

Projet de MPNA : Méthode des itérations simultanées

Matthias BEAUPÈRE, Pierre GRANGER

Rapport MPNA - CHPS - 18 février 2019

Table des matières

1	Introduction	1
2	Problématique	1
3	Approche utilisée	1
4	Cas séquentiel	1
4.1	Description de l'algorithme général	1
4.1.1	Procédé de Gram-Schmidt	2
4.1.2	Décomposition de Schur	2
4.1.3	Sélection de k vecteurs propres	2
4.1.4	Calcul de la précision	3
4.1.5	Locking	3
4.2	Étude de performances théorique	3
4.3	Étude de performances pratique	3
4.4	Ajout d'une méthode de déflation "Locking"	3
4.5	Conclusions	3
5	Cas parallèle	3
5.1	Approche utilisée	3
5.2	Étude de performances théorique	3
5.3	Étude de performances pratique	3
5.4	Conclusions	3
6	Conclusion générale	3

1 Introduction

2 Problématique

3 Approche utilisée

4 Cas séquentiel

4.1 Description de l'algorithme général

Données du problèmes :

- m : taille du sous-espace de Krylov
- k : nombre de vecteurs propres demandé
- p : précision demandé
- A : matrice de taille $n * n$ donnée en entrée

— N : nombre d'itérations

Algorithm 1 Algorithme général

```

1:  $Q \leftarrow rand()$ 
2: while  $i = 0..N - 1$  OU  $\min(\text{precisions}) < p$  do
3:    $Z = AQ$ 
4:   Gram-Schmidt  $Q$ 
5:   Projection  $B = Z^t AZ$ 
6:   Décomposition de Schur  $B = Y^t R Y$ 
7:   Retour dans l'espace d'origine  $Q = ZY$ 
8:   Sélection des  $k$  vecteur propres
9:   Calcul de la précision
10:  Locking
11: end while

```

Dans les paragraphes suivant sont détaillés chaque étape de l'algorithme.

4.1.1 Procédé de Gram-Schmidt

On utilise une décomposition QR avec le procédé de Gram-Schmidt pour orthonormaliser la matrice Q . L'orthogonalisation consiste chaque vecteur de la matrice Z dans un vecteur temporaire tout en lui soustrayant son projeté sur chaque vecteur déjà ajouté. On normalise ensuite en divisant chaque vecteur par sa norme.

Algorithm 2 Algorithme de Gram-Schmidt

```

1: for  $i = 0..m - 1$  do
2:    $q_i^{temp} \leftarrow q_i$ 
3:   for  $k = 0..i$  do
4:      $q_i^{temp} \leftarrow q_i^{temp} - q_k(q_k \cdot q_i)$ 
5:   end for
6: end for
7:  $Q \leftarrow Q^{temp}$ 

```

4.1.2 Décomposition de Schur

La décomposition de Schur permet de calculer les valeurs et vecteurs propre de l'espace de Krylov, aussi appelés valeurs et vecteurs de Ritz. Pour ce calcul a été utilisé la bibliothèque `lapacke`.

4.1.3 Sélection de k vecteurs propres

En entrée du programme est précisé le nombre k de vecteurs propres désirés. La précision est calculé uniquement sur les k vecteurs de plus grande valeur propre associée. Pour sélectionner ces vecteur, on range les vecteurs par valeur propre associée puis on ne garde que les k premiers.

4.1.4 Calcul de la précision

4.1.5 Locking

4.2 Étude de performances théorique

4.3 Étude de performances pratique

4.4 Ajout d'une méthode de déflation "Locking"

Toutes les valeurs propres de la matrice possèdent une vitesse de convergence différente par la méthode des itérations simultanées. Cela nous amène à affiner et recalculer constamment des vecteurs déjà connus avec la bonne précision ce qui entraîne une perte de temps de calcul. De plus, des vecteurs propres qui ont déjà convergés jusqu'à la bonne précision peuvent perdre en précision à cause d'instabilités numériques au fil des calculs : la précision de certains vecteurs propres peut osciller au cours des itérations.

Afin de résoudre ces deux problèmes nous avons décidé de mettre en place une méthode de déflation appelée "locking". Le principe est simple : lorsqu'un vecteur propre a convergé jusqu'à la précision désirée, on le verrouille de sorte à ne plus le remultiplier par la matrice A et on diminue la taille du sous-espace de Krylov d'une unité. Néanmoins, on utilise toujours ce vecteur pour l'orthonormalisation afin qu'il guide la convergence correcte des vecteurs restants.

L'algorithme décrivant ce procédé est présenté ci-après.

Algorithm 3 Algorithme avec "locking"

```
1:  $Q \leftarrow rand()$ 
2: while  $i = 0..N - 1$  OU  $\min(\text{precisions}) < p$  do
3:    $Z = AQ$ 
4:   Gram-Schmidt  $Q$ 
5:   Projection  $B = Z^t AZ$ 
6:   Décomposition de Schur  $B = Y^t R Y$ 
7:   Retour dans l'espace d'origine  $Q = ZY$ 
8:   Sélection des  $k$  vecteur propres
9:   Calcul de la précision
10:  Locking
11: end while
```

4.5 Conclusions

5 Cas parallèle

5.1 Approche utilisée

5.2 Étude de performances théorique

5.3 Étude de performances pratique

5.4 Conclusions

6 Conclusion générale

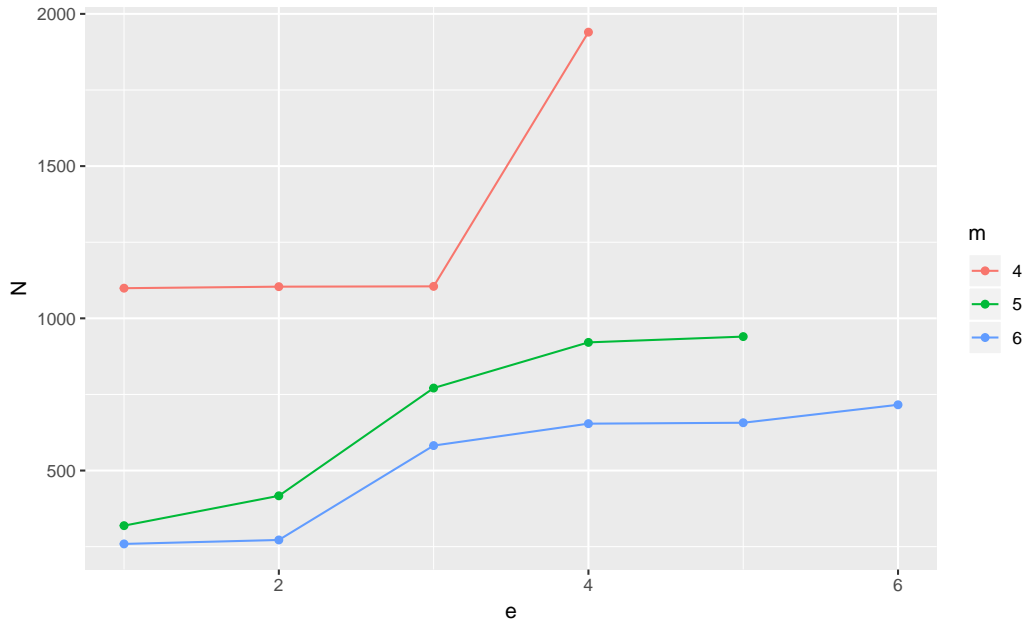


FIGURE 1 – Nombre d'itérations N nécessaires pour faire converger e valeurs propres pour différentes tailles de sous-espace de Krylov m et une précision $p = 10^{-6}$

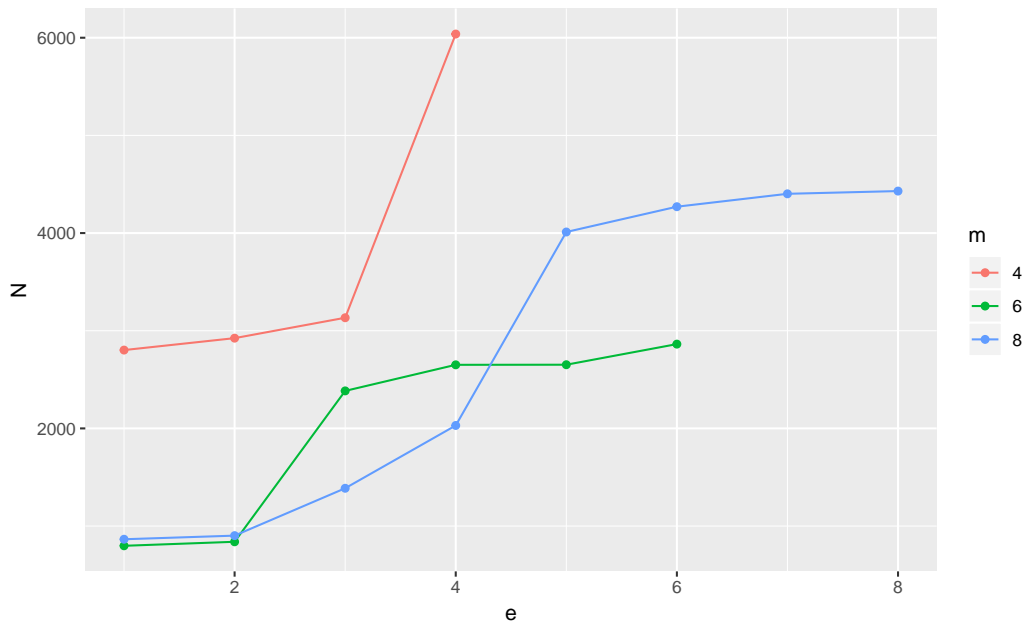


FIGURE 2 – Nombre d'itérations N nécessaires pour faire converger e valeurs propres pour différentes tailles de sous-espace de Krylov m et une précision $p = 10^{-8}$

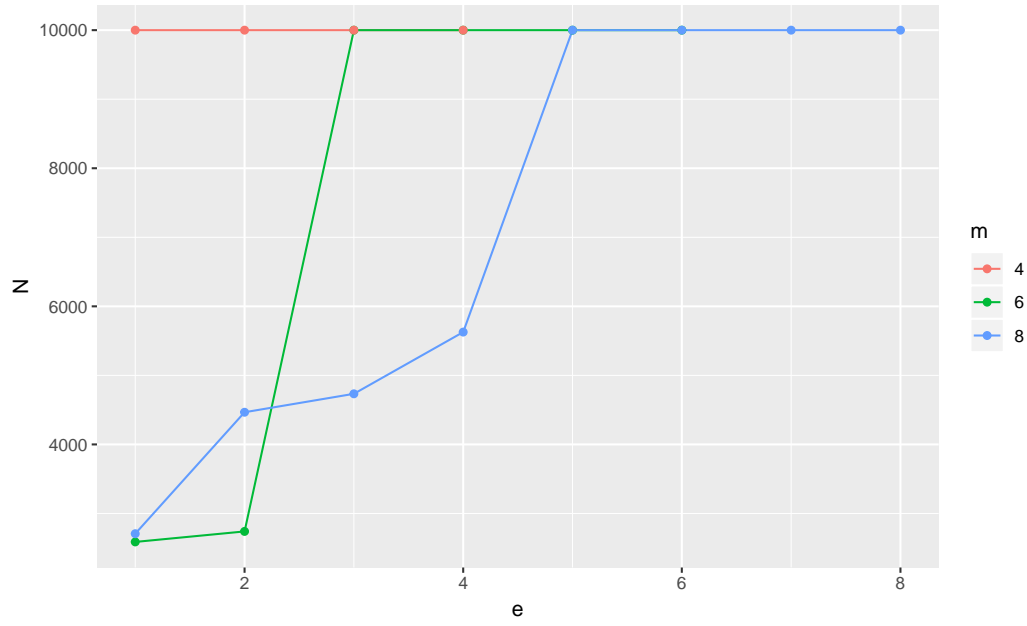


FIGURE 3 – Nombre d'itérations N nécessaires pour faire converger e valeurs propres pour différentes tailles de sous-espace de Krylov m et une précision $p = 10^{-10}$

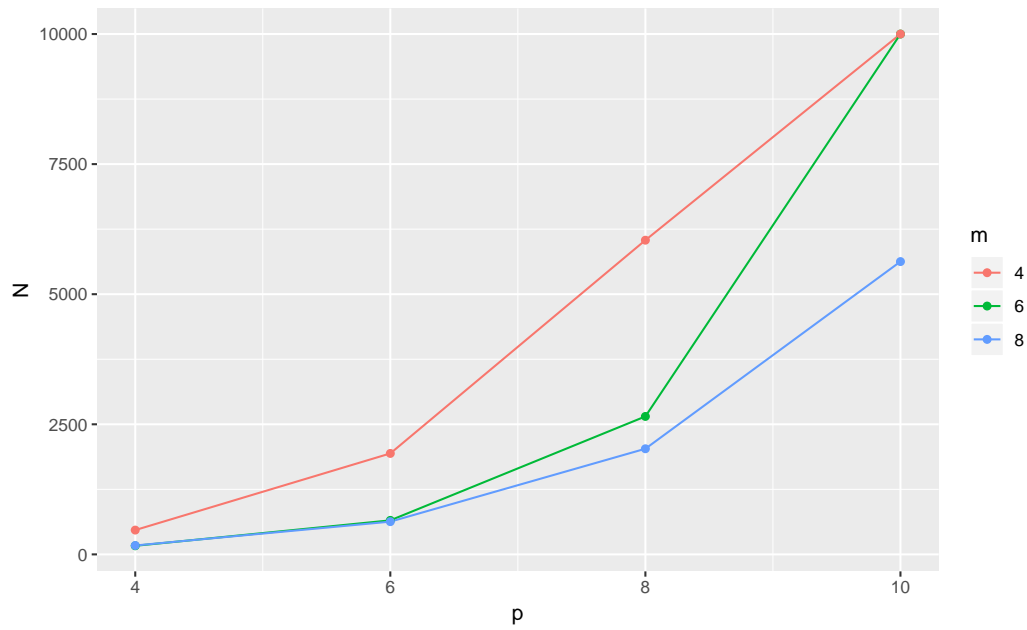


FIGURE 4 – Nombre d'itérations N nécessaires pour faire converger $e = 4$ valeurs propres pour différentes tailles de sous-espace de Krylov m et une précision p