Numerical Methods for Differential Equations
Homework 3th

Lorenzo Schiavone

January 31, 2025

Contents

1 Exercise

Provide the numerical solution with the Galerkin method of the heat equation in the following initial boundary value problem until time $T = 10$: the domain is a horseshoe shape as shown in Figure 1, at $t = 0$ the temperature u is zero everywhere. The boundary condition, for $\Gamma_D = \Gamma_{D1} \cup \Gamma_{D2}$ and $\Gamma_N = \Gamma \setminus \Gamma_D$, are

- null Dirichlet boundary condition on Γ_{D1},
- Dirichlet boundary condition on Γ_{D2}, as in Figure 2,
- no flux Neumann boundary condition on Γ_N.

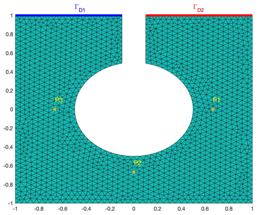


Figure 1: Domain triangularization in mesh2.

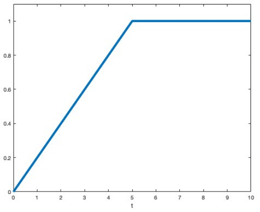


Figure 2: Dirichlet boundary condition in Γ_{D2}.

The `input` folder contains data for five different meshes, ranging from the coarsest (`mesh0`) to the finest (`mesh4`). For each mesh there are four files:

- `coord`: coordinates of all the nodes in the mesh,
- `topol`: list of nodes that form each element in the mesh,
- `bound`: boundary conditions,
- `trace`: indices of the nodes located on the external boundary,
- `track`: indices of the nodes to be tracked at each time interval.

For each of the mesh,

1. monitor the trace of the solution at $T/4, T/2, 3/4T, T$ along the external boundary, the solution at all the times on the three tracking points in `track`, the values of the solution on the three tracking points at the end of the simulation time;
2. write a table of solution values and the associated error, measured as difference with respect to values in the reference Table 1;
3. draw the semi-logarithmic convergence plots for the relative residual norm during the PCG iterations for the linear system arising from the stationary solution. Two plots with the convergence histories for all the refinement levels are required: one using Jacobi as preconditioner, the other with the incomplete Cholesky preconditioner.

t	$u(P_1)$	$u(P_2)$	$u(P_3)$
2.5	0.2434390	0.0772287	0.0183037
5.0	0.6046775	0.2751718	0.0928241
7.5	0.7454968	0.4328630	0.1716526
10.0	0.7751273	0.4805514	0.2008722

Table 1: Reference Value Table.

Moreover, include a summary table for the FEM convergence, with the error value ε_k, computed as in Subsection 2.5, and the ratio

$$r_k = \frac{\varepsilon_{k-1}}{\varepsilon_k} \left(\frac{h_k}{h_{k-1}} \right)^2 , \tag{1}$$

function of the error and the mesh size at the current and previous steps.

2 Solution development

2.1 Solution Schema

We aim to solve the heat equation

$$\frac{\partial}{\partial t} u(x,t) - \Delta u(x,t) = f(x,t) \quad \text{for } (x,t) \in \Omega \times [0,T]$$

in the weak formulation: find $u \in H^1_{\Gamma_D}$ such that

$$\int_\Omega \frac{\partial}{\partial t} uv dx + \int_\Omega \nabla u \cdot \nabla v dx = \int_\Omega f v dx + \int_\Gamma v \nabla u \cdot n d\gamma \tag{2}$$

for every $v \in H^1_{\Gamma_D}$. In our case, the boundary term vanishes due to the no flux Neumann condition in Γ_N. Approximating the solution space with X^1_h, the space of globally continuous functions, linear when restricted to each element, we consider a basis $\{\phi_1, \ldots, \phi_{N_h}\}$ of X^1_h and write $u = \sum_i u_i(t)\phi_i$. By linearity, it is enough to verify Equation 2 against all the basis function, i.e,

$$\sum_{i=1}^{N_h} u_i'(t) \int_\Omega \phi_i \phi_j dx + \sum_{i=1}^{N_h} u_i \int_\Omega \nabla \phi_i \cdot \nabla \phi_j dx = \int_\Omega f \phi_j dx, \tag{3}$$

for all $j = 1, \ldots, N_h$. Defining $m_{ij} = \int_\Omega \phi_i \phi_j dx$ and $a_{ji} = \int_\Omega \nabla \phi_i \cdot \nabla \phi_j dx$, Equation 3 can be written in matrix form as

$$M\mathbf{u}' + A\mathbf{u} = \mathbf{f}, \tag{4}$$

which is a system of ODEs, where M is called mass matrix and A stiffness matrix. Equation 4 can be solved using a θ-method, such as Crank Nicholson or Backward Euler.

2.2 Model Creation

For each mesh, we first load `coord` in `nodes` and `topol` in `elements`. Then, we compute the area of each element and store it in the vector `areas` by

```
for i = 1:tot_el
    matrix = zeros(3,3);
    for j =1:3
        matrix(j,:)= [1 nodes(elements(i, j), :)];
    end
    areas(i,1)=det(matrix)/2;
end
```

Next, we assemble the mass matrix M and the stiffness matrix A. For each element, first we obtain the local mass and stiffness matrices and then we add them to the global matrices at the appropriate locations.

The local stiffness matrix for the triangular element k is given by

$$A_{loc} = \frac{1}{4 \cdot \text{area}} \left(\begin{bmatrix} b_i b_i & b_i b_j & b_i b_m \\ b_j b_i & b_j b_j & b_j b_m \\ b_m b_i & b_m b_j & b_m b_m \end{bmatrix} + \begin{bmatrix} c_i c_i & c_i c_j & c_i c_m \\ c_j c_i & c_j c_j & c_j c_m \\ c_m c_i & c_m c_j & c_m c_m \end{bmatrix} \right) = \frac{1}{4 \cdot \text{area}} \left(\mathbf{bb}' + \mathbf{cc}' \right),$$

where area = $\texttt{areas}(k)$, and $\mathbf{b}$ and $\mathbf{c}$ are column vectors defined as

$$\begin{aligned} b_i &= y_j - y_m, & b_j &= y_m - y_i, & b_m &= y_i - y_j, \\ c_i &= x_m - x_j, & c_j &= x_i - x_m, & c_m &= x_j - x_i \end{aligned}$$

with $(x_i, y_i), (x_j, y_j), (x_m, y_m)$ the coordinates of the element's nodes. On the other hand, the local mass matrix is given by

$$M_{loc} = \frac{\text{area}}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

Finally, these local matrices are added to the global stiffness matrix A and the global mass matrix M as follows:

```
el_nodes = elements(k, :);
for i = 1:3
    row = el_nodes(i);
    for j=1:3
        col = el_nodes(j);
        A(row,col)=A(row,col) + Aloc(i,j);
        M(row,col)=M(row,col) + Mloc(i,j);
    end
end
```

2.3 Imposing Boundary Conditions

The basis functions are chosen to be *Lagrangian*, i.e., there is a basis function ϕ_i for each node N_i and $\phi_i(N_j) = \delta_{ij}$. Therefore for $u(x) = \sum_i u_i \phi_i(x)$ the coefficient u_i directly represents the value of $u(x)$ evaluated at node N_i.

Dirichlet Boundary Conditions on node $N_{\bar{i}}$ are enforced by fixing the corresponding unknown, e.g. $u_{\bar{i}} = \overline{u}$. Since then $u_{\bar{i}}$ is not an unknown anymore, the system can be simplified by replacing its corresponding row with $u_{\bar{i}} = \overline{u}$ and updating other rows by moving terms involving $u_{\bar{i}}$ to the right hand side. For instance, the i-th row

$$\sum_{j=1}^{N_h} m_{ij} u'_j + \sum_{j=1}^{N_h} a_{ij} u_j = f_i$$

becomes

$$\sum_{j=1}^{N_h} m_{ij} u'_j + \sum_{j \neq \bar{i}} a_{ij} u_j = f_i - \overline{u} a_{i\bar{i}}.$$

The implementation follows this approach. First, we store the columns of the stiffness matrix corresponding to the nodes with Dirichlet Boundary conditions. Then, we set to zero the corresponding rows and columns, and set to one the diagonal entries, as in the following code snippet:

```
A4update = A(:,bound(:,1));
nBound = size(bound,1);
A(bound(:,1),:) = 0;
A(:,bound(:,1)) = 0;
for i = 1: nBound
    A(bound(i,1),bound(i,1)) = 1;
end
```

Finally, with the right hand side $\mathbf{f}$ and the boundary conditions `bound`, we enforce them with

```
f = f-A4update *bound(:,2);
f(bound(:,1)) =bound(:,2);
```

2.4 Solution for the System of ODEs

The system of ODEs is numerically solved with the Crank Nicholson's Method as follows. The semidiscretized system

$$M\mathbf{u}' = -A\mathbf{u} + \mathbf{f}$$

is discretized in time with finite differences

$$M\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{dt} = -A\frac{1}{2}(\mathbf{u}^{k+1} + \mathbf{u}^k) + \frac{1}{2}\left(\mathbf{f}^{k+1} + \mathbf{f}^k\right), \quad \text{i.e.,}$$
$$\left(\frac{M}{dt} + \frac{1}{2}A\right)\mathbf{u}^{k+1} = \left(\frac{M}{dt} - \frac{1}{2}A\right)\mathbf{u}^k + \frac{1}{2}\left(\mathbf{f}^{k+1} + \mathbf{f}^k\right)$$

that is a linear system we could solve with the `pcg` method because A and M are sparse symmetric positive definite matrices and so is $M/dt + A/2$.

2.5 Error Computation

For each mesh k with its corresponding triangularization $\mathcal{T}_k$, the error in the stationary solution is computed as

$$\varepsilon_k = \left(\int_\Omega (u_h - u)^2 dx\right)^{1/2},$$

where u_h is the stationary solution for the mesh k and u is the solution provided in `solRef.dat`. To compute numerically the integral, we first interpolate u at the mesh nodes and then approximate the integral with the midpoint rule

$$\varepsilon_k \approx \left\{ \sum_{\text{nodes } i=1}^{n} \left[\frac{(u_i - u(x_i, y_i))^2}{3} \sum_{\substack{K \in \mathcal{T}_k, \\ K \ni N_i}} \text{area}(K) \right] \right\}^{1/2}.$$

In the MATLAB script, for each mesh refinement level k, we proceed as follow:

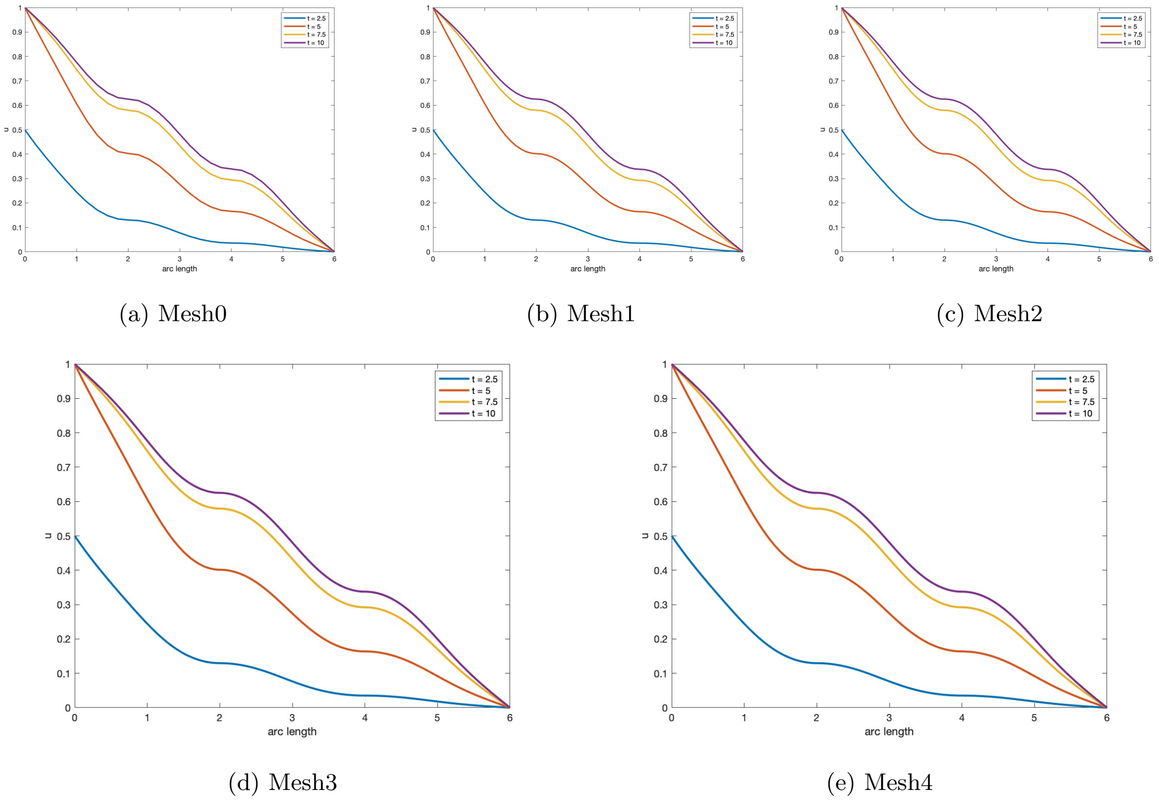
```
Ref = load("solRef.dat");
interp = scatteredInterpolant(Ref(:,1), Ref(:,2), Ref(:,3));
solintp = zeros(tot_nodes);
for j =1:tot_nodes
    solintp(j) = interp(nodes(j,:));
end
err_sq=0; %error squared
for i = 1:tot_el
    id_node = elements(i,:);
    area = areas(i);
    for j = 1:3
        err_sq = err_sq + area/3*(U(id_node(j), end)-solintp(id_node(j
           )))^2;
    end
end
errors(k+1,1)=sqrt(err_sq);
```

where we invert the order of the summation, instead of summing first over the nodes and then over the elements containing them. In this manner, we are sure that each node is counted the correct number of times without having to search for all the elements containing each node.

3 Results and Outputs

3.1 Solution Trace along the External Boundary at Prescribed Times

The solution traces along the external boundary in Figure 3 become smoother for every mesh refinement level, matching the reference plot in the assignment file. Moreover, we can notice that the small bumps in the plots correspond to the two corners of the horseshoe at arc length 2 and 4, and that the boundary conditions are satisfied: the solution is constantly zero on Γ_{D1}, while on Γ_{D2}, it is 0.5 at time 2.5 and 1 for times greater than 5, as prescribed.



(a) Mesh0

(b) Mesh1

(c) Mesh2

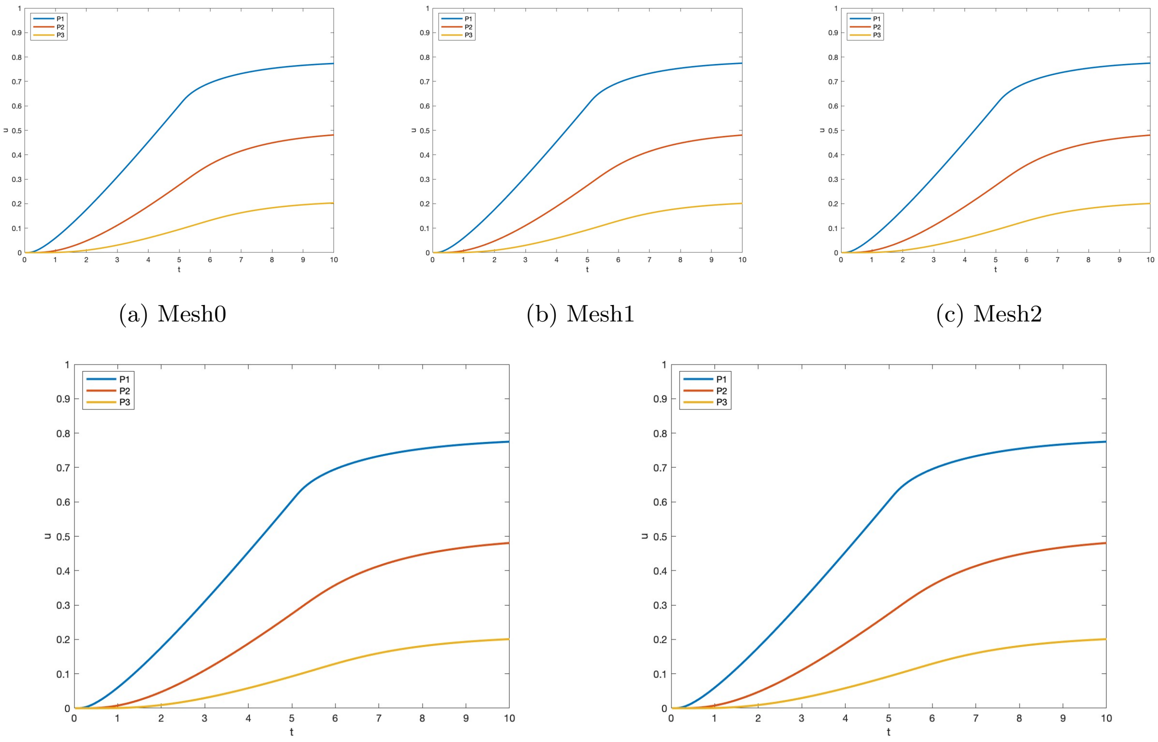
(d) Mesh3

(e) Mesh4

Figure 3: Trace of the solution along the external boundary at time $2.5, 5.0, 7.5, 10.0$.

3.2 Solution Plots at the Tracking Points for all the Times

The plots in Figure 4 show similar behavior to the reference plot in the assignment file. As the point $P1$ is the closest to the heat source on Γ_{D2}, it is the first to be affected by the boundary condition, with an almost direct temperature increase. Point $P2$ has a slower temperature rise receiving heat after a corner, and point $P3$ shows an even slower increase as the heat has to be diffused through two corners. After $t = 5$, the boundary conditions are fixed, and the solution becomes stationary, with the temperature at $P1$ stabilizing first, followed by the temperature at $P2$ and $P3$, as they are further from the heat source.



(a) Mesh0

(b) Mesh1

(c) Mesh2

(d) Mesh3

(e) Mesh4

Figure 4: Solution at the three tracking points for all the times.

3.3 Solution Values on the Tracking Points at Prescribed Times and Differences with Reference Table 1

Table 2 contains for each mesh the table of solution values on the three tracking points at time $2.5, 5.0, 7.5, 10.0$, and the difference compared to the Reference Table 1. These tables show that the solution stabilizes with mesh refinements, and the differences from the reference solution generally decrease.

t	$u(P_1)$	$u(P_2)$	$u(P_3)$	$\Delta u(P_1)$	$\Delta u(P_2)$	$\Delta u(P_3)$
2.5	0.24432	0.078884	0.019164	0.00088445	0.0016555	0.00086044
5.0	0.60531	0.27829	0.095304	0.0006359	0.0031188	0.0024794
7.5	0.74465	0.435	0.17458	-0.00084958	0.0021403	0.0029297
10.0	0.77372	0.4815	0.20329	-0.0014096	0.00094632	0.0024144

(a) Mesh 0

t	$u(P_1)$	$u(P_2)$	$u(P_3)$	$\Delta u(P_1)$	$\Delta u(P_2)$	$\Delta u(P_3)$
2.5	0.24474	0.07814	0.018675	0.0012973	0.00091082	0.0003717
5.0	0.6061	0.27668	0.093726	0.0014254	0.0015085	0.00090154
7.5	0.74556	0.4337	0.17254	6.2212e-05	0.00083883	0.00088568
10.0	0.77491	0.48088	0.2015	-0.00021873	0.00032493	0.00062533

(b) Mesh 1

t	$u(P_1)$	$u(P_2)$	$u(P_3)$	$\Delta u(P_1)$	$\Delta u(P_2)$	$\Delta u(P_3)$
2.5	0.24478	0.077935	0.018553	0.0013405	0.00070622	0.00024955
5.0	0.60617	0.27622	0.093329	0.0014887	0.0010449	0.00050466
7.5	0.74566	0.43331	0.17204	0.00015909	0.00044899	0.00038608
10.0	0.7751	0.48069	0.20109	-2.4534e-05	0.00014087	0.00021617

(c) Mesh 2

t	$u(P_1)$	$u(P_2)$	$u(P_3)$	$\Delta u(P_1)$	$\Delta u(P_2)$	$\Delta u(P_3)$
2.5	0.24479	0.077885	0.018521	0.00135	0.00065623	0.00021701
5.0	0.60619	0.27611	0.093221	0.0015153	0.00093716	0.00039695
7.5	0.7457	0.43323	0.1719	0.0002078	0.00036715	0.00024656
10.0	0.77518	0.48066	0.20097	4.8706e-05	0.00011263	9.714e-05

(d) Mesh 3

t	$u(P_1)$	$u(P_2)$	$u(P_3)$	$\Delta u(P_1)$	$\Delta u(P_2)$	$\Delta u(P_3)$
2.5	0.24478	0.07787	0.018512	0.0013455	0.00064133	0.00020834
5.0	0.60619	0.27608	0.093193	0.0015083	0.00090477	0.00036922
7.5	0.7457	0.4332	0.17186	0.00020724	0.0003409	0.00021168
10.0	0.77518	0.48065	0.20094	5.5363e-05	0.00010025	6.8108e-05

(e) Mesh 4

Table 2: Values at three tracking points and associated error with respect to the Reference Table 1 for each mesh.

3.4 PCG Convergence for the Stationary Solution

The condition number of the stiffness matrix A scales as h^{-2}, hence, when the mesh size h is halved, the condition number of A increases by a factor of 4. Since the number of PCG iterations to solve $Ax = b$ depends on the square root of the condition number of A, halving the mesh size makes the number of iterations double. From Table 3, we can clearly notice that the number of PCG iterations approximately doubles as the mesh size is halved, as expected.

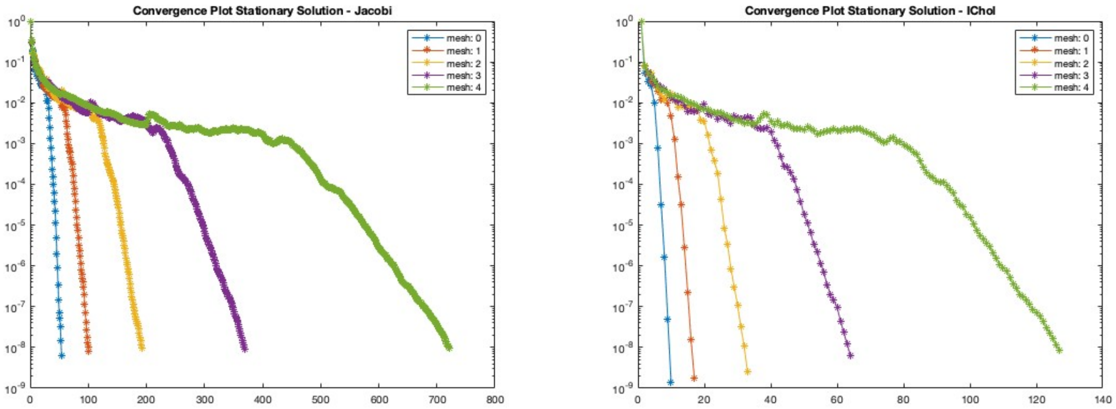


Figure 5: Semilogarithmic convergence plots for the relative residual norm for PCG iterations in the stationary solution linear system with `tol=1e-8` and Jacobi and incomplete Cholesky (`droptol=1e-2`) preconditioners.

Mesh	IterJ	IterIC
0	53	9
1	100	16
2	191	32
3	368	63
4	720	126

Table 3: Number of iterations for PCG convergence in the stationary solution linear system with `tol=1e-8` and Jacobi and incomplete Cholesky (`droptol=1e-2`) preconditioners.

3.5 Summary Table for the FEM convergence

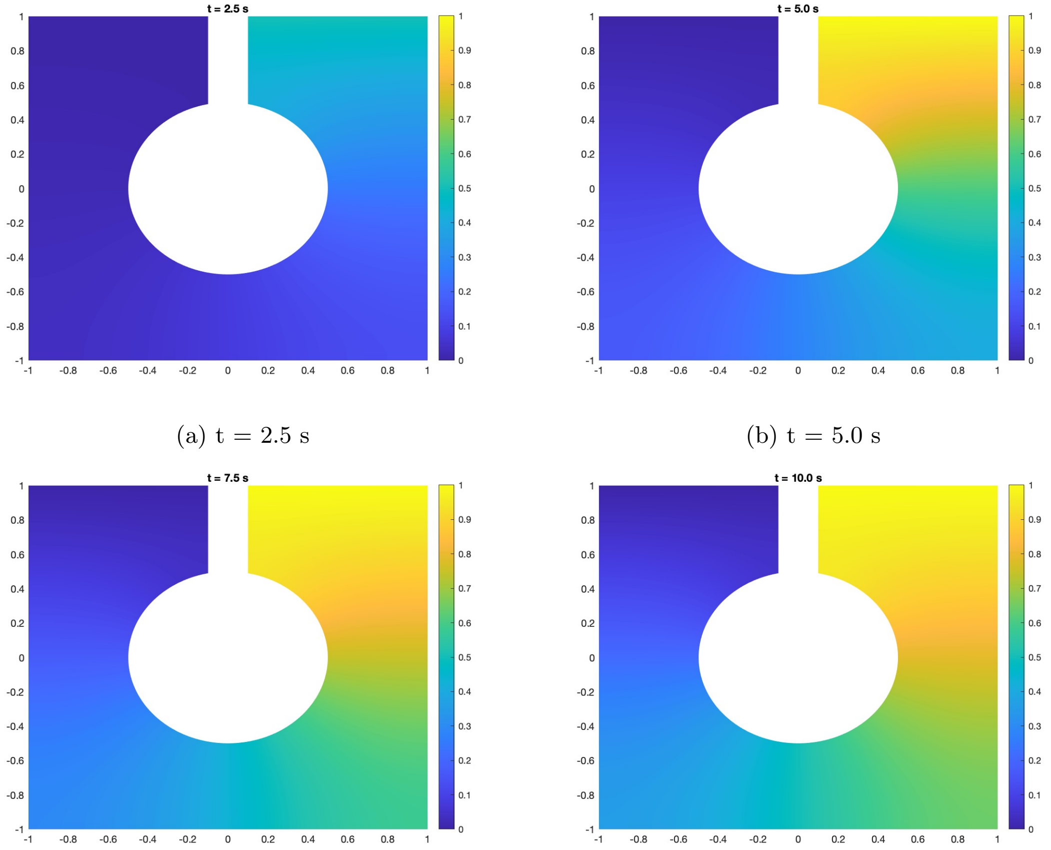
Table 4 contains, for each mesh refinement, the mesh size $h = \max_{K\in\mathcal{T}_h} \operatorname{diam}(K)$, the total error ε_k computed as in 2.5, and the ratio r_k obtained with Equation 1. We can note that for finer grids the total error decreases and the ratio r_k approaches 1, showing that the FEM is converging. Additionally, the error ε decreases as l^2, where l is the average mesh size (i.e., the characteristic edge length). Since the meshes are regular, $h \approx l$, and halving h reduces ε_k by approximately a factor of four, as we can observe from Table 4.

Mesh	h	ε_k	r_k
0	0.23582	0.0022872	-
1	0.12745	0.00060563	1.1032
2	0.06849	0.00014642	1.1944
3	0.034017	3.687e-05	0.97964
4	0.016835	8.7811e-06	1.0284

Table 4: Summary Table for the FEM convergence.

3.6 Solution Plot on the Whole Domain for Mesh4

Figure 6 illustrates the diffusion of the heat from Γ_{D2} to Γ_{D1} and the rise of temperature in time on the whole domain for the solution with Mesh4.



(a) t = 2.5 s

(b) t = 5.0 s

(c) t = 7.5 s

(d) t = 10.0 s

Figure 6: Solution Plot at four times on the whole domain for Mesh4.