



Laboratoire  
Mécanique  
Lille

UNIVERSITÉ DE LILLE 1

LABORATOIRE MÉCANIQUE DE LILLE

MASTER 1 MÉCANIQUE

MÉMOIRE DE STAGE

---

# Étude théorique et numérique du processus de fusion en régime de convection dans une cavité

---

*Etudiants :*

El-Mehdi GOUDA

Mouhcine TAHRI

*Maître de stage :*

Dr. Enrico CALZAVARINI

Maître de Conférences

Mai - Juin 2017



Université  
de Lille  
1 SCIENCES  
ET TECHNOLOGIES

## Résumé

Nous nous sommes intéressés dans ce stage à étudier le phénomène de diffusion dans deux types de cavités semi-circulaire et rectangulaire dans les cas de deux dimensions et de trois dimensions, ceci revient à résoudre l'équation de Laplace dans ces deux cavités, nous avons ensuite simulé ces résultats trouvés sur Python et puis nous les avons comparés avec ceux obtenus grâce à FreeFem++.

Nous avons ensuite étudié le phénomène de convection avec changement de phase (fusion) sur les bords des deux cavités, puis les avons simulé grâce à un programme C.

**Mots Clés** : (*Laplace, Python, C, FreeFem++, Convection, Diffusion, Séries de Fourier*)

## Abstract

We have studied during this internship the heat diffusion phenomena in two types of cavities rectangular and semicircular ones, this comes down to resolve the Laplace Equation in these two cavities in two and three dimensions, we used then Python to simulate these solutions and we compared them with the ones obtained from FreeFem++ simulations.

We have then moved to the natural convection with phase change (fusion) in both cavities, and we studied it using the same boundary conditions, we created then a movie showing the simulation of this phenomenon thanks to a C program.

**Keywords** : (*Laplace, Python, C, FreeFem++, Convection, Diffusion, Fourier series*)

## Remerciements

Nous tenons à remercier notre tuteur Dr. Enrico CALZAVARINI Enseignant-Chercheur à l'Université de Lille 1 qui est, par son expérience, son don d'écoute et sa tolérance, a guidé notre activité tout en préservant notre esprit d'initiative, durant toute la période de notre stage.

Nous adressons aussi nos remerciements à l'ensemble des enseignants du Master 1 Mécanique à l'Université de Lille 1 et, notamment, à notre responsable de formation M<sup>me</sup> Silvia HIRATA qui nous a confié ce sujet de stage qui était en totale adéquation avec nos attentes.

Nous remercions toutes les personnes qui nous ont aidées à réaliser ce stage et en particulier M. Babak RABBANIPOUR qui nous a aidés à vérifier nos résultats et à élaborer la partie concernant la fusion de notre stage.

# Table des matières

<b>1</b>	<b>Introduction : Description du problème</b>	<b>4</b>
1.1	Équation de la chaleur . . . . .	4
1.1.1	Hypothèses du problème . . . . .	4
<b>2</b>	<b>Résolution de l'équation de diffusion de Laplace</b>	<b>6</b>
2.1	Problème en coordonnées cartésiennes en 2D . . . . .	6
2.1.1	Définition des conditions aux limites . . . . .	6
2.1.2	Résolution par la méthode de séparation des variables	8
2.2	Problème en coordonnées cartésiennes en 3D . . . . .	12
2.2.1	Définition des conditions aux limites . . . . .	12
2.2.2	Résolution par la méthode de séparation des variables	14
2.3	Equation de Laplace en coordonnées polaires (2D) . . . . .	18
2.3.1	Description du Problème : . . . . .	18
2.3.2	Résolution de l'équation par la méthode de séparations de variables : . . . . .	18
2.4	Équation de Laplace en coordonnées sphériques 3D . . . . .	22
2.5	Etude numérique en Python . . . . .	23
2.5.1	Etude numérique du problème de la cavité rectangu- laire (2D) : . . . . .	23
2.5.2	Étude numérique et graphique du problème de la ca- vité semi-circulaire (2D) : . . . . .	26
2.5.3	Étude numérique et graphique du problème 3D en co- ordonnées cartésiennes . . . . .	30
2.6	Étude numérique avec FreeFem++ . . . . .	34
2.6.1	L'utilité du logiciel de FreeFem++ . . . . .	34

2.6.2	Résolution FreeFem++ de Laplace 2D en coordonnées cartésiennes . . . . .	34
2.6.3	Résolution FreeFem de Laplace 2D en coordonnées polaires . . . . .	35
2.6.4	Résolution FreeFem++ de Laplace 3D en coordonnées cartésiennes . . . . .	39
<b>3</b>	<b>Système convectif avec changement de phase</b>	<b>43</b>
3.1	Mise en équation : . . . . .	43
3.2	Résolution numérique du problème de changement d'état en C	45
	<b>Conclusion</b>	<b>51</b>
	<b>Table des Figures</b>	<b>53</b>
	<b>Table des Programmes</b>	<b>54</b>
	<b>Bibliographie</b>	<b>55</b>

# Chapitre 1

## Introduction : Description du problème

### 1.1 Équation de la chaleur

En général l'équation de la chaleur s'écrit de la façon suivante :

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (\lambda \nabla T) + q$$

avec  $q$  : la source de chaleur en Joule J

$\lambda$  : La conductivité thermique en W/m.K

$c_p$  : La chaleur spécifique à pression constante en  $m^2 \cdot s^{-2} K^{-1}$

$\rho$  : La masse volumique en  $kg \cdot m^{-3}$

#### 1.1.1 Hypothèses du problème

La convection naturelle nécessite l'application des conditions sur la nature de l'écoulement.

On va considérer que l'écoulement est permanent et que la conductivité ther-

mique est constante et en absence de la source de chaleur on aboutit à une équation à résoudre qui est la suivante :

$$\Delta T = 0$$

Cette équation s'appelle l'**Equation de Laplace**.

## Chapitre 2

# Résolution de l'équation de diffusion de Laplace

### 2.1 Problème en coordonnées cartésiennes en 2D

$$\Delta T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

#### 2.1.1 Définition des conditions aux limites

$$\begin{aligned}0 &\leq x \leq L \\0 &\leq y \leq H \\T(x, 0) &= T_s \\T(x, H) &= T_a \\T(0, y) &= T_s \\T(L, y) &= T_s\end{aligned}$$

Le problème peut être décomposé en quatre problèmes ayant chacun une seule condition non-homogène plutôt que quatre, les quatre problèmes seront



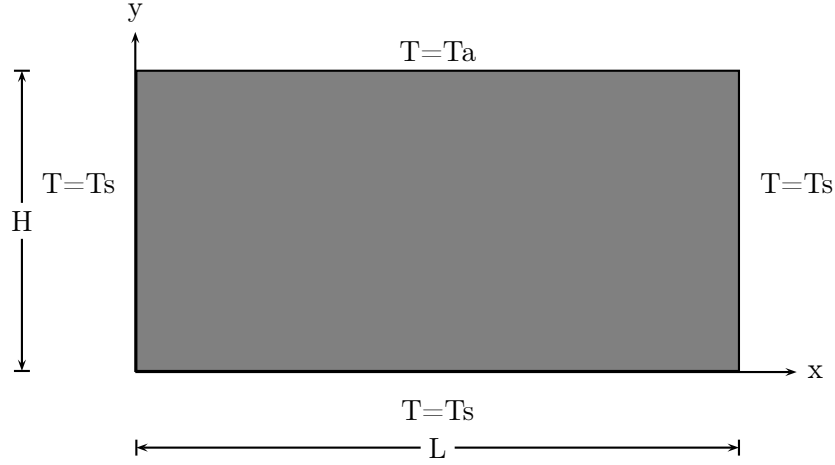


FIGURE 2.1 – Conditions aux limites du problème total en 2D

ensuite regroupés pour former la solution finale par le principe de superposition.

Les quatre problèmes sont :

Problème [1] :

$$\begin{aligned} \frac{\partial^2 T_1}{\partial x^2} + \frac{\partial^2 T_1}{\partial y^2} &= 0 \\ T_1(x, 0) = T_s \quad \text{et} \quad T_1(x, H) &= 0 \quad \forall x \in [0, L] \\ T_1(0, y) = 0 \quad \text{et} \quad T_1(L, y) &= 0 \quad \forall y \in [0, H] \end{aligned}$$

Problème [2] :

$$\begin{aligned} \frac{\partial^2 T_2}{\partial x^2} + \frac{\partial^2 T_2}{\partial y^2} &= 0 \\ T_2(x, 0) = 0 \quad \text{et} \quad T_2(x, H) &= T_a \quad \forall x \in [0, L] \\ T_2(0, y) = 0 \quad \text{et} \quad T_2(L, y) &= 0 \quad \forall y \in [0, H] \end{aligned}$$

Problème [3] :

$$\begin{aligned} \frac{\partial^2 T_3}{\partial x^2} + \frac{\partial^2 T_3}{\partial y^2} &= 0 \quad (1) \\ T_3(x, 0) = 0 \quad \text{et} \quad T_3(x, H) &= 0 \quad \forall x \in [0, L] \end{aligned}$$

$$T_3(0, y) = T_s \quad \text{et} \quad T_3(L, y) = 0 \quad \forall y \in [0, H]$$

Problème [4] :

$$\begin{aligned} \frac{\partial^2 T_4}{\partial x^2} + \frac{\partial^2 T_4}{\partial y^2} &= 0 \\ T_4(x, 0) = 0 \quad \text{et} \quad T_4(x, H) &= 0 \quad \forall x \in [0, L] \\ T_4(0, y) = 0 \quad \text{et} \quad T_4(L, y) &= T_s \quad \forall y \in [0, H] \end{aligned}$$

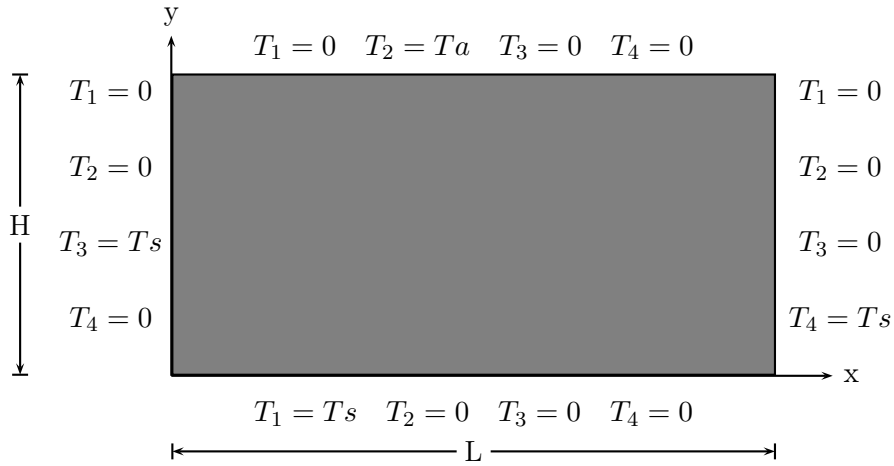


FIGURE 2.2 – Conditions aux limites des quatre problèmes en 2D

Si  $T_i$  est une solution du problème [i] pour  $i = 1; 2; 3; 4$ , alors  $T = T_1 + T_2 + T_3 + T_4$  est une solution du problème de départ.

### 2.1.2 Résolution par la méthode de séparation des variables

$$T_1(x, y) = X(x)Y(y)$$

En injectant l'expression dans l'équation, nous obtenons :

$$\begin{aligned} X''Y + XY'' &= 0 \\ \Rightarrow \frac{X''}{X} &= -\frac{Y''}{Y} \end{aligned}$$

où  $X''$  est la dérivée seconde de  $X$  par rapport à  $x$  et  $Y$  est la dérivée seconde de  $Y$  par rapport à  $y$ . Dans la dernière équation, le terme de gauche est une fonction de  $x$  seulement, alors que le terme de droite est une fonction de  $y$  seulement. Donc, on peut en déduire :

$$\frac{X''}{X} = -\frac{Y''}{Y} = \lambda$$

Nous obtenons ainsi le système suivant :

$$(2) \begin{cases} X'' - \lambda X = 0 \\ Y'' + \lambda Y = 0 \end{cases}$$

En utilisant les conditions aux limites, le problème (1) devient :

$$\begin{aligned} T_1(x, H) = X(x)Y(H) = 0 \quad \forall x \in [0, L] &\Rightarrow Y(H) = 0 \\ T_1(0, y) = X(0)Y(y) = 0 \quad \forall y \in [0, H] &\Rightarrow X(0) = 0 \\ T_1(L, y) = X(L)Y(y) = 0 \quad \forall y \in [0, H] &\Rightarrow X(L) = 0 \end{aligned}$$

D'où, le système (2) devient :

$$X'' - \lambda X = 0 \quad \text{avec } X(0) = 0, X(H) = 0 \quad (2.1)$$

$$Y'' + \lambda Y = 0 \quad \text{avec } Y(0) = 0 \quad (2.2)$$

Si  $\lambda \geq 0$ , nous obtenons que  $X \equiv 0$  et nous pouvons exclure ces valeurs pour  $\lambda$  parce que nous voulons des solutions non triviales. Si  $\lambda = -\mu^2 < 0$ , alors la solution générale de  $X'' - \lambda X = 0$  est  $X(x) = A\cos(\mu x) + B\sin(\mu x)$ . Comme  $X(0) = A = 0$  et  $X(L) = A\cos(\mu L) + B\sin(\mu L) = 0$ , il faut alors que  $B\sin(\mu L) = 0$ . Comme nous cherchons à déterminer des solutions non triviales et que  $A = 0$ , nous pouvons supposer que  $B = 0$ . De ceci, nous pouvons en déduire que :

$$\begin{aligned} \sin(\mu L) = 0 &\Rightarrow \mu = \frac{n\pi}{L}, \lambda_n = -\left(\frac{n\pi}{L}\right)^2 \\ \text{et } X_n(x) &= B_n \sin\left(\frac{n\pi x}{L}\right) \text{ où } n \in \mathbb{Z}, n \neq 0 \end{aligned}$$

Par symétrie de la fonction  $\sin$  et la fonction  $\exp$ , nous pouvons nous restreindre au cas où  $n \in \mathbb{N}^*$ . On a donc la solution du système (2) qui devient :

$$\begin{cases} X_n(x) = B_n \sin\left(\frac{n\pi x}{L}\right) \\ Y_n(y) = C \exp\left(\frac{n\pi y}{L}\right) + D \exp\left(-\frac{n\pi y}{L}\right) \end{cases}$$

Nous avons aussi :

$$Y_n(H) = 0 \Rightarrow C \exp\left(\frac{n\pi H}{L}\right) + D \exp\left(-\frac{n\pi H}{L}\right) = 0 \Rightarrow D = -C \exp\left(\frac{2n\pi H}{L}\right)$$

En substituant cette expression dans la solution  $Y_n$ , on a :

$$Y_n(y) = C \exp\left(\frac{n\pi y}{L}\right) - C \exp\left(\frac{2n\pi y}{L}\right) \exp\left(\frac{-n\pi y}{L}\right) \quad (2.3)$$

$$= C \exp\left(\frac{n\pi N}{L}\right) \left[ \exp\left(\frac{n\pi(y-H)}{L}\right) + \exp\left(-\frac{n\pi(y-H)}{L}\right) \right] \quad (2.4)$$

$$= 2C \exp\left(\frac{n\pi N}{L}\right) \sinh\left(\frac{n\pi(y-H)}{L}\right) \quad (2.5)$$

Donc la solution du problème (1) peut s'écrire :

$$a_n \sin\left(\frac{n\pi x}{L}\right) \sinh\left(\frac{n\pi(y-H)}{L}\right)$$

Avec  $a_n = 2B_n C \exp\left(\frac{n\pi N}{L}\right)$

Par principe de superposition, nous trouvons la solution générale du problème (1) :

$$(3) \quad T_1(x, y) = \sum_{n=0}^{+\infty} a_n \sin\left(\frac{n\pi x}{L}\right) \sinh\left(\frac{n\pi(y-H)}{L}\right)$$

En utilisant les conditions aux limites sur  $T_1$ , on a (3) est solution si et seulement si :[1]

$$T_1(x, 0) = \sum_{n=0}^{+\infty} a_n \sin\left(\frac{n\pi x}{L}\right) \sinh\left(\frac{n\pi(y-H)}{L}\right) = T_0 \quad \forall x \in [0, L]$$

C'est-à-dire que,  $T_1(x, 0)$  est la série de Fourier impaire de la fonction constante  $T_0$ , et conséquemment :

$$a_n = \left( \frac{2T_0}{L \sinh(-n\pi H/L)} \right) \int_0^L \sin\left(\frac{n\pi x}{L}\right) dx \quad \forall n \in \mathbb{N}^* \quad (2.6)$$

$$= \left( \frac{2T_0 \left(\frac{L}{n\pi}\right)}{L \sinh(-n\pi H/L)} \right) (\cos(0) - \cos(n\pi)) \quad (2.7)$$

$$= \left( \frac{2T_0}{n\pi \sinh(-n\pi H/L)} \right) (1 - \cos(n\pi)) \quad (2.8)$$

En procédant de la même façon, nous obtenons les solutions des problèmes [2], [3] et [4]. Ce qui nous obtenons :

Solution du problème [2] :

$$T_2(x, y) = \sum_{n=0}^{+\infty} b_n \sin\left(\frac{n\pi x}{L}\right) \sinh\left(\frac{n\pi y}{L}\right)$$

Avec :

$$b_n = \left( \frac{2T_a}{L \sinh(n\pi H/L)} \right) \int_0^L \sin\left(\frac{n\pi x}{L}\right) dx \quad \forall n \in \mathbb{N}^* \quad (2.9)$$

$$= \left( \frac{2T_0 \left(\frac{L}{n\pi}\right)}{L \sinh(-n\pi H/L)} \right) (\cos(0) - \cos(n\pi)) \quad (2.10)$$

$$= \left( \frac{2T_0}{n\pi \sinh(-n\pi H/L)} \right) (1 - \cos(n\pi)) \quad (2.11)$$

Solution du problème [3] :

$$T_3(x, y) = \sum_{n=0}^{+\infty} c_n \sin\left(\frac{n\pi y}{H}\right) \sinh\left(\frac{n\pi(x-L)}{H}\right)$$

Avec :

$$c_n = \left( \frac{2T_0}{H \sinh(-n\pi L/H)} \right) \int_0^H \sin\left(\frac{n\pi y}{H}\right) dy \quad \forall n \in \mathbb{N}^* \quad (2.12)$$

$$= \left( \frac{2T_0 \left(\frac{L}{n\pi}\right)}{L \sinh(-n\pi H/L)} \right) (\cos(0) - \cos(n\pi)) \quad (2.13)$$

$$= \left( \frac{2T_0}{n\pi \sinh(-n\pi H/L)} \right) (1 - \cos(n\pi)) \quad (2.14)$$

Solution du problème [4] :

$$T_4(x, y) = \sum_{n=0}^{+\infty} d_n \sin\left(\frac{n\pi y}{H}\right) \sinh\left(\frac{n\pi x}{H}\right)$$

Avec :

$$d_n = \left( \frac{2T_0}{H \sinh(n\pi L/H)} \right) \int_0^H \sin\left(\frac{n\pi y}{H}\right) dy \quad \forall n \in \mathbb{N}^* \quad (2.15)$$

$$= \left( \frac{2T_0 \left(\frac{L}{n\pi}\right)}{L \sinh(-n\pi H/L)} \right) (\cos(0) - \cos(n\pi)) \quad (2.16)$$

$$= \left( \frac{2T_0}{n\pi \sinh(-n\pi H/L)} \right) (1 - \cos(n\pi)) \quad (2.17)$$

En utilisant le principe de superposition, on obtient la solution du problème initial :

$$\Delta T = 0 \Leftrightarrow \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

$$T(x, y) = \sum_{n=0}^{+\infty} \sin\left(\frac{n\pi x}{L}\right) \left[ a_n \sinh\left(\frac{n\pi(y-H)}{L}\right) + b_n \sinh\left(\frac{n\pi y}{L}\right) \right] \quad (2.18)$$

$$+ \sum_{n=0}^{+\infty} \sin\left(\frac{n\pi y}{H}\right) \left[ c_n \sinh\left(\frac{n\pi(x-L)}{H}\right) + d_n \sinh\left(\frac{n\pi x}{H}\right) \right] \quad (2.19)$$

## 2.2 Problème en coordonnées cartésiennes en 3D

$$\Delta T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0$$

### 2.2.1 Définition des conditions aux limites

$$\begin{aligned} 0 &\leq x \leq l \\ 0 &\leq y \leq L \\ 0 &\leq z \leq H \\ T(x, y, 0) &= T_s \\ T(x, y, H) &= T_a \\ T(0, y, z) &= T_s \\ T(l, y, z) &= T_s \\ T(x, 0, z) &= T_s \\ T(x, L, z) &= T_s \end{aligned}$$

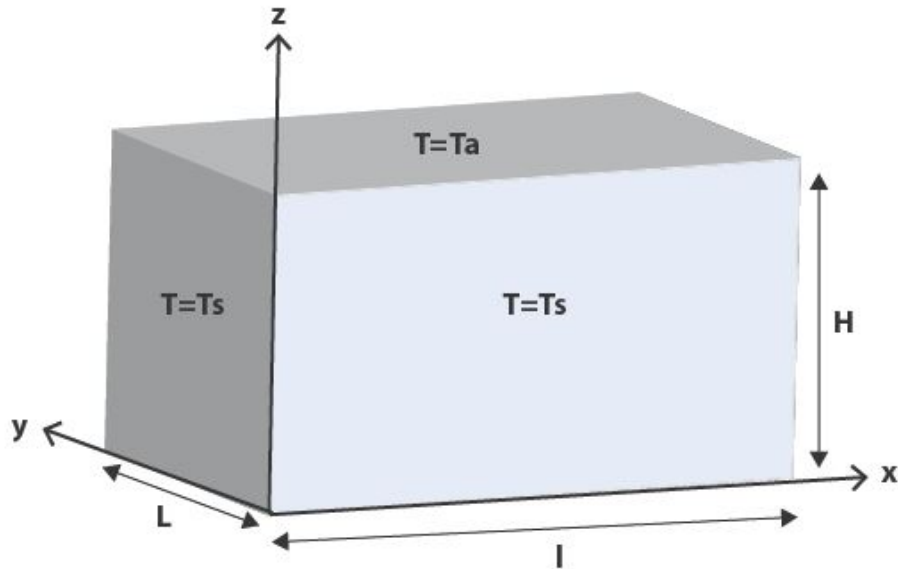


FIGURE 2.3 – Conditions aux limites du problème 3D en coordonnées cartésiennes

Le problème peut être décomposé en six problèmes ayant chacun une seule condition non-homogène plutôt que six (une pour chaque face), les six problèmes seront ensuite regroupées pour former la solution finale par le principe de superposition, comme on a procédé pour le résoudre le problème en 2D

Les six problèmes sont :

Problème [1] :

$$\frac{\partial^2 T_1}{\partial x^2} + \frac{\partial^2 T_1}{\partial y^2} + \frac{\partial^2 T_1}{\partial z^2} = 0$$

$$T_1(x, y, H) = T_a \quad \text{et} \quad T_1(x, y, z) = 0 \quad \text{sur toutes les autres faces}$$

Problème [2] :

$$\frac{\partial^2 T_2}{\partial x^2} + \frac{\partial^2 T_2}{\partial y^2} + \frac{\partial^2 T_2}{\partial z^2} = 0$$

$$T_2(x, 0, z) = T_s \quad \text{et} \quad T_2(x, y, z) = 0 \quad \text{sur toutes les autres faces}$$

Problème [3] :

$$\frac{\partial^2 T_3}{\partial x^2} + \frac{\partial^2 T_3}{\partial y^2} + \frac{\partial^2 T_3}{\partial z^2} = 0$$
$$T_3(x, y, z) = T_s \quad \text{et} \quad T_3(x, y, z) = 0 \text{ sur toutes les autres faces}$$

Problème [4] :

$$\frac{\partial^2 T_4}{\partial x^2} + \frac{\partial^2 T_4}{\partial y^2} + \frac{\partial^2 T_4}{\partial z^2} = 0$$
$$T_4(x, L, z) = T_s \quad \text{et} \quad T_4(x, y, z) = 0 \text{ sur toutes les autres faces}$$

Problème [5] :

$$\frac{\partial^2 T_5}{\partial x^2} + \frac{\partial^2 T_5}{\partial y^2} + \frac{\partial^2 T_5}{\partial z^2} = 0$$
$$T_5(l, y, z) = T_s \quad \text{et} \quad T_5(x, y, z) = 0 \text{ sur toutes les autres faces}$$

Problème [6] :

$$\frac{\partial^2 T_6}{\partial x^2} + \frac{\partial^2 T_6}{\partial y^2} + \frac{\partial^2 T_6}{\partial z^2} = 0$$
$$T_6(x, y, 0) = T_s \quad \text{et} \quad T_6(x, y, z) = 0 \text{ sur toutes les autres faces}$$

Si  $T_i$  est une solution du problème [i] pour  $i = 1; 2; 3; 4, 5, 6$  alors  $T = T_1 + T_2 + T_3 + T_4 + T_5 + T_6$  est une solution du problème de départ.[2]

## 2.2.2 Résolution par la méthode de séparation des variables

$$T_1(x, y) = X(x)Y(y)Z(z)$$

En injectant l'expression dans l'équation, nous obtenons :

$$YZX'' + XZY'' + XYZ'' = 0$$

$$\Rightarrow \frac{X''}{X} = -\alpha^2 \text{ et } \frac{Y''}{Y} = -\beta^2$$

où  $X''$  est la dérivée seconde de  $X$  par rapport à  $x$ ,  $Y''$  est la dérivée seconde de  $Y$  par rapport à  $y$  et  $Z''$  est la dérivée seconde de  $Z$  par rapport à  $z$ . Où  $\alpha$  et  $\beta$  sont des constantes. On peut en déduire que :

$$\frac{Z''}{Z} = \alpha^2 + \beta^2 = \gamma^2$$



Les conditions aux limites du problème nous donnent :

$$X(0) = 0 \Rightarrow X(x) \sin(\alpha x) \quad (2.20)$$

$$Y(0) = 0 \Rightarrow Y(y) \sin(\beta y) \quad (2.21)$$

$$Z(0) = 0 \Rightarrow Z(z) \sinh(\gamma z) = \sinh(\sqrt{\alpha^2 + \beta^2} z) \quad (2.22)$$

$$X(l) = 0 \Rightarrow \alpha = \frac{n\pi}{l} n = 1, 2, 3, \dots \quad (2.23)$$

$$X(L) = 0 \Rightarrow \beta = \frac{m\pi}{L} m = 1, 2, 3, \dots \quad (2.24)$$

$$(2.25)$$

$$\gamma = \sqrt{\alpha^2 + \beta^2} = \sqrt{\left(\frac{n\pi}{l}\right)^2 + \left(\frac{m\pi}{L}\right)^2}$$

La solution générale du problème s'écrit :

$$T_1(x, y, z) = \sum_{n,m=1}^{+\infty} A_{nm} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{m\pi y}{L}\right) \sinh(\gamma z)$$

Les conditions aux limites nous donnent :

$$T_1(x, y, H) = T_1(x, y) = \sum_{n,m=1}^{+\infty} A_{nm} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{m\pi y}{L}\right) \sinh(\gamma H)$$

En multipliant la relation ci dessus par  $\sin\left(\frac{p\pi x}{l}\right) \sin\left(\frac{q\pi y}{L}\right)$  et en intégrant sur  $\int_{x=0}^{x=l} dx \int_{y=0}^{y=L} dy$  pour projeter le  $p^{ime}$  et le  $q^{ime}$  terme en utilisant la propriété de l'orthogonalité du sin afin d'obtenir le coefficient  $A_{nm}$  :

$$\int_0^l \int_0^L T(x, y) \sin\left(\frac{p\pi x}{l}\right) \sin\left(\frac{q\pi y}{L}\right) dx dy = \quad (2.26)$$

$$\sum_{n,m=1}^{+\infty} A_{nm} \sinh(\gamma H) \int_0^l \int_0^L \underbrace{\sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{m\pi y}{L}\right) \sin\left(\frac{p\pi x}{l}\right) \sin\left(\frac{q\pi y}{L}\right)}_{dx dy} dx dy \quad (2.27)$$

Pour  $p$  ou  $q$  des entiers pairs :  $=0$

Pour  $p$  ou  $q$  tous les deux des entiers impairs :  $= \frac{l}{2} \frac{L}{2} \delta_{pn} \delta_{qm}$

Avec symbole de Kroenecker :

$$\delta_{pn} = \begin{cases} 1, & \text{quand } p = n \\ 0, & \text{quand } p \neq n \end{cases} \quad \delta_{qm} = \begin{cases} 1, & \text{quand } q = m \\ 0, & \text{quand } q \neq m \end{cases}. \quad (2.28)$$

Donc :

$$\int_0^l \int_0^L T(x, y) \sin\left(\frac{p\pi x}{l}\right) \sin\left(\frac{q\pi y}{L}\right) dx dy = A_{pq} \sinh(\gamma H) \frac{l}{2} \frac{L}{2} = 0$$

Ce résultat est obtenu pour p ou(et) q des entiers pairs.

On obtient alors :

$$A_{pq} = \frac{4}{lL} * \left[ \frac{\int_0^l \int_0^L T(x, y) \sin\left(\frac{p\pi x}{l}\right) \sin\left(\frac{q\pi y}{L}\right) dx dy}{\sinh(\gamma H)} \right] \text{ pour p et q des entiers impairs}$$

$A_{pq}=0$  pour p ou(et) q des entiers pairs.

Sachant que  $T(x, y) = T_a$  est une constante sur la surface supérieure, on alors :

$$A_{pq} = \frac{4}{lL} \frac{2l}{p\pi} \frac{2L}{q\pi} \frac{T_a}{\sinh(\gamma H)}$$

On trouve finalement :

$$T_1(x, y, z) = \sum_{n,m=1}^{+\infty} A_{nm} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{m\pi y}{L}\right) \sinh(\gamma z)$$

Avec  $\gamma = \sqrt{\alpha^2 + \beta^2}$  et  $A_{nm} = \left( \frac{16T_a}{nm\pi^2 \sinh(\gamma H)} \right)$  et tel que  $m$  et  $n$  sont des entiers impairs !

En procédant de la même façon, on obtient les températures  $T_2, T_3, T_4, T_5$  et  $T_6$

On trouve :

$$T_2(x, y, z) = \sum_{n,m=1}^{+\infty} B_{nm} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{m\pi(y-L)}{L}\right) \sinh(\gamma z)$$

Avec  $\gamma = \sqrt{\alpha^2 + \beta^2}$  et  $B_{nm} = \left( \frac{16T_s}{nm\pi^2 \sin(\beta L)} \right)$  et tel que  $m$  et  $n$  sont des entiers impairs !

$$T_3(x, y, z) = \sum_{n,m=1}^{+\infty} C_{nm} \sin\left(\frac{n\pi(x-l)}{l}\right) \sin\left(\frac{m\pi y}{L}\right) \sinh(\gamma z)$$

Avec  $\gamma = \sqrt{\alpha^2 + \beta^2}$  et  $C_{nm} = \left( \frac{16T_s}{nm\pi^2 \sin(\alpha l)} \right)$  et tel que  $m$  et  $n$  sont des entiers impairs !

$$T_4(x, y, z) = \sum_{n,m=1}^{+\infty} B_{nm} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{m\pi y}{L}\right) \sinh(\gamma z)$$

Avec  $\gamma = \sqrt{\alpha^2 + \beta^2}$  et  $B_{nm} = \left( \frac{16T_s}{nm\pi^2 \sin(\beta L)} \right)$  et tel que  $m$  et  $n$  sont des entiers impairs !

$$T_5(x, y, z) = \sum_{n,m=1}^{+\infty} C_{nm} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{m\pi y}{L}\right) \sinh(\gamma z)$$

Avec  $\gamma = \sqrt{\alpha^2 + \beta^2}$  et  $C_{nm} = \left( \frac{16T_s}{nm\pi^2 \sin(\alpha l)} \right)$  et tel que  $m$  et  $n$  sont des entiers impairs !

$$T_6(x, y, z) = \sum_{n,m=1}^{+\infty} A_{nm} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{m\pi y}{L}\right) \sinh(\gamma(z - H))$$

Avec  $\gamma = \sqrt{\alpha^2 + \beta^2}$  et  $A_{nm} = \left( \frac{16T_a}{nm\pi^2 \sinh(\gamma H)} \right)$  et tel que  $m$  et  $n$  sont des entiers impairs !

La solution globale du problème en 3 dimensions dans une cavité parallélépipédique est alors la somme de ces différentes températures, ce qui nous donne :

$$\begin{aligned} T(x, y, z) = & \sum_{n,m=1}^{+\infty} A_{nm} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{m\pi y}{L}\right) [\sinh(\gamma z) + \sinh(\gamma(z - H))] \\ & + \sum_{n,m=1}^{+\infty} B_{nm} \sin\left(\frac{n\pi x}{l}\right) \sinh(\gamma z) \left[ \sin\left(\frac{m\pi y}{L}\right) \sin\left(\frac{m\pi(y - L)}{L}\right) \right] \\ & + \sum_{n,m=1}^{+\infty} C_{nm} \sin\left(\frac{m\pi y}{L}\right) \sinh(\gamma z) \left[ \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{n\pi(x - l)}{l}\right) \right] \end{aligned}$$

Tel que  $m$  et  $n$  sont des entiers impairs, avec  $\gamma = \sqrt{\alpha^2 + \beta^2}$

$$A_{nm} = \left( \frac{16T_a}{nm\pi^2 \sinh(\gamma H)} \right)$$

$$B_{nm} = \left( \frac{16T_s}{nm\pi^2 \sin(\beta L)} \right)$$

$$C_{nm} = \left( \frac{16T_s}{nm\pi^2 \sin(\alpha l)} \right)$$

## 2.3 Equation de Laplace en coordonnées polaires

(2D)

### 2.3.1 Description du Problème :

Cette partie est consacrée à l'étude de la convection naturelle dans une cavité semi-circulaire. Le schéma ci-dessous montre les étapes de l'étude pour avoir les bonnes conditions aux limites du problème.

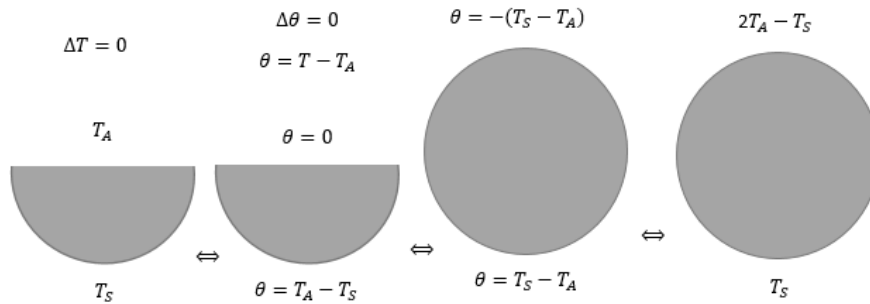


FIGURE 2.4 – Conditions aux limites pour le demi-disque

$$\Delta T = \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial T}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 T}{\partial \theta^2} = 0 \quad (1)$$

### 2.3.2 Résolution de l'équation par la méthode de séparations

de variables :

- Conditions aux limites du problème :

$$T(r = R, 0 \leq \theta \leq \pi) = 2T_A - T_S$$

$$T(r = R, \pi \leq \theta \leq 2\pi) = T_S$$

L'équation de la Laplace est une équation aux dérivées partielles à plusieurs variables, pour la résoudre analytiquement on doit utiliser la méthode de

séparation de variables :[3]

$$T(r, \theta) = f(r)g(\theta)$$

En introduisant cette formule dans (1), on trouve :

$$\frac{r^2(f''(r) + \frac{1}{r}f'(r))}{f(r)} + \frac{g''(\theta)}{g(\theta)} = 0$$

⇒

$$\frac{g''(\theta)}{g(\theta)} = -\lambda \quad (2)$$

et

$$\frac{r^2(f''(r) + \frac{1}{r}f'(r))}{f(r)} = \lambda \quad (3)$$

On aboutit à deux équations différentielles ordinaires à une seule variable. -  
La résolution de (2) :

$$g(\theta) = A_n \cos(n\theta) + B_n \sin(n\theta)$$

La fonction g est  $2\pi$  périodique donc la décomposition de g en série de Fourier va se former alors d'une base de valeurs propres  $n = \sqrt{\lambda}$  associé à des modes propres  $g_n(\theta)$ .

$$g_n(\theta) = A_n \cos(n\theta) + B_n \sin(n\theta)$$

Pour l'équation (3), on a :

$$\frac{r^2(f''(r) + \frac{1}{r}f'(r))}{f(r)} = n^2 \quad (3)$$

pour  $n=0$  la solution est :  $f(r) = A \log r + B$  pour  $n \geq 1$  la solution se transforme en série entière de terme  $r^\alpha$  avec  $\alpha = \pm n$  puisqu'on se place dans le problème de la fusion de la glace donc on va éliminer la solution pour laquelle la fonction tend vers 0 donc on va prendre :

$$f(r) = r^n$$

et alors la forme générale de la solution s'écrit alors en terme de série de Fourier :

$$T_n(r, \theta) = \frac{A_0}{2} + \sum_{n=1}^{+\infty} r^n (A_n \cos(n\theta) + B_n \sin(n\theta)) \quad (4)$$

- Détermination des coefficients  $A_n$  et  $B_n$  de la série de Fourier :

$$A_n = \frac{1}{2R^n\pi} \int_0^{2\pi} f(\phi) \cos(n\phi) d\phi$$

$$B_n = \frac{1}{2R^n\pi} \int_0^{2\pi} f(\phi) \sin(n\phi) d\phi$$

les  $f(\phi)$  sont données par les conditions aux limites :

$$f(\phi) = \begin{cases} f_1(\phi) = 2T_A - T_s \text{ si } \theta \in [0, \pi] \\ f_2(\phi) = T_s \text{ si } \theta \in [\pi, 2\pi] \end{cases}$$

pour avoir la solution  $T$  de ce problème, il suffit alors de remplacer les coefficients  $A_n$  et  $B_n$  dans (4).

$$A_0 = \frac{1}{2\pi} \int_0^{2\pi} f(\phi) d\phi$$

$$\Rightarrow A_0 = \frac{1}{2\pi} \int_0^\pi f_1(\phi) d\phi + \frac{1}{2\pi} \int_\pi^{2\pi} f_2(\phi) d\phi$$

$$\Rightarrow A_0 = \frac{1}{2\pi} \int_0^\pi (2T_A - T_s) d\phi + \frac{1}{2\pi} \int_\pi^{2\pi} T_s d\phi$$

$$\Rightarrow A_0 = T_A$$

L'expression  $T_n$  devient alors sans remplacer  $A_0$  par  $T_A$  :

$$T_n(r, \theta) = \frac{1}{4\pi} \int_0^{2\pi} f(\phi) d\phi$$

$$+ \sum_{n=1}^{+\infty} r^n \left[ \frac{1}{2R^n\pi} \left( \int_0^{2\pi} f(\phi) \cos(n\phi) d\phi \cos(n\theta) + \int_0^{2\pi} f(\phi) \sin(n\phi) d\phi \sin(n\theta) \right) \right]$$

$$\Rightarrow$$

$$T_n(r, \theta) = \frac{1}{2\pi} \int_0^{2\pi} (f(\phi) (1 + 2 \sum_{n=1}^{+\infty} \left(\frac{r}{R}\right)^n \cos(n(\phi - \theta)))$$

et en utilisant la formule trigonométrique :

$\sum_{n=-\infty}^{+\infty} \left(\frac{r}{R}\right)^n \exp(in(\phi - \theta))$  on obtient après les simplifications la formule de Poisson associée au problème de Dirichlet :

$$T(r, \theta) = \frac{1}{2\pi} \int_0^{2\pi} f(\phi) \frac{R^2 - r^2}{R^2 - 2rR \cos(\phi - \theta) + r^2}$$

L'intégrale est définie sur tout le disque or dans ce problème on a deux domaines sur lesquels sont imposées les conditions aux limites , donc il faut

décomposer l'intégrale en deux sous domaines .

$$T(r, \theta) = \frac{1}{2\pi} \int_0^\pi f_1(\phi) \frac{R^2 - r^2}{R^2 - 2rR \cos(\phi - \theta) + r^2} d\phi \\ + \frac{1}{2\pi} \int_\pi^{2\pi} f_2(\phi) \frac{R^2 - r^2}{R^2 - 2rR \cos(\phi - \theta) + r^2} d\phi$$

$\Rightarrow$

$$T(r, \theta) = \frac{1}{2\pi} \int_0^\pi (2T_A - T_s) \frac{R^2 - r^2}{R^2 - 2rR \cos(\phi - \theta) + r^2} d\phi \\ + \frac{1}{2\pi} T_s \int_\pi^{2\pi} \frac{R^2 - r^2}{R^2 - 2rR \cos(\phi - \theta) + r^2} d\phi$$

**- Résolution par la méthode d'orthogonalité des modes :**

Dans cette résolution on va prendre la forme (4) mais dans les deux bords c'est à dire pour  $r = R$  puis on multiplie par un mode propre  $\cos(m\theta)$ . On suppose que  $T_S$  la température à l'interface; est nulle. On procède de la manière suivante :

$$\int_0^{2\pi} T_n(R, \theta) \cos(m\theta) d\theta \\ = \int_0^{2\pi} \frac{T_A}{2} \cos(m\theta) d\theta \\ + \int_0^{2\pi} \sum_{n=1}^{+\infty} R^n (A_n \cos(n\theta) + B_n \sin(n\theta)) \cos(m\theta) d\theta \quad 4)$$

On va utiliser cette formule pour déterminer les coefficients  $A_m$ . De plus on sait que l'orthogonalité de deux modes propres nous donne le résultat immédiat suivant :

$$\int_0^{2\pi} \cos(n\theta) \cos(m\theta) d\theta = \pi$$

pour  $n = m$  et 0 sinon et

$$\int_0^{2\pi} \sin(n\theta) \cos(m\theta) d\theta = 0 \quad \forall (n, m)$$

$\Rightarrow$

$$2T_A \int_0^\pi \cos(m\theta) d\theta = 0 + A_m R^m \pi$$

$\Rightarrow \pi A_m R^m = 0 \Rightarrow$  les  $A_m$  sont tous nuls.  $\forall m \in N^*$

Détermination des coefficients  $B_m$ .

Pour cela on doit multiplier par le deuxième mode propre qui est  $\sin(m\theta)$ .

Et on travaille sur le domaine  $[0, \pi]$  car la fonction est nulle en  $[\pi, 2\pi]$ .

On aboutit après les calculs à  $B_m = \frac{4T_A}{m} \frac{1}{R^m \pi}$  pour  $m = 2n - 1$

La solution générale du problème devient alors :

$$T(r, \theta) = T_A + \sum_{n=1}^{\infty} \frac{4T_A}{\pi} \left(\frac{r}{R}\right)^{2n-1} \sin((2n-1)\theta) \quad (5)$$

C'est cette expression qu'on va la prendre pour faire l'étude numérique du cas d'un demi disque.

## 2.4 Équation de Laplace en coordonnées sphériques

### 3D

L'équation de Laplace en coordonnées sphériques est une généralisation du coordonnées polaires, il suffit de faire une rotation avec un angle  $\phi$  pour obtenir une boule (sphère).

$$\Delta T = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial T}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial T}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 T}{\partial \phi^2} = 0 \quad (5)$$

**-Méthode de séparation de variables**

$$T(r, \theta, \phi) = f(r)G(\theta, \phi)$$

-En introduisant cette expression de l'équation (5) on obtient comme précédemment avec une somme discrète la solution générale suivante :

$$T(r, \theta, \phi) = \sum_{n=1}^{\infty} \left( Cr^n + \frac{D}{r^{n+1}} \right) \sum_{m=1}^{\infty} (a_n^m \cos(m\phi) + b_n^m \sin(m\phi)) P_n^m(\cos\theta)$$

avec  $P_n^m$  est le polynôme de Legendre

Pour l'étude de notre problème on ne va pas utiliser cette expression. Les études graphique et numérique en 2D suffiront pour connaître la distribution de la température dans la cavité semi-circulaire et par la suite dans la cavité semi-sphérique.



## 2.5 Etude numérique en Python

### 2.5.1 Etude numérique du problème de la cavité rectangulaire (2D) :

```
1 # -*- coding : utf-8 -*-
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from math import *
5 from matplotlib import cm
6 from matplotlib.ticker import LinearLocator, FormatStrFormatter
7 from mpl_toolkits.mplot3d import Axes3D
8 N=100
9 L=4
10 H=2
11 Ta=25
12 Ts=-5
13 # Nombre de points sur x et y
14 nx=200
15 ny=200
16 # On définit les fonctions T1,T2,T3 et T4
17 def T1(x,y,n,H,L,Ts):
18     return 2.*Ts/(L*np.sinh(-n*np.pi*H/L))*(L/(n*np.pi))*(1-np.cos(n*
19         np.pi))*np.sin(n*x*np.pi/L)*np.sinh((y-H)*n*np.pi/L)
20 def T2(x,y,n,H,L,Ta):
21     return 2.*Ta/(L*np.sinh(n*np.pi*H/L))*(L/(n*np.pi))*(1-np.cos(n*np
22         .pi))*np.sin(n*x*np.pi/L)*np.sinh(y*n*np.pi/L)
```

```

22
23 def T3(x, y, n, H, L, Ts) :
24     return 2.*Ts/(H*np.sinh(-n*np.pi*L/H))*(H/(n*np.pi))*
25         (1-np.cos(n*np.pi))*np.sin(n*y*np.pi/H)*np.sinh((x-L)*n*np
26             .pi/H)
27
28 def T4(x, y, n, H, L, Ts) :
29     return 2.*Ts/(H*np.sinh(n*np.pi*L/H))*(H/(n*np.pi))*(1-np.cos(n*np
30         .pi))*np.sin(n*y*np.pi/H)*np.sinh(x*n*np.pi/H)
31
32 # On définit T comme la somme de T1, T2, T3 et T4
33 def T(x, y, n, H, L, Ts, Ta) :
34     return T1(x, y, n, H, L, Ts)+T2(x, y, n, H, L, Ta)+T3(x, y, n, H, L, Ts)+T4(x, y, n
35         ,H,L, Ts)
36
37 #maillage
38 x=np.linspace(0,L,nx)
39 y=np.linspace(0,H,ny)
40 X,Y=np.meshgrid(x,y)
41
42 #Sommmation des modes
43 for n in range(1,N+1) :
44     if n==1 :
45         Ttot=T(X,Y,n,H,L, Ts, Ta)
46     else :
47         Ttot=Ttot+T(X,Y, n, H, L, Ts, Ta)
48
49 #Plot 2D
50 CS1=contourf(X, Y, Ttot)
51 cbar1=plt.colorbar(CS1)
52 show()
53 fig = plt.figure()

```

```

47 ax = fig.add_subplot(111, projection='3d')
48 surf = ax.plot_surface(X,Y,Ttot)
49 plt.show()
50 #Plot surfacique en 2D
51 fig = plt.figure()
52 ax = fig.gca(projection='3d')
53 plt.show()
54 # Plot surfacique couleur
55 surf = ax.plot_surface(X, Y, Ttot, cmap=cm.coolwarm,
56                       linewidth=0, antialiased=False)
57 # Echelle de couleur
58 fig.colorbar(surf, shrink=0.5, aspect=5)
59 plt.show()

```

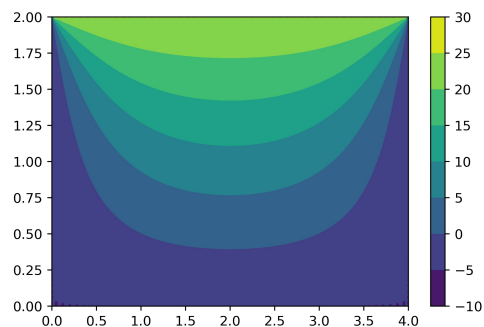


FIGURE 2.5 – *Figure Python : Distribution de la température dans une cavité rectangulaire avec " $T_a=25, T_s=-5, L=4, H=2$ "*

On remarque sur la figure 2.5 que les conditions aux limites sont bien respectées, et sont bien visibles sur l'image, on peut voir aussi les iso-valeurs représentées par différentes couleurs, la forme de répartition de la température a une forme parabolique. Plus le nombre de mode est grand mieux est

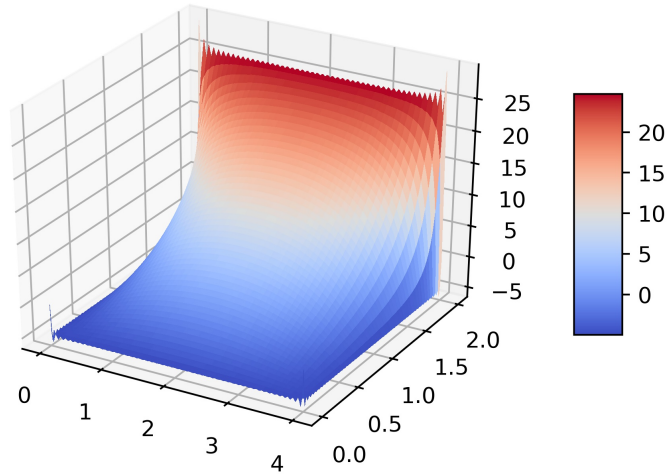


FIGURE 2.6 – *Figure Python :Distribution surfacique de la température dans une cavité rectangulaire (z=axe de la température)*

la répartition de la température sur les bords, toutefois à partir d'un nombre assez grand de modes (500), le programme est saturé et nous permet plus de raffiner encore plus la distribution. Sur le bord ( $y = H$ ), il y a un certain nombre d'irrégularité dans la distribution de la température ( la valeur de la température dépasse en certains points la valeur de  $T_a$  ou est inférieure  $T_s$  ) chose due à l'orthogonalité de la série de Fourier .

### 2.5.2 Étude numérique et graphique du problème de la cavité semi-circulaire (2D) :

Le programme ci-dessous consiste à utiliser la méthode des trapèzes pour le calcul numérique de la formule de Poisson de la solution de Dirichlet.

```
1 from pylab import*
```

```

2 from matplotlib import *
3 import numpy as np
4 from math import *
5 from matplotlib import animation
6 def T1(phi , r , theta ,R, Ta, Ts) :
7     ftemp = ((R**2.-2.*r*np.cos(theta-phi)+r**2.)*2.*pi)
8     return (2*Ta-Ts)*(R**2.-r**2.)/ftemp
9 def T2(phi , r , theta ,R, Ta, Ts) :
10    return Ts*(R**2.-r**2.)/((R**2.-2.*r*np.cos(theta-phi)+r**2.)*2.*
        np.pi)
11 # la méthode des trapèzes pour le calcul de l'intégrale de T1
12 def Q1(r , theta) :
13     n=200
14     dphi= np.pi/n
15     temp1=0.5*(T1(0 , r , theta ,R, Ta, Ts)+T1(np.pi , r , theta ,R, Ta, Ts))
16     for i in range(1,n) :
17         phi = i*dphi
18         temp1 = temp1 + T1(phi , r , theta ,R, Ta, Ts)
19     return dphi*temp1
20 # Le calcul de l'integral de T2 par la même méthode
21 def Q2(r , theta) :
22     n=200
23     dphi= np.pi/n
24     temp2=0.5*(T2(np.pi , r , theta ,R, Ta, Ts)+T2(2*np.pi , r , theta ,R, Ta, Ts))
25     for i in range(1,n) :
26         phi = i*dphi
27         temp2 = temp2 + T2(np.pi+phi , r , theta ,R, Ta, Ts)
28     return dphi*temp2

```

```

29 # Nombre de points sur x et y
30 nx=200
31 ny=200
32
33 # Creer l'ensemble des points x et y
34 R, Ta, Ts=1, 15, 0
35
36 x=np.linspace(-R,R,nx)
37 y=np.linspace(-R,0,ny)
38 X, Y=np.meshgrid(x,y)
39 r=np.sqrt(X**2+Y**2)
40 theta=arctan2(Y,X)
41 Temp1=Q1(r,theta)
42 Temp2=Q2(r,theta)
43
44 Q1=np.linspace(Ts,Ta,200)
45 Q2=np.linspace(Ts,Ta,200)
46 #print(Temp)
47 CS1=contourf(x,y,Temp1,Q1)
48 CS3=contourf(-1*x,y,Temp1,Q1)
49 cbar1=plt.colorbar(CS1)
50 show()

```

*Listing 2.1 – Programme Python 2D en coordonnées cartésiennes*

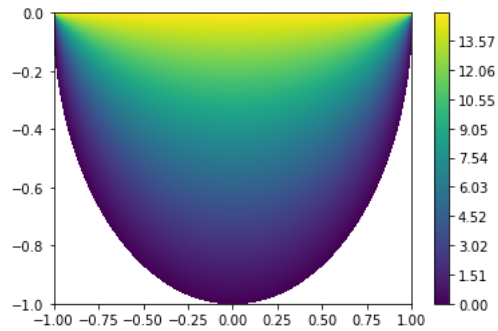


FIGURE 2.7 – *Figure Python de la distribution de la température dans la cavité circulaire(semi-circulaire)*

Le but de cette analyse par la méthode d'orthogonalité est de jouer sur le nombre des modes propres  $N$  ainsi le nombre de mailles sur les axes ( $n_x$ ,  $n_y$ ) jusqu'à avoir la convergence en maillage de la solution donnée de ce problème.

```

1  # -*- coding : utf-8 -*-
2  from pylab import mpl
3  import matplotlib.pyplot as plt
4  import numpy as np
5  import pylab as py
6
7  N=20
8  R=1
9  Ta=15
10 Ts=0
11 DT=Ta-Ts
12 nx=200
13 ny=200
14 # Nombre de points sur x et y

```

```

15
16 x=np.linspace(-R,R,nx)
17 y=np.linspace(-R,0,ny)
18 X,Y=np.meshgrid(x,y)
19
20 def T(X,Y,n,R,DT):
21     r=np.sqrt(X**2.+Y**2.)
22     theta = np.arctan2(-Y,X)
23     Temp1=(4*DT/np.pi)*(np.power((r/R),2.*n-1.) * np.sin((2.*n-1.) *
24         theta ))/(2.*n-1.)
25     return np.piecewise(r, [r<R, r >= R], [1, 0])*Temp1
26
27 Ttot=Ta
28 for n in range(1,N+1):
29     Ttot=Ttot-T(X,Y,n,R,DT)
30
31 CS1=pcolor(X,Y,Ttot)
32 cbar1=plt.colorbar(CS1)
33 show()

```

*Listing 2.2 – Programme Python pour la cavité semi-circulaire en 2D*

Le graphe obtenu correspond à la distribution de la température dans une cavité géométrique de forme circulaire. Le maillage est assez raffiné en choisissant  $nx = ny = 200$  dans les deux programmes. L'approche numérique est donc bien adéquate avec le problème dans la nature puisque la valeur de la température dans le demi disque inférieure diminue en s'approchant de la paroi solide.

### 2.5.3 Étude numérique et graphique du problème 3D en coordonnées cartésiennes

```

1 # -*- coding : utf-8 -*-

```



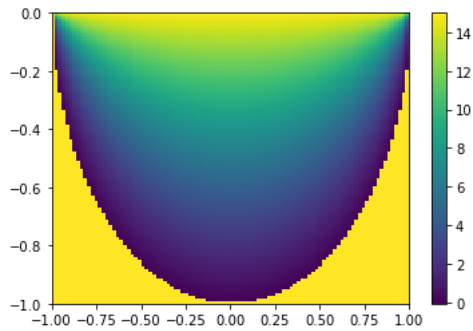


FIGURE 2.8 – Graphe programme 2 Python correspond à la distribution de la température dans la semi-circulaire

```

2 from pylab import*
3 import matplotlib.pyplot as plt
4 import numpy as np
5 from math import *
6 from mpl_toolkits.mplot3d import Axes3D
7
8 N=20
9 l=1
10 L=2
11 H=2
12 Ta=15
13 Ts=0
14 # Nombre de points sur x, y et z
15 nx=40
16 ny=40
17 nz=80
18 #On définit les 6 fonctions chacune correspondant à un problème avec

```

```

    une seule condition aux limites
19 def T1(x, y, z, n, m, l, L, H, Ta) :
20     return 16*Ta/(n*m*np. pi**2*np. sinh(np. sqrt(((n*np. pi/l)**2)+(m*np. pi
        /L)**2)*H))*np. sin(n*np. pi*x/l)*np. sin(m*np. pi*y/L)*np. sinh((np.
        sqrt(((n*np. pi/l)**2)+(m*np. pi/L)**2))*z)
21
22 def T2(x, y, z, n, m, l, L, H, Ts) :
23     return 16*Ts/(n*m*np. pi**2*np. sin(m*np. pi/L*L))*np. sin(n*np. pi*x/l)*
        np. sin(m*np. pi*(y-L)/L)*np. sinh((np. sqrt(((n*np. pi/l)**2)+(m*np.
        pi/L)**2))*z)
24
25 def T3(x, y, z, n, m, l, L, H, Ts) :
26     return 16*Ts/(n*m*np. pi**2*np. sin((n*np. pi/l)*l))*np. sin(n*np. pi*(x-
        l)/l)*np. sin(m*np. pi*y/L)*np. sinh((np. sqrt(((n*np. pi/l)**2)+(m*
        np. pi/L)**2))*z)
27
28 def T4(x, y, z, n, m, l, L, H, Ts) :
29     return 16*Ts/(n*m*np. pi**2*np. sin(m*np. pi/L*L))*np. sin(n*np. pi*x/l)*
        np. sin(m*np. pi*y/L)*np. sinh((np. sqrt(((n*np. pi/l)**2)+(m*np. pi/L
        )**2))*z)
30
31 def T5(x, y, z, n, m, l, L, H, Ts) :
32     return 16*Ts/(n*m*np. pi**2*np. sin((n*np. pi/l)*l))*np. sin(n*np. pi*(x-
        l)/l)*np. sin(m*np. pi*y/L)*np. sinh((np. sqrt(((n*np. pi/l)**2)+(m*
        np. pi/L)**2))*z)
33
34 def T6(x, y, z, n, m, l, L, H, Ts) :
35     return 16*Ts/(n*m*np. pi**2*np. sinh((np. sqrt(((n*np. pi/l)**2)+(m*np.

```

```

    pi/L)**2))*H))*np.sin(n*np.pi*x/l)*np.sin(m*np.pi*y/L)*np.sinh((
    np.sqrt(((n*np.pi/l)**2)+(m*np.pi/L)**2))*(z-H))
36
37 def T(x,y,z,n,m,l,L,H,Ts,Ta):
38     return T1(x,y,z,n,m,l,L,H,Ta)+T2(x,y,z,n,m,l,L,H,Ts)+T3(x,y,z,n,m,
    l,L,H,Ts)+T4(x,y,z,n,m,l,L,H,Ts)+T5(x,y,z,n,m,l,L,H,Ts)+T6(x,y
    ,z,n,m,l,L,H,Ts)
39
40 x=np.linspace(0,l,nx)
41 y=np.linspace(0,L,ny)
42 z=np.linspace(0,H,nz)
43 X,Y,Z=np.meshgrid(x,y,z)
44
45 for m in range(1,2*N+1,2):
46     for n in range(1,2*N+1,2):
47         if n==m==1:
48             Ttot=T(X,Y,Z,n,m,l,L,H,Ts,Ta)
49         else:
50             Ttot=Ttot+T(X,Y,Z,n,m,l,L,H,Ts,Ta)
51
52 fig = plt.figure()
53 ax = fig.add_subplot(projection='3d')
54 surf = ax.plot_surface(X,Y,Z, Ttot)
55 plt.show()

```

*Listing 2.3 – Programme Python pour le pavé droit (3D)*

## 2.6 Étude numérique avec FreeFem++

### 2.6.1 L'utilité du logiciel de FreeFem++

FreeFem++[4] est un logiciel qui permet la résolution numérique des équations différentielles d'ordre 1 ou plus et à plusieurs variables. Notre étude consiste à résoudre l'équation de Laplace qui est une équation parabolique d'ordre 2 à deux variables en 2D et à 3 variables en 3D.

On utilise souvent pour la résolution des équations de ce genre la méthode des éléments finis.

Voici les étapes à suivre :

- Définir le domaine de l'étude en utilisant la syntaxe " border nom1type de coordonnéesdomaine"

- Générer le maillage avec "mesh nom2nom1(nombre de mailles)"

- Choisir un espace fonctionnel avec "fespace nom3 nom2,type des éléments finis"

- Initialisation fonction et fonction test

- Définition des conditions aux limites

- Résolution par formulation variationnelle

problem nom4fonction, fonction test, solver =nom du solveur,tolerance

int2d(TC)( dx(u)\*dx(v) + dy(u)\*dy(v) ) + on(1,u=2\*g-h)

+ on(2,u=h);

on(label du domaine, condition aux limites à imposer sur ce domaine) nom4;

// résoudre le problème

plot (u ,TC ,fill=1,value=true, wait=true); //tracer le graphe

TC=adaptmesh(TC,u,inquire=1,err=error); //adapter le maillage sur les domaines

error = error \* coef; //calculer l'erreur

### 2.6.2 Résolution FreeFem++ de Laplace 2D en coordonnées

#### cartésiennes

```
1 mesh Th1=square(10,10,[4*x,2*y]); //on définit la cavité rectangulaire
2 fespace Vh(Th1,P1); // to change P1 in P2 to make P2 finite element
```

```

3  Vh u=0,v;
4  func f= 0;
5  func g= 25;    // g c'est la température T_a
6  func h= -5;   // h c'est la température T_s
7  int i=0;
8  real error=0.1, coef= 0.1^(1./5.);
9  problem Laplace(u,v, solver=CG, eps=-1.0e-6) =
10     int2d(Th1)( dx(u)*dx(v) + dy(u)*dy(v) ) //formulation
        variationnelle
11     + on(3,u=g)           //On impose T=Ta sur le côté 3 (y=L)
12     + on(1,2,4,u=h) ;    //On impose T=Ts sur les côtés 1,2 et 4
13
14  for (i=0;i< 10;i++)
15  {
16     real d = clock();
17     Laplace; // solve the problem
18     plot(u,Th1, fill=1,wait=1);
19     Th1=adaptmesh(Th1,u, inquire=1,err=error); //maillage automatique
        choisi par freefem
20     error = error * coef;
21 } ;

```

*Listing 2.4 – Programme Freefem pour la cavité rectangulaire 2D*

### 2.6.3 Résolution FreeFem de Laplace 2D en coordonnées polaires

```

1  load "msh3"
2  load "medit"

```

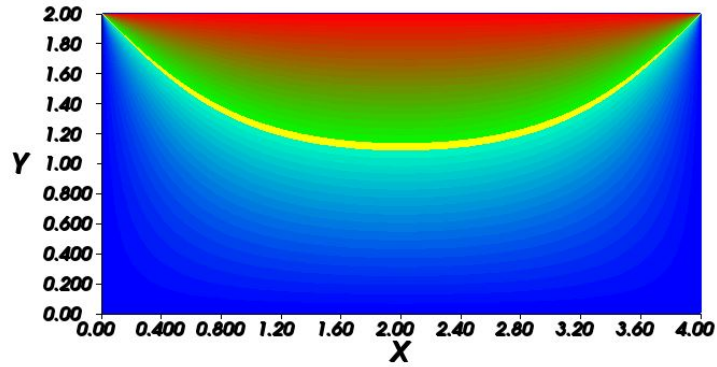


FIGURE 2.9 – *Figure FreeFem++ de la distribution de la température dans la cavité rectangulaire*

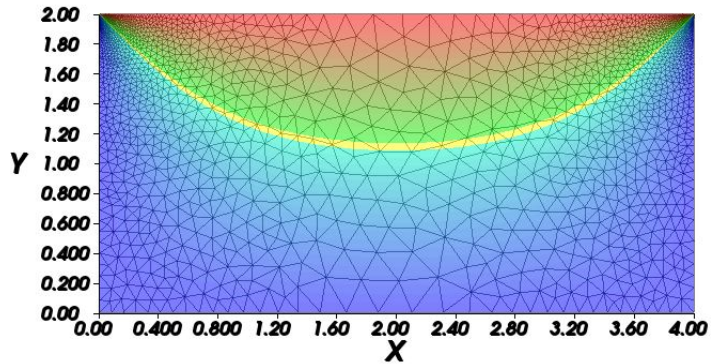


FIGURE 2.10 – *Figure FreeFem++ de la distribution de la température et du maillage choisi dans la cavité rectangulaire*

```

3 border BC1(t=0,pi){ x=cos(t);y=sin(t);label=1;}
4 border BC2(t=pi,2*pi){ x=cos(t);y=sin(t);label=2;}
5 mesh TC= buildmesh(BC1(100)+BC2(100));
6 fespace Vh(TC,P1); // Espace fonctionnel

```

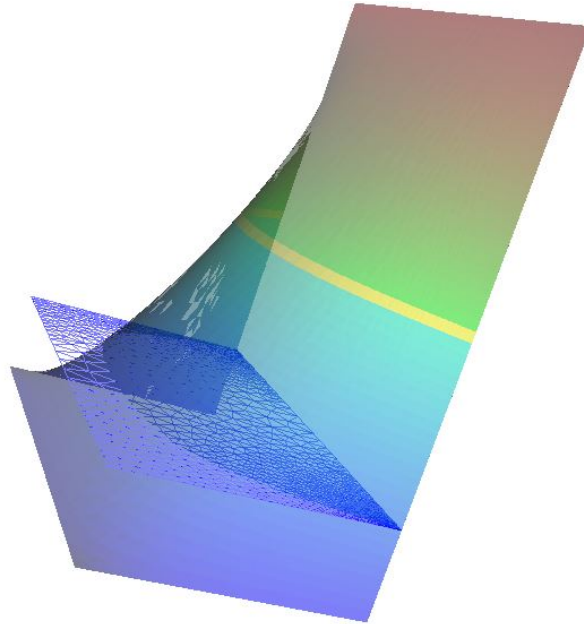


FIGURE 2.11 – *Figure FreeFem++ de la distribution surfacique de la température dans la cavité rectangulaire ( $z=T$ )*

```

7  Vh u=0,v ;
8  func f= 0 ;
9  func g= 20;    // g c'est la température T_A
10 func h= 0;    // h c'est la température T_S
11 int i=0;
12 real error=0.1, coef= 0.1^(1./5.);
13 problem Laplace(u,v,solver=CG,eps=-1.0e-6) =
14     int2d(TC)( dx(u)*dx(v) + dy(u)*dy(v) )
15     + on(1,u=2*g-h) //condition aux limites dans le premier domaine
16     [0,pi]
17     + on(2,u=h); //condition aux limites dans le deuxième domaine [pi,2

```

```

    pi]
17 for (i=0;i< 10;i++)
18 {
19   real d = clock();
20   Laplace; // solve the problem
21   plot (u ,TC ,fill=1,value=true , wait=true) ;
22   TC=adaptmesh(TC,u, inquire=1,err=error);
23   error = error * coef;
24 } ;

```

Listing 2.5 – Programme Freefem++ pour la cavité demi-circulaire (2D)

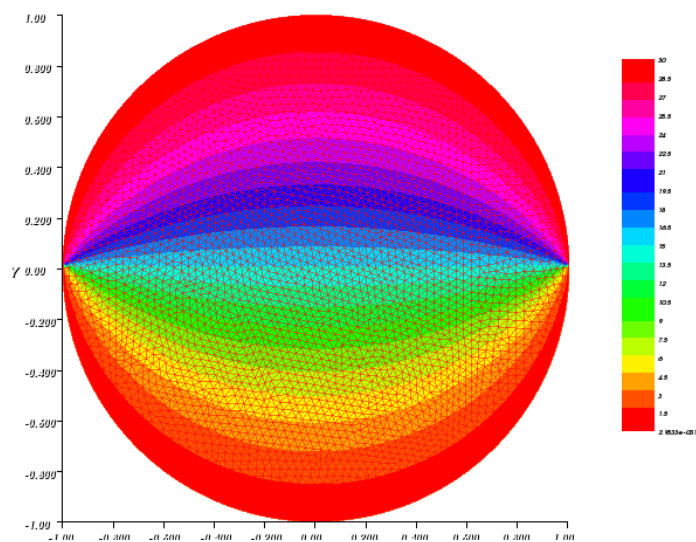


FIGURE 2.12 – Figure FreeFem++ de la distribution de la température dans la cavité circulaire (semi-circulaire)



## 2.6.4 Résolution FreeFem++ de Laplace 3D en coordonnées cartésiennes

```
1  verbosity=2;
2  load "msh3"
3
4  int [int] iparm(16);
5  real [int] dparm(6);
6  iparm=-1;
7  dparm=-1;
8
9  int nn=10;
10 mesh Th2=square(nn,nn); //Création d'un parallélépipède
11 fespace Vh2(Th2,P2);
12 Vh2 ux,uz,p2;
13 int [int] rup=[0,2], rdown=[0,1], rmid=[1,1,2,1,3,1,4,1];
14 real zmin=0,zmax=0.5;
15 mesh3 Th=buildlayers(Th2,nn,
16   zbound=[zmin,zmax],
17   labelmid=rmid,
18   reffaceup = rup,
19   reffacelow = rdown);
20 fespace Vh(Th,P2);
21 func ue = 25; // Condition aux limites ue=Ta
22 func uex= -10; // Condition aux limites uex=uey=uez=Ts
23 func uey= -10;
24 func uez= -10;
25 func f=0 ;
```

```

26 Vh uhe = ue; //
27
28 cout << " uhe min : " << uhe [].min << " max:" << uhe [].max << endl;
29 Vh u,v;
30 macro Grad3(u) [dx(u),dy(u),dz(u)] // EOM
31
32 varf Lapl3d(u,v) = //formulation variationnelle
33 int3d(Th)(Grad3(v)' *Grad3(u)) //') for emacs
34 + int2d(Th,2)(u*v)
35 + int3d(Th)(f*v)
36 + int2d(Th,2) ( ue*v + (uex*N.x + uey*N.y + uez*N.z)*v )
37 + on(1,u=ue); // conditions aux limites Ta
38
39 matrix A = Lapl3d(Vh,Vh,tgv=-1);// to remove line this BC.
40 cout << A.n << " " << A.m << " " << A.nbcoef << endl;
41 verbosity=4;
42 set(A,solver=sparsesolver ,dparams=dparm, lparams=iparm);
43 cout <<A.n <<" " << A.m << " " << A.nbcoef << endl;
44
45 real [int] b = Lapl3d(0,Vh,tgv=-1);
46
47 u[] = A^-1*b;
48 cout << " b min :: " << b.min << " max : " << b.max << endl;
49 cout << " u min :: " << u []. min << " max : " << u [].max << endl;
50 real err= int3d(Th)( square(u-ue) );
51 real aa1= int3d(Th,qfV=qfV1)(u) ;
52 real aa2= int3d(Th,qfV=qfV1lump)(u) ;
53

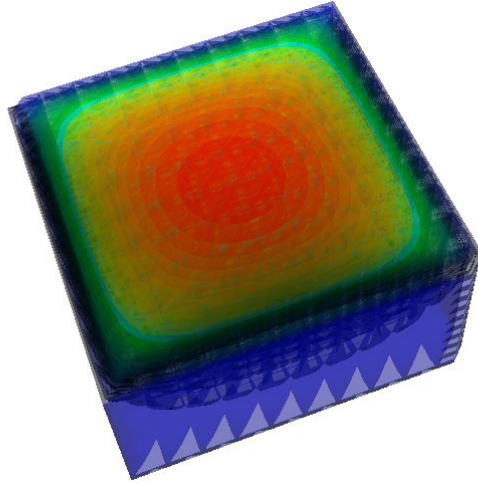
```

```

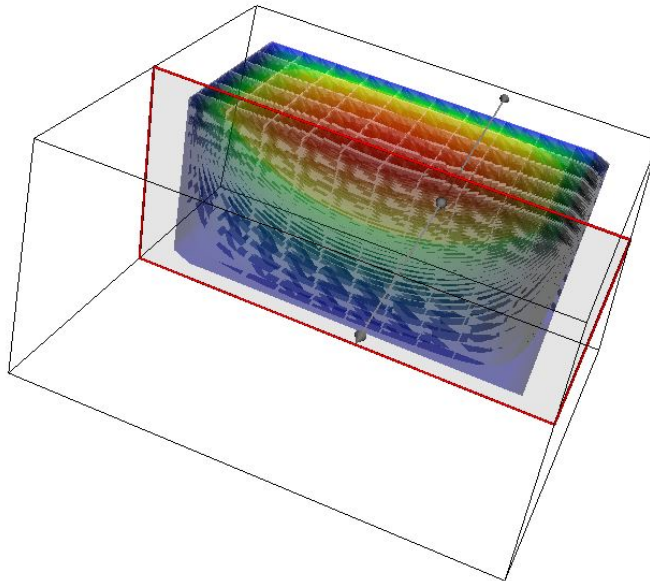
54 cout << " aa1 = " << aa1 << endl;
55 cout << " aa2 = " << aa2 << endl;
56 cout << int3d(Th)(1.) << " = " << Th.measure << endl;
57 cout << int3d(Th,qfV=qfV1)(1.) << " = " << Th.measure << endl;
58 cout << int3d(Th,qfV=qfV1lump)(1.) << " = " << Th.measure << endl;
59 Vh d= ue-u;
60 cout << " err = " << err << " diff  $l^{\int}$  = " << d[.lifty <<
    endl;
61 real aire2=int2d(Th,2)(1.); // bug correct in version 3.0-4
62 cout << " aire2 = " << aire2 << endl;
63 func uuu= 2.+x;
64 cout << uuu(1,1,1) << endl;
65 assert( abs(aire2-1.) < 1e-6);
66 plot(u,wait=1);
67 assert( err < 1e-5);

```

*Listing 2.6 – Programme Freefem++ du pavé droit 3D*



(a) Vue 3D de la distribution de la température



(b) Vue en coupe sur le plan XZ de la distribution de la température

FIGURE 2.13 – *Figure FreeFem++ de la distribution de la température dans un pavé droit*

## Chapitre 3

# Système convectif avec changement de phase

### 3.1 Mise en équation :

Dans les parties précédentes nous avons étudié le phénomène de la convection naturelle dans le cas stationnaire ( permanent) en absence de temps. On va étudier dans ce chapitre le même problème mais en introduisant le temps ainsi on va considérer que le fluide est incompressible. Donc il s'agit bien d'un problème dynamique ( variation de la vitesse de la distribution de la température). Une fois que la fusion débute, nous supposons que la masse volumique (densité) diminue quand la température augmente, mais vu que les différences de températures sont assez faibles, il est possible de dire que l'approximation de Boussinesq reste valable. Les équations de la fusion dans la couche sont ainsi données par :

$$\begin{aligned}
\rho_0 \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) &= -\nabla p + \mu \Delta \mathbf{u} + \rho_L g \\
\nabla \cdot \mathbf{u} &= 0 \\
\rho_L &= \rho(1 - \beta(T - T_A)) \\
\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T &= \kappa \Delta T
\end{aligned}$$

Avec  $\mathbf{u}(x(t), t)$  est le vecteur vitesse,  $\mu$  la viscosité dynamique,  $p(x(t), t)$  la pression,  $\rho_0$  la densité de référence à  $T_A$ ,  $T(x(t), t)$  le champ de température,  $g$  le vecteur de pesanteur,  $t$  le temps et  $\kappa$  la conductivité thermique.

Il faut noter que d'après l'approximation de Boussinesq, la masse volumique dépend linéairement de la température, le coefficient de dilatation thermique volumétrique est  $\beta > 0$ . Dans la suite on supposera que le fluide est incompressible.

Les conditions aux limites du problème en coordonnées cartésiennes sont : (i) : Température fixe sur le mur inférieur, avec une condition de non-glissement de la vitesse, (ii) : conditions à la limite périodiques sur les extrémités verticales, (iii) : Condition de non-glissement et conditions de fusion à l'interface de changement de phase. Ce qui nous donne : **Conditions aux limites en coordonnées cartésiennes :**

$$\begin{aligned}
\mathbf{T}|_{\mathbf{x}=(x,y,0)} &= T_s \quad \forall x, y \in [0, L][0, H] \\
\mathbf{u}|_{\mathbf{x}=(x,y,0)} &= \mathbf{0} \quad \forall x, y \in [0, L][0, H] \\
-\kappa \nabla T|_{\mathbf{x}=x_m(t)} &= \frac{L_f}{c_p} \dot{\mathbf{x}}_m(t) \quad \forall \mathbf{x}_m(t) \in \Gamma(t) \\
\mathbf{u}|_{\mathbf{x}=(x,y,0)} &= \mathbf{0} \quad \forall \mathbf{x}_m(t) \in \Gamma(t)
\end{aligned}$$

**Conditions aux limites en coordonnées sphériques :**

$$\begin{aligned}
\mathbf{T}|_{\mathbf{Y}=(r=R,\theta,\phi)} &= T_s \quad \forall \theta, \phi \in [\pi, 2\pi][0, \pi] \\
\mathbf{u}|_{\mathbf{Y}=(r=R,\theta,\phi)} &= \mathbf{0} \quad \forall \theta, \phi \in [\pi, 2\pi][0, \pi] \\
-\kappa \nabla T|_{\mathbf{Y}=x_m(t)} &= \frac{L_f}{c_p} \dot{\mathbf{x}}_m(t) \quad \forall \mathbf{Y}_m(t) \in \Gamma(t) \\
\mathbf{u}|_{\mathbf{Y}=(R,\theta,\phi)} &= \mathbf{0} \quad \forall \mathbf{x}_m(t) \in \Gamma(t)
\end{aligned}$$

Où  $L_f$  est la chaleur latente,  $c_p$  la capacité thermique,  $\mathbf{x}_m(t)$  sont les coordonnées Lagrangiens des points dans l'interface fluide-solide notée  $\Gamma(t)$ , au temps  $t$  et  $\dot{\mathbf{x}}_m(t)$  est la vitesse de l'interface. X et Y sont respectivement les bases des coordonnées cartésiennes et sphériques.

## 3.2 Résolution numérique du problème de changement d'état en C

On va procéder à résoudre numériquement le système des équations de l'hydrodynamique et donc savoir ce qui se passe pendant le changement de phase dans une cavité semi-sphérique et parallélépipédique. Pour cela on rédigera un programme sur C tenant en compte les formes de solutions trouvées précédemment pour les deux formes de cavités en deux dimensions et en 3 dimensions.

```

1  #ifdef LB_TEMPERATURE_MELTING_INITIAL_LIQUID_CAVITY_TEMPERATURE
2      x = abs(center_V[IDX(i, j, k)].x - property.SX/2.0);
3      y = center_V[IDX(i, j, k)].y;
4      z = abs(center_V[IDX(i, j, k)].z - property.SZ/2.0);
5
6      #ifdef GRID_POP_D2Q9
7          if( x < property.cavity_SX/2 && y < property.cavity_SY )
8          {
9              my_double DT,A,mpx;
10             DT = property.T_bot - property.T_top;
11             for (int m=1;m<100;m++)
12             {
13                 mpx = m*one_pi/property.cavity_SX;
14                 A = 2*DT*(1-cos(m*one_pi))/(m*one_pi*sinh(mpx*property.cavity_SY));
15                 t[IDX(i, j, k)] += A*sin(mpx*(x+property.cavity_SX/2))*sinh(mpx*(property.cavity_SY-y));

```

```

15     }
16 }
17 #endif
18 #ifdef GRID_POP_D3Q19
19 if( x < property.cavity_SX/2 && y < property.cavity_SY && z <
    property.cavity_SZ/2)
20 {
21     my_double DT,mpx,npz,K,A;
22     DT = property.T_bot - property.T_top;
23     for (int m=1;m<100;m++){
24         for (int n=1;n<100;n++){
25             mpx = m*one_pi/property.cavity_SX;
26             npz = n*one_pi/property.cavity_SZ;
27             K = sqrt(mpx*mpx + npz*npz);
28             A = 4*DT*(1-cos(n*one_pi))*(1-cos(m*one_pi))/(m*n*
                one_pi*one_pi*sinh(K*property.cavity_SY));
29             t[IDX(i,j,k)] += A*sin(mpx*(x+property.cavity_SX/2))*
                sin(npz*(z+property.cavity_SZ/2))*sinh(K*(
                property.cavity_SY-y));
30         }
31     }
32 #endif
33 #endif //LB_TEMPERATURE_MELTING_INITIAL_LIQUID_CAVITY

```

*Listing 3.1 – Programme C de la fusion en 2D et 3D en coordonnées cartésiennes*

```

1 #ifdef LB_TEMPERATURE_MELTING_INITIAL_LIQUID_SEMISPHERE_TEMPERATURE
2     x = center_V[IDX(i,j,k)].x - property.SX/2.0;

```



```

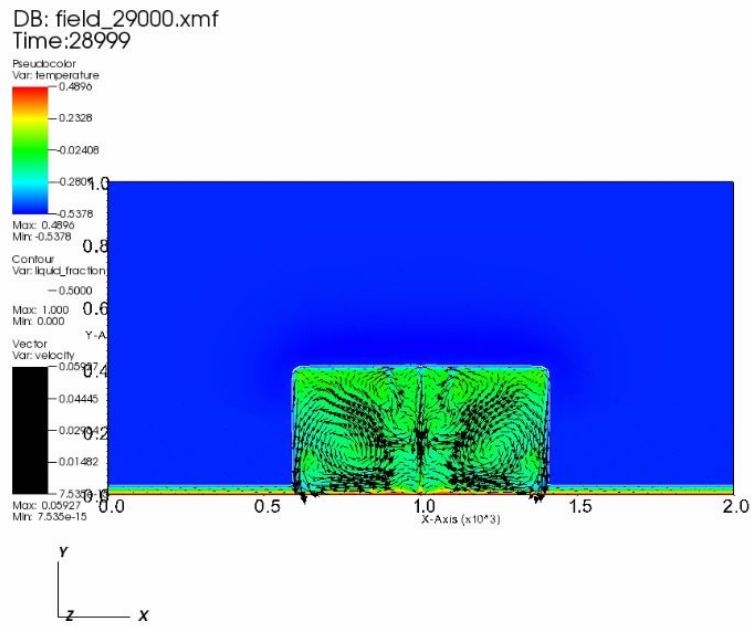
3   y = center_V[IDX(i, j, k)].y;
4   z = center_V[IDX(i, j, k)].z - property.SZ/2.0;
5   val = sqrt(x*x + y*y + z*z);
6   if( val < property.cavity_radius ){
7       my_double DT, radius;
8       DT = property.T_bot - property.T_top;
9       radius = val / property.cavity_radius;
10      t[IDX(i, j, k)] = property.T_bot;
11      for (int m=1; m<100; m++)
12          t[IDX(i, j, k)] += (4*DT/one_pi)*( pow(radius, 2*m-1) * sin( (2*
13              m-1) * atan2(-y, sqrt(x*x+z*z)) ) ) / (2*m-1);
14  #endif //LB_TEMPERATURE_MELTING_INITIAL_LIQUID_SEMISPHERE

```

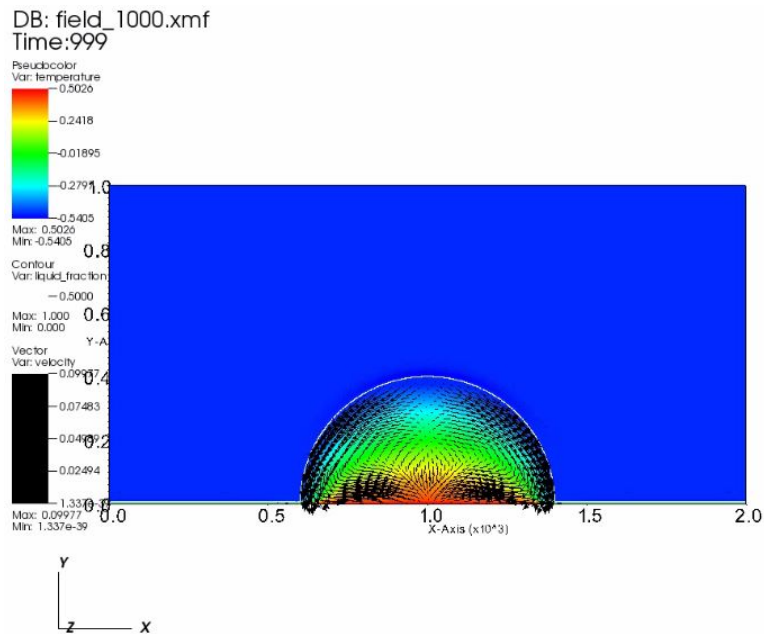
*Listing 3.2 – Programme C de la fusion en 2D et 3D en coordonnées polaires*

On lance le simulateur numérique, et après 10 secondes on constate le début de la fusion et donc un changement de phase.

Nous remarquons que les deux cavités engendrent un même profil de fusion à la fin, même si leurs évolutions semblent différentes -les deux profils prennent des formes différentes au début de la fusion-, leurs 2 vitesses sont égales et donc convergent à la même forme à la fin.

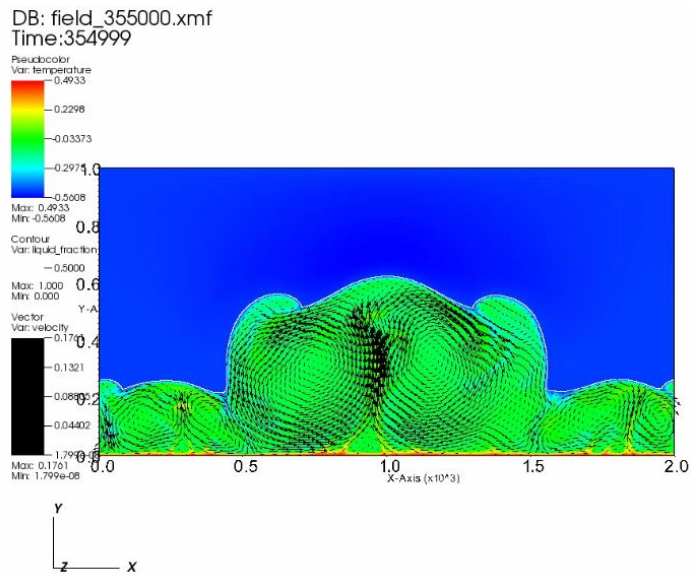


(a) Cavité parallélépipédique avant fusion à  $t=0$

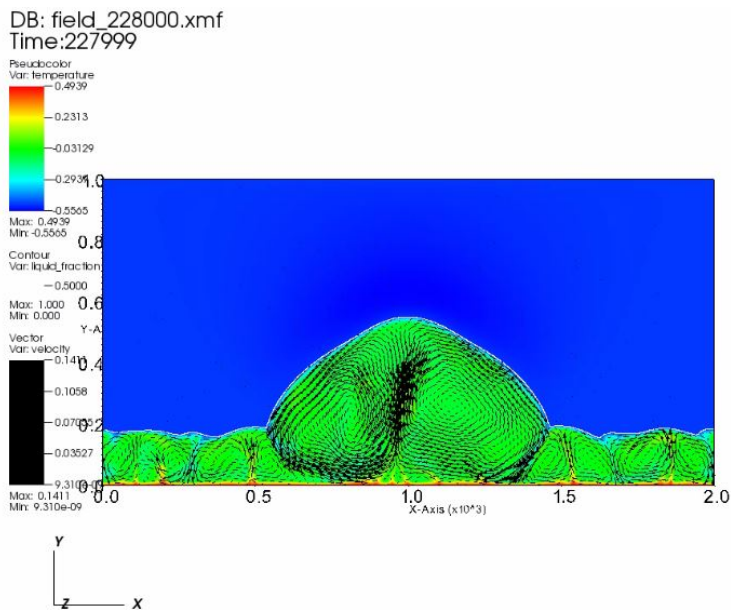


(b) Cavité semi-circulaire avant fusion à  $t=0$

48  
FIGURE 3.1 – Comparaison entre les deux cavités à l'état initial

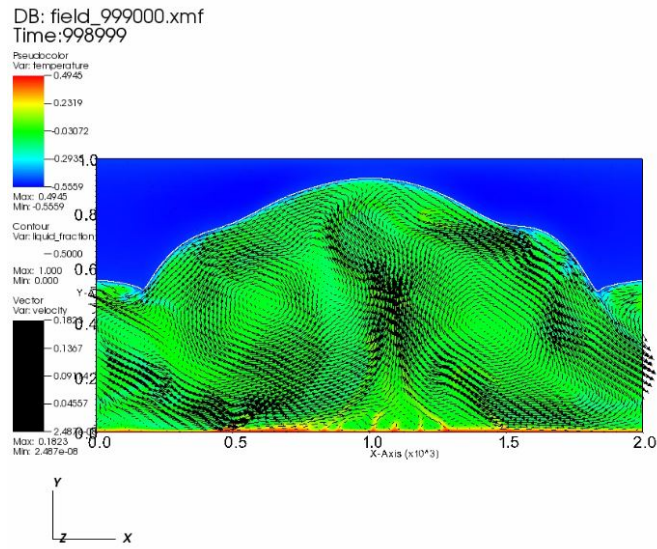


(a) Cavité parallélépipédique pendant la fusion à  $t=25\text{sec}$

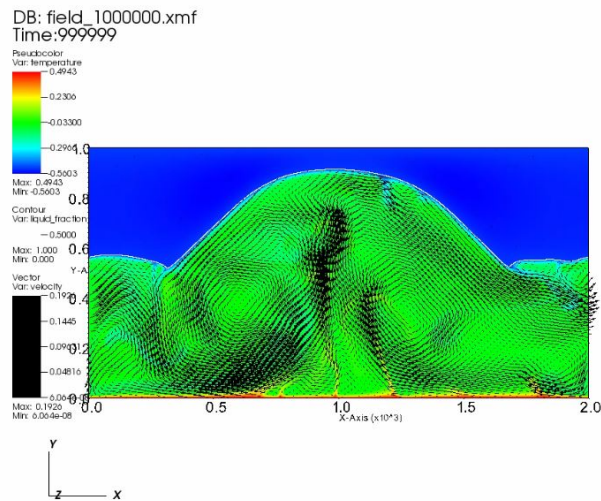


(b) Cavité semi-circulaire pendant la fusion à  $t=23\text{sec}$

FIGURE 3.2 – Comparaison entre les deux cavités pendant le changement d'état



(a) Cavité parallélépipédique à l'état final de fusion  $t=100s$



(b) Cavité semi-sphérique à l'état final de fusion à  $t=100s$

FIGURE 3.3 – Comparaison entre les cavités à la fin du processus de fusion à  $t=100sec$

# Conclusion

L'étude théorique et numérique de la convection naturelle dans deux géométries différentes nous a permis de connaître la distribution de température au niveau de la cavité semi-sphérique et parallélépipédique. À l'aide des équations de l'hydrodynamique nous avons pu suivre graphiquement la fusion de la glace et donc conclure à la fin l'allure du champ de température. Les deux cas sont appliquées à la réalité puisque le flux de la chaleur va du chaud vers le froid, autrement dit la température diminue en s'approchant du solide (glace). L'approche que nous avons faite reste particulière mais valable quelque soit la géométrie et la nature de l'écoulement.

## Table des figures :

2.1	Conditions aux limites du problème total en 2D . . . . .	7
2.2	Conditions aux limites des quatre problèmes en 2D . . . . .	8
2.3	Conditions aux limites du problème 3D en coordonnées cartésiennes . . . . .	13
2.4	Conditions aux limites pour le demi-disque . . . . .	18
2.5	Figure Python : Distribution de la température dans une cavité rectangulaire avec "Ta=25,Ts=-5,L=4,H=2" . . . . .	25
2.6	Figure Python :Distribution surfacique de la température dans une cavité rectangulaire (z=axe de la température) . . . . .	26
2.7	Figure Python de la distribution de la température dans la cavité circulaire(semi-circulaire) . . . . .	29
2.8	Graphe programme 2 Python correspond à la distribution de la température dans la semi-circulaire . . . . .	31
2.9	Figure FreeFem++ de la distribution de la température dans la cavité rectangulaire . . . . .	36
2.10	Figure FreeFem++ de la distribution de la température et du maillage choisi dans la cavité rectangulaire . . . . .	36
2.11	Figure FreeFem++ de la distribution surfacique de la température dans la cavité rectangulaire (z=T) . . . . .	37
2.12	Figure FreeFem++ de la distribution de la température dans la cavité circulaire (semi-circulaire) . . . . .	38
2.13	Figure FreeFem++ de la distribution de la température dans un pavé droit . . . . .	42
3.1	Comparaison entre les deux cavités à l'état initial . . . . .	48
3.2	Comparaison entre les deux cavités pendant le changement d'état . . . . .	49

3.3	Comparaison entre les cavités à la fin du processus de fusion	
	à $t=100\text{sec}$ . . . . .	50

# Listings

2.1	Programme Python 2D en coordonnées cartésiennes . . . . .	26
2.2	Programme Python pour la cavité semi-circulaire en 2D . . . .	29
2.3	Programme Python pour le pavé droit (3D) . . . . .	30
2.4	Programme Freefem pour la cavité rectangulaire 2D . . . . .	34
2.5	Programme Freefem++ pour la cavité demi-circulaire (2D) . .	35
2.6	Programme Freefem++ du pavé droit 3D . . . . .	39
3.1	Programme C de la fusion en 2D et 3D en coordonnées cartésiennes . . . . .	45
3.2	Programme C de la fusion en 2D et 3D en coordonnées polaires	46



# Bibliographie

- [1] Yuxi ZHENG. *PDE on rectangular domains, separation of variables*.  
URL : [www.math.psu.edu/yzheng/m597k/m597kLVI7.pdf](http://www.math.psu.edu/yzheng/m597k/m597kLVI7.pdf).
- [2] *Solution to Laplace Equation In Cartesian Coordinates*. URL : [http://nsmn1.uh.edu/hunger/class/fall\\_2010/lectures/lecture\\_6.pdf](http://nsmn1.uh.edu/hunger/class/fall_2010/lectures/lecture_6.pdf).
- [3] Michael SHEARER et Rachel LEVY. *Partial Differential Equations An Introduction to Theory and Applications*. 2015, p. 142–147.
- [4] F.HECHT. *FreeFem++ Third Edition, Version 3.55-1*. URL : <http://www.freefem.org/ff++/ftp/freefem++doc.pdf>.