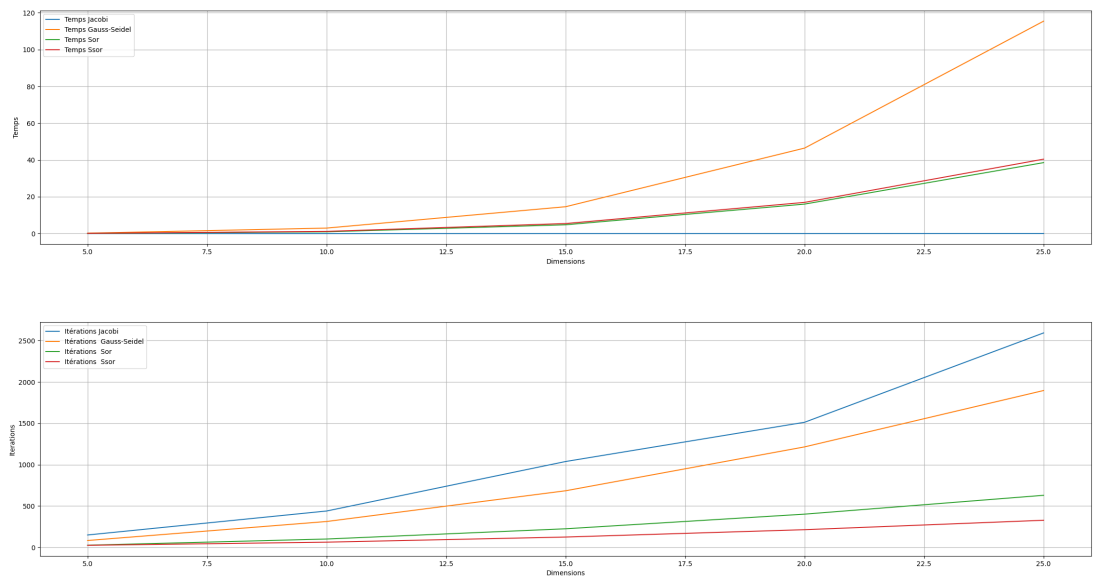


Figure 10: Comparaison temps itérations en fonctions de la dimension (à élever au carré sur le graphique)

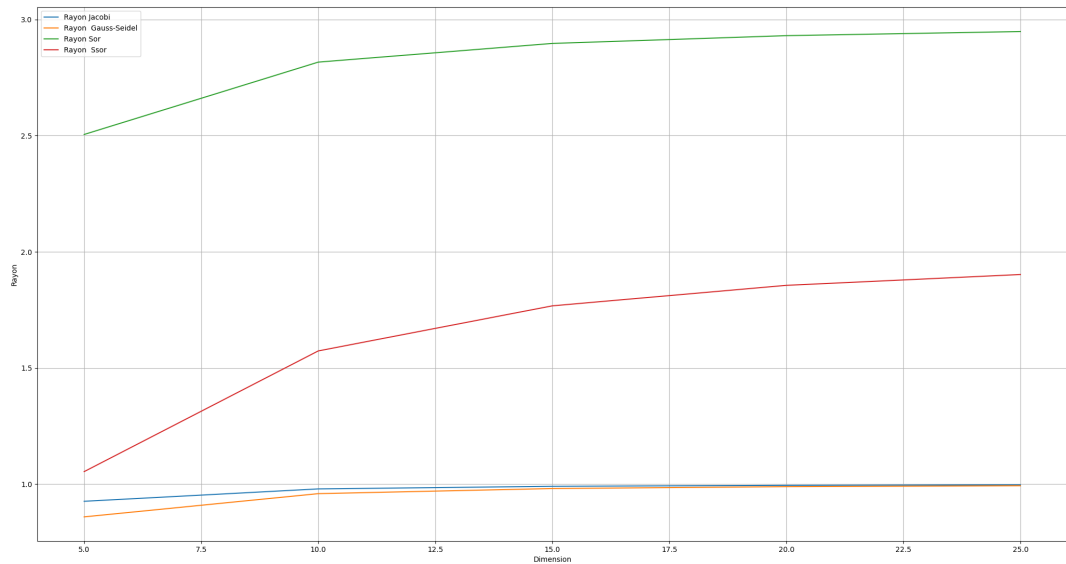


Figure 11: Comparaison temps itérations en fonctions de la dimension (à élever au carré sur le graphique)

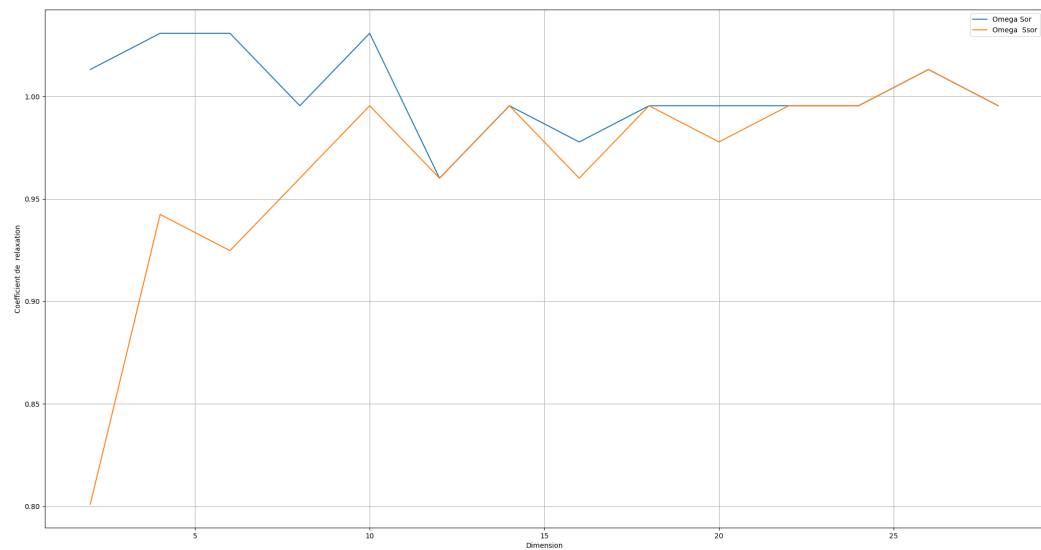


Figure 12: Coefficient de relaxation en fonction de la méthode

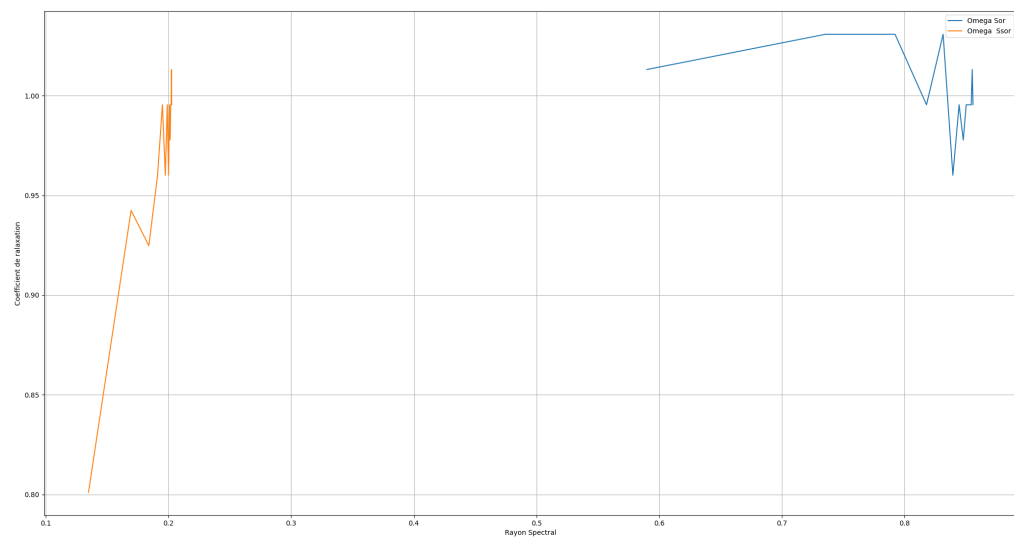


Figure 13: Rayon spectral en fonction du coefficient de relaxation