MATHEMATICS OF COMPUTATION
Volume 70, Number 233, Pages 77–106
S 0025-5718(00)01187-X
Article electronically published on March 3, 2000

ADAPTIVE LAGRANGE–GALERKIN METHODS FOR UNSTEADY CONVECTION-DIFFUSION PROBLEMS

PAUL HOUSTON AND ENDRE SÜLI

Abstract. In this paper we derive an *a posteriori* error bound for the Lagrange–Galerkin discretisation of an unsteady (linear) convection-diffusion problem, assuming only that the underlying space-time mesh is nondegenerate. The proof of this error bound is based on strong stability estimates of an associated dual problem, together with the Galerkin orthogonality of the finite element method. Based on this *a posteriori* bound, we design and implement the corresponding adaptive algorithm to ensure global control of the error with respect to a user-defined tolerance.

1. Introduction and preliminaries

The modelling of the interaction between convective and diffusive processes is of fundamental importance in many areas of applied mathematics; in particular, meteorology, oil reservoir simulation, aerodynamics and physiology, for example. In many such applications convection essentially dominates diffusion, which leads to a "nearly" hyperbolic set of governing partial differential equations. Typically, solutions to these equations exhibit localised phenomena, such as propagating "near-shocks" and sharp transition layers, and their numerical approximation presents a challenging computational task; indeed, it is well documented that many standard numerical methods, developed for diffusion-dominated processes, often behave very poorly when applied to these types of problems. Moreover, the presence of local singularities in the solution may lead to a global deterioration of the numerical approximation. Hence, in order to resolve such localised features, in an accurate and efficient manner, it is essential to implement an adaptive algorithm that is capable of *automatically* refining the discretisation within regions of the computational domain where these transition layers are located.

This paper is concerned with the development of adaptive Lagrange–Galerkin methods for singularly perturbed unsteady convection-diffusion problems. The Lagrange–Galerkin method is based on combining the method of characteristics with the standard Galerkin finite element method. This procedure gives rise to both a highly accurate and stable numerical scheme for transient problems (see Bercovier & Pironneau [3], Douglas & Russell [6] and Pironneau [19], for example). The adaptive algorithm for the Lagrange–Galerkin method developed here is driven by a residual-based *a posteriori* error bound. In particular, we shall derive an *a posteriori* error bound in the norm of the function space $L^2(0,T;L^2(\Omega))$, where

Received by the editor December 16, 1997 and, in revised form, January 4, 1999.
2000 *Mathematics Subject Classification.* Primary 65M15; Secondary 65M25, 65M60.
We acknowledge the financial support of the EPSRC (Grant GR/K76221).

$T > 0$ is a final time and Ω denotes the spatial domain. The proof of this error bound is based on strong stability estimates of an associated dual problem, together with the Galerkin orthogonality of the finite element method, based on the general theoretical framework of *a posteriori* error estimation developed by Eriksson & Johnson [7, 8], Eriksson *et al.* [9] and Hansbo & Johnson [11], for example.

It is worth noting that many of the *a posteriori* error bounds derived in the literature for unsteady problems are based on the assumption that the spatial mesh function h is *continuously differentiable* on $\bar{\Omega}$ and that the gradient of h is uniformly bounded by a constant μ, where μ is assumed to be *sufficiently small*, cf. [7, 8, 15]. Clearly, these conditions do not preclude the use of nonuniform meshes; however, the variation in the size of the elements in the mesh must be very smooth, so that the change in h is sufficiently small to guarantee that $|\nabla h(\mathbf{x})| \leq \mu$ for all $\mathbf{x}$ in $\bar{\Omega}$. For practical computations in two or three space-dimensions, such a restriction on h may be unrealistic.

The *a posteriori* error analysis presented in this paper will assume only that the underlying space-time mesh is *nondegenerate*. To achieve this, the key part of the proof of the error bound relies on the definition in space of the function ϕ_h introduced using the Galerkin orthogonality of the finite element method. We note that for a given $\phi \in L^2(\Omega)$ there are essentially two main requirements that must be satisfied by the mapping $\phi \mapsto \phi_h$. Firstly, that it is stable in the $L^2(\Omega)$ norm, i.e., there exists a positive constant C, independent of h, such that

$$\|\phi_h\| \leq C\|\phi\|. \tag{1.1}$$

Secondly, that the approximation error between the solution ϕ of a suitable dual problem and the corresponding ϕ_h (measured in some appropriate norm) can be bounded *locally* on an element κ (or on a patch of elements surrounding κ), so that the spatial mesh may be arbitrarily nonuniform.

In [7, 8, 15], ϕ_h is defined to be the space-time L^2-projection of ϕ. Clearly, this choice automatically satisfies (1.1) with $C = 1$. However, the spatial projection error cannot be bounded locally due to the global nature of projecting onto continuous piecewise polynomial functions. Hence, it is first necessary to bound the weighted projection error (weighted with powers of h) by the weighted interpolation error. Then, the latter can be locally estimated on each element κ in the mesh. Unfortunately, this process assumes that the weighting factor inside the norm satisfies certain regularity assumptions, which in turn induce the restrictions on the mesh function h mentioned above.

In this paper we propose to define ϕ_h to be the L^2-projection of ϕ in time, but in space we shall define ϕ_h to be a *quasi-interpolant* of ϕ. This quasi-interpolation operator will be constructed in such a way that (1.1) will hold, along with optimal approximation results on arbitrarily nonuniform meshes.

In the final part of this paper, we consider the design and implementation of an adaptive algorithm, driven by our *a posteriori* error bound, for determining both the spatial and temporal mesh parameters in order to ensure global control of the discretisation error with respect to a user-defined tolerance. Numerical experiments indicate that while the norm of the residual of the underlying partial differential equation may be used to bound the error on a global basis (i.e., over the whole computational domain), the norm of the residual, calculated on an individual element, may give a poor estimate of the *local* error on that element. This is particularly evident when convection dominates diffusion, since the residual of the underlying

partial differential equation is dominated by the (direction dependent) hyperbolic part of the differential operator.

The outline of this paper is as follows. In Section 2, we state the model problem to be considered and formulate its Lagrange–Galerkin approximation. Then, in Section 3 we derive an *a posteriori* bound for the error in the norm of the function space $L^2(0,T;L^2(\Omega))$. Based on this error bound, in Section 4 we design an adaptive algorithm to ensure global control of the error with respect to a fixed tolerance. Then, in Section 5 we present the proof of the *a posteriori* error bound. Next, in Section 6 we present some numerical experiments to illustrate the performance of our adaptive strategy. Finally, in Section 7 we summarise the work presented in this paper and draw some conclusions.

Before we proceed, let us introduce some notation. Let $\mathbf{N}$ denote the set of positive integers, $\mathbf{N}_0$ the set of nonnegative integers and $\mathbf{R}$ the set of real numbers. Let ω be a bounded open subset of $\mathbf{R}^d$ ($d \in \mathbf{N}$) with a Lipschitz continuous boundary $\partial\omega$. We write $L^p(\omega)$, $1 \le p \le \infty$, to denote the usual Lebesgue space of real-valued functions with norm $\|\cdot\|_{L^p(\omega)}$. In the case $p = 2$, we denote the usual $L^2(\omega)$ inner product by $(\cdot,\cdot)_\omega$. In particular, for $\omega = \Omega$, where Ω will be specified later, we denote $\|\cdot\|_{L^2(\Omega)}$ by $\|\cdot\|$, and $(\cdot,\cdot)_\Omega$ by $(\cdot,\cdot)$. In the following we shall use the classical Sobolev spaces $W^{m,p}(\omega)$ and $W_0^{m,p}(\omega)$, $m \in \mathbf{N}_0$, $1 \le p \le \infty$, endowed with the norm $\|\cdot\|_{W^{m,p}(\omega)}$ and seminorm $|\cdot|_{W^{m,p}(\omega)}$. For $p = 2$, $W^{m,2}(\omega)$ will be denoted by $H^m(\omega)$ and we drop the subscript $p = 2$ in the corresponding norms and seminorms. The dual space of $H_0^m(\omega)$ will be denoted by $H^{-m}(\omega)$.

2. The Lagrange–Galerkin finite element method

Given a final time $T > 0$, we shall consider the following unsteady convection-diffusion problem. Given that $f \in L^2(0,T;H^{-1}(\Omega))$ and $u_0 \in L^2(\Omega)$, find u such that

$$u_t + \mathbf{a}\cdot\nabla u - \epsilon\Delta u = f, \qquad \mathbf{x} \in \Omega, \quad t \in (0,T], \tag{2.1a}$$

$$u(\mathbf{x},t) = 0, \qquad \mathbf{x} \in \partial\Omega,\ t \in [0,T], \tag{2.1b}$$

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{2.1c}$$

where Ω is a bounded convex polygonal domain in $\mathbf{R}^2$ with boundary $\partial\Omega$. Further, we assume that the diffusion coefficient $\epsilon > 0$, the velocity vector $\mathbf{a}$ lies in the function space $C([0,T];C_0^1(\bar\Omega)^2)$ and that $\mathbf{a}$ is incompressible, i.e., $\nabla\cdot\mathbf{a} = 0\ \forall\mathbf{x} \in \Omega$, $t \in (0,T)$.

Remark 2.1. The assumption that the velocity vector $\mathbf{a}$ is incompressible is not essential to the *a posteriori* error analysis presented in the proceeding sections. However, this restriction leads to "sharper" stability bounds for the corresponding dual or adjoint problem (cf. Section 5.3). In the case of compressible $\mathbf{a}$, such stability estimates have been derived for a system of convection-diffusion problems in [12].

The Lagrange–Galerkin method for (2.1) is based on combining the method of characteristics with the standard Galerkin finite element method (cf. Bercovier & Pironneau [3], Douglas & Russell [6] and Pironneau [19], for example). To define this method, let $0 = t_0 < t_1 < \cdots < t_M < t_{M+1} = T$ be a subdivision (not necessarily uniform) of $[0,T]$, with corresponding time intervals $I_n = (t_{n-1},t_n]$ and time steps $k_n = t_n - t_{n-1}$. For each n, $0 \le n \le M+1$, let $\mathcal{T}_n = \{\kappa\}$ be

an admissible subdivision of Ω into *closed* triangles κ, with corresponding mesh function h_n satisfying

$$c_1 h_\kappa^2 \leq \text{meas}(\kappa) \quad \forall \kappa \in \mathcal{T}_n, \tag{2.2a}$$

$$c_2 h_\kappa \leq h_n(\mathbf{x}) \leq h_\kappa \qquad \forall \mathbf{x} \in \kappa \ \forall \kappa \in \mathcal{T}_n, \tag{2.2b}$$

where $h_\kappa = \text{diam}(\kappa)$ and c_1 and c_2 are positive constants independent of h_n. Further, h is defined to be the global mesh function given by $h(\mathbf{x}, t) = h_n(\mathbf{x})$, for $(\mathbf{x}, t) \in \Omega \times I_n$ and we define the corresponding time step function $k = k(t)$ by $k(t) = k_n$ for $t \in I_n$.

For $n = 0, \ldots, M+1$, we associate with $\mathcal{T}_n$ the set $E_n = \{\tau\}$ consisting of those line segments in $\mathbf{R}^2$ which appear as an edge of some $\kappa \in \mathcal{T}_n$. We also denote by $E_{n,i}$, those τ in E_n which are interior to $\bar{\Omega}$ (i.e., not part of $\partial\Omega$).

Let $S_n = \Omega \times I_n$; for $r \in \mathbf{N}$ we define the following finite element spaces:

$$S_{h_n} = \{v \in C_0(\Omega) : \ v \text{ is a polynomial of degree at most } r \text{ on each } \kappa \text{ in } \mathcal{T}_n\},$$

$$V_{h_n} = \{v \in C(S_n) : \ v \text{ is constant in time and } v(\cdot, t) \in S_{h_n} \text{ for each } t \text{ in } I_n\},$$

$$V_h = \{v : v(\mathbf{x}, t)|_{S_n} \in V_{h_n}, \ n = 1, \ldots, M+1\}.$$

We note that if $v \in V_h$, then v is continuous in space at any time, but may be discontinuous in time at the discrete time levels t_n. To account for this, we introduce the notation $v_{n\pm} := \lim_{s \to 0^+} v(t_n \pm s)$ and $[v_n] := v_{n+} - v_{n-}$.

The construction of the Lagrange–Galerkin method involves writing problem (2.1) in a Lagrangian form. To this end, we define the particle trajectory (or characteristic curve), $\mathbf{X}(\mathbf{x}, s; \cdot)$ for $\mathbf{x} \in \bar{\Omega}$ and $s \in (0, T]$, as the solution of the initial value problem

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{X}(\mathbf{x}, s; t) = \mathbf{a}(\mathbf{X}(\mathbf{x}, s; t), t), \tag{2.3a}$$

$$\mathbf{X}(\mathbf{x}, s; s) = \mathbf{x}. \tag{2.3b}$$

Further, the *material derivative* $D_t u$ may be defined as

$$\begin{aligned} D_t u(\mathbf{x}, s) &:= \frac{\mathrm{d}}{\mathrm{d}t} u(\mathbf{X}(\mathbf{x}, s; t), t) \,|_{t=s} \\ &= \frac{\partial}{\partial t} u(\mathbf{x}, s) + \mathbf{a}(\mathbf{x}, s) \cdot \nabla u(\mathbf{x}, s) \quad \forall \mathbf{x} \in \Omega, \ s \in (0, T]. \end{aligned}$$

Hence, using the material derivative, equation (2.1a) may be rewritten in the following (weak) form. Find $u(t) \in V$, such that

$$\begin{aligned} (D_t u(\cdot, t), v) + (\epsilon \nabla u(\cdot, t), \nabla v) &= (f(\cdot, t), v) \quad \forall v \in V, \\ (u(\cdot, 0), v) &= (u_0(\cdot), v) \quad \forall v \in V, \end{aligned}$$

where $V = H_0^1(\Omega)$ and, for the sake of simplicity, we shall assume that $f \in C([0, T]; L^2(\Omega))$. The Lagrange–Galerkin time-discretisation involves approximating the material derivative by a divided difference operator. The simplest appropriate discretisation is the backward Euler method, giving for $n = 0, \ldots, M$:

$$\left(\frac{u(\cdot, t_{n+1}) - u(\mathbf{X}(\cdot, t_{n+1}; t_n), t_n)}{k_{n+1}}, v \right)$$

$$+ (\epsilon \nabla u(\cdot, t_{n+1}), \nabla v) \approx (f(\cdot, t_{n+1}), v) \quad \forall v \in V, \tag{2.5a}$$

$$(u(\cdot, 0), v) = (u_0(\cdot), v) \qquad \forall v \in V. \tag{2.5b}$$

If we now define u_h^n to be the Galerkin finite element approximation to $u(\cdot, t_n)$ at time t_n, then applying the finite element method to (2.5) yields the Lagrange–Galerkin discretisation of (2.1) as follows. Find $u_h^{n+1} \in S_{h_{n+1}}$ for $0 \le n \le M$ such that

$$\left(\frac{u_h^{n+1} - u_h^n(\mathbf{X}(\cdot, t_{n+1}; t_n))}{k_{n+1}}, v\right) + (\epsilon \nabla u_h^{n+1}, \nabla v) = (f_{n+1}, v) \quad \forall v \in S_{h_{n+1}}, \tag{2.6a}$$

$$(u_h^0, v) = (u_0, v) \qquad \forall v \in S_{h_0}, \tag{2.6b}$$

where $f_{n+1}(\cdot) := f(\cdot, t_{n+1})$.

Further, integrating (2.6a) with respect to t over I_{n+1}, we may write the Lagrange–Galerkin method in the following compact form. Find u_h such that, for $n = 0, 1, \ldots, M$, $u_h|_{S_{n+1}} \in V_{h_{n+1}}$ and satisfies

$$(D_t^h u_h, v)_{n+1} + (\epsilon \nabla u_h, \nabla v)_{n+1} = (\bar{f}, v)_{n+1} \quad \forall v \in V_{h_{n+1}}, \tag{2.7a}$$

$$(u_{h-}^0, v) = (u_0, v) \qquad \forall v \in S_{h_0}, \tag{2.7b}$$

where

$$D_t^h u_h|_{S_{n+1}} := (u_{h-}(\mathbf{x}, t_{n+1}) - u_{h-}(\mathbf{X}(\mathbf{x}, t_{n+1}; t_n), t_n))/k_{n+1},$$

$\bar{f}|_{S_{n+1}} := f(\cdot, t_{n+1})$, and for $v, w \in L^2(I_{n+1}; L^2(\Omega))$, we have used the notation

$$(v, w)_{n+1} := \int_{t_n}^{t_{n+1}} (v, w) \mathrm{d}t.$$

3. A posteriori error analysis

In this section we state an *a posteriori* bound for the error $e = u - u_h$, in the $L^2(0, T; L^2(\Omega))$ norm, where u and u_h are the solutions of (2.1) and (2.7), respectively. However, before we proceed, we first need to introduce some notation. For $v, w \in L^2(0, T; L^2(\Omega))$ and $Q := \Omega \times (0, T)$, we define

$$(v, w)_Q := \sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} (v, w) \mathrm{d}t, \qquad \|v\|_Q := \left((v, v)_Q\right)^{1/2}.$$

Given n, $0 \le n \le M + 1$, and $\tau \in E_{n,i}$, let $\mathbf{n}_\tau$ denote the unit normal to τ in the outward direction to κ, and define for $v \in S_{h_n}$,

$$\left[\frac{\partial v}{\partial \mathbf{n}_\tau}\right] = \lim_{s \to 0^+} (\nabla v(\mathbf{x} + s\mathbf{n}_\tau) - \nabla v(\mathbf{x} - s\mathbf{n}_\tau)) \cdot \mathbf{n}_\tau, \quad \mathbf{x} \in \tau;$$

that is, $[\partial v / \partial \mathbf{n}_\tau]$ is the jump across τ in the normal component of ∇v. Finally, we introduce the discrete second derivatives

$$D_h^2 v|_\kappa = \sum_{\tau \in \partial\kappa \cap E_{n,i}} \left\| \left[\frac{\partial v}{\partial \mathbf{n}_\tau}\right] \right\|_{L^\infty(\tau)} \frac{1}{h_\kappa}, \quad \kappa \in \mathcal{T}_n.$$

We may then state the following *a posteriori* error bound for problem (2.1), which we shall prove in Section 5.

Theorem 3.1. *Let u and u_h be solutions of* (2.1) *and* (2.7)*, respectively, and suppose that $\mathcal{T}_n$, $0 \le n \le M + 1$, satisfies* (2.2)*. Then*

$$\|e\|_Q \equiv \|u - u_h\|_Q \le \mathring{\mathcal{E}}(u_h, h, k, f), \tag{3.1}$$

where

$$\begin{aligned}
\overset{\circ}{\mathcal{E}}(u_h,h,k,f) &= \mathcal{E}(u_h,h,k,f)+\mathcal{E}_0(u_0,u_{h-}^0,h),\\
\mathcal{E}(u_h,h,k,f) &= C_1\|h^2R_1\|_Q+C_2\|kR_1\|_Q+C_3\|h^2R_2\|_Q\\
&\quad +C_4\|kR_3\|_Q+C_5\|kR_4\|_Q,\\
\mathcal{E}_0(u_0,u_{h-}^0,h) &= C_6\|u_0-u_{h-}^0\|,
\end{aligned}$$

and

$$\begin{aligned}
R_1|_\kappa &= [u_h^n]/k_{n+1}+\mathbf{a}\cdot\nabla u_h-\epsilon\Delta u_h-f,\ \textit{for}\ \kappa\in\mathcal{T}_{n+1},\\
R_2 &= D_h^2u_h,\\
R_3|_{S_{n+1}} &= (D_t^hu_h-\bar{f}-([u_h^n]/k_{n+1}+\mathbf{a}\cdot\nabla u_h-f))/k_{n+1},\\
R_4|_{S_{n+1}} &= [u_h^n]/k_{n+1}.
\end{aligned}$$

Furthermore, the C_i, $i=1,\ldots,6$, are (computable) positive constants; namely we have $C_1=C_{i,1}\sqrt{C_{s,2}}/\epsilon$, $C_2=C_{i,2}\sqrt{C_{s,3}}/\sqrt{2}$, $C_3=C_{i,1}C_t\sqrt{C_{s,2}}/(2c_2)$, $C_4=C_{i,2}\sqrt{C_{s,1}T}$, $C_5=\sqrt{C_{s,3}}$ and $C_6=\sqrt{C_{s,1}}$. Here, $C_{i,1}$ and $C_{i,2}$ are (quasi-) interpolation constants depending on c_1 and c_2 (see Lemma 5.1), C_t $(=4\sqrt{2}/c_1)$ is a trace constant (see Lemma 5.4) and $C_{s,1}$, $C_{s,2}$ and $C_{s,3}$ are stability constants of the corresponding dual problem (see subsection 5.3).

Remark 3.2. In general it may not be possible to calculate the particle trajectories $\mathbf{X}$ satisfying (2.3) exactly. Instead, an approximation $\tilde{\mathbf{X}}$ to $\mathbf{X}$ is computed; the Lagrange–Galerkin method will then be defined as in (2.7) with $D_t^hu_h$ replaced by $\tilde{D}_t^hu_h$, where

$$\tilde{D}_t^hu_h|_{S_{n+1}}:=(u_{h-}(\mathbf{x},t_{n+1})-u_{h-}(\tilde{\mathbf{X}}(\mathbf{x},t_{n+1};t_n),t_n))/k_{n+1},$$

for $n=0,\ldots,M$. Thus, the *a posteriori* error bound stated in Theorem 3.1 will still hold with the residual term R_3 defined analogously with $D_t^hu_h$ replaced by $\tilde{D}_t^hu_h$. To ensure that the term $\|kR_3\|_Q$ in the *a posteriori* error bound (3.1) converges to zero as h and k tend to zero, we assume that the approximate particle trajectories $\tilde{\mathbf{X}}$ are constructed so that they converge to $\mathbf{X}$ as the space-time mesh is refined. For example, the Runge–Kutta method of order four could be used in the calculation of $\tilde{\mathbf{X}}$.

The next section describes the implementation of this *a posteriori* error bound into an adaptive finite element algorithm.

4. Adaptive algorithm

For a given tolerance TOL, we now consider the problem of finding a discretisation in space and time $\mathcal{S}_h=\{(\mathcal{T}_n,t_n)\}_{n\geq 0}$ such that:

1. $\|u-u_h\|_Q\leq \mathrm{TOL}$;
2. $\mathcal{S}_h$ is optimal in the sense that the number of degrees of freedom is minimal.

In order to satisfy these criteria we shall use the *a posteriori* error bound (3.1) to choose $\mathcal{S}_h$ such that:

1. $\overset{\circ}{\mathcal{E}}(u_h,h,k,f)\leq \mathrm{TOL}$;
2. The number of degrees of freedom of $\mathcal{S}_h$ is minimal.

The term $\mathcal{E}_0(u_0, u^0_{h-}, h)$ is easily controlled at the start of a computation, so here we shall only consider the problem of constructing $\mathcal{S}_h$ in an efficient way to ensure that

$$\mathcal{E}(u_h, h, k, f) \le \mathrm{TOL}',$$

where $\mathrm{TOL} = \mathrm{TOL}' + \mathcal{E}_0(u_0, u^0_{h-}, h)$. To solve this problem, we first write $\mathcal{E}$ symbolically in terms of two residual terms: one that controls the spatial mesh and one that controls the temporal mesh, i.e., let

$$\mathcal{E}(u_h, h, k, f) \equiv C_1' \|h^2 R_1'\|_Q + C_2' \|k R_2'\|_Q.$$

Similarly, we split the tolerance TOL' into a "spatial part", TOL_h, and a "temporal part", TOL_k. Thus, for reliability we now require that the following two conditions hold:

$$C_1' \|h^2 R_1'\|_Q \le \mathrm{TOL}_h, \tag{4.1}$$

$$C_2' \|k R_2'\|_Q \le \mathrm{TOL}_k. \tag{4.2}$$

To design the space-time mesh $\mathcal{S}_h$ at each time level t_n, we "split up" the norm in (4.1) over each element $\kappa \in \mathcal{T}_n$, and the norm in (4.2) over the domain Ω at time t_n. We do this as follows:

$$\begin{aligned} C_1' \|h^2 R_1'\|_Q &\le C_1' \sqrt{T} \max_{1\le n\le M+1} \|h_n^2 R_1'(u_h^n)\| \\ &\le C_1' \sqrt{N_n T} \max_{1\le n\le M+1} \left(\max_{\kappa\in\mathcal{T}_n} \|h_n^2 R_1'(u_h^n)\|_{L^2(\kappa)} \right), \\ C_2' \|k R_2'\|_Q &\le C_2' \sqrt{T} \max_{1\le n\le M+1} \|k_n R_2'(u_h^n)\|, \end{aligned}$$

where N_n is the number of elements in the spatial mesh at time t_n. Thus, if

$$\begin{aligned} C_1' \sqrt{N_n T}\, \|h_n^2 R_1'(u_h^n)\|_{L^2(\kappa)} &\le \mathrm{TOL}_h \quad \forall \kappa \in \mathcal{T}_n, \text{ for } n = 1, \dots, M+1, \\ C_2' \sqrt{T}\, \|k_n R_2'(u_h^n)\| &\le \mathrm{TOL}_k, \quad \text{for } n = 1, \dots, M+1, \end{aligned}$$

are satisfied, then (4.1) and (4.2) will automatically hold, cf. [23].

For the practical implementation of this method, we consider the following adaptive algorithm for choosing $\mathcal{S}_h$, assuming that the final time T is fixed: for each $n = 1, 2, \dots, M+1$, with $\mathcal{T}_{n,0}$ a given initial mesh and $k_{n,0}$ an initial time step, determine meshes $\mathcal{T}_{n,j}$ with $N_{n,j}$ elements of size $h_{n,j}(\mathbf{x})$ and time steps $k_{n,j}$ and corresponding approximate solution $u^n_{h,j}$ defined on $I_{n,j}$ such that, for $j = 0, 1, \dots, \hat{n}-1$,

(4.3a)

$$C_1 \|h^2_{n,j+1} R_1(u^n_{h,j})\|_{L^2(\kappa)} + C_3 \|h^2_{n,j+1} R_2(u^n_{h,j})\|_{L^2(\kappa)} = \frac{\mathrm{TOL}_h}{\sqrt{N_{n,j} T}} \quad \forall \kappa \in \mathcal{T}_{n,j},$$

$$\begin{aligned} C_2 \|k_{n,j+1} R_1(u^n_{h,j})\| + C_4 \|k_{n,j+1} R_3(u^n_{h,j})\| \\ + C_5 \|k_{n,j+1} R_4(u^n_{h,j})\| = \frac{\mathrm{TOL}_k}{\sqrt{T}}, \end{aligned} \tag{4.3b}$$

where $I_{n,j} = (t_{n-1}, t_{n-1} + k_{n,j}]$ and $\mathrm{TOL}' = \mathrm{TOL}_h + \mathrm{TOL}_k$. We define $\mathcal{T}_n = \mathcal{T}_{n,\hat{n}}$, $k_n = k_{n,\hat{n}}$ and $h_n = h_{n,\hat{n}}$, where for each n, the number of trials $\hat{n}$ is the smallest

integer such that for $j = \hat{n}$, the stopping condition

$$C_1\|h^2_{n,\hat{n}}R_1(u^n_{h,\hat{n}})\|_{L^2(\kappa)} + C_3\|h^2_{n,\hat{n}}R_2(u^n_{h,\hat{n}})\|_{L^2(\kappa)} \leq \frac{\mathrm{TOL}_h}{\sqrt{N_{n,\hat{n}}T}} \quad \forall \kappa \in \mathcal{T}_{n,\hat{n}}, \tag{4.4a}$$

$$C_2\|k_{n,\hat{n}}R_1(u^n_{h,\hat{n}})\| + C_4\|k_{n,\hat{n}}R_3(u^n_{h,\hat{n}})\| + C_5\|k_{n,\hat{n}}R_4(u^n_{h,\hat{n}})\| \leq \frac{\mathrm{TOL}_k}{\sqrt{T}}, \tag{4.4b}$$

is satisfied.

Remark 4.1.

a) By construction, the stopping condition (4.4) will guarantee reliability of the adaptive algorithm. For efficiency, we try to ensure that (4.4) is satisfied with near equality.
b) We should note that because we assume that the final time T is fixed, the time step given by (4.3b) may need to be limited to ensure that $t_M + k_{M+1,\hat{n}} = T$.
c) For the implementation of this adaptive algorithm, we shall assume that $\mathcal{T}_{n,0} = \mathcal{T}_{n-1}$ for $n = 1, 2, \ldots$.
d) The multiplicative factors of $\sqrt{T}$ arising in (4.3) are introduced by switching from the L^2 norm to the L^∞ norm in time. This is not an unrealistic growth in time since the same factor is observed in the case of the ordinary differential equation analogue of a constant coefficient hyperbolic problem with purely imaginary eigenvalues, cf. Eriksson *et al.* [9, p. 137].

5. Proof of the a posteriori error estimate

In this section we present the proof of the *a posteriori* error bound stated in Theorem 3.1. The proof is based on the general theoretical framework of *a posteriori* error estimation developed by Eriksson & Johnson [7, 8], Eriksson *et al.* [9] and Hansbo & Johnson [11], for example.

The basic structure of the proof of the *a posteriori* error bound is as follows:

1. Representation of the error in terms of the residual of the finite element approximation and the solution of the dual problem;
2. Use of Galerkin orthogonality;
3. Local interpolation (projection) error estimates for the solution of the dual problem;
4. Strong stability estimates for the dual problem.

5.1. **Error representation formula.** The (backward) dual or adjoint problem takes the following form. Find ϕ such that

$$-\phi_t - \nabla\cdot(\mathbf{a}\phi) - \epsilon\Delta\phi = e \equiv u - u_h, \quad \mathbf{x} \in \Omega, \quad t \in [0,T), \tag{5.1a}$$

$$\phi(\mathbf{x},t) = 0, \qquad \mathbf{x} \in \partial\Omega, t \in [0,T], \tag{5.1b}$$

$$\phi(\mathbf{x},T) = 0, \qquad \mathbf{x} \in \Omega. \tag{5.1c}$$

We note that for Ω convex, problem (5.1) admits a unique solution ϕ which lies in the function space $H^1(0,T;L^2(\Omega)) \cap L^2(0,T;H^1_0(\Omega)\cap H^2(\Omega))$, cf. [12].

We shall now proceed to prove an error representation formula. Multiplying (5.1a) by e and integrating by parts in both space and time, we get

$$\begin{aligned}\|e\|_Q^2 &= (e, -\phi_t - \nabla\cdot(\mathbf{a}\phi) - \epsilon\Delta\phi)_Q \\ &= (e_t + \mathbf{a}\cdot\nabla e, \phi)_Q + (\epsilon\nabla e, \nabla\phi)_Q - \sum_{n=0}^{M}([u_h^n], \phi(t_n)) + (u_0 - u_{h-}^0, \phi(0)) \\ &= (f - \mathbf{a}\cdot\nabla u_h, \phi)_Q - (\epsilon\nabla u_h, \nabla\phi)_Q - \sum_{n=0}^{M}([u_h^n], \phi(t_n)) + (u_0 - u_{h-}^0, \phi(0)),\end{aligned}$$

where we have used (2.1a). If we now let $\phi_h \in V_h$, then using (2.7) we have

$$\begin{aligned}\|e\|_Q^2 = (f - \mathbf{a}\cdot\nabla u_h, \phi)_Q + (\epsilon\nabla u_h, \nabla(\phi_h - \phi))_Q + \sum_{n=0}^{M}\int_{t_n}^{t_{n+1}}(D_t^h u_h - \bar{f}, \phi_h)\mathrm{d}t \\ - \sum_{n=0}^{M}([u_h^n], \phi(t_n)) + (u_0 - u_{h-}^0, \phi(0)) \quad \forall\phi_h \in V_h. \qquad (5.2)\end{aligned}$$

Finally, we rearrange the terms on the right-hand side of (5.2) in order to ensure that each expression appearing in the error representation formula is of the correct asymptotic form, thereby guaranteeing that the final *a posteriori* error bound is sharp. To this end, we get the following error representation formula:

$$\begin{aligned}\|e\|_Q^2 &= \sum_{n=0}^{M}\int_{t_n}^{t_{n+1}}\sum_{\kappa\in\mathcal{T}_{n+1}}([u_h^n]/k_{n+1} + \mathbf{a}\cdot\nabla u_h - \epsilon\Delta u_h - f, \phi_h - \phi)_\kappa\,\mathrm{d}t \\ &\quad + \sum_{n=0}^{M}\int_{t_n}^{t_{n+1}}\left(\sum_{\kappa\in\mathcal{T}_{n+1}}(\epsilon\Delta u_h, \phi_h - \phi)_\kappa + (\epsilon\nabla u_h, \nabla(\phi_h - \phi))\right)\mathrm{d}t \\ &\quad + \sum_{n=0}^{M}\int_{t_n}^{t_{n+1}}(D_t^h u_h - \bar{f} - ([u_h^n]/k_{n+1} + \mathbf{a}\cdot\nabla u_h - f), \phi_h)\mathrm{d}t \\ &\quad + \sum_{n=0}^{M}\int_{t_n}^{t_{n+1}}([u_h^n]/k_{n+1}, \phi - \phi(t_n))\mathrm{d}t + (u_0 - u_{h-}^0, \phi(0)) \\ &\equiv \mathrm{I} + \mathrm{II} + \mathrm{III} + \mathrm{IV} + \mathrm{V} \quad \forall\phi_h \in V_h. \qquad (5.3)\end{aligned}$$

So far, ϕ_h has been an arbitrary element of V_h. In the next section we make a specific choice of ϕ_h as the interpolant/projection of the dual solution ϕ.

5.2. **Interpolation/projection estimates for the dual problem.** In this section we shall define $\phi_h \in V_h$ in (5.3) to be the quasi-interpolant of ϕ in space and the L^2-projection of ϕ in time. However, before we proceed let us first describe how the quasi-interpolation operator is constructed.

Here, we consider the construction of the quasi-interpolation operator $\mathcal{I}$ based on a modification of the generalised interpolation operators developed by Scott & Zhang [21] and Brenner & Scott [4], Section 4.8; see also Verfürth [24]. The nodal values will be defined by locally averaging the function over an element κ. However, these nodal values will be modified in order to fit homogeneous boundary conditions; this modification is employed in order to ensure that $\mathcal{I}$ is bounded in L^2, cf. Lemma 5.1 below. For further details, see [12, 17].

For a given n, $0 \leq n \leq M+1$, we have the triangulation $\mathcal{T}_n$ of Ω; for simplicity let us drop the subscript/superscript "n", and simply denote $\mathcal{T}_n$ by $\mathcal{T}$. We let $\mathcal{N}_h = \{\mathbf{a}_i\}_{i=1}^L$ denote the set of interpolation nodes of $\mathcal{T}$ and $\{\phi_i\}_{i=1}^L$ denote the set of nodal basis functions of S_h (i.e., S_{h_n}).

For each node $\mathbf{a}_i \in \mathcal{N}_h$ we choose an element κ (recall that κ is closed) such that $\mathbf{a}_i \in \kappa$, and we let $\sigma_i = \kappa$. We note that there may be many such element domains, but we pick just one. Let us denote by n_1 the dimension of $\mathbf{P}_r(\sigma_i)$. Further, let $\mathbf{a}_{i,1} = \mathbf{a}_i$ and $\{\mathbf{a}_{i,j}\}_{j=1}^{n_1}$ be the set of nodal points in σ_i. For the nodal basis $\{\phi_{i,j}\}_{j=1}^{n_1}$ for σ_i, we have the corresponding $L^2(\sigma_i)$-dual basis $\{\psi_{i,j}\}_{j=1}^{n_1}$ defined by

$$\int_{\sigma_i} \psi_{i,j}(\mathbf{x})\phi_{i,k}(\mathbf{x})\mathrm{d}\mathbf{x} = \delta_{jk}, \quad \text{for } j,k = 1,2,\ldots,n_1,$$

where δ_{jk} is the Kronecker delta. To simplify notation, we let

$$\psi_i = \psi_{i,1} \quad \forall \mathbf{a}_i \in \mathcal{N}_h.$$

Hence, it follows that for any nodal basis function ϕ_j of S_h, we have

$$\int_{\sigma_i} \psi_i(\mathbf{x})\phi_j(\mathbf{x})\mathrm{d}\mathbf{x} = \delta_{ij}, \quad \text{for } i,j = 1,2,\ldots,L.$$

We now define the quasi-interpolation operator $\mathcal{I} : L^1(\Omega) \to S_h$ by

$$\mathcal{I}v(\mathbf{x}) = \sum_{i=1}^{L} \mathcal{I}v(\mathbf{a}_i)\phi_i(\mathbf{x}), \tag{5.4}$$

where

$$\mathcal{I}v(\mathbf{a}_i) = \chi_\Omega(\mathbf{a}_i) \int_{\sigma_i} \psi_i(\mathbf{y})v(\mathbf{y})\mathrm{d}\mathbf{y},$$

and χ_Ω is the characteristic function for Ω, i.e., if $\mathbf{a}_i \in \partial\Omega$ then $\mathcal{I}v(\mathbf{a}_i) = 0$.

Following the ideas of [21], the quasi-interpolation operator $\mathcal{I}$ can be shown to be a projection from $L^1(\Omega)$ to S_h, cf. [12, 17]. Furthermore, $\mathcal{I}$ can be shown to satisfy the following optimal approximation property and stability estimate (see [12, 17] for details).

Lemma 5.1. *Given that $v \in H_0^1(\Omega) \cap H^s(\Omega)$, $1 \leq s \leq r+1$, and $\mathcal{T}$ satisfies conditions* (2.2)*, there exist positive constants $C_{i,1}$ and $C_{i,2}$, independent of h, such that*

$$\|h^{-s}(v - \mathcal{I}v)\| + |h^{1-s}(v - \mathcal{I}v)|_{H^1(\Omega)} \leq C_{i,1}|v|_{H^s(\Omega)},$$
$$\|\mathcal{I}v\| \leq C_{i,2}\|v\|.$$

Remark 5.2. We note that the interpolation constants $C_{i,1}$ and $C_{i,2}$ arising in Lemma 5.1 depend on the mesh regularity constant c_1 defined by (2.2a).

We shall now proceed to define $\phi_h \in V_h$; here, we now return to the subscript/superscript "n" notation to distinguish between different time levels t_n. Let us first define the operators

$$\mathcal{I}_n : L^1(\Omega) \to S_{h_n}, \quad \pi_n : L^2(I_n) \to \mathbf{P}_0(I_n),$$

in space and in time, by (5.4) and

$$\int_{t_{n-1}}^{t_n} (\pi_n\phi - \phi)v \,\mathrm{d}t = 0 \quad \forall v \in \mathbf{P}_0(I_n), \tag{5.5}$$

respectively. Then, we can define (locally) $\phi_h|_{S_n} \in V_{h_n}$ by letting

$$\phi_h|_{S_n} = \mathcal{I}_n \pi_n \phi = \pi_n \mathcal{I}_n \phi \in V_{h_n},$$

where $\phi = \phi|_{S_n}$. Further, if we introduce $\mathcal{I}$ and π by

$$(\mathcal{I}\phi)|_{S_n} = \mathcal{I}_n(\phi|_{S_n}), \tag{5.6a}$$

$$(\pi\phi)|_{S_n} = \pi_n(\phi|_{S_n}), \tag{5.6b}$$

then we let $\phi_h \in V_h$ be

$$\phi_h = \mathcal{I}\pi\phi = \pi\mathcal{I}\phi \in V_h. \tag{5.7}$$

It follows from this definition (using Lemma 5.1) that

$$\|\phi_h\|_Q \le C_{i,2}\|\phi\|_Q. \tag{5.8}$$

We next provide error estimates for the operators $\mathcal{I}$ and π in order to estimate $\phi - \phi_h = \phi - \mathcal{I}\pi\phi$. However, let us first give the following stability result (see [10, p.43, Theorem 2.2.1] and [12, 15]) and trace lemma (see [12, 15]).

Lemma 5.3. *Given that Ω is a bounded convex polygonal domain in $\mathbf{R}^2$, then*

$$|v|_{H^2(\Omega)} \le \|\Delta v\| \quad \forall v \in H_0^1(\Omega) \cap H^2(\Omega).$$

Lemma 5.4. *Given n, $0 \le n \le M+1$, let $\kappa \in \mathcal{T}_n$, where $\mathcal{T}_n$ satisfies conditions (2.2). If $v \in W^{1,1}(\kappa)$, then there exists a positive constant C_t such that*

$$\int_\tau |v| \mathrm{d}s \le C_t \left(\int_\kappa |\nabla v| \mathrm{d}\mathbf{x} + h_\kappa^{-1} \int_\kappa |v| \mathrm{d}\mathbf{x} \right), \quad \tau \subset \partial\kappa \ \forall \kappa \in \mathcal{T}_n,$$

where $C_t = 4\sqrt{2}/c_1$.

Lemma 5.5. *Suppose that $R \in L^2(0,T;L^2(\Omega))$ and $v \in V_h$ then*

$$|(R, \mathcal{I}\phi - \phi)_Q| \le C_{p,1}\|h^2 R\|_Q \|\Delta\phi\|_Q, \tag{5.9a}$$

$$\left| \sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} \left(\sum_{\kappa \in \mathcal{T}_{n+1}} (\epsilon\Delta v, \mathcal{I}_{n+1}\phi - \phi)_\kappa + (\epsilon\nabla v, \nabla(\mathcal{I}_{n+1}\phi - \phi)) \right) \mathrm{d}t \right|$$

$$\le C_{p,2}\|h^2 D_h^2 v\|_Q \|\epsilon\Delta\phi\|_Q, \tag{5.9b}$$

where $C_{p,1} = C_{i,1}$, $C_{p,2} = C_{i,1}C_t/(2c_2)$ and $C_t = 4\sqrt{2}/c_1$.

Proof. First, we consider (5.9a): using the Cauchy–Schwarz inequality, we get

$$|(R, \mathcal{I}\phi - \phi)_Q| \le \|h^2 R\|_Q \|h^{-2}(\mathcal{I}\phi - \phi)\|_Q. \tag{5.10}$$

Using Lemma 5.1 and Lemma 5.3, we have

$$\|h^{-2}(\mathcal{I}\phi - \phi)\|_Q = \left(\sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} \|h_{n+1}^{-2}(\mathcal{I}_{n+1}\phi - \phi)\|^2 \mathrm{d}t \right)^{1/2}$$

$$\le C_{i,1} \left(\int_0^T |\phi(t)|_{H^2(\Omega)}^2 \mathrm{d}t \right)^{1/2} \le C_{i,1}\|\Delta\phi\|_Q. \tag{5.11}$$

Substituting (5.11) into (5.10) gives the desired result.

Next, we consider (5.9b). First, let $\mathcal{A}$ denote the left-hand side of (5.9b) inside the modulus signs and let $\rho = \mathcal{I}\phi - \phi$ and $\rho_{n+1} = \mathcal{I}_{n+1}\phi - \phi$. Then, by integrating by parts in space, we have

$$\begin{aligned}\mathcal{A} &= \sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} \left(\sum_{\kappa \in \mathcal{T}_{n+1}} \left(\int_{\partial\kappa} \epsilon \frac{\partial v}{\partial \mathbf{n}_\kappa} \rho_{n+1} \, \mathrm{d}s \right) \right) \mathrm{d}t \\ &= \sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} \left(\sum_{\kappa \in \mathcal{T}_{n+1}} \left(\sum_{\tau \in \partial\kappa \cap E_{n+1,i}} \frac{\epsilon}{2} \int_\tau \left[\frac{\partial v}{\partial \mathbf{n}_\tau} \right] \rho_{n+1} \, \mathrm{d}s \right) \right) \mathrm{d}t.\end{aligned}$$

Using Hölder's inequality and Lemma 5.4, we have

$$\begin{aligned}|\mathcal{A}| &\le \sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} \left(\sum_{\kappa \in \mathcal{T}_{n+1}} \left(\sum_{\tau \in \partial\kappa \cap E_{n+1,i}} \frac{\epsilon}{2} \left\| \left[\frac{\partial v}{\partial \mathbf{n}_\tau} \right] \right\|_{L^\infty(\tau)} \int_\tau |\rho_{n+1}| \mathrm{d}s \right) \right) \mathrm{d}t \\ &\le \sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} \left(\sum_{\kappa \in \mathcal{T}_{n+1}} C_t \frac{\epsilon}{2} D_h^2 v \int_\kappa (h_\kappa |\nabla \rho_{n+1}| + |\rho_{n+1}|) \, \mathrm{d}\mathbf{x} \right) \mathrm{d}t.\end{aligned}$$

Further, using (2.2) and the Cauchy–Schwarz inequality we have

$$|\mathcal{A}| \le \frac{1}{2} C_t c_2^{-1} \|h^2 D_h^2 v\|_Q \|\epsilon(h^{-1}|\nabla\rho| + h^{-2}|\rho|)\|_Q. \tag{5.12}$$

Let us now consider the second term on the right-hand side of (5.12). Using the triangle inequality, Lemma 5.1 and Lemma 5.3, we get

$$\begin{aligned}&\|\epsilon(h^{-1}|\nabla\rho| + h^{-2}|\rho|)\|_Q \\ &\quad\le \epsilon \left(\sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} \left(\|h_{n+1}^{-1} \nabla(\mathcal{I}_{n+1}\phi - \phi)\| + \|h_{n+1}^{-2}(\mathcal{I}_{n+1}\phi - \phi)\| \right)^2 \mathrm{d}t \right)^{1/2} \\ &\quad\le C_{i,1}\epsilon \left(\int_0^T |\phi(t)|_{H^2(\Omega)}^2 \mathrm{d}t \right)^{1/2} \le C_{i,1} \|\epsilon \Delta\phi\|_Q. \qquad (5.13)\end{aligned}$$

Substituting (5.13) into (5.12) completes the proof of the lemma. □

Before presenting the next lemma, we introduce the following space:

$$W_h = \left\{ v : v(\mathbf{x}, t)|_{S_n} = v(\mathbf{x}),\ v \in L^2(\Omega),\ \text{for } n = 1, \ldots, M+1 \right\},$$

i.e., W_h consists of those functions v that are piecewise constant in time and square integrable in space.

Lemma 5.6. *Suppose that $R \in L^2(0,T;L^2(\Omega))$. Then*

$$|(R, \mathcal{I}(\pi\phi - \phi))_Q| \le C_{p,3} \|kR\|_Q \|\phi_t\|_Q, \tag{5.14a}$$

$$\left| \sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} (R, \phi_n - \phi) \mathrm{d}t \right| \le \|kR\|_Q \|\phi_t\|_Q. \tag{5.14b}$$

Moreover, for any $w \in W_h$ and any $\mathbf{w} \in (W_h)^2$,

$$(w, \mathcal{I}(\pi\phi - \phi))_Q = 0, \tag{5.15a}$$

$$(\mathbf{w}, \nabla\mathcal{I}(\pi\phi - \phi))_Q = 0, \tag{5.15b}$$

where $C_{p,3} = C_{i,2}/\sqrt{2}$ and $\phi_n = \phi(\mathbf{x}, t_n)$.

Proof. First, we consider (5.14a). Using the Cauchy–Schwarz inequality, we get

$$|(R, \mathcal{I}(\pi\phi - \phi))_Q| \le \|kR\|_Q \|k^{-1}(\pi(\mathcal{I}\phi) - (\mathcal{I}\phi))\|_Q.$$

By reversing the order of integration, we have

$$\|k^{-1}(\pi(\mathcal{I}\phi) - (\mathcal{I}\phi))\|_Q = \left(\int_\Omega \sum_{n=0}^{M} k_{n+1}^{-2} \|\pi_{n+1}(\mathcal{I}_{n+1}\phi) - (\mathcal{I}_{n+1}\phi)\|^2_{L^2(I_{n+1})} \mathrm{d}\mathbf{x} \right)^{1/2}. \tag{5.16}$$

If we denote the piecewise constant interpolant of v on I_{n+1} evaluated at the point $(t_{n+1} + t_n)/2$ by $\mathcal{I}_{n+1,I}v$, then using the fact that $\pi_{n+1}v$ is the L^2-projection of v onto the set of piecewise constant functions, we have

$$\begin{aligned} \|\pi_{n+1}(\mathcal{I}_{n+1}\phi) - (\mathcal{I}_{n+1}\phi)\|_{L^2(I_{n+1})} &\le \|\mathcal{I}_{n+1,I}(\mathcal{I}_{n+1}\phi) - (\mathcal{I}_{n+1}\phi)\|_{L^2(I_{n+1})} \\ &\le \left(1/\sqrt{2}\right) k_{n+1} \|(\mathcal{I}_{n+1}\phi)_t\|_{L^2(I_{n+1})} \\ &= \left(1/\sqrt{2}\right) k_{n+1} \|\mathcal{I}_{n+1}\phi_t\|_{L^2(I_{n+1})}. \end{aligned} \tag{5.17}$$

Substituting (5.17) into (5.16), reversing the order of integration and applying Lemma 5.1 gives the desired result.

The proof of (5.14b) follows similarly. Here, we use the approximation properties of an interpolation operator defined at the point t_n, rather than at the midpoint of the interval I_{n+1}.

Next, we consider (5.15a):

$$(w, \mathcal{I}(\pi\phi - \phi))_Q = \sum_{n=0}^{M} \left(w_{n+1}, \mathcal{I}_{n+1} \left(\int_{t_n}^{t_{n+1}} (\pi_{n+1}\phi - \phi) \mathrm{d}t \right) \right) = 0,$$

using the definition of π_{n+1} in (5.5). Here, we have used the notation $w|_{S_{n+1}} = w_{n+1}$ for $w \in W_h$. The proof of (5.15b) follows similarly. □

5.3. **Strong stability of the dual problem.** In this section we derive strong stability estimates for the dual problem (5.1) with the aim to provide bounds on the norms of the dual solution ϕ appearing in the inequalities in Lemmas 5.5 and 5.6.

Lemma 5.7. *Let ϕ be the solution of* (5.1)*. Then there is a constant $C_{s,1}(T, \Omega, \epsilon)$ such that*

$$\|\phi\|^2_{L^\infty(0,T;L^2(\Omega))} + \|\epsilon^{1/2}\nabla\phi\|^2_Q \le C_{s,1}\|e\|^2_Q,$$

where $C_{s,1} = 2\min\{c_*^2/\epsilon, \mathrm{e}^T\}$ and $c_* = c_*(\Omega)$ is the constant in the Poincaré inequality

$$\|w\| \le c_*\|\nabla w\| \quad \forall w \in H_0^1(\Omega);$$

namely c_* is the square-root of the reciprocal of the smallest eigenvalue of $-\Delta$ on Ω, subject to homogeneous Dirichlet boundary conditions on $\partial\Omega$.

Proof. Multiply (5.1a) by ϕ and integrate over Ω to obtain

$$-\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\phi(t)\|^2 - \frac{1}{2}(\nabla \cdot \mathbf{a}(t)\phi(t), \phi(t)) + \|\epsilon^{1/2}\nabla\phi(t)\|^2 = (e(t), \phi(t)).$$

Using the incompressibility condition and the Cauchy–Schwarz inequality, we have

$$-\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\phi(t)\|^2+\|\epsilon^{1/2}\nabla\phi(t)\|^2\leq\|e(t)\|\|\phi(t)\|\leq\frac{1}{2}\|e(t)\|^2+\frac{1}{2}\|\phi(t)\|^2. \tag{5.18}$$

Now, integrating with respect to time over the interval (t,T) and using (5.1c), we get

$$\|\phi(t)\|^2+2\int_t^T\|\epsilon^{1/2}\nabla\phi(s)\|^2\mathrm{d}s\leq\|e\|_Q^2+\int_t^T\|\phi(s)\|^2\mathrm{d}s,$$

and by applying Gronwall's lemma, we have

$$\|\phi\|^2_{L^\infty(0,T;L^2(\Omega))}+2\|\epsilon^{1/2}\nabla\phi\|_Q^2\leq 2\mathrm{e}^T\|e\|_Q^2.$$

This proves part of the lemma. In order to get an error constant that does not grow exponentially in time, we use the Poincaré inequality for ϕ in (5.18) as follows:

$$-\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\phi(t)\|^2+\|\epsilon^{1/2}\nabla\phi(t)\|^2\leq\frac{c_*^2}{2\epsilon}\|e(t)\|^2+\frac{1}{2}\|\epsilon^{1/2}\nabla\phi(t))\|^2,$$

where $c_*=c_*(\Omega)$. Hence, we have

$$-\frac{\mathrm{d}}{\mathrm{d}t}\|\phi(t)\|^2+\|\epsilon^{1/2}\nabla\phi(t)\|^2\leq\frac{c_*^2}{\epsilon}\|e(t)\|^2. \tag{5.19}$$

We now integrate (5.19) with respect to time to obtain the desired result. □

Lemma 5.8. *Let ϕ be the solution of* (5.1). *Then there is a constant $C_{s,2}(T,\Omega,\mathbf{a},\epsilon)$ such that*

$$\|\epsilon^{1/2}\nabla\phi\|^2_{L^\infty(0,T;L^2(\Omega))}+\|\epsilon\Delta\phi\|_Q^2\leq C_{s,2}\|e\|_Q^2,$$

where $C_{s,2}=4\min\left\{\exp\left(2\|\mathbf{a}\|^2_{L^2(0,T;L^\infty(\Omega))}/\epsilon\right),\left(1+C_{s,1}\|\mathbf{a}\|^2_{L^\infty(0,T;L^\infty(\Omega))}/\epsilon\right)\right\}$ *and $C_{s,1}$ is as defined in Lemma 5.7.*

Proof. Multiply (5.1a) by $-\epsilon\Delta\phi$ and integrate over Ω to obtain

$$-\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\epsilon^{1/2}\nabla\phi(t)\|^2+\|\epsilon\Delta\phi(t)\|^2=-(e(t)+\nabla\cdot(\mathbf{a}(t)\phi(t)),\epsilon\Delta\phi(t)).$$

Using the incompressibility condition and the Cauchy–Schwarz inequality, we have

$$-\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\epsilon^{1/2}\nabla\phi(t)\|^2+\|\epsilon\Delta\phi(t)\|^2\leq\frac{1}{2}\|e(t)+\mathbf{a}(t)\cdot\nabla\phi(t)\|^2+\frac{1}{2}\|\epsilon\Delta\phi(t)\|^2.$$

If we now apply the triangle inequality and Hölder's inequality, we have

$$-\frac{\mathrm{d}}{\mathrm{d}t}\|\epsilon^{1/2}\nabla\phi(t)\|^2+\|\epsilon\Delta\phi(t)\|^2\leq 2\|e(t)\|^2+\frac{2}{\epsilon}\|\mathbf{a}(t)\|^2_{L^\infty(\Omega)}\|\epsilon^{1/2}\nabla\phi(t)\|^2.$$

Now, integrating with respect to time over the interval (t,T), we get

$$\|\epsilon^{1/2}\nabla\phi(t)\|^2+\int_t^T\|\epsilon\Delta\phi(s)\|^2\mathrm{d}s \leq 2\|e\|_Q^2+\frac{2}{\epsilon}\int_t^T\|\mathbf{a}(s)\|^2_{L^\infty(\Omega)}\|\epsilon^{1/2}\nabla\phi(s)\|^2\mathrm{d}s,$$

and by applying Gronwall's lemma, we have

$$\|\epsilon^{1/2}\nabla\phi\|^2_{L^\infty(0,T;L^2(\Omega))}+\|\epsilon\Delta\phi\|_Q^2\leq 4\|e\|_Q^2\mathrm{e}^{(2/\epsilon)\|\mathbf{a}\|^2_{L^2(0,T;L^\infty(\Omega))}}. \tag{5.20}$$

Alternatively, using Hölder's inequality and Lemma 5.7, gives

$$\|\epsilon^{1/2}\nabla\phi\|^2_{L^\infty(0,T;L^2(\Omega))} + \|\epsilon\Delta\phi\|^2_Q \le 4\left(1 + C_{s,1}\|\mathbf{a}\|^2_{L^\infty(0,T;L^\infty(\Omega))}/\epsilon\right)\|e\|^2_Q. \tag{5.21}$$

The lemma now follows from (5.20) and (5.21). □

Lemma 5.9. *Let ϕ be the solution of* (5.1). *Then there is a constant $C_{s,3}(T,\Omega,\mathbf{a},\epsilon)$ such that*

$$\|\phi_t\|^2_Q + \|\epsilon^{1/2}\nabla\phi(0)\|^2 \le C_{s,3}\|e\|^2_Q,$$

where $C_{s,3} = \left(2 + 2\min\left\{C_{s,1}\|\mathbf{a}\|^2_{L^\infty(0,T;L^\infty(\Omega))}/\epsilon, C_{s,2}\|\mathbf{a}\|^2_{L^2(0,T;L^\infty(\Omega))}/\epsilon\right\}\right)$, *and $C_{s,1}$ and $C_{s,2}$ are as defined in Lemma 5.7 and Lemma 5.8, respectively.*

Proof. This proof is omitted since it is essentially the same as that of Lemma 5.8; although here we initially multiply the dual problem (5.1a) by $-\phi_t$ and integrate over Ω. For full details, see [12, 15]. □

5.4. **Completion of the proof of the a posteriori error bound.** We shall now proceed to estimate the terms I–V on the right-hand side of (5.3). For the first term I, we have

$$\begin{aligned}
\mathrm{I} &= \sum_{n=0}^{M}\int_{t_n}^{t_{n+1}}\sum_{\kappa\in\mathcal{T}_{n+1}}([u_h^n]/k_{n+1} + \mathbf{a}\cdot\nabla u_h - \epsilon\Delta u_h - f, \phi_h - \phi)_\kappa\,\mathrm{d}t \\
&\equiv (R_1, \phi_h - \phi)_Q = (R_1, \mathcal{I}\phi - \phi)_Q + (R_1, \mathcal{I}(\pi\phi - \phi))_Q \\
&\equiv \mathrm{I}_1 + \mathrm{I}_2,
\end{aligned}$$

where $\mathcal{I}$ and π are as defined by (5.6), and

$$R_1|_\kappa = [u_h^n]/k_{n+1} + \mathbf{a}\cdot\nabla u_h - \epsilon\Delta u_h - f, \ \text{ for } \kappa\in\mathcal{T}_{n+1}.$$

By Lemma 5.5 and Lemma 5.8, we have

$$|\mathrm{I}_1| \le C_{p,1}\|h^2R_1\|_Q\|\Delta\phi\|_Q \le \frac{C_{p,1}\sqrt{C_{s,2}}}{\epsilon}\|h^2R_1\|_Q\|e\|_Q.$$

Similarly, using Lemma 5.6 and Lemma 5.9, we have

$$|\mathrm{I}_2| \le C_{p,3}\|kR_1\|_Q\|\phi_t\|_Q \le C_{p,3}\sqrt{C_{s,3}}\,\|kR_1\|_Q\|e\|_Q.$$

Hence,

$$|\mathrm{I}| \le \frac{C_{p,1}\sqrt{C_{s,2}}}{\epsilon}\|h^2R_1\|_Q\|e\|_Q + C_{p,3}\sqrt{C_{s,3}}\,\|kR_1\|_Q\|e\|_Q. \tag{5.22}$$

Analogously, we have

$$\begin{aligned}
\mathrm{II} &= \sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} \left(\sum_{\kappa \in \mathcal{T}_{n+1}} (\epsilon \Delta u_h, \phi_h - \phi)_\kappa + (\epsilon \nabla u_h, \nabla(\phi_h - \phi)) \right) \mathrm{d}t \\
&= \sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} \left(\sum_{\kappa \in \mathcal{T}_{n+1}} (\epsilon \Delta u_h, \mathcal{I}_{n+1}\phi - \phi)_\kappa + (\epsilon \nabla u_h, \nabla(\mathcal{I}_{n+1}\phi - \phi)) \right) \mathrm{d}t \\
&\quad + \sum_{n=0}^{M} \int_{t_n}^{t_{n+1}} \left(\sum_{\kappa \in \mathcal{T}_{n+1}} (\epsilon \Delta u_h, \mathcal{I}_{n+1}(\pi_{n+1}\phi - \phi))_\kappa \right. \\
&\qquad\qquad \left. + (\epsilon \nabla u_h, \nabla \mathcal{I}_{n+1}(\pi_{n+1}\phi - \phi)) \right) \mathrm{d}t \\
&\equiv \mathrm{II}_1 + \mathrm{II}_2.
\end{aligned}$$

By Lemma 5.5 and Lemma 5.8, we have

$$|\mathrm{II}_1| \leq C_{p,2} \|h^2 D_h^2 u_h\|_Q \|\epsilon \Delta \phi\|_Q \leq C_{p,2} \sqrt{C_{s,2}}\, \|h^2 D_h^2 u_h\|_Q \|e\|_Q.$$

Also, by Lemma 5.6 we have

$$\mathrm{II}_2 = 0.$$

Thus, with $R_2 = D_h^2 u_h$ we have that

$$|\mathrm{II}| \leq C_{p,2} \sqrt{C_{s,2}}\, \|h^2 R_2\|_Q \|e\|_Q. \tag{5.23}$$

Next, we consider term III. Applying the Cauchy–Schwarz inequality, inequality (5.8) and Lemma 5.7, we have

$$\begin{aligned}
|\mathrm{III}| &\leq \|kR_3\|_Q \|\phi_h\|_Q \leq C_{i,2} \|kR_3\|_Q \|\phi\|_Q \leq C_{i,2} \sqrt{T}\, \|kR_3\|_Q \|\phi\|_{L^\infty(0,T;L^2(\Omega))} \\
&\leq C_{i,2} \sqrt{C_{s,1} T}\, \|kR_3\|_Q \|e\|_Q, \qquad (5.24)
\end{aligned}$$

where

$$R_3|_{S_{n+1}} = (D_t^h u_h - \bar{f} - ([u_h^n]/k_{n+1} + \mathbf{a} \cdot \nabla u_h - f))/k_{n+1}.$$

Now, we consider term IV. Using Lemma 5.6 and Lemma 5.9, we have

$$|\mathrm{IV}| \leq \|kR_4\|_Q \|\phi_t\|_Q \leq \sqrt{C_{s,3}}\, \|kR_4\|_Q \|e\|_Q, \tag{5.25}$$

where

$$R_4|_{S_{n+1}} = [u_h^n]/k_{n+1} = (u_h^{n+1} - u_h^n)/k_{n+1}.$$

Finally, we consider term V. Using the Cauchy–Schwarz inequality and Lemma 5.7, we get

$$|\mathrm{V}| \leq \|u_0 - u_{h-}^0\| \|\phi(0)\| \leq \sqrt{C_{s,1}}\, \|u_0 - u_{h-}^0\| \|e\|_Q. \tag{5.26}$$

Substituting the estimates (5.22)–(5.26) back into the error representation formula (5.3), recalling the definitions of $C_{p,1}$, $C_{p,2}$ and $C_{p,3}$ from Lemmas 5.5 and 5.6, and dividing through by $\|e\|_Q$ proves the *a posteriori* error bound stated in Theorem 3.1.

6. Numerical experiments

In this section we present some numerical experiments to illustrate the performance of the adaptive algorithm (4.3), (4.4) on a number of convection-diffusion test problems. In the following we let $r = 1$, i.e., so that S_{h_n}, $n = 0, \ldots, M+1$, consists of continuous piecewise linear functions.

We note that, for the practical implementation of (4.3), (4.4) we have used the red-green isotropic refinement strategy. Here, the user must first specify a (coarse) *background* mesh upon which any future refinement will be based. A red refinement corresponds to dividing a certain triangle into four similar triangles by connecting the midpoints of the sides. Green refinement is only temporary and is used to remove any hanging nodes caused by a red refinement. We note that green refinement is only used on elements which have one hanging node; for elements with two or more hanging nodes a red refinement is performed. Within this mesh modification strategy, elements may also be removed from the mesh (i.e., derefined) provided they do not lie in the original background mesh. Mesh coarsening can lead to a loss of information as the elements are deleted and thereby lead to a degradation in the accuracy of the computed numerical solution. To reduce this problem, we ensure that the mesh is not coarsened too quickly from one time step to the next. For the practical implementation of this mesh modification strategy we have used the FEMLAB package developed by K. Eriksson *et al.* [9].

Before we proceed, in the next section we first outline how the error constants arising in the *a posteriori* error bound stated in Theorem 3.1 may be numerically estimated, thereby improving the efficiency of the adaptive algorithm (4.3), (4.4).

6.1. Calibration of the error constants.

The size of the error constants C_i, $i = 1, \ldots, 6$, appearing in the *a posteriori* bound (3.1) may be estimated analytically. Indeed, Section 5.3 provides explicit formulae for the strong stability constants $C_{s,1}$, $C_{s,2}$ and $C_{s,3}$; [12, 17] give analytical upper bounds for the quasi-interpolation constants $C_{i,1}$ and $C_{i,2}$. However, since any value of these constants that is arrived at through such general analytical arguments is necessarily a considerable overestimate, i.e., corresponds to the "worst case" scenario, the error constants must be determined computationally for the problem at hand as part of the process of *a posteriori* error estimation.

To this end, the quasi-interpolation constants $C_{i,1}$ and $C_{i,2}$ are calculated by constructing quasi-interpolants to over 1000 algebraic and trigonometric polynomials defined on $\bar{\Omega}$ of degree up to and including 5, with randomly generated coefficients. The constants $C_{i,1}$ and $C_{i,2}$ are then approximated by taking the supremum over this set of data of normalised quasi-interpolation errors.

The estimation of the strong stability constants $C_{s,j}$, $j = 1, \ldots, 3$, is certainly far from being trivial, since the error function e is not known. We could proceed as we did for the quasi-interpolation constants and replace e by an arbitrary function ψ, compute the solution to the backward dual problem (5.1) numerically, calculate $C_{s,j}$ for each ψ and take the supremum of $C_{s,j}$ over all such ψ. Of course, to do this numerically we would have to choose a finite number of right-hand side functions ψ, and take the supremum for $C_{s,j}$ over this set of trial functions. This approach has been implemented by Sandboge [20] for reactive flow problems. However, numerical computations based on this approach have shown that the individual stability constants may vary by as much as one or even two orders of magnitude depending on the function chosen to represent e.

Clearly, for the purposes of both reliability and efficiency, it is important to "somehow" choose a right-hand side function that is representative of the error function associated with the numerical scheme for the particular physical problem under consideration. For a steady problem we may approximate e by solving the original (primal) problem on two consecutive meshes and setting

$$e \approx e_h = u_h^{\text{fine}} - u_h^{\text{coarse}},$$

cf. Hansbo & Johnson [11]. Then, the corresponding (steady) dual problem is solved with e_h as the right-hand side. However, to apply this strategy for the time-dependent case (cf. (2.1)) we would need to store the function $e_h^n \approx e(\mathbf{x}, t_n)$ at each time level t_n. Clearly, this is not very practical due to the large amount of storage that would be required. Instead we propose to only store a small number of right-hand sides in the set $\{e_h^n\}$, e.g., every tenth, say, and assume the error e, is a piecewise constant function in time at time-levels where e_h^n was not stored (see also Burman [5]).

Computational estimates of the strong stability constants will be given in the next section for the first two model problems considered. We note here that in each case the backward dual problem is solved on a uniform space-time grid using a computationally inexpensive finite difference scheme based on explicit first-order upwind differences for the convection terms and implicit central differences for the diffusion terms. This finite difference scheme can be shown to be both stable and monotone if $\nu_x + \nu_y \leq 1$, where ν_x and ν_y are the Courant numbers calculated in their respective coordinate directions. The strong stability constants are then numerically estimated by interpreting the finite difference approximation of ϕ to be a piecewise constant function in time and a piecewise biquadratic function in space.

Example 6.1. To investigate the computational performance of the *a posteriori* error bound (3.1), we first consider a convection-diffusion problem with a known analytical solution. To this end, we attempt Problem 4 from the Convection-Diffusion Forum [1]; this problem models the transport of a small source in a plane shear flow. Here, we let $\Omega = (0, 24000) \times (-3400, 3400)$, $f = 0$, $\mathbf{a} = (a_0 + \lambda y, 0)^T$, where $a_0 = 0.5$ and $\lambda = 5.0 \times 10^{-4}$. The initial condition u_0 is a point source of mass m at $\mathbf{x}_0 = (x_0, y_0) = (7200, 0)$. The solution to this problem is then given by

$$u(x, y, t) = \frac{m}{4\pi\epsilon t(1 + \lambda t^2/12)^{1/2}}\, \mathrm{e}^{-\xi},$$

where

$$\xi = \frac{(x - \bar{x} - \lambda y t/2)^2}{4\epsilon t(1 + \lambda t^2/12)} + \frac{y^2}{4\epsilon t},$$
$$\bar{x} = x_0 + a_0 t.$$

In order to allow the numerical solution of this problem to begin with a finite source size, the computation is started at a time $t = t_0 = 2400$, with $m = 4\pi\epsilon t_0(1 + \lambda t_0^2/12)^{1/2}$.

Numerical estimates of the strong stability constants $C_{s,j}$, $j = 1, \ldots, 3$, are presented in Figure 1 for $\epsilon = 10$ and $T = 9600$. Here, we plot the square root of each of the stability constants (as these are the values which enter into the *a posteriori* error bound (3.1)) as a function of the number of error functions that are stored within the time interval $[2400, 9600]$ and used as data for the (backward) dual problem.

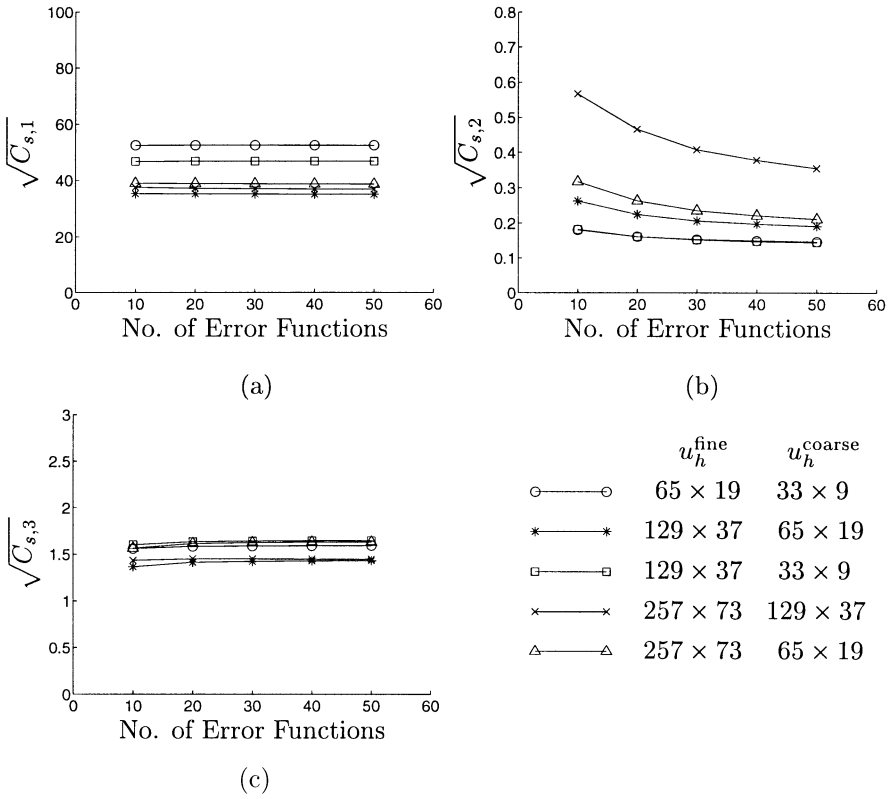
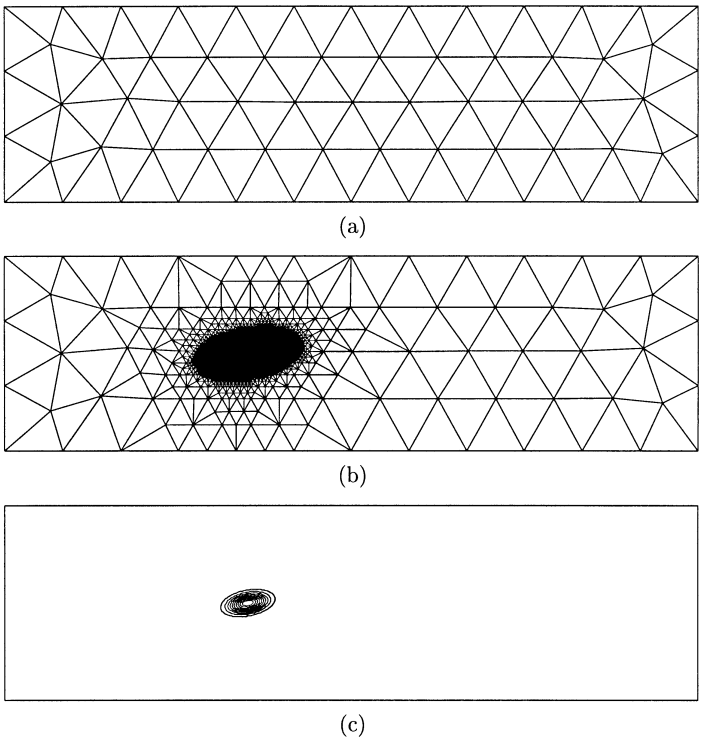


FIGURE 1. Diffusion in a plane shear flow. Numerical estimation of the strong stability constants: (a) $\sqrt{C_{s,1}}$; (b) $\sqrt{C_{s,2}}$; (c) $\sqrt{C_{s,3}}$.

We compute each stability constant for five different error functions constructed by taking the difference between the numerical solution on two consecutive uniform meshes. Here, the finer meshes are constructed using both one and two levels of refinement of the coarse mesh. This is essential in order to ensure that the numerical approximation on the finer mesh is more accurate than u_h^{coarse} in a *critical* way, and thereby that the difference between u_h^{fine} and u_h^{coarse} is representative of the actual error e. In Figure 1 we see that the size of the stability constants does not significantly vary as the number of error functions stored in the time interval increases. Moreover, there is only a small variance between the estimated size of $C_{s,j}$, $j = 1, \ldots, 3$, for each of the error functions constructed.

The background mesh for this problem is shown in Figure 2(a). This mesh is adaptively refined to resolve the "initial" condition $u_0(\mathbf{x}) = u(x, y, t_0)$ at time $t_0 = 2400$ (see Figures 2(b) and 2(c)). Here, $\mathcal{E}_0(u_0, u_{h-}^0, h) = 160.86$.

To investigate the reliability and efficiency of the adaptive algorithm (4.3), (4.4), we provide in Tables 1 and 2 a number of experiments for $T = 7200$ and $T = 9600$, respectively, for different TOL_h and TOL_k. In each case we compute the error estimator $\overset{\circ}{\mathcal{E}}(u_h, h, k, f)$, the $L^2(2400, T; L^2(\Omega))$ norm of the actual error and the effectivity index $\overset{\circ}{\mathcal{E}}/\|e\|_Q$. Here, we see that in each case $\overset{\circ}{\mathcal{E}}(u_h, h, k, f)$ over-estimates the error, which is what we expect, thereby ensuring that the adaptive algorithm



(a)

(b)

(c)

FIGURE 2. (a) Background mesh, with 65 nodes and 98 elements; (b) Background mesh adapted to resolve the "initial" condition, with 2194 nodes and 4353 elements; (c) "Initial" condition at time $t = 2400$.

TABLE 1. Diffusion in a plane shear flow: comparison of the effectivity index for $\epsilon = 10$ and $T = 7200$.

TOL_h	TOL_k	$\mathring{\mathcal{E}}(u_h, h, k, f)$	$\|e\|_Q$	$\mathring{\mathcal{E}}/\|e\|_Q$
10000	12000	15224.27	5210.96	2.92
8000	10000	12825.79	5187.67	2.47
6000	8000	10498.60	5035.01	2.09

TABLE 2. Diffusion in a plane shear flow: comparison of the effectivity index for $\epsilon = 10$ and $T = 9600$.

TOL_h	TOL_k	$\mathring{\mathcal{E}}(u_h, h, k, f)$	$\|e\|_Q$	$\mathring{\mathcal{E}}/\|e\|_Q$
16000	20000	24973.28	7552.86	3.31
12000	16000	20194.49	7511.34	2.69
8000	12000	15364.94	7293.68	2.11

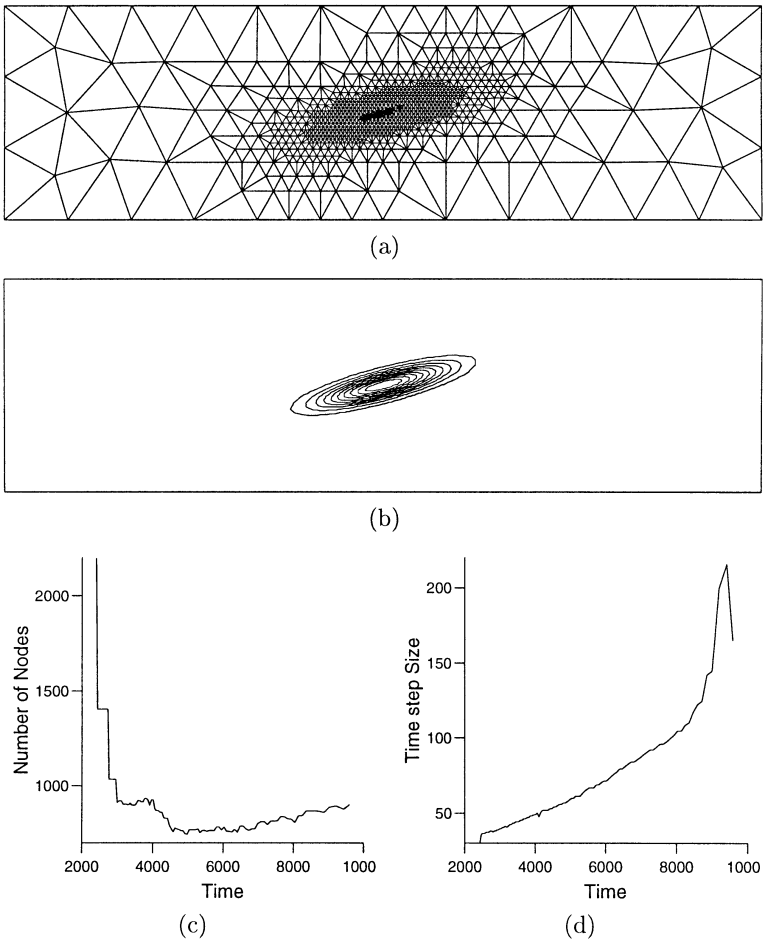


FIGURE 3. Diffusion in a plane shear flow for $\mathrm{TOL}_h = 8000$, $\mathrm{TOL}_k = 12000$, with $\epsilon = 10$ and $T = 9600$: (a) and (b) Mesh and solution (resp.) at the final time $t = 9600$, with 900 nodes and 1764 elements; (c) History of nodes against time; (d) History of time step size against time.

is reliable. To measure the efficiency of the adaptive algorithm we look at the size of the effectivity index: ideally this should be close to one. From Tables 1 and 2 we see that our error estimator $\mathring{\mathcal{E}}(u_h, h, k, f)$ over-estimates the actual error by about 2–3.5 times. We note that the magnitude of the tolerances TOL_h and TOL_k are so large in this example as a result of the size of the computational space-time domain.

Finally, in Figure 3 we present the numerical results for $\mathrm{TOL}_h = 8000$, $\mathrm{TOL}_k = 12000$ and $T = 9600$. Here we see from Figures 3(a) and 3(b) that the mesh is adaptively refined around the numerical solution, as we would expect. In Figures 3(c) and 3(d) we present a history of the number of nodes against time and the time step size against time, respectively. In Figure 3(c) we see that the large number of

nodes introduced at the "initial" time to represent $u_0(\mathbf{x})$ are gradually removed; the number of nodes then slowly increases as the solution spreads out through the diffusion and shearing processes. In Figure 3(d) we observe that the time step increases as the solution becomes smoother through the process of diffusion; the time step is reduced at the end of the computation to ensure that $t_M + k_{M+1,\hat{n}} = T$, cf. Remark 4.1.

Example 6.2. In this example we consider a convection-diffusion problem with both internal and boundary layers; see [22, 23] for the case where $\epsilon = 0$. Here, we let $\Omega = (0,1)^2$, $f = 0$, $\mathbf{a} = (2,1)^T$, with the initial condition, $u_0(\mathbf{x}) = 0$ for $\mathbf{x} \in \Omega_\delta = (\delta, 1) \times (0, 1-\delta)$. For $\mathbf{x} \in \Omega \backslash \Omega_\delta$, $u_0(\mathbf{x})$ is defined to be the linear function which satisfies the boundary conditions

$$u(\mathbf{x},t) = u(x,y,t) = \begin{cases} 1 & \text{for } x = 0,\ 0 \le y \le 1, \\ 1 & \text{for } 0 \le x \le 1, y = 1, \\ (\delta - x)^+/\delta & \text{for } 0 \le x \le 1, y = 0, \\ (y - 1 + \delta)^+/\delta & \text{for } x = 1, 0 \le y \le 1. \end{cases}$$

We note that (for δ small) initially the solution to this problem has boundary layers along $x = 0$ and $y = 1$. The boundary layer along $x = 0$ propagates into the domain Ω and interacts with the outflow boundary at $x = 1$, where a new boundary layer develops at time $t \sim 0.5$. The combination of both internal and boundary layers in this problem make it a challenging model problem. In the following, we shall let $\delta = 7.8125 \times 10^{-3}$, $\epsilon = 1.0 \times 10^{-3}$ and $T = 0.55$ which is the time just before the steady-state solution starts to develop.

In Figure 4 we present numerical estimates of the strong stability constants $C_{s,j}$, $j = 1, \ldots, 3$. Here, we see that the size of the stability constants $C_{s,1}$ and $C_{s,3}$ initially increases as the number of error functions stored increases, before achieving a constant value; the opposite behaviour is observed for the second stability constant $C_{s,2}$. Moreover, in contrast to Example 6.1 we now observe a larger variance between the estimated size of $C_{s,j}$, $j = 1, \ldots, 3$, for each of the error functions constructed. This clearly illustrates the increased complexity of this numerical example.

Here, we specify the background mesh to be the 5×5 triangular mesh shown in Figure 5(a). This mesh is initially refined in order to resolve the boundary layers along $x = 0$ and $y = 1$ at time $t = 0$ (see Figure 5(b)). Here, $\mathcal{E}_0(u_0, u^0_{h-}, h) = 5.4223 \times 10^{-5}$. Numerical results are presented in Figures 6 and 7 for $\mathrm{TOL}_h = \mathrm{TOL}_k = 0.045$; the estimated $L^2(0,T;L^2(\Omega))$ error is $\overset{\circ}{\mathcal{E}}(u_h, h, k, f) = 6.1741 \times 10^{-2}$.

In Figures 6(a), (b) and 7(a), (b) we see that the spatial mesh is concentrated in the internal and boundary layers of the solution. In particular, we see that the spatial mesh has been refined at the points $x = 0$, $y = 0$ and $x = 1$, $y = 1$ where the boundary conditions are "nearly" discontinuous. Further, from Figure 7(a) we see that the mesh emanating from the bottom left-hand corner (i.e., at $x = 0$, $y = 0$) is finer at the top and at the bottom of the internal layer. This is because for $\epsilon = 1.0 \times 10^{-3}$, the spatial part of the adaptive algorithm, i.e., (4.3a), is dominated by the term $C_3 \| h^2 R_2 \|_Q$ which measures the curvature of the numerical solution. From Figures 6(c), (d) and 7(c), (d) we see that the implementation of the adaptive mesh algorithm gives rise to smooth approximations to the very steep features of the solution; we note, however, that some oscillations are observable in the zoom of

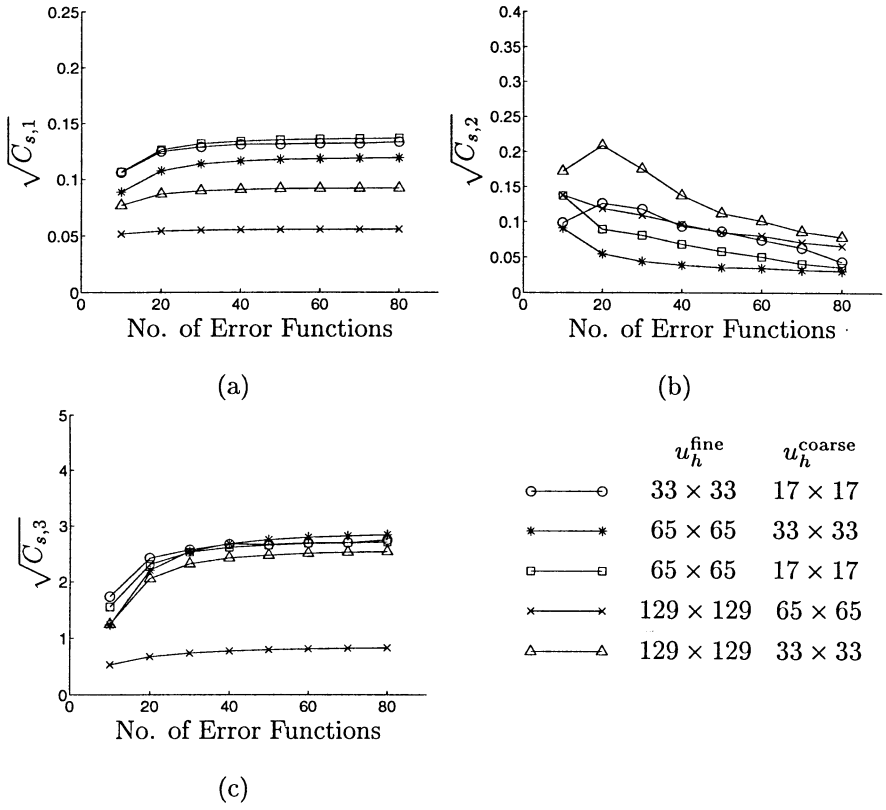


FIGURE 4. Boundary/internal layer problem. Numerical estimation of the strong stability constants: (a) $\sqrt{C_{s,1}}$; (b) $\sqrt{C_{s,2}}$; (c) $\sqrt{C_{s,3}}$.

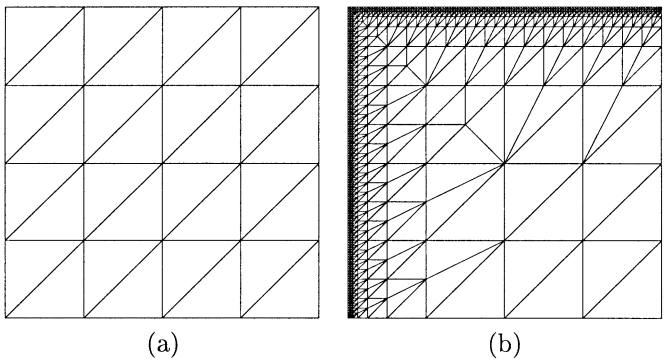


FIGURE 5. (a) Background mesh, with 25 nodes and 32 elements; (b) Background mesh adapted to resolve the initial condition, with 1007 nodes and 1738 elements.

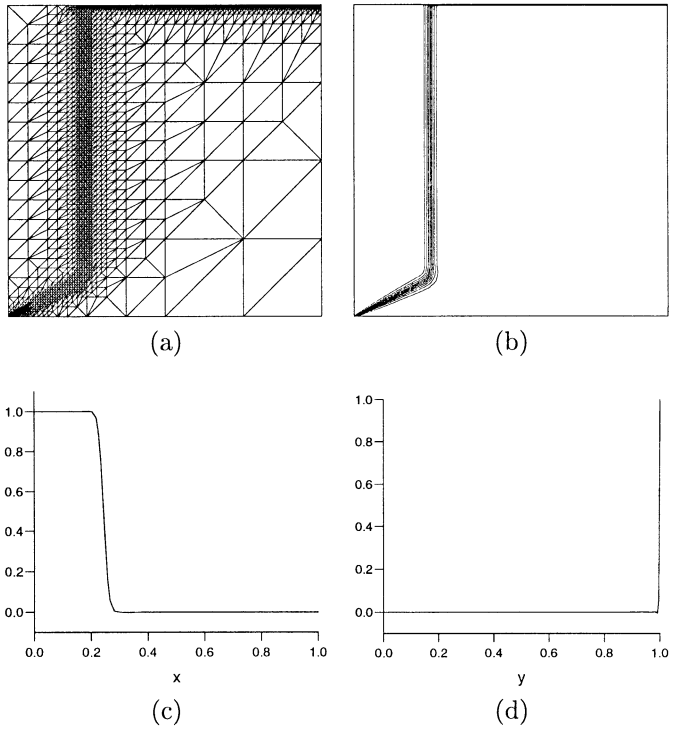


FIGURE 6. Boundary/internal layer problem for $\mathrm{TOL}_h = \mathrm{TOL}_k = 0.045$, with $\epsilon = 1.0 \times 10^{-3}$ and $T = 0.55$: (a) and (b) Mesh and solution (resp.) at time $t = 0.1197$, with 3146 nodes and 5837 elements; (c) Cut through the solution at $y = 0.5$ and $0 \leq x \leq 1$; (d) Cut through the solution at $x = 0.5$ and $0 \leq y \leq 1$.

the outflow boundary layer (cf. Figure 7(d)), although their magnitude is extremely small.

In Figures 7(e) and 7(f) we present a history of the number of nodes against time and the time step size against time, respectively. In Figure 7(e) we see that initially there is a large number of nodes in the spatial mesh as the very steep layers propagate into the domain. However, as these layers become smoother through the process of diffusion, the number of nodes gradually decreases to a minimum before the development of the boundary layer at $x = 1$, when the number of nodes increases again. The reverse of this can be seen in Figure 7(f) for the time step size: initially the time steps are very small in order to correctly capture the very large gradients of the solution. However, the time step size gradually increases as the solution becomes smoother before becoming small again when the outflow boundary layer develops.

Example 6.3. In this example we consider the so-called rotating cylinder problem. Here, we let $\Omega = (0,1)^2$, $f = 0$, $\mathbf{a}(x,y) = -2\pi(2y-1, 1-2x)^T$ and

$$u_0(\mathbf{x}) = \begin{cases} 1, & \text{for } s \leq 1/4, \\ 0, & \text{otherwise,} \end{cases}$$

where $s^2 = (2x-1/2)^2 + (2y-1)^2$.

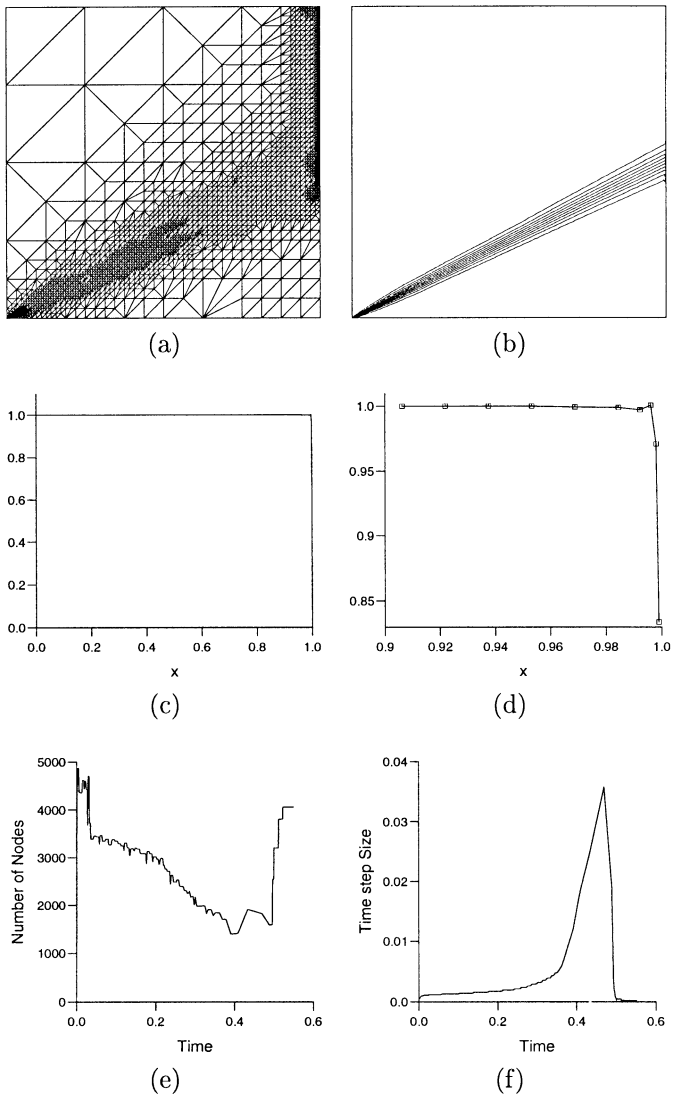


FIGURE 7. Boundary/internal layer problem for $\mathrm{TOL}_h = \mathrm{TOL}_k = 0.045$, with $\epsilon = 1.0 \times 10^{-3}$ and $T = 0.55$: (a) and (b) Mesh and solution (resp.) at the final time $t = 0.55$, with 4057 nodes and 7511 elements; (c) Cut through solution at $y = 0.75$ and $0 \leq x \leq 1$; (d) Zoom of (c); (e) History of nodes against time; (f) History of time step size against time.

We take $\epsilon = 1.0 \times 10^{-5}$ and $T = 2$ (corresponding to four revolutions). The stability constants $C_{s,j}$, $j = 1, \ldots, 3$, are again estimated analogously as in Examples 6.1 and 6.2; for brevity, we omit the figures showing their numerical values. Here, $\sqrt{C_{s,1}} \approx 0.08$, $\sqrt{C_{s,2}} \approx 0.0005$ and $\sqrt{C_{s,3}} \approx 3.5$. The background mesh for this problem is shown in Figure 8(a). This mesh is adaptively refined at time

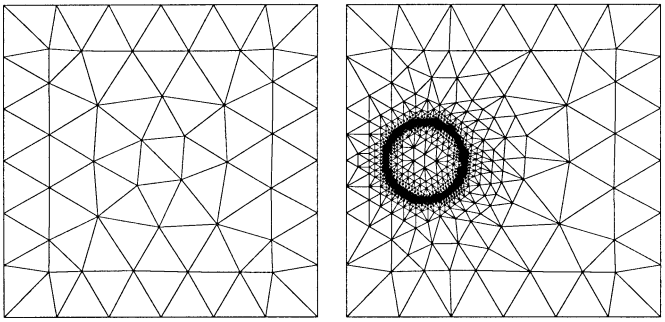


FIGURE 8. (a) Background mesh, with 56 nodes and 86 elements; (b) Background mesh adapted to resolve the initial condition, with 2536 nodes and 5044 elements.

$t = 0$ to resolve the initial condition $u_0(\mathbf{x})$ (see Figure 8(b)); here, $\mathcal{E}_0(u_0, u_{h-}^0, h) = 5.4223 \times 10^{-5}$. Numerical results are presented in Figure 9 for $\mathrm{TOL}_h = 0.01$ and $\mathrm{TOL}_k = 0.45$; the estimated $L^2(0,T;L^2(\Omega))$ error is $\overset{\circ}{\mathcal{E}}(u_h, h, k, f) = 0.4445$. We remark that if we compare this with the actual error calculated for the strictly hyperbolic problem (i.e., where $\epsilon = 0$), this gives rise to an effectivity index of 7.83.

In Figures 9(a), (b), (c), and (d) we see that the mesh "follows" the position of the cylinder. However, we observe that the mesh is coarser in the regions of the cylinder closest to and farthest from the centre of rotation, though this is much more evident in the region closest to the stagnation point. As a result we observe that there is a slight "leakage" of the numerical solution in these coarser regions of the mesh. The reason for this is that when $\epsilon \ll 1$, the spatial refinement algorithm is dominated by the directionally dependent "hyperbolic" part of the residual, i.e., R_1 $(= [u_h^n]/k_{n+1} + \mathbf{a} \cdot \nabla u_h - f$ on $\kappa \in \mathcal{T}_{n+1}$ for $r = 1)$. Indeed, we expect R_1 to concentrate the mesh more in internal layers that are orthogonal to the flow direction, i.e., where $\mathbf{a} \cdot \nabla u_h$ is large, rather than in layers that are aligned with the flow, i.e., where $\mathbf{a} \cdot \nabla u_h \sim 0$. The lack symmetry between the regions in the mesh closest to and farthest from the centre of rotation is attributed to the fact that the magnitude of velocity vector $\mathbf{a}(\mathbf{x})$ increases the further $\mathbf{x}$ is away from the stagnation point $(1/2, 1/2)$. We remark that this behaviour was not observed in the previous numerical example, since for larger ϵ the directionally independent term R_2 $(= D_h^2 u_h)$ dominates the spatial part of the adaptive algorithm. For Example 6.2, $C_1\|h^2 R_1\|_Q = 4.01 \times 10^{-3}$ and $C_3\|h^2 R_2\|_Q = 1.31 \times 10^{-2}$, whereas for the rotating cylinder problem, $C_1\|h^2 R_1\|_Q = 3.25 \times 10^{-3}$ and $C_3\|h^2 R_2\|_Q = 1.21 \times 10^{-4}$. To reduce this problem, in [16] we introduced an artificial diffusion model for the Lagrange–Galerkin method based on calculating second discrete derivatives of the numerical solution u_h. To prevent excessive smearing of the numerical solution, the size of the artificial diffusion coefficient $\hat{\epsilon}$ was limited within the adaptive algorithm by refining elements in the computational mesh where $\hat{\epsilon}$ was larger than some predetermined maximum value.

Finally, in Figures 9(e) and 9(f) we again present a history of the number of nodes against time and the time step size against time, respectively. From Figure 9(e) we observe that the number of nodes in the spatial mesh remains fairly constant

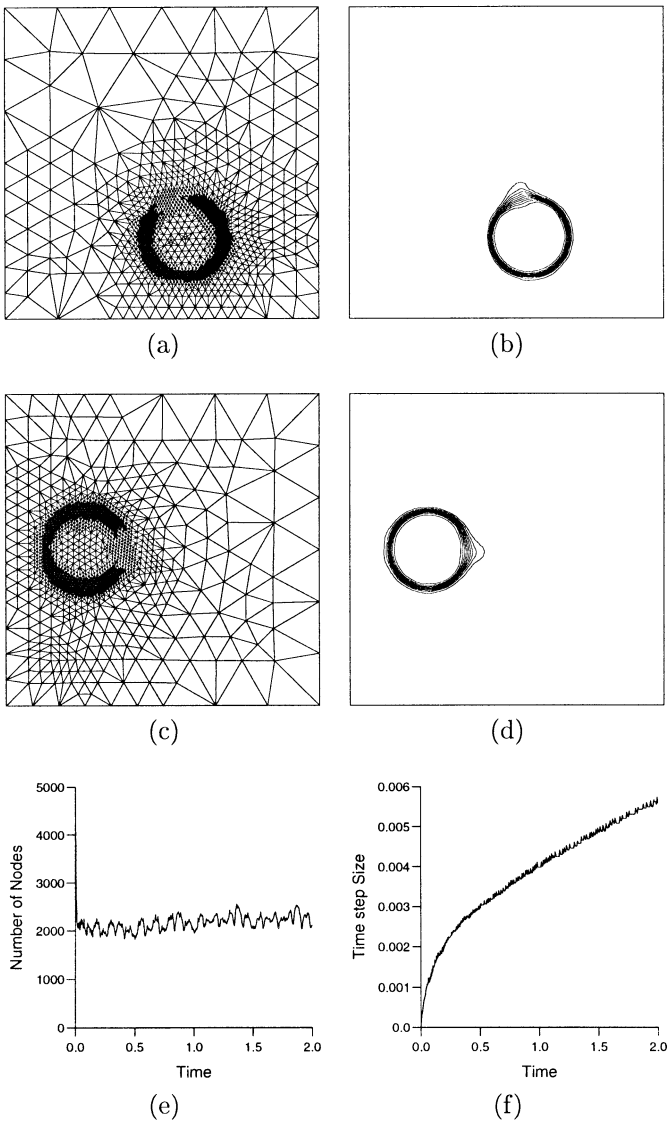


FIGURE 9. Rotating cylinder problem for $\mathrm{TOL}_h = 0.01$, $\mathrm{TOL}_k = 0.45$, with $\epsilon = 1.0 \times 10^{-5}$ and $T = 2$: (a) and (b) Mesh and solution (resp.) at time $t = 1.149$, with 2188 nodes and 4333 elements; (c) and (d) Mesh and solution (resp.) at the final time $t = 2$, with 2105 nodes and 4168 elements; (e) History of nodes against time; (f) History of time step size against time.

over the length of the computation. In contrast with this we see from Figure 9(f) that as the solution becomes smoother through the process of diffusion, the time step slowly increases.

7. Summary

In this paper we have derived an *a posteriori* error bound for the Lagrange–Galerkin discretisation of an unsteady (linear) convection-diffusion problem, assuming only that the underlying space-time mesh is nondegenerate. Moreover, based on this error bound, we have designed an adaptive algorithm to ensure global control of the error with respect to a predetermined tolerance. The reliability and efficiency of such an adaptive algorithm is dependent on the relative size of the error constants arising in the *a posteriori* bound. Since any such estimate of these constants derived by analytical techniques represents a considerable over-estimate, these quantities must be numerically estimated for the problem at hand. While estimating interpolation/quasi-interpolation constants is a relatively easy process, obtaining numerical estimates of the stability factors is a formidable task, especially for time-dependent problems. The latter involves computing the solution to the backward dual problem with "representative" data which approximates the error. In this paper we have constructed approximations to the error by solving the original (primal) problem on two consecutive meshes, cf. [5, 11]. For simplicity, we only considered uniform meshes, though constructing approximations to the error on nonuniform unstructured meshes is certainly possible. In general, the numerical estimation of the error constants, and in particular, the stability constants will mean that the reliability of the adaptive algorithm can no longer be guaranteed. The degree of confidence that one may have about the reliability of an adaptive algorithm will very much depend on how much computational effort is invested into calculating these stability factors.

Numerical experiments presented in this paper indicate that as $\epsilon \to 0$ the spatial part of the adaptive algorithm becomes dominated by the directionally dependent "hyperbolic" part of the residual of the underlying partial differential equation. In particular, this leads to a mesh adaptation strategy that concentrates the mesh within layers that are orthogonal to the flow direction, while the mesh remains coarser within layers that are aligned with the flow. The lack of correlation between the local error and the local residual, calculated on a given element in the computational mesh, has serious ramifications for the study of strictly hyperbolic problems. Indeed, mesh adaptation strategies based solely on refining elements according to the local size of the residual of the underlying partial differential equation may fail to resolve all the regions in the computational domain where the *actual* error is *locally* large. Clearly, there is a great need to develop local error indicators that can detect such regions in the computational domain, irrespective of their orientation with respect to the flow direction. Indeed, as part of our program of future research, in [18] we investigate error creation and error propagation phenomena in the numerical solution of strictly hyperbolic problems, following the ideas presented in [13] and in the review article [22].

Finally, we remark that in the present paper, the objective has been to derive a computable *a posteriori* error bound in order to ensure the analytical solution u is reliably approximated by u_h in the space-time L^2-norm. This involved first deriving an error representation formula in terms of the numerical solution u_h and the dual solution ϕ; ϕ was then eliminated by exploiting well-posedness results for the adjoint problem at the expense of introducing global stability factors into the *a posteriori* error bound. Obtaining good estimates for the stability factors is essential for the success of this approach. Alternatively, less error analysis may be performed and

more numerical estimation done; for example, the norms of the dual solution may be kept in the error representation formula as local weights and approximated by solving the dual problem numerically and recovering the derivatives of ϕ to calculate the appropriate quantities. For further details on these aspects, we refer to the work of D. Estep (see [9] and the references cited therein) and R. Rannacher [2]; see also [14].

References

1. A.M. Baptista, E.E. Adams and P. Gresho. Benchmarks for the transport equation: the convection-diffusion forum and beyond. In, Lynch and Davies, editors, *Quantitative Skill Assessment for Coastal Ocean Models*, AGU Coastal and Estuarine Studies, 47:241–268, 1995.
2. R. Becker and R. Rannacher. Weighted a posteriori error control in FE methods. Technical Report 96-1, Institut für Angewandte Mathematik, Universität Heidelberg, Heidelberg, Germany, 1996.
3. M. Bercovier and O. Pironneau. Characteristics and the finite element method. In T. Kawai, editor, *Proceedings of the Fourth International Symposium on Finite Element Methods in Flow Problems*, pp 67–73. North–Holland, 1982.
4. S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods*, Volume 15 of *Texts in Applied Mathematics*. Springer–Verlag, 1994. MR **95f:**65001
5. E. Burman. *Adaptive Finite Element Methods for Compressible Two-Phase Flow*. PhD thesis, Chalmers University of Technology, Göteborg, 1998.
6. J. Douglas and T.F. Russell. Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Numer. Anal.*, 19(5):871–885, 1982. MR **84b:**65093
7. K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems I: A linear model problem. *SIAM J. Numer. Anal.*, 28(1):43–77, 1991. MR **91m:**65274
8. K. Eriksson and C. Johnson. Adaptive streamline diffusion finite element methods for time dependent convection diffusion problems. Technical Report 1993-23, Department of Mathematics, Chalmers University of Technology, Göteborg, Sweden, 1993.
9. K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Introduction to Adaptive Methods for Differential Equations*, in *Acta Numerica* 1995, pp 105–158. Cambridge University Press, 1995. MR **96k:**65057
10. P. Grisvard. *Singularities in Boundary Value Problems*, *Research Notes in Applied Mathematics*. Masson and Springer–Verlag, 1992. MR **93h:**35004
11. P. Hansbo and C. Johnson. Streamline diffusion finite element methods for fluid flow. von Karman Institute Lectures, 1995.
12. P. Houston. *Lagrange–Galerkin Methods for Unsteady Convection-Diffusion Problems: A Posteriori Error Analysis and Adaptivity*. PhD thesis, University of Oxford, 1996.
13. P. Houston, J. Mackenzie, E. Süli and G. Warnecke. A posteriori error analysis for numerical approximations of Friedrichs systems. *Numer. Math.* 82:433–470, 1999. CMP 99:14
14. P. Houston, R. Rannacher and E. Süli. A posteriori error analysis for stabilised finite element approximations of transport problems. *Comput. Methods Appl. Mech. Engrg.* (to appear).
15. P. Houston and E. Süli. Adaptive Lagrange–Galerkin methods for unsteady convection-dominated diffusion problems. Oxford University Computing Laboratory Technical Report NA95/24, 1995 `(http://www.comlab.ox.ac.uk/oucl/publications/natr/NA-95-24.html).`
16. P. Houston and E. Süli. On the design of an artificial diffusion model for the Lagrange–Galerkin method on unstructured triangular grids. Oxford University Computing Laboratory Technical Report NA96/07, 1996 `(http://www.comlab.ox.ac.uk/oucl/publications/natr/NA-96-07.html).`
17. P. Houston and E. Süli. A posteriori error analysis for linear convection-diffusion problems under weak mesh regularity assumptions. Oxford University Computing Laboratory Technical Report NA97/03, 1997 `(http://www.comlab.ox.ac.uk/oucl/publications/natr/NA-97-03.html).`
18. P. Houston and E. Süli. Local mesh design for the numerical solution of hyperbolic problems. In M. Baines, editor, *Numerical Methods for Fluid Dynamics VI*, pp 17–30. ICFD, 1998.
19. O. Pironneau. On the transport-diffusion algorithm and its applications to the Navier–Stokes equations. *Numer. Math.*, 38:309–332, 1982. MR **83d:**65258

20. R. Sandboge. *Adaptive Finite Element Methods for Reactive Flow Problems*. PhD thesis, Chalmers University of Technology, Göteborg, 1996.
21. L.R. Scott and S. Zhang. Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Math. Comp.*, 54(190):483–493, 1990. MR **90j:**65021
22. E. Süli. A posteriori error analysis and adaptivity for finite element approximations of hyperbolic problems. In D. Kröner, M. Ohlberger and C. Rohde, editors, *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*, Volume 5 of *Lecture notes in Computational Science and Engineering*, pp. 123–144. Springer–Verlag, 1998.
23. E. Süli and P. Houston. Finite element methods for hyperbolic problems: *a posteriori* error analysis and adaptivity. In I.S. Duff and G.A. Watson, editors, *The State of the Art in Numerical Analysis*, pp 441–471. Clarendon Press, 1997. MR **99i:**65107
24. R. Verfürth. Error estimates for some quasi-interpolation operators. M_2AN, 33:695–713, 1999.

Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom
E-mail address: Paul.Houston@mcs.le.ac.uk

Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom
E-mail address: Endre.Suli@comlab.ox.ac.uk