ECOLE
POLYTECHNIQUE
DE BRUXELLES

Projet de Méthodes Numériques

MATH-H401 - Numerical Methodes

Auteurs :

Dietrich colin

Professeurs

Napov Artem

Notay Yvan

année académique 2021 - 2022

Table des matières

1 Introduction **2**

2 Structure du programme **2**

3 Discrétisation **3**

3.1 Définition du problème 3

3.2 Discrétisation du problème 3

4 Méthode Deux-grilles **6**

4.1 Principe de fonctionnement 6

4.2 Résolution numérique pour le domaine P8 7

4.2.1 Convergence 7

4.2.2 Visualisation du résidu 8

5 Méthode Multi-grilles **13**

5.1 Principe de fonctionnement 13

5.2 Résolution numérique pour le domaine P8 15

6 Exploitation du solveur Multi-grilles **17**

6.1 Convergence et stabilité directe 17

6.2 Facteur de convergence asymptotique 17

6.3 Facteur de relaxation 18

6.3.1 Préconditionneur Multi-grilles 18

6.3.2 Application du facteur de relaxation 18

7 Préconditionnement Multi-grilles **21**

A Résultats de la méthode deux-grilles pour Cercle **23**

B Résultats de la méthode deux-grilles pour le domaine P15 **25**

1 Introduction

Ce projet s'inscrit dans le cadre du cours **MATH-H401**. Il a pour objectif la résolution numérique de l'équation de Poisson à deux dimensions. Pour ce faire, un programme a été développé en langage C. Celui-ci permet de définir une forme de membrane quelconque à partir de un ou plusieurs rectangles et de discrétiser le domaine correspondant.
Ensuite, une méthode Multi-grilles a été mise en place afin de résoudre le problème par itérations successives.
De plus, une étude plus approfondie de la méthode Multi-grilles est réalisée afin de vérifier la stabilité et la convergence de l'algorithme, mais également de l'optimiser en introduisant un paramètre de relaxation optimal.
Enfin, un problème aux valeurs propres est résolu avec le solveur PRIMME et l'amélioration apportée par l'application d'un préconditionneur Multi-grilles au solveur est étudiée.

2 Structure du programme

Le programme est constitué de 20 fichiers dont chacun contient des fonctions destinées à un rôle spécifique.
En particulier le fichier principal *main.c* contient 4 options :

- La **1ère** contient la méthode deux-grilles initialement créée pour la question 2 (bien que les options 2 et 3 soient également capable d'implémenter la méthode deux-grilles).
- La **2nd** option permet d'utiliser un solveur multi-grilles avec nombre de niveaux variables en cycle V ou W.
- La **3ème** option met en place un préconditionneur multi-grille (résout $Ax = b - Au$ puis à la fin de chaque itération ajoute la correction à la solution $u = u + \tau x$) afin d'implémenter le facteur de relaxation, de calculer les valeurs propres min/max, le facteur de convergence asymptotique, le τ optimal et de vérifier la stabilité directe de l'algorithme.
- La **4ème** option utilise le solveur PRIMME pour résoudre un problème aux valeurs propres et permet d'implémenter le préconditionneur multi-grille pour accélérer la convergence du solveur.

3 Discrétisation

3.1 Définition du problème

Comme dit précédemment, ce projet consiste en la résolution numérique de l'équation de Poisson à deux dimensions. Plus précisément, pour une membrane $\Omega \subset \mathbb{R}^2$ donnée, on cherche à résoudre le problème aux limites suivant :

$$\begin{cases} -\Delta u = 0 & \text{dans } \ \Omega \subset \mathbb{R}^2 \\ u = e^{\sqrt{x^2+y^2}} & \text{sur } \ \partial\Omega \end{cases} \tag{1}$$

où $\partial\Omega$ est le bord du domaine et u est donné par les conditions aux limites de Dirichlet sur $\partial\Omega$.

3.2 Discrétisation du problème

Les membranes considérées dans ce projet peuvent être obtenues à partir d'une membrane carrée de dimension $[L\text{x}L]$ et en y supprimant une ou plusieurs parties rectangulaires. Ainsi, pour autant que la discrétisation choisie soit suffisamment petite, la membrane peut être définie avec une forme quelconque si on lui enlève les zones rectangulaires nécessaires.

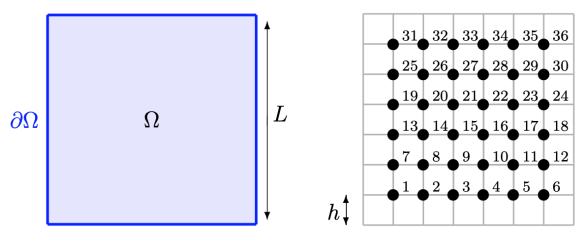


FIGURE 1 – Schéma du domaine discrétiser

Le domaine est discrétisé selon un pas de discrétisation $h = \frac{L}{m-1}$ où L est la longueur d'un côté de la membrane carrée et m est le nombre de points alignés selon une direction de la grille. Ainsi, chacun des points de la grille est défini de la manière suivante :

$$(x_i, y_j) = (ih, jh) \qquad \text{avec } \ i, j = 1, ..., m-2 \tag{2}$$

L'approximation de la solution u est calculée en chacun des points appartenant à l'intérieur du domaine Ω selon la formule des différences finies :

$$\frac{4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1}}{h^2} = 0 + \mathcal{O}(h^2) \tag{3}$$

où les bords du domaine ne sont pas pris en compte car u y est déjà connu par les conditions aux

bords de Dirichlet.

En regroupant toutes les équations on obtient le système linéaire suivant :

$$Au = b \tag{4}$$

où b est un vecteur nul auquel on a ajouté les conditions aux bords de Dirichlet.

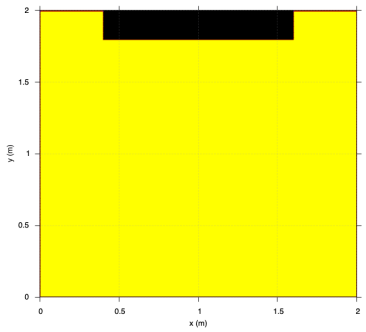


FIGURE 2 – Domaine P8

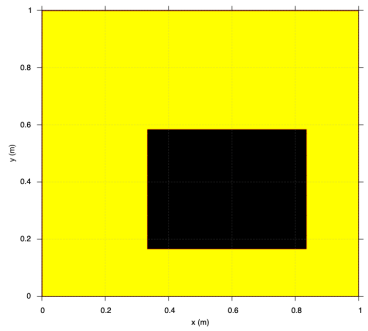


FIGURE 3 – Domaine P15

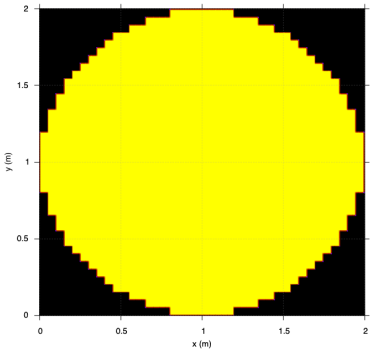


FIGURE 4 – Cercle

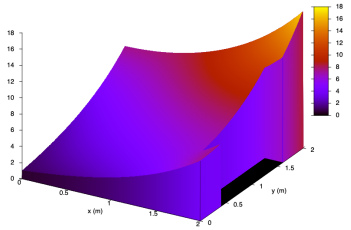


FIGURE 5 – Solution P8

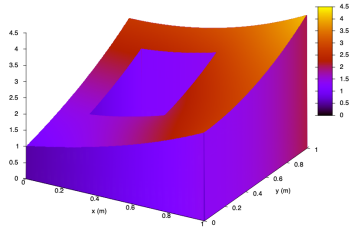


FIGURE 6 – Solution P15

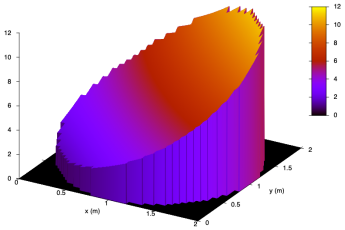


FIGURE 7 – Solution cercle

Le programme développé fonctionne pour tous les domaines proposés et fonctionne également avec plusieurs trous rectangulaires (on peut définir autant de trous que nécessaire). Afin de démontrer la flexibilité du programme, trois exemples de membrane seront traités tout au long du projet (les parties en noir ne font pas parti du domaine) :

— Domaine P8 (figure 2)

  Membrane du projet n°8. le domaine correspondant est $\Omega = [0,2]^2 \setminus [\frac{2}{5}, \frac{8}{5}]$ x $[\frac{9}{5}, 2]$

— Domaine P15 (figure 3)

  Membrane du projet n°15. le domaine correspondant est $\Omega = [0,1]^2 \setminus [\frac{1}{3}, \frac{5}{6}]$ x $[\frac{1}{6}, \frac{7}{12}]$

— Domaine circulaire (figure 4)

  Cette membrane a été réalisée en retirant 48 parties rectangulaires à une membrane carrée de dimension [2x2]. Il serait possible de définir une membrane circulaire encore plus précise pour autant que suffisamment de parties rectangulaires soient retirées à la membranes carrée et que la discrétisassion soit suffisamment petite. Cependant, il a été décidé de garder cette approximation assez grossière afin de minimiser le temps d'exécution du programme.

4 Méthode Deux-grilles

Dans cette section, le programme utilise comme méthode de pré- et post-lissage la méthode itérative de Gauss-Seidel symétrique. De plus les résultats seront présentés pour une discrétisation de m = 641.

4.1 Principe de fonctionnement

La méthode Deux-grilles est une méthode de résolution numérique itérative alternant des itérations de lissage afin d'atténuer les erreurs hautes fréquences, avec des corrections faites sur une grille plus grossière (en général pour un pas de discrétisation valant 2 fois celui de la grille initiale) afin d'atténuer les erreurs basses fréquences. L'algorithme suit les étapes suivantes :

Algorithm 1 méthode Deux-grilles

for $i \leftarrow 0$ to N **do**
  Appliquer ν_1 pré-lissages sur $Au = b \quad \rightarrow \ \mathrm{u}_1$
  Réduire le résidu $r = R(b - Au_1)$
    Résoudre sur la grille grossière : $c = A^{-1}r$
  Prolonger la correction : $u_1' = u_1 + Pc$
  Appliquer ν_2 post-lissages sur $Au = b \quad \rightarrow \ \mathrm{u}_1''$
end for

La restriction est réalisée par full-weighting :

$$\begin{aligned} \tilde{r}_{2i+1,2j+1} &= \frac{1}{4}\, r_{2i+1,2j+1} \\ &+ \frac{1}{8}\left(r_{2i,2j+1} + r_{2i+2,2j+1} + r_{2i+1,2j} + r_{2i+1,2j+2}\right) \\ &+ \frac{1}{16}\left(r_{2i,2j} + r_{2i+2,2j} + r_{2i,2j+2} + r_{2i+2,2j+2}\right) \end{aligned} \tag{5}$$

avec r le résidu sur la grille initiale et $\tilde{r}$ le résidu sur la grille grossière. Cette méthode consiste donc à calculer le résidu équivalent sur la grille grossière en faisant une interpolation des résidus l'entourant sur la grille fine (si un de ces points est sur le bord du domaine ou d'un trou, le résidu y est simplement nul puisque la solution u y est parfaitement connue par les conditions aux bords de Dirichlet).

La prolongation est réalisée selon les formules suivantes :

$$\begin{cases} r_{2i+1,2j+1} &= \tilde{r}_{2i+1,2j+1} \\ r_{2i+1,2j} &= \dfrac{1}{2}\ (\tilde{r}_{2i+1,2j-1} + \tilde{r}_{2i+1,2j+1}) \\ r_{2i,2j+1} &= \dfrac{1}{2}\ (\tilde{r}_{2i-1,2j+1} + \tilde{r}_{2i+1,2j+1}) \\ r_{2i,2j} &= \dfrac{1}{4}\ (\tilde{r}_{2i-1,2j-1} + \tilde{r}_{2i-1,2j+1} + \tilde{r}_{2i+1,2j+1} + \tilde{r}_{2i+1,2j-1}) \end{cases} \tag{6}$$

4.2 Résolution numérique pour le domaine P8

nb : Les mêmes résolutions numériques ont également été réalisées pour le domaine P15 et le cercle. Par soucis de légèreté, les résultats sont présentés en annexe A pour le cercle et B pour le domaine P15.

4.2.1 Convergence

La figure 8 représente la norme du résidu :

$$||r_m|| = \frac{||b - Au||}{||b||} \tag{7}$$

en fonction du nombre d'itérations internes n (itérations de Gauss-Seidel symétrique + résolution sur grille grossière). Elle a été réalisée pour une discrétisation de $m = 641$ et consiste en 5 itérations de l'algorithme Deux-grilles avec un nombre d'itérations pour le pré- et post-lissage $\nu_1 = \nu_2 = 20$. Cela permet de visualiser clairement l'impact de la résolution sur grille grossière par rapport aux itérations de pré- et post-lissage. En effet, on observe d'abord que le résidu commence à 1 (l'initialisation de u étant le vecteur nul) et diminue grâce au pré-lissage. Cependant au-delà de quelques itérations de Gauss-Seidel symétrique, celui-ci converge très lentement. La correction apportée par la résolution sur grille grossière permet ensuite d'accélérer considérablement la convergence (point d'inflexion sur le graphe) qui est alors suivie par le post-lissage.

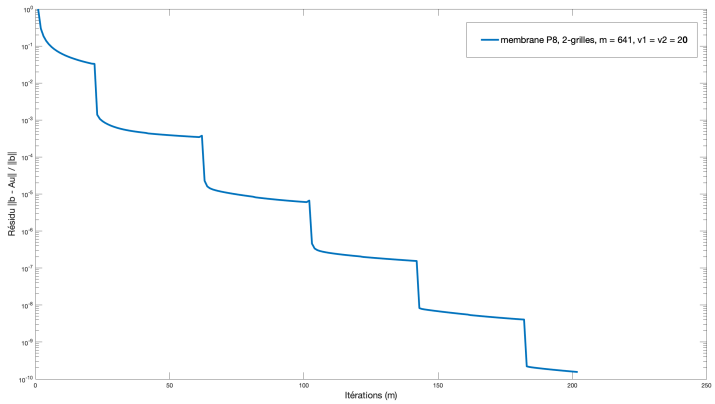


FIGURE 8 – Visualisation de la correction apportée par la coarse grid à la méthode itérative de Gauss-Seidel symétrisée $||r_m|| - vs - m$

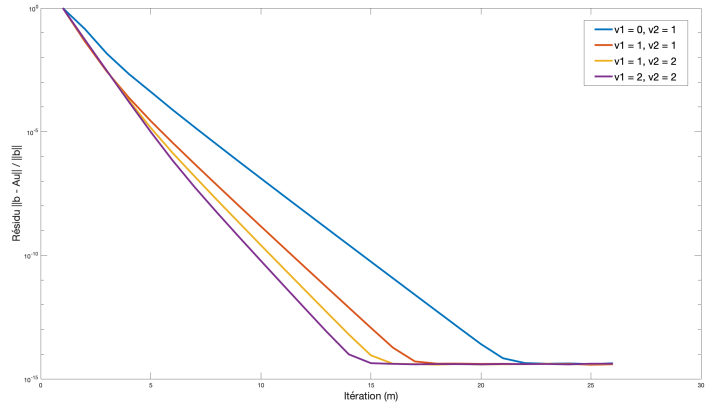


FIGURE 9 – Convergence de la méthode Deux-grilles en fonction du nombre d'itération de pre- et post-lissage (m = 641) $||r_m|| - vs - m$

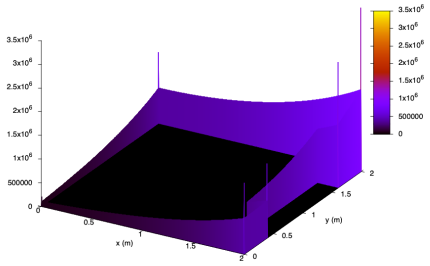
La figure 9 montre la convergence de l'algorithme Deux-grilles pour différentes valeurs d'itérations de pré-/post-lissage. On peut voir que la convergence optimale est obtenue pour $\nu_1 = \nu_2 = 2$. Ces valeurs seront toujours les mêmes dans la suite du rapport. Ainsi, pour $m = 641$ la norme du résidu converge après environ 15 itérations pour atteindre une valeur de l'ordre de 10^{-15}.

4.2.2 Visualisation du résidu

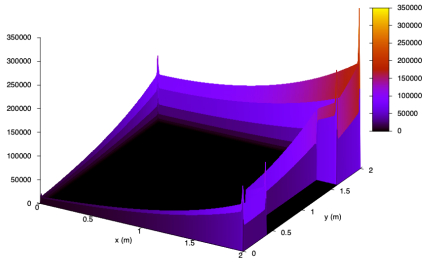
Les figures 10 et 11 représentent l'évolution du résidu à la première et deuxième itération (pour $m = 641$). On observe bien que le résidu à la 1ère itération avant le pré-lissage (fig. 10a) vaut le

terme indépendant b, soit les conditions aux bords de Dirichlet (car la valeur initiale de u est le vecteur nul). Ensuite, le résidu après pré-lissage (fig. 10b) montre que les hautes fréquences sont atténuées avec le pic de résidu maximum passant d'environ $3,5$ x 10^6 à 350000. Ensuite, après la correction de grille grossière (fig 10c), on peut voir que les pics (hautes fréquences) ne sont que légèrement atténués, alors que les courbes présentant des variations lentes sont atténuées. Enfin, le post-lissage (fig 10d) vient de nouveaux atténuer les basses fréquences pour fournir un résidu dont la valeur maximale est passée de $3,5$ x 10^6 à 18000 après une seule itération de l'algorithme Deux-grilles.

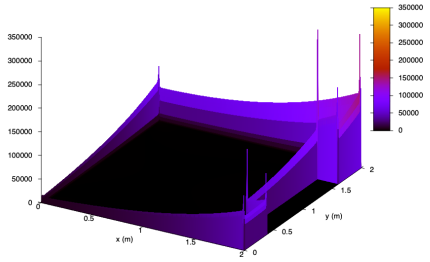
De plus, il est intéressant d'observer le passage du résidu après pré-lissage (fig. 11a) et après correction de grille grossière (fig. 11b) pour la deuxième itération. On pourrait croire en regardant l'échelle selon l'axe z que la correction a augmenté la valeur du résidu. Cependant, si on limite la visualisation du résidu après grille grossière en dessous de 9000 (fig. 12), on comprend que la correction a bien atténué les basses fréquences et que la valeur max du résidu valant 30000 est due à un pique présentant une très haute fréquence. Ainsi, l'erreur sur ce pic augmente sûrement à cause de l'erreur induite par la restriction et la prolongation. Finalement, le résidu après post-lissage atteint une valeur beaucoup plus petite faisant passer le résidu maximum avant pré-lissage de 18000 à 1200, ce qui est un résultat tout à fait satisfaisant.



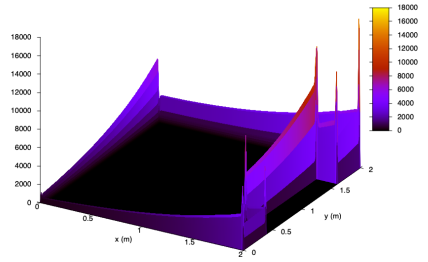
(a) résidu avant pré-lissage



(b) résidu après pré-lissage

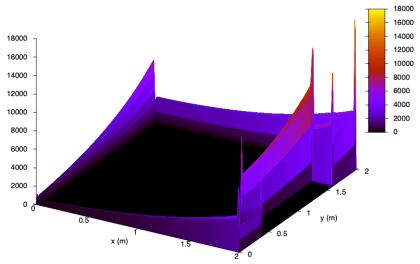


(c) résidu après coarse-grid

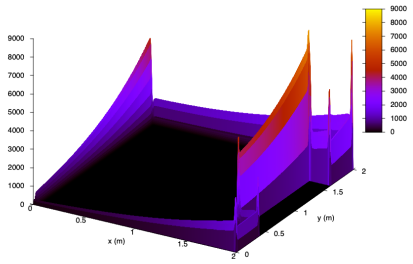


(d) résidu après post-lissage

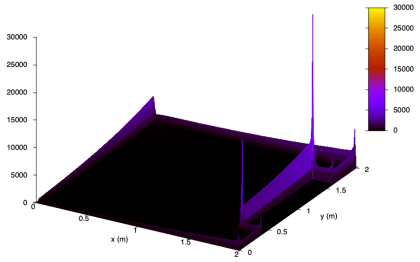
FIGURE 10 – Évolution du résidu de la méthode Deux-grilles pour la 1ère itération (m = 641)



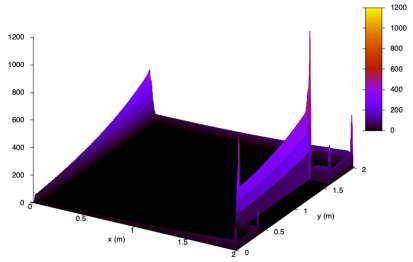
(a) résidu avant pré-lissage



(b) résidu après pré-lissage

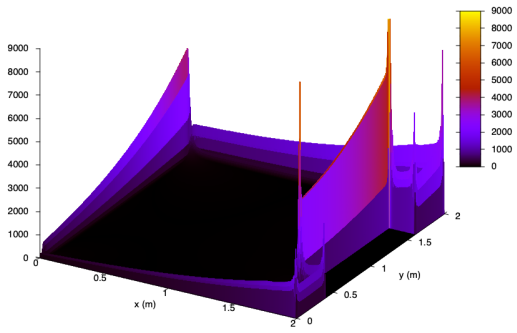


(c) résidu après coarse-grid



(d) résidu après post-lissage

FIGURE 11 – Évolution du résidu de la méthode Deux-grilles pour la 2ème itération (m = 641)



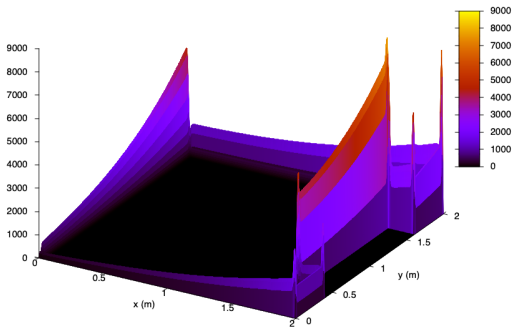


FIGURE 12 – Comparaison du résidu après coarse-grid limité à 9000 selon z (gauche) et du résidu après pré-lissage (droite) pour la 2ème itération

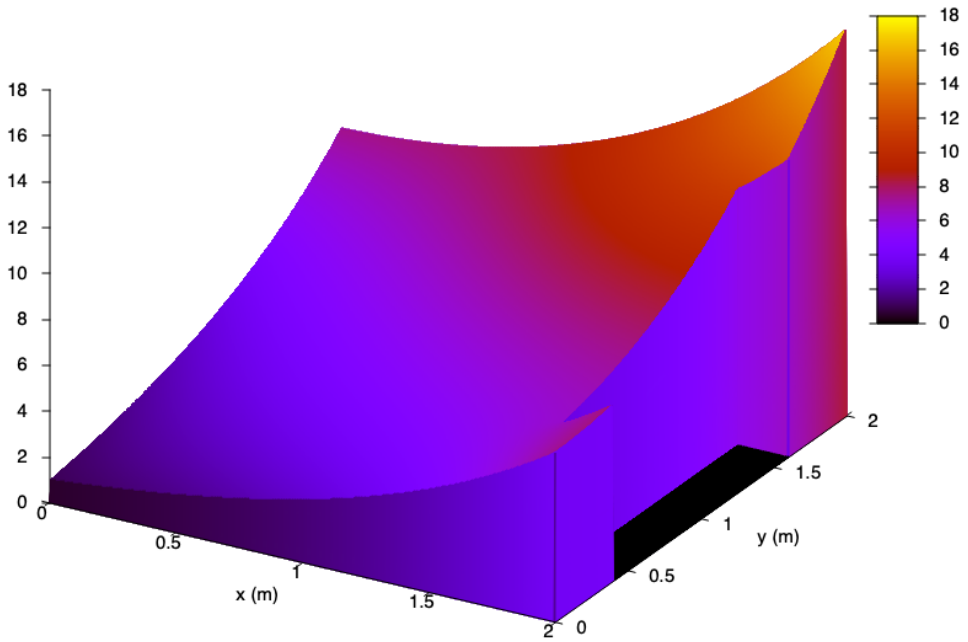


FIGURE 13 – Solution finale fournie par l'algorithme Deux-grilles pour le domaine P8 (m = 641)

5 Méthode Multi-grilles

5.1 Principe de fonctionnement

Dans l'algorithme Deux-grilles, toutes les étapes de résolution sont peu coûteuses excepté la résolution avec le solveur direct UMFPACK sur la grille grossière. La méthode Multi-grilles consiste donc à continuer les itérations de pré-/post-lissage sur des grilles de plus en plus grossières (à chaque nouvelle grille grossière $h_{new} = 2h_{old}$) jusqu'à arriver à une grille avec un pas de discrétisation suffisamment large pour résoudre le problème $u_c = A_c^{-1} r_c$ avec le solveur direct. Cela réduit le coût d'exécution mais les corrections et les prolongations successives ralentissent la convergence. De plus, la correction apportée par la résolution directe sur grille grossière sera d'autant moins efficace que le pas de discrétisation sera large. En effet, on observe bien sur la figure 14 que la convergence de l'algorithme Multi-grilles converge plus lentement avec le nombre de niveaux qui augmentent.

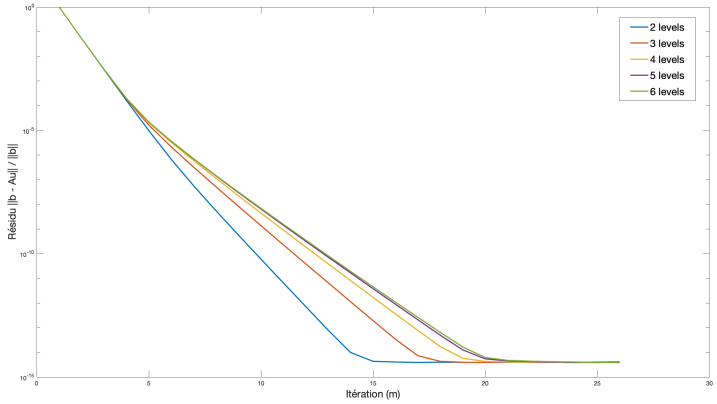


FIGURE 14 – Convergence de l'algorithme Multi-grilles (cycle V) selon le nombre de niveaux (m = 641) $||r_m|| - vs - m$

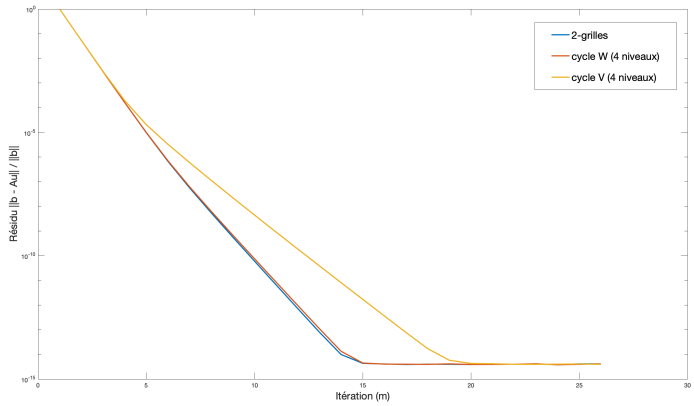


FIGURE 15 – comparaison de l'algorithme Deux-grilles, multigrid en (m = 641) $||r_m|| - vs - m$

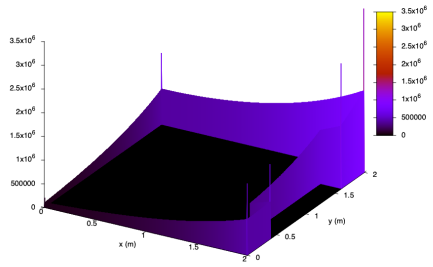
Le programme permet d'exécuter l'algorithme Multi-grilles en cycle V ou W selon un nombre de niveaux variables. On observe bien sur la figure 15 que l'algorithme en cycle W se rapproche plus de l'allure de la méthode Deux-grilles tandis que le cycle V converge plus lentement. Dans la suite de cette section, on utilise un algorithme Multi-grilles en cycle V avec 4 niveaux. Un exemple de cet algorithme est présenté en (15) .

Algorithm 2 méthode Multi-grilles

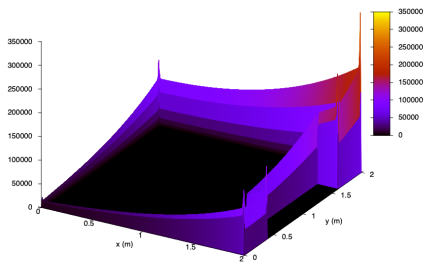
for $i \leftarrow 0$ to N **do**
  Appliquer ν_1 pré-lissages sur $A_1 u = b \quad \rightarrow$ u_1
  Réduire le résidu $r_2 = R_1(b - A_1 u_1)$
    Appliquer ν_1 pré-lissages sur $A_2 c = r_2 \quad \rightarrow$ c_2
    Réduire le résidu $r_3 = R_2(r_2 - A_2 c_2)$
      Appliquer ν_1 pré-lissages sur $A_3 c = r_3 \quad \rightarrow$ c_3
      Réduire le résidu $r_4 = R_3(r_3 - A_3 c_3)$
        Résoudre sur la grille grossière : $c_4 = A_4^{-1} r_4$
      Prolonger la correction : $\tilde{c_3} = c_3 + P_3 c_4$
      Appliquer ν_2 post-lissages sur $A_3 c = r_3 \quad \rightarrow$ c'_3
    Prolonger la correction : $\tilde{c_2} = c_2 + P_2 c'_3$
    Appliquer ν_2 post-lissages sur $A_2 c = r_2 \quad \rightarrow$ c'_2
  Prolonger la correction : $\tilde{u_1} = u_1 + P_1 c'_2$
  Appliquer ν_2 post-lissages sur $A_1 u = b \quad \rightarrow$ u'_1
end for

5.2 Résolution numérique pour le domaine P8

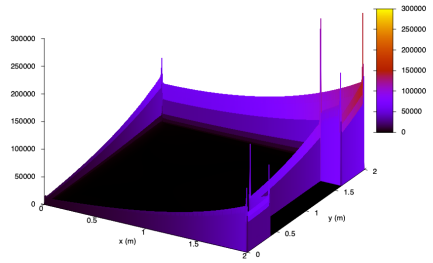
Les figures 16 et 17 représentent respectivement les résidus à la première et à la deuxième itération. Les mêmes conclusions que pour la méthode Deux-grilles peuvent être faites. On remarque d'ailleurs que la première itération est très similaire à celle de la Deux-grilles mais que dès la seconde itération la méthode Multi-grilles commence à converger plus lentement.



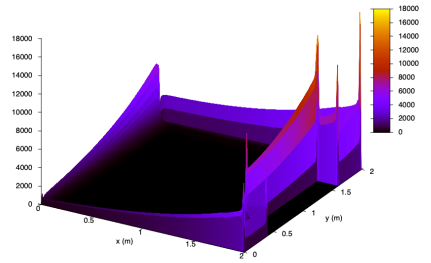
(a) résidu avant pré-lissage



(b) résidu après pré-lissage

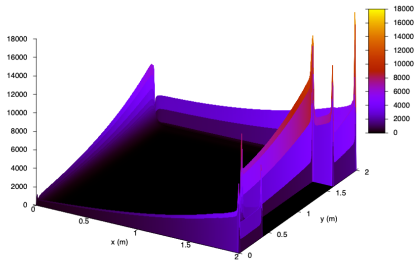


(c) résidu après coarse-grid

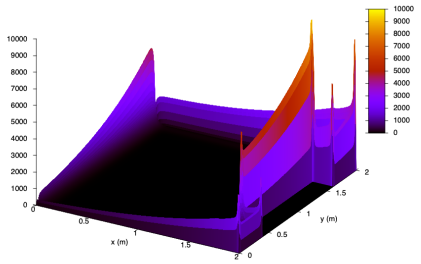


(d) résidu après post-lissage

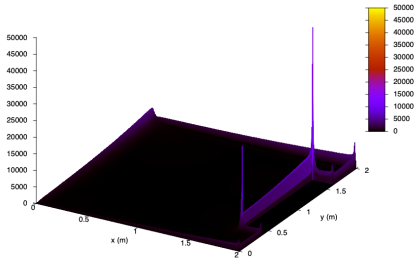
FIGURE 16 – Évolution du résidu de la méthode Multi-grilles pour la 1ère itération (m = 641)



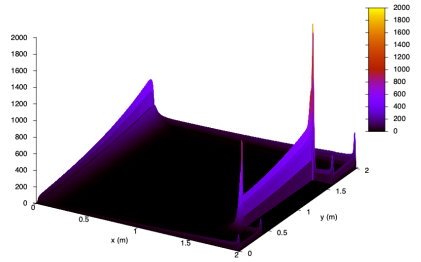
(a) résidu avant pré-lissage



(b) résidu après pré-lissage



(c) résidu après coarse-grid



(d) résidu après post-lissage

FIGURE 17 – Évolution du résidu de la méthode Multi-grilles pour la 2ème itération (m = 641)

6 Exploitation du solveur Multi-grilles

Afin de caractériser plus rigoureusement l'algorithme et d'optimiser sa convergence, celui-ci va être étudié pour deux problèmes différents :

— $h_1 = 0,0031$ ($m_1 = 641$), algorithme Multi-grilles en cycle V avec 4 niveaux.

— $h_2 = 3,9063e$ x 10^{-4} ($m_2 = 5121$), algorithme Multi-grilles en cycle V avec 9 niveaux.

6.1 Convergence et stabilité directe

L'algorithme possède la stabilité directe si il vérifie le critère suivant :

$$||r|| = \mathcal{O}(u_{mach})\ ||A||\ ||u|| \tag{8}$$

avec $||r|| = ||b - Au||$ la norme du résidu, u l'approximation fournie par la méthode Multi-grilles de la solution exacte û, A la matrice du système (4), et $u_{mach} = 1,1$ x 10^{-16} la précision machine. Ce critère nécessite une estimation de la norme de la matrice A du système. Celle-ci étant défini comme $||A|| = \lambda_{max}$ on peut utiliser le théorème de Gershgorin afin d'obtenir une borne supérieure sur cette valeur propre :

$$||A|| = \lambda_{max} \approx \frac{8D_M}{h^2} + c_M \tag{9}$$

Dans le problème considéré en (1), on peut observer que $D_M = C_M = 0$. On obtient ainsi une borne supérieure sur la norme de A :

$$\begin{cases} ||A||_1 = 8,3247 \text{ x } 10^5 & \text{pour } h_1 \\ ||A||_2 = 5,2427 \text{ x } 10^7 & \text{pour } h_2 \end{cases} \tag{10}$$

On peut donc vérifier que l'algorithme possède bien la stabilité directe :

$$\begin{cases} \dfrac{||r||_1}{||A||_1\ ||u_1||} = 1.620022 \text{ x } 10^{-25} & < \quad \mathcal{O}(u_{mach}) \\ \dfrac{||r||_2}{||A||_2\ ||u_2||} = 5,013808e \text{ x } 10^{-17} & < \quad \mathcal{O}(u_{mach}) \end{cases} \tag{11}$$

6.2 Facteur de convergence asymptotique

Le facteur de convergence asymptotique est le rayon spectral de la matrice d'itération $T = I - B^{-1}A$, où B est le préconditionneur correspondant à l'algorithme Multi-grilles.

Ainsi une méthode d'itération stationnaire est convergente si et seulement si $\rho(T)$ vérifie :

$$\rho(T) = \max_{\nu \in \sigma(T)} |\nu| \ < \ 1 \tag{12}$$

avec $\sigma(T)$ le spectre de l'opérateur T. Celui-ci peut être calculé à partir de l'expression :

$$\ln[\frac{||e_m||}{||e_0||}] = \ln[C_m] - m \ . \ \ln[\frac{1}{\rho(T)}] \tag{13}$$

Ainsi, en considérant les deux dernières itérations n et $n-1$ avant la convergence, on peut calculer le facteur ρ de la manière suivante :

$$\rho = \frac{r_n}{r_{n-1}} \tag{14}$$

Ce qui donne pour h_1 et h_2 :

$$\begin{cases} \rho_1 = 0,1476 & \text{pour } h_1 \\ \rho_2 = 0,201 & \text{pour } h_2 \end{cases} \tag{15}$$

6.3 Facteur de relaxation

6.3.1 Préconditionneur Multi-grilles

Afin d'implémenter un facteur de relaxation τ, le solveur Multi-grilles a été modifié en préconditionneur Multi-grilles tel que le vecteur solution soit calculé à chaque itération par :

$$u_{m+1} = u_m + \tau B^{-1}(b - Au_m) \tag{16}$$

avec B le préconditionneur Multi-grille. Ainsi, l'algorithme résout d'abord $Ax = b - Au$ puis à la fin de chaque itération ajoute la correction à la solution $u = u + \tau x$.

6.3.2 Application du facteur de relaxation

Le facteur de convergence asymptotique défini en (12) peut également être exprimé (en prenant en compte le facteur de relaxation) par l'expression suivante :

$$\rho(T) = \max_{\lambda \in \sigma(B^{-1}A)} |1 - \tau\lambda| \tag{17}$$

En considérant $\lambda_{max} \approx 1$ et un facteur de relaxation unitaire $\tau = 1$, on peut calculer une première approximation de la valeur propre minimale :

$$\lambda_{min} = 1 - \rho \tag{18}$$

Ce qui donne pour h_1 et h_2 :

$$\begin{cases} \lambda_{min1} = 0,8524 & \text{pour } h_1 \\ \lambda_{min2} = 0,7990 & \text{pour } h_2 \end{cases} \tag{19}$$

Cette valeur ayant été calculée, on peut alors déterminer le facteur de relaxation optimal :

$$\tau = \frac{2}{\lambda_{max} + \lambda_{min}} \tag{20}$$

Celui-ci vaut pour h_1 et h_2 :

$$\begin{cases} \tau_1 = 1.0797 & \text{pour } h_1 \\ \tau_2 = 1,1117 & \text{pour } h_2 \end{cases} \tag{21}$$

En exécutant à nouveau l'algorithme Multi-grilles, mais cette fois ci avec le facteur de relaxation optimal calculé précédemment, on obtient le facteur de convergence asymptotique de l'algorithme accéléré qui peut être comparé à l'estimation théorique :

$$\rho_t = 1 - \tau\lambda_{min} = \tau - 1 \tag{22}$$

Les valeurs pour h_1 et h_2 sont les suivantes :

$$\begin{cases} \rho_{a1} = 0,2051 & > \ \rho_{t1} = 0,1823 & \text{pour } h_1 \\ \rho_{a2} = 0,3034 & > \ \rho_{t2} = 0,1823 & \text{pour } h_2 \end{cases} \tag{23}$$

Ainsi, ρ_a étant supérieur à ρ_t à la fois pour h_1 et h_2, cela signifie que $\lambda_{max} \approx 1$ n'est pas une bonne approximation. En effet, dans ce cas ci λ_{max} devrait être supérieur à 1 et peut être corrigé selon la formule suivante :

$$\rho_a = \tau\lambda_{max} - 1 \tag{24}$$

ce qui permet de déterminer une meilleur approximation de la valeur propre maximale et donc du facteur de relaxation. On a donc :

$$\begin{cases} \lambda_{max1} = 1,1161 & \rightarrow \tau_{new1} = 1,016 & \text{pour } h_1 \\ \lambda_{max2} = 1,1724 & \rightarrow \tau_{new2} = 1,0145 & \text{pour } h_2 \end{cases} \tag{25}$$

Les figures 18 et 19 représentent la convergence du résidu dans le cas h_1 et h_2 pour les différents facteurs de relaxation. On peut voir que les facteurs de relaxation optimaux n'accélèrent pas la convergence du résidu. Les courbes accélérées passent légèrement en-dessous de la courbe de référence (pour $\tau = 1$) mais repassent rapidement au-dessus pour finalement converger plus lentement. Cela peut possiblement s'expliquer soit par un manque de précision dans les calculs du facteur de relaxation, soit par le fait que l'hypothèse $\lambda_{max} \approx 1$ est mauvaise et la correction de cette valeur

n'a pas suffit à améliorer le calcul de τ.

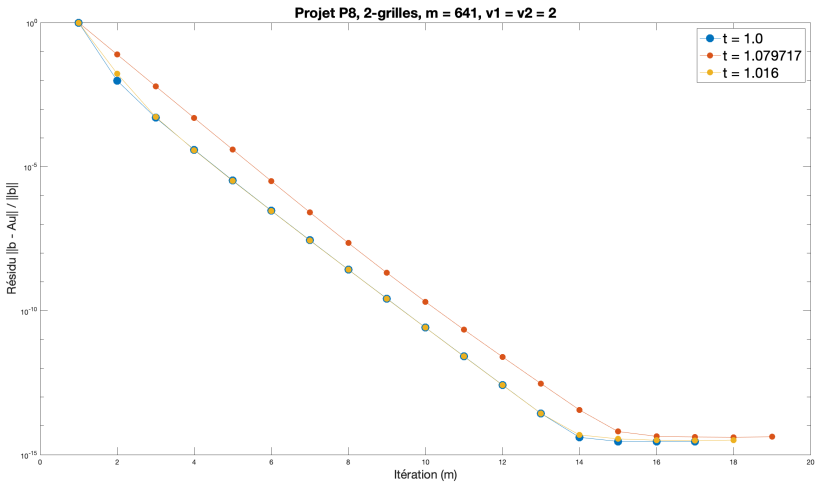


FIGURE 18 – Influence du facteur de relaxation ($h_1 = 0,0031$) $||r_m|| - vs - m$

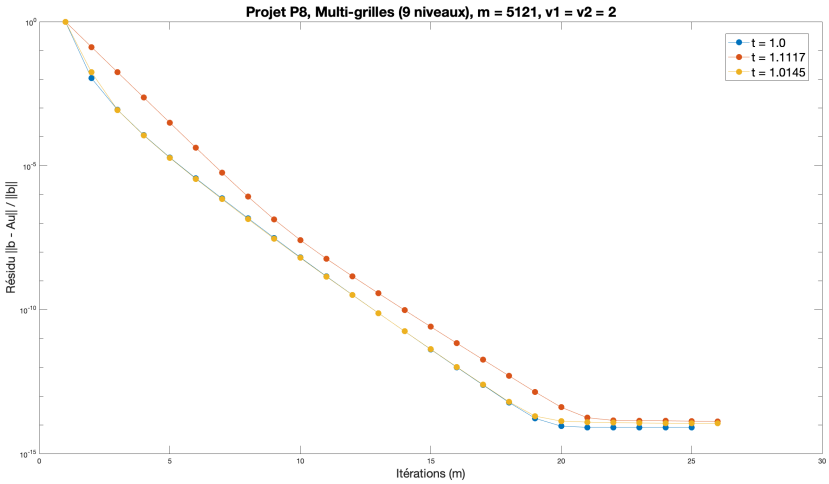


FIGURE 19 – Influence du facteur de relaxation ($h_2 = 3,9063e$ x 10^{-4}) $||r_m|| - vs - m$

7 Préconditionnement Multi-grilles

Le solveur PRIMME, est une bibliothèque permettant de résoudre des problèmes aux valeurs propres. Il est spécialement optimisé pour Les problèmes hermitiens réels symétriques et complexes standards $Ax = \lambda x$ et généralisés $Ax = \lambda Bx$. Il peut déterminer les valeurs propres les plus petites, intérieures, ou les plus grandes ainsi que les vecteurs propres correspondant. De plus, il est possible d'utiliser un préconditionnement afin d'accélérer la convergence du solveur.
On utilise ce solveur afin de déterminer la valeur porpre la plus petite et le vecteur propre correspondant du problème aux valeurs propres suivant :

$$\begin{cases} -\Delta u = \omega^2 u & \text{dans } \Omega \subset \Omega_s = [0, L]^2 \\ u = 0 & \text{sur } \partial\Omega \end{cases} \tag{26}$$

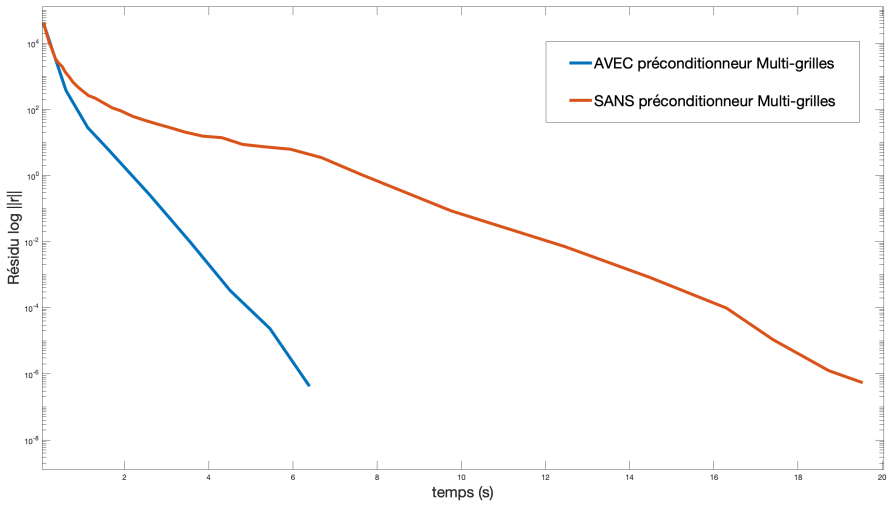


FIGURE 20 – Convergence du solveur PRIMME avec préconditionnement Multi-grilles et sans préconditionnement $||r_m|| - vs - t$

Le problème a été discrétisé selon $m = 641$, soit un pas de discrétisation $h = 0.0031[m]$. Le solveur a d'abord été utilisé sans précondtionnement. Celui-ci a convergé après 39 itérations vers une valeur propre de $5,49624$ correspondant à un temps d'exécution total de $19,5368[s]$ pour un résidu final $||r|| = 5,419386$ x 10^{-7}. Ensuite, l'algorithme Multi-grilles développé précédemment a été fournit à PRIMME afin d'être utilisé comme préconditionneur (le nombre de niveux de l'algorithme Multi-grilles fourni ici a été fixé à 4 mais peut être modifié facilement). Le solveur a convergé vers

une valeur propre de $5,496247$ après 9 itérations correspondant à un temps d'exécution total de $6,39226[s]$ pour un résidu final $||r|| = 4,212514$ x 10^{-7}.

On observe donc bien que le préconditionneur Multi-grilles accélère la convergence du solveur PRIMME. La figure 21 représente la norme du résidu en fonction du temps d'exécution en secondes en échelle semi-logarithmique.

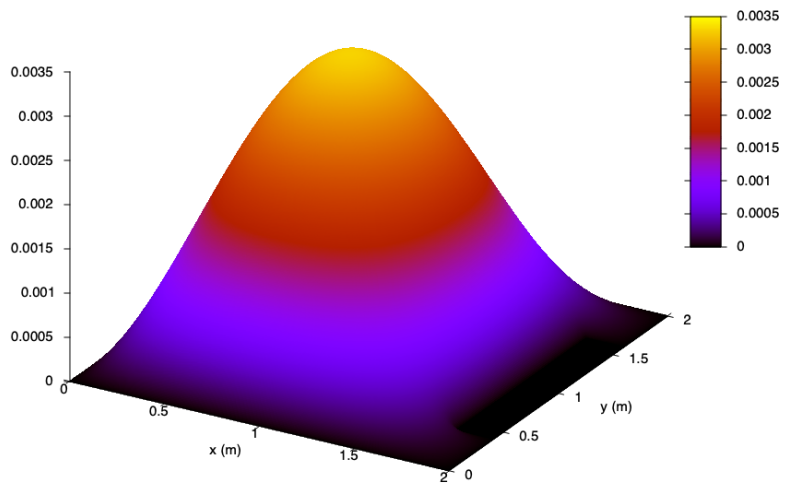


FIGURE 21 – vecteur propre associé à la valeur propre 5,496247 du domaine P8

A Résultats de la méthode deux-grilles pour Cercle

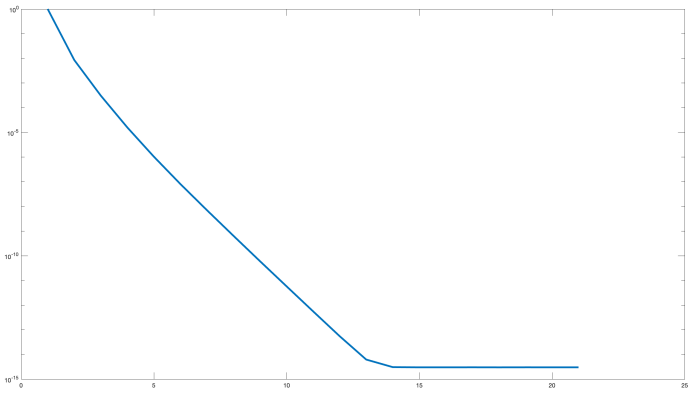
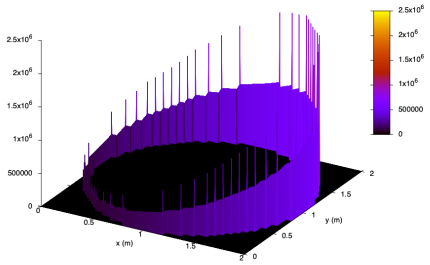
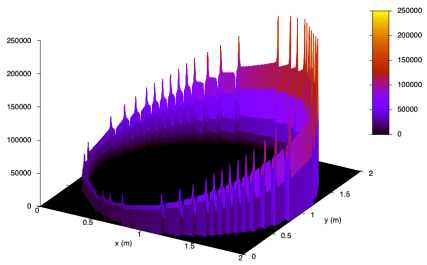


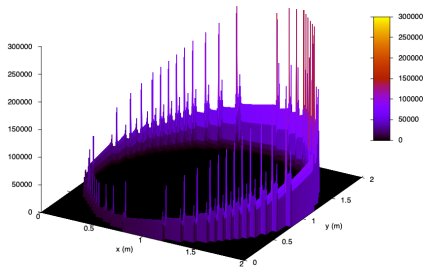
FIGURE 22 – Convergence de l'algorithme deux-grilles $||r_m|| - vs - m$



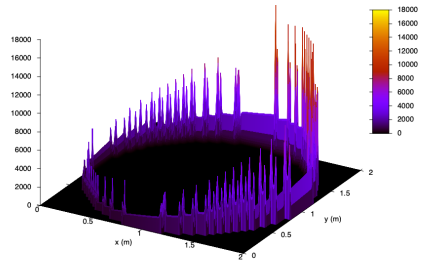
(a) résidu avant pré-lissage



(b) résidu après pré-lissage

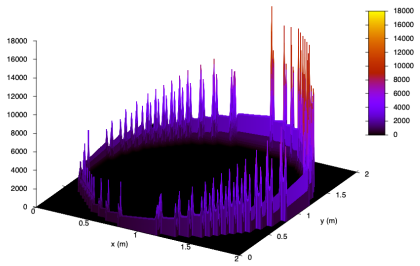


(c) résidu après coarse-grid

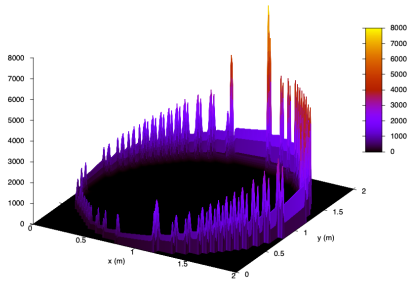


(d) résidu après post-lissage

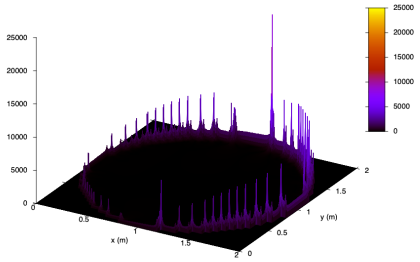
FIGURE 23 – Évolution du résidu de la méthode deux-grilles pour la 1ère itération (m = 641)



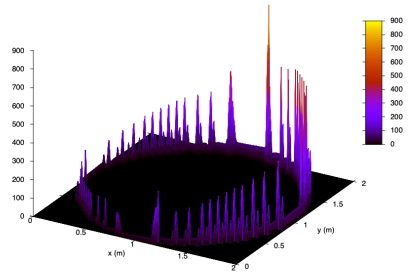
(a) résidu avant pré-lissage



(b) résidu après pré-lissage



(c) résidu après coarse-grid



(d) résidu après post-lissage

FIGURE 24 – Évolution du résidu de la méthode deux-grilles pour la 2ème itération (m = 641)

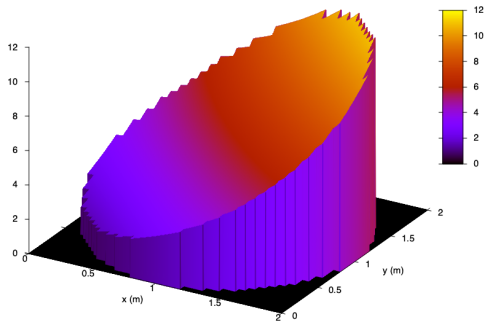


FIGURE 25 – Solution fournie par l'algorithme deux-grilles après 20 itérations

B Résultats de la méthode deux-grilles pour le domaine P15

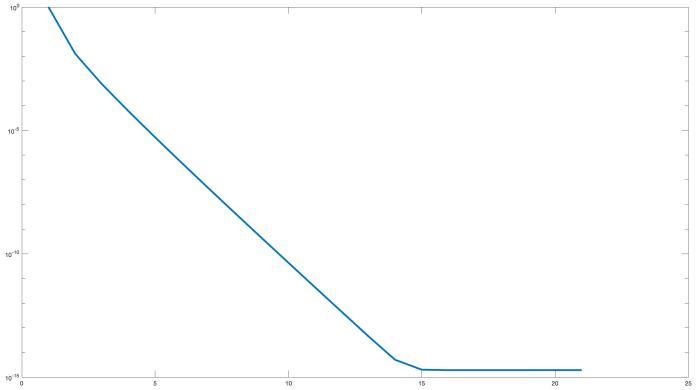
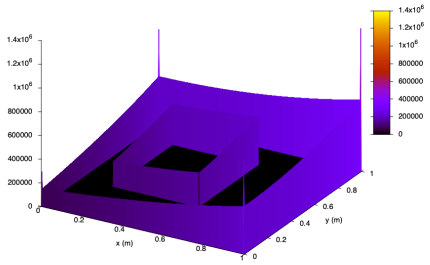
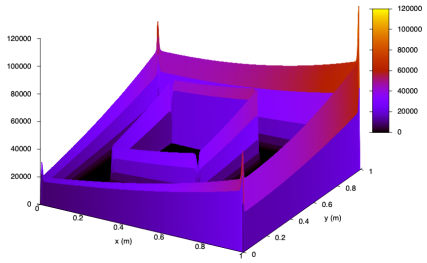


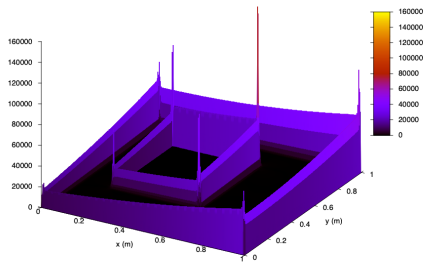
FIGURE 26 – Convergence de l'algorithme deux-grilles $||r_m|| - vs - m$



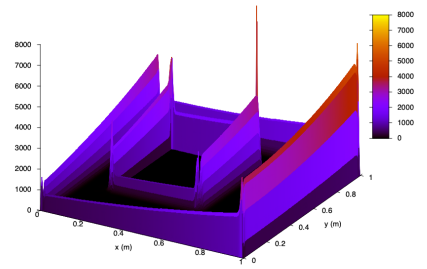
(a) résidu avant pré-lissage



(b) résidu après pré-lissage

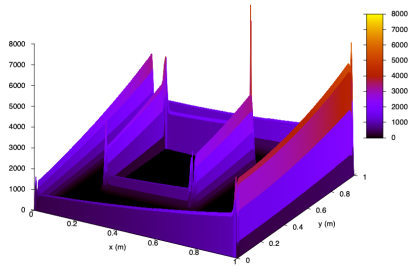


(c) résidu après coarse-grid

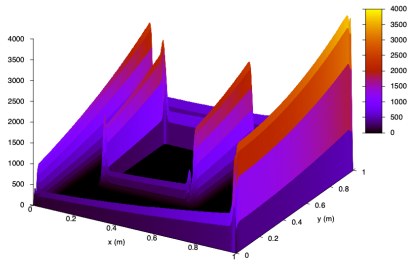


(d) résidu après post-lissage

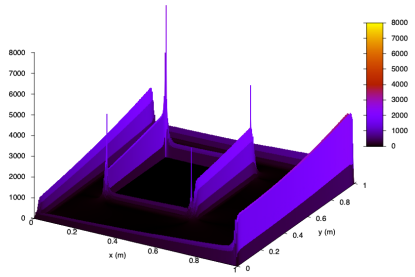
FIGURE 27 – Évolution du résidu de la méthode deux-grilles pour la 1ère itération (m = 385)



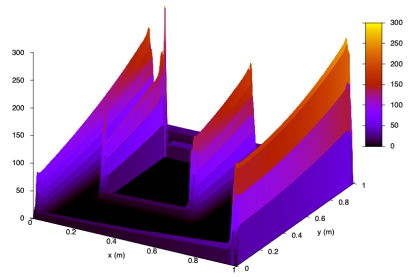
(a) résidu avant pré-lissage



(b) résidu après pré-lissage



(c) résidu après coarse-grid



(d) résidu après post-lissage

FIGURE 28 – Évolution du résidu de la méthode deux-grilles pour la 2ème itération (m = 385)

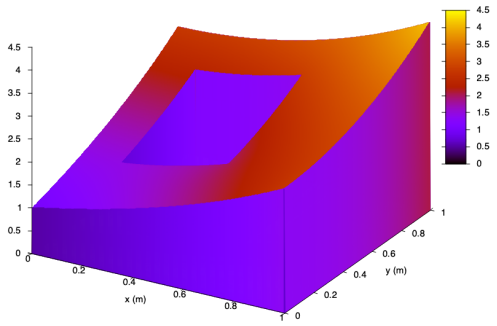


FIGURE 29 – Solution fournie par l'algorithme deux-grilles après 20 itérations