UNIVERSITÀ DEGLI STUDI DI PADOVA

UNIVERSITY OF PADOVA

Civil, Environmental and Architectural department DICEA

Master's degree in Mathematical Engineering

Numerical Methods project

Thermal conduction analysis in a fin with FEM and EbFVM

Andrea Gorgi, Badge 2010658

Academic year 2020/2021

List of Figures

Contents

1 Introduction

1.1 Objective of the project

The purpose of this paper is to solve the heat conduction problem on a fin with proper initial and boundary condition, and specific environment characteristics. Since the length of the bar taken in exam is more than one order of magnitude bigger than the other dimensions, the analysis will be reduced to mono dimensional, assuming that the temperature it's constant in every cross area (no variation of temperature in the "y" and "z" axes). Although this may seem a big approximation to the real problem at first, it could give a quite valuable solution saving from lots of computations required for the more complex 2 or 3 dimensions study. It's important to stress that even though this is a simple problem, its applications are of fundamental importance for the study of almost all the heat conduction problems, because it's at the base of the analysis of more complex geometries and physics cases.

The secondary aim of this project is to test the differences between the *Element based - Finite Volume Method* (EbFVM) and the *Finite element method* (FVM) as numerical solver for a physical problem like the one in analysis, looking deeply to their construction, and comparing the final results. In order to achieve this objectives the project is organized in a systematic way. Firstly will be given the specific datum of the problem, and will be analyzed the physics, secondly will be proposed a way to elaborate the mathematical model to be solved, and than will be implemented the EbFVM and FEM, using the Matlab software ([2]), and got a numerical solution for the initial problem. Finally, those results will be checked with proper analytical solutions and numerical validation, and compared to establish strengths and weaknesses of each method.

1.2 Legends of the terms used in numerical algorithms

In this chapter will be given a brief description of all the terms used in the computation of the two numerical algorithms presented above.

T^O: "old" time level Temperature, stands for the solution computed on the previous iteration, that is now known.

T: "new" time level Temperature, without quotes, stands for the solution that has to be computed in the present iteration, is the unknown.

T_P: Temperature at a generic point P.

T_E: Temperature at the east-neighbour point of P.

T_W: Temperature at the west-neighbour point of P.

T^θ: Temperature in a chosen time level θ between the "old" and the "new" one.

α: $\alpha = \frac{k}{\rho C_P}$, thermal diffusivity coefficient;

Δx: Length of the Control Volume (in EbFVM) and length of the Element (in FEM).

Δx_e: Distance between point P and his east neighbour, E.

Δx_w: Distance between point P and his east neighbour, W.

$q|_e$: flux entering by the east surface of the Control Volume. Is valuated in the point in the middle between point P and E.

$q|_w$: flux entering by the west surface of the Control Volume. Is valuated in the point in the middle between point P and W.

Δt: Step size of the iteration. T is valuated at a time $t + \Delta t$ with respect to T^O.

$N_A; N_B$: Shape functions, used to weight inside the elements the Temperature of theirs extremes.

$\psi_1; \psi_2$: Trial functions of the FEM, used to interpolate the behaviour of the solution inside the elements. For the Galerkin method their equal to the shape functions.

ξ: Coordinate of the local system for each element. $\xi = 0$ corresponds to the first extreme and $\xi = 1$ to the second one.

2 Physics & Mathematical Model

2.1 Description of the problem

The problem taken in consideration is represented by a fin with certain heat generation, q', attached to a wall at left extreme and insulated on the other one, lying in an environment with a certain *global* temperature, T_∞, and convection coefficient, h. The fin is initially supposed to be in thermal equilibrium, so that the heat produced inside the bar is equal to the heat lost to the environment by convection trough the lateral surface. At $t = 0s$ the face attached to the wall is raised up and maintained for time domain, $t > 0$, to the temperature T_0. The aim is to find the evolution of temperature in time along the fin, up to the stationary condition, in which the heat entering from the surface maintained to T_0 plus the heat generated inside the body are equal to the total heat exchanged by convection on the body's surface. A sketch of the geometries and the problem is shown in figure (1). The numerical values of the data

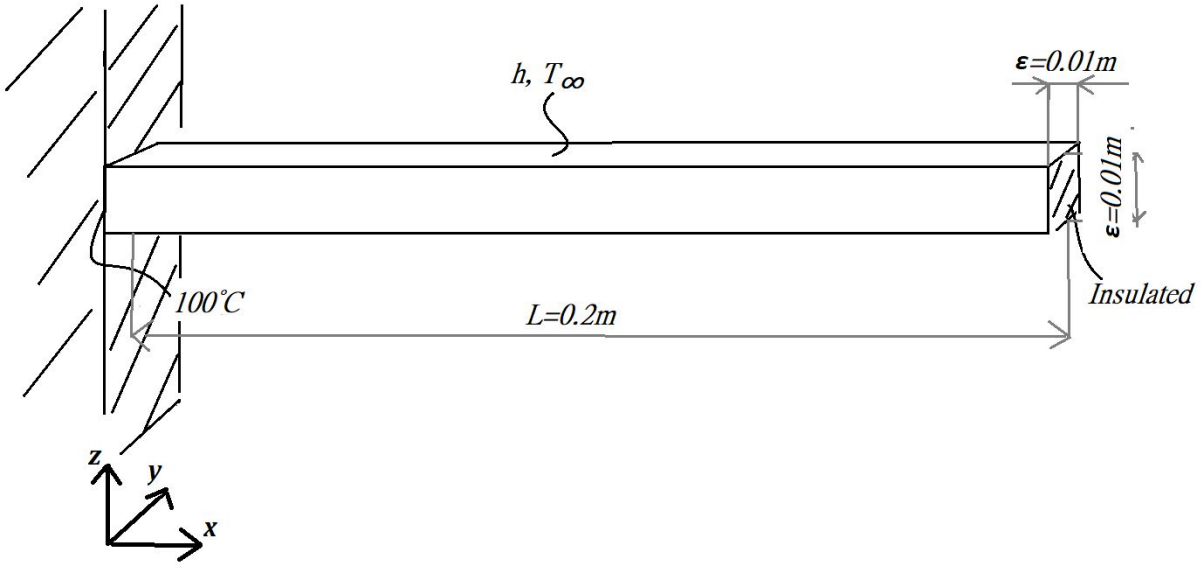

Fig.(1) Sketch of the problem

are the following: $L = 0.2$m, length of the bar; $\epsilon = 0.1$m, base and height of the crossing area; $k = 30$ W/mK, thermal conductivity of the bar; $h = 20$ W$/m^2K$ convection coefficient of the environment; $T_0 = 100$°C, temperature of left extreme; $T_\infty = 20$°C temperature of the environment, $q' = 10^4 W/m^3$, heat generation; $\rho = 8700 kg/m^3$, mass density of the bar; $C_P = 0.42 kJ/kgK$, specific heat of the bar at constant pressure. Before starting to analyzing the problem from a mathematical point of view, it's useful try to understand from a physical point what should happen to the bar. Firstly, notice that the fin starts from a thermal stability condition, that means, as already written above, that the heat produced inside by heat generation must be dissipated completely by convection, because, assuming initial temperature constant and equal to the structure attached at left extreme, there are no other fluxes, as shown in figure (2). Writing explicitly the balance, we get

$$\begin{aligned} Q_{IN} + Q_P = Q_{OUT} \implies q'V = hA_L(T^* - T_\infty) \iff q'\epsilon^2 L = 4h\epsilon L(T^* - T_\infty) \\ \iff T^* = \frac{q'\epsilon}{4h} + T_\infty \implies T^* = 21.25C \end{aligned} \tag{1}$$

where T^* has the meaning of initial temperature, constant along all the fin, V of the bar's volume (crossing area × length) and A_L of lateral surface of the fin (perimeter × length). Q_{IN} and Q_{OUT} here stand for heat entering (that is 0 in this case), and leaving the body, whereas Q_P stands for the heat generated inside the volume. Clearly, from a physical point of view, having the left head of the fin raised to a constant

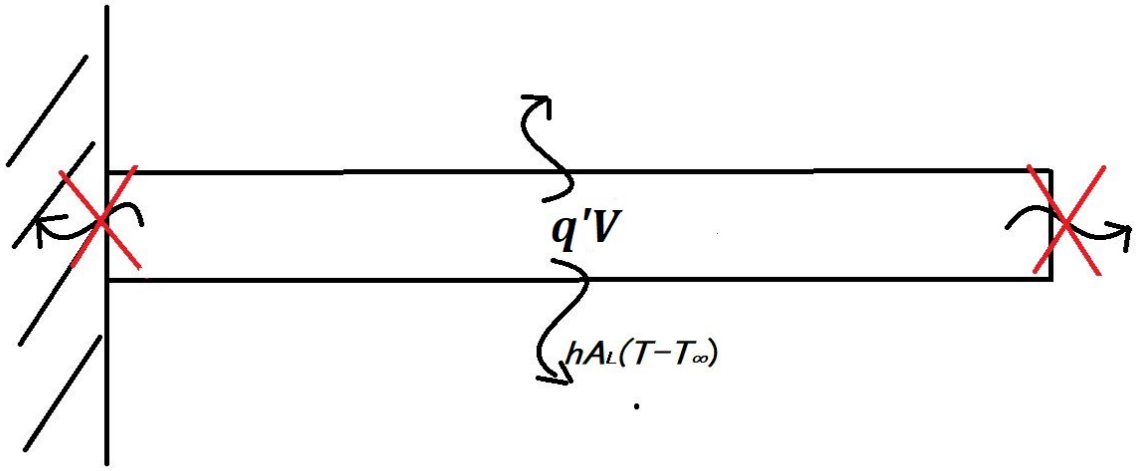

Fig.(2) Initial situation

temperature T_0 greater than the previous equilibrium temperature T^*, we should expect an heat flux passing trough that face and entering additional energy to the system. Therefore the temperature in the fin will tend to increase up to another stability condition, in which the energy (heat) entering trough the left extreme and the heat generated inside of the body will be completely dissipated by convection. This reasoning is what will lead in chapter (4.2) to an analytical solution for the steady state. A banal consequence of having heat entering at left and an insulated face at right, is to have, in every instant of the problem evolution, a decreasing temperature along the "x" positive direction, from the wall attaching point to the other extreme (every point has a bigger temperature than all the remaining right points, and fewer than the one of all the points to his left). At the insulted face, since by definition there cannot be a flux passing trough the face, by Fourier's law

$$q = -k\frac{\partial T}{\partial x} = 0 \implies \frac{\partial T}{\partial x} = 0 \tag{2}$$

that means that the temperature will have the tendency to decrease down to a constant value moving to the right of the bar.

2.2 Mathematical model

To obtain the mathematical model of the problem we consider a generic control volume inside the domain of the bar ($x \in (0, L)$) and analyze the balance of energy. Therefore we compute, looking at figure(3)

$$\begin{aligned} Q_{IN} - Q_{OUT} + Q_P &= \frac{D}{Dt}E \\ q_x A - q_{x+dx}A - hPdx(T - T_\infty) + q'Adx &= \frac{D}{Dt}\int_V \rho C_P T dv \end{aligned} \tag{3}$$

where E stands for total energy inside the control volume and P is the perimeter of the crossing area. Now, assuming T equal to the integral average on the control volume and constant in the crossing area, ρ and C_P constants on the domain of integration and dividing for dx we get

$$A(q_x - q_{x+dx}) - hP(T - T_\infty) + q'A = A\frac{D(\rho C_P T)}{Dt}. \tag{4}$$

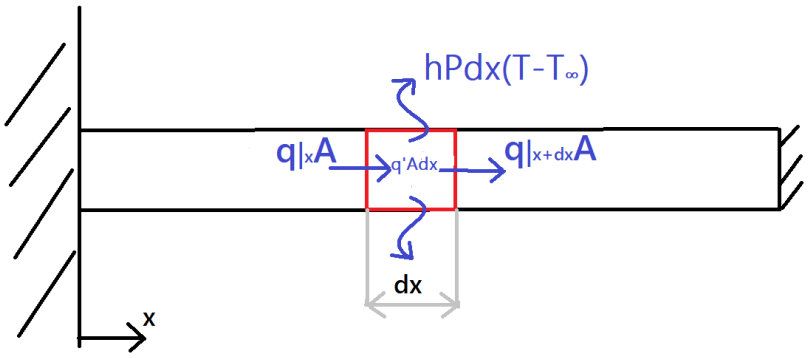

Fig.(3) Balance in a generic control volume inside the bar

Taking the limit $dx \to 0$ the CV used for equation (4) tends to coincide with a single crossing area at distance x from the left extreme of the bar, thus T becomes the real value of the temperature on that surface and we get

$$-A\frac{\partial q}{\partial x} - hP(T - T_\infty) + q'A = A\frac{\partial(\rho C_P T)}{\partial t} \tag{5}$$

where the partial derivative in time comes from the fact that the coordinates of the bar doesn't change in time, so total and partial derivatives are the same. This equation (5) has validity in all the internal points of the domain, because taking the limit we released all the assumption made on T in the discrete case balance. Applying Fourier's law ($q = -k\frac{\partial T}{\partial x}$) on the heat term we finally obtain

$$A\frac{\partial(k\partial T)}{\partial x^2} - hP(T - T_\infty) + q'A = A\frac{\partial(\rho C_P \partial T)}{\partial t} \quad \text{for } x \in (0, L),\ t > 0. \tag{6}$$

This equation is nothing but a balance of energy at every instant of time, in a control volume restricted to be a single face, without thickness (in 3-dim the control volume would be restricted down to a point, whereas in 2-dim to a line), and each term has a specific meaning. The first term has the physical meaning of diffusion inside the fin, and its the one that "transport" the effects of the heat flux entering at left face to the whole body, up to the right extreme, forcing the temperature of every point to be sensitive about the neighbour points temperature. Without diffusion the bar would remain at the same initial temperature along all time values, and we will show that is exactly what would happen with a conductivity coefficient $k = 0$. The second term in 6 is a boundary condition term, derived from the approximation of the problem from 3-dimensional to 1-dimensional, and represents the flux leaving every moment the bar by convection along the perimeter of the crossing area. The third term it's the energy generated inside the face in an infinitesimal time dt. Therefore the left hand side it's the difference between energy entering and leaving the body in every instant of time, taking care also of the generation inside the face. It's clear how this term should be equal to the energy variation in every infinitesimal step of time dt, and that's exactly what it's written on the right hand side of 6. Note that as would be expected from a physical point of view, the solution searched has to be in $\mathcal{C}^2(0, L) \cap \mathcal{C}^1 \times (0, +\infty)$, that means that is surely continuous in both space and time domain. The mathematical model for the whole problem is than the system

$$\left.\begin{aligned} A\frac{\partial(k\partial T)}{\partial x^2} - hP(T(x,t) - T_\infty) + q'A = A\frac{(\rho C_P \partial T(x,t))}{\partial t} \quad & \text{for } x \in (0, L),\ t > 0. \\ T(0,t) = T_0;\ \frac{\partial T(L,0)}{\partial x} = 0 \quad & t > 0 \\ T(x,0) = T^* \quad & x \in (0, L), \end{aligned}\right\} \tag{7}$$

where the second line shows the boundary condition for the solution T(x,t), and the third line is the initial condition for the problem. Those conditions are generally referred as *Parabolic boundary*, since there is no upper condition in time dimension. This last concept, to have condition at both extremes for space, but only for one extreme on time, due to the fact that in the equation appear a second term derivative in space, but only one in time, it's what allows to research a numerical solution *"marching in time"*, generating a solution for the whole space domain at one time level, and then moving to the successive time level, up to the desired time (that in this case is the steady state condition). This characteristic can be summarized saying that this equation is elliptic in space (second time derivatives and two boundary condition for each space dimension), and parabolic in time (only first derivative term and one boundary equation for the beginning of the problem). Physically this means that the any effects would propagate in each direction in space (both left and right), but only upper in time, roughly "what happen in the future doesn't have any effect on the present".

2.3 Scale Analysis

A very powerful tool, useful to have an idea of the order the order of magnitude of the terms in the PDE along the domain, is the *Scale Analysis*. This tool it's nothing more than a reasoning about what's in the equation to solve, comparing the two biggest magnitude terms in it. The first step is to define clearly the region of analysis, in such a way that we can have an idea of what are the external condition for the problem. Then the crucial step is to think at the equation as an equivalence among two dominant terms, simplifying all the others by following proper rules ([1]). In our problem it could be interesting to find at what time, t^*,the heat entering at left surface will arrive to influence the right extreme of the bar. To do this we then consider the domain (0,L)$\times$ (0,t^*), and, by (6) we get

$$A\frac{\rho C_P \Delta T}{t^* - 0} \sim G, \tag{8}$$

where ΔT it's the order of magnitude of the variation of temperature in the domain chosen (equal to $T_{max} - T_{min}$ in the domain, that is $T_0 - T_\infty$) and G is the biggest magnitude term of the LHS in (6). Doing some small calculations, it result that the biggest contribute is represented by the convective term. Then, by trivial computations we obtain

$$t^* \sim \frac{A\rho C_P}{hP} \sim 0.46s, \tag{9}$$

where we have simplified the ΔT terms. We will prove in chapter 5 how good this first approximation is, considering it's absolutely simple computation.

3 Numerical solution

3.1 Introduction to numerical approximations

All the numerical methods presented in this paper are different types of the same general approach for the numerical approximation, the *Weighted Residual Approach* (WRA). Having methods to approximate analytical solution of the problem it's mandatory, because compute an exact solution is usually very difficult or impossible to achieve. The main idea to get an approximate solution of a problem is to find a way to pass from the physical domain, that is continuous, to a discretized one in which the solution has only be found on a finite number of points, one for each sub-volume. In this way we can form a linear

[1] wikipedia

system of equations, each one representing one sub-domain, from which we can obtain the values of the unknowns on every piece, that is the approximate solution. Note that for the real domain this would correspond to a linear system of infinite equations and infinite unknowns (as the point of a physical domain). Calling $\mathcal{L}(\phi)$ the operator of the continuous PDE, obtained moving all the right hand side terms to the left hand side, such that the equation becomes $\mathcal{L}(\phi) = 0$ (ϕ continuous solution of PDE), and the operator of the discretized PDE as $L_N(\phi)$, we would have

$$\mathcal{L}(\phi) = L_N(\phi) + TE, \tag{10}$$

where TE represents the truncation error, due to the approximation made. A numerical algorithm is said to be *convergent* if is both consistent and stable. Consistency means that refining the grid down to points, the approximate solution should tend to be equal the the real one, having the truncation error going to zero as the the dimension of the control volume goes to zero, whereas stability means that the numerical solution is the exact solution of the linear system developed for the method.

3.2 Weighted Residual Approach (WRA)

As already said, every numerical scheme is based on a discrete operator derived from an approximation of the continuous one, and since the original PDE is in the form $\mathcal{L}(\phi) = 0$, we would like the approximate operator with the approximate solution $\tilde{\phi}$ to be such that $\mathcal{L}(\tilde{\phi}) = R \simeq 0$, where R is called *Residue*. One way to construct a method that tries to minimize the residue on the average is the so called *Weighted Residual Approach* (WRA), that impose the following condition

$$\int_\Omega R w d\Omega = 0, \tag{11}$$

where Ω is the whole domain of the PDE (space $\times$ time) and w is a *weighting function*. The choice of the weighting function decides how the method is going to approximate the solution, and for this reason different choices have different names as they are different methods, even if derived all from this basic idea of WRA. Here we referred to w as a set of weighting functions w_i, since there is one equation for each one of them, and so there are as many w_i as the unknowns of the discretized domain For $w_i = \delta_{ij}$ we would have the *Collocation Method*, that interpolates exactly the value of the solution on chosen points of the domain, by using the Dirac delta in the integration; for $w_i = \psi_i$, where ψ_i are the trial functions for the solution, we would have the *Galerkin method*. Like these, there are lots of other possible weighting functions, and one in particular is what generates the first method treated in next paragraph, that is for $w_i = 1$ on the i_{th} sub domain and $w = 0$ on all the others, we have the *Sub-domain method*, also called *Finite Volume Method*.

3.3 Finite Volume Method

The Finite Volume Method (FVM), as written above, is basically a WRA procedure, with weighting function equal to 1 in the sub-domain considered for each equation. This choice of w allows the algorithm to preserve the balance of the quantity in exam in every sub-domain, whatever the size of the grid is, as long as the method is obtained by an initial PDE written in a proper way. This could easily be seen by writing explicitly the method

$$\int_\Omega R w_i d\Omega = \int_{CV_i} R dv,$$

where the CV_i stands for a sub-domain on both space and time dimensions. What we have on the left hand side is basically the integration on the whole domain (that is the Cartesian product of space and time domain), of the PDE times the weighting function, but since w_i is 1 only on the $i - th$ Control Volume, what remains is the integration of the original PDE only on it. So, if the equation is a balance

law, and this represents almost all the practical case of study, the method is nothing but the integration of the continuous balance law of the quantity in exam of the CV. Moreover, if the equation is written in a conservative law, that is the analogous to a divergence form, without splitting the derivatives term and maintaining the structure of the balance, the FVM reduces to basically a discrete balance, as the one used in (3) to derive the mathematical model for the problem. This discrete balance, as we will see, allows to approximate the solution at the center of the CV, and gives particular stability to the algorithm (in the sense of bounding any possible fluctuations or rounding error in initial data), since it tries to represent, should be true from a physical point of view for any realistic solution. For example think at the problem of mass conservation for a fluid flow. If the balance of mass were not respected in the solution, the result wouldn't have any sense, because mass cannot be created or destroyed, therefore it's logical that the numerical algorithm is less stable in the case of balance not preserved, and even if it's consistent, it would probably take considerably more time to converge. Back to the problem described in chapter (2.1), in

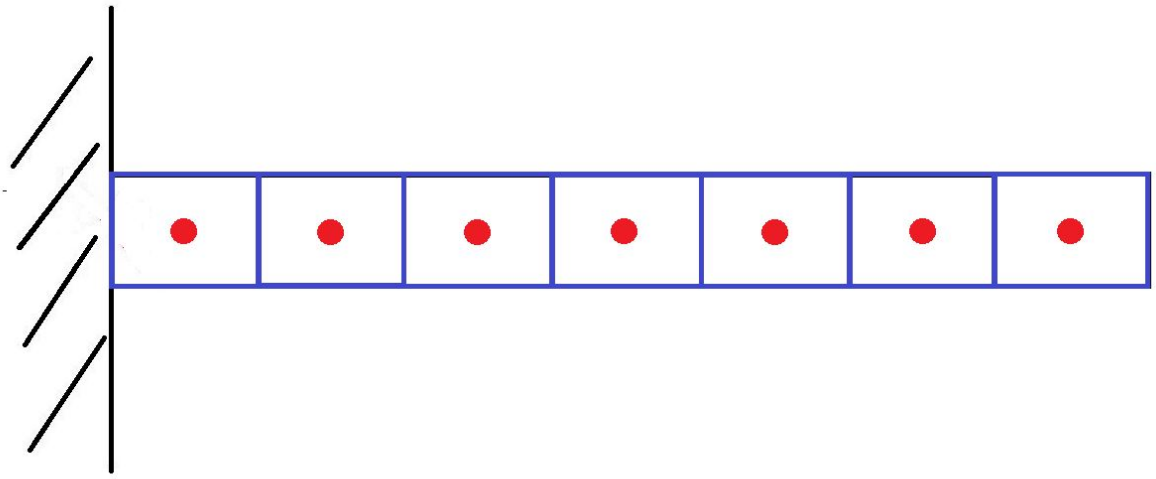

Fig.(4) 2D representation of a 1D Cell-centered method with uniform grid, Control Volumes corresponds to the elements. The red points represent the center of the CVs, where are computed the unknowns

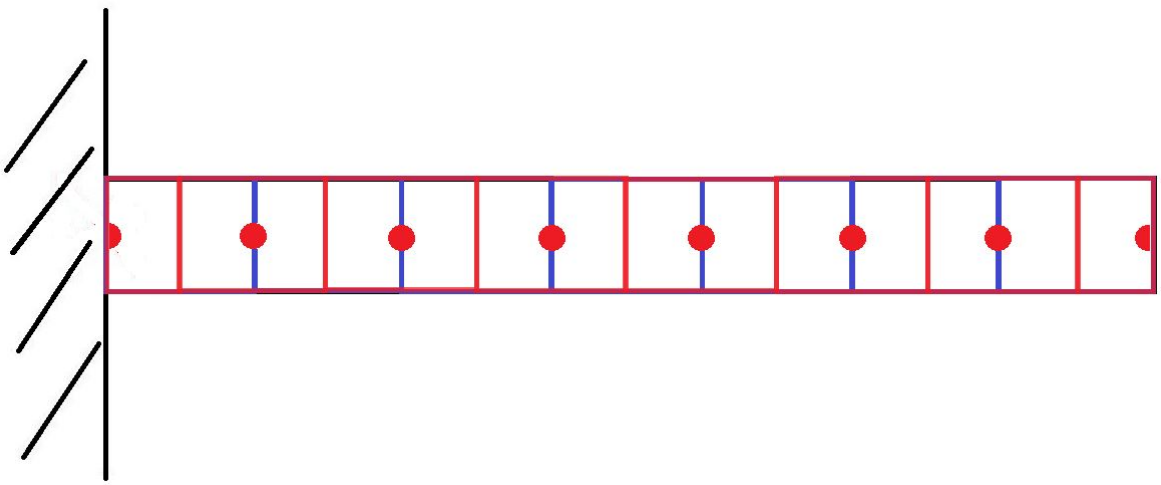

Fig.(5) 2D representation of a 1d Element Based method with a uniform grid, Control Volumes (in red) are built with their centers (red points) in the extremes of the elements (in blue). Every element has parts of different CVs

order to apply this method we first have to divide the physical domain in pieces, called sub-domains, or rather Control Volumes, in whose centers are computed the unknowns. For all modern procedure are commonly used grid generators to split the original domain of the system in convenient pieces, handly for the computations. We referred to this pieces with the name of *elements*. Nonetheless, it's not mandatory to choose this elements as the Control Volumes that are needed for the method. There exists basically two main possibility for the choice of the CVs. The first possibility is to have as CVs exactly the elements created by the grid generator, that leads to the localization of the unknown in the center of the elements, which gives it the name of Cell Centered -FVM (see figure (4)). Of course, since we are treating a 1D problem, the values of the unknown will be the same on the whole crossing area at that "x-coordinate" value. The other main possible choice is to localize the center of the CVs, where the approximate solution is computed, on the elements extremes, or vertex, and for this reason the algorithm is in this case called the *Cell Vertex-FVM* or Element Based-FVM (EbFVM), since here the domain is sweeped by elements, not by the CVs, and the CVs are created by pieces of them. As we will see, this gives a great elasticity to the computation of the solution, and for this reason will be the only one treated in this paper.

3.4 Element based-Finite Volume Method

We now start to compute the Eb-FVM numerical scheme for the problem proposed. Since the equation (6) is already written in a conservative form, the first step is to apply the WRA with $w_i = 1_{CV}$ (1_{CV} here stands for 1 on the i_{th} CV and 0 on all the others). Recalling that the whole domain here is $\Omega = (0, L) \times (0, +\infty)$, assuming the CVs be defined as $(x, x + \Delta x) \times (t, t + \Delta t)$, where x and x+$\Delta x$ are the space coordinate of the two heads of the CV and $t, t + \Delta t$ is the time step taken into consideration (note that time and space steps are discrete now), we get

$$\begin{aligned} \int_\Omega [A\frac{\partial(k\partial T)}{\partial x^2} - hP(T - T_\infty) + q'A]wd\Omega = \int_\Omega A\frac{\partial(\rho C_P T)}{\partial t}wd\Omega \\ \int_{CV_i} [A\frac{\partial(k\partial T)}{\partial x^2} - hP(T - T_\infty) + q'A]dv = \int_\Omega A\frac{\partial(\rho C_P T)}{\partial t}dv \\ \int_t^{t+\Delta t}\int_x^{x+\Delta x} [A\frac{\partial(k\partial T)}{\partial x^2} - hP(T - T_\infty) + q'A]dxdt = \int_t^{t+\Delta t}\int_x^{x+\Delta x} A\frac{\partial(\rho C_P T)}{\partial t}dxdt \end{aligned} \tag{12}$$

Now it's clear how the conservative form of the PDE plays a fundamental role, allowing to simplify the derivative terms without approximations on the first integration, whereas if the derivative term were "broken", splitting the products inside the derivatives, this result couldn't be achieved. In this particular case, since all the PDE parameters inside the equation are constant along the whole domain, the resulting method would be the same also with the derivative terms broken, but it's important to stress that is not true in general. Before proceeding in the computations, it's worthy to remember the analysis made on the PDE at the end of chapter (2.2), that the solution has to be searched "marching in time", estimating the results for the whole space domain every time step before moving to the successive one. It's recommended to look at chapter (1.2) to understand the meaning of the quotes and subscripts used. Therefore from (12), using the constancy of A,ρ, k and C_P along the space $\times$ time domain, and noting that since the solution is searched to be continuous and the domain of integration is bounded, so the order of integration can be exchanged as said by Fubini-Tonelli theorem, we get

$$\begin{aligned} \int_t^{t+\Delta t}[kA(\frac{\partial T}{\partial x}|_e - \frac{\partial T}{\partial x}|_w)]\Delta t - \int_t^{t+\Delta t}\int_x^{x+\Delta x} hPTdxdt + \int_t^{t+\Delta t}\int_x^{x+\delta x}(hPT_\infty + q'A)dxdt = \\ = \int_x^{x+\Delta x}\rho AC_P(T - T^O)dx. \end{aligned} \tag{13}$$

What it's remarkable is that this operation is done without any approximation for the FVM, and all the approximations due to the domain discretization are going to be introduced from now on. To simplify the integration in time we choose arbitrarily a time θ in $(t, t + \Delta t)$, in which we assume the quantities computed in such value to be the integral mean of them as functions of the time. We are than introducing a first approximation to the solution, that obviously goes to 0 as the CVs are refined to points. What we get is

$$\begin{aligned} [kA(\frac{\partial T^\theta}{\partial x}|_e - \frac{\partial T^\theta}{\partial x}|_w)]\Delta t - \int_x^{x+dx} hPT^\theta\Delta tdx + \int_x^{x+dx}(hPT_\infty + q'A)\Delta tdx = \\ = \int_x^{x+dx}\rho AC_P(T - T^O)dx \\ [A(q^\theta|_w - q^\theta|_e)]\Delta t - \int_x^{x+dx} hPT^\theta\Delta tdx + \int_x^{x+dx}(hPT_\infty + q'A)\Delta tdx = \\ = \int_x^{x+dx}\rho AC_P(T - T^O)dx \end{aligned} \tag{14}$$

where all the parameters doesn't shows the θ quote since they are constant along all time values for this problem. Note how the first term of the l.h.s shows the two fluxes on the space boundary in "x" coordinate for the problem, by Fourier's law. To proceed, it's important remark that in general, for every FVM the unknowns between two centers of CVs are computed by interpolation of the nearest known values for each

time step. Depending on how many values are used the interpolation will be of higher or lower degree, and for this reason there exists multiple choices, depending on the order of the approximation desired. In this way the temperature between two centers would be in the form $T(x) = \sum_i N_i(x)T_i$, where T_i is the temperature at the center of the $i-th$ control volume of the grid, whereas N_i is the weighting function, also called *shape function* of that temperature along the domain considered. Clearly the order of approximation increases as the number of centers involved increase, but since the main idea for the FVM is to derive a linear system of the form $[A][T] = [B]$, with T the vector of the centers temperatures, is not convenient to make the temperature of every CV depending on lots of others, because this would fill lots of positions in the matrix A, slowing consistently the computations. The most common procedure is than the linear approximation, which means that between two known values the solution has a linear behaviour, and depends only on the two nearest known points, but is also very popular the quadratic one, which involves a couple more of neighbour centers. Note that to use linear approximation doesn't mean to solve the PDE in a rougher way than with higher order approx. as long as the grid is refined to a sufficient small scale. The use of linear approximations in general allows to have less computations and easier and more elegant code implementation (tridiagonal matrix of the linear system), at the cost of having the grid refined to a smaller scale. In particular for the case of EbFVM, involving only two neighbours for every CV the search of the matrix coefficients can be done by looking every elements at a time, without concatenations, simplifying furthermore the algorithm. This because every element has as extremes two centers of CVs, hence the temperature in it depends only that values that are in the element's domain, allowing to split the integral term in 14 into the pieces of each element in such a way that the algorithm can compute separately the pieces for each element and than sum for each CV the ones involved. The linear approximation for each element is than, for EbFVM

$$T(x) = N_A(x)T_A + N_B(x)T_B, \quad x \in (x_A, x_B) \tag{15}$$

where the subscripts A and B stands respectively for west and east extremes of the element. The same relation could be written using a local coordinate system, built on the element such that the first extreme is at $\xi = 0$ and the other at $\xi = 1$

$$T(\xi) = N_A(\xi)T_A + N_B(\xi)T_B, \quad \xi \in (0, 1). \tag{16}$$

The work now is about finding the shape functions. We clearly want them to be such that the temperature is exactly T_A on left head and T_B on right head, where the values are exactly known. This two conditions determine in a unique way the two linear functions because, in local domain

$$\left.\begin{array}{l} N_A(0)T_A + N_B(0)T_B = T_A \\ N_A(1)T_A + N_B(1)T_B = T_B \end{array}\right\} \implies \left.\begin{array}{l} N_A(0) = 1; N_B(0) = 0 \\ N_A(1) = 0; N_A(1) = 1 \end{array}\right\} \tag{17}$$

as expected, since the weighting functions has to be 0 on the coordinate of a known value point that it's not the one they are weighting. Given that the N_i are linear, this determines their form,

$$N_A = 1 - \xi; N_B = \xi \quad \xi \in (0, 1). \tag{18}$$

The domain transformation, looking at figure(6), has clearly a translation+dilation form, and is

$$\xi = \frac{x - x_A}{x_B - x_A} \quad x \in (x_A, x_B). \tag{19}$$

By this results we can compute explicitly the space derivative terms in (14), or rather the fluxes in the extreme of the CV, given T in the interval by (16), using the chain rule for derivation

$$\begin{aligned} \frac{\partial T}{\partial x} = \frac{\partial(N_A(\xi)T_A + N_B(\xi)T_B)}{\partial x} &= T_A\frac{\partial N_A(\xi)}{\partial x} + T_B\frac{\partial N_B(\xi)}{\partial x} \\ = T_A\frac{\partial N_A(\xi)}{\partial \xi}\frac{\partial \xi}{\partial x} + T_B\frac{\partial N_B(\xi)}{\partial \xi}\frac{\partial \xi}{\partial x} &= \frac{T_B - T_A}{x_B - x_A}, \end{aligned} \tag{20}$$

and it's than constant in the whole element, as expected since is linear. Another useful simplification can be made by assuming the temperature valuated in the center of the CV (here referred to with the letter

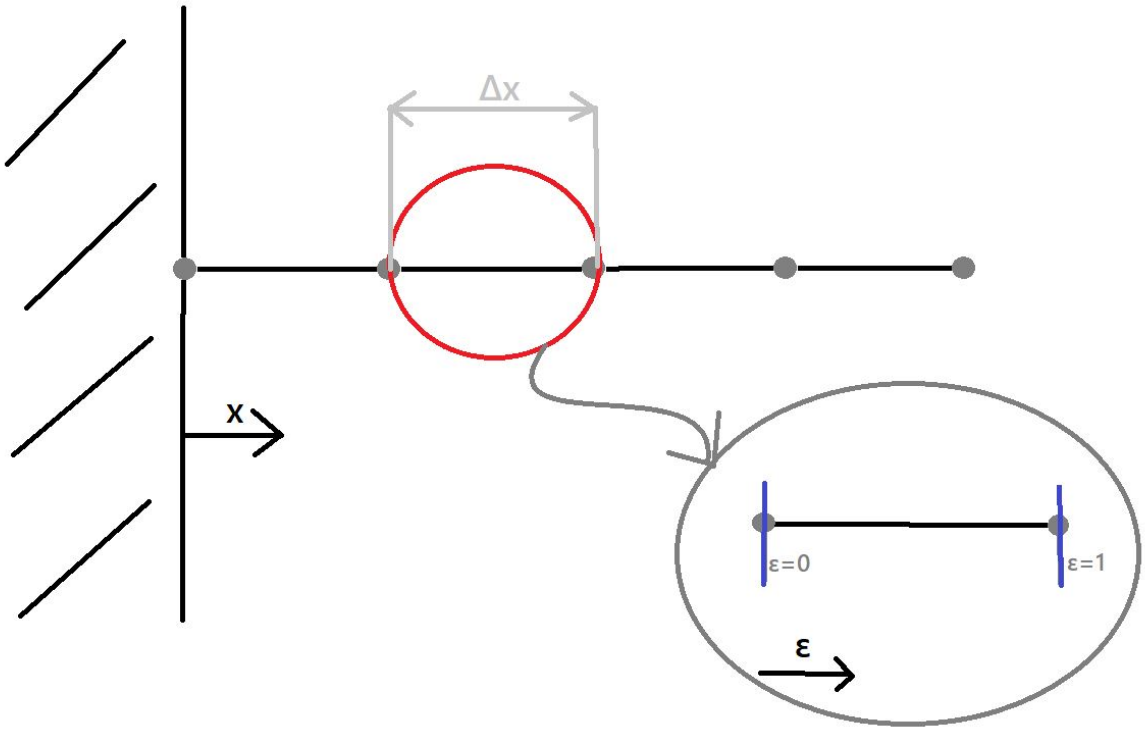

Fig.(6) 1D detail of real and local domain

P) as the integral mean of the $T(x,t)$ in $(x, x+dx)$, but note how this doesn't change the form of the matrix A, since the correlation between the temperature T_P and the two neighbours remain in the space derivative terms. Now, substituting everything inside (14) we have

$$kA(\frac{T_E^\theta - T_P^\theta}{\Delta x_e} - \frac{T_P^\theta - T_E^\theta}{\Delta x_w})\Delta t - hPT^\theta \Delta x \Delta t + (hPT_\infty + q'A)\Delta x \Delta t = \rho A C_P (T_P - T_P^O)\Delta x, \tag{21}$$

that is exactly the balance of energy in a discrete CV, and one can see the great similarity with the result in chapter (3). Dividing all by $\rho C_P \Delta t \Delta x$,

$$\frac{\alpha A}{\Delta x}(\frac{T_E^\theta - T_P^\theta}{\Delta x_e} - \frac{T_P^\theta - T_W^\theta}{\Delta x_w}) - \frac{hP}{\rho C_P}T^\theta + \frac{hPT_\infty + q'A}{\rho C_P} = \frac{A}{\Delta t}(T_P - T_P^O), \tag{22}$$

with $\alpha = \frac{k}{\rho C_P}$. The last step before writing the coefficients is the choice of the parameter θ. Given that θ is a time between t and t+Δt, according to the notation used in previous steps (O quote stands for old value, and no quote stands for new value), we can imagine, by using a linear interpolaton, $T_P(t)$ to be $T_P^\theta = T_P^O(1-\theta) + \theta T$, where $\theta = 0$ stands for the time t and $\theta = 1$ for time $t + \Delta t$. With some trivial computations we finally deduce

$$[A(\frac{1}{\Delta t} + \frac{\alpha\theta}{\Delta x}(\frac{1}{\Delta x_w} + \frac{1}{\Delta x_e}) + \frac{\theta hP}{\rho C_P}]T_P - \frac{\alpha\theta A}{\Delta x \Delta x_w}T_W - \frac{\alpha\theta A}{\Delta x \Delta x_e}T_E = \frac{1}{\rho C_P}(PhT_\infty + q') + \frac{A}{\Delta t}T_P^O + \\ + \frac{\alpha(1-\theta)A}{\Delta x \Delta x_w}T_W^O + \frac{\alpha(1-\theta)A}{\Delta x \Delta x_e}T_E^O. \tag{23}$$

The coefficient of the linear system are then

$$A_E = -\frac{\alpha\theta A}{\Delta x \Delta x_e}; A_W = -\frac{\alpha\theta A}{\Delta x \Delta x_w}; A_P = \frac{A}{\Delta t} - (A_E + A_W) + b\theta;$$
$$A_E^O = \frac{\alpha(1-\theta)A}{\Delta x \Delta x_e}; A_W^O = \frac{\alpha(1-\theta)A}{\Delta x \Delta x_w}; A_P^O = \frac{A}{\Delta t} - b(1-\theta) - (A_E^O + A_W^O);$$
$$B = A_P^O T_P^O + bT_\infty + \frac{q'A}{\rho C_P} + A_E^O T_E^O + A_E^O T_W^O,$$

where b is a parameter $b = \frac{Ph}{\rho C_P}$. The very remarkable property of this result is that every coefficient can be splitted into two components, each one depending only on its element. This because here the terms Δx_e and Δx_w, respectively the distances between point P and right neighbour center and between point P and left neighbour center, by construction are in fact the lengths of the two elements involved, and Δx can be splitted into the sum of half of the east element and half of the west element. Therefore it's easy to see how simple the construction is even for non uniform elements (as for example having the firsts elements with small size with respect to the lasts), because the part of the coefficient specific of one element it's completely independent from the other. Of course, since this is a 1D problem the benefits of this method may doesn't appear so important, but thinking at a 2D or a 3D computation, it's clear that represents all the geometries by Control Volumes (as would be applying the Cell Centered FVM), all with different forms to fill perfectly the domain, would be not so trivial to write explicitly what the derivative terms are, because this would involve distances between two centers of nearby elements that could be arranged in very different forms. Otherwise, with EbFVM, the domain is filled by elements, that are different entities from the CV, and all the coefficient can be seen as the sum of terms depending only on what's inside every element, that it's easily attainable by the local coordinates, known in general because given by the grid generator. The only extra calculation would be to determinate the determinant of the Jacobian of the transformation from physical domain to the local one (required in passages like (20)), but even if it's not known the analytical functions for the transformation, the Jacobian could be approximate easily thanks to the coordinates of the element extremes in both the domains. Note that in every case, with the linear approximation, the matrix developed is tridiagonal, and could be solved with a proper algorithm that allows to save a considerable amount of CPU time.

3.5 Explicit and Implicit formulations

As said above, the value of θ can be chosen arbitrarily between 0 and 1, but it's useful to have an idea of what consequences on the algorithm we should expect for each possibility. For $\theta = 0$, the method is called *Explicit*, because the dependence of the temperature at the new time level depends only on the previous time levels, all already computed, without concatenations between values of other points at this time level, that has not been computed yet. For this reason the matrix of the linear system is reduced to a diagonal one, and this admit the resolution looking each equation at a time, without really involving methods for the solution of a linear system, that normally represents the 50% of the computational time,

$$A_P T_P = A_P^O T_P^O + A_E^O T_E^O + A_W^O T_W^O + bT_\infty + \frac{q'A}{\rho C_P}. \tag{24}$$

Unfortunately, as we will see in next chapter, this formulation leads to a maximum step size in time very low, that makes impossible the study of the steady state without solving first the transient at least in a distorted way. The other extreme choice for θ is 1, called *fully implicit method* which switch the formulation to the dependence only on temperatures of neighbours points at the same level for which we are computing the solution, so in this has to be solved a linear system. The good news is that in this case the algorithm doesn't show any problem for time steps big, and one could easily reach the steady state with few computations. For θ between 0 and 1 the formulation is called (unfully) *implicit*, and uses both the values of the neighbour points at new and old time level. The more θ is near to 1, the more the linear matrix increase his diagonal and the algorithm can jump with higher steps in time. A summary of this formulations is showed in figure(7)

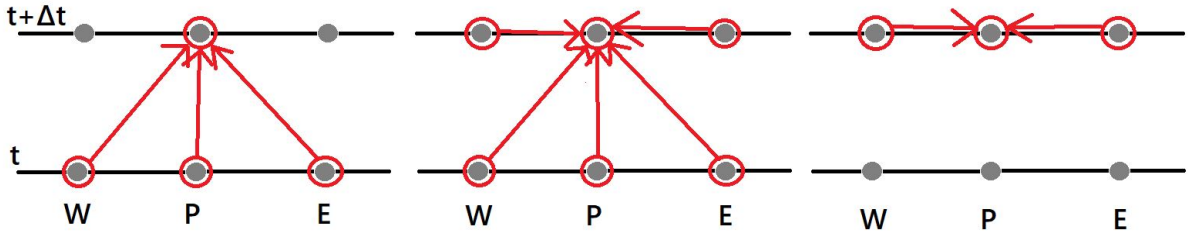

Fig.(7) Summary of the dependence of the three formulations, explicit (left), implicit (center), and fully implicit right)

3.6 Stability of the algorithm

Since the algorithm is computed by a discrete balance on the sub-domains, it's evident that refining the grid up to points would give exactly the real solution of the continuous PDE, which guarantees the consistency of the method. It's not so trivial anyway the stability condition in time for the numerical solution. Choosing arbitrarily a time coordinate t^* in $(0, +\infty)$, we could imagine the errors due by approximations, in every point of the space domain, to be $err_i = |T_{app}(x_i, t^*) - T_{real}(x_i, t^*)|$, where i goes from 1 to n, number of control volumes chosen for the algorithm. It's evident that the more the algorithm uses previous time solutions to compute the future time solution, the more it would be subjected to these errors and as will be showed below, would be mandatory the use of a well refined grid in time. We now call $\epsilon = max_{i=1,\dots,n}(err_i)$ the maximum error in the approximate solution at time t^*, and we try to maximize the error in the worst case, that is, by above considerations, the explicit one ($\theta = 0$), since it uses all the temperatures computed at previous time step. In this case the algorithm is reduced to

$$A_P T_P = A_P^O T_P^O + A_E^O T_E^O + A_W^O T_W^O + bT_\infty + \frac{q'A}{\rho C_P}. \tag{25}$$

In order to find the error, we subtract to this the computation made using a temperature "old" perfectly equal to the real one, named t. What we get is

$$A_P(T_P - t_p) = A_P^O(T_P^O - t_P^O) + A_E^O(T_E - t_E^O)^O + A_W^O(T_W^O - t_W^O), \tag{26}$$

since all the coefficients remains the same. Now we notice that on the left hand side we have the error at the "new" time step at point P, whereas on the right hand side appear errors on the "old" one. Taking the modulus on both sides, applying triangular inequality and bounding the errors with the maximum possible, we obtain

$$A_P\epsilon \leq |A_P^O|\epsilon^O + |A_E^O|\epsilon^O + |A_W^O|\epsilon^O \tag{27}$$

assuming P as the point where is located the maximum error at time $t + \Delta t$. Clearly, in order to have stability, we should verify that the error at new time level is smaller or equal than the previous one. This condition, from (27), becomes

$$\frac{|A_P^O| + A_E^O + A_W^O}{A_P} \leq 1. \tag{28}$$

Recalling that for $\theta = 0$ $A_P^O = \frac{A}{\Delta t} - b - A_E^O - A_W^O$, and $A_E^O = \frac{\alpha A}{\Delta x \delta x_e}$, $A_W^O = \frac{\alpha A}{\Delta x \delta x_w}$, it's clear that if we want to obtain a maximum time step allowable, which guarantees that all the time steps lowers than that are acceptable for the stability, the A_P^O coefficient will tend to be positive as the Δt decreases to 0. Moreover, looking at the coefficient in (28), it's clear that if $A_P^O \geq 0$, the condition would reduce to

$$\frac{\frac{A}{\Delta t} - b}{\frac{A}{\Delta t}} \leq 1, \tag{29}$$

that is clearly true for every time step that makes $A_P^O \geq 0$. This time step is

$$\frac{A}{\Delta t} - b - \frac{\alpha A}{\Delta x}\left(\frac{1}{\Delta x_e} + \frac{1}{\Delta x_w}\right) \geq 0 \implies \Delta t \leq \frac{A}{b + \frac{\alpha A}{\Delta x}\left(\frac{1}{\Delta x_e} + \frac{1}{\Delta x_w}\right)}, \tag{30}$$

and depends in such a way to the space step. We search if it's possible to have time steps bigger than the one that makes $A_P^O \geq 0$, so assuming that A_P^O could be also negative for some time steps. The condition now would be of the form

$$\begin{aligned} \frac{\Delta t(b + 2A_E^O + 2A_W^O)}{A} - 1 \leq 1 \implies \Delta t \leq \frac{2A}{b + \frac{2\alpha}{\Delta x}(\frac{1}{\Delta x_e} + \frac{1}{\Delta x_w})} \\ \Delta t \leq \frac{A}{\frac{b}{2} + \frac{\alpha A}{\Delta x}(\frac{1}{\Delta x_e} + \frac{1}{\Delta x_w})}, \end{aligned} \tag{31}$$

that is only slightly bigger than the previous result. In this way, the maximum time step that allow stability in the algorithm will be the minimum of this time steps for all the control volumes, since we want to proceed with the same velocity in all the CVs to follow the evolution in time of the problem. The numerical algorithm confirms this result, showing that every time step bigger than this will produce an increasing error in time, up to values too large for machine computations. On the other extreme, as written in last chapter, it's now clear that the implicit formulation doesn't have a bound for the time steps allowable, since the computation on the new time level depends only to T_P^O as temperature of the previous time level, and writing the majorization of the error in this case we would have

$$\epsilon(|A_P| - |A_E| - |A_W|) \leq |A_P(T_P - t_p) + A_E(T_E - t_E) + A_W(T_W - t_W)| = \epsilon^O A_P^O = \frac{A}{\Delta t}, \tag{32}$$

which means that the method is stable if

$$\frac{A_P^O}{|A_P| - |A_E| - |A_W||} \leq 1, \tag{33}$$

and this is obviously always true by looking at the coefficients.

3.7 Boundary conditions

We now want to discuss how to implement the boundary conditions (BC) of (7) (Dirichlet BC at left extreme and Neumann BC on the right one) in the algorithm. It's clear, by looking at figure(5) how the first boundary condition should be $T = T^*$ for the first point of the discretized domain, for all time values. It's not so clear anyway what the second boundary condition should involve for the algorithm. One possibility is to consider this BC from a mathematical point of view, which is that the derivative should be 0. This can be implemented by a finite difference scheme backwards, to approximate the derivative at right extreme. What we get is that the temperature on left extreme has to be equal to the previous point temperature, therefore this is the condition that we will apply in the computations. The more powerful solution is instead to approximate in some way the balance inside this half of Control Volume. Looking at (14) what we have here is that the right fluxes is 0 and the dx is the half of the normal one (if uniform grid). Thus we get

$$Aq^\theta|_w \Delta t - \int_x^{x+\frac{dx}{2}} hPT^\theta \Delta t dx + \int_x^{x+fracdx2} (hPT_\infty + q'A)\Delta t dx = \\ = \int_x^{x+fracdx2} \rho A C_P (T - T^O) dx \tag{34}$$

and we can compute the coefficients of the linear system for the boundary Control Volume with exactly the same procedure as above. Finally what we have is

$$\frac{-2\alpha A}{\Delta x}\frac{T_P^\theta - T_W^\theta}{\Delta x_w} - \frac{hP}{\rho C_P}T^\theta + \frac{hPT_\infty + q'A}{\rho C_P} = \frac{A}{\Delta t}(T_P - T_P^O), \tag{35}$$

$$A_W = -\frac{2\alpha\theta A}{\Delta x \Delta x_w}; A_P = \frac{A}{\Delta t} - A_W + b\theta;$$

$$A_W^O = \frac{2\alpha(1-\theta)A}{\Delta x \Delta x_w}; A_P^O = \frac{A}{\Delta t} - b(1-\theta) - A_W^O;$$

$$B = A_P^O T_P^O + bT_\infty + \frac{q'A}{\rho C_P} + A_E^O T_W^O,$$

3.8 Steady state condition

The algorithm is obviously computed step time by step time from a software, but of course we have to established a maximum time value, so that the program doesn't run forever. The time searched is the one after which we cannot obtain any more information on the problem, and for this reason is the time of the achievement of the steady state condition. As said in the introductory chapters, the steady state condition can be computed as the situation in which the energy entering in the system, both by fluxes and internal generation, should be equal to the one leaving the Body, and this could be reduced to every Control Volume. Another condition of very simple implementation is the check of stability by a control on the temperature variation in the points of the discretized domain, since is known that at the steady state condition the solution stay constant along all the times. This could be analyzed by a check that the norm of the temperature difference in time (that is a vector, because it's computed point by point) is below an arbitrarily chosen tolerance.

3.9 Check of convergence

In order to find the check the convergence of the numerical solution to the analytical one, one can operates in two main ways, checking the residual or checking the difference between two solutions with different size of the grid, up to a certain tolerance chosen. Since in general the control over the margin is more effective to approximate the real error, in this paper the control will be made following the second way. Starting from the physical domain, will be computed a first trial solution, then, refining the grids decreasing the size used in previous computation will be obtained a second trial approximation for the problem, always with the same hypothetical time step (for convenience we will take the last one, the one of the steady state condition). If the difference in norm between the two approximations (valuated in the same points, at the same time) is below a chosen tolerance, we will consider the second solution to be valid, otherwise we'll start repeating this operation up to the tolerance condition. Then, when the maximal space size is found, we will repeat the same operation with the time coordinate, considering the norms of the differences in time of the vectors containing the norms in space domain at the first three different time values, t=0s, $t = \Delta t$ and $t = 2\Delta t$ (or by considering the difference in norm between the matrices containing the values of T in space and time), up to convergence. This choice of having the check centered in the first time values is reasonable, due to the fact that the gradient in time for the solution will reach the highest values in the first instants. The extra possibility for the check of convergence is the comparison with the analytical solution, if given, anyway, of course, for all practical problem the analytical solution cannot be computed or is of very hard computation, otherwise we wouldn't have to use numerical methods.

3.10 Finite Element Method

We now start to analyze the problem using the so called *Finite Element Method (FEM)*. This method is substantially a numerical algorithm generated by the WRA applied to a variational formulation of the starting PDE. The variational formulation is a procedure based on Gauss theorem (that implies the integration by parts rule), which allows to transform in some sense the initial PDE to an analogous one, but relaxing the continuity requirements for the solution as we will see below. For this reason, the PDE that we will solve with this method is called *weak formulation* and its solution *weak solution*, since it has has weaker continuity properties. For some problems it's also possible to find a weak solution analytically, using variational principle (as virtual work principle) to compute the variational formulation, and then finding a functional which minimization leads to the solution, but this procedure requires a strong mathematical background and particular information and conditions in the original PDE, that are

usually not given. The starting idea of this method is to find a solution in the form of

$$\phi = \sum_j c_j \psi_j, \tag{36}$$

where the ψ_j are hypothetical trial functions and the c_j are coefficients such that $\mathcal{L}(\phi) = 0$. Anyway it's clear that it's almost impossible to guess trial functions that interpolates perfectly the real behaviour of the solution on the whole domain, so the intention is to split the space domain in pieces so small, that even if we aren't using the exactly right ψ_j the error is sufficiently small, given a proper choice of the coefficients c_j. We call *Elements* the sub-domains generated. In order to find this coefficients, as written above, we would like to apply the WRA to minimize in some sense the residual of the PDE with the solution T written as in 36. Therefore we first write the residual statement, fixing a time value t and integrating the PDE in his domain

$$\begin{aligned}\int_\Omega \mathcal{L}(\phi) = 0 \implies \int_0^\infty \int_0^L w[A\frac{\partial(k\partial T)}{\partial x^2} - hP(T - T_\infty) + q'A]dxdt = \\ = \int_0^\infty \int_0^L wA\frac{\partial(\rho C_P T)}{\partial t}dxdt.\end{aligned} \tag{37}$$

We now want to simplify the second derivative term on the left hand side, in a different way from the FVM, that means without forcing the weighting function to be 1 in each sub-domain at a time and 0 in all others. Anyway, we still consider w to be different from 0 only on the desired sub-domain (that here is the element, since there is not such concept of Control Volumes, times the time step from the previous solution to the "new one"), As already said, the process involves the use of the "Integration by parts law". What we get is

$$\begin{aligned}\int_A^B wA\frac{\partial(k\partial T)}{\partial x^2}dx = \int_A^B \frac{d}{dx}(wkA\frac{dT}{dx})dx - \int_0^L \frac{dw}{dx}kA\frac{dT}{dx}dx \\ = (wkA\frac{dT}{dx})|_B - (wkA\frac{dT}{dx})|_A - \int_A^B \frac{dw}{dx}kA\frac{dT}{dx}dx = (w|_A Q_A - w|_B Q_B) - \int_A^B \frac{dw}{dx}kA\frac{dT}{dx}dx,\end{aligned} \tag{38}$$

where A and B are the extremes of the element chosen. Note how this transformation "moves" one of the derivatives that was on the temperature, to the weighting function, "relaxing" the condition of T in $\mathcal{C}^2(0, L)$ to compute a weak solution of the form T in in $\mathcal{C}^1(0, L)$ for in the space coordinate "x". In this case there is only one partition of the initial domain, and the sub-domains are called only Elements which means that doesn't exist the concept of Control Volume and the resulting algorithm has nothing to do with the balance in a discretized domain, which implies that the approximate solution could not satisfy the balance of energy in each element. We now make the choice of w in each element as $w_i = \psi_i$, or rather equal to the trial functions in each element, and of searching for a solution in the form $T(x,t) = T_A(t)\psi_1(x) + T_2(t)\psi_2(x)$, with the trial function constants in time, and linear interpolation inside of the element (for compatibility to what has been done with the EbFVM). It's clear that definition of the c_i, at a fixed time value, $T(x) = T_A\psi_1(x) + T_B\psi_2(x)$ is nothing but a weighting with apposite shape functions ψ_1 and ψ_2 of the temperature at the extremes of the element, and then could be determined as in paragraph (3.4) with (18)-(19). Therefore, in principle, the method gives for each element one equation for each shape function (2 in the case of linear elements) with 2 unknowns, and for each one could be created a "local" linear system. But considering n adjacent elements, this would bring to 2n equations for n+1 real unknowns, because the number of vertex of n elements all attached to one other from one extreme, except the first and last one, is n+1. The solution to this problem is given by the superposition of effects principle, which implies that on each vertex we can sum together the contribution from the left element and right one, obtaining a new linear system, whose matrix and known term are constructed by the ones of the elements. Considering the i_{th} element, and weight functions equal to shape functions, (37) becomes

$$\begin{aligned}\int_t^{t+\Delta t} \int_A^B [-kA\frac{d\psi_i}{dx}\frac{dT}{dx} - \psi_i hP(T - T_\infty) + \psi_i q'A]dxdt + \int_t^{t+\Delta t} \psi_i|_A Q_A - \psi_i|_B Q_B = \\ = \int_t^{t+\Delta t} \int_A^B \psi_i A\frac{\partial(\rho C_P T)}{\partial t}dxdt,\end{aligned} \tag{39}$$

where $\psi_1 = 1-\xi, \psi_2 = \xi$. Now for since the solution is searched to be continuous and the domain of integration is bounded, we can switch the integrals by Fubini-Tonelli theorem and work first on the integral in time. As in the EbFVM we approximate the integrals of type $constant * T(t)$ or $constant * \frac{dT}{dx}$ to have integral as average themselves valuated at a time value *theta* between t and $t+\Delta t$, where we commonly indicate time t for $\theta = 0$ and time $t+\Delta t$ for $\theta = 1$. Note that since we've chosen the weighting functions as the trial functions, constants in time, last equation reduces to

$$\begin{aligned}\int_A^B [-kA\frac{d\psi_i}{dx}\frac{dT^\theta}{dx} - \psi_i hP(T^\theta - T_\infty) + \psi_i q'A]dx + \psi_i|_A Q_A^\theta - \psi_i|_B Q_B^\theta = \\ = \frac{1}{\Delta t}\int_A^B \psi_i A(\rho C_P(T - T^O))dx.\end{aligned} \tag{40}$$

It's now easier to compute one term at a time. The first one becomes

$$\begin{aligned}&\left.\begin{aligned}-kA\int_A^B \frac{d\psi_1}{dx}\frac{d}{dx}(T_1^\theta\psi_1 + T_2^\theta\psi_2)dx \\ -kA\int_A^B \frac{d\psi_2}{dx}\frac{d}{dx}(T_1^\theta\psi_1 + T_2^\theta\psi_2)dx\end{aligned}\right\} \implies \left.\begin{aligned}-kA(T_1^\theta\int_A^B (\frac{d\psi_1}{dx})^2dx + T_2^\theta\int_A^B \frac{d\psi_1}{dx}\frac{d\psi_2}{dx}dx) \\ -kA(T_1^\theta\int_A^B \frac{d\psi_2}{dx}\frac{d\psi_1}{dx}dx + T_2^\theta\int_A^B (\frac{d\psi_2}{dx})^2dx)\end{aligned}\right\} \\ &\implies \left.\begin{aligned}-\frac{kA}{\Delta x}(T_1^\theta\int_0^1 (\frac{d\psi_1}{d\xi})^2d\xi + T_2^\theta\int_0^1 \frac{d\psi_1}{d\xi}\frac{d\psi_2}{d\xi}d\xi) \\ -\frac{kA}{\Delta x}(T_1^\theta\int_0^1 \frac{d\psi_2}{d\xi}\frac{d\psi_2}{d\xi}d\xi + T_2^\theta\int_0^1 (\frac{d\psi_2}{d\xi})^2d\xi)\end{aligned}\right\} \implies \left.\begin{aligned}\frac{kA}{\Delta x}(-T_1^\theta + T_2^\theta) \\ \frac{kA}{\Delta x}(T_1^\theta - T_2^\theta),\end{aligned}\right\}\end{aligned} \tag{41}$$

where Δx is the length of the element. For both second term and right hand side of (40) it's useful to compute

$$\left.\begin{aligned}\int_A^B \psi_1 T = \Delta x\int_0^1 (1-\xi)(T_1(1-\xi) + T_2\xi)d\xi \\ \int_A^B \psi_2 T = \Delta x\int_0^1 \xi(T_1(1-\xi) + T_2\xi)d\xi\end{aligned}\right\} \implies \left.\begin{aligned}\Delta x(\frac{T_1}{3} + \frac{T_2}{6}) \\ \Delta x(\frac{T_1}{6} + \frac{T_2}{3})\end{aligned}\right\} \tag{42}$$

Note that this result is valid also for the integral of T^θ and T^O. Thus the only integral type left in (40) is the integration of the constant $c = hPT_\infty + q'A$ times the weighting function, that is

$$\left.\begin{aligned}\int_A^B c\psi_1 = c\Delta x\int_0^1 (1-\xi)d\xi \\ \int_A^B c\psi_2 = c\Delta x\int_0^1 \xi d\xi\end{aligned}\right\} \implies \left.\begin{aligned}\frac{c\Delta x}{2} \\ \frac{c\Delta x}{2}\end{aligned}\right\} \tag{43}$$

The resulting equations for the element are, substituting $T^\theta = (1-\theta)T^O + \theta T$ and dividing all by $\rho C_P\Delta x$

$$\begin{aligned}T_1[\frac{A}{3\Delta t} + \theta(\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{3})] + T_2[\frac{A}{6\Delta t} + \theta(-\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{6})] = Q_A + \frac{1}{2}(\gamma T_\infty + \frac{q'A}{\rho C_P}) + \\ + T_1^O[\frac{A}{3\Delta t} - (1-\theta)(\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{3})] + T_2^O[\frac{A}{6\Delta t} - (1-\theta)(-\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{6})]\end{aligned} \tag{44}$$

$$\begin{aligned}T_1[\frac{A}{6\Delta t} + \theta(-\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{6})] + T_2[\frac{A}{3\Delta t} + \theta(\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{3})] = Q_B + \frac{1}{2}(\gamma T_\infty + \frac{q'A}{\rho C_P}) + \\ + T_1^O[\frac{A}{6\Delta t} - (1-\theta)(-\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{6})] + T_2^O[\frac{A}{3\Delta t} - (1-\theta)(\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{3})],\end{aligned} \tag{45}$$

where $\gamma = \frac{hP}{\rho C_P}$, and the terms Q_A and Q_B has been computed noting that $\psi_1(A) = 1, \psi_1(B) = 0, \psi_2(A) = 0, \psi_2(B) = 1$. We have then found the linear system of the element, in the form [K][T]=[B], where

$$K = \begin{bmatrix} \frac{A}{3\Delta t} + \theta(\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{3}) & \frac{A}{6\Delta t} + \theta(-\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{6}) \\ \frac{A}{6\Delta t} + \theta(-\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{6}) & \frac{A}{3\Delta t} + \theta(\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{3}). \end{bmatrix} \tag{46}$$

and

$$B = \begin{bmatrix} Q_A \\ Q_B \end{bmatrix} + \begin{bmatrix} \frac{1}{2}(\gamma T_\infty + \frac{q'A}{\rho C_P}) + T_1^O[\frac{A}{3\Delta t} - (1-\theta)(\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{3})] + T_2^O[\frac{A}{6\Delta t} - (1-\theta)(-\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{6})] \\ \frac{1}{2}(\gamma T_\infty + \frac{q'A}{\rho C_P}) + T_1^O[\frac{A}{6\Delta t} - (1-\theta)(-\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{6})] + T_2^O[\frac{A}{3\Delta t} - (1-\theta)(\frac{\alpha A}{\Delta x^2} + \frac{\gamma}{3})] \end{bmatrix} \quad (47)$$

Notice that K is symmetric, that is a general property for the method, but in this case also the diagonal entries are equal one to each other, $K_{1,2} = K_{2,1}, K_{1,1} = K_{2,2}$. The aim is therefore to *assemble* the linear system of the whole domain by sum in a proper way of the element ones. Since the first equation of this systems regards the left vertex of the element, and the second the right one, the sum is going to be operated by summing the second equation of each element with the first one of the following one. Calling $K_{i,j}$ the entries of the element matrix, the assembled linear system of three elements of equal length is

$$[A][T] = \begin{bmatrix} K_{1,1} & K_{1,2} & 0 & 0 \\ K_{1,2} & 2K_{1,1} & K_{1,2} & 0 \\ 0 & K_{1,2} & 2K_{1,1} & K_{1,2} \\ 0 & 0 & K_{1,2} & K_{1,1} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} B_1 + Q_1 \\ B_2 + B_1 \\ B_2 + B_1 \\ B_2 + Q_4 \end{bmatrix} \quad (48)$$

since all the fluxes in central vertex cancel out each other. Note that this matrix is tri-diagonal, as in the EbFVM.

3.11 Stability of the algorithm

Also for this method the consistency property it's trivial, because if we reduce the elements to be point the interpolation by the trial functions inside the element will tend to establish the exact solution at such point. The stability condition it's anyhow analogous to the one treated in the EbFVM, since both represents the solution of a linear system and from a numerical point of view the only difference is in the values of the coefficients. It could be then found a maximum time step adaptable to the explicit formulation following the same steps. In the case of uniform grid the result is

$$\Delta t \leq \frac{A}{6(\frac{\alpha A}{\Delta x^2} + \gamma)} \quad (49)$$

3.12 Boundary Condition

As for the EbFVM, the Dirichlet boundary condition it's trivial to implement, since here as there we have one point of the discretized domain located at the boundary, so it's just $T_1 = T_0$, where T_1 is the temperature of the first vertex of the first element. The second boundary condition, is a Neumann type, prescribing the flux trough the right face to be 0. But considering the for last element linear system the known term $\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} + \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$, we note that Q_1 will be canceled out by the assembling procedure, and Q_2 is exactly the flux that we want to be 0, so to implement the Neumann BC is just necessary to put that term equal to 0.

3.13 Check of convergence and Steady State condition

As for the EbFVM we will use the convergence to the Steady State condition as a parameter to stop the iterations in time, and the check adopted is the same used for EbFVM. Also the check for convergence to the real solution follows the same steps already seen.

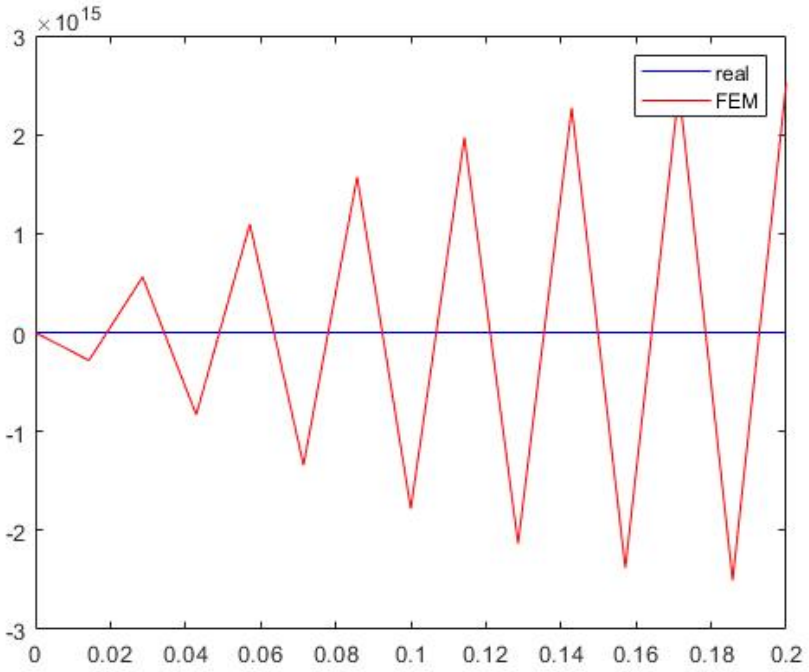

Fig.(8) Example of a explicit formulation solution at steady state with a time step slightly higher than the maximum allowable. See how the solution oscillates more and more around the analytical one

4 Results

In this chapter will be tested the goodness of the algorithm, by both physical and numerical point of view, and will be presented and compared the solutions obtained from the two methods.

4.1 Validation

4.1.1 Tests on physical behaviour

A first step to check the validity of those algorithms is by checking that extreme condition of physical properties produce the trivial solutions that we should expect. For example, since the initial solution is in equilibrium with the environment until the left surface is raise to the temperature T_0, we would like the algorithms to find a constant solution in the case of $T_0 = T^*$. And this is exactly what we get in both methods, see for example figure(9). Another good test is that if the conductivity in the material is decreased to 0, this would have to imply that the heat cannot propagate through the bar, so the solution should stay equal to the initial condition, as before, but with left extreme raised to T_0. Also this property is satisfied by both methods, Fig.(10-12), even if the FEM show an oscillations problem, due to the trial of interpolate the initial discontinuity by the trial functions of the first elements. The opposite should happen if k is raised to ∞, with the temperature along the bar constant to T_0. See Figure(13). Other good checks, done, but not showed are the verification that as L tends to ∞ the flux entering from left surface doesn't change the solution inside the body, that as h goes to ∞ the temperature in the bar stay fixed to T_∞, and the check of the result found by Scale Analysis (that will be prove true for both methods in next chapters).

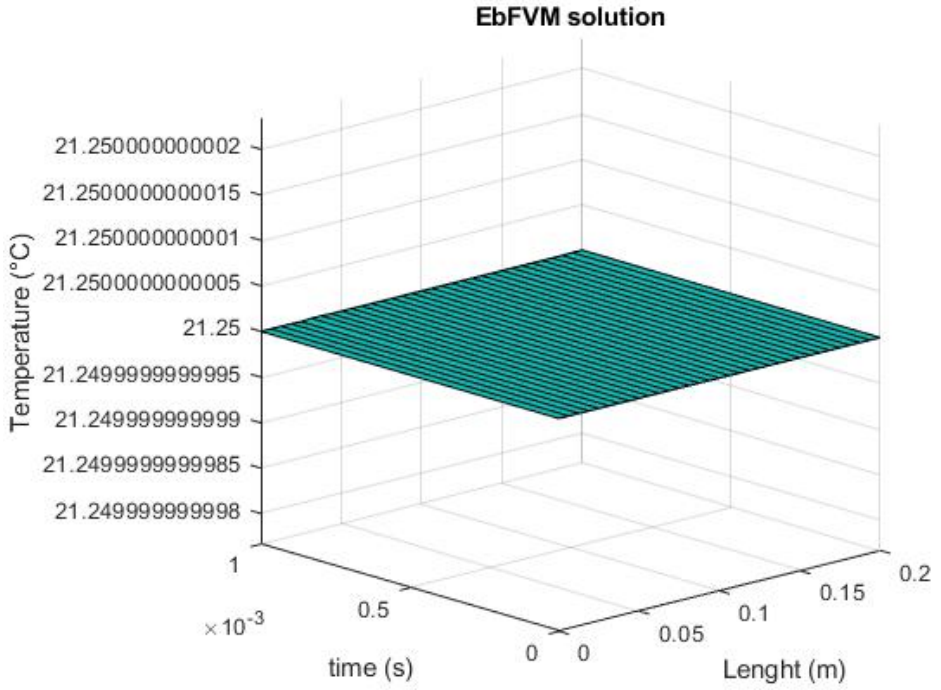

Fig.(9) Solution in the EbFVM for $T_0 = T^*$

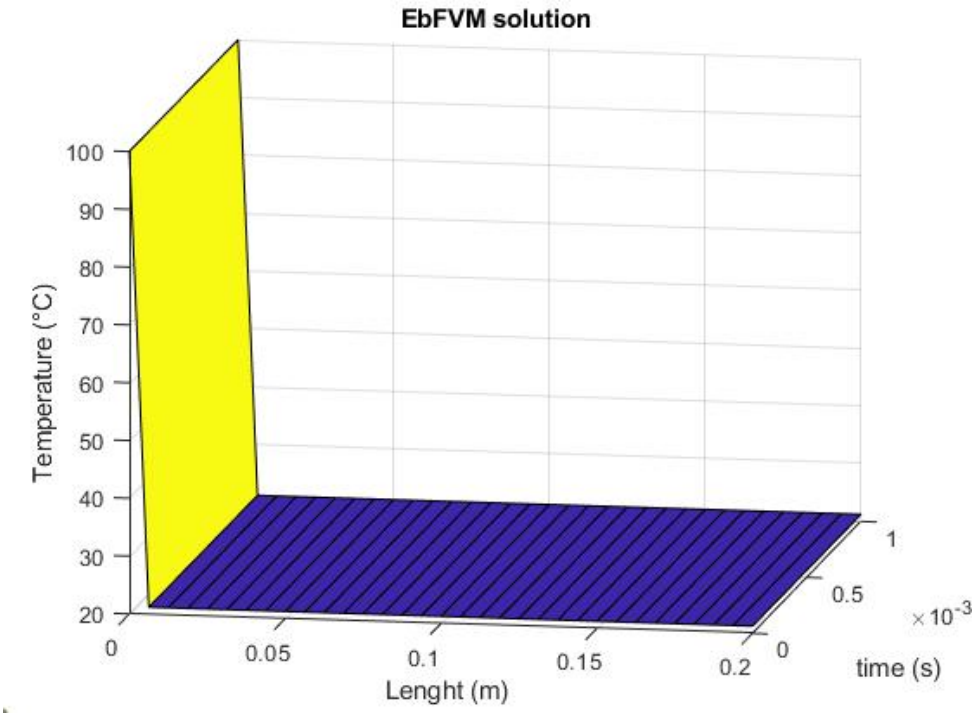

Fig.(10) Solution in the EbFVM for $k = 0$

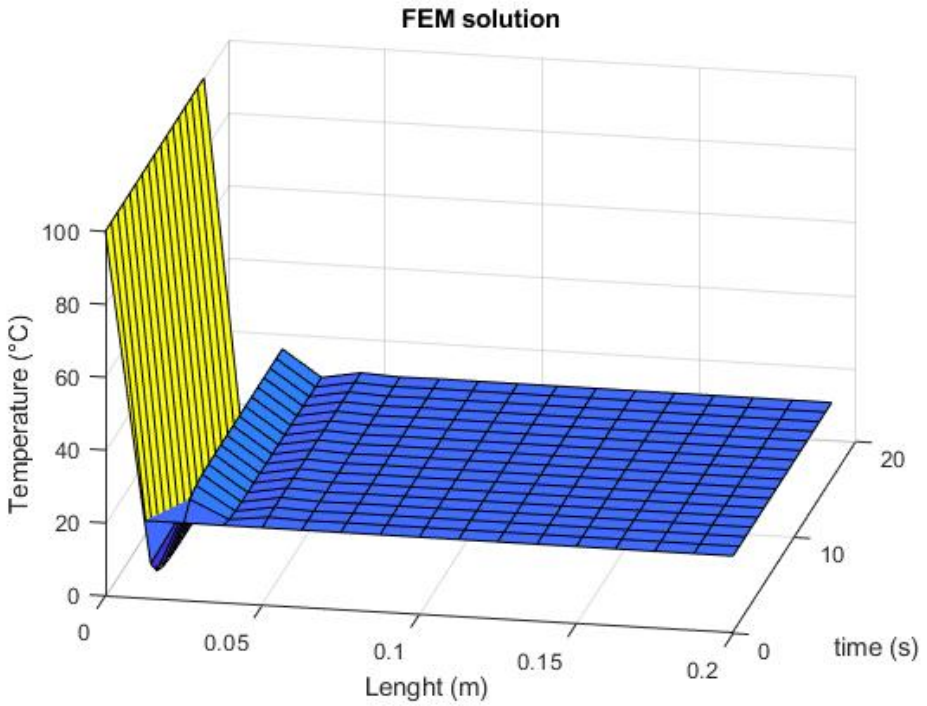

Fig.(11) Solution in the FEM for $k = 0$ and 17 elements. Numerical method tries to interpolate the discontinuities and produce errors. Increasing the number of elements this effect tends to disappear

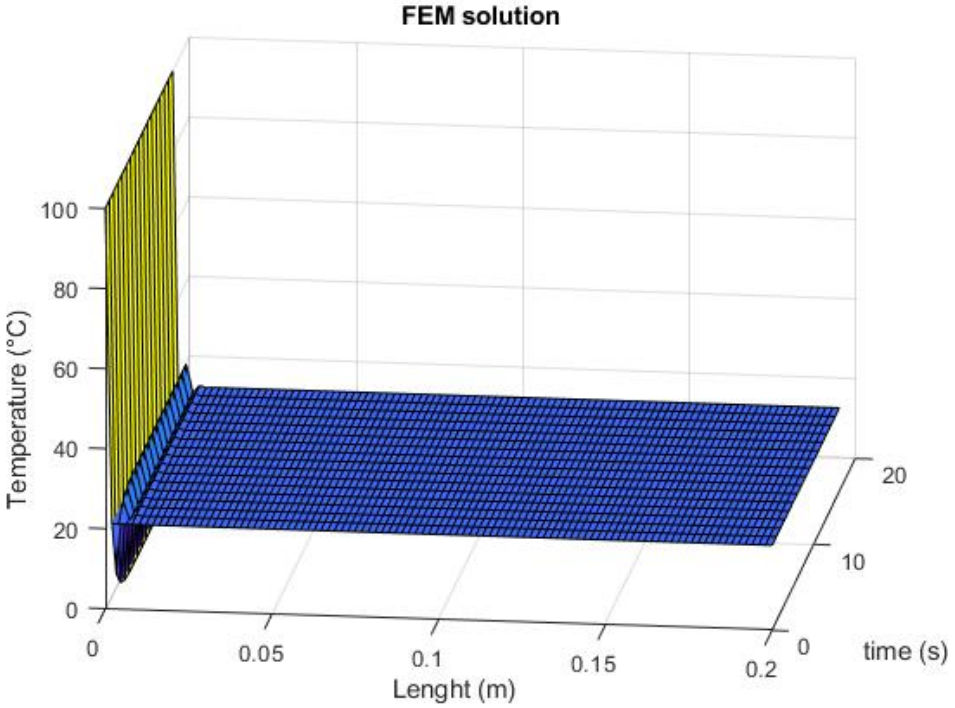

Fig.(12) Solution in the FEM for $k = 0$ and 100 elements. Note how the previous oscillations are redefined in the very first spacial steps.

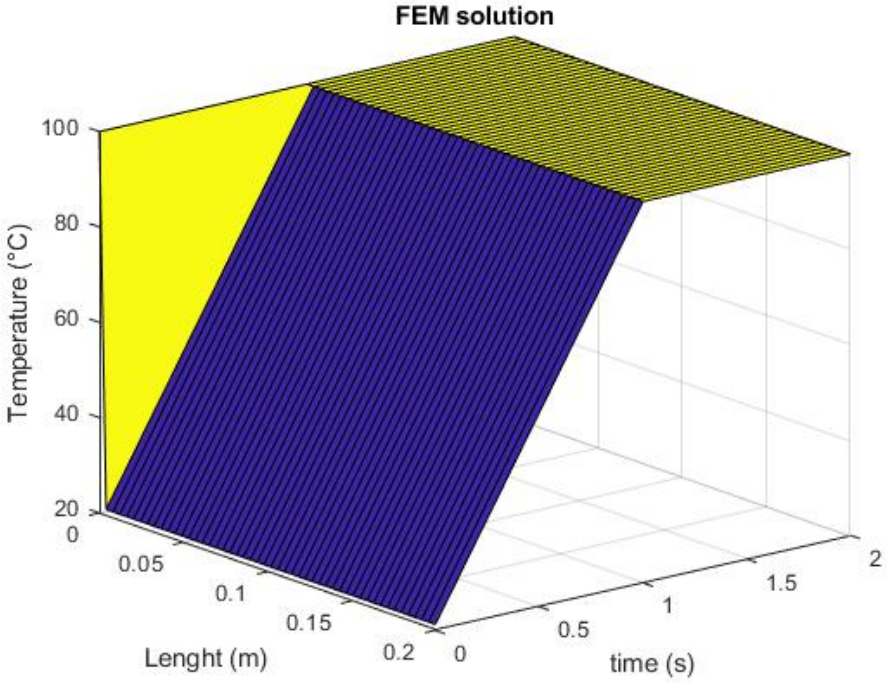

Fig.(13) Solution in the FEM for $k = 1000$. Solution in space tends uniformly to the temperature T_0

4.2 Analytical solution for Steady State

As said in the introductory chapters, it's possible to find an analytical solution for this problem in the Steady State condition. Assuming time independency in (7), we want to solve, dividing all by kA,

$$\left.\begin{aligned} \frac{\partial^2 T(x))}{\partial x^2} - \gamma^2 T(x) = -\gamma^2 T_\infty + \frac{q'}{k} \quad \text{for } x \in (0, L), \\ T(0) = T_0;\ \frac{\partial T(0)}{\partial x} = 0 \end{aligned}\right\} \tag{50}$$

that is a non-homogeneous ODE Cauchy problem, where $\gamma = \sqrt{\frac{hP}{kA}}$. We can split the resolution of this problem into find a u(x) that solves completely the homogeneous form of the ODE and a particular v(x) that solves the non-homogeneous problem, such that T(x)=u(x)+v(x). The particular could be found by a constant, v(x)=$\gamma^2 T_\infty + \frac{q'}{k}$, that clearly satisfies Neumann BC, but non the Dirchlet one. In this way u(x) has to satisfies a redefined Cauchy problem,

$$\left.\begin{aligned} \frac{\partial^2 u(x))}{\partial x^2} - \gamma^2 u(x) = 0 \quad \text{for } x \in (0, L), \\ T(0) = T_0 - v;\ \frac{\partial T(0)}{\partial x} = 0 \end{aligned}\right\} \tag{51}$$

This ODE can be easily solved by applying characteristic polynomial approach, $\lambda^2 - \gamma^2 = 0$, that implies $\lambda = \pm\gamma$. Therefore the solution is

$$u(x) = c_1 e^{-\gamma x} + c_2 e^{\gamma x}, \tag{52}$$

for an appropriate choice of the constants. Imposing the BC we find $c_1 = \frac{T_0 - v}{1+e^{2\gamma L}}$ and $c_2 = \frac{(T_0 - v)e^2\gamma L)}{1+e^{2\gamma L}}$, therefore

$$u(x) = \frac{T_0 - v}{1 + e^{2\gamma L}}(e^{\gamma(2L-x)} + e^{\gamma x}), \tag{53}$$

and

$$T(x) = \frac{T_0 - v}{1 + e^{2\gamma L}}(e^{\gamma(2L-x)} + e^{\gamma x}) + v. \tag{54}$$

This solution will be used below as a test for convergence of the two methods.

4.3 Results

Now that has been proved the validity of the two methods is possible to compute the two numerical solutions, following the steps for convergence discussed above.

4.3.1 Implementation in Matlab solver of the EbFVM

The code implemented in the software Matlab for the computations is, choosing a uniform grid, so that $\Delta x = \Delta x_e = \Delta x_w$,

```
% 1D steady state heat conduction problem
%{
L: length of the rod [m]
k: thermal conductivity of the rod [W/(mK)]
h:  heat transfer coefficient [W/(m^2K)]
T_0: base temperature [K]
T_inf: Temperature of the environment [K]
```

```
A: cross Area of the rod (constant) [m^2]
n: number of control volumes
Cp: specific heat with constant pressure [kJ/(kgK)]
q:heat generation [W/m^3]
rho: density [kg/m^3]
K: number of iterations to achieve steady state condition
toll: tolerance chosen for the check on the steady state condition
%}
clear
close all
T_zero=21.25;
L=0.2;
ex=1e-2;
k=30; h=20; q=1e4;
Cp=0.42; rho=8700;
T_0=100; T_inf=20;
Area=ex*ex;
alpha=k/rho/Cp;
P=4*ex;
n=5;
dx=L/(n-1);  %size of the space domain, length of the elements
b=P*h/rho/Cp;
theta=input('digit 0 for explicit method, 1 for fully implicit, theta= ');
dt=Area/(2*alpha*Area/dx^2+h*P*dx/rho/Cp);
if theta==1
    dt=1;
end
% REAL SOLUTION (STATIONARY)
gamma=sqrt(P*h/k/Area); sol_part=T_inf+q./k/gamma^2; T_gamma=T_0-sol_part;
T_real=@(x) ...
    T_gamma./(exp(-gamma.*L)+exp(gamma.*L))*(exp(gamma.*(L-x))+exp(gamma.*(x-L)))+sol_part;
%IMPLICIT
dN1=-1/dx;
dN2=1/dx;
c=b*T_inf+q*Area/rho/Cp;
%CREATION OF THE LINEAR SYSTEM
Ae=-alpha*Area*theta/dx*dN2;       Aw=alpha*Area*theta/dx*dN1;
Ae_o=alpha*Area*(1-theta)/dx*dN2; Aw_o=-alpha*Area*(1-theta)/dx*dN1;
Ap=Area/dt-Ae-Aw+b*theta;
Ap_o=Area/dt-(1-theta)*b-Ae_o-Aw_o;
K=1;
toll=1e-6;
diff=toll+1;
T=ones(1,n)*T_zero; T(1)=T_0;
A=diag(ones(n,1)*Ap)+diag(ones(n-1,1)*Aw,-1)+diag(ones(n-1,1)*Ae,1); A(1,2)=0;
A(1,1)=1; A(n,n-1)=Aw*2; A(n,n)=Area/dt-2*Aw+b*theta;
B=zeros(1,n);
tic;
while diff>toll
    K=K+1; %UPDATING NUMBER OF ITERATIONS
    for i=2:n-1
        B(i)=c+Ap_o*T(K-1,i)+Ae_o*T(K-1,i+1)+Aw_o*T(K-1,i-1);
    end
    B(1)=T_0; B(n)=c+(Ap_o+Ae_o-Aw_o)*T(K-1,i)+2*Aw_o*T(K-1,i-1);
    T(K,:)=(A\B')';
    diff=norm(T(K-1,:)-T(K,:)); %CONTROL OF CONVERGENCE TO STEADY STATE BY THE WASTE

end
toc;
%CHECK OF STEADY STATE CONDITION BY ENERGY BALANCE
E_in=-k*Area*(T(end,2)-T(end,1))/dx; %ENERGY ENTERING BY LEFT SURFACE
E_out=0;
for i=2:n
E_out=E_out+h*P*dx*((T(end,i)+T(end,i-1))/2-T_inf); %ENERGY LEAVING BODY BY CONVECTION
end
TOLL=abs(E_in+q*L*ex^2-E_out);
if TOLL>0.5
    warning('Error, the steady state is not verified by energy balance');
end
figure(1)
```

```
77
78 %PLOT OF THE SOLUTION IN A 3D GRAPH
79 N=length(T(:,1));
80 x = linspace(0,L,n);
81 t = linspace(0,dt*N,N);
82 [X,tempo]=meshgrid(x,t);
83 surf(X,tempo,T) % mesh
84 xlabel('Lenght (m)');
85 ylabel('time (s)');
86 zlabel('Temperature ( C )')
87 title('EbFVM solution')
88
89 %COMPARISON BETWEEN REAL AND APPROXIMATE SOLUTION FOR STEADY STATE
90 XX=0:dx/2:L;
91 YY=T_real(XX);
92 figure(2)
93 plot(XX,YY,'b-');
94 hold on
95 plot(0:dx:L,T(end,:),'r-')
96 hold off
97 legend('real','EbFVM')
```

The passages used to reach convergence, with $\theta = 1$, fully implicit method, will be shown in the next images (14)-24) . In this case the analytical solution for the steady state is known, as we will see in chapter(4), so the convergence in space domain is shown by both comparing the numerical solution to the real one, and checking the difference in norm with previous solution. We choose tolerance 0.005 for the convergence in both space and time domain, because are values mediated between very large sets of points, so even if this might looks an high tolerance it will guarantee the closeness with the real solution. Anyhow it's useful have in mind what we are looking for: a very smooth function in $\mathcal{C}^2(0, L) \cap \mathcal{C}^1 \times (0, +\infty)$ (2.2), such that there is no discontinuities point also in the derivatives of the solution inside of the domain,

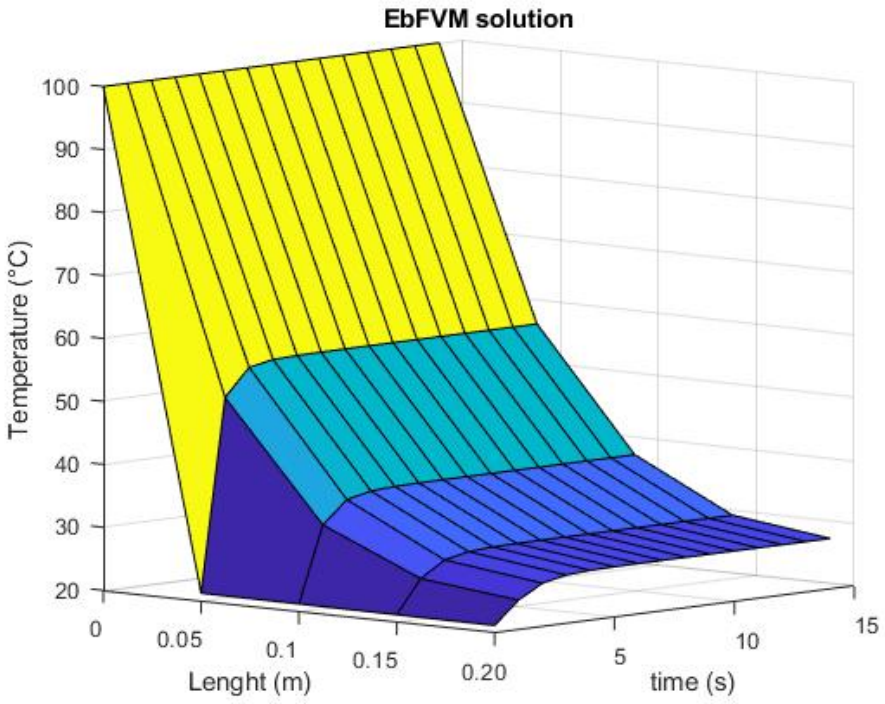

Fig.(14) EbFVM solution for 5 CVs $\Delta x = 0.05m$, $\Delta t = 1s$. Note that the solution is not smooth (in both coordinates), as expected by the physics study, so it's probably far from convergence.

We now concentrate only in he firsts two time steps, in order to find the maximal step that guarantees a good approximation for the initial instants(that intuitively could be seen by the smoothness of the graph in time). Then, since the strongest gradient is obviously located at the initial point that time steps would be certainly enough for the approximation of the rest of the domain.

The final solution will then be implemented with initial step $\Delta x = 0.0125m$ and $\Delta t = \frac{1}{200}$. Anyhow its not necessary to keep those very small steps until the right extreme of the fin on one side, and until the steady state condition from the other, because the real problem in the approximation is in the first instants and at the beginning of the bar. Noting that what would bring very high number of iterations is the time step, we choose to increment the time step in the algorithm little by little after the very first iterations (bounding its maximum with 1 s). What we get is figure(24)

Notice that the tolerance chosen for the convergence to the steady state condition is 10^{-6}, and this coincide with the achievement of the balance of energy condition in the body (this last value is more subject to the grid in space, since it's computed with a very rough approximation in (4.3.1)). It's clear by the picture that this may be a too high tolerance, since it looks like practically the steady state is achieved seconds before the end time founded (approximately 13 seconds), but it all depends on what

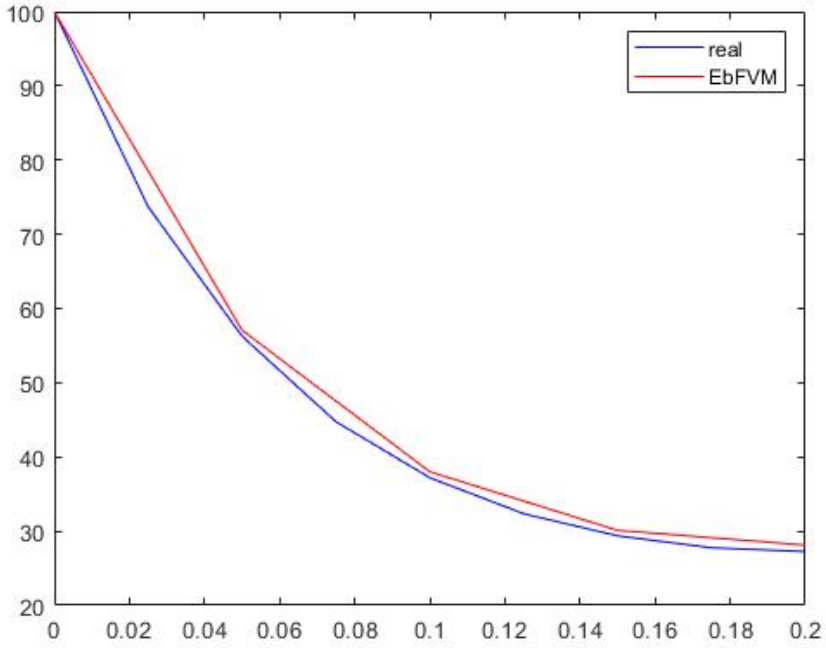

Fig.(15) Comparison between real and approximate solution at steady state for 5 CVs ($\Delta x = 0.05m$), $\Delta t = 1s$. We are clearly far from convergence in space domain, since the numerical solution is not smooth and is different to the real one.

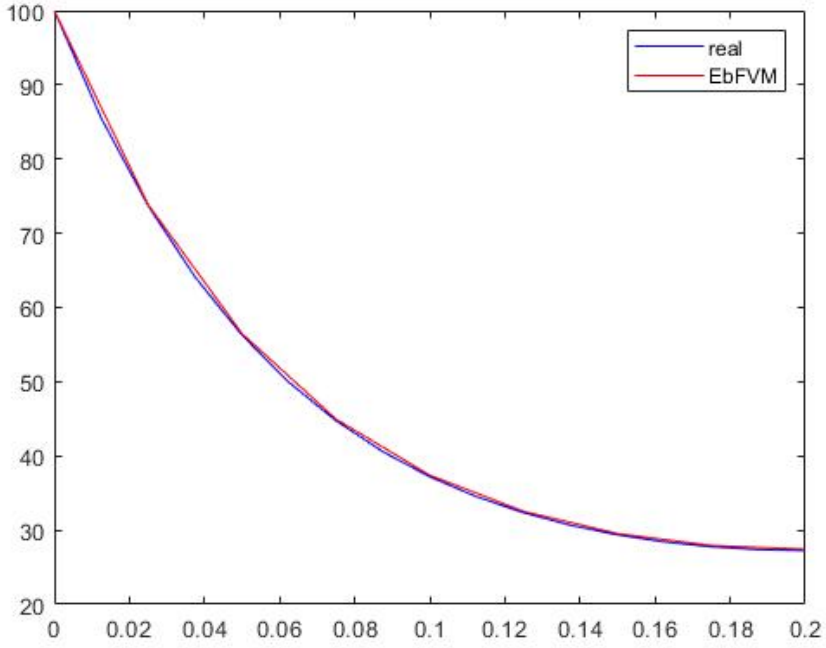

Fig.(16) Comparison between real and approximate solution at steady state for 9 CVs ($\Delta x = 0.025m$), $\Delta t = 1s$. Relative difference, with previous solution in norm on space domain is $\frac{norm(T_{new} - T_old)}{norm(T_{new})} = 0.9\%$, near but above the tolerance, therefore we keep iterate decreasing space step

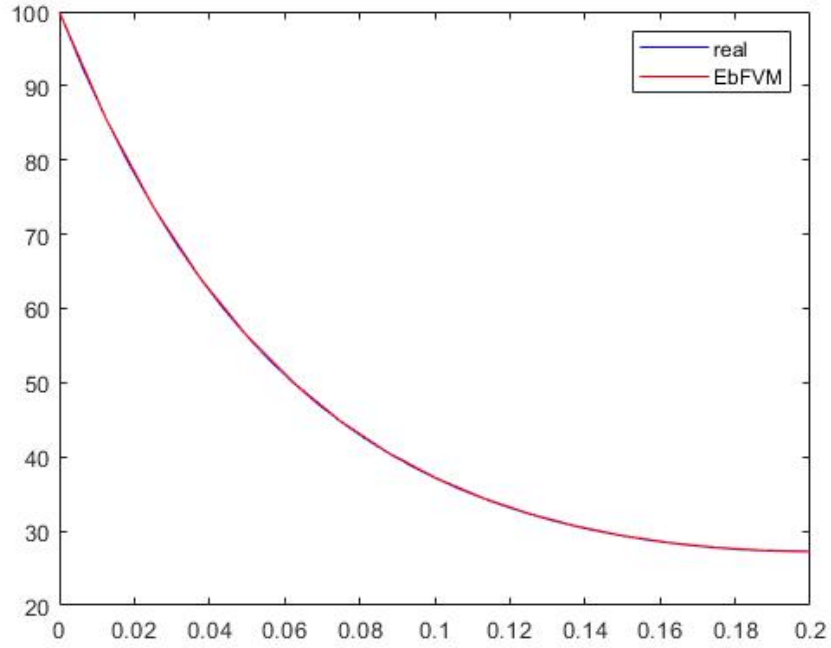

Fig.(17) Comparison between real and approximate solution at steady state for 17 CVs ($\Delta x = 0.0125m$), $\Delta t = 1s$. Relative difference with previous solution in norm on space domain is in fact lower than before 0.25%, below the tolerance, thus we stop and we now start to iterate on the time domain

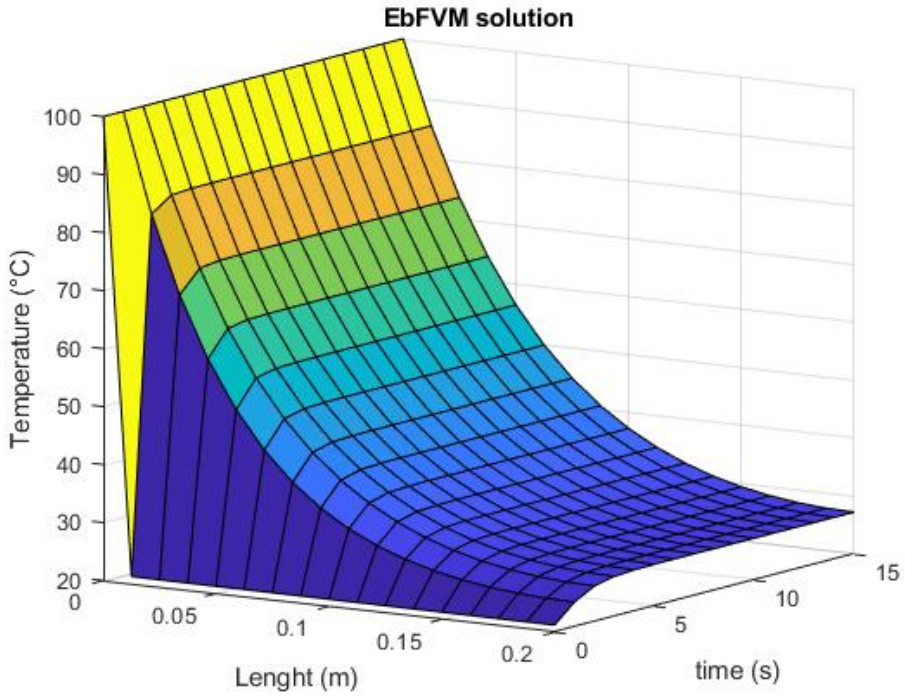

Fig.(18) EbFVM solution for 17 CVs ($\Delta x = 0.0125m$), $\Delta t = 1s$. Note how the solution is not perfectly smooth in time coordinate.

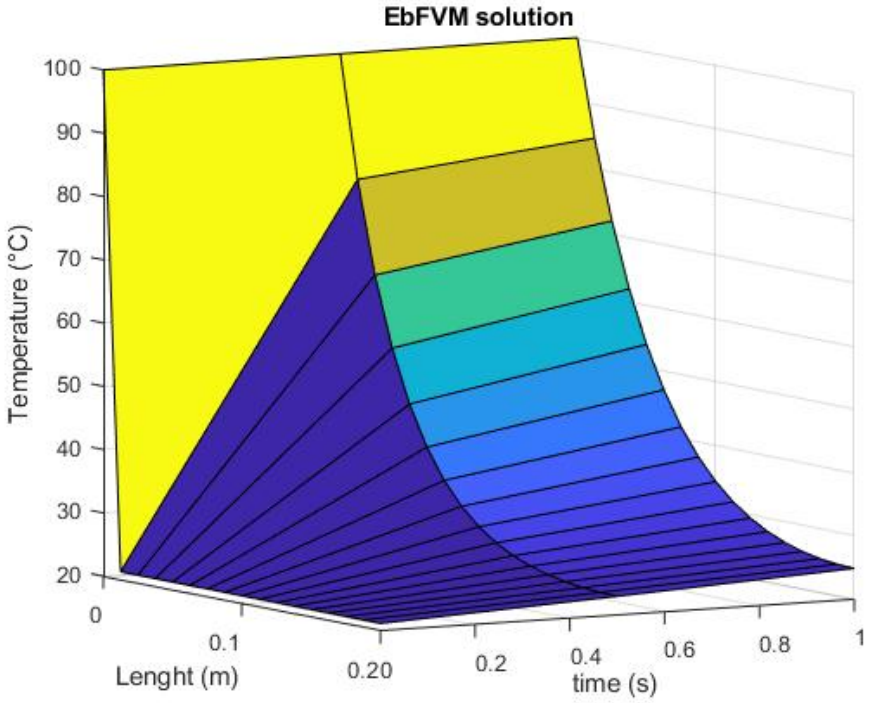

Fig.(19) EbFVM solution for 17 CVs and $\Delta t = \frac{1}{2}s$ for the first two time steps. Note how the solution is not smooth in time coordinate, we then keep iterate until the graph has the form that we want and there we will check the convergence.

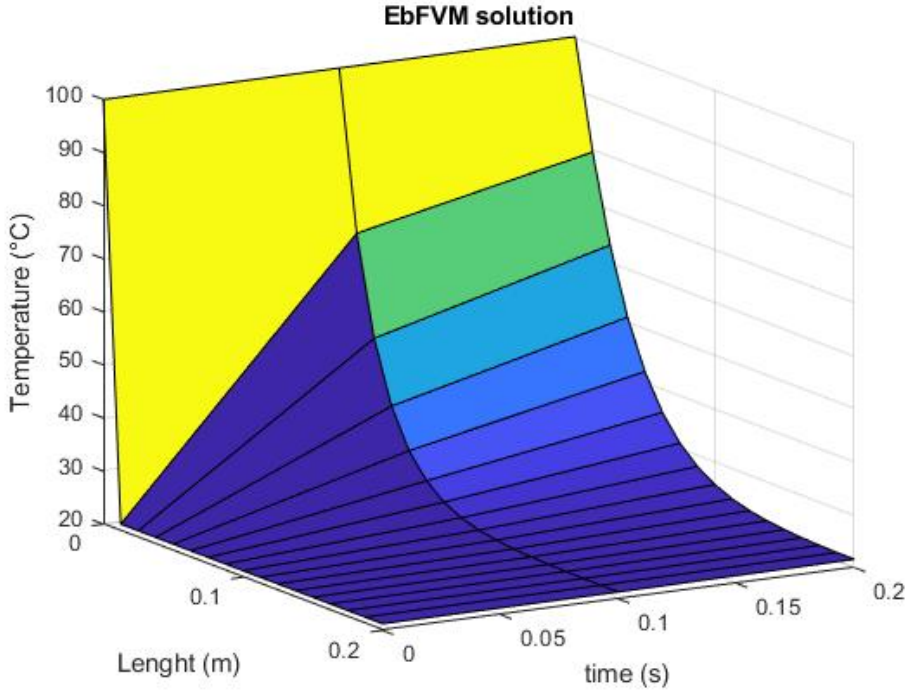

Fig.(20) EbFVM solution for 17 CVs and $\Delta t = \frac{1}{10}s$ for the first two time steps. The solution is still non smooth.

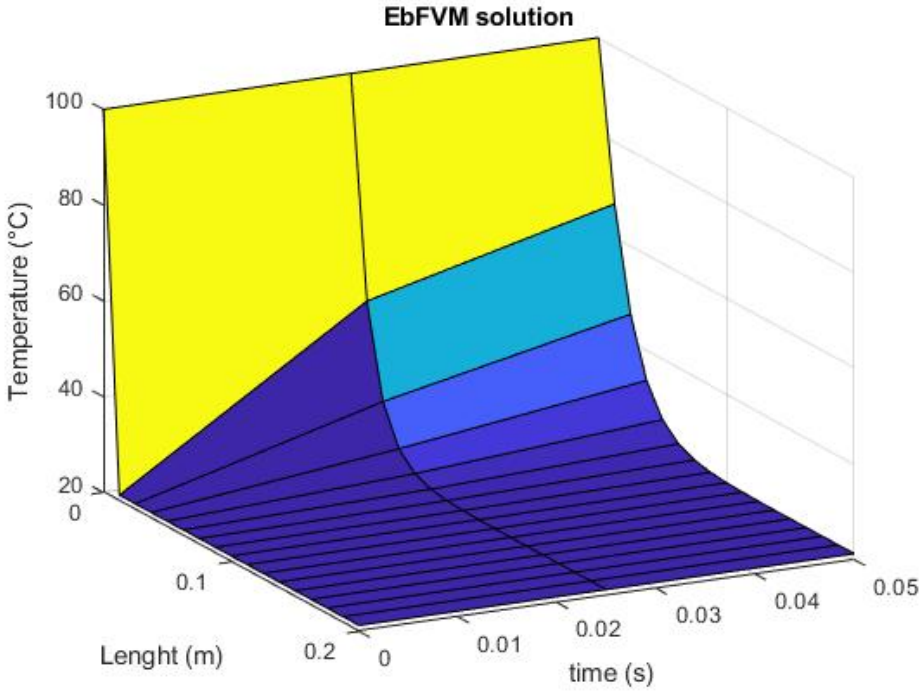

Fig.(21) EbFVM solution for 17 CVs and $\Delta t = \frac{1}{40}s$ for the first two time steps. The solution starts now to look smoother, but it's certainly not enough. We keep iterate.

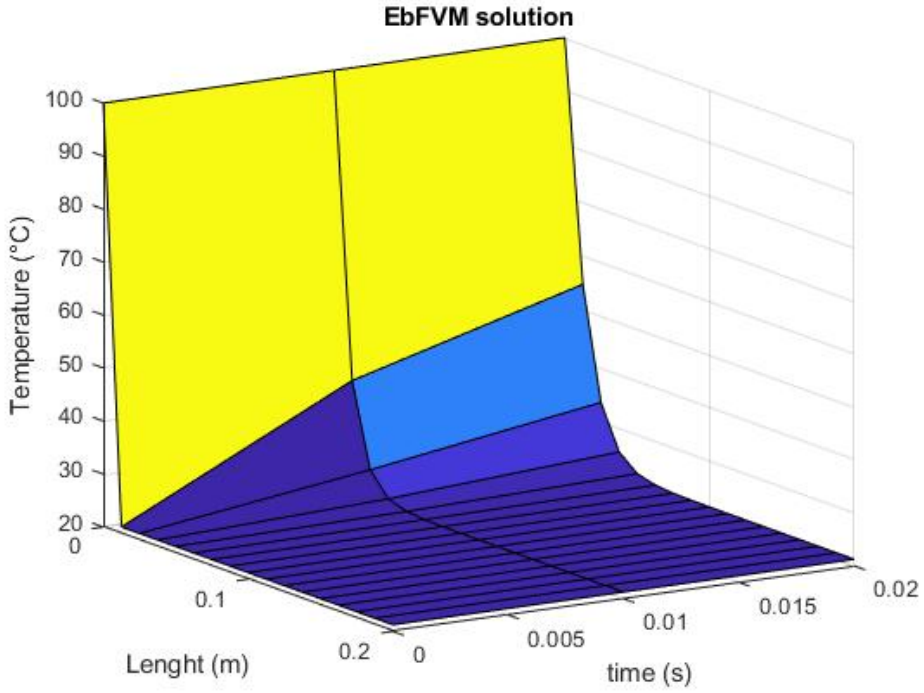

Fig.(22) EbFVM solution for 17 CVs and $\Delta t = \frac{1}{100}s$ for the first two time steps. Finally the solution seems sufficiently smooth, therefore we solve again the problem with half of this time step, but the same final time value, and check the difference between the values valuated at the same time coordinates.

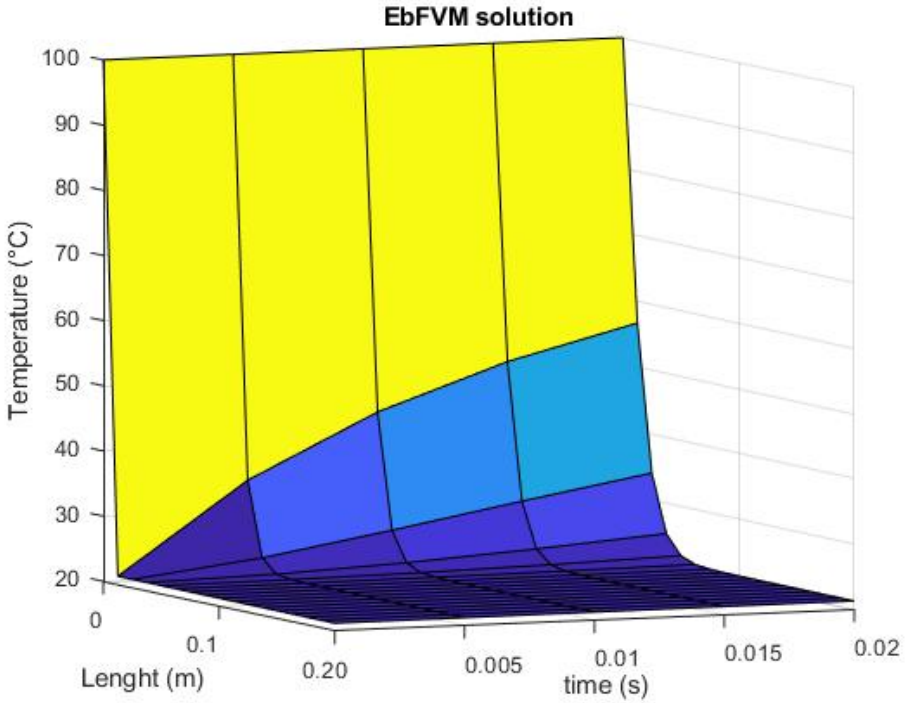

Fig.(23) EbFVM solution for 17 CVs and $\Delta t = \frac{1}{200}s$ *for the first four time steps. The solution is very similar to the previous one, and the difference in norm of the matrices containing the temperature values is 0.41%. We then consider this as the time step that we wanted*

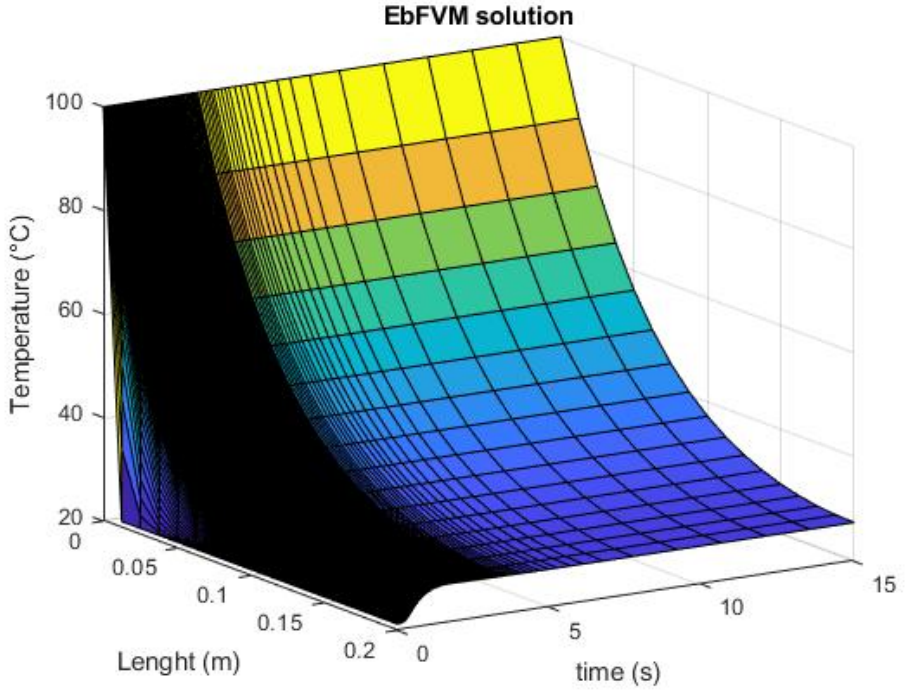

Fig.(24) Final EbFVM solution, 17 CVs and initial $\Delta t = \frac{1}{200}s$.

we want to find solving this problem. Another remarkable thing is that the time step adopted for the first iterations in time is way smaller than the one required for the explicit solution, and would have

been convenient use that formulation to achieve the solution with less CPU time (no linear system to be solved). However, since we used different time steps as the solution grew in time, up to a time step of 1 second, way bigger than the one required for explicit formulation, it's not so clear what would have been the best choice, but one could implement the solution with the explicit method (list of the code used is in the last chapter), and confront the two implementations. In this case the CPU time (tic, toc commands in matlab language) it's approximately 1 seconds, but is variable due to conditions of the CPU processor.

```
while diff>toll && t(end)<tmax
    t=[t,t(end)+dt];
    if t(end)>0.1*l && norm(T(K,:)-T(K-1,:))<1e-1 && dt<1 %CHECK OF TEMPERATURE ...
        VARIATION IN TIME (GRADIENT), AND FIX MAX TIME STEP TO 1
        dt=dt*1.5; %INCREASE TIME STEP
        Ae=-alpha*Area*theta/dx*dN2;        Aw=alpha*Area*theta/dx*dN1;
        Ae_o=alpha*Area*(1-theta)/dx*dN2; Aw_o=-alpha*Area*(1-theta)/dx*dN1;
        Ap=Area/dt-Ae-Aw+b*theta;    Ap_o=Area/dt-(1-theta)*b-Ae_o-Aw_o;
        A=diag(ones(n,1)*Ap)+diag(ones(n-1,1)*Aw,-1)+diag(ones(n-1,1)*Ae,1); A(1,2)=0;
        A(1,1)=1; A(n,n-1)=Aw*2; A(n,n)=Area/dt-2*Aw+b*theta; l=l+5;
    end
    K=K+1; %UPDATING NUMBER OF ITERATIONS
    for i=2:n-1
        B(i)=c+Ap_o*T(K-1,i)+Ae_o*T(K-1,i+1)+Aw_o*T(K-1,i-1);
    end
    B(1)=T_0; B(n)=c+(Ap_o+Ae_o-Aw_o)*T(K-1,i)+2*Aw_o*T(K-1,i-1);
    diff=norm(T(K-1,:)-T(K,:)); %CONTROL OF CONVERGENCE TO STEADY STATE BY THE WASTE
end
```

X COORDINATE																	TIME COORDINATE(S)
100	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	0
100	70.84419	48.93522	35.0839	27.52372	23.86759	22.26686	21.62151	21.37872	21.29259	21.26353	21.25415	21.25123	21.25036	21.2501	21.25003	21.25002	0.05
100	77.90653	59.76394	45.90105	36.0955	29.67294	25.76396	23.5424	22.35711	21.76027	21.47522	21.34551	21.28904	21.26546	21.25606	21.25268	21.25219	0.1
100	80.69768	64.63599	51.75236	41.85838	34.61332	29.56337	26.21338	24.09636	22.81981	22.08387	21.67739	21.46195	21.35247	21.29984	21.27784	21.27462	0.15
100	82.18316	67.34615	55.2448	45.63254	38.22877	32.71524	28.75273	26.00676	24.17228	22.99053	22.25627	21.81653	21.56391	21.42778	21.36562	21.35655	0.2
100	83.08978	69.03765	57.50574	48.20045	40.84515	35.16697	30.89556	27.76952	25.5462	24.01076	22.98222	22.31579	21.90173	21.6621	21.54679	21.52996	0.25
100	83.68886	70.17066	59.05443	50.01444	42.76647	37.05259	32.63356	29.287	26.8092	25.01834	23.75765	22.89721	22.33456	21.99385	21.82451	21.79978	0.3
100	84.10591	70.96664	60.15903	51.33569	44.20368	38.50912	34.02723	30.55677	27.91781	25.95131	24.51982	23.50814	22.8238	22.39706	22.18056	22.14895	0.35
100	84.40719	71.54548	60.97116	52.32209	45.29785	39.64469	35.14466	31.60834	28.87042	26.78727	25.23579	24.11305	23.33615	22.84222	22.58828	22.55119	0.4
100	84.63099	71.97763	61.58262	53.07357	46.14419	40.53956	36.04494	32.47772	29.68178	27.52387	25.89128	24.69058	23.84689	23.30349	23.02164	22.98048	0.45
100	84.80094	72.30715	62.05203	53.65599	46.80827	41.25247	36.77535	33.19831	30.37113	28.16757	26.48234	25.22911	24.33925	23.76108	23.45943	23.41538	0.5
100	84.93237	72.56284	62.41835	54.11416	47.33613	41.82649	37.37262	33.79833	30.95725	28.72794	27.01033	25.72327	24.80285	24.20127	23.88614	23.84013	0.55
100	85.03561	72.76427	62.70834	54.47938	47.7607	42.29335	37.86492	34.30067	31.45678	29.21506	27.47915	26.17161	25.23198	24.61536	24.29148	24.24419	0.6
100	85.11781	72.92507	62.94084	54.77398	48.10589	42.67664	38.2738	34.72356	31.88374	29.63841	27.89375	26.575	25.62419	24.99852	24.6693	24.62122	0.65
100	85.18406	73.05496	63.12938	55.01416	48.38928	42.99402	38.61584	35.08145	32.24979	30.00643	28.25937	26.93571	25.97924	25.34869	25.01649	24.96798	0.7
100	85.23804	73.16101	63.28383	55.21187	48.62401	43.25889	38.90381	35.3858	32.56451	30.32655	28.58115	27.25671	26.29829	25.66568	25.33213	25.28342	0.75
100	85.28245	73.24842	63.41153	55.37603	48.81996	43.48146	39.14765	35.64573	32.8358	30.60517	28.86391	27.54133	26.58336	25.95056	25.61672	25.56797	0.8
100	85.31931	73.32109	63.51796	55.51336	48.98467	43.66962	39.35516	35.86855	33.07018	30.84782	29.1121	27.79296	26.83693	26.20511	25.87168	25.82299	0.85
100	85.35013	73.38194	63.60731	55.62903	49.12398	43.82954	39.53252	36.06017	33.27306	31.05924	29.32974	28.01491	27.0617	26.43156	26.09895	26.05038	0.9
100	85.37609	73.43324	63.68279	55.72702	49.24242	43.96608	39.68467	36.22541	33.44896	31.24356	29.52048	28.21034	27.26038	26.6323	26.30074	26.25233	0.95
100	85.39807	73.47674	63.7469	55.81045	49.34357	44.08311	39.8156	36.36823	33.60168	31.40431	29.68753	28.38215	27.43561	26.80974	26.47935	26.4311	1
100	85.41678	73.5138	63.80161	55.8818	49.43029	44.18375	39.92858	36.49191	33.73443	31.54455	29.83377	28.53303	27.58987	26.96625	26.63704	26.58896	1.05
100	85.43278	73.54551	63.84848	55.94303	49.50487	44.27053	40.02627	36.59918	33.84992	31.66693	29.96175	28.6654	27.72549	27.10404	26.77598	26.72808	1.1
100	85.44651	73.57274	63.88877	55.99574	49.5692	44.34554	40.11091	36.69235	33.95048	31.77375	30.07372	28.78145	27.84457	27.22518	26.89822	26.85047	1.15
100	85.45832	73.59619	63.9235	56.04124	49.62481	44.41049	40.18435	36.77335	34.0381	31.86701	30.17166	28.88311	27.94904	27.33155	27.00561	26.95802	1.2
100	85.46852	73.61644	63.95352	56.08059	49.67298	44.46683	40.24815	36.84385	34.11448	31.94844	30.2573	28.97214	28.04063	27.42488	27.09987	27.05241	1.25
100	85.47734	73.63396	63.9795	56.1147	49.71476	44.51576	40.30364	36.90524	34.18108	32.01955	30.33219	29.05007	28.12086	27.50669	27.18253	27.1352	1.3
100	85.48498	73.64915	64.00203	56.14429	49.75105	44.5583	40.35193	36.95873	34.23919	32.08165	30.39766	29.11826	28.19112	27.57836	27.25497	27.20775	1.35
100	85.49162	73.66233	64.0216	56.17001	49.7826	44.59532	40.39399	37.00537	34.2899	32.1359	30.45489	29.17791	28.25262	27.64113	27.31842	27.27129	1.4
100	85.49738	73.67379	64.03861	56.19238	49.81006	44.62756	40.43066	37.04605	34.33416	32.18328	30.50492	29.23008	28.30643	27.69606	27.37397	27.32693	1.45
100	85.50239	73.68375	64.05342	56.21185	49.83398	44.65566	40.46262	37.08154	34.3728	32.22468	30.54864	29.2757	28.3535	27.74414	27.42258	27.37562	1.5
100	85.50675	73.69243	64.06631	56.22881	49.85482	44.68015	40.4905	37.11251	34.40654	32.26084	30.58685	29.31559	28.39467	27.78619	27.46511	27.41822	1.55
100	85.51055	73.69999	64.07754	56.24359	49.87299	44.70152	40.51483	37.13955	34.43601	32.29243	30.62025	29.35046	28.43067	27.82297	27.50231	27.45548	1.6
100	85.51491	73.70866	64.09043	56.26055	49.89385	44.72604	40.54276	37.17058	34.46977	32.32847	30.65795	29.38888	28.46829	27.85732	27.52968	27.47095	1.665
100	85.51959	73.71795	64.10424	56.27872	49.91615	44.75219	40.57238	37.20318	34.50473	32.36497	30.69491	29.42489	28.50155	27.88549	27.54966	27.47872	1.75125

Fig.(25) Final EbFVM solution up to 1.75s.

4.3.2 Implementation in Matlab solver of the FEM

Also in this case the code is implemented assuming a uniform grid in space and using the implicit formulation.

```
% 1D steady state heat conduction problem
%{
L: length of the rod [m]
k: thermal conductivity of the rod [W/(mK)]
h:  heat transfer coefficient [W/(m^2K)]
T_0: base temperature [K]
T_inf: Temperature of the environment [K]
A: cross Area of the rod (constant) [m^2]
n: number of vertex
Cp: specific heat with constant pressure [kJ/(kgK)]
q:heat generation [W/m^3]
rho: density [kg/m^3]
%}
close all
clear
clear
L=0.2;
ex=1e-2;
k=30; h=20; q=1e4;
Cp=0.42; rho=8700;
T_0=100; T_inf=20;
T_zero=21.25;
Area=ex*ex;
alpha=k/rho/Cp;
P=4*ex;
n=15;
dx=L/(n-1);
theta=input('digit 0 for explicit method, 1 for fully implicit, theta= ');
tmax=18;
gamma=h*P/rho/Cp;
dt=Area/(6*alpha*Area/dx^2+gamma);
if theta==1
    dt=1/2;
end
%creating element's matrices
T=ones(1,n)*T_zero; T(1)=T_0;
K(1,1)=Area/3/dt+theta*(alpha*Area/dx^2+gamma/3); ...
    K(1,2)=Area/6/dt+theta*(-alpha*Area/dx^2+gamma/6);
K(2,1)=K(1,2); K(2,2)=K(1,1);

K_O(1,1)=Area/3/dt-(1-theta)*(alpha*Area/dx^2+gamma/3); ...
    K_O(1,2)=Area/6/dt-(1-theta)*(-alpha*Area/dx^2+gamma/6);
K_O(2,1)=K_O(1,2); K_O(2,2)=K_O(1,1);

%ASSEMBLING
itmax=1e4;
A=zeros(n);
toll=1e-6;
diff=toll+1;
it=1;pp=1;
b=zeros(n,2);
c=0.5*(gamma*T_inf+q*Area/rho/Cp); t=0;
tic;
while diff>toll && t(end)<tmax && it<itmax
    t=[t,t(end)+dt];
%     if t(end)>0.5*pp && norm(T(p,:)-T(p-1,:))<1e-1 && dt<1 %CHECK OF TEMPERATURE ...
    VARIATION IN TIME (GRADIENT), AND FIX MAX TIME STEP TO 1
%         dt=dt*1.5; %INCREASE TIME STEP
%         K(1,1)=Area/3/dt+theta*(alpha*Area/dx^2+gamma/3); ...
    K(1,2)=k*Area/6/dt+theta*(-alpha*Area/dx^2+gamma/6);
%         K(2,1)=K(1,2); K(2,2)=K(1,1);
%
```

```
%         K_O(1,1)=Area/3/dt-(1-theta)*(alpha*Area/dx^2+gamma/3); ...
    K_O(1,2)=k*Area/6/dt-(1-theta)*(-alpha*Area/dx^2+gamma/6);
%         K_O(2,1)=K_O(1,2); K_O(2,2)=K_O(1,1);
%     end
    it=it+1; %UPDATE ITERATION NUMBER
    B=zeros(1,n); A=zeros(n,n);
    for l=1:n-1
    b(l,1)=c+K_O(1,1)*T(it-1,l)+K_O(1,2)*T(it-1,l+1);
    b(l,2)=c+K_O(1,2)*T(it-1,l)+K_O(1,1)*T(it-1,l+1);
    end
    for l=1:n-1
    A(l:l+1,l:l+1)=A(l:l+1,l:l+1)+K;
    B(l)=B(l)+b(l,1); B(l+1)=b(l,2);
    end
    A(1,1)=1; A(1,2:end)=0;
    B(1)=T_0;

    T(it,:)=(A\B')'; %SOLVE THE LINEAR SYSTEM
    diff=norm(T(it-1,:)-T(it,:)); %CONTROL OF CONVERGENCE TO STEADY STATE BY THE WASTE
end
toc;
%CHECK OF STEADY STATE CONDITION BY ENERGY BALANCE
E_in=-k*Area*(T(end,2)-T(end,1))/dx; %ENERGY ENTERING BY LEFT SURFACE
E_out=0;
for i=2:n
E_out=E_out+h*P*dx*((T(end,i)+T(end,i-1))/2-T_inf); %ENERGY LEAVING BODY BY CONVECTION
end
TOLL=abs(E_in+q*L*ex^2-E_out);
if TOLL>1
    warning('Error, the steady state is not verified by energy balance');
end
figure(1)

%PLOT OF THE SOLUTION IN A 3D GRAPH

x = linspace(0,L,n);
t = t;
[X,tempo]=meshgrid(x,t);
surf(X,tempo,T) % mesh
xlabel('Lenght (m)');
ylabel('time (s)');
zlabel('Temperature ( C )')
title('FEM solution')

% REAL SOLUTION (STATIONARY)
gamma=sqrt(P*h/k/Area); sol_part=T_inf+q./k/gamma^2; T_gamma=T_0-sol_part;
T_real=@(x) ...
    T_gamma./(exp(-gamma.*L)+exp(gamma.*L))*(exp(gamma.*(L-x))+exp(gamma.*(x-L)))+sol_part;
%COMPARISON BETWEEN REAL AND APPROXIMATE SOLUTION FOR STEADY STATE
XX=0:dx/2:L;
YY=T_real(XX);
figure(2)
plot(XX,YY,'b-');
hold on
plot(0:dx:L,T(end,:),'r-')
hold off
legend('real','FEM')
```

As for the EbFVM, it's not necessary to keep the initial time step for the resolution of the whole domain. Therefore we obtain the solution as in figure (34) Notice how also in this case the algorithm satisfy the tolerance for convergence to the steady state at about 12-13 seconds.

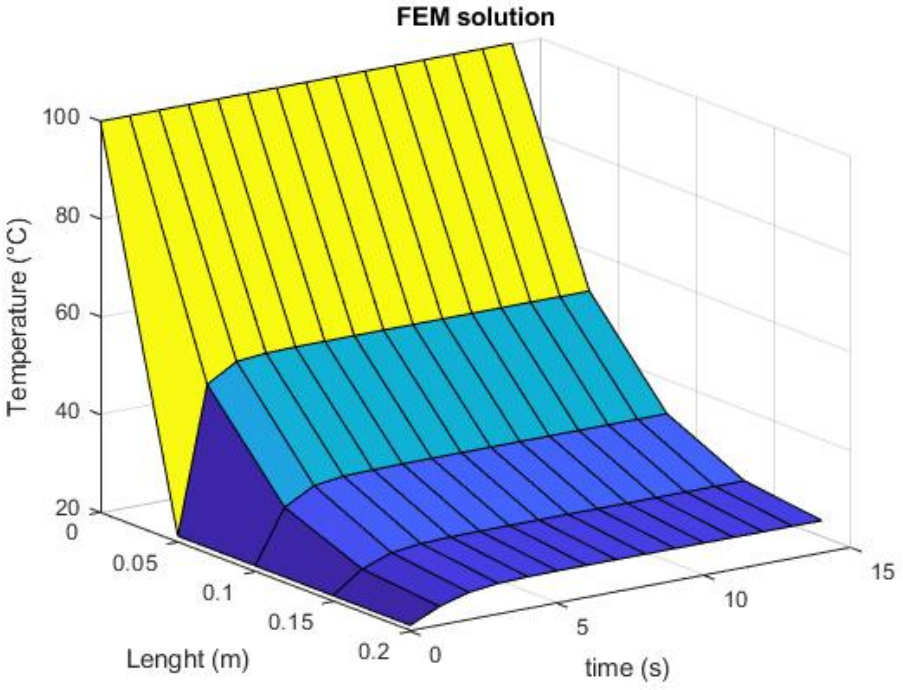

Fig.(26) First FEM trial solution, 4 Elements $\Delta t = 1$. See how non smooth is the solution, meaning the distance from convergence

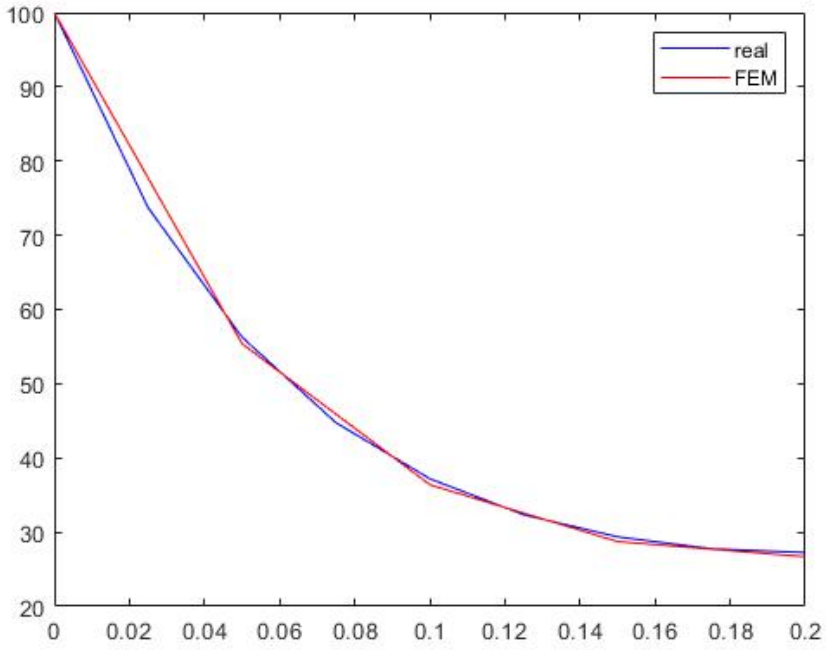

Fig.(27) FEM solution for steady state, 4 Elements $\Delta t = 1$. Solution is not so different from the analytical one even if the first approximation. Keep iterating for check convergence also by difference in successive solutions with grid refined

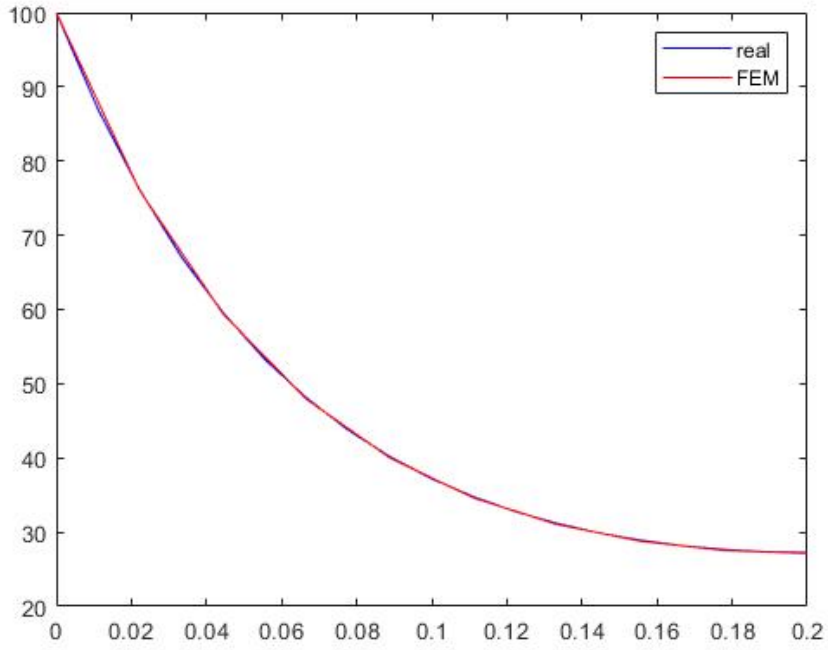

Fig.(28) FEM solution for steady state, 9 Elements $\Delta t = 1$. *Relative error with previous solution is* $\frac{norm(T_{new} - T_old)}{norm(T_{new})} = 4.69\%$, *then we keep iterate*

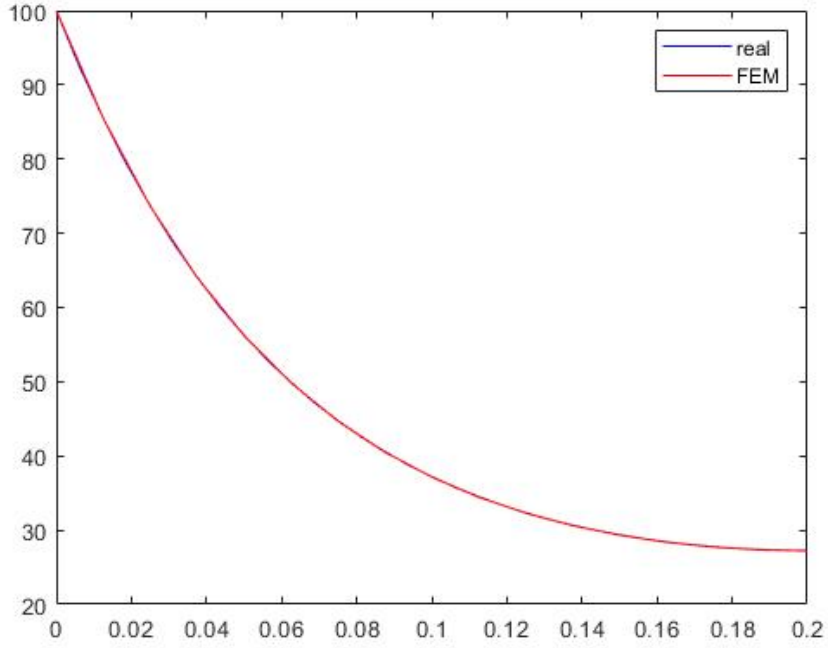

Fig.(29) FEM solution for steady state, 17 Elements $\Delta t = 1$. *Relative error with previous solution is 0.23%, then stop iterating in space and we start iterating in time.*

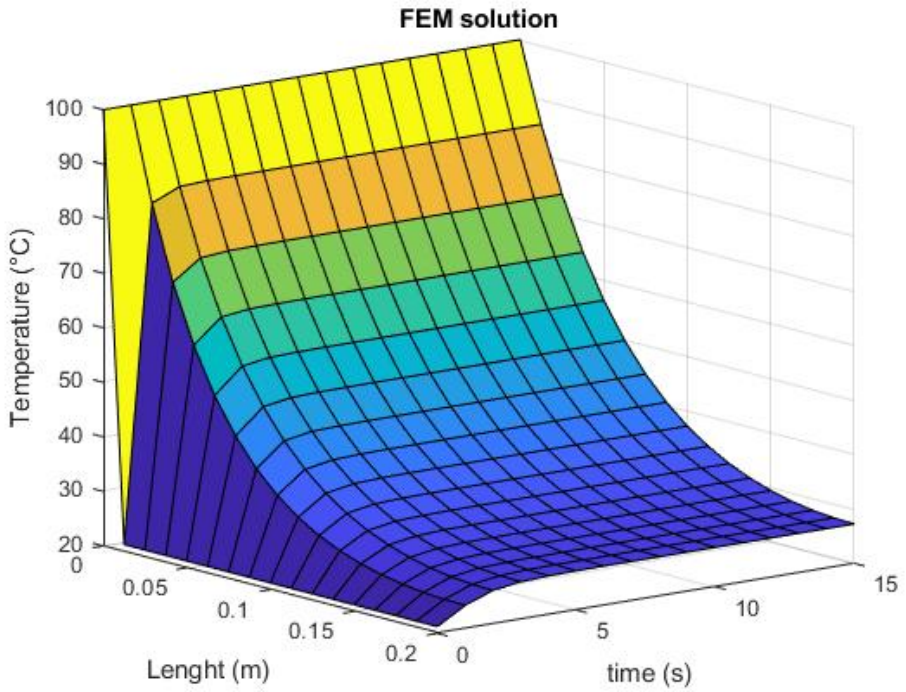

Fig.(30) FEM solution, 17 Elements $\Delta t = 1$. See how the solution seems smooth along "x" coordinate, but clearly not along the time one

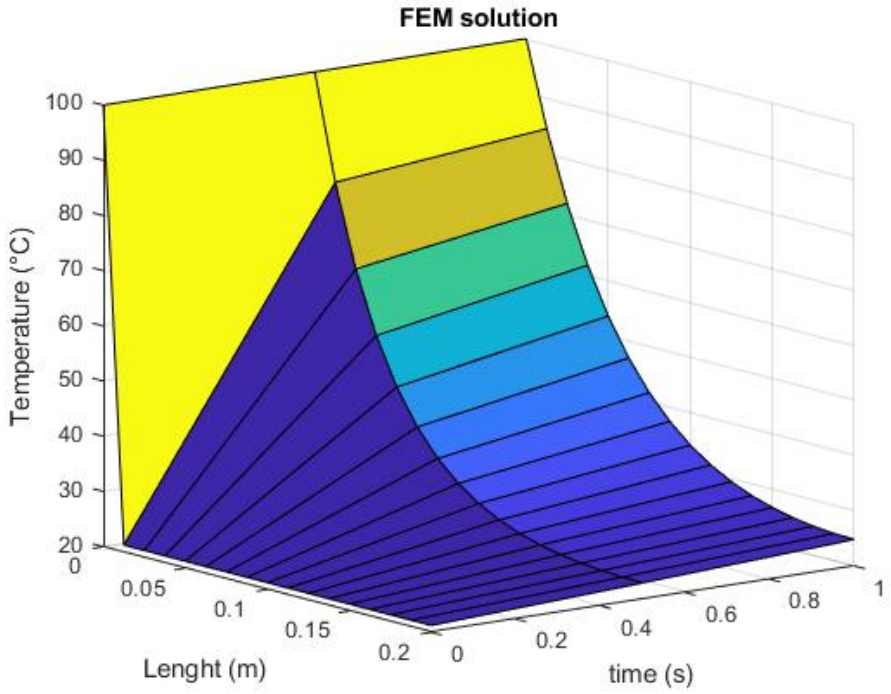

Fig.(31) FEM solution for the first two time steps, 17 Elements $\Delta t = 0.5$. Now it's easy to see the discontinuity of first species in time of the solution

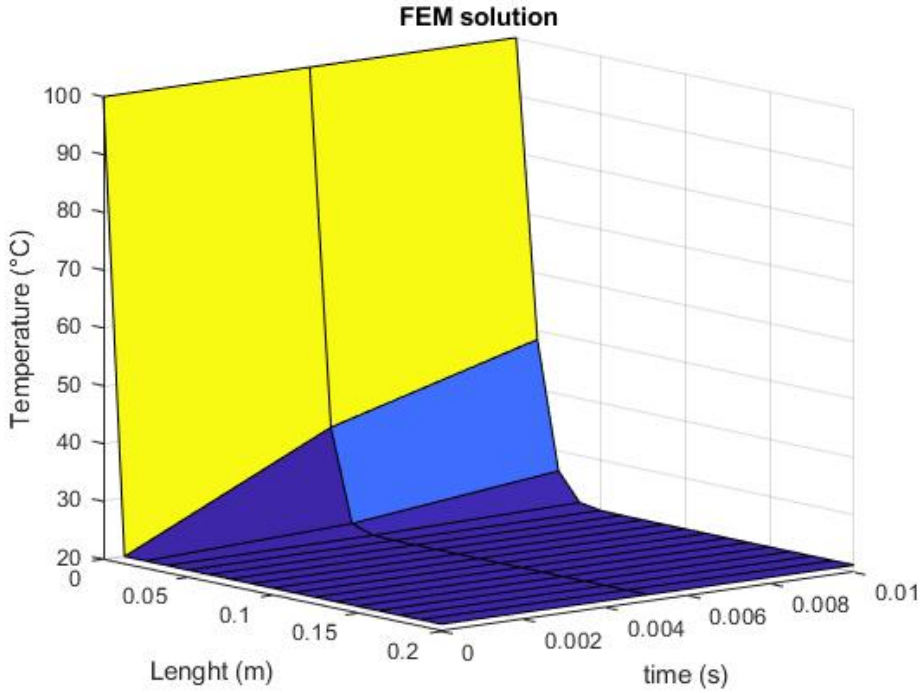

Fig.(32) FEM solution for the first two time steps, 17 Elements $\Delta t = \frac{1}{200}$. The solution seems now sufficiently smooth, so we check the convergence with next iteration

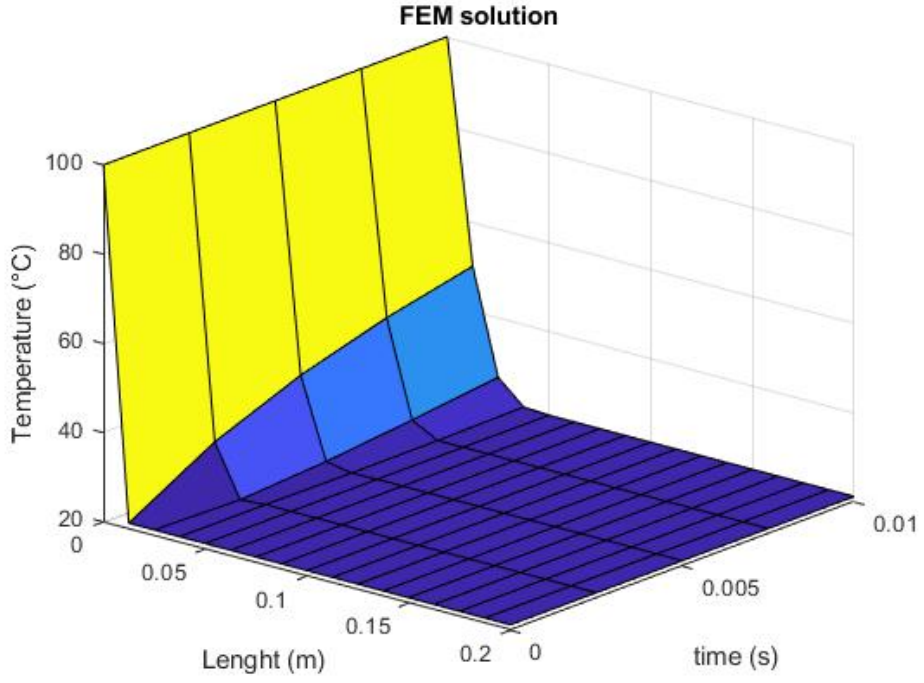

Fig.(33) FEM solution for the first four time steps, 17 Elements $\Delta t = \frac{1}{400}$. The relative difference between last two solutions is 0.25%, then we consider this time step acceptable

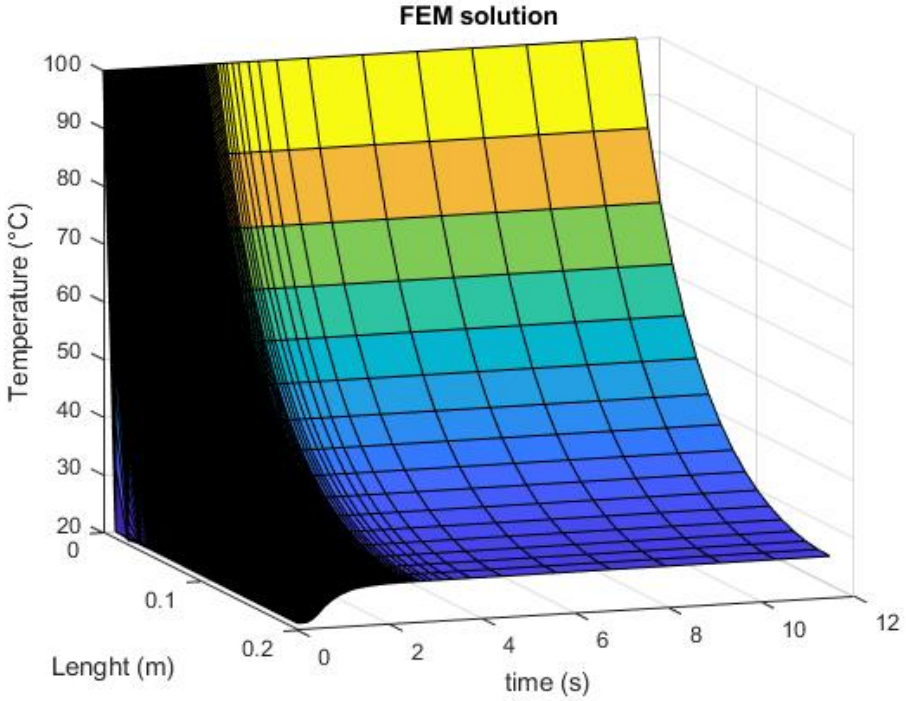

Fig.(34) FEM solution, 17 Elements and initial $\Delta t = \frac{1}{400}$.

X COORDINATE (m)																	TIME COORDINATE (s)
0	0.0125	0.025	0.0375	0.05	0.0625	0.075	0.0875	0.1	0.1125	0.125	0.1375	0.15	0.1625	0.175	0.1875	0.2	
100	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	0
100	76.08074	56.58051	42.10028	32.42189	26.64434	23.58125	22.14573	21.55381	21.34021	21.27321	21.2551	21.25094	21.25014	21.25002	21.25	21.25	0.075
100	80.89595	64.97441	52.12675	42.16718	34.79502	29.6083	26.15184	23.97576	22.68418	21.96267	21.58386	21.39721	21.31104	21.27396	21.25954	21.25585	0.15
100	82.75703	68.41629	56.65863	47.20318	39.77767	34.10462	29.9012	26.88825	24.80327	23.41267	22.52012	21.96987	21.64543	21.46512	21.37625	21.34971	0.225
100	83.70876	70.21754	59.12314	50.09244	42.83887	37.10645	32.66062	29.28457	26.77945	24.96652	23.68977	22.81794	22.24544	21.89203	21.70184	21.64205	0.3
100	84.26327	71.2802	60.60766	51.88419	44.80957	39.12926	34.62328	31.09894	28.38646	26.33682	24.82091	23.7294	22.97287	22.48189	22.20703	22.11869	0.375
100	84.6121	71.95402	61.5615	53.05701	46.13083	40.52604	36.02712	32.45145	29.64325	27.46954	25.81752	24.59298	23.71945	23.13769	22.80549	22.69757	0.45
100	84.84315	72.40283	62.20284	53.85615	47.04678	41.51531	37.04742	33.46498	30.61935	28.38641	26.66307	25.36474	24.42374	23.78821	23.42147	23.30165	0.525
100	85.00215	72.71306	62.64943	54.41842	47.70004	42.2329	37.80279	34.23369	31.38074	29.12483	27.36857	26.03328	25.05686	24.39228	24.00655	23.88014	0.6
100	85.11501	72.93405	62.96948	54.82486	48.17759	42.76483	38.37218	34.8246	31.97927	29.71991	27.95247	26.60182	25.60928	24.9308	24.53576	24.40607	0.675
100	85.19721	73.09552	63.20456	55.12558	48.53427	43.16677	38.8084	35.28457	32.45353	30.20059	28.43364	27.07965	26.08197	25.39838	24.99966	24.86865	0.75
100	85.25845	73.21611	63.38089	55.35254	48.80561	43.47551	39.14728	35.64649	32.83194	30.58983	28.82916	27.47808	26.48116	25.79725	25.39798	25.26672	0.825
100	85.30493	73.30785	63.51553	55.52675	49.01525	43.71594	39.41359	35.9338	33.13562	30.90574	29.15373	27.80847	26.8152	26.1334	25.73518	25.60423	0.9
100	85.34078	73.37874	63.61989	55.66233	49.17929	43.90526	39.62481	36.16348	33.38043	31.16257	29.41981	28.08139	27.09294	26.41429	26.01784	25.88746	0.975
100	85.36879	73.43422	63.70176	55.76906	49.30896	44.05566	39.79356	36.3481	33.57847	31.37166	29.63776	28.30619	27.3228	26.6476	26.25316	26.12343	1.05
100	85.39091	73.47807	63.7666	55.85382	49.41228	44.17598	39.92914	36.49713	33.7391	31.54207	29.81619	28.49099	27.51241	26.84056	26.44809	26.31902	1.125
100	85.40852	73.51302	63.81836	55.92162	49.49514	44.27275	40.03855	36.61781	33.86964	31.68106	29.96222	28.64268	27.66843	26.99964	26.60899	26.48052	1.2
100	85.42263	73.54104	63.85991	55.97613	49.56189	44.35089	40.12712	36.71577	33.9759	31.79448	30.08168	28.76705	27.7966	27.1305	26.74145	26.61352	1.275
100	85.43475	73.56512	63.89565	56.02307	49.61946	44.41841	40.20379	36.80074	34.06826	31.89328	30.18593	28.87577	27.90878	27.24516	26.8576	26.73015	1.35375
100	85.4494	73.59425	63.93891	56.07997	49.68935	44.5005	40.29719	36.90445	34.18119	32.01431	30.31387	29.00939	28.04684	27.3864	27.00075	26.87395	1.485
100	85.46209	73.6195	63.97644	56.12938	49.75011	44.57197	40.37864	36.99504	34.28002	32.12039	30.42618	29.12684	28.16834	27.5108	27.12689	27.00067	1.654531
100	85.46252	73.62036	63.97773	56.13107	49.7522	44.57443	40.38144	36.99816	34.28342	32.12404	30.43005	29.13089	28.17253	27.51509	27.13125	27.00505	1.662187
100	85.46295	73.62121	63.97899	56.13273	49.75424	44.57683	40.38418	37.00121	34.28675	32.12762	30.43384	29.13486	28.17663	27.51929	27.13551	27.00933	1.669844
100	85.46337	73.62204	63.98022	56.13436	49.75624	44.57919	40.38686	37.0042	34.29002	32.13113	30.43755	29.13875	28.18066	27.52341	27.13969	27.01353	1.6775
100	85.46377	73.62286	63.98143	56.13595	49.7582	44.58149	40.38949	37.00713	34.29322	32.13456	30.44119	29.14256	28.1846	27.52745	27.14378	27.01764	1.685156
100	85.46418	73.62365	63.98261	56.13751	49.76012	44.58375	40.39207	37.01	34.29635	32.13793	30.44476	29.14629	28.18846	27.53141	27.1478	27.02168	1.692812
100	85.46457	73.62443	63.98377	56.13904	49.762	44.58597	40.3946	37.01281	34.29942	32.14123	30.44825	29.14994	28.19224	27.53528	27.15173	27.02563	1.700469
100	85.46495	73.6252	63.98491	56.14054	49.76384	44.58814	40.39707	37.01556	34.30242	32.14446	30.45167	29.15353	28.19595	27.53908	27.15558	27.0295	1.708125
100	85.46533	73.62594	63.98602	56.142	49.76565	44.59026	40.39949	37.01826	34.30537	32.14762	30.45503	29.15704	28.19959	27.5428	27.15936	27.03329	1.715781
100	85.4657	73.62668	63.98711	56.14344	49.76741	44.59234	40.40187	37.0209	34.30826	32.15072	30.45831	29.16048	28.20314	27.54645	27.16306	27.03701	1.723437
100	85.46606	73.6274	63.98818	56.14485	49.76915	44.59438	40.40419	37.02349	34.31108	32.15376	30.46153	29.16385	28.20663	27.55002	27.16668	27.04065	1.731094
100	85.46641	73.6281	63.98923	56.14622	49.77084	44.59638	40.40647	37.02603	34.31385	32.15674	30.46469	29.16715	28.21005	27.55352	27.17023	27.04422	1.73875

Fig.(35) FEM solution up to 1.75 s

4.4 Analytical solution for the whole domain

In this particular problem is also possible to find an analytical *weak* solution by using tools like Fourier series. The original aim is to find a solution T(x,t) that satisfies the system

$$\left.\begin{aligned} \frac{\partial^2 T}{\partial x^2} - \gamma^2(T(x,t) - T_\infty) + \frac{q'}{k} = \frac{1}{\alpha}\frac{\partial T(x,t)}{\partial t} \quad \text{for } x \in (0,L),\ t > 0. \\ T(0,t) = T_0;\ \frac{\partial T(L,0)}{\partial x} = 0 \quad t > 0 \\ T(x,0) = T^* \quad x \in (0,L) \end{aligned}\right\} \tag{55}$$

The first step consists in split the problem in pieces, working to find a solution in the form T(x,t)=u(x,t)+z(x), where z(x) is a particular solution for the non homogeneous Cauchy problem

$$\left.\begin{aligned} \frac{\partial^2 T(x))}{\partial x^2} - \gamma^2 T(x) = -\gamma^2 T_\infty + \frac{q'}{k} \quad \text{for } x \in (0,L), \\ T(0) = T_0;\ \frac{\partial T(0)}{\partial x} = 0 \end{aligned}\right\} \tag{56}$$

and u is a solution for the homogeneous one, with Boundary Conditions redefined as

$$\left.\begin{aligned} \frac{\partial^2 partial T}{\partial x^2} - \gamma^2 T(x,t) = \frac{1}{\alpha}\frac{\partial T(x,t)}{\partial t} \quad \text{for } x \in (0,L),\ t > 0. \\ T(0,t) = 0;\ \frac{\partial T(L,0)}{\partial x} = 0 \quad t > 0 \\ T(x,0) = T^* - z(x) \quad x \in (0,L) \end{aligned}\right\} \tag{57}$$

We can immediately see that z(x) is exactly the solution computed in (54) for the Steady State. Without entering too much into the details of the resolution, that is explained properly in [1], we are now going to try to compute the solution u. The main idea is to initially search for a solution in the form u(x,t)=T(t)X(x), as a product of two functions, one depending only on the time coordinate, and the other only on the space coordinate. Now the system (57) is reduced to

$$\left.\begin{aligned} X(x)''T(t) - \gamma^2 X(x)T(t) = \frac{1}{\alpha}X(x)T'(t) \quad \text{for } x \in (0,L),\ t > 0. \\ X(0)T(t) = 0;\ X(L)'T(t) = 0 \quad t > 0 \\ X(x)T(0) = T^* - z(x) \quad x \in (0,L) \end{aligned}\right\} \tag{58}$$

now, assuming that the solution is not trivial, so that X(x)and T(t) are different from 0 in the domain, we can divide all the PDE by X(x)T(t). Doing this passage we obtain

$$\frac{X(x)''}{X(x)} - \gamma^2 = \frac{1}{\alpha}\frac{T'(t)}{T(t)}, \tag{59}$$

which practically tells us that a function depending only on x (first term), minus a constant, it's equal to a function depending only on time, for every x and t in the domain! But clearly this is possible only if both the first and last terms are constants, such that

$$\frac{X''(x)}{X(x)} = C; \frac{1}{\alpha}\frac{T'(t)}{T(t)} = C - \gamma^2 \implies \frac{T'(t)}{T(t)} = \alpha(C - \gamma^2). \tag{60}$$

We have then obtained two ordinary differential equations, that we can solve forcing the boundary conditions. Assuming that the solution is not trivial, the BC are reduced to

$$\left.\begin{aligned} X(0) = 0;\ X(L)' = 0 \\ T(0) = T^* - z(x) \end{aligned}\right\} \tag{61}$$

In order to find a solution using Fourier analysis, it's useful to have the same type of boundary condition at the extremes of the domain. One possibility in this case consists in extending in some way the solution by parity in (0,2L), so that at the middle point the solution has derivative equal to 0. Hence the solution now has in the domain of interest BC as in (61), but we can search it by finding an extension in the domain (0,2L), which satisfies Dirichlet BC on both extremes. Next step to solve the two ODE computed, is to to split the cases for the constant C in C>0,C=0, C<0. One could prove by straightforward calculations that the only solution possible for the cases with $C \geq 0$ is the trivial one, $X(x) \equiv 0$. Instead, for C<0, applying the characteristic polynomial method $\lambda^2 = C$, we get a solution in the form

$$X(x) = c_1 cos(\sqrt{-C}x) + c_2 sin(\sqrt{-C}x), \tag{62}$$

with BC $X(0) = 0, X(2L) = 0$. Forcing the BC we obtain $c_1 = 0$ from the first and

$$0 = X(2L) = c_2 sin(\sqrt{-C}L) \tag{63}$$

from the second. This condition is trivially verified if $c_2 = 0$, but this would lead to the trivial solution $X \equiv 0$, but also if $\sqrt{-C}2L = n\pi$, when the sine it's equal to 0, where n is any natural number, including 0. From this consideration we compute the value of C as a function of n

$$C_n = -(\frac{n\pi}{2L})^2, \tag{64}$$

and inserting this result in the other ODE for T(t) we derive also

$$\begin{aligned} T_n'(t) = \alpha(C_n - \gamma^2)T(t) \implies T(t) = e^{\alpha(C-\gamma^2)} \\ \implies u_n(x,t) = be^{\alpha(C_n-\gamma^2)} sin(\frac{n\pi}{2L}x), \end{aligned} \tag{65}$$

where b is equal to constant c_2. We understand that this is a solution to the PDE that respects the boundary condition, for every value of n natural. Now, by the known property that for an homogeneous PDE of this type the finite sum of solutions is still a solution for the PDE, we get

$$u(x,t) = \sum_{n=0}^{N} b_n e^{\alpha(C_n-\gamma^2)} sin(\frac{n\pi}{2L}x). \tag{66}$$

The final idea is to let N tends to infinite and considerate the result computed, but since with this procedure some conditions of continuity of the solution are relaxed, the solution is in this case called *weak solution*. The last condition which we need to force in u(x,t) to determine a unique solution is the initial condition. For t=0 we obtain

$$\sum_{n=0}^{\infty} b_n sin(\frac{n\pi}{2L}x) = T^* - z(x), \tag{67}$$

that is exactly a Fourier series for an odd function 4L-periodic. Therefore, considering an odd extension, 4L-periodic, of the initial solution. Formally, as said above, the odd extension has to be computed from an initial condition which has been extend by parity to satisfies Dirichlet conditions on both extremes and have a 0 derivative in the middle. Practically that first function is computed translating the solution of -L in x direction, extending it by symmetry, and moving back the result with a further translation of L to the x positive. The function that we are trying to approximate by Fourier series is than the odd extension of

$$\left.\begin{matrix} \phi(x) & x \in (0,L) \\ \phi(2L-x) & x \in (L,2L) \end{matrix}\right\} = f(x) \tag{68}$$

where $\phi(x)$ is $T^* - z(x)$. Hence the coefficients b_n are calculated as

$$b_n = \frac{2}{2L}\int_0^{2L} f(x) sin(\frac{n\pi}{2L}x)dx = [(-1)^n - 1][\frac{2n\pi(T^* - z(x))}{4L\gamma^2 + (n\pi)^2} + \frac{2}{n\pi}(\frac{q'}{k\gamma^2} + T_\infty - T^*)] \tag{69}$$

that means that are non zero only for n odd. The final solution is then T(x,t)=u(x,t)+z(x)

$$T(x,t) = [\sum_{n=0}^{\infty} b_n e^{\alpha(C-\gamma^2)t} sin(\frac{n\pi}{2L}x)] + z(x). \tag{70}$$

The result is shown in figure (36).

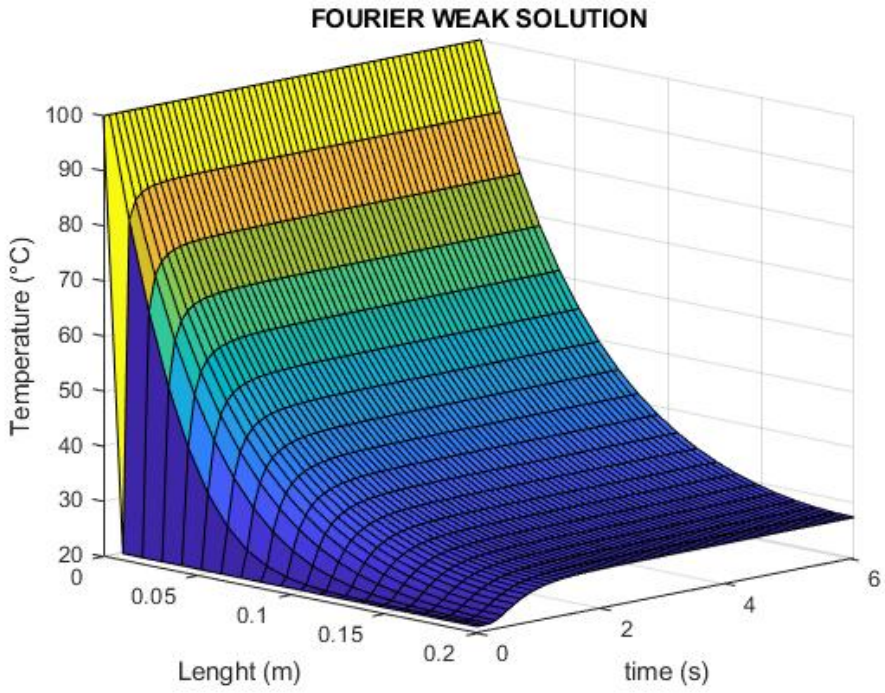

Fig.(36) Analytical solution with Fourier analysis

X COORDINATE																	TIME COORDINATE(S)
100	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	21.25	0
100	78.37124	60.47917	46.57591	36.52954	29.82416	25.70837	23.39174	22.19828	21.63625	21.39451	21.2996	21.2656	21.25449	21.25119	21.2503	21.25013	0.1
100	82.34827	67.64612	55.62064	46.0163	38.56273	32.9642	28.90693	26.07691	24.18054	22.96153	22.21075	21.76832	21.51963	21.38784	21.3255	21.30735	0.2
100	83.7692	70.32651	59.26815	50.2607	43.01801	37.28533	32.83004	29.43779	26.91238	25.07757	23.77952	22.88863	22.30047	21.93552	21.73829	21.67614	0.3
100	84.44902	71.63157	61.09654	52.4765	45.46812	39.81698	35.30634	31.74923	28.9833	26.86769	25.28134	24.12221	23.30695	22.7708	22.46757	22.36957	0.4
100	84.82125	72.35297	62.12348	53.74923	46.91676	41.36817	36.88973	33.30314	30.45893	28.23172	26.51689	25.22824	24.29658	23.66873	23.30702	23.18895	0.5
100	85.04237	72.78427	62.74404	54.53003	47.82316	42.3628	37.93549	34.36592	31.50998	29.24937	27.48747	26.14635	25.1646	24.49577	24.10731	23.97995	0.6
100	85.18164	73.05725	63.14004	55.03407	48.41715	43.02688	38.64948	35.11071	32.26878	30.00859	28.23743	26.88146	25.88327	25.19991	24.80158	24.67073	0.7
100	85.27351	73.23802	63.404	55.37316	48.82152	43.48558	39.15117	35.64437	32.82435	30.57748	28.81284	27.45863	26.45935	25.77381	25.37358	25.24201	0.8
100	85.33641	73.36218	63.58623	55.60896	49.10532	43.81111	39.5118	36.03353	33.23577	31.00555	29.25278	27.90654	26.91229	26.22967	25.83091	25.69978	0.9
100	85.38076	73.44993	63.71554	55.7772	49.30921	44.04692	39.7755	36.321	33.54299	31.32873	29.58845	28.25165	27.2642	26.58613	26.18996	26.05967	1
100	85.41273	73.5133	63.80921	55.89956	49.45826	44.22032	39.9707	36.53533	33.77375	31.57327	29.84424	28.51632	27.53555	26.86211	26.46866	26.33927	1.1
100	85.43616	73.55981	63.87809	55.9898	49.56857	44.34919	40.11645	36.69615	33.94777	31.7586	30.03902	28.71871	27.74377	27.07444	26.68344	26.55485	1.2
100	85.45353	73.59432	63.92928	56.057	49.65092	44.44567	40.22591	36.81734	34.07936	31.89921	30.18725	28.87315	27.90303	27.23713	26.84818	26.72028	1.3
100	85.46652	73.62013	63.96761	56.10738	49.71277	44.51828	40.30846	36.90893	34.17904	32.00596	30.30002	28.99087	28.0246	27.36146	26.97417	26.84681	1.4
100	85.47628	73.63954	63.99645	56.14533	49.75941	44.5731	40.37088	36.9783	34.25464	32.08705	30.38579	29.08051	28.11727	27.4563	27.07032	26.9434	1.5
100	85.48364	73.65419	64.01822	56.17401	49.79468	44.61459	40.41817	37.0309	34.31203	32.14865	30.45102	29.14874	28.18785	27.52857	27.1436	27.01703	1.6
100	85.48921	73.66528	64.03471	56.19571	49.82139	44.64604	40.45403	37.07082	34.35561	32.19547	30.50062	29.20065	28.24157	27.58359	27.19941	27.0731	1.7
100	85.49344	73.67368	64.0472	56.21217	49.84165	44.6699	40.48125	37.10114	34.38872	32.23106	30.53834	29.24013	28.28245	27.62547	27.24189	27.11578	1.8
100	85.49664	73.68005	64.05668	56.22466	49.85703	44.68801	40.50193	37.12417	34.41389	32.25811	30.56702	29.27016	28.31354	27.65732	27.27421	27.14825	1.9
100	85.49907	73.68488	64.06387	56.23415	49.86871	44.70178	40.51764	37.14167	34.43301	32.27868	30.58882	29.29299	28.33718	27.68156	27.29879	27.17296	2
100	85.50091	73.68856	64.06934	56.24136	49.87759	44.71223	40.52957	37.15497	34.44755	32.29431	30.6054	29.31036	28.35516	27.69998	27.31749	27.19174	2.1
100	85.50232	73.69135	64.07349	56.24683	49.88433	44.72018	40.53865	37.16508	34.4586	32.30619	30.618	29.32356	28.36884	27.71399	27.33171	27.20603	2.2
100	85.50338	73.69347	64.07665	56.25099	49.88946	44.72622	40.54554	37.17277	34.467	32.31523	30.62759	29.3336	28.37923	27.72465	27.34252	27.21689	2.3
100	85.50419	73.69508	64.07905	56.25415	49.89335	44.73081	40.55078	37.17861	34.47339	32.3221	30.63487	29.34123	28.38714	27.73275	27.35074	27.22515	2.4
100	85.50481	73.69631	64.08087	56.25656	49.89631	44.7343	40.55477	37.18305	34.47824	32.32732	30.64041	29.34703	28.39315	27.73891	27.35699	27.23143	2.5
100	85.50528	73.69724	64.08226	56.25839	49.89857	44.73695	40.5578	37.18643	34.48193	32.33129	30.64462	29.35144	28.39772	27.74359	27.36174	27.2362	2.6
100	85.50563	73.69795	64.08331	56.25978	49.90028	44.73897	40.5601	37.18899	34.48474	32.33431	30.64782	29.3548	28.40119	27.74715	27.36535	27.23983	2.7
100	85.5059	73.69849	64.08411	56.26083	49.90158	44.7405	40.56185	37.19095	34.48687	32.3366	30.65026	29.35735	28.40383	27.74986	27.3681	27.24259	2.8
100	85.50611	73.6989	64.08472	56.26163	49.90257	44.74167	40.56318	37.19243	34.4885	32.33835	30.65211	29.35928	28.40584	27.75192	27.37019	27.24469	2.9
100	85.50627	73.69921	64.08519	56.26225	49.90332	44.74256	40.5642	37.19356	34.48973	32.33968	30.65351	29.36076	28.40737	27.75348	27.37178	27.24629	3
100	85.50638	73.69945	64.08554	56.26271	49.90389	44.74323	40.56497	37.19442	34.49067	32.34068	30.65458	29.36188	28.40853	27.75467	27.37298	27.2475	3.1
100	85.50647	73.69963	64.08581	56.26306	49.90433	44.74374	40.56555	37.19507	34.49138	32.34145	30.6554	29.36273	28.40941	27.75558	27.3739	27.24842	3.2
100	85.50654	73.69976	64.08601	56.26333	49.90466	44.74413	40.566	37.19556	34.49192	32.34203	30.65602	29.36338	28.41008	27.75626	27.3746	27.24912	3.3

Fig.(37) Table of results of the analytical solutions

```
 %real solution
% 1D steady state heat conduction problem
%{
L: length of the rod [m]
k: thermal conductivity of the rod [W/(mK)]
h:  heat transfer coefficient [W/(m^2K)]
T_0: base temperature [K]
T_inf: Temperature of the environment [K]
A: cross Area of the rod (constant) [m^2]
n: number of control volumes
Cp: specific heat with constant pressure [kJ/(kgK)]
q:heat generation [W/m^3]
rho: density [kg/m^3]
%}
clear
close all
T_zero=21.25;
tmax=6;
L=0.2;
ex=1e-2;
k=30; h=20; q=1e4;
Cp=0.42; rho=8700;
T_0=100; T_inf=20;
Area=ex*ex;
alpha=k/rho/Cp;
P=4*ex;
n=17;
dx=L/(n-1);
% dt=Area/(2*alpha*Area/dx^2+h*P*dx/rho/Cp);
dt=0.1;
gamma=sqrt(P*h/k/Area); sol_part=T_inf+q./k/gamma^2; T_gamma=T_0-sol_part;
N=tmax/dt+1;
t =0:dt:tmax;
x = linspace(0,L,n);
kk=1; Treal=zeros(N,n); toll=20;
const=@(x) -2*(T_gamma*x*pi*2/((2*L*gamma)^2+(x*pi)^2)+2*(q/k/gamma^2+T_inf-T_zero)/x/pi);
% func=@(x,t) ...
    const*exp(-gamma^2/eta*t)-T_gamma/(1+exp(2*gamma*L))*(exp(2*gamma/L)*exp(-gamma*x)+exp(gamma*x))+T_gamma/
for i=1:N
    for j=1:n
        temp=T_gamma/(1+exp(2*gamma*L))*(exp(-gamma*x(j))*exp(2*gamma*L)+exp(gamma*x(j)))+sol_part;
        if i==1
            if j==1
                temp=T_0;
            else
                temp=T_zero;
            end
        else
            while toll>1e-4 || kk<100000
                C=-(kk*pi/2/L)^2;
                more=const(kk)*exp((C-gamma^2)*alpha*t(i))*sin(kk*pi/2/L*x(j));
                temp=temp+more;
                toll=abs(more);
                kk=kk+2;
            end
        end
        Treal(i,j)=temp;
        kk=1; toll=20;

    end
end
figure(1)
[X,tempo]=meshgrid(x,t);
surf(X,tempo,Treal) % mesh
xlabel('Lenght (m)');
ylabel('time (s)');
zlabel('Temperature ( C )')
title('FOURIER WEAK SOLUTION')
```

It's easy to see, comparing the data tables of the two methods with the analytical one, that both methods are incredibly near to the real solution. It result, from a checking on the norm of the values, that the real relative error on the whole discretized domain of the EbFVM is 0.29%, whereas is 0.23 for the FEM, that is then slightly better, probably due to the very redifined grid in time with respect to the EbFVM one. Note that this analytical solution is still subjected to approximation when computed, because practically the solver has to consider a finite value of N, a finite sum, and the higher the requirements for the approximation are, the more time the CPU will take to compute the sum of the series for each point, therefore it's not so trivial that this analytical solution would compute all the desired points of the domain in less time than the numerical methods. The good point of this solution is that any point of the domain can be derived without involving the solution on the others.

4.5 Final considerations

We first remark, after have seen the real results, how powerful was the Scale Analysis tool since it gave a very good idea of the time scale for the problem with a very easy computation (we found an approximate value of 0.46 seconds as the time in which the flux entering though left surface starts influence the right extreme, whereas the real value computed is approximately 0.12 seconds. Analyzing both methods from the development point of view, it's clear how simpler and easier to understand is the FVM with respect to the FEM. This because the FVM is basically a balance in a discrete set, and it could be computed even without involving the PDE, just doing explicitly the balance. For a 2D or 3D problem the derivation of the algorithms for the two methods is very similar, only with different numbers of entries in the matrix of the linear systems (5 and 7 for each row respectively for 2D and 3D problems, with linear approximation). Looking at the approximations, the EbFVM has proved to be more effective also in this field, since we computed an incredibly good solution with half of the time step of the FEM. On the other side, even if with more iterations, the FEM approximates the solution in less CPU time (approximately 0.01 seconds against 1 of the EbFVM) and thus it's not completely true that the FVM is the best approach for the numerical solution of the problem, but it's up to the solver to look at strengths and weaknesses of every method and choose the best one depending on his purposes. The key point of the EbFVM is that is simple, strongly correlated with the physic of the problem. Has been also proved that in this particular case it's possible to compute an analytical solution in a weaker sense by Fourier analysis, but clearly this is not always possible, so it's mandatory to have numerical algorithms to compute the solution for practical purpose. Moreover even when it's possible to compute an analytical solution, as in this case, the computations are not trivial, and requires a lot of developing time, whereas numerical solution could be simpler to implement and more rapid to achieve an approximate solution.

Bibliography

[1] G. Gaeta. "L'equazione del calore o di diffusione". In: (2019).

[2] MATLAB. *version R2020a*. Natick, Massachusetts: The MathWorks Inc., 2020.