Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 339 (2018) 389–415
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A stabilised finite element method for the convection–diffusion–reaction equation in mixed form

Gabriel R. Barrenechea[a,*], Abner H. Poza[b], Heather Yorston[a]

[a] Department of Mathematics and Statistics, University of Strathclyde, 26 Richmond Street, Glasgow G1 1XH, Scotland
[b] Departamento de Matemática y Física Aplicadas, Facultad de Ingeniería, Universidad Católica de la Santísima Concepción, Casilla 297, Concepción, Chile

Received 16 November 2017; received in revised form 11 April 2018; accepted 12 April 2018
Available online 30 May 2018

Abstract

This paper is devoted to the approximation of the convection–diffusion–reaction equation using a mixed, first-order, formulation. We propose, and analyse, a stabilised finite element method that allows equal order interpolations for the primal and dual variables. This formulation, reminiscent of the Galerkin least-squares method, is proven stable and convergent. In addition, a numerical assessment of the numerical performance of different stabilised finite element methods for the mixed formulation is carried out, and the different methods are compared in terms of accuracy, stability, and sharpness of the layers for two different classical test problems.

Keywords: First-order formulation; Convection–diffusion–reaction equation; Stabilised finite element method; Numerical comparison

1. Introduction

Despite the large amount of work that has been devoted to the numerical approximation of convection-dominated problems, there is still the open question of finding a method that 'ticks all the boxes'. By this, we mean a method that provides stable results while not smearing the sharp layers appearing in the solution. For example, the SUPG method (cf. [1,2]) has been accepted as an efficient method that produces sharp layers, but at the cost of producing over- and undershoots in the regions close to them. In order to avoid these non-physical oscillations, several methods have been proposed over the years, including Continuous Interior Penalty (e.g. [3]), LPS methods (e.g. [4]), or using shock-capturing related ideas (see, e.g., [5,6] for a review, and [7–10] for more recent developments). Several alternatives were compared in the relatively recent paper [11], and the conclusion was that, up to that date, no method could be considered to be completely satisfactory.

* Corresponding author.
E-mail addresses: gabriel.barrenechea@strath.ac.uk (G.R. Barrenechea), apoza@ucsc.cl (A.H. Poza), heather.yorston@strath.ac.uk (H. Yorston).

https://doi.org/10.1016/j.cma.2018.04.019
0045-7825/

Alternatively, some attempts have been made to approximate this problem by first rewriting it as a first-order system. To the best of our knowledge, the first papers that addressed this possibility were [12,13]. Different first-order formulations were tried in these papers, and the discretisation was carried out by means of Raviart–Thomas finite element methods (cf. [14]). Nevertheless, two issues remain that are not covered by those papers. Firstly, the numerical stability of the resulting scheme was only proven when the mesh discretisation parameter was small enough, which limits the applicability of such a discretisation to the diffusion-dominated case. Secondly, since the discretisation did not include any form of stabilisation, the same instabilities from the plain Galerkin scheme are to be expected for this mixed method. With the aim of addressing that issue, in [15] the author proposes a new method, which also uses Raviart–Thomas spaces, but adds an upwind-based stabilisation. Nevertheless, the resulting method is only applicable to higher order discretisations. A more modern approach, including a posteriori error estimation and different choices for finite element spaces, can be found in [16].

Several works have tried to address the points raised in the previous paragraph. For example, one possibility is to consider a least-squares method, such as FOSLS. This leads to an elliptic problem, thus freeing the choice of the finite element spaces, see, e.g., [17–19] and the references therein, or [20] for more general least-squares methods and an extensive review. One disadvantage of this sort of approach is that it leads to fairly diffusive layers, thus, again, making its interest for convection-dominated problems limited; see [21] for a discussion on this issue, [22] for the possibility of using a FOSLS method combined with an enrichment of the finite element space with bubble functions, or [23] for a streamline-based FOSLS method. To address this issue, in [24] a weighted FOSLS method was proposed, combined with a weak imposition of the boundary conditions. Alternatively, some finite volume-inspired methods have been proposed in conjunction with Raviart–Thomas elements (see, e.g. [25]). However, their performance for problems that contain strong layers is still to be tested. Other approaches to stabilise this mixed problem include the hybridised discontinuous Galerkin methods (see, e.g., [26]), the discontinuous Petrov–Galerkin method with optimal test functions (see, e.g. [27,28]), and augmented formulations (see, e.g., [29,30]). It is interesting to remark that almost none of the references just quoted use Lagrangian elements for both variables. In fact, several of them make use of the Raviart–Thomas' space for the vector variable, even in the case the final formulation is driven by an elliptic bilinear form.

In this work we pursue a different approach. Our interest is to approximate the convection–diffusion–reaction equation using a mixed, first-order formulation, but using standard Lagrangian elements in both variables. Thus, stabilisation is needed in order to prove stability and convergence. As far as we are aware, the only method that has been proposed with this purpose is the one presented in [31], which is a modification of the method proposed in [32] for the Darcy equation. In the work [31] no stability, or error estimates, are proven. Our first aim is to bridge this gap. In the process of trying to analyse the method from [31], the need to modify its definition appeared. Thus, in this work we propose a new stabilised mixed finite element method for the first order writing of the convection–diffusion–reaction equation, which can be proven to be stable and convergent. To assess the performance of the new method, we have also carried out intensive comparisons with several previously existing alternative methods. More precisely, by means of two standard test cases for the convection–diffusion equation we have compared the new method to the original method from [31] and two variants of the FOSLS approach. As a reference, we have also considered the results provided by the SUPG method.

The rest of this manuscript is organised as follows. In Section 2 the problem of interest and the main notations are introduced. The stabilised finite element method is presented in Section 3, and its stability is proven. In Section 4 error estimates are shown, and these are corroborated numerically in Section 5. In Section 6 some alternative finite element methods for the mixed formulation of the convection–diffusion equation are reviewed, and then a detailed comparison of the performance of these alternatives with the present approach is given.

2. Notation and preliminaries

We consider $\Omega \subseteq \mathbb{R}^d, d = 2, 3$, an open, bounded, polyhedral domain with Lipschitz boundary Γ. Standard notations for Sobolev spaces and their corresponding norms are used throughout. For $D \subseteq \Omega$, the inner product in $L^2(D)$, or $L^2(D)^d$, is denoted by $(\cdot,\cdot)_D$. In the case $D = \Omega$ the subscript will be dropped. The norm and semi-norm in $W^{m,p}(D)$ will be denoted by $\|\cdot\|_{m,p,D}$ and $|\cdot|_{m,p,D}$, respectively, with the convention $\|\cdot\|_{m,D} = \|\cdot\|_{m,2,D}$, where $H^m(D) = W^{m,2}(D)$ and $L^2(D) = H^0(D)$. We also introduce the subspace of $L^2(\Omega)^d$:

$$H(\text{div};\Omega) = \left\{ \boldsymbol{w} \in L^2(\Omega)^d : \nabla\cdot\boldsymbol{w} \in L^2(\Omega) \right\}.$$