

# Projet de MPNA : Méthode des itérations simultanées

Matthias BEAUPÈRE, Pierre GRANGER

Rapport MPNA - CHPS - 19 février 2019

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Problématique</b>	<b>1</b>
<b>3</b>	<b>Approche utilisée</b>	<b>1</b>
<b>4</b>	<b>Cas séquentiel</b>	<b>1</b>
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 . . . . .	2
4.2	Étude de performances théorique . . . . .	2
4.3	Étude de performances pratique . . . . .	2
4.4	Ajout d'une méthode de déflation "Locking" . . . . .	2
4.5	Conclusions . . . . .	3
<b>5</b>	<b>Cas parallèle</b>	<b>3</b>
5.1	Approche utilisée . . . . .	3
5.2	Étude de performances théorique . . . . .	3
5.3	Étude de performances pratique . . . . .	3
5.4	Conclusions . . . . .	3
<b>6</b>	<b>Conclusion générale</b>	<b>3</b>

## 1 Introduction

De nombreux problèmes modernes impliquent le calcul de valeurs propres et de vecteurs propres de grandes matrices creuses. Par exemple, ce problème peut être rencontré lors de l'étude de certains systèmes physiques ou bien par exemple dans les moteurs de recherche qui reposent sur ce type de technique. Des algorithmes spécifiques doivent être utilisés afin de résoudre ces problèmes de valeur propre en un temps raisonnable. De plus les algorithmes utilisés doivent pouvoir être facilement utilisés sur les architectures massivement parallèles actuelles ce qui représente un grand défi.

## 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  $\max(\text{precisions}) < p$  do
3:    $Z = AQ$ 
4:   Gram-Schmidt  $Q$ 
5:   Projection  $B = Z^t AZ$ 
6:   Décomposition de Schur  $B = Y^t RY$ 
7:   Retour dans l'espace d'origine  $Q = ZY$ 
8:   Sélection des  $k$  vecteur propres
9:   Calcul de la précision
10: 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 orthonormalisé la matrice  $Q$ . L'ortogonalisation 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

### 4.2 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 entraine 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 osciler 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.

### 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

On représente sur la figure 6 la précision au cours des itérations pour  $e$  et  $m$  fixés dans le cas de la méthode initiale et de la méthode avec locking. On peut tout d'abord observer que la convergence est bien plus rapide lorsque le locking est utilisé (environ 2000 contre 3800 itérations). On peut en outre constater sur la figure 6a que la convergence avec locking semble moins instable.

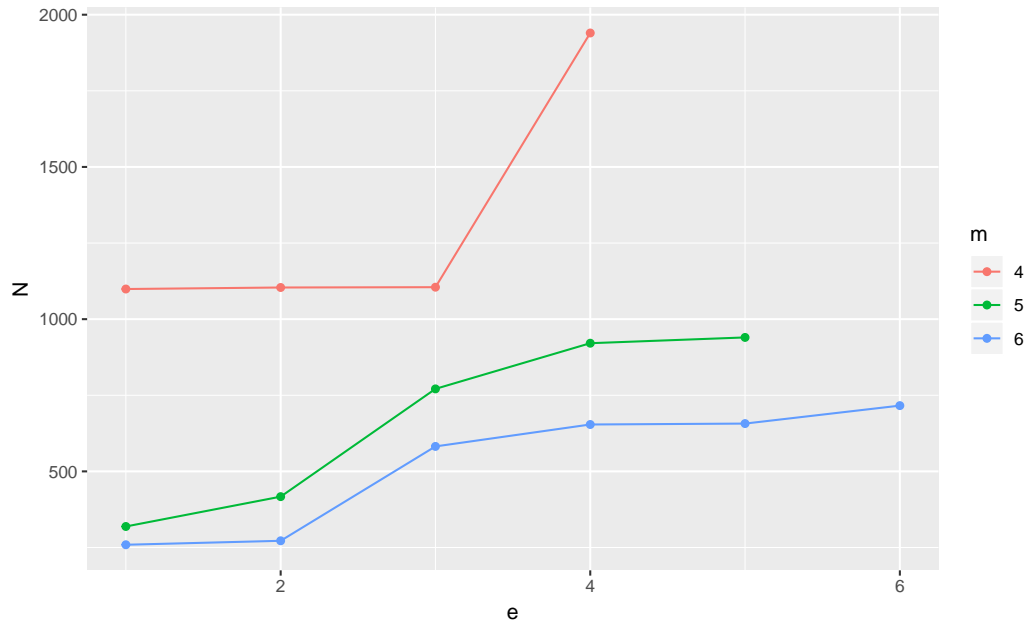


FIGURE 1 – Nombre d'itérations  $N$  nécessaires pour faire converger  $e$  valeurs propres pour différentes tailles de sous-espace de Krylov  $m$  et une précision  $p = 10^{-6}$

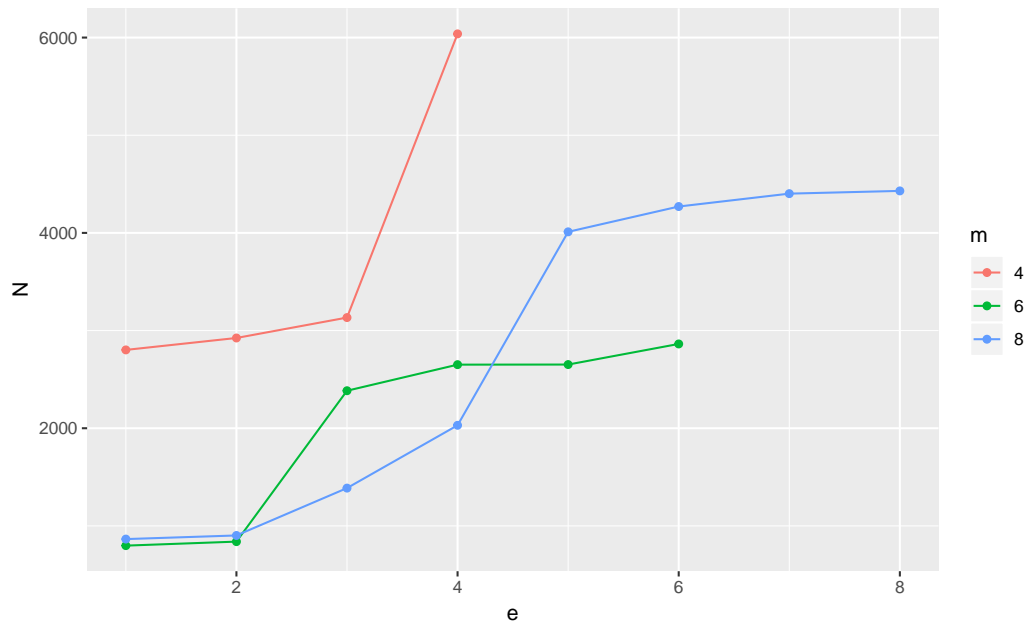


FIGURE 2 – Nombre d'itérations  $N$  nécessaires pour faire converger  $e$  valeurs propres pour différentes tailles de sous-espace de Krylov  $m$  et une précision  $p = 10^{-8}$

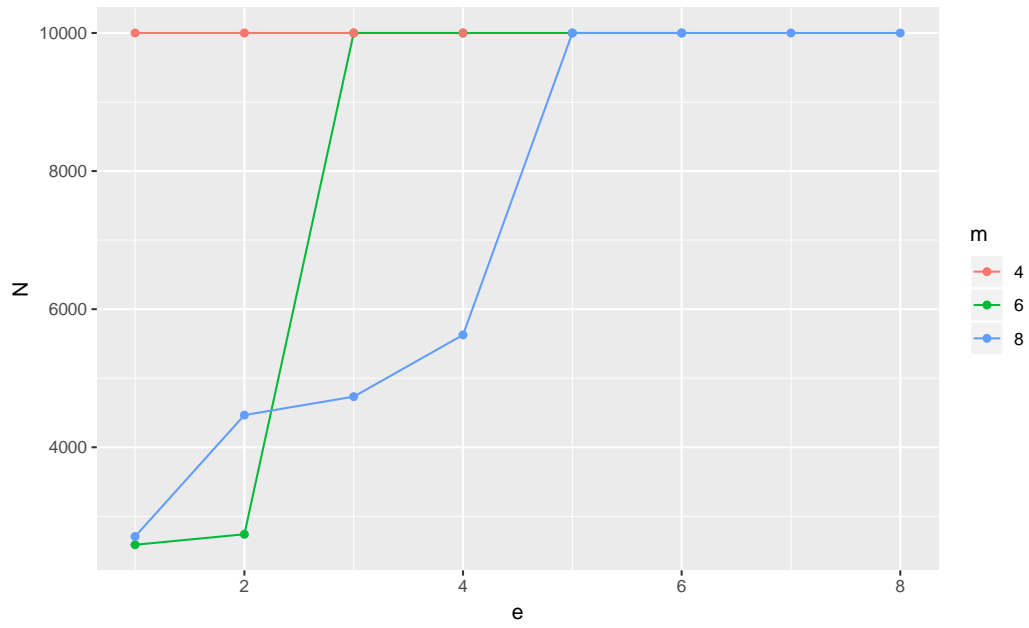


FIGURE 3 – Nombre d'itérations  $N$  nécessaires pour faire converger  $e$  valeurs propres pour différentes tailles de sous-espace de Krylov  $m$  et une précision  $p = 10^{-10}$

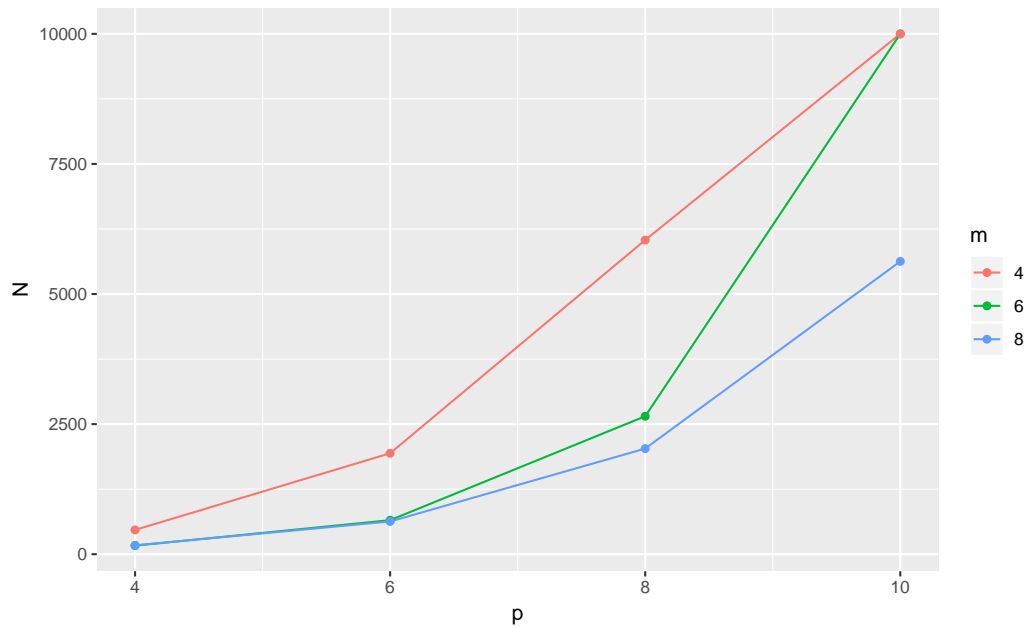


FIGURE 4 – Nombre d'itérations  $N$  nécessaires pour faire converger  $e = 4$  valeurs propres pour différentes tailles de sous-espace de Krylov  $m$  et une précision  $p$

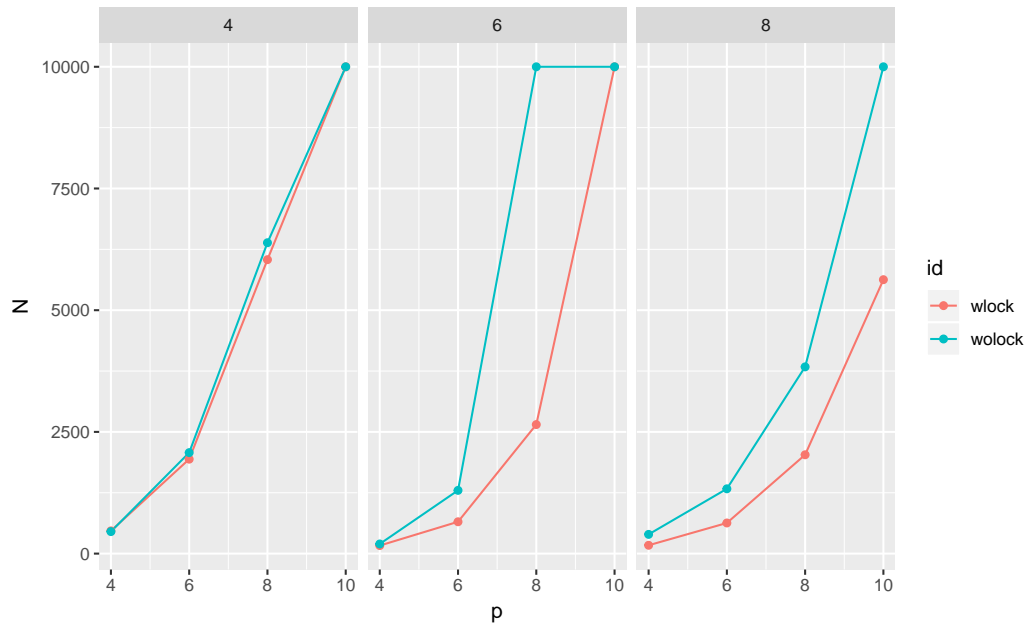
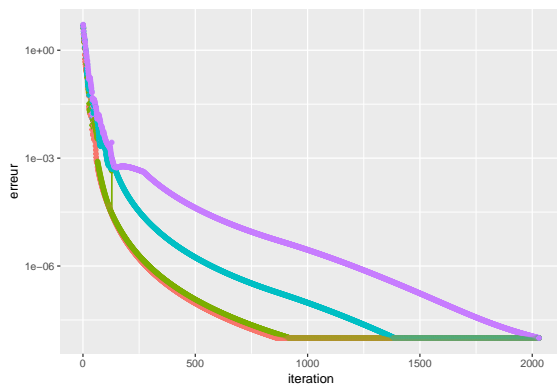
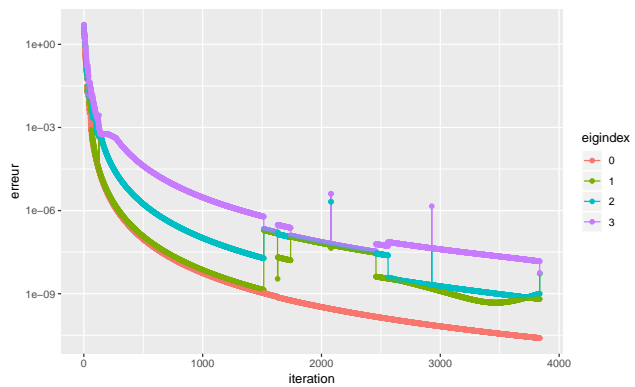


FIGURE 5 – Nombre d'itérations  $N$  nécessaires pour faire converger  $e = 4$  valeurs propres pour différentes tailles de sous-espace de Krylov  $m$  et une précision  $p$  avec et sans utilisation du locking



(a) Avec locking



(b) Sans locking

FIGURE 6 – Précision au cours des itérations  $N$  pour  $e = 4$  valeurs propres pour une taille de sous-espace de Krylov  $m = 8$