# A local projection stabilisation for convection-diffusion-reaction equations using a biorthogonal system and adaptive refinement

Bishnu P. Lamichhane[1] Jordan Shaw-Carmody[2]

(Received 22 February 2022; revised 22 July 2022)

**Abstract**

We consider a local projection stabilisation based on biorthogonal systems and adaptive refinement for convection-diffusion-reaction differential equations. The local projection stabilisation and adaptive finite element method are both based on a biorthogonal system. We investigate the numerical efficiency of the approach when compared to the standard finite element method. Numerical examples are presented to demonstrate the performance of the approach.

## Contents



DOI:10.21914/anziamj.v63.17179,  Published 2022-07-29, as part of the Proceedings of the 15th Biennial Engineering Mathematics and Applications Conference. ISSN 1445-8810. (Print two pages per sheet of paper.) 

# 1 Introduction

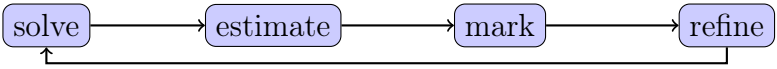
The applications of convection-diffusion-reaction differential equations are vast in the areas of science and engineering. However, when attempting to apply the standard finite element method to these equations, a stable approximation solution is not obtained due to spurious modes [9, 14]. This suggests that for the finite element method to be a viable approach, a stabilisation technique must be applied to stop the formation of spurious modes in the approximation. To this end, a wide variety of stabilisation techniques have been developed [1, 2, 13, 14].

For the advection dominated problem, the spurious modes come primarily from boundary and internal layers present in the exact solution to the equation. The development of stabilisation techniques to stop the formation of spurious modes when using the standard finite element method has been a popular area of research. Some of the methods of stabilisation that have been developed include the streamline-diffusion methods, least square methods, residual-free bubbles, local projection schemes, continuous interior penalty methods, discontinuous Galerkin methods, and the Galerkin least-square method [10, 14]. In this article we are focusing on a new adaptive finite element technique, which uses a local projection scheme based on a biorthogonal system.

Adaptive finite element methods have also been applied to the reaction-

diffusion-convection problem, but previously the focus has been on grid refinements for the standard finite element method rather than the use of stabilisation in conjunction with the adaptive mesh. The techniques that have been used with adaptive techniques are the streamline diffusion finite element method [15] and the multilevel homotopic adaptive finite element method [6].

Adaptive finite element schemes follow four steps that repeat until a stopping criterion is satisfied (as shown below).




Let the domain $\Omega \subset \mathbb{R}^2$ be bounded by a polygonal boundary $\partial\Omega$, where $\partial\Omega$ is divided into two disjoint parts $\Gamma_D$ and $\Gamma_N$. We consider the convection-diffusion-reaction equation

$$\begin{aligned} -\epsilon\Delta u + \mathbf{b}\cdot\nabla u + cu &= f \quad \text{in } \Omega\,, \\ u &= g_D \quad \text{on } \Gamma_D\,, \\ \epsilon\frac{\partial u}{\partial \mathbf{n}} &= g_N \quad \text{on } \Gamma_N\,, \end{aligned} \tag{1}$$

for given functions $\mathbf{b} \in \left[W^{1,\infty}(\Omega)\right]^2$, $c \in L^\infty(\Omega)$, $g_N \in L^2(\Gamma_N)$, $g_D \in H^{1/2}(\Gamma_D)$ and $f \in L^2(\Omega)$ satisfying

$$\sigma := c - \tfrac{1}{2}\nabla\cdot\mathbf{b} \geqslant \sigma_0 > 0\,, \tag{2}$$

for constants $\sigma_0$ and $0 < \varepsilon \ll 1$. Here $\mathbf{n}$ is the unit outward normal vector of $\Gamma_N$.

The purpose of this article is to explore the viability of local projection stabilisation using a biorthogonal system combined with an adaptive refinement technique, and compare it numerically with the uniformly refined stabilised finite element method.

To apply the finite element method we must consider the weak formulation of (1). Let $V_0 := \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}$ and $V_D := \{v \in H^1(\Omega) : v|_{\Gamma_D} = g_D\}$. The weak formulation is to find $u \in V_D$ such that

$$a(u, v) = \ell(v), \quad v \in V_0, \tag{3}$$

with

$$a(u, v) = \varepsilon \int_\Omega \nabla u \cdot \nabla v \, dx + \int_\Omega [(\mathbf{b} \cdot \nabla u) v + cuv] \, dx$$

$$\text{and} \quad \ell(v) = \int_\Omega f v \, dx + \int_{\Gamma_N} g_N v \, ds.$$

Assuming that the inflow boundary is part of the Dirichlet boundary $\Gamma_D$; that is

$$\{x \in \partial\Omega : \mathbf{b} \cdot \mathbf{n} < 0\} \subset \Gamma_D, \tag{4}$$

the condition (2) guarantees the coercivity of the bilinear form $a(\cdot, \cdot)$ on $V_0$. Hence the boundary value problem (3) has a unique solution by the Lax–Milgram Lemma [4, 8].

# 2 Finite element method

## 2.1 Finite element discretisation

We consider a shape regular triangulation $\mathcal{T}_h$ of the polygonal domain $\Omega$, where $\mathcal{T}_h$ consists of triangles. The diameter of an element $T \in \mathcal{T}_h$ is denoted by $h_T$. Let a reference triangle be defined as

$$\hat{T} = \{\mathbf{x} \in \mathbb{R}^2 : x_1, x_2 > 0, \text{ and } x_1 + x_2 < 1\}.$$

The linear finite element space based on the triangles $\mathcal{T}_h$ is then given by [3, 4]

$$V_h := \{v_h \in H^1(\Omega) : v_h|_T \in \mathcal{P}_1(T), T \in \mathcal{T}_h\}, \quad V_{0,h} := V_h \cap V_0, \tag{5}$$

where $\mathcal{P}_1(T)$ is the set of all linear polynomials on $T$. This then allows us to write our discretised formulation of (3) as finding $u_h \in V_h$ with $u_h|_{\Gamma_D} = g_{D,h}$, which is a projection of $g_D$ onto $V_h$, such that

$$a(u_h, v_h) = \ell(v_h), \quad v_h \in V_{0,h}. \tag{6}$$

## 2.2 Local projection stabilisation

Let the basis functions of $V_h$ with respect to the mesh $\mathcal{T}_h$ be the set $\mathcal{B}_1 := \{\eta_1, \ldots, \eta_n\}$. This is now used to construct a second basis $\mathcal{B}_2 := \{\psi_1, \ldots, \psi_n\}$ which satisfies the biorthogonality condition

$$\int_\Omega \eta_i \psi_j \, dx = c_j \delta_{i,j}, \quad c_j \neq 0, \quad 1 \leqslant i, j \leqslant n, \tag{7}$$

where $\delta_{i,j}$ is the Kronecker delta function, and $c_j$ is a scaling factor that is chosen to be proportional to the area $|\operatorname{supp} \eta_j|$. Basis $\mathcal{B}_2$ allows us to define another finite element space $Q_h := \operatorname{span}\{\mathcal{B}_2\}$.

Using this new space we now construct two quasi-projection operators: $\Pi_h : L^2(\Omega) \mapsto V_h$ and $\Pi_h^* : L^2(\Omega) \mapsto Q_h$ defined by

$$\int_\Omega \Pi_h v \psi_h \, dx = \int_\Omega v \psi_h \, dx, \quad v \in L^2(\Omega), \quad \psi_h \in Q_h, \tag{8}$$

and

$$\int_\Omega \Pi_h^* v v_h \, dx = \int_\Omega v v_h \, dx, \quad v \in L^2(\Omega), \quad v_h \in V_h. \tag{9}$$

An analysis on the properties of the quasi-projection operators is given by Lamichhane [11]. The projection operator $\Pi_h^*$ is required to implement the stabilisation of the standard finite element method. The stabilisation term

$$S(u_h, v_h) = \sum_{T \in \mathcal{T}_h} \int_T h_T \left[\mathbf{b} \cdot \nabla u_h - \Pi_h^* (\mathbf{b} \cdot \nabla u_h)\right] \left[\mathbf{b} \cdot \nabla v_h - \Pi_h^* (\mathbf{b} \cdot \nabla v_h)\right] dx \tag{10}$$

is added to the left hand side of (6) to stabilise the formulation by finding $u_h \in V_h$ with $u_h|_{\Gamma_D} = g_{D,h}$ such that for $v_h \in V_{0,h}$:

$$\varepsilon \int_\Omega \nabla u_h \cdot \nabla v_h \, dx + \int_\Omega \left[ (\mathbf{b} \cdot \nabla u_h) + c u_h \right] v_h \, dx + S(u_h, v_h) = \ell(v_h).$$

This is written in a similar form to (3) by first defining the bilinear form

$$A(u_h, v_h) = \varepsilon \int_\Omega \nabla u_h \cdot \nabla v_h \, dx + \int_\Omega \left[ (\mathbf{b} \cdot \nabla u_h) + c u_h \right] v_h \, dx + S(u_h, v_h).$$

Then our problem is written as finding $u_h \in V_h$ with $u_h|_{\Gamma_D} = g_{D,h}$ such that

$$A(u_h, v_h) = \ell(v_h), \quad v_h \in V_{0,h}. \tag{11}$$

The norm used for error analysis is the LP-norm:

$$\|v\|_{LP} = \left( \varepsilon |v|^2_{1,\Omega} + \sigma_0 \|v\|^2_{0,\Omega} + S(v,v) + \frac{1}{2} \sum_{e \in \Gamma_N^h} \int_e |\mathbf{b} \cdot \mathbf{n}| \, v^2 \, ds \right)^{1/2},$$

where $\Gamma_N^h$ is the set of all Neumann edges. This allows us to use the following theorem.

**Theorem 1.** *Let $u \in H^2(\Omega)$ and $u_h \in V_h$ be the solutions of (3) and (6), respectively. Then there exists a constant $C$ independent of the mesh-size $h$ such that the following a priori error estimate holds true:*

$$\|u - u_h\|_{LP} \leqslant C \left( \sum_{T \in \mathcal{T}_h} \left( \varepsilon + h_{\omega(T)} \right) h^2_{\omega(T)} \|u\|^2_{2,\omega(T)} \right)^{\frac{1}{2}}, \tag{12}$$

*where $\omega(T)$ is the patch of elements touching $T$ and $h_{\omega(T)}$ is the diameter of $\omega(T)$.*

The proof of this theorem is given by Lamichhane and Shaw-Carmody [12].

# 3 Adaptivity

When applying the estimation step, *a posteriori* error estimator needs to be chosen as some measure of the estimated error of the solution over each element. This is used to inform the marking step. Let $\mathcal{T}$ be a triangulation of $\Omega$. We define the global error estimator

$$\eta\left(\mathcal{T};v\right)=\left(\sum_{T\in\mathcal{T}}\eta_T\left(\mathcal{T};v\right)^2\right)^{1/2}\qquad\text{for all } v\in V_h\,,$$

where the localised estimation for each $T\in\mathcal{T}$ satisfies

$$\eta\left(\mathcal{T};\cdot\right):V_h\mapsto[0,\infty)\,.$$

Here we use the local estimator

$$\eta_T\left(\mathcal{T};v\right)=\left(h_T\int_T\left[\mathbf{b}\cdot\nabla v-\Pi_h^*\left(\mathbf{b}\cdot\nabla v\right)\right]^2\mathrm{d}x\right)^{1/2},\tag{13}$$

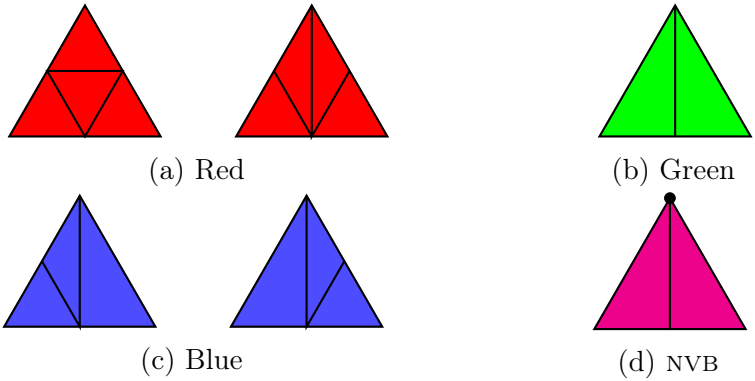
which is motivated by the stabilisation term.

The results of applying the local error estimator are used to inform the marking step. The Dörfler marking scheme is a popular strategy in adaptive finite element to determine which elements are to be marked for refinement.

The marking scheme is to select $\mathcal{M}$, a subset of $\mathcal{T}$, with minimal cardinality such that

$$\theta\eta\left(\mathcal{T};u_h\right)^2\leqslant\sum_{T\in\mathcal{M}}\eta_T\left(\mathcal{T};u_h\right)^2\tag{14}$$

holds true for some fixed parameter $0<\theta<1$. The set $\mathcal{M}$ represents which elements are marked to be refined because they constitute the higher proportion of the estimated error. If $\theta=0$ then no elements will be chosen, and if $\theta=1$ then all of the elements, except those with $\eta_T\left(\mathcal{T};u_h\right)=0$, will be chosen.



(a) Red (b) Green

(c) Blue (d) NVB

Figure 1: Examples of a single refinement using each of the named refinement types.

## 3.1 Refinement schemes

The refinement step of the adaptive algorithm is broken down into two parts: the refining of the elements that have been marked, and the closure of the mesh. The closure step involves checking each element of the mesh for hanging nodes, which for linear triangles are nodes that lie on the boundaries of the element that are not the vertices that define the triangle. If an element is found to have a hanging node, then it is marked for further refinement according to the refinement scheme. These two steps alternate until the closure step finds no elements that need to be marked.

### Red Refinement

Red refinement is where a triangle is split into four sub-triangles. This is traditionally performed either by placing nodes at the midpoints of the edges and connecting the midpoint edges creating four elements, demonstrated in the left picture of Figure 1a, or by applying a Green refinement to the original element and then applying a further Green refinement to the resultant triangles, shown in the right picture of Figure 1a. A Green refinement is where a triangle is split into two sub-triangles by connecting the midpoint of

an edge of a triangle to its opposite vertex. This Green refinement is shown in Figure 1b, where the selected edge is the bottom edge of the triangle. When Red refinement is used on all of the elements, then this is equivalent to uniform refinement, which we use as our baseline for comparison.

### Red-Green (RG) Refinement

Red-Green refinement is where the marked elements have a Red refinement applied to them and then a closure cycle occurs to deal with all of the hanging nodes. The closure cycle occurs by marking all elements that contain a hanging node. If an element has more than one edge with a hanging node, then it will have a Red refinement applied to it. Once all Red refinements are completed and no new Red refinements are required, then the elements with hanging nodes have Green refinements, shown in Figure 1b, applied to them to resolve the hanging node. There is a potential problem with the Green refinement component, where the child triangles can become degenerate since one of the angles of the parent is guaranteed to be cut in half.

### Red-Green-Blue (RGB) Refinement

The implementation of the Red-Green-Blue refinement is more complicated than the Red-Green or Red refinement methods because it does not have a clear decision path for how to apply the Blue refinement, which has left and right refinements as shown in Figure 1c, and thus a choice has to be made while implementing the method. Clarity can be gained at the price of additional overhead by having one of the edges of the triangle denoted as the reference edge. This reference edge then dictates the orientation of the triangle. One such implementation of this method is demonstrated by Funken and Schmidt [7], which we adopt here.

### Newest-Vertex-Bisection (NVB) Refinement

Newest-Vertex-Bisection is a refinement method that has a vertex of each triangle which is denoted as the newest vertex. When the triangle is marked

for refinement the new edge which is constructed to split the triangle is drawn between the newest vertex and the midpoint of the edge opposite the newest vertex. This is shown in Figure 1d where the vertex with the black dot denotes the newest vertex. The newest vertex information can be stored as easily in the element information as the first node which defines the triangle. This method has an advantage over the other refinement methods since it is known to be compatible with the axioms of adaptivity [5], which guarantee quasi-optimal convergence if satisfied. This is not true of Red-Green-Blue refinement.

# 4 Numerical results

## 4.1 Example 1

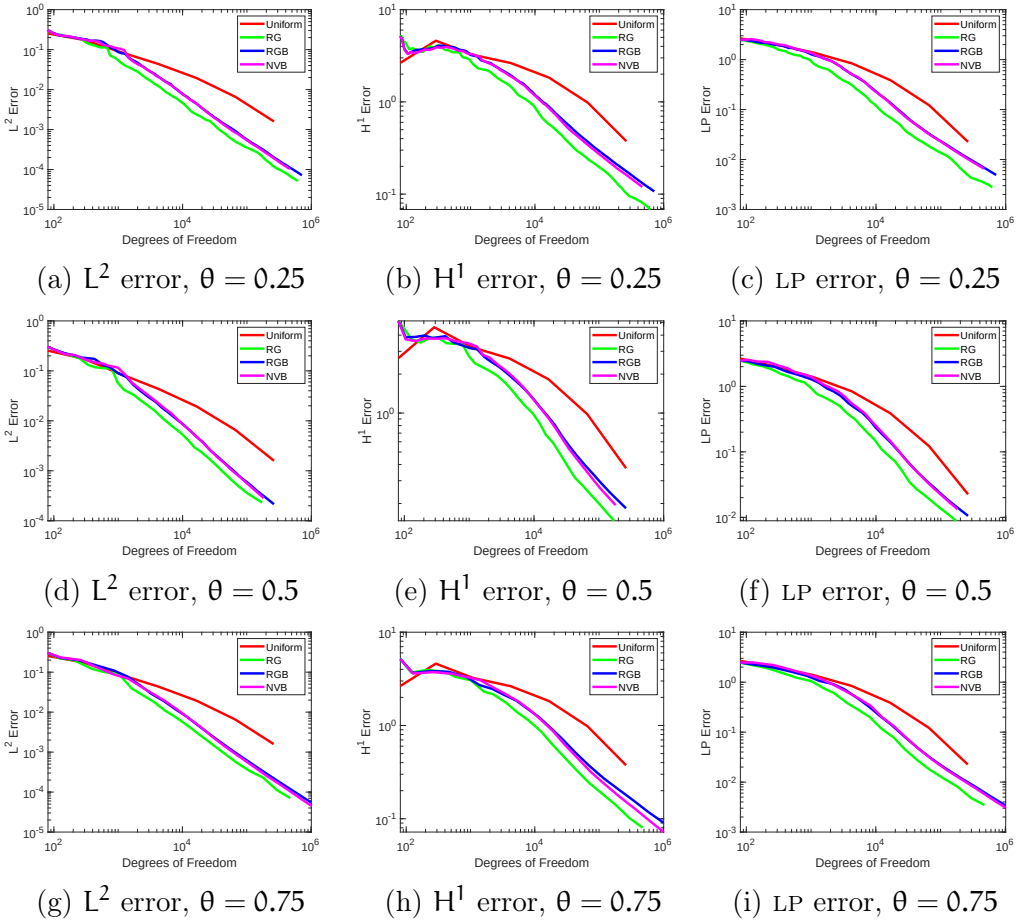
Here, we consider the PDE (1) with coefficients $\varepsilon = 10^{-8}$, $\mathbf{b} = (2,3)$, $c = 2$, and where $\Omega = (0,1)^2$, $\Gamma_D = \partial\Omega$, $\Gamma_N = \emptyset$ for which the exact solution is

$$u(x,y) = 16x(1-x)y(1-y) \times \left( \frac{1}{2} + \frac{\tan^{-1}\left(200\left[\frac{1}{4^2} - \left(x-\frac{1}{2}\right)^2 - \left(y-\frac{1}{2}\right)^2\right]\right)}{\pi} \right).$$

This PDE contains a transition layer on a circle in the domain with radius $0.25$ and centre $(0.5, 0.5)$.

Here we compare the different refinement schemes over various values of $\theta$ to investigate the convergence of each method for their individual marking regiments. For each test we use the uniform refinement as the benchmark for comparison, which is independent of $\theta$.

For $\theta = 0.25, 0.5, 0.75$, the adaptive refinement schemes were run for 70, 30 and 20 iterations, respectively, to obtain a similar number of degrees of freedom used for eight iterations of uniform refinement. Figure 2 shows that all of the adaptive refinements perform better than the uniform refinement.




(a) $L^2$ error, $\theta = 0.25$ (b) $H^1$ error, $\theta = 0.25$ (c) LP error, $\theta = 0.25$

(d) $L^2$ error, $\theta = 0.5$ (e) $H^1$ error, $\theta = 0.5$ (f) LP error, $\theta = 0.5$

(g) $L^2$ error, $\theta = 0.75$ (h) $H^1$ error, $\theta = 0.75$ (i) LP error, $\theta = 0.75$

Figure 2: The $L^2$, $H^1$ and LP error convergence comparison over Uniform, RG, RGB and NVB refinement regimes for Example 1 with $\theta = 0.25, 0.5, 0.75$ for each row, respectively.

For each value of $\theta$ the Red-Green refinement scheme performs better than the Red-Green-Blue and Newest-Vertex-Bisection schemes, which have similar errors.

## 4.2 Example 2

Here, we consider the PDE (1) with coefficients $\varepsilon = 10^{-9}$, $\mathbf{b} = (1, 0)$, $c = 1$, and where $\Omega = (0, 1)^2$, $\Gamma_D = \partial\Omega$, $\Gamma_N = \emptyset$ for which the exact solution is

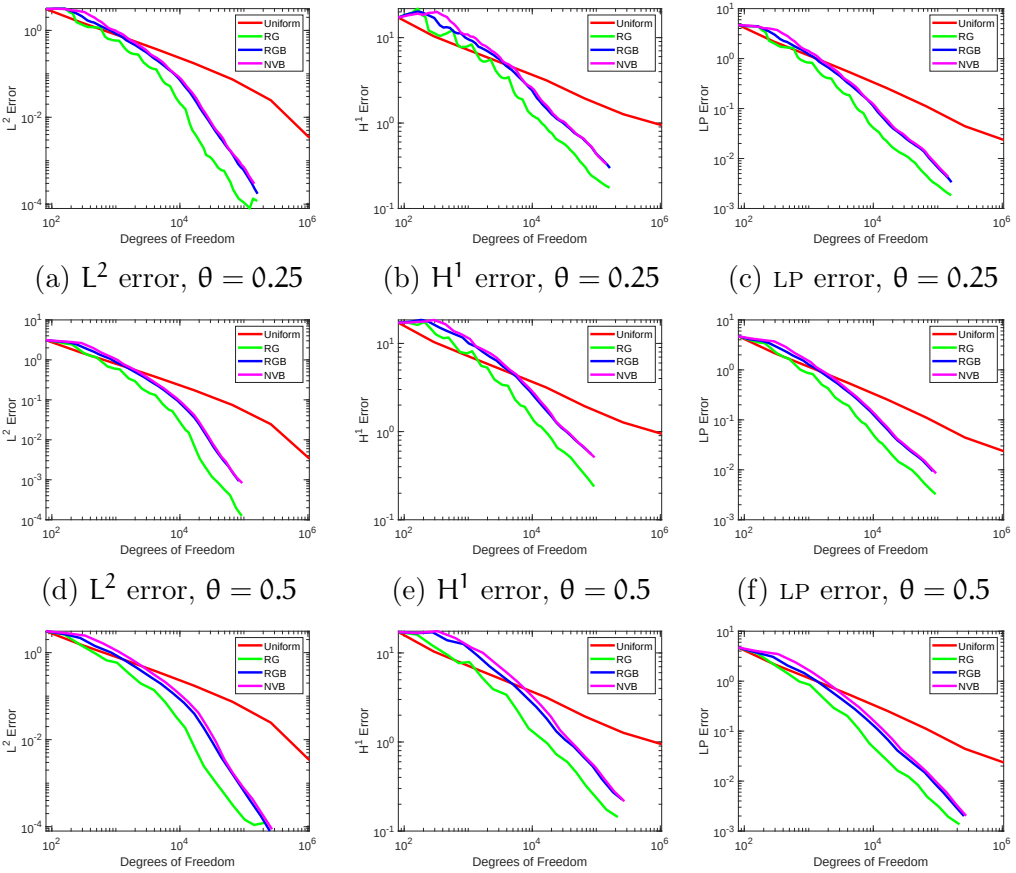
$$u(x, y) = \frac{xy}{2}(1 - x)(1 - y)\left[1 - \tanh\left(\frac{\alpha - x}{\gamma}\right)\right].$$

In this problem the parameters $\alpha$ and $\gamma$ control the location and thickness of the interior layer. Here $\alpha = 0.5$ and $\gamma = 0.0005$. The adaptive method works better where the transition layer is quite sharp.

For $\theta = 0.25, 0.5, 0.75$, the adaptive refinement schemes are run for 70, 30 and 20 iterations, respectively. Figure 3 shows that all of the adaptive refinements perform better than the uniform refinement which represents the performance of the stabilised standard finite element method. For each value of $\theta$ the Red-Green refinement scheme performs better than the Red-Green-Blue and Newest-Vertex-Bisection schemes, which have similar errors. However, in this example the Red-Green scheme suffers from oscillation in the error, from which the Red-Green-Blue and Newest-Vertex-Bisection schemes do not suffer. This shows that while the error is decreasing overall for the Red-Green scheme, there is no guarantee that the error would be decreasing from one iteration to the next. However, the oscillation appears to be larger for smaller values of $\theta$. Further investigation of this property is warranted in future research.

# References

[1] R. Becker and M. Braack. “A finite element pressure gradient stabilization for the Stokes equations based on local projections”. In:




(a) $L^2$ error, $\theta = 0.25$ (b) $H^1$ error, $\theta = 0.25$ (c) LP error, $\theta = 0.25$

(d) $L^2$ error, $\theta = 0.5$ (e) $H^1$ error, $\theta = 0.5$ (f) LP error, $\theta = 0.5$

(g) $L^2$ error, $\theta = 0.75$ (h) $H^1$ error, $\theta = 0.75$ (i) LP error, $\theta = 0.75$

Figure 3: The $L^2$, $H^1$ and LP error convergence comparison over Uniform, RG, RGB and NVB refinement regimes for Example 2 with $\theta = 0.25, 0.5, 0.75$ for each row, respectively.

*Calcolo* 38.4 (2001), pp. 173–199. DOI: 10.1007/s10092-001-8180-4 (cit. on p. C139).

[2] M. Braack and E. Burman. "Local projection stabilization for the Oseen problem and its interpretation as a variational multiscale method". In: *SIAM J. Numer. Anal.* 43.6 (2006), pp. 2544–2566. DOI: 10.1137/050631227 (cit. on p. C139).

[3] D. Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, 2001. URL: https://www.cambridge.org/au/academic/subjects/mathematics/numerical-analysis/finite-elements-theory-fast-solvers-and-applications-solid-mechanics-3rd-edition?format=PB (cit. on p. C141).

[4] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer–Verlag, 1994. DOI: 10.1007/978-0-387-75934-0 (cit. on p. C141).

[5] C. Carstensen, M. Feischl, M. Page, and D. Praetorius. "Axioms of adaptivity". In: *Comput. Math. Appl.* 67.6 (2014), pp. 1195–1253. DOI: 10.1016/j.camwa.2013.12.003. (Cit. on p. C147).

[6] L. Chen, P. Sun, and J. Xu. "Multilevel homotopic adaptive finite element methods for convection dominated problems". In: *Domain Decomposition Methods in Science and Engineering*. Ed. by T. J. Barth, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, T. Schlick, R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu. Springer, 2005, pp. 459–468. DOI: 10.1007/3-540-26825-1_47 (cit. on p. C140).

[7] S. A. Funken and A. Schmidt. "Adaptive mesh refinement in 2D—An efficient implementation in Matlab". In: *Comput. Meth. Appl. Math.* 20.3 (2020), pp. 459–479. DOI: 10.1515/cmam-2018-0220. (Cit. on p. C146).

[8] D. Gilbarg and N. S. Trudinger. *Elliptic partial differential equations of second order,* Springer-Verlag, 2001. DOI: 10.1007/978-3-642-61798-0 (cit. on p. C141).

[9] V. John, P. Knobloch, and J. Novo. "Finite elements for scalar convection-dominated equations and incompressible flow problems: A never ending story?" In: *Comput. Visual. Sci.* 19.5 (2018), pp. 47–63. DOI: 10.1007/s00791-018-0290-5 (cit. on p. C139).

[10] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Dover Books on Mathematics. Dover Publications, 2012. URL: https://store.doverpublications.com/048646900x.html (cit. on p. C139).

[11] B. P. Lamichhane. "Higher order mortar finite elements with dual Lagrange multiplier spaces and applications". PhD thesis. University of Stuttgart, 2006. DOI: 10.18419/opus-4770 (cit. on p. C142).

[12] B. P. Lamichhane and J. A. Shaw-Carmody. "A local projection stabilisation for convection-diffusion-reaction equations using biorthogonal systems". In: *J. Comput. Appl. Math.* 393, 113542 (2020). DOI: 10.1016/j.cam.2021.113542 (cit. on p. C143).

[13] G. Matthies, P. Skrzypacz, and L. Tobiska. "Stabilization of local projection type applied to convection-diffusion problems with mixed boundary conditions". In: *Elec. Trans. Numer. Anal.* 32 (2008), pp. 90–105. URL: https://etna.math.kent.edu/volumes/2001-2010/vol32/abstract.php?vol=32&pages=90-105 (cit. on p. C139).

[14] H.-G. Roos, M. Stynes, and L. Tobiska. *Robust Numerical Methods for Singularly Perturbed Differential Equations: Convection-Diffusion and Flow Problems.* Springer, 2008. DOI: 10.1007/978-3-540-34467-4 (cit. on p. C139).

[15] P. Sun, L. Chen, and J. Xu. "Numerical Studies of Adaptive Finite Element Methods for Two Dimensional Convection-Dominated Problems". In: *J. Sci. Comput.* 43 (2010), pp. 24–43. DOI: 10.1007/s10915-009-9337-6 (cit. on p. C140).

## Author addresses

1. **Bishnu P. Lamichhane**, School of Information & Physical Sciences, University of Newcastle, University Drive, Callaghan, NSW 2308, Australia
mailto:Bishnu.Lamichhane@newcastle.edu.au

2. **Jordan Shaw-Carmody**, School of Information & Physical Sciences, University of Newcastle, University Drive, Callaghan, NSW 2308, Australia
mailto:Jordan.Shaw-Carmody@uon.edu.au