École Nationale Supérieure d'Électronique, Informatique, Télécommunications,
Mathématique et Mécanique de Bordeaux

Rapport du stage d'application

Estimation d'erreur et optimisation de méthodes numériques pour la théorie de la fonctionnelle de densité (DFT)

Réalisé par

BAMHAOUTE TAHA

Filière Matmeca, Spécialité Fluide et Énergétique

Tuteurs de stage :
Gaspard Kemlin
Jean Paul Chehab

Tuteur école :
Kevin Santugini Repiquet

Jury :
Kevin Santugini Repiquet
Antoine Lemoine

Année académique : 2023-2024

Table des matières

Résumé

Ce stage a été réalisé au sein du *Laboratoire Amiénois de Mathématique Fondamentale et Appliquée* (LAMFA), situé à l'Université de Picardie Jules Verne. L'objectif principal du stage était l'estimation d'erreur et l'optimisation de méthodes utilisées en calcul de structure électronique pour la physique de la matière condensée. Les modèles dans ce domaine de recherche reposent sur l'équation de Schrödinger à N corps, une EDP linéaire de grande dimension ($3N$, où N est le nombre d'électrons du système). Sa résolution pratique n'est possible que pour de petites valeurs de N à cause du fléau de la dimension. Différents modèles d'approximation ont donc été introduits, comme la théorie de la fonctionnelle de densité de Kohn–Sham (DFT), au prix d'une non-linéarisation du problème initial. De nombreuses méthodes ont été développées pour résoudre numériquement les équations sous-jacentes, qui prennent souvent la forme d'un problème aux valeurs propres non linéaire, résolu par exemple par des méthodes de type point fixe.

La première partie de ce stage a consisté à optimiser des méthodes numériques pour la résolution de l'équation de Schrödinger en utilisant le modèle de Gross-Pitaevskii (section 3), tandis que la deuxième partie s'est concentrée sur l'estimation de l'erreur en utilisant une discrétisation en ondes planes, basée sur une linéarisation des équations.

Outils

Pour répondre à la problématique de ce stage, nous avons utilisé les outils informatiques suivants :

Le langage Julia : Un langage de programmation de haut niveau, performant et dynamique, conçu pour le calcul scientifique et les applications de calcul numérique. Julia se distingue par sa capacité à combiner la facilité d'utilisation des langages dynamiques avec les performances des langages compilés, en particulier pour le calcul intensif.

DFTK (Density Functional Tool Kit) : Un ensemble d'outils développé en Julia pour les calculs basés sur la théorie de la fonctionnelle de la densité (DFT). Conçu pour simplifier le développement de nouvelles méthodes, DFTK permet d'explorer une large gamme de modèles, allant des systèmes simples en 1D à des systèmes physiques complexes comportant jusqu'à 1 000 électrons. Cette flexibilité fait de DFTK un outil puissant à l'intersection de l'analyse numérique, du calcul haute performance et des simulations de matériaux. Il offre des fonctions prédéfinies pour le calcul de la densité électronique et de l'énergie des atomes, et prend en charge des simulations pour des systèmes périodiques.

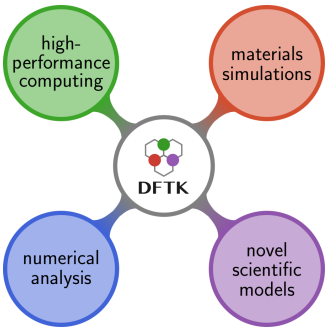


FIGURE 1 – DFTK Toolkit

Cluster informatique : Grâce à la plateforme MatriCS, j'ai pu exécuter des calculs lourds à distance, bénéficiant ainsi de ressources de calcul importantes et d'une grande capacité de mémoire, essentielles pour traiter les simulations complexes et les calculs intensifs.

1 Introduction

L'équation de Schrödinger, formulée par Erwin Schrödinger en 1925, est une pierre angulaire de la mécanique quantique, décrivant l'évolution temporelle des systèmes quantiques tels que les atomes et les molécules. Cette équation nous permet de déterminer la fonction d'onde d'un système, qui contient toutes les informations sur l'état quantique des électrons. On peut ensuite déterminer de nombreuses propriétés physiques du système considéré. L'énergie, dans ce contexte, joue un rôle crucial : elle apparaît sous forme de valeurs propres quantifiées, c'est-à-dire que l'énergie d'un système ne peut prendre que des valeurs discrètes spécifiques. L'équation de Schrödinger dépendante du temps, décrivant le comportement d'un électron soumis à un potentiel externe V, s'écrit :

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\Delta\psi(x,t) + V(x)\psi(x,t) \tag{1}$$

où :

- $\psi(x,t)$ est la fonction d'onde, qui dépend de la position x et du temps t.
- i est l'unité imaginaire telle que i^2=-1.
- $\hbar$ est la constante de Planck réduite.
- m est la masse de l'électron
- Δ est l'opérateur laplacien, qui donne la dérivée seconde par rapport à la position, représentant l'énergie cinétique de l'électron.
- $V(x)$ est le potentiel externe, qui dépend uniquement de la position x.

En utilisant les unités atomiques, où l'on suppose que la constante de Planck réduite $\hbar$ et la masse m valent 1, l'équation de Schrödinger se réduit à :

$$i\frac{\partial\psi(x,t)}{\partial t} = -\frac{1}{2}\Delta\psi(x,t) + V(x)\psi(x,t) =: (H_0\psi)(x,t) \tag{2}$$

L'équation de Schrödinger dépendante du temps peut être simplifiée dans le cas stationnaire, où la fonction d'onde $\psi(x,t)$ ne dépend explicitement du temps qu'à travers un facteur de phase. Dans ce cas, on peut séparer la fonction d'onde en un produit de la forme $\psi(x,t) = \phi(x)e^{-iEt}$, où E représente l'énergie de l'état quantique dans lequel de trouve l'électron. En substituant cette forme dans l'équation de Schrödinger dépendante du temps, on obtient l'équation de Schrödinger indépendante du temps :

$$H_0\phi(x) = E\phi(x) \tag{3}$$

où $H_0 = -\frac{1}{2}\Delta + V(x)$ est l'opérateur hamiltonien représentant l'énergie totale du système. Cette équation constitue un problème aux valeurs propres, où les valeurs propres E représentent les niveaux d'énergie possibles, et $|\phi(x)|^2$ représente la probabilité de localiser une particule dans une région donnée de l'espace. Résoudre cette équation, surtout pour des systèmes complexes, pose des défis numériques significatifs qui nécessitent des méthodes numériques avancées. Pour résoudre ce problème, nous verrons qu l'on peut utiliser une méthode de minimisation de l'énergie sur la variété de Grassmann. La variété de Grassmann $\mathrm{Gr}(k,n)$ est un espace qui paramètre les sous-espaces vectoriels de dimension k d'un espace vectoriel de dimension n. Dans le contexte des systèmes quantiques à 1 corps, cela revient à minimiser l'énergie $E[\phi] = \frac{\langle\phi|H_0|\phi\rangle}{\langle\phi|\phi\rangle}$. Cette approche est aussi particulièrement utile pour les systèmes à plusieurs corps, où la minimisation sur la variété de Grassmann permet d'obtenir les états d'énergie minimale via des méthodes numériques efficaces.

1.1 Mécanique quantique des électrons non-interagissants

Considérons un système de N_{el} électrons non-interagissants. Dans ce cadre, deux principes fondamentaux s'appliquent :

1. **Le principe d'exclusion de Pauli** : Ce principe stipule que deux électrons ne peuvent pas se trouver dans le même état quantique. Cela signifie que pour des systèmes sans spin, chaque état quantique est occupé par au maximum un électron.
2. **Le principe Aufbau** (ou principe de construction en allemand) : Les électrons occupent les états d'énergie les plus bas disponibles, remplissant progressivement les niveaux d'énergie de plus en plus élevés jusqu'à ce que tous les N_{el} électrons soient placés.

L'équation de Schrödinger pour N_{el} électrons non-interagissants dans ce système est donnée par :

$$H_0\varphi_n = \epsilon_n\varphi_n \tag{4}$$

où :

- $H_0 = -\frac{1}{2}\Delta + V$ est l'hamiltonien de base du système, comprenant le terme cinétique -$\frac{1}{2}\Delta$ (opérateur laplacien) et le potentiel V.

— ϵ_n est l'énergie associée à l'état φ_n.
— Les fonctions d'onde φ_n sont orthonormales, c'est-à-dire que $\langle \varphi_n, \varphi_m \rangle_{L^2(\mathbb{R}^3)} = \delta_{nm}$, où δ_{nm} est le symbole de Kronecker.

Les niveaux d'énergie sont ordonnés de manière croissante :(principe Aufbau)

$$\epsilon_1 \leq \epsilon_2 \leq \cdots \leq \epsilon_{N_{\text{el}}}$$

L'énergie de l'état fondamental du système, c'est-à-dire l'énergie la plus basse possible lorsque tous les électrons sont dans les états d'énergie les plus bas, est donnée par la somme des N_{el} plus basses énergies :

$$E = \sum_{n=1}^{N_{\text{el}}} \epsilon_n$$

La densité électronique de l'état fondamental, $\rho(x)$, est exprimée comme :

$$\rho(x) = \sum_{n=1}^{N_{\text{el}}} |\varphi_n(x)|^2$$

Cette densité est normalisée de sorte que l'intégrale sur tout l'espace soit égale au nombre total d'électrons :

$$\int_{\mathbb{R}^3} \rho(x)\, dx = N_{\text{el}}$$

Pour résoudre numériquement ce problème, on commence par choisir une base orthonormale discrétisée de taille N_b. Les orbitales discrètes $(\varphi_n) \in \mathbb{R}^{N_b \times N_{\text{el}}}$ ne sont pas uniques en raison des dégénérescences possibles. (plusieurs états d'énergie identique, lorsque la valeur propre est dégenerée). Par conséquent, il est plus aisé de travailler avec le projecteur orthogonal P^* sur l'espace engendré par la famille orthonormale $(\varphi_n)_{1 \leq n \leq N_{\text{el}}}$:

$$P^* = \sum_{n=1}^{N_{\text{el}}} |\varphi_n\rangle\langle\varphi_n| = \sum_{n=1}^{N_{\text{el}}} \varphi_n \varphi_n^T \in \mathbb{R}^{N_b \times N_b}.$$

P^* est un projecteur orthogonal de rang N_{el} (matrice densité de l'état fondamental). L'énergie de l'état fondamental peut alors être calculée comme :

$$E = \sum_{n=1}^{N_{\text{el}}} \epsilon_n = \sum_{n=1}^{N_{\text{el}}} \langle \varphi_n | H_0 \varphi_n \rangle = \text{Tr}(H_0 P^*) \tag{5}$$

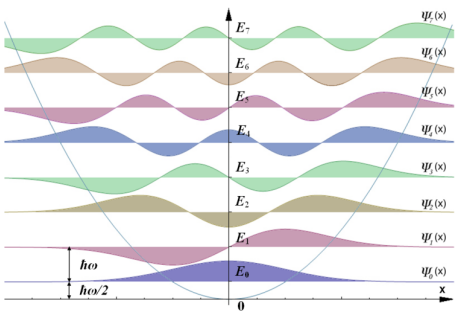
Il est facile de montrer que le projecteur P^* minimise la trace $\text{Tr}(H_0 P)$ sur l'ensemble des projecteurs orthogonaux de rang N_{el} (cf. annexe). Ce problème de minimisation peut être reformulé de manière équivalente comme suit :

$$P^* = \min_{P \in \mathcal{M}_{N_{\text{el}}}} \text{Tr}(H_0 P)$$

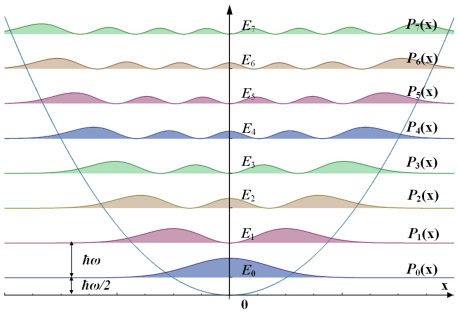
où $\mathcal{M}_{N_{\text{el}}}$ est l'ensemble des projecteurs orthogonaux de rang N_{el}, défini par :

$$\mathcal{M}_{N_{\text{el}}} := \left\{ P \in \mathbb{R}^{N_b \times N_b} \mid P = P^T,\ \text{Tr}(P) = N_{\text{el}},\ P^2 = P \right\}$$

Cet ensemble est difféomorphe à la variété de Grassmann $\text{Grass}(N_{\text{el}}, N_b)$, qui paramètre les sous-espaces vectoriels de dimension N_{el} dans un espace de dimension N_b.



((a)) Représentation des sept premiers niveaux d'énergie et des fonctions d'onde associées



((b)) Représentation des sept premiers niveaux d'énergie et des densités de probabilité de présence associées

FIGURE 2 – Illustrations des niveaux d'énergie et des fonctions d'onde dans le cadre de la théorie quantique [4]

La figure 2(a) montre les sept premiers niveaux d'énergie et les fonctions d'onde associées d'un oscillateur harmonique quantique unidimensionnel ($V(x) = -\frac{1}{2}x^2$). La figure 2(b) montre la même chose mais cette fois-ci, ce sont les allures des densités de probabilité de présence $|\varphi(x)|^2$ qui sont représentées.

1.2 Mécanique quantique des électrons interagissants

Dans la réalité, les électrons interagissent entre eux, ce qui modifie la forme générale de l'énergie en ajoutant un terme non linéaire, l'expression de l'énergie totale devient alors :

$$E(P) := \mathrm{Tr}(H_0 P) + E_{\mathrm{nl}}(P) \tag{6}$$

Le choix de $E_{\mathrm{nl}}(P)$ modélise l'interaction entre les électrons et dépend du modèle choisi pour approximer l'équation de Schrödinger à N corps (comme la DFT de Kohn-Sham ou le modèle de Hartree-Fock). Au niveau continu, on obtient les équations de Kohn-Sham en écrivant les conditions d'Euler-Lagrange associées au problème (6). L'hamiltonien se décompose en termes linéaires et non linéaires :

$$\begin{cases} \left(-\frac{1}{2}\Delta + V_{\mathrm{nuc}}\right)\varphi_n + V_{\mathrm{Hxc}}(\rho)\varphi_n = \epsilon_n \varphi_n, \\ \langle \varphi_n, \varphi_m \rangle_{L^2(\mathbb{R}^3)} = \delta_{nm}, \\ \rho = \sum_{n=1}^{N_{\mathrm{el}}} |\varphi_n|^2 \end{cases} \tag{7}$$

où $V_{\mathrm{Hxc}}(\rho)$ est le potentiel de Hartree-échange-correlation, qui dépend de la densité électronique ρ. Au niveau discret, le problème de minimisation associé est alors :

$$\min_{P \in \mathcal{M}_{N_{\mathrm{el}}}} E(P) = \mathrm{Tr}(H_0 P) + E_{\mathrm{nl}}(P) \tag{8}$$

2 Méthodes numériques

Notre objectif est de minimiser la fonction d'énergie $E(P) = \mathrm{Tr}(H_0 P) + E_{nl}(P)$ sous la contrainte que la matrice P appartienne à l'ensemble $\mathcal{M}_{N_{el}}(\mathbb{R})$ des projecteurs orthogonaux symétriques de rang N_{el}. $\mathcal{M}_{N_{el}}(\mathbb{R})$ est une variété riemannienne régulière : l'espace tangent $T_P\mathcal{M}_{N_{el}}$ en un point $P \in \mathcal{M}_{N_{el}}$ est défini par :

$$T_P\mathcal{M}_{N_{el}} = \{X \in H \mid PX + XP = X,\ \mathrm{Tr}(X) = 0\} = \{X \in H \mid PXP = 0 \text{ et } (1-P)X(1-P) = 0\}. \tag{9}$$

Pour toute matrice X, l'opérateur de projection sur $T_P\mathcal{M}_{N_{el}}$ est défini par :

$$\Pi_P(X) = PX(I-P) + (I-P)XP. \tag{10}$$

La condition d'optimalité du premier ordre stipule que la projection du gradient de l'énergie $H^* := \nabla E(P_*)$ appliqué à la solution du problème P_*, sur l'ensemble $\mathcal{M}_{N_{el}}(\mathbb{R})$ soit nulle, c'est-à-dire que $\Pi_{P_*}(H_*) = 0$. Cela implique que $P_* H_* (1-P_*) = 0$ et $(1-P_*)H_* P_* = 0$, caractérisant ainsi P_* comme un point critique de l'énergie sur la variété.

Pour résoudre ce problème de minimisation, nous commencerons par deux algorithmes de base : l'algorithme de point fixe (SCF) et l'algorithme de descente de gradient. Ensuite, nous améliorerons ces deux algorithmes en introduisant des paramètres de relaxation pour accélérer la convergence. Afin de garantir que notre solution reste dans l'ensemble $\mathcal{M}_{N_{el}}(\mathbb{R})$ à chaque itération, nous utiliserons une fonction de rétraction R, définie pour une matrice symétrique P_e proche de $\mathcal{M}_{N_{el}}(\mathbb{R})$. Cette rétraction s'appuie sur la décomposition en valeurs propres $P_e = VD_eV^*$, où $D_{ii} = 1$ si $D_{e,ii} > 0.5$, et 0 sinon. Ainsi, la rétraction $R(P_e)$ est donnée par $R(P_e) = VDV^*$.

2.1 Point fixe

L'algorithme SCF (Self-Consistent Field), également connu sous le nom de méthode du point fixe, est une technique fondamentale en chimie quantique et en physique des matériaux pour résoudre les équations de Schrödinger dans les systèmes électroniques complexes. Ce processus itératif commence par une estimation initiale P_0 de la matrice de densité, puis met à jour cette matrice à chaque étape k selon la relation suivante :

$$\begin{cases} H(P_k)\phi_i^k = \epsilon_i^k \phi_i^k, \quad \epsilon_1^k \le \epsilon_2^k \le \cdots \le \epsilon_{N_{el}}^k, \\ \langle \phi_i^k, \phi_j^k \rangle = \delta_{ij}, \\ P_{k+1} := R\left(P_k + \beta \Pi_{P_k}\left(\Phi\left(P_k\right) - P_k\right)\right), \end{cases} \tag{11}$$

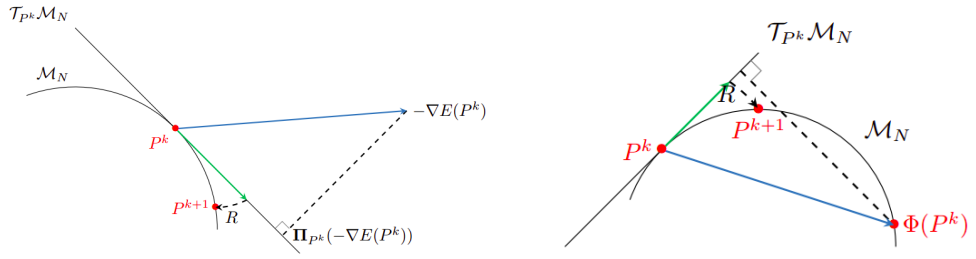
où R est la fonction de rétraction, β est un paramètre de relaxation, et $\Phi(P_k) := \sum_{i=1}^{N_{el}} \phi_i^k \left(\phi_i^k\right)^*$, avec ϕ_i^k représentant les vecteurs propres orthonormés associés aux N_{el} plus petites valeurs propres du gradient de l'énergie $H(P_k)$. Le processus continue jusqu'à convergence, c'est-à-dire jusqu'à ce que la différence entre P_k et P_{k+1} soit suffisamment petite, indiquant que le champ est auto-cohérent.

2.2 La descente de gradient

La descente de gradient est une approche itérative qui exploite le gradient de l'énergie pour minimiser une fonction en mettant à jour les matrices densités P_k à chaque itération selon :

$$P_{k+1} = R(P_k - \beta \Pi_{P_k}(\nabla E(P_k))),$$

où β est un pas fixe de relaxation, Π_{P_k} est l'opérateur de projection sur l'espace tangent, et R est la fonction de rétraction



((a)) Algorithme de la descente de gradient ((b)) Algorithme du point fixe

FIGURE 3 – Comparaison des algorithmes de descente de gradient et de point fixe [1]

2.3 Convergence du deux algorithmes

Les taux de convergence des algorithmes de descente de gradient et de SCF dépendent du rayon spectral d'opérateurs agissant sur l'espace $\mathbb{R}^{Nb\times Nb}$. Pour ces algorithmes, ces opérateurs sont de la forme $1-\beta J$, où β est le pas de relaxation fixe, et J la jacobienne (cf. **Théorème 1.1** et **Théorème 1.2** en annexe).
Pour la descente de gradient, l'opérateur est définit par $J = \Omega_* + K_*$, et pour l'algorithme SCF, $J = 1 + \Omega_*^{-1} K_*$, où Ω_* et K_* sont des opérateurs définis comme suit. L'opérateur $\Omega_* : T_{P_*}\mathcal{M}_N \to T_{P_*}\mathcal{M}_N$ est défini par :

$$\Omega_* X = P_* X (1 - P_*) H_* - H_* P_* X (1 - P_*) + \text{sym}$$

où "sym" représente la transposée de l'expression précédente. L'opérateur $K(P) = \Pi_P \nabla^2 E(P) \Pi_P$ est la Hessienne projetée sur l'espace tangent en P, et $K_* = K(P_*)$ où P_* est la solution de notre problème de minimisation. La condition du second ordre pour une fonction d'énergie $E(P)$, où P est une matrice de densité dans l'espace $\mathcal{M}_{N_{\text{el}}}$, s'exprime comme suit,sous des hypothéses raisonnables pour E :

$$\forall X \in \mathcal{T}_{P_*}\mathcal{M}_{N_{\text{el}}}, \quad \langle X, (\Omega_* + K_*)\, X \rangle_{\text{F}} \geqslant \eta \|X\|_{\text{F}}^2,$$

où $K_* = \Pi_{P_*} \nabla^2 E(P_*) \Pi_{P_*}$ est l'Hessienne de l'énergie projetée sur l'espace tangent $\mathcal{T}_{P_*}\mathcal{M}_{N_{\text{el}}}$ au point critique P_*. Ici, Π_{P_*} est l'opérateur de projection sur cet espace tangent, et η est une constante positive. Cette condition garantit que la fonction d'énergie est localement convexe autour de P_* dans l'espace tangent, ce qui assure que P_* est un minimum local et que les variations dans la direction de l'espace tangent ne conduisent pas à des diminutions d'énergie imprévues, assurant ainsi la stabilité et la convergence ds algorithmes.

3 L'équation de Gross–Pitaevskii

Dans cette section, nous abordons l'équation de Gross–Pitaevskii, un modèle non linéaire couramment utilisé en physique de la matière condensée. Au niveau continu, la fonction d'énergie est donnée par :

$$\mathcal{E}_\alpha(\gamma) = \text{Tr}_{L^2_{\text{per}}}\left(-\frac{1}{2}\Delta\gamma\right) + \int_0^1 V\rho_\gamma\, dx + \frac{\alpha}{2}\int_0^1 \rho_\gamma^2\, dx,$$

où ρ_γ représente la densité associée à l'opérateur γ, et V est le potentiel externe.
Pour obtenir une approximation numérique, nous discrétisons cette équation en utilisant la méthode des différences finies sur une grille uniforme de pas $\delta = \frac{1}{N_b}$. Cela conduit à un modèle de dimension finie :

$$E_\alpha(P) = \text{Tr}(H_0 P) + \frac{\alpha}{2}\delta \sum_{i=1}^{N_b}\left(\frac{P_{ii}}{\delta}\right)^2.$$

La matrice $H_0 = -\frac{1}{2}\Delta + V \in \mathbb{R}^{N_b\times N_b}$ est une matrice tridiagonale qui incorpore les conditions aux limites périodiques :

$$\forall 1 \leq i \leq N_b, \quad h_{ii} = \frac{1}{\delta^2} + V(i\delta), \quad h_{i,i+1} = h_{i,i-1} = -\frac{1}{2\delta^2}.$$

Pour implémenter les conditions aux limites périodiques, nous identifions les sites 0 et N_b ainsi que $N_b + 1$ et 1. Avec cette discrétisation, la densité discrète est approximée par $\rho(i\delta) \approx \rho_i := \frac{P_{ii}}{\delta}$, de sorte que :

$$\int_0^1 \rho \, dx \approx \sum_{i=1}^{N_b} \delta \cdot \rho_i = 1.$$

Le potentiel $V(x)$ est défini par :

$$V(x) = -C \left(\exp\left(-c \cos^2(\pi(x - 0.20))\right) + 2 \exp\left(-c \cos^2(\pi(x + 0.25))\right) \right),$$

avec $c = 30$ et $C = 20$. Ce potentiel à double puits devrait localiser la densité ρ principalement dans les régions correspondantes aux puits. Pour résoudre ce problème de minimisation avec contraintes, nous utiliserons des méthodes itératives telles que la descente de gradient à pas fixe et la méthode du point fixe décrite dans la section précédente. Nous chercherons également à optimiser ces méthodes en employant un paramètre de relaxation variable afin d'accélérer la convergence.

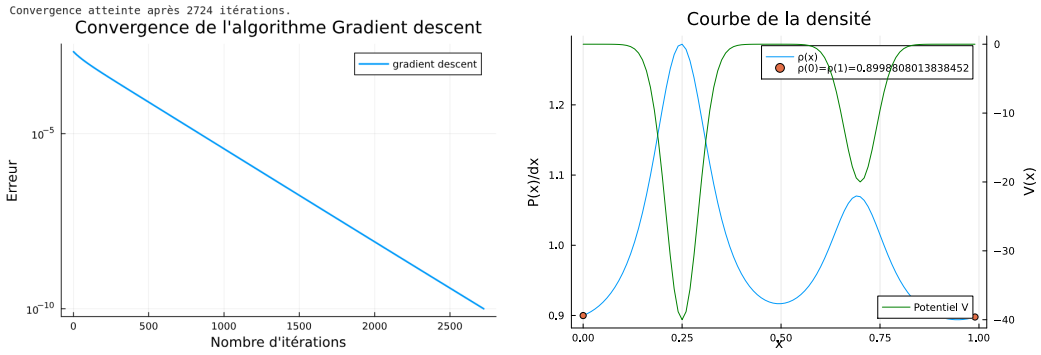
3.1 Descente de gradient à pas fixe

La descente de gradient définie dans la section 2.1 met à jour P_k de manière itérative avec :

$$P_{k+1} = R(P_k - \beta \Pi_{P_k}(\nabla E(P_k)),$$

où β est le pas de descente fixe et $\nabla E_\alpha(P_k)$ est le gradient de E_α évalué en P_k et R est la fonction de rétraction.

Pour bien converger vers la solution exacte il faut initialiser avec une matrice proche de la solution exacte. Pour cela on initialise avec P_0 qui est égal à $\phi_1 \phi_1^T$, où ϕ_1 est le vecteur propre de la matrice hermitienne h associé à sa plus petite valeur propre. On choisit un β très petit, de l'ordre de 10^{-5}.



((a)) Convergence du gradient descent ((b)) variation de la densité en fonction de x

FIGURE 4 – Convergence de l'algorithme de descente de gradient pour $\alpha = 50$ et $N_b = 100$

L'erreur dans la k-ème itération est définie par $\|P_{k+1} - P_k\|_F$, où $\| \ \|_F$ désigne la norme de Frobenius. L'algorithme s'arrête lorsque l'erreur atteint une tolérance de 10^{-10} après 2724 itérations pour un β de 5.10^{-5}. La figure 4 représente la densité définie par $\rho(x) = \frac{P(x)}{dx}$, où P désigne notre solution qui minimise la fonction d'énergie trouvée en utilisant la descente de gradient avec $P(x) = P(i \cdot dx) = P_{ii}$.

3.2 Descente de gradient à pas variable

On peut calculer un pas β variable pour accélérer la convergence de la descente de gradient en calculant $\mathrm{Tr}(H_0 P_{k+1}^2) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} (P_{k+1})_{i,i}^2$. Étant donné que P_{k+1} est une matrice densité, en utilisant l'expression de P_{k+1} en fonction de P_k et de β_k, on peut exprimer β_k sous forme quadratique et trouver par dérivation le β_k qui minimise la fonction d'énérgie. (cf. annexe)
On trouve :

$$\beta_k = \frac{\mathrm{Tr}(h P_k \Pi_{P_k}(\nabla E(P_k))) + \frac{\alpha}{2\delta} \sum_{i=1}^{\mathrm{Nb}} (P_k)_{i,i} (\Pi_{P_k}(\nabla E(P_k)))_{i,i}}{\mathrm{Tr}(h(\Pi_{P_k}(\nabla E(P_k)))^2) + \frac{\alpha}{2\delta} \sum_{i=1}^{\mathrm{Nb}} (\Pi_{P_k}(\nabla E(P_k)))_{i,i}^2}$$

Cette méthode converge rapidement par rapport à la descente de gradient à pas fixe.
On peut obtenir une meilleure convergence en multipliant β_k à chaque itération par un facteur aléatoire issu d'une loi uniforme sur l'intervalle [0, 2] (cf. figure 19).

La multiplication par un facteur de probabilité issu de la loi uniforme permet de réduire le nombre d'itérations de plus de 1000 à moins de 400. Cette approche fonctionne bien lorsqu'elle est appliquée à la direction de la descente et lorsque l'erreur décroît de manière linéaire.

3.3 Descente de gradient à pas variabe 2

On peut également chercher un paramètre β variable qui annule $\langle r^k, J(r^{k+1})\rangle$, où $J = \Omega + K$ désigne la jacobienne appliquée au résidu r^{k+1}, défini par $\Pi_{P_{k+1}}(\nabla E(P_{k+1}))$. En utilisant la méthode de gradient, on obtient la mise à jour suivante :

$$P_{k+1} = R\left(P_k - \beta \Pi_{P_k}(\nabla E(P_k))\right)$$

En appliquant une projection sur P_{k+1} avec une approximation de premier ordre, le β optimal se calcule sous la forme :

$$\beta_{\text{opt}} = \frac{\langle r^k, Jr^k\rangle}{\langle Jr^k, Jr^k\rangle}$$

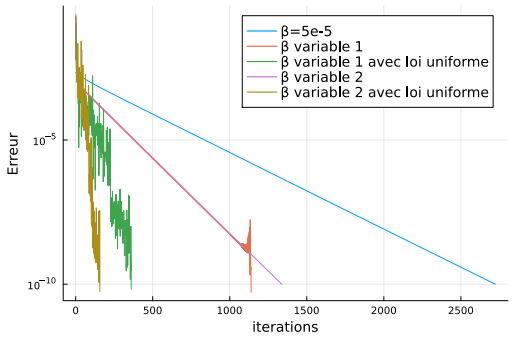
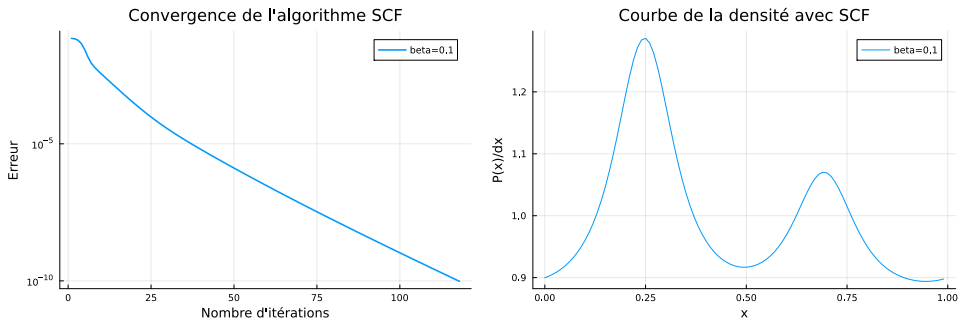


FIGURE 5 – Convergence de l'algorithme avec différentes β variables pour $\alpha = 50$ et $N_b = 100$

3.4 Algorithme du point fixe (SCF)

L'algorithme de point fixe défini dans la section 2.1 s'écrit comme suit :

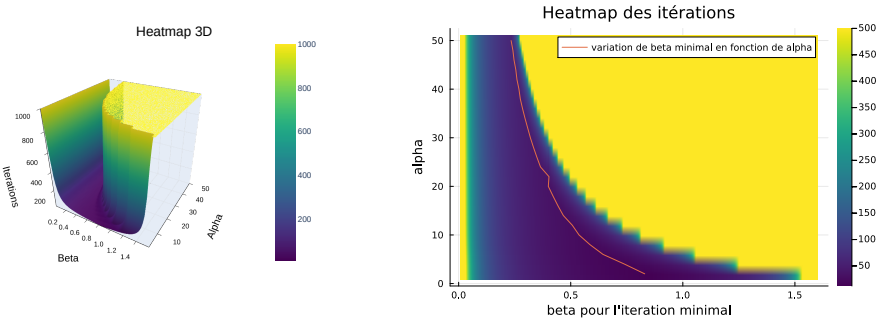
$$P_{k+1} := R\left(P_k + \beta \Pi_{P_k}\left(\Phi\left(P_k\right) - P_k\right)\right),$$



((a)) Convergence de SCF ((b)) variation de la densité

FIGURE 6 – Convergence de l'algorithme SCF pour $\alpha = 50$ et $N_b = 100$

Afin de choisir un β fixe optimal, nous lançons la simulation pour des valeurs de β comprises entre 0.01 et 1.6 et des valeurs de α de 2 à 50. Nous traçons ensuite une heatmap pour visualiser le nombre d'itérations nécessaires pour atteindre une tolérance de 10^{-12}. Nous fixons un nombre maximal d'itérations à 500. Pour réduire le temps de calcul on utilise la bibliothèque `Base.Threads` de Julia pour effectuer les calculs en parallèle du nombre d'itérations associé à chaque paire (α, β).



((a)) Heatmap 3D de la variation du nombre d'iteration en fonction de α et β

((b)) Heatmap 2D de la variation du nombre d'iteration en fonction de α et β

FIGURE 7 – Heatmap de l'algorithme point fixe

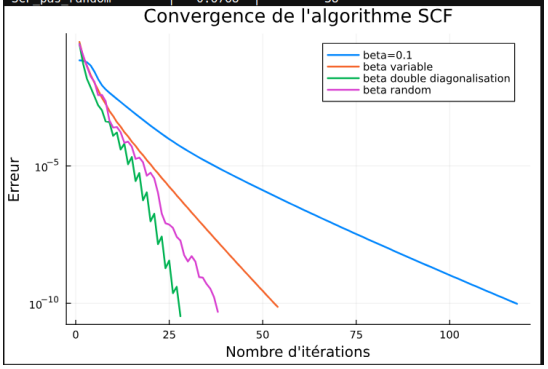
Pour un α de 50 et un β de 0.25, la méthode du point fixe converge en moins de 250 itérations, ce qui est considérablement rapide comparé à la méthode de descente de gradient. La figure 7 montre que le β optimal diminue lorsque α augmente. Il faut toutefois garder en tête qu'il est nécessaire de diagonaliser une matrice à chaque itération.

3.5 Point fixe avec relaxation

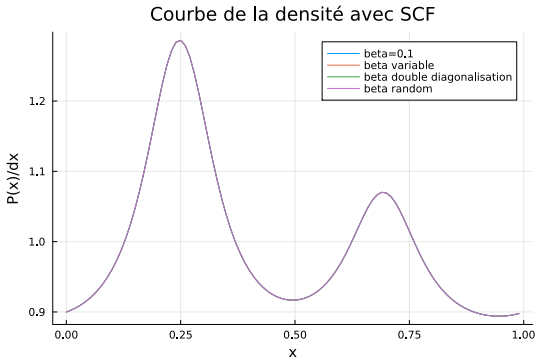
On peut calculer un pas variable pour la méthode du point fixe de la même manière que pour la descente de gradient, en évaluant la fonction d'énergie en P_{k+1}^2 et en injectant P_{k+1} par sa formule. En dérivant par rapport à β_k (cf. annexe 5.2), on obtient :

$$\beta_k = -\frac{\mathrm{Tr}\left(h\,\Pi_{P_k}\nabla(\phi(E(P_k))-P_k)P_k\right)+\left(\frac{\alpha}{2\delta}\right)\sum_{i=1}^{\mathrm{Nb}}P_k[i,i]\,\Pi_{P_k}\nabla(\phi(E(P_k))-P_k)[i,i]}{\mathrm{Tr}\left(h\,\left(\Pi_{P_k}\nabla(\phi(E(P_k))-P_k)\right)^2\right)+\left(\frac{\alpha}{2\delta}\right)\sum_{i=1}^{\mathrm{Nb}}\left(\Pi_{P_k}\nabla(\phi(E(P_k))-P_k)[i,i]\right)^2}$$

Method	Temps (s)	Iterations
SCF	1.1656	118
SCF_pas_variable	0.9649	54
SCF_pas_variable2	0.6827	28
SCF_pas_random	0.6768	38



((a)) Convergence des algorithmes de point fixe



((b)) Courbe de la densité

FIGURE 8 – Convergence des algorithmes de point fixe

Les algorithmes de points fixes convergent rapidement en termes de nombre d'itérations et de temps de calcul par rapport aux méthodes de descente de gradient. Pour le point fixe avec un paramètre de relaxation fixe, la méthode converge en moins de 120 itérations, ce qui est plus rapide que la la descente de gradient avec β variable. On remarque également que la multiplication du paramètre de relaxation variable par un facteur aléatoire uniforme entre $[0, 2]$ réduit le temps de calcul et le nombre d'itérations par rapport à la méthode avec uniquement le paramètre de relaxation variable, mais n'accélère pas la convergence de manière significative comme le fait la descente de gradient. De plus, la deuxième méthode pour calculer le paramètre de relaxation variable, en utilisant une double diagonalisation (cf. annexe 5.2), accélère la convergence, bien que la courbe de convergence n'évolue pas de manière linéaire et oscille légèrement. Le temps de calcul de cette méthode est proche de celui utilisant un facteur de probabilité, bien qu'elle présente un nombre d'itérations plus faible. Cela est dû au fait qu'à chaque itération, il faut diagonaliser deux fois. Pour des matrices de taille plus grande, cet algorithme pourrait être coûteux en termes de temps de calcul.

3.6 Calcul du beta fixe optimal

Le β optimal est défini par (cf. annexe 5.2) :

$$\beta_{\text{opt}} = \frac{2}{\lambda_{\max} + \lambda_{\min}}$$

où $\lambda_{\max}$ et $\lambda_{\min}$ désignent respectivement la valeur propre maximale et minimale de la jacobienne J de l'algorithme choisi.
Dans le cas général, les matrices associées à ces opérateurs peuvent atteindre des dimensions très grandes. Par exemple, pour notre problème, la matrice densité P a une dimension de 100×100, tandis que la matrice associée à la Jacobienne est de dimension 10000×10000. Le calcul direct des valeurs propres dans cet espace est souvent impraticable en raison du coût élevé des calculs.

Pour surmonter cette difficulté, nous utilisons la bibliothèque `Linearmap` en Julia, pour stocker non pas des matrices elles-mêmes, mais leur action sur des vecteurs, permettant ainsi de calculer leurs valeurs propres extrêmes à l'aide de méthodes itératives. Pour la Jacobienne dans la méthode SCF, nous employons la méthode itérative gradient conjugué pour calculer l'action de Ω^{-1}. Nous nous intéressons particulièrement aux valeurs propres de la Jacobienne définie sur l'espace tangent, car les valeurs propres dans l'orthogonal de cet espace sont toutes nulles.

Une première approche consistait à utiliser la bibliothèque `ARPACK` pour calculer la plus grande valeur propre en utilisant la méthode itérative de puissance et la plus petite valeur propre en utilisant la méthode de puissance inverse, puis faire une boucle pour trouver la première petite valeur propre non nulle. Cependant, cette méthode est coûteuse pour la Jacobienne de la méthode SCF car elle nécessite à chaque fois l'inversion de l'opérateur Ω^*.

Pour optimiser ce processus, nous pouvons d'abord calculer la base des vecteurs propres de l'opérateur Ω^*, défini par :

$$\forall a \in \{2, 3, \ldots, N_b\}, \quad \Omega^* \left(\phi_1 \phi_a^T + \phi_a \phi_1^T\right) = (\varepsilon_a - \varepsilon_1) \left(\phi_1 \phi_a^T + \phi_a \phi_1^T\right).$$

où ϕ_a sont les vecteurs propres associés à chaque valeur propre, triés de la plus petite à la plus grande, de $\nabla E(P^*)$, avec P^* étant la solution de notre problème de minimisation.
La projection de la Jacobienne sur cette base réduite permet d'obtenir une version simplifiée de la Jacobienne, de dimension 99×99, ce qui facilite considérablement le calcul des valeurs propres.

Après avoir obtenu la Jacobienne projetée, nous utilisons des méthodes itératives fournies par la bibliothèque `ARPACK`, pour calculer directement les plus grandes et les plus petites valeurs propres de la matrice. La projection sur une base réduite diminue significativement la dimension du problème, rendant le calcul des valeurs propres plus rapide et plus efficace.

Cette approche permet de déterminer le paramètre de relaxation β optimal avec une précision accrue, tout en optimisant le temps de calcul.

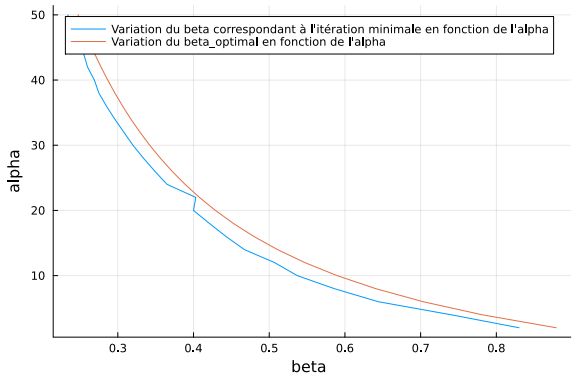


FIGURE 9 – Comparaison du beta optimal

La figure 9 montre la variation du paramètre β correspondant aux itérations minimales calculées en utilisant le heatmap 7(b) ainsi que la variation du β optimal calculé à partir des valeurs propres du jacobien. On remarque que les deux courbes ont des profils similaires et sont presque superposées, avec une erreur moyenne de 0,01. Cette petite différence est due à la liste des β prises dans le heatmap. Cet écart indique que notre approche pour calculer le β optimal est bien cohérente et correspond au nombre minimal d'itérations.

3.7 Comparaison entre différents algorithmes

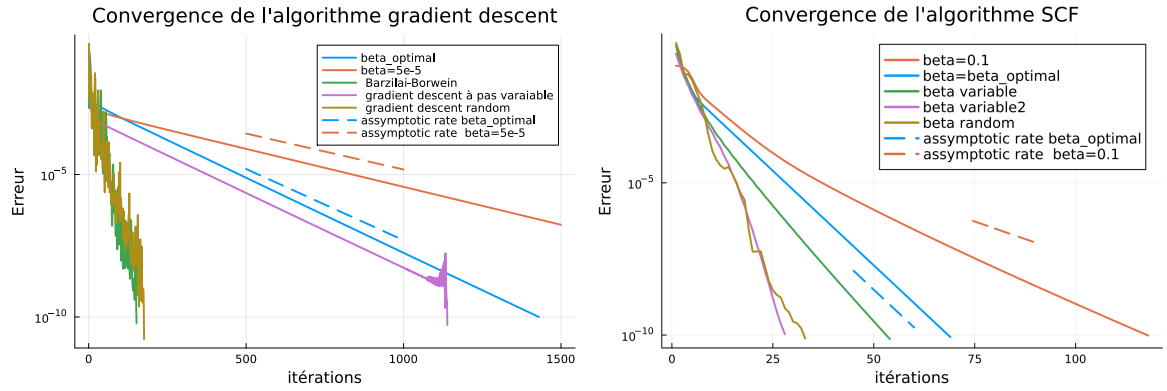


FIGURE 10 – Comparaison de la convergence des algorithmes de gradient descent et de point fixe pour $N_b = 100$ et $\alpha = 50$.

La méthode de descente de gradient avec un pas variable, inspirée de la méthode de Barzilai-Borwein (voir annexe), converge rapidement et atteint une tolérance de 10^{-10} en moins de 200 itérations. En parallèle, la méthode probabiliste, qui multiplie le β précédent par une loi normale entre $[0, 2]$, converge avec des performances distinctes. Le gradient à pas variable converge plus rapidement que celui à pas fixe, mais il demeure moins efficace que la méthode de Barzilai-Borwein. Le β fixe optimal, obtenu en calculant les valeurs propres de la jacobienne, permet de converger en 1430 itérations, comparé à un β fixe de 5×10^{-5} qui nécessite plus de 2000 itérations pour atteindre la convergence. Le calcul des valeurs propres de la jacobienne permet également de déterminer le taux asymptotique défini par le rayon spectral de $1 - \beta J$, où J est la jacobienne. On peut l'exprimer comme

$$r = \max\left(|1 - \beta\lambda_{\max}|, |1 - \beta\lambda_{\min}|\right).$$

On obtient alors le taux de convergence suivant :

$$\exists C \in \mathbb{R} \text{ tel que } \|P_k - P_{k-1}\|_{\mathrm{F}} \leq Cr^k,$$

où P_k est la solution à l'itération k et $\|\cdot\|_{\mathrm{F}}$ désigne la norme de Frobenius. En traçant r^k en échelle logarithmique, on peut observer une droite parallèle à la courbe de convergence pour chaque valeur fixe de β.

Il est également observé que les algorithmes à point fixe tendent à être plus efficaces que ceux basés sur la descente de gradient, car ils permettent généralement de converger en moins d'itérations, chaque itération étant cependant plus coûteuse.

4 Convergence avec un facteur de relaxation aléatoire

Dans cette section, nous cherchons à comprendre pourquoi multiplier le paramètre de relaxation par un facteur aléatoire issu d'une loi uniforme accélère la convergence.

Pour mieux appréhender l'impact de la loi uniforme sur la convergence, nous étudions un cas linéaire simple de minimisation de la fonction de coût quadratique :

$$\frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$

Comme le gradient de la fonction de coût est $\nabla f(x) = Ax - b$, avec A une matrice symétrique définie positive, minimiser cette fonction revient à résoudre le système linéaire $Ax = b$. Nous utilisons l'algorithme de descente de gradient pour atteindre cette solution et cherchons à calculer un pas de relaxation optimal qui annule $\langle Ar^k, r^{k+1}\rangle$, où le résidu r^k est donné par :

$$r^k = b - Ax^k.$$

On trouve alors un β_k donné par :

$$\beta_k = \frac{\langle r^k, Ar^k\rangle}{\langle Ar^k, Ar^k\rangle}.$$

Nous travaillons avec A, la matrice du laplacien, et comparons la convergence de la méthode avec ce pas variable en le multipliant par un facteur tiré d'une loi uniforme.

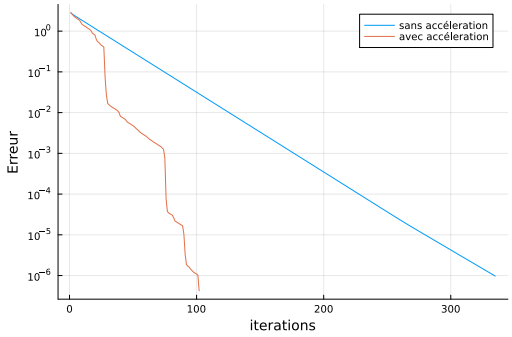
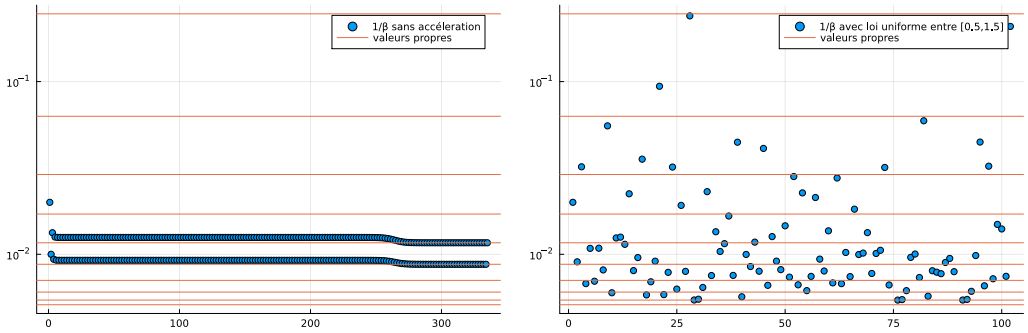


FIGURE 11 – Comparaison entre la convergence sans et avec la loi uniforme entre [0.5,1.5]



((a)) Distribution des β sans loi uniforme ((b)) Distribution des β avec loi uniforme

FIGURE 12 – Comparaison entre la distribution du facteur de relaxation sans et avec la loi uniforme

La multiplication par une loi uniforme permet d'accélérer la convergence. En effet, sur la figure 12(a), nous remarquons que les facteurs $\frac{1}{\beta_k}$ oscillent entre deux valeurs. Cependant, après avoir multiplié par une loi uniforme, la distribution des facteurs s'approche des inverses des valeurs propres. Cette approche accélère la convergence, en effet la mise à jour du résidu dans la méthode de gradient descent est donnée par :

$$r_{k+1} = (I - \beta_k A) r_k$$

nous exprimons r_k dans la base des vecteurs propres de A :

$$r_k = \sum_{i=1}^{n} (r_k^T v_i) v_i$$

où v_i sont les vecteurs propres associés aux valeurs propres λ_i de A. En substituant cette expression dans la mise à jour du résidu, nous obtenons :

$$r_{k+1} = \sum_{i=1}^{n} (r_k^T v_i)(1 - \beta_k \lambda_i) v_i$$

Cette expression montre comment chaque composante du résidu est ajustée en fonction des valeurs propres de A et du pas de mise à jour β_k. En choisissant β_k comme l'inverse des valeurs propres λ_i pour $i = 1, \ldots, n$, nous annulons le résidu dans la direction des vecteurs propres. Ainsi, à l'itération k=n, r_k=0.
Cela signifie que, pour une matrice de taille 10, la méthode va converger en 10 itérations si $\beta_k = \frac{1}{\lambda_k}$, comme le montre la figure 20. Toutefois, cela nécessite de connaître les valeurs propres de A. En multipliant le pas de mise à jour β par une loi uniforme, nous augmentons nos chances de nous approcher des inverses des valeurs propres, ce qui contribue à accélérer la convergence.
On utilise maintenant la descente de gradient à pas variable, où le paramètre de relaxation est multiplié par un facteur tiré d'une loi uniforme entre $[1 - \sigma, 1 + \sigma]$. Nous faisons varier σ entre [0, 1] et comparons le nombre moyen minimal d'itérations nécessaires pour atteindre la convergence avec celui obtenu par la méthode de descente de gradient à pas fixe. Cette comparaison est effectuée sur 1000 simulations pour chaque valeur de σ.

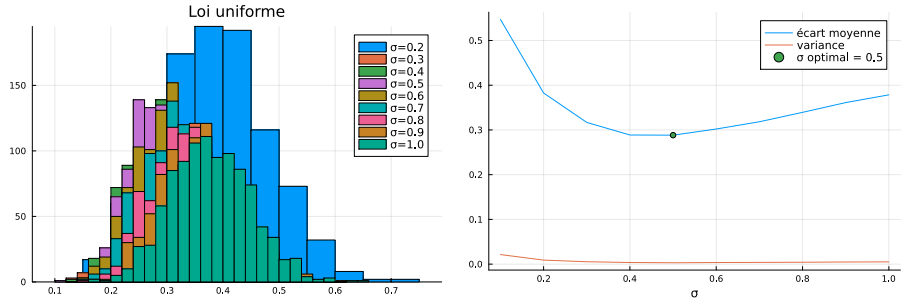


FIGURE 13 – Variation de l'écart en fonction de σ

Le σ optimal trouvé est autour de 0.4, c'est-à-dire que multiplier le paramètre de relaxation par un facteur tiré d'une loi uniforme entre [0.4, 1.4] permet d'obtenir en moyenne le nombre minimal d'itérations.
On observe un comportement similaire si on travaille avec l'équation de Gross-Pitaevskii (cf. figure 21 en annexe).

En effet, cela revient au fait que nous avons pris une matrice A correspondant à la matrice laplacienne utilisée pour construire le hamiltonien dans le cas de Gross-Pitaevskii. Le laplacien est perturbé par un potentiel V, ce qui change le σ optimal. Cependant, lorsque nous utilisons d'autres matrices mal conditionnées importées depuis Matrix Market, cela modifie significativement le σ optimal, car il dépend de la distribution des valeurs propres. Dans les deux cas, multiplier par un facteur aléatoire de moyenne 1 permet d'accélérer la convergence, même dans le cas où la matrice est mal conditionnée, où la méthode de gradient descent, même avec un pas optimal variable, converge lentement.
En remplaçant la loi aléatoire par une loi normale de moyenne 1 et de variance $\frac{\sigma}{\sqrt{3}}$, nous avons observé que le profil des résidus demeure similaire à celui obtenu avec la loi initiale. Cette invariance suggère que, malgré le changement de distribution, les caractéristiques fondamentales de la convergence restent intactes. De plus, nous continuons à obtenir le même σ optimal (cf. figure 22 en annexe). Cependant, il est important de noter que la variance de la loi normale et l'écart type influencent directement le comportement de la convergence. Une variance plus élevée peut entraîner une plus grande dispersion des valeurs générées, ce qui peut affecter la précision de l'approximation et la rapidité de la convergence.
On essaie maintenant de construire une fonction de répartition qui suit une distribution normale centrée sur l'inverse des valeurs propres, avec un écart-type égal à $\min(|\frac{1}{\lambda_i} - \frac{1}{\lambda_j}|)$. Ensuite, nous générons des β qui suivent cette loi et nous utilisons ces β pour calculer la solution de notre problème. Nous remarquons que cette méthode donne des résultats différents par rapport à la loi utilisée précédemment. Parfois, cela permet de converger très rapidement, comme le montre la figure 23 en annexe, et parfois l'inverse, comme illustré également par la figure 24 en annexe.

Cela peut s'expliquer par le fait que, pour assurer la convergence et la diminution du résidu après chaque itération, il faut que $\beta < \frac{2}{\lambda_{\max}}$. En revanche, en générant des β avec notre fonction, nous pouvons tomber plusieurs fois sur des points supérieurs à $\frac{2}{\lambda_{\max}}$, on remarque la même chose dans la figure 20 où nous prenons β qui vaut l'inverse des valeurs propres.

5 Estimations d'erreurs pour la DFT de Kohn-Sham en Ondes Planes

La résolution de problèmes de minimisation dans l'espace réel, notamment lorsqu'on travaille avec des matrices de grande taille, s'avère souvent coûteuse en termes de calcul. Dans le cas précédent, nous avons réussi à minimiser l'énergie en utilisant une discrétisation unidimensionnelle par différences finies avec des matrices de taille 100, ce qui a permis de déterminer la densité électronique pour un seul électron. Cependant, l'étude d'atomes plus complexes augmente considérablement la complexité computationnelle et le temps de calcul devient prohibitif.

Dans cette partie on va passer de l'espace réel à l'espace de Fourier, ce qui est justifié par la périodicité intrinsèque des systèmes cristallins que nous étudions. Un cristal parfait est décrit par une disposition spécifique d'atomes dans une maille élémentaire qui se répète périodiquement, formant ainsi un réseau de Bravais $\mathcal{R}$ avec son réseau réciproque $\mathcal{R}^*$. Cette périodicité nous permet d'utiliser une base de fonctions propres constituée des modes de Fourier (ondes planes) associées aux vecteurs du réseau réciproque.

Dans ce contexte, nous utilisons la méthode de discrétisation par ondes planes, qui est une approximation de Galerkin spécifique. Nous définissons un espace d'approximation de dimension finie $\mathcal{X}_{E_{\text{cut}}}$ comme :

$$\mathcal{X}_{E_{\text{cut}}} := \text{Span}\left\{ e_G, G \in \mathcal{R}^* \left| \frac{1}{2} \right| G \right|^2 \leqslant E_{\text{cut}} \right\}$$

où

$$\forall x \in \mathbb{R}^3, \quad e_G(x) := \frac{1}{\sqrt{|\Gamma|}} \exp(\mathrm{i}G \cdot x)$$

L'énergie de coupure E_{cut} détermine ainsi la taille de notre base d'orbitales : plus E_{cut} est élevé, plus notre solution approchée se rapproche de la solution exacte, mais le temps de calcul nécessaire augmente également.

Pour estimer l'erreur entre une solution calculée avec une énergie de coupure modérée E_{cut} et une solution de référence obtenue avec une énergie de coupure élevée $E_{\text{cut,ref}}$, nous effectuons une décomposition en basses et hautes fréquences.

$$\mathcal{X}_{E_{\text{cut,ref}}} = \mathcal{X}_{E_{\text{cut}}} \oplus \mathcal{X}^{\perp}_{E_{\text{cut}}}$$

où $\mathcal{X}_{E_{\text{cut}}} = \text{Span}\left(e_G, \frac{|G|^2}{2} \leqslant E_{\text{cut}}\right)$ et $\mathcal{X}^{\perp}_{E_{\text{cut}}} = \text{Span}\left(e_G, E_{\text{cut}} < \frac{|G|^2}{2} \leqslant E_{\text{cut,ref}}\right)$ Cette décomposition permet d'isoler l'effet des hautes fréquences, qui peuvent être négligées pour une énergie de coupure suffisante, sur la solution approchée.

Enfin, pour résoudre efficacement les équations de Kohn-Sham discrétisées et accélérer la convergence vers la solution auto-cohérente du champ (SCF), nous utilisons l'outil DFTK (*Density Functional Toolkit*). Cet outil permet de manipuler les orbitales électroniques et de calculer les densités en s'appuyant sur des fonctions intrinsèques, ainsi que sur des solveurs prédéfinis. DFTK est particulièrement adapté aux calculs de structure électronique des cristaux, exploitant la périodicité du système et permettant une manipulation efficace des orbitales dans l'espace de Fourier. En combinant ces approches, nous pouvons estimer l'erreur introduite par une énergie de coupure finie et optimiser le compromis entre précision et coût computationnel dans nos simulations.

5.1 Estimation d'erreur

Dans cette section on s'intéresse aux estimations pratiques des erreurs de discrétisation pour les approximations numériques des calculs de structure électronique. Pour ce faire, nous utilisons une approche générale basée sur une linéarisation des équations de Kohn-Sham. dans un contexte général. Supposons que nous souhaitons trouver $x \in \mathbb{R}^n$ tel que $f(x) = 0$, pour une fonction non linéaire $f : \mathbb{R}^n \to \mathbb{R}^n$ (le résidu). Près d'une solution x_*, nous avons $f(x) \approx f'(x)\,(x - x_*)$, et par conséquent, si $f'(x)$ est inversible, nous avons la relation erreur-résidu suivante :

$$x - x_* \approx f'(x)^{-1} f(x) \tag{12}$$

Il s'agit de la même approximation qui conduit à l'algorithme de Newton. Supposons maintenant que nous souhaitions calculer une quantité d'intérêt réelle $A\,(x_*)$, où $A : \mathbb{R}^n \to \mathbb{R}$ est une fonction C^1 (par exemple, l'énergie, une composante des forces interatomiques, ou la densité, etc.) ; nous avons alors l'égalité approximative avec un côté droit calculable :

$$A(x) - A\,(x_*) \approx \nabla A(x) \cdot \left(f'(x)^{-1} f(x)\right) \tag{13}$$

nous obtenons l'estimation simple suivante :

$$|A(x) - A\,(x_*)| \leqslant |\nabla A(x)| \left\|f'(x)^{-1}\right\|_{\text{op}} |f(x)| \tag{14}$$

où $|\cdot|$ est une norme choisie sur $\mathbb{R}^n$, et $\|\cdot\|_{\text{op}}$ est la norme opérateur induit sur $\mathbb{R}^{n\times n}$.
La structure de notre problème ne peut pas être facilement formulée comme ci-dessus en raison de la présence de contraintes et de dégénérescences. D'après la section 2-3 nous identifions l'analogue approprié du Jacobien $f'(x)$: l'opérateur $\boldsymbol{\Omega}_* + \boldsymbol{K}_*$ est le Jacobien de la carte résiduelle $R : P \mapsto \boldsymbol{\Pi}_P H(P)$ qui correspond à notre fonction f dans ce cas., qui s'annule pour $P = P_*$. Notre approche repose donc sur l'approximation du premier ordre suivante :

$$P - P_* \approx \left(\boldsymbol{\Omega}_* + \boldsymbol{K}_*\right)^{-1} R(P) \tag{15}$$

où P est notre solution dans $\mathcal{X}_{E_{\text{cut}}}$ et P_* est la solution sur $\mathcal{X}_{E_{\text{cut,ref}}}$.

Pour améliorer l'estimation des erreurs dans le contexte des méthodes numériques basées sur la séparation des fréquences, nous commençons par décomposer les vecteurs et opérateurs de l'espace tangent en deux parties distinctes, associées respectivement aux composantes basse fréquence et haute fréquence. Nous étiquetons ces parties comme $\boldsymbol{\Pi}_{E_{\text{cut}}} \mathcal{T}_P \mathcal{M}_{\mathcal{N}}$ et $\boldsymbol{\Pi}^{\perp}_{E_{\text{cut}}} \mathcal{T}_P \mathcal{M}_{\mathcal{N}}$, notées respectivement par 1 et 2 pour simplifier. Ainsi, la relation erreur-résidu peut être écrite de manière concise comme suit :

$$\begin{bmatrix} (\boldsymbol{\Omega} + \boldsymbol{K})_{11} & (\boldsymbol{\Omega} + \boldsymbol{K})_{12} \\ (\boldsymbol{\Omega} + \boldsymbol{K})_{21} & (\boldsymbol{\Omega} + \boldsymbol{K})_{22} \end{bmatrix} \begin{bmatrix} P_1 - P_{*1} \\ P_2 - P_{*2} \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

L'inversion de l'opérateur $(\boldsymbol{\Omega}(P) + \boldsymbol{K}(P))|_{\mathcal{T}_P \mathcal{M}_{\mathcal{N}}}$ dans tout l'espace est coûteuse, car elle inclut des valeurs non nulles sur les composantes associées aux différentes fréquences. Cependant, nous pouvons simplifier cette tâche en effectuant l'inversion uniquement sur la grille grossière $\mathcal{X}_{E_{\text{cut}}}$ et en approximant les composantes d'erreur de basse fréquence. Nous appliquons les approximations suivantes :

$$(\boldsymbol{\Omega}+\boldsymbol{K})_{21} \approx 0 \quad \text{et} \quad (\boldsymbol{\Omega}+\boldsymbol{K})_{22} \approx \boldsymbol{M}_{22}$$

L'opérateur M est défini sur le sous-espace orthogonal $\mathbf{\Pi}_{E_{\text{cut}}}^{\perp}\mathcal{T}_P\mathcal{M}_{\mathcal{N}}$ par la relation

$$M = P^{\perp}T^{1/2}P^{\perp}T^{1/2}P^{\perp},$$

où $P^{\perp}$ est l'opérateur de projection sur l'orthogonal de l'image de P et T est un opérateur coercif,diagonal en Fourier.
En particulier, si T est choisi comme une discrétisation de l'opérateur $1-\Delta$, on retrouve la norme de Sobolev classique H^1. M est positif défini, induisant ainsi une métrique sur cet espace. Cette formulation simplifie le calcul de $M^{1/2}$ et permet d'effectuer efficacement des calculs impliquant $M^{-1/2}$ à l'aide d'algorithmes itératifs. Cela transforme le système d'équations en :

$$\begin{bmatrix} (\boldsymbol{\Omega}+\boldsymbol{K})_{11} & (\boldsymbol{\Omega}+\boldsymbol{K})_{12} \\ 0 & \boldsymbol{M}_{22} \end{bmatrix} \begin{bmatrix} P_1 - P_{*1} \\ P_2 - P_{*2} \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

Nous pouvons alors résoudre pour $P_2 - P_{*2}$ comme suit :

$$P_2 - P_{*2} \approx \boldsymbol{M}_{22}^{-1} R_2$$

Puis, substituant ce résultat dans la première équation, nous obtenons :

$$P_1 - P_{*1} \approx (\boldsymbol{\Omega}+\boldsymbol{K})_{11}^{-1}\left(R_1 - (\boldsymbol{\Omega}+\boldsymbol{K})_{12}\boldsymbol{M}_{22}^{-1}R_2\right)$$

Ce calcul nécessite seulement un pas de Newton complet sur la grille grossière $\mathcal{X}_{E_{\text{cut}}}$, rendant l'approche plus économique. En tenant compte de la correction apportée par $(\boldsymbol{\Omega}+\boldsymbol{K})_{12}$, le résidu corrigé est donné par :

$$R_{\text{Schur}}(P) = \begin{bmatrix} (\boldsymbol{\Omega}+\boldsymbol{K})_{11}^{-1}\left(R_1 - (\boldsymbol{\Omega}+\boldsymbol{K})_{12}\boldsymbol{M}_{22}^{-1}R_2\right) \\ \boldsymbol{M}_{22}^{-1}R_2 \end{bmatrix}$$

5.2 Choix des paramètres

Le but est d'approcher la solution de référence P^* et de calculer l'erreur $P - P^*$ sans avoir à calculer directement la solution de référence P^*, qui nécessite un E_{cut} très grand et donc un temps de calcul important. Pour cela, on introduit une base variable de taille supérieure à celle de la petite base (correspondant à $\mathbf{\Pi}_{E_{\text{cut}}}\mathcal{T}_P\mathcal{M}_{\mathcal{N}}$ ou 1) et inférieure à la base de référence. En séparant les composantes de haute et basse fréquence et en utilisant l'approximation du résidu $R_{\text{Schur}}(P)$, on peut arriver à une approximation de l'erreur.

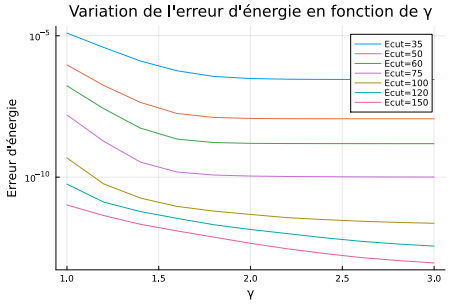


FIGURE 14 – Variation de l'erreur d'énergie en fonction de γ

L'erreur diminue lorsque le paramètre γ augmente, avec γ représentant le rapport entre l'énergie de coupure de la petite base et celle de la base variable, soit $\frac{E_{\text{init}}}{E_\gamma}$. La courbe d'erreur diminue progressivement et se stabilise à un certain facteur γ. On observe la même tendance pour différentes valeurs de E_{init} : plus on augmente E_{init}, plus l'erreur diminue, ce qui est conforme aux attentes. Une fois que la courbe d'erreur se stabilise, il devient inutile d'augmenter davantage γ, car la courbe reste plate et l'erreur ne varie plus à partir de ce point.

Nous nous intéressons maintenant à identifier ce facteur γ tout en minimisant le coût de calcul. Pour cela, on calcule la norme $\|M_{22}e_2 - (\Omega+K)_{22}e_2\|$ en utilisant e_2 qui vaut $M_{22}^{-1}R_2$. On remarque que cette norme augmente en fonction de γ et se stabilise à partir d'une certaine valeur de γ. Nous pouvons maintenant détecter le point de stagnation en utilisant un algorithme de dichotomie avec une erreur relative d'environ $\approx 1 \times 10^{-2}$ et le comparer à la stagnation de l'erreur précédemment obtenue avec le résidu R_{Schur}.

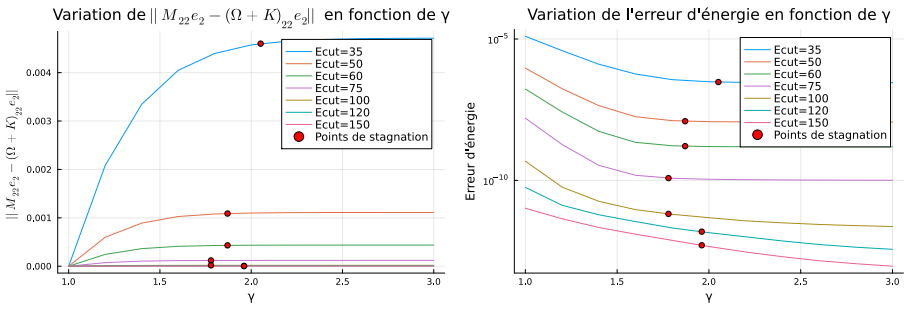


FIGURE 15 – Comparaison entre la stagnation de l'erreur

On constate que les points de stagnation sont très proches de celui obtenu avec le résidu R_{Schur}. Toutefois, pour des E_{cut} initiaux très élevés, il est nécessaire d'ajuster davantage l'erreur relative dans notre fonction de dichotomie pour se rapprocher du point de stagnation de la courbe d'erreur. Ainsi, on peut trouver le point de stagnation γ en inversant uniquement M_{22} sur les hautes fréquences, ce qui est moins coûteux, et obtenir avec ce γ une approximation de l'erreur.

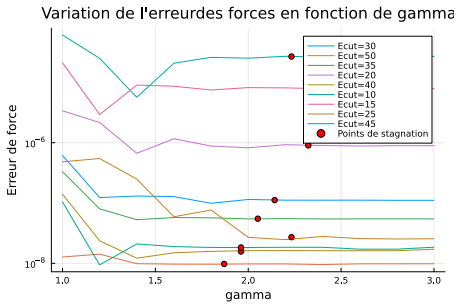


FIGURE 16 – variation de l'erreur des forces en fonction de γ

De manière similaire, on peut calculer l'erreur entre les forces en utilisant le résidu R_{Schur} et en déterminant les points de stagnation avec la même norme. Il est nécessaire d'ajouter un léger déplacement au niveau des atomes afin d'obtenir des forces non nulles. Ainsi, les points de stagnation calculés en utilisant la norme correspondent bien à la stagnation de l'erreur des forces.
Nous avons pu effectuer des calculs similaires pour des systèmes plus complexes, comme le TiO_2 et le GaAs, dans un cluster de calcul en 3D, et nous avons obtenu des résultats similaires. La stagnation calculée avec la méthode de dichotomie correspond à celle observée dans la stagnation de l'erreur.

Conclusion

Ce stage a approfondi la compréhension des méthodes numériques appliquées à la théorie de la fonctionnelle de densité (DFT), avec un accent sur l'estimation des erreurs et l'optimisation des algorithmes. Nous avons exploré diverses approches, comme la discrétisation par ondes planes et des méthodes de minimisation telles que la descente de gradient et l'algorithme de point fixe (SCF). La combinaison de ces méthodes avec des outils numériques avancés comme DFTK a permis d'améliorer l'efficacité tout en maintenant une précision satisfaisante dans les simulations.

L'étude des algorithmes a montré l'efficacité de la relaxation appliquée aux algorithmes de point fixe et de descente de gradient grâce à l'introduction d'un paramètre de relaxation variable, accélérant ainsi la convergence et réduisant les itérations. L'utilisation d'algorithmes stochastiques, comme l'ajustement aléatoire du paramètre de relaxation, a montré des résultats prometteurs en termes de réduction du temps de calcul. La multiplication par un paramètre aléatoire n'ajoute pas de complexité au code mais accélère considérablement la convergence.

L'approximation de Galerkin, propre à la méthode des ondes planes, s'est révélée particulièrement adaptée à la simulation des systèmes périodiques comme les cristaux. Les résultats ont montré que l'augmentation de l'énergie de coupure E_{cut} améliore la précision, mais au prix d'un coût computationnel accru. Nous avons proposé une méthode d'estimation de l'erreur basée sur la décomposition en basses et hautes fréquences, permettant de réduire le coût sans sacrifier la précision.

Enfin, l'estimation d'erreur via la méthode de Schur appliquée à la décomposition en fréquences a permis d'obtenir une approximation fiable tout en minimisant le coût des calculs. Cette approche pourrait être étendue à des systèmes plus complexes et servir de base à des méthodes d'optimisation futures pour des simulations à grande échelle.

Références

[1] Gaspard Kemlin. *Thèse : Analyse numérique pour la théorie de la fonctionnelle de densité* ,2022.

[2] Barzilai, J. and Borwein. *Journal : IMA Journal of Numerical Analysis*, 1988.

[3] Marcos Raydan et Benar F. Svaiter. *Publication : Relaxed Steepest Descent and Cauchy-Barzilai-Borwein Method.* ,2002

[4] Wikipedia *: Oscillateur harmonique quantique.*

[5] Laboratoire Amiénois de Mathématique Fondamentale et Appliquée (LAMFA), "Presentation of LAMFA," https://www.lamfa.u-picardie.fr/presentation

Annexes

1.Problème de minimisation

Soit H_0 un opérateur hermitien dont les valeurs propres ϵ_n sont associées aux vecteurs propres $|\varphi_n\rangle$. Nous souhaitons minimiser la trace $\mathrm{Tr}(H_0P)$ pour un projecteur P de rang N_{el}.

Pour un projecteur P de rang N_{el}, nous pouvons écrire :

$\mathrm{Tr(H_0}P) = \mathrm{Tr}\left(H_0 \sum_{i=1}^{N_{\mathrm{el}}} |\psi_i\rangle\langle\psi_i|\right) = \sum_{i=1}^{N_{\mathrm{el}}} \mathrm{Tr}(H_0|\psi_i\rangle\langle\psi_i|) = \sum_{i=1}^{N_{\mathrm{el}}} \langle\psi_i|H_0|\psi_i\rangle$, où $|\psi_i\rangle$ sont des vecteurs orthonormaux formant une base de dimension N_{el}. Sachant que H_0 a des vecteurs propres $|\varphi_n\rangle$, nous pouvons exprimer chaque $|\psi_i\rangle$ comme une combinaison linéaire de ces vecteurs propres :

$$|\psi_i\rangle = \sum_n c_{in}|\varphi_n\rangle,$$

où $c_{in} = \langle\varphi_n|\psi_i\rangle$. En utilisant cette décomposition, nous avons :

$$\langle\psi_i|H_0|\psi_i\rangle = \sum_n |c_{in}|^2\epsilon_n.$$

où ϵ_n sont les N_{el} plus petites valeurs propres de H_0. L'inégalité de Rayleigh stipule que pour un opérateur hermitien H et un sous-espace de dimension N_{el}, la valeur minimale de la forme quadratique associée est atteinte lorsque les vecteurs de base correspondent aux vecteurs propres associés aux plus petites valeurs propres de H. Ainsi, nous avons :

$$\mathrm{Tr}(H_0P) \geq \sum_{n=1}^{N_{\mathrm{el}}} \epsilon_n,$$

Or

$$\mathrm{Tr}(H_0P^*) = \sum_{n=1}^{N_{\mathrm{el}}} \langle\varphi_n|H_0|\varphi_n\rangle = \sum_{n=1}^{N_{\mathrm{el}}} \epsilon_n.$$

Cela implique que :

$$\mathrm{Tr}(H_0P) \geq \mathrm{Tr}(H_0P^*),$$

ce qui montre que le projecteur optimal P^* minimise la trace $\mathrm{Tr}(H_0P)$ parmi tous les projecteurs orthogonaux de rang N_{el}.

2.Théorèmes [1]

Théorème.1. *Sous des hypothèses appropriées, si $P_0 \in MN_{el}$ est suffisamment proche de P^*, l'Algorithme du gradient descent avec un β fixe converge linéairement vers P^* pour $\beta > 0$ suffisamment petit, avec un taux asymptotique $r(1 - \beta J_{grad})$ où $J_{grad} = \Omega^* + K^*$.*

Théorème.2. *Sous des hypothèses appropriées et si le principe de forte Aubfau est satisfait, alors, pour $\beta > 0$ suffisamment petit et $P_0 \in MN_{el}$ suffisamment proche de P^*, l'Algorithme SCF avec un β fixe converge linéairement vers P^*, avec un taux asymptotique $r(1 - \beta J_{SCF})$ où $J_{SCF} = 1 + \Omega^{-1}K^*$.*

3. Calcul du beta obtimal variable

3.1 Descente de gradient à pas variable 1

L'approximation de P_{k+1} en fonction de P_k et de β_k peut être exprimée comme :

$$P_{k+1} = P_k - \beta_k \Pi_{P_k}(\nabla E(P_k)),$$

où $\Pi_{P_k}(\nabla E(P_k))$ est la projection du gradient de l'énergie $\nabla E(P_k)$ sur le plan tangent passant par P_k.

On souhaite minimiser la fonction d'énergie suivante par rapport à β_k, en utilisant $P_{k+1}^2 = P_{k+1}$:

$$f(\beta_k) = \mathrm{Tr}(H_0P_{k+1}^2) + \frac{\alpha}{2\delta}\sum_{i=1}^{N_b}(P_{k+1})_{i,i}^2.$$

En substituant l'expression de P_{k+1} dans cette fonction, on obtient :

$$f(\beta_k) = \mathrm{Tr}\left(H_0 \left(P_k - \beta_k \Pi_{P_k}(\nabla E(P_k))\right)^2\right) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \left(\left(P_k - \beta_k \Pi_{P_k}(\nabla E(P_k))\right)_{i,i}\right)^2.$$

Pour le premier terme :

$$\mathrm{Tr}(H_0 P_{k+1}^2) = \mathrm{Tr}(H_0 P_k^2) - 2\beta_k \,\mathrm{Tr}\left(H_0 P_k \Pi_{P_k}(\nabla E(P_k))\right) + \beta_k^2 \,\mathrm{Tr}\left(H_0 (\Pi_{P_k}(\nabla E(P_k)))^2\right).$$

Pour le second terme :

$$\frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \left((P_k)_{i,i} - \beta_k (\Pi_{P_k}(\nabla E(P_k)))_{i,i}\right)^2 = \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \left((P_k)_{i,i}^2 - 2\beta_k (P_k)_{i,i} (\Pi_{P_k}(\nabla E(P_k)))_{i,i} + \beta_k^2 (\Pi_{P_k}(\nabla E(P_k)))_{i,i}^2\right).$$

Ainsi, la fonction d'énergie devient :

$$f(\beta_k) = \text{Constante} - 2\beta_k \left(\mathrm{Tr}\left(H_0 P_k \Pi_{P_k}(\nabla E(P_k))\right) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} (P_k)_{i,i} \left(\Pi_{P_k}(\nabla E(P_k))\right)_{i,i}\right)$$

$$+\beta_k^2 \left(\mathrm{Tr}\left(H_0 \left(\Pi_{P_k}(\nabla E(P_k))\right)^2\right) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \left(\Pi_{P_k}(\nabla E(P_k))\right)_{i,i}^2\right).$$

Pour minimiser $f(\beta_k)$, nous prenons la dérivée par rapport à β_k et la mettons à zéro :

$$\frac{df(\beta_k)}{d\beta_k} = -2 \left(\mathrm{Tr}\left(H_0 P_k \Pi_{P_k}(\nabla E(P_k))\right) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} (P_k)_{i,i} \left(\Pi_{P_k}(\nabla E(P_k))\right)_{i,i}\right)$$

$$+2\beta_k \left(\mathrm{Tr}\left(H_0 \left(\Pi_{P_k}(\nabla E(P_k))\right)^2\right) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \left(\Pi_{P_k}(\nabla E(P_k))\right)_{i,i}^2\right) = 0.$$

Ce qui nous donne :

$$\beta_k = \frac{\mathrm{Tr}(H_0 P_k \Pi_{P_k}(\nabla E(P_k))) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} (P_k)_{i,i} (\Pi_{P_k}(\nabla E(P_k)))_{i,i}}{\mathrm{Tr}(H_0 (\Pi_{P_k}(\nabla E(P_k)))^2) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} (\Pi_{P_k}(\nabla E(P_k)))_{i,i}^2}.$$

3.2 Descente de gradient à pas variable 3

La mise à jour de P est donnée par :

$$P_{k+1} = P_k - \beta_k r^k.$$

Le gradient d'énergie à l'itération $k+1$ est donné par :

$$H(P_{k+1}) = \nabla E(P_{k+1}) = r^k - \frac{\alpha}{\delta} \sum_{i=1}^{N_b} L_{k_{i,i}}.$$

où $L_k = \frac{\alpha}{\delta} \cdot \sum_{i=1}^{N_b} r_{k_{ii}}$ Le résidu à l'itération suivante est défini par :

$$r^{k+1} = \Pi_{P_{k+1}}(H(P_{k+1})) = \Pi_{P_{k+1}}(r^k) - \frac{\alpha}{\delta} \sum_{i=1}^{N_b} L_{k_{i,i}}.$$

Pour minimiser l'interaction entre r^k et r^{k+1}, nous voulons que :

$$\langle r^{k+1}, r^k \rangle = 0.$$

En substituant cette forme dans la condition de minimisation, nous obtenons une équation de troisième degré en β_k.

$$a_k \beta_k^3 + b_k \beta_k^2 + c_k \beta_k + d_k = 0,$$

où

- $a_k = -\,\mathrm{tr}\left(r^{k\prime} C\left(r^k, L_k\right)\right)$, où $L_k = \frac{\alpha}{\delta} \cdot \sum_{i=1}^{N_b} r_{k_{ii}}$, et $C(r_k, L_k) = -2 r_k L_k r_k$
- $b_k = \mathrm{tr}\left(r^{k\prime}\left(-B\left(r^k, L_k, P_k\right) + C\left(r^k, H_0\right)\right)\right)$, où $B(r_k, L_k, P_k) = -r_k L_k - L_k r_k + 2 P_k L_k r_k + 2 r_k L_k P_k$
- $c_k = \mathrm{tr}\left(r^{k\prime}\left(-\Pi_{P_k}\left(L_k\right) + B\left(r^k, H_0, P_k\right)\right)\right)$
- $d_k = \mathrm{tr}\left(r^{k\prime} \Pi_{P_k}(H_0)\right)$

3.3 Point fixe

L'approximation de P_{k+1} en fonction de P_k et de β_k peut être exprimée comme :

$$P_{k+1} = R\left(P_k + \beta_k \Pi_{P_k}\left(\Phi(P_k) - P_k\right)\right),$$

où $\Pi_{P_k}\left(\Phi(P_k) - P_k\right)$ est la projection du terme $\Phi(P_k) - P_k$ sur le plan tangent passant par P_k.

On souhaite minimiser la fonction d'énergie suivante par rapport à β_k :

$$f(\beta_k) = \operatorname{Tr}\left(H_0 (P_{k+1})^2\right) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \left((P_{k+1})_{i,i}\right)^2.$$

En substituant l'expression de P_{k+1} dans cette fonction, on obtient :

$$\begin{aligned} f(\beta_k) = \operatorname{Tr}\Big(H_0 \left(P_k + \beta_k \Pi_{P_k}\left(\Phi(P_k) - P^k\right)\right)^2\Big) \\ + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \left(P_k[i,i] + \beta_k \Pi_{P_k}(\Phi(P_k) - P_k)[i,i]\right)^2. \end{aligned}$$

Pour le premier terme :

$$\begin{aligned} \operatorname{Tr}\left(H_0 (P_{k+1})^2\right) = \operatorname{Tr}\left(H_0 (P_k)^2\right) + 2\beta_k \operatorname{Tr}\left(H_0 P_k \Pi_{P_k}(\Phi(P_k) - P_k)\right) \\ + \beta_k^2 \operatorname{Tr}\Big(H_0 \left(\Pi_{P_k}(\Phi(P_k) - P_k)\right)^2\Big). \end{aligned}$$

Pour le second terme :

$$\begin{aligned} \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \left(P_k[i,i] + \beta_k \Pi_{P_k}(\Phi(P_k) - P_k)[i,i]\right)^2 = \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \Big((P_k[i,i])^2 + 2\beta_k P_k[i,i] \Pi_{P_k}(\Phi(P_k) - P_k)[i,i] \\ + \beta_k^2 \left(\Pi_{P_k}(\Phi(P_k) - P_k)[i,i]\right)^2\Big). \end{aligned}$$

Ainsi, la fonction d'énergie devient :

$$\begin{aligned} f(\beta_k) = \text{Constante} + 2\beta_k \left(\operatorname{Tr}\left(H_0 P_k \Pi_{P_k}(\Phi(P_k) - P_k)\right) + \beta_k \frac{\alpha}{\delta} \sum_{i=1}^{N_b} P_k[i,i] \Pi_{P_k}(\Phi(P_k) - P_k)[i,i] \right. \\ \left. + \beta_k^2 \left(\operatorname{Tr}\Big(H_0 \left(\Pi_{P_k}(\Phi(P_k) - P_k)\right)^2\Big) + \beta_k^2 \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \left(\Pi_{P^k}(\Phi(P_k) - P_k)[i,i]\right)^2. \right. \end{aligned}$$

Pour minimiser $f(\beta_k)$, nous prenons la dérivée par rapport à β_k et la mettons à zéro :

$$\begin{aligned} \frac{df(\beta_k)}{d\beta_k} = 2\left(\operatorname{Tr}\left(H_0 P_k \Pi_{P_k}(\Phi(P_k) - P_k)\right) + \frac{\alpha}{\delta} \sum_{i=1}^{N_b} P_k[i,i] \Pi_{P_k}(\Phi(P_k) - P_k)[i,i] \right. \\ \left. + 2\beta_k \Big(\operatorname{Tr}\Big(H_0 \left(\Pi_{P_k}(\Phi(P_k) - P_k)\right)^2\Big) + \frac{\alpha}{\delta} \sum_{i=1}^{N_b} \left(\Pi_{P_k}(\Phi(P_k) - P_k)[i,i]\right)^2 = 0. \right. \end{aligned}$$

Ce qui nous donne :

$$\beta_k = -\frac{\operatorname{Tr}\left(H_0 \Pi_{P_k}\left(\Phi(P_k) - P_k\right) P_k\right) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} P_k[i,i] \Pi_{P_k}\left(\Phi(P_k) - P_k\right)[i,i]}{\operatorname{Tr}\Big(H_0 \left(\Pi_{P_k}\left(\Phi(P_k) - P_k\right)\right)^2\Big) + \frac{\alpha}{2\delta} \sum_{i=1}^{N_b} \left(\Pi_{P_k}\left(\Phi(P_k) - P_k\right)[i,i]\right)^2}.$$

4. Calcul du beta obtimal fixe

Pour obtenir la convergence la plus rapide avec les méthodes de point fixe et gradient descent avec un beta fixe, il faut que les valeurs propres de $1 - \beta J$ soient aussi proches que possible de 0. La vitesse de convergence dépend donc des valeurs propres de la matrice J et du paramètre β.

Soit λ_1 la plus petite et λ_N la plus grande valeur propre de J. Nous cherchons à minimiser la distance maximale entre 1 et ces valeurs propres multipliées par β. Autrement dit, nous voulons minimiser :

$$\max\left\{\left|1 - \beta\lambda_1\right|, \left|1 - \beta\lambda_N\right|\right\}.$$

Le pas optimal β_* est celui qui minimise cette quantité. Pour le trouver, nous résolvons :

$$\beta_* = \arg\min_{\beta} \max\left\{\left|1-\beta\lambda_1\right|, \left|1-\beta\lambda_N\right|\right\}.$$

La solution est :

$$\beta_* = \frac{2}{\lambda_1+\lambda_N}.$$

En utilisant ce pas optimal, le taux de convergence r est donné par :

$$r = \frac{\kappa-1}{\kappa+1},$$

où $\kappa = \frac{\lambda_N}{\lambda_1}$ est le conditionnement spectral de J. Ce taux mesure la rapidité de la convergence : plus r est proche de 0, plus la convergence est rapide. Si κ est grand (les valeurs propres sont très dispersées), r sera proche de 1, indiquant une convergence plus lente.

5. Autre méthodes

5.1 Descente de gradient à pas variable 3

Le paramètre β est calculé comme solution d'une équation de troisième degré dans le cadre de la descente de gradient pour minimiser l'interaction entre le résidu r^k à l'itération k et le résidu r^{k+1} à l'itération suivante. Cette minimisation se fait en cherchant à réduire $\left\langle r^k, r^{k+1}\right\rangle$, où r^{k+1} est défini par $\Pi_{P_{k+1}}\left(\nabla E\left(P_{k+1}\right)\right)$.

L'expression de P_{k+1}, obtenue par la mise à jour

$$P_{k+1} = R\left(P_k - \beta\Pi_{P_k}\left(\nabla E\left(P_k\right)\right)\right),$$

conduit à une équation cubique en β de la forme :

$$a_k\beta_k^3 + b_k\beta_k^2 + c_k\beta_k + d_k = 0.$$

Les coefficients a, b, c, et d dépendent du résidu, de la projection, ainsi que de la rétraction utilisée pour rester sur la variété. Ils sont donnés par :

- $a_k = -\operatorname{tr}\left(r^{k\prime}C\left(r^k, L_k\right)\right)$, où $L_k = \frac{\alpha}{\delta}\cdot\sum_{i=1}^{N_b} r_{k_{ii}}$, et $C(r_k, L_k) = -2r_kL_kr_k$
- $b_k = \operatorname{tr}\left(r^{k\prime}\left(-B\left(r^k, L_k, P_k\right) + C\left(r^k, H_0\right)\right)\right)$, où $B(r_k, L_k, P_k) = -r_kL_k - L_kr_k + 2P_kL_kr_k + 2r_kL_kP_k$
- $c_k = \operatorname{tr}\left(r^{k\prime}\left(-\Pi_{P_k}\left(L_k\right) + B\left(r^k, H_0, P_k\right)\right)\right)$
- $d_k = \operatorname{tr}\left(r^{k\prime}\Pi_{P_k}(H_0)\right)$

Ces coefficients sont obtenus en linéarisant l'expression du résidu après mise à jour, puis en rétractant cette expression sur la variété (voir annexe). On utilise la bibliothèque Roots.jl sur Julia pour calculer les racines de cette équation à chaque itération, et on prend la plus petite racine réelle. La multiplication de ce β_k par une loi uniforme dans l'intervalle $[0, 2]$, comme dans le cas des pas variables calculés précédemment, permet d'accélérer la convergence .

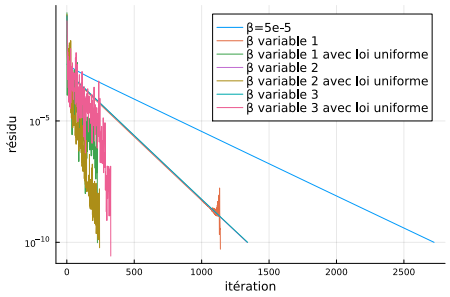


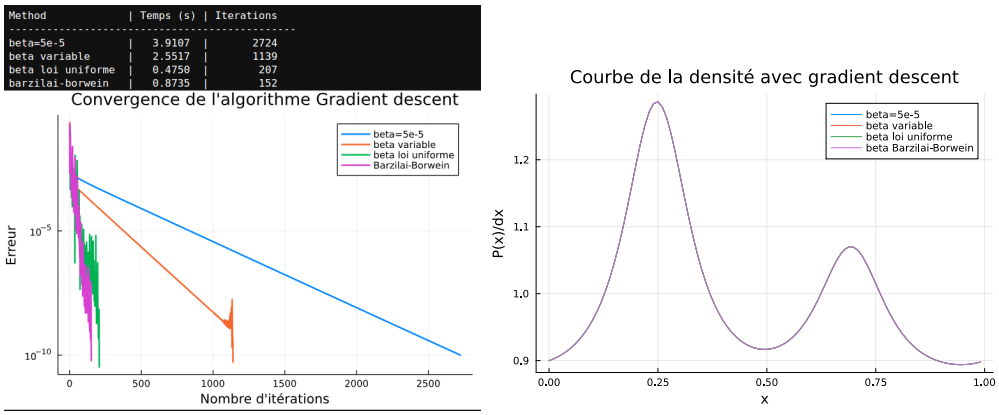
FIGURE 17 – Convergence de l'algorithme

5.2 Méthode de Barzilai-Borwein

La méthode de Barzilai-Borwein [2] est une méthode numérique utilisée pour résoudre des problèmes d'optimisation non linéaires. On choisit un β_k variable de la forme :

$$\beta_k = \frac{\|P_k - P_{k-1}\|^2}{\operatorname{Tr}((P_k - P_{k-1})^T(\Pi_{P_k}\nabla E(P_k) - \Pi_{P_{k-1}}\nabla E(P_{k-1})))}$$

Cette méthode permet d'accélérer la convergence. Cependant, la courbe d'erreur ne décroît pas de manière uniforme ; elle oscille avant d'atteindre la convergence.



((a)) Convergence des différentes descentes de gradient

((b)) Densité avec différentes descentes de gradient

FIGURE 18 – Convergence des algorithmes

La méthode de Barzilai-Borwein converge plus rapidement que la méthode utilisant un β variable. Cependant, lorsque le β variable est multiplié par un facteur issu d'une loi uniforme, la convergence de cette méthode s'améliore considérablement et se rapproche de celle de la méthode de Barzilai-Borwein.

5.3 Point fixe avec double diagonalisation

Une autre méthode pour optimiser la convergence de la méthode SCF en ajustant la matrice de densité de manière sophistiquée. À chaque itération, le Hamiltonien est mis à jour selon :

$$H_k = H_0 + \left(\frac{\alpha}{\delta}\right) \sum_{i=1}^{\mathrm{Nb}} P_k[i,i]$$

où h est le Hamiltonien de base et α est un paramètre d'ajustement. La fonction Φ, qui projette sur les vecteurs propres associés aux plus petites valeurs propres de la matrice, est utilisée pour obtenir les matrices F_k et F_{2k} :

$$\begin{cases} F_k = \Phi(H_k), \\ H'_k = H_0 + \frac{\alpha}{\delta} \sum_{i=1}^{\mathrm{Nb}} F_k[i,i] \\ F_{2k} = \Phi(H'_k). \end{cases}$$

On définit la différence δ_k comme :

$$\delta_k = F_{2k} - 2F_k + P_k.$$

Le paramètre de relaxation β_k est alors calculé pour optimiser la mise à jour :

$$\beta_k = -\frac{\mathrm{Tr}(\delta_k (F_k - P_k)')}{\|\delta_k\|^2},$$

où $\|\delta_k\|_F$ représente la norme de Frobenius de δ_k. La mise à jour de la matrice densité est ensuite effectuée comme :

$$P_{k+1} = R(P_k + \beta_k \Pi_{P_k}(\Phi(H_k) - P_k))$$

Cette méthode ajuste le pas de mise à jour de manière optimale pour converger plus rapidement vers une solution auto-cohérente.

6. Figures

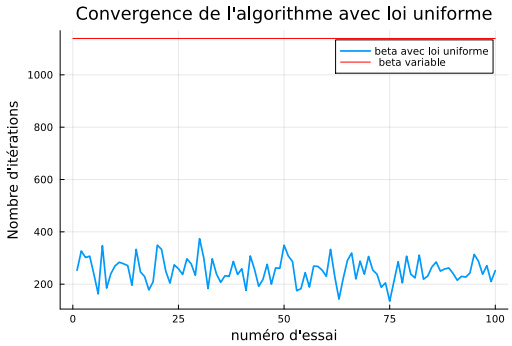


FIGURE 19 – Convergence du gradient descent avec beta qui suit la loi uniforme dans 100 essais

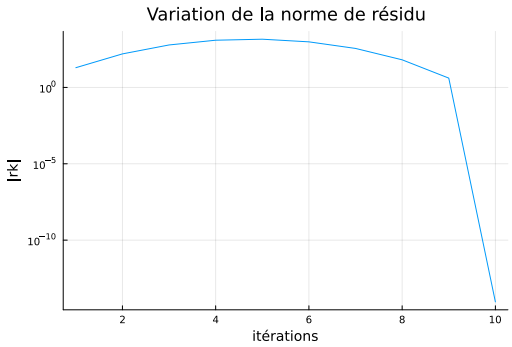


FIGURE 20 – Variation du résidu pour $\beta_k = \frac{1}{\lambda_k}$

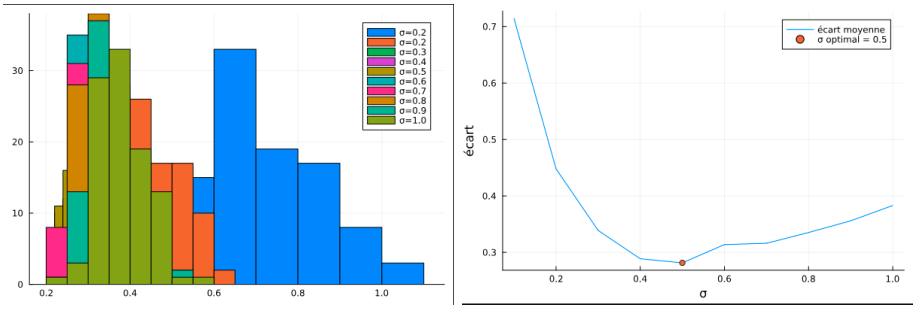


FIGURE 21 – Variation de l'écart en fonction de σ pour l'équation de Gross-Pitaevskii

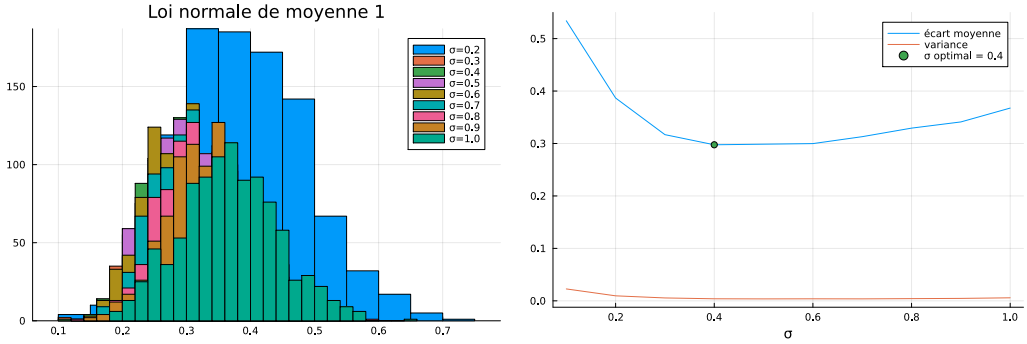


FIGURE 22 – Variation de l'écart en fonction de σ pour $Ax = b$ en utilisant la loi normale

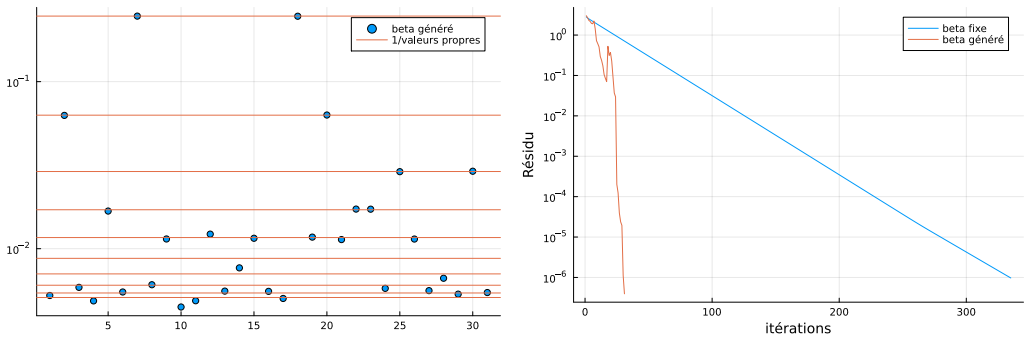


FIGURE 23 – Convergence avec la loi de probabilité construite

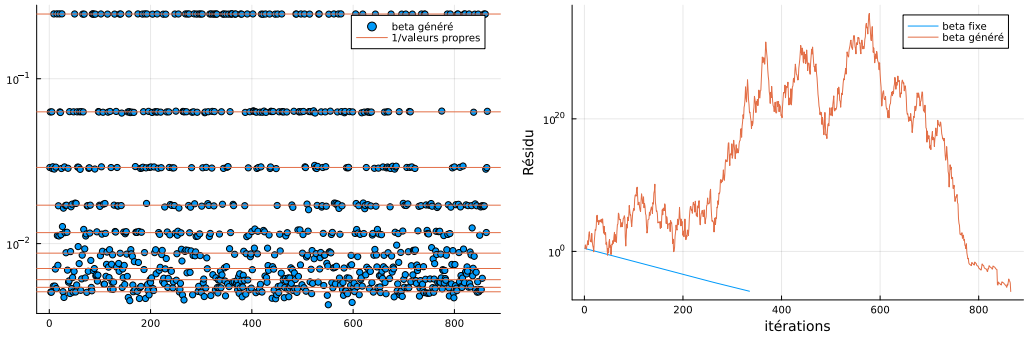


FIGURE 24 – Convergence avec la loi de probabilité construite

Annexe CREGE

Le Laboratoire Amiénois de Mathématique Fondamentale et Appliquée (LAMFA) est une unité mixte de recherche affiliée au CNRS (UMR CNRS 7352) et à l'Université de Picardie Jules Verne. Il regroupe les mathématiciennes et mathématiciens de l'Université, jouant un rôle essentiel dans le développement des mathématiques, en combinant recherche fondamentale et applications pratiques. Le LAMFA est structuré en trois équipes de recherche :

- **Analyse Appliquée** (A^3)
- **Groupes, Algèbre et Topologie (GAT)**
- **Systèmes Dynamiques - Probabilités - Arithmétique (SymPA)**

De plus, une équipe transverse a pour objectif de promouvoir l'image des mathématiques auprès du grand public.

Structure du LAMFA

Le LAMFA est organisé de manière à favoriser la collaboration et l'efficacité au sein du laboratoire. Voici un résumé de l'organigramme :

- **Direction** :
  - Directeur
  - Directeur adjoint
- **Secrétariat** :
  - Secrétariat général
  - Secrétariat CNRS
- **Responsables d'équipes** :
  - Responsable de l'équipe Analyse Appliquée
  - Responsable de l'équipe GAT
  - Responsable de l'équipe SymPA
  - Responsable de l'équipe Mathématiques et Grand Public
- **Référents** :
  - Référent parité
  - Référent développement durable
  - Correspondant diffusion
  - Correspondant communication
  - Correspondant valorisation
  - Correspondant international
- **Conseil du laboratoire** :
  - Membres représentant les professeurs
  - Membres représentant les maîtres de conférences
  - Membres du personnel administratif
  - Représentant des doctorants

Environnement Interne

Au sein du LAMFA, la gestion de projet est un élément clé qui contribue à l'efficacité et à la réussite des recherches menées. Le laboratoire est structuré en plusieurs équipes spécialisées, favorisant la collaboration et la synergie entre les chercheurs. La communication interne est facilitée par des réunions régulières, des séminaires hebdomadaires et l'utilisation de plateformes collaboratives, ce qui permet un partage efficace des informations et des ressources.

Chaque lundi matin, l'équipe (A^3) organise des séminaires où des intervenants de différentes universités présentent leurs travaux. J'ai assisté à l'un de ces séminaires, qui portait sur la mécanique quantique. Cette présentation m'a fourni des informations précieuses et a élargi ma compréhension des applications mathématiques dans ce domaine. Ces événements sont une excellente occasion de découvrir de nouvelles perspectives et de renforcer la cohésion interne.

La culture organisationnelle du LAMFA valorise l'innovation, la rigueur scientifique et la collaboration. Le laboratoire collabore étroitement avec d'autres instituts de recherche, tant au niveau national qu'international, ce qui favorise l'échange d'idées et la réalisation de projets interdisciplinaires. Ces collaborations enrichissent les travaux menés au sein du laboratoire et ouvrent la voie à de nouvelles découvertes.

De plus, le LAMFA accorde une grande importance à la formation des doctorants et des stagiaires à la recherche. Les jeunes chercheurs sont encadrés par des professeurs expérimentés, bénéficiant ainsi d'un environnement propice au développement de leurs compétences. Le laboratoire est également impliqué dans l'enseigne-

ment des mathématiques à l'université, contribuant à la formation des étudiants en licence et en master. Les membres du LAMFA participent activement à la préparation à l'agrégation de mathématiques, formant ainsi les futurs professeurs et renforçant le lien entre recherche et enseignement.

Expérience Personnelle : Journée du LAMFA

J'ai eu l'opportunité d'assister à la Journée du LAMFA 2024, un événement important où quatre chercheurs du laboratoire ont présenté leurs sujets de recherche. Parmi eux se trouvait mon tuteur, ce qui m'a permis de mieux comprendre mon sujet et les travaux auxquels il s'applique. Cette journée a été particulièrement enrichissante, offrant un aperçu des différentes thématiques abordées au sein du laboratoire.

L'événement incluait également un petit-déjeuner et un déjeuner offerts par le laboratoire, favorisant les échanges informels entre les participants. Ces moments de convivialité ont renforcé les liens au sein de la communauté du LAMFA et ont permis d'engager des discussions constructives sur les projets en cours.

Missions du LAMFA

L'équipe Mathématiques et Grand Public est une équipe transverse du LAMFA. Son objectif est de promouvoir l'image des mathématiques auprès du grand public, notamment des collégiens et des lycéens.

Formation des Étudiants en Licence et Master Le LAMFA s'investit non seulement dans la formation des étudiants en master, mais également dans celle des étudiants en licence après le lycée. Il offre des cours et des séminaires spécialisés pour former les futurs enseignants et chercheurs. Le laboratoire est impliqué dans la préparation à l'agrégation de mathématiques et assure un encadrement de qualité pour les doctorants et post-doctorants, favorisant ainsi leur insertion dans la communauté scientifique.

Interventions dans les Établissements Scolaires de Picardie Le LAMFA intervient activement dans les établissements scolaires de la région Picardie à travers divers programmes éducatifs et ateliers de sensibilisation aux mathématiques. Il organise des exposés dans les collèges et lycées, ainsi que des stages MathC2+ axés sur des thématiques mathématiques comme l'atelier sur le triangle de Pascal en 2024. Le laboratoire participe également au Rallye Mathématiques Inter-classes, dont la finale se tient à l'Université de Picardie Jules Verne (UPJV). Par ailleurs, des ateliers *Maths en Jeans* sont animés par des chercheurs du LAMFA dans plusieurs établissements, comme les collèges et lycées de Beauvais, Chauny et Rue. Ces actions visent à promouvoir l'excellence en mathématiques et à encourager les élèves à s'engager dans cette discipline.

Collaborations Interdisciplinaires Les membres du LAMFA interviennent dans diverses disciplines scientifiques, pas seulement en mathématiques mais aussi en biologie et dans d'autres domaines. Le laboratoire entretient des collaborations avec des partenaires industriels et des scientifiques d'autres disciplines, ce qui permet d'appliquer les outils mathématiques à des problématiques concrètes en physique, ingénierie, sciences de l'information, biologie, etc. Ces partenariats contribuent au développement de solutions innovantes et enrichissent les travaux du LAMFA en ouvrant de nouvelles perspectives pour les applications des mathématiques.

Stages pour Collégiens et Lycéens Un stage d'observation d'une semaine peut être effectué au sein du laboratoire par des élèves de $3^{\text{ème}}$ ou de 2^{nde}. Au programme : entretiens avec le personnel de l'université, participation à des séminaires de recherche, des groupes de travail, des cours et activités de réflexion. Les élèves intéressés sont invités à contacter le laboratoire pour organiser leur venue et remplir les formalités administratives.

Vulgarisation des Mathématiques et Mise en Valeur des Mathématiciennes Le LAMFA est activement impliqué dans la promotion des mathématiques auprès du grand public. Il participe à des manifestations nationales comme la Fête de la Science, où il tient un stand "Mathématiques", et au Salon des Jeux Mathématiques à Paris, en tenant le stand de la SMAI. De plus, le laboratoire présente des exposés lors de la remise des prix des Olympiades de Mathématiques, contribuant ainsi à valoriser l'excellence des élèves et à susciter l'intérêt du public pour la discipline.

Dans le cadre de la mise en valeur des mathématiciennes, le LAMFA s'engage dans des actions visant à promouvoir la place des femmes en mathématiques. De courtes présentations de mathématiciennes sont diffusées au sein de l'université, mettant en lumière leurs parcours et contributions. Le laboratoire met également à disposition des établissements l'exposition *Portraits de mathématiciennes* (voir http://womeninmath.net/), une exposition itinérante retraçant le parcours scientifique et personnel de treize mathématiciennes de divers pays d'Europe.