

Projet 2 : Résolution de l'équation stationnaire de la chaleur en 2D

Serge Bays

Avril 2010

Nous allons étudier la question de la propagation de la chaleur à travers un problème physique particulier : celui de l'évolution de la température dans un four.

Ce problème sera modélisé afin d'obtenir des équations aux dérivées partielles avec des conditions aux limites en deux dimensions.

Une étude numérique sera menée avec la méthode des éléments finis utilisée à partir de la formulation variationnelle du problème. Les calculs se feront d'une part à l'aide du logiciel `FreeFem++` et d'autre part avec des programmes écrits en `C++`. Les résultats obtenus seront alors comparés.

L'intérêt d'une simulation numérique est évident si l'on compare son coût avec celui d'essais expérimentaux.

La présentation pourra se faire à l'aide d'une interface programmée en langage Java. Cette interface permettra de définir dans un tableau éditable les différents paramètres du problème, de lancer le calcul `C++`, d'écrire les résultats dans des fichiers exploitables par le logiciel `Gnuplot` ou d'afficher les résultats obtenus sous forme graphique en 3D avec `OpenGL`.

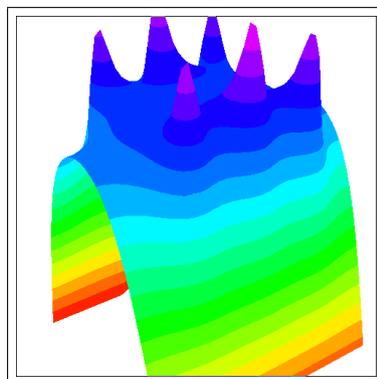


Table des matières

1	Modélisation du problème	2
2	Formulation variationnelle	3
2.1	Problème général	3
2.2	Décomposition du problème	4
2.3	Discretisation	4
3	Problème sans source	7
3.1	Utilisation du logiciel FreeFem++	7
3.2	Programmation en C++	7
4	Problème avec sources	10
4.1	Une résistance chauffante	10
4.1.1	Problème Pb	10
4.1.2	Problème Pb_r	12
4.2	Six résistances chauffantes	14
5	Sensibilité aux paramètres	17
5.1	Température fonction de la source	17
5.1.1	Etude par différences finies	17
5.1.2	Etude par différentiation automatique	17
5.2	Température fonction des conditions limites	20
5.3	Température fonction de la conductivité de la pièce	22
6	Conclusion	24
7	Annexe : les programmes	26

1 Modélisation du problème

Considérons un four destiné à la cuisson d'une pièce de résine thermo-formée. Le four est modélisé par un rectangle et la pièce par un rectangle plus petit à l'intérieur du four. Les éléments chauffants sont des résistances électriques dont la valeur est connue et placées dans le four. On cherche à déterminer le champ de température à l'intérieur du four et en particulier la température de l'objet à cuire.

Le four est donc représenté par le domaine rectangulaire noté Ω , avec

$$\Omega = [-L/2; L/2] \times [-H/2; H/2]$$

de frontière $\Gamma = \partial\Omega$.

Nous aurons besoin de distinguer les quatre côtés du four et posons donc $\Gamma = \Gamma_1 + \Gamma_2 + \Gamma_3 + \Gamma_4$.

La pièce est représentée par le domaine rectangulaire noté Ω_1 , avec

$$\Omega_1 = [-\ell/2; \ell/2] \times [-h/2; h/2]$$

de frontière $\gamma = \partial\Omega_1$.

Les sources de chaleur sont les résistances notées q et modélisées par des cercles.

Le domaine obtenu est visible sur la figure 1.

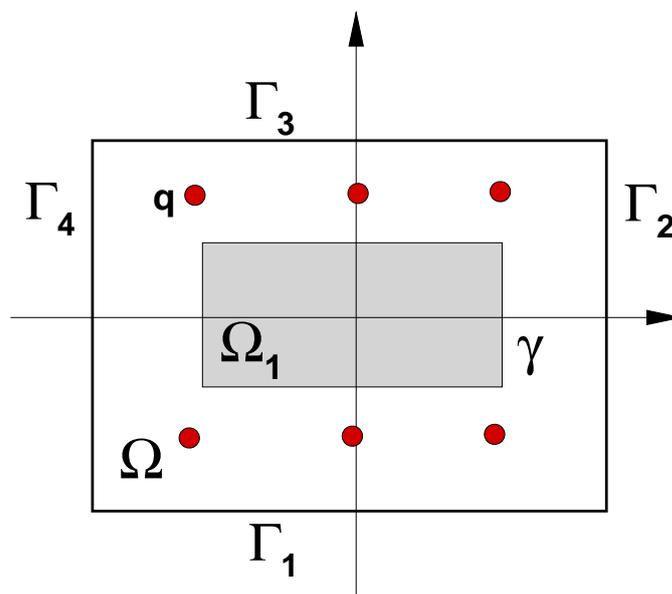


FIGURE 1 – Modélisation du four et de la pièce

La température d'équilibre dans le four est la solution de l'équation :

$$-\nabla(k\nabla u) = f$$

équation aux dérivées partielles qui s'écrit pour $(x, y) \in \Omega$:

$$-\left[\frac{\partial}{\partial x} \left(k(x, y) \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(k(x, y) \frac{\partial u}{\partial y} \right) \right] = f(x, y)$$

avec les conditions aux limites sur Γ :

$$\begin{cases} \text{sur } \Gamma_1 & u = u_s & \text{température sud imposée} \\ \text{sur } \Gamma_2 & \partial u / \partial x = 0 & \text{paroi isolée} \\ \text{sur } \Gamma_3 & u = u_n & \text{température nord imposée} \\ \text{sur } \Gamma_4 & \partial u / \partial x = 0 & \text{paroi isolée} \end{cases}$$

k est la diffusivité thermique :

$$k(x, y) = \begin{cases} k_1 & \text{sur } \Omega \setminus \Omega_1 \\ k_2 & \text{sur } \Omega_1 \end{cases}$$

$f(x, y)$ est le terme source qui sera défini par la suite.

2 Formulation variationnelle

2.1 Problème général

Nous nous plaçons dans l'espace $H^1(\Omega) = \{v \in L^2(\Omega), \nabla v \in L^2(\Omega)\}$, où $L^2(\Omega)$ est l'espace des fonctions de Ω dans \mathbb{R} , de carré intégrable ; une fonction de $H^1(\Omega)$ appartient donc à $L^2(\Omega)$ ainsi que ses dérivées partielles par rapport à x et y .

Nous allons prendre en compte la discontinuité de la fonction k en séparant le domaine Ω en deux domaines $\Omega \setminus \Omega_1$ et Ω_1 sur chacun desquels k est constante. Si u est une fonction de $H^1(\Omega)$ solution du problème et v une fonction quelconque de $H^1(\Omega)$, à partir de l'égalité

$$-\nabla(k\nabla u) = f$$

nous multiplions par v et en intégrons sur les deux domaines :

$$\int_{\Omega \setminus \Omega_1} -\nabla(k\nabla u)v = \int_{\Omega \setminus \Omega_1} fv \quad \text{et} \quad \int_{\Omega_1} -\nabla(k\nabla u)v = \int_{\Omega_1} fv$$

pour toute fonction $v \in H^1(\Omega)$.

En appliquant la formule de Green nous obtenons les deux égalités :

$$\int_{\Omega \setminus \Omega_1} (k\nabla u)\nabla v - \int_{\partial(\Omega \setminus \Omega_1)} (k\nabla u \cdot n)v = \int_{\Omega \setminus \Omega_1} fv$$

et

$$\int_{\Omega_1} (k\nabla u)\nabla v - \int_{\partial\Omega_1} (k\nabla u \cdot n)v = \int_{\Omega_1} fv$$

où n est le vecteur normal à chaque frontière, dirigé vers l'extérieur, soit :

$$\int_{\Omega \setminus \Omega_1} (k\nabla u)\nabla v - \int_{\Gamma} k \frac{\partial u}{\partial n} v - \int_{\gamma} k \frac{\partial u}{\partial n} v = \int_{\Omega \setminus \Omega_1} fv$$

et

$$\int_{\Omega_1} (k\nabla u)\nabla v - \int_{\gamma} k \frac{\partial u}{\partial n} v = \int_{\Omega_1} f v$$

Physiquement la quantité $k \frac{\partial u}{\partial n}$ est continue à la traversée de γ . Cette frontière γ a une "épaisseur très fine", à l'intérieur de laquelle les quantités examinées varient très rapidement mais de manière continue.

Alors les termes correspondants aux intégrales sur γ s'éliminent lorsque nous ajoutons les deux égalités précédentes membres à membres :

$$\int_{\Omega \setminus \Omega_1} (k\nabla u)\nabla v - \int_{\Gamma} k \frac{\partial u}{\partial n} v + \int_{\Omega_1} (k\nabla u)\nabla v = \int_{\Omega \setminus \Omega_1} f v + \int_{\Omega_1} f v$$

soit :

$$\int_{\Omega} (k\nabla u)\nabla v - \int_{\Gamma} k \frac{\partial u}{\partial n} v = \int_{\Omega} f v$$

Plus précisément, avec les conditions imposées sur Γ_2 et Γ_4 , (où $\partial u / \partial x = 0$), la fonction solution u vérifie, pour toute fonction $v \in H^1(\Omega)$:

$$\int_{\Omega} (k\nabla u)\nabla v - \int_{\Gamma_1 \cup \Gamma_3} k \frac{\partial u}{\partial n} v = \int_{\Omega} f v$$

2.2 Décomposition du problème

Nous allons décomposer le problème général noté Pb avec $f(x, y) = f_r$, u_n et u_s non nuls, en deux sous problèmes notés Pb_r et Pb_0 .

Le problème Pb_r correspond à $f(x, y) = f_r$ et $u_n = u_s = 0$, le problème Pb_0 correspond à $f(x, y) = f_0 = 0$, u_n et u_s non nuls.

Si l'on note U , U_r , U_0 les solutions correspondantes, il est clair que $U = U_0 + U_r$.

En effet, si U_0 est solution du problème Pb_0 et si U_r est solution du problème Pb_r alors

$$-\nabla(k\nabla U_0) = 0 \quad \text{et} \quad -\nabla(k\nabla U_r) = f_r$$

Donc, par linéarité : $-\nabla(k\nabla(U_0 + U_r)) = f_r$

De plus : $U_n = 0 + u_n = u_n$ et $U_s = 0 + u_s = u_s$.

2.3 Discrétisation

Le problème est posé dans l'espace $H^1(\Omega)$ de dimension infinie. Pour obtenir une approximation numérique de la solution u , nous allons le simplifier par un problème approché posé dans un espace de dimension finie. Nous utilisons la méthode des éléments finis.

Nous commençons par construire un maillage du domaine Ω qui sera noté T_h , c'est-à-dire un recouvrement de Ω par des triangles K où h désigne le diamètre maximal des triangles.

La réunion des triangles est égale à Ω , l'intersection de deux triangles distincts est soit vide, soit un sommet commun, soit une arête commune.

Les figures 2 et 3 nous présentent un maillage construit à l'aide du logiciel FreeFem++ et la valeur de la conductivité constante sur chacun des sous domaines $\Omega \setminus \Omega_1$ et Ω_1 .

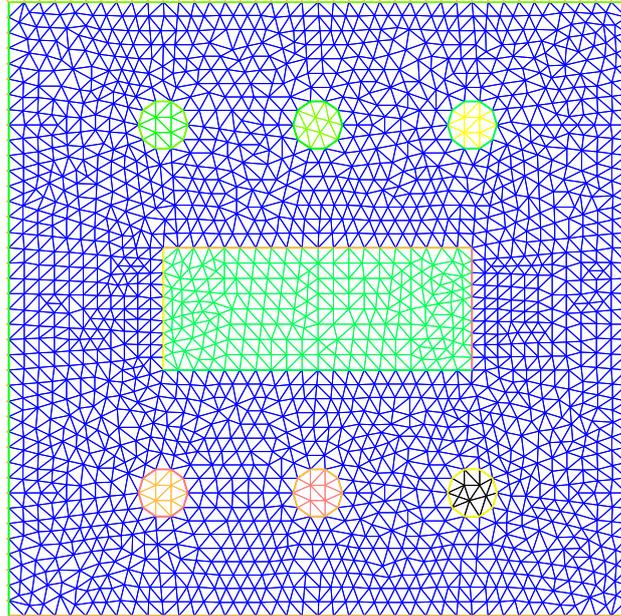


FIGURE 2 – *Maillage du domaine*

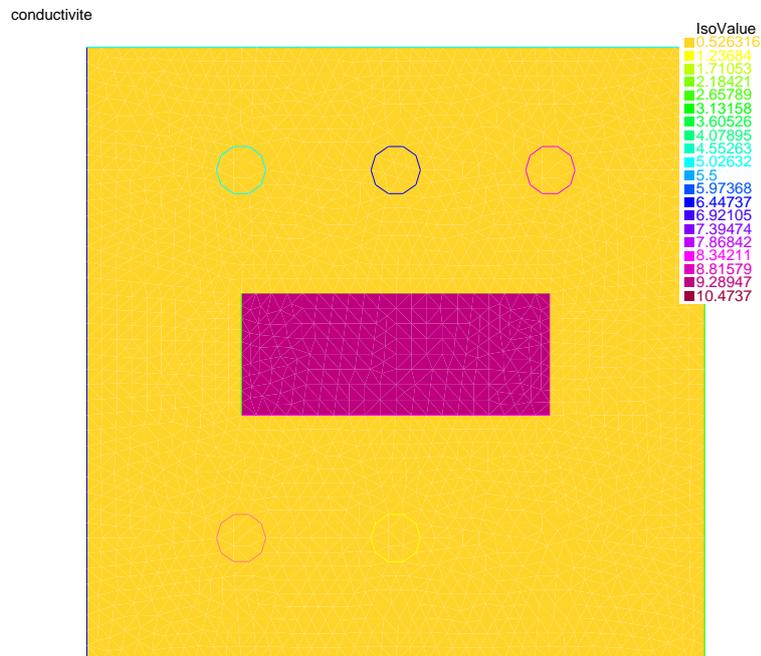


FIGURE 3 – *Conductivité dans le domaine et la pièce*

Un élément fini P_1 construit sur un triangle K est la donnée d'un triangle K , de l'espace des fonctions polynômiales de degré inférieure ou égal à 1 et de trois formes linéaires φ_i telles que $\varphi_i(v) = v(a_i)$ où a_i est un sommet du triangle K . Une base associée à cet élément fini est formée des fonctions coordonnées barycentriques du triangle. De plus $V_h^1 \subset H^1(\Omega)$.

Nous considérons alors l'espace d'approximation noté V_h^1 . Cet espace vectoriel est l'ensemble des fonctions v continues sur Ω , dont la restriction à chaque triangle K du maillage est un polynôme de degré inférieur ou égal à 1.

Si $\{s_i\}$ est l'ensemble des sommets du maillage, une base de cet espace vectoriel est constituée des fonctions $\phi_j \in V_h^1$ définies par $\phi_j(s_i) = \delta_{j,i}$.

La dimension de cet espace est donc égale au nombre de sommets N du maillage et une fonction v de V_h^1 se décompose sur cette base :

$$v = \sum_{j=1}^N v(s_j) \phi_j$$

Le problème approché se pose alors ainsi : trouver $u \in V_h^1$ telle que pour tout fonction $v \in V_h^1$,

$$\int_{\Omega} (k \nabla u) \nabla v - \int_{\Gamma_1 \cup \Gamma_3} k \frac{\partial u}{\partial n} v = \int_{\Omega} f v$$

Dans le cadre de notre problème, il nous restait à traduire les conditions imposées sur Γ_1 et Γ_3 . Si les conditions sont $u_s = u_n = 0$ alors nous restreignons l'espace V_h^1 à l'espace $V_{0h}^1 \subset V_h^1$, dont les fonctions v sont nulles sur $\Gamma_1 \cup \Gamma_3$. La solution u et les fonctions v appartiennent au même espace. Cet espace a pour dimension n , le nombre de sommets qui ne sont pas sur $\Gamma_1 \cup \Gamma_3$.

Le problème se traduit alors par :

trouver $u_1, u_2, \dots, u_n \in \mathbb{R}$ tels que pour toute famille $v_1, v_2, \dots, v_n \in \mathbb{R}$:

$$\sum_{1 \leq i, j \leq n} u_i v_j \int_{\Omega} (k \nabla \phi_i) \nabla \phi_j = \sum_{j=1}^n \int_{\Omega} f \phi_j$$

où n est le nombre de sommets qui ne sont pas sur $\Gamma_1 \cup \Gamma_3$.

Si u est non nulle sur $\Gamma_1 \cup \Gamma_3$, nous posons $u = u_{\Gamma} + w$ où $u_{\Gamma} = u$ sur $\Gamma_1 \cup \Gamma_3$ et $w \in V_{0h}^1$. Nous obtenons un problème identique au précédent avec la condition $u - u_{\Gamma} \in V_{0h}^1$.

Notre problème est donc ramené à la résolution d'un système $Au = Mf$, en dimension n , avec A matrice symétrique définie positive, ou en dimension N si on garde les équations correspondant à $u = u_{\Gamma}$ sur $\Gamma_1 \cup \Gamma_3$.

3 Problème sans source

Nous considérons le problème sans source qui se traduit donc par :

$$\int_{\Omega} (k \nabla u) \nabla v = 0$$

pour toute fonction $v \in V_{0h}^1$ avec les conditions $u = u_s$ sur Γ_1 et $u = u_n$ sur Γ_3 .

3.1 Utilisation du logiciel FreeFem++

A l'aide du logiciel FreeFem++, nous avons déjà défini le maillage de Ω et la conductivité. Nous définissons alors l'espace d'éléments finis P_1 et traduisons ensuite le problème à résoudre.

Le programme permet également de comparer la solution avec celle qui sera obtenue par le programme C++.

Le résultat pour le champ de température est présenté sur la figure 4 et nous voyons clairement l'influence de la conductivité.

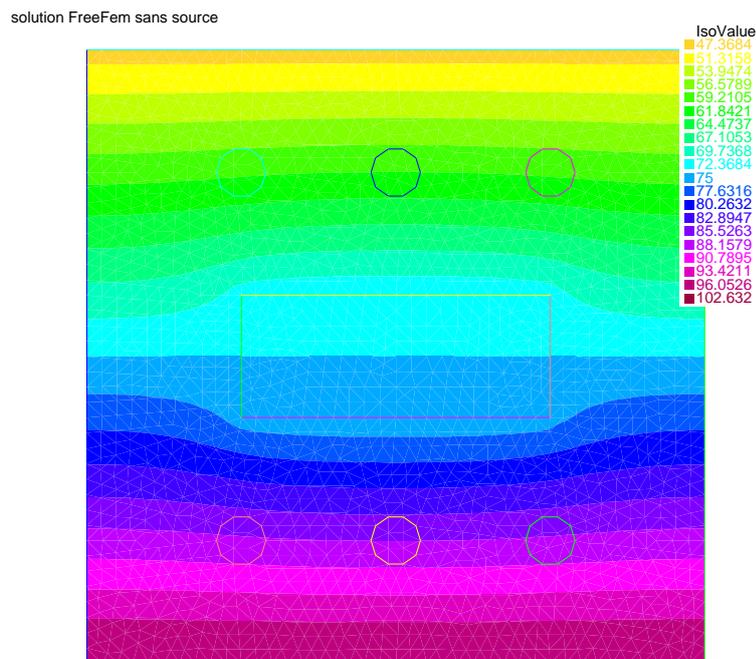


FIGURE 4 – Solution sans sources avec FreeFem

3.2 Programmation en C++

Le programme C++ est construit à l'aide de classes pour les sommets et les triangles du maillage afin de pouvoir lire le fichier de maillage généré par le programme FreeFem++.

Il permet de définir le second membre du problème ainsi que les conditions sur la frontière.

Le système linéaire $Au = Mf$ est résolu par la méthode du gradient conjugué. Grâce à la fonction "addMatMul" implémentée dans la classe "MatLap2dD", on ne stocke pas les matrices mais on calcule seulement les produits matrice par vecteur utilisés par la méthode du gradient conjugué.

Le vecteur solution est alors calculé et le programme écrit trois fichiers afin de présenter la solution sous des formes graphiques différentes :

- un fichier lisible par le programme FreeFem++ qui calculera les lignes de niveau,
- un fichier donnant les lignes de niveau,
- un fichier permettant une vision en 3D du champ de température.

Les deux derniers fichiers sont utilisables par le logiciel Gnuplot.

Les lignes de niveau sont visibles sur la figure 5.

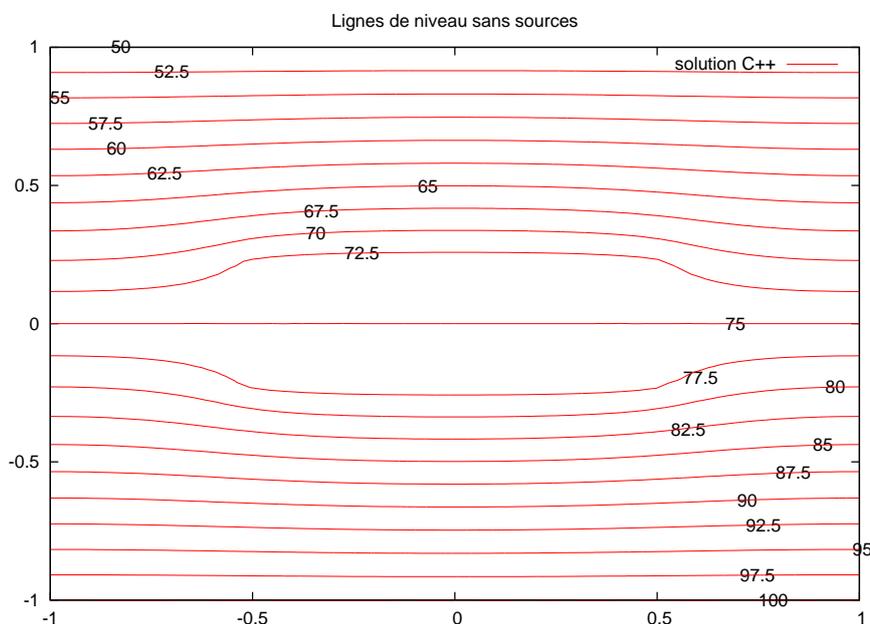


FIGURE 5 – Lignes de niveau du champ de température sans sources

La figure 6 présente une vue en 3D du champ de température.

La différence entre les solutions calculées par le programme C++ et par FreeFem++ est de l'ordre de 10^{-8} .

Cette différence est représentée sur la figure 7.

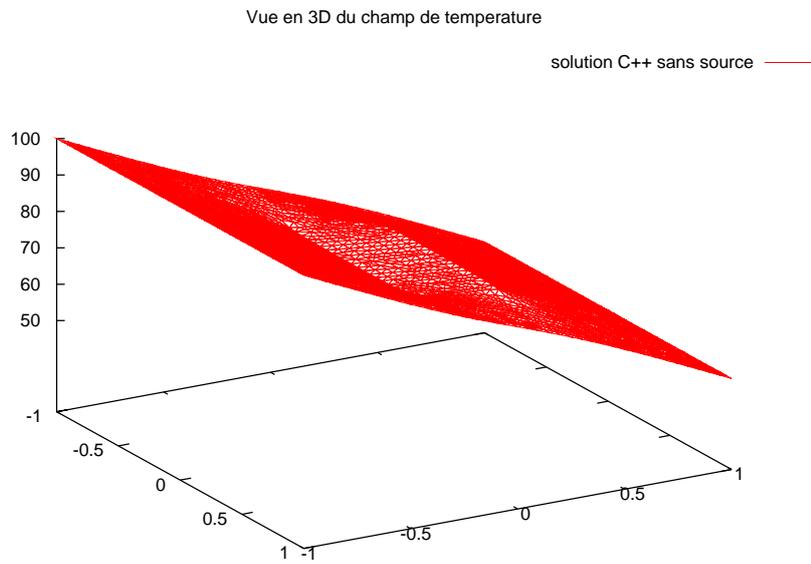


FIGURE 6 – *Champ de température sans source*

différence entre les solutions FreeFem et C++

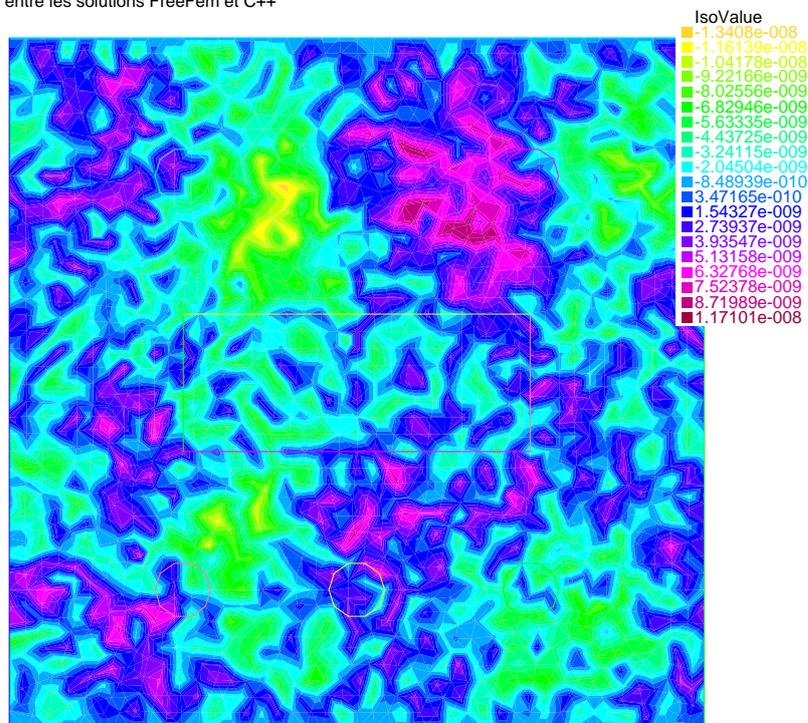


FIGURE 7 – *Différence entre les solutions C++ et FreeFem++*

4 Problème avec sources

Nous considérons maintenant le problème avec une source, (les résistances), qui se traduit donc par :

$$\int_{\Omega} (k \nabla u) \nabla v = \int_{\Omega} f v$$

pour toute fonction $v \in V_{0h}^1$ avec les conditions $u = u_s$ sur Γ_1 et $u = u_n$ sur Γ_3 .

4.1 Une résistance chauffante

4.1.1 Problème P_b

Une seule résistance fonctionne. Nous résolvons le problème P_b qui correspond à $f(x, y) = f_r$ avec les conditions u_s et u_n respectivement sur Γ_1 et Γ_3 .

La solution obtenue avec FreeFem++ est présentée sur la figure 8.

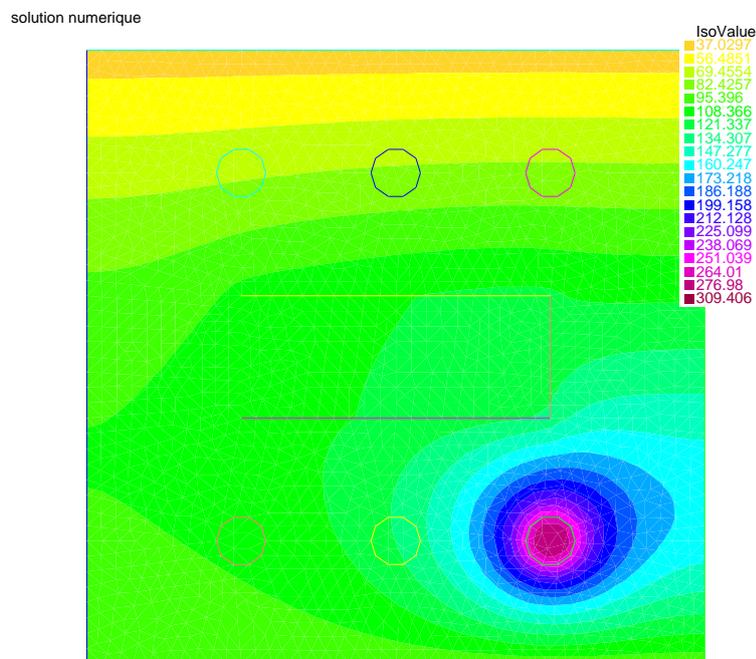


FIGURE 8 – Solution FreeFem avec une résistance, $u_n=100$, $u_s=50$

Les lignes de niveau calculées par le programme C++ sont présentées sur la figure 9. La figure 10 nous montre la vue en 3D du champ de température calculé également par le programme C++. Ce programme permet aussi d'exploiter la solution obtenue par FreeFem++ et écrite dans un fichier, et de calculer la différence entre les deux solutions, en norme sup, qui est de l'ordre de 10^{-8} .

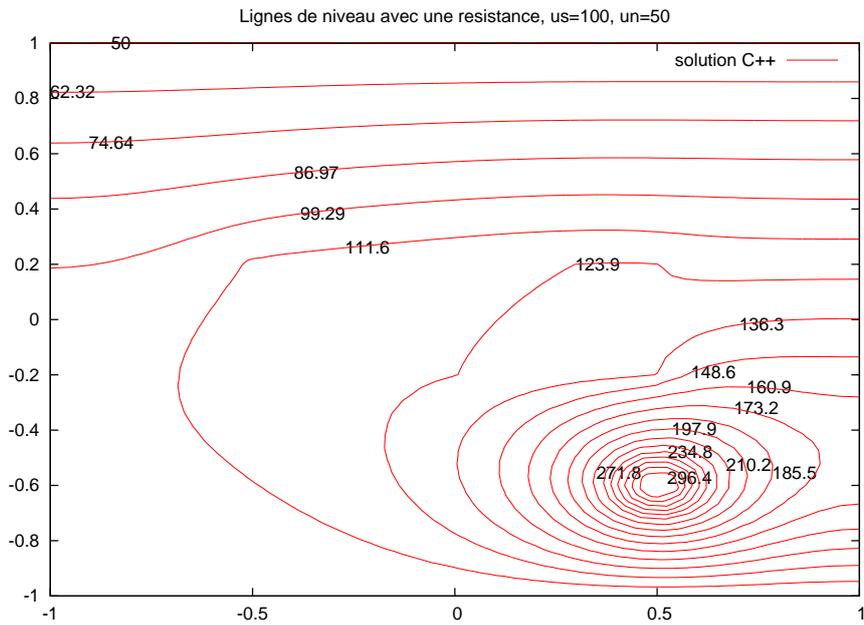


FIGURE 9 – Lignes de niveau, une résistance avec $us=100$ et $un=50$

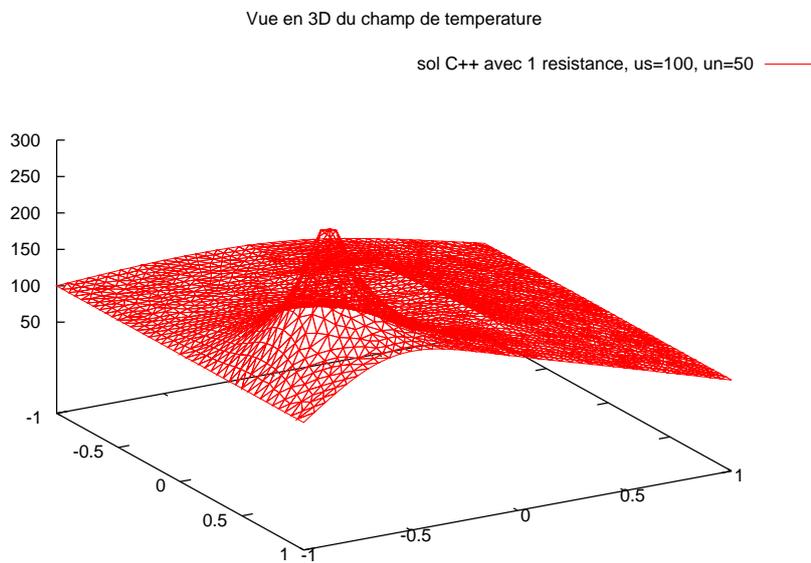


FIGURE 10 – Une résistance avec $us=100$ et $un=50$

4.1.2 Problème Pb_r

Nous résolvons maintenant le problème Pb_r qui correspond à $f(x, y) = f_r = 25000$ sur une résistance et les conditions $u_s = 0$ et $u_n = 0$ respectivement sur Γ_1 et Γ_3 .

Nous obtenons les résultats présentés sur les figures 11, 12 et 13.

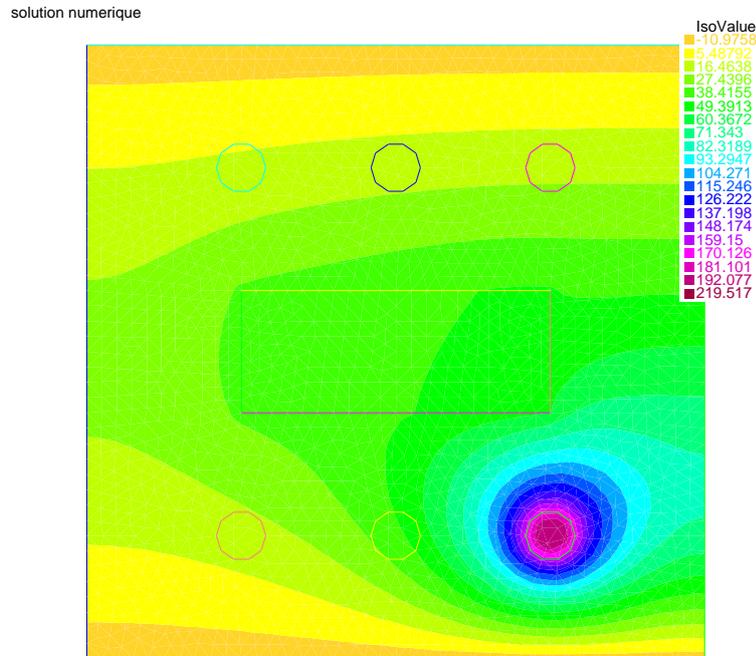


FIGURE 11 – Solution FreeFem avec une résistance, $u_n=us=0$

Les lignes de niveaux sont visibles sur la figure 12.

La figure 13 présente une vue en 3D.

La différence entre les solutions C++ et FreeFem++, calculée par le programme C++ est inférieure à 5.10^{-8} . Cette différence est représentée sur la figure 14 réalisée par le programme FreeFem++.

Le programme C++ permet de stocker les solutions aux problèmes Pb , Pb_0 et Pb_r et de vérifier la relation qui a été démontrée entre les solutions U , U_0 et U_r , soit : $U = U_0 + U_r$.

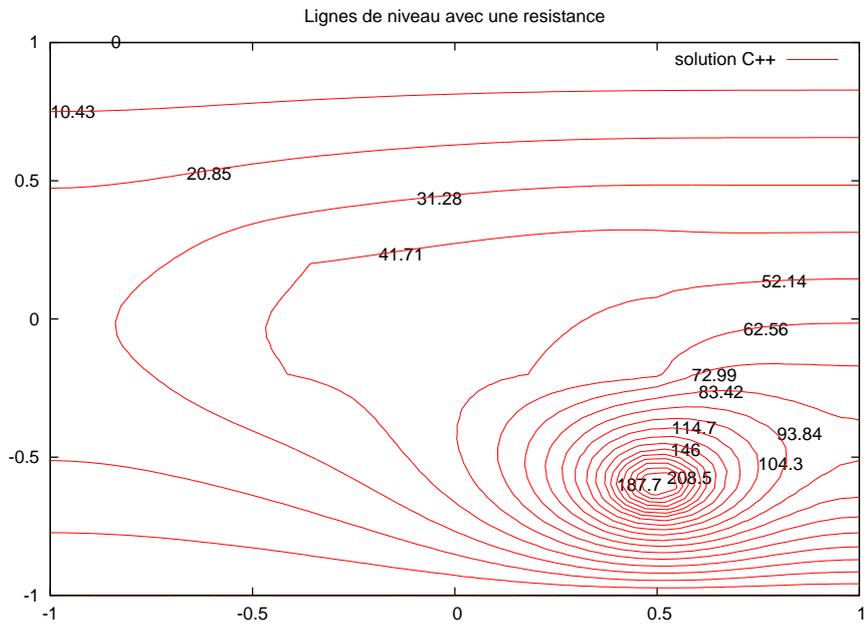


FIGURE 12 – Une resistance $un=us=0$

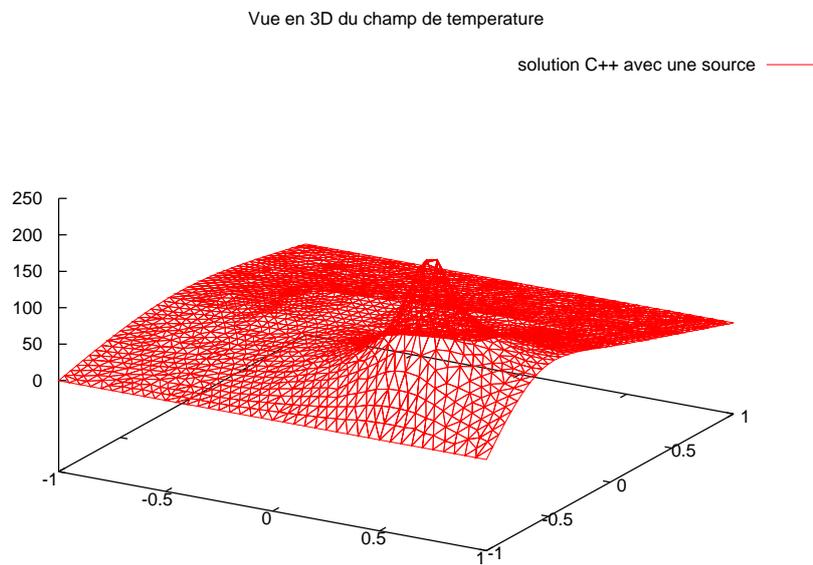


FIGURE 13 – Une resistance $un=us=0$

différence entre les deux solutions

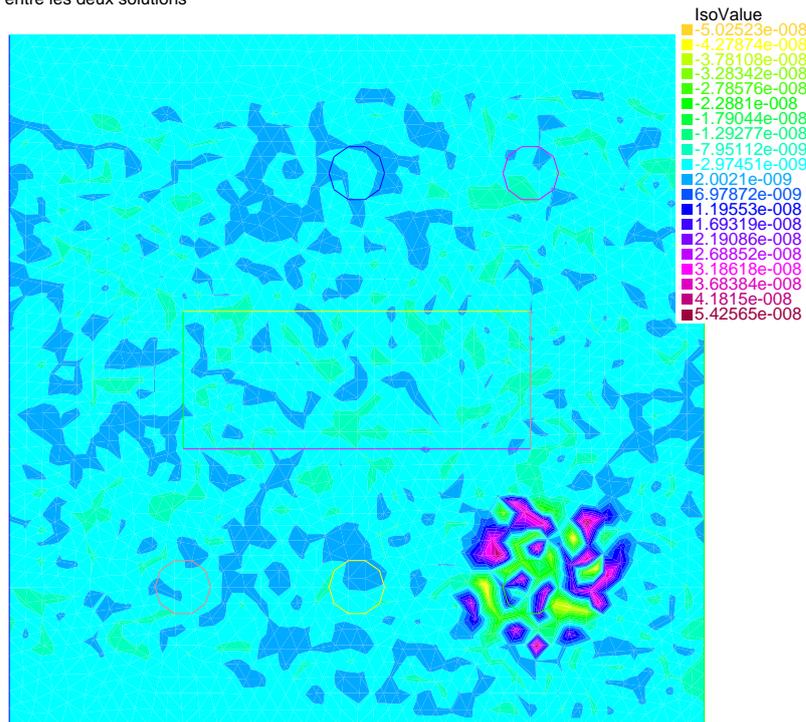


FIGURE 14 – Différence entre les solutions C++ et FreeFem++

4.2 Six résistances chauffantes

Les programmes FreeFem++ et C++ vont résoudre ici le problème Pb .

Ce problème correspond à $f(x, y) = f_r = 25000$ sur les six résistances avec les conditions $u_s = 100$ et $u_n = 50$ respectivement sur Γ_1 et Γ_3 .

Nous avons ajouté dans le programme C++ une fonction `"int NumduTriangle(const Mesh2 &TTh, R2 &P) { ...}"` qui détermine, à l'aide des coordonnées barycentriques, et renvoie le numéro du triangle auquel le point P appartient.

A partir du label de ce triangle, nous pouvons déterminer la région correspondant aux résistances ou à la pièce. Ceci permet d'éviter la recherche "à la main" de ces régions dans le fichier `mesh.msh` du maillage généré par FreeFem++.

Le numéro du triangle et l'utilisation des coordonnées barycentriques permettront ensuite de calculer la valeur du champ de température au point P.

Les résultats obtenus sont présentés sur les figures 15, 16, 17 et 18.

solution cpp

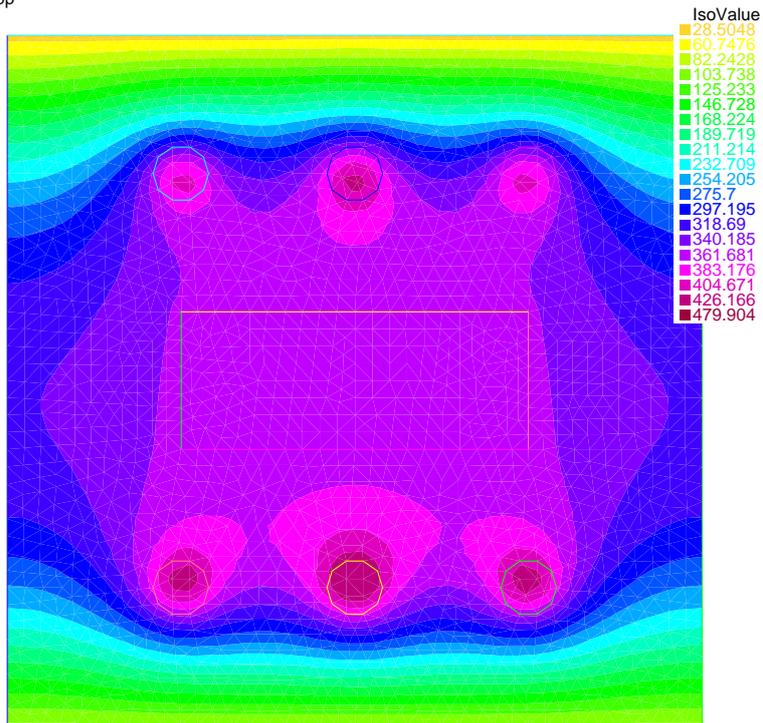


FIGURE 15 – Solution avec 6 résistances, $un=100$, $us=50$ FreeFem++

différence entre les deux solutions

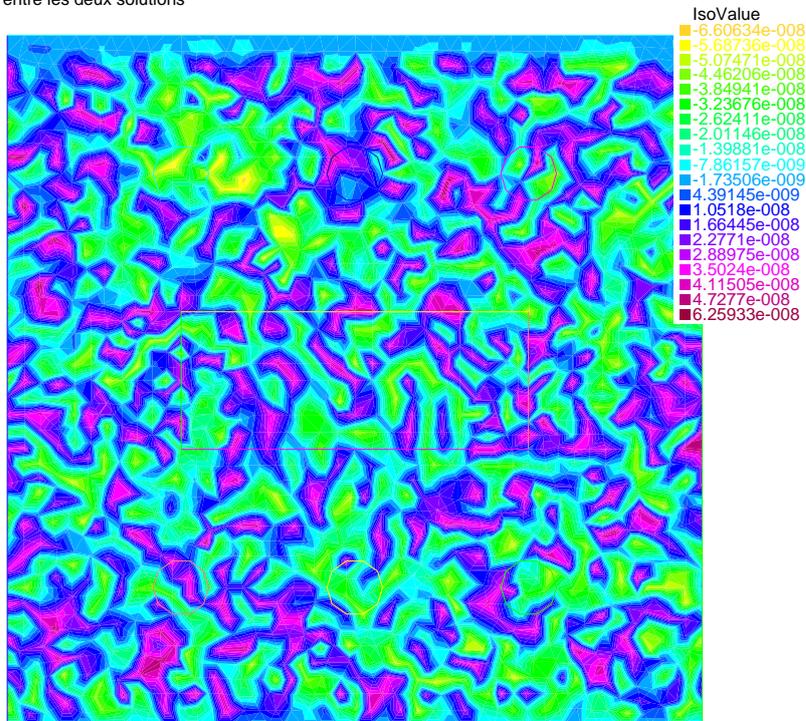


FIGURE 16 – Différence entre les solutions C++ et FreeFem++

Vue en 3D du champ de température

solution C++ avec 6 resistances

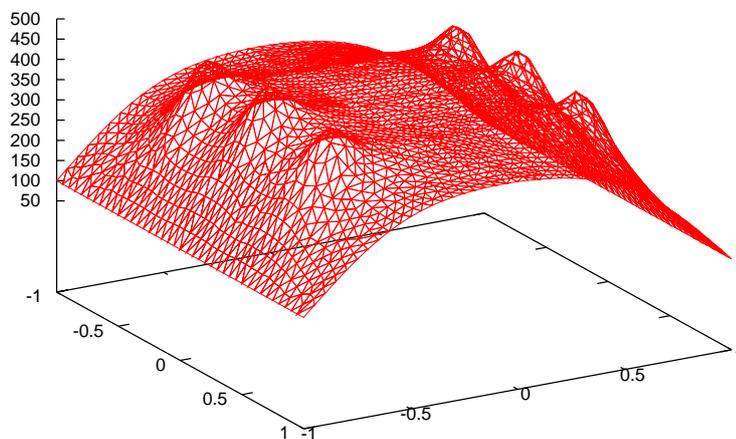


FIGURE 17 – Solution avec 6 résistances, $un=100$, $us=50$

Lignes de niveau avec 6 resistances

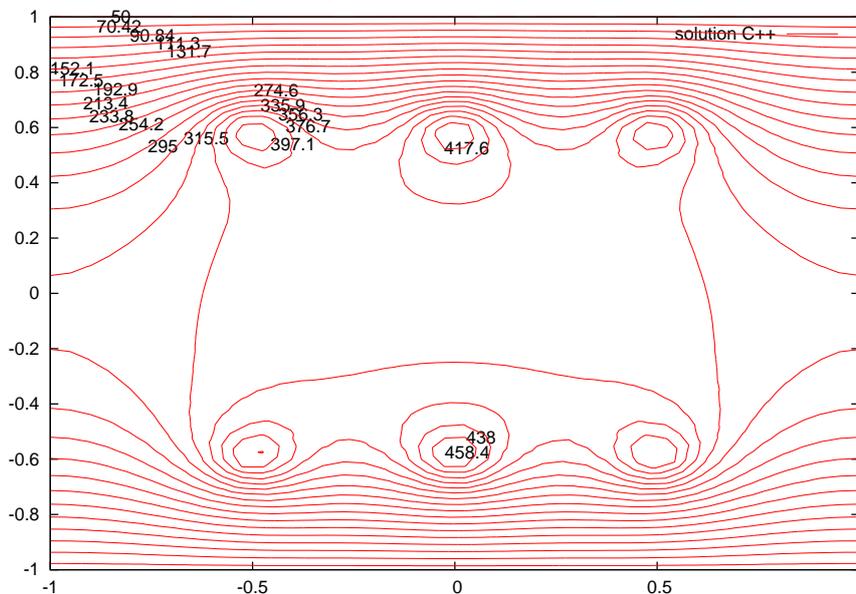


FIGURE 18 – Solution avec 6 résistances, $un=100$, $us=50$

5 Sensibilité aux paramètres

5.1 Température fonction de la source

Dans le cadre du problème précédent, nous allons étudier les variations de la température u en fonction du paramètre $q = f_r$. Les autres paramètres sont fixés : $u_s = 100$, $u_n = 50$ et $k = 10$.

L'étude numérique sera traitée par deux méthodes : les différences finies et la différentiation automatique.

5.1.1 Etude par différences finies

A partir du programme C++ utilisé jusqu'à présent, nous pouvons calculer les valeurs de u pour des valeurs de f_r égales à $q + dq$ ou $q - dq$, puis évaluer le rapport $\frac{u(q + dq) - u(q - dq)}{2dq}$.

Nous faisons ce calcul au point $P(0, 0)$, le centre de la pièce. Le programme écrit alors les résultats dans un fichier.

Pour $q = 25000$, nous obtenons $u \simeq 370$ et les résultats suivants :

valeur de dq	dérivée approchée
$q/10$	0.0117979
$q/10^2$	0.0117979
$q/10^3$	0.0117979
$q/10^4$	0.0117979
$q/10^5$	0.0117979

Nous pouvons donc en déduire :

$$\left(\frac{\partial u}{\partial q}\right)_{q=25000} \simeq 0.0118$$

D'après le tableau, il semble que cette dérivée est indépendante de q . Nous allons le vérifier plus loin.

5.1.2 Etude par différentiation automatique

Nous ajoutons au programme les fichiers `ddouble.hpp` et `ddouble_GC.hpp` afin d'utiliser la classe `ddouble` qui a pour membre un tableau de deux réels : `double val[2];`.

`val[0]` est la valeur de u et `val[1]` sa dérivée par rapport à q .

L'intérêt est de pouvoir calculer ces deux valeurs simultanément et en n'importe quel point. Nous obtenons de plus les mêmes résultats graphiques pour le champ des dérivées que ceux obtenus pour le champ des températures.

Les figures 19, 20 et 21 nous présentent ces résultats calculés pour $q = 25000$. Nous constatons que la dérivée est pratiquement constante dans la pièce.

solution cpp

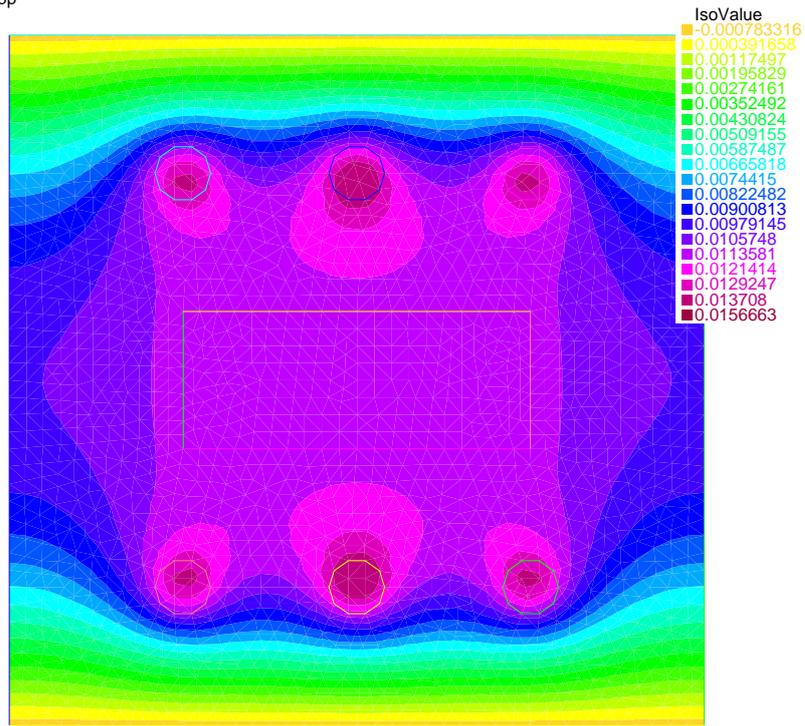


FIGURE 19 – Dérivée du champ de température par rapport à q

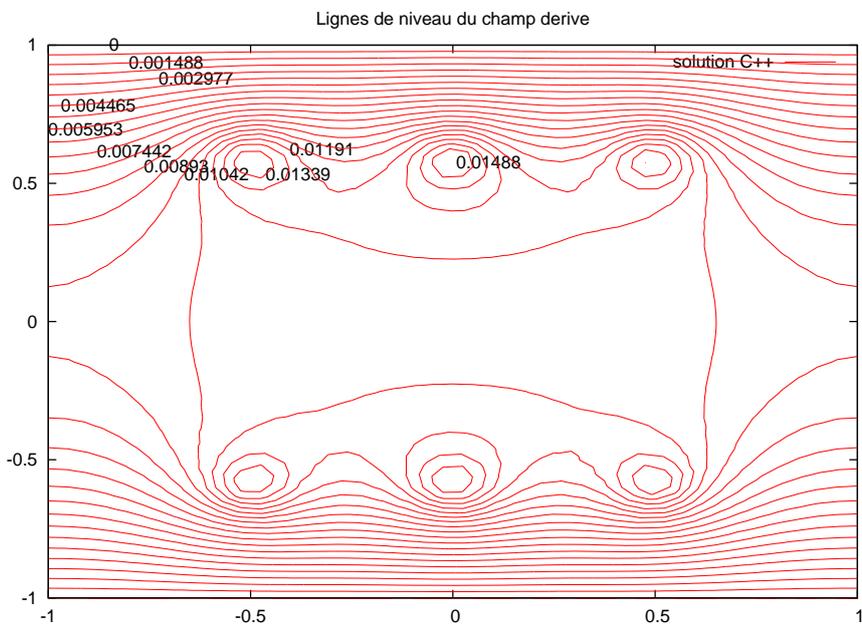


FIGURE 20 – Lignes de niveau du champ dérivé par rapport à q

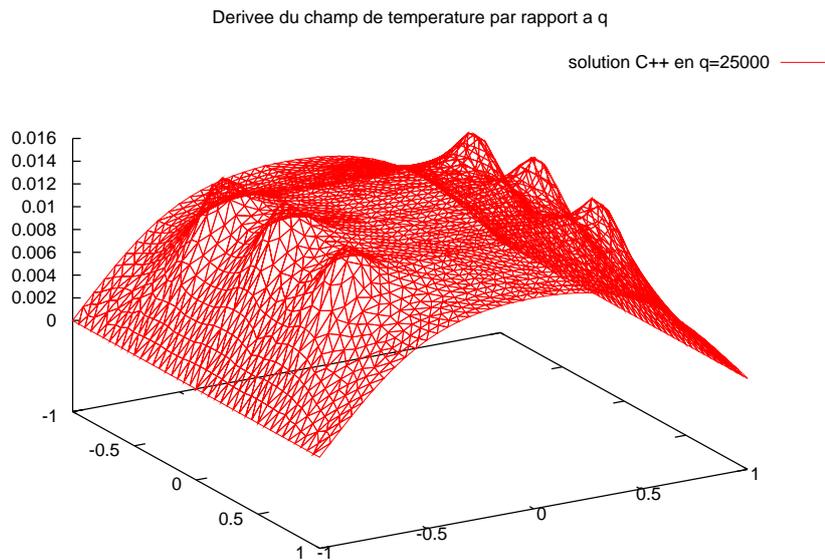


FIGURE 21 – *Vue en 3D du champ des dérivées par rapport à q*

Le programme précédent donne les mêmes résultats pour des valeurs de q quelconques, différentes de 0. Il nous permet de vérifier la relation $u = \lambda q + u_0$.

Plus généralement, nous supposons : $u = \lambda(x, y, k)q + u_0$

Au point $P(0, 0)$, avec $k = 10$, $u_s = 100$ et $u_n = 50$, cette équation s'écrit :

$$u \simeq 0,0118q + 75$$

La figure 22 nous présente cette droite.

Remarque : nous vérifions à nouveau la relation déjà démontrée, à savoir que la solution du problème Pb est la somme des solutions des problèmes Pb_r et Pb_0 , soit :

$$u = u_r + u_0$$

Cette relation $u = \lambda(x, y, k)q + u_0$ va être précisée encore davantage.

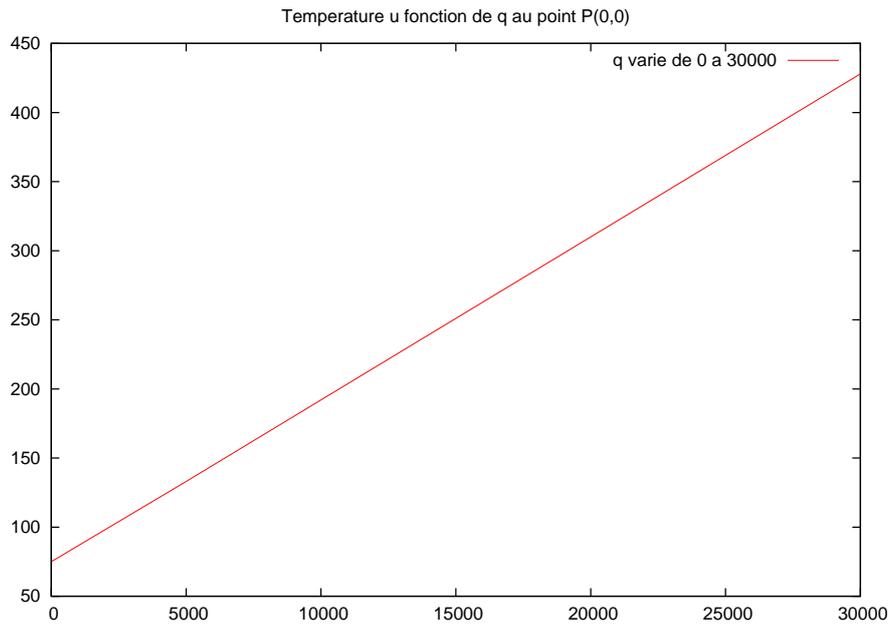


FIGURE 22 – u fonction de q au centre de la pièce

5.2 Température fonction des conditions limites

Nous étudions la variation de u par rapport à la condition limite u_n sur Γ_3 . La même étude pourrait être menée pour u_s sur Γ_1 .

Les autres paramètres sont fixés, $q = 25000$, $k = 10$, $u_s = 100$, et nous obtenons les résultats suivants :

valeur de u_n	valeur de u
0	344.947
10	349.947
20	354.947
30	359.947
40	364.947
50	369.947
60	374.947
70	379.947
80	384.946
90	389.946
100	394.946

La représentation graphique est donnée par la figure 23.

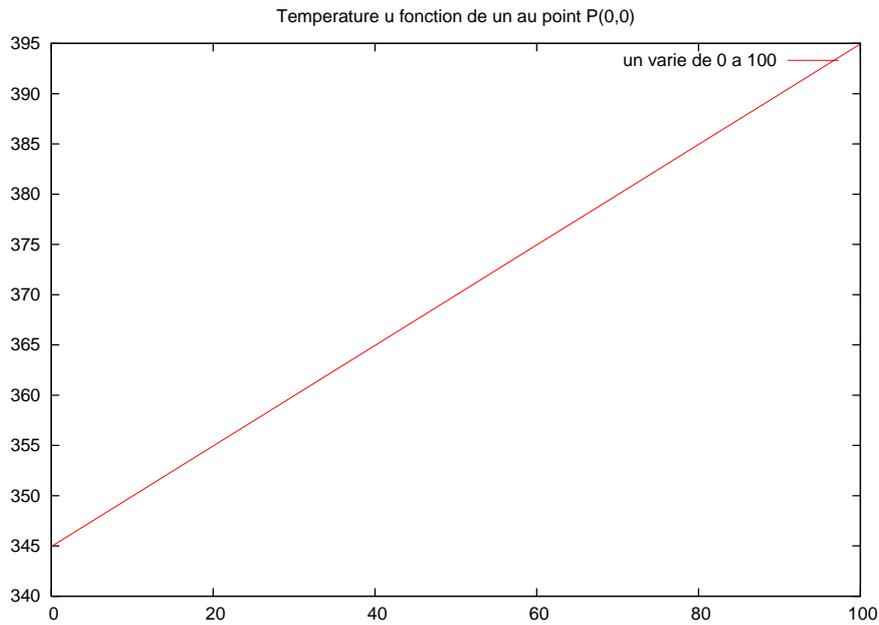


FIGURE 23 – u fonction de un au centre de la pièce

Nous obtenons ce qui était attendu, à savoir que la température u au centre de la pièce est une fonction affine de u_n . Il en serait de même pour u fonction de u_s .

Nore relation s'écrit donc maintenant, au point $P(0, 0)$, sous la forme plus précise :

$$u = \lambda(0, 0, k)q + \frac{u_s + u_n}{2}$$

Il nous reste à examiner le terme $\lambda(0, 0, k)$ que nous notons plus simplement $\lambda(k)$.

C'est l'objet de la section suivante.

5.3 Température fonction de la conductivité de la pièce

Tous les autres paramètres sont fixés : $q = 25000$, $u_s = 100$, $u_n = 50$.

Un premier examen nous laisse penser que λ peut être approximativement de la forme

$$\lambda(k) = a + \frac{b}{k + c}$$

Notre programme nous permet de calculer les différentes valeurs et d'obtenir un tableau donnant les valeurs de λ en fonction de la conductivité k , ou un tableau donnant les valeurs de u en fonction de la conductivité k . Une partie de ces tableaux est présentée ci-dessous :

valeur de k	valeur de λ
1	0.01206
2	0.01196
3	0.01191
4	0.01187
5	0.01185
6	0.01183
7	0.01182
8	0.01181
9	0.0118
10	0.0118

valeur de k	valeur de u
1	376.541
2	374.006
3	372.675
4	371.848
5	371.282
6	370.87
7	370.557
8	370.31
9	370.111
10	369.947
12	369.692
13	369.591
16	369.357
20	369.147

L'étude du premier tableau nous permet d'obtenir pour les coefficients a , b , c :

$$a \simeq 0.01174 \quad b \simeq 6.4 \times 10^{-4} \quad c \simeq 1$$

Puis, en multipliant par $q = 25000$:

$$\lambda q \simeq 293.5 + \frac{16}{k + 1}$$

Et finalement, en ajoutant $u_0 = \frac{u_s + u_n}{2}$

$$u \simeq 368.5 + \frac{16}{k+1}$$

La formule générale, au point $P(0,0)$, donnant u en fonction des paramètres q, k, u_s, u_n étant :

$$u \simeq \left(0.01174 + \frac{6.4 \times 10^{-4}}{k+1}\right) q + \frac{u_s + u_n}{2}$$

La figure 24 nous permet de vérifier notre approximation.

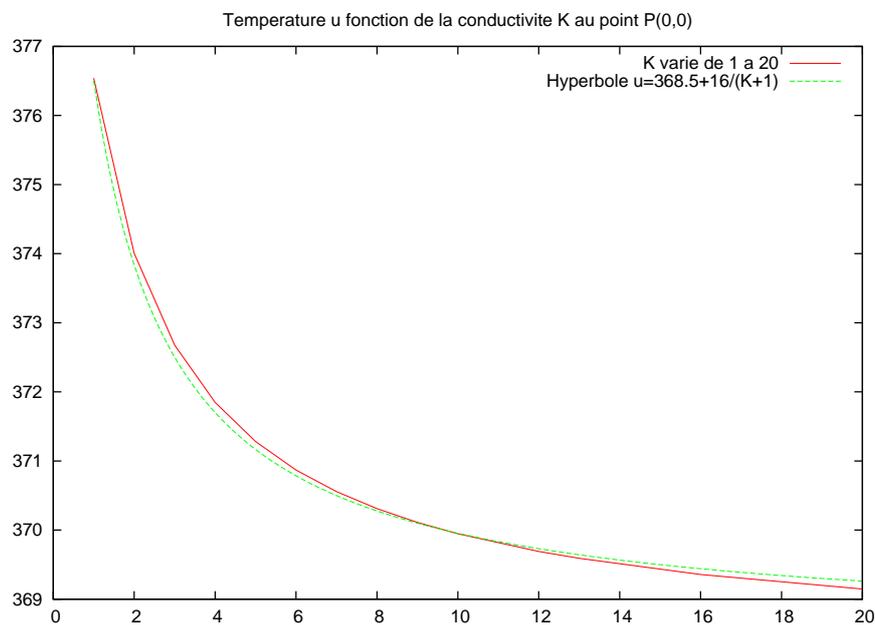


FIGURE 24 – u fonction de la conductivité de la pièce au centre

6 Conclusion

Le programme C++ final permet de résoudre les différents problèmes P_b , P_{b_r} et P_{b_0} . Il est possible de choisir les valeurs u_s et u_n , la valeur de la conductivité de la pièce, la valeur de la source f_r et le fichier du maillage.

Afin d'avoir une présentation plus agréable, un programme écrit en Java propose dans une fenêtre ces différents choix, puis lance le programme C++ qui écrit les résultats dans des fichiers exploitables ensuite par Gnuplot pour une visualisation graphique. L'apport d'OpenGL, avec le fichier `glplotiso.hpp` permet également d'obtenir une vue en 3D que l'on peut manipuler.

Enfin, la valeur de la source, ici les résistances, n'étant pas toujours une fonction constante, le programme C++ qui lit les données dans un fichier, doit pouvoir interpréter la lecture d'une écriture du type $f(x, y)$, et la transformer en une fonction exploitable par le programme. Ceci se fait avec l'apport du fichier `calculette.hpp` dans notre programme.

La figure 25 nous montre un aperçu de la fenêtre qui apparaît au lancement du programme java.

La figure 26 nous montre une vue en 3D du résultat extrait de l'animation créée avec OpenGL.

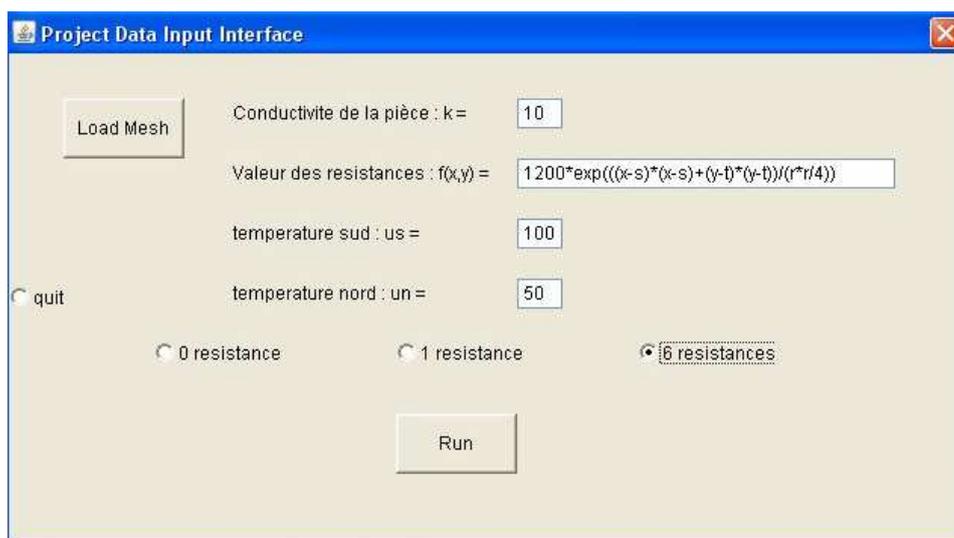


FIGURE 25 – Fenêtre java

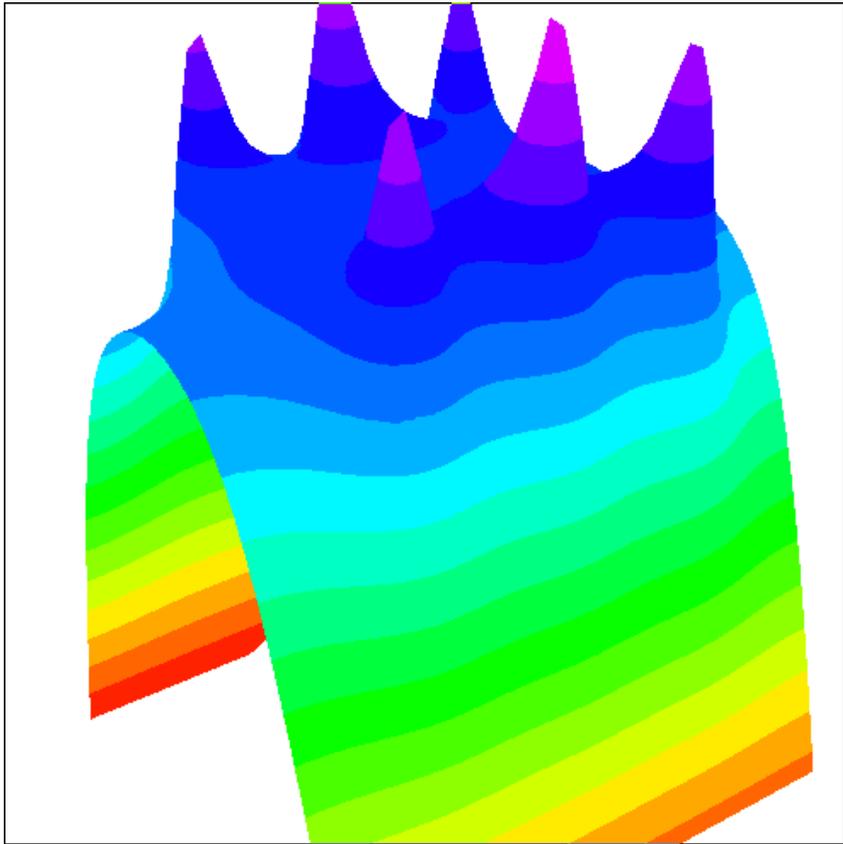


FIGURE 26 – *Résultat avec OpenGL*

7 Annexe : les programmes

Les programmes sont répartis dans divers dossiers en fonction des questions auxquelles ils peuvent répondre. Le fichier principal est `EF2d-2010.cpp`. Les fichiers `Mesh2d.hpp` et `Mesh2d.cpp` permettent de définir toutes les classes nécessaires au maillage. Les fichiers `RNM` sont regroupés dans un dossier nommé "RNM".

La complexité des programmes évolue avec les questions posées.

1. Dossier "0resist"
 - Résoud le problème sans source avec conditions aux limites.
 - Comparaison avec la solution `FreeFem++` calculée par le programme `four0.edp`.
2. Dossier "1resist"
 - Résoud les problèmes avec source, (1 résistance), Pb et Pb_r .
 - Vérification de la relation $U = U_0 + U_r$.
 - idem avec la solution `FreeFem++` calculée par le programme `four1.edp`.
3. Dossier "6resist"
 - Résoud le problème avec source, (6 résistances).
 - Une fonction implémentée dans le fichier `EF2d-2010.cpp` permet de récupérer le numéro du triangle contenant un point donné à l'aide des coordonnées barycentriques.
 - Comparaison avec la solution calculée par le programme `FreeFem++` nommé `four6.edp`.
4. Dossier "diff_finies"
 - Calcul de la dérivée de u par différence finies avec le programme précédent.
5. Dossier "diff_auto"
 - Calcul de la dérivée de u par différenciation automatique.
 - Utilisation de la class `ddouble` implémentée dans le fichier `ddouble.hpp`.
 - Le gradient conjugué est programmé dans un fichier fonctionnant avec la classe `ddouble`.
6. Dossier "prog_fin"
 - Programme final qui résoud le problème avec différents paramètres.
 - Utilisation d'une interface java permettant d'entrer tous les paramètres.
 - Interprétation d'une formule à l'aide du fichier `calcullette.hpp` afin de pouvoir entrer dans l'interface une fonction source analytique.
7. Dossier "prog_fin_GL"
 - Même programme que le précédent avec l'utilisation d'"OpenGL" pour une visualisation et une animation de la solution en 3D ;