Projet d'éléments finis en C++

Naossa MAILLE-OKADA

Elève-Ingénieur à l'E.N.S.G.

15 mai 2024

École Nationale Supérieure de Géologie
2 rue du Doyen Roubault – BP 10162
54505 Vandœuvre-lès-Nancy CEDEX
Tél : 03 72 74 46 00
www.ensg.univ-lorraine.fr

Notice analytique

Titre du document	Projet d'Eléments Finis en C++
Cadre et nature du travail	Ecriture d'un code en C++ pour résoudre des problèmes de Poisson 2D par éléments finis.
Date de fin	15/05/2024
Auteur(s)	Naossa MAILLE-OKADA
Encadrement	Paul Cupillard
Résumé	Ce rapport vise à présenter les différents tests réalisés et résultats de simulation obtenus, suite à l'implémentation d'une méthode aux éléments finis de résolution de problèmes de Poisson 2D, avec des éléments triangulaires linéaires, en C++.
Nombre de pages	9
Annexes	

Table des matières

1. Introduction

L'implémentation d'une méthode aux éléments finis de résolution de problèmes de Poisson 2D, sur des éléments triangulaires linéaires a été faite en C++. Dans ce cadre, un certain nombre de tests ont été réalisés afin de vérifier pas à pas le bon fonctionnement du code. En outre, des simulations ont aussi été effectuées, sur différents modèles, permettant une analyse supplémentaire des résultats.

Les tests, ainsi que leurs résultats seront abordés dans une première partie, suivis des différentes simulations.

2. Tests

2.1. Vérifications des quadratures

Une fonction de quadrature était fournie au préalable. Ce test permet donc d'en vérifier les valeurs. En calculant l'intégrale d'une fonction unitaire sur le triangle à l'aide de sa quadrature d'ordre 2, l'aire obtenue pour un triangle est, conformément aux attentes, ½.

2.2. Vérification du mapping

Le bon fonctionnement du mapping peut être vérifié en affichant des points. Ici, le choix a été fait sur le bord 4, ainsi que le triangle 4. La vérification du fonctionnement de la construction de la matrice jacobienne, et du calcul de son déterminant est aussi réalisée. Les résultats obtenus, et validés, sont les suivants :

```
[Test Element Mapping]
[ElementMapping] constructor for element 4
Border vertices
x y
0.5 0
0.625 0

[ElementMapping] constructor for element 4

Triangle vertices
x y
0.625 1
0.639235 0.846329
0.734711 0.900627

[ElementMapping] transform reference to world space
Reference vertex 0.2 0.4
World space vertex 0.671731 0.929517

[ElementMapping] compute jacobian matrix
Reference vertex (xi=0.2, eta=0.4)
triangle
Matrice jacobienne
0.0142345 0.109711
-0.153671 -0.0993728

Determinant
0.0154449
```

2.3. Vérification de la shape function

Le fonctionnement de la fonction de forme est réalisé en vérifiant, d'une part, que le bon nombre de fonctions est construit (3 en dimension 2, ordre 1), et d'autre part, le résultat après application à un point. Les résultats obtenus étant cohérents, la fonction est validée.

```
[Test Shape Functions]
Nombre de fonctions en dim2, ordre1 3
[ShapeFunctions] evaluate shape function 2
Reference vertex (xi=0.2, eta=0.4)
0.4
[ShapeFunctions] evaluate gradient shape function 2
Reference vertex (xi=0.2, eta=0.4)
1
```

2.4. Vérification de la construction de la matrice élémentaire

Pour la vérification que la matrice Ke est bien construite, celle-ci est calculée et affichée pour un élément connu, ici le triangle 4. Les résultats obtenus correspondent à ce qui était attendu.

```
[Test Elementary Matrix]

compute elementary matrix
Ke (pour k=1), pour le triangle 4
0.390552  -0.164425  -0.226127
-0.164425  0.709343  -0.544918
-0.226127  -0.544918 0.771045
```

2.5. Vérification de l'assemblage de la matrice

Lors de l'assemblage de la matrice K, les correspondances d'indices priment. De ce fait, les correspondances d'indices entre Ke et K sont affichés, pour un élément connu. Ici, pour le triangle 4, comme attendu, le vertex 0 correspond au vertex 20 en coordonnées générales, le 1 au 49, et le 2 au 65.

En outre, une vérification générale visuelle (absence de valeur totalement aberrante, d'erreurs sur des points particuliers) est faite, en affichant tout le vecteur K (non recopié ici, du fait de sa taille).

```
[Test Assemble Matrix]
Ke -> K for triangle no.4
Le point de Ke en (0,0) correspond à (20,20) dans K
Le point de Ke en (0,1) correspond à (20,49) dans K
Le point de Ke en (0,2) correspond à (20,65) dans K
Le point de Ke en (1,0) correspond à (49,20) dans K
Le point de Ke en (1,1) correspond à (49,49) dans K
Le point de Ke en (1,2) correspond à (49,65) dans K
Le point de Ke en (2,0) correspond à (65,20) dans K
Le point de Ke en (2,1) correspond à (65,49) dans K
Le point de Ke en (2,2) correspond à (65,65) dans K
```

2.6. Vérification de l'application des conditions de Dirichlet

Pour la vérification de l'application des conditions de Dirichlet, les valeurs obtenues après simulation sont affichées. Ici, ceci a été réalisé pour une condition u=x+y sur les bords.

Cependant, il aurait sans doute mieux valu prendre une condition nulle, et afficher les valeurs des points en bordure après la résolution du problème, à partir des informations du maillage.

[Test Apply Dirichlet Conditions]

solving system with 109 unknowns ..

0 : 4.2756401090e+09 -- 4.2756401090e-15

in OpenNL : ||Ax-b||/||b|| = 9.959769e-13

.. system solved

U:

0.000148 1 2 1 0.125 0.25 0.375 0.5 0.625 0.75 0.875 1.13 1.25 1.38 1.5 1.63 1.75 1.88 1.88 1.75 1.63 1.5 1.38 1.25 1.13 0.875 0.75 0.625 0.5 0.375 0.25 0.125 7.83 6.64 6.29 6.26 5.87 5.89 5.31 5.88 5.27 4.3 3.64 3.64 2.97 7.3 7.07 7.17 6.98 5 4.3 4.72 4 4.7 4.96 3.99 4.26 7.03 6.71 7.18 6.85 3.94 3.13 4.01 3.2 3.9 3.09 3.43 2.61 4.01 3.22 3.54 2.75 6.4 5.91 6.18 5.63 3.76 2.93 3.65 2.81 6.17 6.4 5.66 5.89 3.03 2.22 2.22 1.41 5.55 4.97 5.11 4.51 5.58 5.15 4.99 4.55 7.69 7.74 7.49 7.55 3.27 2.41 2.63 1.77 3.29 2.43 2.65 1.79

2.7. Vérification de la construction du vecteur élémentaire

Pour la vérification que le vecteur élémentaire Fe est construit est fait de la même manière que pour la matrice Ke : celui-ci est calculé et affiché pour un élément connu, ici le triangle 4. Les résultats obtenus correspondent à ce qui était attendu.

[Test Elementary Vector]

compute elementary vector (source term)
create Fe
Fe (pour f=1), pour le triangle 4
0.00257 0.00257 0.00257

2.8. Vérification de l'assemblage du vecteur

De même, la vérification de l'assemblage du vecteur F est vérifié en affichant les correspondances d'indices. Celui-ci concorde bien avec ce qui était attendu, et avait été obtenu plus tôt.

En outre, une vérification générale visuelle (absence de valeur totalement aberrante, d'erreurs sur des points particuliers) est faite, en affichant tout le vecteur F.

[Test Assemble Vector]

compute elementary vector (source term)
create Fe
Fe -> F no.4
Numero global du point 0 : 20
Numero global du point 1 : 49
Numero global du point 2 : 65

F (pour f=1)
0.00405 0.00405 0.00405 0.00405 0.0048 0.00493 0.00836 0.00537 0.00936 0.00514 0.00482 0.0048 0.00493 0.00836 0.00537 0.00936 0.00514 0.00482 0.0048 0.00493 0.00836 0.00537 0.00936 0.00514 0.00482 0.0048 0.00493 0.00836 0.00537 0.00936 0.00514 0.00482 0.00674 0.0101 0.0102 0.0103 0.0107 0.0172 0.0174 0.0172 0.0178 0.0112 0.0112 0.0112 0.0112 0.0145 0.0138 0.014 0.0168 0.012 0.012 0.012 0.0121 0.0126 0.0126 0.0126 0.0126 0.0122 0.0119 0.012 0.0117 0.00999 0.00999 0.00999 0.00999 0.0111 0.0111 0.0111 0.0111 0.0112 0.0112 0.0112 0.0112 0.00921 0.00931 0.00916 0.00992 0.00936 0.00936 0.00936 0.00936 0.00828 0.00836 0.00839 0.00844 0.00912 0.00912 0.00912 0.00912 0.0106 0.0106

```
0.0106 0.0106 0.01 0.01 0.01 0.01 0.0094 0.00953 0.00757 0.00767 0.00682 0.00682 0.00682 0.00682
0.0068 0.0068 0.0068 0.0068
```

2.9. Vérification de l'application des conditions de Neumann

Le bon fonctionnement de l'application des conditions de Neumann est vérifiée à travers les valeurs obtenues, puis, au cours de l'assemblage, en affichant des correspondances. Encore une fois, ceci est appliqué sur le triangle 4, dont les résultats sont connus : la fonction est validée.

```
[Test Neumann Conditions]
Fe (pour f=1), pour le bord 4
0.0625 0.0625
Fe -> F no.4
Numero global du point 0 : 7
Numero global du point 1 : 8
```

2.10. Vérification de la résolution des problèmes de Poisson

La vérification du fonctionnement de la fonction de résolution de problèmes de Poisson est effectuée en réalisant une simulation. Dans le cas présent, il s'agit du problème de Neumann, dont les résultats ont ensuite comparés à travers la visualisation des résultats.

```
 [Test Poisson Problem Solver]
solving problem on data/square.mesh
solve poisson problem

[CREATING QUADRATURE AND SHAPE FUNCTIONS]

[CREATE EMPTY K, F]

[CREATE MATRIX K AND VECTOR F]

[APPLY DIRICHLET CONDITIONS]

[APPLY NEUMANN CONDITIONS]

[SOLVING PROBLEM]
solving system with 109 unknowns ..
0 : 7.9472796680e-02  -- 7.9472796680e-26
100 : 1.1190882914e-16  -- 7.9472796680e-26
in OpenNL : ||Ax-b||/||b|| = 7.996708e-13
.. system solved
```

3. Simulations

3.1. Problème de Dirichlet pur

Dans ce problème, seules des conditions de Dirichlet, imposant une valeur précise sur des bords(de température, par exemple), sont utilisées. Ici, celles-ci sont de u = x + y, et sont appliquées à un modèle carré unitaire. Le coefficient de diffusion étant constant égal à 1, des bandes d'intensité croissante, allant de 0 à 2 respectivement entre (0,0) et (1,1), sont attendues.

C'est en effet ce qui est obtenu, comme il est possible de le voir dans les figures obtenues avec deux maillages différents sur le même milieu. Si les valeurs ne vont pas strictement de 0 à 2, les valeurs obtenues en restent néanmoins raisonnablement proches pour que la différence puisse être négligés.

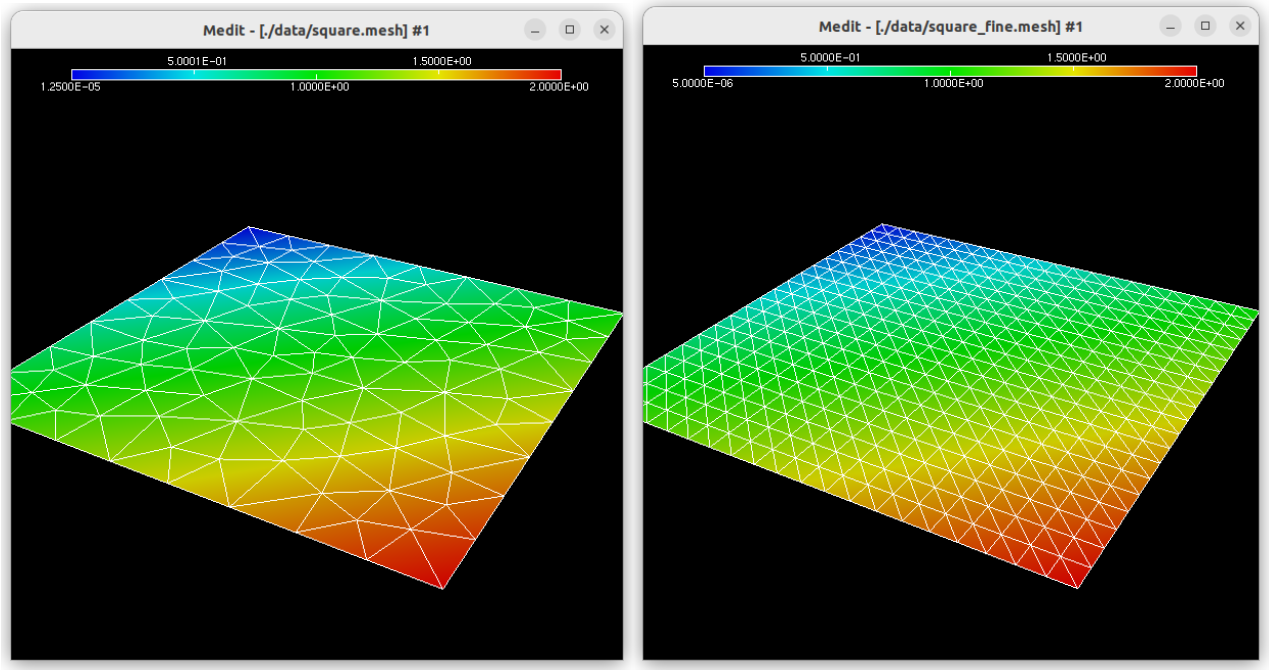

Figure 1 Résultats de simulations pour des maillages large (gauche) et fin (droite) pour un problème de Dirichlet pur

3.2. Problème de Dirichlet avec terme source

Dans cette simulation, le même modèle est conservé, ainsi que le coefficient de diffusion égal à 1. En plus de cela, un terme source constant égal à 1 est ajouté, et des conditions de Dirichlet nulles sont utilisées.

Le résultat attendu correspond donc à des valeurs nulles sur les bords, croissant vers le centre du modèle. C'est par ailleurs ce qui est observé, à l'exception de l'erreur due au processus de modélisation.

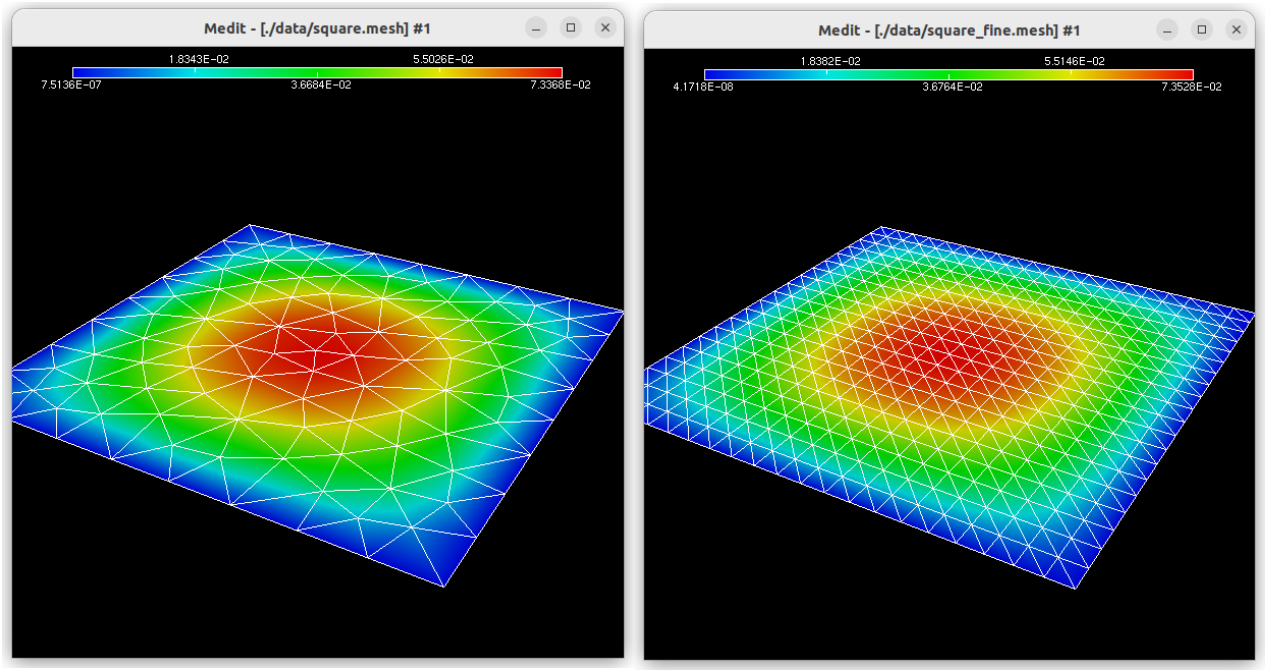

Figure 2 Résultats de simulations pour des maillages large (gauche) et fin (droite) pour un problème de dirichlet avec terme source

Une différence plus importante que précédemment est visible entre les deux images. En effet, dans la simulation sur le maillage fin, et plus régulier, l'influence des conditions de Dirichlet associée à la forme du milieu se fait plus sentir : les zones de plus grande « température », sont moins arrondies qu'avec le maillage large.

3.3. Problème de sinus bump

Ici, le problème ressemble au précédent : le même modèle carré est utilisé, avec un coefficient de diffusion de 1, et des conditions de Dirichlet nulles sur tous les bords. Le terme source, lui est variable et vaut : $f = 2\pi^2 sin(\pi x) sin(\pi y)$.

Une allure similaire à celle obtenue pour le problème de Dirichlet avec terme source est attendue dans la figure créée (Figure 3). Les conditions aux bords sont identiques, et les valeurs du terme source sont réparties de façon similaire aux résultats du problème précédent. Cependant, des valeurs de « température » plus importantes sont attendues au centre du domaine, puisque le terme source dépasse par endroits celui du problème précédent.

L'affichage du résultat de ce problème confirme cela. En effet, un maxima de 0.99 est observé ici, tandis qu'il n'était que de 0.074 avec le terme source constant.

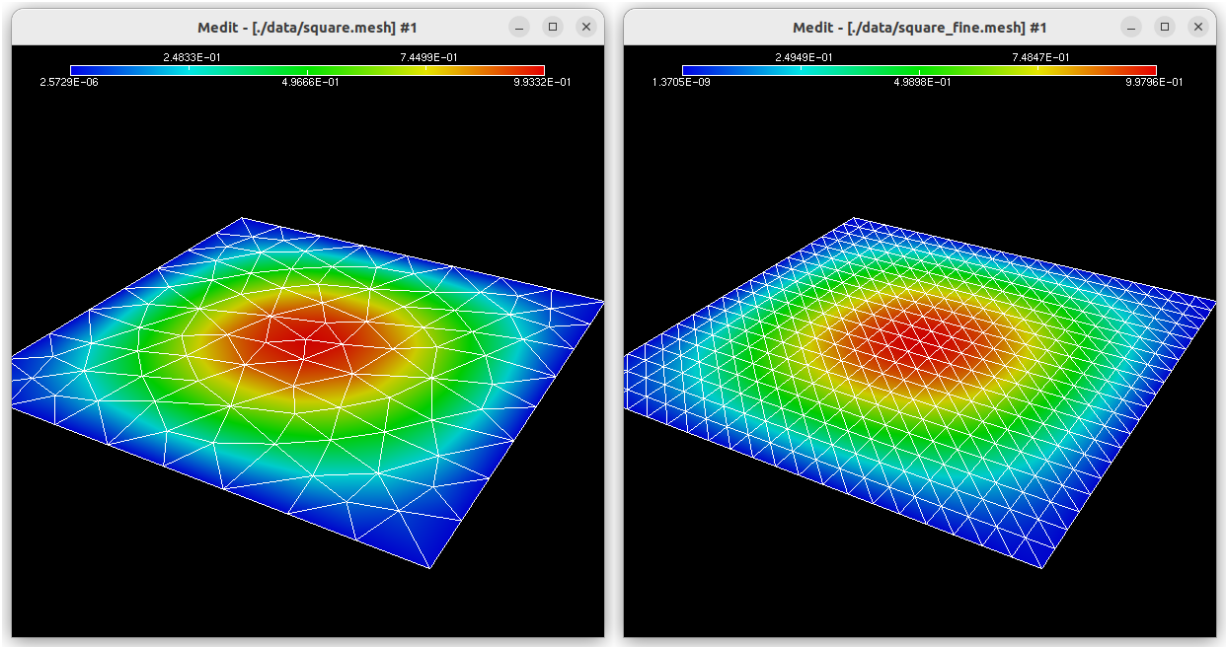

Figure 3 Résultats de simulations pour des maillages large (gauche) et fin (droite) pour un problème de sinus bump

En outre, pour ce problème une solution analytique est connue, de la forme $u = sin(\pi x) * sin(\pi y)$. Il est donc possible de les comparer visuellement (Figure 4), mais aussi de calculer l'erreur du résultat obtenu par éléments finis (Figure 5).

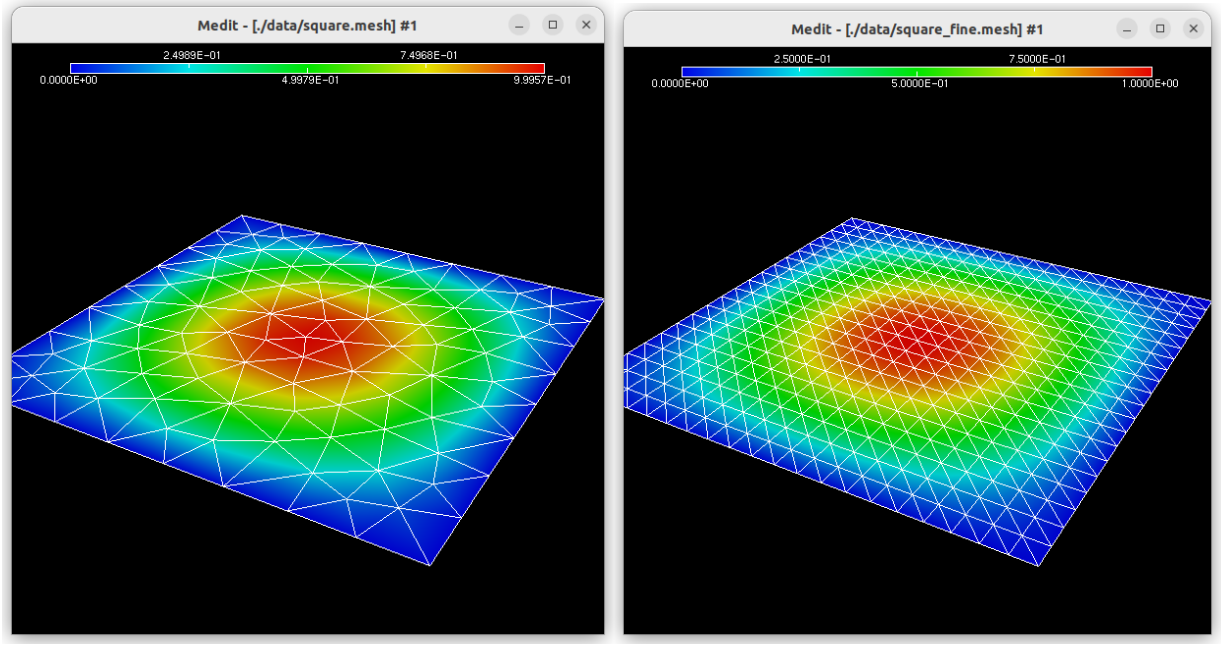

Figure 4 Solution analytique pour des maillages large (gauche) et fin (droite) pour un problème de sinus bump

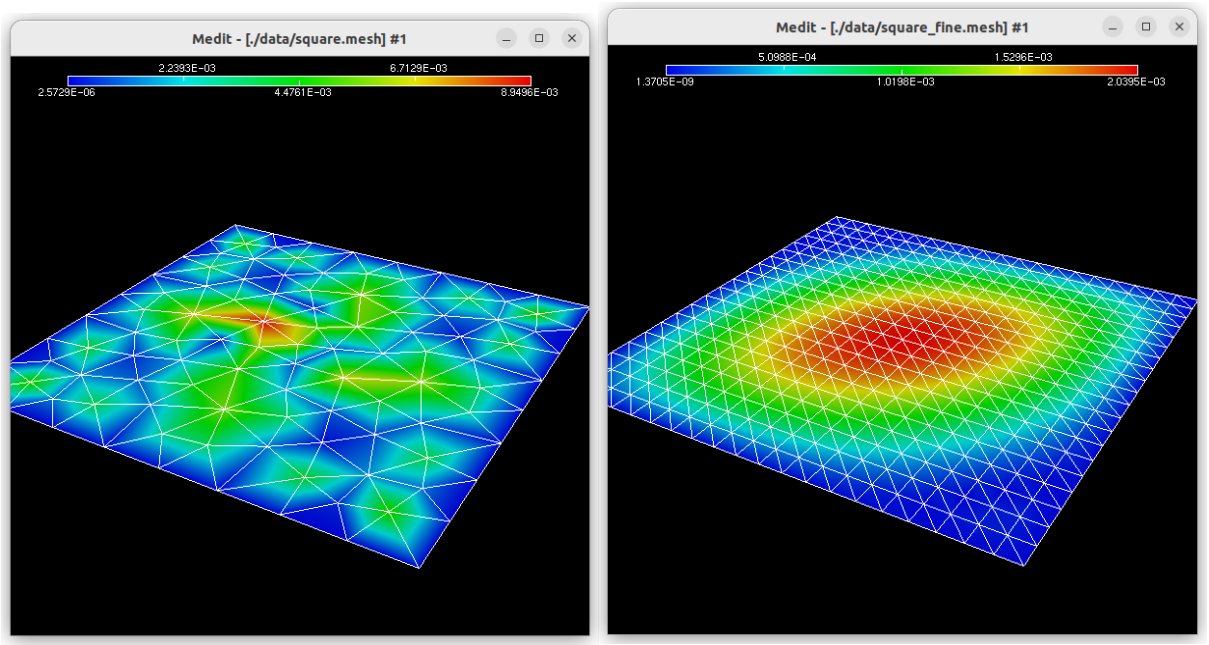

Figure 5 Erreur (simulation - solution analytique) pour des maillages large (gauche) et fin (droite) dans un problème de sinus bump

L'allure de l'erreur avec le maillage plus large sans doute due au fait qu'elle est moins régulière, et l'on constate que celle-ci atteint ses extrema localement autour 1 à quelques points. A l'inverse, dans le maillage fin, régulier, l'erreur suit une tendance générale, et est plus petite que celle du maillage plus large. Il serait donc possible qu'elle soit comprise dans les résultats obtenus avec ce dernier, mais non visible du fait des autres erreurs plus grandes. Il pourrait par exemple s'agir d'une conséquence d'approximations utilisées lors de l'application de la fonction du terme source, ou des calculs matriciels.

3.4. Problème de Neumann

Cette fois, si le modèle considéré reste le même, les conditions imposées aux frontières sont différentes : certains bords sont soumis à des conditions de Neumann, imposant des valeurs de flux.

A droite, une condition de Dirichlet de u=0 est imposée. Sur les frontières horizontales, une condition de Neumann nulle est imposée. Enfin, à gauche, une condition de Neumann de h=sin(πy) est utilisée. En outre, un terme source et un coefficient de diffusion unitaires sont utilisés.

Une diminution des valeurs de température est donc attendue de gauche à droite, où celle-ci doit atteindre 0, chose que l'on obtient bien (Figure 6). En plus de cela, la condition de Neumann sur le bord gauche devrait résulter en une répartition sinusoïdale des températures.

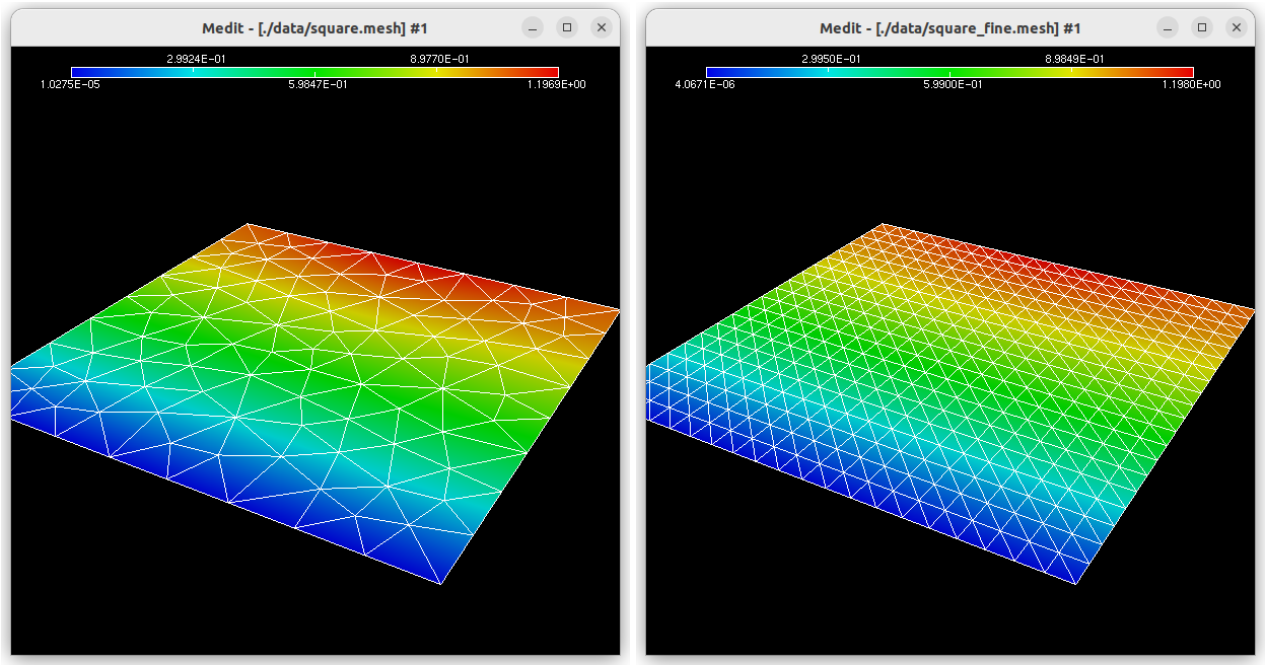

Figure 6 Résultats de simulations pour des maillages large (gauche) et fin (droite) pour un problème de Neumann

3.5. Problème de diffusion dans un mug

Enfin, une dernière simulation est réalisée, de diffusion de chaleur dans un mug, rempli d'eau à 100°C jusqu'à mi-hauteur. De ce fait, la partie de la porcelaine en contact avec l'eau est soumise à une condition de Dirichlet de 100. La chaleur étant évacuée depuis le mug vers l'extérieur, une condition de Neumann de h=-0.1 est imposée sur le restant des bords.
En outre, un coefficient de diffusion constant unitaire est utilisé.

Les résultats de la simulation sont visualisés (Figure 7). Dans celui-ci, les conditions de Dirichlet sont visiblement respectées, et la répartition générale semble cohérente, puisque la température baisse à mesure que l'on s'éloigne de cela.

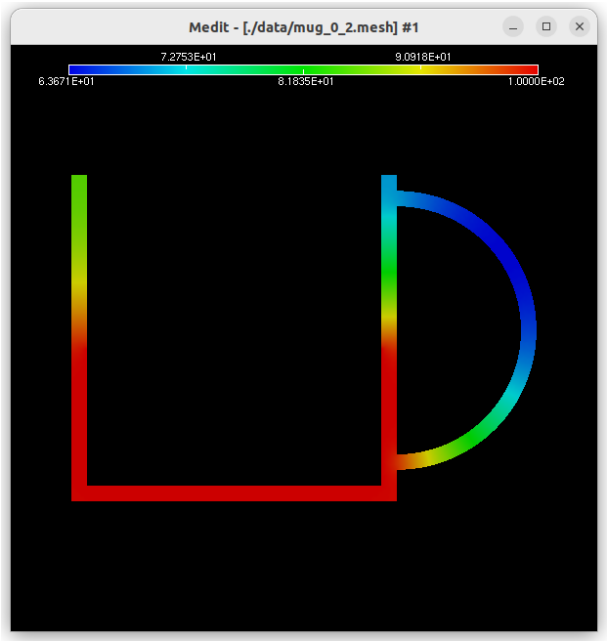

Figure 7 Résultat de simulation dans un mug