STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, April 2026, pp 3238–3254.

Numerical methods for evolutionary problems considering advection dominated and control

Andrés Felipe Camelo Muñoz, Carlos Alberto Ramírez Vanegas, Guillermo Villa Martínez*

afkamelo@utp.edu.co, caramirez@utp.edu.co, gvilla@utp.edu.co
Department of Mathematics, Universidad Tecnológica de Pererira, Colombia

Abstract We study a time–dependent advection–diffusion equation with spatially varying advection and heterogeneous diffusivity under homogeneous Dirichlet conditions. The strong and weak formulations are derived and discretized by a conforming Galerkin finite element method, leading to the standard semi–discrete system with mass, stiffness, and advection matrices. Temporal integration is performed with an unconditionally stable implicit Euler scheme. A practical 2D assembly procedure based on a 7–point Gaussian quadrature is detailed. To assess discretization accuracy and mesh independence, we employ L^1, L^2, and H^1 norms together with the Grid Convergence Index (GCI), including Richardson extrapolation and an asymptotic range check via the convergence ratio. Beyond baseline simulations with elementwise constant advection, we formulate and solve a convex optimization problem for an advection field γ_{opt} that minimizes a quadratic functional and steers the solution within a prescribed subdomain. Numerical experiments on structured meshes ($n = 9, 18, 36$ per direction) demonstrate consistent convergence, CAR values near unity, and reduced dispersion when using γ_{opt}, while quantifying uncertainty through GCI. The results confirm the robustness and effectiveness of the proposed FEM framework for evolutionary advection–diffusion problems and provide a reproducible pathway for accuracy verification and transport-field design.

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

AMS 2010 subject classifications 65M60

DOI: 10.19139/soic-2310-5070-3153

1. Introduction

Let $\Omega \subset \mathbb{R}^2$ be a polygonal domain. We aim to solve the following advection–diffusion equation with medium-dependent advection [1, 2]:

$$\partial_t u + \gamma(\boldsymbol{x}) \cdot \nabla u - \nabla \cdot (\kappa(\boldsymbol{x}) \nabla u) = f(\boldsymbol{x}, t), \quad \text{in } \Omega \times (0, T], \tag{1}$$

subject to homogeneous Dirichlet boundary conditions and an initial condition $u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x})$. For the parabolic differential equation (1), the solution $u : \Omega \to \mathbb{R}$ must satisfy the PDE within the rectangular domain $\Omega \times (0, T]$, together with the boundary conditions:

$$u(\boldsymbol{x}) = 0, \quad \forall \boldsymbol{x} \in \partial\Omega,$$

which corresponds to homogeneous Dirichlet conditions assumed throughout this work. In this study, the strong and weak formulations of a class of partial differential equations are presented. The numerical approximation

---

*Correspondence to: Guillermo Villa Martínez (Email: gvilla@utp.edu.co). 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)

of evolutionary problems with variational formulations is applied to the modeling of the heat equation with a dominant advection term. The solution approach relies on transforming the PDE into a linear system through the finite element method, considering complete discretization in both time and spatial domains, together with initial and boundary conditions. Finally, several numerical experiments are carried out to analyze the effect of mesh refinement on the minimization of a functional within an arbitrary subdomain, demonstrating the effectiveness of the proposed methodology.

1.1. Notation

In the numerical solution of time-dependent advection–diffusion equations, several fundamental terms must be clearly defined. The test function $\varphi(\boldsymbol{x})$ plays a central role in the variational formulation, while $\partial_t u$ represents the temporal derivative of the unknown solution. The external influence of the system is modeled by the forcing term $f(\boldsymbol{x}, t)$. The advection term $\gamma(\boldsymbol{x})$, assumed constant within each element, governs directional transport, whereas the diffusivity coefficient $\kappa(\boldsymbol{x})$ accounts for spatial dispersion of the solution. For numerical integration, Gauss quadrature is employed using points ξ_i and weights ω_i. The operator ∇ is used to express gradients and fluxes in the spatial coordinates $(\boldsymbol{x}) = (x, y)$, enabling the finite element discretization to capture both transport and diffusion phenomena accurately.

1.2. Organization

Section (2) introduces the model problem advection–diffusion equation with spatially dependent advection under homogeneous Dirichlet conditions and derives its weak formulation. We then present the finite element discretization: the Galerkin setting, the semi-discrete matrix form, and the implicit Euler time integrator, followed by the $2D$ assembly procedure using a 7–point quadrature rule. Next, we define the error norms and the Grid Convergence Index (GCI) employed to assess mesh–dependent accuracy, establishing the metrics used in the convergence study. The numerical section (Section (4)) reports experiments on structured meshes, first with elementwise constant γ and subsequently with the optimized advection field γ_{opt}, including GCI-based verification and Richardson extrapolation. We conclude with a discussion of accuracy, stability, and computational costs, and outline directions for future extensions.

2. Basic algorithm and extensions

In this work, we study the advection–diffusion equation with spatially dependent advection under homogeneous Dirichlet conditions. The weak formulation is derived to transform the PDE into a variational problem suitable for the finite element method [3]. This framework provides the basis for the numerical experiments and the subsequent mesh convergence analysis.

2.1. Weak Formulation

Consistent with the homogeneous boundary conditions defined above, we search for a solution $u \in H_0^1(\Omega)$. Multiplying equation (1) by a test function $v \in V := H_0^1(\Omega)$ and integrating over Ω yields:

$$\int_\Omega \frac{\partial u}{\partial t} v \, d\boldsymbol{x} + \int_\Omega \gamma(\boldsymbol{x}) \cdot \nabla u \, v \, d\boldsymbol{x} - \int_\Omega \nabla \cdot (\kappa(\boldsymbol{x}) \nabla u) \, v \, d\boldsymbol{x} = \int_\Omega f v \, d\boldsymbol{x}. \tag{2}$$

Defining the bilinear form

$$(u, v) = \int_\Omega uv \, d\boldsymbol{x},$$

the diffusion-related term becomes

$$-\int_\Omega \nabla \cdot (\kappa(\boldsymbol{x}) \nabla u) \, v \, d\boldsymbol{x} = \int_\Omega \kappa(\boldsymbol{x}) \nabla u \cdot \nabla v \, d\boldsymbol{x} - \int_{\partial\Omega} \kappa(\boldsymbol{x}) \nabla u \cdot \vec{n} \, v \, d\boldsymbol{x}.$$

Since the test functions vanish on the boundary ($v = 0$ on $\partial\Omega$), the boundary integral disappears, leaving only the domain contribution.

$$b(u, v) = \int_{\Omega} (\kappa(\boldsymbol{x})\nabla u) \cdot \nabla v \, d\boldsymbol{x}. \tag{3}$$

The advection terms $\gamma(\boldsymbol{x})$ are given by:

$$a(u, v) = \int_{\Omega} \gamma(\boldsymbol{x}) \cdot \nabla u \, v \, d\boldsymbol{x}, \tag{4}$$

and the linear functional is defined as:

$$L(v) = \int_{\Omega} f(\boldsymbol{x}, t) v \, d\boldsymbol{x}. \tag{5}$$

Thus, the complete weak formulation reads [3]: find $u \in H_0^1(\Omega)$ such that

$$\left(\frac{\partial u}{\partial t}, v\right) + a(u, v) + b(u, v) = L(v), \quad \forall v \in H_0^1(\Omega). \tag{6}$$

This formulation provides the basis for deriving the nodal equations and assembling the linear system obtained through the finite element discretization of the domain [3, 4].

2.2. FEM for Parabolic Problems

The Finite Element Method (FEM) approximates the solution of PDEs by representing it as a linear combination of basis functions and solving the weak formulation of the problem. The domain Ω is subdivided into a finite set of elements $\{\Omega_i\}$, typically triangles or quadrilaterals, and the approximate solution is expressed as $u_h = \sum \alpha_i(t)\varphi_i(\boldsymbol{x})$, where φ_i are piecewise polynomial basis functions. These are continuous, low–degree functions with compact support [4], which simplifies the assembly of global matrices and ensures computational efficiency. The Galerkin formulation of equation (1) reads:

$$\left(\frac{\partial u_h}{\partial t}, v_h\right) + a(u_h, v_h) + b(u_h, v_h) = L(v_h), \quad \forall v_h \in V_h. \tag{7}$$

Expressing $u_h = \sum \alpha_i(t)\varphi_i(\boldsymbol{x})$, the matrix form is given by:

$$M\frac{\partial u_h}{\partial t} + (A + S)u_h = B, \tag{8}$$

where the global matrices are defined as:

$$M_{ij} = \int_{\Omega} \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x}) \, d\boldsymbol{x}, \qquad \text{(Mass matrix)} \tag{9}$$

$$S_{ij} = \int_{\Omega} \kappa(\boldsymbol{x})\nabla\varphi_i(\boldsymbol{x}) \cdot \nabla\varphi_j(\boldsymbol{x}) \, d\boldsymbol{x}, \qquad \text{(Stiffness matrix)} \tag{10}$$

$$A_{ij} = \int_{\Omega} \big(\gamma(\boldsymbol{x}) \cdot \nabla\varphi_i(\boldsymbol{x})\big)\varphi_j(\boldsymbol{x}) \, d\boldsymbol{x}, \qquad \text{(Advection matrix)} \tag{11}$$

$$B_i = \int_{\Omega} f(\boldsymbol{x}, t)\varphi_i(\boldsymbol{x}) \, d\boldsymbol{x}, \qquad \text{(Load vector)} \tag{12}$$

This formulation provides the algebraic system required for the numerical solution of parabolic advection–diffusion equations using FEM.

2.3. Implicit Euler Method

To numerically approximate the solution of the semi-discrete system (8), we apply the implicit Euler scheme for the temporal discretization. Let the time interval $[0, T]$ be divided into N_t steps of size $\Delta t = T/N_t$. For u_h^n denoting the approximation at time $t^n = n\Delta t$, the scheme reads:

$$M\frac{u_h^{n+1} - u_h^n}{\Delta t} + (A + S)u_h^{n+1} = B^{n+1}. \tag{13}$$

This leads to the following linear system to be solved at each time step:

$$\big(M + \Delta t(A + S)\big)u_h^{n+1} = Mu_h^n + \Delta t B^{n+1}. \tag{14}$$

The implicit Euler method is unconditionally stable for parabolic problems, making it suitable for advection–diffusion equations with temporal dynamics. However, it requires solving a linear system at every time step, which increases the computational cost compared to explicit schemes [5]. The stability and robustness of this approach ensure accurate results, even for relatively large time steps.

2.4. Assembling in 2D with 7–point quadrature

To describe any two-dimensional mesh, we define a coordinate matrix:

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}, \tag{15}$$

and a connectivity matrix that associates nodes to each element:

$$[T] = \begin{bmatrix} n_1^{(1)} & n_2^{(1)} & n_3^{(1)} \\ n_1^{(2)} & n_2^{(2)} & n_3^{(2)} \\ \vdots & \vdots & \vdots \\ n_1^{(m)} & n_2^{(m)} & n_3^{(m)} \end{bmatrix}, \tag{16}$$

where triangular P_1 elements are considered. The global approximation is expressed as:

$$u_h(\boldsymbol{x}, t) = \sum_{i=1}^{n} \alpha_i(t)\varphi_i(\boldsymbol{x}), \tag{17}$$

with $\varphi_i(\boldsymbol{x})$ the linear shape functions.

For each element e, the local matrices are computed using the 7–point Gaussian quadrature rule:

$$[M^e]_{ij} = \sum_{k=1}^{7} \omega_k\, \varphi_i(\xi_k, \eta_k)\, \varphi_j(\xi_k, \eta_k)\, |J|, \tag{18}$$

$$[S^e]_{ij} = \sum_{k=1}^{7} \omega_k\, \kappa(\xi_k, \eta_k)\, \nabla\varphi_i(\xi_k, \eta_k) \cdot \nabla\varphi_j(\xi_k, \eta_k)\, |J|, \tag{19}$$

$$[A^e]_{ij} = \sum_{k=1}^{7} \omega_k\, \left(\gamma(\xi_k, \eta_k) \cdot \nabla\varphi_i(\xi_k, \eta_k)\right) \varphi_j(\xi_k, \eta_k)\, |J|, \tag{20}$$

$$[B^e]_i = \sum_{k=1}^{7} \omega_k\, f(\xi_k, \eta_k, t)\, \varphi_i(\xi_k, \eta_k)\, |J|, \tag{21}$$

where (ξ_k, η_k) are the quadrature points, ω_k are the corresponding weights, and $|J|$ is the determinant of the Jacobian mapping from the reference element to the physical element.
Finally, the assembly step consists of inserting each M^e, S^e, A^e, B^e into their global counterparts:

$$M = \sum_{e=1}^{m} M^e, \quad S = \sum_{e=1}^{m} S^e, \quad A = \sum_{e=1}^{m} A^e, \quad B = \sum_{e=1}^{m} B^e.$$

Thus, the assembled system in 2D is:

$$M \frac{\partial u_h}{\partial t} + (A + S) u_h = B, \tag{22}$$

which represents the finite element discretization of the advection–diffusion equation with temporal dynamics, evaluated through high–order quadrature for greater accuracy.

3. Error norms definitions

Within the framework of the Grid Convergence Index (GCI), different mathematical norms are employed to quantify the differences between discrete solutions obtained on meshes of varying resolutions. The L^1 norm evaluates global deviations through a weighted sum of absolute values, while the L^2 norm, associated with the mass matrix, captures the quadratic distribution of errors and guarantees positive definiteness. The H^1 seminorm, linked to the stiffness matrix, reflects variations in the gradient and becomes particularly relevant in problems dominated by diffusion and advection phenomena. Together, these norms provide a rigorous framework to assess convergence, quantify discretization errors, and validate the asymptotic behavior of numerical solutions in time-dependent advection–diffusion equations.

3.1. L^1 norm

When implementing the Grid Convergence Index (GCI), we need to measure how the numerical solution u_h changes as the mesh is refined. To this end, a norm is required to quantify the difference between solutions obtained on different meshes. In particular, the L^1 norm is defined by:

$$||u_h||_{L^1(\Omega)} = \int_\Omega |u_h| d\boldsymbol{x}. \tag{23}$$

In order to compare two solutions obtained on different meshes, the coarser mesh h_1 is interpolated onto the finer mesh h_2, using:

$$||u_{h_1} - u_{h_2}||_{L^1(\Omega)} = \int_\Omega |u_{h_1} - u_{h_2}| d\boldsymbol{x}. \tag{24}$$

In practice, since Ω is discretized into finite elements, where:

$$u_h(\boldsymbol{x}, t) = \sum_{i=1}^{n} \alpha_i(t) \varphi_i(\boldsymbol{x}),$$

the integral is approximated by the quadrature weight associated with node i:

$$||u_h||_{L^1(\Omega)} = \sum_{i=1}^{n} |\alpha_i(t)| w_i, \tag{25}$$

in addition, the nodal weights are:

$$w_i = \int_\Omega \varphi_i(\boldsymbol{x}) d\boldsymbol{x}. \tag{26}$$

Since $\sum_{j=1}^{n} \varphi_j(\boldsymbol{x}) = 1$, the connection with the mass matrix is given by:

$$w_i = \sum_{j=1}^{n} \int_{\Omega} \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x}) d\boldsymbol{x} = \sum_{j=1}^{n} M_{ij}, \tag{27}$$

in vector form:

$$\boldsymbol{w} = M\mathbf{1}, \tag{28}$$

where $\mathbf{1}$ is a column vector of ones, therefore:

$$||u_h||_{L^1(\Omega)} = \sum_{i=1}^{n} |\alpha_i(t)| w_i = \sum_{i=1}^{n} |\alpha_i(t)| M\mathbf{1}. \tag{29}$$

This transforms the L^1 norm into a weighted sum of absolute values at quadrature points. When applying the GCI within the finite element method, the L^1 norm is employed to evaluate the discrete approximations on each mesh $u_{h_1}, u_{h_2}, u_{h_3}$. Instead of considering a pointwise value of the solution, the error is assessed in terms of the L^1 norm.

3.2. L^2 Norm

Let the $L^2(\Omega)$ norm be defined as:

$$||u_h||^2_{L^2(\Omega)} = \int_{\Omega} |u(\boldsymbol{x})|^2 d\boldsymbol{x}. \tag{30}$$

If $u_h(\boldsymbol{x}, t) = \sum_{i=1}^{n} \alpha_i(t)\varphi_i(\boldsymbol{x})$ is the finite element approximation with shape functions $\varphi_i(\boldsymbol{x})$, then:

$$||u_h||^2_{L^2(\Omega)} = \sum_{i,j=1}^{n} \alpha_i(t)\alpha_j(t) \int_{\Omega} \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x}) d\boldsymbol{x}. \tag{31}$$

The term

$$M_{ij} = \int_{\Omega} \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x}) d\boldsymbol{x}, \tag{32}$$

corresponds precisely to the mass matrix. In the two-dimensional case, it is computed using equation (9):

$$||u_h||^2_{L^2(\Omega)} = \alpha^T M \alpha. \tag{33}$$

3.2.1. Properties of the Mass Matrix Each entry M_{ij} is non–negative, since $\varphi_i, \varphi_j \geq 0$ within their support. The matrix M is symmetric and positive definite:

$$\alpha^T M \alpha = \int_{\Omega} \left(\sum_{i=1}^{n} \alpha_i(t)\varphi_i(\boldsymbol{x}) \right)^2 d\boldsymbol{x} \geq 0, \tag{34}$$

moreover, $\alpha^T M \alpha = 0$ if and only if $u_h(\boldsymbol{x}) = 0$.

3.3. Seminorm H^1

The energy norm induced by the diffusion operator is defined as

$$||\, u \,||^2_E = \int_{\Omega} \kappa(x) \mid \nabla u \mid^2 dx. \tag{35}$$

Using the stiffness matrix S defined in (10), this is computed discretely as

$$\| u_h \|_E^2 = \alpha^T S\alpha. \tag{36}$$

Note that for constant $\kappa = 1$, this coincides with the standard H^1 seminorm. Using the finite element method:

$$||u||^2_{H^1(\Omega)} = \sum_{i,j=1}^{n} \alpha_i(t)\alpha_j(t) \int_\Omega \nabla\varphi_i(\boldsymbol{x}) \cdot \nabla\varphi_j(\boldsymbol{x}) d\boldsymbol{x}. \tag{37}$$

The term:

$$S_{ij} = \int_\Omega \nabla\varphi_i(\boldsymbol{x}) \cdot \nabla\varphi_j(\boldsymbol{x}) d\boldsymbol{x}, \tag{38}$$

where the matrix S is computed using equations (19) in the one–dimensional case and equation (10) in the two–dimensional case, yields:

$$||u_h||^2_{H^1(\Omega)} = \alpha^T S\alpha. \tag{39}$$

The energy norm, or equivalently the $H^1(\Omega)$ seminorm, is directly computed through the stiffness matrix.

4. Numerical results

4.1. solution considering γ constant

Using a polygonal discretization of 9 squares in the domain $\Omega = [0, 1] \times [0, 1]$, we aim to solve the partial differential equation introduced in (1). Figure (1) shows a structured quadrilateral mesh (Q_1), composed of uniform rectangular cells. Each vertex corresponds to a FEM node, represented by red points. This mesh discretizes the domain $\Omega \subset \mathbb{R}^2$ and is used to assemble the global matrices of the linear system obtained by the finite element method, i.e., the mass, stiffness, advection matrices, and the load vector.

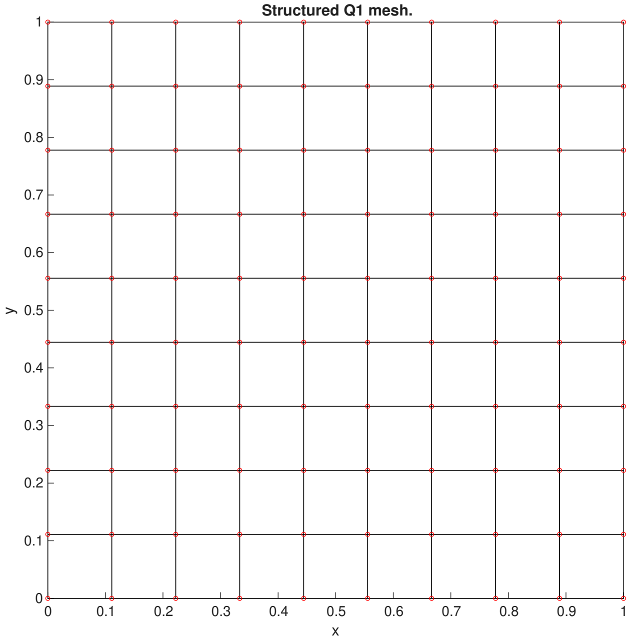


Figure 1. Structured Q1 mesh.

The initial condition is given by the expression:

$$u(x,y,0) = 10000e^{\dfrac{-((x-x_0)^2+(y-y_0)^2)}{2\sigma^2}},$$

Figure (2) depicts a surface describing a Gaussian bell centered at $x_0 = 0.5$, $y_0 = 0.5$ with $\sigma = 0.05$. The maximum amplitude occurs at the center, decaying rapidly toward the boundaries, under homogeneous Dirichlet boundary conditions. This initial condition simulates a localized distribution of concentration/temperature, serving as the input to the system to observe how the initial perturbation propagates through the medium under advection and diffusion.

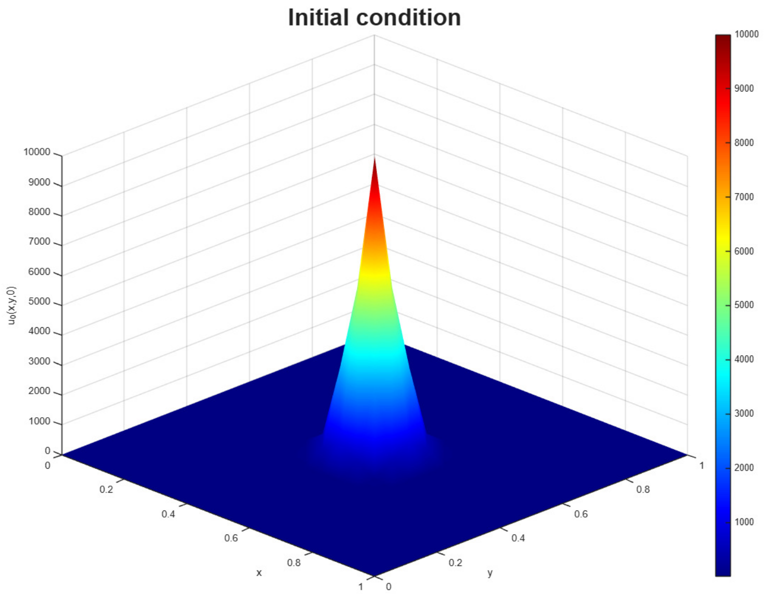


Figure 2. Initial condition $u(x,y,0)$.

A constant $\gamma(\boldsymbol{x})$ is considered, centered at each square element. Red arrows (vectors) are displayed at the center of each cell in Figure (3), indicating the direction and magnitude of the advection field. $\gamma(\boldsymbol{x})$ represents the convective transport in the equation. The arrows illustrate how the transported quantity is driven across the domain. Different directions affect the symmetry and displacement of the solution profile.

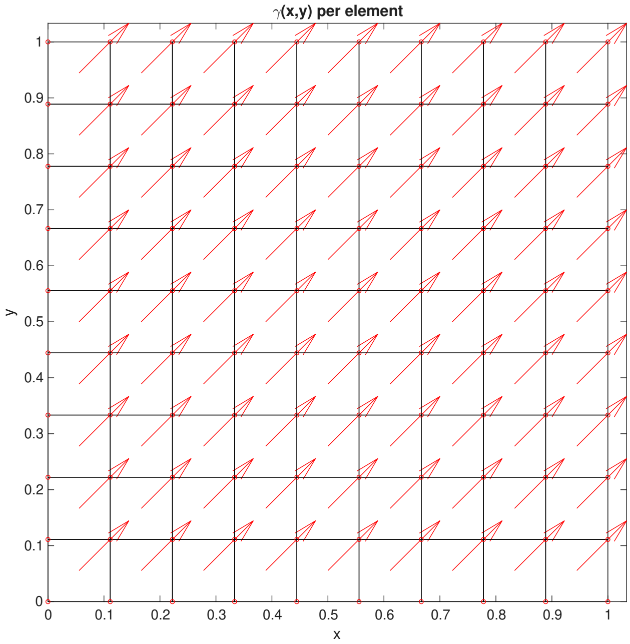


Figure 3. Advection per element $\gamma(\boldsymbol{x})$.

On the other hand, the values of $\kappa(\boldsymbol{x})$ are constant across elements (Figure (4)). $\kappa(\boldsymbol{x})$ models the heterogeneous diffusion of the medium.

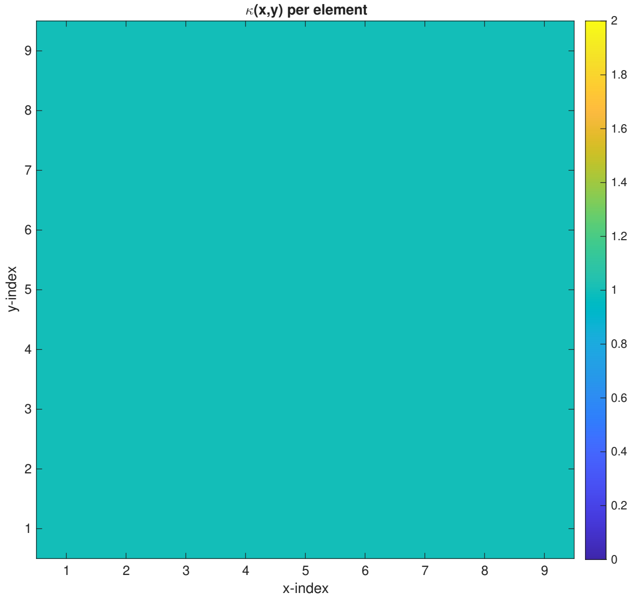


Figure 4. $\kappa(\boldsymbol{x})$ per element.

The solution $u(\boldsymbol{x}, t)$ from a top view exhibits the following distribution at different time instants $t = [0, \quad 0.25, \quad 0.5, \quad 0.75]$.

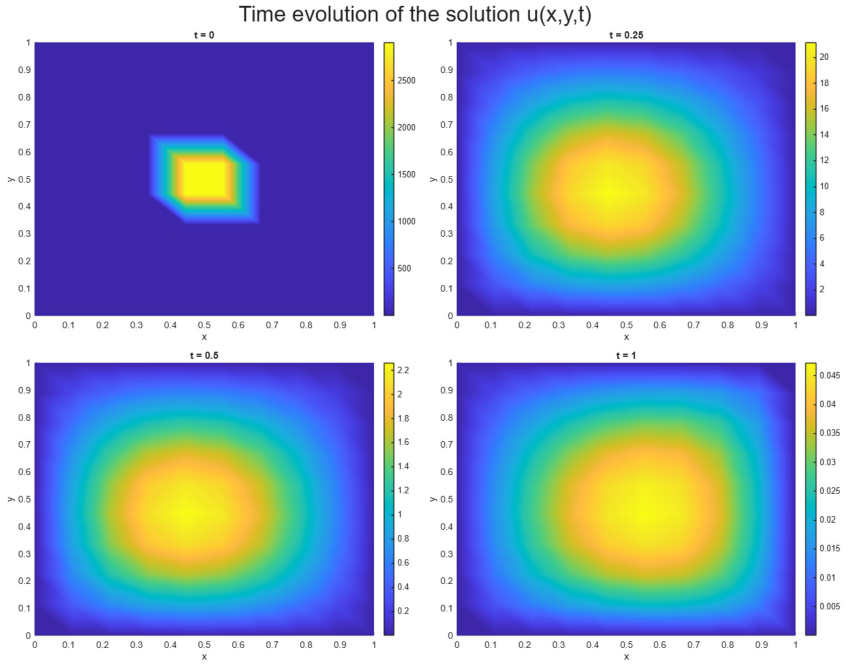


Figure 5. Solution $u(x, y, t)$ for $t = [0, \quad 0.25, \quad 0.5, \quad 0.75]$.

The behavior of the solution $u(\boldsymbol{x}, t)$ with elevation is illustrated in Figure (5), which displays the temporal evolution of the solution. It can be observed that the Gaussian bell is displaced (advection) and smoothed (diffusion) over time. This figure highlights the combined effect of both phenomena.

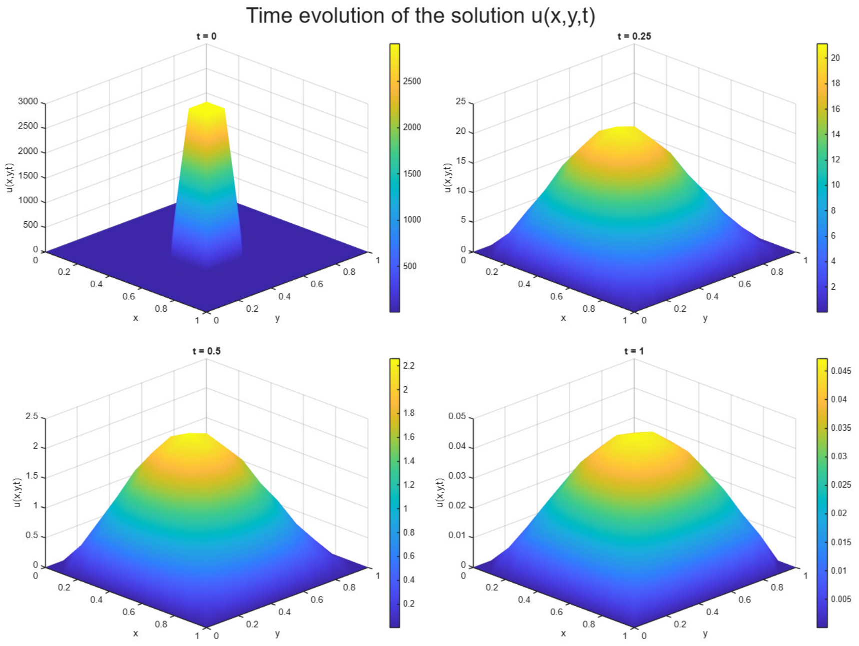


Figure 6. Solution $u(x, y, t)$ for $t = [0, \quad 0.25, \quad 0.5, \quad 0.75]$

The plots in Figure (6) allow visualization of the three-dimensional deformation of the distribution over time. This is useful for comparing the decay rate and directional propagation of the initial peak due to the advection field and constant diffusion.

4.2. Grid Convergence Index

We now aim to compute the L^1, L^2, and H^1 errors for three meshes (coarse h_1, medium h_2, fine h_3). To evaluate the Grid Convergence Index (GCI) using the three numerical solutions associated with each mesh, the following procedure is applied to the results of the numerical experiments:

1. Place the solutions on the same mesh. Let u_{h_k} denote the solution on mesh h_k (with $k = 1, 2, 3$ and $h_1 > h_2 > h_3$). To compare two solutions, they must lie in the same discrete space. This requires interpolating the solutions onto the fine mesh:

   $$P_{2\to 3} : u_{h_2} \to u_{h_3}, \quad P_{1\to 3} : u_{h_1} \to u_{h_3},$$

   so that the comparisons are performed on h_3:

   $$e_{32} = u_{h_3} - u_{h_2}, \quad e_{21} = u_{h_2} - u_{h_1}.$$

   In the finite element method it is common to employ non–nodal interpolation. In this work, interpolation was carried out using the *griddata* function in *Matlab*, e.g. $P_{1\to 3} : u_{h_1} \to u_{h_3} = griddata(x1, y1, u1, x3, y3,' linear')$.
2. Norms with mass and stiffness matrices. On the mesh where solutions are compared (fine mesh h_3), assemble:
   - Mass matrix M_{h_3} using equation (9).
   - Stiffness matrix S_{h_3} using equation (10).
3. Let e be the error vector on mesh h_3 (either e_{32} or e_{21}). Then:
   - $L^1(\Omega)$ norm:

     $$||e||_{L^1(\Omega)} = \sum_{i=1}^{n} |e| M\mathbf{1}. \tag{40}$$

   - L^2 norm:

     $$||e||_{L^2(\Omega)} = \sqrt{e^T M e}. \tag{41}$$

   - $H^1(\Omega)$ seminorm:

     $$||e||_{H^1(\Omega)} = \sqrt{e^T S e}. \tag{42}$$

4. Since the meshes are uniformly refined with ratio $r = \dfrac{h_k}{h_{k+1}}$ ($r = 2$), the following calculations are performed [9, 10]:
   - Relative error:

     $$RE = \left|\frac{e_{32}}{e_{21}}\right|, \quad RE \leq 1. \tag{43}$$

   - The observed order p in each norm can be estimated with error pairs:

     $$\rho = \frac{\ln\left(\dfrac{e_{13}}{e_{23}}\right)}{\ln(r)}. \tag{44}$$

5. With ρ, the GCI is computed using a safety factor Fs (see [**?**], where $1.25 \leq Fs \leq 3$). In this work, $Fs = 1.25$ was employed:

   $$GCI_{12} = Fs\left(\frac{|e_{21}|}{u_{h_2}(r^\rho - 1)}\right), \tag{45}$$

and

$$GCI_{23} = Fs\left(\frac{|e_{32}|}{u_{h_3}(r^{\rho}-1)}\right). \tag{46}$$

6. The asymptotic range of convergence is verified by computing the coefficient:

$$CAR = \frac{r^{\rho} GCI_{32}}{GCI_{21}} \approx 1, \tag{47}$$

which verifies that the solutions lie within the asymptotic convergence range. It is important to emphasize that the generalized Richardson extrapolation method must be applied within the monotonic convergence regime.

7. Finally, the exact solution is estimated using Richardson extrapolation:

$$u_{exact} \approx u_h(h \to 0) \approx u_{h_3} + \frac{u_{h_3} - u_{h_2}}{r^{\rho} - 1}. \tag{48}$$

4.2.1. γ constant per element The following presents the mesh–independence analysis for the numerical solution of the time–dependent advection–diffusion equation. For each function on each mesh, the solution at the final time $u(\boldsymbol{x}, t_f)$ was employed:

- $Fs = 1.25$
- $h1(N = 9) = \frac{1}{9}$
- $h2(N = 18) = \frac{1}{18}$
- $h3(N = 36) = \frac{1}{36}$
- $r21 = 2$
- $r32 = 2$

p	ρ	RE	$\|\|e_{extrap}\|\|_p$	$\|\|u_1\|\|_p$	$\|\|u_2\|\|_p$	$\|\|u_3\|\|_p$	GCI_{12}	GCI_{23}	R
L^1	3.1170	0.1153	0.0537	0.0485	0.0531	0.0536	1.4394%	0.1643%	0.9905
L^2	3.1674	0.1113	0.0652	0.0589	0.0645	0.0651	1.3744%	0.1516%	0.9909
H^1	1.2122	0.4315	0.2941	0.2671	0.2896	0.2916	15.4003%	6.5986%	0.9930

Table 1. GCI mesh–independence analysis.

4.3. *Function* $J(\gamma)$

We aim to optimize a vector field $\gamma(\boldsymbol{x}) = (\gamma_x, \gamma_y) \in \mathbb{R}^2$ in the domain $\Omega \in \mathbb{R}^2$, which is associated with the advection term in (1):

$$\gamma(\boldsymbol{x}) \cdot \nabla u = (\gamma_x, \gamma_y) \cdot \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) = \gamma_x \frac{\partial u}{\partial x} + \gamma_y \frac{\partial u}{\partial y}, \tag{49}$$

where $\gamma(\boldsymbol{x}) \cdot \nabla u$ represents a vector term projected onto the gradient of u. Furthermore, the goal is to find $\gamma(\boldsymbol{x})$ that minimizes the following functional [11]:

$$J(\gamma) = \frac{1}{2}\int_0^T \int_{\Omega} |\gamma(\boldsymbol{x})|^2 d\boldsymbol{x}dt + \frac{1}{2}\int_0^T \int_R |u(\boldsymbol{x}, t)|^2 d\boldsymbol{x}dt + \frac{1}{2}\int_{\Omega} |u(\boldsymbol{x}, t)|^2 d\boldsymbol{x}, \tag{50}$$

The objective is to control the dynamics of $u(\boldsymbol{x}, t)$ through the advection term, where the aim is to minimize:

$$J(\gamma_{opt}) = \min_{\gamma \in \mathbb{R}^2} J(\gamma). \tag{51}$$

Now, by considering random values for $\gamma(\boldsymbol{x})$, we aim to numerically demonstrate the convexity of the functional (50). This is achieved by computing a linear combination proposed in equation (14).

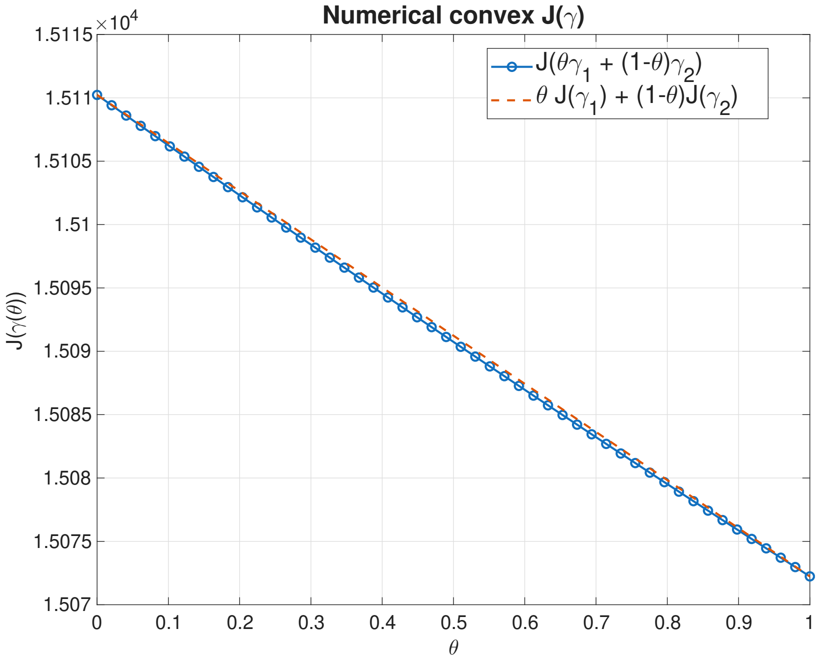
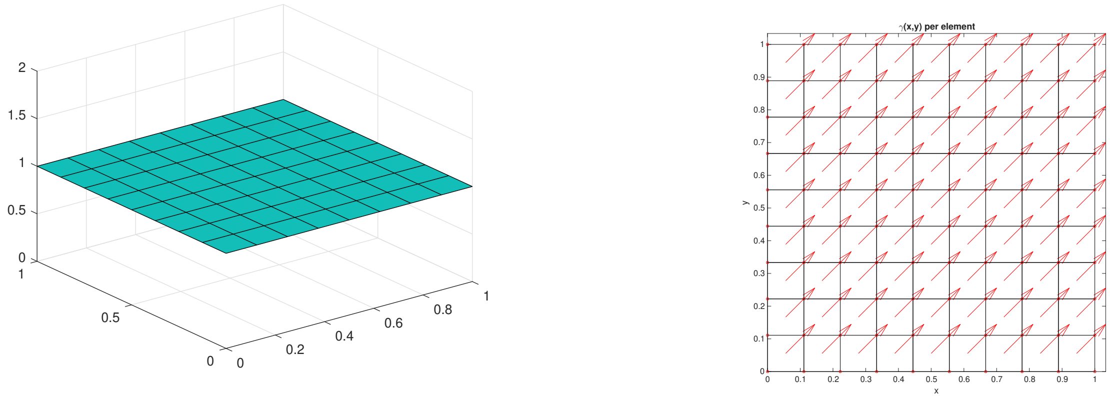


Figure 7. Behavior of the convex functional evaluated along different linear trajectories between two γ configurations.

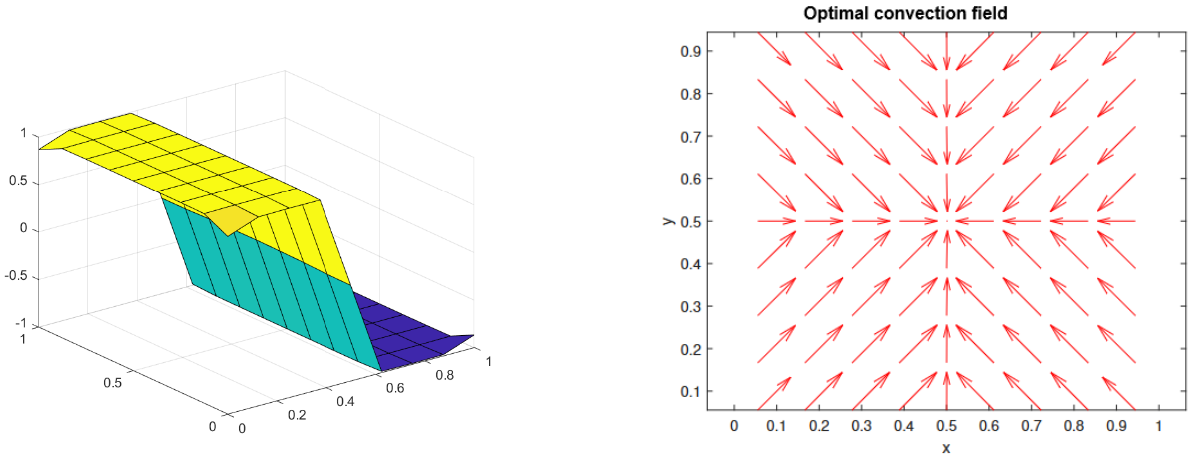
In the case of convex functionals, optimization exhibits a fundamental property: any local minimum coincides with the global minimum. This implies that the optimal values of the functional are found in a stable and unique manner within the feasible domain. In particular, when constraints or bounds are imposed on the domain, the optimal values tend to be located at the extremes or boundaries of such a domain, reflecting the geometric property of convexity. The values shown in Figure (7) were computed on a coarse mesh with $n = 9$ elements in each direction. The optimal value was found to be $J(\gamma(\boldsymbol{x})_{opt}) = 1.539823 \times 10^4$.

4.3.1. Solution considering γ_{opt} Discrete Optimization Implementation To minimize the functional defined in (50), the control variable $\gamma(x) = (\gamma_x, \gamma_y)$ is discretized using a piecewise constant approximation on each element Ω_e, consistent with the finite element approximation of the state variable. For the coarse mesh ($n = 9$), this results in a optimization problem with $2 \times 9^2 = 162$ unknowns. The functional includes an L^2 regularization term, $\frac{1}{2}\int_\Omega |\gamma(x)|^2 dx2$, which ensures the problem remains well-posed and bounded. The optimization was solved using the interior-point algorithm via MATLAB's fmincon. The stopping criteria were set to a function tolerance of 10^{-8} and a maximum of 500 iterations.The optimization was performed on the coarse mesh to reduce computational overhead, with a total execution time of 16.1203 seconds. The resulting optimal parameters γ_{opt} were then interpolated to finer meshes ($n = 18, 36$) for the verification analysis presented in Section 4.3.2. Initial condition centered at the origin $(0, 0)$. Optimization completed. Results saved. Total execution time: 16.1203 seconds. $J(\gamma_{opt}(\boldsymbol{x})) = 1.539823 \times 10^4$



(a) $\gamma(\boldsymbol{x})$.

(b) Initial vector field $\gamma(\boldsymbol{x})$.



(c) $\gamma(\boldsymbol{x})_{opt}$.

(d) Optimized vector field $\gamma(\boldsymbol{x})_{opt}$.

Figure 8. Initial advection field $\gamma(\boldsymbol{x})$ (a),(b) and optimized field $\gamma(\boldsymbol{x})_{opt}$ (c),(d).

In subfigures (a) and (b), the initial advection field is observed to be uniform and constant: the flat surface in (a) indicates that $\gamma(\boldsymbol{x})$ exhibits no local variations in magnitude. The vector field in (b) shows that all vectors have the same direction and magnitude, implying that the initial convective transport is homogeneous throughout the domain. This corresponds to a simplified initial configuration that does not capture the specific characteristics of problem (51) nor the geometry of the domain. In subfigure (c), the scalar result of the optimization for $\gamma(\boldsymbol{x})_{opt}$ is shown. The surface now varies compared to the initial case. This reflects that the optimization process introduced local variations in the magnitude of $\gamma(\boldsymbol{x})$ to minimize the associated functional and achieve improved transport effects. In subfigure (d), the optimized vector field shows a significant change: the vectors are no longer parallel nor equal in magnitude. A radial distribution is observed around the center of the domain, indicating that the transport has been adjusted to direct the flow toward the center. This vectorial pattern reflects the capability of the method to capture cavitation or localized dispersion phenomena, adapting the dynamics of the advection field to the PDE solution.

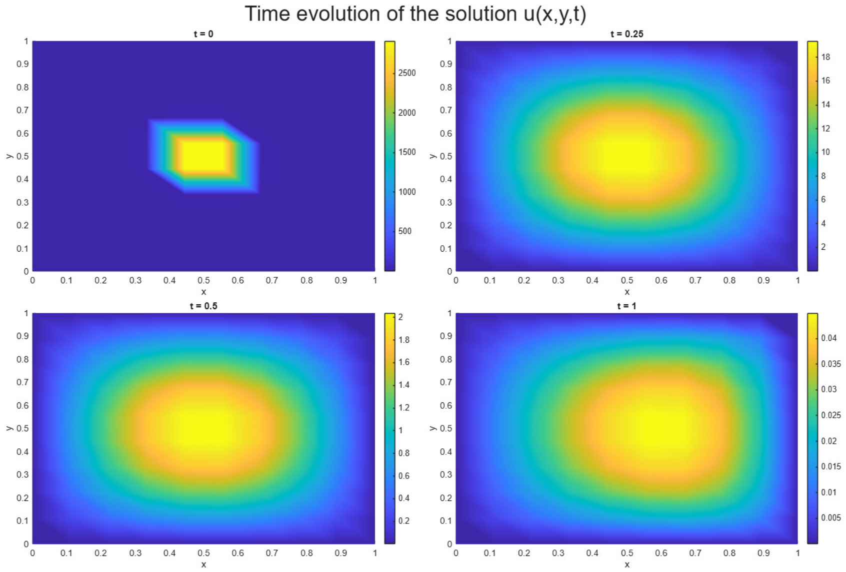


Figure 9. Solution $u(\boldsymbol{x}, t)$ for $t = [0, \quad 0.25, \quad 0.5, \quad 1]$ including $\gamma(\boldsymbol{x})_{opt}$.

Figure (9) shows the temporal evolution of the solution $u(\boldsymbol{x}, t)$ considering the case where the advection field is constant per element, but now employing the optimized values $\gamma(\boldsymbol{x})_{opt}$ obtained from the optimization process. The use of an optimized advection field enables controlling the propagation dynamics of the initial perturbation within a subdomain R. Whereas with the initial field the solution rapidly dispersed toward the domain boundaries, the optimized field $\gamma(\boldsymbol{x})_{opt}$ attenuates the expansion rate and maintains a more concentrated and stable profile over time. This confirms that the optimization process successfully reduces uncontrolled dispersion, enhances numerical robustness, and adjusts the transport dynamics according to the problem objectives.

4.3.2. GCI considering γ_{opt} For the following analysis, the considerations regarding the obtained solutions are as follows:

1. The differential equation (1) is solved on a coarse mesh ($N = 9$ elements in each direction).
2. The computation of the optimal values per element is performed by optimizing the functional (50).
3. With the obtained values in the advection field, these are extrapolated to the medium mesh ($N = 18$ elements in each direction), and the solution of (1) is computed with these interpolated values.
4. Subsequently, the values from the coarse mesh are interpolated to the fine mesh ($N = 36$ elements in each direction), and the solution of the differential equation is computed with these values.
5. The mesh–independence analysis is performed in the same way as in subsection (4.2.1).
6. Finally, the values of each term of the Grid Convergence Index are recorded according to the procedure described in Section (4.2).

In order to evaluate the numerical accuracy and the mesh independence of the solution, an analysis was carried out using the Grid Convergence Index (GCI). For this purpose, three spatial discretizations with systematic refinements were considered: $n = 9$, $n = 18$, and $n = 36$ elements per spatial direction, corresponding to coarse, intermediate, and fine meshes, respectively. The calculation was performed using the final–time solution $u(\boldsymbol{x}, t_f)$ with the optimized values $\gamma(\boldsymbol{x})_{opt}$.

p	ρ	RE	$\|\|e_{extrap}\|\|_p$	$\|\|u_1\|\|_p$	$\|\|u_2\|\|_p$	$\|\|u_3\|\|_p$	GCI_{12}	GCI_{23}	R
L^1	2.7176	0.1520	0.1509	0.1347	0.1485	0.1505	2.0944%	0.3141%	0.9865
L^2	2.7597	0.1477	0.1818	0.1629	0.1790	0.1814	1.9708%	0.2872%	0.9871
H^1	1.2122	0.4316	0.8194	0.7365	0.8016	0.8103	15.4254%	6.5859%	0.9892

Table 2. GCI mesh–independence analysis for γ_{opt}.

For the coarse mesh with $n = 9$, the following execution values were obtained: optimization completed, results stored, and a total execution time of 16.1203 seconds. Once the values of $\gamma(\boldsymbol{x})_{opt}$ were obtained on the coarse mesh, they were interpolated to the medium and fine meshes, yielding the equivalent values in each case. The transient state of the partial differential equation was then solved for each $\gamma(\boldsymbol{x})_{opt}$ on each mesh. Afterwards, interpolation of the solutions at the final time was carried out. The GCI method enables a quantitative estimation of the discretization error by comparing the solutions obtained on successively refined meshes. In this way, the asymptotic convergence behavior is verified, and the numerical uncertainty associated with the spatial discretization of the problem is quantified. The primary contribution of this work lies in the synergistic integration of a rigorous accuracy verification framework—using the Grid Convergence Index (GCI) across L^1, L^2, and H^1 norms—with a convex optimization approach for controlling evolutionary advection-diffusion dynamics1. Unlike traditional FEM studies that focus solely on discretization error, or existing PDE-constrained optimization works that often overlook the quantification of numerical uncertainty, this study bridges the gap by demonstrating that the optimized advection field (γ_{opt}) significantly mitigates dispersion while simultaneously satisfying strict asymptotic convergence criteria (CAR ≈ 1). This establishes a reproducible pipeline for designing transport fields where numerical reliability is explicitly guaranteed alongside control performance.

5. Future work

This study establishes a finite element framework for time–dependent advection–diffusion with mesh–verified accuracy via GCI and Richardson extrapolation, and explores transport-field design through the optimization of $\gamma(\boldsymbol{x})$. Several research directions arise naturally:

- **Discretization enhancements:** extend the spatial approximation to higher–order (P_k) and discontinuous Galerkin schemes, and incorporate stabilization, this will allow for the extension of the current optimization framework to high-Peclet number flows where standard Galerkin methods fail, (e.g., SUPG/GLS) for advection–dominated regimes; explore space–time FEM and higher–order time integrators (BDF2, Crank–Nicolson) with adaptive time stepping.
- **Adaptive strategies:** develop goal–oriented and a posteriori error estimators (in L^1, L^2, H^1) to drive h– and hp–adaptivity; compare uniform versus adaptive refinement in the GCI pipeline and analyze the effect on the observed order ρ and CAR.
- **Model generalization:** treat anisotropic and strongly heterogeneous $\kappa(\boldsymbol{x})$, curved geometries, and 3D domains; include reaction terms and nonlinear/state–dependent advection to assess robustness of the assembly and verification procedures.
- **Optimization and control:** formulate PDE–constrained optimization with adjoints for γ_{opt} under regularization and box constraints on subdomains R; compare deterministic versus data–assimilative formulations and study well–posedness and convexity beyond quadratic costs.
- **Uncertainty quantification:** incorporate stochastic γ and κ (random fields) and propagate uncertainty to GCI metrics; investigate Bayesian inversion of transport parameters from sparse observations.
- **Performance and reproducibility:** design parallel solvers with multigrid/preconditioned Krylov methods for the implicit systems; provide open benchmarks (manufactured solutions) and formalize convergence proofs linking monotone regimes to CAR ≈ 1.

These extensions aim to improve accuracy in advection–dominated settings, broaden applicability to complex media and geometries, and establish scalable, verifiable pipelines for simulation and control of evolutionary transport phenomena.

Acknowledgement

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

REFERENCES

1. Boštjan Mavrič, Božidar Šarler., *Equivalent-PDE based stabilization of strong-form meshless methods applied to advection-dominated problems.*, Engineering Analysis with Boundary Elements, vol. 113, pp 315-327 2020.
2. Xiu Ye, Shangyou Zhang *Two-Order Superconvergent CDG Finite Element Method for the Heat Equation on Triangular and Tetrahedral Meshes*, Journal of Computational and Applied Mathematics, vol. 404, 113878, 2022.
3. Lishen He, Albert J. Valocchi, C.A. Duarte, *An adaptive global–local generalized FEM for multiscale advection–diffusion problems*, Computer Methods in Applied Mechanics and Engineering, vol. 418, 116548, 2024.
4. Fabio Nobile , Thomas Trigo Trindade, *Petrov–Galerkin Dynamical Low Rank Approximation: SUPG stabilisation of advection-dominated problems*, Computer Methods in Applied Mechanics and Engineering, vol. 433, 117495, 2025.
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. Bürger, Raimund and Gavilán, Elvis and Inzunza, Daniel and Mulet, Pep and Villada, Luis Miguel, *Exploring a Convection–Diffusion–Reaction Model of the Propagation of Forest Fires: Computation of Risk Maps for Heterogeneous Environments*, Mathematics MDPI, vol. 8(10), 1674, 2020.
7. Sharma Rohit, Yadav Om Prakash, *The Evolution of Finite Element Approaches in Reaction-Diffusion Modeling*, Archives of Computational Methods in Engineering, vol. 32, pp. 2745–2766, 2025.
8. Lishen He, Albert J. Valocchi, C.A. Duarte, *A transient global-local generalized FEM for parabolic and hyperbolic PDEs with multi-space/time scales*, Journal of Computational Physics, vol. 488, 112179, 2023.
9. Rutger A. Biezemans, Claude Le Bris, Frédéric Legoll, Alexei Lozinski, *MsFEM for advection-dominated problems in heterogeneous media: Stabilization via nonconforming variants*, Computer Methods in Applied Mechanics and Engineering, vol. 433, 117496, 2025.
10. Tatar, Farzad and Brighenti, Roberto, *Structural topology optimization based on an immersed FEM Level-Set method*, Structural and Multidisciplinary Optimization, vol. 68, 149, 2025.
11. Thawait Amit K, Tandaiya Parag, Jain Prakash C, Chandy Abhilash J, *Numerical investigation of aerothermoelastic characteristics of a thin heated panel in high supersonic and hypersonic flow*, Acta Mechanica, vol. 234, pp. 2269–2291, 2023.