Numerical 2D resolution of 4 conductive materials

Authors: Gerard Morales Riera
Gas dynamics and heat transfer, Carlos David Pérez Segarra

Aerospace Technologies Engineering degree, ESEIAAT

Abstract

Heat transfer problems are, in general, analytically solvable. However, interesting case studies come from complex geometries and challenging situations. In them, the numerical approach is a good enough tool to find a valid solution within a considerably competitive amount of computational time. This task will require of a 2D transitory state problem, which is, among the options to choose, a pretty expensive one to compute. Here, the resolution time will be worth a deep analysis.

1 Description

The case of this problem is a 2D plate which combines 4 different materials. The overall dimensions of these are determined by the user as the positioning of 3 points:

$$\boldsymbol{p_1} = (x_1, y_1) \quad \boldsymbol{p_2} = (x_2, y_2) \quad \boldsymbol{p_3} = (x_3, y_3) \tag{1}$$

These coordinates have two restrictions:

1. $\boldsymbol{p_3}$ determines width and length of the overall surface, so its coordinates must be the largest of all.
2. x_1 and x_2 shall be the same to correctly define the materials since, if it was not true, there would exist an intermediate zone where the material would be unexpected.

The present case looks like shown below.

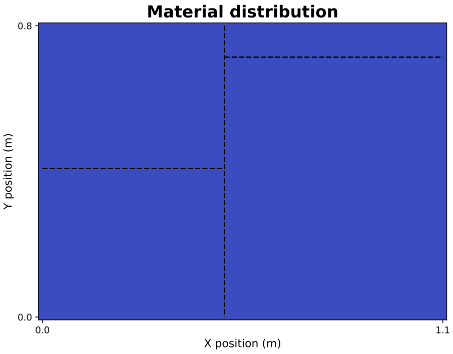

Figure 1. Current distribution.

To clarify, material 1 will be for the bottom left, 2 for bottom right, 3 for top left and 4 for top right. One defined the geometry, there will be a section dedicated to assign each node its material, which will include a set of properties: ρ, c_p and λ.

How are these nodes and what kind mesh is used?

Given the highly cartesian aspect of the problem, a structured mesh with rectangular elements was chosen. Each of the elements has a centered node. These elements are generally rectangular, so that the geometry does not compromise the precision in any direction. In addition, boundary elements have a centered nodes in the wall side.

This sums up a total of (N+2) × (M+2) nodes for the entire mesh (temperature matrix size). Notice that this total number claims the existence of corner nodes. These can not be physically well-defined, so the temperature solution in them will be given by qualitative terms.

What heat transfer methods does it include?

As stated in the suggested exercises, the boundary conditions are

- Bottom: isothermal wall.
- Top: uniform inlet heat flux.
- Left: convective wall.
- Right: time-dependent wall temperature.

and, since the inner heat flux is transferred by conduction, this problem includes both conduction and convection but not radiation.

1.1 Language used

Given this is the case of a 2D transient resolution, it is highly recommended to use a programming language capable to execute all the necessary commands in an acceptable amount of time. Given this is probably the biggest of all constraints, *C++* is chosen as the main solver language.

In addition, *Python* will be used to perform the post-processing features, like the material distribution form figure (1) and the output temperature distributions.

It is worth to mention that the overall compilation and execution will be done with *Makefile*. That is in order to achieve a certain knowledge on automatic code running. *Json* will be used for input parameters.

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <chrono>
#include <algorithm>
#include "json.hpp"

using namespace std;
using json = nlohmann::json;
```

Above is shown the first ten lines of the main code. With it, see that there is no inclusion of partial differential equations solvers nor linear algebra libraries.

The first 4 libraries are basic libraries to work with *C++*. The <*chrono*> library will be used to perform temporal duration metrics during the code execution. The <*algorithm*> one is for regular data management operations. And the included file *"json.hpp"* is a header created by a developer called *Niels Lohmann* which allows reading *Json* files for input data.

1.2 Resolution method

Generally speaking, the aim is to solve the fundamental equations of thermodynamics and heat transfer for each of the nodes simultaneously, and using a step by step method for time evolution. This will require a solver type to find a solution to each set of equations. Considering the *Tri-Diagonal Matrix Algorithm* (TDMA) is not available for 2D cases, the *Gauss-Seidel* (GS) is the chosen one.

Once found a solution to each of them, the next time step will be analysed. For it, say that the implicit integration method will be used, arbitrarily predefined and aiming for better convergence.

The equation to solve in each and every case is the first principle of thermodynamics,

$$U = Q - W \tag{2}$$

where the work done by each control volume (i.e. element) is null. This equation is simplified to:

$$U = Q \tag{3}$$

With this, may be analysed the left side first. In it, the variation of the internal energy in the control volume is stated. Its development follows as shown below.

$$U = \int_V \rho u \, dV = \rho_P \bar{u}_P V_P \tag{4}$$

Where the definition of $\bar{u}_P$,

$$\bar{u}_P = \frac{1}{\rho_P V_P} \int \rho u \, dV \tag{5}$$

has been used, alongside the premise of not having control volumes nor physical properties time dependency.

Now, using the expression

$$du = c_v \, dT \tag{6}$$

which in solids can be approximated to

$$du \approx c_p \, dT \tag{7}$$

one can find the control volume's internal energy in a discrete manner as:

$$U = \rho_P \bar{c}_{p,P} V_P (T_P^{n+1} - T_P^n) \tag{8}$$

Notice that the n and $n+1$ notation refers to the current and next time step, respectively.

Now, the transferred heat in a certain time interval is given by the integration

$$Q = \int_{t^n}^{t^{n+1}} \dot{Q} \, dt \tag{9}$$

where the superscripts of the integration limits have the same meaning that has just been explained (temporal references). This is turned into the expression

$$Q = \Delta t \left(\beta \sum \dot{Q}^{n+1} + (1 - \beta) \sum \dot{Q}^n \right) \tag{10}$$

where, as said before, the implicit method will be used ($\beta = 1$), and this equation results in:

$$Q_P = \Delta t \sum \dot{Q}_P^{n+1} \tag{11}$$

All these equivalences found imply a certain level of error given discrete behaviour, which was not stated as approximation. Precision will be treated in section (2). To summarize, the general equation for each and every node states as:

$$\boxed{\rho_P \bar{c}_{p,P} V_P (T_P^{n+1} - T_P^n) = \Delta t \sum \dot{Q}_P^{n+1}} \tag{12}$$

The left side includes general parameters and variables that need no longer definition. However, it is important to see how the volume is included. Yet, this problem does not contain any volume at all. It is necessary to remark that volumes and surfaces will be one dimension less in this case study. Hence, the surface of an element will be name volume, as well as its sides will be named surfaces. As long as the equations dimensions are equivalent, there is no issue to apply this.

Once it is clarified, see that the boundary nodes will not have a volume associated to them, since they are placed in walls. Hence, equation (12) turns into:

$$\boxed{0 = \sum \dot{Q}_P^{n+1}} \tag{13}$$

Of course, these equations will entangle with Fourier's law to let the final formulations be found. This law states that (in 1D):

$$\dot{Q}_x = \lambda \frac{\partial T}{\partial x} S \tag{14}$$

Known these two equations and one law, it is time to accurately find the expression that defines the behaviour of each node. Of course, it depends on the positioning of the node. Each kind of node will be defined under the following format:

$$\begin{aligned} a_P T_P^{n+1} &= a_E T_E^{n+1} + a_W T_W^{n+1} + a_N T_N^{n+1} \\ &+ a_S T_S^{n+1} + b_P \end{aligned} \tag{15}$$

All temperatures to solve belong to the next time step, since the resolution is done step by step and knowing the initial map of temperatures. Hence, T_P^n will be inside the b_P term.

Considering the indexation used is

$$i \in [1,\ \text{N+2}] \quad j \in [1,\ \text{M+2}] \tag{16}$$

let's find the equation that describes best each of the nodes

- **Top {i = 1; j = 2, ..., M+1}**: This wall includes a constant heat flux from the north face, as well as conduction from the south.

$$0 = \dot{q}_{wall} S_n + \lambda_H \frac{T_S^{n+1} - T_P^{n+1}}{\Delta y} S_s \tag{17}$$

  ⋆ $a_S = a_P = \lambda_H \frac{S_s}{\Delta y}$

  ⋆ $b_P = \dot{q}_{wall} S_n$

- **Left {i = 2, ..., N+1; j = 1}**: Convective wall which includes conduction too.

$$0 = \alpha_{ext}(T_{ext} - T_P^{n+1}) S_w + \lambda_H \frac{T_E^{n+1} - T_P^{n+1}}{\Delta x} S_e \tag{18}$$

  ⋆ $a_E = \lambda_H \frac{S_e}{\Delta x}$

  ⋆ $a_P = a_E + \alpha_{ext} S_w$

  ⋆ $b_P = \alpha_{ext} S_w T_{ext}$

- **Top left {i = 1; j = 1}**: Node without volume nor surface. The temperature will arbitrarily be the average of the two adjacent nodes. This does not affect the verification procedures.

$$T_P^{n+1} = \frac{T_E^{n+1} + T_S^{n+1}}{2} \tag{19}$$

  In this node, the constants are not necessary, since the calculation is direct.

- **Right {i = 1, ..., N+1; j = M+2}**: Wall the temperature of which varies only depending on time.

$$T_P^{n+1} = T_{right,0} + \Delta T_{right} t^{n+1} \tag{20}$$

  Here, constant are not necessary too. The value will be computed directly.

- **Bottom {i = N+2; j = 1, ..., M+2}**: Isothermal wall.

$$T_P^{n+1} = T_{bottom} \tag{21}$$

  Same as before, no constants required.

- **Bulk {i = 2, ..., N+1; j = 2, ..., M+1}**: Nodes with volume. These only undergo conduction.

$$\begin{aligned} \rho_P \bar{c}_{p,P} V_P \frac{(T_P^{n+1} - T_P^n)}{\Delta t} &= \lambda_H \frac{T_S^{n+1} - T_P^{n+1}}{\Delta y} S_s \\ &+ \lambda_H \frac{T_N^{n+1} - T_P^{n+1}}{\Delta y} S_n \\ &+ \lambda_H \frac{T_W^{n+1} - T_P^{n+1}}{\Delta x} S_w \\ &+ \lambda_H \frac{T_E^{n+1} - T_P^{n+1}}{\Delta x} S_e \end{aligned} \tag{22}$$

  ⋆ $a_E = \lambda_H \frac{S_e}{\Delta x}$

  ⋆ $a_W = \lambda_H \frac{S_w}{\Delta x}$

  ⋆ $a_N = \lambda_H \frac{S_n}{\Delta y}$

  ⋆ $a_S = \lambda_H \frac{S_s}{\Delta y}$

  ⋆ $a_P = a_E + a_W + a_N + a_S + \frac{\rho_P \bar{c}_{p,P} V_P}{\Delta t}$

  ⋆ $b_P = \frac{\rho_P \bar{c}_{p,P} V_P T_P^n}{\Delta t}$

Notice that internal heat generation was not considered in this case, as it was not suggested in the problem definition.

Also, an important feature not mentioned yet is how the heat transfer constant is treated. Since there are material changes within the body, an harmonic average of the constant is required in the general case. This is applied by using the function *harmonic_lambda()* for every constant used. This function replicates the well-known expression:

$$\lambda_H = \frac{d_{A,B}}{\frac{d_{a,wall}}{\lambda_A} + \frac{d_{b,wall}}{\lambda_B}} \tag{23}$$

Finally, it is necessary to explain a little how the *GS* solver works in this case. The general functioning of the method will not be treated here, since it is a common knowledge. However, it is worth saying that it is applied the use of a relaxation factor if necessary. It allows the code to accelerate more or less and favour convergence more or less.

The code looks for the maximum difference there is between the unsolved case (previous temperature field) and the solved case. With it, there is a convergence criterion which varies on the user's input and, of course, must be fulfilled.

2 Verification

The verification procedure is performed inside the code during the *solve_transitory()* function. This means that, for every time step solved, the overall energy balance is checked through the *energy_balance()* function.

In this process, the first principle of thermodynamics is applied to the full area, without considering control volumes. This ensures that the discretization performed and the results obtained are compatible within the physical constraints of the entire body.

The equation,

$$U = Q - W \quad \xrightarrow{\text{no work}} \quad U = Q \tag{24}$$

uses the internal energy accumulated (in every time step) of the full span of bulk nodes and the heat transferred from the wall nodes to them. This means using the previous temperature too, just like in equation (8) and using the full volume.

The wall uses Fourier's law from (14) to transmit the heat from them to the volume. However, analysing equations (17) and (18), one sees that these computations can be easily obtained by using the already known input parameters. This saves some calculation time.

After all, the statement

$$\frac{|U - Q|}{U} < 0.001\% \tag{25}$$

takes the decision on accepting the step or not.

3 Results

Now, along with the material properties and other parameters suggested in the problem definition, the input below is used to solve accurately the case of study.

```
{
    "Control Volumes": {
        "N": 100,
        "M": 100
    },
    "Time Parameters": {
        "dt": 1,
        "ti": 0.0,
        "tf": 5000.0
    },
    "Solver Parameters": {
        "delta": 1e-8,
        "T0": 281.0,
        "Relaxation Factor": 1,
        "Maximum Iterations": 1e6
    }
}
```

See that 100 × 100 mesh grid was chosen, alongside a unity relaxation factor. In addition, the temperature error is quite restrictive, which complicates convergence but makes the energy balance more secure.

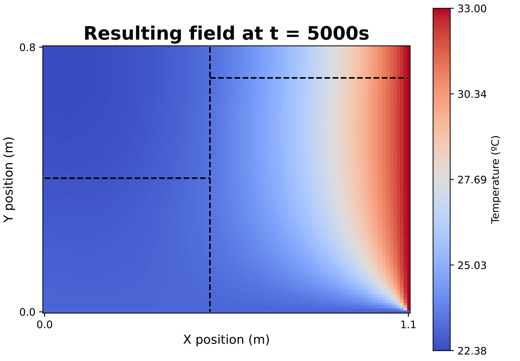

Figure 2. Temperature field.

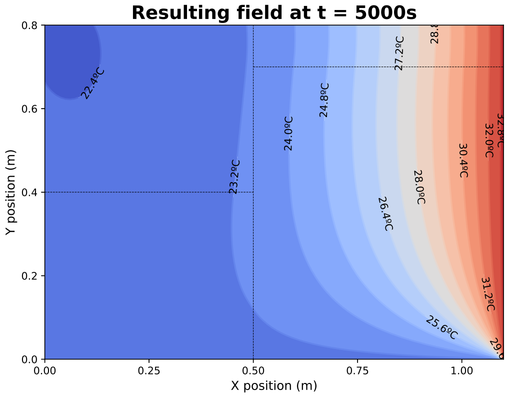

Figure 3. Isothermal curves.

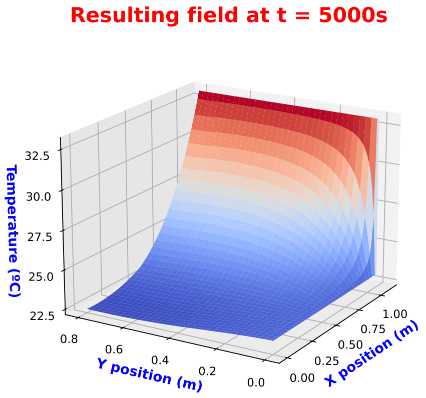

Figure 4. 3D field.

Above are shown, in three forms of representation, the results of this iteration. In it, all the time steps were correctly done, as of the result in the energy balance.

These results clearly satisfy basic requirements such as the isothermal bottom and right (time dependent) walls. Also, the curves shape are pretty similar to those on the shown results in the problem definition. By changing the labels used, one could for sure get exactly the same representation.

4 Numerical study

The numerical study will focus on three aspects of the simulation. Using a total emulated time of 500 seconds, the following three parameters and their resolution time analysed:

- **Number of elements**: using N = M meshes, different number of nodes will be used.
- **Relaxation factor (rf)**: over-relaxed and under-relaxed situations will take place.
- **Temperature's *GS* precision**: the variable δ will take different values (all of them will still be relatively low).

By using the code several times, it seems like, at some point of the simulation (around the 50th to 100th emulated second), the number of iterations used for the *GS* remains constant. This converged value of number of iterations will also be recorded and compared among other cases.

First of all, let's iterate over the number of elements. The other parameters used will be:

$$\text{rf} = 1 \quad \delta = 10^{-8} \tag{26}$$

See tabulated the values obtained, alongside their graphical representation to notice the tendency:

N, M	40	60	80	100	120	140
t (s)	4.1	11.4	28.6	53.7	112.6	174.8
Iter.	19	31	45	61	79	100

Table 1. Element study.

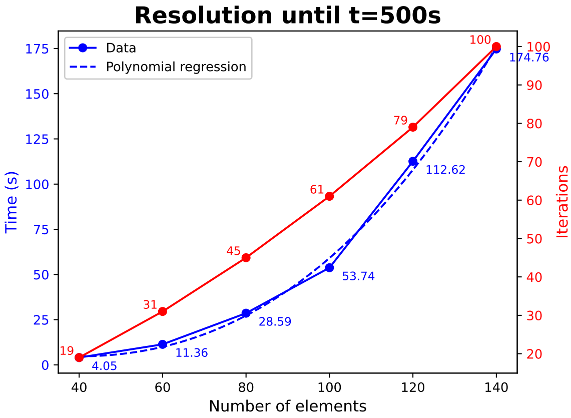

Figure 5. Element study regression.

As can be seen, the tendency is that the more divisions there are, the more time and iterations are required. Notice how the iterations follow a regression which is more than linear, probably between linear and quadratic. And, considering that the time required could be related to the operation needed to compute, it is logical that these are proportional to the square of elements (since it is a 2D case) times the number of iterations required. The emulated time would also be a factor, but here all of the samples have the same, so it does not affect regression.

Known this, one could state that

$$t(N) \propto N^2 \times iter(N^n) \propto N^{n+2} \tag{27}$$

where, as said before, n could belong to

$$n \in (1, 2] \tag{28}$$

and, by using $n \approx 1$, for instance, the regression in figure (5) is drawn. The resulting polynomial is:

$$t(N) \approx 0.000051N^3 + 0.0059N^2 - 0.71N + 21 \tag{29}$$

It is worth to mention that, of course, even though high amount of elements require more time, they lead to more precise representations of the body, since more nodes are calculated and more values of temperature are obtained. The elements definition represents the compromise between field precision and computation time.

Next, let's iterate over the relaxation factor. The other parameters:

$$\mathrm{N} = \mathrm{M} = 80 \quad \delta = 10^{-8} \tag{30}$$

See the results, displayed below.

rf	0.4	0.7	1	1.2	1.4	1.6
t (s)	79.5	44.1	29.3	23.3	22.8	-
Iter.	122	67	45	36	32	-

Table 2. Relaxation factor study.

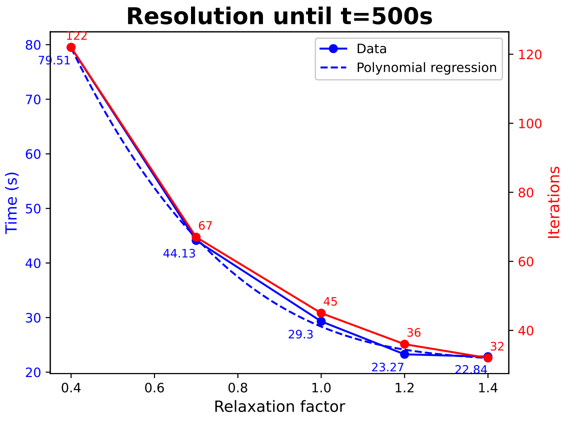

Figure 6. Relaxation factor study regression.

Here, the more the relaxation factor is increased, the less time and iterations are required. This is good, since the relaxation factor can be arbitrarily increased without limit. However, there is a convergence limitation to that. See how, at 1.6 of relaxation factor, not very far from the optimal value, the *GS* no longer converges, at least in a realistic amount of time. This means that the manipulation of the relaxation factor is highly powerful (notice it does not affect precision), but requires knowledge on its convergence limitations.

The time depending on the relaxation factor was surprisingly well fitted by a third degree polynomial, as seen in picture (6). Such regression is:

$$t(rf) \approx -46.7rf^3 + 202rf^2 - 294rf + 168 \tag{31}$$

Finally, it is the turn of δ, where these parameters have been used:

$$\mathrm{N} = \mathrm{M} = 80 \quad \mathrm{rf} = 1 \tag{32}$$

The results found are shown below.

δ	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
t (s)	16.3	19.7	24.3	28.6	31.4	40.2
Iter.	21	29	37	45	53	61

Table 3. Precision study.

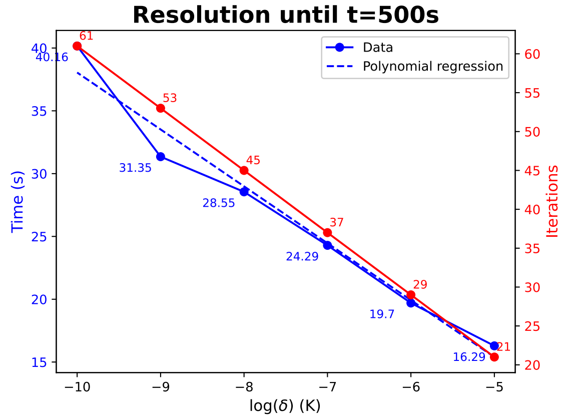

Figure 7. Precision study regression.

The precision follows a pattern that could be expected too. The more precision the user imposes, the more time and iterations are required. However, given the criterion to give one solution as valid, remember expression (25), there are some which do not fulfill such statement. Of course, these are the highest of the δ values, marked in red on table (3).

Since the number of elements are always the same, there must be a linear pattern that correlates the logarithmic δ with the time required. See how the iterations can be represented by a perfectly linear curve. Time regression was calculated for a linear one too, as seen in figure (7), and the result of which is:

$$t(log_{10}(\delta)) \approx -4.53log_{10}(\delta) - 7.25 \tag{33}$$

5 Physical study

The materials used in the resolution of this problem have characteristics of metallic elements. See how all four materials are comprised around the values of properties of

- $\rho_{M1-4} \approx 2000$ kg/m^3
- $c_{p,M1-4} \approx 800$ J/(kgK)
- $\lambda_{M1-4} \approx 160$ W/(mK)

and these belong to very dense and thermal conductive materials. For instance, see these same properties stated in aluminum are:

- $\rho_{Al} \approx 2700$ kg/m^3
- $c_{p,Al} \approx 900$ J/(kgK)
- $\lambda_{Al} \approx 220$ W/(mK)

Of all, the one to analyse in this study is specially the thermal conductivity. For instance, an experiment will be conducted, which will show the effect of different materials in the heat propagation.

The simulations will consist on 500 emulated seconds with 100 × 100 meshes and using a precision of

$$\delta = 10^{-7} \tag{34}$$

while maintaining a bit lower requirement than in (25), since extreme cases usually lead to slightly higher errors in a few of the time steps. The changing material will be material 3, since it has a big enough area in the whole rectangle and has both convective and conductive walls. The materials' area will not be changed, maintaining the points in (1).

Since this is a qualitative study and very different materials will be sampled, very precise values of material properties is not an important feature.

To begin with, see the case of the **wood** from an Oak tree. There are several types of woods, which can be classified in soft woods and hard woods. The first have lower density than the second and lower thermal conductivity too. The Oak's wood is a hard wood.

Having the properties

- $\rho_{wood} \approx 650$ kg/m^3
- $c_{p,wood} \approx 1500$ J/(kgK)
- $\lambda_{wood} \approx 0.15$ W/(mK)

the results from heat transfer in a transient state are shown below.

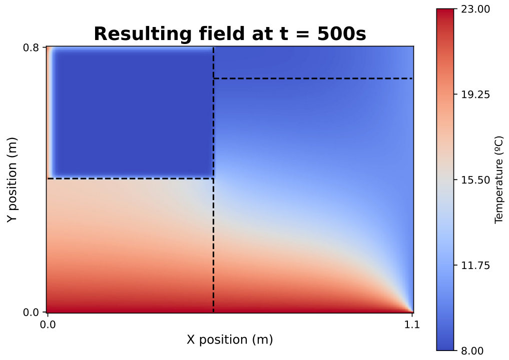

Figure 8. Temperature field.

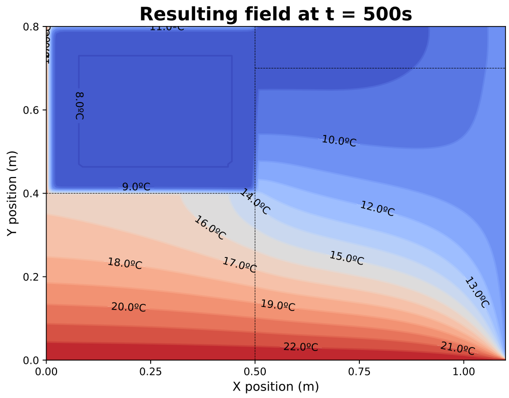

Figure 9. Isothermal curves.

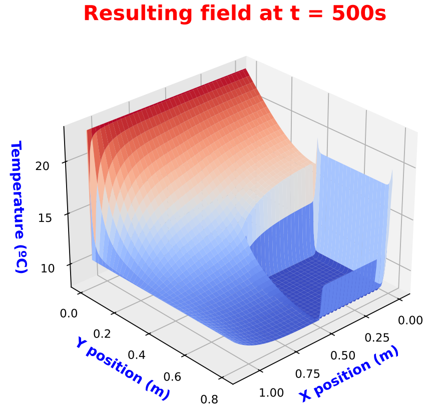

Figure 10. 3D field.

Notice how there is a considerable fall in temperate on the wooden section, but not that much on its sides. This is a natural behaviour of extremely low conduction in the inner part of the area.

Now, it is the time to study the behaviour of **hexagonal Boron Nitride (h-BN)** in the same environment. This choice is given the the nature of very high thermal conductivity it has in-plane, which is applicable to this case of study.

Its properties are:

- $\rho_{BN} \approx 2200$ kg/m^3
- $c_{p,BN} \approx 700$ J/(kgK)
- $\lambda_{BN} \approx 3000$ W/(mK)

Doing the simulation, it was noticed that 22/500 time steps did not comply with (25). However, these were still clearly below $\delta = 10^{-4}$ criterion. This was given because the high conductivity and difference in heat transfer leads to lower precision on the final computation. By using elements like graphene, which has a thermal conductivity in-plane one order of magnitude higher, it can be seen how these errors increase.

The results are displayed right below.

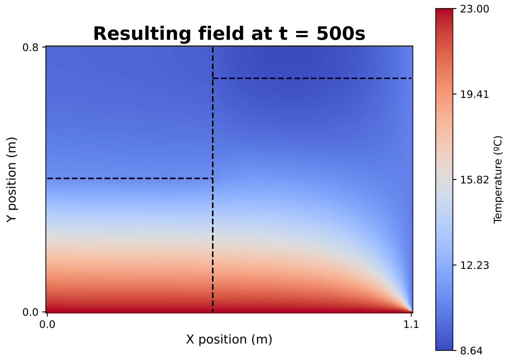

Figure 11. Temperature field.

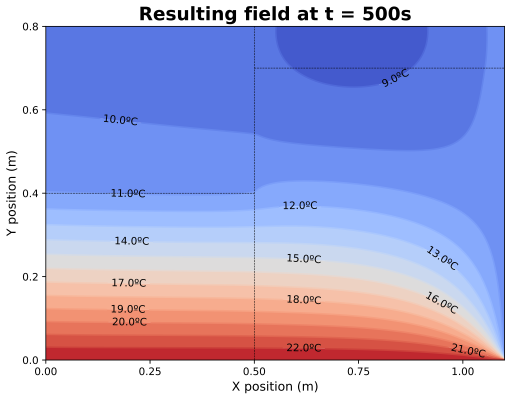

Figure 12. Isothermal curves.

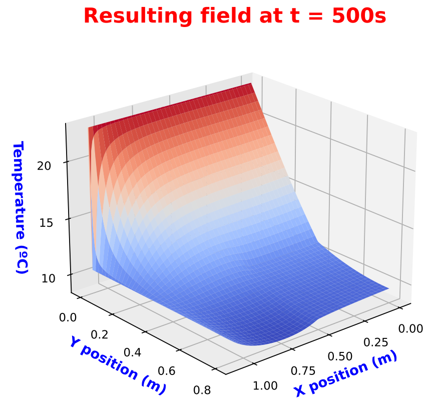

Figure 13. 3D field.

In this case, the behaviour is totally opposite to the wood one. Here, the temperature of the entire h-BN was almost the same. See how isothermal curves propagate in figure (12). They are clearly changing at a high pace in material one, but constant in h-BN. This is a behaviour directly correlated to the conductivity of the material.

Next, it is time to see the way **Uranium Dioxide** works. It is a very dense material, which will affect the internal energy side of the equation (12).

See its material properties:

- $\rho_{Ur} \approx 11000$ kg/m^3
- $c_{p,Ur} \approx 520$ J/(kgK)
- $\lambda_{Ur} \approx 2.5$ W/(mK)

Shall be remarked that this case has not presented any precision issues. All of the iterations have fulfilled the imposed criterion of this study and even the one used before for general purposes, stated in (25).

The results are displayed below.

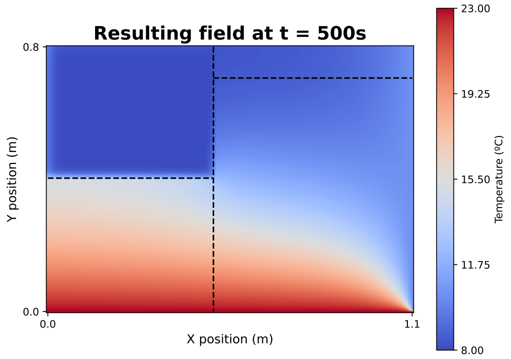

Figure 14. Temperature field.

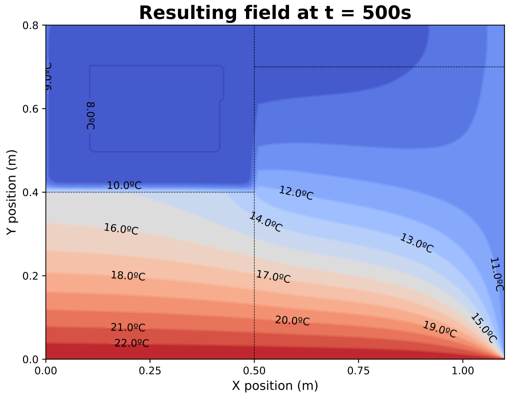

Figure 15. Isothermal curves.

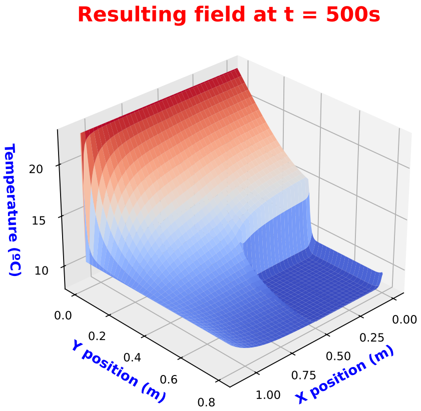

Figure 16. 3D field.

This temperature field looks a bit familiar. The behaviour is quite similar to the wood's. This is because the low thermal conductivity isolates the innermost region of the area. Also, since density is very high, one could look at equation (5) and understand why the walls do not have such a high value as it was shown in figure (10). The high density is responsible to accumulate a very low amount of specific internal energy, thereby leading to a low temperature gradient in the walls.

Finally, the opposite case to the latter can be analysed too. The plastic called **Expanded Polystyrene (EPS)** is known for having a ridiculously low density.

The properties used are:

- $\rho_{EPS} \approx 20$ kg/m^3
- $c_{p,EPS} \approx 1300$ J/(kgK)
- $\lambda_{EPS} \approx 0.04$ W/(mK)

Its results are shown below.

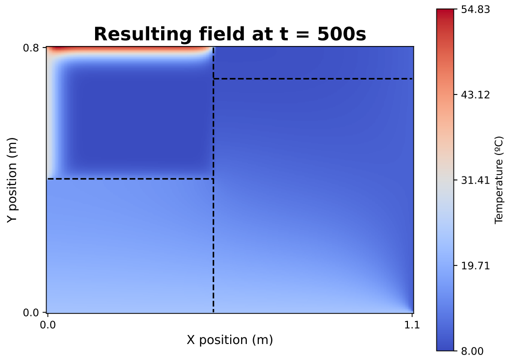

Figure 17. Temperature field.

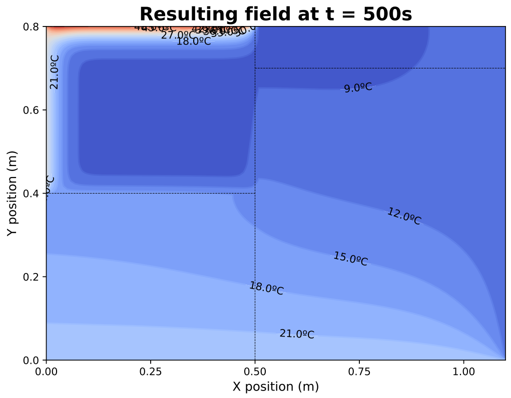

Figure 18. Isothermal curves.

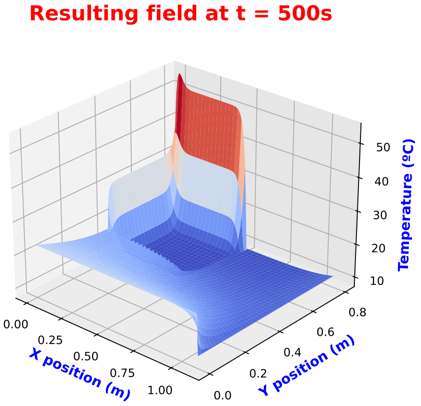

Figure 19. 3D field.

As can be perceived, the inner area is just like wood's and Uranium's. This is, as said before, given by the low conductivity. However, it is the boundary that clearly differs from the high-density case. Here, the sides of the EPS section comprise the highest temperature peaks of the field. This is because, by citing equation (5), the specific internal energy can be very high due to the low density of the material. This enhances heat absorption, yet it does not affect its transmission.

Speaking in numbers, the temperature at the boundary is reaching values of 50 degrees Celsius, which is twice the previous peak temperatures.

6 Conclusions

The conclusions are divided into three sections, one related to the general discussion of the problem resolution and the other two focused on the studies performed.

6.1 Resolution conclusions

Regarding the overall technical proceedings, they have to be computed inside the physical margins previously established. This, alongside a good enough precision in the mesh, the verification calculation should lead to ridiculously low errors. Once this is fulfilled, one knows the code works as expected.

Said this, the important features to ensure code reliability come across a good understanding of the equation every node follows, plus the interaction between nodes of different nature. Also, the running costs are very high, specially in transient cases of more than one dimension. This is a restrictive aspect that enhances good coding practices.

6.2 Numerical conclusions

These conclusions focus specially on the time required, since this is a very demanding problem in terms of computational time. Hence, one has to understand the the relation between mesh refinement and resolution time is extremely sensitive. This is a trade-off where the user has to find a suitable midpoint.

However, there are some behaviours which are not that much of a trade-off. It has been seen, for instance, how the relaxation factor and the temperature precision are key when asking low resolution times. Of course, they affect precision more or less. Yet, at the end, they are getting the expected results in highly different amounts of time. Hence, this is an aspect the user must consider when writing solver parameters. The only characteristic to consider here is the convergence. As well as these features are highly effective when being at their optimal, a lack of knowledge on their behaviour can lead to problems. These are no *GS* convergence with the relaxation factor and precision errors with δ's value.

6.3 Physical conclusions

Finally, the physical behaviour of different materials has already explained in its section. From it, the key ideas to extract are that heat transfer processes can highly differ on the material used. Extreme cases have been presented, but some of them are materials that can be easily obtained. Hence, this shows that one can be capable to create diverse heat transfer systems within a reasonable spectre of materials available.

Some of the materials will be suitable for thermal isolation and some other will be extremely good conductors. And, given a study like the one just done, it is not necessary to create a resolution code straightaway to know which element is better than the other at some task. Observing the general equations that rule the case and understanding the properties that are involved in them, one can guess accurately lots of aspects about how a material will behave.