STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–12.
Published online in International Academic Press (www.IAPress.org)

Numerical methods for evolutionary problems in partial differential equations and control

Guillermo Villa Martínez [1,*], Carlos Alberto Ramírez Vanegas [1], Oscar Danilo Montoya Giraldo [2]

[1] *Department of Mathematics, Universidad Tecnológica de Pererira, Colombia*
[2] *Grupo de Compatibilidad e interferencia Electromagnética, Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas, Bogotá, 110231, Colombia*

Abstract
In this paper we implement the finite element method for parabolic problems with dominant transport terms. The formulation of the linear system of equations coming from a partial differential equation takes its form from the weakening of the problem. Several numerical experiments are performed to know the convergence of the solution and an error analysis in a Sobolev space for linear functions is implemented.

Keywords finite element method, finite element analysis, partial differential equation, Poission problem, weak formulation, discretization, assembly, reference element, Gaussian quadrature.

AMS 2010 subject classifications ???, ???

DOI: 10.19139/soic-2310-5070-2489

1. Introduction

In this study, the strong and weak formulation for a class of partial differential equations is presented. We address numerical approximation of this class of evolutionary problems combined with variational problems and their applications in the modeling of the heat equation with dominant transport term is performed. The solution of the problem is approached from the point of view of weak formulation and strong formulation to transform a partial differential equation into a linear system of equations of constant terms via the finite element method. For the solution of the proposed partial differential equation, the complete discretization in both time and spatial domains is taken into account. Finally, initial and boundary conditions are considered. Several numerical experiments are performed to know the interaction of the number of elements with the minimization of a functional in an arbitrary subdomain. The numerical experiments show the effectiveness of the method and the proposed solution model.

1.1. Notation

- $\varphi(x)$ test function.
- $\dfrac{\partial u}{\partial t}$ partial derivative with respect to time.
- $\dfrac{\partial u}{\partial x}$ partial derivative of the displacement.
- $f(x,t)$ forcing term.
- γ; thermal diffusivity constant.
- β; transport term .
- ξ_i Gauss points.
- ω_i Gauss weights.

*Correspondence to: Guillermo Villa Martínez (Email: gvilla@utp.edu.co). Department of Department of Mathematics, Universidad Tecnológica de Pererira, Carrera 27 #10-02 Barrio Alamos - Pereira - Risaralda - Colombia (660003).

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 202x International Academic Press

1.2. Organization

This paper is organized as follows: In the preliminaries section, the finite element method for a second order differential equation is raised. The galerkin projection is expressed with linear elements. Then, some numerical experiments with different interval partitions are performed to verify the convergence of the solution as well as, the values that minimize the functional to find the transport term at different finite element values. Finally, it is concluded for the parabolic problem in partial differential equations.

2. Basic algorithm and extensions

To begin with, the linear form functions are presented are introduced for discretizing the solution function of the given differential equation. At the first level, the shape functions are defined globally. At the second level, the shape functions are defined at the element level. Along the way, the concepts of assembly, integration by substitution and Gaussian quadrature are explained.

2.1. Test functions

So far you have seen simple linear shape functions and elements that are defined over a 1D subdomain with two nodes. This paper focuses on the implementation of linear elements for the 1D time dependent problem. In higher dimensional space [1, 2], the same key concepts can be used. The central idea is that the solution space is reduced with the introduction of shape functions.

$$\varphi_i(x) = \begin{cases} \dfrac{nx}{L} - (i-1), & \dfrac{(i-1)L}{n} \leq x \leq \dfrac{iL}{n}. \\ \\ -\dfrac{nx}{L} + i, & \dfrac{iL}{n} \leq x \leq \dfrac{(i+1)L}{n}. \end{cases} \quad i = 0, 1, 2, \cdots, n. \tag{1}$$

The solution is given by:

$$u(x) = \sum_{i=0}^{n} u_i \varphi_i(x) = [\varphi(x)]^T \mathbf{U} \tag{2}$$

Strong formulation: Given $f(x)$, find $u(x)$ sucha that:

$$\begin{gathered} u''(x) = f(x) \\ \text{for all } 0 \leq x \leq 1 \\ u(0) = 0, u'(1) = 0 \end{gathered}$$

Weak formulation: Given $f(x)$, find $u(x)$ such that:

$$\int_0^1 u'(x)v'(x)dx = -\int_0^1 f(x)v(x)dx \tag{3}$$

for all $v(x)$ with $v(0) = 0$, $u(0) = 0$
The derivative of the solution is:

$$u'(x) = \sum_{i=0}^{n} u_i \varphi_i'(x) = [\varphi'(x)]^T \mathbf{U} \tag{4}$$

The test function:

$$v(x) = \sum_{i=0}^{n} v_i \varphi_i(x) = \mathbf{V}^T [\varphi(x)] \tag{5}$$

The derivative of the test function:

$$v'(x) = \sum_{i=0}^{n} v_i \varphi_i'(x) = \mathbf{V}^T[\varphi'(x)] \tag{6}$$

Where:

$$\mathbf{U} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_n \end{bmatrix} \quad (7) \qquad \mathbf{V} = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \end{bmatrix} \quad (8) \qquad [\varphi(x)] = \begin{bmatrix} \varphi_0(x) \\ \varphi_1(x) \\ \vdots \\ \varphi_n(x) \end{bmatrix} \quad (9) \qquad [\varphi'(x)] = \begin{bmatrix} \varphi_0'(x) \\ \varphi_1'(x) \\ \vdots \\ \varphi_n'(x) \end{bmatrix} \quad (10)$$

Then equation (3) can be written matrixially as follows [1]:

$$\int_0^1 \mathbf{V}^T[\varphi'(x)][\varphi'(x)]^T \mathbf{U} dx = -\int_0^1 f(x)\mathbf{V}^T[\varphi(x)]dx. \tag{11}$$

The following terms can be extracted from equation (10):

$$\mathbf{V}^T \int_0^1 [\varphi'(x)][\varphi'(x)]^T dx \mathbf{U} = \mathbf{V}^T \int_0^1 -f(x)[\varphi(x)]dx. \tag{12}$$

Where:

$$\mathbf{K} = \int_0^1 [\varphi'(x)][\varphi'(x)]^T dx, \tag{13}$$

and

$$\mathbf{F} = \int_0^1 -f(x)[\varphi(x)]dx. \tag{14}$$

This allows us to rewrite equation (3) in terms of a linear system of equations of the form [3]:

$$\mathbf{V}^T \mathbf{K} \mathbf{U} = \mathbf{V}^T \mathbf{F}, \tag{15}$$

$\forall \mathbf{V}$ with $v_0 = 0$ and $u_0 = 0$

Afterwards, globally defined shape functions $\varphi_i(x)(i = 0, 1, \ldots, n)$, element wise defined shape functions $\varphi_j^e(x)(e = 1, 2, \ldots n, j = 0, 1)$. Now the contribution of each element to the solution has the form:

$$u^e(x) = \sum_{j=0}^{1} u_j^e \varphi_j^e(x) = [\varphi^e]^T(x)\mathbf{U}^e \tag{16}$$

therefore, the equation (12), written by the contribution of each element:

$$\sum_{e=1}^{n} \mathbf{V^{eT}} \int_{\Omega_e} [\varphi_e'(x)][\varphi_e'(x)]^T dx \mathbf{U}^e = \sum_{e=1}^{n} \mathbf{V}^{eT} \int_{\Omega_e} -f(x)[\varphi^e(x)]dx. \tag{17}$$

In terms of each element the matrices (13) and (14) are given by:

$$\mathbf{K}^e = \int_{\Omega_e} [\varphi_e'(x)][\varphi_e'(x)]^T dx, \tag{18}$$

and,

$$\mathbf{F}^e = \int_{\Omega_e} -f(x)[\varphi^e(x)]dx. \tag{19}$$

In the interval [a,b] the element Ω_e. x in terms of a new variable ξ taking values between $[-1, 1]$ as a reference has the following expression:

$$x = \phi_e(\xi) = \frac{1}{2}(1-\xi)\,a + \frac{1}{2}(1+\xi)\,b. \tag{20}$$

On the other hand, the inverse process is given by:

$$\xi = \phi_e^{-1}(x) = \frac{\frac{1}{2}(a+b) - x}{\frac{1}{2}(a-b)}. \tag{21}$$

Now, in terms of the contribution of each element:

$$\varphi_j^e(x) = \varphi_j^e(\phi_e(\xi)) = \varphi_j(\xi) \tag{22}$$

And the derivatives:

$$\frac{\partial \varphi_j^e(x)}{\partial x} = \frac{\partial \varphi_j(\xi)}{\partial \xi}\frac{\partial \xi}{\partial x} = \frac{\partial \varphi_j(\xi)}{\partial \xi}\left(\frac{\partial \varphi_e(\xi)}{\partial \xi}\right)^{-1} \tag{23}$$

Using Gaussian quadrature, goven a polynomial $P(\xi)$ and a set of Gauss points ξ_i and Gauss weights ω_i, te integral $P(\xi)$ from $\xi = -1$ to $\xi = 1$ can be computed as:

$$\int_{-1}^{1} P(\xi)d\xi = \sum_i P(\xi_i)\omega_i \tag{24}$$

In this case, polynomial order 5 (or less), numeber od Gauss points: 3, Gauss point ξ_i : $\left(-\sqrt{\frac{3}{3}}, 0, \sqrt{\frac{3}{5}}\right)$ and Gauss weights $\omega_i = \left(\frac{5}{9}, \frac{8}{9}, \frac{5}{9}\right)$

3. PDE with time dynamics

In this section we consider the temporal dynamics of the following partial differential:

$$\frac{\partial u}{\partial t} + \gamma\frac{\partial u}{\partial x} - \frac{\partial}{\partial x}\left(\beta\frac{\partial u}{\partial x}\right) = f(x,t) \tag{25}$$

Following the same methodology proposed in the previous section, it is multiplied by a test function $v(x)$ and integrated into x:

$$\int \frac{\partial u}{\partial t}vdx + \int \gamma\frac{\partial u}{\partial x}vdx - \int \frac{\partial}{\partial x}\left(\beta\frac{\partial u}{\partial x}\right)vdx = \int f(x,t)vdx \tag{26}$$

When multiplying by a test function $\varphi(x)$ (1), the solution is given by:

$$u(x,t) = \alpha_1(t)\varphi_1(x) + \cdots + \alpha_n(t)\varphi_n(x) + \alpha_{n+1}(t)\varphi_{n+1}(x) = \sum_{i=1}^{n+1}\alpha_i(t)\varphi_i(x) \tag{27}$$

The derivative with respect to time involves deriving the equation (27):

$$\frac{\partial u(x,t)}{\partial t} = \alpha_1'(t)\varphi_1(x) + \cdots + \alpha_n'(t)\varphi_n(x) + \alpha_{n+1}'(t)\varphi_{n+1}(x) = \sum_{i=1}^{n+1} \alpha_i'(t)\varphi_i(x) \tag{28}$$

The product $\int \frac{\partial u}{\partial t} v dx = \int \sum_{i=1,j=1}^{n+1} \alpha_i(t)^{'} \varphi_i(x)\varphi_j(x)dx$ and this in turn generates a matrix $M = [m_{ij}]$ which we will call the mass matrix [1, 2]:

$$M = [m_{ij}] = \int_a^b \varphi_i(x)\varphi_j(x)dx \qquad i,j = 1,2,\cdots,n+1 \tag{29}$$

The diagonal entries $i = j$ for the equation matrix (29):

$$[m_{ii}] = \int_{iL/n}^{(i+1)L/n} \left(\frac{nx}{L} - i\right)^2 dx + \int_{(i+1)L/n}^{(i+2)L/n} \left(-\frac{nx}{L} + (i+1)\right)^2 dx \tag{30}$$

Now, for $i \neq j$ the off-diagonal terms are calculated with the integral:

$$[m_{ij}] = \int_0^L P_i P_j dx \tag{31}$$

Where, P_i and P_j have the form of the equation (1). Then, the product defined by $\int \gamma \frac{\partial u}{\partial x} v dx = -\gamma \int \sum_{i=1,j=1}^{n+1} \alpha_i(t)\varphi_i'(x)\varphi_j(x)dx$ which generates a matrix $S = [s_{ij}]$ that takes the name of *stiffness* matrix:

$$S = [s_{ij}] = -\gamma \int_a^b \varphi_i'(x)\varphi_j(x)dx \qquad i,j = 1,2,\cdots,n+1 \tag{32}$$

For the diagonal terms $i = j$ it is calculated with the integral:

$$[s_{ii}] = -\gamma \left(\int_{iL/n}^{(i+1)L/n} \left(\frac{n}{L}\right)\left(\frac{nx}{L} - i\right) dx + \int_{(i+1)L/n}^{(i+2)L/n} \left(-\frac{n}{L}\right)\left(-\frac{nx}{L} - (i+1)\right) dx \right) \tag{33}$$

When $i \neq j$ you have the off-diagonal values:

$$[s_{ij}] = -\gamma \int_{(i+1)L/n}^{(i+2)L/n} \left(-\frac{n}{L}\right)\left(\frac{nx}{L} - i\right) dx \tag{34}$$

On the other hand, the product $-\beta \int \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) v$

generates $-\beta \int \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) v = \beta \int_a^b \sum_{i=1,j=1}^{n+1} \alpha_i(t)\varphi_i'(x)\varphi_j'(x)dx$. That is, the recurrence equations (13,14) and change the constants multiplying this matrix by $-\beta$:

$$A = [a_{ij}] = \beta \int_a^b \varphi_i'(x)\varphi_j'(x)dx \qquad i,j = 1,2,\cdots,n+1 \tag{35}$$

The diagonal terms are given by the following expression:

$$[a_{ii}] = \beta \left(\int_{iL/n}^{(i+1)L/n} \left(\frac{n}{L}\right)^2 dx + \int_{(i+1)L/n}^{(i+2)L/n} \left(-\frac{n}{L}\right)^2 dx \right) \tag{36}$$

The values outside the diagonal (the integration interval is constant) are calculated as follows:

$$[a_{ij}] = \beta \left(\int_{iL/n}^{(i+1)L/n} \left(\frac{n}{L}\right)\left(-\frac{n}{L}\right) \right) dx \tag{37}$$

Now, $\int f(x)vdx = \int_a^b f(x)\varphi_i(x)dx$ allows to calculate the terms of the load matrix $B = [b_i] = \int_a^b f(x)\varphi_i(x)dx$

$$[b_i] = \int_a^b f(x)\varphi_i(x)dx \tag{38}$$

Finally, the linear system of differential equations is given by:

$$[M]\frac{d\vec{\alpha}}{dt} - [S]\vec{\alpha} + [A]\vec{\alpha} = B \tag{39}$$

Grouping terms:

$$[M]\frac{d\vec{\alpha}}{dt} + ([A] - [S])\,\vec{\alpha} = B \tag{40}$$

In a full discretization in time, consider the equation (41):

$$[C]\frac{d\vec{U}}{dt} + [k]\vec{U} = [Q] \tag{41}$$

Where:

$$\vec{U} = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_N \end{bmatrix} \tag{42}$$

$N \to$ for x (Spatial dimension). In the time discretization we will use the *Backward Eueler* or implicit Euler algorithm (n for time), for all Δt equally spaced:

$$\frac{d\vec{U_{n+1}}}{dt} = \frac{1}{\Delta t}\left(\vec{U_{n+1}} - \vec{U_n}\right) \tag{43}$$

Evaluating equation (43) in equation (42) we have:

$$\frac{1}{\Delta t}[C]\left(\vec{U_{n+1}} - \vec{U_n}\right) + [K]\vec{U_{n+1}} = [Q] \tag{44}$$

By assuming $\vec{U_n}$ known, it is required to calculate $\vec{U_{n+1}}$. In the iteration $k+1$ you have the approximation $\vec{U_{n+1}^{(k)}}$ for $\vec{U_{n+1}}$. We are looking for the approach $\vec{U_{n+1}^{(k+1)}}$ that satisfies [3]:

$$\vec{U_{n+1}^{(k+1)}} = \vec{U_{n+1}^{(k)}} + \Delta\vec{U_{n+1}^{(k)}} \tag{45}$$

Returning to the equation (40), $\frac{d\vec{\alpha_m}}{dt} = \frac{1}{\Delta t}\left(\vec{\alpha_m} - \vec{\alpha_{m-1}}\right)$ and making $[A] - [S] = [SA]$ it holds:

$$[M]\frac{1}{\Delta t}\left(\vec{\alpha_m} - \vec{\alpha_{m-1}}\right) + [SA]\vec{\alpha_m} = B \tag{46}$$

By factoring and grouping terms we obtain:

$$\vec{\alpha_m}\left(\frac{1}{\Delta t}[M]+[SA]\right)=B+[M]\frac{1}{\Delta t}\vec{\alpha_{m-1}} \qquad m=1,2,3,\cdots,n+1 \tag{47}$$

Given the initial conditions $\vec{\alpha_0}$, the matrices $[M]$, $[S]$, $[A]$ and B are the same as the previous problem. La solución de la ecuación (47) can be realized by the inverse of $\left(\frac{1}{\Delta t}[M]+[SA]\right)$ or implementing the **MatLab** function *ode45*.

Algorithm 1 Finite Element Method (FEM)

1: **Define the problem domain and boundary conditions.**
2: **Discretize the domain:** divide the domain into finite elements and define the mesh.
3: **Select shape functions** for each element to approximate the solution.
4: **Formulate element matrices and vectors:**
5: **for** each element **do**
6: Compute local Mass matrix 29.
7: Compute local stiffness matrix 32.
8: Compute local diffusion stiffness matrix 35.
9: Compute local load vector 38.
10: **end for**
11: **Assemble global matrices and vectors:**
12: **for** each element **do**
13: Map local matrices to global matrices.
14: **end for**
15: **Apply boundary conditions** to modify the global system.
16: **Solve the global system of equations** to obtain nodal values (Euler's Method, ODE45).
17: **Post-process the solution:**
18: **Apply Fminsearch:** Use fminsearch (Matlab function) to find the minimun of (61) options = optimset('Display','iter','PlotFcns',@optimplotfval);
19: **Post-process the solution** Visualize results, compute derived quantities, and check convergence (GCI).

4. Error estimation

This section shows the energy norm for determining the error in a linear function space. If u is the solution, v is some function such that $v(0)=0$ and without loss of generality it is denoted by $S=\mathcal{P}^1(\tau^h)$ y $V=V^h$ as a finite dimensional subspace [4], such that:

$$u_s \in S \mid a(u_s,v)=(f,v) \quad \forall v \in S \tag{48}$$

It can be said that the solution u is characterized by:

$$u \in V \mid a(u,v)=(f,v) \quad \forall v \in V \tag{49}$$

To observe the orthogonality relationship between u and u_s, subtract the equations (49) and (48):

$$a(u-u_s,\omega)=0 \quad \forall \omega \in S \tag{50}$$

Now, energy is defined as the norm:

$$||v||_E=\sqrt{a(v,v)} \quad \forall v \in V \tag{51}$$

The relationship between the norm and the inner product is given by the Schwarz inequality:

$$|a(v,\omega)| \leq ||v||_E ||\omega||_E \quad \forall v,\omega \in V \tag{52}$$

Then, for some vS:

$$||u-u_S||_E^2 = a(u-u_S, u-u_S)$$

(Here it is for the Schwarz inequality).

$$||u-u_S||_E^2 = a(u-u_S, u-v) + a(u-u_S, v-u_S)$$

From equation (50) it is known that $a(u-u_s,\omega)=0$ with $\omega = v-u_S$,

$$||u-u_S||_E^2 = a(u-u_S, u-v)$$

Then, using Schwarz's inequality (52),

$$||u-u_S||_E^2 \leq ||u-u_s||_E ||u-v||_E. \tag{53}$$

If $|||u-u_s||_E \neq 0$, it can be divided by $||u-u_s||_E$ to obtain $||u-u_s||_E \neq |||u-v||_E$ for some $v \in S$. Moreover, in the case $||u-u_s||| = 0$, the inequality is satisfied since the obtained solution coincides with the solution in S [4, 5]. Taking the infimum over $v \in S$:

$$||u-u_S||_E \leq \inf\{||u-v||_E \mid v \in S\}.$$

Since $u_S in S$, one has:

$$\inf\{||u-v||_E \mid v \in S\} \leq ||u-u_s||_E.$$

Therefore,

$$||u-u_s||_E = \inf\{||u-v||_E \mid v \in S\}$$

This means that, there is an element (u_s) for which the infimum is reached. If the infimum is replaced by the minimum it can be proved that the error estimate can be expressed as follows:

$$||u-u_s||_E = \min\{||u-v||_E \mid v \in S\}.$$

That is, the error in the energy defined by the standard (53) is optimized.

5. Experimental results

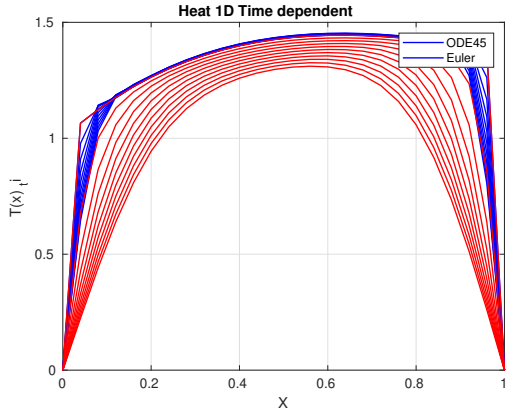
In the following subsection the proposed differential equation is solved with the proposed conditions at different interval partitions for a constant time delta. The following parameters were implemented to solve the differential equation (25)

5.1. PDE time dependent

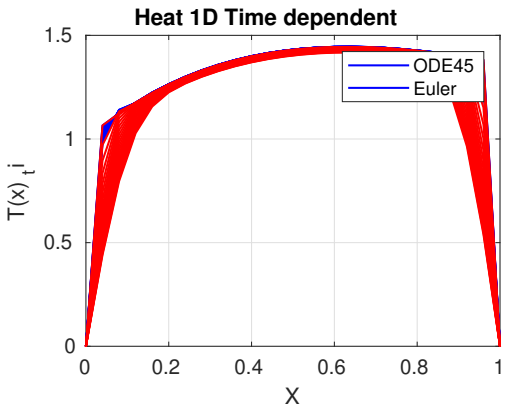
Parameters:

- $n = [25, 50, 100]$
- $a = 0$; a lower limit
- $L = 1$; b upper limit
- $\gamma = 0.1$; thermal diffusivity constant
- $\beta = 1$; transport term
- $tf = 0.1$; final Time

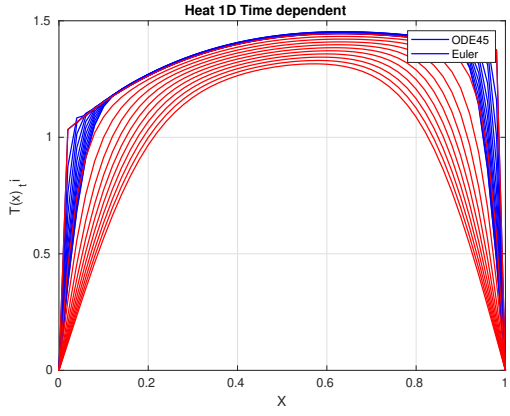
- $Nt = [100, 1000]$; time partition
- $dt = tf/Nt$; delta time
- $Ts = 0$ Boundary conditions
- $Ta = Ts$; Temperature at a $T(x = a) = Ts$
- $Tb = Ts$; Temperature at a $T(x = b) = Ts$



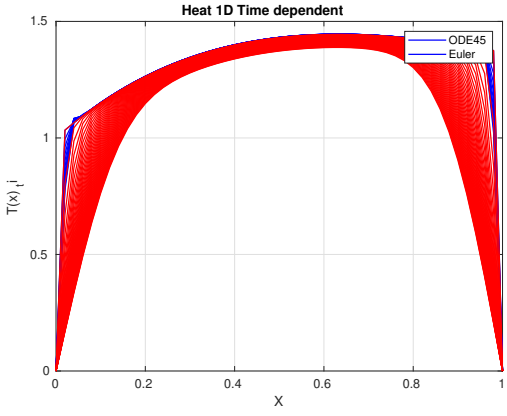
(a) $\Delta t = 0.01s$, $n = 25$



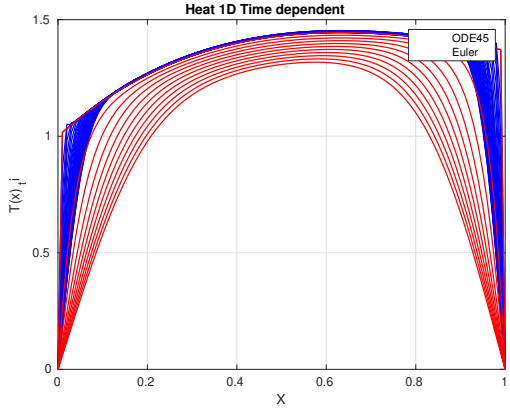
(b) $\Delta t = 1.0000 \times 10^{-03}s$, $n = 25$



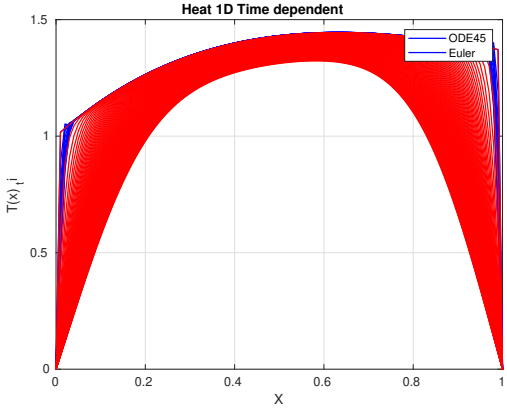
(c) $\Delta t = 0.01s$, $n = 50$



(d) $\Delta t = 1.0000 \times 10^{-03}s$, $n = 50$



(e) $\Delta t = 0.01s$, $n = 100$



(f) $\Delta t = 1.0000 \times 10^{-03}s$, $n = 100$

Figure 1. Solutions considering different scenarios. Column 1 figures (a), (c) and (e) for $\delta t = 0.01$ and $n = 25$, $n = 50$, $n = 100$. Column 2 figures (b), (d) and (f) $\delta t = 1 \times 10 - 03$ and $n = 25$, $n = 50$ and $n = 100$.

The euler algorithm (red) is used to calculate the answers, with a runtime $= 49.8124s$ and the $ODE45$ matlab function (blue) is used to calculate the numerical solutions with runtime $= 10.5213s$.

5.2. *Constant* β

As shown in the previous section, we solved the problem posed by the equation (25) [6]. A value of $\gamma \in \mathbb{R}$ for the calculation of the matrix S [7]:

$$S_{ii} = \beta \int_0^L \varphi_i'(x)\varphi_j(x). \tag{54}$$

Expanding the terms for n, the diagonal terms are given by the following expression:

$$S_{ii} = \beta \left(\int_{\frac{iL}{n}}^{\frac{(i+1)L}{n}} \left(\frac{n}{L}\right)\left(\frac{nx}{L} - i\right) dx + \int_{\frac{(i+1)L}{n}}^{\frac{(i+2)L}{n}} \left(-\frac{n}{L}\right)\left(-\frac{nx}{l} + i + 1\right) dx \right). \tag{55}$$

The off-diagonal terms are given by:

$$S_{ij} = \beta \int_{\frac{(i+1)L}{n}}^{\frac{(i+2)L}{n}} \left(-\frac{n}{L}\right)\left(\frac{nx}{L} - i\right) dx. \tag{56}$$

The formulation of the $J(\beta)$ functional is presented below.

5.3. *Grid Convergence Index*

The grid convergence index (GCI) is an estimate of the discretisation error for the local [11] or global quantify we are studying (e.g. potential/presure (stress/displacement)) at a point.

1. One way to compute the GCI is to perform the calculation using 3 separate meshes with successively decreasing refinement (or corarsing) i.e the element size h_1 for mesh 1, h_2 for mesh 2 and h_3 for mesh 3, $h \to 0$, $h_1 > h_2 > h_3$.
2. Calculate the order of accuracy p using the equation below, where f_1 is the value for the quantity of interest on the finest mesh (smallest elements), f_2 is the value on the middle density mesf and f_3 is the value on the corasest mesh (largest elements). r es 2 in the case where the mesh sizing is halved each iteration:

$$p = \frac{\ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right)}{\ln(r)} \tag{57}$$

3. Estimate the error on the finest mesh using the equation (57):

$$E_1 = \frac{f_2 - f_1}{1 - r^p} \tag{58}$$

4. Calculate the GCI using the equation:

$$GCI_1 = F_s|E_1| \tag{59}$$

5. We can now say we are 95% confident that answer is:

$$f_{true} = f_1 \pm F_s|E_1| \tag{60}$$

-	**Mesh spacing**	**Refinement ratio** (r)	**Aproximation**
Coarsest mesh	0.04	–	1.3737
Medium mesh	0.02	2	1.3742
Finest mesh	0.01	2	1.3744

Table 1. Refinement ratio.

Coarsest mesh is the mesh with the largest element size, finest mesh is the mesh with the smallest element size. Mesh spacing is the average element width. Refinement ratio is the ratio of the coarser mesh element size to the finer mesh element size. Aproximation represents the predicted value of interest at $t = 0.022$ and $x = 0.4$, (e.g. displacement or stress at a point, or the total force on a boundary, or total energy in the model).

Check refinement ratios	OK
Order of accuracy (p)	1.321928095
Estimate of the finest mesh error	0.000133333
Fs	1.25
GCI	0.0001666666667
Estimate of the true solution	1.37
GCI 95% confidence interval min	1.37
GCI 95% confidence interval max	1.37

Table 2. Estimating discretisation error using the grid convergence index (GCI).

Check refinement ratios, if "NOT OK" is displayed then you need to check that the refinement ratio between the coarse mesh and medium mesh is the same as the refinement ratio between the medium mesh and the fine mesh. Order of accuracy (p) if this is negative then the prediction is far from the mesh independent solution and the GCI method cannot be used. Estimate of the finest mesh error has the same units as the prediction; this is the signed error, i.e. it can be positive or negative. Fs, factor of safety used by the GCI method. GCI on the finest mesh. This is a conservative estimate of the absolute value of the error on the finest mesh. The GCI method says with 95% confidence that the true solution is between the min and the max

5.4. Functional $J(\beta)$

Due to optimal control, the objective is not only to find a solution to the PDE but also to *control* certain aspects of the system. This leads to the introduction of a **cost functional** that quantifies the performance of a control input β. The functional $J(\beta)$ measures both the energy of the control and the deviation of the state u from desired behaviors.

This functional includes three main terms:

- The **control cost**, which penalizes the magnitude of the control β.
- The **state cost**, which measures the magnitude of the state variable u over the control horizon.
- The **terminal cost**, which penalizes deviations of the state u at the final time t_f.

The minimization of $J(\beta)$ under the dynamics imposed by the PDE aims to balance control effort and system performance, providing an optimal control β that drives the state u towards a desired behavior while minimizing costs. This subsection presents the formulation of a convex functional. The following is the formulation of the functional $J(\beta)$, with $\beta \in [-1, 1]$ [8]. Consider $|u|^2_{L^2}$ norm of L^2 u.

$$J(\beta) = \frac{1}{2}\int_{[a,b]\times[0,tf]} |\beta|^2 dxdt + \frac{1}{2}\int_{R\times[t_i,t_f]} |u|^2 dxdt + \frac{1}{2}\int_{\Omega} |u(t_f)|^2 dx. \tag{61}$$

The above integrals can be calculated as follows:

$$\frac{1}{2}\int_{[0,1]\times[0,tf]} |\beta|^2 dxdt = \frac{1}{2}\beta(b-a)t_f. \tag{62}$$

In the second integral, a part of the domain Ω is selected to match the mass matrix [9], i.e., $\alpha_R^T M_R \alpha_R$ with T elements such that $T \subset R$.

$$\frac{1}{2}\int_{R\times[t_i,t_f]} |u|^2 dxdt = \frac{1}{2}\sum_{l=0}^{t_u}\left(\int_R u(x,t_l)dx\right)\Delta t = \frac{1}{2}\sum_{l=0}^{t_u}\left(\alpha_R^T M_R \alpha_R\right)\Delta t. \tag{63}$$

On the other hand $\alpha(t_f)$ is taken as the last vector of α thus:

$$\frac{1}{2}\int_\Omega |u(t_f)|^2 dx = \frac{1}{2}\alpha^T(t_f)M\alpha(t_f) = \frac{1}{2}u(x,t_f). \tag{64}$$

This functional is in principle convex to guarantee the minimum [10]. The minimum value of the functional is at one of the extremes or within the interval.

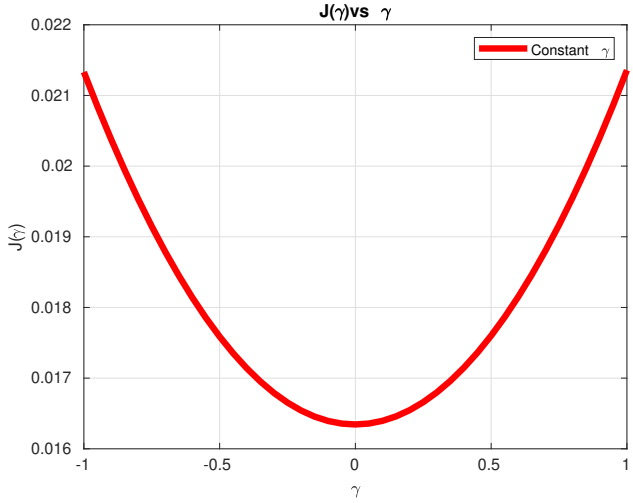


Figure 2. Functional $J(\beta)$. (Graph created in $MATLAB$, authors).

The table calculates the values of β and the functional for different values of elements in the domain partition.

Table 3. β values for different domain partitions.

n	β	$J(\beta)$
4	-0.037792	0.64783
10	-0.12632	0.81218
100	-0.14703	0.81459
102	-0.12635	0.81218

6. Conclusion

We successfully implemented the finite element method for parabolic problems with dominant transport terms, providing a robust approach to address the numerical approximation of such evolutionary problems. By leveraging both the strong and weak formulations, we transformed the governing partial differential equations into a linear

system of equations. The numerical experiments conducted validated the effectiveness of the proposed method, demonstrating convergence of the solution and the influence of element discretization on the minimization of the functional. Finally, Runtime was compared in both methods: Euler and $ODE45$; the Euler's method was Throughput-focused: Use lower-order fixed-step or inexact implicit methods for massive ensembles and $ODE45$ was fast startup, small memory overhead.

Acknowledgement

This work was supported by master in mathematic program at Universidad Tecnológica de Pereira.

REFERENCES

1. Justin Mouyedo Loufouilou, Joseph Bonazebi Yindoula, Gabriel Bissanga, *A new approach for pseudo hyperbolic partial differential equations with nonLocal conditions using Laplace Adomian decomposition method*, International Journal of Applied Mathematics and Theoretical Physics, 7(1), 28–39 2024.
2. Xiu Ye, Shangyou Zhang *Two-Order Superconvergent CDG Finite Element Method for the Heat Equation on Triangular and Tetrahedral Meshes*, Communications on Applied Mathematics and Computation, 2024.
3. Markus Merkel, Andreas Öchsner, *One-Dimensional Finite Elements An Introduction To The Method*, Book, Springer, 2023.
4. Eduardo Casas, Karl Kunisch, *Infinite Horizon Optimal Control for a General Class of Semilinear Parabolic Equations*, Applied Mathematics Optimization, vol. 88, article 47, 2023.
5. Bhagyashree Prabhune, Krishnan Suresh, *An isoparametric tangled finite element method for handling higher-order elements with negative Jacobian*, Springer, Computational Mechanics, vol. 73, pp. 159-176, 2024.
6. Li Chen, Veniamin Gvozdik, Yue Li, *Rigorous derivation of the degenerate parabolic-elliptic Keller-Segel system from a moderately interacting stochastic particle system. Part I Partial differential equation*, Journal of Differential Equations, Volume 375, pp. 567-617, 2023.
7. Hongliang Liu, Yilin You, Haodong Li, Shoufu Li, *Canonical Euler splitting method for parabolic partial functional differential algebraic equations*, Applied Numerical Mathematics, vol. 190, pp. 65-83, 2023.
8. Markus Bachmayr, Manfred Faldum, *A space-time adaptive low-rank method for high-dimensional parabolic partial differential equations*, Journal of Complexity, Volume 82, 2024.
9. N.N. Nefedov, *Development of methods of asymptotic analysis of transition layers in reaction–diffusion–advection equations: Theory and applications*, Differential Equations, vol. 57, no. 12, pp. 1701–1721, 2021.
10. Yan-ping Chen, Jian-wei Zhou and Tian-liang Hou , *Two-grid Method of Expanded Mixed Finite Element Approximations for Parabolic Integro-differential Optimal Control Problems*, Acta Mathematicae Applicatae Sinica, English Series, 2024.
11. E. Siva Prasad and K. Phaneendra, *A Computational Scheme for 1D Time-Dependent Singularly Perturbed Parabolic Differential-Difference Equations*, Computational Mathematics and Mathematical Physics,Volume 65, pages 236–251, 2025.