Stabilised finite element methods for the convection-diffusion equation

Finite Element Methods for PDEs

Candidate Number: 1092278

1 Introduction

Convective–diffusive processes are ubiquitous in science and engineering. The convection–diffusion equation governs the transport of heat and mass in fluids, as well as species concentrations in complex reactive systems and turbulence parameters for computational fluid dynamics solvers [1, 9]. A further motivation for developing robust solution methods to this equation lies in its close connection to the steady-state Navier–Stokes momentum equation, which can be seen as a convection–diffusion equation for the velocity field u with a physics-based source term.[1] As such, understanding numerical methods for the convection–diffusion equation provides an essential foundation for tackling the Navier–Stokes equations.

This paper reviews finite element methods for solving the convection–diffusion equation. Finite element methods are well-suited for approximating solutions in complex geometries and for systematically reducing error. However, in convection-dominated regimes, the presence of boundary layers, where derivatives become large, suggests that numerical methods will struggle to resolve the solution in this layer. Indeed this is the case, but to make matters worse, the standard Galerkin finite element method can introduce spurious oscillations that propagate throughout the domain, corrupting the solution even far from the layer. Grid refinement can mitigate this to some extent, but it is not always practical. This motivates the use of stabilised methods, which aim to suppress numerical oscillations and provide accurate solutions even on relatively coarse meshes [10, 5].

In the remainder of this report, we explore both standard and stabilised finite element methods for solving the steady-state convection–diffusion equation. We begin with the necessary background on the analytical behaviour of the equation, after which we formulate the problem variationally and prove the well-posedness of this form. We then implement the standard Galerkin finite element method and demonstrate its oscillatory behaviour in convection-dominated regimes. To address this, we introduce the stabilised Streamline Upwind Petrov–Galerkin (SUPG) method. Theoretical error bounds are obtained for both the standard Galerkin and SUPG method, and a numerical study is performed on a two model problem to compare theoretical error bounds with experiments. We conclude with a summary and possible extensions.

[1] An example being the inviscid source term $\nabla p + \rho g$, the pressure and gravitational body force.

2 Governing Equation and Analytical Behaviour

In this report we focus on the steady-state convection-diffusion equation for transport in a fluid medium. The general form, with constant diffusivity, is given by

$$-\epsilon\nabla^2 u(x) + \vec{w}(x)\cdot\nabla u(x) = f(x), \quad x \in \Omega \subset \mathbb{R}^n, \tag{1}$$

where $\epsilon > 0$ is the diffusion coefficient, $u(x)$ is the scalar field of interest, $\vec{w}(x)$ is the given fluid vector field, and $f(x)$ is the source term [10]. Note that $\nabla u = \left(\frac{\partial u}{\partial x_n}, ..., \frac{\partial u}{\partial x_n}\right)^T$ denotes the gradient operator, and $\nabla^2 u = \frac{\partial^2 u}{\partial x_1^2} + ... + \frac{\partial^2 u}{\partial x_n^2}$ denotes the Laplacian operator. We are interested in transport in an incompressible fluid medium, hence we assume that $\nabla \cdot \vec{w} \equiv 0$. Without loss of generality, we assume throughout this paper that $\vec{w}(x) = \mathcal{O}(1)$[2].

Equation (1) is supplemented with the Dirichlet and Neumann boundary conditions

$$u = g_D(x), \quad x \in \Gamma_D, \qquad \frac{\partial u}{\partial \vec{n}} = g_N(x), \quad x \in \Gamma_N, \tag{2}$$

with $\Gamma_D \cup \Gamma_N = \Gamma = \partial\Omega$.

We subdivide the domain boundary $\Gamma = \partial\Omega$ according to the direction of the vector field $\vec{w}$ and outward pointing boundary normal $\vec{n}$:

$$\begin{aligned} \text{Outflow:} \quad & \Gamma^+ := \{x \text{ on } \Gamma : \vec{w}\cdot\vec{n} > 0\} \\ \text{Wall:} \quad & \Gamma^0 := \{x \text{ on } \Gamma : \vec{w}\cdot\vec{n} = 0\} \\ \text{Inflow:} \quad & \Gamma^- := \{x \text{ on } \Gamma : \vec{w}\cdot\vec{n} < 0\} \end{aligned}$$

The first term in (1) models diffusion of the quantity u through the domain, while the second term represents convection of u along the streamlines of the velocity field $\vec{w}$. In many physical applications, the diffusion parameter ϵ is very small, often on the order of $\mathcal{O}(10^{-5})$ [1], leading to convection-dominated behavior. Equation (1) is frequently referred to in the literature as a 'singularly perturbed' boundary value problem because its behaviour is fundamentally different compared to the case when $\epsilon = 0$ [10].

As an example, consider

$$\begin{gathered} -\epsilon u''(x) + wu'(x) = 0, \quad x \in (0,1) \\ u(0) = b, \quad u(1) = c \end{gathered}$$

[2] We make this assumption to simplify analysis when investigating $\epsilon \ll 1$.

With $\epsilon, w \in \mathbb{R}$, $0 < \epsilon \ll w$. When $\epsilon = 0$, the reduced equation becomes $u'(x) = 0$, whose general solution $u = Ax$ cannot in general satisfy both boundary conditions unless $b = c$. However, when $\epsilon > 0$ but small, the second-order term becomes significant only in a narrow region near the outflow boundary at $x = 1$. In this *boundary layer*, the solution rapidly adjusts to satisfy the boundary condition, and its width is asymptotically $\mathcal{O}(\epsilon)$ [10].

In higher-dimensional domains ($n > 1$), convection still dominates at leading order. Formally, in the limit $\epsilon \to 0$ and under the assumption that cross–stream gradients remain small compared to streamwise variations, one may approximate the full convection–diffusion operator along each streamline s by a one–dimensional balance. Writing the Laplacian in curvilinear coordinates yields both streamwise $\partial^2/\partial s^2$ and transverse $\partial^2/\partial \eta^2$ terms, as well as curvature corrections. Neglecting the latter (and assuming weak cross–stream diffusion) gives the leading–order equation along the streamline:

$$-\epsilon \frac{\partial^2 u}{\partial s^2} + w_s \frac{\partial u}{\partial s} \approx f, \tag{3}$$

where $w_s = \vec{w} \cdot \vec{t}$. This is a formal or asymptotic reduction, not an exact equivalence: the full $\nabla^2 u$ remains isotropic and retains transverse and geometric terms that are omitted here. Nonetheless, this one–dimensional model provides a useful approximation for the dominant balance in the convection–dominated regime.

Additional dimensions introduce transverse boundary–layer phenomena beyond the purely streamwise layer seen in one dimension. Asymptotic analysis is performed in Appendix A to show that streamwise layers (parallel to streamlines) occur at outflow boundaries with width $\mathcal{O}(\epsilon)$, cross-stream layers occur where flow runs tangentially to a surface with width $\mathcal{O}(\sqrt{\epsilon})$, and mixed layers of either type can occur where distinct values of u meet via advection in the domain interior, with the type depending on the angle of the conincident velocity vectors.

Since the interplay between convection and diffusion is crucial, we introduce the global Péclet number

$$\mathcal{P} := \frac{\|\vec{w}\|_\infty L}{2\epsilon}, \qquad \|\cdot\|_\infty := \sup_{x \in \Omega}(\cdot),$$

following [6]. High values of $\mathcal{P}$ indicate convection-dominated regimes in which the boundary, shear, or interior layers described above become extremely thin and exhibit steep gradients that challenge numerical resolution.

3 Variational Formulation

In this section we explain how to derive the variational formulation of (1) and demonstrate its well-posedness.

3.1 The Formulation

Let Ω be a bounded open set in $\mathbb{R}^n$, with Lipschitz boundary $\Gamma = \partial\Omega$. Assume that $\vec{w}, f \in C(\overline{\Omega})$. We seek a solution $u \in U$ satisfying (1).

We multiply (1) by a test function $v \in V$ and integrate over Ω,

$$\int_\Omega -\epsilon(\nabla^2 u)v + \int_\Omega (\vec{w} \cdot \nabla u)v = \int_\Omega fv. \tag{4}$$

We lower the regularity requirement on our function u by integrating the first term by parts, enabling efficient approximation when we later discretise the solution into a finite element basis,

$$\epsilon \int_\Omega \nabla u \cdot \nabla v \, d\Omega - \int_\Gamma v(\nabla u \cdot \vec{n}) \, d\Gamma + \int_\Omega v(\vec{w} \cdot \nabla u) \, d\Omega = \int_\Omega fv \, d\Omega. \tag{5}$$

Next, we use linearity of the integral to split the boundary term into the contributions of the Dirichlet and Neumann sections. We know the value of u on Γ_D, so we restrict the test functions to those satisfying $v = 0$ on Γ_D, whilst imposing $u = G_D$ on Γ_D. After moving the Neumann term to the right hand side,

$$\epsilon \int_\Omega \nabla u \cdot \nabla v \, d\Omega + \int_\Omega v(\vec{w} \cdot \nabla u) \, d\Omega = \int_\Omega fv \, d\Omega + \int_{\Gamma_N} vg_N \, d\Gamma, \tag{6}$$

where $g_N = \frac{\partial u}{\partial n} = (\nabla u \cdot \vec{n})$ on Γ_N.

We require that (6) holds for all test functions $v \in W$, providing an alternative definition for our solution u.[3]

Since the left hand side (6) is linear in the arguments u and v, which follows from the linearity of the gradient and integral operators, we may abstractly write it as the bilinear form

$$a(u, v) \coloneqq \epsilon \int_\Omega \nabla u \cdot \nabla v \, d\Omega + \int_\Omega v(\vec{w} \cdot \nabla u) \, d\Omega. \tag{7}$$

The right hand side is linear in v, so we write it as the linear functional

$$F(v) \coloneqq \int_\Omega fv \, d\Omega + \int_{\Gamma_N} vg_N \, d\Gamma \tag{8}$$

[3] Note that if u satisfies (1), it automatically satisfies (6), but (6) also admits solutions which are not twice- or even once-continuously differentiable over the whole of Ω, so we have increased the space of admissible solutions.

The variational formulation is then:
Find $u \in U$ such that

$$a(u, v) = F(v), \qquad \forall v \in W \tag{9}$$

3.2 Function Spaces

We now define the function spaces U and W that u and v belong to. We use the standard Galerkin method, so U and W are closed subspaces of the same ambient function space (the underlying space unrestricted by boundary restrictions) [7].

We define the L^2-inner product (for scalar or vector valued functions) over arbitrary closed domain Θ,

$$(u, v)_\Theta := \int_\Theta u \cdot v \, d\Theta, \tag{10}$$

which induces the L^2 norm, $\|v\|_\Theta = \sqrt{\int_\Theta v \cdot v \, d\Theta}$. We usually omit the domain because it is clear from context. We say that a function belongs to $L^2(\Theta)$ if its L^2 norm is finite on Θ.

We also define the H^1 Sobolev inner product over arbitrary closed domain Θ,

$$(u, v)_{H^1(\Theta)} := (u, v) + (\nabla u, \nabla v), \tag{11}$$

which induces the H^1 norm $\|v\|_{H^1(\Theta)} := (\|v\|^2 + \|\nabla v\|^2)^{1/2}$, and the H^1 seminorm $|v|_1 := (\|v\|)^{1/2}$. We say that a function belongs to $H^1(\Theta)$ if its H^1 norm is finite on Θ, which is equivalent to its v and ∇v L^2 norms being finite.

We note that (6) is a collection of L^2 inner products, and by the Cauchy schwarz inequality, requires that v, ∇v, and ∇u are in $L^2(\Omega)$. Since we seek u and v in closed subspaces of the same underlying space, we require that both u and v belong to the H^1 Sobolev space.

Putting things together, the ambient space is $V = H^1(\Omega)$, and using the restrictions we set on u and v earlier, the trial space is

$$U = V_D = H^1_D(\Omega) := \{u \in H^1(\Omega) : u = g_D \text{ on } \Gamma_D\}, \tag{12}$$

which is a closed affine subspace of V, and

$$W = V_0 = H^1_0(\Omega) := \{u \in H^1(\Omega) : u = 0 \text{ on } \Gamma_D\}, \tag{13}$$

which is a closed affine subspace of V (a translation of V_0) determined by the Dirichlet data.

3.3 Existence and Uniqueness

We now have our weak formulation, but before we tackle finding solutions to the problem we should ensure that a unique, stable solution exists. This is the bedrock upon which the finite element method convergence guarantees lie. We will use the Lax-Milgram Lemma, which applies to bilinear forms which are not necessarily symmetric [7].

Lemma 3.1 (Lax-Milgram Lemma)**.** *Let V be a closed subspace of a Hilbert space H. Let $a : H \times H \to \mathbb{R}$ be a continuous bilinear form which is coercive on V. Let $F \in V^*$. Then the following problem has a unique, stable solution:*

$$\textit{Find } u \in V \textit{ such that } a(u,v) = F(v), \textit{ for all } v \in V$$

To use Lemma 3.1 we must show that the bilinear form $a : H \times H \to \mathbb{R}$ is both coercive and bounded, and that $F \in V^*$, which is equivalent to showing that F is a bounded linear functional. We assume that $u = 0$ holds on a subset of Γ_D with nonzero measure. [4]

3.3.1 Coercivity of a.

Henceforth we assume that the homogeneous Dirichlet boundary Γ_D contains the entire inflow portion $\Gamma_D \supseteq \Gamma^-$. Hence test functions $v \in V_0$ satisfy $v = 0$ on Γ^-. Under this assumption and incompressibility, the bilinear form $a(u,v)$ is coercive on $V_0 \subset H^1(\Omega)$ with coercivity constant ϵ:

$$a(v,v) \;\geq\; \epsilon \|\nabla v\|^2, \quad \forall\, v \in V_0. \tag{14}$$

Proof:
Starting from

$$a(v,v) = \epsilon \int_\Omega |\nabla v|^2 \, d\Omega + \int_\Omega (\vec{w} \cdot \nabla v)\, v \, d\Omega, \tag{15}$$

we rewrite the convective term via the divergence theorem. Using the product rule, $2\,(\vec{w} \cdot \nabla v)\, v = \nabla \cdot (v^2\, \vec{w}) - v^2\, (\nabla \cdot \vec{w})$ and applying $\nabla \cdot \vec{w} = 0$, we get

$$\int_\Omega (\vec{w} \cdot \nabla v)\, v \, d\Omega = \tfrac{1}{2} \int_\Omega \nabla \cdot (v^2\, \vec{w})\, d\Omega. \tag{16}$$

[4] By linearity, one can always transform the equation to achieve this.

Then we apply the divergence theorem

$$\int_{\Omega}(\vec{w}\cdot\nabla v)\,v = \tfrac{1}{2}\int_{\Gamma} v^2\,(\vec{w}\cdot n)\,d\Gamma = \tfrac{1}{2}\Bigg(\int_{\Gamma^+} v^2\,(\vec{w}\cdot n)\,d\Gamma + \underbrace{\int_{\Gamma^-} v^2\,(\vec{w}\cdot n)\,d\Gamma}_{\text{0, by assumption}}\Bigg) \;\geq 0. \quad (17)$$

Hence,

$$a(v,v) = \epsilon\|\nabla v\|^2 + \tfrac{1}{2}\int_{\Gamma^+} v^2\,(\vec{w}\cdot n)\,d\Gamma \;\geq\; \epsilon\,\|\nabla v\|^2, \quad (18)$$

as required. □

3.3.2 Norm Equivalence and Selection

Using the Poincare-Friedrichs inequality (see Appendix C)

$$\|v\|_{L^2(\Omega)} \;\leq\; K\,\|\nabla v\|_{L^2(\Omega)}, \quad \forall\, v \in V_0,$$

one obtains the bound

$$\|v\|_{H^1(\Omega)} = \big(\|v\| + \|\nabla v\|\big)^{1/2} \;\leq\; \sqrt{1+K^2}\,\|\nabla v\|.$$

Thus controlling the H^1–seminorm $\|\nabla v\|$ automatically controls the full H^1–norm.

We can also define the quasi-optimal "energy" norm

$$\|v\|_E = \sqrt{a(v,v)} = \sqrt{\epsilon}\,\|\nabla v\|.$$

Since $\|v\|_E \sim \|\nabla v\|$, all three norms $\|\nabla v\|$, $\|v\|_{H^1}$, and $\|v\|_E$ are equivalent on V_0. For simplicity, we will measure errors in the H^1–seminorm, $\|\nabla(u-u_h)\|$, throughout.

3.3.3 Boundedness of $a(u,v)$

The bilinear form, $a(u,v)$, is bounded with constant $\epsilon + \|\vec{w}\|_{L^\infty}$. That is,

$$|a(u,v)| \leq (\epsilon + K\|\vec{w}\|_\infty)\|\nabla u\|\|\nabla v\| \quad (19)$$

The proof is straightforward successive applications of the Cauchy-Schwarz inequality, and is given in Appendix D.

3.3.4 Boundedness of $F(v)$

The linear form, $F(v)$, is bounded with constant $\|f\|_{L^2(\Omega)} + \epsilon K_\Gamma \|g_n\|_{L^2(\Gamma_N)}$, where K_Γ is a constant that depends on the geometry of the domain boundary. The proof is standard and is given in Appendix D.

By Lax–Milgram, the variational problem (9) is well-posed: there exists a unique solution $u \in V$ that depends continuously on the data (f, g_D, g_N), providing a stable foundation for the error and convergence analyses that follow.

4 The Standard Galerkin Method

In this section we show how the standard Galerkin method can fail to provide accurate results. Error analysis is performed to understand why this is the case.

4.1 Galerkin Approximation

In its current form, (9) is posed in an infinite-dimensional function space. To cast it into a finite-dimensional form that we can solve, we approximate the trial and test functions, u and v, in a finite-dimensional subspace $V^h \subset V\ \big(= H^1(\Omega)\big)$. We select a linearly independent basis $V^h = \text{span}\{\phi_1, \phi_2, ..., \phi_n\}$, in which the approximate solution can be uniquely determined by a finite set of parameters.

Mathematically, we can express $u_h, v_h \in V^h$ as an expansion in our selected basis:

$$u_h = \sum_{j=1}^{n} u_j \phi_j, \ \ u_i \in \mathbb{R}, \qquad v_h = \sum_{i=1}^{n} v_i \phi_i, \ \ v_i \in \mathbb{R} \tag{20}$$

Expressing our variation problem in the finite-dimensional subspace V^h, we have:
Find u_h in V^h such that

$$a(u_h, v_h) = F(v_h), \quad \forall v_h \in V^h. \tag{21}$$

By the same coercivity and continuity arguments as (9), Lax-Milgram applies on the finite-dimensional subspace V^h, so the discrete problem (21) has a unique, stable solution u_h. Expanding v_h in our variation problem and using the linearity of a and F:

$$\sum_{i=1}^{n} v_i a(u_h, \phi_i) = \sum_{i=1}^{n} v_i F(\phi_i) \tag{22}$$

Since this must hold for all v, we can express this as $a(u_h, \phi_i) = F(\phi_i),\ \forall i \in [1, n]$. Then we expand u_h:

$$a(\sum_{j=1}^{n} u_j \phi_j, \phi_i) = \sum_{j=1}^{n} u_j a(\phi_j, \phi_i) = F(\phi_i), \qquad \forall i \in [1, n] \tag{23}$$

Which is easily recognisable in matrix formulation $AU = F$. Explicitly:

$$AU := \begin{pmatrix} (a(\phi_1, \phi_1)) & ... & a(\phi_1, \phi_n) \\ \vdots & & \vdots \\ a(\phi_n, \phi_1)) & ... & a(\phi_n, \phi_n) \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} F(\phi_1) \\ \vdots \\ F(\phi_n) \end{pmatrix} =: F, \tag{24}$$

where A is the stiffness matrix, F is the load vector, and U is the vector of coefficients uniquely determining the Galerkin approximation u_h in the chosen basis.

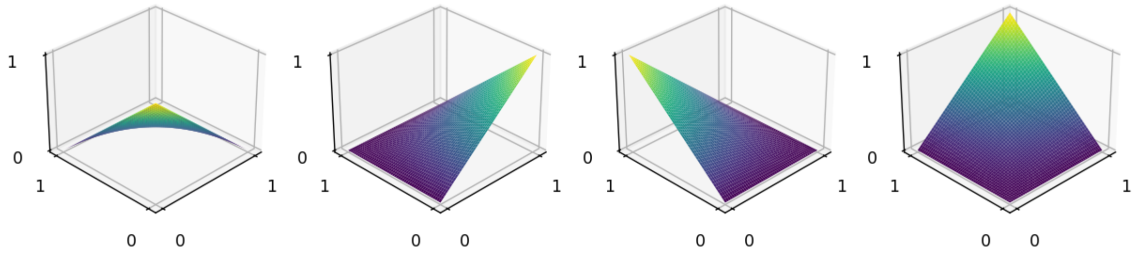

Figure 1: Basis functions for the Q_1 reference element.

4.2 Implementation

The choice of basis $\{\phi_i\}_{i=1}^n$ has a direct impact on accuracy and computational cost. We seek basis functions that have good approximation properties; lead to sparse system matrices; are easy to implement and integrate; and align naturally with our domain geometry.

This report considers both one- and two-dimensional examples. Here, we describe the more complex two-dimensional case. The one-dimensional implementation is briefly described in Appendix H. Let $\Omega \subset \mathbb{R}^2$ be a bounded domain. We partition $\Omega = [-1, 1]^2$ into a mesh $\mathcal{T}_h$ composed of non-overlapping elements. We construct a Cartesian mesh with uniform spacing h in both directions:

$$h = \frac{x_{\max} - x_{\min}}{N_x} = \frac{y_{\max} - y_{\min}}{N_y}, \quad \text{with } N_x, N_y \in \mathbb{N}.$$

Each cell in $\mathcal{T}_h$ is an axis-aligned square, with edges parallel to the coordinate axes. This regular structure simplifies both the basis definition and the assembly process.

On each element $K \in \mathcal{T}_h$, we define local finite element spaces. We use bilinear quadrilateral elements, denoted $\mathbf{Q}_1$, which are continuous and span bilinear polynomials on each cell. Rather than defining basis functions on every element individually, we define them on a reference element $\hat{K} = [0, 1]^2$ and map to each cell.

The affine map $F_K : \hat{K} \to K$ takes a point $\hat{x}$ to $x = F_K(\hat{x})$ via scaling and translation. For a Cartesian mesh with uniform spacing, this mapping simplifies to a scaling by h and a shift.

The basis functions on the reference element,

$$\hat{\phi}_1(x, y) = (1 - x)(1 - y), \quad \hat{\phi}_2(x, y) = x(1 - y),$$
$$\hat{\phi}_3(x, y) = xy, \quad \hat{\phi}_4(x, y) = (1 - x)y$$

are plotted in Figure 1. These form a nodal basis for the element K and are

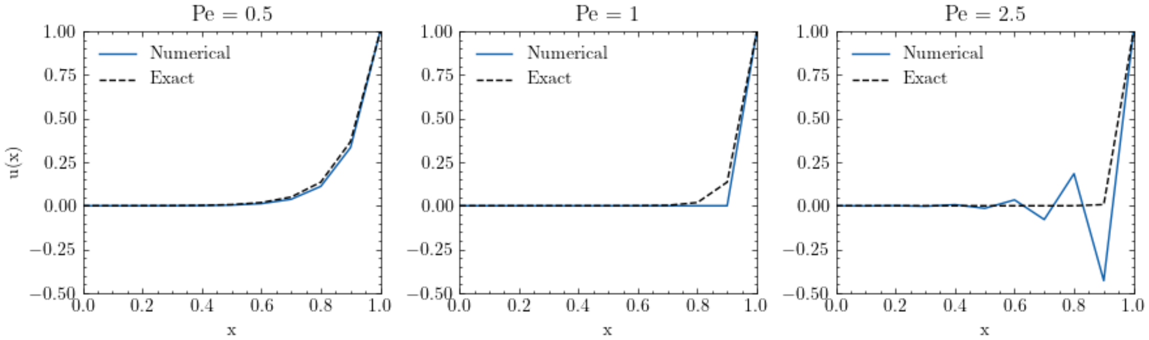

Figure 2: Numerical vs exact solution of (25) using a linear Galerkin approximation with varying Péclet number.

nonzero only on elements connected to the corresponding node, leading to sparse system matrices.

Now we have defined our mesh and finite element basis, the weak formulation requires assembling a global stiffness matrix A and load vector F from local element contributions, defined in (24). The steps and pseudo-code are provided in Appendix H.

4.3 Standard Galerkin Method Result

We apply the Galerkin method to the following one-dimensional model problem:

$$\epsilon u'' - wu' = 0, \qquad u(0) = 0, \ \ u(1) = 1 \tag{25}$$

Where $\epsilon \equiv 0.1$ and $w \equiv 1$. To aid in our analysis, we define the mesh Péclet number: $\mathcal{P}_h = \frac{\|\vec{w}\|h}{2\epsilon}$, where h is the largest element length in the direction of $\vec{w}$.

Figure 2 compares the numerical solution and exact solution for a range of $\mathcal{P}_h$. At $\mathcal{P}_h = 0.5$, the numerical solution approximates the exact solution closely. At $\mathcal{P}_h = 1$, the numerical solution has begun to deviate from the exact solution, but still displays the overall correct behaviour. Unforunately, at $\mathcal{P}_h = 2.5$, The numerical solution no longer resembles the exact solution, and now contains spurious oscillations.

4.4 Error Analysis

By Lax–Milgram, the variational problem (9) admits a unique solution $u \in V$ that depends continuously on the data (f, g_D, g_N). Consequently, any oscillations (e.g. in subsubsection 4.4.1) are simply perturbations of that single solution—no new spurious solutions can appear or disappear. Therefore, $u - u_h$ is purely a discretisation error, so we can meaningfully analyse and bound it.

4.4.1 Negative Diffusion

To demonstrate why standard Galerkin discretisations can oscillate when $\epsilon \ll 1$, we analyse the one dimensional model problem (25). Its exact solution is

$$u = \frac{e^{\frac{w}{\epsilon}x} - 1}{e^{\frac{w}{\epsilon}} - 1}. \tag{26}$$

For $w > 0$ the rapid exponential variation occurs in an $\mathcal{O}(\epsilon)$ layer at $x = 1$. By incompressibility and the small-ϵ limit, identical local layers form along any streamline ending on a 2D/3D outflow boundary, so this 1D analysis accurately predicts the leading-order behaviour there.

The Galerkin method with linear elements and integration using the trapezoid rule is equivalent to finite difference scheme

$$\epsilon\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - \frac{u_{i+1} - u_{i-1}}{2h} = 0, \tag{27}$$

where $g_i \coloneqq g(x_i)$.

By comparing the computed solution $\{u\}_i$ at the nodes, one discovers that the truncation error induced by the first-order central difference approximation is equivalent to the exact solution of

$$(\epsilon - \epsilon^*)u_i'' + wu_i' \tag{28}$$

Where ϵ^* can be considered as negative diffusion, and is given as a function of $\mathcal{P}_h$ by [4]

$$\epsilon^* = \epsilon\mathcal{P}_h\Big(\coth\mathcal{P}_h - \frac{1}{\mathcal{P}_h}\Big). \tag{29}$$

In Figure 3 we have plotted $\hat{\epsilon}_{\text{total}} \coloneqq (\epsilon - \epsilon^*)/\epsilon$ against $\mathcal{P}_h$. We observe that for $\mathcal{P}_h \gtrsim 2$, $\hat{\epsilon}_{\text{total}}$ becomes negative, signifying the onset of anti-diffusion. Since positive diffusion serves to damp perturbations, the emergence of negative diffusion instead amplifies any small errors in the discrete solution, thereby giving rise to the spurious oscillations seen at high Péclet numbers.

4.4.2 Error bounds

Next, we bound the Galerkin error and show it deteriorates as $\epsilon \to 0$. By linearity of the continuous and discrete problems, we have the Galerkin orthogonality

$$a(u - u_h, v_h) = 0, \qquad \forall v_h \in V^h \tag{30}$$

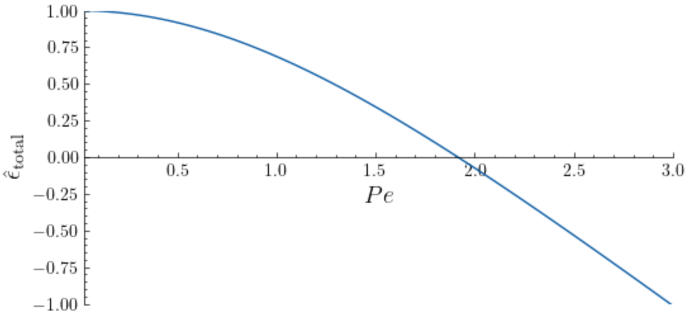

Figure 3: Total diffusion coefficient of the one-dimensional model problem (25) solved by the Galerkin FEM approximation with linear basis elements

Then, applying our coercivity and continuity bounds in the H^1 seminorm gives

$$\begin{aligned}\epsilon\|u-u_h\|_1^2 &\leq a(u-u_h, u-u_h)\\ &= a(u-u_h, u-v_h) + a(u-u_h, v_h-u_h)\\ &= a(u-u_h, u-v_h)\\ &\leq (\epsilon + \|\vec{w}\|_\infty)\|\nabla(u-u_h)\|\|\nabla(u-v_h)\|,\end{aligned}$$

where we used linearity from the first line to the second, and the bilinear form containing $v_h - u_h$ vanishes because $v_h - u_h \in V^h$, allowing us to use the Galerkin orthogonality property. Thus, we obtain Céa's Lemma,

$$\|\nabla(u-u_h)\| \leq (1 + \frac{K\|\vec{w}\|_\infty}{\epsilon})\|\nabla(u-v_h)\|. \tag{31}$$

This approximation is quasi-optimal, meaning optimal up to the constant $1 + \frac{\|\vec{w}\|_\infty}{\epsilon} = \mathcal{O}(\mathcal{P}_h)$. Unfortunately, this constant grows unboundedly as $\epsilon \to 0$.[5]

To illuminate this error estimate further, one can obtain an error bound in terms of the norm of the next highest derivative of the true solution, D^2u - the Hessian of u,

$$\|D^2u\|^2 := \int_\Omega \left((\frac{\partial^2 u}{\partial x^2})^2 + (\frac{\partial^2 u}{\partial y^2})^2 + (\frac{\partial^2 u}{\partial x \partial y})^2 \right) d\Omega. \tag{32}$$

By choosing v_h to be the standard finite-element interpolant of u and invoking Bramble–Hilbert (see e.g [6]), on can show immediately that

$$\|\nabla(u-u_h)\| \;\leq\; C\,h\,\|D^2u\|, \quad C = \mathcal{O}(\mathcal{P}_h). \tag{33}$$

[5] Higher-order Galerkin elements can reduce the effect of this factor (the so-called super-approximation property), but achieving a constant indepedent of ϵ requires a stabilised formulation.

(33) demonstrates that the errors within boundary layers are worse than (31) initially suggests, because not only does $C \to \infty$ as $\epsilon \to 0$, but $\|D^2 u\|$ is $\mathcal{O}(\epsilon^{-2})$, which also blows up within exponential boundary layers as $\epsilon \to 0$. We can then obtain a bound in terms of a fixed ϵ, recognising that $C = \mathcal{O}(\mathcal{P}_h)$, we factor out $\mathcal{P}_h$ and factor the rest of C and $\|D^2 u\|$ into D, giving

$$\|\nabla(u - u_h)\| \leq D\mathcal{P}_h h, \tag{34}$$

where D does not depend on h.

5 The Streamline Diffusion Method

We have shown that the standard Galerkin finite element method struggles to solve convection-dominated transport problems accurately, because the even upstream and downstream weighting introduces negative diffusion along the streamlines, resulting in non-physical oscillations when the element Péclet number becomes large.

One simple solution to this problem is mesh refinement, which is explored in Appendix E. However, this can be computationally expensive and/or complex to develop. In particular, the requirement that $\mathcal{P}_h \lesssim 2 \Rightarrow h \lesssim \frac{4\epsilon}{\|\vec{w}\|}$ locally, which for $\epsilon \ll \|\vec{w}\|$ can force impractically fine meshes. To avoid this route, we instead introduce stabilisation, which allows for larger mesh sizes by artificially stabilising the Galerkin method. In this section, we outline three progressively refined stabilisation strategies:

1. **Isotropic artificial diffusion**, which restores stability by augmenting the Laplacian with an isotropic stability parameter $\delta > 0$;
2. **Streamline diffusion**, which confines added diffusion to the flow direction but remains inconsistent for the original PDE; and
3. **Streamline Upwind/Petrov–Galerkin (SUPG)**, which isolates the artificial diffusion along streamlines and recovers consistency by basing the stabilization on the PDE residual.

5.1 Isotropic Artificial Diffusion

A classical remedy is to add an artificial diffusion term isotropically

$$-(\delta + \epsilon)\nabla^2 u + \vec{w} \cdot \nabla u = f \tag{35}$$

and then apply the Galerkin weak form. Here, $\delta > 0$ is chosen large enough to suppress oscillations but small enough to avoid excessive smearing. Provided δ is selected appropriately, this method produces a stable solution. However, there are two shortcomings.

Firstly, the Laplacian $\delta\nabla^2 u$ diffuses equally in all directions, not only along the streamlines of the flow, and thus excessively smears layers perpendicular to the flow direction. In simulations with significant variation (sharp layers) aligned (closely parallel to) with the flow, this renders the result very inaccurate. Note that this does not affect the quality of the solution in the trivial one-dimensional case.

Secondly, because the underlying PDE has been altered, the exact solution of the original problem no longer satisfies the discrete variational formulation. When we replace the original bilinear form (Eq. 7) by the modified bilinear form

$$a_\delta(u, v) = \big((\epsilon + \delta)\nabla u, \nabla v\big) + \big(\vec{w} \cdot \nabla u, v\big),$$

the exact solution u of the original PDE no longer satisfies $a(u, v_h) = F(v_h)$, $\forall v_h \in V^h$, but instead satisfies $a_\delta(u, v_h) = F(v_h)$. Hence, the crucial Galerkin orthogonality

$$a(u - u_h, v_h) = 0 \quad \forall v_h \in V^h$$

is lost (in fact, $a(u - u_h, v_h) = a_\delta(u, v_h) - a(u_h, v_h) \neq 0$ in general). Since Galerkin orthogonality is the linchpin of Céa's lemma, its error bound and convergence guarantee no longer hold. In practical terms, even as $h \to 0$, the discrete solution u_h will converge to the solution of the $(\epsilon + \delta)$-modified problem rather than the true solution of the original PDE. This is undesirable because it makes the solution unreliable.

Note that some authors call this an 'upwind finite element' method because with one-dimensional linear elements and optimal selection of the parameter δ, one reproduces the classical first-order upwind finite difference scheme,

$$\underbrace{-\,\epsilon\,\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}}_{\text{diffusion}} + \underbrace{w\,\frac{u_i - u_{i-1}}{h}}_{\text{upwind convection}}$$

in which the discrete convective operator biases information coming from the upstream side of each node [4]. We note that there are other ways to achieve this upwinding effect, through modifying the numerical quadrature rule and Petrov-Galerkin methodsc [10].

5.2 Streamline Diffusion

A simple approach to solving the crosswind diffusion was proposed by Brooks and Hughes in 1979 [8]. Once can replace the artificial diffusivity scalar δ by the rank-one projection tensor

$$D \coloneqq \tau\, \vec{w} \otimes \vec{w} \;(= \tau\, \vec{w}\vec{w}^T\,), \tag{36}$$

where $\tau = \delta/\|\vec{w}\|^2$, and $\|\vec{w}\|$ is the ℓ^2 norm.[6]

The tensor is used to introduce diffusivity only in the direction of the flow, resulting in an upwinded convective term along the streamlines of $\vec{w}$. This can be seen by investigating the eigenvalue decomposition. D has the eigenvalue τ in the $\vec{w}$ direction, and the eigenvalue 0 in directions perpendicular to $\vec{w}$. The bilinear form becomes

$$\begin{aligned} a(u_h, v_h)_D &\coloneqq \int_\Omega \big((\epsilon I + D)\nabla u_h\big)\cdot\nabla v_h \, d\Omega + \int_\Omega v_h(\vec{w}\cdot\nabla u_h)\, d\Omega \\ &= \epsilon \int_\Omega \nabla u_h \cdot \nabla v_h \, d\Omega + \tau \int_\Omega (\vec{w}\cdot\nabla u_h)(\vec{w}\cdot\nabla v_h)\, d\Omega + \int_\Omega v_h(\vec{w}\cdot\nabla u_h)\, d\Omega, \end{aligned}$$

where from the first to the second line we expand D and separate the contributions from ϵ and τ. Intuitively, one observes that this is the original bilinear form with the added τ correction term.

Unfortunately, this method still suffers from inconsistency. The second term in general does not vanish when u_h is equal to the true solution. Additionally, the diffusive term is not consistent with the centrally weighted source term, resulting in overly diffuse solutions if this term is present[4].

5.3 Streamline Upwind Petrov Galerkin (SUPG)

To solve the consistency problem, one must consistently weight all terms in the weak formulation. This can be done using the Petrov-Galerkin method, in which one modifies the test space $V_0^h \to \tilde{V}_0^h$, so that it differs from the trial space V_D^h. Then one can use an anisotropic test function to produce the desired stability-inducing $\tau \int_\Omega (\vec{w}\cdot\nabla u)(\vec{w}\cdot\nabla v)\, d\Omega$ term, within a consistent finite element framework. We will show that this is because the new test function introduces an additional term which ensures that exact solution which satisfy the regularity conditions also satisfy the variational form.

Explicitly, let V^h be the finite dimensional subspace of the space V defined in (**??**). We define the SUPG test space to be spanned by discontinuous test functions of form

$$\tilde{V}_0^h \coloneqq \operatorname{span}\big\{\tilde{v}_h : \tilde{v}_h = v_h + \tau\vec{w}\cdot\nabla v_h,\; v_h \in V_0^h\big\}.$$

[6] Note that Brooks and Hughes define this tensor instead using $\tau = \delta$ and $D \coloneqq \delta\,\hat{\vec{w}} \otimes \hat{\vec{w}}$, where $\hat{\vec{w}} \coloneqq \vec{w}/\|\vec{w}\|$. The two definitions are identical, we opted for ours because factoring out $\|\vec{w}\|^2$ early links better with the parameter choice τ in subsection 5.4.

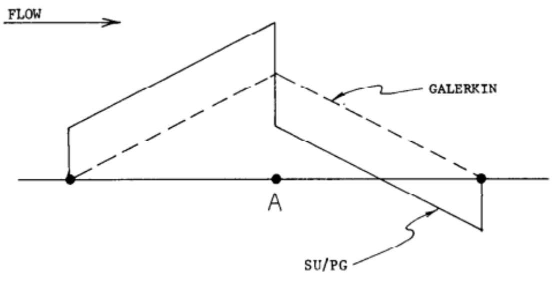

Figure 4: SUPG test functions aligned with the flow. [4]

Figure 4, taken from [4], compares a linear reference SUPG test function profile $\tilde{v}_h = v_h + \tau\vec{w} \cdot \nabla v_h$ along streamlines with a linear Galerkin test function profile. We see that the function applies greater weighting upstream of the central node than downstream, with a discontinuous jump at at the node.

In practice, one does not want to work with the complicated test functions $\tilde{v}_h$. Instead, we follow the same steps as in with $v \to \tilde{v}_h$, multiplying the strong form by $\tilde{v}_h$ and integrating to obtain the new variational formulation in terms of the standard $v_h \in V_0^h$ test functions. After applying the Galerkin approximation, the new bilinear form is

$$a_{supg}(u_h, v_h) = \epsilon \int_\Omega \nabla u_h \cdot \nabla v_h \, d\Omega + \int_\Omega (\vec{w} \cdot \nabla u_h) v_h \, d\Omega + \tau \int_\Omega (\vec{w} \cdot \nabla u_h)(\vec{w} \cdot \nabla v_h) \, d\Omega \\ - \ \tau\epsilon \int_\Omega (\nabla^2 u_h)(\vec{w} \cdot \nabla u_h) \, d\Omega$$

This requires $u_h \in H^2(\Omega)$, which is an issue because currently we only assume that $u_h \in H^1(\Omega)$. To remedy this, we restrict our discrete function space V^h to require that $u_h \in H^2(\Delta_K)$ on each individual element Δ_K. We define our new ambient discrete function space S^h to avoid confusion, defined formally as

$$S^h := H^1_D(\Omega) \ \cap \ H^2(\mathcal{T}_h),$$

where $H^2(\mathcal{T}_h)$ is a broken Sobolev space, defined by

$$H^2(\mathcal{T}_h) \ := \ \big\{ v \in L^2(\Omega) : v|_k \in H^2(\Delta_k), \forall k \in \mathcal{T}_h \big\}.$$

We can then replace the $\tau\epsilon$ term by the element-wise sum $-\tau\epsilon \sum_K \int_{\Delta_K} (\nabla^2 u_h)(\vec{w} \cdot \nabla u_h) \, d\Omega$. The right hand side is

$$F_{supg} = \int_\Omega f v_h \, d\Omega + \tau \int_\Omega f(\vec{w} \cdot \nabla v_h) \, d\Omega + \int_{\Gamma_N} v_h g_N \, d\Gamma$$

Thus, the SUPG finite-dimensional variational form is:
Find $u_h \in S_D^h$ such that

$$a_{supg}(u_h, v_h) = F_{supg}(v_h), \quad \forall v_h \in S_0^h \tag{37}$$

Based on the SUPG bilinear form, we define the SUPG norm (omitting the first order convective term due to skew-symmetry, using the same assumptions as in (**??**), and the second order term because it is sign-indefinite)

$$\|v\|_{supg} \coloneqq \left(\epsilon\|\nabla v\|^2 + \tau\|\vec{w}\cdot\nabla v\|^2\right)^{1/2}. \tag{38}$$

To prove that the SUPG formulation is consistent, we assume that an exact solution $\hat{u} \in W_h$ of the strong form exists and substitute it into the new variational form. Taking the second term across to the left hand side and noting that the residual of the (1) is $R(u) \coloneqq -\epsilon\nabla^2 u + \vec{w}\cdot\nabla u - f$, the variational formulation is

$$\epsilon\int_\Omega \nabla\hat{u}\cdot\nabla v_h\, d\Omega + \int_\Omega (\vec{w}\cdot\nabla\hat{u})v_h\, d\Omega + \tau\int_\Omega R(\hat{u})(\vec{w}\cdot\nabla v_h) = \int_\Omega f v_h\, d\Omega + \int_{\Gamma_N} v_h g_N\, d\Gamma. \tag{39}$$

Since $R(\hat{u}) = 0$ for the exact solution $\hat{u}$, this term vanishes and (39) becomes the variational formulation (6), which is satisfied by the exact solution $\tilde{u}$. Thus, the SUPG formulation is consistent. The terms $\tau\epsilon\sum_k\int_{\Delta_k}(\nabla^2 u_h)(\vec{w}\cdot\nabla u_h)\, d\Omega$ and $\tau\int_\Omega f(\vec{w}\cdot\nabla v_h)\, d\Omega$ can be viewed as a correction terms to correct the inconsistency of the classic streamline diffusion method. In fact, when using linear or bilinear basis elements the $(\nabla^2 u_h)$ term vanishes, simplifying the method.

5.4 Selecting the SUPG parameter τ

The analysis presented in Section 4.4.1 for the one-dimensional model problem (25) shows that the negative diffusion introduced is $\epsilon^* = \epsilon\Big(\coth\mathcal{P}_h - \frac{1}{\mathcal{P}_h}\Big)$. The SUPG stabilisation term (with homogeneous Neumann conditions and a linear element basis) is $\tau\int_\Omega(\vec{w}\cdot\nabla u)\,(\vec{w}\cdot\nabla v)\,\mathrm{d}\Omega$, and scales like $\|\vec{w}\|^2$.

Approximating multi-dimensional behaviour along streamlines by our one-dimensional model (3), and setting $\tau\|\vec{w}\|^2 = \epsilon^*$ to cancel the spurious diffusion, yields from (29)

$$\tau \;=\; \frac{h}{2\|\vec{w}\|}\Big(\coth\mathcal{P}_h - \tfrac{1}{\mathcal{P}_h}\Big). \tag{40}$$

Since the local Péclet number $\mathcal{P}_h^K$ varies per element K, one should ideally estimate τ locally in each element, which we denote as τ_K,

$$\tau_K = \frac{h_K}{2\|\vec{w}\|_K}\Big(\coth\mathcal{P}_h^K - \tfrac{1}{\mathcal{P}_h^K}\Big) =: \frac{h_K}{2\|\vec{w}\|_K}\xi_K, \tag{41}$$

where h_K is a measure of the element length in the direction of the wind [6]. The presence of coth can make (40) computationally expensive to implement element-wise when dealing with large meshes. Hughes and Brooks proposed two common approximations [4]. The asymptotic approximation is given by

$$\xi_K^*(\mathcal{P}_h^K) = \begin{cases} \dfrac{\mathcal{P}_h^K}{3}, & \mathcal{P}_h^K > 3, \\ 1, & \mathcal{P}_h^K \leq 3. \end{cases} \tag{42}$$

The critical approximation simply approximates $\coth(x) \approx 1$ when $x > 1$:

$$\xi_K^*(\mathcal{P}_h^K) = \begin{cases} 1, & \mathcal{P}_h^K \leq 1, \\ 1 + \dfrac{1}{\mathcal{P}_h^K}, & \mathcal{P}_h^K > 1. \end{cases} \tag{43}$$

5.5 A priori error analysis

We begin by stating an error bound in terms of $\|D^2u\|$, then use this to derive more specific bounds on the error in terms of $\mathcal{P}_h$ and h for fixed ϵ. The specific proof is given by Elman et al, based on a general analysis of Roos et al [6, 10].

Theorem 5.1. *Let the same assumptions apply as in the standard Galerkin bound, with the additional assumption that $|\vec{w}| = 1$ and $\mathcal{P}_h > 1$. Suppose* (9) *with constant coefficients is solved using the SD formulation. Use τ given by the critical approximation:*

$$\tau := \max\left\{0, \frac{h}{2}\left(1 - \frac{1}{\mathcal{P}_h}\right)\right\} \tag{44}$$

Then there exists a constant C_{supg}, bounded independently of ϵ, such that

$$\|u - u_h\|_{SD} \leq C_{supg} h^{3/2} \|D^2u\|. \tag{45}$$

Firstly, we note that in terms of ϵ-convergence, the constant C_{supg} is now independent of ϵ, which is much better than the constant for the Galerkin method (33).

Next, we obtain error bounds for fixed ϵ in terms of h and $\mathcal{P}_h$ by investigating the streamwise and crosswind directons. By expanding the streamline diffusion norm and using the identity $\sqrt{A+B} \geq \sqrt{B}$ for non-negative A, B, we have $\tau^{1/2}\|\vec{w}\cdot\nabla(u-u_h)\| \leq Ch^{3/2}\|D^2u\|$, where τ is $\mathcal{O}(h)$, as defined above. This gives a streamline error estimate for fixed ϵ as

$$\|\vec{w} \cdot \nabla(u - u_h)\| \lesssim h, \tag{46}$$

where $\lesssim$ means bounded by a D_s independent of h.

For bilinear (Q_1) elements the best-approximation error in the H^1-seminorm scales exactly like $\mathcal{O}(h)$, so our $\mathcal{O}(h)$ bound is the optimal h-rate attainable without raising the polynomial degree [2, 3].

Investigating the other term in the streamline diffusion norm, we notice $\epsilon^{1/2}\|\nabla(u-u_h)\| \lesssim h^{3/2}$. We get

$$\|\vec{w}_\perp \cdot \nabla(u-u_h)\| \leq \|\vec{w}\|\|\nabla(u-u_h)\| \leq \frac{C_{supg}\|\vec{w}\|h^{3/2}}{\sqrt{\epsilon}}\|D^2u\| \lesssim \mathcal{P}_h^{1/2}h. \tag{47}$$

Therefore, both the streamwise and crosswind components of the error are better than the Galerkin bound (34) as $h \to 0$. Since $\mathcal{P}_h^{1/2}h \gg h$ when $\mathcal{P}_h \gg 1$, the crosswind direction error is also worse than the stream wise direction, and dominates the total error in this limit.

Unfortunately, the term $\|D^2u\|$ is still present, which blows up within exponential boundary layers as $\epsilon \to 0$. So we expect that elements within these layers will have large errors in the derivatives when h is not well-refined. This is a simple fact of polynomial interpolation, whether the method is stabilised or not, one cannot resolve a sharp boundary layer (or any steep gradients) without having a h which is smaller than the width of the layer. The benefit of SUPG is that the solution is stable, and much more accurate everywhere else on the domain.

6 Numerical Example

6.1 Problem Description

We now use the SUPG method to obtain numerical solutions to the following two-dimensional problem, which contains an exponential outflow boundary layer. The solution is relatively simple, which is intentional so that we can use the analytical solution for exact error analysis.[7] The equation is

$$-\epsilon\nabla^2u + (0\ \ 1)^T\nabla u = 0, \quad x \in [-1,1]\times[-1,1] \tag{48}$$

with boundary conditions

$$u(x,y) = \begin{cases} x^3+1 & \text{on } y=-1, \\ 0 & \text{on } y=1 \text{ and } x=-1, \\ 2 & \text{on } x=1. \end{cases} \tag{49}$$

[7] One limitation of this particular model problem is that it only contains an outflow type boundary layer, and the uniform velocity field means that cross-wind diffusion has a negligible effect. A second interesting problem is in Appendix F.

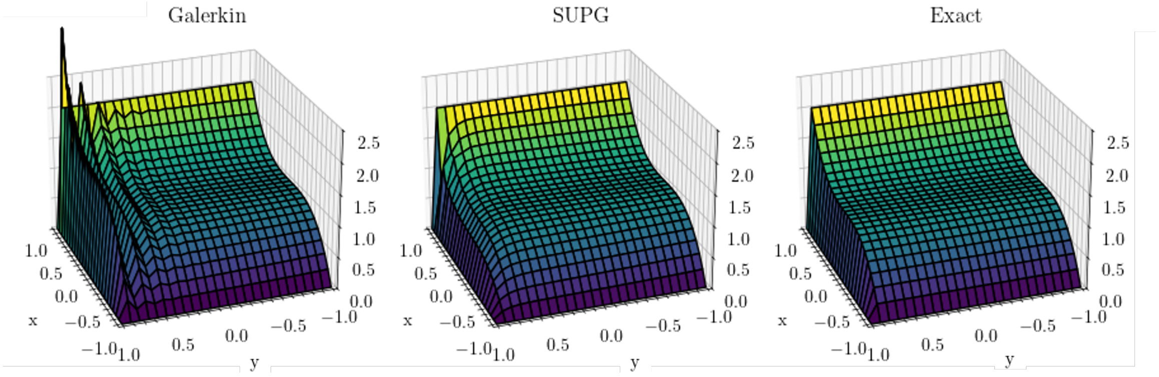

Figure 5: The numerical solution to (48), using the Galerkin method, SUPG method with critical approximation, and interpolated exact solution. $\mathcal{P}_h = 10$.

Which has the exact solution

$$u = (x^3 + 1)\frac{1 - e^{(y-1)/\epsilon}}{1 - e^{-2/\epsilon}} \tag{50}$$

The numerical solution was obtained using a uniform quadrilateral mesh of 24 elements in both the x and y direction. The $\mathcal{P}_h$ was varied, and the $\|\nabla(u - u_h)\|_{L^2(\Omega)}$ was obtained using *dblquad* from the *Scipy* Python library with a tolerance of of 10^{-8}. Both the critical and asymptotic parameter choice were used at each $\mathcal{P}_h$ for comparison.

6.2 Results

Figure 5 shows the standard Galerkin, SUPG, and interpolated exact solutions.[8] As we can see, the exact solution follows the initial condition closely along the streamlines $(0 \;\; 1)^T$, then encounters an exponential boundary layer at the boundary $y = 1$. The Galerkin solution is oscillatory and bears little resemblance to the exact solution when close to the boundary layer. In fact, although not visible on this particular plot, as $\mathcal{P}_h$ increases, the oscillations propagate through the entire domain, rendering the solution useless. The SUPG solution visibly approximates the solution well up until the boundary, where it is over-diffusive. This is in line with what we expect based on our error analysis in (45), in which the error is bounded by $\|D^2 u\|$ up to a constant. The magnitude of $\|D^2 u\|$ is greatest in this boundary layer.

[8] The exact solution was interpolated onto the quadrilateral basis using *plot_surface* from the Numpy library.

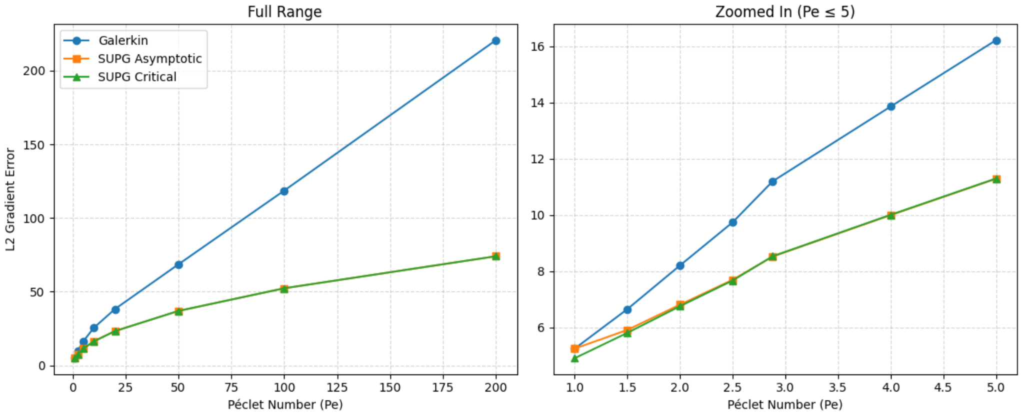

Figure 6: $\|\nabla(u - u_h)\|_{L^2(\Omega)}$ for each method at a range of $\mathcal{P}_h$.

Figure 6 shows the error measure $\|\nabla(u - u_h)\|_{L^2(\Omega)}$ for each method at a range of $\mathcal{P}_h$ from 1 to 200. The full table of data is provided in Appendix G. Firstly, we note that the SUPG error is improved compared to the Galerkin error, as expected. Secondly, the Galerkin error is linear in $\mathcal{P}_h$ as $\mathcal{P}_h \to \infty$, and the SUPG error scales with $\mathcal{O}(\mathcal{P}_h^{1/2})$, as our theoretical error bounds (34) and (47) showed. To compare the SUPG parameter approximation, we zoomed in to the data on the right-hand side. We see that $\mathcal{P}_h \lesssim 2.5$ the asymptotic approximation is more accurate, as would be expected, but beyond this they converge. This suggests that for convection-dominated problems, the choice of SUPG is not important provided it is close to 1.

7 Conclusion

To conclude, we have demonstrated why the standard Galerkin FEM fails to provide an accurate solution in of the steady-state convection-diffusion equation in dominated regimes. This was backed up with numerical examples and theoretical error bounds. Following this, we formulated the SUPG method conceptually, and presented further error bounds which demonstrate its dramatic improvement and stabilisation most of the domain, minus the exponential boundary layers which polynomial based methods struggle to approximate. Further work could include investigating SUPG combined with modern adaptive meshing alogrithms.

References

[1] R. Byron Bird, Warren E. Stewart, and Edwin N. Lightfoot. *Transport Phenomena*. 2nd ed. New York: John Wiley & Sons, 2002. ISBN: 978-0-471-41077-5.

[2] J. H. Bramble and S. R. Hilbert. “Estimation of Linear Functionals on Sobolev Spaces with Applications to Fourier and Spline Interpolation”. In: *SIAM Journal on Numerical Analysis* 7.1 (1970), pp. 112–124.

[3] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. 3rd. Vol. 15. Texts in Applied Mathematics. Springer, 2008.

[4] Alexander N. Brooks and Thomas J.R. Hughes. “Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations”. In: *Computer Methods in Applied Mechanics and Engineering* 32.1 (1982), pp. 199–259. ISSN: 0045-7825. DOI: `https://doi.org/10.1016/0045-7825(82)90071-8`. URL: `https://www.sciencedirect.com/science/article/pii/0045782582900718`.

[5] Jean Donea and Antonio Huerta. *Finite Element Methods for Flow Problems*. Chichester: John Wiley & Sons, 2003. ISBN: 978-0-471-49666-3.

[6] Howard C Elman, David J Silvester, and Andrew J Wathen. *Finite elements and fast iterative solvers : with applications in incompressible fluid dynamics*. Second edition. Oxford: Oxford University Press, 2014. ISBN: 0-19-178074-X.

[7] Patrick Farrell. *Finite Element Methods for PDEs*. 2022.

[8] T. J. R. Hughes and A. N. Brooks. “A Multidimensional Upwind Scheme with no Crosswind Diffusion”. In: *Finite Element Methods for Convection Dominated Flows*. Ed. by T. J. R. Hughes. Vol. 34. AMD. New York: ASME, 1979, pp. 19–35.

[9] Stephen B. Pope. *Turbulent Flows*. Cambridge: Cambridge University Press, 2000. ISBN: 978-0-521-59886-6.

[10] Hans-Görg Roos, Martin Stynes, and Lutz Tobiska. *Numerical Methods for Singularly Perturbed Differential Equations. Convection-Diffusion and Flow Problems*. Vol. 24. Springer Series in Computational Mathematics. Springer-Verlag Berlin Heidelberg, 1996. ISBN: 978-3-662-03206-0. DOI: `10.1007/978-3-662-03206-0`. URL: `https://doi.org/10.1007/978-3-662-03206-0`.

A Boundary layer types

To analyse these we introduce streamwise and cross-stream coordinates (s, η), where s is arc-length along a streamline, with tangent $\vec{t}$ and velocity component $w_s = \vec{w} \cdot \vec{t}$, and η is the distance normal to that streamline, with unit normal $\vec{e}_\eta$.

We assume that across any thin layer of half-width δ, derivatives scale as

$$u_s = \mathcal{O}\big(u/L\big), \quad u_{ss} = \mathcal{O}\big(u/L^2\big), \quad u_\eta = \mathcal{O}\big(u/\delta\big), \quad u_{\eta\eta} = \mathcal{O}\big(u/\delta^2\big),$$

where L is a characteristic length along the streamline. Then we have the following three boundary layer types:

1. Streamwise boundary layer: When a streamline meets a boundary at any angle, the rapid adjustment to the boundary condition occurs along the streamline direction. We balance convective transport along s against diffusion along s:
$$w_s\, u_s \;\sim\; \epsilon\, u_{ss} \quad\Longrightarrow\quad w_s\, \frac{u}{L} \;\sim\; \epsilon\, \frac{u}{L^2} \;\Longrightarrow\; \delta_s \sim \frac{\epsilon}{w_s} = \mathcal{O}(\epsilon).$$
Here δ_s is the layer width measured along the streamline itself.

2. Shear (cross-stream) layer: Where flow runs tangentially to a surface (wall or interior characteristic), there is no streamwise convective flux across that surface. Instead, tangential advection along s balances diffusion across η:
$$w_s\, u_s \;\sim\; \epsilon\, u_{\eta\eta} \quad\Longrightarrow\quad w_s\, \frac{u}{L} \;\sim\; \epsilon\, \frac{u}{\delta^2} \;\Longrightarrow\; \delta \sim \sqrt{\frac{\epsilon\, L}{w_s}} = \mathcal{O}\big(\sqrt{\epsilon}\big).$$
This cross-stream layer smooths gradients normal to streamlines when they slide along boundaries or meet internal discontinuities.

3. Mixed Interior (Characteristic) Layers: These layers form on internal characteristic surfaces—surfaces along which fluid parcels with different constant values of u are convected together so that the leading-order convective derivative along the surface vanishes. If $w_\eta \neq 0$, streamwise convection and diffusion dominate, giving $\mathcal{O}(\epsilon)$. If $w_\eta = 0$, there is no cross-stream convective transport, so tangential advection balances cross-stream diffusion, yielding $\mathcal{O}\big(\sqrt{\epsilon}\big)$.

B Equivalence of the H^1 norm and H^1 semi-norm

C Equivalence of the H^1 norm and H^1 semi–norm

Theorem C.1 (Poincaré–Friedrichs, [7])**.** *Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain and let $\Gamma_D \subset \partial\Omega$ be a closed subset of positive measure. If $v \in H^1(\Omega)$ satisfies $v = 0$*

on Γ_D, then there exists a constant $C_P = C_P(\Omega, \Gamma_D) < \infty$ such that

$$\|v\|_{L^2(\Omega)} \;\le\; C_P \,\|\nabla v\|_{L^2(\Omega)} \qquad \forall v \in H^1(\Omega) \text{ with } v|_{\Gamma_D} = 0.$$

Norm equivalence on $H_0^1(\Omega)$. For such functions we have

$$\|v\|^2_{H^1(\Omega)} = \|v\|^2_{L^2(\Omega)} + \|\nabla v\|^2_{L^2(\Omega)} \;\le\; C_P^2 \,\|\nabla v\|^2_{L^2(\Omega)} + \|\nabla v\|^2_{L^2(\Omega)} = (1 + C_P^2)\,\|\nabla v\|^2_{L^2(\Omega)}.$$

Conversely, since $\|\nabla v\|_{L^2} \le \|v\|_{H^1}$ trivially,

$$\|\nabla v\|^2_{L^2(\Omega)} \;\le\; \|v\|^2_{H^1(\Omega)} \qquad \forall v \in H_0^1(\Omega).$$

Taking square roots yields the desired two-sided estimate

$$\|\nabla v\|_{L^2(\Omega)} \;\le\; \|v\|_{H^1(\Omega)} \;\le\; \sqrt{1 + C_P^2}\,\|\nabla v\|_{L^2(\Omega)} \qquad \forall v \in H_0^1(\Omega).$$

Thus the H^1 norm and the H^1 semi-norm are equivalent on $H_0^1(\Omega)$.

D Boundedness of $a(u, v)$ and $F(v)$ proofs

D.1 Boundedness of $a(u, v)$ proof

Using the triangle inequality

$$|a(u, v)| \le \epsilon |b(u, v)| + |c(u, v)| \tag{51}$$

where $b(u, v) = \int_\Omega \nabla u \cdot \nabla v \, d\Omega$, $c(u, v) = \int_\Omega (\vec{w} \cdot \nabla u) v \, d\Omega$. We bound b first. Using the Cauchy-Schwarz inequality:

$$|b(u, v)| = \langle \nabla u, \nabla v \rangle_{L^2(\Omega)} \le \|\nabla u\|^2_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)}$$

$c(u, v)$ is bounded using successive applications of the Cauchy-Schwarz inequality:

$$\begin{aligned}
|c(u, v)| = |\int_\Omega (\vec{w} \cdot \nabla u) v \, d\Omega| &= \int_\Omega |\vec{w} \cdot \nabla u| |v| \, d\Omega \\
&\le \int_\Omega \|\vec{w}\|_2 \|\nabla u\|_2 |v| \, d\Omega \\
&\le \|\vec{w}\|_{L^\infty} \|\nabla u\|_{L^2} \|v\|_{L^2} \\
&\le K \|\vec{w}\|_{L^\infty} \|\nabla u\|_{L^2} \|\nabla v\|_{L^2}
\end{aligned}$$

From the first line to the second we use the Cauchy-Schwarz inequality on the term $|\vec{w} \cdot \nabla u|$.

Combining the bounds on $b(u, v)$ and $c(u, v)$, (19) follows directly. □

D.2 Boundedness of $F(v)$ proof

First, we apply the triangle inequality.

$$|F(v)| \leq \left| \int_\Omega f v \, d\Omega \right| + \epsilon \left| \int_\Gamma v g_N \, d\Omega \right|$$

The first term is equaivalent to $(f, v)_{L^2_\Omega}$ and can be bounded with a simple application of Cauchy-Schwarz. The second term requires the boundary analogue to the Poincare-Friedrichs inequality, known as the Trace Inequality. A proof can be found in [6]. We state it below:

Theorem D.1 (Trace Inequality)**.** *Let Ω be a bounded domain with a sufficiently smooth boundary, Γ. Then there exists a constant K_Γ such that*

$$\|v\|_{L^2(\Gamma)} \leq K_\Gamma \|v\|_{H^1(\Omega)}.$$

Using this inequality, we obtain

$$\begin{aligned} |F(v)| &\leq \|f\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} + \epsilon \|v\|_{L^2(\Gamma)} \|g_N\|_{L^2(\Gamma)} \\ &\leq \|f\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} + \epsilon K_\Gamma \|v\|_{H^1(\Omega)} \|g_N\|_{L^2(\Gamma)} \\ &\leq (\|f\|_{L^2(\Omega)} + \epsilon K_\Gamma \|g_N\|_{L^2(\Gamma)}) \|v\|_{H^1(\Omega)} \end{aligned}$$

□

E Grid Refinement

The standard Galerkin approximation produces inaccurate solutions and is inadequate for approximating the convection-diffusion equation when convection dominates.

One simple solution is to refine the grid, decreasing $\mathcal{P}_h$ so that the magnitude of the negative diffusion introduced by the Galerkin approximation remains $\ll \epsilon$. However, refining the grid uniformly significantly increases the computational cost, with the number of elements increasing in $\mathcal{O}(N^2)$ in two-dimensional domains. A better solution would be to only refine the grid in areas where the error is large.

(33) shows us that this is where $\|D^2 u\|$ is large, which occurs within boundary layers. Thus, one could create a mesh which is much more fine on the boundaries of a mesh. Conceptually, we can also picture this method as reducing $\mathcal{P}_h$ locally to bring it under the limit of negative diffusion at the domain boundaries.

Figure 7 shows a simple implementation of this for our 1-D model problem (25).

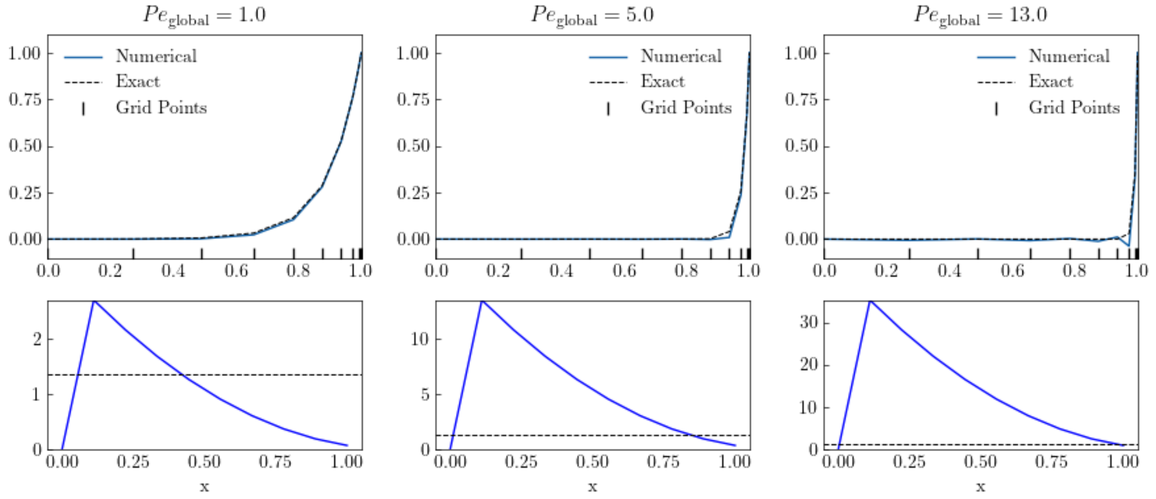

Figure 7: Grid refinement results. Grid points are shown above the x-axis of the top plots. The lower plots show the local element Peclet number

F Example problem 2: Interior boundary layer

This follows Example 6.1.3 defined in [6].

This allows us to meaningfully compare the standard Galerkin method, isotropic diffusion method (subsection 5.1), and SUPG method for a more realistic problem. This is a qualitative comparison, validated against an accurate finite element solution for the same problem provided in [4].

Figure 8 and Figure 9 compare FEM soltions using different methods for this problem.

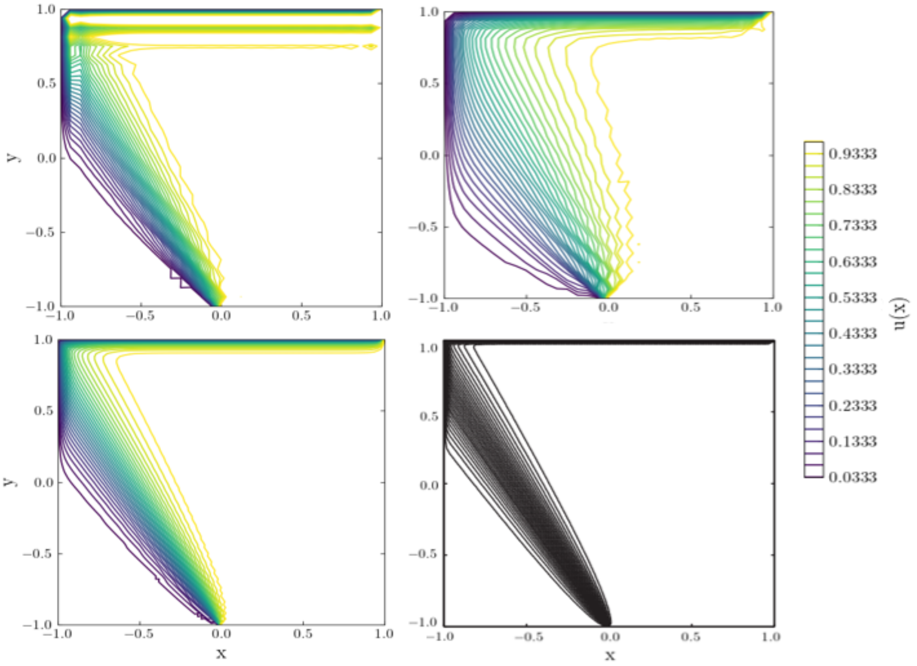

Figure 8: Contour plots of Example 6.1.3 in [6]. Top right: Standard Galerkin method. Top left: Artificial diffusion method. Bottom left: SUPG method with critical approximation. Bottom right: Accurate finite element solution from [6]

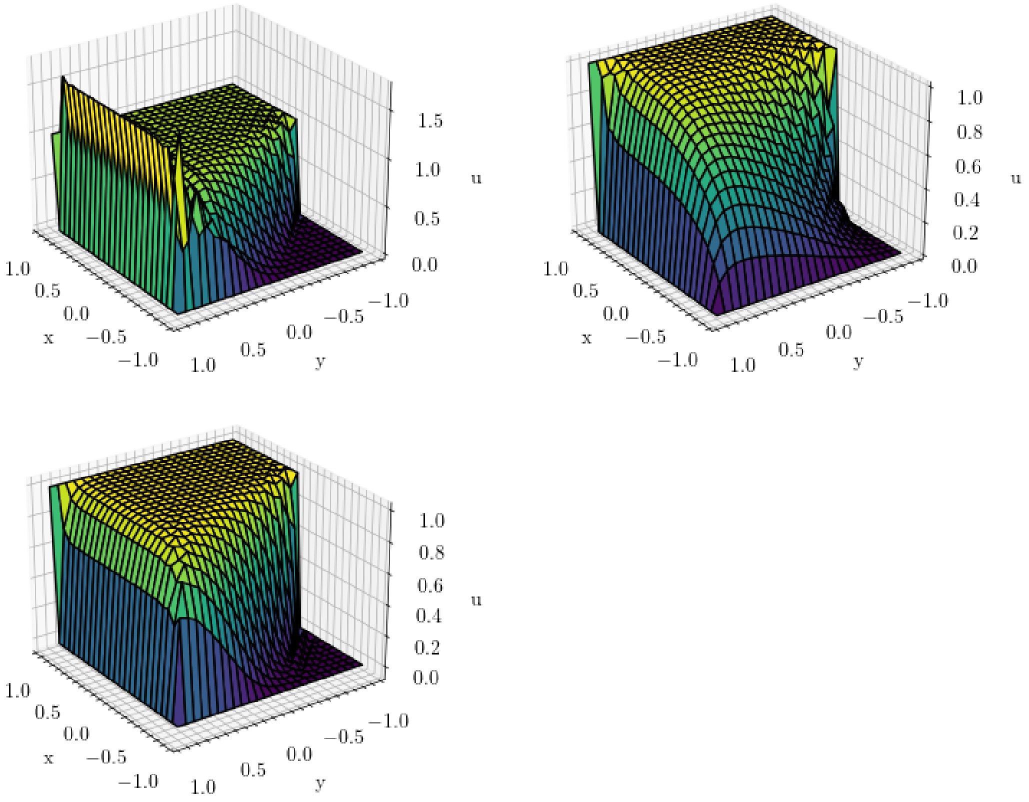

Figure 9: Surface plots of Example 6.1.3 in [6]. Top right: Standard Galerkin method. Top left: Artificial diffusion method. Bottom left: SUPG method with critical approximation.

G L^2 Error Table

$\mathcal{P}_h$	ϵ	Galerkin	SUPG Asymptotic	SUPG Critical
1	1/24	5.24326	5.24326	4.90291
2.5	1/60	9.74016	7.69315	7.66768
5	1/120	16.21868	11.28647	11.29689
10	1/240	25.39173	16.28448	16.29261
20	1/480	38.19257	23.25083	23.25680
50	1/1200	68.53897	36.95271	36.95585

Table 1: $\|\nabla(u-u_h)\|_{L^2(\Omega)}$ of (48) for a range of $\mathcal{P}_h$

H Implementation

H.1 One-dimensional implementation

In the one-dimensional examples, $\Omega = [0,1]$ is divided into N uniform intervals of width $h = 1/N$. We use continuous piecewise linear functions (denoted CG$_1$). Our reference element $\hat{K} = [0,1]$ has the linear shape functions:

$$\hat{\phi}_1(x) = 1 - x, \quad \hat{\phi}_2(x) = x.$$

This yields continuous, piecewise-linear functions (denoted CG$_1$).

H.2 Stiffness matrix assembly algorithm

For each element we:

1. Evaluate the local element matrix and load vector using the basis functions on the reference element $\hat{K}$. In our case, all integrals involved in the stiffness matrix were computed symbolically using exact integration over $\hat{K}$. This is possible due to the simplicity of the bilinear basis functions and the affine mapping, which results in constant gradients on each element. No numerical quadrature is required for the stiffness matrix. The load vector is assembled using a lumped (midpoint) approximation, evaluating the source function at each node and multiplying by the approximate integral of the basis function, $\int_K \phi_a \approx |K|/4$.
2. Use the affine map F_K to scale integrals appropriately. On a uniform Cartesian mesh, this mapping is trivial (constant Jacobian), and element transformations reduce to uniform scaling and translation.

3. Accumulate the local element contributions into the global stiffness matrix and load vector using the local-to-global node mapping.

Note: In our implementation, the affine map is not defined explicitly. Its effect is incorporated into the precomputed gradients of the basis functions, which are scaled according to the constant Jacobian of the transformation. This is valid on a Cartesian mesh where the mapping is the same for every element.

Because the mesh is Cartesian and the affine mapping is uniform across all elements, the assembly is efficient. We compute the local stiffness matrix once and reuse it for all elements.

Algorithm 1 Element-wise Assembly with Local $\mathcal{P}_h^K$-Based Stabilization

```
for each element $K$ in the mesh do
    Get element size $h_K$ (here $h_K = dx = dy$)
    Compute element area $|K| = dx \cdot dy$
    Compute local Péclet number $Pe_K = \frac{|\mathbf{v}| \cdot h_K}{2\varepsilon}$.
    Determine $\tau_K$ based on $Pe_K$:
    if method = critical_approximation then
        if $Pe_K > 1$ then
            $\tau_K \leftarrow \frac{h_K}{2}(1 - \frac{1}{Pe_K})$
        else
            $\tau_K \leftarrow 0$
        end if
    else if method = asymptotic_approximation then
        if $Pe_K < 3$ then
            $\tau_K \leftarrow \frac{h_K}{2} \cdot \frac{Pe_K}{3}$
        else
            $\tau_K \leftarrow \frac{h_K}{2}$
        end if
    else
        $\tau_K \leftarrow 0$
    end if
    Get global node indices: $[i_1, i_2, i_3, i_4]$
    Initialize local matrix $A^{(K)} \leftarrow 0$
    for $a = 1$ to 4 do
        for $b = 1$ to 4 do
            Compute:
            diffusive $\leftarrow \varepsilon \, \nabla\phi_a \cdot \nabla\phi_b \cdot |K|$
            convective $\leftarrow \mathbf{v} \cdot \nabla\phi_b \cdot |K|/4$
            stabilization $\leftarrow \tau_K \, (\mathbf{v} \cdot \nabla\phi_a)(\mathbf{v} \cdot \nabla\phi_b) \cdot |K|$
            $A_{ab}^{(K)} \leftarrow$ diffusive + convective + stabilization
        end for
    end for
    Assemble both global stiffness matrix $A$ and load vector $F$:
    for $a = 1$ to 4 do
        for $b = 1$ to 4 do
            $A[i_a, i_b]$ += $A_{ab}^{(K)}$
        end for
        $F[i_a]$ += $f(i_a) \cdot \frac{|K|}{4}$    ▷ $f(i_a)$ is the value of the source function at global node $i_a$
    end for
```