A Thesis submitted for the degree of Doctor of Philosophy in the Faculty of Science

Mixed Formulations for the Convection-Diffusion Equation

Heather Yorston

2020

Department of Mathematics and Statistics
University of Strathclyde
Glasgow, UK

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed:

Date:

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor Dr Gabriel Barrenechea for all his guidance, knowledge an patience in enabling me to complete this work.

I would also like to thank my husband for his support during the numerous years that this work has taken. Also thanks for the support and encouragement of my daughter and son and my friends within the department and outwith.

Finally, I am grateful for funding received from the Centre for Numerical Analysis and Intelligent Software (NAIS) that supported this work.

Abstract

This thesis explores the numerical stability of the stationary Convection-Diffusion-Reaction (CDR) equation in mixed form, where the second-order equation is expressed as two first-order equations using a second variable relating to a derivative of the primary variable. This first-order system uses either a total or diffusive flux formulation. We start by numerically testing the unstabilised Douglas and Roberts classical discretisation of the mixed CDR equation using Raviart-Thomas elements. The results indicate that, as expected, for both total and diffusive flux, the stability of the formulation degrades dramatically as diffusion decreases.

Next, we investigate stabilised formulations that are designed to improve the ability of the discrete problem to cope with problems containing layers. We test the Masud and Kwack method that uses Lagrangian elements but whose analysis has not been developed. We then significantly modify the formulation to allow us to prove existence of a solution and facilitate the analysis. Our new method, which uses total flux, is then tested for convergence with standard tests and found to converge satisfactorily over a range of values of diffusion.

Another family of first-order methods called First-Order System of Least-Squares (FOSLS/LSFEM) is also investigated in relation to solving the CDR equation. These symmetric, elliptic methods do not require stabilisation but also do not cope well with sharp layers and small diffusion. Modifications have been proposed and this study includes a version of Chen et al. which uses diffusive flux, imposing boundary conditions weakly in a weighted formulation.

We test our new method against all the aforementioned methods, but we find that other methods do not cope well with layers in standard tests. Our method compares favourably with the standard Streamline-Upwind-Petrov-Galerkin method (SUPG/SDFEM), but overall is not a significant improvement. With further fine-tuning, our method could improve but it has more computational overhead than SUPG.

Chapter 1

Introduction

The combination of the differing scales of convection and diffusion in the second-order Convection-Diffusion-Reaction (CDR) equation provides us with a challenging problem. The typical form of the stationary CDR equation is: find a scalar variable p such that

$$\begin{cases} -\varepsilon\Delta p + \boldsymbol{\alpha}\cdot\nabla p + \mu\, p &= f \quad \text{in } \Omega, \\ p &= 0 \quad \text{on } \Gamma := \partial\Omega, \end{cases} \tag{1.1}$$

where $\Omega \subseteq \mathbb{R}^d$ with $d = 2, 3$ is a polygonal open domain with a Lipschitz boundary Γ, $\boldsymbol{\alpha}$ is a solenoidal convective field ($\nabla \cdot \boldsymbol{\alpha} = 0$) for incompressible flow in Ω, $\varepsilon > 0$ is a constant diffusion coefficient, $\mu \geq 0$ is a scalar reaction coefficient and the source term f.
A typical situation modelled by this equation would be the spread of pollution in a river, where there is a strong current, this is convection (sometimes called advection), ongoing diffusion of the pollution molecules and some chemical reaction, with a possible source producing more pollution.

The challenge is to produce a stable approximation for convection-dominated problems that copes well with sharp layers but does not smear the solution. This has continued to be a great unsolved problem in over the last 30 years, which Stynes has compared to the thirty years war in Europe [Sty13] and is described by John et al. as the 'never-ending story' [JKN18]. In many cases the Streamline–Upwind Petrov-Galerkin (SUPG), or Streamline–Diffusion Finite Element Method (SDFEM) as it is alternatively named, which first emerged in 1982 [BH82, HMA86] , still remains the method of choice as it produces sharp layers. However, in spite of refinements and the use of Shiskin meshes to try and capture the layers [LS01, ST03], it also produces non-physical over- and undershoots in regions close to the layers [Kop04]. The layers were further classified

into outflow (exponential boundary) layers [LS01], characteristic (parabolic) boundary [ST03] or interior layers [FLR08] to find parameters for constructing the Shiskin meshes. Further consideration of corner singularities is added in [KS05, KS07, FKS12]. However, it is difficult to choose a value a priori that works for all layers. This has led to the development of techniques, such as adaptive anisotropic meshing, which use a posteriori techniques in order to automatically detect the layers and prevent non-physical oscillations [FMP04, NGJB09, HJVC14, DXY18].

Other different methods have been suggested including Continuous Interior Penalty methods [BH04], Local Projection Stablisation [Kno10], SOLD methods with shock-capturing related ideas [JK07, JK08] and co-volume methods which mix finite element and finite volume methods [BMO96, SS97, CJMR97, BMM$^+$06, RST08].

There has been an emergence of Discontinuous Galerkin (DG) methods applied to the CDR problem [CS01, Kan08]. These methods can deal with discontinuities but have high numbers of degrees of freedom. However, it is easy to solve DG formulations in parallel which makes the use of super-computers possible. We will not include the large field of DG methods [AM09, YHK13] or Hybrid DG methods [NPC09, DF16] in this study instead we will focus on unstabilised and stabilised solutions.

A relatively recent, comprehensive study by Augustin et al [ACF$^+$11] in 2011, finished with the conclusion, that a fresh look should be taken at the methods for solving the Convection–Diffusion–Reaction (CDR) problem, as little overall progress had been made. A more recent survey of current methods [JKN18], which included the words 'a never ending story' in its title, concurred with this conclusion. The authors searched for an ideal discretisation where the numerical solution satisfied three properties: it possessed sharp layers; it did not exhibit spurious oscillations; and there was an efficient computational method. They concluded that there was currently no method that satisfied all of these properties.

So the search for improvement continues and latest methods using the Discrete Maximum Principle developed by [MH85] and improved by [BE05a, Kno06a], together with algebraic flux-correction schemes, can be found in [Kno06b, BE05b, BH14, BBK17, BJK17].

An alternative method which is explored in this thesis, is to start by rewriting the problem as a first-order system of two linear equations, often referred to as the Mixed Finite Element Method (MFEM). One method of doing this is to write the total flux $\boldsymbol{v} = -\varepsilon\nabla p + \boldsymbol{\alpha} p$ as an independent variable, so that Equation (1.1) becomes

$$\begin{cases} \boldsymbol{v} + \varepsilon\, \nabla p - \boldsymbol{\alpha} p = \boldsymbol{0} & \text{in } \Omega, \\ \nabla \cdot \boldsymbol{v} + \mu\, p = f & \text{in } \Omega, \\ p = 0 & \text{on } \Gamma\,. \end{cases} \tag{1.2}$$

Another method is to use only diffusive flux $\boldsymbol{v} = -\varepsilon\nabla p$ and adjust the equations accordingly. Note that there are now two variables, $\boldsymbol{v}$ a vector variable and p a scalar variable.

To the best of our knowledge, the first time the CDR equation was discretised using the mixed form was by Douglas and Roberts [DR82] and again later in their well-known paper [DR85]. In these studies, the discretisation was carried out using Raviart-Thomas finite elements, which consist of vector basis functions. Neither of these papers contained any numerical tests and both stated that convergence would be obtained, provided that the mesh parameter was 'sufficiently small', but no explicit bounds for h were given.

A few years later, Thomas presented a short paper [Tho87] at the Polytechnic University of Madrid in 1987 during the International Symposium on Numerical Analysis conference. This used Raviart-Thomas finite elements again and included 'jump' terms for the scalar variable as a stabilisation method.

Despite the length and body of work dedicated to mixed formulations (see the recent monograph [BBF13] for a review), little attention seems to have been paid to the mixed formulation of the CDR equation. Thus, the mixed method, while popular for solving other elliptic equations, in particular Stokes equations, seems to have been discarded for solving the CDR equation, but without any clearly published reasons.

Another branch of accepted first-order system methods for the CDR problem called First-Order Least Squares (FOSLS) or Least Squares FEM (LSFEM) has many adherents and a large volume of literature [CLMM94, CMM97, CMMR01, BG09, FMM98]. The advantages of FOSLS methods is that they propose a weak formulation that is an elliptic problem rather than the saddle-point problem of MFEM and thus the choice of

finite element spaces is freed. As a result, no stabilisation is necessary and no consideration of the inf-sup condition is needed; however, the disadvantage is that the method is too diffusive with a problem containing both interior and boundary layers [HY09]. Various modifications of the basic formulations have been tried; exponentially-weighted least-squares[FMM98], an added curl equation [JP93], interior bubbles [HY10], use of the adjoint equation [CMMR01] and streamline diffusion[LTV97] amongst others. This thesis includes a recent formulation by Chen et al. [CFLQ14] which imposes the boundary conditions weakly in order to try to achieve better adaptation for problems with layers.

In 2008 Masud and Kwack published a new stabilised method [MK08] using the MFEM approach and Lagrangian elements for both variables. Several tests of this new method were included in [MK08] but no analysis was included.

This study came to our attention and we began to investigate the convergence of this method [MK08] numerically; firstly modifying LehrFEM [BFM14] (a finite element training package produced by ETHZ) and secondly using FreeFEM++[Hec12]. After establishing that the MK method did have convergent properties but was not stable in various tests with layers, we looked for a way to improve on this method. Accordingly, an attempt was made to analyse the MK method but this proved intractable. This analytic challenge together with extensive numerical tests, led to the new method proposed in Chapter 4, which can be seen as an extension of [MK08] but also has unique features and the analysis in this case proved to be tractable.

1.1 Outline

We end this chapter with the outline of this thesis.

In Chapter 2 we look at the background for solving the second order CDR Equation (1.1) and introduce the SUPG method, which will be used as a benchmark for our new method. We introduce the unstabilised mixed formulations of [DR82, DR85], the Raviart-Thomas finite elements and some simple tests which will be used for the initial investigations of the DR methods and the modification of Thomas [Tho87]. We investigate the convergence and stability of the MK method [MK08] and examine some of the

standard FOSLS approaches. The FOSLS method [CFLQ14] that will be used in the comparative studies is introduced.

In Chapter 3 we investigate the inf-sup constant for the total flux method of [DR82, DR85] in order to find any connection between the size of the mesh parameter and the diffusion term. This could help us come to a conclusion about the statement in [DR82, DR85] about convergence being obtained when the mesh parameter is 'sufficiently small' and our difficultly in obtaining convergence in numerical tests of Chapter 2 .

Then, in Chapter 4 we set out the analyses for a new stabilised method and test its convergence with some standard tests. This chapter is based on published joint work in [BPY18], augmented with extra convergence tests.

Our new method is computationally tested further in Chapter 5 and a computational review of mixed methods for solving CDR is conducted. The new method's performance is compared to that of SUPG and the FOSLS method of Chen [CFLQ14] using more exacting tests involving boundary and interior layers. Other methods could not be included due to failing the stringent nature of these tests. This chapter has also been published in [BPY18]

We draw our conclusions and present possible further extensions in Chapter 6.

Finally, we include an Appendix with FreeFem++ code for the implementation of the present method and other computational methods used in this thesis.

Chapter 2

Background

In this chapter we introduce the standard stationary, second-order formulation of the Convection Diffusion Reaction (CDR) equation and its solution by the Streamling Upwind Petrov-Galerkin (SUPG) method. We then give the background to the alternative solution using a system of first-order equations, or as they are also called, 'mixed' formulations for the CDR equation and their solution.
We numerically test the classical, unstabilised Douglas and Roberts formulation and the modified stabilisation of Thomas which both used Raviart-Thomas elements. We then move on to numerical tests on the stabilised formulation of Masud and Kwack which uses Lagrangian elements. We finish by introducing first-order least-squares methods (FOSLS) and the chosen formulation for our comparative study in the sequel.

2.1 Notation

Standard notations are used in this thesis for Sobolev spaces and corresponding norms. In particular, the inner product in $L^2(\Omega)$ or $L^2(\Omega)^d$ is denoted by $(\cdot,\cdot)$, where $\Omega \subseteq \mathbb{R}^d$ with $d = 2, 3$ as a polygonal open domain with a Lipschitz boundary $\partial\Omega := \Gamma$.

The norm and semi-norm in $W^{m,p}(\Omega)$ will be denoted by $\|\cdot\|_{m,p,\Omega}$ and $|\cdot|_{m,p,\Omega}$, respectively, with the convention $\|\cdot\|_{m,\Omega} = \|\cdot\|_{m,2,\Omega}$, where $H^m(\Omega) = W^{m,2}(\Omega)$ and $L^2(\Omega) = H^0(\Omega)$. Functions in $H_0^1(\Omega)$ belong to $H^1(\Omega)$ and vanish on the boundary $\partial\Omega$.

The polynomial space $\mathcal{P}_k$ consists of the space of polynomials $p(x)$ in the variables

$x_1, ..., x_n$ with real coefficients $\alpha_{i_1}, ..., \alpha_{i_n}$ and of global degree at most k,

$$\mathcal{P}_k = \left\{ p(x) = \sum_{\substack{0 \le i_1, ... i_n \le k \\ i_1 + ... + i_n \le k}} \alpha_{i_1 ... i_d} x_1^{i_1} ... x_n^{i_n} ; \alpha_{i_1 ... i_d} \in \mathbb{R} \right\},$$

[EG13, p.22].

Let $\{\mathcal{T}_h\}_{h>0}$ be a family of triangulations of Ω, built up using simplices T, diameter $h_T := \operatorname{diam}(T)$, and $h := \max\{h_T : T \in \mathcal{T}_h\}$. A family of triangulations are called shape regular if there exists a constant $M \ge 1$ with $\dfrac{h_T}{\rho_T} < M$ [QV08, p. 90] where

$$\rho_T := \{\operatorname{diam}(S) | S \text{ is a ball contained in } T\} .$$

We also introduce the subspace of $L^2(\Omega)^d$:

$$H(\operatorname{div}, \Omega) = \left\{ \boldsymbol{v} \in L^2(\Omega)^d : \operatorname{div} \boldsymbol{v} \in L^2(\Omega) \right\}.$$

with the associated norm $\|\boldsymbol{v}\|_{\operatorname{div},\Omega} = \{\|\nabla \cdot \boldsymbol{v}\|_{0,\Omega}^2 + \|\boldsymbol{v}\|_{0,\Omega}^2\}^{\frac{1}{2}}$.

For a polynomial order $k \ge 1$, we introduce the finite element space for the flux variable as

$$\boldsymbol{H}_h := \left\{ \boldsymbol{\varphi} \in C^0(\overline{\Omega})^d : \boldsymbol{\varphi}|_T \in \mathcal{P}_k(T)^d \quad \forall T \in \mathcal{T}_h \right\}, \tag{2.1}$$

and the discrete subspace for the scalar variable p as

$$Q_h^0 := Q_h \cap H_0^1(\Omega) \text{ where } Q_h := \left\{ q_h \in C^0(\overline{\Omega}) : q_h|_T \in \mathcal{P}_k(T) , \forall T \in \mathcal{T}_h \right\}. \tag{2.2}$$

The inner product in $L^2(D)$ is denoted by

$$(f, g)_D = \int_D f(x) g(x) \mathrm{d}x , \quad \forall f, g \in L^2(D). \tag{2.3}$$

Norms and inner products for functions in $L^2(D)^2$ or in $L^2(D)^{d\times d}$ are defined componentwise and the same notation is used.
In the case $D = \Omega$, the subscript will be dropped.
The duality pairing between a Hilbert space and its dual space will be denoted by $\langle . , . \rangle$.

We state the Lax–Milgram lemma which will be used in the sequel [EG13, p.133].

Theorem 2.1. *(Lax–Milgram lemma).*
The Problem:

$$\text{Find } u \in V : \mathcal{A}(u,v) = \mathcal{F}(v) \qquad \forall v \in V\,. \tag{2.4}$$

Let V be a (real) Hilbert space, endowed with the norm $\|.\|$, $\mathcal{A}(u,v) : V \times V \to \mathbb{R}$ a bilinear form and $\mathcal{F}(v) : V \to \mathbb{R}$ a linear, continuous functional (i.e. $\mathcal{F} \in V'$, where V' denotes the dual space of V). Assume, moreover, that $\mathcal{A}(.,.)$ is continuous, i.e.

$$\exists\, \gamma \geq 0 : |\mathcal{A}(w,v)| \leq \gamma \|w\|\|v\| \qquad \forall w, v \in V$$

,
and coercive, i.e.

$$\exists\, \alpha \geq 0 : \mathcal{A}(v,v) \geq \alpha \|v\|^2 \qquad \forall v \in V\,.$$

Then, there exists an unique $u \in V$ solution to (2.4) *and*

$$\|u\| \leq \frac{1}{\alpha} \|\mathcal{F}\|_{V'}\,.$$

2.2 The stationary second-order CDR equation and its solution

We consider the stationary *convection–diffusion–reaction problem* in its second-order form: find p such that

$$\begin{cases} -\varepsilon \Delta p + \boldsymbol{\alpha} \cdot \nabla p + \mu\, p &= f \quad \text{in } \Omega, \\ p &= 0 \quad \text{on } \Gamma, \end{cases} \tag{2.5}$$

$\boldsymbol{\alpha} \in L^\infty(\Omega)^d$ is a convective field such that $\nabla \cdot \boldsymbol{\alpha} = 0$ in Ω, $0 < \varepsilon \leq 1$ is a constant diffusion coefficient, $\mu \geq 0$ is a reaction coefficient and f is a source term.

The weak formulation of Equation (2.5) is given by: find $p \in V := H_0^1(\Omega)$ such that

$$a(p,q) := \varepsilon(\boldsymbol{\nabla} p, \boldsymbol{\nabla} q) + (\boldsymbol{\alpha} \cdot \boldsymbol{\nabla} p, q) + \mu(p,q) = (f,q)\,, \tag{2.6}$$

for all $q \in V$.

The bilinear form $a(.,.)$ is elliptic in $H_0^1(\Omega)$ since

$$a(q,q) = \varepsilon |q|_{1,\Omega}^2 + \mu \|q\|_{0,\Omega}^2 \tag{2.7}$$

for all $q \in H_0^1(\Omega)$. This leads to the following stability result for the solution of (2.5) for some constant C

$$\varepsilon |p|_{1,\Omega}^2 + \mu \|p\|_{0,\Omega}^2 \leq C \min\left\{\frac{1}{\varepsilon}, \frac{1}{\mu}\right\} \|f\|_{0,\Omega}^2 \,, \tag{2.8}$$

Clearly, for $\varepsilon \ll 1$, and especially when $\mu = 0$, this gives very weak control on the solution in the $H^1(\Omega)$ norm. In fact, since the problem in Equation (2.6) with $\mu = 0$ is linear, for $f_1, f_2 \in L^2(\Omega)$, and denoting the solutions of Equation (2.6) for the right-hand sides f_1 and f_2 by $p_{f_1}, p_{f_2} \in H_0^1(\Omega)$, we conclude from Equation (2.8) that

$$|p_{f_1} - p_{f_2}|_{1,\Omega} \leq \frac{C}{\varepsilon} \|f_1 - f_2\|_{0,\Omega} \,. \tag{2.9}$$

This result in Equation (2.9) states that p, the solution of Equation (2.6), depends continuously on the right-hand side f. Nevertheless, for $\varepsilon \ll 1$, the continuity constant $\frac{C}{\varepsilon}$ is extremely large, with the result that small changes in the right-hand side can lead to extremely large changes in the solution. This phenomenon is referred to as 'weak stability'.

The standard Galerkin method for Equation (2.6) in the discrete space $V_h \subset V$ reads: find $p_h \in V_h$ such that

$$a(p_h, q_h) = (f, q_h), \qquad \forall q_h \in V_h \tag{2.10}$$

where $a(.\,,.)$ is defined in Equation (2.6).

This discrete problem, Equation (2.10), inherits the same problem of weak control as Equation (2.6). This is evident especially when layers are present, as the gradient of p_h is only bounded by a negative power of ε and so spurious oscillations are usually present. Figure 2.1 shows a 3-D visualisation of the Galerkin method tested with the 'advection skew to the mesh' (Test B) outlined in Section 2.4.2. The instablility of the method with using $\varepsilon = 10^{-4}$ and a mesh that does not resolve the layers, is clearly visible and can be compared with the SUPG solution in Figure 2.4b.

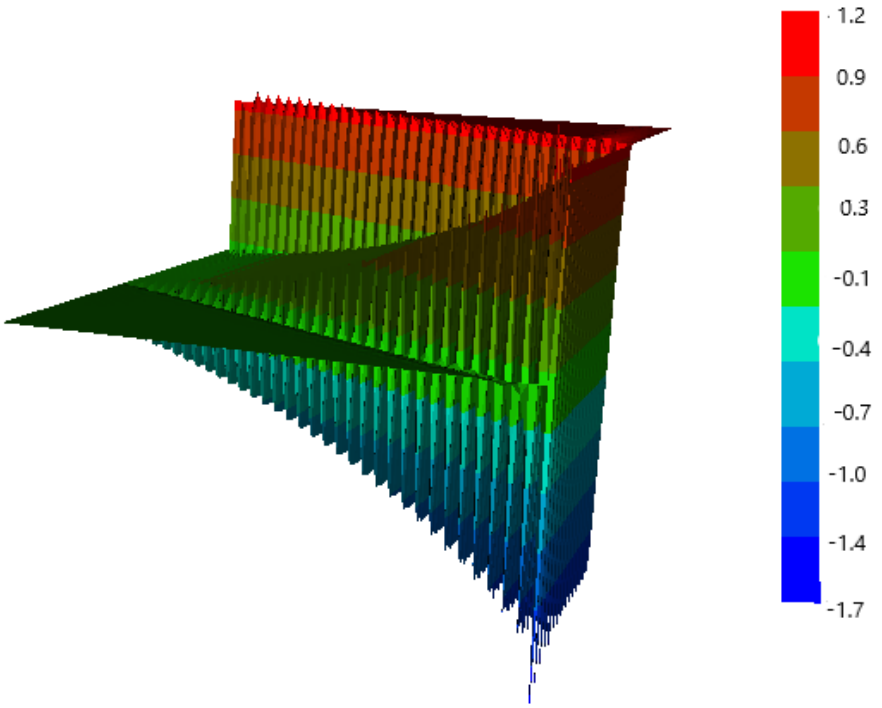


Fig: 2.1 Unstabilised Galerkin method for Test B with layers $\varepsilon = 10^{-4}$, mesh 64×64 of piecewise linear, $\mathcal{P}_1$, Lagrangian elements, Number of triangles = 8192

2.2.1 SUPG/SDFEM

The Streamline Diffusion method (SDFEM) or Streamline Upwind Petrov-Galerkin (SUPG) [BH82] is the most commonly accepted standard method of stabilisation for the second-order form. It stabilises the solution by adding diffusion in the upwind direction of the streamlines to the Galerkin method Equation (2.10). This adds another term to the left-hand-side of Equation (2.6) and the discrete formulation becomes: find $p_h \in V_h$ such that

$$a(p_h, q_h) + a_s(p_h, q_h) = (f, q_h) \tag{2.11}$$

with $a_s(p_h, q_h) = \sum_{T \in \mathcal{T}_h} (R_h(p_h), \tau_T \boldsymbol{\alpha} \cdot \boldsymbol{\nabla} q_h)_T$,
for all $q_h \in V_h$, where $R_h(p_h) = -\varepsilon \Delta p_h + \boldsymbol{\alpha} \cdot \boldsymbol{\nabla} p_h + \mu p_h - f$ is the discrete residual of the strong form of Equation (2.5) and τ_T is a parameter introduced to improve the stability of the method.
The stabilisation used in this thesis is that recommended in a similar comparative study of the second-order system [FFH92]

$$\tau_T = \frac{h_T}{2|\boldsymbol{\alpha}|} \min \left\{ 1, \frac{m_k h_T |\boldsymbol{\alpha}|}{2\varepsilon} \right\} \qquad \forall\, T \in \mathcal{T}_h \,, \tag{2.12}$$

and m_k is a constant appearing in an inverse inequality related to Equation (4.6) (for details, see [FFH92]).
Let $\omega \geq 0$ such that

$$\omega = \min \left\{ -\frac{1}{2} \nabla \cdot \boldsymbol{\alpha}(\boldsymbol{x}) + \mu(x) : x \in \Omega \right\} .$$

Then we define the SUPG norm in V_h by (cf. [JN13])

$$|||p_h|||_{SUPG} := \left\{ \varepsilon |p_h|^2_{1,\Omega} + \omega \|p_h\|^2_{0,\Omega} + \sum_{T \in \mathcal{T}_h} \tau_T \|(\boldsymbol{\alpha} \cdot \nabla p_h)\|^2_{0,T} \right\}^{\frac{1}{2}} . \tag{2.13}$$

This is stronger than the norm for the (unstabilised) Galerkin method, which would not have the last term on the right-hand side.

2.3 The CDR equation and the Mixed Formulation

There are two alternative first-order methods of rewriting Equation (2.5): one method uses the combined total convective–diffusive flux and the other solely the convective flux.

In this work we follow the approach of defining the total flux by $\boldsymbol{v} = -\varepsilon\nabla p + \boldsymbol{\alpha} p$ as an independent variable, so that Equation (2.5) becomes

$$\begin{cases} \dfrac{1}{\varepsilon}\,\boldsymbol{v} + \nabla p - \dfrac{1}{\varepsilon}\,\boldsymbol{\alpha} p = \boldsymbol{0} & \text{in } \Omega, \\ \nabla \cdot \boldsymbol{v} + \mu\, p = f & \text{in } \Omega, \\ p = 0 & \text{on } \Gamma. \end{cases} \tag{2.14}$$

With this notation, multiplying the first equation in Equation (2.14) by $\boldsymbol{w} \in H(\mathrm{div}, \Omega)$ and integrating by parts, the weak formulation of Equation (2.14) is given by: find $(\boldsymbol{v}, p) \in \boldsymbol{V} \times Q := H(\mathrm{div}\,, \Omega) \times L^2(\Omega)$ such that

$$\frac{1}{\varepsilon}\,(\boldsymbol{v}, \boldsymbol{w}) - (p, \nabla \cdot \boldsymbol{w}) - \frac{1}{\varepsilon}\,(\boldsymbol{\alpha}\, p, \boldsymbol{w}) + (\nabla \cdot \boldsymbol{v}, q) + \mu\,(p, q) \;=\; (f, q)\,, \tag{2.15}$$

for all $(\boldsymbol{w}, q) \in \boldsymbol{V} \times Q$.

An alternative formulation arises if, instead of the total flux, the diffusive flux $\boldsymbol{v}_d = -\varepsilon\nabla p$ is introduced as the extra unknown. This has been done in [DR82, Tho87, CFLQ14]. In this case the first order system for Equation (2.5) becomes

$$\begin{cases} \dfrac{1}{\varepsilon}\,\boldsymbol{v}_d + \nabla p = \boldsymbol{0} & \text{in } \Omega, \\ \nabla \cdot \boldsymbol{v}_d + \nabla p \cdot \boldsymbol{\alpha} + \mu\, p = f & \text{in } \Omega, \\ p = 0 & \text{on } \Gamma. \end{cases} \tag{2.16}$$

Similarly, multiplying the first equation in Equation (2.16) by $\boldsymbol{w} \in \boldsymbol{V}$ and integrating by parts, together with the substitution of $-\frac{1}{\varepsilon}\boldsymbol{v}_d$ for ∇p into the second equation, leads to the weak variational form for Equation (2.16) becoming: find $(\boldsymbol{v}_d, p) \in \boldsymbol{V} \times Q$ such that

$$\frac{1}{\varepsilon}\,(\boldsymbol{v}_d, \boldsymbol{w}) - (p, \nabla \cdot \boldsymbol{w}) + (\nabla \cdot \boldsymbol{v}_d, q) - \frac{1}{\varepsilon}\,(\boldsymbol{\alpha} \cdot \boldsymbol{v}_d, q) + \mu\,(p, q) \;=\; (f, q)\,, \tag{2.17}$$

for all $(\boldsymbol{w}, q) \in \boldsymbol{V} \times Q$.

Remark 2.2. *Using the Lax-Milgram Lemma,Theorem* (2.1)*, it can be proven that Equation* (2.5)*, with* $-\frac{\nabla\cdot\boldsymbol{\alpha}}{2} + \mu \geq 0$*, has a unique weak solution* $p \in H_0^1(\Omega)$*; see [EG13] for details. Thus, the existence and uniqueness of solution of the problem in Equations* (2.15)*, or* (2.17)*, follow from the fact that a solution of either of these problems is a weak solution of Equation* (2.5)*, and vice–versa.*

2.3.1 Solving the Mixed Formulation: A general framework

When discussing the linear algebraic challenges of solving the mixed formulation, sets of equations such as Equations (2.15) and (2.17) or Stokes (given in Equation (2.19)), are typically represented in a general framework (see [BG03, BGL05, CJHZ03, BBF13, QV08]) for some specified Hilbert spaces $\boldsymbol{V}$ and Q and their dual spaces $\boldsymbol{V}'$ and Q'. Introducing the bilinear forms
$a : \boldsymbol{V} \times \boldsymbol{V} \to \mathbb{R}$, $b_1 : \boldsymbol{V} \times Q \to \mathbb{R}$, $b_2 : \boldsymbol{V} \times Q \to \mathbb{R}$, $c : Q \times Q \to \mathbb{R}$ and given $\boldsymbol{g} \in \boldsymbol{V}'$ and $f \in Q'$: find $(\boldsymbol{v}, p) \in \boldsymbol{V} \times Q$ the solution of

$$\begin{cases} a(\boldsymbol{v}, \boldsymbol{w}) + b_1(\boldsymbol{w}, p) = (\boldsymbol{g}, \boldsymbol{w}) & \forall \boldsymbol{w} \in \boldsymbol{V} \\ b_2(\boldsymbol{v}, q) + c(p, q) = (f, q) & \forall q \in Q \ . \end{cases} \tag{2.18}$$

In our case of the CDR Equations (2.15) and (2.17), $a(\boldsymbol{v}, \boldsymbol{w}) = \dfrac{1}{\varepsilon}(\boldsymbol{v}, \boldsymbol{w})$, $\boldsymbol{g} = \boldsymbol{0}$ and $c(p, q) = \mu(p, q)$. The $b_1(.\,,.)$ and $b_2(.\,,.)$ terms will vary according to whether total or diffusive flux is used for $\boldsymbol{v}$, as this determines the location of the convective flux term. In the case of total flux,

$$b_1(\boldsymbol{w}, p) = -(p, \nabla \cdot \boldsymbol{w}) - \frac{1}{\varepsilon}\,(\boldsymbol{\alpha}\, p, \boldsymbol{w}) \quad \text{and} \quad b_2(\boldsymbol{v}, q) = (\nabla \cdot \boldsymbol{v}, q)\,.$$

Alternatively, in the case of diffusive flux,

$$b_1(\boldsymbol{w}, p) = -(p, \nabla \cdot \boldsymbol{w}) \quad \text{and} \quad b_2(\boldsymbol{v}, q) = (\nabla \cdot \boldsymbol{v}, q) - \frac{1}{\varepsilon}\,(\boldsymbol{\alpha} \cdot \boldsymbol{v}, q)\,.$$

Remark 2.3. *There is a large volume of literature concerning the solution of the Stokes equations, given in Equation* (2.19)*, which control the flow of a steady, viscous, incompressible, Newtonian fluid (see [BBF13, EG13, BGL05, BDG06]) and arise from a simplification of the incompressible Navier-Stokes equations without the presence of a convective term.*

$$\begin{cases} -\Delta \boldsymbol{v} + \boldsymbol{\nabla} p = \boldsymbol{g} & \text{in } \Omega \\ \nabla \cdot \boldsymbol{v} = 0 & \text{in } \Omega, \\ \boldsymbol{v} = 0 & \text{on } \Gamma, \end{cases} \tag{2.19}$$

where $\boldsymbol{v}$ and p are the velocity and pressure of the fluid, respectively and $\boldsymbol{g}$ is a given vector function.

A weak formulation of the Stokes problem is given here for comparison with Equation (2.18) *and will be referred to again in Chapter 3: find* $\boldsymbol{v} \in \boldsymbol{W} := [H_0^1(\Omega)]^2$ *and* $p \in Q_0 := \{q \in L^2(\Omega) : \int_\Omega q\, dx = 0\}$*, such that*

$$\begin{cases} \tilde{a}(\boldsymbol{v}, \boldsymbol{w}) + \tilde{b}(\boldsymbol{w}, p) &= (\boldsymbol{g}, \boldsymbol{w}) \quad \forall \boldsymbol{w} \in \boldsymbol{W}, \\ \tilde{b}(\boldsymbol{v}, q) &= 0 \qquad \forall q \in Q_0, \end{cases} \tag{2.20}$$

where $\tilde{a}(\boldsymbol{v}, \boldsymbol{w}) = (\boldsymbol{\nabla v}, \boldsymbol{\nabla w})$ *and* $\tilde{b}(\boldsymbol{v}, q) = -(\nabla \cdot \boldsymbol{v}, q)$ *.*

In the Stokes formulation, Equation (2.20)*, there is a single bilinear term* $\tilde{b}(.\,,.)$ *due to the symmetry of the formulation. However, in the case of the CDR equation, the bilinear forms, which are denoted by* $b_1(.\,,.)$ *and* $b_2(.\,,,)$*, are not the same due to the different location of the convection term in the cases of total flux and diffusive flux.*
It is known (see [GR86]) that the problem in Equation (2.20) *is well-posed if and only if*

$$\inf_{q \in Q} \sup_{\boldsymbol{v} \in \boldsymbol{V}} \frac{\tilde{b}(\boldsymbol{v}, q)}{\|\boldsymbol{v}\|_{\boldsymbol{V}} \|q\|_Q} \geq \beta_\Omega > 0 \,. \tag{2.21}$$

This condition Equation (2.21) *is termed the Ladyzhenskaya-Babuska-Brezzi (LBB) constraint [BBF13] and is also commonly referred to as the inf-sup condition.*

Returning to the case of the CDR equation, we introduce the following continuous operators: $A : \boldsymbol{V} \to \boldsymbol{V'}$; $B_1 : \boldsymbol{V} \to Q'$ with its adjoint $B_1' : Q \to \boldsymbol{V'}$; $B_2 : \boldsymbol{V} \to Q'$; and $C : Q \to Q'$. (Here $\boldsymbol{V} = H(\mathrm{div}, \Omega), Q = L^2(\Omega)$ and $\boldsymbol{V'}$ and Q' denote their dual spaces.) Then

$$\begin{aligned} \langle A\boldsymbol{v}\,,\, \boldsymbol{w} \rangle_{\boldsymbol{V'} \times \boldsymbol{V}} &= a(\boldsymbol{v}, \boldsymbol{w}), \\ \langle B_1' p\,,\, \boldsymbol{w} \rangle_{\boldsymbol{V'} \times \boldsymbol{V}} &= b_1(\boldsymbol{w}, p), \\ \langle B_2 \boldsymbol{v}\,,\, q \rangle_{Q' \times Q} &= b_2(\boldsymbol{v}, q) \,, \\ \langle Cp\,,\, q \rangle_{Q' \times Q} &= c(p, q) \,. \end{aligned}$$

For $\boldsymbol{g} = \boldsymbol{0}$ and $f \in Q'$, Equation (2.18) can then be written in operator form as

$$\begin{cases} A\boldsymbol{v} + B_1' p = \boldsymbol{0} & \text{in } \boldsymbol{V'} \\ B_2 \boldsymbol{v} + Cp = f & \text{in } Q' \,. \end{cases} \tag{2.22}$$

The corresponding matrix equation is

$$\begin{bmatrix} A & B_1' \\ B_2 & C \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ f \end{bmatrix} . \tag{2.23}$$

When the reaction term is zero and there are no additional stabilisation terms, then $C = 0$ and Equation (2.23) reduces to

$$\begin{bmatrix} A & B_1' \\ B_2 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ f \end{bmatrix} \qquad \in \boldsymbol{V}' \times Q' . \tag{2.24}$$

The matrix in Equation (2.24) is not positive-definite and more sophisticated tools have appeared to analyse the stability of this type of problem and will be discussed in Chapter 3.

2.4 Preliminary tests used in this Chapter

2.4.1 Test A: testing for convergence of a method with a known solution

We consider a domain $\Omega = (0,1)^2$ and use structured Friedrichs–Keller–type meshes in these computations, as shown in Figure 2.2 where N is the number of segments along one side and the mesh parameter h, where $h = \frac{\sqrt{2}}{N}$ in this case. A series of these meshes are obtained by subdividing the mesh in two along both sides during the mesh refinement. The exact solution for p is chosen as $p(x,y) = \sin(2\pi x)\sin(2\pi y)$ with $\mu = 0$ and we use uniform, constant values of ε in the range 1 to 10^{-5} with homogenous Dirichlet conditions for simplicity.

A value of $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]^T = \left[\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}}\right]^T$ is used (giving $|\boldsymbol{\alpha}| = 1$) as this does not align with the triangular mesh being used. Substituting for p in Equation (2.14) or Equation (2.16), with $\mu = 0$ gives $f(x,y) = 8\varepsilon\pi^2 \sin(2\pi x)\sin(2\pi y) + 2\pi\alpha_1 \cos(2\pi x)\sin(2\pi y) + 2\pi\alpha_2 \sin(2\pi x)\cos(2\pi y)$.
A graph of the exact solution is shown in Figures 2.3a and 2.3b

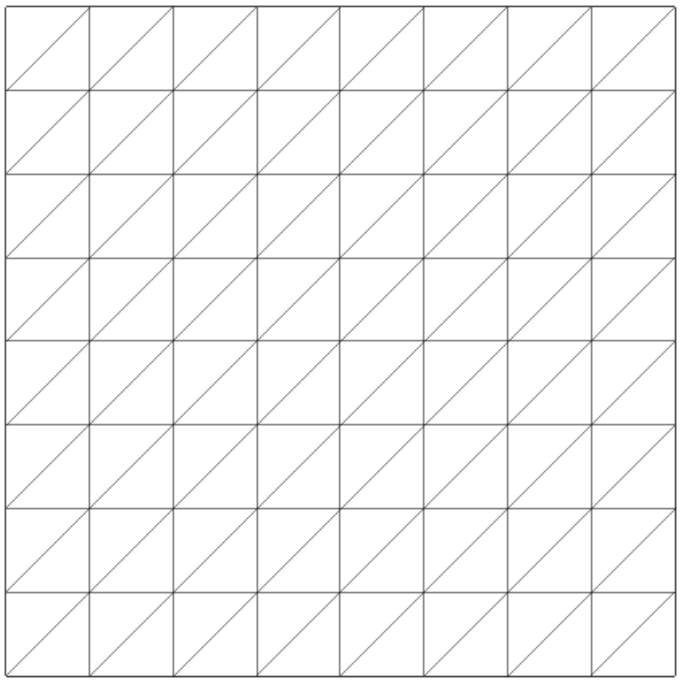
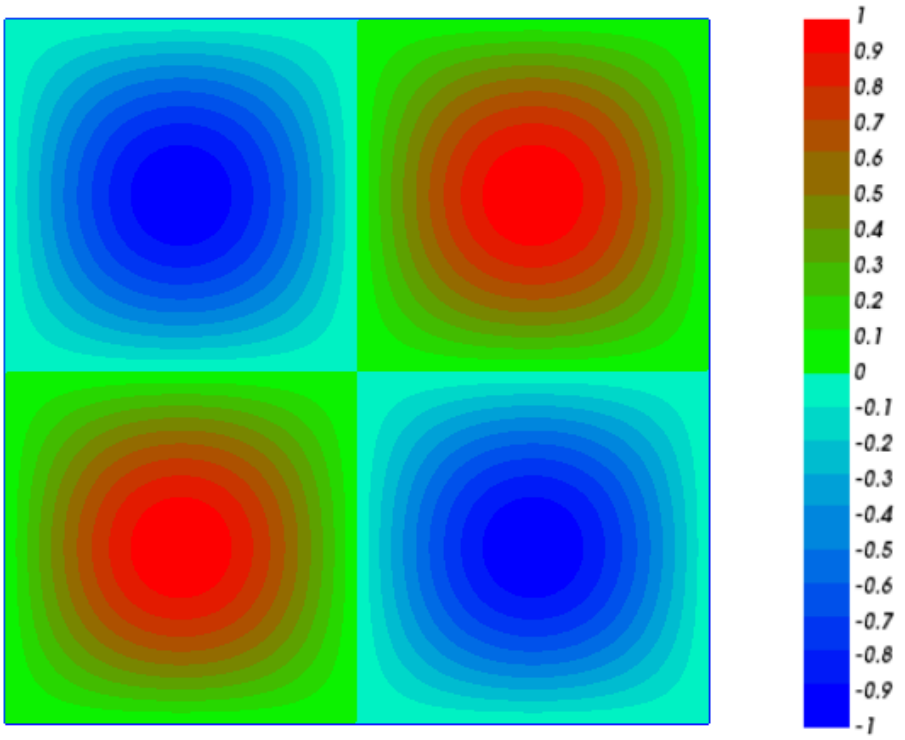
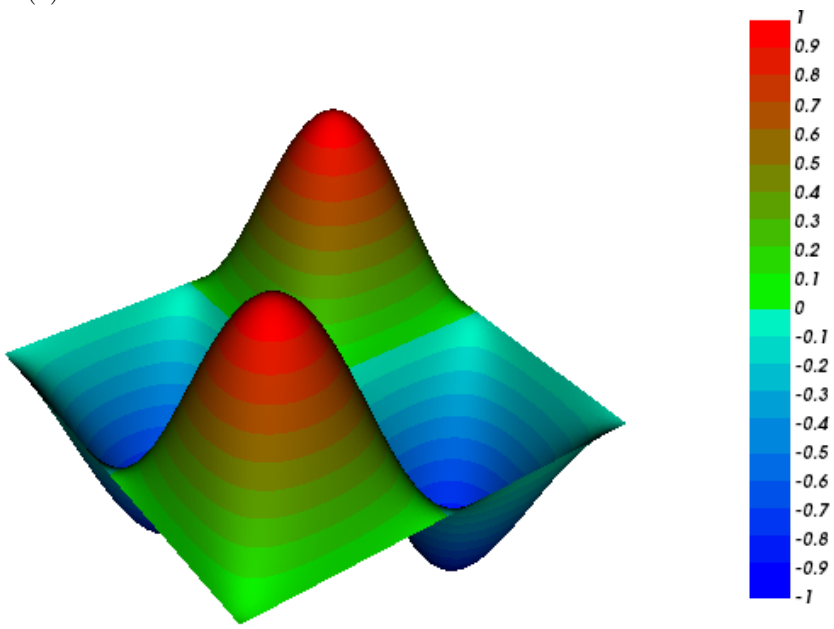


Fig: 2.2 Friedrichs–Keller mesh, $N = 8$



(a) Exact solution for Test A: a 2-D visualistion



(b) Exact solution for Test A: a 3-D visualistion

2.4.2 Test B: Advection skew to the mesh test

This test is designed to show how well a particular method of solution deals with the existence of an interior layer due to the skewing of advection and a boundary layer at the outflow (defined as $\boldsymbol{\alpha}(x) \cdot \boldsymbol{n}(x) > 0$). It is based on a slightly modified version of the test in [BH82]. The advective velocity is chosen as $\boldsymbol{\alpha} = \frac{1}{\sqrt{5}}[1, 2]^T$, and the same family of meshes is used as in Figure 2.2 on the unit square domain of $\Omega = (0, 1)^2$ with $f = 0$, $\mu = 0$ and ε in a range of values from 1 to 10^{-5}. The boundary conditions are given by

$$p(x, y) = \begin{cases} 1 \text{ on } \{(0, y) : 0 \leq y \leq 1\} \cup \{(x, 1) : 0 \leq x \leq 1\} \\ 0 \text{ on } \{(1, y) : 0 \leq y < 1\} \cup \{(x, 0) : 0 < x \leq 1\} \,. \end{cases}$$

The analytical solution to this problem is not known. Therefore, we compare it to a reference solution using the SUPG method (see Section 2.2.1), computed on a very fine mesh using $N = 2^{11}$ (giving 8,388,608 triangles) with quadratic $\mathcal{P}_2$ Lagrangian elements. In Figure 2.4b and Figure 2.4a, we depict 3-D and 2-D visualisations respectively of the Reference Solution. The cross-section depicting the internal layer is taken at $y = 0.5$, shown in Figure 2.5a, and the cross-section depicting the boundary layer at the outlet is taken at $x = 0.7$, shown in Figure 2.5b.

2.5 Raviart–Thomas based mixed methods

The Raviart-Thomas pair of finite elements introduced in [RT77] is one of the first and most popular discrete inf-sup stable pairs for first-order mixed problems.
For a simplex $T \in \mathcal{T}_h$ with $d = 2$ or 3, the $\mathcal{RT}$ space of order k, where $k \geq 0$, is defined as

$$\mathcal{RT}_k(T) = \mathcal{P}_k(T)^d + \boldsymbol{x}\mathcal{P}_k(T)\,, \tag{2.25}$$

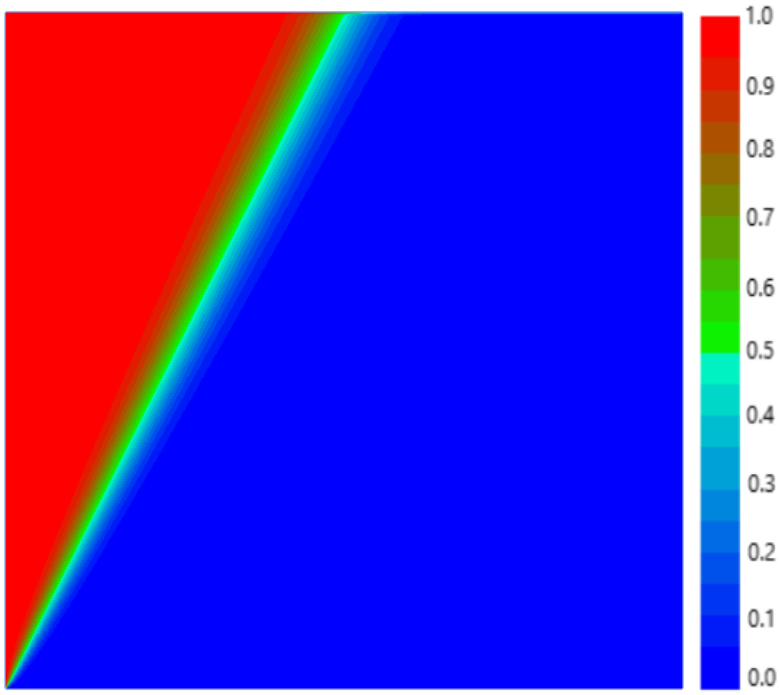
where $\boldsymbol{x}$ are coordinates of the nodes.

Then the associated Raviart-Thomas global space is

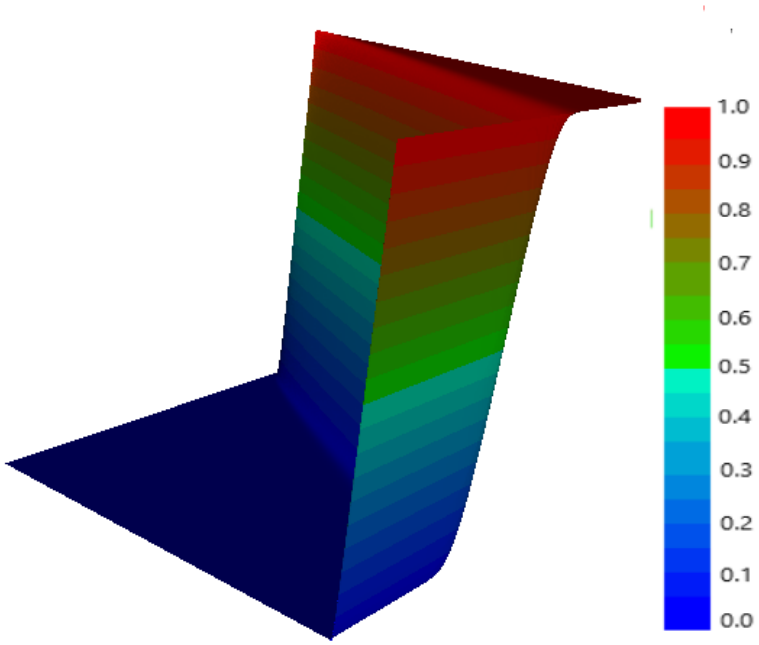
$$\mathcal{RT}_k(\Omega) = \{\boldsymbol{v}_h \in H(\mathrm{div}; \Omega) : \boldsymbol{v}_h|_T \in \mathcal{RT}_k(T),\ \forall T \in \mathcal{T}_h\}\,. \tag{2.26}$$

The primal variable p is approximated using the space of discontinuous piecewise polynomial function of degree $k \geq 0$ given by

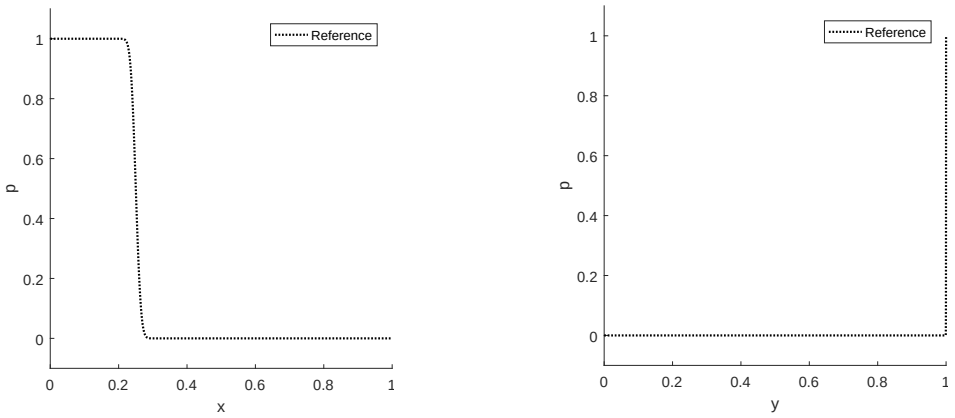
$$\mathcal{P}_k^{dc}(\Omega) = \{q_h \in L^2(\Omega) : q_h|_T \in \mathcal{P}_k(T), \forall T \in \mathcal{T}_h\}\,. \tag{2.27}$$



(a) 2-D visualisation of the reference solution for Test B



(b) 3-D visualisation of the reference solution for Test B



(a) Reference cross-section at $y = 0.5$ (b) Reference cross-section at $x = 0.7$

Fig: 2.5 SUPG $\mathcal{P}_2$ reference solution, $N = 2^{11}$, $\varepsilon = 10^{-4}$, for the advection skew to the mesh test, Test B

This pairing will be referred to as $\mathcal{RT}_k \times \mathcal{P}_k^{\mathrm{dc}}$ with $k \geq 0$. This choice of finite element spaces in the Raviart–Thomas pairs are compatible, so that the discrete LBB constraint is satisfied (for a proof see [RT77]). That is, there exists $\beta > 0$ such that

$$\sup_{\boldsymbol{v}_h \in \mathcal{RT}_k(\Omega)} \frac{(q_h, \nabla \cdot \boldsymbol{v_h})}{\|\boldsymbol{v}_h\|_{\mathrm{div},\Omega}} \geq \beta \|q_h\|_{0,\Omega} \qquad \forall q_h \in \mathcal{P}_k^{dc}(\Omega) \ . \tag{2.28}$$

This inf-sup constraint is examined further in Chapter 3.

2.5.1 Unstabilised Douglas-Roberts discretisation

Douglas and Roberts published a short paper on MFEM [DR82], which showed the existence and uniqueness of the solution and derived the error bounds for both total and diffusive flux, using both the formulations in Equations (2.15) and (2.17). Their analysis was purely theoretical and no attention was paid to the value of ε. Their implementation used the Raviart-Thomas pairs of elements for the analysis, but no numerical studies were included in the paper. They derived error bounds using the properties of Raviart-Thomas elements and the mass matrix for both methods, with the warning that stability and convergence could only be achieved for a mesh parameter 'sufficiently small'.

Later, Douglas and Roberts published a second paper [DR85] which is the more well-known of the two papers. In it they extended and reworked the error analysis, again stating that a unique solution existed for a mesh parameter 'sufficiently small' for the CDR problem. However, this study also did not include any numerical studies.

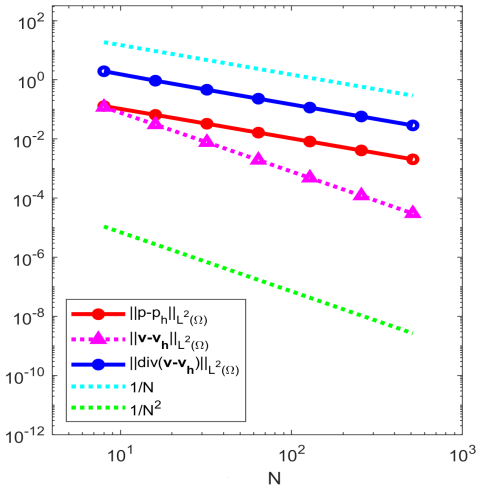
For this present study, extensive numerical studies are carried out on [DR82, DR85] for both total and diffusive flux given in Equations (2.15) and (2.17). Details of the solvers used, the FreeFem++ program listings, quadrature employed and convergence calculations can be found the Appendices A1-3.
The convergence of each formulation to a known solution is investigated to see the effect of mesh refinement using both $\mathcal{RT}_0 \times \mathcal{P}_0$ and $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ pairs of elements with a standard convergence test detailed in Section 2.4.1, called Test A. The 'advection skew to the mesh', called Test B, detailed in Section 2.4.2 is also conducted to test the stability of this method in the presence of layers.
Since [DR82, DR85] use a mixed formulation, the Dirichlet condition, which is imposed strongly in the space for the primal formulation, becomes a natural boundary condition that is imposed within the formulation for both total and diffusive flux formulations. As the stability and error estimates for the mixed method are proven in the $H(\mathrm{div}, \Omega)$ norm for $\boldsymbol{v}$ and the $L^2(\Omega)$ norm for p, we depict their behaviour with respect to h.

This formulation suffers from the same instabilities as the unstabilised Galerkin method and our studies that follow confirm this fact. (It will be referred to as DR in our numerical studies.)

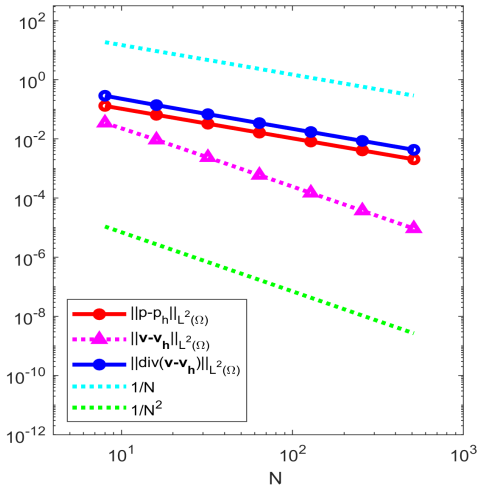
2.5.1.1 Studies using Unstabilised Total Flux Formulation in Equation (2.14)

For the total flux formulation using $\mathcal{RT}_0 \times \mathcal{P}_0$ pairs of elements, convergence of the order of $1/N^2$ ($\mathcal{O}(h^2)$) is observed for $\|\boldsymbol{v} - \boldsymbol{v}_h\|_{0,\Omega}$ and the order of $1/N$ for $\|p - p_h\|_{0,\Omega}$ with values of diffusion greater than 10^{-2}. However, in the cases where diffusion became smaller than 10^{-2} , convergence occurs only as the mesh becomes fine enough. As can be seen in Figure 2.6, when $\varepsilon = 10^{-5}$, h must be less than $\frac{1}{128}$ for the convergence to begin. However, when $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ pairs of elements are used (Figure 2.7), although good convergence of the order of $1/N^2$ is found for both $\|p - p_h\|_{0,\Omega}$ and $\|\boldsymbol{v} - \boldsymbol{v}_h\|_{\mathrm{H(div},\Omega)}$ with $\varepsilon \geq 10^{-2}$, serious instability of the solution is observed when $\varepsilon < 10^{-2}$. Similarly, convergence for $\|\boldsymbol{v} - \boldsymbol{v}_h\|_{0,\Omega}$ was of order $1/N^3$ but with the same instability occurring when $\varepsilon < 10^{-2}$.

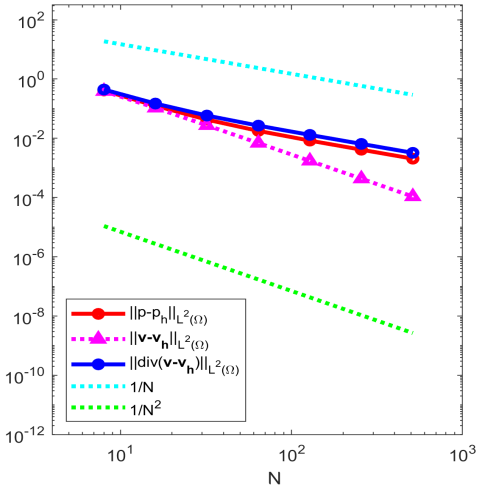
The advection skew to the mesh test in Figure 2.8, shows that for $\mathcal{RT}_0 \times \mathcal{P}_0$ and using the total flux formulation, the boundary layer at the outflow disappears from the solution when $\varepsilon < 10^{-1}$ and when $\varepsilon = 10^{-4}$ the solution is unstable. The situation improves with $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, as shown in Figure 2.9, with the outlet boundary layer visible until $\varepsilon < 10^{-2}$ but then a stability loss in the solution is seen when $\varepsilon < 10^{-3}$.



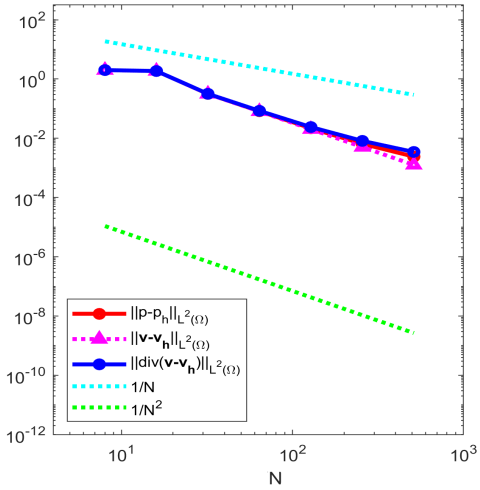
(a) DR total flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 1$



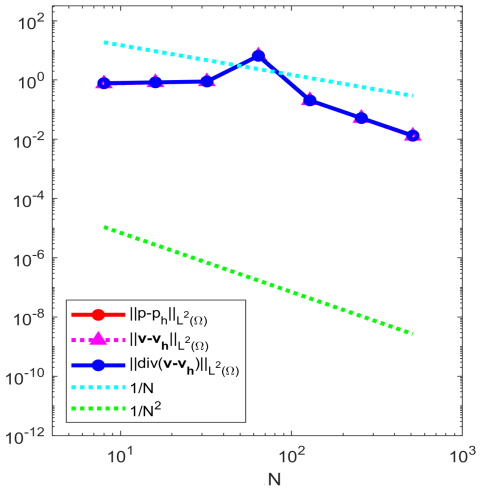
(b) DR total flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 10^{-1}$



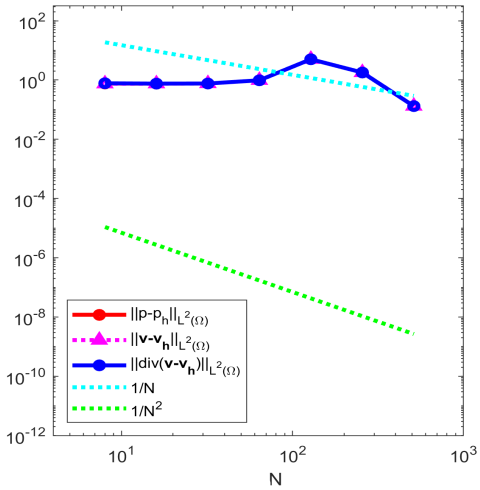
(c) DR total flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 10^{-2}$



(d) DR total flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 10^{-3}$

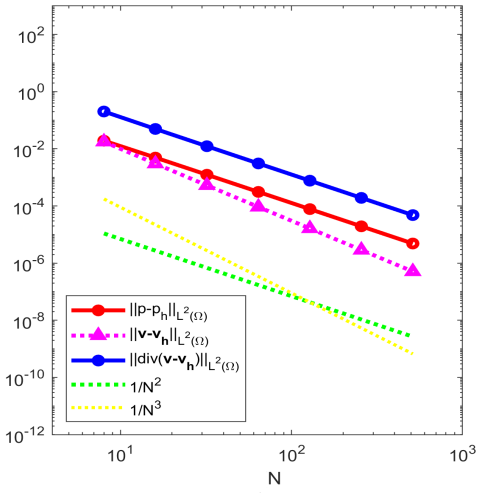


(e) DR total flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 10^{-4}$

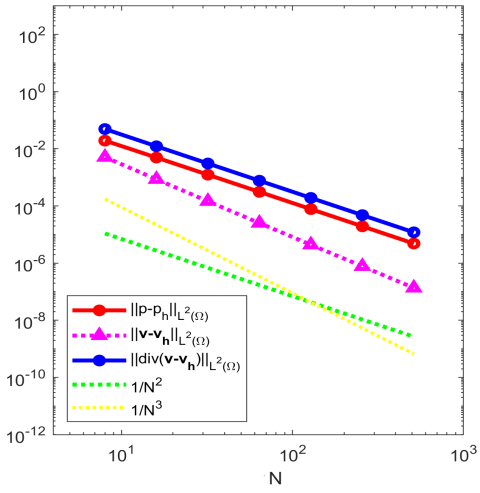


(f) DR total flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 10^{-5}$

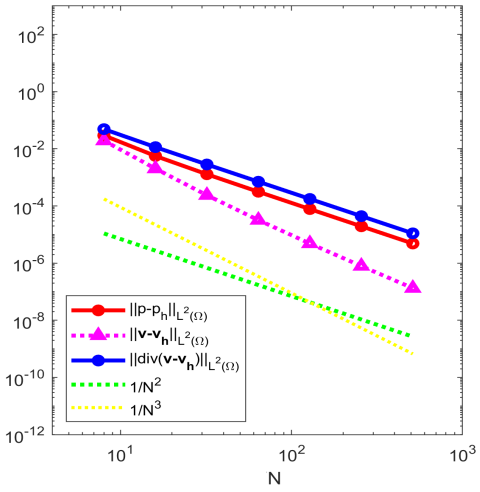
Fig: 2.6 DR $\mathcal{RT}_0 \times \mathcal{P}_0$ Total flux method: convergence graphs for Test A



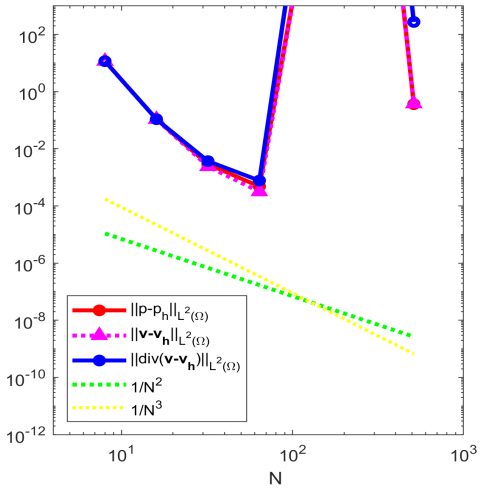
(a) DR total flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 1$



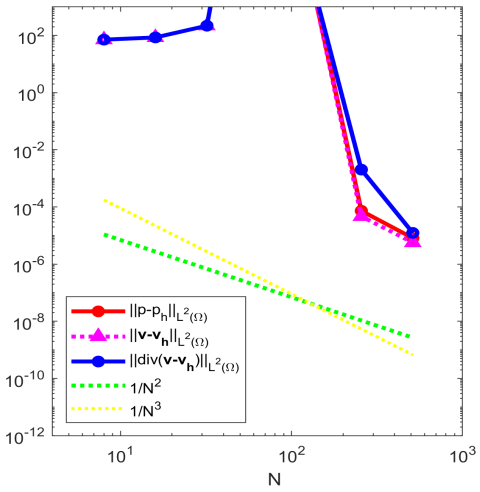
(b) DR total flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-1}$



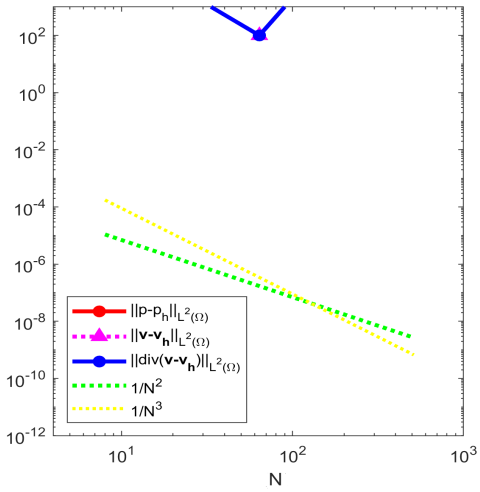
(c) DR total flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-2}$



(d) DR total flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-3}$

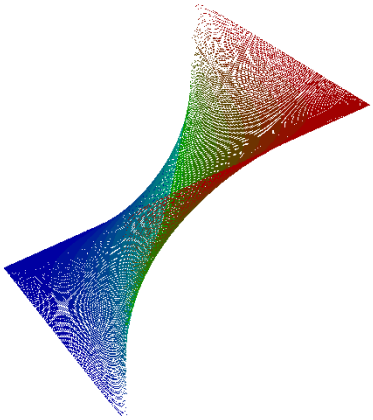


(e) DR total flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-4}$

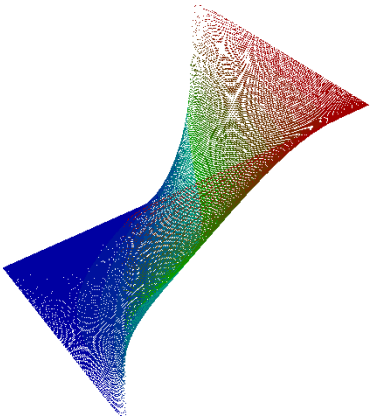


(f) DR total flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-5}$

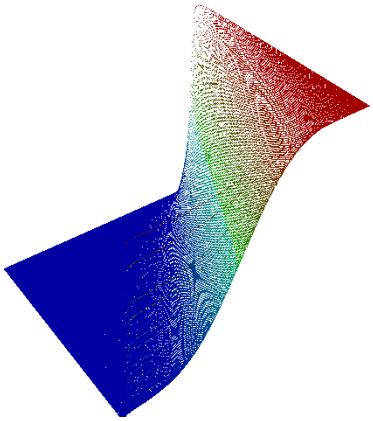
Fig: 2.7 DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ Total flux method: convergence graphs for Test A



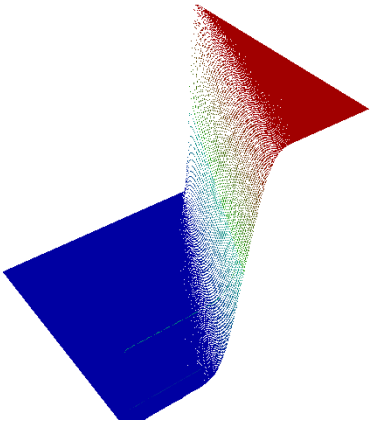
(a) DR $\mathcal{RT}_0 \times \mathcal{P}_0$ total flux $\varepsilon = 10^0$



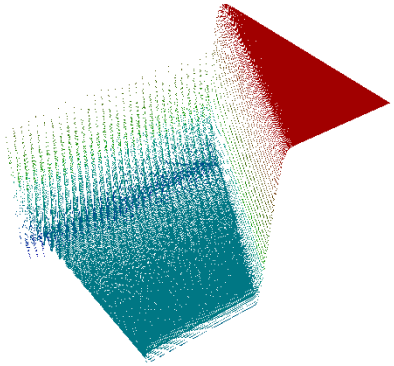
(b) DR $\mathcal{RT}_0 \times \mathcal{P}_0$ total flux $\varepsilon = 10^{-1}$



(c) DR $\mathcal{RT}_0 \times \mathcal{P}_0$ total flux $\varepsilon = 10^{-2}$

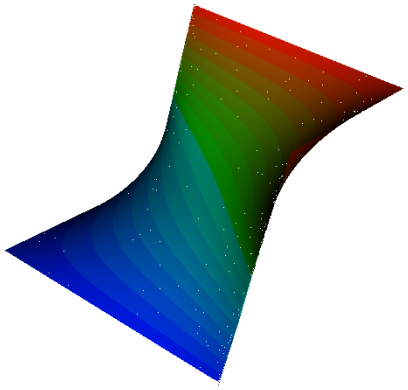


(d) DR $\mathcal{RT}_0 \times \mathcal{P}_0$ total flux $\varepsilon = 10^{-3}$

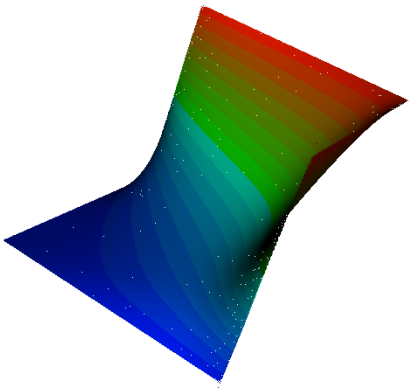


(e) DR $\mathcal{RT}_0 \times \mathcal{P}_0$ total flux $\varepsilon = 10^{-4}$

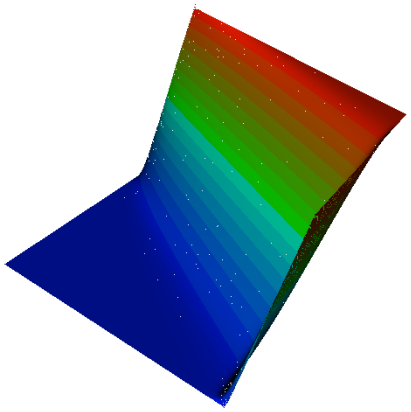
Fig: 2.8 DR $\mathcal{RT}_0 \times \mathcal{P}_0$ Test B: advection skew to the mesh test for total flux, $N = 2^8$



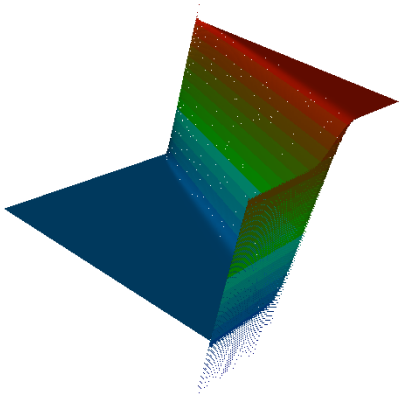
(a) DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\text{dc}}$ total flux $\varepsilon = 10^0$



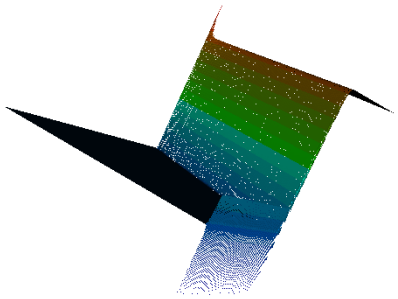
(b) DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\text{dc}}$ total flux $\varepsilon = 10^{-1}$



(c) DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\text{dc}}$ total flux $\varepsilon = 10^{-2}$



(d) DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\text{dc}}$ total flux $\varepsilon = 10^{-3}$



(e) DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\text{dc}}$ total flux $\varepsilon = 10^{-4}$

Fig: 2.9 DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\text{dc}}$ Test B: advection skew to the mesh test for total flux, $N = 2^8$

2.5.1.2 Studies using Unstabilised Diffusive Flux Formulation in Equation (2.16)

The same convergence studies are repeated with diffusive flux for both $\mathcal{RT}_0 \times \mathcal{P}_0$ and $\mathcal{RT}_1 \times \mathcal{P}_1^{\text{dc}}$ pairs of elements. Less instability in the formulation is observed in Figure 2.10, with a clear turning point for each value of ε when the mesh becomes small enough for convergence to begin. However, Figure 2.11 shows that the same dramatic instabilities are found when using $\mathcal{RT}_1 \times \mathcal{P}_1^{\text{dc}}$ elements. For $\varepsilon = 10^{-5}$, it is not clear if a small enough mesh can actually be found that will lead to convergence.

The interior layer tests for diffusive flux and $\mathcal{RT}_0 \times \mathcal{P}_0$ pairs of elements in Figure 2.10 are very similar to the total flux case, with the boundary layer disappearing when $\varepsilon < 10^{-1}$ and instability of the solution when $\varepsilon = 10^{-4}$. Again using $\mathcal{RT}_1 \times \mathcal{P}_1^{\text{dc}}$ pairs of elements, the solution is stable until $\varepsilon = 10^{-3}$ with a visible, partial boundary layer on the outlet but some signs of instability visible. However, by $\varepsilon = 10^{-4}$ the solution is unstable again (see Figure 2.11).

2.5.2 Stabilised MFEM with Raviart-Thomas pairs of elements

After the publication of Douglas and Roberts papers, in 1987, Thomas published a separate paper [Tho87] titled 'Mixed Finite Elements for the Convection-Diffusion' problem. His formulation was based on a variation of the diffusive flux Equation (2.16) using Raviart-Thomas pairs of elements with $k \geq 1$ and added stabilising terms with means and jumps across the boundary for the discontinuous scalar quantities. This could perhaps be seen as resembling the currently popular Discontinuous Galerkin (DG) methods for solving CDR problems.

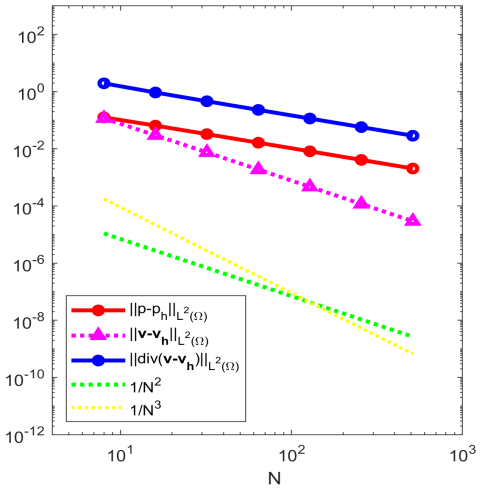
We start with

$$\begin{cases} \boldsymbol{v} + \varepsilon \nabla p = \boldsymbol{0} & \text{in } \Omega, \\ \nabla \cdot \boldsymbol{v} + \nabla p \cdot \boldsymbol{\alpha} + \mu\, p = f & \text{in } \Omega, \\ p = 0 & \text{on } \Gamma. \end{cases} \tag{2.29}$$

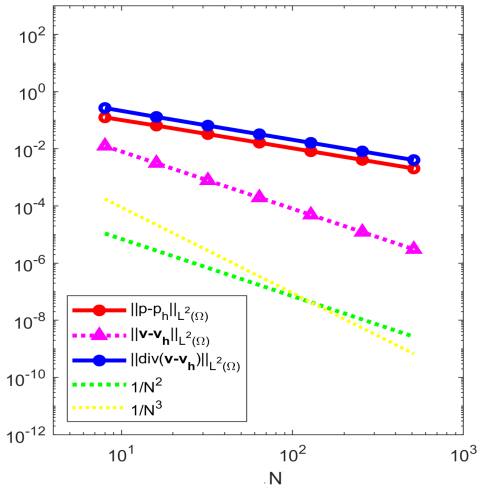
This leads to the variational form for Equation (2.29) becoming: find $(\boldsymbol{v}, p) \in \boldsymbol{V} \times Q$ such that

$$(\boldsymbol{v}, \boldsymbol{w}) - \varepsilon(p, \nabla \cdot \boldsymbol{w}) + (\nabla \cdot \boldsymbol{v}, q) - \underbrace{(\boldsymbol{\alpha} \cdot \nabla p, q)}_{c_0(p,q)} + \mu\,(p, q) \quad = \quad (f, q)\,, \tag{2.30}$$

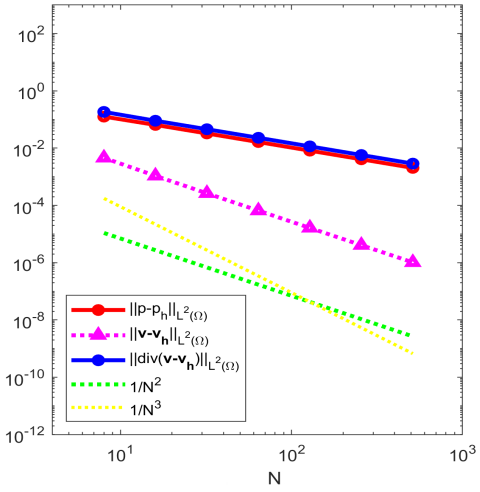
for all $(\boldsymbol{w}, q) \in \boldsymbol{V} \times Q$.



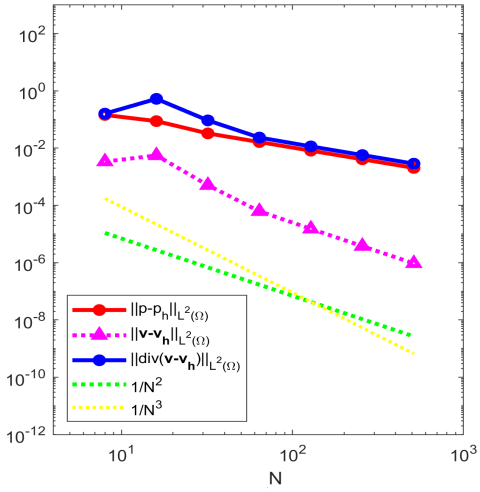
(a) DR diffusive flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 1$



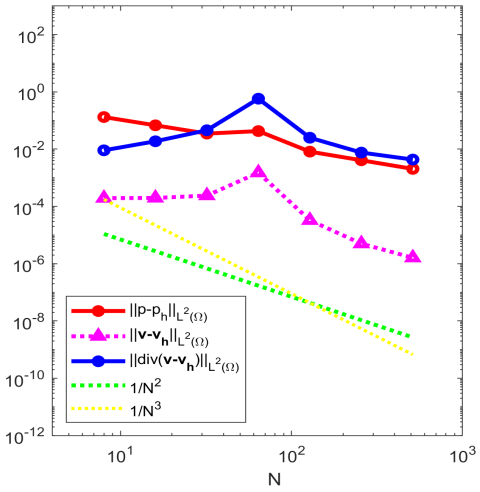
(b) DR diffusive flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 10^{-1}$



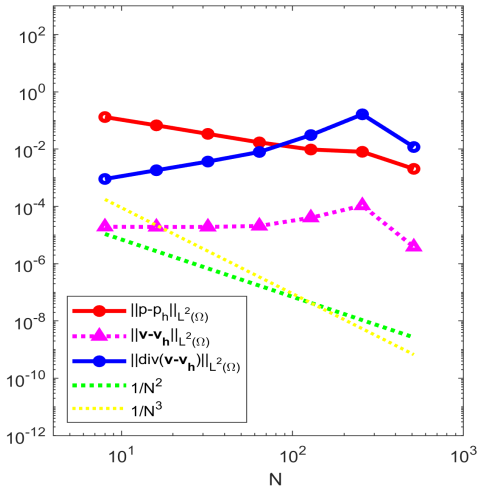
(c) DR diffusive flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 10^{-2}$



(d) DR diffusive flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 10^{-3}$

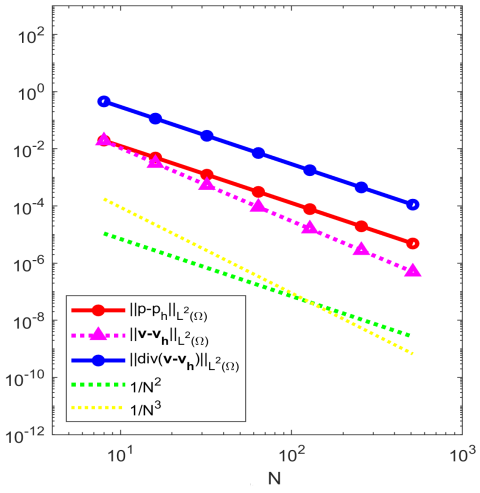


(e) DR diffusive flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 10^{-4}$

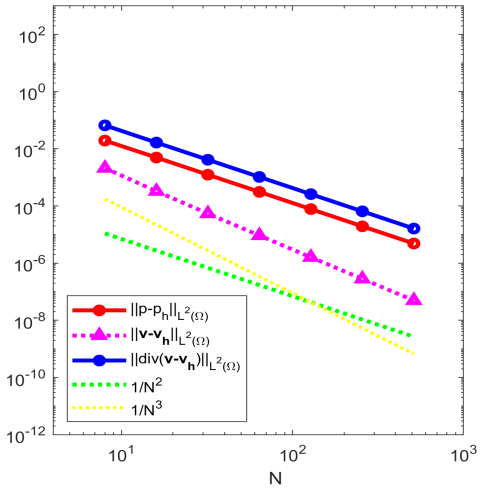


(f) DR diffusive flux $\mathcal{RT}_0 \times \mathcal{P}_0\,, \varepsilon = 10^{-5}$

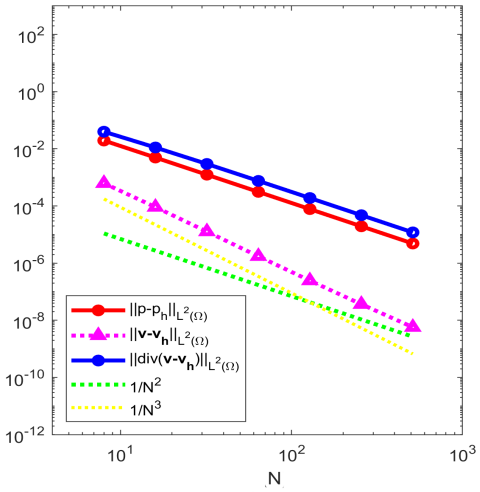
Fig: 2.10 DR $\mathcal{RT}_0 \times \mathcal{P}_0$ Diffusive flux method: convergence graphs for Test A



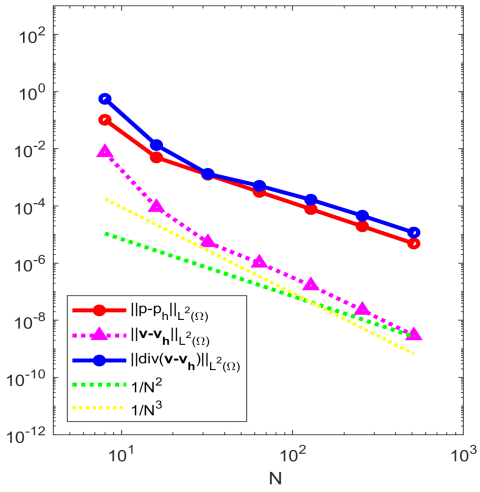
(a) DR diffusive flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}, \varepsilon = 1$



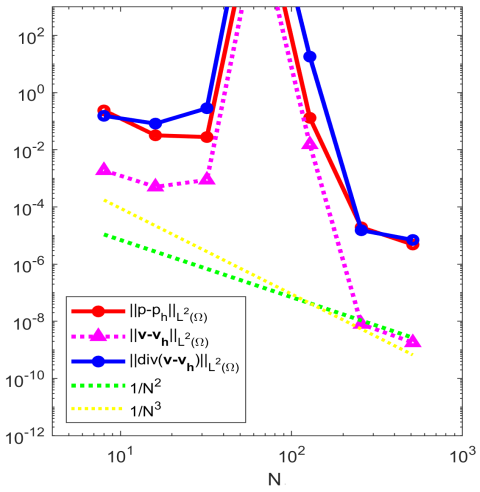
(b) DR diffusive flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}, \varepsilon = 10^{-1}$



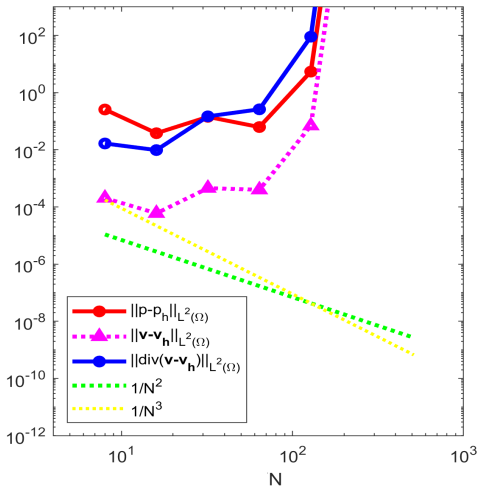
(c) DR diffusive flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}, \varepsilon = 10^{-2}$



(d) DR diffusive flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}, \varepsilon = 10^{-3}$

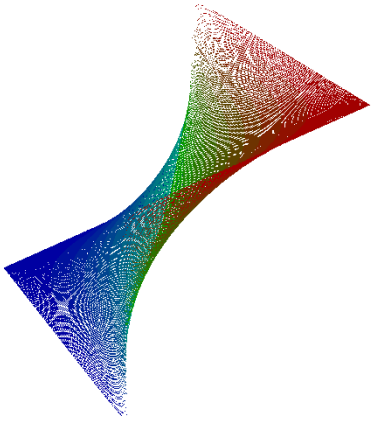


(e) DR diffusive flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}, \varepsilon = 10^{-4}$

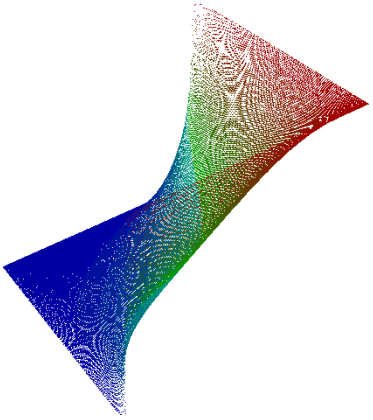


(f) DR diffusive flux $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}, \varepsilon = 10^{-5}$

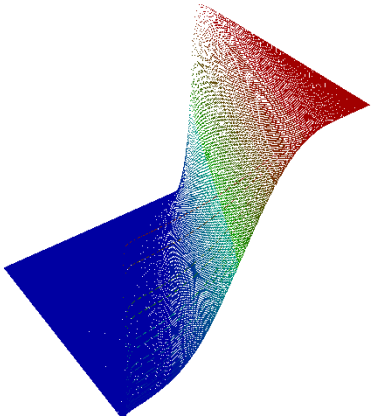
Fig: 2.11 DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ Diffusive flux method: convergence graphs for Test A



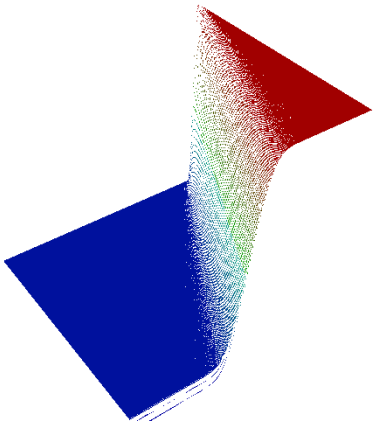
(a) DR $\mathcal{RT}_0 \times \mathcal{P}_0$ diffusive flux $\varepsilon = 10^0$



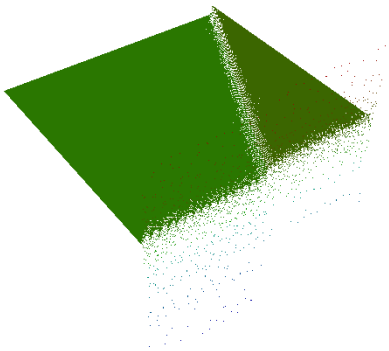
(b) DR $\mathcal{RT}_0 \times \mathcal{P}_0$ diffusive flux $\varepsilon = 10^{-1}$



(c) DR $\mathcal{RT}_0 \times \mathcal{P}_0$ diffusive flux $\varepsilon = 10^{-2}$

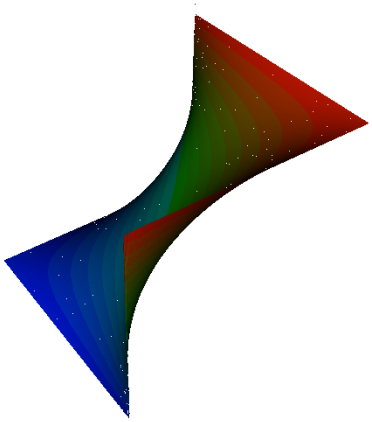


(d) DR $\mathcal{RT}_0 \times \mathcal{P}_0$ diffusive flux $\varepsilon = 10^{-3}$

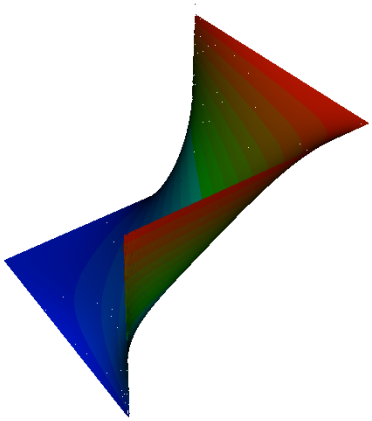


(e) DR $\mathcal{RT}_0 \times \mathcal{P}_0$ diffusive flux $\varepsilon = 10^{-4}$

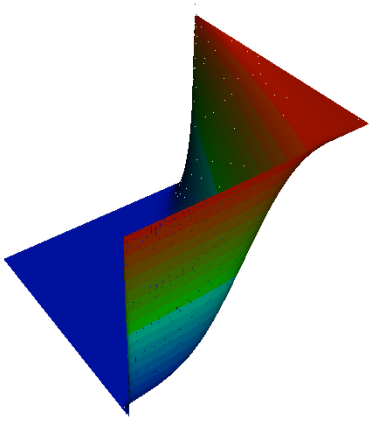
Fig: 2.12 DR $\mathcal{RT}_0 \times \mathcal{P}_0$ Test B: advection skew to the mesh test for diffusive flux, $N = 2^8$



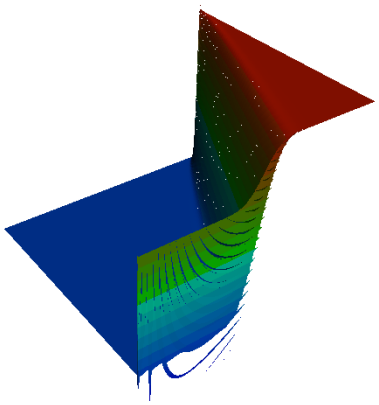
(a) DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ diffusive flux $\varepsilon = 10^0$



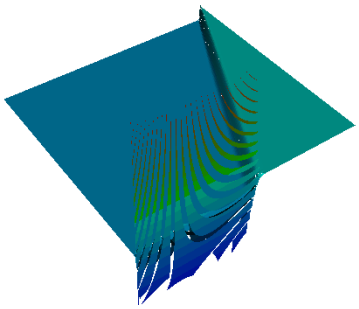
(b) DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ diffusive flux $\varepsilon = 10^{-1}$



(c) DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ diffusive flux $\varepsilon = 10^{-2}$



(d) DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ diffusive flux $\varepsilon = 10^{-3}$



(e) DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ diffusive flux $\varepsilon = 10^{-4}$

Fig: 2.13 DR $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ Test B: advection skew to the mesh test for diffusive flux, $N = 2^8$

Now the term $c_0(p,q)$ is reformulated in terms of the change of gradient and jump across the boundaries of each element,

$$c_0(p,q) = \int_\Omega \boldsymbol{\alpha}\cdot\ \boldsymbol{\nabla} pq = \frac{1}{2}\sum_{T\in\mathcal{T}_h}\int_T \boldsymbol{\alpha}\cdot(q\boldsymbol{\nabla}p - p\boldsymbol{\nabla}q)dx + \sum_{T\in\mathcal{T}_h}\int_{\partial T}\boldsymbol{\alpha}\cdot\boldsymbol{n_T}\left(\langle p\rangle - \frac{1}{2}p\right) q\ ds\ , \tag{2.31}$$

where $\boldsymbol{n_T}$ is an outwards facing normal to the chosen edge or face and $\langle p\rangle$ is the average value of p between the two triangles either side of the edge (or face).

As this reformulation is still not stable when diffusion is small compared to convection, stabilisation is to be achieved by adding an extra term, $d(p,q)$, which combined together with $c_o(p,q)$ as $-c(p,q)$, becomes the C term in the matrix equation Equation (2.23). Thus $c(p,q) = c_o(p,q) + d(p,q)$. The main stabilisation term suggested by Thomas is

$$d(p,q) = -\frac{1}{2}\sum_{T\in\mathcal{T}_h}\int_{\partial T}\boldsymbol{\alpha}\cdot\boldsymbol{n_T}[\![p]\!]q\ ds\ , \tag{2.32}$$

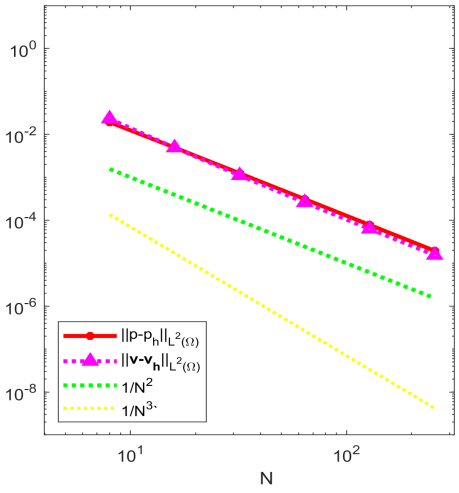
where $[\![p]\!]$ is the jump of the discontinuous scalar quantity p across the edge or face between the two elements either side. This term was proposed earlier by Jaffre [Jaf84] and leads to an upwind discretisation of the convective term corresponding to the Lesaint-Raviart upwinding method [LR74].

Thus the full discrete form for Thomas' method becomes: find $(\boldsymbol{v},p)\in\mathcal{RT}_1\times\mathcal{P}_1^{dc}$ such that

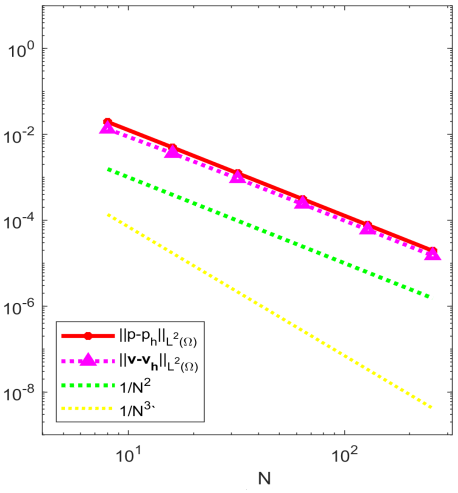
$$\begin{aligned}&(\boldsymbol{v},\boldsymbol{w}) - \varepsilon(p,\nabla\cdot\boldsymbol{w}) + (\nabla\cdot\boldsymbol{v},q) + \mu\,(p,q) - \\ &\quad\frac{1}{2}\,(\boldsymbol{\alpha},(q\boldsymbol{\nabla}p - p\boldsymbol{\nabla}q)) - (\boldsymbol{\alpha}\cdot\boldsymbol{n_T}(\langle p\rangle - \frac{1}{2}p),q)_{\partial T} + \frac{1}{2}(\boldsymbol{\alpha}\cdot\boldsymbol{n_T}[\![p]\!],q)_{\partial T} = (f,q)\,,\end{aligned} \tag{2.33}$$

for all $(\boldsymbol{w},q)\in\mathcal{RT}_1\times\mathcal{P}_1^{dc}$.

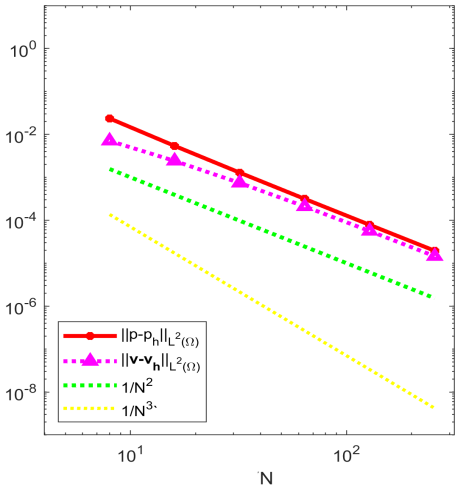
Thomas' method performs well in convergence studies with Test A for diffusion values of $\varepsilon \geq 10^{-2}$, shown in Figure 2.14, but does not converge so well if diffusion is smaller. For the advection skew to the mesh study (Test B) depicted in Figure 2.15, Thomas' method fails to capture the interior or boundary layer as diffusion gets smaller.



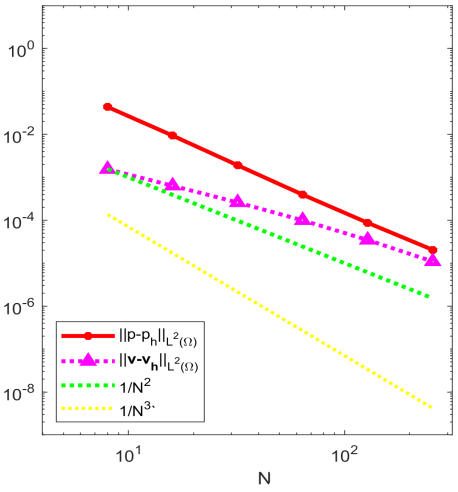
(a) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 1$



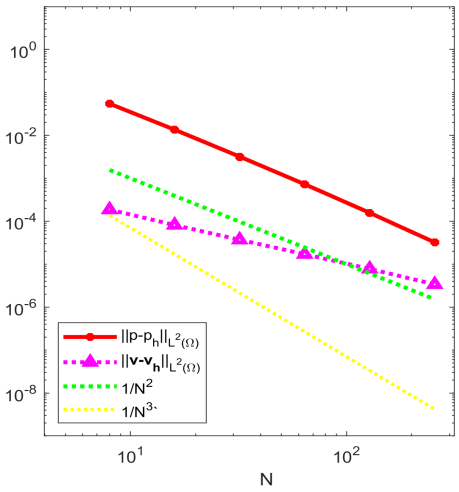
(b) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-1}$



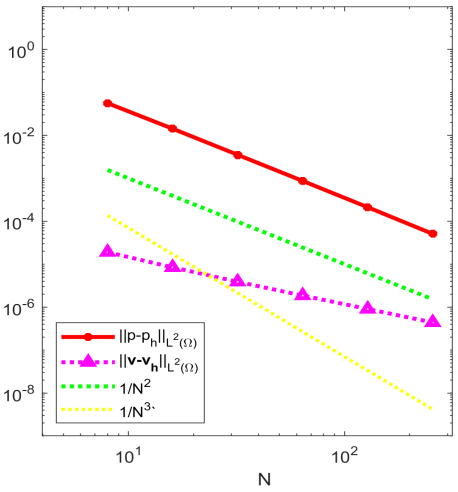
(c) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-2}$



(d) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-3}$

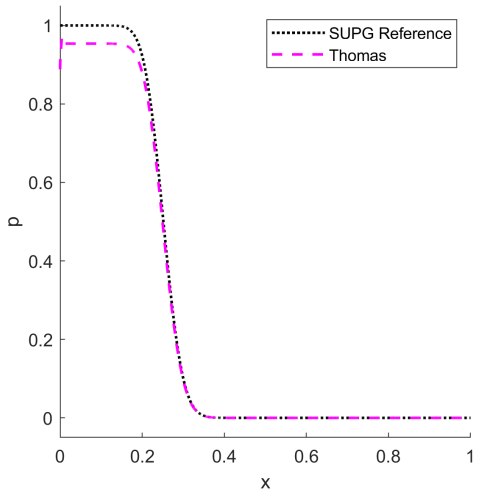


(e) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-4}$

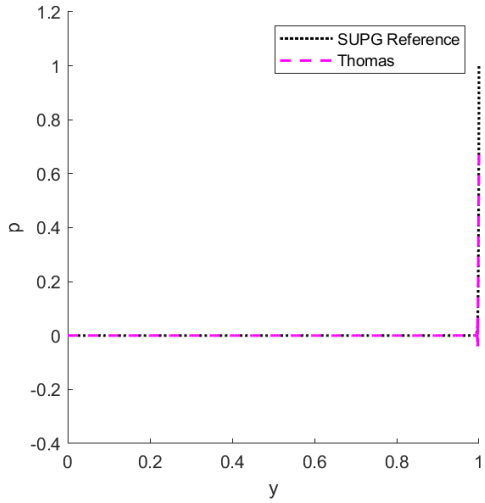


(f) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-5}$

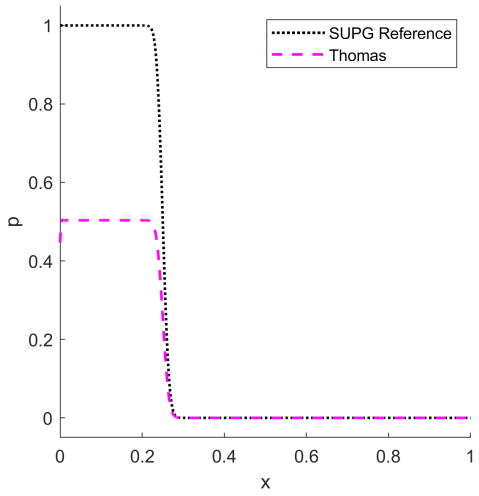
Fig: 2.14 Thomas method: Convergence graphs for Test A



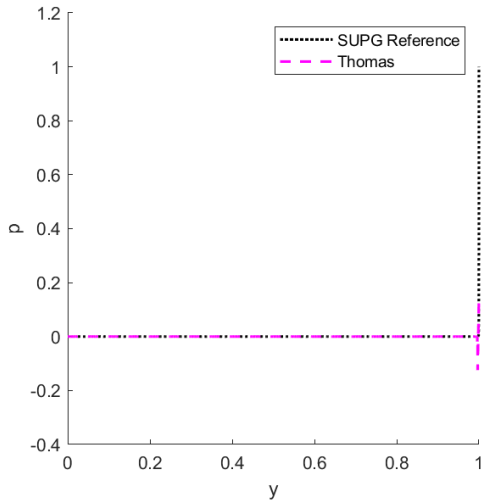
(a) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-3}$ $x = 0.7$



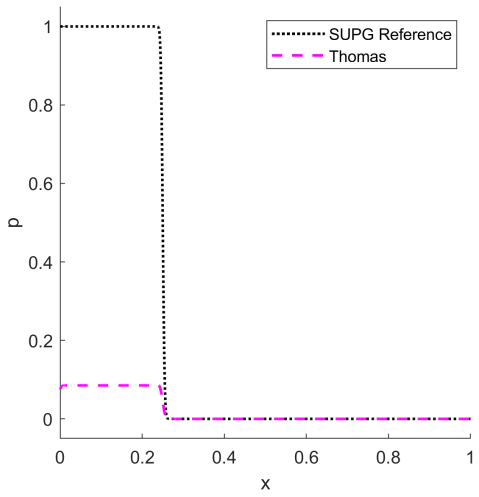
(b) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-3}$ $y = 0.5$



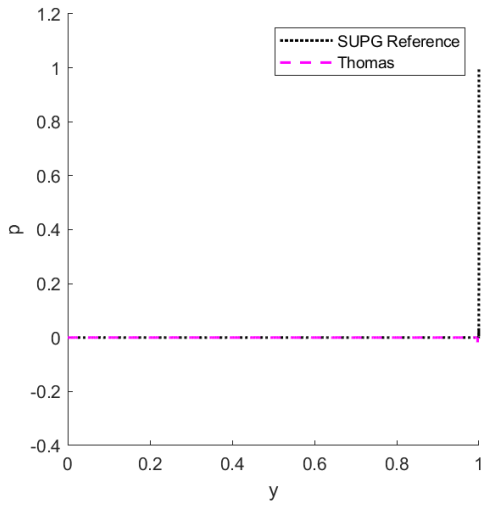
(c) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-4}$ $x = 0.7$



(d) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-4}$ $y = 0.5$



(e) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-5}$ $x = 0.7$



(f) Thomas $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\varepsilon = 10^{-5}$ $y = 0.5$

Fig: 2.15 Thomas method for Test B: advection skew to the mesh test

2.6 Stabilised Mixed Methods with Lagrangian Elements

2.6.1 Masud and Kwack method

Masud and Kwack [MK08] used stabilised $\mathcal{P}_k$ Lagrangian elements (with $k = 1$ or $k = 2$) for both variables p_h and $\boldsymbol{v_h}$ based on the total flux formuation Section 2.3. A stabilising, residual-based term, $(-\tau(\boldsymbol{v}_h - \boldsymbol{\alpha} p_h + \varepsilon \boldsymbol{\nabla} p_h), \frac{1}{\varepsilon}\boldsymbol{w}_h - \boldsymbol{\nabla} q_h)_\Omega$, is added to the left-hand side with boundary conditions applied strongly, giving the following discrete, weak formulation: find $(\boldsymbol{v}_h, p_h) \in \boldsymbol{H}_h \times Q_h^0$, such that

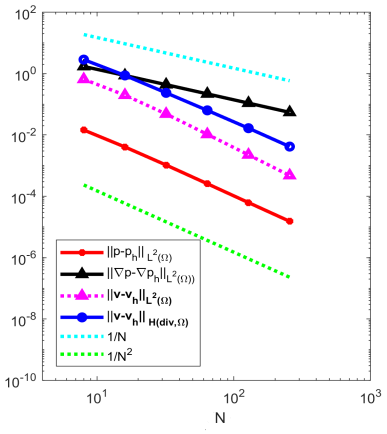
$$\begin{aligned} &\frac{1}{\varepsilon}(\boldsymbol{v_h}, \boldsymbol{w_h}) - (p_h, \nabla \cdot \boldsymbol{w_h}) - \frac{1}{\varepsilon}(\boldsymbol{\alpha} p_h, \boldsymbol{w_h}) \\ &\quad + (\nabla \cdot \boldsymbol{v_h}, q_h) + \mu(p_h, q_h) - (\tau(\boldsymbol{v_h} - \boldsymbol{\alpha} p_h + \varepsilon \boldsymbol{\nabla} p_h), \frac{1}{\varepsilon}\boldsymbol{w_h} - \boldsymbol{\nabla} q_h) \ = (f, q_h) \ , \end{aligned} \tag{2.34}$$

for all $(\boldsymbol{w}_h, q_h) \in \boldsymbol{H}_h \times Q_h^0$. This method is referred to as MK in our numerical results. The value of τ was estimated in [MK08] from calculations using bubble functions and the formula $\tau(Pe) = -\dfrac{a}{Pe + 2a} + 1$ where $a = 4.5$ was used in the study. For small values of diffusion, when the Péclet number, defined as $Pe = \dfrac{|\boldsymbol{\alpha}|h}{2\varepsilon}$ is large, then $\tau \approx 1.0$. There was no analysis published in their study and the numerical tests were carried out with flux parallel to the mesh at 45°.

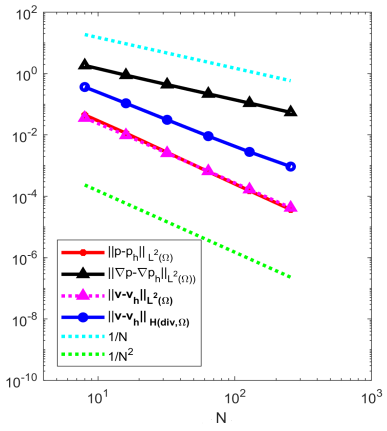
2.6.1.1 Convergence studies

We carry out Test A (see Section 2.4.1) to test the convergence of this stabilised formulation in Equation (2.34) and the results can be found in Figures 2.16 and 2.17. When using $\mathcal{P}_1\mathcal{P}_1$ Lagrangian elements, Figure 2.16 shows convergence of the order of $\dfrac{1}{N}$ for $\|\boldsymbol{\nabla} p - \boldsymbol{\nabla} p_h\|_{0,\Omega}$ and $\dfrac{1}{N^2}$ for both $\|\boldsymbol{v} - \boldsymbol{v}_h\|_{0,\Omega}$ and $\|p - p_h\|_{0,\Omega}$. In the case of $\mathcal{P}_2\mathcal{P}_2$ Lagrangian elements, Figure 2.17 shows convergence of the order $\dfrac{1}{N^2}$ for both $\|\boldsymbol{\nabla} p - \boldsymbol{\nabla} p_h\|_{0,\Omega}$ and $\|\boldsymbol{v} - \boldsymbol{v}_h\|_{0,\Omega}$ and $\dfrac{1}{N^3}$ for $\|p - p_h\|_{0,\Omega}$. It is notable that when diffusion is less than 10^{-3} initially before the mesh is refined, then the errors are large. However, as the mesh is refined, super-convergence occurs until the error is greatly reduced.

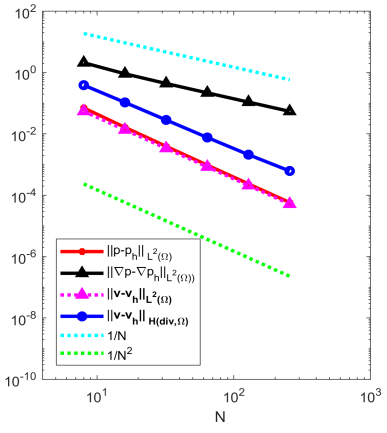
Further tests will be carried out on the MK formulation in the comparative study in Chapter 5.



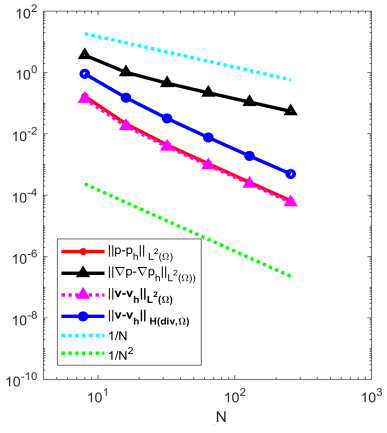
(a) MK $\mathcal{P}_1\mathcal{P}_1$, $\varepsilon = 1$



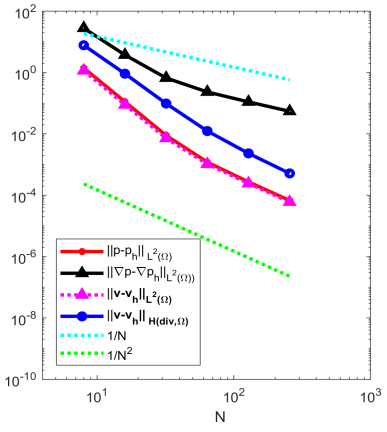
(b) MK $\mathcal{P}_1\mathcal{P}_1$, $\varepsilon = 10^{-1}$



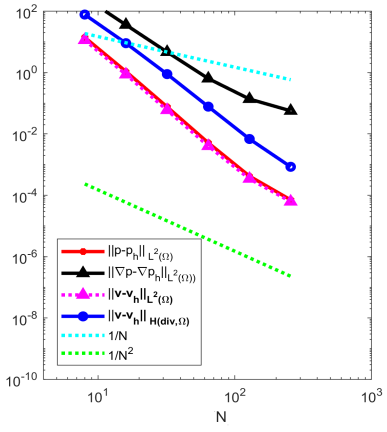
(c) MK $\mathcal{P}_1\mathcal{P}_1$, $\varepsilon = 10^{-2}$



(d) MK $\mathcal{P}_1\mathcal{P}_1$, $\varepsilon = 10^{-3}$

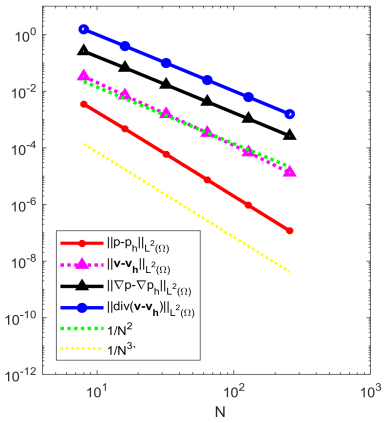


(e) MK $\mathcal{P}_1\mathcal{P}_1$, $\varepsilon = 10^{-4}$

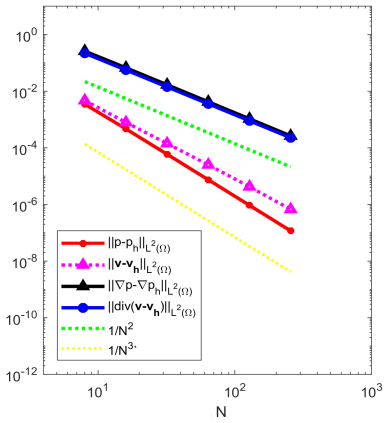


(f) MK $\mathcal{P}_1\mathcal{P}_1$, $\varepsilon = 10^{-5}$

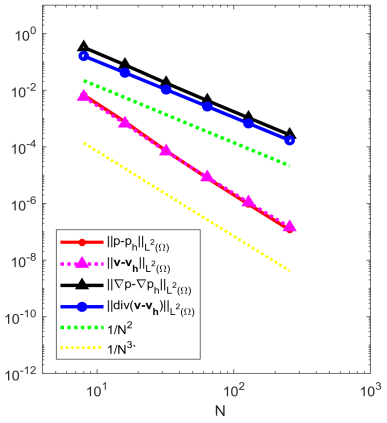
Fig: 2.16 Masud and Kwack $\mathcal{P}_1\mathcal{P}_1$ convergence graphs for Test A



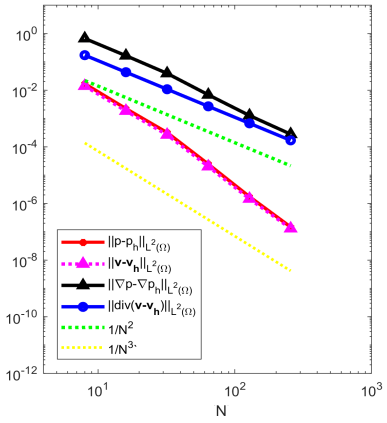
(a) MK $\mathcal{P}_2\mathcal{P}_2$, $\varepsilon = 1$



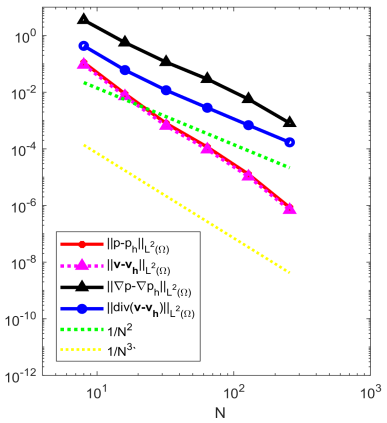
(b) MK $\mathcal{P}_2\mathcal{P}_2$, $\varepsilon = 10^{-1}$



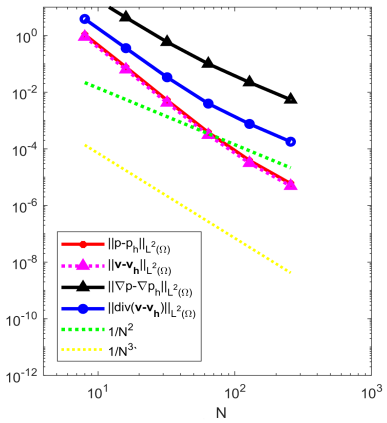
(c) MK $\mathcal{P}_2\mathcal{P}_2$, $\varepsilon = 10^{-2}$



(d) MK $\mathcal{P}_2\mathcal{P}_2$, $\varepsilon = 10^{-3}$



(e) MK $\mathcal{P}_2\mathcal{P}_2$, $\varepsilon = 10^{-4}$



(f) MK $\mathcal{P}_2\mathcal{P}_2$, $\varepsilon = 10^{-5}$

Fig: 2.17 Masud and Kwack $\mathcal{P}_2\mathcal{P}_2$ convergence graphs for Test A

2.7 First Order System of Least Squares (FOSLS): Lagrangian and Raviart-Thomas Elements

When the First-Order Systems Least Squares method (FOSLS) or Least-Square Finite Element Method (LSFEM) is used, the equations are in a form similar to those used in MFEM but the test function is different. It is often termed the 'div-grad' system, and the equations can also be formulated with the methods of total and diffusive flux. An extended version, which may give better results and dampen instabilities, is also often used for both fluxes, called the 'div-grad-curl' system, and includes an extra curl equation. The CDR problem is now a symmetric, elliptic problem and no stabilisation or consideration of the inf-sup condition is necessary, making this an attractive method. However, as mentioned in the Introduction, FOSLS methods lead to diffusive layers. The FOSLS formulation can be used with many finite elements choices including straightforward Lagrangian element combinations such as $\mathcal{P}_1\mathcal{P}_1$ and $\mathcal{P}_2\mathcal{P}_2$. However, recent methods have used Raviart-Thomas $\mathcal{RT}_k$ elements, which have weaker continuity requirements, for the vector variable and Lagrangian $\mathcal{P}_k$ elements for the scalar variable, with $k \geq 0$.

Firstly, here is a short survey of some of the main FOSLS methods that exist. These give the background for our FOSLS method we selected for our comparative study. (Note the symmetry of the left-hand side in the formulations as compared to Equations (2.15) or (2.17)). Using the set of equations Equation (2.14) for total flux, with the method of FOSLS gives: find $(\boldsymbol{v}, p) \in H(\mathrm{div}, \Omega) \times H_0^1(\Omega)$ such that,

$$\begin{cases} (\boldsymbol{v} + \varepsilon\boldsymbol{\nabla} p - \boldsymbol{\alpha} p, \boldsymbol{w} + \varepsilon\boldsymbol{\nabla} q - \boldsymbol{\alpha} q) = 0 \qquad \forall (\boldsymbol{w}, q) \in H(\mathrm{div}, \Omega) \times H_0^1(\Omega) \\ (\nabla \cdot \boldsymbol{v} + \mu p, \nabla \cdot \boldsymbol{w} + \mu q) = (f, \nabla \cdot \boldsymbol{w} + \mu q) \\ p = 0 \qquad \text{on } \Gamma\,. \end{cases} \tag{2.35}$$

In a similar fashion, using the set of equations Equation (2.16) for diffusive flux, with the method of FOSLS gives: find $(\boldsymbol{v}, p) \in H(\mathrm{div}, \Omega) \times H_0^1(\Omega)$, such that,

$$\begin{cases} (\boldsymbol{v} + \varepsilon\boldsymbol{\nabla} p, \boldsymbol{w} + \varepsilon\boldsymbol{\nabla} q) = 0 \qquad \forall (\boldsymbol{w}, q) \in H(\mathrm{div}, \Omega) \times H_0^1(\Omega) \\ (\nabla \cdot \boldsymbol{v} + \boldsymbol{\alpha} \cdot \boldsymbol{\nabla} p + \mu p, \nabla \cdot \boldsymbol{w} + \boldsymbol{\alpha} \cdot \boldsymbol{\nabla} q + \mu q) = (f, \nabla \cdot \boldsymbol{w} + \boldsymbol{\alpha} \cdot \boldsymbol{\nabla} q + \mu q) \\ p = 0 \qquad \text{on } \Gamma\,. \end{cases} \tag{2.36}$$

Both these formulations of Equations (2.35) and (2.36) are commonly referred to as 'div-grad'.

A modification to the first-order system of Equation (2.36) was made by Cai et al. [CLMM94], by adding a curl constraint and extra boundary conditions to the original diffusive flux formulation. This extended first-order system can be termed the 'div-grad-curl' formulation, with the extended system for the diffusive flux given by Equation (2.37) is: find $(\boldsymbol{v}, p) \in H(\mathrm{div}, \Omega) \times H_0^1(\Omega)$, such that,

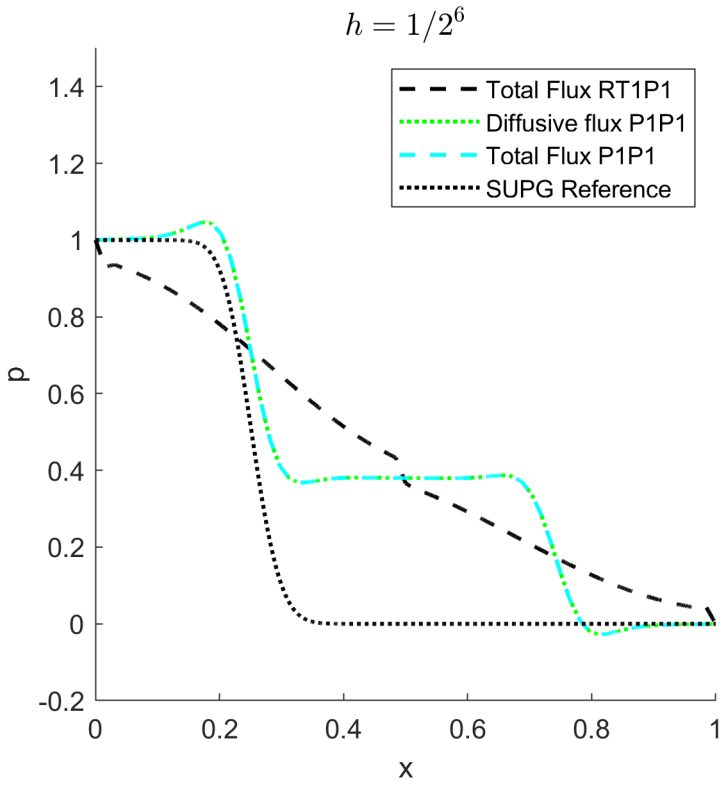
$$\begin{cases} (\boldsymbol{v} + \varepsilon \boldsymbol{\nabla} p, \boldsymbol{w} + \varepsilon \boldsymbol{\nabla} q) = 0 \qquad \forall (\boldsymbol{w}, q) \in H(\mathrm{div}, \Omega) \times H_0^1(\Omega) \\ (\nabla \cdot \boldsymbol{v} + \boldsymbol{\alpha} \cdot \boldsymbol{\nabla} p + \mu p, \nabla \cdot \boldsymbol{w} + \boldsymbol{\alpha} \cdot \boldsymbol{\nabla} q + \mu q) = (f, \nabla \cdot \boldsymbol{w} + \boldsymbol{\alpha} \cdot \boldsymbol{\nabla} q + \mu q) \\ (\boldsymbol{\nabla} \times \boldsymbol{v}, \boldsymbol{\nabla} \times \boldsymbol{w}) = 0 \qquad \text{on } \Gamma \\ p = 0 \qquad \text{on } \Gamma \, . \end{cases} \tag{2.37}$$

It is also possible, although less common, to create an extended system for the total flux of Equation (2.35).

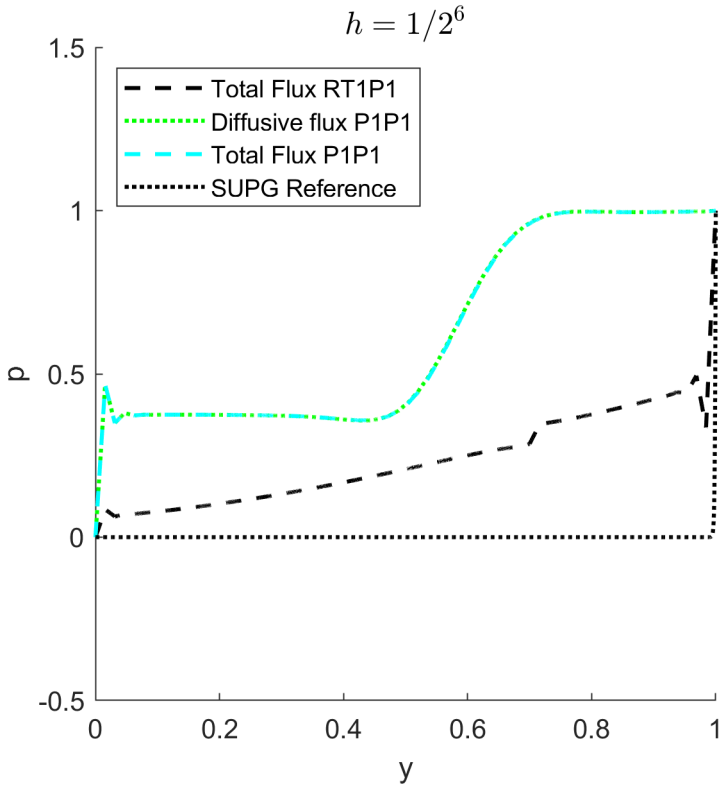
Figure 2.18 depicts the result of applying Test B, the advection skew to the mesh test found in Section 2.4.2, for some of the FOSLS formulations that have been mentioned. It illustrates the difficulties that classical FOSLS methods have in coping with both interior (Figure 2.18a) and boundary layers (Figure 2.18b). Note that this test was conducted at the modestly small value of diffusion $\varepsilon = 10^{-3}$ and the situation deteriorates rapidly as diffusion becomes smaller.

In order to improve the outcomes when layers are present, alternative modifications to the FOSLS system have been considered which involve weighting (or scaling) the diffusive flux equation Equation (2.16) by various factors:$\sqrt{\varepsilon}$ in [CMM97] and exponentially weighted in [FMM98]. Thus it is hoped that the balance of the equations in the system will be improved and lead to better stability and convergence.

In a method similar to SUPG/SDFEM, Lazarov et al. introduced a streamline diffusion [LTV97] method to FOSLS formulations, which they prefer to call LSFEM. This method still has stability problems and in their paper [HY09], Hsieh et al. try modifications of the streamline diffusion method with various stabilisations applied. After applying several layer tests, they admit in their conclusion to the difficulties of finding a FOSLS method that copes with both boundary and interior layers. Their ensuing work introduced interior residual-free bubble functions [HY10] and resulted in significant im-



(a) FOSLS $\varepsilon = 10^{-3}, x = 0.7$ cross-section



(b) FOSLS $\varepsilon = 10^{-3}, y = 0.5$ cross-section

Fig: 2.18 FOSLS methods Test B:advection skew to the mesh test.

provements but at greater computational cost.

2.7.1 Weakly imposed boundary conditions and a weighted FOSLS approach

In a more recent paper, Chen et al. [CFLQ14] combined a $\sqrt{\varepsilon}$ weighted diffusive flux approach with weak imposition of boundary conditions, in order to improve the adaptation of FOSLS to the presence of layers. Their method was tested for both interior and boundary layers in their publication and is now detailed in this section, along with a variation that they proposed. It will be used in the comparative studies in Chapter 5.

$$\boldsymbol{v} + \varepsilon^{1/2}\boldsymbol{\nabla} p = 0 \qquad \text{in } \Omega\,, \tag{2.38a}$$

$$\varepsilon^{1/2}\nabla\cdot\boldsymbol{v} + \boldsymbol{\alpha}\cdot\boldsymbol{\nabla} p + \mu p = f \qquad \text{in } \Omega\,. \tag{2.38b}$$

$$p = 0 \qquad \text{on } \partial\Omega \tag{2.38c}$$

$$. \tag{2.38d}$$

The solution is sought in the finite element space $\mathcal{U}_h = \mathcal{RT}_k(\Omega)\times Q_h$, where Q_h is defined in Equation 4.2 using the degree $k \geq 1$. The method for Equation (2.38) proposed in [CFLQ14] is given by: Find $(\boldsymbol{v}_h, p_h) \in \mathcal{U}_h$ such that

$$\begin{aligned}
&(\boldsymbol{v}_h + \varepsilon^{1/2}\boldsymbol{\nabla} p_h\ ,\ \boldsymbol{w}_h + \varepsilon^{1/2}\boldsymbol{\nabla} q_h)\\
&+ (\varepsilon^{1/2}\nabla\cdot\boldsymbol{v}_h + \boldsymbol{\alpha}\cdot\boldsymbol{\nabla} p_h + \mu p_h\ ,\ \varepsilon^{1/2}\nabla\cdot\boldsymbol{w}_h + \boldsymbol{\alpha}\cdot\boldsymbol{\nabla} q_h + \mu q_h)\\
&\qquad + \sum_{F\in\xi_h^\partial} h_F^{-1}\langle(\varepsilon + \max(-\boldsymbol{\alpha}\cdot\boldsymbol{n}(x), 0))p_h, q_h\rangle_F\\
&\qquad\quad = (f\ ,\ \varepsilon^{1/2}\nabla\cdot\boldsymbol{w}_h + \boldsymbol{\alpha}\cdot\boldsymbol{\nabla} q_h + \mu q_h)
\end{aligned} \tag{2.39}$$

for all $(\boldsymbol{w}_h, q_h) \in \mathcal{U}_h$. Here, ξ_h^∂ is the set of edges or faces of the triangulation (denoted by F) that lie in the boundary Γ, $h_F = h_T$ on the edges on the boundary Γ , $\langle .\,, .\rangle_F$ stands for the inner product in $L^2(F)$, $\boldsymbol{n}$ denotes the unit normal vector outward to Γ. This method will be referred to as FOSLS in our numerical experiments.
As an alternative even weaker imposition of the boundary conditions in [CFLQ14, Re-

mark 2.2], the following method is proposed: find $(\boldsymbol{v}_h, p_h) \in \mathcal{U}_h$ such that

$$\begin{aligned}
&(\boldsymbol{v}_h + \varepsilon^{1/2}\boldsymbol{\nabla} p_h \ , \ \boldsymbol{w}_h + \varepsilon^{1/2}\boldsymbol{\nabla} q_h) \\
&+ (\varepsilon^{1/2}\nabla\cdot\boldsymbol{v}_h + \boldsymbol{\alpha}\cdot\boldsymbol{\nabla} p_h + \mu p_h \ , \ \varepsilon^{1/2}\nabla\cdot\boldsymbol{w}_h + \boldsymbol{\alpha}\cdot\boldsymbol{\nabla} q_h + \mu q_h) \\
&\qquad + \sum_{F\in\xi_h^\partial} \langle (h_F^{-1}\varepsilon + \max(-\boldsymbol{\alpha}\cdot\boldsymbol{n}(x), 0)) p_h, q_h\rangle_F \\
&\qquad\qquad = (f \ , \ \varepsilon^{1/2}\nabla\cdot\boldsymbol{w}_h + \boldsymbol{\alpha}\cdot\boldsymbol{\nabla} q_h + \mu q_h)
\end{aligned} \tag{2.40}$$

for all $(\boldsymbol{w}_h, q_h) \in \mathcal{U}_h$. This alternative will be referred to as FOSLSb in our experiments that follow and imposes the boundary conditions more weakly that the first method.

The central idea of both FOSLS methods is that the weak imposition of the boundary conditions will prevent the error along the boundary layers propagating into the whole domain. This contrasts with the strong imposition of the boundary conditions which pollutes the numerical solution on almost all the domain because of the existence of boundary layers or interior layers.

2.8 Chapter Review

In this chapter we reviewed the existing discretisations for solving the mixed formulation of the CDR equation and conducted numerical studies. The classical, unstabilised method of Douglas and Roberts [DR82, DR85] was found to be unstable when diffusion was small, lacking convergence and not coping with layers. (Note that solutions could only be obtained for the convergence tests or layer tests using the newer 64-bit versions of FreeFem++ and UMFPACK (UMFPACK64) when $N \geq 6$ and $\mathcal{RT}_1 \times \mathcal{P}_1^{\text{dc}}$ elements are used, suggesting that we are working at the limits of machine accuracy.)
The stabilised method introduced by Thomas [Tho87] was also found to suffer from similar problems. The method of Masud and Kwack [MK08] that inspired our new method showed good convergence when diffusion was small and will be included in later comparative studies. The FOSLS methods were reviewed briefly and the selected method and its variation of Chen et al. that will be used in the comparative study was introduced [CFLQ14].

Chapter 3

A numerical investigation of the stability of Douglas and Robert's formulation, especially for small values of diffusion

In this chapter, we further investigate the stability of the classical, unstabilised, standard mixed formulation proposed by Douglas and Roberts using the total flux formulation, by looking at both the LBB inf-sup constant and the overall stability of the formulation. The formulation is outlined in the Background in Section 2.5.1 of Chapter 2.

3.1 Theory

In Chapter 2, the numerical implementation of the formulation of Douglas and Roberts using Raviart-Thomas elements for the CDR equation was shown not to be stable in numerical experiments when diffusion is small compared to convection. Here we investigate the behaviour of the LBB inf-sup constant of Equation (2.21) and the overall stability of the formulation when the meshes are refined and diffusion takes a very small value. This is of particular interest due to the instability observed in the numerical tests in Section 2.5.1.

For convenience of the reader, we restate the general form of Equation (2.18) with homogenous boundary conditions in the form used in this chapter: Find $(\boldsymbol{v}, p) \in \boldsymbol{V} \times Q := H(\mathrm{div}, \Omega) \times L^2(\Omega)$

$$\begin{cases} a(\boldsymbol{v}, \boldsymbol{w}) + b_1(\boldsymbol{w}, p) = 0 \qquad \forall \boldsymbol{w} \in \boldsymbol{V} \\ \quad b_2(\boldsymbol{v}, q) + c(p, q) = (f, q) \qquad \forall q \in Q \ . \end{cases} \tag{3.1}$$

where $a(\boldsymbol{v}, \boldsymbol{w}) = (\boldsymbol{v}, \boldsymbol{w})$, and $c(p, q) = \mu(p, q)$ and we recall that $p = 0$ is enforced weakly on the Dirichlet boundary, $\Gamma := \partial\Omega$. We choose to investigate the case of total flux and therefore

$$b_1(\boldsymbol{w}, p) = -\varepsilon(\nabla \cdot \boldsymbol{w}, p) - (\boldsymbol{w}, \boldsymbol{\alpha}\, p) \quad \text{and} \quad b_2(\boldsymbol{v}, q) = (\nabla \cdot \boldsymbol{v}, q)\,.$$

Both papers [DR82, DR85] were written for the general case with $\mu \geq 0$, but in our case, we focus on $\mu = 0$, (as $\mu > 0$ is believed to aid the stability) so that $c(p, q) = 0$.

By introducing the continuous operators $A : \boldsymbol{V} \to \boldsymbol{V}'$, $B_1 : \boldsymbol{V} \to Q'$ and $B_2 : \boldsymbol{V} \to Q'$ where $B_1' : Q \to \boldsymbol{V}'$ is the adjoint of B_1 ($\boldsymbol{V}'$and Q' are the duals of $\boldsymbol{V}$ and Q) with

$$\begin{aligned} \langle A\boldsymbol{v}\,,\, \boldsymbol{w}\rangle_{\boldsymbol{V}'\times\boldsymbol{V}} &= a(\boldsymbol{v}, \boldsymbol{w}) \\ \langle B_1' p\,,\, \boldsymbol{w}\rangle_{\boldsymbol{V}'\times\boldsymbol{V}} &= b_1(\boldsymbol{w}, p) \\ \langle B_2\boldsymbol{v}\,,\, q\rangle_{Q'\times Q} &= b_2(\boldsymbol{v}, q)\,, \end{aligned} \tag{3.2}$$

then for $f \in Q'$ Equation (3.1) becomes

$$\begin{cases} A\boldsymbol{v} + B_1' p = 0 & \text{in } \boldsymbol{V}' \\ B_2\boldsymbol{v} + 0 = f & \text{in } Q' \end{cases} \tag{3.3}$$

In matrix form using these operators this is

$$\underbrace{\begin{bmatrix} A & B_1' \\ B_2 & 0 \end{bmatrix}}_{\mathbb{A}} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix} \tag{3.4}$$

and we will refer to the matrix on the left-hand side as $\mathbb{A}$.

There are two problems to address for this system: first, the solvability and secondly, the stability of the numerical method used in the discrete implementation.
For solvability, there is a unique solution if matrix $\mathbb{A}$ has an inverse. If this condition is met, then we can prove that the solution is determined by the data so that

$$\|\boldsymbol{v}\|_{\text{div},\Omega} + \|p\|_{0,\Omega} \leq c\,\|f\|_{0,\Omega}\,, \tag{3.5}$$

where $c > 0$ and is independent of the data of the problem.

For our numerical method, we also need stability and this means examining a sequence of discretised problems on increasingly finer meshes, which results in matrices of increas-

ing dimensions. The saddle-point matrix $\mathbb{A}$ must stay uniformly invertible as the mesh parameter h tends to zero.
Therefore, we need to investigate the inf-sup constants as a measure of stability, of both the system as a whole and $a(.,.), b_1(.,.)$ and $b_2(.,.)$ as detailed later in Equations (3.11),(3.12),(3.13),(3.23). We will first set out the background discrete formulation.

The discrete form of Equation (3.1) with $\mu = 0$ is: Find $(\boldsymbol{v}_h, p_h) \in \boldsymbol{V}_h \times Q_h := \mathcal{RT}_k \times \mathcal{P}_k^{\mathrm{dc}}$ such that

$$\begin{cases} a(\boldsymbol{v_h}, \boldsymbol{w_h}) + b_1(\boldsymbol{w_h}, p_h) = 0 \qquad \forall \boldsymbol{w_h} \in \boldsymbol{V}_h \\ \qquad\qquad\qquad b_2(\boldsymbol{v_h}, q_h) = f_h(q_h) \qquad \forall q_h \in Q_h \ , \end{cases} \tag{3.6}$$

with

$$a(\boldsymbol{v}_h, \boldsymbol{w}_h) := \sum_{T \in \mathcal{T}_h} \int_T \boldsymbol{v}_h \cdot \boldsymbol{w}_h \, \mathrm{d}x$$

$$b_1(\boldsymbol{w_h}, p_h) := \sum_{T \in \mathcal{T}_h} \int_T \left(-\varepsilon \nabla \cdot \boldsymbol{w}_h p_h - \boldsymbol{\alpha} \cdot \boldsymbol{w}_h p_h\right) \, \mathrm{d}x$$

$$b_2(\boldsymbol{v_h}, q_h) := \sum_{T \in \mathcal{T}_h} \int_T \nabla \cdot \boldsymbol{v}_h \, q_h \, \mathrm{d}x$$

and

$$f_h(q_h) := \sum_{T \in \mathcal{T}_h} \int_T f \, q_h \, \mathrm{d}x \, .$$

The spaces $\boldsymbol{V}_h$ and Q_h are finite dimensional and, if we let $\dim(\boldsymbol{V}_h) = N_h$ and $\dim(Q_h) = M_h$ (usually $M_h \leq N_h$), then the basis of our finite element space can be represented by $\{\boldsymbol{\phi}\}_{n=1}^{N_h}$ for $\boldsymbol{V}_h$ and a basis $\{\psi_m\}_{m=1}^{M_h}$ for Q_h and we write

$$\boldsymbol{v}_h = \sum_{n=1}^{N_h} \hat{\boldsymbol{v}}_n \boldsymbol{\phi}_n \qquad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h \, , \tag{3.7}$$

and

$$p_h = \sum_{m=1}^{M_h} \hat{p}_m \psi_m \qquad \forall p_h \in Q_h \, , \tag{3.8}$$

with $\hat{\boldsymbol{v}}_n = (\boldsymbol{v}_n)_{n=1}^{N_h} \in \mathbb{R}^{N_h}$ and $\hat{p}_m = (p_m)_{m=1}^{M_h} \in \mathbb{R}^{M_h}$.

With these bases, Equation (3.6) can be represented in matrix form (with some abuse

of notation as A , B_1 and B_2 now represent algebraic vectors), for $(\hat{\boldsymbol{v}}_h, \hat{p}_h) \in \boldsymbol{V}_h \times Q_h$ as

$$\begin{bmatrix} A & B_1^T \\ B_2 & 0 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{v}}_h \\ \hat{p}_h \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{f}_h \end{bmatrix} , \tag{3.9}$$

where the dimension of submatrix A is $N_h \times N_h$, B_1 is $N_h \times M_h$ and B_2 is $M_h \times N_h$ with $N_h, M_h \to \infty$ as $h \to 0$.
Thus, we would like a stability estimate of the form

$$\|\boldsymbol{v}_h\|_{\mathrm{div},\Omega} + \|p_h\|_{0,\Omega} \leq c \, \|f_h\|_{0,\Omega} , \tag{3.10}$$

where the important result would be that c is again independent of h.

For these results to hold, following the theory presented in [BBF13, p. 231], the bilinear forms $a(.,.), b_1(.,.)$ and $b_2(.,.)$ need to satisfy the following four conditions:

$$\inf_{q_h \in Q_h} \sup_{\boldsymbol{v}_h \in V_h} \frac{b_1(\boldsymbol{v}_h, q_h)}{\|\boldsymbol{v}_h\|_{\mathrm{div},\Omega} \|q_h\|_{0,\Omega}} \geq \beta_1 , \tag{3.11}$$

$$\inf_{q_h \in Q_h} \sup_{\boldsymbol{v}_h \in V_h} \frac{b_2(\boldsymbol{v}_h, q_h)}{\|\boldsymbol{v}_h\|_{\mathrm{div},\Omega} \|q_h\|_{0,\Omega}} \geq \beta_2 , \tag{3.12}$$

$$\inf_{\boldsymbol{v}_0 \in Ker B_1} \sup_{\boldsymbol{w}_0 \in Ker B_2} \frac{a(\boldsymbol{v}_0, \boldsymbol{w}_0)}{\|\boldsymbol{v}_0\|_{\mathrm{div},\Omega} \|\boldsymbol{w}_0\|_{\mathrm{div},\Omega}} \geq \alpha_1 , \tag{3.13}$$

$$\inf_{\boldsymbol{w}_0 \in Ker B_1} \sup_{\boldsymbol{v}_0 \in Ker B_2} \frac{a(\boldsymbol{v}_0, \boldsymbol{w}_0)}{\|\boldsymbol{v}_0\|_{\mathrm{div},\Omega} \|\boldsymbol{w}_0\|_{\mathrm{div},\Omega}} \geq \alpha_2 , \tag{3.14}$$

where Ker $B_1 := \{\boldsymbol{v}_h \in \boldsymbol{V}_h$ such that $b_1(\boldsymbol{v}_h, q_h) = 0, \forall q_h \in Q_h\}$, Ker $B_2 := \{\boldsymbol{v}_h \in \boldsymbol{V}_h$ such that $b_2(\boldsymbol{v}_h, q_h) = 0, \forall q_h \in Q_h\}$ and $\alpha_1, \alpha_2, \beta_1$ and β_2 are positive constants independent of h.

Remark 3.1. *For invertibility of $\mathbb{A}$, it is enough that $\alpha_1, \alpha_2, \beta_1, \beta_2 > 0$, but for stability we also need them to be independent of h.*

While this is the theoretical basis for stability, in practice it is difficult to test the validity of Equations (3.11) - (3.14) within the finite element formulation. Thus, it is simpler

to either test the inf-sup stability for $a(\boldsymbol{v}_h, \boldsymbol{w}_h)$ over the full space of $H(\mathrm{div}, \Omega)$, rather trying to isolate the stability on both kernel B_1 and kernel B_2, or to examine the inf-sup stability of $b_1(\boldsymbol{v}_h, q_h)$ alone. As we are using the total flux formulation, then we do not need to look at the validity of Equation (3.12), where $b_2(\boldsymbol{v}_h, q_h) = (q_h\,, \nabla \cdot \boldsymbol{v}_h)$, since this is the stability of the Raviart-Thomas discretisation, which has been proved to be bounded away from zero for $\mathcal{RT}_k \times \mathcal{P}_k^{\mathrm{dc}}$ [RT77]. The difficulty is the validity of the inf-sup constant β_1 in Equation (3.11), which, to the best of our knowledge, is an open problem. One of the purposes of this chapter is to test this numerically.

Another way of proving stability on the complete bilinear formulation of Equations (3.1) and (3.3), is to show that an inf-sup condition holds on a sequence of refined meshes for $h \to 0$ with

$$\mathcal{A}\left((\boldsymbol{v}_h, p_h), (\boldsymbol{w}, q_h)\right) = a(\boldsymbol{v}_h, \boldsymbol{w}_h) + b_1(p_h, \boldsymbol{w}_h) + b_2(\boldsymbol{v}_h, q_h)\ . \tag{3.15}$$

This inf-sup condition reads

$$\sup_{(\boldsymbol{w}_h, q_h)} \frac{\mathcal{A}\left((\boldsymbol{v}_h, p_h), (\boldsymbol{w}_h, q_h)\right)}{\|(\boldsymbol{w}_h, q_h)\|_R} \geq \sigma \|(\boldsymbol{v}_h, p_h)\|_R \qquad \forall (\boldsymbol{v}_h, p_h), (\boldsymbol{w}_h, q_h)\ \in R = \mathcal{RT}_k \times \mathcal{P}_k^{\mathrm{dc}}, \tag{3.16}$$

where

$$\|(\boldsymbol{w}_h, p_h)\|_R = \{\|\boldsymbol{w}_h\|^2_{\mathrm{div},\Omega} + \|p_h\|^2_{0,\Omega}\}^{\frac{1}{2}}\ . \tag{3.17}$$

Thus, another goal of this chapter is to explore if Equation (3.16) holds numerically.

3.2 Methods to compute the two inf-sup conditions

3.2.1 Investigating the inf-sup constant β_1

Following [Wac15], which builds on [HSV12], the Stokes problem in Equations (2.19) and (2.20) can be written as an eigenvalue problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = -\lambda \begin{bmatrix} 0 & 0 \\ 0 & M_p \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix}\ , \tag{3.18}$$

where A, B, M_p are induced by the bilinear forms $\tilde{a}(\boldsymbol{\nabla} \boldsymbol{v}_h, \boldsymbol{\nabla} \boldsymbol{w}_h), \tilde{b}(\boldsymbol{w}_h, p_h)$ and (p_h, q_h) respectively. (Note that M_p is the pressure mass matrix). Expanding out this matrix

and rearranging gives

$$\begin{cases} A\boldsymbol{v} + B^T p = 0 & \Rightarrow \boldsymbol{v} = -A^{-1}B^T p \\ B\boldsymbol{v} + \lambda M_p p = 0 \end{cases} \tag{3.19}$$

We note that M_p is symmetric and positive-definite, so it has a square-root $M_p^{\frac{1}{2}}$, which is also symmetric and positive-definite and thus invertible. Substituting for $\boldsymbol{v}$ into the second equation gives

$$\begin{aligned} -BA^{-1}B^T p &= -\lambda M_p p \\ \Rightarrow BA^{-1}B^T M_p^{-1/2} M_p^{1/2} p &= \lambda M_p^{1/2} M_p^{1/2} p \\ \Rightarrow M_p^{-1/2} BA^{-1}B^T M_p^{-1/2} (M_p^{1/2} p) &= \lambda (M_p^{1/2} p) \\ \Rightarrow D\tilde{p} &= \lambda \tilde{p}\,, \end{aligned}$$

where $\tilde{p} = M_p^{1/2} p$ and $D = M_p^{-1/2} BA^{-1}B^T M_p^{-1/2}$.

Thus, the smallest singular value $\sigma_{\min}(M_p^{-1/2} BA^{-1/2})$ is the square root of the smallest eigenvalue, $\lambda_{\min}(M_p^{-1/2} BA^{-1}B^T M_p^{-1/2})$ and $\sigma_{\min}(M_p^{-1/2} BA^{-1/2}) = |\gamma|$, where γ is the inf-sup constant following the results in [CF03] .

Our difficulty with the CDR equation is that it is non-symmetric and we have two different operators B_1 and B_2. The proof in [HSV12] for the Stokes equation relies on the ellipticity of the linear form

$$\tilde{a}(\boldsymbol{v}, \boldsymbol{w}) = \int_\Omega \boldsymbol{\nabla v} : \boldsymbol{\nabla w} \qquad \text{in } H_0^1(\Omega)^d\,. \tag{3.20}$$

In our case $a(\boldsymbol{v}, \boldsymbol{v}) = (\boldsymbol{v}, \boldsymbol{w})$ is not elliptic in $H(\mathrm{div}, \Omega)$, so we introducte the bilinear form

$$a_v(\boldsymbol{v}, \boldsymbol{w}) = (\boldsymbol{v}, \boldsymbol{w}) + (\nabla \cdot \boldsymbol{v}, \nabla \cdot \boldsymbol{w})\,. \tag{3.21}$$

Inspired by the previous discussion for the Stokes problem, in order to gain an idea of the size of the inf-sup constant β_1 in Equation (3.11), we use an approximation and investigate the smallest eigenvalue, $\lambda_{\min} = \beta_1^2$, of

$$\begin{bmatrix} A_{\boldsymbol{v}} & B_1^T \\ B_1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = -\lambda \begin{bmatrix} 0 & 0 \\ 0 & M_p \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix}\,, \tag{3.22}$$

where the operators $B_1, A_{\boldsymbol{v}}$ and M_p are induced by the bilinear forms $b_1(\boldsymbol{w}, p) = -\varepsilon(p, \nabla\cdot \boldsymbol{w}) - (\boldsymbol{\alpha}p, \boldsymbol{w})$, $a_v(\boldsymbol{v}, \boldsymbol{w})$ and (p, q), respectively. This is now a symmetric problem, where we focus on discovering β_1 by removing the contribution of β_2, which is proven to be bounded away from zero for $\mathcal{RT}_k \times \mathcal{P}_k^{\mathrm{dc}}$. It is straightforward to use standard routines from the ARPACK [LS98] library to find the smallest eigenvalues and then to calculate the magnitude of the smallest values of the LBB inf-sup constant for different values of diffusion and different meshes.

3.2.1.1 Numerical results for β_1

We carry out numerical tests with the total flux formulation on the system in Equation (3.22), using a series of structured Friedrichs-Keller meshes as in Section 2.4.1, depicted in Figure 2.2, where N is the number of segments along one side and domain $\Omega = (0, 1)^2$. First, we use values of $\mu = 0$, $\boldsymbol{\alpha} = [1, 0]^T$, homogenous Dirichlet and Neumann boundary conditions and the lowest order Raviart-Thomas elements $\mathcal{RT}_0 \times \mathcal{P}_0$ and then $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$. Secondly, we change the flux to $\boldsymbol{\alpha} = [-y^3, x^3]^T$ and repeat the tests. The value of ε is varied from 10^{-4} to 1 and N is varied from 2^3 to 2^8 (or 2^9) depending on computer memory restrictions for different Raviart-Thomas pairs.
The results are shown in Tables 3.1–3.4, where the decimal values of $h = \dfrac{1}{N}$ have been added for ease of comparison. Note: It was not possible to use FreeFem++ on finer discretisations , particularly for $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, due to computer memory limitations and eigenvalues are obtained accurate to 10^{-10}, giving the inf-sup constant accurate to 5 decimal places. The approximate value of β_1 decreases when ε decreases, but it remains uniformly bounded away from zero. Since problem (3.22) is not definite, no monotonic behaviour is expected for its eigenvalues, which can be observed in the tables. In the second case, with the flux $\boldsymbol{\alpha} = [-y^3, x^3]^T$, the inf-sup constant for both $\mathcal{RT}_0 \times \mathcal{P}_0$ and $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ is almost identical and $\beta_1 \approx 0.10256\varepsilon$, where ε is the value of diffusion.

Table 3.1 Variation of inf-sup constant β_1 with h and ε for $\mathcal{RT}_0 \times \mathcal{P}_0$, $\boldsymbol{\alpha} = [1, 0]^T$

ε	$h = 2^{-3}$ $= 0.125$	$h = 2^{-4}$ $= 0.0625$	$h = 2^{-5}$ $= 0.0313$	$h = 2^{-6}$ $= 0.0156$	$h = 2^{-7}$ $= 0.0078$	$h = 2^{-8}$ $= 0.0039$	$h = 2^{-9}$ $= 0.0020$	$h = 2^{-10}$ $= 0.0010$
1.0	0.09740	0.09692	0.08571	0.09134	0.09667	0.09486	0.09591	0.09612
0.1	0.00954	0.00940	0.00890	0.00954	0.00920	0.00950	0.00940	0.00947
0.01	0.00217	0.00148	0.00116	0.00106	0.00103	0.00089	0.00094	0.00094
0.001	0.00197	0.00093	0.00050	0.00029	0.00017	0.00012	0.00011	0.00010
0.0001	0.00196	0.00093	0.00053	0.00027	0.00013	7.0E-05	3.0E-05	2.0E-05

Table 3.2 Variation of inf-sup constant β_1 with h and ε for $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\boldsymbol{\alpha} = [1, 0]^T$

ε	$h = 2^{-3}$ $= 0.125$	$h = 2^{-4}$ $= 0.0625$	$h = 2^{-5}$ $= 0.0313$	$h = 2^{-6}$ $= 0.0156$	$h = 2^{-7}$ $= 0.0078$	$h = 2^{-8}$ $= 0.0039$	$h = 2^{-9}$ $= 0.0020$
1.0	0.08886	0.09627	0.09593	0.09497	0.09674	0.08679	0.09797
0.10	0.00956	0.00899	0.00946	0.00950	0.00937	0.00951	0.00856
0.01	0.00125	0.00104	0.00097	0.00092	0.00096	0.00090	0.00084
0.001	0.00082	0.00042	0.00021	0.00006	0.00011	0.00010	0.00010
0.0001	0.00075	0.00040	0.00019	0.00010	5.0E-05	3.0.0E-05	2.0.0E-05

Table 3.3 Variation of inf-sup constant β_1 with h and ε for $\mathcal{RT}_0 \times \mathcal{P}_0$,$\boldsymbol{\alpha} = [-y^3, x^3]^T$

ε	$h = 2^{-3}$ $= 0.125$	$h = 2^{-4}$ $= 0.0625$	$h = 2^{-5}$ $= 0.0313$	$h = 2^{-6}$ $= 0.0156$	$h = 2^{-7}$ $= 0.0078$	$h = 2^{-8}$ $= 0.0039$	$h = 2^{-9}$ $= 0.0020$	$h = 2^{-10}$ $= 0.0010$
1.0	0.10245	0.10252	0.10254	0.10256	0.10256	0.10256	0.10256	0.10256
0.1	0.01024	0.01025	0.01026	0.01026	0.01026	0.01026	0.01026	0.01026
0.01	0.00102	0.00103	0.00103	0.00103	0.00103	0.00103	0.00103	0.00103
0.001	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010
0.0001	1.0E-05	1.0E-05	1.0E-05	1.0E-05	1.0E-05	1.0E-05	1.0E-05	1.0E-05

Table 3.4 Variation of inf-sup constant β_1 with h and ε for $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\boldsymbol{\alpha} = [-y^3, x^3]^T$

ε	$h = 2^{-3}$ $= 0.125$	$h = 2^{-4}$ $= 0.0625$	$h = 2^{-5}$ $= 0.0313$	$h = 2^{-6}$ $= 0.0156$	$h = 2^{-7}$ $= 0.0078$	$h = 2^{-8}$ $= 0.0039$	$h = 2^{-9}$ $= 0.0020$
1.0	0.10248	0.10252	0.10254	0.10256	0.10256	0.10256	0.10256
0.1	0.01024	0.01025	0.01026	0.01026	0.01026	0.01026	0.01026
0.01	0.00103	0.00103	0.00103	0.00103	0.00103	0.00103	0.00103
0.001	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010
0.0001	1.0E-05	1.0E-05	1.0-05	1.0E-05	1.0E-05	1.0E-05	1.0E-05

3.2.2 An exploration of the inf-sup constant σ of $\mathcal{A}$ in Equation (3.16)

We rearrange Equation (3.16) and state here for convenience of the reader.

$$\inf_{(\boldsymbol{v}_h,p_h)} \sup_{(\boldsymbol{w}_h,q_h)} \mathcal{A}\left((\boldsymbol{v}_h,p_h),(\boldsymbol{w}_h,q_h)\right) \geq \sigma \|(\boldsymbol{v}_h,p_h)\|_R \|(\boldsymbol{w}_h,q_h)\|_R \,, \tag{3.23}$$

$$\forall (\boldsymbol{v}_h,p_h),(\boldsymbol{w}_h,q_h) \ \in R = \mathcal{RT}_k \times \mathcal{P}_k^{\mathrm{dc}} \,. \tag{3.24}$$

At present, we are measuring stability with the $\mathbb{R}^N$ norms which are dependant on the size of the two discretisations used for $\boldsymbol{v}$ and p. However, an alternative investigation can be carried out on the stability of the total formulation using norms induced by the matrix $\mathbb{B}$, where $\mathbb{B}$ is the combined mass matrix given by right-hand side in Equation (3.23), consisting of $A_{\boldsymbol{v}}$ and M_p. This results in the inf-sup constant σ for the global system being independent of the level of the mesh refinement and the approach uses singular value decomposition as an alternative to eigenvalue decomposition. This is advantageous as the left-hand side of Section 3.2.2 with the bilinear form $\mathcal{A}(.,.)$ denoted by matrix $\mathbb{A}$, is not symmetric and would result in complex pairs of eigenvalues; whereas singular value decomposition results in positive, real values. This allows us to explore σ in Section 3.2.2.

The challenge now is to find equivalent inequalities for Equations (3.11)–(3.12) using a norm induced by the matrix $\mathcal{B}$ instead of the $\mathbb{R}^N$ norm.

Starting from the result of [CF03] in terms of our variables p and $\boldsymbol{v}$, and a general matrix $\widetilde{\mathbb{A}}$, with the inf-sup constant as σ gives

$$\sigma \|p\|_2 \leq \max_{\boldsymbol{v}\in\mathbb{R}^n} \frac{\boldsymbol{v}^T \widetilde{\mathbb{A}}\, p}{\|\boldsymbol{v}\|_2} \,, \tag{3.25}$$

where $\|\cdot\|_2$ is the Euclidian norm in $\mathbb{R}^N$
Then, following the analysis of [BW18] by changing the variables $\widetilde{\mathbb{A}} = L^{-T}\mathbb{A}L^{-1}$ where L is defined by the Cholesky decomposition $\mathbb{B} = L^T L$, $\boldsymbol{v} = L\widetilde{\boldsymbol{v}}$ and $p = L\widetilde{p}$, leads to

$$\begin{aligned}
\sigma \|L\widetilde{p}\|_2 &\leq \max_{\widetilde{\boldsymbol{v}}\in\mathbb{R}^n} \frac{(L\widetilde{\boldsymbol{v}})^T \left(L^{-T}\mathbb{A}L^{-1}\right)(L\widetilde{p})}{\|L\widetilde{\boldsymbol{v}}\|_2} && (3.26)\\
&\leq \max_{\widetilde{\boldsymbol{v}}\in\mathbb{R}^n} \frac{\widetilde{\boldsymbol{v}}^T L^T L^{-T}\mathbb{A}L^{-1}L\widetilde{p}}{\|L\widetilde{\boldsymbol{v}}\|_2} \\
&\leq \max_{\widetilde{\boldsymbol{v}}\in\mathbb{R}^n} \frac{\widetilde{\boldsymbol{v}}^T \widetilde{\mathbb{A}}\widetilde{p}}{\|L\widetilde{\boldsymbol{v}}\|_2} \,, \quad \forall \widetilde{p} \in \mathbb{R}^n \,. && (3.27)
\end{aligned}$$

Now we use the following to change the induced norms

$$\|\widetilde{p}\|_{\mathbb{B}}^2 = \langle \mathbb{B}\widetilde{p}, \widetilde{p}\rangle = \langle L^T L\widetilde{p}, \widetilde{p}\rangle = \langle L\widetilde{p}, L\widetilde{p}\rangle = \|L\widetilde{p}\|_2^2$$

and

$$\|\widetilde{\boldsymbol{v}}\|_{\mathbb{B}}^2 = \langle \mathbb{B}\widetilde{\boldsymbol{v}}, \widetilde{\boldsymbol{v}}\rangle = \langle L^T L, \widetilde{\boldsymbol{v}}\rangle = \langle L\widetilde{\boldsymbol{v}}, L\widetilde{\boldsymbol{v}}\rangle = \|L\widetilde{\boldsymbol{v}}\|_2^2 \,,$$

which leads to

$$\begin{aligned} \sigma\|\widetilde{p}\|_{\mathbb{B}} &\leq \max_{\widetilde{\boldsymbol{v}}\in\mathbb{R}^n} \frac{\widetilde{\boldsymbol{v}}^T\widetilde{\mathbb{A}}\widetilde{p}}{\|\widetilde{\boldsymbol{v}}\|_{\mathbb{B}}} \quad \text{and} \quad \text{finally} \\ \sigma &\leq \min_{\widetilde{p}\in\mathbb{R}^n}\max_{\widetilde{\boldsymbol{v}}\in\mathbb{R}^n} \frac{\widetilde{\boldsymbol{v}}^T\widetilde{\mathbb{A}}\widetilde{p}}{\|\widetilde{\boldsymbol{v}}\|_{\mathbb{B}}\|\widetilde{p}\|_{\mathbb{B}}} \,. \end{aligned} \tag{3.28}$$

Equation (3.28) now gives the inf-sup value with both norms scaled by matrix $\mathbb{B}$. Therefore, from Equation (3.26) the inf-sup constant σ is defined as the smallest singular value of $\widetilde{\mathbb{A}} = L^{-T}\mathbb{A}L^{-1}$ where L is defined by the Cholesky decomposition $\mathbb{B} = L^T L$.

The values of the inf-sup constant σ for the whole system can be seen in Tables 3.5-3.7. These values are calculated using Matlab double precision with an machine accuracy of 2.2E-16. There is a memory limit as the larger matrices exceed 32GB for storage, even before calculations are started, so it was only possible to obtain a few values without using super-computing facilities. We note that as diffusion decreases to $\varepsilon = 10^{-3}$ and 10^{-4} and as the meshes are refined, the inf-sup value for the whole system deteriorates more rapidly than that of β_1.

Table 3.5 Variation of inf-sup constant σ with h and ε for $\mathcal{RT}_0 \times \mathcal{P}_0$, $\boldsymbol{\alpha} = [1, 0]^T$

ε	$h = 2^{-3}$ $= 0.125$	$h = 2^{-4}$ $= 0.0625$	$h = 2^{-5}$ $= 0.0313$	$h = 2^{-6}$ $= 0.0156$
1.0	1.01E-03	3.55E-04	5.99E-05	1.49E-05
0.1	1.07E-03	2.77E-04	5.84E-05	1.47E-05
0.01	7.84E-05	3.37E-05	1.48E-05	5.29E-06
0.001	8.14E-07	1.19E-06	6.20E-07	3.09E-07
0.0001	9.04E-10	1.62E-09	3.44E-09	5.91E-09

Table 3.6 Variation of inf-sup constant σ with h and ε for $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$, $\boldsymbol{\alpha} = [1, 0]^T$

ε	$h = 2^{-3}$ $= 0.125$	$h = 2^{-4}$ $= 0.0625$	$h = 2^{-5}$ $= 0.0313$
1.0	1.68E-07	1.22E-08	7.53E-10
0.1	1.76E-07	1.23E-08	7.57E-10
0.01	8.05E-08	6.73E-09	6.07E-10
0.001	1.22E-11	1.36E-10	5.47E-11
0.0001	7.01E-12	4.08E-13	2.00E-16

Table 3.7 Variation of inf-sup constant σ with h and ε for both $\mathcal{RT}_0 \times \mathcal{P}_0$ and $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$,$\boldsymbol{\alpha} = [-y^3, x^3]^T$

ε	$h = 2^{-3}$ $= 0.125$	$h = 2^{-4}$ $= 0.0625$	$h = 2^{-5}$ $= 0.0313$
1.0	8.40E-06	2.67E-06	5.13E-07
0.1	8.40E-06	2.67E-06	5.13E-07
0.01	8.40E-06	2.67E-06	5.13E-07
0.001	8.40E-06	2.67E-06	5.13E-07
0.0001	6.01E-06	2.63E-06	5.13E-07

Table 3.8 Variation of the condition number of $\mathcal{A}$ with mesh width h and diffusion ε

ε	$h = \frac{1}{8}$	$h = \frac{1}{16}$	$h = \frac{1}{32}$	$h = \frac{1}{64}$
0.1	1.42E+02	4.78E+02	1.83E+03	7.25E+03
0.01	1.45E+03	1.63E+03	3.57E+03	1.25E+04
0.001	8.60E+04	3.48E+04	4.14E+04	4.02E+04
0.0001	7.48E+07	1.97E+07	4.76E+06	1.46E+06

3.3 Conclusion

The LBB inf-sup constant β_1 remains above zero but becomes extremely small as mesh size is increased and diffusion becomes smaller. In the case of $\boldsymbol{\alpha} = [-y^3, x^3]^T$, β_1 is directly proportional to diffusion. The inf-sup constant σ for the total system is notably smaller than β_1. This means that the stability of the formulation is very tenuous. Using $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ pairs of elements does not improve matters, but rather it makes the discretisation less stable. This accords with Figure 2.7f for $\varepsilon \leq 10^{-4}$. The return to convergence as the mesh size increases in Figures 2.7d and 2.7e is hopeful, but the values for σ in Tables 3.6 and 3.7 show a definite trend to zero, particularly for $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$. There is also a more rapid deterioration in the condition number of $\mathcal{A}$ as ε gets smaller than 10^{-3} that is seen in Table 3.8. Unsurprisingly, this unstabilised formulation with either $\mathcal{RT}_0 \times \mathcal{P}_0$ or $\mathcal{RT}_1 \times \mathcal{P}_1^{\mathrm{dc}}$ pairs of elements does not cope with layers when diffusion is small.

In their papers [DR82, DR85] Douglas and Roberts' proof is that 'h small enough' implies existence and convergence. Our results seem to imply that existence is guaranteed for any h, but that the stability constants are extremely small for small ε and therefore the method is unstable in practice.

Chapter 4

The new stabilised method: Analysis and Tests

This chapter is organised as follows. First the stability analysis and error analysis developed with colleagues G. R. Barrenechea and A. Poza and published in our paper [BPY18] for our new method outlined in Equations (4.7) and (4.8) is presented. Then the convergence of this new method is tested with various standard tests in 2-D ($d = 2$) and some testing in 3-D ($d = 3$). Finally we explore the effect of changing the values of the parameter δ_{div} in Equation (4.9).

This chapter is the paper published in [BPY18] without Convergence tests A and C. Test A has been added for consistency with Chapter 2 and it is used with both positive and negative combinations of the convective flux $\boldsymbol{\alpha}$. Test C includes non-homogenous Dirichlet conditions which differ on adjacent sides and reaction terms of $\mu = 0$ and $\mu = 1$. We consider $k \geq 1$ and recall the definition of the spaces as defined in Section 2.1

$$\boldsymbol{H}_h := \left\{ \boldsymbol{\varphi} \in C^0(\overline{\Omega})^d : \boldsymbol{\varphi}|_T \in \mathcal{P}_k(T)^d \quad \forall T \in \mathcal{T}_h \right\}, \tag{4.1}$$

and the discrete subspace for the scalar variable p as

$$Q_h^0 := Q_h \cap H_0^1(\Omega) \text{ where } Q_h := \left\{ q_h \in C^0(\overline{\Omega}) : q_h|_T \in \mathcal{P}_k(T) \ , \ \forall T \in \mathcal{T}_h \right\}. \tag{4.2}$$

All statements made in this chapter are valid for all $k \geq 1$. The constants involved in general depend on k.

4.1 Preliminary Results

We denote by $\boldsymbol{\Pi}_h$ the L^2–orthogonal projection onto $\boldsymbol{H}_h$ used in the sequel and defined by

$$(\boldsymbol{\Pi}_h(\boldsymbol{v}), \boldsymbol{w}_h) = (\boldsymbol{v}, \boldsymbol{w}_h) \qquad \forall \boldsymbol{w}_h \in \boldsymbol{H}_h. \tag{4.3}$$

Lemma 4.1. *There exists a positive constant C, independent of h, such that*

$$\|\boldsymbol{\Pi}_h(\boldsymbol{v})\|_{0,\Omega} \leq \|\boldsymbol{v}\|_{0,\Omega} \quad \forall \boldsymbol{v} \in L^2(\Omega)^d, \tag{4.4}$$

$$\|\boldsymbol{v} - \boldsymbol{\Pi}_h(\boldsymbol{v})\|_{0,\Omega} \leq C\, h\, |\boldsymbol{v}|_{1,\Omega} \quad \forall \boldsymbol{v} \in H^1(\Omega)^d. \tag{4.5}$$

Proof. See Lemma 1.131 in [EG13]. □

We recall the following inverse inequality, which will be used throughout, and whose proof is a direct consequence of classical inverse inequalities for polynomial functions (see e.g., [EG13, Lemma 1.138]): There exists $C_k > 0$, depending only on k and the regularity of the mesh, such that, for all $\boldsymbol{w}_h \in \boldsymbol{H}_h$:

$$h_T \, \|\nabla \cdot \boldsymbol{w}_h\|_{0,T} \leq C_k \|\boldsymbol{w}_h\|_{0,T} \qquad \forall\, T \in \mathcal{T}_h\,. \tag{4.6}$$

4.2 The stabilised finite element method

As mentioned in the introduction, our method is a modification of the one from [MK08] (see Section 2.6.1 for details). More precisely, our stabilised finite element method studied in this thesis and termed the Present Method reads: find $(\boldsymbol{v}_h, p_h) \in \boldsymbol{H}_h \times Q_h^0$ such that

$$B((\boldsymbol{v}_h, p_h), (\boldsymbol{w}_h, q_h)) = (f, q_h) + \sum_{T \in \mathcal{T}_h} \delta_{div}^T (f, \nabla \cdot \boldsymbol{w}_h + \mu\, q_h)_T, \tag{4.7}$$

for all $(\boldsymbol{w}_h, q_h) \in \boldsymbol{H}_h \times Q_h^0$, where the bilinear form $B(\cdot,\cdot)$ is given by

$$\begin{aligned} & B((\boldsymbol{v}_h, p_h), (\boldsymbol{w}_h, q_h)) \\ := \;& \frac{1}{\varepsilon}(\boldsymbol{v}_h, \boldsymbol{w}_h) - (p_h, \nabla \cdot \boldsymbol{w}_h) + (\nabla \cdot \boldsymbol{v}_h, q_h) - \frac{1}{\varepsilon}(\boldsymbol{\alpha}\, p_h, \boldsymbol{w}_h) + \mu\,(p_h, q_h) \\ & - \frac{\varepsilon}{2}\left(\frac{1}{\varepsilon}\boldsymbol{v}_h + \nabla p_h - \frac{1}{\varepsilon}\boldsymbol{\alpha}\, p_h, \frac{1}{\varepsilon}\boldsymbol{w}_h - \nabla q_h + \frac{1}{\varepsilon}\boldsymbol{\alpha}\, q_h\right) \\ & + \sum_{T \in \mathcal{T}_h} \delta_{div}^T (\nabla \cdot \boldsymbol{v}_h + \mu\, p_h, \nabla \cdot \boldsymbol{w}_h + \mu\, q_h)_T, \end{aligned} \tag{4.8}$$

and the stabilisation parameter δ_{div} is defined as

$$\delta_{div}^T := \delta \min\left\{h_T, \frac{h_T^2}{4\varepsilon}\right\} \quad \text{where } \delta > 0 \text{ is arbitrary.} \tag{4.9}$$

In what follows we will denote $\delta_{div} := \max_{T\in\mathcal{T}_h} \delta_{div}^T$.

Remark 4.2. *Although of similar shape, Method* (4.7) *and Masud-Kwack's method [MK08] contain significant differences. The first is the addition of the convective term in the test function for the stabilising term. This is added to make the analysis possible (in fact, to the best of our knowledge, there is no analysis for the original method from [MK08]). Moreover, the div-div term added to the formulation improves the numerical results significantly.*

The stability and error analysis will be carried out using the following mesh-dependent norm:

$$\|(\boldsymbol{w}, q)\|_h := \left\{\frac{1}{2\varepsilon}\|\boldsymbol{w} - \boldsymbol{\alpha}\, q\|_{0,\Omega}^2 + \varepsilon\,|q|_{1,\Omega}^2 + \mu\,\|q\|_{0,\Omega}^2 + \sum_{T\in\mathcal{T}_h} \delta_{div}^T\,\|\nabla\cdot\boldsymbol{w} + \mu\, q\|_{0,T}^2\right\}^{1/2}. \tag{4.10}$$

Using this norm, we present the main result about stability of the method.

Remark 4.3. *For the continuous problem* $\boldsymbol{v} = -\varepsilon\nabla p + \boldsymbol{\alpha} p$ *and thus*

$$h^2\|\nabla\cdot\boldsymbol{w} + \mu q\|_{0,T}^2 \approx h^2\| - \varepsilon\Delta q + \nabla\cdot(\boldsymbol{\alpha} q) + \mu q\|_{0,T}^2$$

and

$$\frac{1}{2\varepsilon}\|\boldsymbol{v} - \boldsymbol{\alpha} p\|_{0,\Omega}^2 = \frac{\varepsilon}{2}\|\nabla p\|_{0,\Omega}^2\,.$$

So the norm $\|.\|_h$ *is similar to the SUPG norm for the continuous problem. For the discrete problem,* $\|.\|_h$ *is slightly stronger as it involves an* $L^2(\Omega)$ *norm of* $\boldsymbol{v}_h$.

Theorem 4.4. *Let* $B(\cdot,\cdot)$ *be the bilinear form given by* (4.8). *Then, there exists a positive constant* C, *independent of* $\varepsilon, \mu, h, \delta$, *and* $\boldsymbol{\alpha}$, *such that*

$$\sup_{(\boldsymbol{w}_h, q_h)\in \boldsymbol{H}_h\times Q_h^0} \frac{B\left((\boldsymbol{v}_h, p_h), (\boldsymbol{w}_h, q_h)\right)}{\|(\boldsymbol{w}_h, q_h)\|_h} \geq C\,\|(\boldsymbol{v}_h, p_h)\|_h, \tag{4.11}$$

for all $(\boldsymbol{v}_h, p_h) \in \boldsymbol{H}_h \times Q_h^0$. *Thus,* (4.7) *is well-posed.*

Proof. Let $(\boldsymbol{v}_h, p_h) \in \boldsymbol{H}_h \times Q_h^0$. First, using the definition of $B(\cdot,\cdot)$, and Cauchy–Schwarz and Young inequalities we arrive at

$$\begin{aligned} B((\boldsymbol{v}_h, p_h), (\boldsymbol{v}_h, p_h)) =& \frac{1}{\varepsilon} \|\boldsymbol{v}_h\|_{0,\Omega}^2 - \frac{1}{\varepsilon} (\boldsymbol{\alpha}\, p_h, \boldsymbol{v}_h) - \frac{1}{2\varepsilon} \{\|\boldsymbol{v}_h\|^2 - \|\varepsilon \nabla p_h - \boldsymbol{\alpha} p_h\|^2\}_{0,\Omega}^2 \\ &+ \mu \|p_h\|_{0,\Omega}^2 + \sum_{T \in \mathcal{T}_h} \delta_{div}^T \|\nabla \cdot \boldsymbol{v}_h + \mu\, p_h\|_{0,T}^2 \\ \geq& \frac{\varepsilon}{2} |p_h|_{1,\Omega}^2 + \mu \|p_h\|_{0,\Omega}^2 + \sum_{T \in \mathcal{T}_h} \delta_{div}^T \|\nabla \cdot \boldsymbol{v}_h + \mu\, p_h\|_{0,T}^2 . \end{aligned} \tag{4.12}$$

Let now $\boldsymbol{w}_h \in \boldsymbol{H}_h$. The definition of $B(\cdot,\cdot)$ and integration by parts give

$$\begin{aligned} & B((\boldsymbol{v}_h, p_h), (\boldsymbol{w}_h, 0)) \\ =& \frac{1}{\varepsilon} (\boldsymbol{v}_h, \boldsymbol{w}_h) - (p_h, \nabla \cdot \boldsymbol{w}_h) - \frac{1}{\varepsilon} (\boldsymbol{\alpha}\, p_h, \boldsymbol{w}_h) - \frac{\varepsilon}{2} \left(\frac{1}{\varepsilon} \boldsymbol{v}_h + \nabla p_h - \frac{1}{\varepsilon} \boldsymbol{\alpha}\, p_h, \frac{1}{\varepsilon} \boldsymbol{w}_h \right) \\ & + \sum_{T \in \mathcal{T}_h} \delta_{div}^T (\nabla \cdot \boldsymbol{v}_h + \mu\, p_h, \nabla \cdot \boldsymbol{w}_h)_T \\ =& \frac{1}{2\varepsilon} (\boldsymbol{v}_h - \boldsymbol{\alpha}\, p_h, \boldsymbol{w}_h) + \frac{1}{2} (\nabla p_h, \boldsymbol{w}_h) + \sum_{T \in \mathcal{T}_h} \delta_{div}^T (\nabla \cdot \boldsymbol{v}_h + \mu\, p_h, \nabla \cdot \boldsymbol{w}_h)_T . \end{aligned}$$

Thus, using (4.9), (4.6), taking $\tilde{\boldsymbol{w}}_h := \boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)$, and using the Cauchy-Schwarz, Young, and inverse inequalities we obtain

$$\begin{aligned} & B((\boldsymbol{v}_h, p_h), (\tilde{\boldsymbol{w}}_h, 0)) \\ =& \frac{1}{2\varepsilon} \|\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)\|_{0,\Omega}^2 + \frac{1}{2} (\nabla p_h, \boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)) \\ & + \sum_{T \in \mathcal{T}_h} \delta_{div}^T (\nabla \cdot \boldsymbol{v}_h + \mu\, p_h, \nabla \cdot (\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)))_T \\ \geq& \frac{1}{4\varepsilon} \|\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)\|_{0,\Omega}^2 - \frac{\varepsilon}{4} |p_h|_{1,\Omega}^2 \\ & - \sum_{T \in \mathcal{T}_h} \left\{ \frac{\delta_{div}^T C_k^2 \delta}{2} \|\nabla \cdot \boldsymbol{v}_h + \mu\, p_h\|_{0,T}^2 + \frac{\delta_{div}^T}{2 C_k^2 \delta} \|\nabla \cdot (\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h))\|_{0,T}^2 \right\} \\ \geq& \frac{1}{4\varepsilon} \|\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)\|_{0,\Omega}^2 - \frac{\varepsilon}{4} |p_h|_{1,\Omega}^2 \\ & - \sum_{T \in \mathcal{T}_h} \left\{ \frac{\delta_{div}^T C_k^2 \delta}{2} \|\nabla \cdot \boldsymbol{v}_h + \mu\, p_h\|_{0,T}^2 + \frac{\delta_{div}^T}{2 \delta h_T^2} \|\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)\|_{0,T}^2 \right\} \\ \geq& \frac{1}{8\varepsilon} \|\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)\|_{0,\Omega}^2 - \frac{\varepsilon}{4} |p_h|_{1,\Omega}^2 - \sum_{T \in \mathcal{T}_h} \frac{\delta_{div}^T C_k^2 \delta}{2} \|\nabla \cdot \boldsymbol{v}_h + \mu\, p_h\|_{0,T}^2 . \end{aligned} \tag{4.13}$$

Adding (4.12) and (4.13), and defining $\gamma := \min\{1, (\delta C_k^2)^{-1}\}$, the following holds

$$
\begin{aligned}
&B\left((\boldsymbol{v}_h, p_h), (\boldsymbol{v}_h + \gamma\tilde{\boldsymbol{w}}_h, p_h)\right) \\
&\geq \frac{\varepsilon(4-\gamma)}{8} |p_h|^2_{1,\Omega} + \mu \|p_h\|^2_{0,\Omega} + \frac{\gamma}{8\varepsilon} \|\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)\|^2_{0,\Omega} \\
&\quad + \sum_{T\in\mathcal{T}_h} \delta^T_{div} \left(1 - \frac{C_k^2 \delta\gamma}{2}\right) \|\nabla\cdot\boldsymbol{v}_h + \mu\, p_h\|^2_{0,T} \\
&\geq C \|(\boldsymbol{v}_h, p_h)\|^2_h. \qquad (4.14)
\end{aligned}
$$

Finally, from (4.9), (4.6), and using that $\gamma \leq 1$, it follows that

$$
\begin{aligned}
\|(\boldsymbol{v}_h + \gamma\tilde{\boldsymbol{w}}_h, p_h)\|_h &\leq \left\{ \|(\boldsymbol{v}_h, p_h)\|_h + \frac{1}{\varepsilon^{1/2}} \|\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)\|_{0,\Omega} \right. \\
&\qquad \left. \left(\sum_{T\in\mathcal{T}_h} \delta^T_{div} \|\nabla\cdot(\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h))\|^2_{0,T} \right)^{\frac{1}{2}} \right\} \\
&\leq C \left\{ \|(\boldsymbol{v}_h, p_h)\|_h + \left(\sum_{T\in\mathcal{T}_h} \frac{\delta^T_{div} C_k^2}{h_T^2} \|\boldsymbol{v}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha}\, p_h)\|^2_{0,T} \right)^{\frac{1}{2}} \right\} \\
&\leq \tilde{C} \|(\boldsymbol{v}_h, p_h)\|_h,
\end{aligned}
$$

where $\tilde{C}$ is independent of ε, μ, h and $\boldsymbol{\alpha}$. Hence, from (4.14) the discrete inf–sup condition

$$
\begin{aligned}
\sup_{(\boldsymbol{w}_h, q_h)\in \boldsymbol{H}_h \times Q_h^0} \frac{B\left((\boldsymbol{v}_h, p_h), (\boldsymbol{w}_h, q_h)\right)}{\|(\boldsymbol{w}_h, q_h)\|_h} &\geq \frac{B\left((\boldsymbol{v}_h, p_h), (\boldsymbol{v}_h + \gamma\tilde{\boldsymbol{w}}_h, p_h)\right)}{\|(\boldsymbol{v}_h + \gamma\tilde{\boldsymbol{w}}_h, p_h)\|_h} \\
&\geq C \|(\boldsymbol{v}_h, p_h)\|_h,
\end{aligned}
$$

follows, which concludes the proof. □

4.3 Error analysis

Let $k \geq 1$. We introduce the Scott-Zhang interpolation operators $\boldsymbol{\mathcal{I}}_h : H^1(\Omega)^d \longrightarrow \boldsymbol{H}_h$ and $\mathcal{J}_h : H^1_0(\Omega) \longrightarrow Q_h^0$. These interpolation operators satisfy (see, e.g., [EG13])

$$
\begin{aligned}
|\eta^{\boldsymbol{v}}|_{m,\Omega} := |\boldsymbol{v} - \boldsymbol{\mathcal{I}}_h\boldsymbol{v}|_{m,\Omega} &\leq Ch^{s-m} |\boldsymbol{v}|_{s,\Omega} \qquad \forall \boldsymbol{v} \in H^s(\Omega)^d, \qquad (4.15) \\
|\eta^{p}|_{m,\Omega} := |p - \mathcal{J}_h p|_{m,\Omega} &\leq Ch^{s-m} |p|_{s,\Omega} \qquad \forall p \in H^s(\Omega)\cap H^1_0(\Omega), \qquad (4.16)
\end{aligned}
$$

for $0 \leq m \leq 2$ and $\max\{m, 1\} \leq s \leq k+1$.
Finally, we recall the discrete commutator property (see e.g., [EG13, Lemma 1.137]). This is a powerful tool used to analyse nonlinear problems which will be useful in the sequel:
There exists $c > 0$ such that, for all $h, v_h \in \mathcal{P}^k_{c,h}, \phi \in W^{s+1,\infty}(\Omega)$, and $0 \leq m \leq s \leq 1$

$$\|\phi v_h - \boldsymbol{\mathcal{I}}_h(\phi v_h)\|_{m,p,\Omega} \leq c\, h^{1+s-m} \|v_h\|_{s,p,\Omega} \|\phi\|_{s+1,\infty,\Omega}\,. \tag{4.17}$$

Since $\boldsymbol{\Pi}_h$ is the orthogonal projection on to $\boldsymbol{H}_h$, then, clearly

$$\|\boldsymbol{\alpha} q_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha} q_h)\|_{0,\Omega} \leq \|\boldsymbol{\alpha} q_h - \boldsymbol{\mathcal{I}}_h(\boldsymbol{\alpha} q_h)\|_{0,\Omega} \tag{4.18}$$

and then the properties of the Scott-Zhang interpolation operator in Equation (4.15) or the discrete commutator property in Equation (4.17) may be used.

The main error estimate for Method (4.7) is stated next.

Theorem 4.5. *Let $(\boldsymbol{v}, p) \in H^{k+1}(\Omega)^d \times \left[H^{k+1}(\Omega) \cap H^1_0(\Omega)\right]$ be the solution of (2.15) and $(\boldsymbol{v}_h, p_h) \in \boldsymbol{H}_h \times Q^0_h$ the solution of (4.7). Then, there exists a positive constant C, independent of ε, μ, and h, but dependent on k and δ such that*

$$\|(\boldsymbol{v} - \boldsymbol{v}_h, p - p_h)\|_h \leq C h^k \Big(M_1 \, |\boldsymbol{v}|_{k+1,\Omega} + M_2 \, |p|_{k+1,\Omega} \Big)\,, \tag{4.19}$$

where

$$M_1 = C_1 \frac{h}{\varepsilon} \quad , \quad M_2 = \mu^{1/2} h + \mu h^{3/2} + C_1 \left(\frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega} h}{\varepsilon} + 1 \right)\,,$$

and

$$C_1 = \min \left\{ \frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega}}{\mu^{1/2}}, \frac{\|\boldsymbol{\alpha}\|_{1,\infty,\Omega} h}{\mu^{1/2}} \right\} + \varepsilon^{1/2}\,. \tag{4.20}$$

Remark 4.6. *In the case where $\mu = 0$, we obtain*

$$C_1 = \min \left\{ \frac{h\|\boldsymbol{\alpha}\|_{0,\infty,\Omega}}{\varepsilon^{1/2}}, \frac{\|\boldsymbol{\alpha}\|_{1,\infty,\Omega} h}{\varepsilon^{1/2}} \right\} + \varepsilon^{1/2}\,. \tag{4.21}$$

This estimate blows up with ε when $\varepsilon \to 0$. An improved result for the case $\mu = 0$ has recently been obtained and will be communicated elsewhere.

Proof. First, using the definition of $\|\cdot\|_h$, the triangle inequality and estimates (4.15)-(4.16), we obtain

$$
\begin{aligned}
&\|(\eta^{\boldsymbol{v}},\eta^p)\|_h \\
&\leq \left\{ \frac{1}{\varepsilon^{1/2}}\,\|\eta^{\boldsymbol{v}}\|_{0,\Omega} + \frac{1}{\varepsilon^{1/2}}\,\|\boldsymbol{\Pi}_h(\boldsymbol{\alpha}\,\eta^p)\|_{0,\Omega} + \varepsilon^{1/2}\,|\eta^p|_{1,\Omega} + \right. \\
&\qquad\qquad \left. \mu^{1/2}\,\|\eta^p\|_{0,\Omega} + \delta_{div}^{1/2}\,\|\nabla\cdot\eta^{\boldsymbol{v}}\|_{0,\Omega} + \delta_{div}^{1/2}\mu\,\|\eta^p\|_{0,\Omega} \right\} \qquad (4.22)\\
&\leq \left\{ \frac{1}{\varepsilon^{1/2}}\,\|\eta^{\boldsymbol{v}}\|_{0,\Omega} + \frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega}}{\varepsilon^{1/2}}\,\|\eta^p\|_{0,\Omega} + \varepsilon^{1/2}\,|\eta^p|_{1,\Omega} + \right. \\
&\qquad\qquad \left. \mu^{1/2}\,\|\eta^p\|_{0,\Omega} + \delta_{div}^{1/2}\,|\eta^{\boldsymbol{v}}|_{1,\Omega} + \delta_{div}^{1/2}\mu\,\|\eta^p\|_{0,\Omega} \right\} \qquad (4.23)\\
&\leq Ch^k \left\{ \frac{h}{\varepsilon^{1/2}}\,|\boldsymbol{v}|_{k+1,\Omega} + \left[\varepsilon^{1/2}\left(\frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega} h}{\varepsilon} + 1 \right) + \mu^{1/2}h + \mu\,h^{3/2} \right] |p|_{k+1,\Omega} \right\}. \qquad (4.24)
\end{aligned}
$$

Next, let $(\boldsymbol{w}_h, q_h) \in \boldsymbol{H}_h \times Q_h^0$. Then, applying (4.4) to $id - \boldsymbol{\Pi}_h$ (where id denotes the identity operator) we get

$$
\|\boldsymbol{\alpha} q_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha} q_h)\|_{0,\Omega} \;\leq\; \|\boldsymbol{\alpha}\|_{0,\infty,\Omega}\,\|q_h\|_{0,\Omega} \leq \frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega}}{\mu^{1/2}}\,\|(\boldsymbol{w}_h, q_h)\|_h\,. \qquad (4.25)
$$

Alternatively, if we use a discrete commutator property (see Lemma 1.137 in [EG13]) we obtain

$$
\|\boldsymbol{\alpha} q_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha} q_h)\|_{0,\Omega} \;\leq\; C\,h\|\boldsymbol{\alpha}\|_{1,\infty,\Omega}\,\|q_h\|_{0,\Omega} \leq C\,\frac{\|\boldsymbol{\alpha}\|_{1,\infty,\Omega} h}{\mu^{1/2}}\,\|(\boldsymbol{w}_h, q_h)\|_h. \qquad (4.26)
$$

So, from (4.25) and (4.26), we get

$$
\|\boldsymbol{\alpha} q_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha} q_h)\|_{0,\Omega} \leq C\,\min\left\{ \frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega}}{\mu^{1/2}}, \frac{h\|\boldsymbol{\alpha}\|_{1,\infty,\Omega}}{\mu^{1/2}} \right\}\,\|(\boldsymbol{w}_h, q_h)\|_h. \qquad (4.27)
$$

Thus, using the triangle inequality and (4.27) we arrive at

$$
\|\boldsymbol{w}_h - \boldsymbol{\alpha} q_h\|_{0,\Omega} \leq \|\boldsymbol{w}_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha} q_h)\|_{0,\Omega} + \|\boldsymbol{\alpha} q_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha} q_h)\|_{0,\Omega} \leq C\,C_1\,\|(\boldsymbol{w}_h, q_h)\|_h, \quad (4.28)
$$

where C_1 is given by (4.21), for all $(\boldsymbol{w}_h, q_h) \in \boldsymbol{H}_h \times Q_h^0$.

Using the definition of B and integration by parts, we arrive at

$$\begin{aligned}
& B((\eta^{\boldsymbol{v}}, \eta^p), (\boldsymbol{w}_h, q_h)) \\
= \;& \frac{1}{2\varepsilon}(\eta^{\boldsymbol{v}} - \boldsymbol{\alpha}\eta^p, \boldsymbol{w}_h) + \frac{1}{2}\left(\nabla\eta^p, \boldsymbol{w}_h - \boldsymbol{\alpha}q_h\right) - \frac{1}{2\varepsilon}\left(\eta^{\boldsymbol{v}} - \boldsymbol{\alpha}\eta^p, \boldsymbol{\alpha}q_h\right) \\
& + \frac{\varepsilon}{2}(\nabla\eta^p, \nabla q_h) - \frac{1}{2}\left(\eta^{\boldsymbol{v}} + \boldsymbol{\alpha}\eta^p, \nabla q_h\right) + \mu\left(\eta^p, q_h\right) \\
& + \sum_{\mathcal{T}\in\mathcal{T}_h} \delta_{div}^T \left(\nabla\cdot\eta^{\boldsymbol{v}} + \mu\,\eta^p, \nabla\cdot\boldsymbol{w}_h + \mu\, q_h\right)_T \qquad (4.29) \\
= \;& \frac{1}{2}\left(\frac{1}{\varepsilon}\eta^{\boldsymbol{v}} - \frac{1}{\varepsilon}\,\boldsymbol{\alpha}\eta^p + \nabla\eta^p, \boldsymbol{w}_h - \boldsymbol{\alpha}q_h\right) + \frac{1}{2}\left(\varepsilon\,\nabla\eta^p - \eta^{\boldsymbol{v}} - \boldsymbol{\alpha}\eta^p, \nabla q_h\right) + \mu\left(\eta^p, q_h\right) \\
& + \sum_{\mathcal{T}\in\mathcal{T}_h} \delta_{div}^T \left(\nabla\cdot\eta^{\boldsymbol{v}} + \mu\,\eta^p, \nabla\cdot\boldsymbol{w}_h + \mu\, q_h\right)_T \\
= \;& I_1 + I_2 + I_3 + I_4. \qquad (4.30)
\end{aligned}$$

We bound the expression above term by term. First, I_1 is bounded using Cauchy–Schwarz inequality, estimate (4.15)-(4.16) and (4.27) as follows

$$\begin{aligned}
I_1 \;&\leq\; \left\{\frac{1}{\varepsilon}\,\|\eta^{\boldsymbol{v}}\|_{0,\Omega} + \frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega}}{\varepsilon}\|\eta^p\|_{0,\Omega} + |\eta^p|_{1,\Omega}\right\}\,\|\boldsymbol{w}_h - \boldsymbol{\alpha}q_h\|_{0,\Omega} \\
&\leq\; C\,C_1 h^k\,\left\{\frac{h}{\varepsilon}\,|\boldsymbol{v}|_{k+1,\Omega} + \left(\frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega}h}{\varepsilon} + 1\right)\,|p|_{k+1,\Omega}\right\}\,\|(\boldsymbol{w}_h, q_h)\|_h. \qquad (4.31)
\end{aligned}$$

Using the Cauchy-Schwarz inequality and (4.15)-(4.16), I_2 is bounded as follows

$$\begin{aligned}
I_2 &= \frac{\varepsilon}{2}\,\left(\nabla\eta^p, \nabla q_h\right) - \frac{1}{2}\left(\eta^{\boldsymbol{v}}, \nabla q_h\right) - \frac{1}{2}\left(\boldsymbol{\alpha}\eta^p, \nabla q_h\right) \\
&\leq Ch^k\left\{\frac{h}{\varepsilon^{1/2}}|\boldsymbol{v}|_{k+1,\Omega} + \varepsilon^{1/2}\left(1 + \frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega}h}{\varepsilon}\right)\,|p|_{k+1,\Omega}\right\}\,\|(\boldsymbol{w}_h, q_h)\|_h. \qquad (4.32)
\end{aligned}$$

For the third term in (4.30), we have

$$I_3 \;\leq\; C\,\mu\,\|\eta^p\|_{0,\Omega}\|q_h\|_{0,\Omega} \leq C\mu^{1/2}\,h^{k+1}|p|_{k+1,\Omega}\,\|(\boldsymbol{w}_h, q_h)\|_h. \qquad (4.33)$$

Finally, the last term in (4.30) is bounded as follows

$$\begin{aligned}
I_4 \;&\leq\; Ch^k\left\{\delta_{div}^{1/2}\,|\eta^{\boldsymbol{v}}|_{1,\Omega} + \delta_{div}^{1/2}\mu\,\|\eta^p\|_{0,\Omega}\right\}\,\|(\boldsymbol{w}_h, q_h)\|_h \\
&\leq\; C\delta h^k\left\{\frac{h}{\varepsilon^{1/2}}\,|\boldsymbol{v}|_{k+1,\Omega} + \mu h^{3/2}\,|p|_{k+1,\Omega}\right\}\,\|(\boldsymbol{w}_h, q_h)\|_h. \qquad (4.34)
\end{aligned}$$

Thus, defining $e_h^{\boldsymbol{v}} := \boldsymbol{v}_h - \boldsymbol{\mathcal{I}}_h \boldsymbol{v}$ and $e_h^p := p_h - \mathcal{J}_h p$, using the consistency of the scheme, (4.11), and combining (4.31)-(4.34) with (4.30), we arrive at

$$\begin{aligned}
\|(e_h^{\boldsymbol{v}}, e_h^p)\|_h &\leq C \sup_{(\boldsymbol{w}_h, q_h) \in \boldsymbol{H}_h \times Q_h^0} \frac{B((e_h^{\boldsymbol{v}}, e_h^p), (\boldsymbol{w}_h, q_h))}{\|(\boldsymbol{w}_h, q_h)\|_h} \\
&= C \sup_{(\boldsymbol{w}_h, q_h) \in \boldsymbol{H}_h \times Q_h^0} \frac{B((\eta^{\boldsymbol{v}}, \eta^p), (\boldsymbol{w}_h, q_h))}{\|(\boldsymbol{w}_h, q_h)\|_h} \\
&\leq C h^k \left\{ \frac{C_1 h}{\varepsilon} |\boldsymbol{v}|_{k+1,\Omega} + \left[C_1 \left(\frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega} h}{\varepsilon} + 1 \right) + \mu^{1/2} h + \mu h^{3/2} \right] |p|_{k+1,\Omega} \right\}. \quad (4.35)
\end{aligned}$$

Then using the triangle inequality we arrive at

$$\|(\boldsymbol{v} - \boldsymbol{v}_h, p - p_h)\|_h \leq \|(\eta^{\boldsymbol{v}}, \eta^p)\|_h + \|(e_h^{\boldsymbol{v}}, e_h^p)\|_h,$$

and the result follows using (4.35) and (4.24). □

Remark 4.7. *If we suppose $\boldsymbol{\alpha} \in W^{2,\infty}(\Omega)^d$ then a further use of the discrete commutator property gives*

$$\|\boldsymbol{\alpha} q_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha} q_h)\|_{0,\Omega} \leq C\, h^2 \|\boldsymbol{\alpha}\|_{2,\infty,\Omega}\, |q_h|_{1,\Omega} \leq C\, \frac{\|\boldsymbol{\alpha}\|_{2,\infty,\Omega} h^2}{\varepsilon^{1/2}}\, \|(\boldsymbol{w}_h, q_h)\|_h\,. \quad (4.36)$$

Thus, combining this estimate with (4.26) *we obtain*

$$\|\boldsymbol{\alpha} q_h - \boldsymbol{\Pi}_h(\boldsymbol{\alpha} q_h)\|_{0,\Omega} \leq C\, C_1\, \|(\boldsymbol{w}_h, q_h)\|_h, \quad (4.37)$$

but now with $C_1 := \min\left\{ \frac{\|\boldsymbol{\alpha}\|_{0,\infty,\Omega}}{\mu^{1/2}}, \frac{\|\boldsymbol{\alpha}\|_{1,\infty,\Omega} h}{\mu^{1/2}}, \frac{\|\boldsymbol{\alpha}\|_{2,\infty,\Omega} h^2}{\varepsilon^{1/2}} \right\} + \varepsilon^{1/2}$ *in Theorem 4.5.*

4.4 Convergence testing of our Present Method

For the initial numerical testing of our Present Method, the value of δ in Equation (4.9) is set to 1.0. A later exploration of the effect of different values of δ in the δ_{div} term on the convergence is carried out in Section (4.4.4). The same meshes are used in these computations as shown in Figure 2.2.

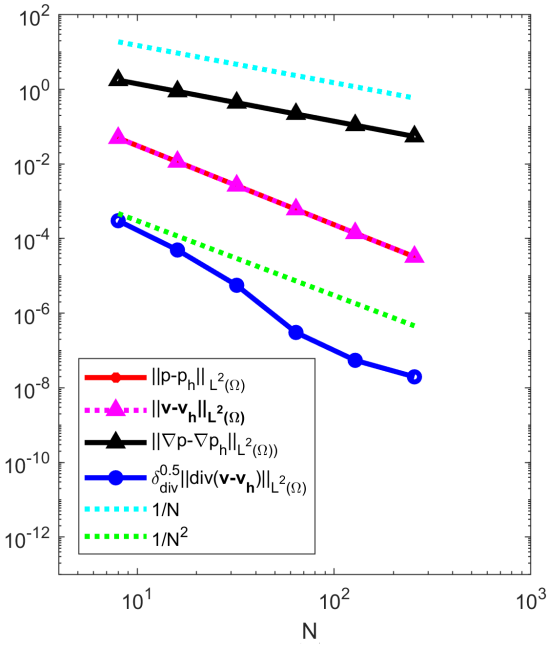
4.4.1 Results of Convergence Test A

We first test our new method using convergence test A (see Section 2.4.1). The results depicted in Figure 4.1 show our Present Method converges in the case of $\mathcal{P}_1\mathcal{P}_1$ linear elements, with errors of the order of $\frac{1}{N}$ for grad p and $\frac{1}{N^2}$ for both the primary variable p, secondary variable $\boldsymbol{v}$ and the divergence of $\boldsymbol{v}$. It is difficult to distinguish p and $\boldsymbol{v}$ when diffusion is small and both lines are identical in the scale of the graph, so $\boldsymbol{v}$ is drawn with a dotted line. It should also be noted that the δ_{div} term is included in the calculation of the divergence errors for all the error plots for the Present Method. This test is repeated with different values of $\boldsymbol{\alpha}$, which include both positive and negative values, and we find that all values work equally well.
Similarly the graphs for the errors in Figure 4.1 in the case of the quadratic $\mathcal{P}_2\mathcal{P}_2$ elements also converge with appropriate rates.

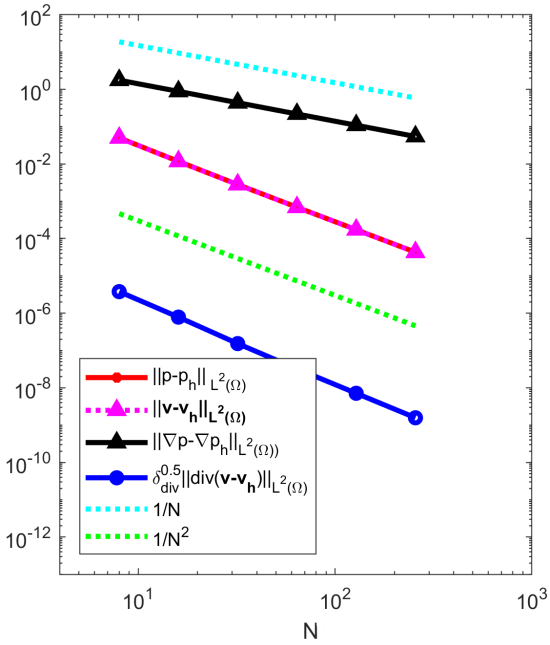
4.4.2 Testing for Convergence using Test C: Method and results

We now test the convergence of our Present Method in a two-dimensional example with a smooth, known solution and variable convective flux $\boldsymbol{\alpha}$. This is a more demanding test of convergence due to the rotating, variable nature of the flux compared with the constant flux applied in Test A.
We consider $\Omega = (0,1)^2$, $\boldsymbol{\alpha} = [y, -x]^T$, $\mu = 0$, and test with different values of ε ranging from 10^{-5} to 1. Both f and the boundary conditions are chosen such that the solution of (2.5) is given by $p(x,y) = 100x^2(1-x)^2y(1-y)(1-2y)$, depicted in Figure 4.2a.

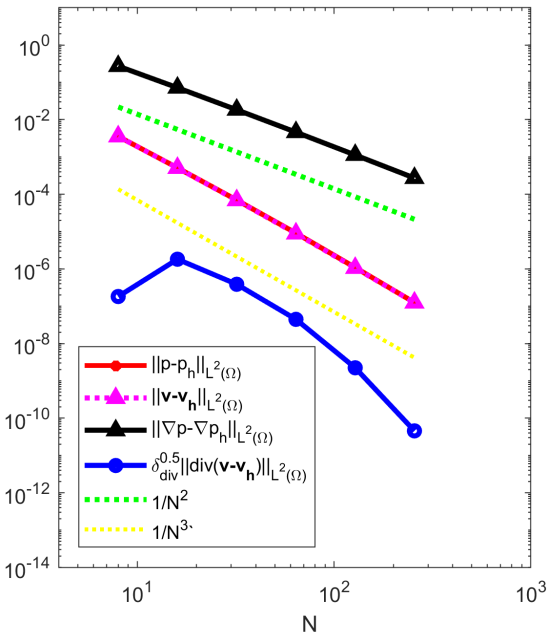
In Figure 4.3 we depict the errors obtained on implementing our Present Method in a sequence of uniformly refined meshes obtained by increasing the value of N. The first two plots correspond to the results obtained by using $\mathcal{P}_1$ elements for both variables, p and $\boldsymbol{v}$, with $\varepsilon = 10^{-3}$ (Figure 4.3a) and $\varepsilon = 10^{-5}$ (Figure 4.3b) . We observe that all the errors tend to zero with a ratio which is consistent with the results of Section 4.2. The same comments are applicable to the cases depicted in the Figures 4.3c and 4.3d, where quadratic $\mathcal{P}_2$ elements are considered for both variables.



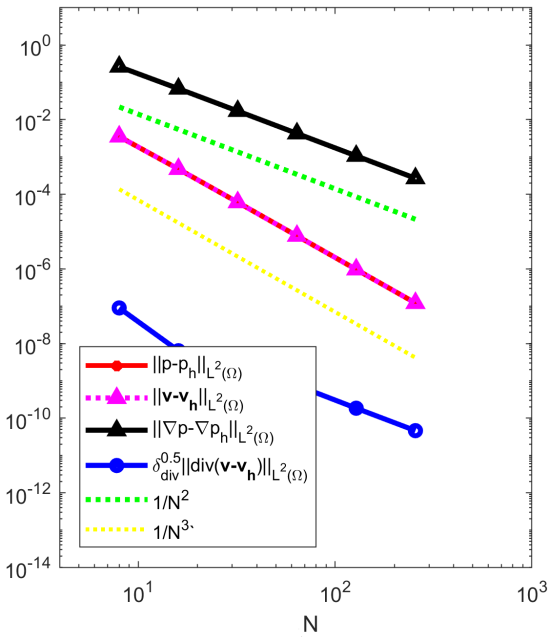
(a) $\mathcal{P}_1\mathcal{P}_1$ Convergence study $\varepsilon = 10^{-3}$.



(b) $\mathcal{P}_1\mathcal{P}_1$ Convergence study $\varepsilon = 10^{-5}$.

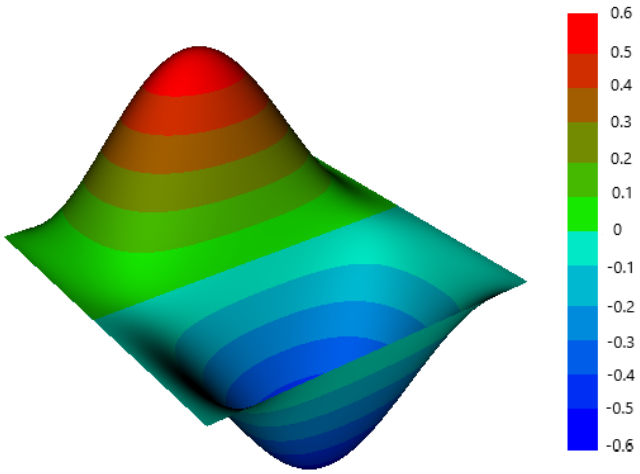


(c) $\mathcal{P}_2\mathcal{P}_2$ Convergence study $\varepsilon = 10^{-3}$.

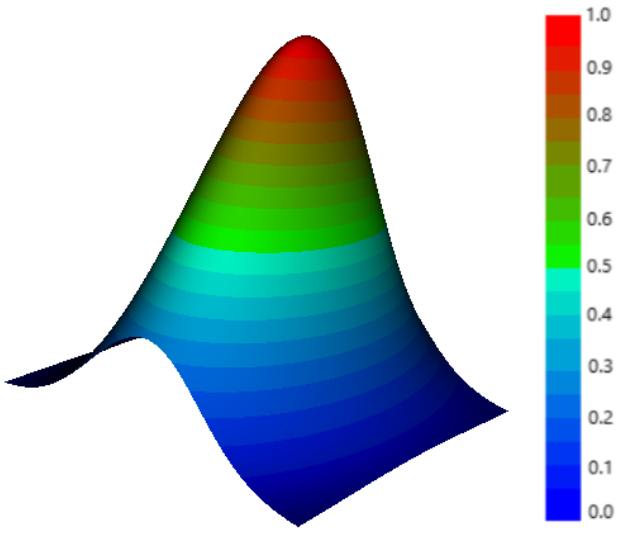


(d) $\mathcal{P}_2\mathcal{P}_2$ Convergence study $\varepsilon = 10^{-5}$.

Fig: 4.1 Convergence test A for the Present Method, $\mu = 0$

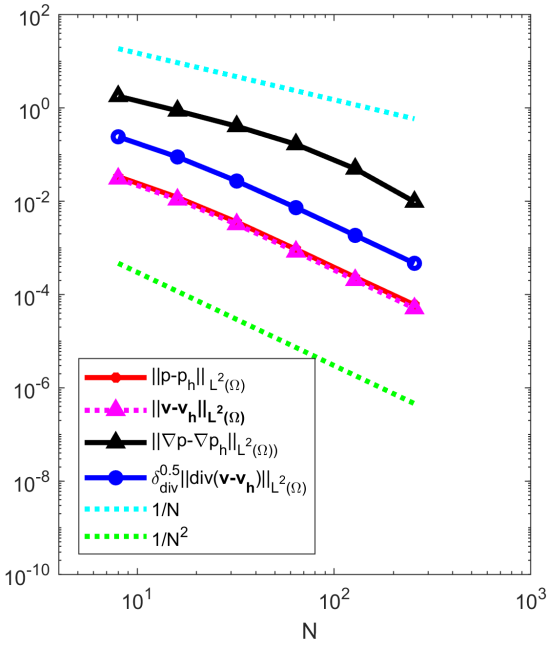


(a) Exact solution for test C.

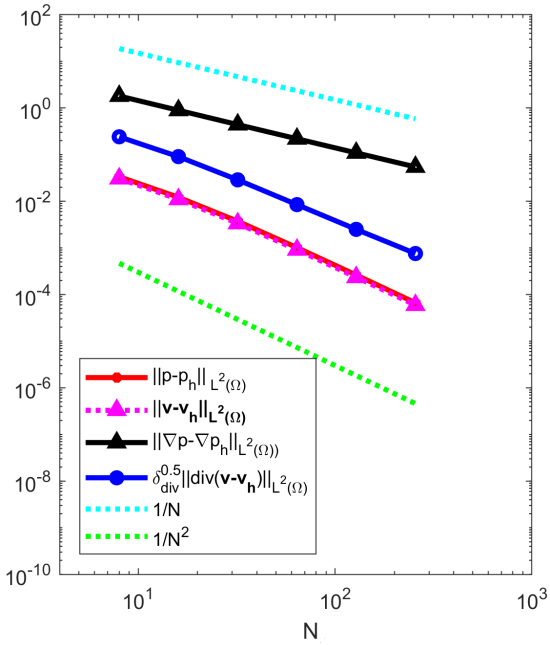


(b) Exact solution for test D.

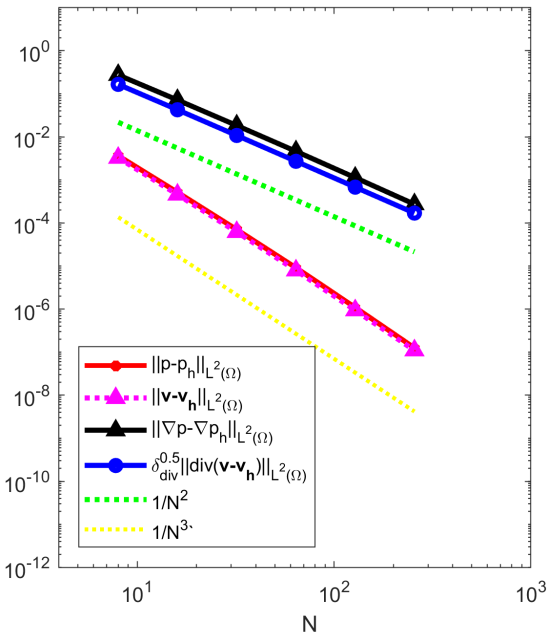
Fig: 4.2 Exact solutions for tests C and D



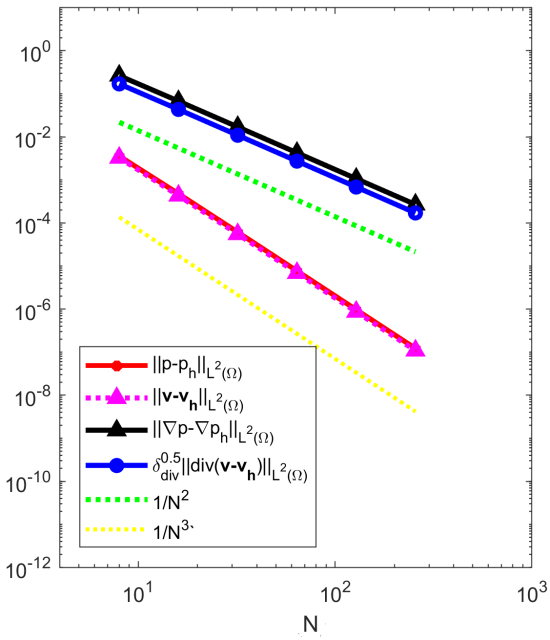
(a) $\mathcal{P}_1\mathcal{P}_1$ Convergence study $\varepsilon = 10^{-3}$.



(b) $\mathcal{P}_1\mathcal{P}_1$ Convergence study $\varepsilon = 10^{-5}$.



(c) $\mathcal{P}_2\mathcal{P}_2$ Convergence study $\varepsilon = 10^{-3}$.



(d) $\mathcal{P}_2\mathcal{P}_2$ Convergence study $\varepsilon = 10^{-5}$.

Fig: 4.3 Convergence studies for the Present Method: Test C, $\mu = 0$

4.4.3 Testing for Convergence using Test D: Method and results

Next, as both of our earlier tests have homogenous Dirichlet conditions and no reaction term, we test the Present Method with a Gaussian function which has non-homogenous Dirichlet conditions and with two different values of the reaction term $\mu = 0$ and $\mu = 1$. We again consider $\Omega = (0,1)^2$, let $\boldsymbol{\alpha} = [1,0]^T$ and test with a value of $\varepsilon = 10^{-5}$. Both f and the boundary conditions are chosen such that the solution of (2.5) is given by

$$p(x,y) = \exp\left(-\frac{(x-0.5)^2}{0.2} - \frac{3(y-0.5)^2}{0.2} \right).$$

The exact solution is shown in Figure 4.2b, where we can see that it does not have a Dirichlet value of zero on any side and there are different exponential functions on adjacent sides.
In this test

$$f(x,y) = -\frac{\varepsilon}{0.2}\left(\frac{4}{0.2}(x-0.5)^2 - 8 + \frac{36}{0.2}(y-0.5)^2\right)p - \frac{2}{0.2}(x-0.5)p\alpha_1 - \frac{6}{0.2}(y-0.5)p\alpha_2 + \mu p\,,$$

$$p_{\Gamma_1} = p_{\Gamma_3} = \exp\left(\frac{-(x^2-x+1)}{0.2}\right) \text{ and } p_{\Gamma_2} = p_{\Gamma_4} = \exp\left(\frac{-(3y^2-3y+1)}{0.2}\right).$$

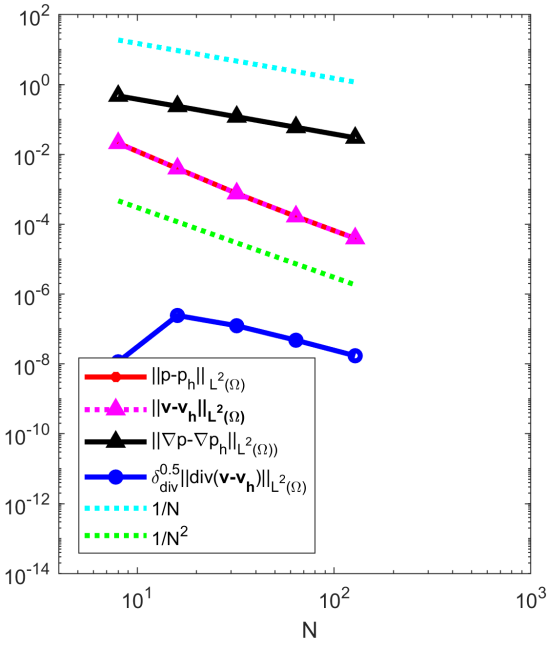
For this Gaussian function, our Present Method calculates both the primary and secondary variables very accurately and is able to show good convergence with small values of diffusion. Therefore, it is only necessary to examine $\varepsilon = 10^{-5}$ and the results are depicted in Figure 4.4. The errors converge with the correct orders without the reaction term. Also with the reaction term, $\mu = 1$, similar results for convergence are obtained, with a little more variation for the $\mathcal{P}_2\mathcal{P}_2$ case with $\varepsilon = 10^{-5}$.

4.4.4 A three-dimensional numerical convergence test: Test E

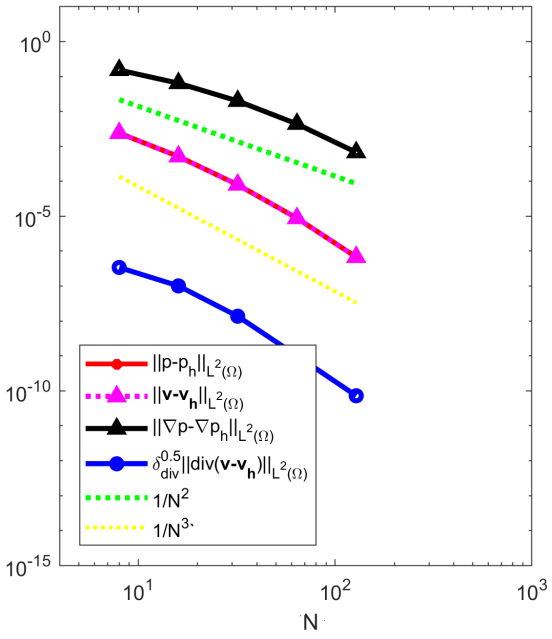
As the analysis holds for 3D configurations we now carry out a suitable test. We consider $\Omega = (0,1)^3$, $\varepsilon = 10^{-3}$, $\mu = 0$, $\boldsymbol{\alpha} = [1,2,1]^T$, and f is chosen such that the exact solution is given by

$$u(x,y,z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)\,. \tag{4.38}$$

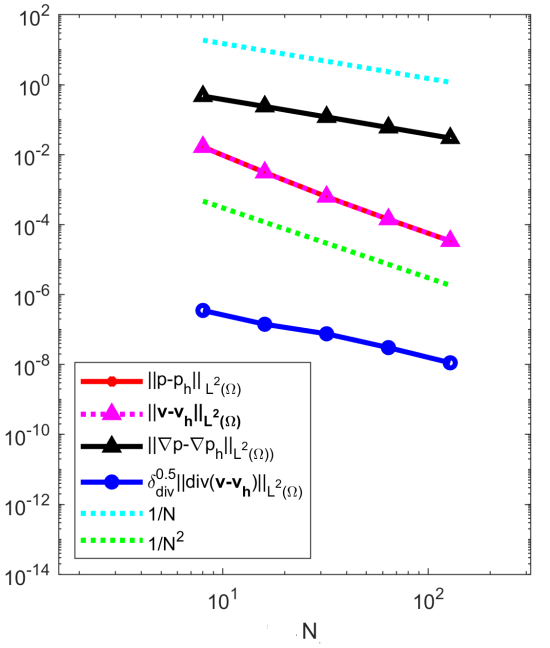
The domain is partitioned by dividing each side of the unit cube into N segments of equal length. This generates a structured mesh of each face of the unit cube, which is then propagated inside the domain (for details, see the Freefem++ documentation, or [Hec12]). We have measured the errors in the different norms, and the results are



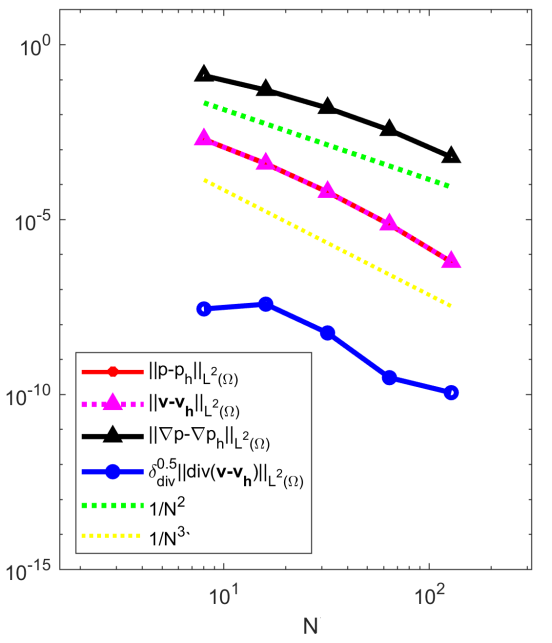
(a) $\mathcal{P}_1\mathcal{P}_1$ Convergence study $\varepsilon = 10^{-5},\ \mu = 0.$



(b) $\mathcal{P}_2\mathcal{P}_2$ Convergence study $\varepsilon = 10^{-5},\ \mu = 0.$



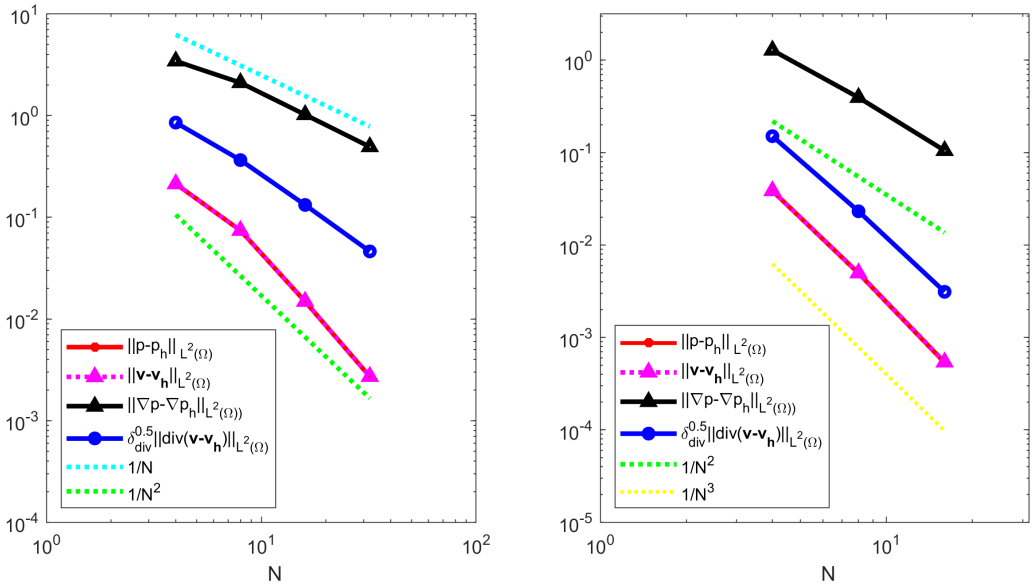
(c) $\mathcal{P}_1\mathcal{P}_1$ Convergence study $\varepsilon = 10^{-5},\ \mu = 1$



(d) $\mathcal{P}_2\mathcal{P}_2$ Convergence study $\varepsilon = 10^{-5},\ \mu = 1.$

Fig: 4.4 Convergence studies for the Present Method:Test D with differing reaction terms and non-homogenous Dirichlet conditions

depicted in Figure 4.5, where we can see that they have orders of convergence that are consistant with the theoretical results.



(a) $3D - \mathcal{P}_1\mathcal{P}_1$ Convergence study $\varepsilon = 10^{-3}$. (b) $3D - \mathcal{P}_2\mathcal{P}_2$ Convergence study $\varepsilon = 10^{-3}$.

Fig: 4.5 Three-dimensional testing of convergence with $\varepsilon = 10^{-3}$: Test E

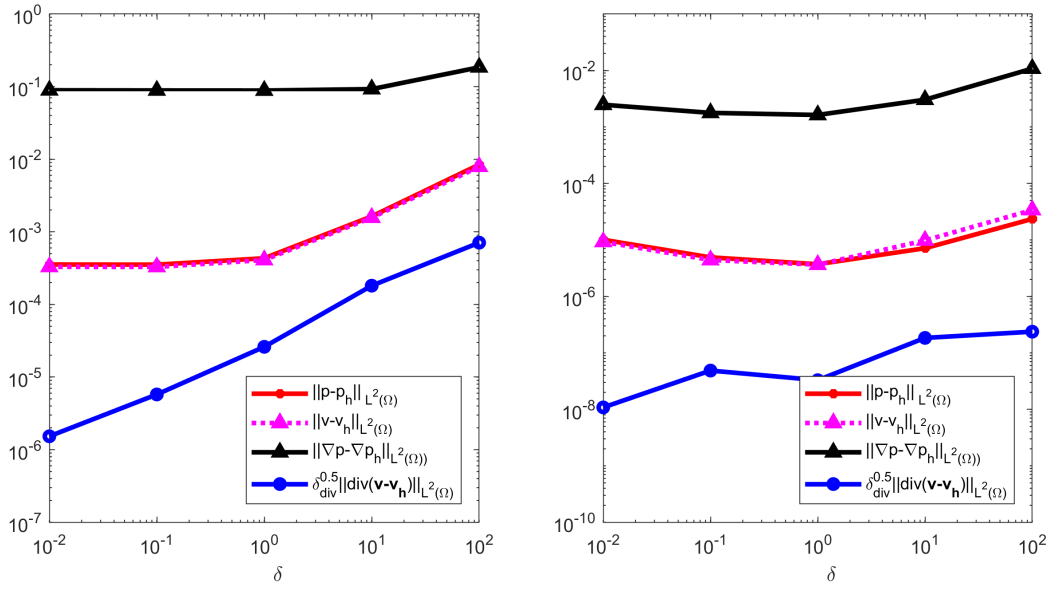
4.4.5 Testing possible values of δ parameter in the div-div term

In order to justify choosing a value of 1 for our stabilisation parameter δ, we carry out the following experiment. We fix a mesh, of the type depicted in Figure 2.2 with $N = 2^6$, $\varepsilon = 10^{-3}$, and compute the errors for the method using a range of values for δ, spanning from 10^{-2} to 10^2. The results are depicted in Figure 4.6. For this smooth solution all the errors, except for the one associated to the divergence of $\boldsymbol{v}$ (which is multiplied by $\delta^{\frac{1}{2}}$), show a fairly robust behavior with respect to δ in this range. It should be noted that the errors do deteriorate for more extreme choices of $\varepsilon = 10^{-3}$. However, the deciding factor is the advection skew to the mesh test (see Section 2.4.2) depicted in Figures 4.7 and 4.8. With the cross-section at $y = 0.5$, both $\mathcal{P}_1\mathcal{P}_1$ and $\mathcal{P}_2\mathcal{P}_2$ show clearly that, when diffusion is small, $\delta = 1$ is the best choice. The same applies to the cross-section at $x = 0.7$, where too small a value for δ leads to instablity in the outflow layer and too large a value results in a very diffuse outflow layer.

The results of Figures 4.9 and 4.10 also confirm our conclusions about the optimum value of δ. As $\delta \to 0$, the solution is only controlled by the part of the sum that does

not include the divergence of $\boldsymbol{w}$. This an effect on performance, as even if existence and uniqueness of the solution can be proven, the performance of the method deteriorates as $\delta \to 0$. Alternatively, when $\delta \to \infty$ we are imposing $\nabla \cdot \boldsymbol{w_h} + \mu q_h = f$ strongly.

Thus a value of $\delta = 1$ is justified, although no further fine tuning has been attempted.

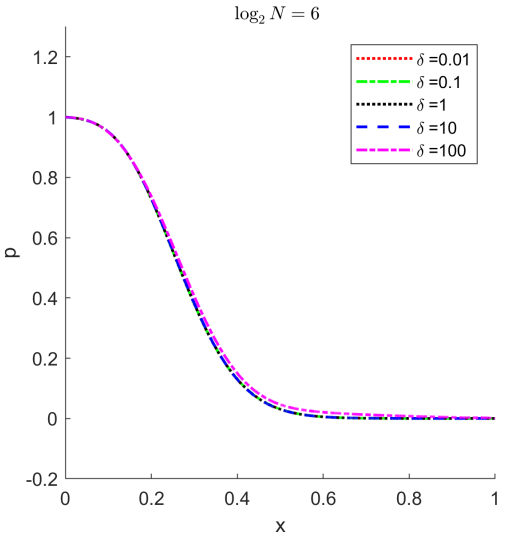


(a) Our new method using $\mathcal{P}_1\,\mathcal{P}_1$ elements. (b) Our new method using $\mathcal{P}_2\,\mathcal{P}_2$ elements.

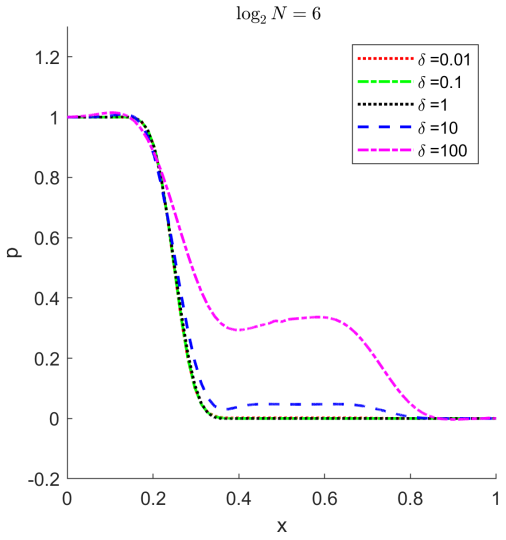
Fig: 4.6 Errors for the Present Method for $\varepsilon = 10^{-3}$, and different values for δ.

4.5 Conclusions

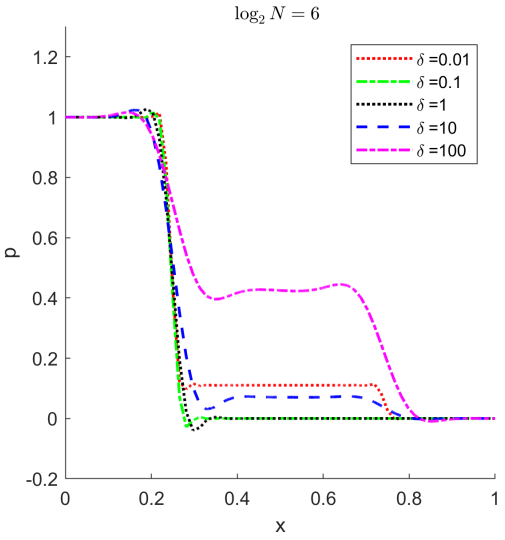
The analysis showing existence and uniqueness of the solution was not avaliable for [MK08]. Our new method was successfully analysed and found to converge as expected in numerical tests for both 2-D and 3-D with values of diffusion ranging from 1 to 10^{-5}. A test with non-homogenous Dirichlet boundary conditions and non-zero values of the reaction term was also carried out successfully. Different constant values of convective flux $\boldsymbol{\alpha}$ were used and a variable flux was also used successfully in the tests. The sensitivity of the results with respect to δ was tested and $\delta = 1$ proved an optimal choice.



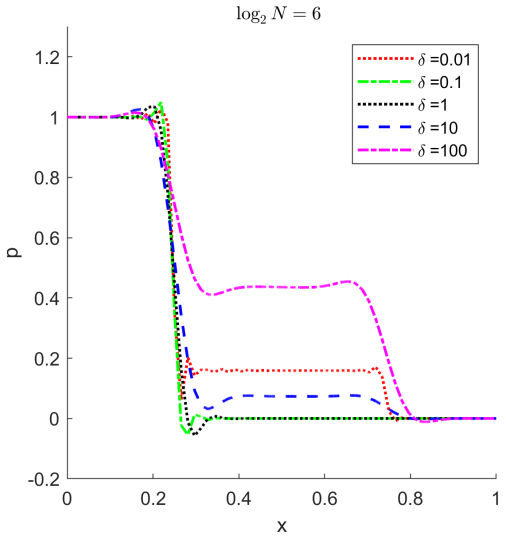
(a) BPY $\mathcal{P}_1\mathcal{P}_1$ variation with δ, $\varepsilon = 10^{-2}$



(b) BPY $\mathcal{P}_1\mathcal{P}_1$ variation with δ, $\varepsilon = 10^{-3}$

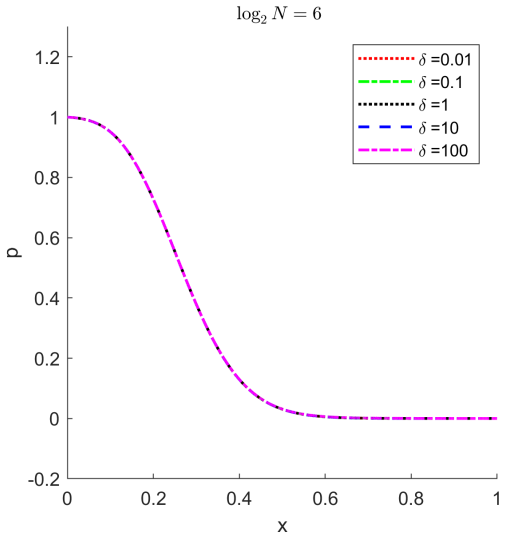


(c) BPY $\mathcal{P}_1\mathcal{P}_1$ variation with δ, $\varepsilon = 10^{-4}$

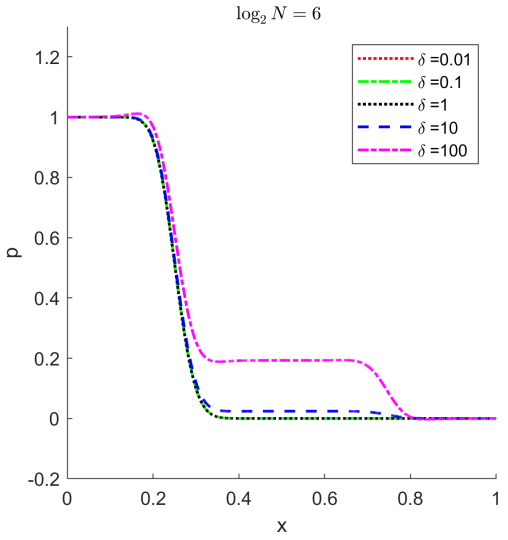


(d) BPY $\mathcal{P}_1\mathcal{P}_1$ variation with δ, $\varepsilon = 10^{-5}$

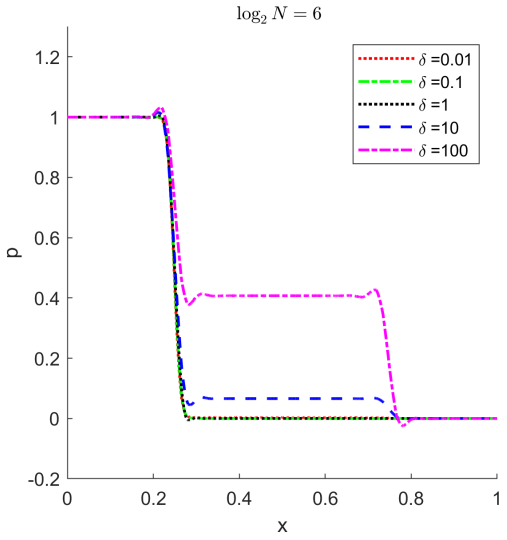
Fig: 4.7 Effect of δ with $y = 0.5$ for $\mathcal{P}_1\mathcal{P}_1$ elements



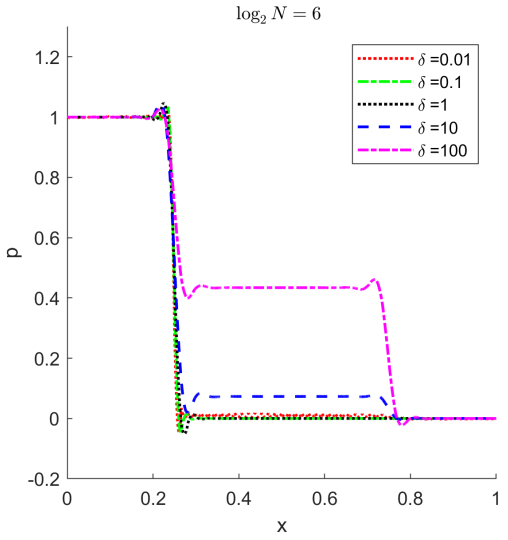
(a) BPY $\mathcal{P}_2\mathcal{P}_2$ variation with δ, $\varepsilon = 10^{-2}$



(b) BPY $\mathcal{P}_2\mathcal{P}_2$ variation with δ, $\varepsilon = 10^{-3}$

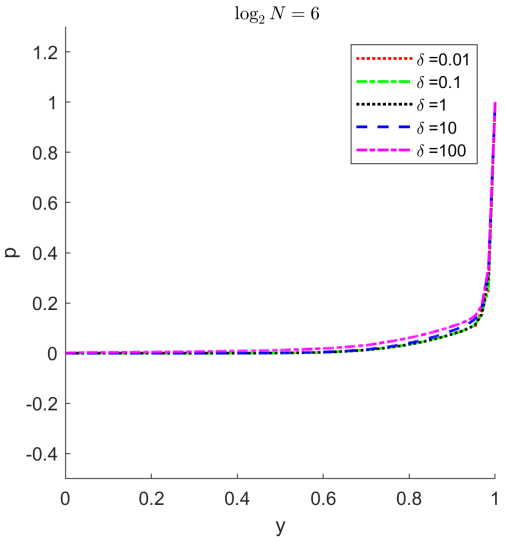


(c) BPY $\mathcal{P}_2\mathcal{P}_2$ variation with δ, $\varepsilon = 10^{-4}$

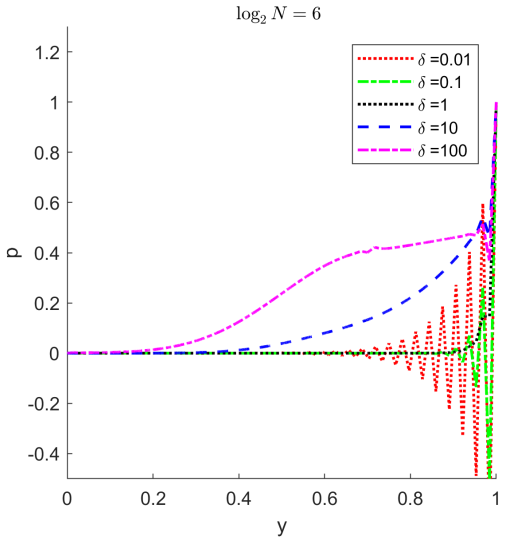


(d) BPY $\mathcal{P}_2\mathcal{P}_2$ variation with δ, $\varepsilon = 10^{-5}$

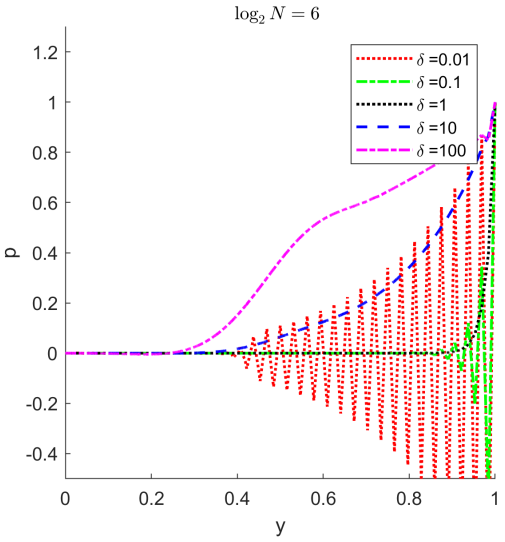
Fig: 4.8 Effect of δ with cross-section at $y = 0.5$ for $\mathcal{P}_2\mathcal{P}_2$ elements



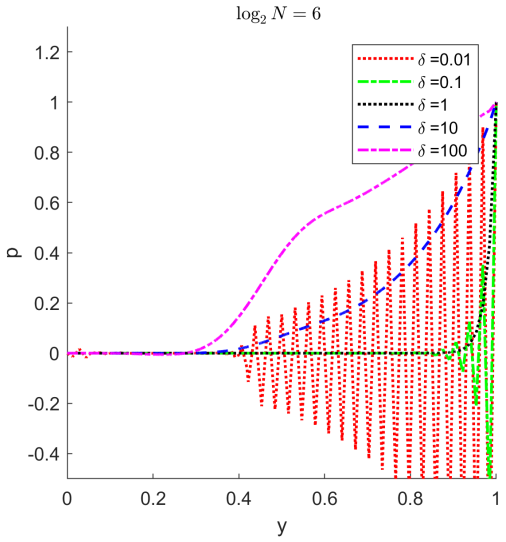
(a) BPY $\mathcal{P}_1\mathcal{P}_1$ variation with δ, $\varepsilon = 10^{-2}$



(b) BPY $\mathcal{P}_1\mathcal{P}_1$ variation with δ, $\varepsilon = 10^{-3}$

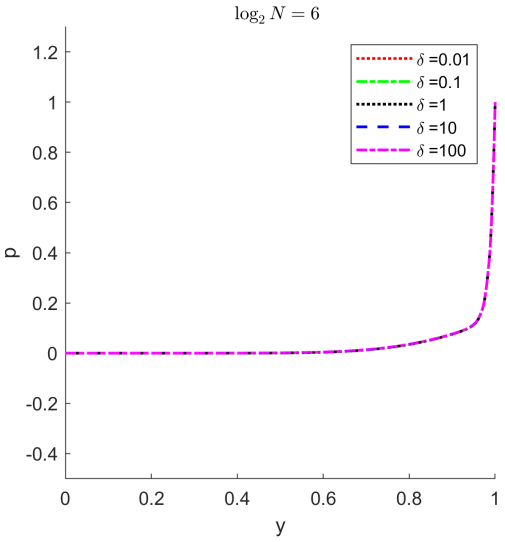


(c) BPY $\mathcal{P}_1\mathcal{P}_1$ variation with δ, $\varepsilon = 10^{-4}$

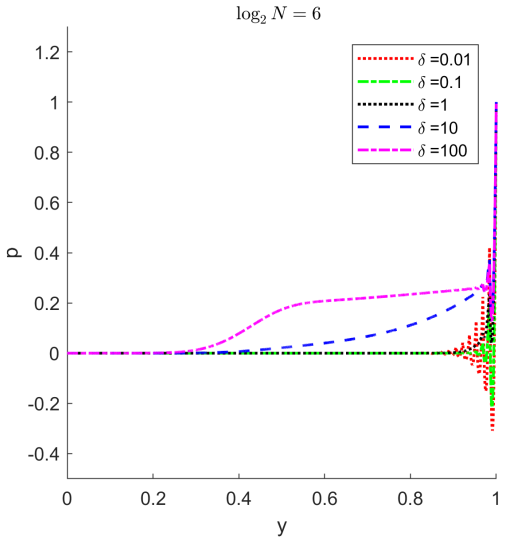


(d) BPY $\mathcal{P}_1\mathcal{P}_1$ variation with δ, $\varepsilon = 10^{-5}$

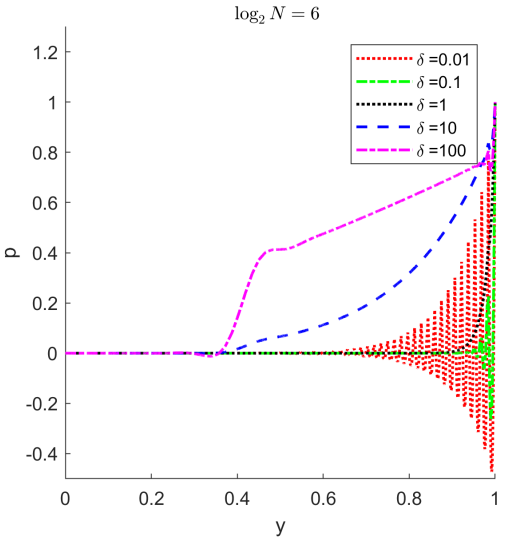
Fig: 4.9 Effect of δ with cross-section at $x = 0.7$ for $\mathcal{P}_1\mathcal{P}_1$ elements



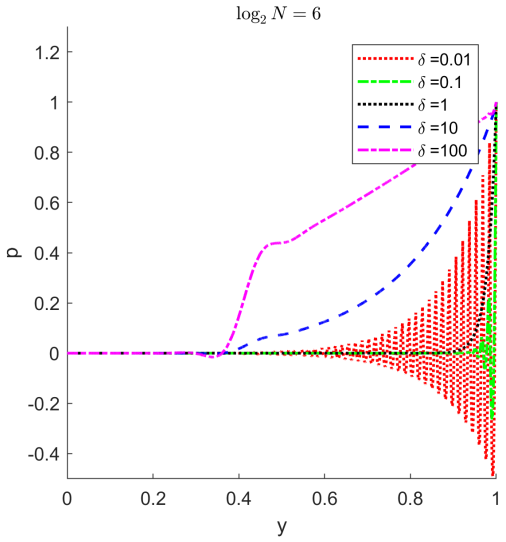
(a) BPY $\mathcal{P}_2\mathcal{P}_2$ variation with δ, $\varepsilon = 10^{-2}$



(b) BPY $\mathcal{P}_2\mathcal{P}_2$ variation with δ, $\varepsilon = 10^{-3}$



(c) BPY $\mathcal{P}_2\mathcal{P}_2$ variation with δ, $\varepsilon = 10^{-4}$



(d) BPY $\mathcal{P}_2\mathcal{P}_2$ variation with δ, $\varepsilon = 10^{-5}$

Fig: 4.10 Effect of δ with cross-section at $x = 0.7$ for $\mathcal{P}_2\mathcal{P}_2$ elements

Chapter 5

Comparative Studies

This chapter presents the results of a comparative study of our new method (termed 'the Present Method') from our published paper [BPY18]. It uses two challenging tests which involve layers: Test B, the 'Advection skew to the mesh test' described previously in Section 2.4.2 and the Hemker test [Hem96] described in Section 5.2. The mixed formulations included are the stabilised formulation of Masud and Kwack [MK08], the FOSLS method of Chen et al. [CFLQ14] and our new method. These are benchmarked against the SUPG method, outlined in Section 2.2.1, for the second-order formulation of the CDR equation.

5.1 The resuts of Test B: Advection skew to the mesh test

We start by showing elevations of the different solutions in Figure 5.1 and then cross-sections of the numerical solution given by the Present Method using linear elements in Figure 5.2, with $\delta = 1$. For comparison, we also include cross-sections of the solution given by the SUPG method using $k = 1$ in the same mesh along with the MK method and the reference solution. In order to capture both the interior and outflow layers the cross-sections are obtained by taking separate cut-lines with values interpolated from the computed solution at $x = 0.7$ and $y = 0.5$. The cross-sections of the reference solution using these cut-lines are shown in Figures 2.5a and 2.5b respectively. In this comparative study, 10,000 equally distributed points are used along the cut lines. We observe that the MK method exhibits oscillations near the outflow layer and that these are not eliminated even with a further level ($N = 2^8$) of mesh refinement.
In Figure 5.3 we depict the same cross-sections using quadratic elements. We also include the results given by methods (2.39) and (2.40), since these are second order methods. The same comments as before are valid for this case with the MK method exhibiting oscillations. The weak imposition of outflow boundary conditions in FOSLS appears to

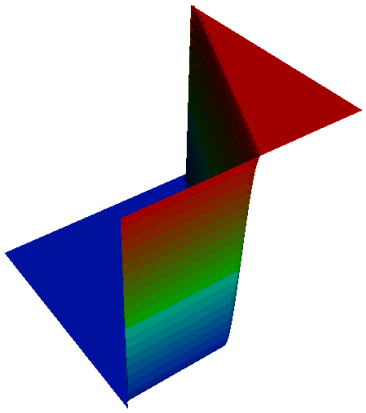
help suppress the oscillations present in the MK solution. However, this comes at the price of the FOSLS solutions completely missing the outflow boundary layer, unless the mesh is extremely refined. This can be observed in Figure 5.4a where we zoom in on the plots with all the solutions (except [MK08]) for $N = 2^7$. Here we observe that SUPG and the Present Method capture the outflow boundary layer while the FOSLS methods do not.

We continue by examining the over- and undershoots produced by all of the methods. These are computed as follows:

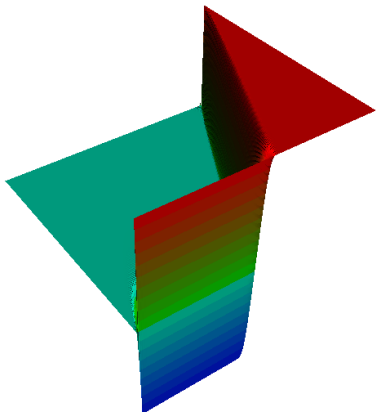
$$p_{max} = \max_{x \in \bar{\Omega}} p_h(x) - 1,$$
$$p_{min} = \min_{x \in \bar{\Omega}} p_h(x) \ .$$

For quadratic elements, we have approximated these values by using the maximum and minimum over the degrees of freedom. We observe that the present approach dramatically improves on the results of [MK08]. We also note briefly that some of the results given by the MK method lie outside the range of the plots, especially for small values of ε. The over- and undershoots given by the present method show a comparable behaviour to SUPG, with both outperforming the results given by both FOSLS methods.

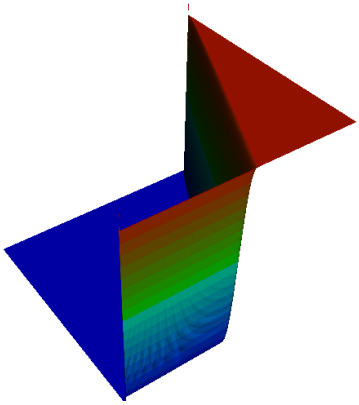
Next we compare the layer thickness of both the internal and outflow layers. In the graphs in Figure 5.6, the width of the interior layer is defined as the width of the interval taken for the value of the solution along $p(x, 0.5)$ to decrease from 0.9 to 0.1. Similarly, in Figure 5.6 the width of the outflow layer is defined as the width of the interval taken for the value of the solution along $p(0.7, y)$ to rise from 0.1 to 0.9. We observe that the present method produces sharper results than the ones given by [MK08], but it is out-performed by SUPG for linear elements and comparable for quadratic elements. The instabilities of the method in [MK08] led to us removing those results from the $\varepsilon = 10^{-5}$ graphs. It is worth mentioning that the increase of the interior layer width with increasing refinement in both FOSLS methods is due to the fact that the weak imposition of the outflow boundary conditions makes the method only capture the outflow layer if the mesh is refined enough. To illustrate this, in Figure 5.8 we plot elevations of the discrete solution given by both FOSLS methods with $\varepsilon = 10^{-3}$ and $N = 2^8$.



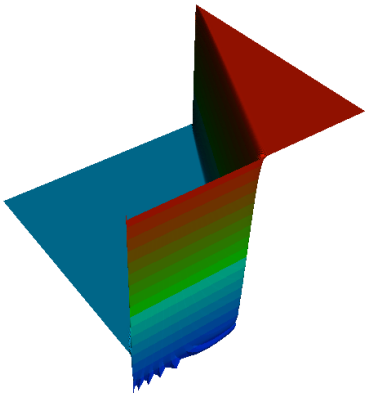
(a) Present method $\mathcal{P}_1\mathcal{P}_1$



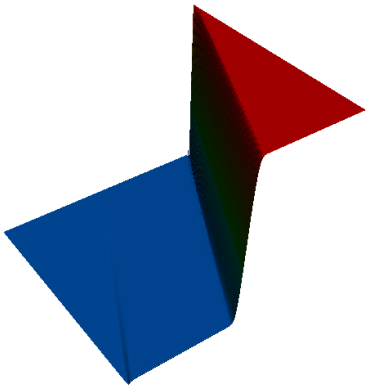
(b) MK method $\mathcal{P}_1\mathcal{P}_1$



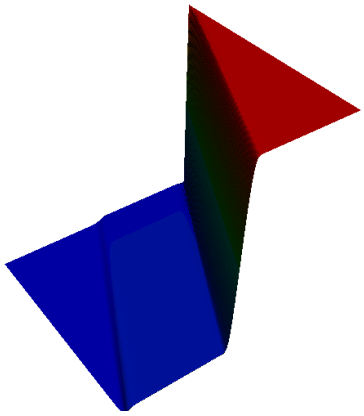
(c) Present method $\mathcal{P}_2\mathcal{P}_2$



(d) MK method $\mathcal{P}_2\mathcal{P}_2$

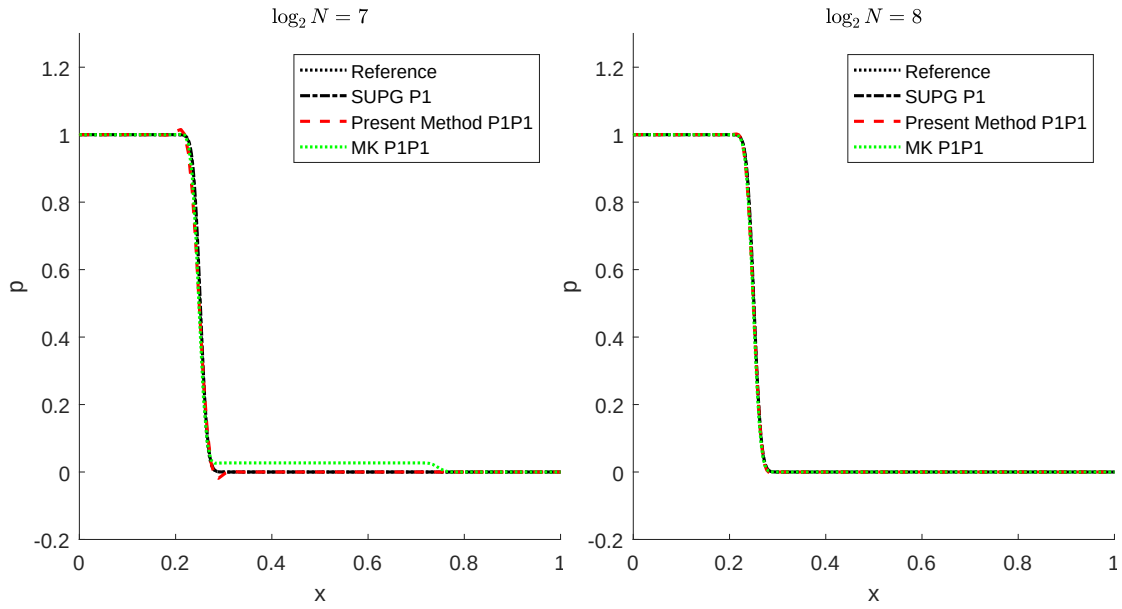


(e) FOSLS method $\mathcal{RT}_1 \times \mathcal{P}_1$

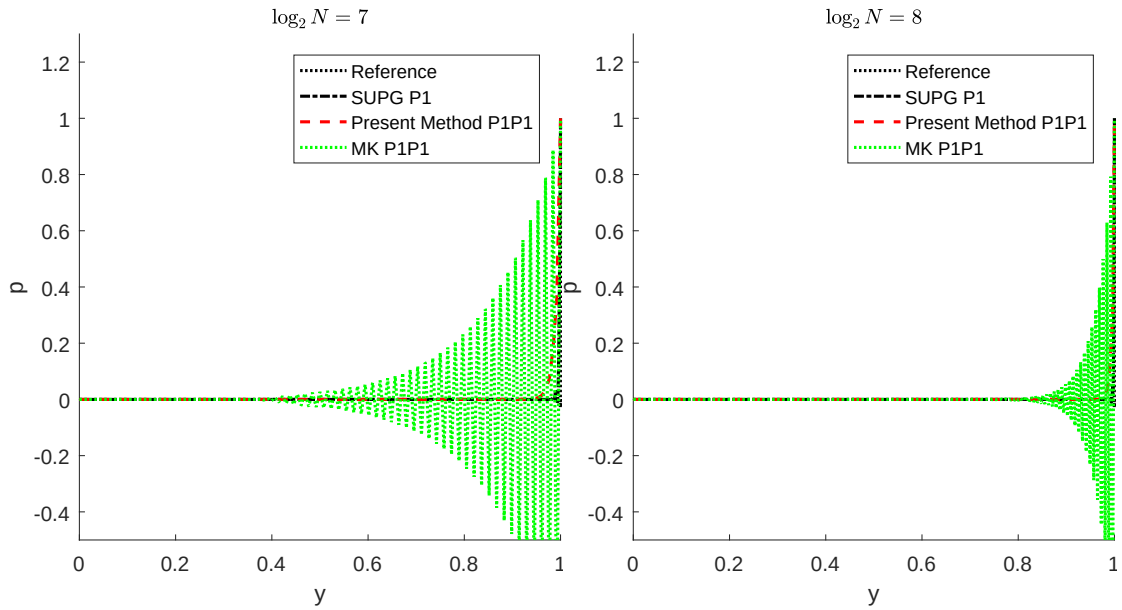


(f) FOSLSb method $\mathcal{RT}_1 \times \mathcal{P}_1$

Fig: 5.1 Elevations with quadratic elements, $N = 2^8$, $\varepsilon = 10^{-4}$.

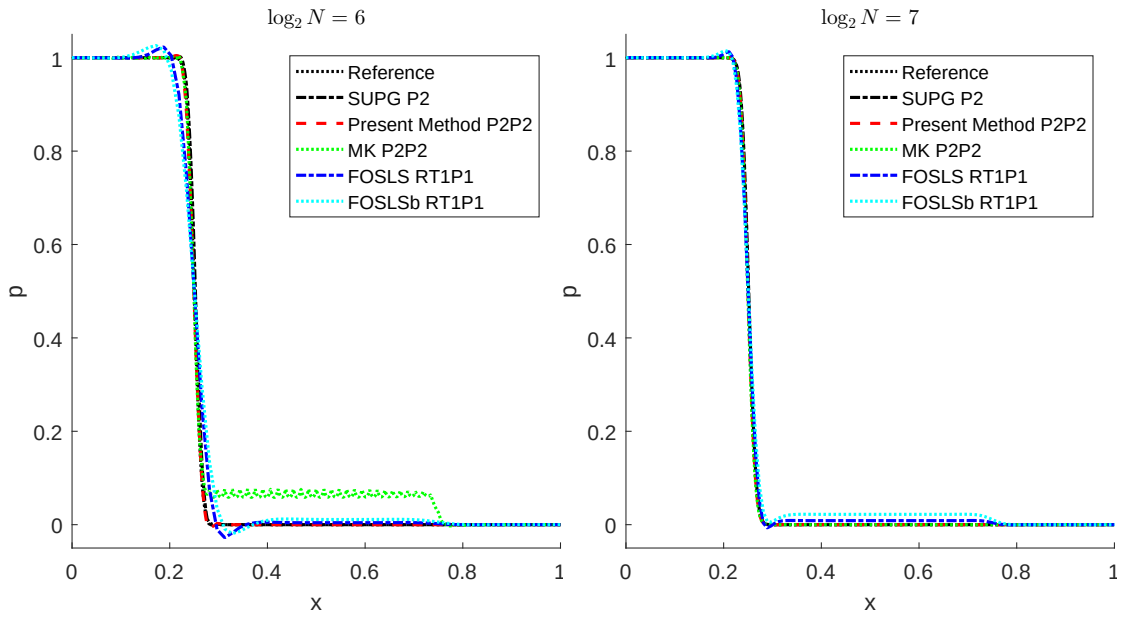


(a) cross-section at y =0.5, $\varepsilon = 10^{-4}$ (b) cross-section at y =0.5, $\varepsilon = 10^{-4}$



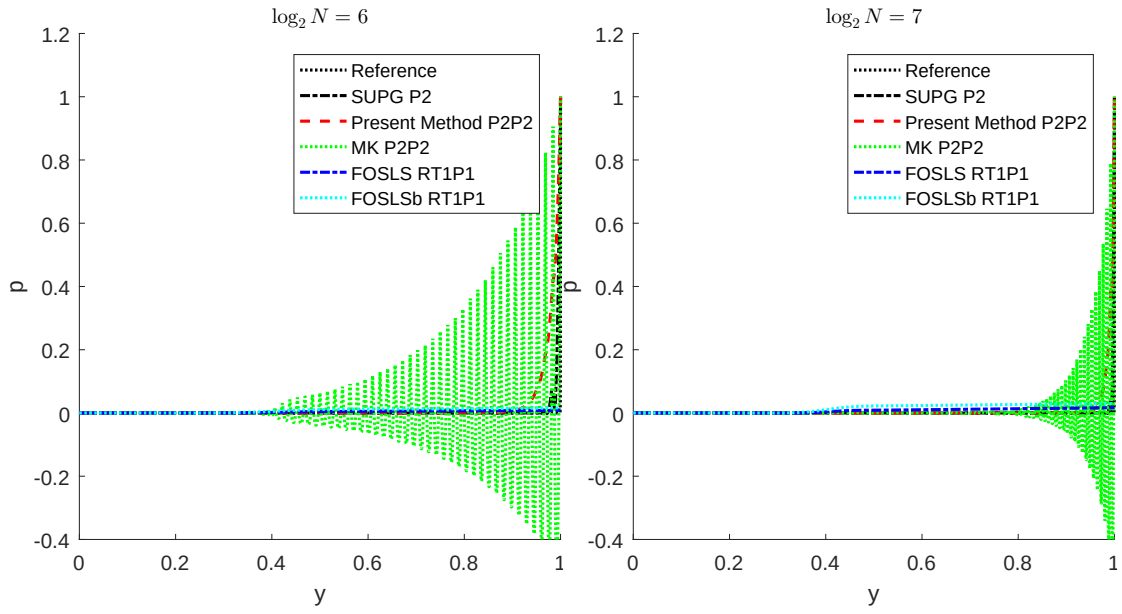
(c) cross-section at x=0.7, $\varepsilon = 10^{-4}$ (d) cross-section at x = 0.7, $\varepsilon = 10^{-4}$

Fig: 5.2 Cross-sections of the different methods considered using linear elements.



(a) cross-section at y =0.5, $\varepsilon = 10^{-4}$

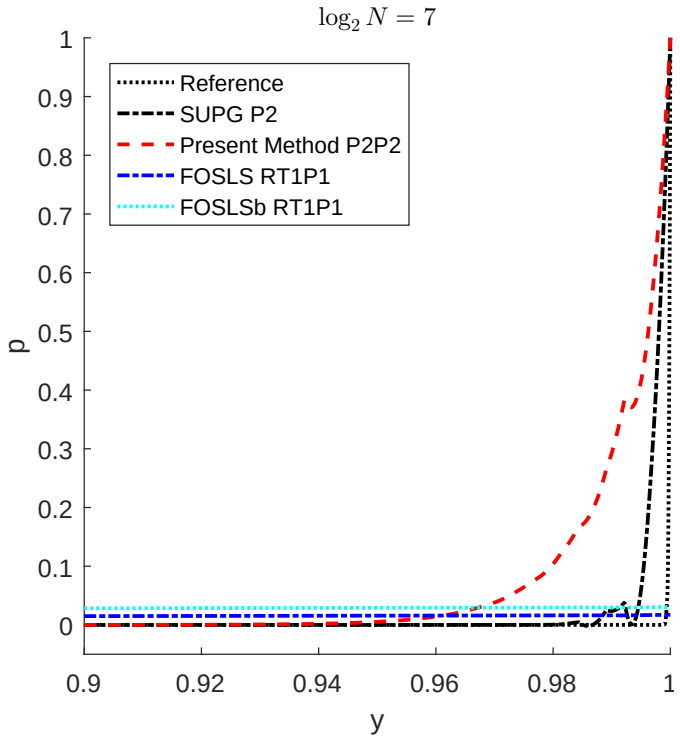
(b) cross-section at y =0.5, $\varepsilon = 10^{-4}$



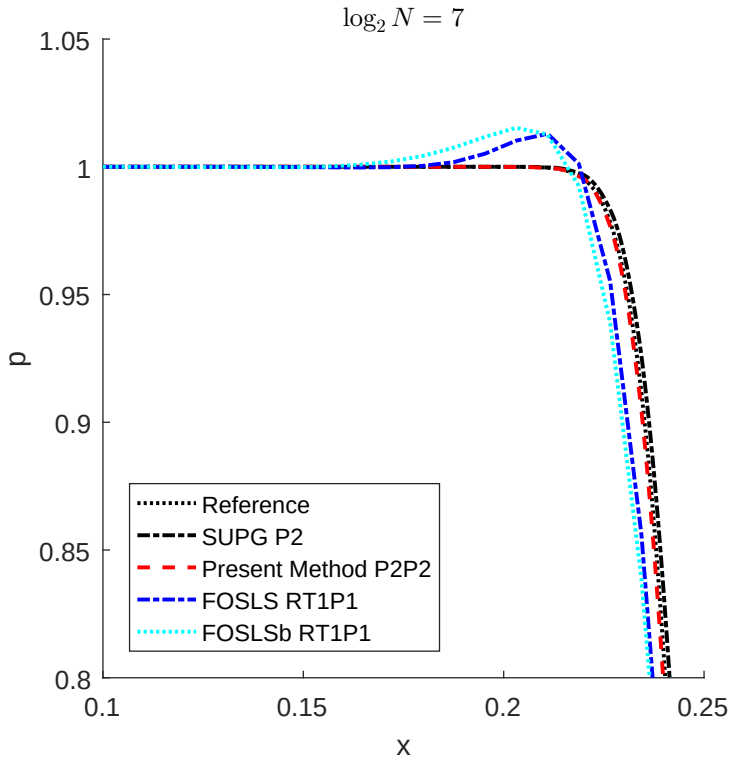
(c) cross-section at x=0.7, $\varepsilon = 10^{-4}$

(d) cross-section at x = 0.7, $\varepsilon = 10^{-4}$

Fig: 5.3 Cross-sections of the different methods considered using quadratic elements.

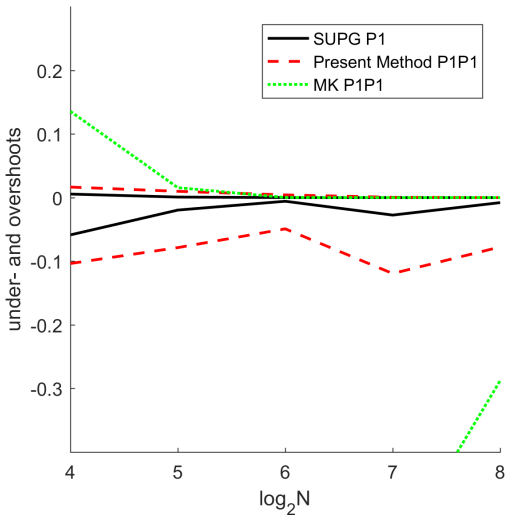


(a) cross-section at y =0.5, $\varepsilon = 10^{-4}$

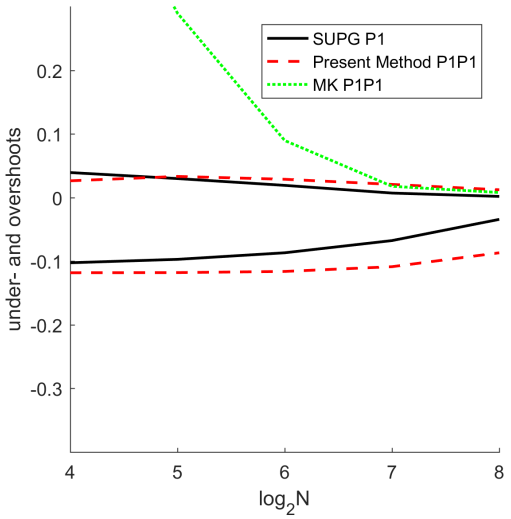


(b) cross-section at y =0.5, $\varepsilon = 10^{-4}$

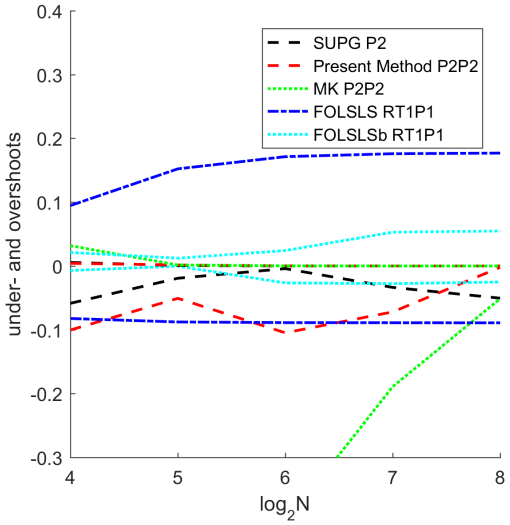
Fig: 5.4 Close-up of cross-sections of the different methods considered using quadratic elements.



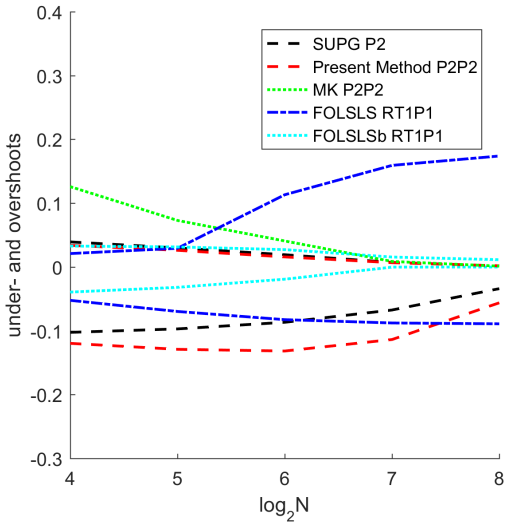
(a) Linear elements $\varepsilon = 10^{-3}$



(b) Linear elements $\varepsilon = 10^{-4}$

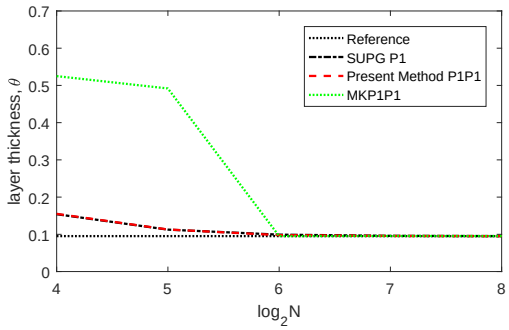


(c) Quadratic elements $\varepsilon = 10^{-3}$

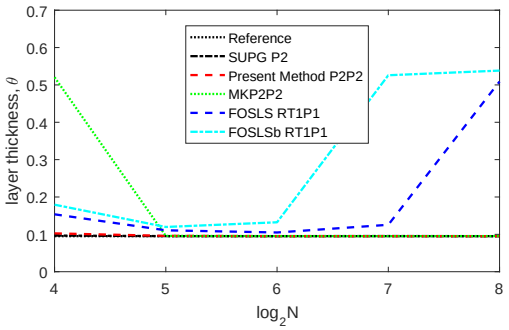


(d) Quadratic elements $\varepsilon = 10^{-4}$

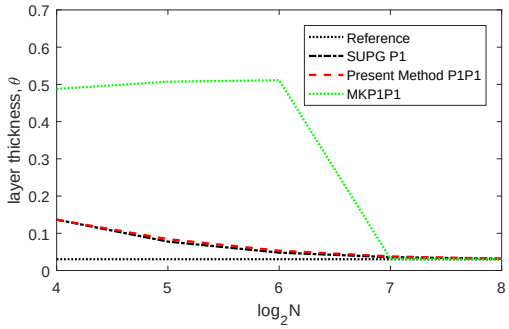
Fig: 5.5 Over- and undershoots for the different methods.



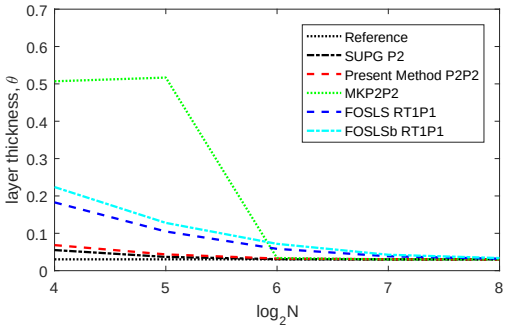
(a) Linear elements, $\varepsilon = 10^{-3}$



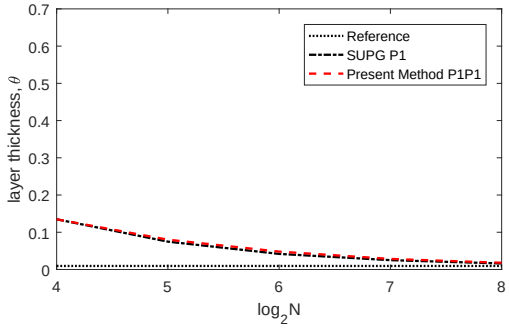
(b) Quadratic elements $\varepsilon = 10^{-3}$



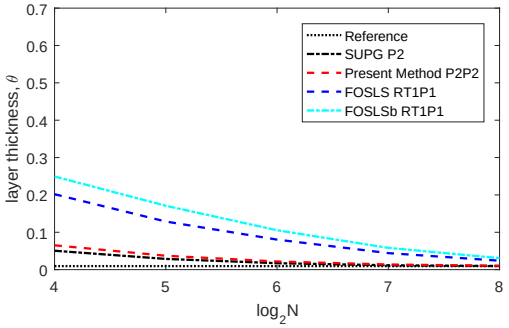
(c) Linear elements $\varepsilon = 10^{-4}$



(d) Quadratic elements, $\varepsilon = 10^{-4}$

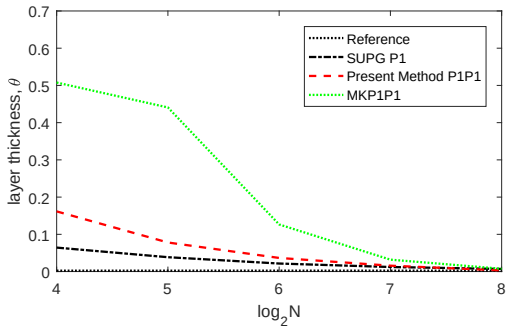


(e) Linear elements, $\varepsilon = 10^{-5}$

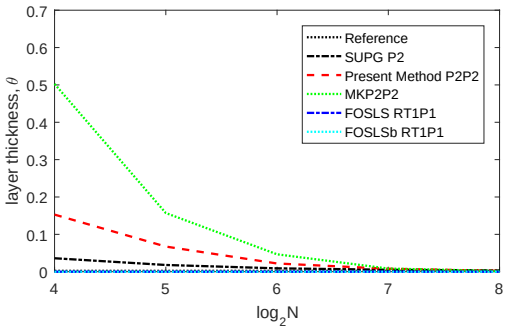


(f) Quadratic elements, $\varepsilon = 10^{-5}$

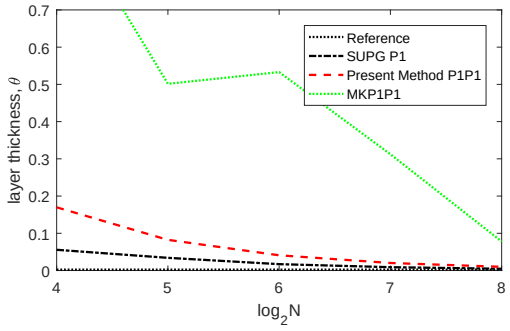
Fig: 5.6 Internal layer thickness, θ, for $0.1 < p(x, 0.5) < 0.9$ with refinement level.



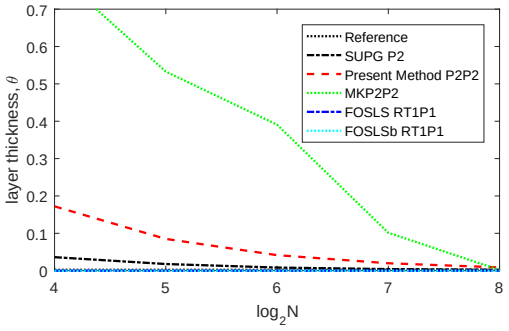
(a) Linear elements, $\varepsilon = 10^{-3}$



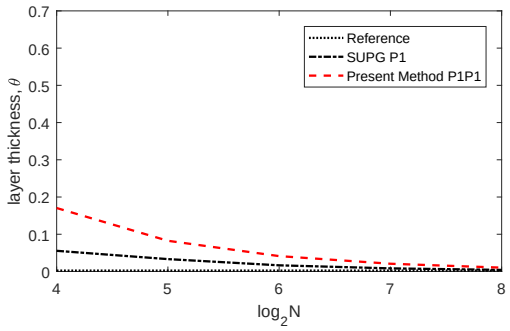
(b) Quadratic elements, $\varepsilon = 10^{-3}$



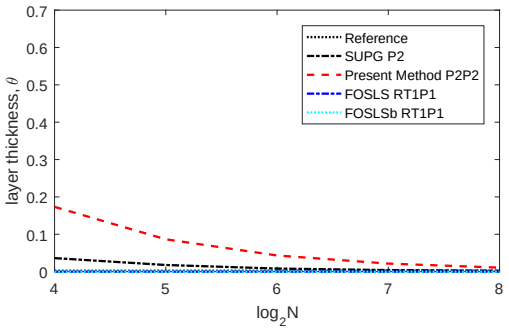
(c) Linear elements, $\varepsilon = 10^{-4}$



(d) Quadratic elements, $\varepsilon = 10^{-4}$

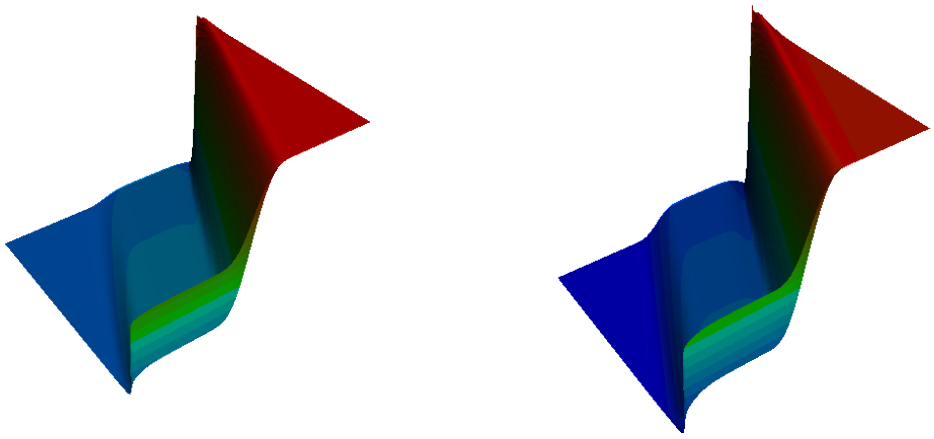


(e) Linear elements, $\varepsilon = 10^{-5}$



(f) Quadratic elements, $\varepsilon = 10^{-5}$

Fig: 5.7 Outflow layer thickness, θ, for $0.1 < p(0.7, y) < 0.9$ with refinement level.



(a) FOSLS method

(b) FOSLSb method

Fig: 5.8 Elevations for FOSLS methods, $N = 2^8$, $\varepsilon = 10^{-3}$.

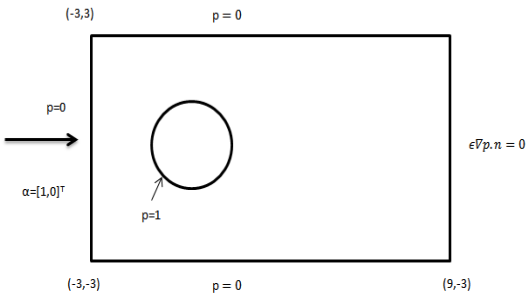
5.2 The Hemker problem

The domain and boundary conditions for the Hemker problem, similar to the one from [Hem96], is depicted on Figure 5.9a. This test is a simple 2-D model for a hot rod (the circle) with temperature of $T = 1$ in a room (the rectangle) with a wind blowing on the left-hand side and heat being convected across the room in the direction of the wind. A boundary layer will appear on the left-hand side of the circle, while the characteristic (interior) start from the top and bottom of the circle in the direction of the convection stretching out to the right-hand side.

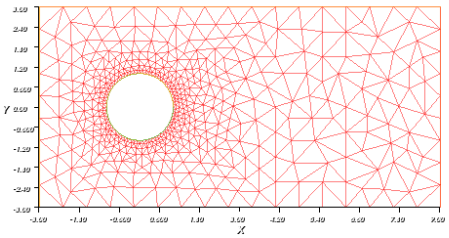
The meshes used were generated from an initial unstructured grid (Figure 5.9b). Successive refinements lead to meshes whose characteristics are detailed in Table 5.1. Details of the parameterisation of the mesh in FreeFem++ can be found in Appendix A.3.5 with comments on the implementation. When using linear elements we used meshes up to level 6 and with quadratic elements we used meshes up to level 5. For this problem we have not included a comparison with the MK method since several plots lie outside the scale of the plots shown. A reference solution for the Hemker problem with $\varepsilon = 10^{-4}$ is shown in Figure 5.10b for the SUPG method on a very fine mesh (level 6) with $\mathcal{P}_2$ triangular Lagrange elements. The reference solution can also be seen on Figures 5.11 and 5.12 and in more detail in the close-up in Figure 5.13.

Table 5.1 Details of Hemker meshes

level	No of Triangles	No of Vertices	SUPG $\mathcal{P}_2$ DOFs	Present DOFs $\mathcal{P}_2\mathcal{P}_2$	FOSLS DOFs $\mathcal{RT}_1 \times \mathcal{P}_1$	h_{min}
0	978	549	2076	6228	5559	0.098
1	3918	2079	8076	24228	21909	0.047
2	15522	8001	31524	94572	86091	0.023
3	61494	31227	123948	371844	339657	0.011
4	247542	124731	497004	1491012	1364361	0.0056
5	988588	496214	1981016	5943048	5442994	0.0026
6	3951688	1979624	7910816	–	–	0.0012

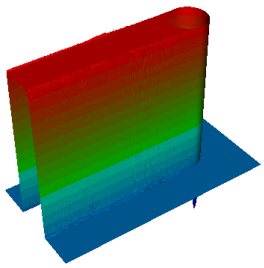


(a) Hemker test: geometry and boundary conditions

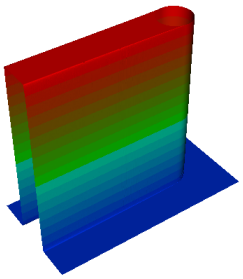


(b) mesh for Hemker test-level 0

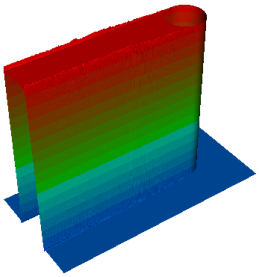
Fig: 5.9 Hemker test details and initial mesh.



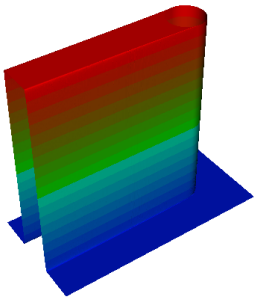
(a) SUPG $\mathcal{P}_1$ solution



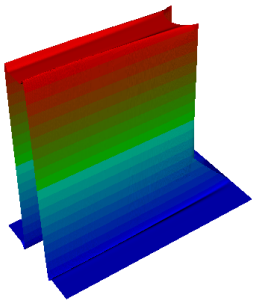
(b) SUPG Reference $\mathcal{P}_2$ solution



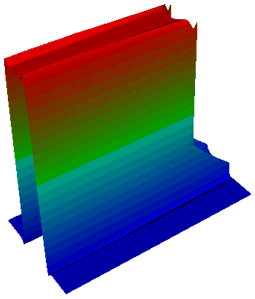
(c) Present Method $\mathcal{P}_1\mathcal{P}_1$ solution



(d) Present Method $\mathcal{P}_2\mathcal{P}_2$ solution

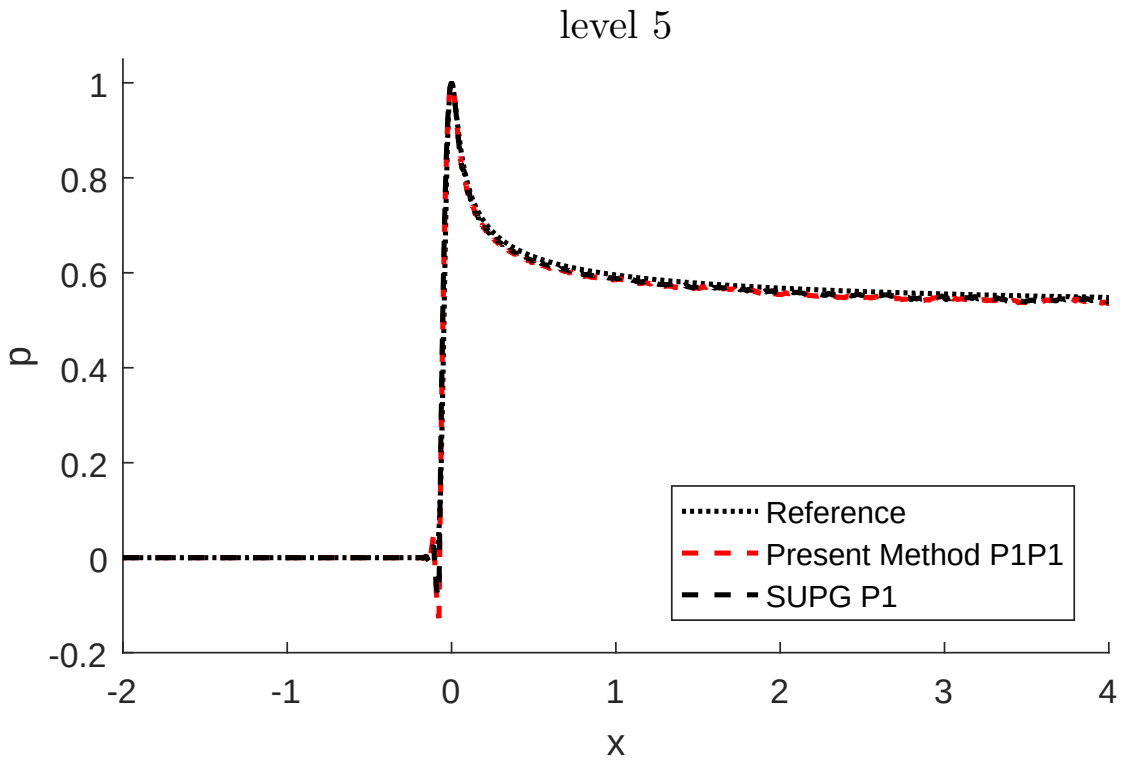


(e) FOSLS $\mathcal{RT}_1 \times \mathcal{P}_1$ solution

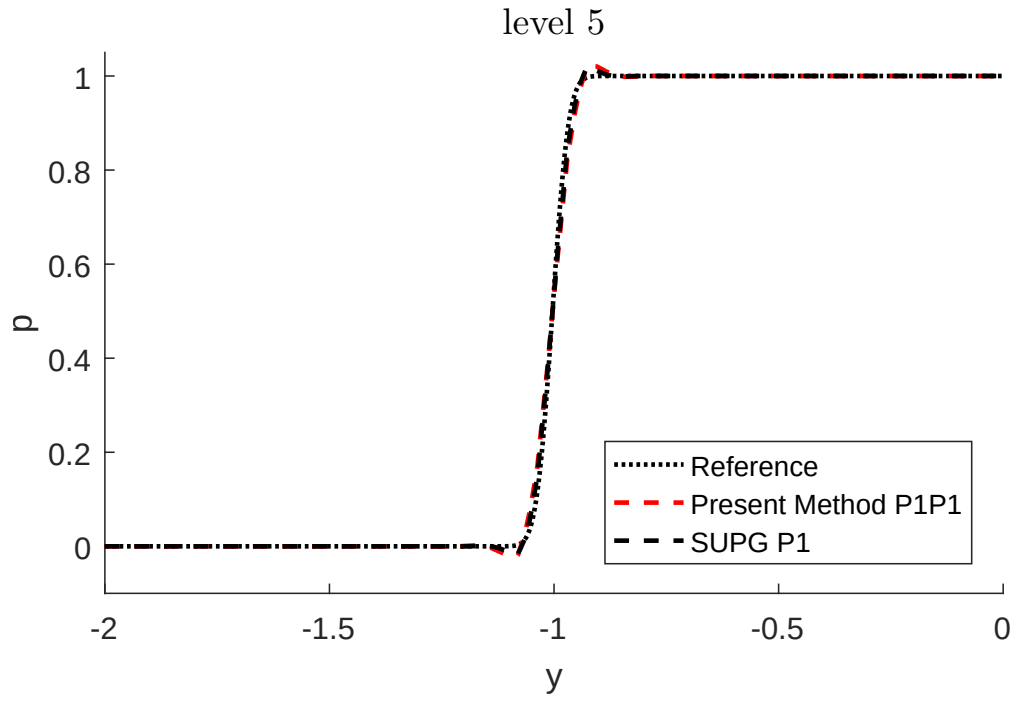


(f) FOSLSb $\mathcal{RT}_1 \times \mathcal{P}_1$ solution

Fig: 5.10 Elevation of the solutions for level 5, $\varepsilon = 10^{-4}$ for methods used in the Hemker Study.

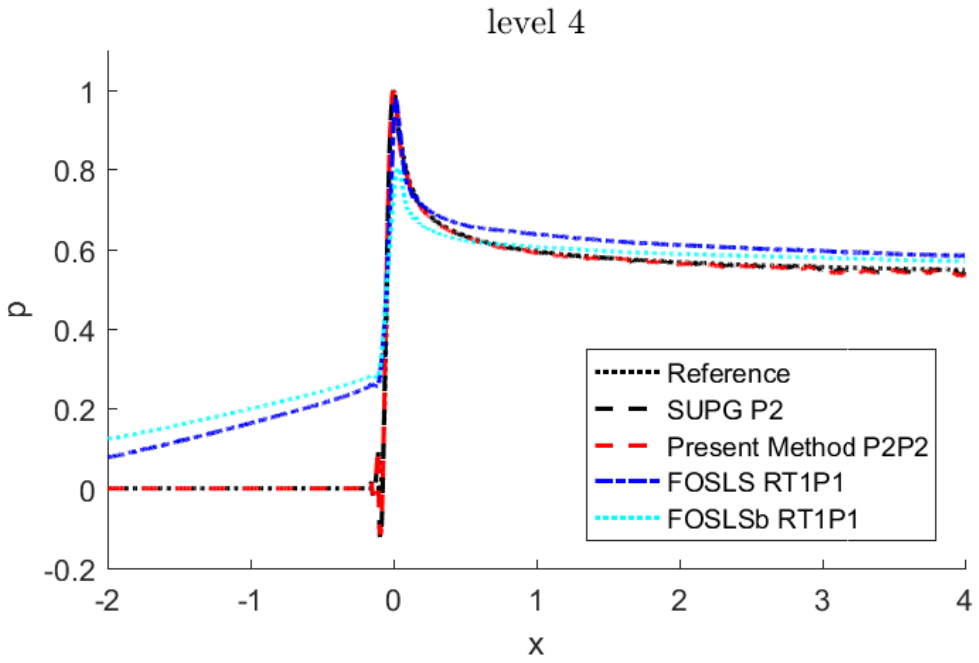


(a) x cross-section at $y = 1$

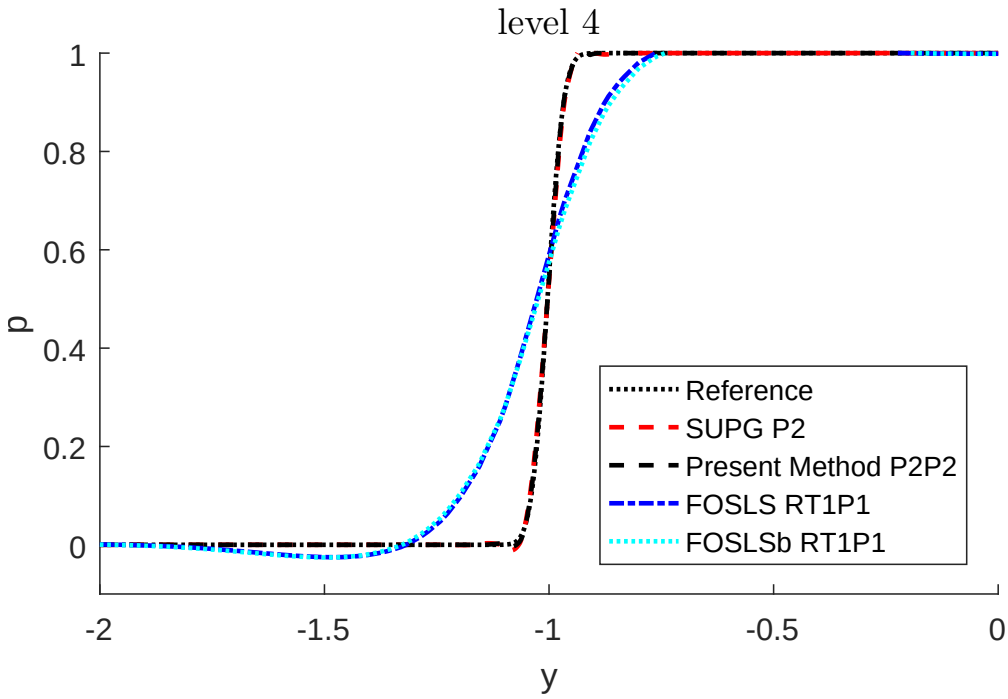


(b) y cross-section at $x = 4$

Fig: 5.11 Cross-sections using linear elements, level 5, $\varepsilon = 10^{-4}$ in the Hemker Study.

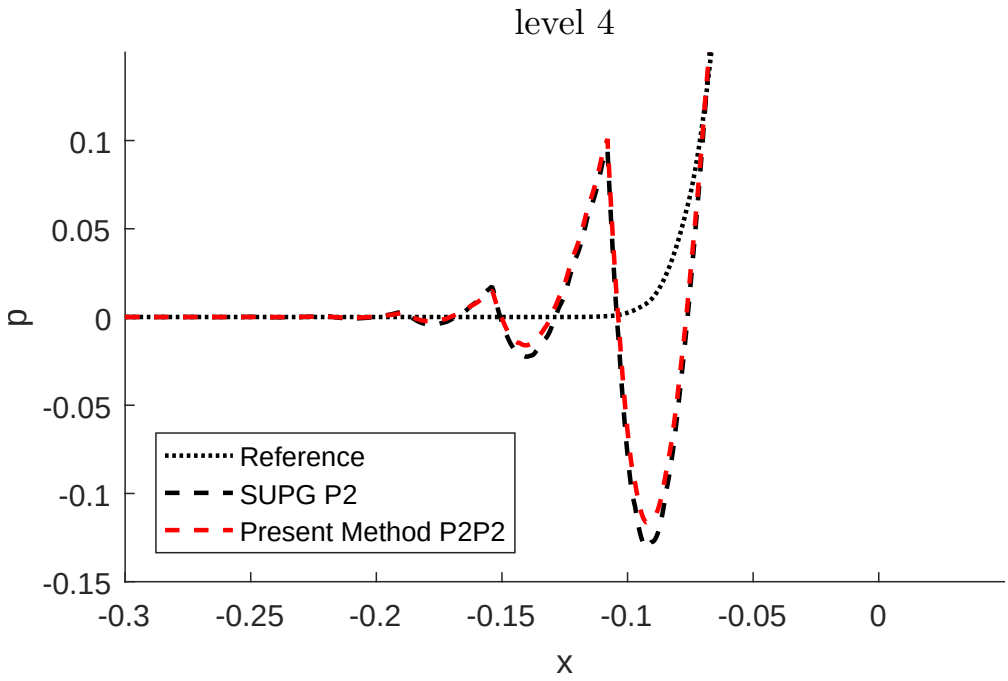


(a) x cross-section at $y = 1$

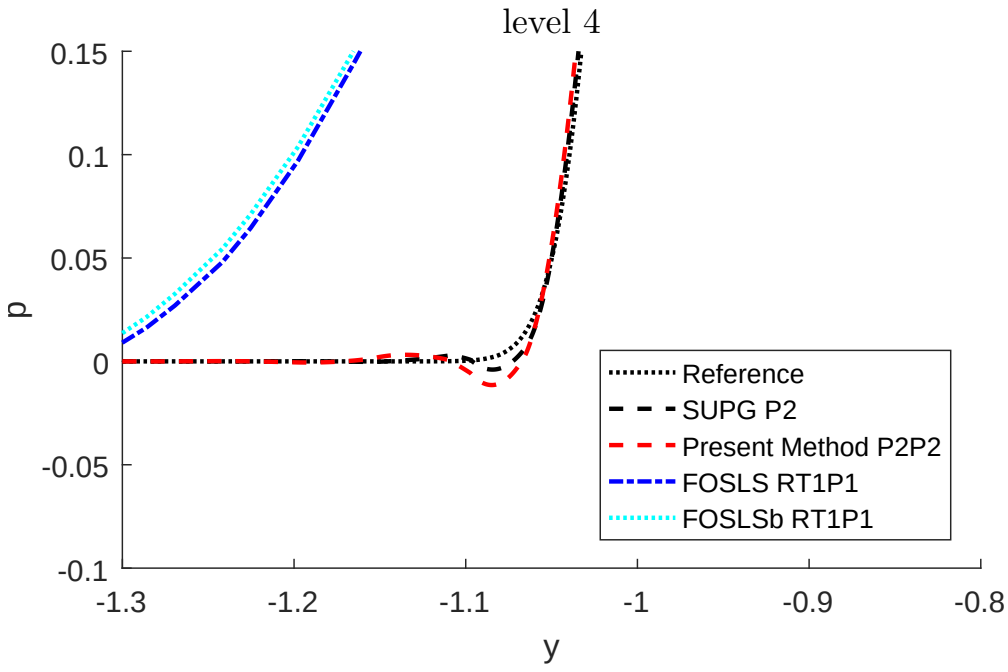


(b) y cross-section at $x = 4$

Fig: 5.12 Cross-sections using quadratic elements, level 4, $\varepsilon = 10^{-4}$ in the Hemker Study.

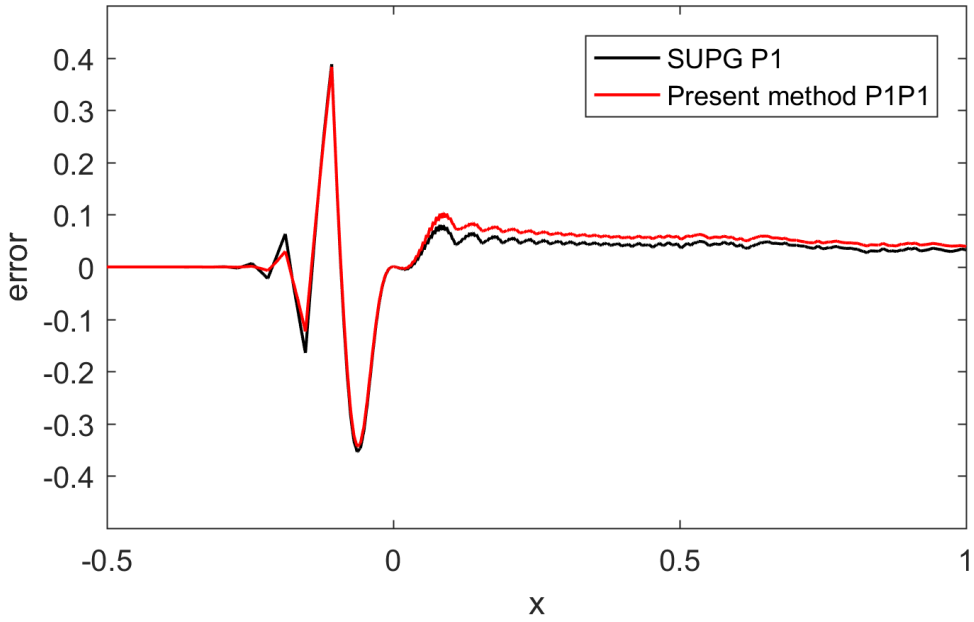


(a) x cross-section at $y = 1$

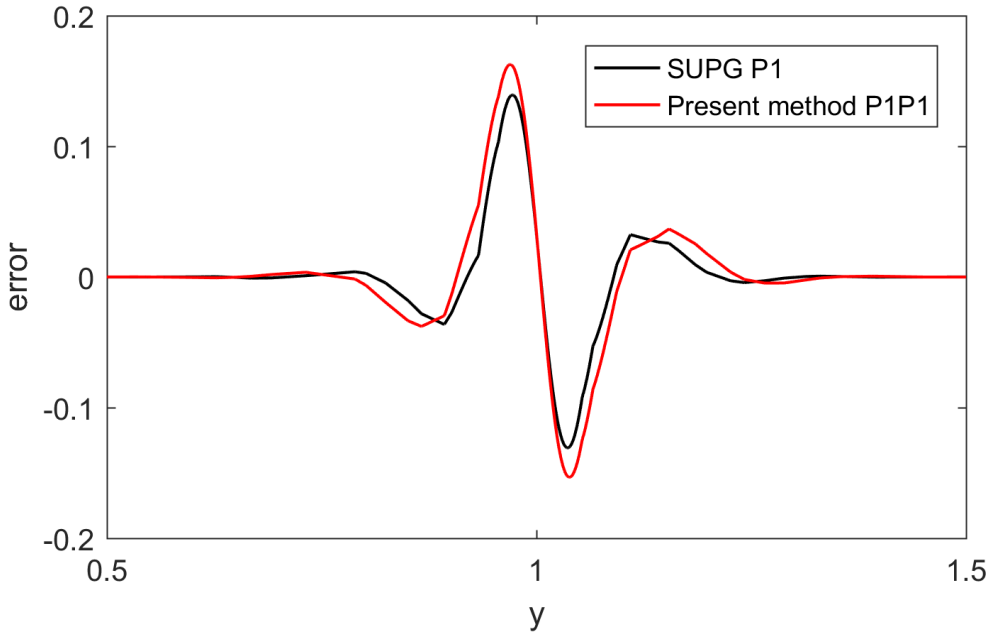


(b) y cross-section at $x = 4$

Fig: 5.13 Close-ups of cross-sections using quadratic elements, level 4, $\varepsilon = 10^{-4}$ in the Hemker Study.

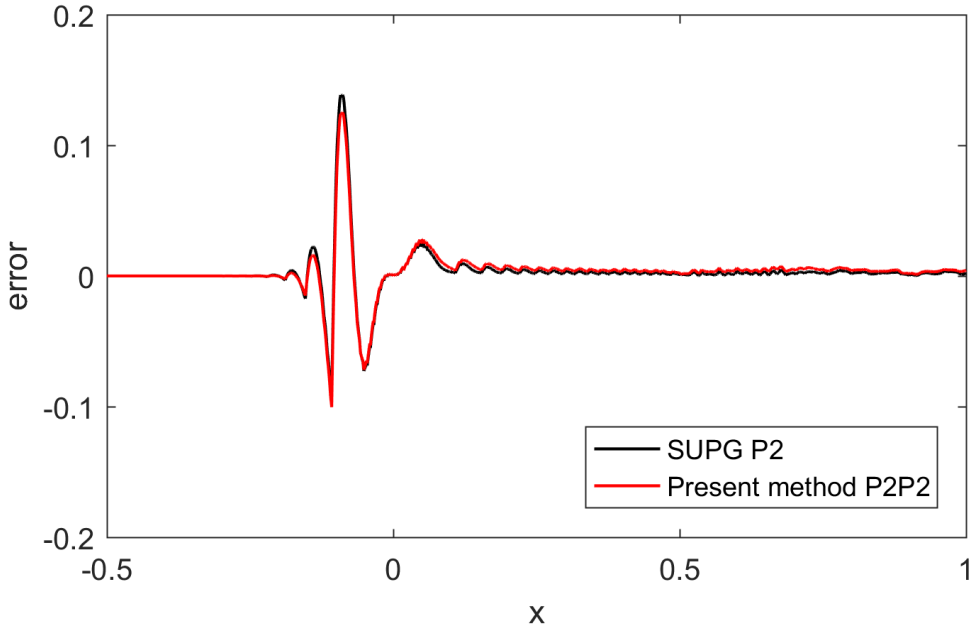


(a) x cross-section at $y = 1$

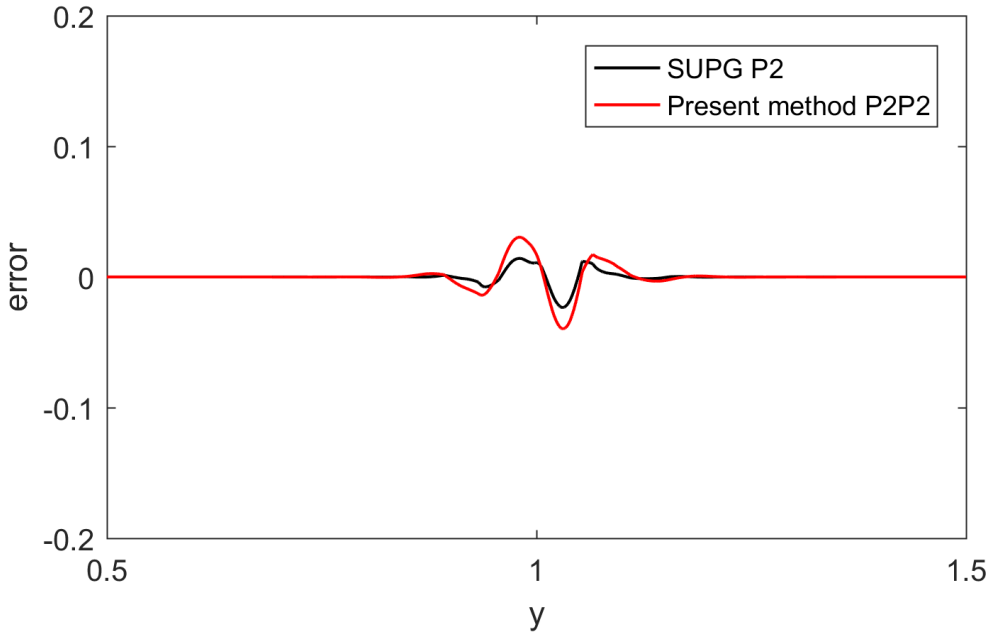


(b) y cross-section at $x = 4$

Fig: 5.14 Error with respect to the Reference Solution for linear elements in the Hemker study, $\varepsilon = 10^{-4}$, level 4.

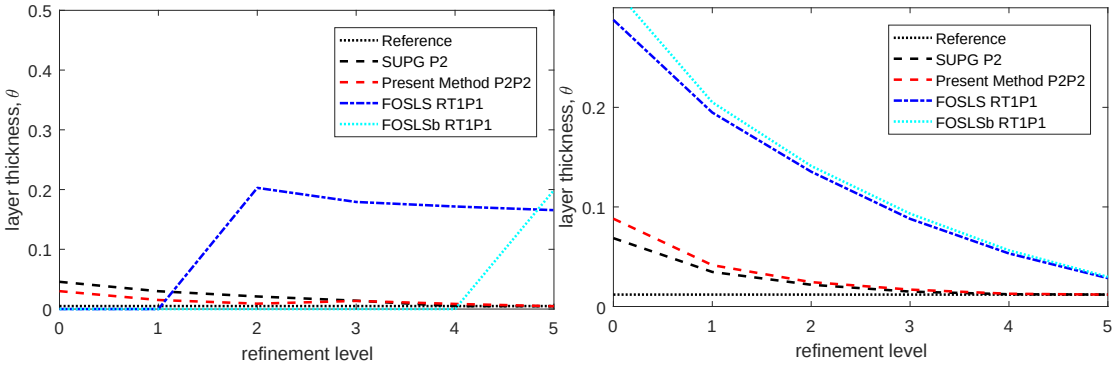


(a) x cross-section at $y = 1$



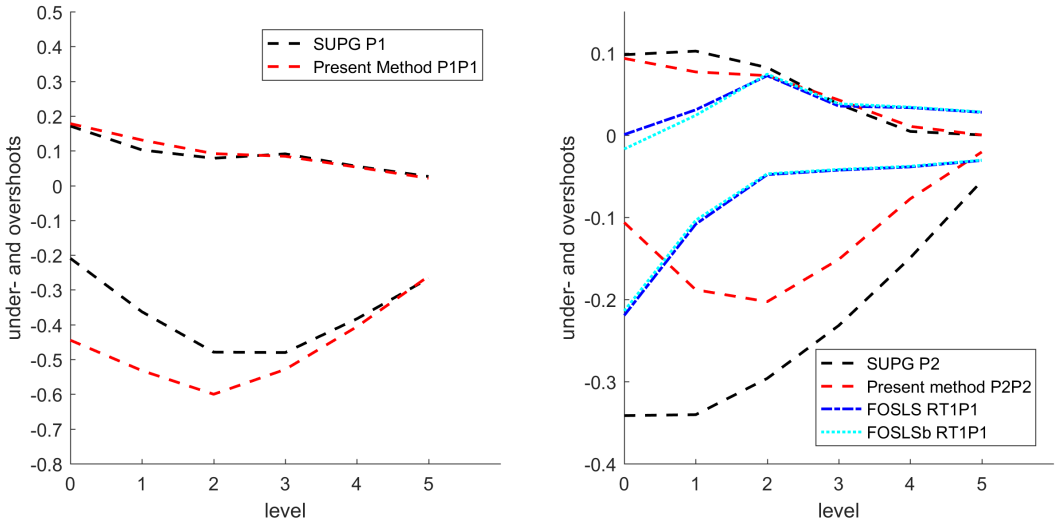
(b) y cross-section at $x = 4$

Fig: 5.15 Error with respect to the Reference Solution for quadratic elements in the Hemker study, $\varepsilon = 10^{-4}$, level 4.



(a) x cross-section at $y = 1$ (b) y cross-section at $x = 4$

Fig: 5.16 Layer thickness, θ, using quadratic elements for solution with $0.1 < p < 0.9$ in the Hemker study, $\varepsilon = 10^{-4}$.



(a) Linear elements (b) Quadratic elements

Fig: 5.17 Over- and undershoots in the Hemker study, $\varepsilon = 10^{-4}$.

In Figure 5.10, we depict elevations of solutions given by the Present Method using both linear and quadratic elements and also the solutions provided by both the FOSLS methods. For a more detailed comparison, in Figure 5.11 we depict the cross-sections of the solutions along the cut-lines $y = 1$ and $x = 4$. In this figure we use linear elements and also include both the reference solution and the SUPG solution on the same mesh using $\mathcal{P}_1$ elements. We repeat this process for the quadratic elements and in Figure 5.12 we depict cross-sections of the Present Method, the reference solution, the solution given by the different versions of the FOSLS methods presented in Section 2.7 and SUPG solutions on the same mesh. We observe that FOSLS fails to provide sharp layers. Close-ups of the regions near the layer on the left-hand side of the circle are shown in Figure 5.13. In Figure 5.13a the Present Method is a slight improvement on SUPG and in Figure 5.13b SUPG is slightly better on this very close-up comparison.
We proceed with further quantitative comparisons. In Figure 5.15 we depict the error of the computed solution with respect to that of the reference solution on level 4 along the cut-lines. We observe that the Present Method's results are at least comparable to the ones given by SUPG on the same mesh. The results of the other methods have been excluded since in some cases they lie outside the scale of the plot. Finally, we compute the layer thickness for all methods and the results in Figure 5.16 confirm that the Present Method provides steeper layers than the other mixed approaches. Then in Figure 5.17, we plot the over- and undershoots of all methods tested. The lower undershoots that occur in FOSLS are consistent with the wider, more diffuse layers.

5.3 Chapter Review

In this chapter, which was published in [BPY18], we conducted a comparative study of our new method, termed the 'Present Method' for challenging test problems with layers. Also included in the study were the MK [MK08] method. This proved unstable and had to be omitted from the layer thickness graphs for the advection skew to the mesh test when diffusion was less than 10^{-4} and the Hemker test. The FOSLS methods of Chen et al. [CFLQ14], with the weak imposition of the boundary conditions, were included in all studies but failed to cope with the internal or boundary layer problems. The cost of our new $\mathcal{P}_2\mathcal{P}_2$ method in terms of degrees of freedom is only slightly greater that [CFLQ14], but we obtain convergence of $\mathcal{O}(h^2)$ while the latter method only obtains $\mathcal{O}(h)$. A thorough comparison of our Present Method with the standard SUPG method was carried out and, while our new method was comparable, it did not show any outstanding improvement, particularly if the computational overhead is considered.

Chapter 6

Conclusions and Future Work

In this chapter we draw conclusions from the body of work carried out in the previous chapters and outline ideas for future work and lines of enquiry.

6.1 Conclusions

6.1.1 Unstabilised Mixed FE Methods

The unstabilised mixed finite element formulation of Douglas and Roberts [DR82, DR85], unsurprisingly, failed to give convergence when diffusion is much smaller than convection until the mesh size h was very small (see Figures 2.6, 2.7, 2.10, 2.11). It failed to converge in the case of $\varepsilon = 10^{-5}$ for both total and diffusive flux (see Figures 2.7f and 2.11f) and proved unstable when sharp layers were present. It also did not capture the outflow layer which can be seen in Figures 2.8, 2.9, 2.12, 2.13. The examination of the inf-sup constant for this method in Chapter 3 appears to show that, while the LBB inf-sup constant is greater than zero, it is very small and approaches zero as the diffusion becomes smaller. The inf-sup constant for the total system, within the range of our tests, was very small indeed and indicates that the stability is limited so the formulation is not able to cope with sharp layers.

FOSLS methods, while attractive for their elliptic nature and the avoidance of the inf-sup condition and the need for stabilisation, also struggled in tests with sharp layers and when diffusion is small compared with convection. The attempt to improve on this by the method of [CFLQ14], which imposes the boundary conditions weakly and uses a weighted FOSLS approach, also failed to meet the tests with sharp layers adequately. This method also struggled as the diffusion became smaller and did not capture the outflow layer in the test with advection skew to the mesh, which we found to be a

common problem with the FOSLS methods (see Figures 5.1 and 5.8). This defect is known to occur in most methods with weakly imposed boundary conditions. Applying established tests such as the advection skew to the mesh and the Hemker test revealed the deficiencies of this method clearly.

6.1.2 Other Stabilised FE Mixed Methods

The method of Thomas [Tho87], which modified the diffusive flux formulation with added jump terms on the scalar variable, failed to be capture sharp layers as the diffusion decreased. This can be seen clearly in Figure 2.15.

From our tests, the Masud and Kwack [MK08] methods, while showing some properties of convergence (Figures 2.16 and 2.17), became unstable in tests that featured sharp layers, especially when the convective flux was not aligned with the triangulation of the mesh as in their published work. In this case, oscillations developed with large undershoots which are depicted in Figures 5.2, 5.3 and 5.5. As a result of this instability the MK method was unable to cope with the Hemker test. This method also, to our knowledge, lacks formal analysis.

6.1.3 Our new method

The results presented in Chapter 5 show that our new method gave comparable results to SUPG on the same level of mesh refinement, especially if over- and undershoots using $\mathcal{P}_2\mathcal{P}_2$ Lagrangian elements shown in graph Figure 5.17b are considered. The analyses are a useful basis for other researchers who are interested in pursuing this line of enquiry.

The obvious disadvantage of our new method is that it is more expensive, having both the flux (vector) variable $\boldsymbol{v}$ and the scalar variable p. This results in three times as many degrees of freedom as a second-order method such as SUPG, which has just the one scalar variable. This leads to rapid growth in the size of the matrices for the whole system.

The SUPG/SDFEM method has had over 30 years of fine-tuning, particularly in the choice of the parameter, while the present method is giving comparable results already.

It could well be possible to improve the analysis or fine-tune the implementation to improve the performance. In particular, more local implementation of the δ_{div} parameter, which currently uses the maximum value of the mesh parameter and is not related to the direction of the convective flux. Also, the value of δ could be investigated further. Figures 4.7 and 4.9 indicate that $\delta = 1$ is a good choice for $\mathcal{P}_1\mathcal{P}_1$ elements and Figures 4.8 and 4.10 also suggest a value of $\delta = 1$ for $\mathcal{P}_2\mathcal{P}_2$ elements. However, no further fine tuning was attempted.

As both variables use the same Lagrangian finite elements and mesh, the computational over-head is kept to a minimum and, with the advances in the efficiency of direct solver routines, this is not such a big handicap. It might also be beneficial to use GPU methods for FE calculations when both variables use the same Lagrangian elements (as these perform better with increased workload), now that later versions of programming languages for mixes of CPU and GPU systems make it much easier and faster to transfer data between GPUs and the CPU.

6.2 Future Work

The method may be more accurate for obtaining the flux variable $\boldsymbol{v}$, which in some applications is the main variable that the user is interested in obtaining. This would be interesting to investigate.

The results obtained for inf-sup constant in Chapter 3 could be extended by the use of super-computing facilities which have high memory nodes, in order to confirm the trends we have seen and draw more definite conclusions suitable for publishing. The use of the PRIMME library [SM10] to optimise the retrieval of the smallest singular value for the inf-sup constant, as outlined in Section 3.2.2, could improve the performance with the very large matrices and enable extension of the table. At present, it seems both the LBB and the overall system inf-sup constants decrease as diffusion decreases and do not improve by decreasing the mesh parameter. However, although the inf-sup constant is very small it is greater than zero, but the unstabilised [DR82, DR85] formulation does not prove stable enough to cope with sharp layers.

While we were able to implement the method of [Tho87], the method of [SS97], which is a quasi- finite volume method with a stabilised term that relies on the direction of the normal on a finite element edge and knowledge of the neighbouring elements, proved

impossible to implement with either FreeFEM++ or FEniCS at present. There is also a problem with cell-centred finite volume formulations as FreeFEM++, for example, does not store information about neighbouring elements in the mesh in order to optimise performance of the code. There has been a project to incorporate Finite Volume into FreeFEM++ for over 10 years but only in the last few months are there signs that a solution is emerging. However, by modifying the training program LehrFem to incorporate $\mathcal{RT}_k \times \mathcal{P}_k^{\mathrm{dc}}$ elements fully, together with the modifications we made so that the edge normals are consistently directed, it would be possible to implement this method. This would also allow a comparative numerical study of other finite volume based methods as it is straightforward to store information about neighbouring elements. The drawback is that LehrFEM is not optimised for performance on large HPC applications, but it would be interesting to use it for simple tests on these types of combined finite element and finite volume formulations.

It would be informative to try further three-dimensional tests, including those with sharp layers to fully examine the abilities of the present method when deployed in 3D.

There is a significant body of literature of pre-conditioning for saddle-point matrices for elliptic equations [LS03, AFW97, PS03, VL96] and it is a topic of active research. However, it is less common to consider an asymmetric matrix such as that obtained in the CDR equation. The use of a block matrix pre-conditioner could aid rapid convergence to a solution using iterative methods and avoid the use of a direct multi-frontal solver such as UMFPACK. This would be an interesting line of future research.

Chapter A

Appendix

A.1 Software used

MATLAB [MAT18] was used initially to build finite element code following [ACF99] and then in the deployment and refactoring of LehrFEM [BFM14], the finite element training code of ETHZ. MATLAB was also used extensively to postprocess results and produce all the graphs in this thesis and to find the smallest singular values.

FreeFEM++ [Hec12] was used in all the Finite Element numerical calculations.
FEniCS [Log07, ABH$^+$15] was used for several months but the orientation of the edge normals was not well documented and did seem suitable for finite volume methods.

PARAVIEW [Aya15] was used for the visualisation of plots but had large memory overheads when finite elements were over 50,000 and does not deal with a variety of finite elements. Therefore ffglut that was provided with the FreeFEM++ [Hec12] program was mainly used.

A.2 Solvers used

The multifrontal solver UMFPACK [Dav04, Dav06] for sparse, unsymmetric matrices was used in the 64 bit version for Windows 10. This is now called UMFPACK64 in FreeFEM++ and, since its recent incorporation it, enables the building and solving of larger matrices. It proved more reliable than MUMPs or SuperLu.

A.3 FreeFem++ Programs

The quadrature used for RTo elements uses 7 points, with one Gauss point in the middle of each edge for the vectorial fluxes and one in the centre of each element. Other quadratures used for $\mathcal{P}_1$ and $\mathcal{P}_2$ are the standard quadratures detailed in the FreeFem++ manual [Hec12]. The refinement of the the mesh is achieved by using the command *mesh* Th = *square*$(2^n, 2^n)$ and nesting n in a loop. This simple method gives the effect of subdividing the each triangle into 4 with each iteration. For error calculations, a higher value of $\mathcal{P}_k$ than that being tested is always used to calculate the exact solution and increase the accuracy of the calculations.

A.3.1 Raviart-Thomas elements: Douglas-Roberts and Thomas Methods

```
load "Element_P3"
load "UMFPACK64"
macro div(u,v) (dx(u)+dy(v)) //
macro Grad(u)  [dx(u), dy(u)] //
load "Element_Mixte"
int power,k,maxiter=9;
func eps = 1.0*(10.^(-power));
func alpha1x=1.0; func alpha2y = 2.0; func ralphan = 1.0/sqrt((
   alpha1x^2 + alpha2y^2));
func alpha1 = alpha1x*ralphan; func alpha2= alpha2y*ralphan;
string pth="ThomasRT1P1";
real[int] pL2error(maxiter -3),gradpL2error(maxiter-3), divVerror(
   maxiter -3),vL2error(maxiter -3);
real hk, tau, tauk, realn, mu = 0.0;func g0 = 0.0;
func pexact = sin(2*pi*x)*sin(2*pi*y);
func gradpexact = [2*pi*cos(2*pi*x)*sin(2*pi*y),2*pi*sin(2*pi*x)*
   cos(2*pi*y)];
func vexact = +eps*gradpexact;
func f1 = 8* pi*pi*eps*pexact;
func f=  f1+ 2*pi*alpha1*cos(2*pi*x)*sin(2*pi*y)+2*pi*alpha2*sin
   (2*pi*x)*cos(2*pi*y)+ mu*pexact;

for( power = 0;power<6;power++){
        k =-1;
  cout<<"POWER"<<power<<endl;
  for(int n=3; n < maxiter; n++)
        {k =k+1;
```

```
24          mesh Th = square(2^n,2^n);
25          fespace Uh(Th,RT1); Uh [v1,v2],[w1,w2], [verr1,verr2];
26          fespace Ph(Th,P1dc); Ph p,q;
27          fespace Qh(Th,P3);
28          Qh pex = pexact,gradpex1=gradpexact[0],gradpex2=gradpexact
                [1],  vex1=vexact[0],vex2=vexact[1];
29          func reps = 1.0/eps;
30
31  // Thomas method
32          problem Thomas([v1,v2,p],[w1,w2,q], solver = sparsesolver,
                tgv= 1e30) =
33          int2d(Th)(reps*(v1*w1 + v2*w2)) +
34          int2d(Th)( p*div(w1,w2) + div(v1,v2)*q  + mu*p*q )
35          -int2d(Th)(0.5*alpha1*(q*dx(p)-p*dx(q))+0.5*alpha2*(q*dy(p
                )-p*dy(q)))
36          -intalledges(Th)((alpha1*N.x+alpha2*N.y)*(mean(p)-0.5*p)*q
                )
37          +intalledges(Th)(0.5* abs( alpha1*N.x+alpha2*N.y)*jump(p)*
                q)
38          + int2d(Th)(f*q)
39          + int1d(Th, 1,2,3,4) ((w1*N.x+w2*N.y)*g0 );
40
41  // DR total flux form
42
43          problem DRreps([v1,v2,p],[w1,w2,q], solver = sparsesolver,
                tgv= 1e30) = int2d(Th)(
44          reps*(v1*w1 + v2*w2) -  p*div(w1,w2)
45           - reps*p* (alpha1*w1+ alpha2*w2)
46          +div(v1,v2)*q + mu*p*q )
47          -int2d(Th)(f*q)
48          + int1d(Th, 1,2,3,4) ((w1*N.x+w2*N.y)*g0);//dirichlet
49
50
51  // DR diffusive flux form
52          problem DRnondivreps([v1,v2,p],[w1,w2,q], solver =
                sparsesolver, tgv= 1e30) = int2d(Th)(
53          reps*(v1*w1 + v2*w2) -  p*div(w1,w2)
54           - reps*q* (alpha1*v1+ alpha2*v2)
55         + div(v1,v2)*q + mu*p*q )
56           -  int2d(Th)(f*q)
57           + int1d(Th, 1,2,3,4) ((w1*N.x+w2*N.y)*g0 );
58
```

```
59          Thomas;
60
61          [verr1,verr2] = [v1-vex1 , v2-vex2];
62          vL2error[k] = sqrt(int2d(Th)(verr1^2 +verr2^2));
63          divVerror[k] = sqrt(int2d(Th)((verr1)^2 +(verr2)^2+ div(
                verr1,verr2)^2));
64          gradpL2error[k] =sqrt(int2d(Th)(( gradpex1-dx(p))^2 + (
                gradpex2-dy(p))^2));
65          pL2error[k]= sqrt(int2d(Th)((p-pex)^2));
66
67          if(n== maxiter-1){
68                  plot(p,dim = 3,fill=1,wait=1,value=true,cmm="power
                        :1e^-"+power);
69          }
70      }// n loop
71             cout<<"pL2error    "<<"vL2error   "<<"divVerror   "<<"grad
                   perror"<<endl;
72           for( int ii = 0; ii <maxiter-3; ii++){
73                  cout<<pL2error[ii]<<"    "<< vL2error[ii]<<"     "<<
                       gradpL2error[ii]<<"    "<<divVerror[ii]<<endl;
74           }
75
76             ofstream filepL2 ( pth+"-"+string(power)+"p_L2.dat");
77           filepL2 << pL2error << endl;
78           ofstream filevL2 (pth+"-"+string(power)+"v_L2.dat");
79           filevL2 << vL2error << endl;
80           ofstream filevDiv (pth+"-"+string(power)+"v_Div.dat");
81                  filevDiv << divVerror << endl;
82  }// power loop
```

A.3.2 Stabilsed Lagrangian Methods with total flux: MK, BPY

```
1
2  macro div(u,v) (dx(u)+dy(v)) //
3  macro Grad(u)  [dx(u), dy(u)] //
4  macro curl(u1,u2)  [dx(u2), -dy(u1)]//
5  load "UMFPACK64"
6  load "Element_P3"
7  load "Element_P4"
8
9  int power, k,maxiter =9; //ten for P1P1
10 real hk,  realn, taubpy;
11 func eps = 1.0*(10.^(-power));func reps = 1.0/eps;
12 func alpha1x=1.0; func alpha2y = 2.0; func ralphan = 1.0/sqrt((
       alpha1x^2 + alpha2y^2));
13 func alpha1 = alpha1x*ralphan; func alpha2= alpha2y*ralphan;func
       alpha= [alpha1,alpha2];
14 real real mu =0.0;string pth="MasudP2P2conv";
15 func gn2 = 2*pi*sin(2*pi*y)*eps; func gn3=2*pi*sin(2*pi*x)*eps;
16 real epsr= 1e-8;real a = 4.5, Pe;real gd =0.0, gd1 =1.0;
17 func Pe1 = sqrt(alpha1^2+ alpha2^2)/ eps; func tau= -a/(Pe+2*a)
       +1;func tauk=tau/eps;
18 real[int] pL2error(maxiter-3),gradpL2error(maxiter-3), divVerror(
       maxiter-3), vL2error(maxiter-3);
19 func pexact = sin(2*pi*x)*sin(2*pi*y);
20 func gradpexact = [2*pi*cos(2*pi*x)*sin(2*pi*y),2*pi*sin(2*pi*x)*
       cos(2*pi*y)];
21 func vexact = -eps*gradpexact+[alpha1, alpha2]*pexact;
22 func f1 = 8* pi*pi*eps*sin(2*pi*x)*sin(2*pi*y);
23 func f = f1+ 2*pi*alpha1*cos(2*pi*x)*sin(2*pi*y)+2*pi*alpha2*sin
       (2*pi*x)*cos(2*pi*y);
24
25  for( power = 0;power<6;power++){
26         k =-1;
27         cout<<"POWER"<<power<<endl;
28         for(int n=3; n <maxiter; n++)
29         {
30                 hk = sqrt(2.0)/10./n; Pe= Pe1*hk; cout << " hk="<<
                       hk <<endl;
31                 taubpy=0.5; k = k+1;
32                 mesh Th = square(2^n,2^n);
33                 plot(Th);
```

```
34                  fespace Uh(Th,P2); Uh v1,v2,w1,w2;
35                  fespace Qh(Th,P3);
36                   Qh pex = pexact,gradpex1=gradpexact[0],gradpex2=
                         gradpexact[1],                          vex1=
                         vexact[0],vex2=vexact[1];
37                  fespace Ph(Th,P2); Ph p,q;
38
39  //MK method
40  problem MK([v1,v2,p],[w1,w2,q]) =  int2d(Th)(
41                  (1-tau)* reps*(v1*w1 + v2*w2) -   reps*(1-tau)* p*
                         (alpha1*w1+ alpha2*w2)
42                  - p * div(w1,w2) + q* div(v1,v2) +epsr*p*q
43                   - tau* (dx(p)*w1+dy(p)* w2) + tau*( dx(q)*v1+dy(q
                         )*v2)
44                   - tau*p*(alpha1*dx(q)+alpha2*dy(q)) +  tau*eps*(
                         dx(p)*dx(q)+dy(p)*dy(q))
45                   + hTriangle*div(v1,v2)*div(w1,w2))
46                  - int2d(Th)(f*(q+hTriangle*div(w1,w2)))
47                  + on(1,2,3,4, p=0.0)
48                  + int1d(Th,1,2,3,4) ( gd*(w1*N.x+w2*N.y));//
                        dirichlet
49
50  //BPY method
51  problem BPY([v1,v2,p],[w1,w2,q]) =  int2d(Th)(
52                  (1-taubpy)* reps*(v1*w1 + v2*w2) -   reps*(1-
                        taubpy)* p* (alpha1*w1+ alpha2*w2)
53                  - p * div(w1,w2) + q* div(v1,v2)
54                   - taubpy* (dx(p)*w1+dy(p)* w2) + taubpy*( dx(q)*
                         v1+dy(q)*v2)
55                   - taubpy*p*(alpha1*dx(q)+alpha2*dy(q)) +  taubpy*
                         eps*(dx(p)*dx(q)+dy(p)*dy(q))
56                   -reps*taubpy*q*(v1*alpha1+v2*alpha2)+
                                                  taubpy*reps*p*q*(
                         alpha1*alpha1+alpha2*alpha2)
57                              -taubpy*q*(dx(p)*alpha1+dy(p)*alpha2)
58                  + hTriangle*min(hTriangle/4.*reps,1.0)*div(v1,v2)*
                        div(w1,w2))
59                  - int2d(Th)(f*(q+hTriangle*min(hTriangle/4.*reps
                        ,1.0)*div(w1,w2)))
60                              + on(1,2,3,4, p=0.0)
61                  + int1d(Th,1,2,3,4) ( gd*(w1*N.x+w2*N.y));//
                        dirichlet
```

```
62                  MK;
63  //              BPY;
64                  pL2error[k]= sqrt(int2d(Th)((pex-p)^2));
65                  gradpL2error[k] =sqrt(int2d(Th)(( gradpex1-dx(p))
                        ^2 + (gradpex2-dy(p))^2));
66                  vL2error[k]= sqrt(int2d(Th)((hTriangle*min(
                        hTriangle/4.*reps,1.0)*( v1-vex1)^2 + (v2-vex2
                        )^2)));
67                  divVerror[k]= sqrt(int2d(Th)((dx(v1)-dx(vex1)+dy(
                        v2)-dy(vex2))^2+( v1-vex1)^2+ (v2-vex2)^2));
68
69                  if(k>1){
70                  cout<<"p convergence rate = "<< log(pL2error[k-1]/
                        pL2error[k])/log(2.)  <<endl;
71                  cout<<"grad(p)convergence rate = "<< log(
                        gradpL2error[k-1]/gradpL2error[k])/log(2.) <<
                        endl;
72                  cout<<"v convergence rate = "<< log(vL2error[k-1]/
                        vL2error[k])/log(2.)  <<endl;
73                  cout<<"div convergence rate = "<< log(divVerror[k
                        -1]/divVerror[k])/log(2.)  <<endl;
74                  }
75                  if(n== maxiter-1){
76                  plot(p,dim = 2,fill=1,wait=1,value=true,cmm="power
                        :1e^-"+power);
77                  }
78    }// n loop
79                  cout<<"pL2error   "<<"vL2error  "<<"grad perror"<<
                        "divVerror  "<<endl;
80          for( int ii = 0; ii <maxiter-3; ii++){
81                  cout<<pL2error[ii]<<"   "<< vL2error[ii]<<"    "<<
                        gradpL2error[ii]<<"   "<<divVerror[ii]<<endl;
82          }
83                  ofstream filepL2 ( pth+"-"+string(power)+"p_L2.dat
                        ");
84                  filepL2 << pL2error << endl;
85                  ofstream filegradpL2 ( pth+"-"+string(power)+"
                        gradp_L2.dat");
86                  filegradpL2 <<gradpL2error << endl;
87                  ofstream filevL2 (pth+"-"+string(power)+"v_L2.dat"
                        );
88                  filevL2 << vL2error << endl;
```

```
89                  ofstream filevDiv (pth+"-"+string(power)+"v_Div.
                        dat");
90                  filevDiv << divVerror << endl;
91  }// power loop
```

A.3.3 FOSLS with Total Flux

```
1   macro div(u,v) (dx(u)+dy(v)) //
2   macro Grad(u)  [dx(u), dy(u)] //
3   macro curl(u1,u2)  [dx(u2), -dy(u1)]//
4
5   // Total flux formulation
6   problem FOSLStotflux([v1,v2,p],[w1,w2,q]) = int2d(Th)(
7              [v1,v2]`*[w1,w2] +eps*Grad(p)`*[w1,w2] +eps*[v1,v2]`*
                   Grad(q)
8                + eps^2*Grad(p)`*Grad(q)
9                - eps*q*[alpha1,alpha2]`*Grad(p) - p* [alpha1,
                     alpha2]`*[w1,w2]
10               - eps*p*[alpha1,alpha2]`*Grad(q)  -q* [alpha1,
                     alpha2]`*[v1,v2]
11               + p*q*(alpha1^2+alpha2^2)
12            + div(v1,v2)* div(w1,w2)
13               + mu*p*div(w1,w2) +mu*q*div(v1,v2)+mu^2*p*q)
14               -int2d(Th)(f* (div(w1,w2)+ mu*q))
15               + on(1,2,3,4, p=0.0);
```

A.3.4 FOSLS with Diffusive Flux and formulations [CFLQ14] used in Comparative Study

```
macro div(u,v) (dx(u)+dy(v)) //
macro Grad(u)  [dx(u), dy(u)] //
macro curl(u1,u2)  [dx(u2), -dy(u1)]//

//Diffusive flux formulation
 problem FOSLSdifflux([v1,v2,p],[w1,w2,q]) = int2d(Th)(
            ([v1,v2]+eps*Grad(p))'*([w1,w2]+eps*Grad(q))
            + (div(v1,v2)+Grad(p)'*alpha+mu*p)*( div(w1,w2)+Grad(q
                )'*alpha+mu*q))
            - int2d(Th)(f* ( div(w1,w2)+Grad(q)'f*alpha+mu*q))
                + on(1,2,3,4, p=0.0);


//Main formulation of Chen et al used in Comparative Study named
   FOSLS and Hemker Problem configuration
problem FOSLSdifflux([v1,v2,p],[w1,w2,q]) = int2d(Th)(
            ([v1,v2]+sqeps*Grad(p))'*([w1,w2]+sqeps*Grad(q))
                  + (sqeps*div(v1,v2)+Grad(p)'*alpha+mu*p)*( sqeps
                     *div(w1,w2)+Grad(q)'*alpha+mu*q))
                 -int2d(Th)(f* ( sqeps*div(w1,w2)+Grad(q)'*alpha+
                    mu*q))
                + int1d(Th,1,3,4)((1.0/ lenEdge*
                (eps + max(-1.0*(alpha1*N.x +alpha2*N.y),0.0)))*p*
                   q)
                +int1d(Th,2)((1.0/lenEdge *
                (eps + max(-1.0*(alpha1*N.x +alpha2*N.y),0.0)))*(
                   v1*N.x+v2*N.y)*(w1*N.x+w2*N.y))
                - int1d(Th,6,7)((1.0/lenEdge *
                (eps + max(-1.0*(alpha1*N.x +alpha2*N.y),0.0)))
                   *1.0*q)
                //      +on(4,p =0.0)
                 + on(5,8,p = 1.0);
```

```
//Alternative formulation of Chen et al. used in Comparative Study
//named FOSLSb and Hemker Problem configuration

problem FOSLSdiffluxtypeb([v1,v2,p],[w1,w2,q]) = int2d(Th)(
            ([v1,v2]+sqeps*Grad(p))'*([w1,w2]+sqeps*Grad(q))
                  + (sqeps*div(v1,v2)+Grad(p)'*alpha+mu*p)*( sqeps
                     *div(w1,w2)+Grad(q)'*alpha+mu*q))
                 -int2d(Th)(f* ( sqeps*div(w1,w2)+Grad(q)'*alpha+
                    mu*q))
                + int1d(Th,1,3)((eps/lenEdge + max(-1.0*(alpha1*N.
                   x +alpha2*N.y),0.0))*p*q)
                       +int1d(Th,2)( (
                eps/lenEdge + max(-1.0*(alpha1*N.x +alpha2*N.y)
                   ,0.0))*(v1*N.x+v2*N.y)*(w1*N.x+w2*N.y))
                - int1d(Th,6,7)((eps/lenEdge + max(-1.0*(alpha1*N.
                   x +alpha2*N.y),0.0))*1.0*q)
                +on(4,p =0.0) +on(5,8,p=1.0);
```

A.3.5 Hemker configuration

The parameterisation of the mesh using the parameter t on each part of the boundary can be seen in the listing below. The mesh is then built in FreeFem++ using a further parameter n, which can be scaled by a factor of 2 in each level of refinement. The Delauney meshes were constructed within FreeFem++ (see [Hec12] for details) and saved at each level of refinement. These were then loaded at the beginning of each run for the various levels in the test.

```
//Hemker geometric configuration
border circle1(t=0,pi/2){x=cos(t);y=sin(t);label =5;};
border circle2(t=pi/2,pi){x=cos(t);y=sin(t);label =6;};
border circle3(t=pi,3*pi/2){x=cos(t);y=sin(t);label =7;};
border circle4(t=3*pi/2,2*pi){x=cos(t);y=sin(t);label =8;};
border south(t=-3,9) {x=t;y=-3;label=1;};border east(t=-3,3){x=9;y
   =t;label =2;};
border north(t=9,-3){x=t;y=3;label =3;};border west(t=3,-3){x=-3;y
   =t;label =4;};
mesh Th = buildmesh(south(n)+east(n)+north(n)+west(n)+circle1(-n)
                +circle2(-n) +circle3(-n)+circle4(-n));
 savemesh(Th,"testHem.msh");
```

Bibliography

[ABH+15] Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells, *The FEniCS Project Version 1.5*, Archive of Numerical Software **3** (2015), no. 100.

[ACF99] Jochen Alberty, Carsten Carstensen, and Stefan A Funken, *Remarks around 50 lines of Matlab: short finite element implementation*, Numerical Algorithms **20** (1999), no. 2-3, 117–137.

[ACF+11] Matthias Augustin, Alfonso Caiazzo, André Fiebach, Jürgen Fuhrmann, Volker John, Alexander Linke, and Rudolf Umla, *An assessment of discretizations for convection-dominated convection-diffusion equations*, Computer Methods in Applied Mechanics and Engineering **200** (2011), no. 47-48, 3395–3409.

[AFW97] Douglas Arnold, Richard Falk, and Ragnar Winther, *Preconditioning in H(div) and applications*, Mathematics of Computation of the American Mathematical Society **66** (1997), no. 219, 957–984.

[AM09] Blanca Ayuso and L. Donatella Marini, *Discontinuous Galerkin methods for advection-diffusion-reaction problems*, SIAM Journal on Numerical Analysis **47** (2009), no. 2, 1391–1420.

[Aya15] Utkarsh Ayachit, *The PARAVIEW guide: a parallel visualization application*, Kitware, Inc., 2015.

[BBF13] Daniele Boffi, Franco Brezzi, and Michel Fortin, *Mixed Finite Element Methods and Applications*, Springer Series in Computational Mathematics, vol. 44, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[BBK17] Gabriel R. Barrenechea, Erik Burman, and Fotini Karakatsani, *Edge-based nonlinear diffusion for finite element approximations of convection–diffusion equations and its relation to algebraic flux-correction schemes*, Numerische Mathematik **135** (2017), no. 2, 521–545.

[BDG06] Pavel B. Bochev, Clark R. Dohrmann, and Max D. Gunzburger, *Stabilization of low-order mixed finite elements for the Stokes equations*, SIAM Journal on Numerical Analysis **44** (2006), no. 1, 82–101.

[BE05a] Erik Burman and Alexandre Ern, *Stabilized Galerkin approximation of convection-diffusion-reaction equations: discrete maximum principle and convergence*, Mathematics of computation **74** (2005), no. 252, 1637–1652.

[BE05b] ———, *Stabilized Galerkin approximation of convection-diffusion-reaction equations: discrete maximum principle and convergence*, Mathematics of computation **74** (2005), no. 252, 1637–1652.

[BFM14] Annegret Y. Burtscher, Eivind Fonn, and Patrick Meury, *LehrFEM – a 2d finite element toolbox*.

[BG03] Ivo Babuška and Gabriel N. Gatica, *On the mixed finite element method with Lagrange multipliers*, Numerical Methods for Partial Differential Equations **19** (2003), no. 2, 192–210.

[BG09] Pavel B. Bochev and Max D. Gunzburger, *Least-squares finite element methods*, Springer Science & Business Media, 2009.

[BGL05] Michele Benzi, Gene H. Golub, and Jörg Liesen, *Numerical solution of saddle point problems*, Acta Numerica **14** (2005), 1–137 (English).

[BH82] 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*, Computer Methods in Applied Mechanics and Engineering **32** (1982), no. 1, 199–259.

[BH04] Erik Burman and Peter Hansbo, *Edge stabilization for Galerkin approximations of convection–diffusion–reaction problems*, Computer Methods in Applied Mechanics and Engineering **193** (2004), no. 15-16, 1437–1453.

[BH14] Santiago Badia and Alba Hierro, *On monotonicity-preserving stabilized finite element approximations of transport problems*, SIAM Journal on Scientific computing **36** (2014), no. 6, A2673–A2697.

[BJK17] Gabriel R. Barrenechea, Volker John, and Petr Knobloch, *An algebraic flux correction scheme satisfying the discrete maximum principle and linearity preservation on general meshes*, Mathematical Models and Methods in Applied Sciences **27** (2017), no. 03, 525–548.

[BMM+06] Franco Brezzi, L. Donatella Marini, Stefano Micheletti, Paola Pietra, and Riccardo Sacco, *Stability and error analysis of mixed finite-volume methods for advection dominated problems*, Computers & Mathematics with Applications **51** (2006), no. 5, 681–696.

[BMO96] Jacques Baranger, Jean-François Maitre, and Fabienne Oudin, *Connection between finite volume and mixed finite element methods*, ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique **30** (1996), no. 4, 445–465.

[BPY18] Gabriel R. Barrenechea, Abner H. Poza, and Heather Yorston, *A stabilised finite element method for the convection–diffusion–reaction equation in mixed form*, Computer Methods in Applied Mechanics and Engineering **339** (2018), 389–415.

[BW18] Gabriel R. Barrenechea and Andreas Wachtel, *Stabilised finite element meth-*

ods for the Oseen problem on anisotropic quadrilateral meshes, ESAIM: Mathematical Modelling and Numerical Analysis **52** (2018), no. 1, 99–122.

[CF03] Zhi-Hao Cao and Li-Hong Feng, *A note on variational representation for singular values of matrix*, Applied Mathematics and Computation **143** (2003), no. 2-3, 559–563.

[CFLQ14] Huangxin Chen, Guosheng Fu, Jingzhi Li, and Weifeng Qiu, *First-order least squares method with weakly imposed boundary condition for convection dominated diffusion problems*, Computers and Mathematics with Applications **68** (2014), no. 12, Part A, 1635 – 1652.

[CJHZ03] Patrick Ciarlet Jr, Jianguo Huang, and Jun Zou, *Some observations on generalized saddle-point problems*, SIAM Journal on Matrix Analysis and Applications **25** (2003), no. 1, 224–236.

[CJMR97] Zhiqiang Cai, Jim E. Jones, Stephen F. McCormick, McCormick, and Thomas F. Russell, *Control-volume mixed finite element methods*, Computational Geosciences **1** (1997), no. 3-4, 289–315.

[CLMM94] Zhiqiang Cai, Raytcho Lazarov, Thomas A. Manteuffel, and Stephen F. McCormick, *First-order system least squares for second-order partial differential equations: Part I*, SIAM Journal on Numerical Analysis **31** (1994), no. 6, 1785–1799.

[CMM97] Zhiqiang Cai, Thomas A. Manteuffel, and Stephen F. McCormick, *First-order system least squares for second-order partial differential equations: Part II*, SIAM Journal on Numerical Analysis **34** (1997), no. 2, 425–454.

[CMMR01] Zhiqiang Cai, Thomas A. Manteuffel, Stephen F. McCormick, and John Ruge, *First–Order System (FOSLL*): Scalar Elliptic Partial Differential Equations*, SIAM Journal on Numerical Analysis **39** (2001), no. 4, 1418–1445.

[CS01] Bernardo Cockburn and Chi-Wang Shu, *Runge–kutta discontinuous Galerkin methods for convection-dominated problems*, Journal of scientific computing **16** (2001), no. 3, 173–261.

[Dav04] Timothy A. Davis, *Algorithm 832: UMFPACK V4.3—an Unsymmetric-pattern Multifrontal Method*, ACM Trans. Math. Softw. **30** (2004), no. 2, 196–199.

[Dav06] ———, *Direct methods for sparse linear systems*, vol. 2, Siam, 2006.

[DF16] Rafail Z. Dautov and E. M. Fedotov, *Hybridized schemes of the discontinuous Galerkin method for stationary convection–diffusion problems*, Differential Equations **52** (2016), no. 7, 906–925.

[DR82] Jim Douglas and Jean E. Roberts, *Mixed Finite Element Methods for Second Order Elliptic Problems*, Mat. Aplic. Comp **1** (1982), no. 1, 91–103.

[DR85] ———, *Global Estimates for Mixed Methods for Second Order Elliptic Equations*, Mathematics of Computation **44** (1985), no. 169, 39–52.

[DXY18] Yana Di, Hehu Xie, and Xiaobo Yin, *Anisotropic meshes and stabilization parameter design of linear supg method for 2d convection-dominated convection–diffusion equations*, Journal of Scientific Computing **76** (2018), no. 1, 48–68.

[EG13] Alexandre Ern and Jean-Luc Guermond, *Theory and practice of finite elements*, vol. 159, Springer Science & Business Media, 2013.

[FFH92] Leopoldo P. Franca, Sergio L. Frey, and Thomas J.R. Hughes, *Stabilized finite element methods: I. application to the advective-diffusive model*, Computer Methods in Applied Mechanics and Engineering **95** (1992), no. 2, 253–276.

[FKS12] Sebastian Franz, R. Bruce Kellogg, and Martin Stynes, *Galerkin and streamline diffusion finite element methods on a Shishkin mesh for a convection-diffusion problem with corner singularities*, Mathematics of Computation **81** (2012), no. 278, 661–685.

[FLR08] Sebastian Franz, Torsten Linß, and Hans-Görg Roos, *Superconvergence analysis of the SDFEM for elliptic problems with characteristic layers*, Applied Numerical Mathematics **58** (2008), no. 12, 1818–1829.

[FMM98] Jean Michel Fiard, Thomas A. Manteuffel, and Stephen F. McCormick, *First-order system least squares (FOSLS) for convection-diffusion problems: Numerical results*, SIAM Journal on Scientific Computing **19** (1998), no. 6, 1958–1979.

[FMP04] Luca Formaggia, Stefano Micheletti, and Simona Perotto, *Anisotropic mesh adaptation in computational fluid dynamics: application to the advection–diffusion–reaction and the stokes problems*, Applied Numerical Mathematics **51** (2004), no. 4, 511–533.

[GR86] Vivette Girault and Pierre-Arnaud Raviart, *Finite element methods for navier-stokes equations: theory and algorithms*, vol. 5, Springer Science & Business Media, 1986.

[Hec12] Frédéric Hecht, *New development in FreeFem++*, J. Numer. Math. **20** (2012), no. 3-4, 251–265. MR 3043640

[Hem96] Piet W. Hemker, *A singularly perturbed model problem for numerical computation*, Journal of Computational and Applied Mathematics **76** (1996), no. 1, 277–285.

[HJVC14] Elie Hachem, Ghina Jannoun, Jérémy Veysset, and Thierry Coupez, *On the stabilized finite element method for steady convection-dominated problems with anisotropic mesh adaptation*, Applied Mathematics and Computation **232** (2014), 581–594.

[HMA86] Thomas J. R. Hughes, Michel Mallet, and Mizukami Akira, *A New Finite Element Formulation for Computational Fluid Dynamics: II. Beyond SUPG*, Computer Methods in Applied Mechanics and Engineering **54** (1986), no. 3, 341–355.

[HSV12] Antti Hannukainen, Rolf Stenberg, and Martin Vohralík, *A unified framework for a posteriori error estimation for the Stokes problem*, Numerische Mathematik **122** (2012), no. 4, 725–769.

[HY09] Po-Wen Hsieh and Suh-Yuh Yang, *On efficient least-squares finite element methods for convection-dominated problems*, Computer Methods in Applied Mechanics and Engineering **199** (2009), no. 1, 183–196.

[HY10] ———, *A novel least-squares finite element method enriched with residual-free bubbles for solving convection-dominated problems*, SIAM Journal on Scientific Computing **32** (2010), no. 4, 2047–2073.

[Jaf84] Jerome Jaffré, *Décentrage et éléments finis mixtes pour les équations de diffusion-convection*, Calcolo **21** (1984), no. 2, 171–197.

[JK07] Volker John and Petr Knobloch, *On spurious oscillations at layers diminishing (SOLD) methods for convection—diffusion equations: Part I— A review*, Computer Methods in Applied Mechanics and Engineering **196** (2007), no. 17, 2197–2215.

[JK08] ———, *On spurious oscillations at layers diminishing (SOLD) methods for convection—diffusion equations: Part II—Analysis for P1 and Q1 finite elements*, Computer Methods in Applied Mechanics and Engineering **197** (2008), no. 21, 1997–2014.

[JKN18] Volker John, Petr Knobloch, and Julia Novo, *Finite elements for scalar convection-dominated equations and incompressible flow problems: a never ending story?*, Computing and Visualization in Science **19** (2018), no. 5-6, 47–63.

[JN13] Volker John and Julia Novo, *A robust supg norm a posteriori error estimator for stationary convection–diffusion equations*, Computer Methods in Applied Mechanics and Engineering **255** (2013), 289–305.

[JP93] Bo-Nan Jiang and Louis A Povinelli, *Optimal least-squares finite element method for elliptic problems*, Computer Methods in Applied Mechanics and Engineering **102** (1993), no. 2, 199–212.

[Kan08] Guido Kanschat, *Discontinuous Galerkin methods for viscous incompressible flow*, Springer, 2008.

[Kno06a] Petr Knobloch, *Improvements of the mizukami–hughes method for convection–diffusion equations*, Computer methods in applied mechanics and engineering **196** (2006), no. 1-3, 579–594.

[Kno06b] ———, *Improvements of the Mizukami–Hughes method for convection–*

diffusion equations, Computer methods in applied mechanics and engineering **196** (2006), no. 1-3, 579–594.

[Kno10] ———, *A generalization of the local projection stabilization for convection-diffusion-reaction equations*, SIAM Journal on Numerical Analysis **48** (2010), no. 2, 659–680.

[Kop04] Natalia Kopteva, *How accurate is the streamline-diffusion fem inside characteristic (boundary and interior) layers?*, Computer methods in applied mechanics and engineering **193** (2004), no. 45-47, 4875–4889.

[KS05] R. Bruce Kellogg and Martin Stynes, *Corner singularities and boundary layers in a simple convection–diffusion problem*, Journal of Differential Equations **213** (2005), no. 1, 81–120.

[KS07] ———, *Sharpened bounds for corner singularities and boundary layers in a simple convection–diffusion problem*, Applied mathematics letters **20** (2007), no. 5, 539–544.

[Log07] Anders Logg, *Automating the Finite Element Method*, Archives of Computational Methods in Engineering **14** (2007), no. 2, 93–138.

[LR74] Pierre Lesaint and Pierre-Arnaud Raviart, *On a finite element method for solving the neutron transport equation*, Mathematical aspects of finite elements in partial differential equations (1974), no. 33, 89–123.

[LS98] Richard B. Lehoucq and Chao Sorensen, Danny C .and Yang, *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, vol. 6, Siam, 1998.

[LS01] Torsten Linß and Martin Stynes, *The SDFEM on Shishkin meshes for linear convection-diffusion problems*, Numerische Mathematik **87** (2001), no. 3, 457–484.

[LS03] Leigh Little and Yousef Saad, *Block preconditioners for saddle point problems*, Numerical Algorithms **33** (2003), no. 1, 367–379.

[LTV97] Raytcho D. Lazarov, Lutz Tobiska, and Panayot S Vassilevski, *Streamline diffusion least-squares mixed finite element methods for convection-diffusion problems*, East West Journal of Numerical Mathematics **5** (1997), 249–264.

[MAT18] MATLAB, *version 9.5.0 (R2018b)*, The MathWorks Inc., Natick, Massachusetts, 2018.

[MH85] Akira Mizukami and Thomas J.R. Hughes, *A Petrov-Galerkin finite element method for convection-dominated flows: an accurate upwinding technique for satisfying the maximum principle*, Computer Methods in Applied Mechanics and Engineering **50** (1985), no. 2, 181–193.

[MK08] Arif Masud and JaeHyuk Kwack, *A stabilized mixed finite element method for the first-order form of advection—diffusion equation*, International Journal for Numerical Methods in Fluids **57** (2008), no. 9, 1321–1348.

[NGJB09] Hoa Nguyen, Max Gunzburger, Lili Ju, and John Burkardt, *Adaptive anisotropic meshing for steady convection-dominated problems*, Computer Methods in Applied Mechanics and Engineering **198** (2009), no. 37-40, 2964–2981.

[NPC09] Ngoc Cuong Nguyen, Jaume Peraire, and Bernardo Cockburn, *An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations*, Journal of Computational Physics **228** (2009), no. 9, 3232–3254.

[PS03] Catherine E. Powell and David Silvester, *Optimal preconditioning for Raviart–Thomas mixed formulation of second-order elliptic problems*, SIAM journal on matrix analysis and applications **25** (2003), no. 3, 718–738.

[QV08] Alfio M Quarteroni and Alberto Valli, *Numerical Approximation of Partial Differential Equations*, vol. 23, Springer, 2008.

[RST08] Hans-Görg Roos, Martin Stynes, and Lutz Tobiska, *Robust numerical methods for singularly perturbed differential equations: convection–diffusion–reaction and flow problems*, vol. 24, Springer Science & Business Media, 2008.

[RT77] Pierre-Arnaud Raviart and Jean-Marie Thomas, *A mixed finite element method for 2–nd order elliptic problems*, Mathematical aspects of finite element methods, Springer, 1977, pp. 292–315.

[SM10] Andreas Stathopoulos and James R. McCombs, *PRIMME: PReconditioned Iterative MultiMethod Eigensolver: Methods and software description*, ACM Transactions on Mathematical Software **37** (2010), no. 2, 21:1–21:30.

[SS97] Riccardo Sacco and Fausto Emilio Saleri, *Stabilized mixed finite volume methods for convection-diffusion problems*, East-West Journal of Numerical Mathematics **5** (1997), 291–311.

[ST03] Martin Stynes and Lutz Tobiska, *The SDFEM for a convection–diffusion problem with a boundary layer: optimal error analysis and enhancement of accuracy*, SIAM Journal on Numerical Analysis **41** (2003), no. 5, 1620–1642.

[Sty13] Martin Stynes, *Numerical methods for convection–diffusion problems or the 30 years war*, arXiv preprint arXiv:1306.5172 (2013).

[Tho87] Jean-Marie Thomas, *Mixed Finite Elements Methods for Convection-Diffusion Problems* , Numerical Approximation of Partial Differential Equations –Selection of Papers Presented at the International Symposium on Numerical Analysis held at the Polytechnic University of Madrid (Eduardo L. Ortiz, ed.), North-Holland Mathematics Studies, vol. 133, North-Holland, 1987, pp. 241 – 250.

[VL96] Panayot S. Vassilevski and Raytcho D. Lazarov, *Preconditioning Mixed Finite Element Saddle-point Elliptic Problems*, Numerical linear algebra with applications **3** (1996), no. 1, 1–20.

[Wac15] Andreas Wachtel, *Stabilised mixed finite element methods on anisotropic meshes*, Ph.D. thesis, University of Strathclyde, 2015.

[YHK13] Hamdullah Yücel, Matthias Heinkenschloss, and Bülent Karasözen, *Distributed optimal control of diffusion-convection-reaction equations using Discontinuous Galerkin methods*, Numerical Mathematics and Advanced Applications 2011, Springer, 2013, pp. 389–397.