

Deriverable 2: Adaptivity, transient problems and Navier-Stokes

Pol Navarro Pérez

July 1, 2025

Abstract

Abstract

This work addresses three computational problems using finite element methods (FEM). First, a gradient smoothing technique was implemented for a 1D boundary value problem, achieving accurate error estimations through the Zienkiewicz-Zhu error indicator and demonstrating effective adaptive mesh refinement. Second, the transient heat equation was solved using both explicit and implicit Euler schemes, with stability conditions analyzed and solutions computed to model the temperature evolution of a thin film. Finally, the flow problem was solved for Reynolds numbers up to $Re = 3560$, utilizing a Q2-Q1 element pair and the Picard iteration method. The streamlines reveal the evolution from laminar to complex flow structures, with secondary vortices emerging at higher Reynolds numbers. These results validate the robustness and versatility of FEM for diverse applications in mechanics and fluid dynamics.

1. Gradient smoothing

We start with a 1D boundary value problem

$$-u'' = s, \quad x \in (0, 1), \quad u(0) = u(1) = 0 \quad (1)$$

a) Gradient smoothing q

The first section is about implementing the gradient smoothing q of the FEM solution.

As seen in theory q is the L_2 projection of $\frac{du^h}{dx}$ in $V^h = \langle N_i \rangle$. The nodal values of this variable q are the solution of $Mq = \bar{b}$ with $m_{ij} = \int_{\Omega} \frac{du^h}{dx} N_i dx = \sum_e \int_{\Omega} (\sum_j u_j \frac{dN_j}{dx}) N_i dx$, or alternative $Mq = B\bar{u}$.

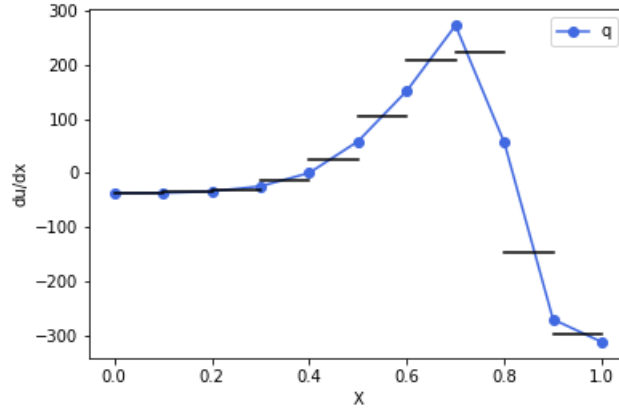


Figure 1: Plot of the gradient smoothing q compared to the differential of the approximation $\frac{du^h}{dx}$, using 10 elements.

As seen in 1 there is a good approximation using this derivative smoothing. The procedure has been computing the derivative for the nodal values u_i using the values u obtained with the system $Ku = f$. Then, using the already mentioned equation to obtain q . Remark that we have used the previously computed elemental matrices

$$M_e = \frac{h_e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

$$B_e = \frac{1}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}.$$

b) Zienkiewicz and Zhu error

Now, we have to evaluate what is the error of our procedure using the expression

$$|u^h - u|_{H^1(\Omega_e)}^2 \simeq \int_{\Omega_e} \left(\frac{du^h}{dx} - q \right)^2 dx \quad (2)$$

and the corresponding estimate of the density of the square error

$$\rho_e := \frac{1}{|\Omega_e|} |u^h - u|_{H^1(\Omega_e)}^2. \quad (3)$$

This is implemented computationally by implementing 2 for each Gauss-Legendre point and then the sum of all of them for each element. This is accomplished using an interpolation of q , to get it at the point we want. Eventually, by definition, the ZZ error is the square root of this error per element.

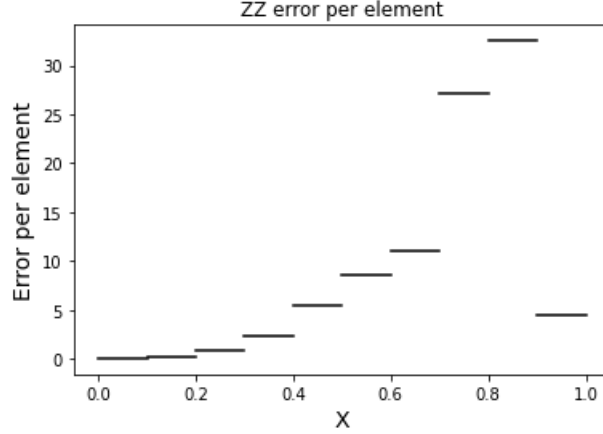


Figure 2: ZZ error for each element, using 10 elements.

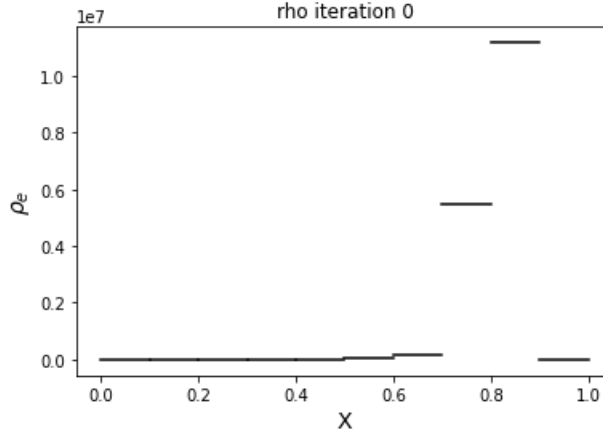


Figure 3: Error density of the square error for each element, using 10 elements.

c) Total error density

Given a density error $\rho_e \leq \tau$ we are asked to check that it implies that the total error density $\rho \leq \tau$, with $\rho := \frac{1}{|\Omega|} |u^h - u|_{H^1(\Omega)}^2$.

$$\begin{aligned} \rho_e &= \frac{1}{|\Omega_e|} \int_{\Omega_e} \left(\frac{du^h}{dx} - q \right)^2 dx, \\ \rho &= \frac{1}{|\Omega|} \int_{\Omega} \left(\frac{du^h}{dx} - q \right)^2 dx = \frac{1}{|\Omega|} \sum_e \int_{\Omega_e} \left(\frac{du^h}{dx} - q \right)^2 dx, \\ &\int_{\Omega_e} \left(\frac{du^h}{dx} - q \right)^2 dx = \rho_e |\Omega_e|, \\ \rho &= \frac{1}{|\Omega|} \sum_e \rho_e |\Omega_e|, \end{aligned}$$

$$\rho \leq \frac{1}{|\Omega|} \sum_e \tau |\Omega_e|,$$

$$\rho \leq \frac{\tau}{|\Omega|} \sum_e |\Omega_e| = \frac{\tau}{|\Omega|} |\Omega| = \tau.$$

d) e) Refinement of the mesh

As seen in previous sections, we have some error density that we could try to decrease by refining the mesh. This is implemented by splitting the elements with higher $\rho_e > \tau$, with τ a given parameter. Eventually, we can see how the error for every element is changing in each iteration.

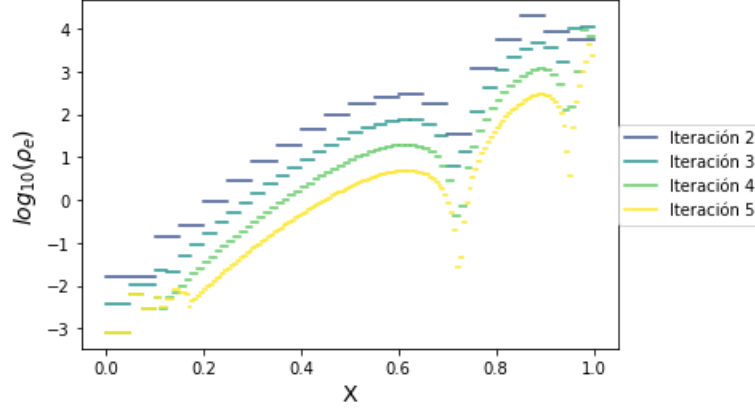


Figure 4: Plot of the logarithm of the error density per error for each iteration of the mesh refinement. We have used the parameter $\tau = 0.01$.

As seen in 4, the logarithm of the error density decreases as the element size gets smaller. In addition, we can repeat the previous parts for the 4 first iterations and see how the error map changes each iteration.

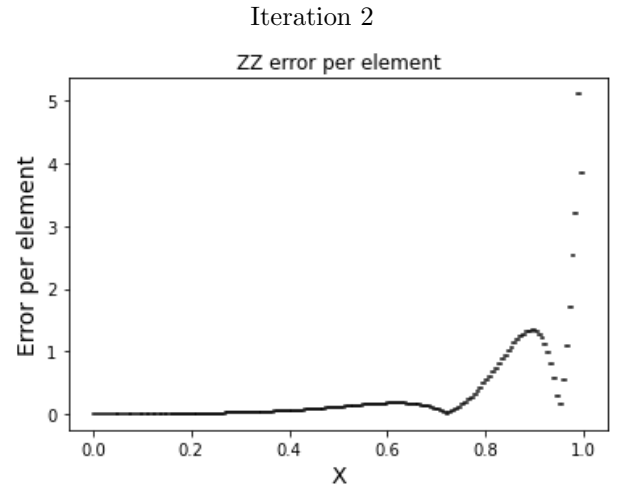
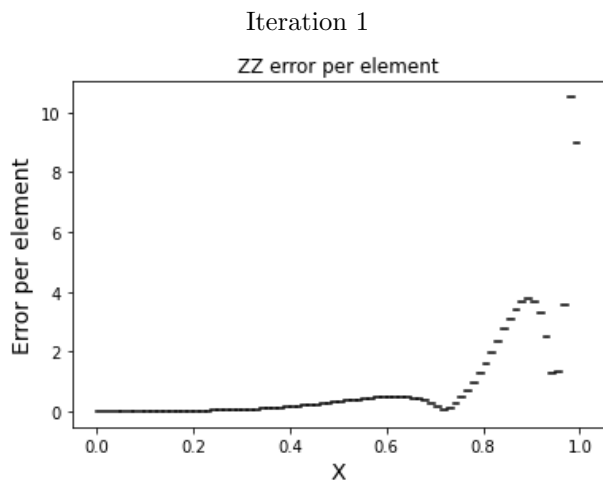
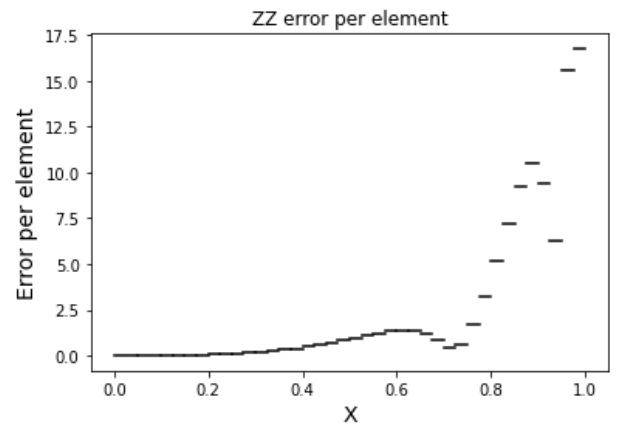
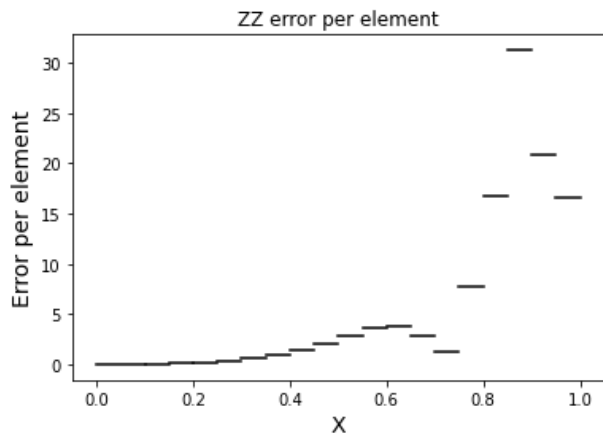


Figure 5: Error per element for the first 4 iterations of the mesh refinement, using an initial value of 10 elements for our domain.

As shown in Figure 5, the error is decreasing as we refine the mesh. We can observe how the y-axis has smaller values and the top height of our plot, this is the max error, is getting smaller as we iterate. This refinement strategy is based on the local error indicator ρ_e . The refinement process is driven by evaluating the error ρ_e for each element, therefore our h_2 may depend on the element. If ρ_e exceeds a predefined threshold τ , the corresponding element is marked for refinement. By refining only where necessary and smoothing the gradient, we achieve a more accurate solution while minimizing computational costs.

2. Heat equation as a transient problem

Considering a thin film of a certain material, initially at a temperature of 16°C , and then introduced into an oven, such that:

$$\frac{\partial u}{\partial t} = \Delta u, \quad \mathbf{x} \in \Omega = (0, 5)^2, \quad t > 0, \quad (4)$$

$$u(\mathbf{x}, t) = g(t), \quad \forall (\mathbf{x}, t) \in \partial\Omega \times (0, +\infty), \quad (5)$$

$$u(\mathbf{x}, 0) = 16, \quad \forall \mathbf{x} \in \Omega, \quad (6)$$

$$g(t) = -\frac{300}{t+3} + 116^\circ\text{C} \quad (7)$$

Representing the heating of the walls of the thin film as time evolves.

Weak form

The first step is to derive the weak form and the initial value problem resulting from the discretization.

1. Multiply the PDE by a test function $v \in H^1(\Omega)$ and integrate over Ω :

$$\int_{\Omega} \frac{\partial u}{\partial t} v \, d\Omega = \int_{\Omega} \Delta u v \, d\Omega.$$

2. Apply the integration by parts to the right-hand side:

$$\int_{\Omega} \Delta u v \, d\Omega = - \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega + \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, d\Gamma.$$

3. Since $u(x, t) = g(t)$ on $\partial\Omega$, and choosing $v = 0$ on $\partial\Omega$, the boundary term vanishes:

$$\int_{\partial\Omega} \frac{\partial u}{\partial n} v \, d\Gamma = 0.$$

4. The weak form becomes:

$$\int_{\Omega} \frac{\partial u}{\partial t} v \, d\Omega + \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega = 0, \quad \forall v \in H^1(\Omega). \quad (8)$$

5. Discretize $u(x, t)$ using a finite element approximation: $u_h(x, t) = \sum_{i=1}^n u_i(t) N_i(x)$,

where $N_i(x)$ are the basis functions.

6. Substitute $u_h(x, t)$ into the weak form and use $v = N_j(x)$:

$$\int_{\Omega} \frac{\partial}{\partial t} \left(\sum_{i=1}^n u_i(t) N_i(x) \right) N_j(x) \, d\Omega + \int_{\Omega} \nabla \left(\sum_{i=1}^n u_i(t) N_i(x) \right) \cdot \nabla N_j(x) \, d\Omega = 0.$$

7. This leads to the system of equations:

$$M \frac{d\mathbf{u}}{dt} + K \mathbf{u} = 0,$$

$$\text{where } M_{ij} = \int_{\Omega} N_i N_j \, d\Omega, \quad K_{ij} = \int_{\Omega} \nabla N_i \cdot \nabla N_j \, d\Omega, \quad \mathbf{u} = [u_1(t), u_2(t), \dots, u_n(t)]^T.$$

Remark that we have not applied the Dirichlet boundary conditions yet.

Stability condition for Eulers schemes

Assuming that the eigenvalues of the matrix $-M^{-1}K$ are denoted by $\{\lambda_i\}_{i=1}^n$, and they satisfy:

$$0 \geq \lambda_i \geq \lambda, \quad \text{with } \lambda = -\frac{\alpha}{h^2}, \quad \alpha > 0,$$

We analyze the stability conditions for both the explicit and implicit Euler time integration schemes. As seen in theory sessions we need two different conditions dependent if we used the lumped matrix M_L or not.

- **Euler Explicit:**

$M\bar{u}^{n+1} = M\bar{u}^n + \Delta T(\bar{f}^n - k\bar{u}^n)$, with $\frac{d\bar{u}}{dt} = M^{-1}(\bar{f}(t) - k\bar{u})$.
In our case $f(t) = 0$. Therefore, we get $\bar{u}^{n+1} = (I + \Delta T M^{-1} K)\bar{u}^n$.

We know that the eigenvalues, as proved in theory sessions, must satisfy

$$|1 + \Delta t \lambda_i| \leq 1$$

And we know $0 \geq \lambda_i \geq \lambda = \frac{-\alpha}{h^2}$, with $\alpha > 0$, this means

$$-2 \leq \Delta t \lambda \leq 0$$

Therefore,

$$\frac{\Delta t}{h^2} \leq \frac{2}{\alpha}$$

- **Euler Implicit:** $(M + \Delta t K)\bar{u}^{n+1} = M\bar{u}^n + \Delta t \bar{f}^{n+1}$

In this case, the matrix to get the eigenvalues is defined as:

$$(I - \Delta t M^{-1} K)^{-1}$$

, whose eigenvalues are

$$\mu_i = \frac{1}{1 - \Delta t \lambda_i}$$

Because of the definition of $\lambda_i \leq 0$, this implies the stability conditions will be always satisfied $\forall \Delta t$.

Solution using Euler explicit

In this section, we compute the solution of the above by using quadrilateral elements (Q1) and Euler explicit time integration.

Our problem consists of a rectangular thin film with heated walls that is placed inside an oven. We are provided with a function, $g(t)$, which describes the temperature of these walls over time. Once the film is introduced into the oven, heat will propagate throughout the entire body according to Equation 4. Therefore, if we want to know the time T , at which the walls reach a temperature of 66°C we just have to compute:

$$g(t) = 66^\circ\text{C} = -\frac{300}{t+3} + 166^\circ\text{C}$$

and we obtain a value of $T = 3$ a.u.

Another interesting thing to find is the evolution over time for a fixed point in our domain, in this case, $x = (2.5, 2.5)$. To implement the Euler explicit method, we use the lumped mass matrix M_L , which is defined as:

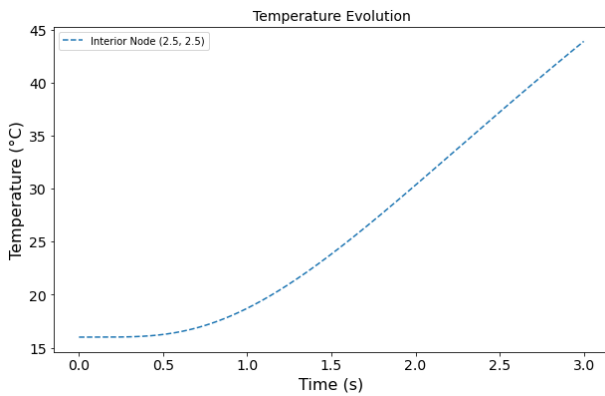
$$(M_L)_{ij} = \begin{cases} \sum_{k=1}^n M_{ik}, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

The lumped mass matrix M_L is diagonal, making its inverse computationally efficient to calculate. Additionally, since none of the eigenvalues of M are close to zero in our context, the stability condition for the Euler explicit method requires computing:

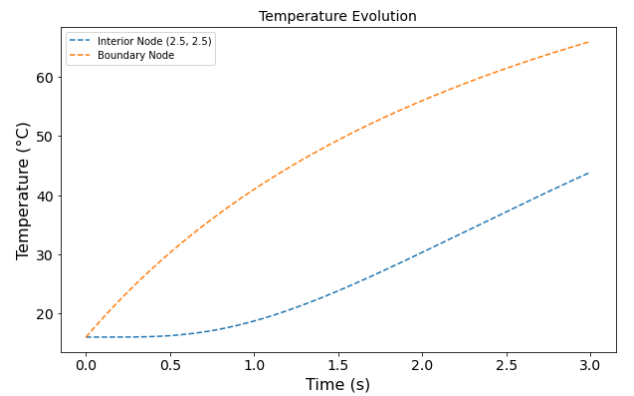
$$\lambda_{\max} = \max\{|\lambda| \mid \lambda \text{ is an eigenvalue of } -M_L^{-1}K\},$$

where K is the stiffness matrix. The time step Δt is then taken as:

$$\Delta t = \frac{2}{\lambda_{\max}}.$$



Temperate evolution for $x = (2.5, 2.5)$ using Euler explicit method.



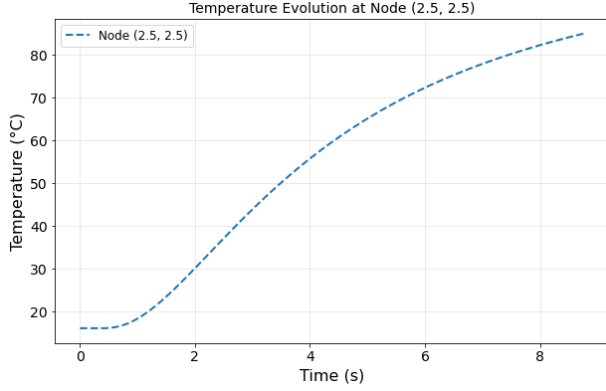
Temperature evolution for $x = (2.5, 2.5)$ and a boundary node $x = (0,0)$ using Euler explicit method.

Figure 6: Temperature evolution for some x values after computing the Euler explicit method.

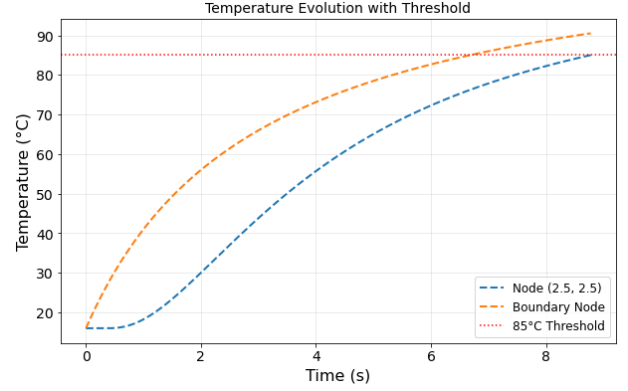
As Figure 6 shows, the boundary node behaves as expected, this is $g(t)$ see Equation 7. On the other hand, the node $x = (2.5, 2.5)$ heats up with a slight delay compared to the boundary node. At the beginning, the temperature remains constant for a short period.

Solution using Euler implicit

We have already introduced the procedure to implement the Euler implicit time integration. This time using a chosen small Δt we have obtained similar results than before.



Temperate evolution for $x = (2.5, 2.5)$ using Euler implicit method.



Temperature evolution for $x=(2.5, 2.5)$ and a boundary node $x = (0,0)$ using Euler implicit method.

Figure 7: Temperature evolution for some x values until the whole domain reaches $85^{\circ}C$.

We have obtained the value of $t = 8.78s$ to achieve the temperature $T = 85^{\circ}C$ in the whole domain.

3. Cavity flow problem and Reynolds number

To solve the cavity flow problem for high Reynolds numbers (Re), we employed an incremental strategy combined with the Picard iteration method. The Navier-Stokes equations were discretized using the finite element method (FEM) with the Q2-Q1 element pair, which provides quadratic velocity and linear pressure approximations, ensuring compatibility and numerical stability. The system of equations is expressed in matrix form as:

$$\begin{bmatrix} A(U^k) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} U^{k+1} \\ P^{k+1} \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix},$$

where $A(U^k)$ represents the linearized momentum equation, including contributions from viscosity and the convective term evaluated at the previous iteration U^k . B and B^T enforce the divergence-free condition for incompressible flow, and F accounts for body forces and boundary conditions. The Picard iteration solves this system iteratively until the relative error between successive velocity fields satisfies

$$\frac{\|U^{k+1} - U^k\|}{\|U^{k+1}\|} < \epsilon,$$

ensuring convergence.

To handle the increased nonlinearity and complexity at high Re , an incremental approach was used. Starting from a low Re , the Reynolds number was increased in fixed increments, and the solution from the previous step served as the initial guess for the next step. This strategy ensures numerical stability and accelerates convergence by leveraging the proximity of solutions for adjacent Re . Plots of the velocity magnitude and streamlines at each step illustrate the progressive intensification of flow features, including sharper gradients near boundaries and the emergence of secondary vortices. The method successfully resolved the flow up to $Re = 4600$, demonstrating the robustness of the approach.

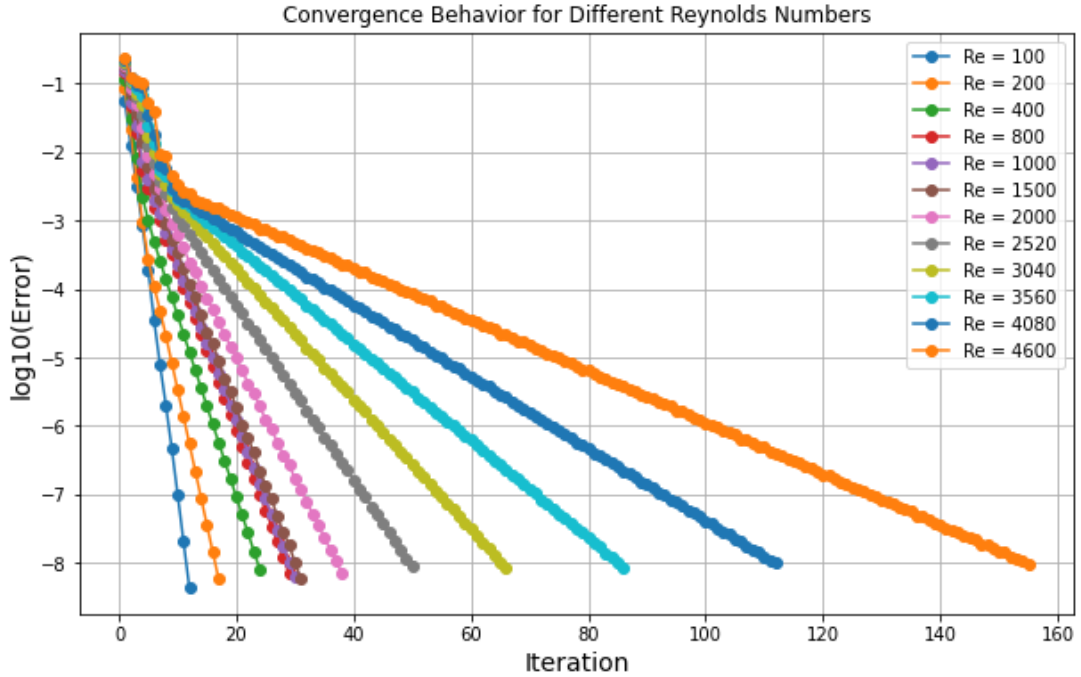


Figure 8: Logarithmic error decay during the Picard iteration process for different Reynolds numbers.

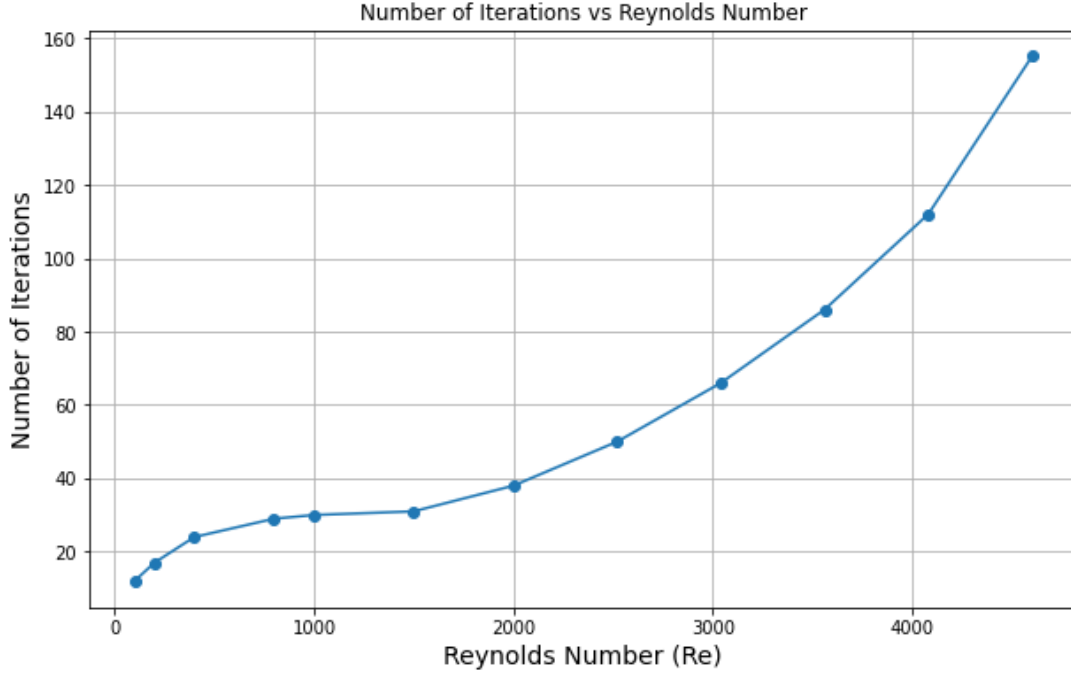


Figure 9: Number of iterations required for convergence as a function of the Reynolds number. Convergence was achieved when the relative error satisfied the tolerance criterion $\epsilon = 10^{-8}$.

Figure 8 shows the logarithmic decay of the error during the Picard iteration process for different Reynolds numbers. The error, represented as $\log_{10}(\text{Error})$, decreases steadily as the iterations progress, demonstrating the convergence of the iterative method. For higher Reynolds numbers, the initial error tends to be larger due to the increased nonlinearity of the system, but the method consistently achieves the tolerance criterion of $\epsilon = 10^{-8}$, ensuring reliable results.

On the other hand, Figure 9 highlights the number of iterations required to reach convergence for various Reynolds numbers. As expected, the computational effort increases with higher Reynolds numbers, reflecting the growing dominance of nonlinear convective effects. This behavior underscores the challenge of solving high- Re flows and the importance of the incremental strategy, which enables the method to maintain stability and convergence even at $Re = 4600$.

As a final result we plotted the streamlines once the code converged for each Reynolds number.

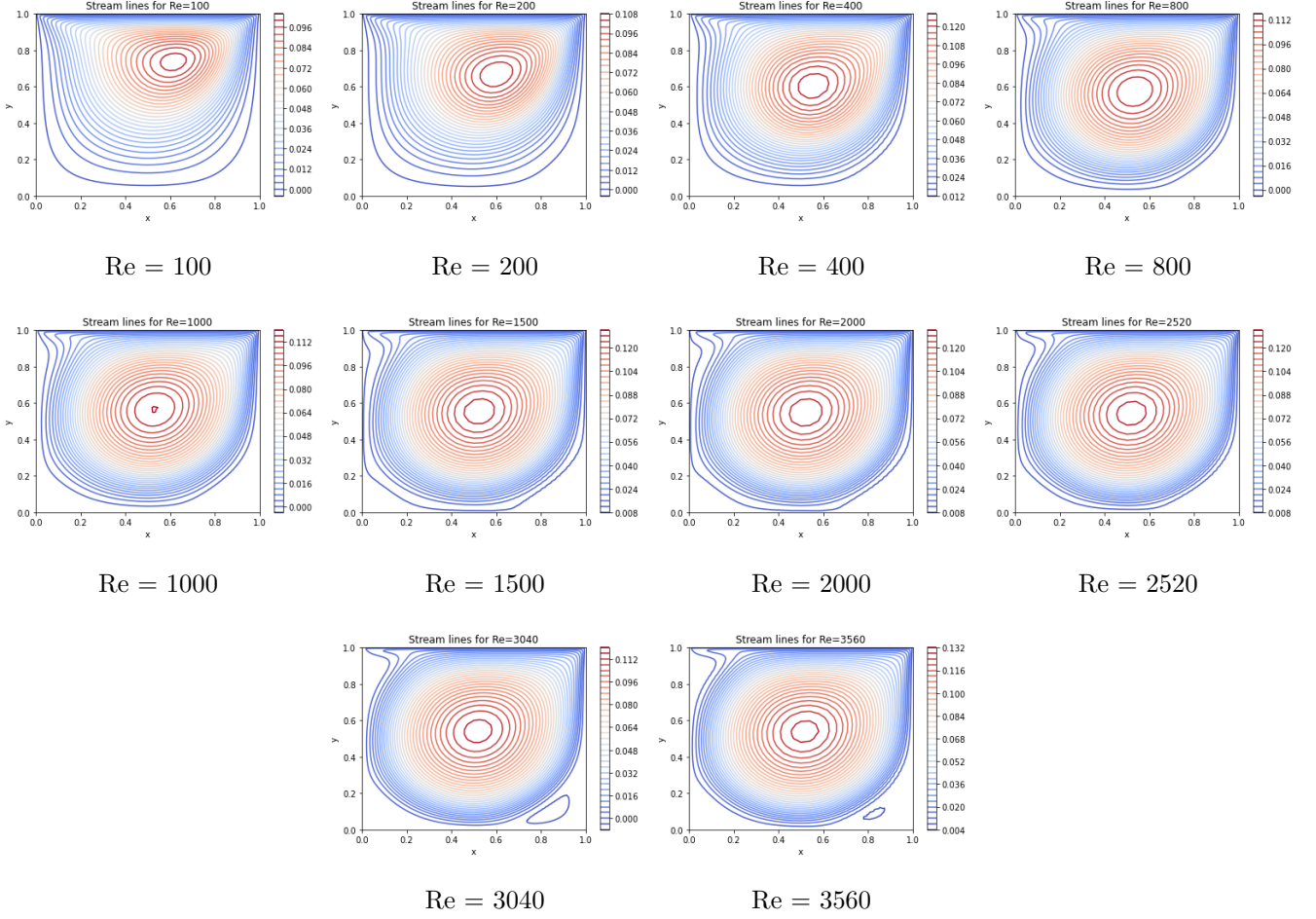


Figure 10: Streamline patterns for various Reynolds numbers, from $Re = 100$ to $Re = 3560$.

Using a computational grid of 50×50 elements and an iterative solution approach with a tolerance of 1×10^{-8} , the streamline plots (Figure 10) clearly demonstrate the evolution of flow structures as the Reynolds number increases:

- For $Re = 100$, the flow is characterized by a single, smooth central vortex. The streamlines remain well-ordered, with minimal distortions, indicating a laminar regime.
- At $Re = 200$ and $Re = 400$, the primary vortex undergoes a noticeable displacement and elongation toward the center of the cavity due to stronger inertial effects. However, secondary vortices are not yet present.
- For $Re = 800$ and $Re = 1000$, the main vortex becomes further displaced, and the streamlines show increasing asymmetry, indicating a growing complexity in the flow.
- Secondary vortices begin to emerge distinctly at $Re = 3040$, with a noticeable recirculation zone appearing near the bottom-right corner of the cavity. At $Re = 3560$, these secondary vortices become more pronounced, and the flow exhibits multiple regions of recirculation, indicative of the transition toward turbulence.

The use of a fine computational grid ($n_x = 50, n_y = 50$) provided sufficient resolution to capture these flow features accurately. Convergence was achieved across all cases despite the increasing nonlinearity of the convective term at higher Reynolds numbers. We conclude the finite element method is robust for solving the cavity flow problem and provide clear insights into the transition from laminar to more complex flow regimes.