Report of PAP project : heat equation in 1D and 2D

Hao Liu
Faryadi Mohammed

21^{st} January 2018

Contents

1 Introduction

The advanced processing project consists in handling one of the four proposed subjects. The subject which we chose is the one about heat equations in 1D and 2D which aims to feign the evolution of the heat in a material, by means of the partial differential equation which describes the propagation of heat. We will dedicate our study to four types of materials which are iron, copper, polystyrene and glass.

We are brought to solve the equation of the heat for the previous materials by using the finite differences. At first, we are invited to study this distribution in an one-dimensional bar, and then in the second part the study of the distribution in a two-dimensional plate.
The conditions to respect are the following ones :

- For the bar :
- x = 0, Newman condition.
- x = L, Dirichlet condition.

- For the plate :
- x = 0; y = 0, Newman condition.
- x = L; y = L, Dirichlet condition.

To model the approached solutions, we have to implement a C++ program, and finally to display the obtained results we have to create an animation on Scilab. To find an approached solution for the heat equation we will build an approximation with the finite differences method. For this purpose we will divide our interval [0,L] to $\Delta x = 1000$ intervals and also our time interval [0,t] with a step of Δt.

2 Approach 1D : The bar

2.1 Mathematical approach

We consider the one-dimensional heat equation where we seek a u(x,t) satisfying:

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\lambda}{\rho c}\frac{\partial^2 u}{\partial x^2} + \frac{F}{\rho c}, \forall (t,x) \in [0, T_{max}] \times [0, L] \\ u(0,x) = u_0, \forall x \in [0, L] \\ \frac{\partial u}{\partial x}(t,0) = 0, \forall t \in [0, T_{max}] \\ u(t,L) = u_0, \forall t \in [0, T_{max}] \end{cases}$$

We will also have a heat source F that simulates 2 additions of heat of temperature f and $\frac{3}{4}f$:

$$\begin{cases} F(x) = t_{max} f^2, on[\frac{L}{10}, \frac{L}{10}] \\ F(x) = \frac{3}{4} t_{max} f^2, on[\frac{5L}{10}, \frac{6L}{10}] \\ F(x) = 0, else \end{cases}$$

Now we will derive the scheme by using backward difference approximation. Then we write the scheme at the point (x_i, t^n) so that the difference equation now becomes:

$$\frac{U_i^n - U_i^{n-1}}{\Delta t} = \frac{\lambda}{\rho c} \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{(\Delta x)^2} + \frac{1}{\rho c} F(x_i, t^n)$$

for i = 1,...,10001 because it is requested to split our interval into 1000 points. Now when we simplify this expression, we see that only U_i^{n-1} is known and we move it to the right hand side. The equation then becomes :

$$-\alpha U_{i+1}^n + (1 + 2\alpha) U_i^n - \alpha U_{i-1}^n = U_i^{n-1} + \frac{\Delta t}{\rho c} F(x_i, t^n)$$

with

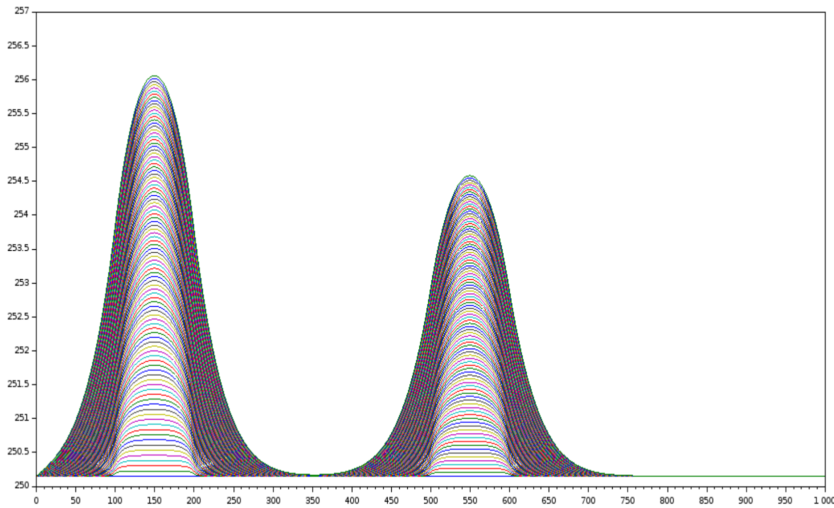
$$\alpha = \frac{\lambda \Delta t}{\rho c (\Delta x)^2}$$

We can no longer solve for U_1^n and then U_2^n , etc. All of the values U_1^n , U_2^n... are coupled. We must solve for all of them at once. This requires us to solve a linear system at each time-step. This is the **implicit way**.
To solve our linear system we should be able to solve a tridiagonal system with the help of Cholesky's decomposing and it is in this spirit that comes our C++ program.

2.2 Computing approach with C++ and Scilab

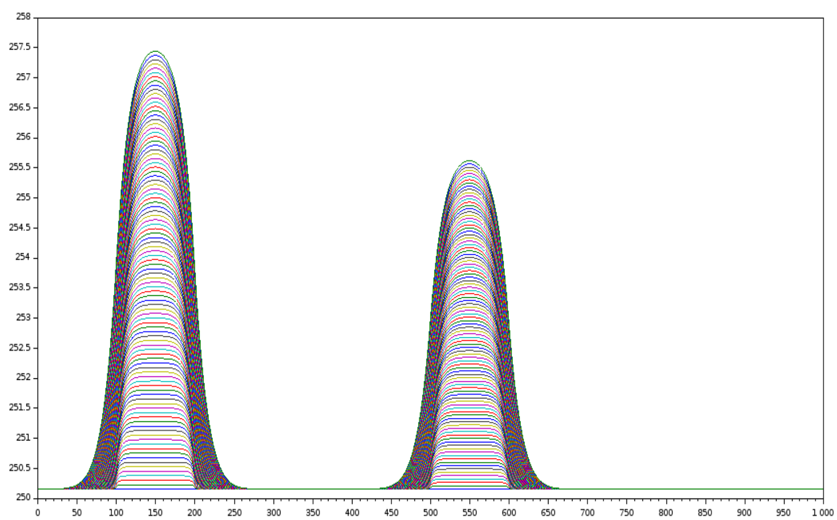
After doing the simulations on Scilab we achieved the following graphs :

For the copper :



- The 1^{rst} heat source heats the material up to a maximum temperature of 256 kelvin.
- The 2^{nd} heat source heats the material up to a maximum temperature of 254.75 kelvin.

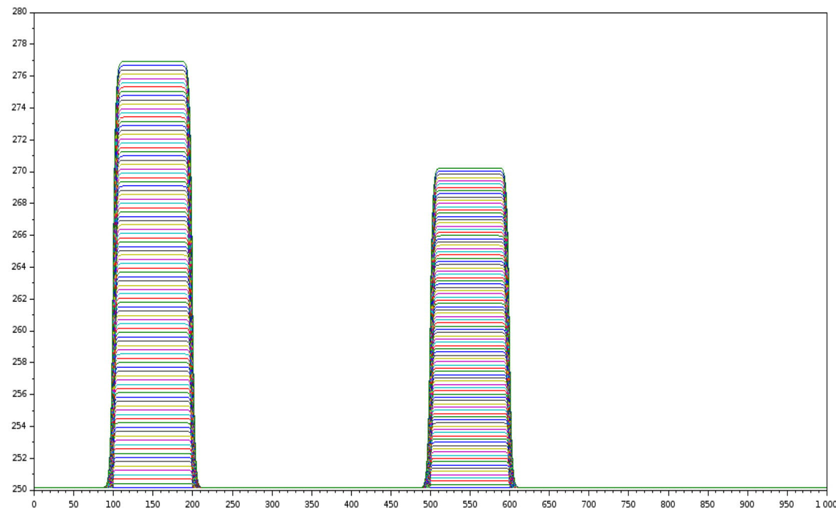
For the iron :



- The 1^{rst} heat source heats the material up to a maximum temperature of 257.5 kelvin.
- The 2^{nd} heat source heats the material up to a maximum temperature of 255.5

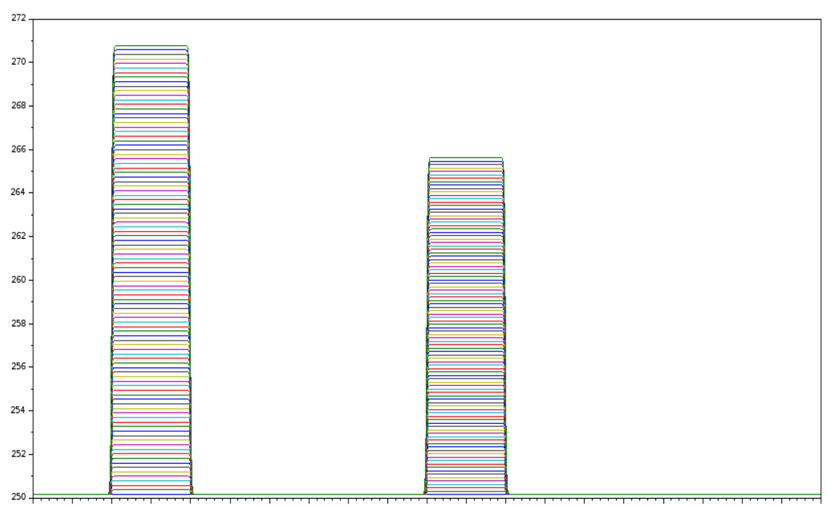
kelvin.

For the glass :



- The 1^{rst} heat source heats the material up to a maximum temperature of 277 kelvin then remains constant for a while before decreasing.
- The 2^{nd} heat source heats the material up to a maximum temperature of 271 kelvin then remains constant for a while before decreasing.

For the polystyrene :



- The 1^{rst} heat source heats the material up to a maximum temperature of 271 kelvin then remains constant for a while before decreasing.
- The 2^{nd} heat source heats the material up to a maximum temperature of 266 kelvin then remains constant for a while before decreasing.

Note :
We could say that there is a loss of heat for the iron and the copper we could see that in the edges of each curve, unlike the glass or polystyrene where we cannot note if there is an important loss of heat.

3 Approach 2D : The plate

3.1 Mathematical approach

We consider the two-dimensional heat equation where we seek a u(x,y,t) satisfying:

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\lambda}{\rho c}(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) + \frac{F}{\rho c}, \forall (t,x,y) \in [0,T_{max}] \times [0,L]^2 \\ u(0,x,y) = u_0, \forall (x,y) \in [0,L]^2 \\ \frac{\partial u}{\partial x}(t,0,y) = 0, \forall (t,y) \in [0,T_{max}] \times [0,L] \\ \frac{\partial u}{\partial y}(t,x,0) = 0, \forall (t,x) \in [0,T_{max}] \times [0,L] \\ u(t,L,y) = u_0, \forall (t,y) \in [0,T_{max}] \times [0,L] \\ u(t,x,L) = u_0, \forall (t,x) \in [0,T_{max}] \times [0,L] \end{cases}$$

We will also have a heat source F :

$$\begin{cases} F(x) = t_{max} f^2, on[\frac{L}{6}, \frac{2L}{6}] \times [\frac{L}{6}, \frac{2L}{6}] \\ F(x) = t_{max} f^2, on[\frac{4L}{6}, \frac{5L}{6}] \times [\frac{L}{6}, \frac{2L}{6}] \\ F(x) = t_{max} f^2, on[\frac{L}{6}, \frac{2L}{6}] \times [\frac{4L}{6}, \frac{5L}{6}] \\ F(x) = t_{max} f^2, on[\frac{4L}{6}, \frac{5L}{6}] \times [\frac{4L}{6}, \frac{5L}{6}] \\ F(x) = 0, else \end{cases}$$

Let $U_{ij}^n = u(x_i, y_j, t^n$. Our difference equation then becomes :

$$\frac{U_{ij}^n - U_{ij}^{n-1}}{\Delta t} = \frac{\lambda}{\rho c}(\frac{U_{i+1,j}^n - 2U_{ij}^n + U_{i-1,j}^n}{(\Delta x)^2} + \frac{U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n}{(\Delta y)^2}) + \frac{1}{\rho c} F(x_i, y_j, t^n)$$

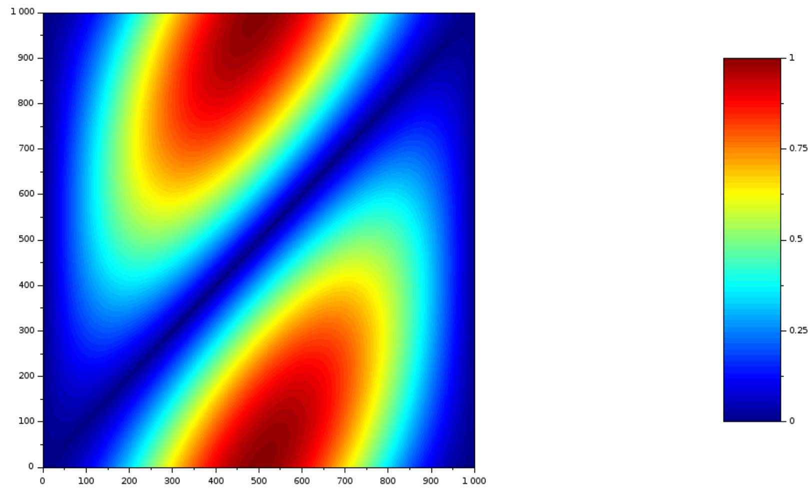
for i = 1,...,1001 and j = 1,...,1001
For simplicity of exposition, let's assume that $\Delta x = \Delta y$. Then simplifying the expression we arrive at :

$$-\alpha U_{i+1,j}^n + (1+4\alpha)U_{ij}^n - \alpha U_{i-1,j}^n - \alpha U_{i,j+1}^n - \alpha U_{i,j-1}^n = U_{ij}^{n-1} + \frac{\Delta t}{\rho c} F(x_i, y_j, t^n)$$

If we number the unknowns as U_{11}^n, U_{12}^n etc, then the linear system we must solve is a block tridiagonal matrix that we will solve like before in the first section (approach 1D).

3.2 Computing approach with C++ and Scilab

For this part we were able to build up a working program but unfortunately the results were not as expected at all and here is our figure result :



4 Conclusion

In the term of this project we have tried to combine the knowledge acquired during lectures and practical works in order to go through the relation between the theoretical and practical aspects. Honestly we were able to understand and complete the first approach but unfortunately we had less chance on the second so we will give you our code in order to have a look on it and give us some explanations and clues to fix our problem if it is possible.