UNIVERSITÉ CHEIKH ANTA DIOP

FACULTÉ DES SCIENCES ET TECHNIQUES
DÉPARTEMENT DE MATHÉMATIQUES ET INFORMATIQUE

Numerical Methods Project

Presented by

Jean-Michel Amath Sarr

Greetings

We thanks our Professor Mountaga Lam, Doctor at the Faculty of Science and Technology in Cheikh Anta Diop University of Dakar, to have taught us numerical simulation through the software Freefem++ and C++.

Chapitre Un

Heat diffusion simulation with the finite element method

Introduction

The goal of this project is to solve heat equation in two dimensions with the finite element method. We study the evolution of the temperature $u(x_1, x_2, t)$ on a domain Ω through time under specific boundary conditions on a fins of a given material.

When using the finite element methods, we first discretize the equation in time, then we express the weak formulation. From there, we mesh the domain, to obtain a completely discrete problem, hence we can solve it using linear algebra methods. In particular, we use the conjugate gradient method. Finally, we will compare and interpret the results.

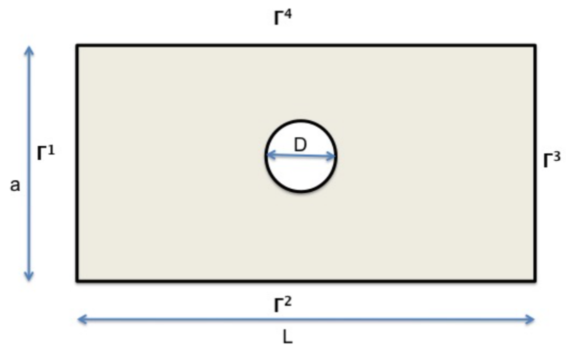
We suppose that the reader has already knowledge in weak formulation of elliptic partial derivative equations, and in the theory behind the finite element method.Our purpose is to share our approach to simulate the heat diffusion, not to prove any theorem nor result. For further information there is a good course available on the web [1] on weak formulation of elliptic problems. The code is also well documented and comes from the following books on C++ simulation : [2], [3].

The mesh generation is obtained using the software Freefem++, the code to solve the problem is written in C++.

1.1 System modelization

Notations : $x = (x_1, x_2) \in \Omega$
Let us consider the following system :

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} - \Delta u(x,t) = 0 & x \in \Omega,\ 0 \leq t \leq t_{max} \\ u = \frac{T_0 - T_e}{T_e} & x \in \Gamma^1,\ 0 \leq t \leq t_{max} \\ \frac{\partial u(x,t)}{\partial n} & x \in \Gamma^0 \cup \Gamma^2 \cup \Gamma^4,\ 0 \leq t \leq t_{max} \\ \frac{\partial u(x,t)}{\partial n} + (\frac{h_c}{L} K) u(x,t) = 0 & x \in \Gamma^3,\ 0 \leq t \leq t_{max} \\ u(x_1, x_2, t = 0) = u^0(x_1, x_2) \end{cases}$$



$L = 40;\ a = 25;\ D = 8$

FIGURE 1.1 – Domain Ω in yellow

1.2 Time discretization

For the discretization, we use the implicit Euler method, let partition the interval $[0, t_{max}]$ in N uniform subdivisions. , We denote u^n the temperature at time $n\delta t$ and $N\delta t = t_{max}$. The we get :

$$u(x, t + \delta t) - u(x,t) = \frac{\partial u(x,t)}{\partial t} \delta t + o(\delta t)$$

We can use the following approximation :

$$\frac{\partial u(x,t)}{\partial t} \simeq \frac{u(x, t+\delta t) - u(x,t)}{\delta t}$$

Using our notation the discretize equation is :

$$\frac{u^{n+1} - u^n}{\delta t} - \Delta u^{n+1} = 0\ \forall x \in \Omega \tag{1.1}$$

1.3 Weak formulation

Let us introduce the following notations :

— $\Gamma^D = \Gamma^1$ (Dirichlet).
— $\Gamma^N = \Gamma^0 \cup \Gamma^2 \cup \Gamma^4$ (Neumann).
— $\Gamma^R = \Gamma^3$ (Robin)
— $L^2(\Omega) = \{u : \Omega \to \mathbb{R} | \int_\Omega u^2\, dx < \infty\}$
— $H^1 = \{u \in L^2(\Omega) | \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2} \in L^2(\Omega)$
— $V = \{u \in H^1(\Omega),\, u_{|\Gamma^D} = 0\}$

We suppose that $\Delta u \in L^2(\Omega)$. In the following we suppose u regular enough so that our calculations are correct We multiply the equation (1.1) by $v \in V$

$$\int_\Omega (\frac{u^{n+1} - u^n}{\delta t} - \Delta u^{n+1}) v = 0$$

Then with the Green identity we get :

$$\int_\Omega (\frac{u^{n+1} - u^n}{\delta t}) v\, dx + \int_\Omega \nabla u^{n+1} \nabla v\, dx = \int_{\partial\Omega} \frac{\partial u^{n+1}}{\partial n} v\, d\sigma$$

Developing the boundary term we get :

$$\int_{\partial\Omega} \frac{\partial u^{n+1}}{\partial n} v\, d\sigma = \int_{\Gamma^D} \frac{\partial u^{n+1}}{\partial n} v\, d\sigma + \int_{\Gamma^N} \frac{\partial u^{n+1}}{\partial n} v\, d\sigma + \int_{\Gamma^R} \frac{\partial u^{n+1}}{\partial n} v\, d\sigma$$

Thus

$$\int_{\partial\Omega} \frac{\partial u^{n+1}}{\partial n} v\, d\sigma = -\int_{\Gamma^R} (\frac{h_c}{L} K) u^{n+1} v\, d\sigma$$

Finally the weak formulation is as follows :
Find $u^{n+1} \in H^1(\Omega)$, verifying $u^{n+1} = \frac{T_0 - T_e}{T_e}$ so that $\forall v \in V$

$$\int_\Omega (\frac{u^{n+1} - u^n}{\delta t}) v\, dx + \int_\Omega \nabla u^{n+1} \nabla u^n\, dx + \int_{\Gamma^R} (\frac{h_c}{L} K) u^{n+1} v\, d\sigma = 0 \tag{1.2}$$

Given the regularity Lemma we know that a weak solution of (1.2) is also a solution of (1.1) if the following regularity condition : $u^{n+1} \in H^1(\Omega)$ et $\Delta u^{n+1} \in L^2(\Omega)$ is held.

1.4 Discretization of Ω

A mesh $\mathcal{T}$ of Ω verify :

— $|\mathcal{T}| < \infty$

— $\forall K \in \mathcal{T}$, $\mathring{K} \neq \emptyset$ et $\overline{K} = K$

— $\overline{\Omega} \subset \bigcup_{K\in\mathcal{T}} K$

— $K, U \in \mathcal{T} \Rightarrow \mathring{K} \cap \mathring{U} = \emptyset$

We make a triangulation of Ω and we denote $(T_i)_{i=1,\cdots,n_t}$ the set of all triangles in the mesh of Ω, n_t the number of triangles and n_v the number of vertices. We define the following spaces : $H_h = \{v \in C^0(\Omega),\ v_{|T_i} \in \mathcal{P}_1(T_i),\ i = 1, \cdots, n_t\}$ and $V_h = \{v \in H_h,\ v_{|\Gamma^D} = 0\}$ as approximation of $H^1(\Omega)$ and V respectively.

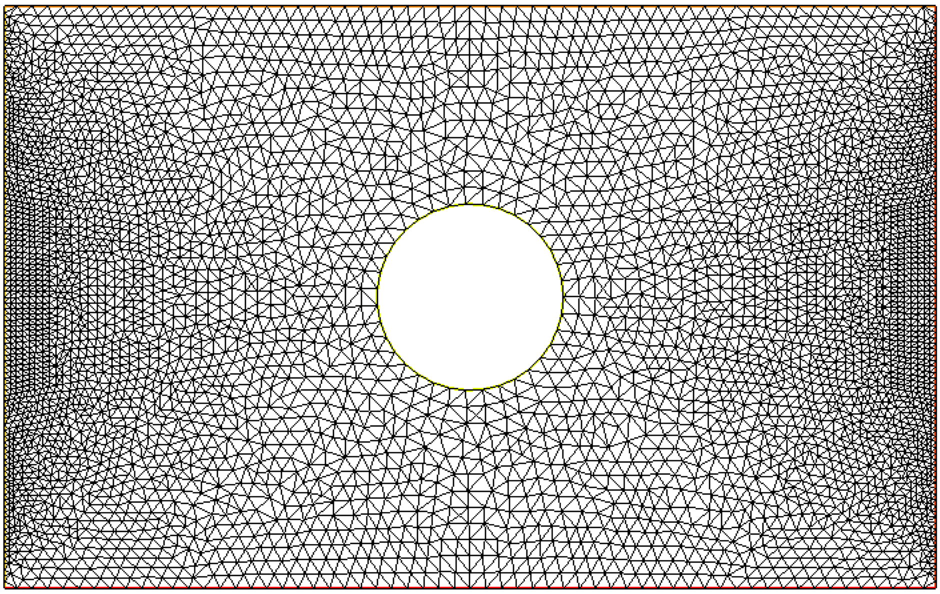


FIGURE 1.2 – Meshing of the Domain

We then get the following discrete weak formulation :

1.5 Discrete Weak Formulation

We introduce the notation :

— Ω_h the meshed domain ; $\Omega_h = \bigcup_{k=0}^{n_t} T_k$

— J the set of all vertices on Γ_h^D.

— I the set of all vertices on $\Omega_h \setminus \Gamma_h^D$.

— $q^i \in \mathbb{R}^2$ the i-th vertice and $u_i^n = u_h^n(q^i)$.

The problem then becomes :

Find $u_h^{n+1} \in H_h$ verifying $u_h^{n+1}(q^j) = \frac{T_0 - T_e}{T_e}\ \forall j \in J$ so that $\forall v \in V_h$

$$\int_{\Omega_h} (\frac{u_h^{n+1} - u_h^n}{\delta t}) v \, dx + \int_{\Omega_h} \nabla u_h^{n+1} \nabla v \, dx + \int_{\Gamma_h^R} (\frac{h_c}{L} K) u_h^{n+1} v \, d\sigma = 0 \tag{1.3}$$

Proposition 1.1. *The dimension of the space H_h is equal to the number of vertices in the triangulation.*
Proposition 1.2. *The functions in H_h are totally characterized by their values at the vertices of the triangulation.*

We can show that a basis of H_h is given by the following set :

$$\{w^i \in H_h, \ w^i(q^j) = \delta_{ij} | i, j = 1, \cdots, n_v\}$$

with δ_{ij} the Kronecker symbol : $\delta_{ij} = \begin{cases} 1 & if \ i = j \\ 0 & elsewhere \end{cases}$

Hence we can write any $f_h \in H_h$ in the following manner :

$$u_h^n(x) = \sum_{i \in I \cup J} f_i \, w^i(x)$$

Thus, developing the equation (1.3) we get the more general formula :

$$\sum_{i \in I \cup J} \Big[\frac{1}{\delta t} \int_{\Omega_h} w^i w^j dx + \int_{\Omega_h} \nabla w^i \nabla w^j dx + \int_{\Gamma^R} (\frac{h_c}{L} K) w^i w^j d\sigma \Big] u_i^{n+1} = \tag{1.4}$$
$$\sum_{i \in I \cup J} \Big[\frac{1}{\delta t} \int_{\Omega_h} w^i w^j dx \Big] u_i^n \quad \forall j \in I \cup J$$

To simplify our expression, we note the constant $c = (\frac{h_c}{L} K)$, we then obtain a linear system :

$$\mathcal{A}^{(1,1,c)} \, \mathcal{U}^{n+1} = \mathcal{A}^{(1,0,0)} \, \mathcal{U}^n \tag{1.5}$$

With $\mathcal{U}$ the solution, a vector of dimension n_v and $\mathcal{A}^{(\alpha,\beta,\gamma)}$ a matrix of the same dimension define as follows $\forall i, j \in I \cup J$:

$$\mathcal{A}_{(i,j)}^{(\alpha,\beta,\gamma)} = \frac{1}{\delta t} \int_{\Omega_h} \alpha \, w^i w^j dx + \int_{\Omega_h} \beta \, \nabla w^i \nabla w^j dx + \int_{\Gamma^R} \alpha_R \, w^i w^j d\sigma \tag{1.6}$$

Where α, β, γ are functions defined over the triangulation.

The linear system defined in (1.5) is then solved with the gradient conjugate method.

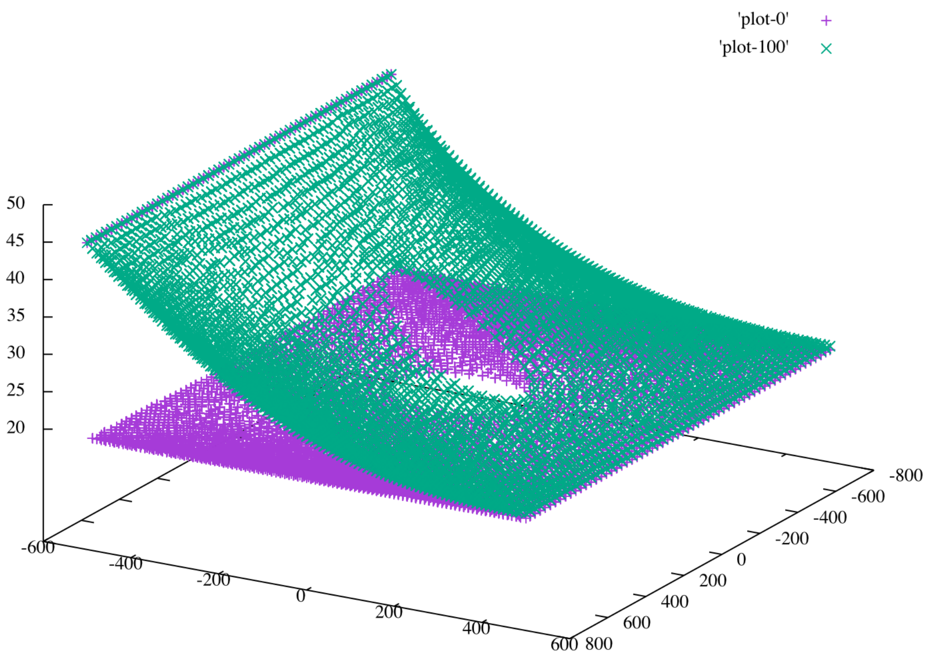


FIGURE 1.3 – Comparison between the solutions at t=0 and at t = 100

1.6 Results

We notice that the heat diffuse on the fins relatively quickly, but even after 100 iterations the temperature on the Robin boundary as not evolved very much.

Between the iterations 100 et 1000, the heat has diffused in a homogeneous manner on the fins.

One of the main remark when executing this simulation is the high number of iterations before convergence. It can be explained by the requirements in the main program. In fact we demand that the norm of the difference between $\mathcal{U}^{n-1}$ and $\mathcal{U}^{n}$ to be less than $\varepsilon = 1^{-5}$.

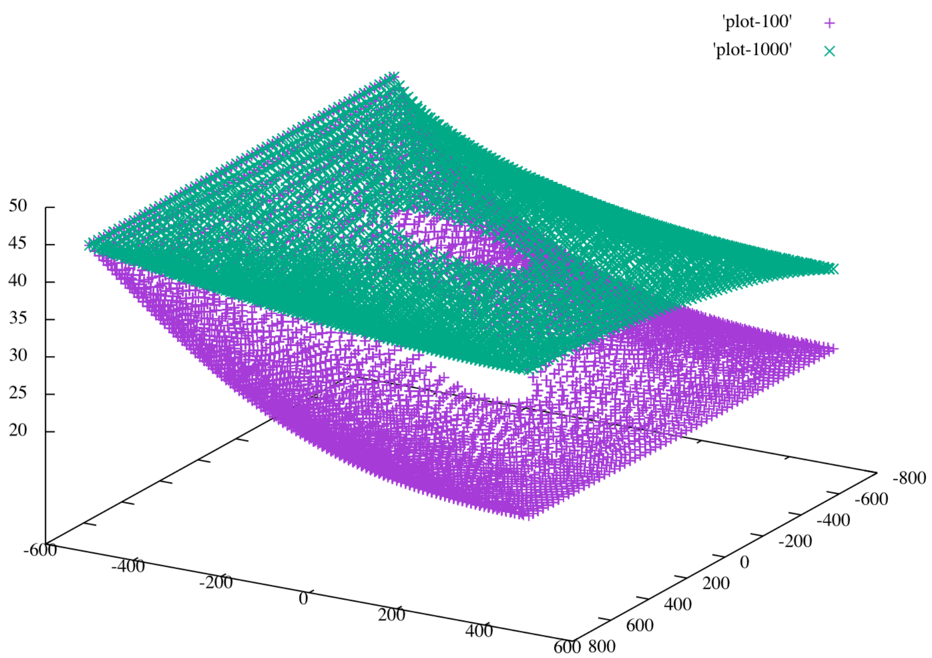


FIGURE 1.4 – Comparison between the solutions at t=100 and at t = 1000

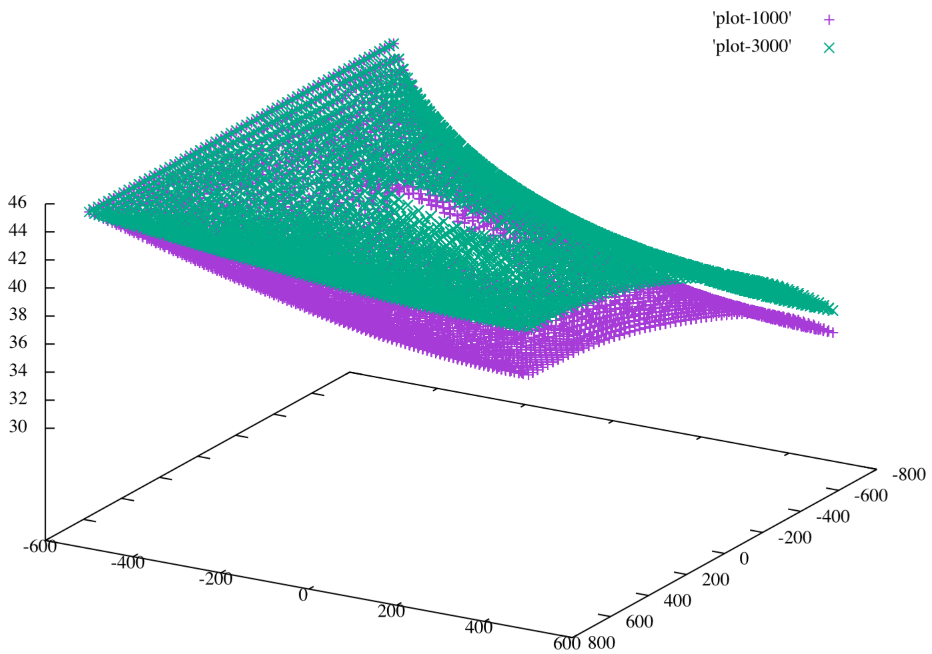


FIGURE 1.5 – Comparison between the solutions at t=1000 and at t = 3000

Conclusion

To conclude we can say that the finite element method give us an efficient and flexible approach to solve partial derivative equations. Efficient because it allows us to solve the equation using solutions on approximation spaces, that the theory shows it converges to the analytical solution. Flexible because the meshing method used to discretize the domain, can be adapted to any open set.

Remarque 1.1. The main disadvantage of this simulation is that the data generated are quite big ($> 2\,Gb$), and the computing time is consequent. We have made other simulations, with finer mesh or with other dimensions of the fins. Here are what we notice :

- A finer mesh increase the computing time. In a sense it is to expect ; in fact, at each iteration, we compute the norm between two vectors in $\mathbb{R}^{n_v}$. Hence a coarse mesh can make the computation faster and give us an idea of the number of iteration required to converge.
- A large area increases the number of iterations required for the solution to converge. This is also comprehensible, In our macroscopic world, heat a little room will be quicker than a large one given the same source.

We can also act on the value of ε to modify the convergence requirements.

Bibliographie

[1] Course UPMC, UDC : *Chapter 4 Weak formulation of elliptic problems*, $https://www.ljll.math.upmc.fr/frey/cours/UdC/ma691$

[2] Frédéric Hech : *Quelques Idées d'utilisation du C++*, Cours Informatique de Base 2011-2012, Master 1 Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie

[3] Ionut Danaila, F. Hecht, Pironneau : *Simulation numérique en C++*, 2015 Dunod, Sciences sup