

Compte-rendu du TP3 : Méthodes Multigrilles

Ines BESBES
Sara ROOL

March 2025

1 Introduction

Durant ce TP, on a implémenté une méthode multigrilles en une dimension (1D) pour résoudre des systèmes linéaires issus d'équations différentielles partielles. Ici, on souhaite résoudre un problème de Poisson avec des conditions aux limites de Dirichlet homogènes, en utilisant une solution exacte pour valider notre implémentation.

2 Multigrilles 1D

2.1 Formulation du problème

Le problème de Poisson en une dimension est défini par l'équation suivante :

$$\begin{aligned} -u'' &= f, & \text{in } \Omega = [0, 1] \\ u &= 0, & \text{on } \Gamma \end{aligned} \tag{1}$$

On suppose que la solution exacte est donnée par

$$u(x) = x(1 - x) \tag{2}$$

ce qui permet de déterminer le membre de droite

$$f = -u'' = 2 \tag{3}$$

Cette solution exacte est utilisée pour évaluer les erreurs introduites par la discrétisation par différences finies et pour valider notre implémentation.

2.2 Vérification des conditions aux limites de Dirichlet

Pour vérifier que la solution $u(x) = x(1 - x)$ respecte les conditions aux limites de Dirichlet pour le problème de Poisson en une dimension, on doit examiner les valeurs de $u(x)$ aux points frontières du domaine $\Omega = [0, 1]$.

Les conditions aux limites de Dirichlet spécifient que $u = 0$ sur Γ , où Γ représente les points frontières du domaine. Dans ce cas, Γ inclut les points $x = 0$ et $x = 1$.

- À $x = 0$:

$$u(0) = 0(1 - 0) = 0$$

La solution $u(x)$ est donc 0 à $x = 0$.

- À $x = 1$:

$$u(1) = 1(1 - 1) = 0$$

La solution $u(x)$ est donc 0 à $x = 1$.

Puisque $u(x) = 0$ aux deux points frontières $x = 0$ et $x = 1$, la solution $u(x) = x(1 - x)$ respecte bien les conditions aux limites de Dirichlet homogènes pour ce problème.

2.3 Maillage

On a développé un solveur à deux niveaux, utilisant deux maillages : un maillage fin et un maillage grossier. Pour un intervalle $[0, 1]$ avec N éléments, on obtient les deux domaines suivants :

- **Maillage fin** (Ω_h) : $x_i = i \cdot h$, où $h = \frac{1}{N}$ et $i = 0, \dots, N$.
- **Maillage grossier** (Ω_{2h}) : $x_i = 2i \cdot h$, où $i = 0, \dots, \frac{N}{2}$.

3 Discrétisation par différences finies

On utilise l'approche différence centrée :

$$u_i'' \approx \frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} = f_i, \quad i = 1, \dots, N-1 \quad (4)$$

Avec cette discrétisation, on obtient, sur la grille fine, le système linéaire :

$$\mathbf{A}\mathbf{u} = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{pmatrix} \quad (5)$$

Étant donné les conditions aux limites de Dirichlet homogènes, il n'est pas nécessaire d'inclure les valeurs u_0 et u_N dans la matrice. Par conséquent, la matrice du problème sur la grille fine est de dimension $N-1$.

4 Lisseur de Jacobi pondéré

Résoudre directement ce système linéaire avec des méthodes directes (comme la décomposition LU) peut être coûteux en termes de calcul et de mémoire, surtout pour les grands problèmes (grilles fines). Les méthodes itératives, comme les lisseurs, sont donc une alternative. Leur but n'est pas de résoudre précisément le système, mais d'essayer d'éliminer les composantes haute fréquence de l'erreur.

D'où le fait qu'on ait choisi, pour ce TP, d'utiliser la méthode de Jacobi pondérée :

$$\mathbf{u}^{(m+1)} = (I - \omega D^{-1}A) \mathbf{u}^{(m)} + \omega D^{-1}\mathbf{f} \quad (6)$$

avec $0 \leq \omega \leq \frac{2}{\lambda_{\max}(D^{-1}A)}$.

En effet, la méthode de Jacobi pondérée a la propriété d'atténuer rapidement les composantes de haute fréquence de l'erreur dans la solution. En d'autres termes, après quelques itérations, l'erreur devient plus "lisse".

On a donc implémenté en Matlab une fonction *weighted_jacobi* qui effectue m itérations de la méthode de Jacobi pondérée pour améliorer l'approximation de la solution du système linéaire $Au = f$, en utilisant le facteur de pondération ω .

La matrice A , issue de la discrétisation du Laplacien, possède un ensemble de vecteurs propres orthogonaux. Chaque vecteur propre correspond à une fréquence spatiale particulière. Les vecteurs propres associés aux plus petites valeurs propres correspondent aux basses fréquences, tandis que ceux associés aux plus grandes valeurs propres correspondent aux hautes fréquences.

La méthode de Jacobi pondérée amortit efficacement les composantes de haute fréquence de l'erreur beaucoup plus rapidement que les composantes de basse fréquence. Plus la fréquence est élevée (plus k est grand), plus le mode est amorti.

On va maintenant observer l'effet d'amortissement de la méthode de Jacobi pondérée pour différents modes, et en faisant varier le nombre d'itérations m ainsi que le paramètre ω .

4.1 Mode $k=3$ (basse fréquence)

Effet d'amortissement de la méthode de Jacobi pondérée pour $k = 3$

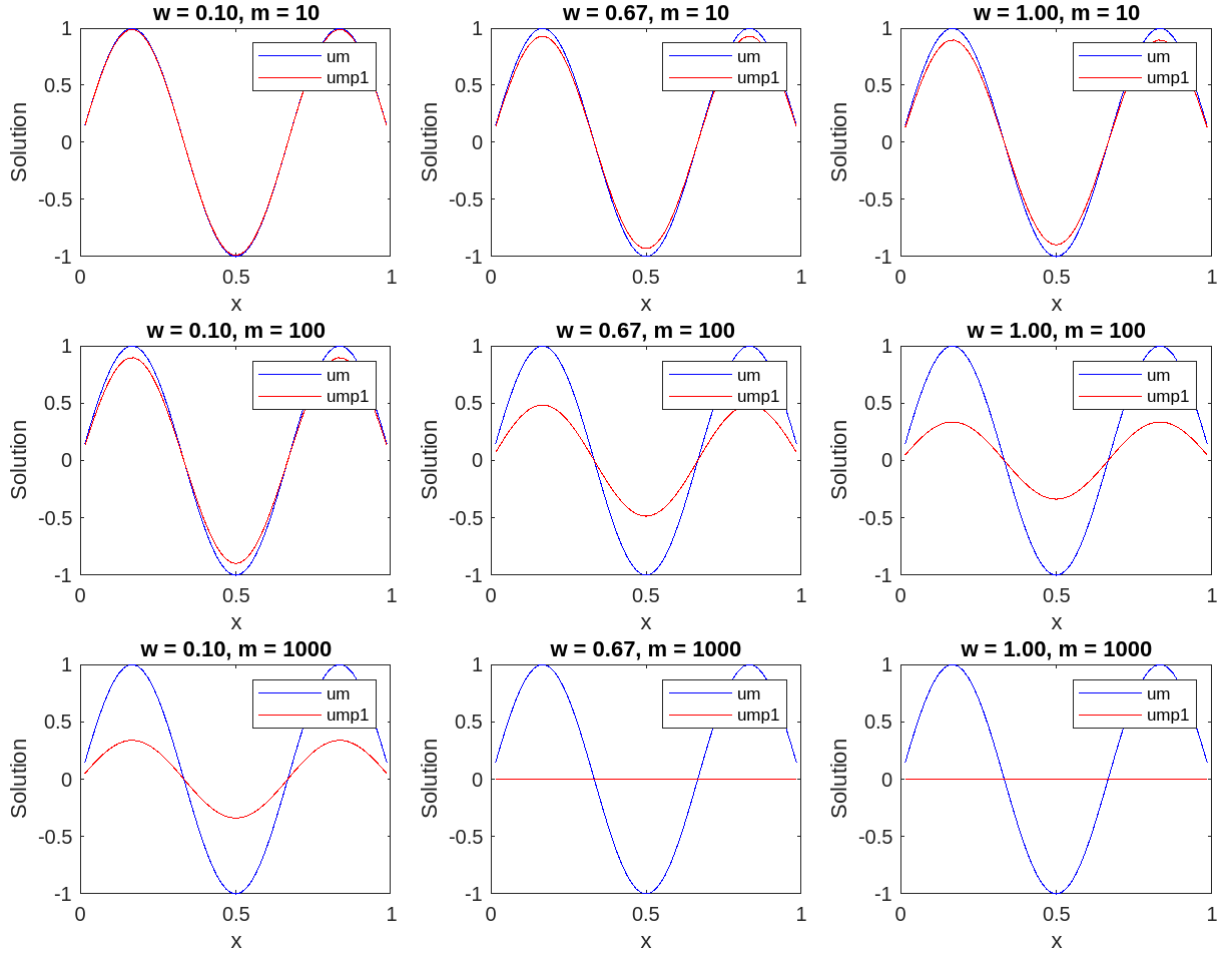


Figure 1: Effet d'amortissement de la méthode de Jacobi pondérée pour $k=3$

On observe que l'amortissement est très lent. Avec $m = 10$, l'effet est minime, surtout pour $\omega = 0.1$. Un amortissement un peu plus important commence à apparaître avec 100 itérations, particulièrement pour $\omega = \frac{2}{3}$ et $\omega = 1$. Avec 1000 itérations, l'amortissement devient significatif même pour $\omega = 0.1$, et il est presque total pour les deux autres valeurs de ω . Ce mode est donc très difficile à amortir avec la méthode de Jacobi pondérée.

4.2 Mode $k=12$ (fréquence moyenne)

Effet d'amortissement de la méthode de Jacobi pondérée pour $k = 12$

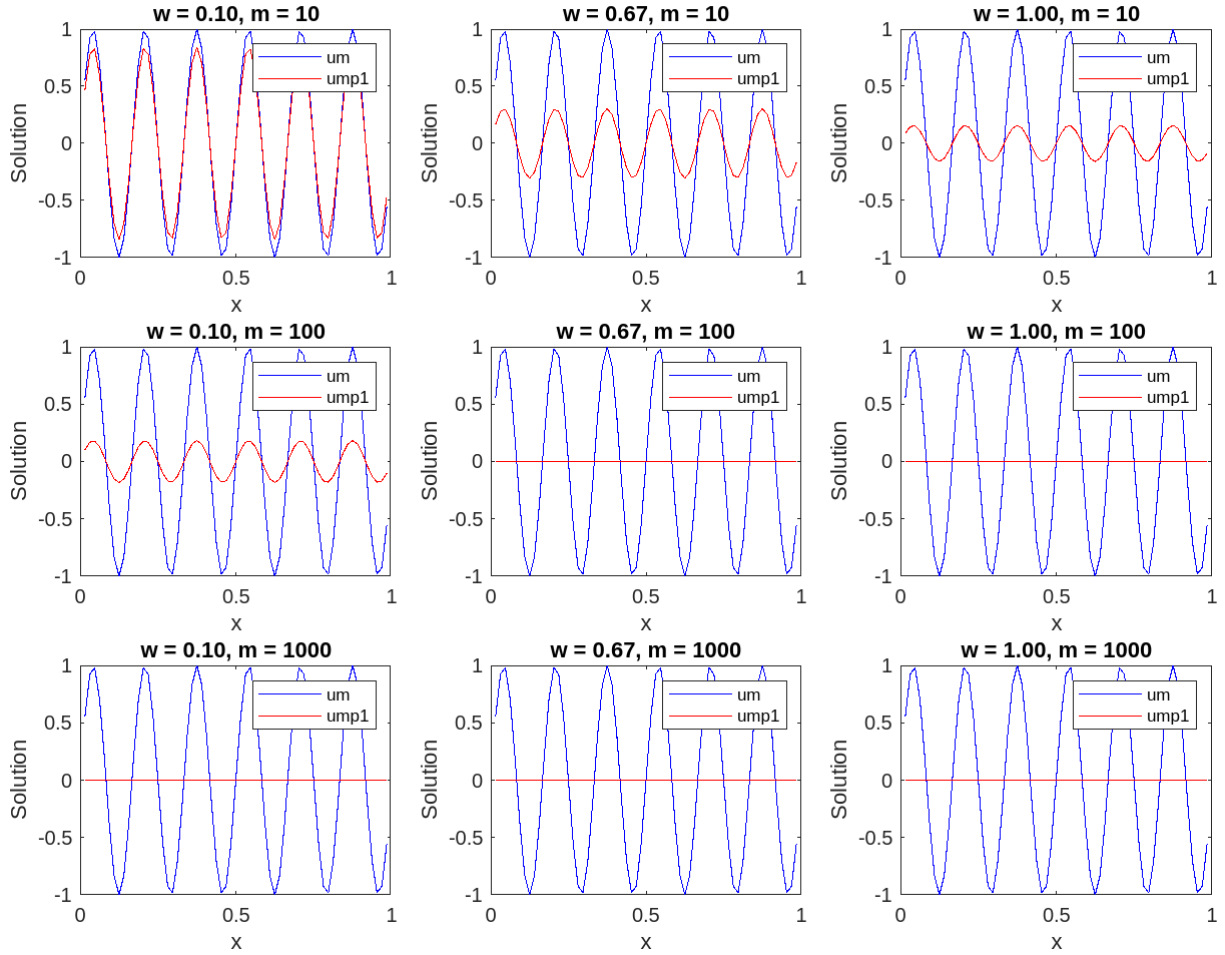


Figure 2: Effet d'amortissement de la méthode de Jacobi pondérée pour $k=12$

Ce mode a un amortissement plus rapide que le mode $k = 3$. Après 10 itérations, l'amortissement est déjà visible, surtout pour $\omega = \frac{2}{3}$ et $\omega = 1$. Après 1000 itérations, l'amortissement est pratiquement complet pour toutes les valeurs de ω .

4.3 Effet de lissage pour $k=48$

Effet d'amortissement de la méthode de Jacobi pondérée pour $k = 48$

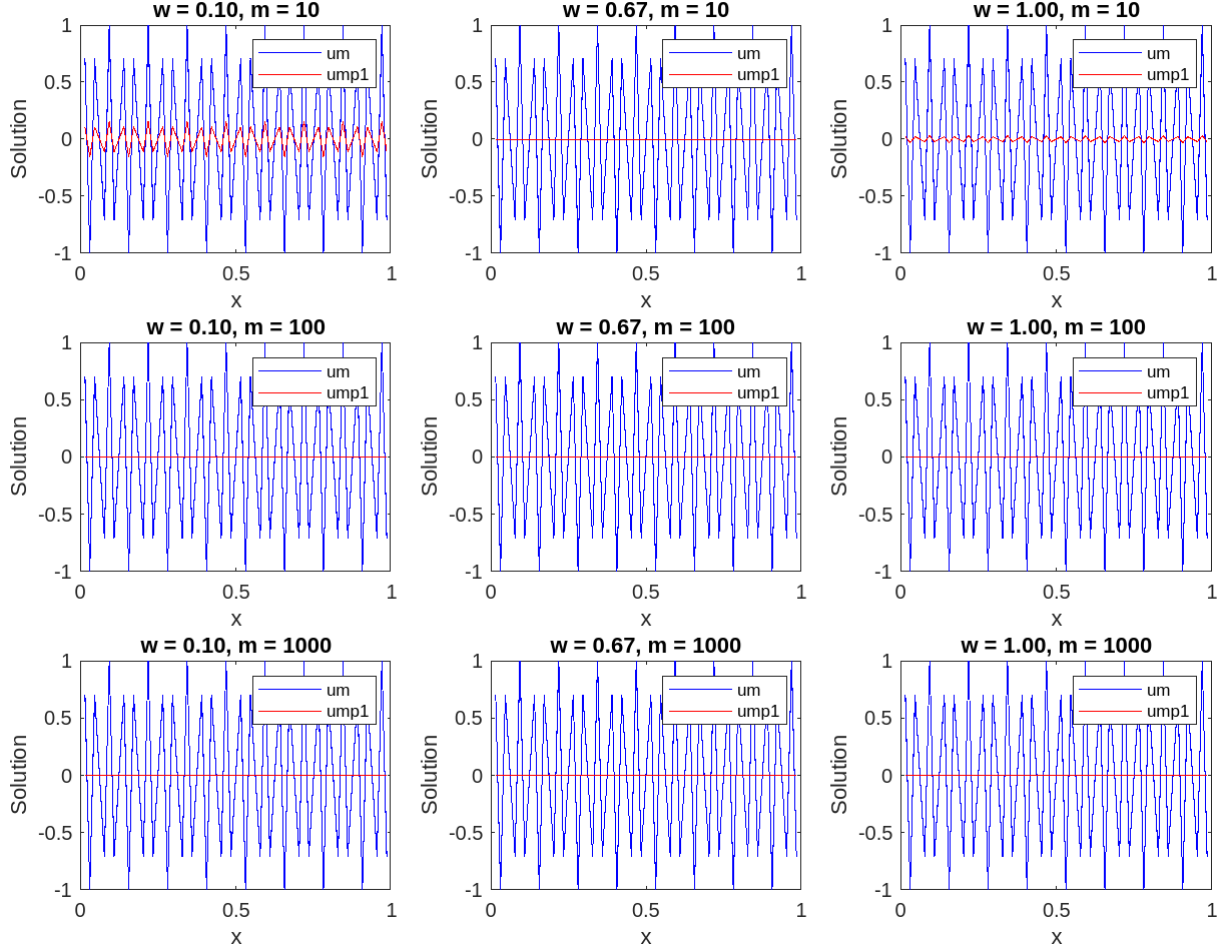


Figure 3: Effet d'amortissement de la méthode de Jacobi pondérée pour $k=48$

On observe que l'amortissement est très rapide. Même avec seulement 10 itérations, l'amplitude est considérablement réduite, surtout pour $\omega = \frac{2}{3}$ et $\omega = 1$. Avec 100 et 1000 itérations, la solution est pratiquement nulle pour $\omega = \frac{2}{3}$ et $\omega = 1$. Même avec $\omega = 0.1$, l'amplitude de la solution est fortement réduite. Ce mode est donc le plus efficacement amorti par la méthode de Jacobi pondérée.

4.4 Interprétation globale

On constate que les modes à haute fréquence sont amortis beaucoup plus rapidement que les modes à basse fréquence, ce qui confirme l'intérêt de la méthode. Le paramètre de relaxation ω joue un rôle crucial, et on remarque que $\omega = \frac{2}{3}$ semble le plus optimal, particulièrement pour les hautes fréquences. Enfin, le nombre d'itérations m influence l'amplitude de l'amortissement. En effet, un nombre plus élevé d'itérations conduit à un amortissement plus prononcé. Ainsi, la méthode de Jacobi pondérée agit comme un filtre passe-bas.

5 Opérateurs de transfert

Le passage d'une grille fine à une grille grossière requiert des opérateurs de transfert. Ici, la prolongation s'effectue par interpolation linéaire $I_{2h}^h : \mathbb{R}^{\frac{N}{2}-1} \rightarrow \mathbb{R}^{N-1}$ avec

$$\begin{aligned} v_{2j}^h &= v_j^{2h}, \\ v_{2j+1}^h &= \frac{1}{2}(v_j^{2h} + v_{j+1}^{2h}) \end{aligned}$$

et la restriction par le full weighting $I_h^{2h} : \mathbb{R}^{N-1} \rightarrow \mathbb{R}^{\frac{N}{2}-1}$ avec

$$v_j^{2h} = \frac{1}{4}(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h)$$

La prolongation interpole linéairement les valeurs de la grille grossière sur la grille fine, et la restriction calcule les valeurs de la grille grossière comme une moyenne pondérée des valeurs voisines sur la grille fine. De plus, on a la relation suivante :

$$\begin{aligned} I_{2h}^h &: \mathbb{R}^{n/2-1} \rightarrow \mathbb{R}^{n-1} \\ I_h^{2h} &= c(I_{2h}^h)^T \text{ avec } c \in \mathbb{R} \end{aligned}$$

L'avantage de choisir le couple interpolation linéaire et full weighting vient du fait qu'ils puissent bien représenter les composantes basses fréquences de la solution sur les différentes grilles. Le full weighting, qui est aussi la transposée de l'interpolation linéaire, assure une bonne transmission de l'information entre les grilles et contribue à la stabilité de la méthode multigrille. Cette combinaison permet de réduire efficacement l'erreur sur toutes les grilles, accélérant ainsi la convergence globale de la méthode.

6 Schéma de multigrilles à deux niveaux

Après avoir choisi nos opérateurs de transfert, on a implémenté la méthode de multigrilles sur deux niveaux, afin de résoudre notre système linéaire.

6.1 Fonctionnement de la méthode

Soit $v^{(k)}$ la solution approximative à l'itération k , et u_{ref} la solution de référence exacte du système linéaire. L'erreur à l'itération k est définie comme :

$$e^{(k)} = u_{ref} - v^{(k)}$$

Pour chaque itération, le processus comprend les étapes suivantes :

- **Lissage avec Jacobi pondéré** : Cette étape réduit les composantes de haute fréquence de l'erreur sur la grille fine en utilisant la méthode de Jacobi pondérée.

$$v^{(k)} \leftarrow \text{weighted_jacobi}(A, v^{(k)}, f, \omega, m)$$

- **Calcul du résidu** : On calcule le résidu, qui représente l'erreur actuelle dans la solution sur la grille fine.

$$r^{(k)} = f - Av^{(k)}$$

- **Restriction** : Le résidu est transféré de la grille fine à la grille grossière. Cela permet de représenter l'erreur avec moins de points, ce qui réduit le coût de calcul.

$$r_{2h}^{(k)} = I_h^{2h} r^{(k)}$$

- **Résolution sur la grille grossière** : L'équation d'erreur est résolue directement sur la grille grossière. C'est une étape clé pour corriger les composantes de basse fréquence de l'erreur.

$$A_{2h} e_{2h}^{(k)} = r_{2h}^{(k)}$$

- **Prolongation** : La solution de l'équation d'erreur sur la grille grossière est interpolée (prolongée) vers la grille fine.

$$e_h^{(k)} = I_{2h}^h e_{2h}^{(k)}$$

- **Correction** : La solution sur la grille fine est mise à jour en ajoutant la correction interpolée de la grille grossière.

$$v^{(k+1)} = v^{(k)} + e_h^{(k)}$$

- **Calcul de l'erreur** : L'erreur est calculée en comparant la solution actuelle avec la solution de référence.

$$e^{(k+1)} = u_{ref} - v^{(k+1)}$$

6.2 Résultat

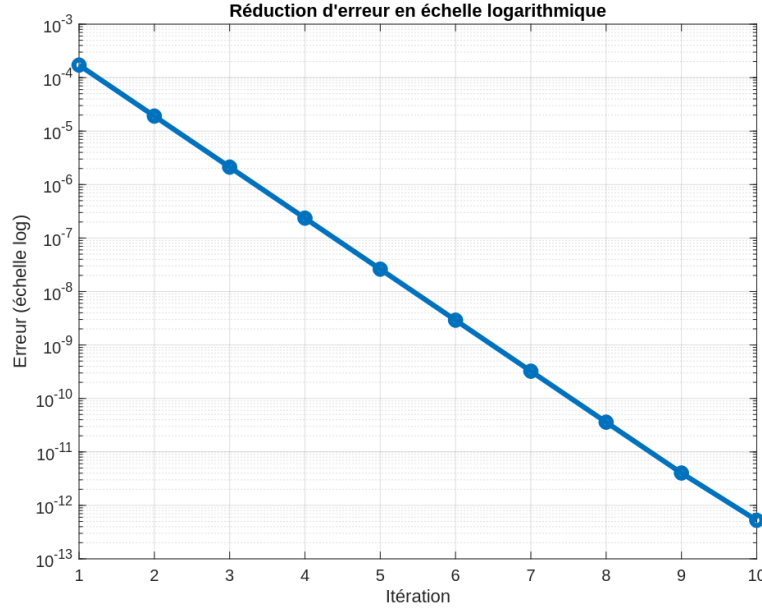


Figure 4: Erreur en fonction du nombre d'itérations (en log)

On observe que l'erreur diminue de manière linéaire au fil des itérations. Cela signifie que la méthode multigrille converge de manière exponentielle, réduisant l'erreur d'un facteur constant (à peu près $\frac{1}{9}$) à chaque itération. Ainsi, la méthode multigrilles est efficace et converge rapidement vers la solution exacte du système linéaire.

7 Adaption du critère d'arrêt

Dans la section précédente, on a mesuré la performance du modèle en utilisant l'erreur, qui s'appuie sur la solution exacte du problème. Toutefois, cette dernière n'est généralement pas disponible. De fait, au lieu de calculer l'erreur, on va calculer le résidu à chaque étape du processus itératif pour constater une réduction de l'erreur.

On va maintenant étudier la convergence de la méthode multigrille en fonction de la taille du maillage N et de la tolérance ϵ , en utilisant le résidu comme critère d'arrêt. L'algorithme s'arrêtera lorsque le résidu deviendra inférieur à ϵ .

Convergence multigrille V-cycle : Résidu en échelle logarithmique

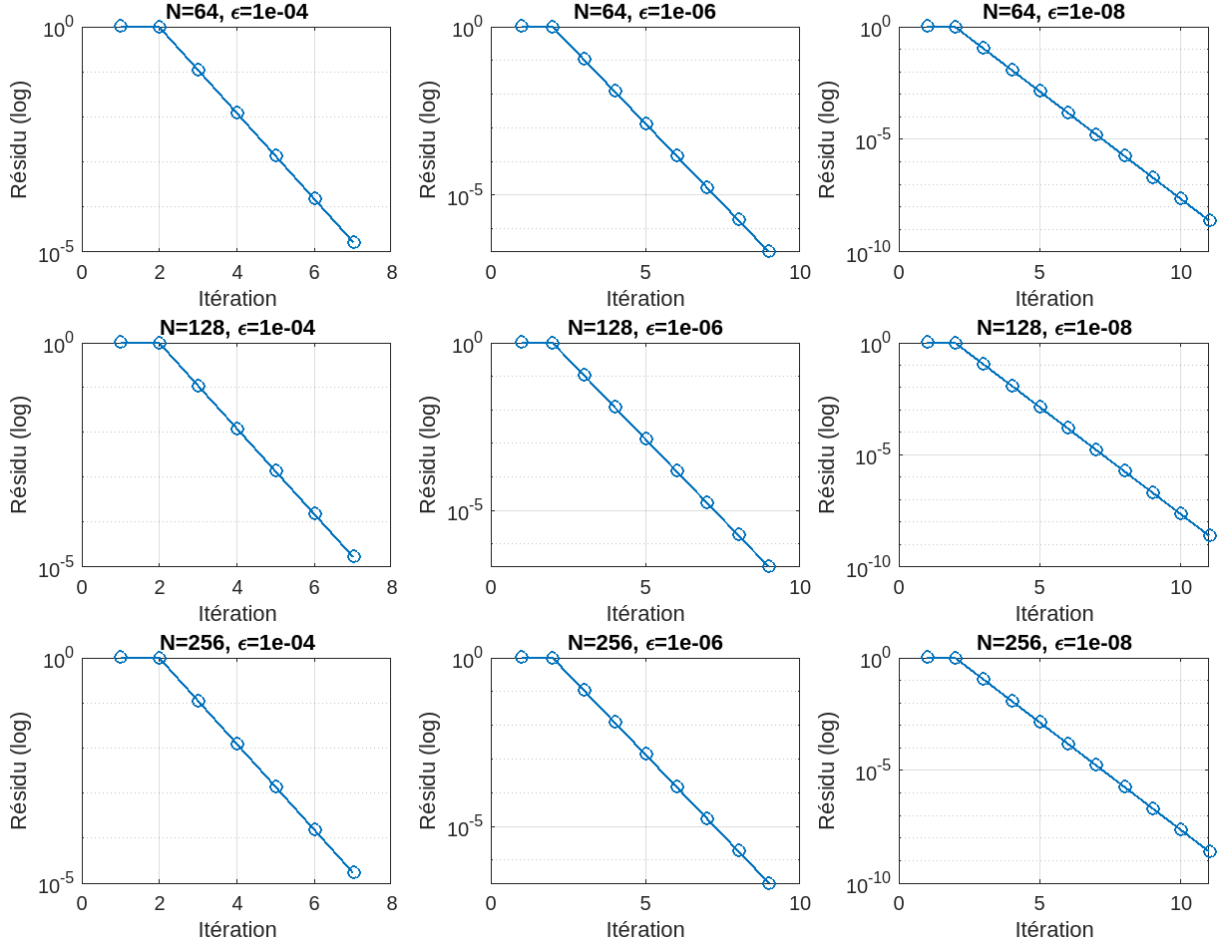


Figure 5: Evolution du résidu au fil des itérations

Chaque courbe montre comment le résidu diminue à chaque itération. La pente descendante indique une réduction exponentielle du résidu, ce qui est témoin de l'efficacité des méthodes multigrilles.

D'une part, on observe que pour une taille de maillage N donnée, les graphiques avec une tolérance plus petite (par exemple, 1×10^{-8}) nécessitent plus d'itérations que ceux avec une tolérance plus grande (par exemple, 1×10^{-4}). En effet, en réduisant ϵ , le résidu final devient plus petit, ce qui donne une solution plus précise. Ainsi, les courbes montrent que la méthode multigrilles peut atteindre des tolérances très faibles en un nombre relativement petit d'itérations, démontrant ainsi son efficacité.

D'autre part, on observe qu'à mesure que N augmente, le nombre d'itérations nécessaires pour atteindre une certaine tolérance reste relativement stable, ce qui confirme que le taux de convergence est relativement indépendant de la taille du maillage.

Pour conclure, la méthode multigrille montre une convergence rapide pour toutes les combinaisons de N et ϵ . La réduction rapide du résidu indique que la méthode est capable de corriger efficacement les erreurs à chaque itération, en utilisant les informations des grilles grossières pour améliorer la solution sur les grilles fines.