Anisotropic Residual-Based Mesh Adaptation for Reaction-Diffusion Systems: Applications to Cardiac Electrophysiology

Edward Boey

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Mathematics[1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1]The Ph.D. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Abstract

Accurate numerical simulation of reaction-diffusion systems can come with a high cost. A system may be stiff, and solutions may exhibit sharp localized features that require fine grids and small time steps to properly resolve the physical phenomena they represent. The development of efficient methods is crucial to cut down the demands of computational resources.

In this thesis we consider the use of adaptive space and time methods driven by *a posteriori* error estimation. The error estimators for the spatial discretization are built from a variety of sources: the residual of the partial differential equation (PDE) system, gradient recovery operators and interpolation estimates. The interpolation estimates are anisotropic, not relying on classical mesh regularity assumptions. The adapted mesh is therefore allowed to include elements elongated in specified directions, as dictated by the type of solution being approximated.

This thesis proposes an element-based adaptation method to be used for a residual estimator. This method avoids the usual conversion of the estimator to a metric, and instead applies the estimator to directly control the local mesh modifications. We derive a new error estimator for the L^2-norm in the same anisotropic setting and adjust the element-based adaptation algorithm to the new estimator.

This thesis considers two new adaptive finite element settings for reaction-diffusion problems. The first is the extension to a PDE setting of an estimator for the time discretization with the backward difference formula of order 2 (BDF2), based on an estimator for ordinary differential equation (ODE) problems. Coupled with the residual estimator, we apply a space-time adaptation method. The second is the derivation of anisotropic error estimates for the monodomain model from cardiac electrophysiology. This model couples a nonlinear parabolic PDE with an ODE and this setting presents challenges theoretically as well as numerically.

In addition to theoretical considerations, numerical tests are performed throughout to assess the reliability and efficiency of the proposed error estimators and numerical methods.

Acknowledgements

First I would like to thank my thesis supervisors Yves Bourgault and Thierry Giordano. Their advice, encouragement and suggestions have been invaluable throughout my time in Ottawa. I would also like to thank them for their careful readings of this thesis.

I would like to acknowledge the researchers and professionals at GIREF for making their code mef++ available and for many discussions clarifying the use and functionality of their code.

I would like to thank my parents for their support throughout my university studies.

Last I would like to thank Erin for all her support and patience during these past few years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Reaction-diffusion systems represent a large class of time-dependent problems in the field of partial differential equations. They are used to model balance laws, including a very broad range of physical applications, such as population dynamics, chemical processes, and physiology. General resources dealing with the analysis of these systems and the illustration of the various physical applications can be found in [89], [93], [99], [52] and the references therein. Reaction-diffusion systems are known to admit solutions exhibiting a wide variety of pattern formation and wave propagation phenomena. The existence and behaviour of planar wave solutions have been studied in a variety of situations, see for instance [59]. Spiral waves, which propagate with a curved wave profile, were observed as early as in the 1950s with the so-called Belousov-Zhabotinsky reaction, and can been studied in a geometrical framework, for instance see [61].

The problems tend to be nonlinear, and in general cannot be solved by analytical means alone. An important tool in the study of the solutions is to compute accurate numerical approximations. A variety of approaches have been taken, including finite difference schemes, finite volume and finite element methods. The finite element method in particular admits flexibility in terms of geometry and choice of domain. It is therefore the method used in this thesis for space discretization. Standard references for the finite element method are [40] and [23].

In general, a sufficiently accurate numerical solution is difficult to obtain. For example, in cardiac electrophysiology, the wave-front of the action potential requires resolution of approximately 0.1mm, so that for realistic 3D models to reach this resolution for a heart, measuring several cubic cm in volume, a uniform mesh would have to contain on the order of 10^7 unknowns for the simplest models. For more complex models, the number of unknowns could increase by orders of magnitude, and the memory requirement to set up the system becomes prohibitive [92, Chapter 6]. At the same time, in a normal heartbeat, which takes on order of 1 second, phenomena occur at multiple time scales, ranging in some cases from about $0.001 - 0.1$ms, so to

capture the lowest scale with a constant time step will on the order of about 10^6 steps [46].

It is clear that steps should be taken to reduce the prohibitive cost and to improve the efficiency of numerical methods. To name a few approaches, domain decomposition methods [79] can be applied to solve the problem in parallel, and the use of higher-order methods in space [3], and in time [42] can often be advantageous. In this thesis, we consider the use of adaptive finite element methods. The main idea is that if one can approximate the exact error of the discrete solution, the regions that contribute the most and least amount of error can be identified. Then the discretization can be changed locally, with respect to the mesh and time-step size, to either improve accuracy or reduce complexity. For a general overview of error estimation for finite elements, see [1] and [40, Chapter 10], and for a modern introduction to mesh adaptation method, see [48].

For the remainder of this chapter, we introduce the concepts and background material required for this thesis. In Section 1.1 we discuss the general setting of reaction-diffusion systems, including the functional spaces required for the variational framework. Following, we formally introduce the example of the bidomain and monodomain models from cardiac electrophysiology. Next, we introduce the basics of the Lagrangian finite element method for the discretization of the problem in Section 1.2. We discuss a posteriori error estimation in Section 1.2.4 and adaptative methods in Sections 1.2.5 and 1.2.6. Finally, in Section 1.3 we state the goals and problems addressed in the thesis.

1.1 Reaction-diffusion systems

1.1.1 Model problem and functional spaces

Let $T > 0$, and $\Omega \subseteq \mathbb{R}^d$ be a bounded domain with Lipschitz boundary. For i with $1 \leq i \leq m$, consider source terms $S_i : (0,T) \times \Omega \to \mathbb{R}$, reaction terms $F_i : \mathbb{R}^m \to \mathbb{R}$, and initial conditions $U_{i,0} : \Omega \to \mathbb{R}$. For some k with $1 \leq k \leq m$ consider for $i \leq k$ the diffusion coefficients $D_i : \Omega \to M_{d\times d}(\mathbb{R})$, where the matrices $D_i(x)$ are positive definite, and boundary data $G_i : (0,T) \times \partial\Omega \to \mathbb{R}$. For convenience, extend the definitions of D_i, G_i to be zero for $i > k$. In this thesis we are concerned with solving the following system for the unknowns $U_i : (0,T) \times \Omega \to \mathbb{R}$:

$$\begin{cases} \dfrac{\partial U_i}{\partial t} - \operatorname{div}(D_i \nabla U_i) + F_i(U_1, \cdots, U_m) = S_i, & \text{in } (0,T) \times \Omega, \\ D_i \nabla U_i \cdot n = G_i, & \text{in } (0,T) \times \partial\Omega, \\ U_i(0) = U_{i,0}, & \text{in } \Omega. \end{cases} \tag{1.1.1}$$

The solution to system (1.1.1) will be defined in a weak sense. With the Lebesgue measure on Ω we consider the functional spaces $(L^2(\Omega), \|\cdot\|_{0,\Omega})$ and $(H^1(\Omega), \|\cdot\|_{1,\Omega})$

with norms

$$\|u\|_{0,\Omega} = \left(\int_\Omega |u|^2 \,\mathrm{d}x\right)^{1/2}, \qquad u \in L^2(\Omega)$$
$$\|u\|_{1,\Omega} = \left(\int_\Omega (|u|^2 + |\nabla u|^2) \,\mathrm{d}x\right)^{1/2}, \qquad u \in H^1(\Omega).$$

We denote by $(\cdot,\cdot)_\Omega$ the inner product on $L^2(\Omega)$. Let $(V, \|\cdot\|)$ be a Banach space. For a measurable function $v : (0,T) \to V$, define the norm

$$\|v\|_{L^2(0,T;V)} = \left(\int_0^T \|v\|^2 \,\mathrm{d}t\right)^{1/2}, \tag{1.1.2}$$

and define the space $L^2(0,T;V) = \{v : (0,T) \to V : \|v\|_{L^2(0,T;V)} < \infty\}$, which is a Banach space with norm given by 1.1.2. Given reflexive Banach spaces V_1, V_2 with $V_1 \subseteq V_2$, define

$$\mathcal{W}(V_1,V_2) = \{w : [0,T] \to V_1 \,;\, w \in L^2(0,T;V_1),\, \partial_t w \in L^2(0,T;V_2)\},$$

which is a Banach space with norm

$$\|w\|_{\mathcal{W}(V,V')} = \|w\|_{L^2(0,T;V_1)} + \|\partial_t w\|_{L^2(0,T;V_2)}.$$

The time derivative here is meant in the vector-valued distributional sense (see [67] or [93]). Let V, H be two Hilbert spaces such that there exists a continuous embedding $V \subset H$, generally not isometric, such that the image under the embedding of V is dense in H. Let V', H' denote their topological duals, and identify H with H' in the usual way. We naturally have inclusions $V \subset H = H' \subset V'$. We therefore view the duality pairing $(\cdot,\cdot)_{V',V}$ as an extension of the inner product on H. Then, there is a continuous embedding $\mathcal{W}(V,V') \subseteq C([0,T];H)$, see [67, Lemme 1.2]. In particular, for $w \in \mathcal{W}(V,V')$, the point evaluations $w(0)$, $w(T) \in H$ are well-defined.

Under this functional setting, take $V = H^1(\Omega)$ and $H = L^2(\Omega)$. Choose the initial conditions $U_{i,0} \in H$ and boundary data $G_i \in L^2(0,T;L^2(\partial\Omega))$ and source terms $S_i \in L^2(0,T;H)$. Then $\{U_i\} \in \mathcal{W}(V,V')^k \oplus \mathcal{W}(H,H')^{m-k}$ is the solution to the system of variational equations

$$\frac{\mathrm{d}}{\mathrm{d}t}(U_i,\, v)_\Omega + (D_i \nabla U_i,\, \nabla v)_\Omega + (F_i(U_1,\cdots,U_m),\, v)_\Omega = (S_i,\, v)_\Omega + (G_i,\, v)_{\partial\Omega}, \tag{1.1.3}$$
$$\forall v \in \mathcal{W}(V,V')$$
$$U_i(0) = U_{i,0}. \tag{1.1.4}$$

Results for existence and uniqueness to (1.1.4) depend on the choice of reaction terms F_i, and can be found for instance in [89], [93], [99]. In Chapter 3 we will concentrate

on the scalar case, where $m = 1$, and in Chapter 4 we concentrate on the monodomain model, introduced in Section 1.1.2 below, which is a two-variable model with diffusion only for the first variable (so $D_2 = 0$).

Additionally, in Chapter 2 we discuss a stationary diffusion problem which is now described. Consider source term $f \in L^2(\Omega)$, boundary term $g \in H^{1/2}(\partial\Omega)$ and diffusion coefficient $D : \Omega \to M_{d\times d}(\mathbb{R})$. Suppose that the matrix coefficients $D_{i,j}$ belong to $L^\infty(\mathbb{R})$ and that the uniform ellipticity assumption is satisfied: there exists a constant $C > 0$ such that for a.e. $x \in \Omega$

$$\xi^T D(x)\xi \geq C|\xi|_2^2, \quad \forall \xi \in \mathbb{R}^d,$$

where $|\xi|_2 = \left(\sum_{i=1}^d \xi_i^2\right)^{1/2}$. Let $u \in V$ be the solution to the following:

$$\begin{cases} -div(D\nabla u) = f, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega. \end{cases} \tag{1.1.5}$$

Equation (1.1.5) has a unique solution, see for instance [40]. Let $V_0 = \{v \in V : v|_{\partial\Omega} = 0\}$ and $V_g = \{v \in V : v|_{\partial\Omega} = g\}$. Then $u \in V_g$ is the solution to the variational problem

$$(D\nabla u, \nabla v)_\Omega = (f, v)_\Omega, \quad \forall v \in V_0.$$

1.1.2 Equations from electrophysiology

The heart is composed of muscle, which is arranged in fibres. In the course of a normal heartbeat, electrical current flows from excited cells to surrounding cells, with the direction of propogation influenced by the orientation of the fibres. The depolarization of the cells induces the muscle to contract, and blood is pumped from the heart.

We briefly introduce the models from electrophysiology relevant for this thesis. For a thorough derivation of the models, see for instance [59], [60], [47], [92]. The heart at the cell-level is divided into two domains: the intra- and extra-cellular domains Ω_i and Ω_e, separated by the cell membrane. Inside each domain, the current densities J_i and J_e are assumed to be purely Ohmic, satisfying

$$J_i = -M_i\nabla u_i, \qquad J_e = -M_e\nabla u_e,$$

where M_i, M_e are positive definite conductivity matrices. The heart muscle fibres are arranged in sheets, with the eigenvectors of the conductivity matrices corresponding to the direction parallel to the fibre (the direction of highest conductivity), orthogonal to the fibres parallel to the sheets, and orthogonal to the sheets. The membrane acts as an insulator, with capacitance C_m, with most of the ionic transport between

domains facilitated by ionic channels. The separation of ions between the intra- and extra-cellular domains results in a potential difference across the membrane: $u = u_i - u_e$, where u_i and u_e denote the potential in each domain. Through a homogenization argument, for instance in [59], [47], the domains are assumed to coincide $\Omega_i = \Omega_e = \Omega$. The total current is modelled as a circuit in parallel, as a sum of the ionic and capacitive current:

$$I_m = \chi(I_{ion} + I_c),$$

where the constant $\chi > 0$ is the cell membrane surface area to volume ratio, required since the ionic and capacitive currents are measured per unit area of surface membrane, while the transmembrane current is measured per unit volume of tissue. The bidomain model for the potential in an isolated heart in terms of the unknowns u, u_i, $u_e : (0,T) \times \Omega \to \mathbb{R}$ and $W = (w_1, \cdots, w_m) : (0,T) \times \Omega \to \mathbb{R}^m$ is given by

$$\begin{cases} \chi C_m \dfrac{\partial u}{\partial t} - div(\sigma_i \nabla u_i) + \chi I_{ion}(u, W) = 0, & \text{in } (0,T) \times \Omega, \\ \chi C_m \dfrac{\partial u}{\partial t} + div(\sigma_e \nabla u_e) + \chi I_{ion}(u, W) = 0, & \text{in } (0,T) \times \Omega, \\ \dfrac{\partial W}{\partial t} = G(u, W), & \text{in } (0,T) \times \Omega, \\ \sigma_i \nabla u_i \cdot n = 0, & \text{in } (0,T) \times \partial\Omega, \\ \sigma_e \nabla u_e \cdot n = 0, & \text{in } (0,T) \times \partial\Omega, \\ u(0) = u_0, & \text{in } \Omega, \\ W(0) = W_0, & \text{in } \Omega. \end{cases} \tag{1.1.6}$$

The first two equations represent balance laws for the charges in the two domains Ω_i, Ω_e. The boundary conditions are made under the assumption that the heart is insulated. The third equation represents a system of ODEs for the ionic activity, for instance channel gating variables or ionic concentrations, which is coupled to the transmembrane potential by the nonlinear reaction term I_{ion}. The above also assumes there is no external current source, which may be added to the first two equations. Existence results for weak solutions of the bidomain model can be found in [30] for the FitzHugh-Nagumo model, in [20] under some growth assumptions for the reaction terms, including the FitzHugh-Nagumo and Aliev-Panfilov models, and in [19] extending results to the heart-torso coupled problem, as well as a version of the Mitchell-Schaeffer model. Uniqueness so far has only been proven for models related to the FitzHugh-Nagumo, as in [20] and [19], for systems which satisfy a strong monotonicity condition.

If we assume equal anisotropy ratios, so that $M_e = \lambda M_i$ for some $\lambda > 0$, then setting $\sigma = \frac{\lambda}{1+\lambda} M_i$, the first two equations in (1.1.6) may be replaced by a single equation

$$\chi C_m \frac{\partial u}{\partial t} - div(\sigma \nabla u) + \chi I_{ion}(u, W) = 0. \tag{1.1.7}$$

The monodomain system consists of replacing (1.1.7) for the first two equations in (1.1.6) with boundary condition $(\sigma\nabla u)\cdot n = 0$.

Phases of the action potential

The primary importance of the bidomain and monodomain models, is the simulation at the macroscopic scale of the action potential, which is a physiological reaction that results in the heartbeat. The cells of the heart muscle are excitable, reacting to outside stimulus. The ionic channels of cells in the ventricle are controlled by gates, which open and close in a voltage-dependent way. When the potential difference is raised above a certain threshold, an action potential results. In the case of the ventricle, this process can be described as follows. In the upstroke phase, which is short (short scale, a few milliseconds), the sodium channels open, and a large influx of sodium ions enter until the cell is completely depolarized. In the following excited phase (long scale, hundreds of ms), the inward sodium current is balanced by an outward potassium current, and the cell slowly begins to repolarize. During the repolarization phase (short scale, tens of ms), the sodium channel closes, while the outward potassium current completes the repolarization of the cell. After the cell has repolarized, a long scale recovery period begins (long scale, hundreds of ms), during which equilibrium is attained for the ionic currents and gates, for instance by the Na-K pump, which pumps sodium out of the cell and potassium in. See [86], [92], [66], [59] for further details on the physiology of the action potential.

FitzHugh-Nagumo model

The simplest ionic model we consider is the FitzHugh-Nagumo (FHN) model, see [43], with 0D model given by

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t} &= f(u) - w, \\ \frac{\mathrm{d}w}{\mathrm{d}t} &= \epsilon(\kappa u - w), \end{aligned} \tag{1.1.8}$$

where $f(u) = u(u-a)(1-u)$, with constants $0 < a < 1$, $\epsilon > 0$ and $\kappa > 0$. The model is a simplified version of the Hodgkin-Huxley model, which was designed to model the nerve axon of a giant squid [56]. While the model is not considered physically accurate for the purposes of the human heart, it is a simple model that qualitatively captures some features of the action potential. Usually $1 \gg \epsilon$, so w generally varies more slowly. The action potential resulting from the system may be described by "fast-slow" dynamics, by studying the nullclines of (1.1.8). When $f(u) \approx w$, the second equation dominates, and u varies slowly, which roughly corresponding to plateau and recovery phases. Depolarization and repolarization are then modeled by the rapid transitions between branches, moving respectively towards the upper and lower

stable equilibria. See for instance [59] for details. In Chapter 3 we discuss a simplified version of this model, which does not involve the variable w. This version is sufficient to model the depolarization front, but not the repolarization. In Chapter 4 we study the full two-variable model.

Mitchell-Schaeffer model

We also consider the Mitchell-Schaeffer (MS) model from [75], with 0D model given by

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{1}{\tau_{in}} w u^2 (1-u) - \frac{1}{\tau_{out}} u \tag{1.1.9}$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \begin{cases} \frac{1}{\tau_{open}}(1-w), & u < u_{gate}, \\ -\frac{1}{\tau_{close}} w, & u \geq u_{gate}, \end{cases} \tag{1.1.10}$$

where $0 < \tau_{in} \ll \tau_{out} \ll \tau_{open}, \tau_{close}$ are positive constants and $0 < u_{gate} < 1$. In a regular heartbeat simulation, the variables u, w are between 0 and 1. The first term on the right for the first equation represents the inward current, which is gated by w, so that the gate is open when $w = 1$, and closed when $w = 0$, while the second term represents an ungated outward current. Due to the relation between the parameters, the action potential can be described by fast-slow dynamics by studying the phase-plane. The nullcline for the first equation consists of a short depolarization of order τ_{in}, a long plateau of order τ_{close}, a short repolarization of order τ_{out} and recovery of order τ_{open}. See [75] for details. Unlike the FHN model, the paramters of the MS model can be chosen to match the duration of each phase of the action potential (see [86]).

1.2 Finite element method

1.2.1 Lagrangian finite elements

In this thesis, a mesh $\mathcal{T}_h$ is an assemblage of a finite number of d-dimensional simplicial elements K with finite and nonzero volume such that $\overline{\Omega} = \cup_{K \in \mathcal{T}_h} K$, and such that any pair of elements K_1, K_2 have disjoint interior $\mathring{K}_1 \cap \mathring{K}_1 = \emptyset$. The subscript h refers to the largest diameter $h = \max_{K \in \mathcal{T}_h} h_K$, where h_K is the diameter of the element K. Fixing a reference simplex $\hat{K}$, for every element $K \in \mathcal{T}_h$ there exists an invertible affine transformation $F_K : \hat{K} \to K$. Denoting the vertices $\{\hat{v}_i\}$ and $\{v_i\}$ for the respective elements, the transformation is unique up to the condition $F_K(\hat{v}_i) = v_i$. The mesh is assumed to be geometrically conformal: given two elements K_1, K_2 such that $K_1 \cap K_2$ is a $(d-1)$-dimensional face E, then there is a face $\hat{E}$ of the reference simplex such that up to a reordering of the nodes of K_1, K_2, $E = F_{\hat{K}_1}(\hat{E}) = F_{\hat{K}_2}(\hat{E})$

and $F_{\hat{K}_1}|_{\hat{E}} = F_{\hat{K}_2}|_{\hat{E}}$. In particular, for $d = 2$, the intersection of any two elements will be either empty, a vertex, or an edge.

Denote p_K the diameter of the largest ball inscribed in K. A family of meshes $\{\mathcal{T}_h\}_h$, indexed by the maximum diameter h, is said to be *shape-regular* if there exists a constant $c > 0$ such that $\max_h\{\max_{K\in\mathcal{T}_h} \frac{h_K}{p_K}\} \leq c$. For $d = 2$, denoting by θ_K the smallest angle in K, we have the inequality $\frac{h_K}{p_K} \leq \frac{1}{\sin\theta_K}$, so a shape-regular family of meshes satisfies a minimum-angle condition. In what follows, we say that a family of meshes satisfying a shape-regularity condition is *isotropic*. Otherwise, it is *anisotropic*.

We specialize to the case $d = 2$. Let $\ell \geq 1$ be an integer. For an element $K \in \mathcal{T}_h$, let $P_\ell(K)$ be the space of polynomials on K of degree at most ℓ. This space has dimension $n_\ell = \frac{1}{2}(\ell + 1)(\ell + 2)$. We define a basis as follows. Consider the local nodes $\{a_{K,i}\}_{i=1}^{n_\ell}$ in K with Barycentric coordinates $\left(\frac{i_0}{\ell}, \frac{i_1}{\ell}, \frac{i_2}{\ell}\right)$, where i_0, i_1, i_2 are all triplets of non-negative integers such that $i_0 + i_1 + i_2 = \ell$. Define the linear forms $\sigma_i : P_\ell(K) \to \mathbb{R}$ by $\sigma_i(p) = p(a_{K,i})$. Then $\Sigma = \{\sigma_i\}_{i=1}^{n_\ell}$ forms a basis of the dual of $P_\ell(K)$. The dual basis $\{\theta_i\}_{i=1}^{n_\ell} \subseteq P_\ell(K)$, consisting of the polynomials satisfying $\sigma_i(\theta_j) = \delta_{ij}$, are called the local shape functions, and the set $(K, P_\ell(K), \Sigma)$ is called a Lagrangian finite element.

1.2.2 Finite element approximation

Denote by V_h the Lagrangian finite element space:

$$V_h = \{v_h \in C(\overline{\Omega}) : v_h|_K \in P_\ell(K),\ \forall K \in \mathcal{T}_h\}.$$

The space V_h is $H^1(\Omega)$-conformal, in the sense that $V_h \subseteq H^1(\Omega)$. The global nodes $\{a_i\}_{i=1}^N$ are obtained as the union of all the local nodes for each element K. A basis $\{\phi_i\}$ of V_h consists of the functions that satisfy $\phi_i(a_j) = \delta_{ij}$. Define the space $V_{0,h} = H_0^1(\Omega) \cap V_h$, with basis consisting of the ϕ_i for which a_i is an internal node.

Let g_h be a piecewise linear interpolation of g. The finite element approximation $u_h \in V_h$ to problem (1.1.5) is obtained by restricting the variational form:

$$\begin{aligned}(D\nabla u_h,\, \nabla v_h)_\Omega = (f,\, v_h)_\Omega\,, \quad \forall v_h \in V_{0,h},\\ u_h|_{\partial\Omega} = g_h.\end{aligned}$$

The convergence of this method under certain regularity assumptions can be summarized in the following.

Theorem 1.2.1. *[40] Suppose that the solution u to (1.1.5) belongs to $H^{\ell+1}(\Omega)$, and let $\{\mathcal{T}_h\}_h$ be a family of shape regular meshes. Then there exists a constant $C > 0$ independent of h such that the P_ℓ approximation satisfies*

$$\|u - u_h\|_{0,\Omega} + h\|u - u_h\|_{1,\Omega} \leq Ch^{\ell+1}\|u\|_{\ell+1,\Omega}.$$

Due to Cea's Lemma [40, Lemma 2.8] the finite element approximation u_h is in some sense the best approximation to u in the approximation space V_h. Therefore, bounds on local interpolation error lead to bounds for the global finite element approximation error. The dependence on shape-regularity for the constant C in Theorem 1.2.1 is due to classical bounds on the Lagrange interpolation operator. In practice, the shape-regularity requirement may be too restrictive. This restriction can be weakened somewhat by replacing the minimum-angle condition with a maximum-angle condition, see [5]. In this thesis, we do not want to impose any a priori global constraint on the type of elements allowed.

1.2.3 Anisotropic setting

In this thesis, we make use of the anisotropic setting from [44]. For a triangular element K, consider the affine mapping $F_K : \hat{K} \to K$. The reference element $\hat{K}$ is taken to be the equilateral triangle centred at the origin with vertices at the points $(0,1)$, $(\frac{-\sqrt{3}}{2}, \frac{-1}{2})$, $(\frac{\sqrt{3}}{2}, \frac{-1}{2})$. The Jacobian J_K of F_K is non-degenerate, so that the singular value decomposition (SVD) $J_K = \mathcal{R}_K^T \Lambda_K \mathcal{R}_K \mathcal{Z}_K$ consists of orthogonal matrices $\mathcal{R}_K$, $\mathcal{Z}_K$, and a positive definite diagonal matrix Λ_K. The matrices $\mathcal{R}_K$, Λ_K take the form

$$\mathcal{R}_K = \begin{pmatrix} r_{1,K}^T \\ r_{2,K}^T \end{pmatrix}, \qquad \Lambda_K = \begin{pmatrix} \lambda_{1,K} & 0 \\ 0 & \lambda_{2,K} \end{pmatrix},$$

where $\lambda_{1,K} \geq \lambda_{2,K} > 0$, $r_{1,K}$, $r_{2,K}$ are orthogonal unit vectors. Geometrically, the singular values and vectors represent the deformation of the unit ball in $\mathbb{R}^2$ to an ellipse with axes of length $\lambda_{1,K}$, $\lambda_{2,K}$ in directions $r_{1,K}$, $r_{2,K}$, respectively. Moreover, they represent K in the sense that the ellipse circumscribes the element.

Denote by Δ_K the patch of all elements which contain a vertex of K. While shape-regularity is not imposed for individual elements, some mild conditions are imposed on the type of patches allowed, see for instance [80], [74]. For $v \in H^1(\Omega)$, define the following "Hessian" type matrix:

$$\tilde{G}_K(v) = \left(\int_{\Delta_K} \frac{\partial v}{\partial x_i} \frac{\partial v}{\partial x_j} \, \mathrm{d}x \right)_{i,j},$$

and let

$$\tilde{\omega}_K(v) = (\lambda_{1,K}^2 r_{1,K}^T \tilde{G}_K(v) r_{1,K} + \lambda_{2,K}^2 r_{2,K}^T \tilde{G}_K(v) r_{2,K})^{1/2}.$$

As $d \geq 2$, there is no embedding $H^1(\Omega) \subseteq C(\overline{\Omega})$, so instead of the usual Lagrange interpolation operator, one uses a quasi-interpolation operators $Q_h : H^1(\Omega) \to V_h$, for instance the Clement or the Scott-Zhang operators [29], [88]. Then we have the following estimates.

Lemma 1.2.2 ([44][45]). *Let $Q_h : H^1(\Omega) \to V_h$ denote the Clement or Scott-Zhang interpolation operator. Then there exists a constant $C_{\hat{K}} > 0$, such that for all $v \in H^1(\Omega)$*

1. $\|v - Q_h v\|_{0,K} \leq C_{\hat{K}} \omega_K(v)$,
2. $\|v - Q_h v\|_{0,\partial K} \leq C_{\hat{K}} \left(\frac{h_K}{\lambda_{1,K}\lambda_{2,K}}\right)^{1/2} \omega_K(v)$.

As noted in [74], the constant $C(\hat{K})$ does not depend on a minimum-angle condition, but instead on mild conditions related to the patch Δ_K. More specifically, it is assumed that the cardinality (number of elements) of all such patches, as well as the diameter of $F_K^{-1}(\Delta_K)$ are uniformly bounded. These interpolation estimates have been used to derive optimal order a priori error estimates, see for instance [45] and [74]. More significantly, they have been used to derive a posteriori error estimates in a variety of settings. In this thesis we build on these results.

1.2.4 A posteriori error estimates

A posteriori error estimates do not involve any information about the unknown solution, as in Theorem 1.2.1. Additionally, the estimate should be "local", giving information about the error on an element, edge, patch, etc. In what follows we give some examples from the literature.

Gradient recovery and superconvergence

For Lagrange finite elements, the gradient ∇u_h generally converges at a lower rate (i.e. convergence in the H^1-seminorm) than u_h (i.e. convergence in the L^2-norm) due to the fact that the gradient belongs to a lower order finite element space. To recover higher-order convergence, one can apply a gradient recovery operator of the form $\Pi_h : V_h \to V_h \oplus V_h$. The operator should be superconvergent in the sense that $\Pi(u_h)$ converges to ∇u faster than ∇u_h.

A popular recovery operator is the Zienkiewicz-Zhu (ZZ) estimator used in [87], simplified from the original [102]. The estimator is constructed by a mass-lumped integral:

$$ZZ(u_h)(p) = \sum_{K \in \Delta_p} \frac{|K|}{|\Delta_p|} \nabla u_h|_K,$$

where Δ_p is the patch of elements containing the vertex p, and where $|\cdot|$ denotes the Lebesgue measure on $\mathbb{R}^d$. Define the estimator $\eta^{ZZ} = \|\nabla u_h - ZZ(u_h)\|_{0,\Omega}$. In [87] it is shown that η^{ZZ} is equivalent to the exact error in the sense that there exists C_1, $C_2 > 0$ independent of h (under shape-regularity requirements) such that

$C_1\eta^{ZZ} \leq |u - u_h|_{1,\Omega} \leq C_2\eta^{ZZ}$. Additionally, it is shown that for certain classes of uniform meshes (i.e. parallel), the gradient is superconvergent.

Another recovery operator that was introduced in [101], based on a local least squares approximation at solution nodes, improves on the approximation properties of the ZZ recovery operator. Superconvergence results were extended to further classes of regular meshes for which the ZZ estimator is not superconvergent, including the chevron pattern. Moreover, their recovery operator Π_h^k preserves polynomials of order $k+1$. That is, if u is polynomial of order $k+1$, and u_h is the Lagrange interpolant of order k, then $\Pi_h^k u_h = \nabla u$. This property holds for the ZZ estimator for only certain classes of uniform meshes.

Hierarchical

Hierarchical estimates use the idea that higher-order approximate solutions should be more accurate. From the initial solution space V_h, one considers the "enriched" solution space $\tilde{V}_h = V_h \oplus W_h$. The error $e_h = u - u_h \in V$ is then approximated by a function $\tilde{e_h} \in W_h$. For example, if V_h is a P_ℓ approximation space, then one may take $\tilde{V}_h$ to be the $P_{\ell+1}$ approximation space. Locally, one has $P_{\ell+1}(K) = P_\ell(K) \oplus C_{\ell+1}(K)$, where $C_{\ell+1}(K)$ is the set of monomials of degree $\ell+1$. A variety of approaches exist for constructing hierarchical estimates. In [57], the error is approximated by solving an auxiliary problem. For a finite element approximation $u_h \in V_h$ which solves $a(u_h, v_h) = f(v_h), \forall v_h \in V_h$, where a is a bilinear form on V_h, and f is a linear functional, the error is approximated by function $\tilde{e}_h \in W_h$ by solving $a(\tilde{e}_h, w_h) = f(w_h) - a(u_h, w_h), \forall w_h \in W_h$. A similar approach is taken in [98] by solving a local Neumann or Dirichlet problem. For general results on such methods, see for instance [40]. A problem-independent approach was taken in [17], where the approximate error function is constructed with the use of gradient recovery techniques. In addition, their estimator does not rely on a mesh regularity condition, and therefore applies in anisotropic settings.

Hessian

Hessian type estimators are based on reconstruction of the Hessian of the solution. This idea is related to the observation that the interpolation error can be related to the Hessian of the solution, for example see [53], [48]. The Hessian is treated as a Riemannian metric, defined discretely at nodal values. This point of view leads very naturally to anisotropic mesh adaptation. A crucial step in applying such estimators is approximating the Hessian from the approximate solution. For the P_1 finite element space V_h, the Hessian recovery operator is a function $P_H : V_h \to M_2(V_h)$, where $P_h(a_i)$ is a symmetric matrix for each node a_i. A simple, yet effective, method can be found in [53], which applies a variational formulation with lumped mass integration.

Comparison studies on the performance and approximation properties of various Hessian recovery techniques can be found in [97] and in [82]. In general, the recovered Hessian does not need to converge in norm to the exact Hessian of the solution in order for the estimated error to properly represent the exact error for the problem, but a much weaker "closeness" assumption should be satisfied, see [58, Theorem 2.1].

Residual estimates

In this thesis, we focus on explicit residual type estimators for the energy norm of the error. These methods take advantage of Galerkin orthogonality for the variational equation of the finite element approximation to represent the error in terms of element and edge residuals. These residuals are local estimates, generally obtained by integration by parts. For standard results on such estimators, see [6], [1], [98], and for a more modern approach, see [45], [80], [72], [63]. The latter group vary from the former by applying anisotropic interpolation estimates and gradient recovery techniques. The estimators are generally of two different forms: explicit or dual-based. In the dual-based approach, one is interested in controlling an arbitrary (possibly nonlinear) functional of the error. The estimator consists of products of stability constants, computed by approximating a dual problem, with residual terms. See [41] for a detailed review of the methods.

Error estimation and this thesis

From the methods mentioned above, in this thesis we are only concerned with anisotropic error estimation. In particular, we focus on the general anisotropic setting from [80]. We are interested in extending the results in a few directions. Typically, an explicit residual estimator is developed to control the energy norm of the error for elliptic problems. We address the question of how well such estimators, in this setting, can be used to control the error in other norms, specifically the L^2-norm. Additionally, we would like to extend the results on these error estimates to further nonlinear settings, namely to nonlinear reaction-diffusion systems, and to the systems for electrophysiology. There has been some work on residual a posteriori error estimation in electrophysiology, however, none of the results have been for anisotropic estimators and generally apply a dual-based method, for instance in [41].

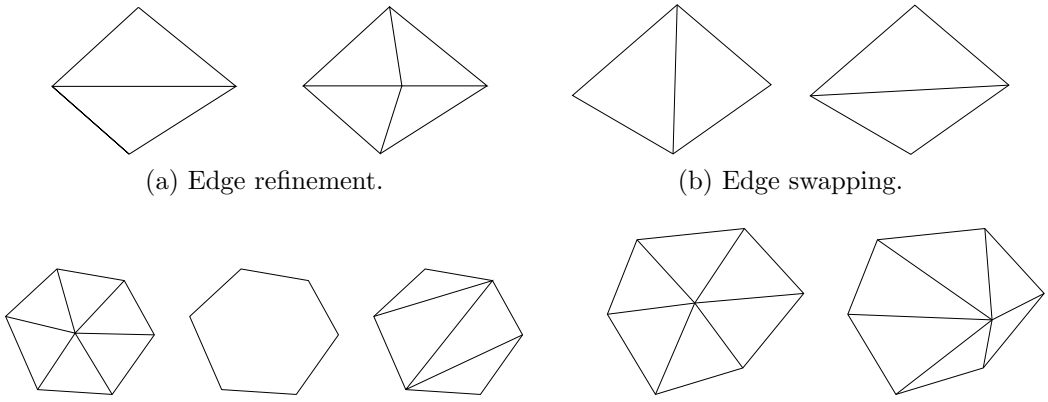
1.2.5 Mesh adaptation

The technique of mesh adaptation uses computed data, such as the approximate solution, error estimates, etc., to find a new optimal mesh. There are many approaches, ranging from isotropic methods, which respect conditions such as mesh-regularity, see [98], to anisotropic methods where as few such mesh requirements are considered as possible. In this thesis, we consider anisotropic methods for $2D$ simplicial meshes.

The adaptation will combine local operations, illustrated in Figure 1.1. The operations involve two types of element patches: the patch Δ_e denotes the patch of elements containing the edge e, and the patch Δ_p denotes the patch of elements containing the vertex p.

- **Edge refinement**: Given an edge e, add a new node at the midpoint, splitting each element adjacent to e into two new elements.
- **Edge swapping**: Given an edge e with two adjacent elements, reconnect the edge using the other two vertices in the patch Δ_e.
- **Node removal**: Remove the node p and all elements in the patch Δ_p, then remesh the resulting "hole".
- **Node displacement**: Choose a new coordinate for the node p within the patch Δ_p.

Note that edge swapping cannot be performed if the reconnected edge results in overlapping elements. Also, when performing node displacement, the patch Δ_p must remain star-shaped with respect to the new coordinate for p. To drive these local operations, one uses error estimators. A few approaches are used in this thesis.



(a) Edge refinement. (b) Edge swapping.

(c) Node removal. (d) Node displacement.

Figure 1.1: Local mesh operations.

Metric based adaptation

A popular method for anisotropic mesh adaptation is to use a Riemannian metric. The metric introduces the notion of path length in the domain, and in particular, defines edge length. The idea is to construct an optimal mesh where all edges have a length of 1 in the Riemannian metric, and this is what determines the criteria for local mesh

modification. Edge refinement is performed to edges which are too long, while node removal is performed if an edge attached to a node is too short. The other operations attempt to equidistribute the edge length, applying a non-Euclidean analogy of the Delaunay criterion for edge swapping, and a spring energy minimization analogy for node displacement. For a modern introduction to the theory and techniques, see [68], [69], [48].

Element-based adaptation

Another approach is to work with estimators defined on elements. In this context, the error can be approximated by local values:

$$\|e_h\| \approx \eta = \left(\sum_{K \in \mathcal{T}_h} \eta_K^2 \right)^{1/2},$$

where η_K is computed locally on each element. The goal of mesh adaptation is to control the global level of error by requiring that $\eta \leq TOL$ for some global tolerance $TOL > 0$. Moreover, the error should be equidistributed over the elements in some way. Classically, this requirement has been achieved by refinement where the error is large and derefinement where the error is small, for instance [98], done in such a way to respect mesh regularity conditions. More recently in [17], this approach was extended to a completely anisotropic setting using a hierarchical error estimator. The operations edge swapping and node discplacement were performed in order to minimize the energy norm of the error, while edge refinement and node removal were used to control the global L^2-norm of the error.

Element to metric conversion

With an element-based estimator, a common approach is to convert the estimator into a metric in order to apply mesh adaptation. Thus the control of the error on elements is transformed into a control of the error on edges. See [80], [72] for details.

Mesh adaptation and this thesis

A significant number of anisotropic mesh adaptation methods have been introduced in the literature, which highlights the need for a comparative study of the performance of these methods. In particular, if we would like to control the error in a given norm, what is the best method to achieve a level of error in the least amount of CPU time, or with the least amount of memory usage? Note that while an adaptation method might be developed to control the error in one norm, it might effectively control the error in other norms as well, so that it is worthwhile to compare methods designed for various situations. Another question arises for the anisotropic residual estimator.

While they are typically used for mesh adaptation by converting the error into a metric, this requires an averaging over adjacent nodes. Does this averaging affect the equidistribution of the estimator over the elements, and therefore the control of the error, and might we achieve greater control by employing an element-based adaptation method?

1.2.6 Space-time discretization

To approximate the full problem (1.1.1) it is necessary to discretize the partial differential equation in time as well. Partition the interval $[0, T]$ as $0 = t_0 < t_1 < \cdots < t_N = T$, with time steps $\tau_k = t_k - t_{k-1}$, and denote the step-size ratio $\gamma_n = \frac{\tau_n}{\tau_{n-1}}$. The multistep backward difference formula (BDF) method of order s with $1 \leq s \leq 6$ which solves for u_h^n using the s previously computed values $\{u_h^{n-s+k}\}_{k=0}^{s-1}$ is obtained by solving the variational problem

$$\begin{aligned}&(\pi_s'(t_n),\, v_h)_\Omega + (D\nabla u_h^n,\, \nabla v_h)_\Omega + (F(u_h^n),\, v_h)_\Omega \\ &= (S,\, v_h)_\Omega + (G,\, v_h)_{\partial\Omega}\,, \qquad \forall v_h \in V_h,\end{aligned}$$

where $\pi_s : [t_{n-s}, t_n] \to V_h$ is the order s Newton polynomial that interpolates the values u_k at t_k, see [37]. The first step u_h^0 is the interpolation of the initial value u_0, while the step u_h^k for $1 \leq k \leq s-1$ are computed using a lower order method.

The BDF2 method (with $s = 2$) is known to have good stability properties applied in the context of electrophysiology, see [42], [85]. The stability is preserved in the variable step method under reasonable restrictions on the step-size ratio, see [9], [38]. Additionally, in [38], optimal order convergence results are given for a linear and nonlinear problem.

In general, if the full discretization of the problem is order $O(h^r + \tau^s)$, it is natural to seek an error estimator of the form:

$$\begin{aligned}\|e_h\| &\approx \left((\eta^S)^2 + (\eta^T)^2\right)^{1/2} \\ &= \left(\sum_{n=1}^{N} \left(\eta_n^S\right)^2 + \left(\eta_n^T\right)^2\right)^{1/2},\end{aligned}$$

where, η^S and η^T are of optimal order $O(h^r)$ and $O(\tau^s)$, corresponding respectively to the error for the space and time discretization. Adaptation in both space and time may be applied by choosing two tolerances TOL_S, TOL_T, with the goal of achieving $\eta^S \approx TOL_S$ and $TOL_T \approx \eta^T$. To give a few examples of space-time adaptation methods that are based on a posteriori residual estimators, see [73], [70] for single step methods, [41] for continuous and discontinuous Galerkin polynomial approximation methods and [46] for a Rosenbrock method. Note that it is not necessary to adapt both in space and in time. For instance, in [11] and [24] mesh adaptation is performed to control the error in space, while a constant time step is employed throughout.

Space-time adaptation and this thesis

The appeal of space-time adaptation is that we can, in principle, very accurately control the error of the solution by tuning the local mesh and step size. The hope is a gain in efficiency over uniform refinement methods, both in terms of CPU time, memory usage, and in the number of time steps performed. On the other hand, the adaptive process may introduce significant overhead. After the mesh is adapted, the system needs to be reassembled, the error estimators need to be computed throughout, and the solution may need to be recomputed with a smaller time step. The potential for gains in efficiency will be explored when applying such methods to reaction-diffusion systems and to the monodomain problem in electrophysiology. Gains in efficiency have been noticed in the literature when adapting in space with a constant time step, so this situation will also be considered for comparison. An interesting question that arises in time-dependent partial differential equations is what is the best time step? For a given space discretization, taking the time step to zero approximates the solution of the semidiscrete problem in space, not of the original problem, so it is not always the smallest time step that gives the most accurate result. If a constant time step is used, should its size be determined by the interval on which we expect the highest error? This question appears to be problem dependent, and we consider to what extent time-step adaptation may resolve this issue.

1.3 Plan of thesis

The goal of this thesis is to further build upon existing work related to a posteriori error estimation and mesh adaptation techniques for the finite element method applied to elliptic and parabolic equations. This work relates to the improved efficiency and accuracy that is to be gained from adaptive techniques. Moreover, the a posteriori error estimates that guide the adaptation process should be theoretically justified in their representation of the exact error of the problem. We address these issues in the thesis as follows.

- **Chapter 2**: This chapter deals with a purely diffusive linear elliptic equation. The first goal of the chapter is to introduce a new mesh adaptation method to control the energy norm of the error using the error estimator from [80] and [72]. The performance of the implementation of the new adaptation algorithm is thoroughly investigated. The second goal of the chapter is to compare the efficiency of the new method with three existing adaptation methods. The third goal of the chapter is to introduce a new estimator for the L^2-norm of the error.
- **Chapter 3**: We consider a nonlinear scalar reaction-diffusion equation. The primary goal is to introduce a new estimator for the full discretization of the

problem, by employing a fully implicit BDF2 discretization in time. The reliability of the estimator is proven using energy techniques, and illustrated with numerical examples. A space-time adaptation algorithm is proposed and is demonstrated to effectively control the error. The chapter addresses the question of whether improved efficiency can be expected when applying the space-time adaptation method compared to a space-only adaptation method on a case-by-case basis. The method is shown to correctly predict a suitable time-step choice, leading to improved user-independence.

- **Chapter 4**: This chapter extends the results of Chapter 3 to the monodomain problem. The primary difficulty with the system is the lack of diffusion operator for the recovery variable. The theoretical aspects of applying the estimator are discussed for different ionic models. A modified estimator for the recovery variable is proposed due to limitations of the residual estimator. Numerical examples are provided to illustrate the suitability of the adaptation algorithm for error control, and to assess the reliability of the estimator. Efficiency of the method is discussed.

Chapter 2 has been submitted for publication to The Journal of Computational and Applied Mathematics, co-authored with my thesis advisors Yves Bourgault and Thierry Giordano. My contribution, under the direction of my advisors, is the writing, the implementation and analysis of the numerical results, and the theoretical derivation of the new L^2 estimator. Chapter 3 is currently in preparation to be submitted for publication, and is therefore presented in article format.

Chapter 2

Anisotropic residual-based a posteriori mesh adaptation in 2D: element-based approach

Abstract. An element-based adaptation method is developed for an anisotropic a posteriori error estimator. The adaptation does not make use of a metric, but instead equidistributes the error over elements using local mesh modifications. Numerical results are reported, comparing with three popular anisotropic adaptation methods currently in use. It was found that the new method gives favourable results for controlling the energy norm of the error in terms of degrees of freedom at the cost of increased CPU usage. Additionally, we considered a new L^2 variant of the estimator. The estimator is shown to be conditionally equivalent to the exact L^2 error. We provide examples of adapted meshes with the L^2 estimator, and show that it gives greater control of the L^2 error compared with the original estimator.

2.1 Introduction

In the last twenty years, anisotropic mesh adaptation has seen great activity. Since the work of D'Azevedo and Simpson in [34] and [33] for piecewise linear approximation of quadratic functions there has been a significant amount of research dedicated to producing practical adaptation procedures based on their results. In addition, there has been much software written for the implementation, which either construct an entirely new mesh, such as BAMG [54], BL2D [65], GAMANIC3D [49], or apply local modifications to a previous mesh, such as MEF++ [50], MMG3D [36], YAMS [78]. The main idea they share in common is to construct a non-Euclidean metric from the Hessian of the solution. We will refer to them as Hessian adaptation methods, see for instance [68], [69], [48], [18], [53].

Residual a posteriori error estimation for elliptic equations has been around for some time. In [7] and [8], Babuska and Rheinboldt introduced a local estimator, constructed entirely from the approximate solution, that is globally equivalent to the energy norm of the error. Numerical results showed that it was suitable for the purposes of mesh adaptation by determining regions in which the mesh could be refined or coarsened. While initially an entirely isotropic method, recently, the residual method was modernized by the introduction of anisotropic interpolation estimates from [44]. Unlike classical results, the new estimates did not require a minimum (or maximum) angle condition, and instead took into account the geometric properties of the element. In [80] and [45] these interpolation results were combined with the standard a posteriori estimates to drive mesh adaptation by constructing a metric. We will refer to this method as the residual metric method. The method results in highly anisotropic meshes, reducing the error by an order of magnitude compared to isotropic methods [80]. Moreover, the procedure has been successfully applied to a variety of nonlinear situations, including a reaction-diffusion system to model solutal dendrites in [24] and the Euler equations to model the supersonic flow over an aircraft in [22].

Recent work in [17] demonstrates the potential advantages of element-based anisotropic mesh adaptation over the usual metric based mesh adaptation methods used so far. The error estimator they use is hierarchical: from a given approximate solution, they construct a higher-order, more accurate approximation. For the Hessian method it is necessary to take the absolute value of the eigenvalues of the Hessian, thus treating positive and negative curvature as essentially equal, while the distinction can be seen very clearly in meshes adapted with the hierarchical method. Further, the hierarchical estimator has the advantage that it can naturally be applied to finite elements of arbitrary order.

The primary goal of this paper is to introduce, and numerically assess, an element-based adaptation approach to be used with the residual estimator from [80]. We will refer to this method as the element-based residual method. Motivation for implementing such a method includes avoiding the additional steps involved in converting the estimator defined on elements, to a metric defined on the nodes, during which information could be lost. Additionally, we would like to attempt to mimic the success of the hierarchical method. The adaptation will be implemented by interfacing the estimator with the hierarchical adaptation code MEF++. We also introduce a variant of the estimator for the L^2 norm error, which is shown to be reliable and efficient under certain assumptions, and show that the estimator is also suitable for anisotropic mesh adaptation. A secondary goal of the paper will be to provide a comparative performance analysis between four different adaptation techniques: element-based residual, metric based residual, Hessian, and hierarchical.

The outline of this paper is as follows: in Section 2 we introduce the model problem and error estimator, as well as recall some results from the literature; in

Section 3 we discuss both the metric and element-based adaptation procedures; in Section 4 we produce numerical results, validating the element-based method, and comparing it with other anisotropic adaptation procedures.

2.2 The estimator

We discuss the model problem and introduce a residual estimator. Main results will be summarized from the literature. Full details can be found for instance in [44], [80], and [72].

2.2.1 Model problem

Let $\Omega \subseteq \mathbb{R}^2$ be a bounded polygonal domain, with boundary $\partial\Omega$. Let $V = H^1(\Omega)$ and $V_0 = \{v \in H^1(\Omega) : v|_{\partial\Omega} = 0\}$. For $g \in H^{1/2}(\partial\Omega)$, let $V_g = \{v \in H^1(\Omega) : v|_{\partial\Omega} = g\}$, which may be thought of as the translation of V_0 by g. For $f \in L^2(\Omega)$, and a positive definite matrix A, let $u \in V_g$ be the solution of the equation

$$\begin{cases} -\mathrm{div}(A\nabla u) = f, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega. \end{cases} \tag{2.2.1}$$

Then u is the solution to the variational equation

$$B(u, v) = F(v), \qquad \forall v \in V_0,$$

where

$$B(u, v) = \int_\Omega A\nabla u \cdot \nabla v \, \mathrm{d}x, \qquad u \in V_g, v \in V_0,$$
$$F(v) = \int_\Omega fv \, \mathrm{d}x, \qquad v \in V_0.$$

For $h > 0$, let $\mathcal{T}_h$ be a conformal triangulation of Ω consisting of triangles K with diameter $h_K \leq h$. Denote by V_h the finite element space of continuous, piecewise linear functions (P_1) on $\mathcal{T}_h$ and $V_{h,0}$ the subspace of functions vanishing on $\partial\Omega$. Let g_h be a piecewise linear approximation of g on $\partial\Omega$ and let $V_{h,g} = \{v_h \in V_h : v_h|_{\partial\Omega} = g_h\}$. Then the finite element approximation $u_h \in V_{h,g}$ of u satisfies the discrete variational equation

$$B(u_h, v_h) = F(v_h), \qquad \forall v_h \in V_{h,0}. \tag{2.2.2}$$

For details on the finite element method for elliptic problems, see for instance [84].

2.2.2 Anisotropic residual error estimator

Define the energy norm by $|||v||| = B(v,v)^{1/2}$ for $v \in V$. The residual mesh adaptation procedure is based on controlling the energy norm of the discretization error $e_h = u - u_h$. The error estimator, which will be outlined below, combines information of the residual with anisotropic interpolation estimates.

Define the localized residual by

$$R_K(u_h) = f + \operatorname{div}(A\nabla u_h),$$

where the divergence operator is local to K. The jump of the derivative for an element K with edges e_i is defined by

$$r_K(u_h) = \sum_{i=1}^{3} [A\nabla u_h]_{e_i},$$

where the jump $[A\nabla v_h]_{e_i}$ over e_i is defined as follows: denoting the outward unit normal by n_i and the adjacent element (if it exists) by K', then

$$[A\nabla v_h]_{e_i} = \begin{cases} 0, & e_i \in \partial\Omega, \\ A\nabla(v_h)|_K \cdot n_i - A\nabla(v_h)|_{K'} \cdot n_i, & \text{otherwise.} \end{cases}$$

For a triangular element K, the anisotropic information comes from the affine mapping $F_K : \hat{K} \to K$. The reference element $\hat{K}$ is taken to be the equilateral triangle centred at the origin with vertices at the points $(0,1)$, $(\frac{-\sqrt{3}}{2}, \frac{-1}{2})$, $(\frac{\sqrt{3}}{2}, \frac{-1}{2})$. The Jacobian J_K of F_K is non-degenerate, so the singular value decomposition (SVD) $J_K = \mathcal{R}_K^T \Lambda_K \mathcal{R}_K \mathcal{Z}_K$ consists of orthogonal matrices $\mathcal{R}_K$, $\mathcal{Z}_K$, and positive definite diagonal matrix Λ_K. The matrices $\mathcal{R}_K$, Λ_K take the form

$$\mathcal{R}_K = \begin{pmatrix} r_{1,K}^T \\ r_{2,K}^T \end{pmatrix}, \qquad \Lambda_K = \begin{pmatrix} \lambda_{1,K} & 0 \\ 0 & \lambda_{2,K} \end{pmatrix},$$

where $\lambda_{1,K} \geq \lambda_{2,K} > 0$, $r_{1,K}$, $r_{2,K}$ are orthogonal unit vectors. Geometrically, these eigenvalues and eigenvectors represent the deformation of the unit ball in $\mathbb{R}^2$ to an ellipse with axes of length $\lambda_{1,K}$, $\lambda_{2,K}$ in directions $r_{1,K}$, $r_{2,K}$ respectively. Moreover, they represent K in the sense that the ellipse circumscribes the element.

Denote by Δ_K the patch of elements containing a vertex of K. As noted in [74], for the bounds for the quasi-interpolation operator to be uniform, there must be an integer $\Gamma > 0$ and a constant $C > 0$ such that all such patch satsifies $\operatorname{card}(\Delta_K) \leq \Gamma$ (cardinality) and $\operatorname{diam}(F_K^{-1}(\Delta_K)) \leq C$ (diameter). For $v \in V$, define the following "Hessian" type matrix:

$$\tilde{G}_K(v) = \left(\int_{\Delta_K} \frac{\partial v}{\partial x_i} \frac{\partial v}{\partial x_j} \, \mathrm{d}x \right)_{i,j}, \tag{2.2.3}$$

and let

$$\tilde{\omega}_K(v) = (\lambda_{1,K}^2 r_{1,K}^T \tilde{G}_K(v) r_{1,K} + \lambda_{2,K}^2 r_{2,K}^T \tilde{G}_K(v) r_{2,K})^{1/2},$$

Finally, define

$$\hat{\eta}_K^2 = \left(\|R_K(u_h)\|_{0,K} + \left(\frac{h_K}{\lambda_{1,K}\lambda_{2,K}} \right)^{1/2} \|r_K(u_h)\|_{0,\partial K} \right) \tilde{\omega}_K(e_h). \tag{2.2.4}$$

Theorem 2.2.1 ([80][81][72]). *There exist constants $C_{1,\hat{K}}$, $C_{2,\hat{K}} > 0$ such that*

$$C_{1,\hat{K}} \sum_K \hat{\eta}_K^2 \leq |||e_h|||^2 \leq C_{2,\hat{K}} \sum_K \hat{\eta}_K^2.$$

The upper and lower bounds in Theorem 2.2.1 still depend on the unknown solution due to the $\tilde{\omega}_K(e_h)$ term. The approach taken in [80] is to remove this dependency by using a gradient recovery operator of the form $\Pi : V_h \to V_h \oplus V_h$. The operator should be super-convergent in the sense that $\Pi(u_h)$ converges to ∇u faster than ∇u_h, at least of order $1+\epsilon$ where $\epsilon > 0$. For full details on the derivation of upper and lower bounds using super-convergence assumptions, see [72]. Therefore, for the remainder we replace $\tilde{G}_K(e_h)$ by

$$G_K(u_h) = \left(\int_K \left(\frac{\partial u_h}{\partial x_i} - \Pi(u_h)_i \right) \left(\frac{\partial u_h}{\partial x_j} - \Pi(u_h)_j \right) \mathrm{d}x \right)_{i,j},$$

and $\tilde{\omega}_K(e_h)$ by $\omega_K(u_h) = (\lambda_{1,K}^2 r_{1,K}^T G_K(u_h) r_{1,K} + \lambda_{2,K}^2 r_{2,K}^T G_K(u_h) r_{2,K})^{1/2}$ and the estimator becomes

$$\eta_K^2 = \left(\|R_K(u_h)\|_{0,K} + \left(\frac{h_K}{\lambda_{1,K}\lambda_{2,K}} \right)^{1/2} \|r_K(u_h)\|_{0,\partial K} \right) \omega_K(u_h). \tag{2.2.5}$$

Note furthermore, that the integral for the matrix $\tilde{G}_K(e_h)$ is taken on the patch Δ_K while that for $G_K(u_h)$ is taken only on the element K. We found simplification works in practice and greatly reduces the computational complexity of the estimator, and has been used for instance in [80], [70].

2.2.3 Gradient recovery

Here we discuss briefly our choice of gradient recovery method. A popular choice is the simplified Zienkiewicz-Zhu (ZZ) operator, see [87], which generally performs very well. For instance, on certain regular meshes (parallel) it is asymptotically exact. Moreover, despite the fact that it cannot be proven to be super-convergent for non-regular meshes, in practice superconvergence has been observed for adapted meshes, as in [80] and [72].

An improved method is proposed by Zhang and Naga in [101]. The main idea is that for each node, one fits the solution values to a higher-order polynomial on a surrounding patch, the fit being obtained in a least-square sense. The value of the recovered gradient at the node is obtained by taking the gradient of the higher-order polynomial. They prove that the method is super-convergent for any regular mesh pattern, including situations where the ZZ estimator is not, such as the chevron pattern [87]. In addition, while the ZZ estimator only preserves polynomials of degree 1, their method can be extended to higher-order elements.

In this paper we have chosen to use the recovery method of Zhang and Naga due to an observed increase in performance. We remark that the usual justification of use of the ZZ estimator is its low cost. However, the gradient recovery is only computed once at the start of each iteration of the adaptation loop. As it turns out in our case, calculation of the Zhang/Naga gradient recovery accounted for less than 0.5% of the total CPU time.

2.3 Adaptive procedure

In this section, we describe the four mesh adaptation methods that will be compared in Section 2.4, starting with the new, element-based adaptation procedure for the residual estimator η_K. The section concludes with a discussion of the control of the L^2 norm error vs. the H^1 seminorm error.

2.3.1 Element-based adaptation

By Theorem 2.2.1, the estimator $\eta = (\sum_K \eta_K^2)^{1/2}$ is globally equivalent to the energy norm of the error $|||e_h|||$. Given an error tolerance $TOL > 0$, the adaptation algorithm will attempt to control the error so that $\eta \approx TOL$. Moreover, the mesh should have the least possible number of elements N_T. Therefore, the primary goal of the adaptation algorithm is to equidistribute the estimated error by asking that every element K satisfies $\eta_K^2 \approx \frac{TOL^2}{N_T}$. From an initial calculation of η_K we adapt the mesh by performing the following local mesh modifications: edge refinement, edge swapping, node removal, and node displacement. For a complete description of local mesh modifications, see for instance [17], [32], [48], [53].

For convenience we define two element patches, which will be referred to frequently while discussing the local modifications. For an edge e, the patch Δ_e will denote the patch of elements containing e. Similarly, for a vertex p, the patch Δ_p will denote the patch of elements containing p.

Edge Refinement

Edge refinement is used to decrease the level of error where it is too large. The candidate edges for refinement are those belonging to an element K for which $\eta_K^2 > 1.5\frac{TOL^2}{N_T}$. For such an edge e with associated edge patch Δ_e, denote by Δ_e' the resulting patch after refining e, and suppose they have respectively $N_{T,e}$, $N_{T,e'}$ elements. Denote respctiveley by $\eta_{\Delta_e}^2$ and $\eta_{\Delta_e'}^2$ the error on the patch before and after refinement. The refinement is accepted if the new error is closer to the goal in the following sense:

$$\left|\frac{\eta_{\Delta_e'}^2}{N_{T,e'}} - \frac{TOL^2}{N_T}\right| < \left|\frac{\eta_{\Delta_e}^2}{N_{T,e}} - \frac{TOL^2}{N_T}\right|. \tag{2.3.1}$$

Edge swapping

Edge swapping is used to minimize the error without changing the number of elements. For an internal edge e, consider the edge patch Δ_e, test the reconnection of the edge, and denote this patch $\Delta_{e'}$. Note that it may be geometrically impossible to swap an edge, for instance if the patch is not convex, or degenerates to a triangle. Edge swapping is performed if the global error decreases. At first, one might try swapping if the following criterion holds:

$$\sum_{K' \in \Delta_{e'}} \eta_{K'}^2 < \sum_{K \in \Delta_e} \eta_K^2.$$

However, for the residual estimator, the above criteria is not enough, and we had to enlarge the patch as in Figure 2.1. Note that swapping the edge changes the normal jump of the derivative for elements adjacent to Δ_e. Including these elements in the error calculation means that we have included all elements for which η_K is changed by swapping, so that if the error decreases on the patch, then in fact the error will have decreased globally.

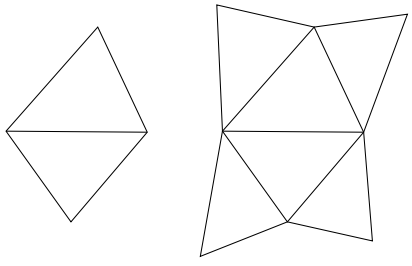


Figure 2.1: Extension of the edge patch for edge swapping.

The edges are stored in an ordered list, and edge swapping is carried out by looping over this list and checking all internal edges. When an edge is swapped, the

new edge is placed at the end of the list, so will be considered for swapping again. After the list is exhausted, if any edges were swapped the entire procedure will be repeated.

We remark that the loop will in fact terminate. There are only finitely many ways to reconfigure the mesh by swapping, and the edges are only swapped if the global error decreases. Furthermore, because no interpolation of the solution takes place during edge swapping, the map from edge configuration to global error calculation is well-defined, so it is not possible to arrive at a previous configuration but with smaller error. Also, as noted on p. 1339 of [80], the choice of reference element means that the contribution of the SVD will not depend on a reordering of the nodes.

Node removal

Node removal is used to reduce the number of mesh elements where possible, particularly where the error is small. Node removal consists in removing a node p from the mesh, as well as the patch of elements, Δ_p, attached to the node. The resulting "hole" then is remeshed, and we will call the resulting patch Δ_p'. The initial choice of remeshing is not important because the optimal choice will be determined by edge swapping. One compares the error before and after the procedure, denoted η_{Δ_p}, $\eta_{\Delta_p'}$, and the node is accepted for removal if the following analogue to (2.3.1) holds

$$\left| \frac{\eta_{\Delta_p'}^2}{N_{T,p'}} - \frac{TOL^2}{N_T} \right| < \left| \frac{\eta_{\Delta_p}^2}{N_{T,p}} - \frac{TOL^2}{N_T} \right|. \tag{2.3.2}$$

Node displacement

The goal of node displacement is to equidistribute the error over the mesh elements. Node displacement is applied to each vertex p to determine the optimal position of the vertex within the vertex patch Δ_p. Note that this patch might not be convex, so care has to be taken to avoid overlapping elements. We consider the value of the error on the elements as a discrete distribution, and find the position within the patch which minimizes the variance: $\min_p \left(\mathrm{Var}_{K \in \Delta_p} \{\eta_K^2\} \right)$. No attempt is made to solve the minimization problem fully for each vertex, but only to find an approximate solution with one iteration of a gradient recovery method. Computing the full solution could be costly, and moreover might not even be possible depending on the shape of the function being minimized. Instead, one applies several iterations of the global node displacement procedure. As we will see in Section 2.3.2, node movement minimizes a different function in the hierarchical method from [17].

Edge refinement and node removal work towards achieving the error tolerance, node movement "smooths" the mesh by equidistributing the error, while edge swapping minimizes the error.

After a mesh operation is performed the error estimator needs to be recalculated. First we interpolate the continuous data, which in this case is u_h and the recovered gradient $\Pi(u_h)$. The discontinuous data needs to be recalculated on each element, i.e. the singular value decomposition, discontinuous gradient, jump of the derivative, and the residual. Additionally, after each operation is performed there is a check to ensure degenerate elements were not produced.

All numerical results are produced with MEF++. The hierarchical estimator adaptation driver was used, described in [17], suitably adjusted.

2.3.2 Hierarchical

We summarize the ideas from [17]. Given a P_k approximation $u_{h,k}$, construct a higher-order solution P_{k+1}, $\tilde{u}_{h,k+1}$, which is supposed to be more accurate. From this, one obtains an approximation of the error

$$e_h \approx \tilde{u}_{h,k+1} - u_{h,k}. \tag{2.3.3}$$

Taking $k = 1$, and the barycentric representation of the element K by $u_{h,1}|_K = u_1\lambda_1 + u_2\lambda_2 + u_3\lambda_3$, one builds $\tilde{u}_{h,2}$ in the "hierarchical" basis

$$\tilde{u}_{h,2}|_K = u_1\lambda_1 + u_2\lambda_2 + u_3\lambda_3 + 4(e_1\lambda_1\lambda_2 + e_2\lambda_1\lambda_3 + e_3\lambda_2\lambda_3),$$

where e_i denotes the mid-edge values. Taking the Zhang-Naga (or any other sufficiently accurate) recovered gradient $\Pi(u_{h,1}) = (\Pi(u_{h,1})_1, \Pi(u_{h,1})_2)$, the mid-edge values are found by enforcing consistency between the Hessian of $\tilde{u}_{h,2}$ and the derivatives of $\Pi(u_h)$:

$$\frac{\partial^2 \tilde{u}_{h,2}}{\partial x_1^2} = \frac{\partial \Pi(u_{h,1})_1}{\partial x_1}, \qquad \frac{\partial^2 \tilde{u}_{h,2}}{\partial x_2^2} = \frac{\partial \Pi(u_{h,1})_2}{\partial x_2},$$
$$\frac{\partial^2 \tilde{u}_{h,2}}{\partial x_1 \partial x_2} = \frac{1}{2}\left(\frac{\partial \Pi(u_{h,1})_1}{\partial x_2} + \frac{\partial \Pi(u_{h,1})_2}{\partial x_1}\right).$$

Having computed the higher-order solution, the adaptation process follows similarly to that used in Section 2.3.1. But since (2.3.3) gives a direct representation of the error field, one has considerably more freedom in how to calculate the error on each element. The choice in [17], and as implemented by the authors in MEF++, is to target the global error in the L^2 norm. The operations of edge refinement and node removal will be used to achieve a global level of error, while node displacement and edge swapping are used to locally equidistribute the error by minimizing the gradient of the error, i.e. the H^1-seminorm error.

In this paper, we only consider hierarchical adaptation for P_1 finite elements for the sake of comparison. However, note that (2.3.3) is quite general, and it is very easy to generalize these ideas to higher-order finite elements. The hierarchical method has been successfully applied to P_2 finite elements in [16].

2.3.3 Metric adaptation

Currently, the most popular anisotropic mesh adaptation methods in use are metric based. Here, the main idea is to control the edge length in a Riemannian metric. For a planar domain Ω, an inner product is given by a set $\{\mathcal{M}(x)|x \in \Omega\}$ of 2×2 positive definite matrices. In practice we only have a discrete approximation, consisting of a metric defined at the nodes of the mesh, the values at other points being obtained by interpolation [18]. For an edge $e = PQ$, the edge length is given by

$$|e|_{\mathcal{M}} = \int_0^1 \sqrt{e^T \mathcal{M}(P + te)e}\, \mathrm{d}t. \tag{2.3.4}$$

The goal of a metric based adaptation algorithm will be to generate meshes which are "unit" with respect to the metric. For 2D meshes this simply means that, up to some tolerance, the edges have unit length.

The metric adaptation will be done using MEF++, applying the same mesh modification operations discussed Section 2.3.1. The goal of edge refinement and node removal is to achieve unit edge length, while the second two locally equidistribute the error. More precisely, edge swapping applies a non-Euclidean variant of the classical Delaunay edge swapping criterion to maximize the minimum angle. For node movement, the edges attached to a node are seen as a network of springs with stiffness proportional to metric edge length, and the goal is to minimize the "energy" of the system. For full details see for instance [53].

2.3.4 Residual metric based

Now we describe how the residual estimator introduced in Section 2.2 can be used to define a metric. There exist at least two approaches used in the literature, both following similar principles. The one we will use is that from [72] since it resulted in unit meshes in only a few iterations. The metric is constructed locally for the element K by finding the shape of a new element K_{new} which minimizes $\eta_{K_{new}}$ up to a fixed area. From [72], Propostion 26, the minimzing shape is given by $\tilde{r}_{1,K} = p_2$, $\tilde{r}_{2,K} = p_1$, and $\tilde{s}_K = \sqrt{\frac{\alpha_{1,K}}{\alpha_{2,K}}}$, where $\alpha_{1,K} \geq \alpha_{2,K} > 0$ and p_1, p_2 are respectively the eigenvalues and eigenvectors of the normalized matrix $\frac{G_K}{|K|}$. Under these conditions, one obtains a simple relation for the error [72, p. 826]. Imposing $\eta_K \equiv \tau$, where $\tau >$ is the local error tolerance, one determines the area from this relation, and then easily recovers the optimal values $\tilde{\lambda}_{1,K}$, $\tilde{\lambda}_{2,K}$. Finally, defining $\tilde{R}_K$, $\tilde{\Lambda}_K$ in the obvious way, the metric on K is then given by $\tilde{\mathcal{M}}_K = \tilde{R}_K^T \tilde{\Lambda}_K^{-2} \tilde{R}_K$. We remark that the mesh adaptation software used for this paper requires the metric to be defined on vertices, and for this purpose we apply metric intersection. For a vertex p, the metric $\tilde{\mathcal{M}}_p$ will be defined as the intersection of all the metrics $\tilde{\mathcal{M}}_K$ over the patch Δ_p. For details on metric

intersection, see for instance [48]. Note that, alternatively to metric intersection, we found that a simple averaging procedure gave satisfactory results.

2.3.5 Hessian

The Hessian metric approach indroduced here follows the expositions from [18], [53]. The P_1 interpolation error $e_h^I = u - I_h(u)$ of a function u on an edge $\ell = [x_i, x_j]$ satisfies

$$|e_h^I|_{L^\infty(\ell)} = \frac{|\ell|^2}{8} \left| \frac{\mathrm{d}^2 u}{\mathrm{d}x^2}(\xi) \right|,$$

where ξ is some point in ℓ. The error on the edge can then approximated using the end-points of the interval

$$|e_h^I|_{L^\infty(\ell)} \approx \frac{1}{2} \left(\ell^T |H(x_i)| \ell + \ell^T |H(x_j)| \ell \right), \tag{2.3.5}$$

where $H(x)$ is the Hessian of u at x, and $|H(x)|$ is the positive semi-definite matrix obtained by taking the absolute value of the eigenvalues of the symmetric matrix $H(x)$. Fixing an error level e_D, defining the metric $\mathcal{M}_x = \frac{1}{8 \cdot e_D} |H(x)|$, and computing the edge length from (2.3.4) with the trapezoid rule give $|\ell|_{\mathcal{M}} = 1$ precisely when the approximate error (2.3.5) is equal to e_D. For a slightly different approach to Hessian methods, see [68] and [69].

Note that (2.3.5) depends on the unknown Hessian of the solution. One may obtain a piecewise linear approximation of the Hessian from the computed solution, for instance, with the least square fitting method used in [13].

2.3.6 L^2 error vs. H^1 seminorm error estimation

While the residual estimator targets the energy norm of the error, it is also interesting to determine whether we can expect to control the L^2 norm of the error. Recall that for elliptic problems such as (2.2.1), if a set of meshes uniformly satisfies the minimum (or maximum) angle condition for some angle $\theta > 0$, then there exists $C_1 > 0$ such that for any such mesh with maximum edge length $h > 0$,

$$|e_h|_{1,\Omega} \leq C_1 h. \tag{2.3.6}$$

Furthermore, the Aubin-Nitsche Lemma states that there is a $C_2 > 0$ such that the L^2 error satisfies

$$\|e_h\|_{0,\Omega} \leq C_2 h |e_h|_{1,\Omega}. \tag{2.3.7}$$

Combining (2.3.6) with (2.3.7) one concludes that there is a $C_3 > 0$ such that

$$\|e_h\|_{0,\Omega} \leq C_3 h^2. \tag{2.3.8}$$

Therefore, if we adapt the mesh to control the H^1 seminorm error, which is the case for the residual estimator, we expect higher-order convergence for the L^2 error coming from an upper bound similar to (2.3.8).

In the absence of an Aubin-Nitsche Lemma in the context of anisotropic meshes, we attempt to find a comparison at the element level between the L^2 and energy norm of the error. The ultimate goal of the analysis is to derive an L^2 norm variant of the residual estimator η_K, which could be used for mesh adaptation.

To motivate our results, we first recall two existing a posteriori L^2 error estimators. The first, in the isotropic setting is of the form $\eta_{L^2(K)} = h_K \eta_{H^1(K)}$, where $\eta_{H^1(K)}$ is an estimator for the energy norm, see (3.20) and (3.31) from [1] and 5.1 from [26]). Similarly, the authors in [63] derived an anisotropic estimate of the form $\tilde{\eta}_E = (h_{\min,E})\eta_E$, where η_E, $\tilde{\eta}_E$ are respectively estimates for the H^1 seminorm and L^2 norm of the error for the edge E, and where $h_{\min,E}$ plays an analogous role to $\lambda_{2,K}$ in the current setting. These observations lead us to propose the following candidate for an L^2 error estimator:

$$\tilde{\eta}_K = \lambda_{2,K}\eta_K, \tag{2.3.9}$$

where η_K is the energy norm estimator (2.2.5). In what follows we present some partial results towards the reliability and efficiency of this estimator.

We make the following strong assumption on the equivalence of the energy norm error: there exist C_1, $C_2 > 0$ such that for every element K,

$$C_1\eta_K \leq |e_h|_{1,K} \leq C_2\eta_K. \tag{2.3.10}$$

Given this assumption, the strategy will be to relate the L^2 error and energy norm locally. Note that in the literature, the upper bound in (2.3.10) only appears globally (for the entire domain Ω), while the lower bound holds on a patch related to a quasi-interpolation operator, see [72, Propositions 16, 21]. Numerical results in Section 2.4.1 suggest that provided the mesh is not too coarse, the inequality holds with $C_1 = 1$ and $C_2 = 10$, see Figure 2.4.

We begin with a technical lemma. In what follows we let W_h denote the space of continuous piecewise quadratic functions on $\mathcal{T}_h$. We note, however, that W_h could be replaced by a higher-order finite element space, with different constants for the inequalities.

Lemma 2.3.1. *Let $v_h \in V_h$.*

1. *There exists $C_{\hat{K}} > 0$ depending only on the reference element $\hat{K}$ such that for all $w_h \in W_h$ and $K \in \mathcal{T}_h$,*

$$\|v_h - w_h\|_{0,K} \geq C_{\hat{K}}\lambda_{2,K}|v_h - w_h|_{1,K}. \tag{2.3.11}$$

2. Suppose that $w_h \in W_h$ and $C_{\hat{K},1} > 0$ such that for all $K \in \mathcal{T}_h$

$$\lambda_{1,K}\|\nabla(v_h - w_h)\cdot r_{1,K}\|_{0,K} \leq C_{\hat{K},1}\lambda_{2,K}\|\nabla(v_h - w_h)\cdot r_{2,K}\|_{0,K}. \tag{2.3.12}$$

Then there exists $C_{\hat{K}} > 0$ depending only on the reference element $\hat{K}$ and $C_{\hat{K},1}$ such that for all $K \in \mathcal{T}_h$,

$$\|v_h - w_h\|_{0,K} \leq C_{\hat{K},2}\lambda_{2,K}|v_h - w_h|_{1,K}. \tag{2.3.13}$$

Proof: By [44], Lemma 2.2 we have

$$|w_h - v_h|_{1,K} \leq \left(\frac{\lambda_{1,K}}{\lambda_{2,K}}\right)^{1/2} |\hat{w}_h - \hat{v}_h|_{1,\hat{K}},$$

where $\hat{w} = w \circ F_K$ for a function w on K. Since $P_2(\hat{K})$ is finite dimensional, there exist positive constants $\tilde{C}_1$, $\tilde{C}_2$ such that

$$\tilde{C}_1\|\hat{w}\|_{0,\hat{K}} \leq |\hat{w}|_{1,\hat{K}} \leq \tilde{C}_2\|\hat{w}\|_{0,\hat{K}}, \qquad \forall \hat{w} \in P_2(\hat{K}).$$

Therefore,

$$\begin{aligned} |w_h - v_h|_{1,K} &\leq \tilde{C}_2 \left(\frac{\lambda_{1,K}}{\lambda_{2,K}}\right)^{1/2} \|\hat{w}_h - \hat{v}_h\|_{0,\hat{K}} \\ &= \tilde{C}_2 \left(\frac{1}{\lambda_{1,K}\lambda_{2,K}}\right)^{1/2} \left(\frac{\lambda_{1,K}}{\lambda_{2,K}}\right)^{1/2} \|w_h - v_h\|_{0,K} \\ &= \frac{\tilde{C}_2}{\lambda_{2,K}}\|w_h - v_h\|_{0,K}, \end{aligned} \tag{2.3.14}$$

and (2.3.11) follows. For (2.3.13), we have

$$\begin{aligned} \|v_h - w_h\|_{0,K} &= (\lambda_{1,K}\lambda_{2,K})^{1/2} \|\hat{u}_h - \hat{u}_{h,2}\|_{0,\hat{K}} \\ &\leq \tilde{C}_1 (\lambda_{1,K}\lambda_{2,K})^{1/2} |\hat{u}_h - \hat{u}_{h,2}|_{1,\hat{K}}. \end{aligned}$$

Applying [44], equation (17) to the right side of the inequality, and applying assumption (2.3.12),

$$\begin{aligned} &\|v_h - w_h\|_{0,K} \\ &\leq \tilde{C}_1 \left(\lambda_{1,K}^2\|\nabla(v_h - w_h)\cdot r_{1,K}\|_{0,K}^2 + \lambda_{2,K}^2\|\nabla(v_h - w_h)\cdot r_{2,K}\|_{0,K}^2\right)^{1/2} \\ &\leq \tilde{C}_1 C_{\hat{K},1}\lambda_{2,K}|v_h - w_h|_{1,K}. \end{aligned}$$

∎

In the present situation we take $v_h = u_h$. To apply Lemma 2.3.1 in a meaningful way, we would like to find functions $\{w_h \in W_h\}_h$ that converge to u faster than $\{u_h \in V_h\}_h$ and that moreover satisfy (2.3.12) uniformly. Let $g_h = \Pi_h(u_h)$ denote the recovered gradient, which is assumed to be superconvergent. In the literature, the adaptive algorithm is designed to achieve the equality

$$\lambda_{1,K}\|(\nabla u_h - g_h)\cdot r_{1,K}\|_{0,K} = \lambda_{2,K}\|(\nabla u_h - g_h)\cdot r_{2,K}\|_{0,K},$$

In the context of [72], this equality means that η_K has been minimized with respect to the choice of $r_{1,K}$, $r_{2,K}$ and aspect ratio s_K. The adaptive algorithm discussed in Section 2.4.1 will ensure that the equality holds by minimizing of η_K with edge swapping and node movement. In general, g_h is not the gradient of a function in W_h. Instead, we take $\tilde{u}_{h,2} \in W_h$ to be the hierarchical reconstruction introduced in [17], which in practice provides a higher-order approximation to u, for instance [17, Figure 17]. Additionally, it will be assumed that $\nabla\tilde{u}_{h,2}$ and g_h are close enough so that (2.3.12) holds with $w_h = \tilde{u}_{h,2}$.

Proposition 2.3.2. *With the notation and assumptions of the preceding paragraph, there exist positive constants $C_{\hat{K},1}$, $C_{\hat{K},2}$ such that for all $K \in \mathcal{T}_h$,*

$$\|e_h\|_{0,K} \geq C_{\hat{K},1}\left(\lambda_{2,K}|e_h|_{1,K} - \|u - \tilde{u}_{h,2}\|_{0,K} - \lambda_{2,K}|u - \tilde{u}_{h,2}|_{1,K}\right), \tag{2.3.15}$$

and

$$\|e_h\|_{0,K} \leq C_{\hat{K},2}\left(\lambda_{2,K}|e_h|_{1,K} + \|u - \tilde{u}_{h,2}\|_{0,K} + \lambda_{2,K}|u - \tilde{u}_{h,2}|_{1,K}\right) \tag{2.3.16}$$

Proof: This follows directly from Lemma 2.3.1 and straightforward triangle inequality arguments. ∎

Finally, if we apply the superconvergence assumptions on $\tilde{u}_{h,2}$ and $\nabla\tilde{u}_{h,2}$, and the strong energy norm error assumption (2.3.10), we conjecture that there exists $C_{\hat{K},1}$, $C_{\hat{K},2} > 0$ such that for every element K, up to the addition of higher-order terms, the L^2 norm error satisfies

$$C_{\hat{K},1}\tilde{\eta}_K \leq \|e_h\|_{0,K} \leq C_{\hat{K},2}\tilde{\eta}_K. \tag{2.3.17}$$

This local estimate will be verified numerically in the following section.

2.4 Numerical results

In this section we provide numerical validation for the new, element-based adaptation method for the residual estimator. The full adaptation loop is given in Algorithm 1.

The first test case will begin with an illustration of the convergence of the loop for the element-based residual method introduced in this paper, the notion of convergence to be made more precise in the following section. Next, we will assess how well the method performs in achieving the goal of equidistributing the error over the elements of the mesh. Since what we are really interested in is controlling the actual error, the analysis will include an element-level comparison of the estimated versus exact error. Following will be a numerical validation of the L^2 error control results from Section 2.3.6. Finally, the remainder of the section will be devoted to the comparison of the adaptation methods outlined in Section 2.3.

Algorithm 1 Solution-adaptation loop

1 Compute the solution and error estimator on the current mesh.

2 Adapt the current mesh by performing the following loop one or more times:

(a) Refine edges where the error is too large.

(b) Minimize the error by swapping edges until the algorithm terminates, then equidistribute the error by applying node displacement. Repeat the procedure one or more times.

(c) Remove nodes where the error is too small, or when the impact on the error is minimal.

(d) Apply 2(b).

2.4.1 First test case

We consider the problem (2.2.1) using $A = I$, with domain $\Omega = (0,1) \times (0,1)$, and f, g chosen so that the exact solution is

$$u_1(x,y) = 4(1 - e^{-100x} - x(1 - e^{-100}))y(1-y).$$

Due to the boundary layer near $x = 0$, this function can be used to check the anisotropy of an error estimator and adaptation method, and appears for instance in [44] and [80].

Assessment of the residual element-based method

Convergence of the adaptation loop. The convergence of the algorithm for the element-based residual method is assessed. Starting from a relatively coarse initial mesh, the tolerance TOL is set to 0.125 and the loop is run for 40 iterations (which

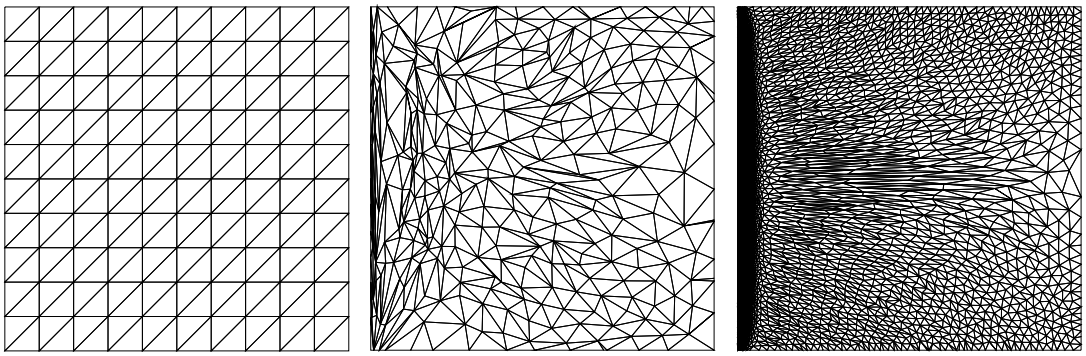


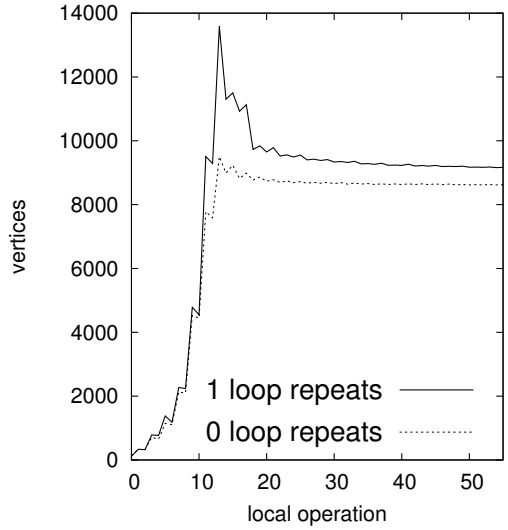
Figure 2.2: Left: initial uniform mesh with 121 vertices. Middle: first adapted mesh with 324 vertices. Right: final mesh with 8559 vertices.

for our purposes will be more than sufficient). See Figure 2.2 for initial and adapted meshes. In the context of Algorithm 1, the edge swapping/node movement step is always run 3 times. Additionally, the adaptation step 2 is only run once before the solution and the estimator are recomputed. The point of view taken here is that the adaptation algorithm should not be run too long before recomputing the solution and error estimator. For comparison, we computed an example where step 2 from Algorithm 1 is run twice, as opposed to just once. From Figure 2.3a, repeating the loop initially calls for too much refinement. This is likely due to a loss of accuracy of the estimator on coarse meshes (see Figure 2.4 and the related discussion).

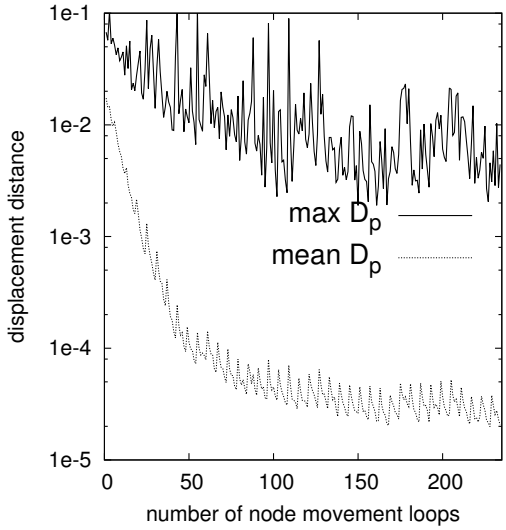
Table 2.1 records the number of refinements, derefinements and edge swappings performed at each iteration. The columns for edge swapping include the sum of three separate edge swapping loops. In all cases, note that the number of operations performed becomes small by about 10 iterations.

Convergence of node movement is measured by norm of the displacement of individual nodes. For a node p denote by D_p the norm of its displacement. Figure 2.3b plots the max and mean displacement for each iteration of node movement during the adaptation loop. While the maximum displacement remains of the order 10^{-2}, this value represents only a few outlier cases, with the average displacement occuring between 10^{-5} and 10^{-4}.

Control of the energy norm of the error. Next we assess the performance of the algorithm towards equidistributing the error. The distribution of the estimated error for different iterations is plotted in Figure 2.4. The error is normalized by taking $e_K = \log_{10}\left(\frac{\eta_K}{TOL/\sqrt{N_T}}\right)$. From the figure, we see that after successive iterations the error increasingly tends to cluster towards the target error and the distribution tends to be more normal. Furthermore, the standard deviation of the error decreases from 0.58 on the intial mesh to 0.041 on the final mesh.



(a) Solid line: applying the adaptation step from Algorithm 1 twice before recalculating the solution. Dotted line: applying the adaptation step once.



(b) Solid line: maximum displacement. Dotted line: average displacement.

Figure 2.3: Left: Number of vertices after refinement/derefinement. Right: Maximum and average displacement of nodes for cumulative node movement loops.

Next we establish numerically the equivalence between the exact and estimated error. The motivation is to assess to what degree we can expect to control the exact error, both locally and globally, by equidistributing the estimated error as in Figure 2.4. Define the global effectivity index with respct to the energy norm by $ei = \frac{\eta}{|||e_h|||}$, where $\eta = \sqrt{\sum_K \eta_K^2}$. Theorem 2.2.1 says that globally the energy norm of the error is equivalent to the estimated error, so that the effectivity index assesses this equivalence on a given mesh. Ideally, the effectivity index satisfies $ei \to 1$ as the mesh element size goes to 0, in which case we say that the estimator is asymptotically exact. This effectivity index is studied for instance in [80] and [72], where it was observed to remain reasonably low for adapted meshes (between 2 to 5). Furthermore, for meshes adapted with target error TOL, the index remains bounded as $TOL \to 0$, see for instance [80, Table 3.7]. Thus, while the estimator is not asymptotically exact, it is clearly equivalent to the exact error.

We define the local effectivity index for a triangle K by

$$ei_K = \frac{\eta_K}{|e_h|_{1,K}}. \tag{2.4.1}$$

(Recall that since $A = I$ in (2.2.1) the energy norm is just the H^1 seminorm.) The

it.	refinement		derefinement		swapping			
					after refinement		after derefinement	
	edges	%	nodes	%	edges	%	edges	%
1	212	175.21	9	2.70	500	53.48	251	27.61
2	415	128.09	47	6.36	1049	49.23	418	20.93
3	487	70.38	55	4.66	1179	34.40	569	17.41
4	996	88.61	20	0.94	1616	25.96	768	12.44
5	2383	113.48	78	1.74	3275	24.66	1235	9.45
6	3295	74.80	256	3.32	4556	19.91	1675	7.57
7	2031	27.28	507	5.35	3926	13.95	1380	5.18
8	265	2.95	497	5.38	1902	6.94	990	3.82
9	189	2.16	223	2.50	1173	4.43	677	2.62
10	87	1.00	132	1.50	777	2.98	535	2.08
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
40	5	0.06	4	0.05	153	0.60	110	0.43

Table 2.1: Number of local operations for complete adaptation loop.

quantity (2.4.1) measures the equivalence of the exact and estimated error at the level of the element. In Figure 2.4 we plotted the distribution of (2.4.1) for a few meshes. For the coarse uniform mesh with 200 elements, note that while the global effectivity index is quite low ($ei = 1.08$), the distribution of the local effectivity index is spread out, with a large upper tail. On the other hand, the finer uniform mesh with 20000 elements has a higher global effectivity index ($ei = 1.70$) with a smaller tail, suggesting that the accuracy of the estimator improves with refinement. We also show the distribution for a relatively coarse adapted mesh with 4500 elements. While the global effectivity index is higher ($ei = 2.42$), the local effectivity index is more closely distributed about the global effectivity index. What appears to happen is that refinement exaggerates the overestimation of the error that already occurs in uniform meshes.

Control of the L^2 norm of the error. In the remainder of the subsection, we will assess numerically the lower and upper bounds for the L^2 error given in (2.3.17), and briefly present some results using the estimator (2.3.9) for mesh adaptation.

Setting $TOL = 0.125$, the final adapted mesh using η_K mesh has about 18000 elements. Figure 2.5a records for each element the estimated error η_K (in blue) and $\tilde{\eta}_K$ (in black) vs. the exact L^2 error $\|e_h\|_{0,K}$. While η_K remains within less than 1 order of magnitude, the exact L^2 error is spread by about 3 orders. Therefore, equidistributing the estimator η_K does not lead to equidistribution of the L^2 error. In verifying the lower and upper from (2.3.17), we see that the local effectivity index $\tilde{ei}_K = \frac{\tilde{\eta}_K}{\|e_h\|_{0,K}}$ remains between about 0.1 and 10, with the lower bound appearing sharp.

Next we adapt the mesh using the scaled error $\tilde{\eta}_K$. The mesh is adapted using

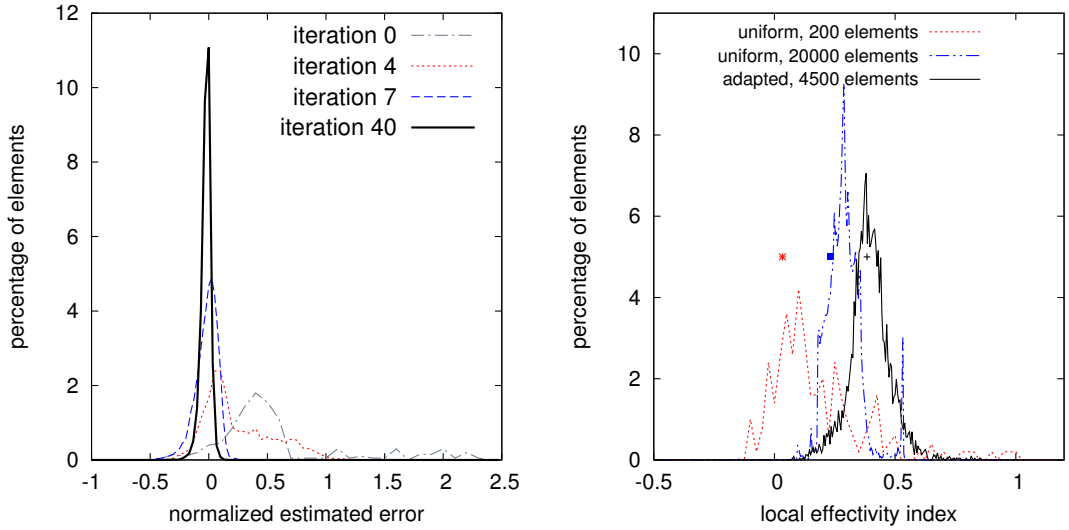
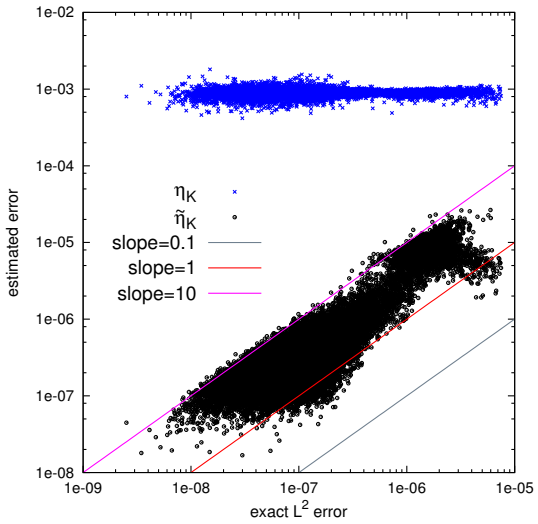


Figure 2.4: Left: distribution of the error over elements. Middle: standard deviation of error distribution. Right: Distribution of the local effectivity. The global effectivity index is indicated by a red star for the coarse uniform mesh, by a blue square for the fine uniform mesh, and a black plus for the adapted mesh.

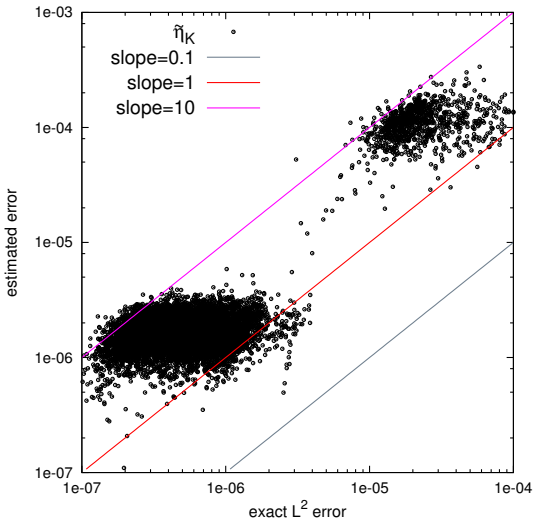
the hybrid error approach from [17] for the hierarchical estimator. That is, edge refinement and node removal are used to control the global L^2 error level (here using (2.3.9)), while edge swapping and node movement are used to equidistribute the error by minimizing the energy norm (here using (2.2.5)). Results of two meshes adapted using different target error levels are presented in Figure 2.5b: a mesh with about 1000 elements (top right) and one with 14000 (bottom left). The spread of the L^2 error is significantly lower, going from 3 orders of magnitude to about 1.5.

We compare global error calculations using the scaled and non-scaled estimator in Figure 2.6. Clearly, the non-scaled estimator results in lower energy norm error for the same degrees of freedom. Moreover, as predicted, the scaled error significantly improves the results for the L^2 error.

Lastly, we compare meshes adapted with the scaled and unscaled estimators in Figure 2.7. Both meshes have roughly the same number of vertices/elements, but the distribution of the elements for the mesh adapted using the scaled estimator is much more spread throughout the domain, while that in the original estimator tends to concentrate near the boundary $x = 0$. This oberservatioin is really not surprising, since scaling the estimator by the smallest eigenvalue will permit elements to be larger in areas where the H^1 error is largest, such as the boundary layer. Generally, we expect that the error converges at a higher order in the L^2 norm than for the H^1



(a) Estimated error (2.2.5) and (2.3.9) vs. the exact L^2 error for a mesh adapted using the original estimator (2.2.5).



(b) Estimated error (2.3.9) vs. the exact L^2 error for two meshes adapted using the scaled estimator (2.3.9).

Figure 2.5: Estimated error vs. exact L^2 error over elements.

seminorm error. But as seen in Figure 2.5b, this higher order is not just global (at the level of the domain), but local (element level). This observation about mesh quality being related to the norm will be further confirmed in what follows when we consider the hierarchical estimator, which natively controls the L^2 error.

Comparison of the adaptation methods

Qualitative comparison. Figure 2.8 presents examples of adapted meshes with about 2500 vertices produced by each method. In all cases, we see that the meshes contain elements that are very stretched near the boundary layer. Note that in general the meshes obtained from the residual estimators tend to have more elements near the boundary layer, while the meshes from Hessian and hierarchical methods tends to be more spread out. The difference in mesh density is likely due to the target norm used by each method. As discussed in the previous section, the target norm is related to the local order of convergence, which affects local element size.

Another note is that the mesh for the Hessian is quite regular in the top and bottom right corners. The initial mesh is regular, consisting of right triangles as in Figure 2.2a, so what seems to be happening is that in these regions the main operation performed is edge refinement. In particular, node displacement appears to be less smooth for the Hessian. Repeating the adaptation loop starting from a

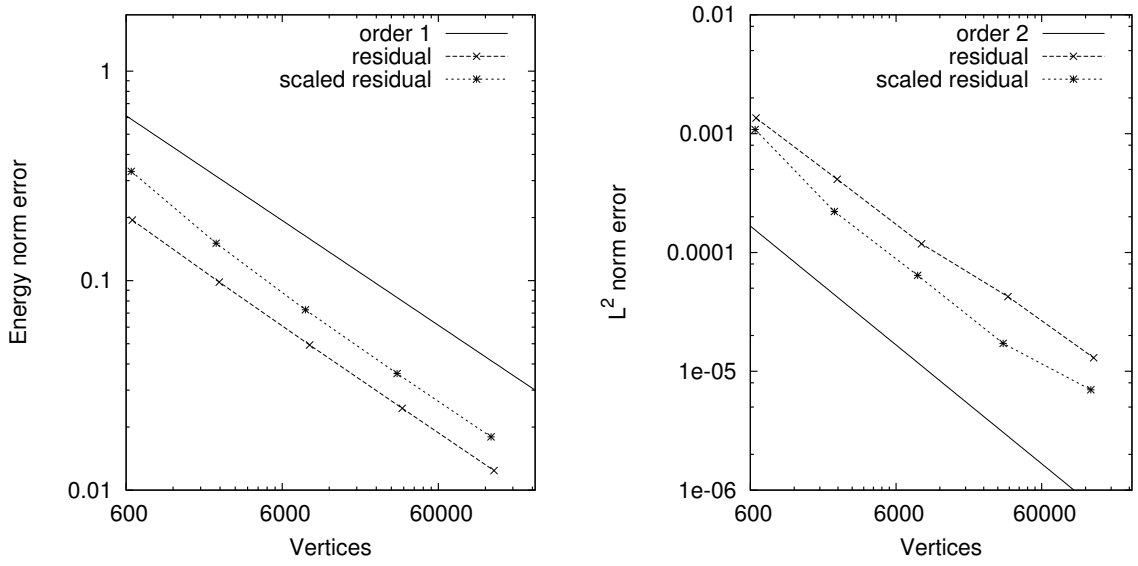


Figure 2.6: Energy norm (left) and L^2 norm (right) error calculations for u_1 for residual estimators.

non-uniform mesh does in fact result in a final mesh which is not regular.

Analytical comparison. The error is reported in Figure 2.9 as a function of the number of vertices. Recall that for a regular mesh in $2D$, the number of vertices is roughly proportional to $(\frac{1}{h})^2$, so that the theoretically optimal (logarithmic) slope corresponding to (2.3.6) is $-1/2$, while for (2.3.8) it is -1.

Figure 2.9 reports the error in the energy and L^2 norm. We see that all methods approach the theoretical rate of convergence for the energy norm. Moreover the hierarchical method, which reports the largest error, remains about 1.3 times higher than the residual element-based method, which reports the smallest error. The convergence for the L^2 norm, on the other hand, appears to be more erratic, with none of the methods achieving the optimal rate of convergence. Here the hierarchical method reports the lowest error, the residual methods report an error 2 to 3 times as large, while the Hessian method reports an error about 4 to 5 times as large. Note that for both, the energy and L^2 norms, the results for both residual methods are close.

In Table 2.2 we record the mean and variance of the distribution of the error over the elements. While Figure 2.9 shows that the global energy norm is lowest for the residual methods and highest for the hierarchical, the situation is reversed here, with the hierarchical reporting the lowest mean error. This can be partially accounted for by the fact that the residual methods result in the lowest standard deviation for the energy norm of the error, which is likely the result of the equidistribution of the

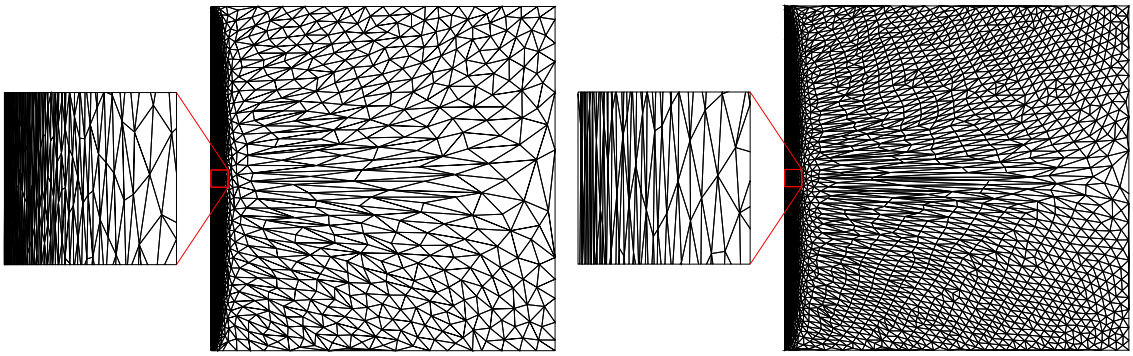


Figure 2.7: Adapted meshes with 2500 vertices using the H^1 estimator (left) and scaled estimator (right).

estimated error achieved by the adaptive method. For the L^2 error, as expected the hierarchical method reports the lowest mean and standard deviation.

We remark that the target norm for the Hessian adaptation is L^∞, so it is possible that the Hessian does the best job equidistributing the error in the L^∞ norm. We did not calcluate the L^∞ norm. Furthermore, it should be noted that in [69], a Hessian-based error estimator was developed to control the the L^p error for $1 \leq p < \infty$. Adaptation is, as before, done by constructing a metric, which turns out to be the the same as that discussed in Section 2.3.5 with the eigenvalues appropriately scaled for the choice of p, see [69, Section 2]. It is reasonable to expect that the results for the L^2 error could be improved using their estimates.

Computational performance. Figure 2.10a records the CPU time for the adaptation part of each iteration of the loop. For each method, we chose the global error level so that the final mesh has about 9000 vertices. The number of vertices at each iteration is recorded in Figure 2.10b, and the gap between lowest and highest at the last step is about 6%. We find that metric adaptation requires much less time than the other methods, and in the plot both metric based methods appear superimposed. This result is not surprising. The only that value we really need to keep track of is the metric tensor at each vertex. The local error calculations are relatively insubstantial compared to those required for element-based adaptation.

In addition, note that the element-based residual method takes roughly 5 to 6 times that of the hierarchical estimator. For one thing, the residual estimator requires the contribution from discontinuous functions. These functions cannot be directly interpolated after performing local modifications, and must be recomputed on each element/edge. Especially problematic is the calculation of the singular value decomposition. Even for a 2×2 matrix A, it can be numerically disastrous to calculate the singular value decomposition of A directly by first computing AA^T [51], and instead

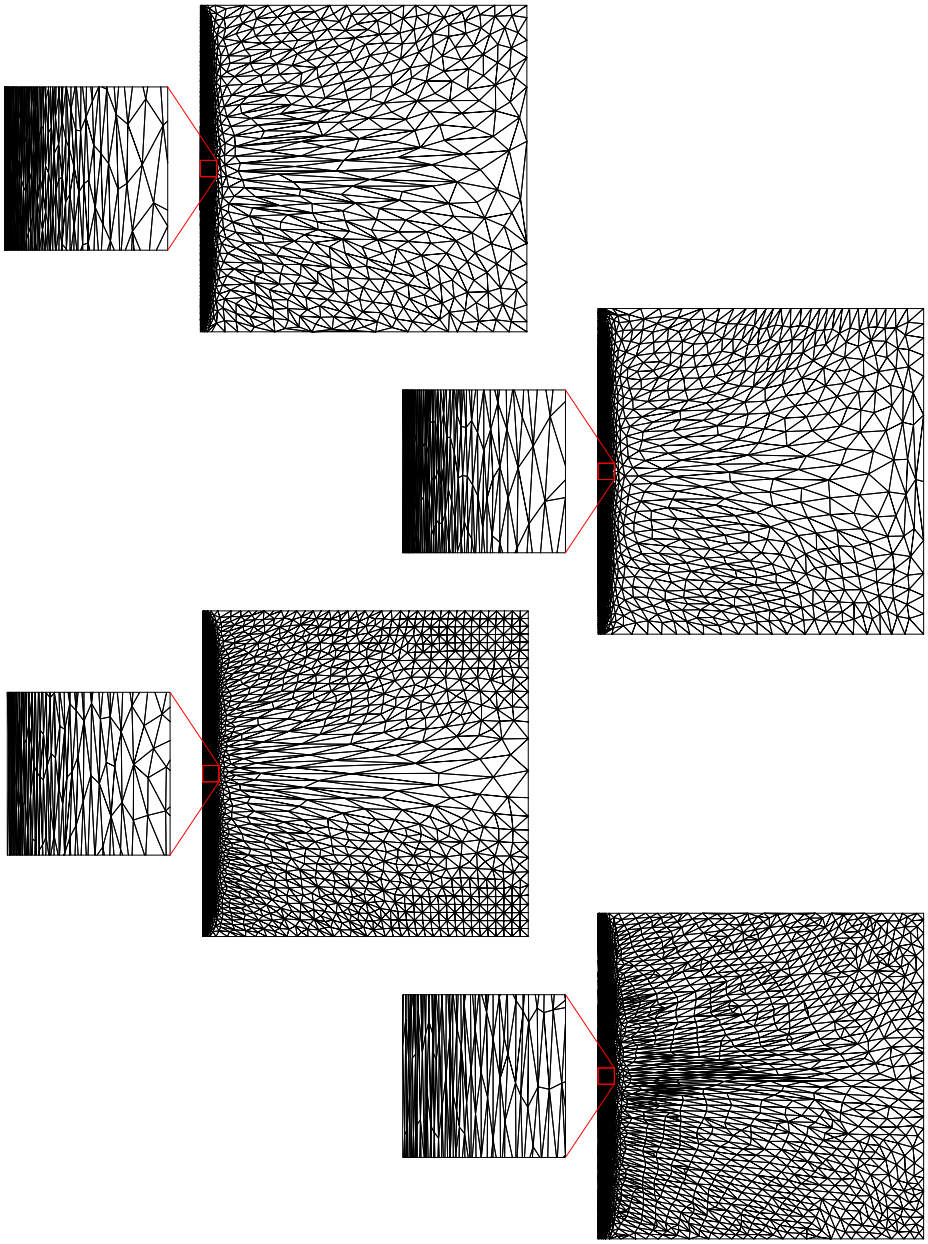


Figure 2.8: Adapted meshes for u_1 with approximately 2500 vertices, with zoom near the boundary at $x = 0$. From top to bottom: residual (element), residual (metric), Hessian, hierarchical.

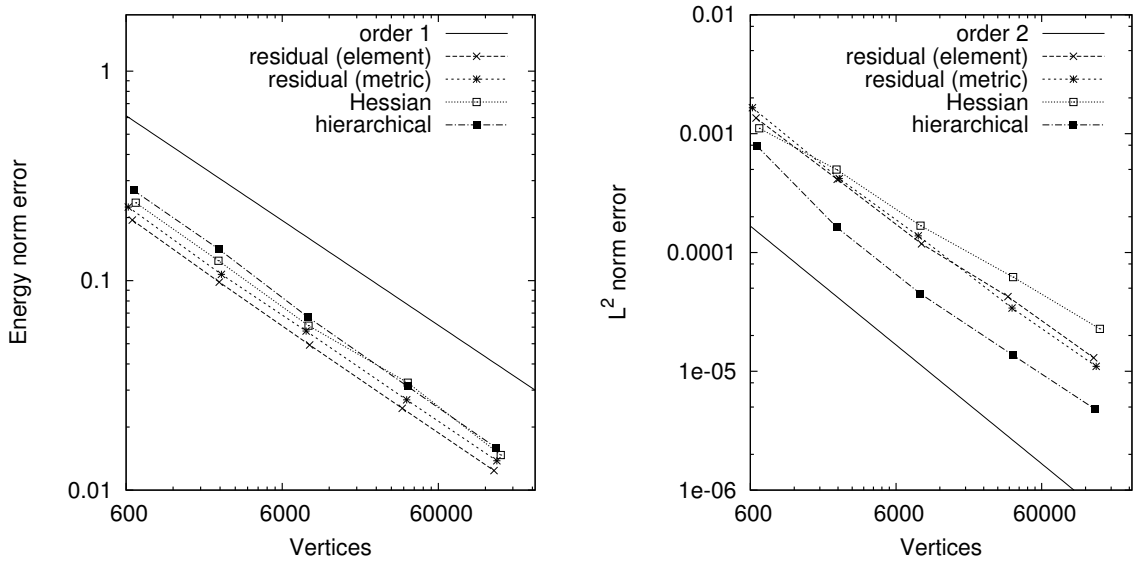


Figure 2.9: Energy norm (left) and L^2 norm (right) error calculations for u_1.

it is recommended to use an iterative method. We have used the implementation provided by DGESVD from LAPACK. Overall, it was found that this computation takes between $13-18\%$ of the total adaptation process. Another contribution towards increased CPU time is due to the fact that the jump term depends on more than one element. As mentioned in Section 2.3.1, when performing edge swapping, the error needs to be calculated on an enlarged patch as in Figure 2.1 in order to accurately compute the jump term. The construction and handling of this patch introduces significant computational overhead.

2.4.2 Second test case

With the same parameters for problem (2.2.1) as test case 1, we consider the function taken from [76]

$$u_2 = \tan^{-1}(\alpha(r - r_0)),$$

where $r = \sqrt{(x+0.05)^2 + (y+0.05)^2}$, and $r_0 = 0.7$. Thus, we have a circular a wave-front type solution, centered at $(-0.05, -0.05)$ with a transition region with thickness of order α^{-1}. We will run simulations with both $\alpha = 100$ and $\alpha = 1000$.

Qualitative comparison

In Figures 2.11 we show some examples of adapted meshes. In each case, the mesh follows what we would expect from the solution. The elements are mainly concentrated near the wavefront where the gradient is steep in the direction orthogonal to

method	vertices	mean $\|\nabla(e_h)\|_K$	st. dev. $\|\nabla(e_h)\|_K$	mean $\|e_h\|_K$	st. dev. $\|e_h\|_K$
residual (element)	657	5.37e-03	1.20e-03	2.16e-05	3.16e-05
residual (metric)	639	6.04e-03	1.74e-03	2.63e-05	3.93e-05
Hessian	691	5.27e-03	3.84e-03	2.27e-05	2.07e-05
hierarchical	670	5.42e-03	5.30e-03	1.84e-05	1.21e-05
residual (element)	2369	1.42e-03	2.82e-04	3.25e-06	5.13e-06
residual (metric)	2251	1.62e-03	4.09e-04	3.55e-06	4.99e-06
Hessian	2342	1.51e-03	1.08e-03	4.40e-06	5.98e-06
hierarchical	2356	1.41e-03	1.54e-03	2.14e-06	1.08e-06
residual (element)	8992	3.63e-04	7.24e-05	4.51e-07	7.65e-07
residual (metric)	8701	4.01e-04	8.88e-05	4.99e-07	8.20e-07
Hessian	8842	3.85e-04	2.56e-04	6.63e-07	1.09e-06
hierarchical	8786	3.55e-04	3.62e-04	2.96e-07	1.72e-07
residual (element)	35218	9.12e-05	1.76e-05	7.13e-08	1.44e-07
residual (metric)	33448	1.02e-04	2.14e-05	8.16e-08	1.58e-07
Hessian	38290	8.78e-05	7.92e-05	1.06e-07	1.85e-07
hierarchical	38205	7.90e-05	8.10e-05	3.95e-08	2.97e-08

Table 2.2: Distribution of error.

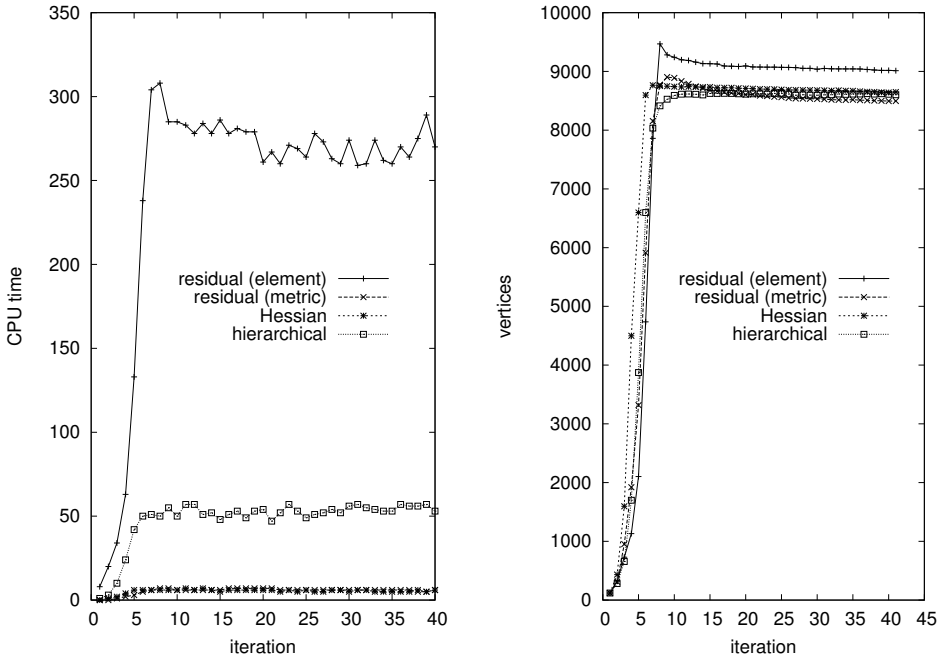
the wave, and with the alignment of the elements in this region reflecting the curvature. Outside this region, variation in the solution is reduced significantly, so that the elements can be much larger. What is striking, however, is the difference between the mesh produced by the hierarchical method compared to the others. For the hierarchical method, the mesh is more spread out and less concentrated near the wave-front. As discussed in Section 2.4.1, we attribute this difference to the target norm used. Another feature of interest, seen in the zoom to the wavefront, is a sub-layer of elements where the mesh is coarser. In this region, the function is almost linear in the direction orthogonal to the wave-front, so that the error is somewhat smaller than in the immediate surroundings.

Analytical comparison

Calculation of the residual term. This test case highlights one of the drawbacks of residual estimators. To calculate the error η_K, we need to evaluate the integrals $\int_K f^2 \, \mathrm{d}x$ of the source term f. (Since $A = I$ and u_h is piecewise linear, we get $R_K(u_h) = f$.) At the wave front with $\alpha = 1000$ this value is very difficult to compute accurately. The immediate effect on adaptation was that some elements were not being refined despite being flagged as having large error. In particular, the algorithm reported

$$\int_{K_1} f^2 \, \mathrm{d}x + \int_{K_2} f^2 \, \mathrm{d}x \gg \int_K f^2 \, \mathrm{d}x, \tag{2.4.2}$$

where K_1, K_2 are obtained by refining an edge of K.



(a) CPU-time in seconds (b) Number of nodes

Figure 2.10: Left: CPU time for each iteration. Right: number of nodes at each iteration.

We improve the accuracy of the integral by subdivision. For a given quadrature rule $\mathcal{Q}_K$ on an element, we divide the triangle into 4 by splitting the edges in half, then define the subdivided quadrature rule $\mathcal{Q}_{K,1}$ to be that composed of four copies of the original, each weighted $\frac{1}{4}|K|$. The effect is that if the rule we had before was Ch^k accurate, the subdivided scheme is $\frac{C}{2^k}h^k$ accurate. We therefore increase the accuracy without introducing a large constant from a higher-order method. See Algorithm 2 for implementation details.

Since each time we subdivide, we multiply the number of Gauss points by 4, subdivision can quickly become expensive. We always subdivide at least once, so that at the very least we need to compute values at $(1+4)B_G$ points, where B_G is the base number of Gauss points. Therefore, higher-order quadrature rules are virtually unusable for subdivision, and the total number of subdivisions never exceeds 3. Fortunately, in our case it was sufficient to use the single point (barycenter) integration scheme. Even still, this comes at the high cost of 64 Gauss points for the third subdivision. The percentage of subdivisions that occur for an adaptation loop with ϵ from Algorithm 2 set to 0.05 are reported Figure 2.3. By the tenth iteration, additional subdivision is not significant.

We remark that subdivision integration is not necessary if adapting using a met-

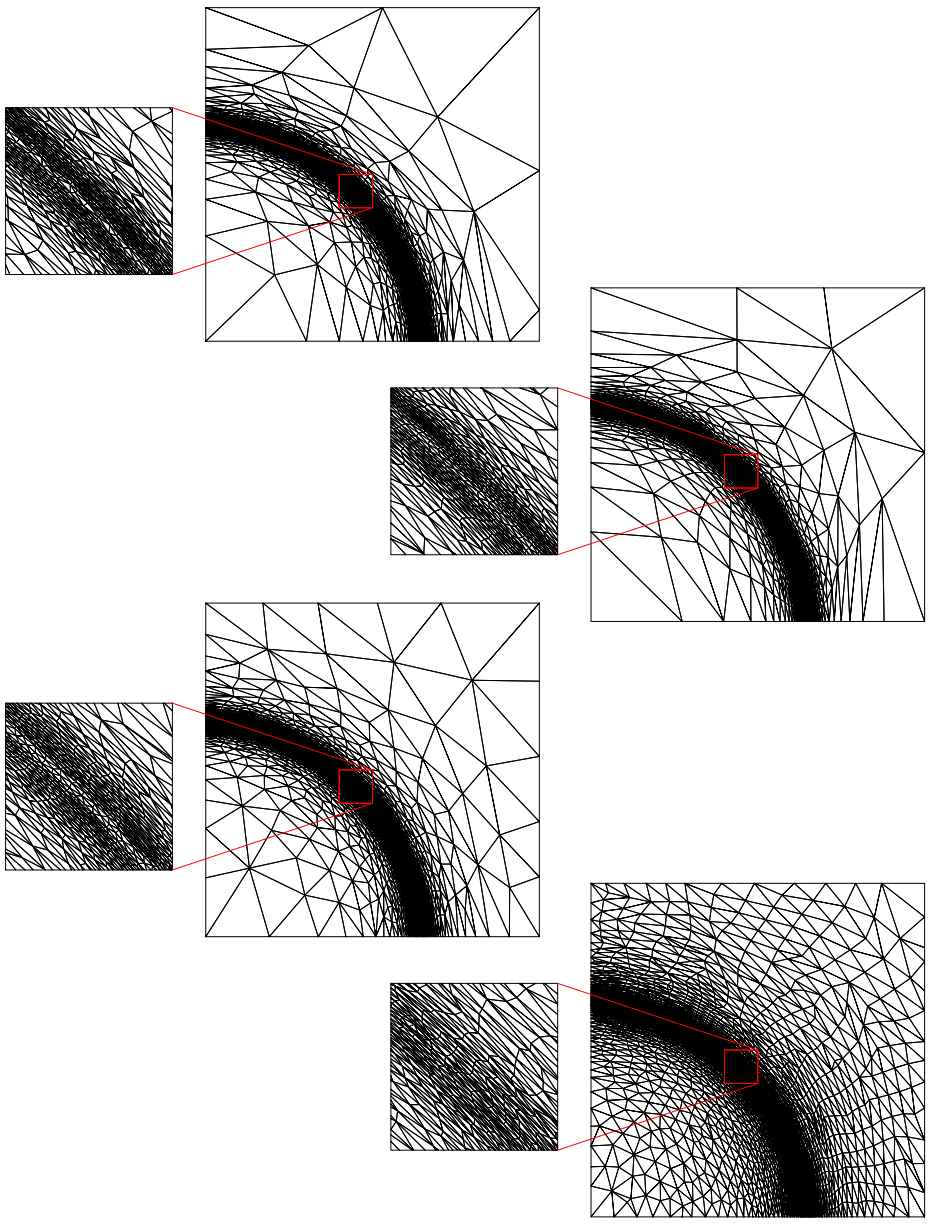


Figure 2.11: Adapted meshes for u_2, $\alpha = 100$ with approximately 3000 vertices, with zoom to wave front. From top to bottom: residual (element), residual (metric), Hessian, hierarchical.

Algorithm 2 Calculation of the residual on element K.

1. Calculate the residual $\tilde{R}_0$ with the original quadrature $\mathcal{Q}_{K,0}$.
2. Choose $\epsilon > 0$ to be small. For $i = 0, 1, 2$ do the following:
   (a) Subdivide the current quadrature $\mathcal{Q}_{K,i}$ into $\mathcal{Q}_{K,i+1}$ and compute the residual $\tilde{R}_{i+1}$.
   (b) If $i = 2$ or if $\frac{|\tilde{R}_i - \tilde{R}_{i+1}|}{\tilde{R}_{i+1}} \leq \epsilon$ then accept $\tilde{R}_{i+1}$ as the residual and exit.

it.	2 sub. %	3 sub. %
1	7.55	4.02
5	10.69	2.69
10	1.79	0.88
20	1.49	0.77

Table 2.3: Percentage of elements where additional subdivision occurs at each iteration of the global adaptation step.

ric. There, the residual is calculated only once to compute the metric so that issues such as (2.4.2) will not be seen during adaptation. Furthermore, when computing the metric, the residual term is often left out altogether to save computational time, as is done in [24]. Theoretically, this simplification can be justified for the Laplace equation as proven in [63]. In the case of element-based adaptation, we found that including the residual term was necessary, since experiments with removing the residual term generally resulted in meshes of poor quality.

Global error comparison. In Figures 2.12 and 2.13, we record the error convergence for u_2 for $\alpha = 100$ and $\alpha = 1000$. The results are very similar to that for u_1: for the energy norm, the results are close, with the element-based residual method reporting the lowest error, while for the L^2 error, as in Section 2.4.1, the hierarchical method reports the lowest. The results for the L^2 error for $\alpha = 1000$ will be discussed in some detail here, for they clearly highlight the issue of controling the L^2 norm with an estimator for the H^1 seminorm. We found that the L^2 error oscillates over consecutive iterations when adapting with the residual in some situations. To illustrate this issue, we reported the results in a different way in Figure 2.13. For each target error, after the number of local modifications and vertices has stabilized, we take the smallest and largest error after 10 further adaptation iterations, giving an upper and lower envelope. With the exception of the residual element-based, where we see a persistent spread of about 5 to 10%, the envelope becomes narrow as the number of nodes increases.

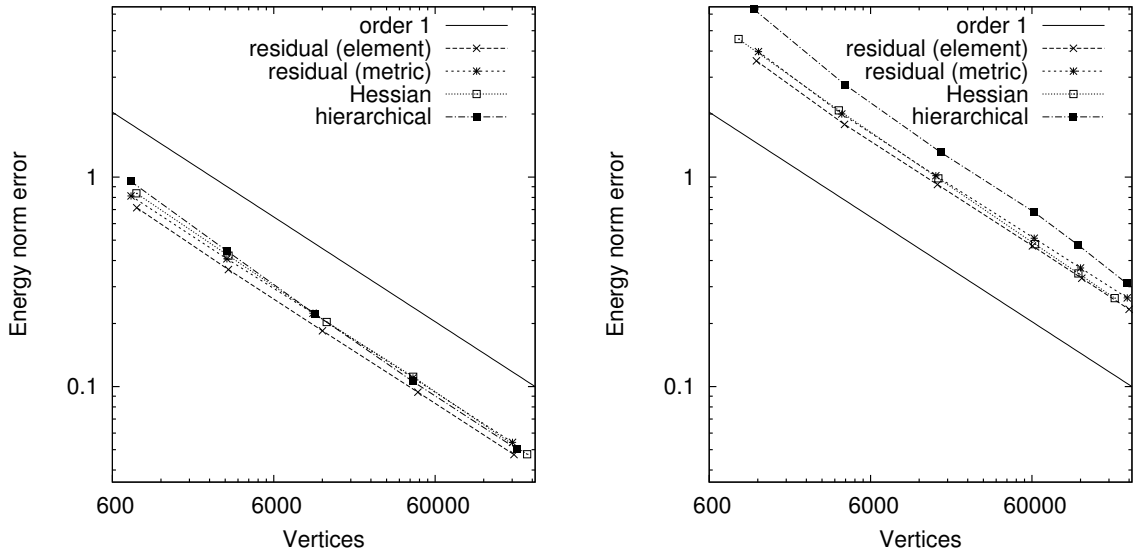


Figure 2.12: Energy norm error calcultations for u_2 with $\alpha = 100$ (left) and $\alpha = 1000$ (right).

In Figure 2.14, we illustrate that the oscillation is due to a few outlier elements, appearing just before and after the wave-front. These elements account for a significant percentage of the overall error, and at each iteration, slight variations in this region cause significant fluctuation in the error. From the Figure 2.13b we see that for coarse meshes, this instability arises for all methods. For the hierarchical method, as we decrease the target error, the region is refined, and the L^2 error stabilizes. The lack of stability for the L^2 error in the case of the residual estimator is the result of two combined factors. First, the estimator does not detect the fact that the L^2 error is still quite large outside the wave-front, and therefore, even at very fine meshes of over 250000 vertices, the mesh is not refined in those regions. This observation fits within the context of Proposition 2.3.2 very well, because while we have equidistributed the H^1 seminorm error over the elements, the value of $\lambda_{2,K}$ is much smaller for elements at the wave-front, which predicts that the L^2 error should also be much lower. The other contributing factor is that the mesh is not completely stationary in this region, so that slight variations in the mesh, which barely registered as far as the H^1 seminorm is concerned, cause large variations in the L^2 error.

2.5 Conclusion

We introduced an element-based mesh adaptation method for the anisotropic a posteriori error estimator appearing in [80]. The method is done by interfacing with

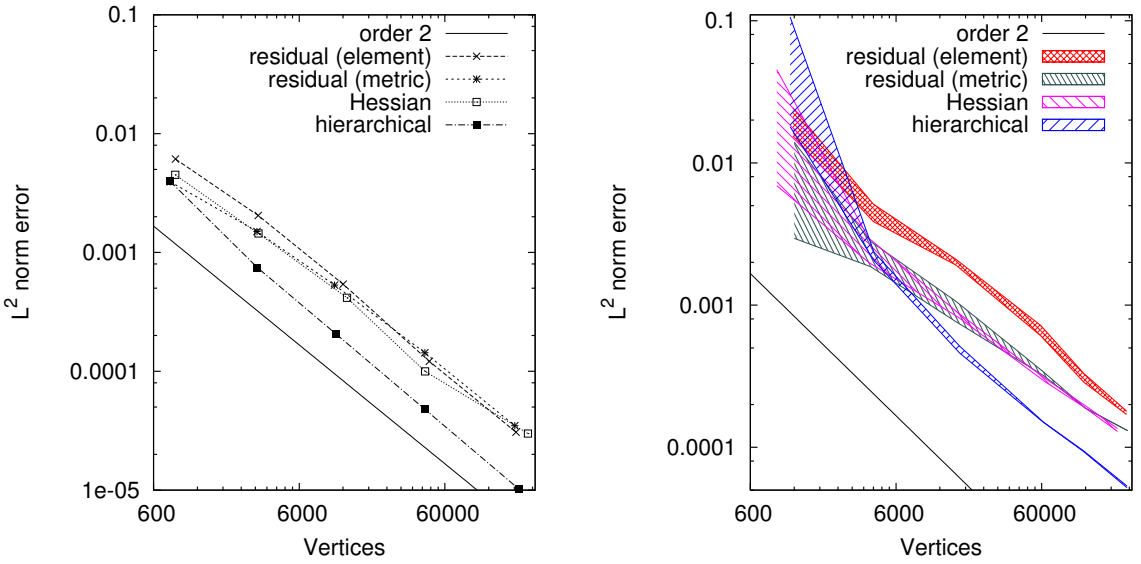


Figure 2.13: L^2 norm error calcultations for u_2 with $\alpha = 100$ (left) and $\alpha = 1000$ (right). The plot on the right depicts the envelope of the oscillating error.

the hierarchical estimator driver from MEF++, as introduced in [17]. We tested the method in numerical test cases that feature significant anisotropic behaviour and verified that the adaptation algorithm produces anisotropic meshes and converges. Additionally, we considered an L^2 norm error variant of the estimator, which, under some hypotheses on the mesh, is equivalent to the exact L^2 error. Numerical examples were provided to confirm the equivalence with the exact error. Examples of adapted meshes using the modified estimator were provided, and it was found to give improved performance for control of the L^2 error over the original estimator.

The new element-based method was compared with three existing anisotropic mesh adaptation methods for P_1 finite elements: residual metric based, Hessian metric, and hierarchical. In terms of controlling the level of error with respect to degree of freedom, the new method generally performed slightly better for the energy norm, while the hierarchical method performed significantly better than the other methods for the L^2 norm. However, the new method is significantly more expensive from a computational standpoint. We note that the results for both element and metric based methods for the residual estimator were generally very close for both norms. Given the results presented in Section 2.4, it seems likely that the method that obtains a given level of error in the energy norm in the shortest time would be the residual metric method, while for L^2 error it would be one of the residual metric or Hessian methods.

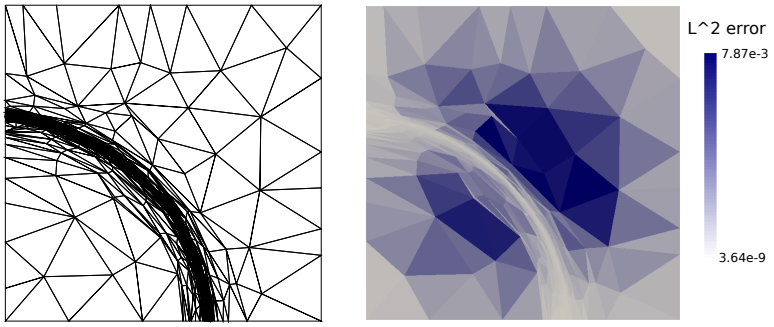


Figure 2.14: Mesh with about 1100 vertices adapted with the hierarchical method (left) and the distribution of the exact L^2 error (right).

Currently the authors are working on optimizing the computational aspects of the method to make it more competitive with the other methods in terms of CPU efficiency. Additionally, an investigation is being made to determine why the element residual cannot be dropped from the computation, as for instance in [24].

Chapter 3

Space-time adaptation for nonlinear reaction-diffusion equations

Abstract. A residual error estimator is proposed for the energy norm of the error for a scalar reaction-diffusion problem. The problem is discretized using P_1 finite elements in space, and the backward difference formula of second order (BDF2) in time. The estimator for space makes use of anisotropic interpolation estimates, assuming only minimal regularity. Reliability of the estimator is proven under certain mild assumptions on the convergence of the approximate solution. A space-time adaptation algorithm is proposed to control the global error, using a Riemannian metric for mesh adaptation and a simple method to adjust the time step. Numerical examples are used to verify the reliability and efficiency of the estimator, and to test the adaptive algorithm. The potential gains in efficiency of the proposed algorithm compared to uniform methods is discussed.

3.1 Introduction

Reaction-diffusion equations model physical problems in a large variety of situations. Numerically, these problems tend to be stiff, and can be demanding in terms of computational resources, particularly in regions that develop sharp wave fronts, solitons, etc., and for solutions that exhibit multiscale behaviour. Achieving an accurate solution with uniform spatial and temporal resolution can be impractical or even impossible. This chapter addresses the problem of improving efficiency and accuracy by the use of adaptive techniques. The adaptation should be based, when possible, on theoretically justified *a posteriori* error estimates that are computable from the approximate solution.

A variety of approaches have been considered to estimate the error for reaction-diffusion equations. Energy techniques are applied in [24], [27] to derive explicit error estimates, the former for the energy norm and the latter in the $L^\infty(0,T;L^1(\Omega))$ norm. The error is bounded by a sum of residual terms, together with interpolation estimates. A similar class of estimators in various norms is found in [39], [41], where the error is estimated using duality techniques. The residual terms are weighted by stability terms, obtained by approximating a dual problem, which indicates the rate of accumulation of error. While such techniques are generally more expensive, owing to the requirement of solving a dual problem, they are useful if one is interested in controlling the value of an arbitrary functional of the error. Another approach can be found in [46], [57], [64], where the error is approximated by solving an auxiliary problem with a hierarchical basis.

Equally important to error estimation is the application of adaptation techniques. Mesh adaptation based on a posteriori error estimates for reaction-diffusion problems can be found in [14], [39], [41], [46] in an isotropic context. Here the primary mesh operations performed are refinement in regions where the estimator is large, and coarsening where the estimator is small. These estimators generally employ finite element interpolation or projection estimates, which rely on a mesh regularity assumption such as the minimum (or maximum) angle condition. With the aid of anisotropic interpolation estimates, for instance from [25], [44], [62], the classical a posteriori estimates have been applied to mesh adaptation in anisotropic framework in [57], [62], [73], [80] for linear problems, and in [22], [24] for nonlinear problems. The adaptation is driven by constructing a non-Euclidean metric from the estimator, often employing a gradient and Hessian recovery technique. See for instance [48], [53], [68] for details on metric mesh adaptation in a general context. The use of anisotropic methods is generally found to result in significantly lower error for a given number of elements compared to isotropic methods; see for instance [45], [57].

The application of adaptation in both space and time has been explored for a number of problems and time-stepping methods. To give some examples, estimates for arbitrary order continuous and discontinuous Galerkin methods are considered in [41], space-time adaptation in an anisotropic setting for the first-order discontinuous Galerkin method is given in [73], while the Crank-Nicolson method is used in [70], for linear problems and in [83] for nonlinear ones. A popular choice for reaction-diffusion problems is the fully implicit backward difference formula of second order (BDF2). The method is second-order accurate, and has good stability properties applied to nonlinear problems, for instance see [42], [95], for constant time step, [11] for use in mesh adaptation, and [9], [38] for stability of the variable time-step method. As far as the author is aware, a space-time adaptation algorithm driven by a posteriori error estimates has not been considered for the BDF2 discretization. In [2] optimal order a posteriori error estimates were derived for the method applied to linear ODEs. The primary goal of this chapter is to extend the estimator in [2] to the full discretization

of nonlinear problems. First-order simplicial elements will be used for the spatial discretization, and we will employ a simple explicit a posteriori error estimator using energy techniques. The estimator will then be used to drive a complete space-time adaptation algorithm to control the error. The mesh adaptation will be performed in the anisotropic context of [44], [80].

The rest of the chapter will proceed as follows. In Section 3.2 we introduce the problem studied and all functional notation. In Section 3.3 we introduce the a posteriori error estimators and prove some upper bounds: first for the semidiscrete problem in space in Section 3.3.1. In Section 3.3.2 we consider the full BDF2 discretization. In Section 3.4 we verify numerically the equivalence of the estimator with the energy norm error. Following, we test the space-time adaptation algorithm. It is shown that applying the algorithm leads to optimal second-order behaviour in time. The efficiency of the method is considered.

3.2 Functional spaces and model problem

Let Ω be a bounded polygonal domain in $\mathbb{R}^2$ with finitely many edges. For two measurable functions u, $v : \Omega \to \mathbb{R}^n$ whose inner product is integrable we will denote by $(u, v)_\Omega = \int_\Omega (u, v)\,\mathrm{d}x$ their L^2 inner product. Let $V = H^1(\Omega)$, $H = L^2(\Omega)$, with topological duals V', H'. There exist continuous dense embeddings $V \subseteq H = H' \subseteq V'$. Therefore, defining $(u,\, f)_{V,V'} = (u,\, f)_\Omega$ for $u \in V$, $f \in H$, the duality between V and V' can be expressed in terms of the duality between H and itself.

Let $T > 0$ and define

$$\mathcal{W}(V, V') = \{w : [0, T] \to V \,;\, w \in L^2(0, T; V),\, \partial_t w \in L^2(0, T; V')\},$$

where the derivative is meant in the vector-valued distributional sense. Then, it can be shown that there is a continuous embedding $\mathcal{W}(V, V') \subseteq C([0, T]; H)$, so that in particular $w(0)$, $w(T) \in H$ are well-defined, see for instance [67].

Let $f \in C(\mathbb{R})$. We will assume that there exists $2 \leq p < \infty$, with conjugate exponent $1 < q \leq 2$ such that $f(u) \in L^q(0, T; L^q(\Omega))$ for $u \in \mathcal{W}(V, V') \cap L^p(0, T; L^p(\Omega))$, for instance if f satisfies the growth assumptions from [71]. As $\Omega \subseteq \mathbb{R}^2$, the Sobolev embedding $V \subseteq L^p(\Omega)$ holds, so that the integral $\int_0^T (f(u),\, v)_\Omega \,\mathrm{d}t$ is defined for $v \in V$. Additionally, f is assumed to satisfy one of the following:

(F1) f is continuously differentiable and for some $\alpha \geq 0$ its derivative satisfies $f'(x) \geq -\alpha$, $\forall x \in \mathbb{R}$,

(F2) f is locally Lipschitz continuous.

A typical example of (F1) is $f(x) = \sum_{k=0}^{2p-1} b_k x^k$, that is, a polynomial of odd degree with real coefficients such that $b_{2p-1} > 0$. We can now define the model initial value

problem. For $u_0 \in H$, $g \in L^2(0,T;L^2(\partial\Omega))$ and $s \in L^2(0,T;H)$, let $u \in \mathcal{W}(V,V') \cap L^p(0,T;L^p(\Omega))$ be the solution to the initial value problem

$$\begin{cases} \dfrac{\partial u}{\partial t} - \Delta u + f(u) = s, & \text{in } (0,T)\times\Omega, \\ \nabla u \cdot n = g, & \text{in } (0,T)\times\partial\Omega, \\ u(0) = u_0, & \text{in } \Omega. \end{cases} \tag{3.2.1}$$

Problem (3.2.1) will be considered in the following variational formulation:

$$\frac{\mathrm{d}}{\mathrm{d}t}(u,\,v)_\Omega + (\nabla u,\,\nabla v)_\Omega + (f(u),\,v)_\Omega = (s,\,v)_\Omega + (g,\,v)_{\partial\Omega}\,, \quad \forall v \in V$$
$$u(0) = u_0. \tag{3.2.2}$$

For existence and uniqueness of solutions to the initial value problem (3.2.2), see for instance [67] or [93].

3.3 A posteriori estimates

3.3.1 Semidiscrete problem

Notation and background

Let $\mathcal{T}_h$ be a conformal triangulation of the domain Ω with elements K of diameter h_K and consider the finite element approximation space $V_h = \{v_h \in C(\overline{\Omega}); v_h|_K \in P_1(K), \forall K \in \mathcal{T}_h\}$. Let $\overline{u}_0 \in V_h$ be an approximation of u_0. The semidiscrete approximation $u_h \in C^1(0,T;V_h)$ of u satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}(u_h,\,v_h)_\Omega + (\nabla u_h,\,\nabla v_h)_\Omega + (f(u_h),\,v_h)_\Omega = (s,\,v_h)_\Omega + (g,\,v_h)_{\partial\Omega}\,, \quad v_h \in V_h,$$
$$u_h(0) = \overline{u}_0. \tag{3.3.1}$$

Define the error $e_h = u - u_h$. Below, we introduce the notation required to derive an anisotropic residual estimator for the energy norm of the error. Relevant results are cited from the literature.

The estimator combines information on the residual with anisotropic interpolation estimates. Define the local residual $R_K(u_h) \in L^2(0,T;L^2(K))$ by

$$R_K(u_h) = \frac{\partial u_h}{\partial t} - \Delta u_h + f(u_h) - s.$$

Here Δ denotes the Laplacian operator on K. The jump of the derivative of u_h for an element K with edges e_i is defined by

$$r_K(u_h) = \sum_{i=1}^{3} [\nabla u_h]_{e_i},$$

where the jump $[\nabla v_h]_{e_i}$ over e_i is defined as follows: denoting the outward unit normal by n_i and the adjacent element (if it exists) by K', then

$$[\nabla u_h]_{e_i} = \begin{cases} 2\left(g - \nabla u_h \cdot n\right), & e_i \text{ is a boundary edge,} \\ \nabla(u_h)|_K \cdot n_i - \nabla(u_h)|_{K'} \cdot n_i, & e_i \text{ is an interior edge.} \end{cases}$$

Next, we introduce the interpolation estimates from [44] and [45]. For a triangular element K, the anisotropic information comes from the affine mapping $F_K : \hat{K} \to K$. The reference element $\hat{K}$ is taken to be the equilateral triangle centred at the origin with vertices at the points $(0,1)$, $(\frac{-\sqrt{3}}{2}, \frac{-1}{2})$, $(\frac{\sqrt{3}}{2}, \frac{-1}{2})$. The Jacobian J_K of F_K is non-degenerate, so the singular value decomposition (SVD) $J_K = \mathcal{R}_K^T \Lambda_K \mathcal{R}_K \mathcal{Z}_K$ consists of orthogonal matrices $\mathcal{R}_K$, $\mathcal{Z}_K$, and a positive definite diagonal matrix Λ_K. The matrices $\mathcal{R}_K$, Λ_K take the form

$$\mathcal{R}_K = \begin{pmatrix} r_{1,K}^T \\ r_{2,K}^T \end{pmatrix}, \qquad \Lambda_K = \begin{pmatrix} \lambda_{1,K} & 0 \\ 0 & \lambda_{2,K} \end{pmatrix},$$

where $\lambda_{1,K} \geq \lambda_{2,K} > 0$, and $r_{1,K}$, $r_{2,K}$ are orthogonal unit vectors. Geometrically, the SVD represents the deformation of the unit ball in $\mathbb{R}^2$ to an ellipse with axes of length $\lambda_{1,K}$, $\lambda_{2,K}$ in directions $r_{1,K}$, $r_{2,K}$ respectively. Moreover, the SVD also represents K in the sense that the ellipse circumscribes the element.

Let $I_h : H^1(\Omega) \to V_h$ denote a Scott-Zhang interpolation operator, see [88]. Define the following "Hessian" type matrix:

$$\tilde{G}_K(v) = \left(\int_{\Delta_K} \frac{\partial v}{\partial x_i} \frac{\partial v}{\partial x_j} \, \mathrm{d}x \right)_{i,j}, \qquad \text{for } v \in V,$$

and let

$$\tilde{\omega}_K(v) = \left(\lambda_{1,K}^2 r_{1,K}^T \tilde{G}_K(v) r_{1,K} + \lambda_{2,K}^2 r_{2,K}^T \tilde{G}_K(v) r_{2,K} \right)^{1/2}.$$

Here Δ_K is the patch of elements containing a vertex common to K. Recall that as in [74], the usual minimum-angle condition is not required, but instead, the uniform bound of the interpolation operator I_h requires a mild patch condition to hold. In what follows, a constant $C_{\hat{K}}$ will denote a positive constant which relies on such a patch condition.

Define the local anisotropic residual estimator by

$$\eta_{K,t}^2 = \left(\|R_K(u_h)\|_{0,K} + \left(\frac{h_K}{\lambda_{1,K}\lambda_{2,K}} \right)^{1/2} \|r_K(u_h)\|_{0,\partial K} \right) \tilde{\omega}_K(e_h), \tag{3.3.2}$$

and the global estimator

$$\eta_t^2 = \sum_K \eta_{K,t}^2, \qquad \eta = \left(\int_0^T \eta_t^2 \, \mathrm{d}t \right)^{1/2}. \tag{3.3.3}$$

Lemma 3.3.1. *There exists a constant $C_{\hat{K}} > 0$ independent of the mesh such that the following distributional inequality holds:*

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left\|e_h\right\|_{0,\Omega}^2 + \left|e_h\right|_{1,\Omega}^2 \leq C_{\hat{K}}\eta_t^2 - \left(f(u) - f(u_h),\, e_h\right)_\Omega . \tag{3.3.4}$$

Proof:

From the variational formulation (3.2.2) of u,

$$\begin{aligned}
&\left(\frac{\partial e_h}{\partial t},\, e_h\right)_\Omega + \left(\nabla e_h,\, \nabla e_h\right)_\Omega \\
&= \left(s - \frac{\partial u_h}{\partial t},\, e_h\right)_\Omega - \left(\nabla u_h,\, \nabla e_h\right)_\Omega + \left(g,\, e_h\right)_{\partial\Omega} - \left(f(u),\, e_h\right)_\Omega \\
&= \left(s - \frac{\partial u_h}{\partial t} - f(u_h),\, e_h\right)_\Omega - \left(\nabla u_h,\, \nabla e_h\right)_\Omega + \left(g,\, e_h\right)_{\partial\Omega} - \left(f(u) - f(u_h),\, e_h\right)_\Omega .
\end{aligned}$$

Using the fact that $I_h(e_h) \in V_h$ a.e. $t \in (0,T)$ and applying integration by parts for each triangle K

$$\begin{aligned}
&\left(s - \frac{\partial u_h}{\partial t} - f(u_h),\, e_h\right)_\Omega - \left(\nabla u_h,\, \nabla e_h\right)_\Omega + \left(g,\, e_h\right)_{\partial\Omega} \\
&= \left(s - \frac{\partial u_h}{\partial t} - f(u_h),\, e_h - I_h(e_h)\right)_\Omega - \left(\nabla u_h,\, \nabla(e_h - I_h(e_h))\right)_\Omega + \left(g,\, e_h - I_h(e_h)\right)_{\partial\Omega} \\
&= \sum_K \left(\left(R_h(u_h),\, e_h - I_h(e_h)\right)_K + \frac{1}{2}\left(r_h(u_h),\, \nabla(e_h - I_h(e_h))\right)_{\partial K}\right).
\end{aligned}$$

Therefore, applying the interpolation estimates from [44] and [45], and the Cauchy-Bunyakowsky-Schwartz inequality

$$\begin{aligned}
&\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left\|e_h\right\|_{0,\Omega}^2 + \left|e_h\right|_{1,\Omega}^2 \\
&\leq \sum_K \left(\left\|R_h(u_h)\right\|_{0,K}\left\|e_h - I_h(e_h)\right\|_{0,K} + \frac{1}{2}\left\|r_h(u_h)\right\|_{0,\partial K}\left\|\nabla(e_h - I_h(e_h))\right\|_{0,\partial K}\right) \\
&\quad - \left(f(u) - f(u_h),\, e_h\right)_\Omega \\
&\leq C_{\hat{K}}\eta_t^2 - \left(f(u) - f(u_h),\, e_h\right)_\Omega .
\end{aligned}$$

∎

From Lemma 3.3.1, it follows that the main difficulty to proceed is to deal with the last term on the right side of (3.3.4), which depends on the nonlinear function f.

Upper bounds

Here we derive two theoretical upper bounds for the energy norm of the error in terms of the estimator. Recall that the energy seminorm for $v \in L^2(0,T;V)$ is given by

$$|||v||| = \left(\int_0^T |v(t)|_{1,\Omega}^2 \, \mathrm{d}t \right)^{1/2}. \tag{3.3.5}$$

We would like to find an upper bound for the error of the form

$$|||e_h||| \leq C\eta, \tag{3.3.6}$$

where $C > 0$ is close to 1, and does not depend on the choice of mesh.

Propositions 3.3.2 and 3.3.4 can be proved if f satisfies (F1) or (F2). To simplify the presentation, we only prove the results in terms of (F1), and after we illustrate how (F2) can be substituted.

Proposition 3.3.2. *Suppose that f satisfies (F1). Then for $C_{\hat{K}} > 0$, independent of the mesh, we have*

$$\frac{1}{2} \|e_h(T)\|_{0,\Omega}^2 + \int_0^T e^{2\alpha(T-t)} |e_h(t)|_{1,\Omega}^2 \, \mathrm{d}t \leq \frac{1}{2} e^{2\alpha T} \|e_h(0)\|_{0,\Omega}^2 + C_{\hat{K}} \int_0^T e^{2\alpha(T-t)} \eta_t^2 \, \mathrm{d}t. \tag{3.3.7}$$

Proof:

From (F1) and the mean value theorem we conclude $(f(u) - f(u_h))(e_h) \geq -\alpha(e_h)^2$. Inequality (3.3.4) implies

$$\frac{1}{2} \frac{\mathrm{d}}{\mathrm{d}t} \|e_h\|_{0,\Omega}^2 + |e_h|_{1,\Omega}^2 \leq C_{\hat{K}} \eta_t^2 + \alpha \|e_h\|_{0,\Omega}^2 . \tag{3.3.8}$$

Taking the term $|e_h|_{1,\Omega}^2$ to the right hand side, we can apply Gronwall's inequality to (3.3.8) to get

$$\frac{1}{2} \|e_h(T)\|_{0,\Omega}^2 \leq \frac{1}{2} e^{2\alpha T} \|e_h(0)\|_{0,\Omega}^2 + \int_0^T e^{2\alpha(T-t)} \left(C_{\hat{K}} \eta_t^2 - |e_h(t)|_{1,\Omega}^2 \right) \mathrm{d}t, \tag{3.3.9}$$

and the result follows. ∎

In the context of (3.3.6), using the inequality $1 \leq e^{2\alpha(T-t)} \leq e^{2\alpha T}$ and supposing that we may ignore the initial error term, we are led to consider the upper bound

$$|||e_h||| \leq e^{\alpha T} C_{\hat{K}}^{1/2} \eta. \tag{3.3.10}$$

While the constant on the right hand side of (3.3.10) will not be close to 1, we can at least conclude that the upper bound holds for fixed $T > 0$. For long time scales, however, we do not expect the upper bound to be sharp. Furthermore, the value α can be quite large as will be seen in Section 3.4. For instance, solutions to the bistable equation are traveling waves with width scaled proportional to $\alpha^{-1/2}$. Therefore, the stiffer the solution, the less optimistic is the theoretical result. Recall that classical a priori estimates for (3.2.1) are typically of the form $C(u;T)e^{MT}O(h^k)$, where $C(u;T)$ depends on various norms of u, and where M is a large constant depending on the derivatives of f. It should be remarked, however, that (3.3.7) is essentially a worst case estimate in the sense that it is valid for any solution of the initial value problem (3.2.1), including solutions with finite time blowup. In what follows, we derive a stronger estimate under the assumption that the approximate solution converges at an optimal rate.

We now get to the alternative upper bound in Proposition 3.3.4. The following result is based on the assumption that the error converges faster in the L^2 norm than the H^1 seminorm. We use a condition similar to [24, (3.4)]. For the general anisotropic case, we have some partial results. We borrow from [17], where the idea is that given a P_1 approximation u_h, by utilizing the recovered gradient one may construct a P_2 approximation $u_{h,2}$ which converges faster to u, both in L^2 norm and H^1 seminorm. While this superconvergence property would be difficult to prove in general, test cases in [17] suggests that it is true in practice. Condition (3.3.11) is similar to conditions used in [81] and [72].

Lemma 3.3.3 ([15], Proposition 1, Propsition 2.3.2 in this thesis)**.** *Using the notation above, suppose that there exists $C_{\hat{K},1} > 0$ such that, for all $K \in \mathcal{T}_h$ and a.e. $t \in [0,T]$,*

$$\lambda_{1,K}\|\nabla(u_h - u_{h,2}) \cdot r_{1,K}\|_{0,K} \leq C_{\hat{K},1}\lambda_{2,K}\|\nabla(u_h - u_{h,2}) \cdot r_{2,K}\|_{0,K}. \tag{3.3.11}$$

Then there exists $C_{\hat{K},2} > 0$ independent of t such that

$$\|e_h\|_{0,K} \leq C_{\hat{K},2}\left(\lambda_{2,K}|e_h|_{1,K} + \|u - u_{h,2}\|_{0,K} + \lambda_{2,K}|u - u_{h,2}|_{1,K}\right). \tag{3.3.12}$$

From (3.3.12), we conclude that there exists $C_{\hat{K}} > 0$ for a.e. $t \in [0,T]$, such that

$$\|e_h\|_{0,\Omega} \leq C_{\hat{K}}\left(\left(\sum_K \lambda_{2,K}^2|e_h|_{1,K}^2\right)^{1/2} + \|u - u_{h,2}\|_{0,\Omega} + \left(\sum_K \lambda_{2,K}^2|u - u_{h,2}|_{1,K}^2\right)^{1/2}\right).$$

Under superconvergence assumptions for $u_{h,2}$, the last two terms on the right are assumed to be higher order, so neglected. Note that this is stronger than the usual superconvergence assumptions, which are for stationary problems. Additionally, we will assume that there exists a constant $C > 0$ such that for all $K \in \mathcal{T}_h$ and a.e. $t \in [0,T]$

$$|e_h|_{1,K} \leq CN_T^{-1/2}|e_h|_{1,\Omega}, \tag{3.3.13}$$

where N_T is the number of elements. The mesh adaptation algorithm will attempt to equidistribute the error over all elements, so that (3.3.13) should hold in practice. Then noting that $\sum_K \lambda_{2,K}^2 \leq \sum_K \lambda_{1,K}\lambda_{2,K} = |\Omega||\hat{K}|^{-1}$, there exists a constant $C_{AN} > 0$ such that up to higher-order terms

$$\|e_h\|_{0,\Omega}^2 \leq C_{AN} N_T^{-1} |e_h|_{1,\Omega}^2. \tag{3.3.14}$$

Proposition 3.3.4. *Suppose that f satisfies (F1) and the error satisfies (3.3.14). Then there exists $C_{\hat{K}} > 0$ such that*

$$\|e_h(T)\|_{0,\Omega}^2 + \left(1 - \alpha C_{AN} N_T^{-1}\right) |||e_h|||^2 \leq \|e_h(0)\|_{0,\Omega}^2 + C_{\hat{K}}\eta^2. \tag{3.3.15}$$

Proof:

Integrate (3.3.8) over t and apply condition (3.3.14). ∎

Ignoring the error on the initial condition $\|e_h(0)\|_{0,\Omega}$, estimate (3.3.15) implies

$$|||e_h||| \leq C_{\hat{K}}^{1/2} \left(1 - \alpha C_{AN} N_T^{-1}\right)^{-1/2} \eta \to C_{\hat{K}}^{1/2}\eta,$$

where $N_T \to \infty$. Therefore, we achieve an upper bound of the form (3.3.6) asymptotically with respect to the mesh size. However, note that the value of α can be large, and the constant C_{AN} is not known a priori, so it is not clear how fine the mesh needs to be so that $1 - \alpha C_{AN} N_T^{-1} > 0$.

***Remark* 1.** In general the constant C_{AN} depends on the class of meshes considered. For isotropic meshes, where $\lambda_{1,K} \simeq \lambda_{2,K} \simeq h_K \simeq h$, the error is expected to converge $O(h^2)$ in the L^2 norm and $O(h)$ in the H^1 seminorm. Then from the relation $N_T \approx Ch^{-2}$, this translates to $O(N_T^{-1})$ for the L^2 norm and $O(N_T^{-1/2})$ for the H^1 seminorm, and we obtain (3.3.14).

***Remark* 2.** Propositions 3.3.2 and 3.3.4 were proven under the assumption of (F1), which is used to prove estimate (3.3.8). On the other hand, if f satisfies (F2), if we assume that $u \in L^\infty(0,T;L^\infty(\Omega))$ and moreover, that the collection $\{u_h\}_h$ is uniformly bounded in $L^\infty(0,T;L^\infty(\Omega))$, then there exists a Lipschitz constant $C_f(u) > 0$ for f such that $|(f(u) - f(u_h))(e_h)| \leq C_f(u)(e_h)^2$ a.e., and we obtain an analogue to (3.3.8). The rest of the proof follows as before. The boundedness of u is easy to prove, provided that the initial data is smooth enough and that the equation admits an invariant region [89]. For uniform boundedness of the approximate solutions, see for instance [41]. If the solution blows up in finite time (in the L^∞ sense), then the estimates can only be local in time.

3.3.2 Space-time discretization

To simplify the presentation, for the remainder of the section we assume that g and s are both 0. Let $0 = t_0 < t_1 < \cdots < t_N = T$, with time steps $\tau_k = t_k - t_{k-1}$. We use the following notation for backward finite difference formulas

$$\begin{aligned}\partial_n^0 u_h &= u_h^n,\\ \partial_n^k u_h &= \frac{\partial_n^{k-1} u_h - \partial_{n-1}^{k-1} u_h}{\left(\frac{\tau_n + \ldots + \tau_{n-k+1}}{k}\right)}, \quad k \geq 1.\end{aligned}$$

We will also need the Newton polynomials of degree 1 and 2, given by

$$\begin{aligned}u_{h\Delta t} &= u_h^n + (t - t_n)\partial_n^1 u_h, \qquad t \in [t_{n-1}, t_n],\\ \tilde{u}_{h\Delta t} &= u_h^n + (t - t_n)\partial_n^1 u_h + \frac{1}{2}(t - t_{n-1})(t - t_n)\partial_n^2 u_h, \qquad t \in [t_{n-2}, t_n],\end{aligned}$$

that satisfy $u_{h\Delta t}(t_k) = u_h^k$ for $k = n-1$, n and $\tilde{u}_{h\Delta t}(t_k) = u_h^k$ for $k = n-2$, $n-1$, n. We define the Gear derivative $\partial_n^G u_h = \frac{\partial \tilde{u}_{h\Delta t}}{\partial t}(t_n)$, which is a second-order accurate approximation of the first derivative. In practical computation, it is a two step approximation of the following form:

$$\partial_n^G u_h = \frac{1}{\tau_n}\left[\frac{1 + 2\gamma_n}{1 + \gamma_n} u_h^n - (1 + \gamma_n) u_h^{n-1} + \frac{\gamma_n^2}{1 + \gamma_n} u_h^{n-2}\right],$$

where $\gamma_n = \frac{\tau_n}{\tau_{n-1}}$ is the step-size ratio. Denote by $\pi_h : C(\overline{\Omega}) \cap H^1(\Omega) \to V_h$ the Lagrange interpolation operator. The variable time step BDF2 method starting with backward Euler is defined as follows: find $\{u_h^n\}_n \in V_h$ that solve

$$\begin{cases} u_h^0 = \pi_h(u_0) \\ (\partial_1^1 u_h,\, v_h)_\Omega + (\nabla u_h^1,\, \nabla v_h)_\Omega + (f(u_h^1),\, v_h)_\Omega = 0, \\ \left(\partial_n^G u_h,\, v_h\right)_\Omega + (\nabla u_h^n,\, \nabla v_h)_\Omega + (f(u_h^n),\, v_h)_\Omega = 0, \qquad n \geq 2. \end{cases} \tag{3.3.16}$$

As with constant step BDF2, the method is second-order accurate in time. It is shown in [9] that the method is stable for linear problems provided $\gamma_n \leq (2 + \sqrt{13})/3 \approx 1.86$, and that the constant $\Gamma_N = \sum_{n=2}^{N-2} [\gamma_n - \gamma_{n+2}]_+$ remains bounded, where $[\cdot]_+$ denotes the non-negative part. In [38], this result is extended to semilinear parabolic problems provided $\gamma_n \leq \gamma_{max} \approx 1.910$, and provided $\tau_n \leq \tau_{max}$, where τ_{max} depends only on the nonlinear term f. For the purposes of implementing adaptive step-size control, these restrictions are not too severe. The maximum for the step-size ratio allows the step size to increase by an order of magnitude in three steps, a rapid transition. Furthermore, the boundedness of Γ_N will hold provided the variation in γ_n is not too erratic.

For the next lemma we need a linear reconstruction of $f(u_h)$:

$$I_1(f(u_h))(t) = f(u_h^n) + \frac{t-t_n}{\tau_n}(f(u_h^n) - f(u_h^{n-1})).$$

For convenience, we will define the function $\overline{\partial}^3 u_h : [0,T] \to V_h$ to be 0 for $t \leq t_2$ and otherwise to be $\partial_n^3 u_h$ on $(t_{n-1}, t_n]$. The following lemma extends results from [2], where only a constant step size was considered.

Lemma 3.3.5. *For $v_h \in V_h$ and $t \in (t_{n-1}, t_n]$ with $n \geq 3$*

$$\left(\frac{\partial \tilde{u}_{h\Delta t}}{\partial t},\, v_h\right)_\Omega + (\nabla u_{h\Delta t},\, \nabla v_h)_\Omega = -\left(I_1(f(u_h)),\, v_h\right)_\Omega - Q_n(t)\left(\partial_n^3 u_h,\, v_h\right)_\Omega,$$

where $Q_n(t) = \frac{\tau_{n-1}(\tau_n+\tau_{n-1}+\tau_{n-2})}{6\tau_n}(t-t_n)$. Furthermore, if the collection $\left\{\overline{\partial}^3 u_h\right\}_h$ is uniformly bounded in $L^2(0,T;H)$, then $\|Q_n(t)\partial_n^3 u_h\|_{L^2(0,T;H)}$ is order $O(\tau_*^2)$ where $\tau_* = \max_{n\geq 2}\tau_n$.

Proof:

If $n \geq 3$, we can apply the two step variational equality for u_h^{n-1} and u_h^n so that

$$\begin{aligned}&(\nabla u_{h\Delta t},\, \nabla v_h)_\Omega \\ &= -\left(I_1 f(u_h),\, v_h\right)_\Omega - \left(\partial_n^G u_h + \frac{t-t_n}{\tau_n}(\partial_n^G u_h - \partial_{n-1}^G u_h),\, v_h\right)_\Omega.\end{aligned}$$

Then combined with the relation $\frac{\partial \tilde{u}_{h\Delta t}}{\partial t} = \partial_n^G u_h + (t-t_n)\partial_n^2 u_h$

$$\begin{aligned}&\left(\frac{\partial \tilde{u}_{h\Delta t}}{\partial t},\, v_h\right)_\Omega + (\nabla u_{h\Delta t},\, \nabla v_h)_\Omega \\ &= -\left(I_1 f(u_h),\, v_h\right)_\Omega + (t-t_n)\left(\partial_n^2 u_h - \frac{1}{\tau_n}(\partial_n^G u_h - \partial_{n-1}^G u_h),\, v_h\right)_\Omega.\end{aligned}$$

Since $\partial_n^G u_h = \partial_n^1 u_h + \frac{\tau_n}{2}\partial_n^2 u_h$, we get

$$\begin{aligned}\partial_n^2 u_h - \frac{1}{\tau_n}(\partial_n^G u_h - \partial_{n-1}^G u_h) &= \partial_n^2 u_h - \frac{1}{\tau_n}\left(\frac{\tau_n+\tau_{n-1}}{2}\partial_n^2 u_h + \frac{\tau_n}{2}\partial_n^2 u_h - \frac{\tau_{n-1}}{2}\partial_{n-1}^2 u_h\right) \\ &= -\frac{\tau_{n-1}}{2\tau_n}(\partial_n^2 u_h - \partial_{n-1}^2 u_h) \\ &= -\frac{\tau_{n-1}(\tau_n+\tau_{n-1}+\tau_{n-2})}{6\tau_n}\partial_n^3 u_h\end{aligned}$$

and we conclude the result. Finally, for the order of convergence we note that

$$\int_{t_{n-1}}^{t_n} Q_n(t)^2 \left\|\partial_n^3 u_h\right\|_{0,\Omega}^2 \,\mathrm{d}t$$

$$= \frac{1}{108}\tau_n\tau_{n-1}^2\left(\tau_n+\tau_{n-1}+\tau_{n-2}\right)^2\left\|\partial_n^3 u_h\right\|_{0,\Omega}^2$$
$$\leq C\tau_*^4\|\partial_n^3 u_h\|_{L^2(t_{n-1},t_n;H)}^2. \tag{3.3.17}$$

∎

***Remark* 3.** In [2] it is shown that the uniform boundedness assumption of $\overline{\partial}^3 u_h$ need not hold. The optimality of the order of convergence of the estimator will be addressed in Section 3.4 for numerical examples.

For convenience, we will denote by $p_n = p_n(\tau_n, \tau_{n-1}, \tau_{n-2})$ the coefficient for $\|\partial_n^3 u_h\|_{0,\Omega}^2$ appearing in the second line of (3.3.17).

The techniques used in the proof of Theorem 3.3.6 are essentially from [70, Theorem 4.4], which is an estimate for the heat equation solved with Crank-Nicolson. We include the proof for the sake of completeness. In what follows, we denote $e_h(t) = u - u_{h\Delta t}$ and $\tilde{e}_h(t) = u - \tilde{u}_{h\Delta t}$. For the fully discrete problem, the element residual defined on $(t_{n-1}, t_n) \times K$ is

$$R_{K,n}^S(u_h) = f(\tilde{u}_{h\Delta t}) + \partial_n^G u_h - \Delta u_{h\Delta t},$$

while the edge residual defined on $(t_{n-1}, t_n) \times \partial K$ is

$$r_{K,n}^S(u_h) = [\nabla u_{h\Delta t}].$$

Theorem 3.3.6. *Suppose that (3.3.14) holds. There exists a constant $C_{\hat{K}} > 0$ independent of the mesh and step size such that for $n \geq 3$*

$$\|e_h(t_n)\|_{0,\Omega}^2 + \int_{t_{n-1}}^{t_n} |e_h(t)|_{1,\Omega}^2 \,\mathrm{d}t \leq \|e_h(t_{n-1})\|_{0,\Omega}^2$$
$$+ C_{\hat{K}}\left(\sum_K \int_{t_{n-1}}^{t_n}\left(\|R_{K,n}^S(u_h)\|_{0,K} + \frac{1}{2}\left(\frac{h_K}{\lambda_{1,K}\lambda_{2,K}}\right)^{1/2}\|r_{K,n}^S(u_h)\|_{0,\partial K}\right)\tilde{\omega}_K(\tilde{e}_h)\,\mathrm{d}t\right.$$
$$+ \sum_K \lambda_{2,K}^2\tau_n^3\left\|\partial_n^2 u_h\right\|_{0,K}^2 + p_n\left\|\partial_n^3 u_h\right\|_{0,\Omega}^2 + \frac{1}{120}\tau_n^5\left|\partial_n^2 u_h\right|_{1,\Omega}^2$$
$$\left. + \int_{t_{n-1}}^{t_n}\|f(\tilde{u}_{h\Delta t}) - I_1(f(u_h))\|_{0,\Omega}^2\,\mathrm{d}t\right). \tag{3.3.18}$$

Proof:

For $v \in V$, using the variational formulation for u

$$\left(\frac{\partial \tilde{e}_h}{\partial t}, v\right)_\Omega + (\nabla e_h, \nabla v)_\Omega$$

$$
\begin{aligned}
&= -\left(\frac{\partial \tilde{u}_{h\Delta t}}{\partial t},\, v\right)_{\Omega} - \left(\nabla u_{h\Delta t},\, \nabla v\right)_{\Omega} - \left(f(u),\, v\right)_{\Omega} \\
&= -\left(\frac{\partial \tilde{u}_{h\Delta t}}{\partial t} + f(\tilde{u}_{h\Delta t}),\, v\right)_{\Omega} - \left(\nabla u_{h\Delta t},\, \nabla v\right)_{\Omega} - \left(f(u) - f(\tilde{u}_{h\Delta t}),\, v\right)_{\Omega}. \qquad (3.3.19)
\end{aligned}
$$

For $v_h \in V_h$, we apply Lemma 3.3.5 to conclude

$$
\begin{aligned}
&\left(\frac{\partial \tilde{e}_h}{\partial t},\, v\right)_{\Omega} + \left(\nabla e_h,\, \nabla v\right)_{\Omega} \\
&= -\left(\frac{\partial \tilde{u}_{h\Delta t}}{\partial t} + f(\tilde{u}_{h\Delta t}),\, v - v_h\right)_{\Omega} - \left(\nabla u_{h\Delta t},\, \nabla (v - v_h)\right)_{\Omega} + Q_n(t)\left(\partial_n^3 u_h,\, v_h\right)_{\Omega} \\
&- \left(f(u) - f(\tilde{u}_{h\Delta t}),\, v\right)_{\Omega} - \left(f(\tilde{u}_{h\Delta t}) - I_1(f(u_h)),\, v_h\right)_{\Omega}. \qquad (3.3.20)
\end{aligned}
$$

Choose $v = \tilde{e}_h$, $v_h = I_h(\tilde{e}_h)$. Note that

$$
\begin{aligned}
\left(\nabla e_h,\, \nabla \tilde{e}_h\right)_{\Omega} &= \frac{1}{2}\left|e_h\right|_{1,\Omega}^2 + \frac{1}{2}\left|\tilde{e}_h\right|_{1,\Omega}^2 - \frac{1}{2}\left|e_h - \tilde{e}_h\right|_{1,\Omega}^2 \\
&= \frac{1}{2}\left|e_h\right|_{1,\Omega}^2 + \frac{1}{2}\left|\tilde{e}_h\right|_{1,\Omega}^2 - \frac{1}{2}\left|\tilde{u}_{h\Delta t} - u_{h\Delta t}\right|_{1,\Omega}^2 \\
&= \frac{1}{2}\left|e_h\right|_{1,\Omega}^2 + \frac{1}{2}\left|\tilde{e}_h\right|_{1,\Omega}^2 - \frac{1}{4}(t - t_n)^2 (t - t_{n-1})^2 \left|\partial_n^2 u_h\right|_{1,\Omega}^2, \qquad (3.3.21)
\end{aligned}
$$

so substituting (3.3.21) into (3.3.20), applying integration by parts over each element K, and integrating from t_{n-1} to t_n, and applying Cauchy-Bunyakowsky-Schwartz we get

$$
\begin{aligned}
&\frac{1}{2}\left\|e_h(t_n)\right\|_{0,\Omega}^2 - \frac{1}{2}\left\|e_h(t_{n-1})\right\|_{0,\Omega}^2 + \int_{t_{n-1}}^{t_n} \frac{1}{2}\left(\left|e_h(t)\right|_{1,\Omega}^2 + \left|\tilde{e}_h(t)\right|_{1,\Omega}^2\right) \mathrm{d}t \\
&\leq \underbrace{\sum_K \int_{t_{n-1}}^{t_n} \left\|f(\tilde{u}_{h\Delta t}) + \partial_n^G u_h - \Delta u_{h\Delta t}\right\|_{0,K} \left\|\tilde{e}_h - I_h(\tilde{e}_h)\right\|_{0,K} \mathrm{d}t}_{\mathrm{I}} \\
&+ \sum_K \int_{t_{n-1}}^{t_n} \underbrace{\frac{1}{2}\left\|[\nabla u_{h\Delta t}]\right\|_{0,\partial K} \left\|\tilde{e}_h - I_h(\tilde{e}_h)\right\|_{0,\partial K} \mathrm{d}t}_{\mathrm{II}} + \frac{1}{120}\tau_n^5 \left|\partial_n^2 u_h\right|_{1,\Omega}^2 \\
&+ \underbrace{\sum_K \int_{t_{n-1}}^{t_n} |t - t_n| \left\|\partial_n^2 u_h\right\|_{0,K} \left\|\tilde{e}_h - I_h(\tilde{e}_h)\right\|_{0,K} \mathrm{d}t}_{\mathrm{III}} \\
&+ \underbrace{\int_{t_{n-1}}^{t_n} \left\|Q_n(t)\partial_n^3 u_h\right\|_{0,\Omega} \left\|I_h(\tilde{e}_h)\right\|_{0,\Omega} \mathrm{d}t}_{\mathrm{IV}}
\end{aligned}
$$

$$\underbrace{-\int_{t_{n-1}}^{t_n} (f(u) - f(\tilde{u}_{h\Delta t}),\, \tilde{e}_h)_\Omega \,\mathrm{d}t}_{\mathrm{V}} \underbrace{-\int_{t_{n-1}}^{t_n} (f(\tilde{u}_{h\Delta t}) - I_1(f(u_h)),\, I_h(\tilde{e}_h))_\Omega \,\mathrm{d}t}_{\mathrm{VI}} \quad (3.3.22)$$

(Note that $e_h = \tilde{e}_h$ at $t = t_n,\, t_{n-1}$).

We will deal with each term of (3.3.22). Applying the interpolation estimates we get

$$\mathrm{I} + \mathrm{II} \leq \int_{t_{n-1}}^{t_n} C_{\hat{K}} \sum_K \left(\|R^S_{K,n}(u_h)\|_{0,K} + \frac{1}{2} \left(\frac{h_K}{\lambda_{1,K}\lambda_{2,K}} \right)^{1/2} \|r^S_{K,n}(u_h)\|_{0,\partial K} \right) \tilde{\omega}_K(\tilde{e}_h) \,\mathrm{d}t.$$

As in [70], we assume there exists $C_{eq} > 0$ independent of the mesh such that $\tilde{\omega}_K(\tilde{e}_h) \leq C_{eq}\lambda_{2,K}|\tilde{e}_h|_{1,\Omega}$. Then applying the interpolation estimates and Young's inequality, for any $\gamma > 0$

$$\begin{aligned}
\mathrm{III} &\leq \sum_K \frac{\gamma}{2} \int_{t_{n-1}}^{t_n} \lambda^2_{2,K}(t - t_n)^2 \left\|\partial_n^2 u_h\right\|^2_{0,K} \,\mathrm{d}t + \frac{C^2_{\hat{K}} C^2_{eq}}{2\gamma} \int_{t_{n-1}}^{t_n} |\tilde{e}_h|^2_{1,\Omega} \,\mathrm{d}t \\
&\leq \sum_K \frac{\gamma}{6} \lambda^2_{2,K} \tau_n^3 \left\|\partial_n^2 u_h\right\|^2_{0,K} + \frac{C^2_{\hat{K}} C^2_{eq}}{2\gamma} \int_{t_{n-1}}^{t_n} |\tilde{e}_h|^2_{1,\Omega} \,\mathrm{d}t.
\end{aligned}$$

Let $C_{I_h} > 0$ be a constant such that for all $v \in H^1(\Omega)$, $\|I_h(v)\|_{0,\Omega} \leq C_{I_h}\|v\|_{0,\Omega}$. Then

$$\mathrm{IV} \leq \frac{\gamma}{2} p_n \left\|\partial_n^3 u_h\right\|^2_{0,\Omega} + \frac{C^2_{I_h}}{2\gamma} \int_{t_{n-1}}^{t_n} \|\tilde{e}_h\|^2_{0,\Omega} \,\mathrm{d}t.$$

Assuming (F1) we get

$$\mathrm{V} \leq \alpha \int_{t_{n-1}}^{t_n} \|\tilde{e}_h\|^2_{0,\Omega} \,\mathrm{d}t.$$

Finally,

$$\mathrm{VI} \leq \frac{\gamma}{2} \int_{t_{n-1}}^{t_n} \|f(\tilde{u}_{h\Delta t}) - I_1(f(u_h))\|^2_{0,\Omega} \,\mathrm{d}t + \frac{C^2_{I_h}}{2\gamma} \int_{t_{n-1}}^{t_n} \|\tilde{e}_h\|^2_{0,\Omega} \,\mathrm{d}t$$

If we apply assumption (3.3.14) and collect the coefficients to $\int_{t_{n-1}}^{t_n} |\tilde{e}_h|^2_{1,\Omega} \,\mathrm{d}t$ on the left side we have

$$\frac{1}{2} - \frac{C^2_{\hat{K}} C^2_{eq}}{2\gamma} - \frac{C_{AN}}{N_T} \left(\alpha + \frac{C^2_{I_h}}{2\gamma} \right).$$

This can be made non-negative, for instance choosing $\gamma \geq \max\{2C_{\hat{K}}^2 C_{eq}^2, C_{I_h}^2\}$ and provided $N_T \geq 4C_{AN}(\alpha + \frac{1}{2})$. The proof is then completed by gathering all terms in (3.3.22).

∎

Note that the right hand side of the inequality in Theorem 3.3.6 still depends on the unknown ∇u. However, as is done for instance in [80], we let $\Pi_h : V_h \to V_h \times V_h$ denote a gradient recovery operator. In general, we expect $\Pi_h(u_h)$ to converge faster than ∇u_h. In this chapter, we found it sufficed to apply the simplified Zienkiewicz-Zhu operator studied in [87]. We have the following estimator for the error in space

$$\eta_{K,n}^S = \left(\int_{t_{n-1}}^{t_n} \left(\|R_{K,n}^S(u_h)\|_{0,K} + \frac{1}{2} \left(\frac{h_K}{\lambda_{1,K}\lambda_{2,K}} \right)^{1/2} \|r_{K,n}^S(u_h)\|_{0,\partial K} \right) \omega_K(\tilde{u}_{h\Delta t}) \, \mathrm{d}t \right)^{1/2}, \tag{3.3.23}$$

where for $v_h \in V_h$

$$\begin{aligned} \omega_K(v_h) &= \left(\lambda_{1,K}^2 r_{1,K}^T G_K(v_h) r_{1,K} + \lambda_{2,K}^2 r_{2,K}^T G_K(v_h) r_{2,K} \right)^{1/2}, \\ G_K(v_h) &= \left(\int_K \left(\frac{\partial v_h}{\partial x_i} - \Pi_h(v_h)_i \right) \left(\frac{\partial v_h}{\partial x_j} - \Pi_h(v_h)_j \right) \, \mathrm{d}x \right)_{i,j}. \end{aligned}$$

Note that while $\tilde{\omega}_K(\tilde{e}_h)$ is defined by taking the integral over the patch Δ_K, in the definition of $\omega_K(u_h)$ the integral is only taken over the element K. We found this simplification to produce satisfactory results in practice. The estimator for the error in time is given by

$$\eta_n^T = \left((\eta_n^{1,T})^2 + (\eta_n^{2,T})^2 + (\eta_n^{3,T})^2 + (\eta_n^{4,T})^2 \right)^{1/2}, \tag{3.3.24}$$

where

$$\begin{aligned} \eta_n^{1,T} &= \left(\frac{1}{120} \tau_n^5 \left| \partial_n^2 u_h \right|_{1,\Omega}^2 \right)^{1/2}, \qquad \eta_n^{2,T} = \left(\frac{1}{12} \sum_K \lambda_{2,K}^2 \tau_n^3 \left\| \partial_n^2 u_h \right\|_{0,K}^2 \right)^{1/2}, \\ \eta_n^{3,T} &= \left(p_n \left\| \partial_n^3 u_h \right\|_{0,\Omega}^2 \right)^{1/2}, \qquad \eta_n^{4,T} = \left(\int_{t_{n-1}}^{t_n} \| f(\tilde{u}_{h\Delta t}) - I_1(f(u_h)) \|_{0,\Omega}^2 \, \mathrm{d}t \right)^{1/2}. \end{aligned} \tag{3.3.25}$$

Adaptive algorithm

Let $TOL_S, TOL_T > 0$ denote error tolerances for the space and time error respectively. As in [70], the goal will be to adapt both the mesh and time step in order to

satisfy the following inequalities on each interval

$$0.1875\ TOL_S \leq \frac{\eta_n^S}{|||u_h|||_n} \leq 0.75\ TOL_S, \tag{3.3.26}$$

$$0.5\ TOL_T \leq \frac{\eta_n^T}{|||u_h|||_n} \leq 1.5\ TOL_T, \tag{3.3.27}$$

where $|||v|||_n = \left(\int_{t_{n-1}}^{t_n} |v(t)|_{1,\Omega}^2 \, \mathrm{d}t\right)^{1/2}$. The choice of constants appearing in the upper and lower bounds in (3.3.26) may seem unusual, particularly the upper bound 0.75 which is less than 1. However, we found that with the adaptation algorithm we describe below, the error after adaptation was generally significantly lower than the target error.

The mesh is adapted using a non-Euclidean metric, derived from η_K using the techniques from [72], which we briefly describe. The metric consists of a positive definite matrix $\mathcal{M}_K$ corresponding to each element. The Jacobian J_K relates to the metric by the relation $\mathcal{M}_K = \mathcal{R}_K^T \Lambda_K^{-2} \mathcal{R}_K$. Therefore, given the prescribed global error tolerance $\widetilde{TOL}$, the idea is to determine a new optimally defined element $\widetilde{K}$ by choosing new directions $\tilde{r}_{1,K}$, $\tilde{r}_{2,K}$ and aspect ratio $\tilde{s}_K = \frac{\tilde{\lambda}_{1,K}}{\tilde{\lambda}_{2,K}}$ which minimize $\eta_{\widetilde{K}}$ and scaling the area $\tilde{\lambda}_{1,K}\tilde{\lambda}_{2,K}$ so that the error satisfies $\eta_{\widetilde{K}} = \frac{\widetilde{TOL}}{\sqrt{N_T}}$. Since the mesh adaptation software we use in this chapter, MEF++, requires the metric to be defined on vertices, the metric needs to be averaged. We use the simple averaging for each vertex p:

$$\mathcal{M}_p = \frac{1}{N_p} \sum_{K \in \Delta_p} \mathcal{M}_K,$$

where Δ_p is the patch of elements containing p as a vertex, and N_p is the number of elements of Δ_p. Since the BDF2 method involves the variables u_h^{n-2}, u_h^{n-1}, u_h^n, the mesh should be adapted to the solution on the entire interval $[t_{n-2}, t_n]$. Therefore, following an idea from [85], for each sub-interval $[t_{s-1}, t_s]$ we construct the metric $\mathcal{M}_s$ under the condition $\frac{\eta^S}{|||u_h|||_n} = TOL_S$. Then we take the average $\mathcal{M}^n = \frac{1}{3}\sum_{s=0}^{2} \mathcal{M}_{n-s}$. We have made the common assumption that the edge residual r_K dominates the space residual R_K, and so have dropped the latter from the calculation, see [26], [22], [24].

The time-step adaptation is implemented with a standard method involving multiplicative factors. After computing the solution u_h^n the estimator η^T is computed. If (3.3.27) is not satisfied, a multiplicative factor $m > 0$ determines a new step $\tau_{new} = m\tau$. For our purposes, we chose $m = 0.67$ if the error was too large, and $m = 1.5$ if the error was too small. Note that the estimator is only defined for $n = 3$ onwards. We therefore cannot completely control the accumulation of error in the first two time steps. We will assume that if the error does not satisfy the upper bound (3.3.27) for the third step, then it also does not satisfy it for the first two. We will therefore terminate the algorithm and restart using a smaller time step.

In order to avoid asking for too much control of either the space or time error, we fix the ratio $\frac{TOL_T}{TOL_S} = \mu$. In [70], this ratio is fixed at $\mu = 1$. The choice of μ used in this chapter will be discussed further in Section 3.4, as it is observed that a suitable choice depends on the particular problem being solved.

Algorithm 3 Space-time solution-adaptation loop.

1. *Initialization:* Find a suitable starting mesh.
   (a) Repeat the following for a fixed number of iterations (5 or 10):
      i. Interpolate $u_h^0 = \pi_h(u_0)$ on the current mesh and solve for u_h^1, u_h^2, u_h^3.
      ii. Adapt the mesh using u_h^0, u_h^1, u_h^2, u_h^3.
   (b) Interpolate u_h^0 on the current mesh and solve for u_h^1, u_h^2, u_h^3, u_h^4.
2. *Space Loop:*
   (a) Compute η_n^S.
   (b) If (3.3.26) is satisfied, move on to step 3.
   (c) If (3.3.26) is not satisfied, adapt the mesh using u_h^{n-2}, u_h^{n-1}, u_h^n.
   (d) Interpolate u_h^{n-2}, u_h^{n-1} on the new mesh, and solve for u_h^n, u_h^{n+1}. Go to step 2.
3. *Time Loop:*
   (a) Compute η_n^T.
   (b) If (3.3.27) is satisfied, or if the ratio $\frac{\tau_n}{\tau_{n-1}}$ is too large or too small, compute u_h^{n+1} and go to step 2.
   (c) If (3.3.27) is not satisfied, adjust the time step, recompute u_h^n and go to step (3).

3.4 Numerical results

All numerical computations in this section are performed with the MEF++ software developed by GIREF [50].

3.4.1 Test case 1

Let $\Omega = (0,1) \times (0,1)$, $T = 0.04$, and take u to be the solution to

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u + 10^4 u(u-1)(u-0.25) = 0, & \text{in } (0,T) \times \Omega, \\ \nabla u \cdot n = 0, & \text{in } (0,T) \times \partial\Omega, \\ u(0) = e^{-100(x^2+y^2)}, & \text{in } \Omega. \end{cases} \tag{3.4.1}$$

The solution is a traveling wave with a circular profile, moving upwards and right from the bottom left corner until $u \approx 1$ in the entire domain (see Figure 3.4). We plot the space and time error estimators as a function of time (Figure 3.1) on a relatively fine uniform mesh with 12800 elements ($h = 0.0125$), and with constant time step $\tau = 0.0004$. The error estimator in space follows more or less the area of the wave-front, increasing until the front hits the top and right boundaries, and decreasing as the wave exits the domain. The time estimator behaves similarly, but has two peaks, corresponding to the wave hitting the boundary at $t = 0.027$ and just as it exits at $t = 0.038$.

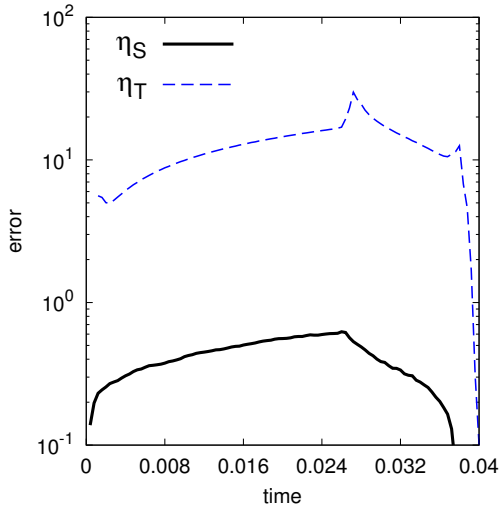


Figure 3.1: Test case 1: plot of the error estimators in time for the solution on a uniform mesh.

Effectivity index

As in [70], we define the effectivity indices

$$ei = \frac{\left((\eta^S)^2 + (\eta^T)^2\right)^{1/2}}{|||e_h|||}, \qquad ei^S = \frac{\eta^S}{|||e_h|||}, \qquad ei^T = \frac{\eta^T}{|||e_h|||},$$

where

$$|||v||| = \left(\int_0^T |v(t)|_{1,\Omega}^2 \, \mathrm{d}t \right)^{1/2}, \quad \eta^S = \left(\sum_{n \geq 1} \left(\eta_n^S \right)^2 \right)^{1/2}, \quad \eta^T = \left(\sum_{n \geq 3} \left(\eta_n^T \right)^2 \right)^{1/2}.$$

In the absence of an exact solution, we approximate the effectivity index with the use of a reference solution. Normally, a reference solution is computed on a very fine uniform mesh with a very small time step. However, we found that for this example, computing a reference solution with a uniform mesh is impractical, as we illustrate below. For fixed time step $\tau = 0.0002$, denote by $u_{\Delta t}$ the semidiscrete in time solution to (3.4.1), and $u_{H\Delta t}$ the fully discrete solution computed on a uniform mesh with $H = 0.025$. Table 3.1 illustrates the approximation of the error $|(u_{H\Delta t} - u_{\Delta t})(0.01)|_{1,\Omega}$ by the value $e_H^{h,ref} = |(u_{H\Delta t} - u_{h\Delta t})(0.01)|_{1,\Omega}$, where $u_{h\Delta t}$ is the fully discrete solution computed on various uniform meshes, and solutions using space adaptation. If we assume that the error at time $t = 0.01$ is $O(H)$ and $O(h)$, respectively, then at worst

$$\frac{\left| e_H^{h,ref} - |(u_{H\Delta t} - u_{\Delta t})(0.01)|_{1,\Omega} \right|}{|(u_{H\Delta t} - u_{\Delta t})(0.01)|_{1,\Omega}} \leq \frac{|(u_{h\Delta t} - u_{\Delta t})(0.01)|_{1,\Omega}}{|(u_{H\Delta t} - u_{\Delta t})(0.01)|_{1,\Omega}}$$

is $O\left(\frac{h}{H}\right)$. Unless $h << H$, one cannot trust the accuracy of the reference solution. With $H = 0.025$, in this case a mesh with 6400 elements, this restriction on h calls for a uniform mesh with millions of elements, and unrealistic demands for memory and CPU usage. On the other hand, with adapted meshes the approximate error approaches the value 0.953 in reasonable CPU time. In what follows, the reference solution will be computed using space adaptation with $TOL_S = 0.015625$ and with a finer time step $\tau_{ref} = 2.5 \times 10^{-5}$. The resulting meshes range from 43000 elements at time $t = 0$ to 180000 elements at $t = 0.01$.

	h (uniform meshes)			TOL_S (adapted meshes)				
	0.0125	0.00625	0.003125	0.25	0.125	0.0625	0.03125	0.015625
$e_H^{h,ref}$	0.7380	0.8992	0.9396	0.9487	0.9541	0.9556	0.9530	0.9531
max. elements	25600	102400	409600	1113	3431	10450	42172	160563
CPU time (sec)	168	871	5229	26	65	202	878	4089

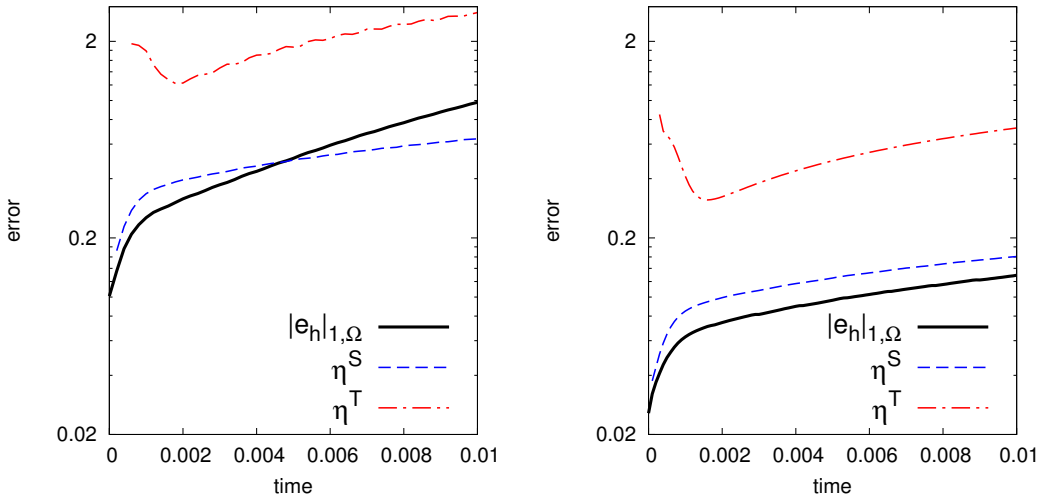
Table 3.1: Approximation of error in space at $t = 0.01$ with $H = 0.025$, $\tau = 0.0002$.

We compute the error and effectivity indices for a few values of h and τ in Table 3.2. The effectivity index ei remains at a good value ($1 \leq ei \leq 10$) provided the space and time estimators remain close in magnitude. Moreover, the index ei^S remains close to 1. The effectivity index increases as we decrease h and τ. For the coarse mesh $h = 0.025$ the solution has not yet reached the asymptotic convergence, noting that

the error decreases by a factor of 6 (first and last rows) instead of the theoretic factor of 4 predicted by the general theory. The evolution of the error in time is shown in Figure 3.2. Note that the space and time error estimators grow at the same rate for each value of h, τ, while the exact error grows slower for the lower value of h, τ. As a result, the growth of the exact error is more closely captured by the estimators for finer mesh and time step.

h	τ	$\|\|\|e_h\|\|\|$	η^S	η^T	ei	ei^S	ei^T
0.025	0.0002	0.0582	0.0494	0.199	3.52	0.849	3.42
0.025	0.0001	0.0560	0.0497	0.0528	1.29	0.888	0.943
0.0125	0.0002	0.0240	0.0247	0.199	8.34	1.03	8.29
0.0125	0.0001	0.0216	0.0250	0.0531	2.72	1.16	2.46
0.00625	0.0001	0.00966	0.0124	0.0533	5.66	1.28	5.52

Table 3.2: Error and effectivity indices for uniform mesh and constant time step.



(a) $h = 0.025$, $\tau = 0.0002$. (b) $h = 0.00625$, $\tau = 0.0001$.

Figure 3.2: Plot of the exact and estimated error in time for uniform mesh and constant time step.

***Remark* 4.** If the reference solution is computed on an adapted mesh $\mathcal{T}_h$, the integral is approximated by interpolating the gradient of $u_{H\Delta t}$ at the Gauss points of $\mathcal{T}_h$. To verify the accuracy of the integral, we apply successive subdivisions of the quadrature formula. That is, the quadrature rule on each element is obtained by splitting the element into four copies by dividing each edge in half. If the initial quadrature rule is

accurate order of h^ℓ, the resulting subdivided rule is accurate of order $\left(\frac{h}{2}\right)^\ell$. Generally, on a fine adapted mesh, the difference is less than 1% after 3 subdivisions.

We now address the optimality of the time estimator. In Table 3.3 we compute the estimators on uniform meshes and time steps, and refine with a fixed ratio $\frac{\tau^2}{h}$. We conclude that asymptotically, the space error η^S converges $O(h)$, the time estimators $\eta^{i,T}$ for $i = 1, 3, 4$ converge $O(\tau^2)$, while the estimator $\eta^{2,T}$ converges $O(h\tau)$. It was observed in [2] that the estimator may be of sub-optimal order for a given ODE system when the first step is computed with backward Euler. However, in our case we are not interested in the best approximation of the semidiscretization in space, but the approximation in both space and time of the PDE. We expect that the behaviour of the estimators in these two situations may be different.

h	τ	η^S	$\eta^{1,T}$	$\eta^{2,T}$	$\eta^{3,T}$	$\eta^{4,T}$	η^T
0.1	8.00×10^{-4}	1.69×10^{-1}	1.27×10^{-2}	8.81×10^{-2}	2.14	3.49	4.10
0.05	5.66×10^{-4}	1.40×10^{-1}	8.37×10^{-3}	2.53×10^{-2}	8.76×10^{-1}	1.35	1.61
0.025	4.00×10^{-4}	9.02×10^{-2}	4.84×10^{-3}	8.61×10^{-3}	4.36×10^{-1}	6.03×10^{-1}	7.45×10^{-1}
0.0125	2.83×10^{-4}	4.93×10^{-2}	2.60×10^{-3}	3.08×10^{-3}	2.47×10^{-1}	2.90×10^{-1}	3.81×10^{-1}
0.0625	2.00×10^{-4}	2.52×10^{-2}	1.39×10^{-3}	1.13×10^{-3}	1.38×10^{-1}	1.46×10^{-1}	2.01×10^{-1}
0.03125	1.41×10^{-4}	1.26×10^{-2}	7.11×10^{-4}	4.06×10^{-4}	7.32×10^{-2}	7.32×10^{-2}	1.04×10^{-1}
0.1	2.00×10^{-4}	1.62×10^{-1}	1.18×10^{-3}	3.09×10^{-2}	3.73×10^{-1}	2.89×10^{-1}	4.73×10^{-1}
0.05	1.41×10^{-4}	1.41×10^{-1}	6.85×10^{-4}	7.82×10^{-3}	1.08×10^{-1}	9.39×10^{-2}	1.43×10^{-1}
0.025	1.00×10^{-4}	9.17×10^{-2}	3.59×10^{-4}	2.39×10^{-3}	3.54×10^{-2}	3.97×10^{-2}	5.32×10^{-2}
0.0125	7.07×10^{-5}	4.98×10^{-2}	1.82×10^{-4}	8.34×10^{-4}	1.90×10^{-2}	1.89×10^{-2}	2.68×10^{-2}
0.0625	5.00×10^{-5}	2.53×10^{-2}	9.29×10^{-5}	2.97×10^{-4}	1.01×10^{-2}	9.41×10^{-3}	1.38×10^{-2}
0.03125	3.53×10^{-5}	1.26×10^{-2}	4.66×10^{-5}	1.05×10^{-4}	5.23×10^{-3}	4.68×10^{-3}	7.02×10^{-3}

Table 3.3: Error estimators for uniform mesh and constant time step.

Dealing with overshoot of the time estimator

We address a technical difficulty in implementing the space-time adaptation algorithm. The time estimator is observed to strongly spike at times when the mesh is adapted. To assess what is happening, we apply mesh adaptation with a constant time step for various levels of TOL_S and various time steps. We found that for fixed TOL_S, as the time step is decreased, the magnitude of the overshoot increases (see top row of Figure 3.3). Moreover, from the bottom row of Figure 3.3 we see that the overshoot of the estimator does not reflect the nature of the true error, and therefore, decreasing the time step in order to attempt to control the time estimator would not be worthwhile. The most likely explanation for the overshoot is that interpolating the solution on the new adapted mesh introduces high frequency transients, which are quickly damped. While the finite element solution itself remains good, the estimator η^T is built using finite difference schemes for second and third-order derivatives, so the

transients are magnified by small time steps. In Figure 3.3, note that the overshoot is largest for $\eta^{3,T}$, which requires the solution at four different time steps. As TOL_S is decreased, the interpolation error decreases, so smaller time steps may be taken. To apply the space-time adaptation algorithm, this implies that the ratio $\frac{TOL_T}{TOL_S}$ cannot be taken too small. In practice, for this test case, we found that the algorithm could be used with $\frac{TOL_T}{TOL_S}$ as low as 8, however, this is still unnecessarily restrictive. In Figure 3.3, we observe that the dominant terms are $\eta^{3,T}$ and $\eta^{4,T}$, and that after the initial overshoot, $\eta^{3,T}$ quickly settles back to the level of $\eta^{4,T}$. Moreover, note that in Table 3.3 for uniform meshes, the column for $\eta^{3,T}$ is closely matched by that for $\eta^{4,T}$. In what follows, we replace η^T with the modified estimator:

$$\tilde{\eta}^T = \left((\eta_n^{1,T})^2 + (\eta_n^{2,T})^2 + (\eta_n^{4,T})^2\right)^{1/2}, \tag{3.4.2}$$

While oscillations are also observed in $\eta^{1,T}$, they are small relative to $\eta^{4,T}$. Therefore, including $\eta^{1,T}$ in the estimator did not result in oscillations in (3.4.2). Another possible solution, which was not explored here, would be to use a more accurate interpolation operator as was done in [21].

Space-time adaptation

We fix the ratio $\frac{TOL_T}{TOL_S} = 0.75$ and apply space-time adaptation (Algorithm 3). Note that as the wave exits the domain, the normalizing factor $|u_h|_{1,\Omega}$ appearing in (3.3.26) quickly decreases to zero. To avoid pathological behaviour, we use the modified form $\max(|u_h|_{1,\Omega}, 1)$. Examples of adapted meshes and the solution are shown in Fig. 3.4. The majority of the elements are located near the wave-front and elongated parallel to the wave. Note that the mesh is somewhat coarser near the center of the wave, where the wave is nearly linear in the direction orthogonal to the wave. Figure 3.5 shows the evolution of the error estimators, the time step, and the number of elements for a complete computation using $TOL_S = 0.125$, $TOL_T = 0.09375$. The adaptation maintains the relative space and time error near a constant value until the normalizing factor $|u_h|_{1,\Omega}$ quickly drops off as the wave exits the domain. Recalling the results form uniform meshes (3.1), the space and time estimators grow and decrease slowly as the surface of the wave-front increases and eventually exits. This behaviour is reflected in the adaptive algorithm with a slow growth and decrease in the number of elements, and in the fact that the time step is mostly constant. Moreover, the decreases in the time step and corresponding spikes in the time estimator coincide with the wave hitting the boundary. Note that as the wave exits the time step becomes large and the number of elements drops significantly. The evolution of the time step follows the trend noted in Figure 3.1, with a nearly constant time step where the error grows linearly, and with two sharp decreases when the wave hits the boundary. The oscillation in η^S is a reflection of the mesh adaptation, with the error dropping off suddenly after adaptation, and growing quickly as the wave-front moves

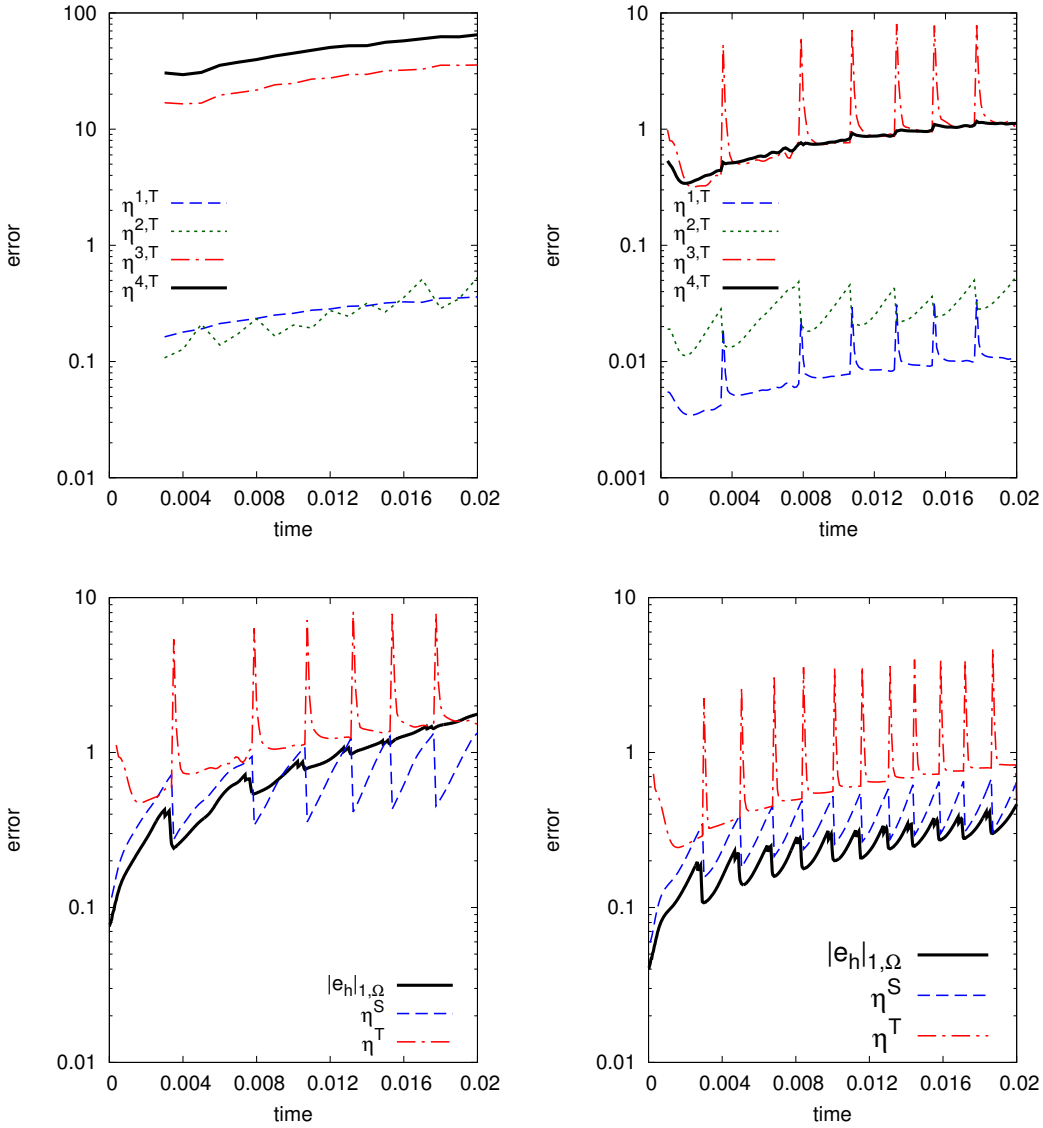


Figure 3.3: Test case 1. Top: evolution of time estimator applying mesh adaptation with $TOL_S = 0.5$ with constant time step $\tau = 0.001$ (left) and $\tau = 0.000125$ (right). Bottom: evolution of the estimators and exact error for $TOL_S = 0.5$, $\tau = 0.000125$ (left) and $TOL_S = 0.52$, $\tau = 8.85625 \times 10^{-5}$ (right).

beyond the refined region. In Table 3.4 it is shown that the estimated error terms converge proportionally to TOL_S, TOL_T when applying the space-time adaptation algorithm. In Table 3.5 we collect some additional statistics. Columns three, four and six address the efficiency of the time-step adaptation algorithm. From column three, we note that the number of time-step modifications is essentially independent of the error tolerance. Recall that each time the time step is modified, the current time step needs to be recomputed after which condition (3.3.27) is checked once again. The current time step may possibly need to be recomputed several times before the error condition is satisfied. Column four collects the total number of times that a time step needs to be recomputed, and we conclude that in this case, no additional recomputations are required to satisfy the error condition. Then from column 6, it is shown that the total percentage of CPU time spent computing the time estimator is low (no more than 0.3%). From column five, we observe that the total number of time steps has a moderate growth as a function of TOL_T^{-1}. More precisely, as we decrease TOL_T by 2, the number of time steps increases by approximately a factor of $\sqrt{2}$ (as was observed in [70] for the Crank-Nicolson method). This is consistent with the fact that the BDF2 method is second-order accurate. Lastly, note that the number of mesh adaptations grows sublinearly as TOL_S is decreased.

TOL_S	TOL_T	η^S	$\tilde{\eta}^T$	$\eta^{1,T}$	$\eta^{2,T}$	$\eta^{4,T}$
0.5	0.375	1.38×10^{-1}	2.36×10^{-1}	2.53×10^{-3}	8.84×10^{-3}	2.36×10^{-1}
0.25	0.1875	6.98×10^{-2}	1.21×10^{-1}	1.46×10^{-3}	2.72×10^{-3}	1.21×10^{-1}
0.125	0.09375	3.65×10^{-2}	6.07×10^{-2}	7.52×10^{-4}	1.07×10^{-3}	6.07×10^{-2}
0.0625	0.046875	1.92×10^{-2}	3.05×10^{-2}	3.67×10^{-4}	4.43×10^{-4}	3.05×10^{-2}
0.03125	0.0234375	9.77×10^{-3}	1.52×10^{-2}	1.76×10^{-4}	1.65×10^{-4}	1.52×10^{-2}

Table 3.4: Estimated error after space-time adaptation for Test case 1 with $TOL_T = 0.75 \cdot TOL_S$.

TOL_S	remeshings	step adapts	step restarts	total steps	CPU time est (%)
0.5	21	11	11	305	0.22
0.25	38	11	11	432	0.18
0.125	48	9	9	611	0.19
0.0625	62	10	10	862	0.24
0.03125	73	11	11	1219	0.28

Table 3.5: Mesh and time-step statistics for space-time adaptation for Test case 1 with $TOL_T = 0.75 \cdot TOL_S$.

3.4.2 Test case 2

The goal of the following test case is to determine the gain in efficiency using space-time adaptation. In particular, we are interested in situations where the time error

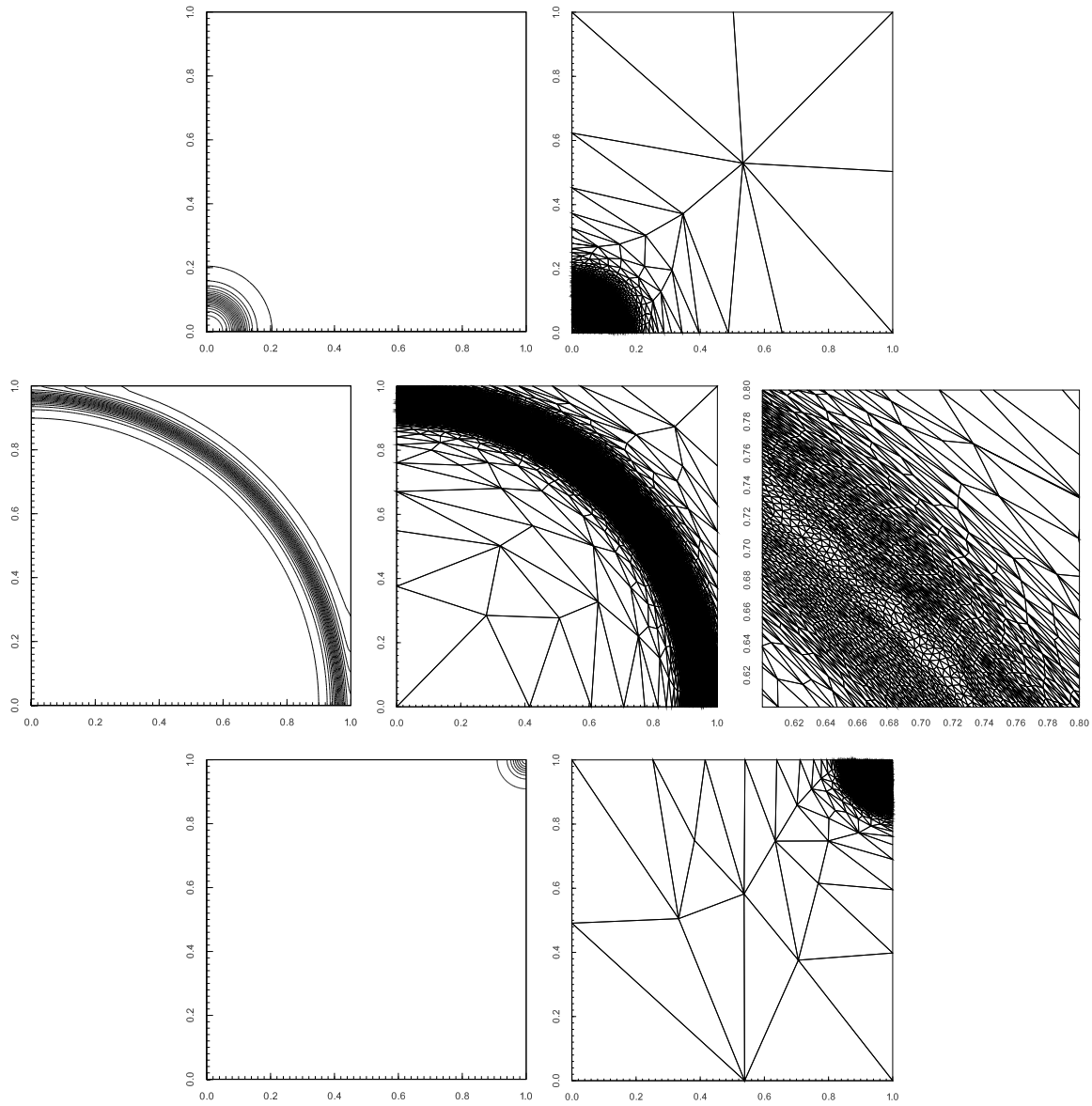


Figure 3.4: Examples of contours and adapted meshes with $TOL_S = 0.125$, $TOL_T = 0.09375$ at $t = 0.00141933$ (top), $t = 0.0265008$ (center) with zoom to the wave-front (center right), and at $t = 0.0382107$ (bottom).

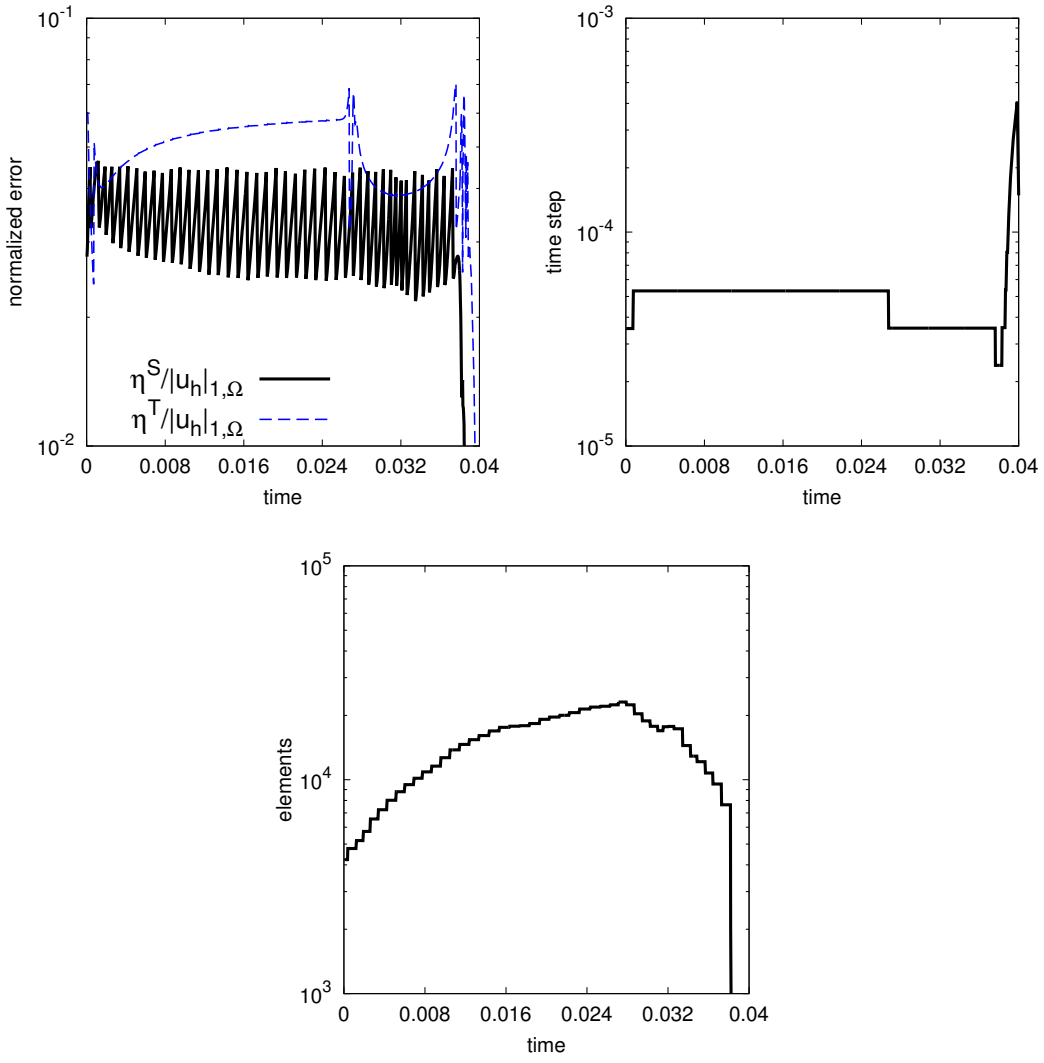


Figure 3.5: Test case 1: space-time adaptive algorithm with $TOL_S = 0.125$, $TOL_T = 0.09375$: the relative space and time error estimators (top left), the evolution of the time step (top right), and number of elements (bottom).

varies more rapidly and in magnitude than in the previous test case. We choose the same domain, initial data and reaction term as in (3.4.1) and take $T = 0.01$. We replace the diffusion term with $-\text{div}(A\nabla u)$ where

$$A(x,y) = 1 + 19\left(1 - \tanh^2\left(20\left(\left(x^2+y^2\right)^{1/2} - 0.3\right)\right)\right).$$

The wave slows down and spreads outwards as it enters the region with higher diffusion where $(x^2+y^2)^{1/2} \approx 0.3$, and then picks up speed as the wave exits this region. The theory in this case follows as in Section 3.3 with appropriate changes to the definition of the edge and element residuals. To simplify the computation of the estimator, on each element K we replace A by its value A_K at the barycenter of K. Then we define the new edge residual $r_K^A(u_h) = A_K[\nabla u_h]$. This definition is somewhat different from what appears in the literature, for instance in [80] and [73], but was easier to implement.

As for test case 1, we compute a reference solution using mesh adaptation with $TOL_S = 0.015625$, with meshes ranging from 55000 to 250000 elements, and a relatively small constant time step $\tau_{ref} = 1.25 \times 10^{-5}$. A solution is then computed on a uniform mesh with $h = 0.025$, with 6400 elements, and time step $\tau = 2 \times 10^{-4}$ and the estimated error and exact error are approximated from the reference solution, plotted in Figure 3.6 (left). We see that the estimated error η^S is generally close to the exact error. The time estimator $\tilde{\eta}^T$ decreases as the wave enters the area of higher diffusion and slows down. The estimator then increases as the wave exits this region speeds up again. Additionally, we compute the estimated and exact error when applying mesh adaptation with $TOL_S = 0.25$, with meshes of about 1000 elements, and constant time step 2×10^{-4}, plotted in Figure 3.6 (right). The observed trend for $\tilde{\eta}^T$ is similar to that for uniform meshes. The estimated error η^S is again close to the exact error, both dropping whenever the mesh is adapted. As noted for test case 1, the space error grows more quickly for adapted meshes compared to uniform meshes since the wave moves beyond the refined region of the mesh. Finally, Figure 3.7 illustrates the evolution of the error and time step applying the space-time adaptation algorithm with $TOL_S = 0.25$, $TOL_T = 0.375$, also resulting in meshes of about 1000 elements.

The efficiency of the method is assessed by determining the level of error that can be obtained in a given total CPU time. We compare computations on uniform meshes with constant time step, adapted meshes with constant time step, and applying the full space-time adaptation algorithm. For each uniform mesh, we compute the solution with a variety of time steps, collected in Table 3.6, to determine the "best" time step for a given value of h. For instance, for $h = 0.05$, as the time step is decreased below $\tau = 10^{-4}$, the error changes very little while the total CPU time increases. The same approach is taken to get Table 3.7 for mesh adaptation with constant time step, finding the best time step for a fixed value TOL_S. The results for space-time adaptation are reported in Table 3.8. For each value of TOL_S, we compute the solution with various levels of TOL_T. Figure 3.8 plots the CPU time vs. the

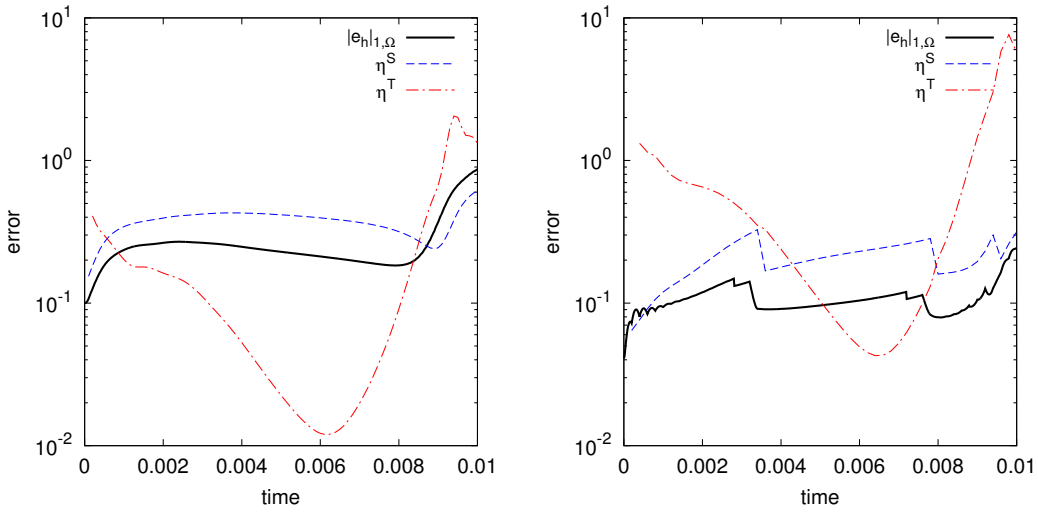


Figure 3.6: Plot of error for Test case 2. Plot of error in time on a uniform mesh with $h = 0.025, \tau = 2 \times 10^{-4}$ (left) and using space-only adaptation with $TOL_S = 0.25, \tau = 2 \times 10^{-4}$ (right).

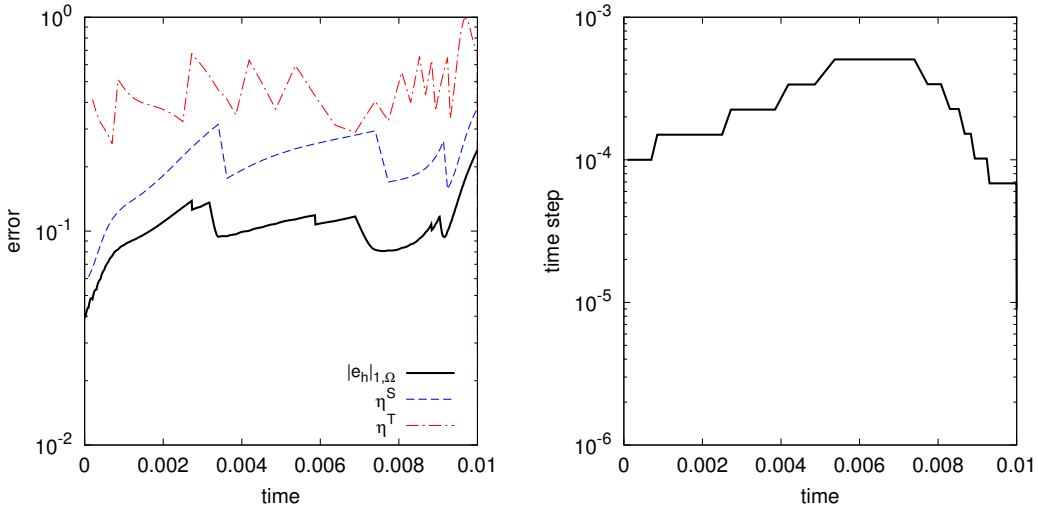


Figure 3.7: Test case 2: plot of the error (left) and time step (right) using space-time adaptation with $TOL_S = 0.25$ and $TOL_T = 0.375$.

h	τ	$\lvert e_h\rvert_{1,\Omega}$	η^S	ei^S	CPU(sec)	steps
0.05	4.00×10^{-4}	0.0874	0.0740	0.85	3.5	25
0.05	2.00×10^{-4}	0.0772	0.0742	0.96	6.3	50
0.05	1.00×10^{-4}	0.0748	0.0743	0.99	11.3	100
0.05	5.00×10^{-5}	0.0742	0.0743	1.00	21.1	200
0.05	2.50×10^{-5}	0.0741	0.0743	1.00	41.9	400
0.05	1.25×10^{-5}	0.0741	0.0743	1.00	80.0	800
0.025	4.00×10^{-4}	0.0478	0.0380	0.79	14.7	25
0.025	2.00×10^{-4}	0.0336	0.0379	1.13	25.3	50
0.025	1.00×10^{-4}	0.0303	0.0380	1.25	45.9	100
0.025	5.00×10^{-5}	0.0296	0.0380	1.28	85.6	200
0.025	2.50×10^{-5}	0.0294	0.0380	1.29	172.4	400
0.0125	4.00×10^{-4}	0.0321	0.0188	0.59	62.5	25
0.0125	2.00×10^{-4}	0.0159	0.0188	1.18	108.8	50
0.0125	1.00×10^{-4}	0.0128	0.0188	1.47	189.0	100
0.0125	5.00×10^{-5}	0.0122	0.0188	1.54	350.6	200
0.0125	2.50×10^{-5}	0.0121	0.0188	1.56	683.4	400
0.0125	1.25×10^{-5}	0.0120	0.0188	1.56	1289.1	800

Table 3.6: Test case 2: error and total CPU time for uniform meshes with constant time step.

error, with the plot on the right including only the best points for each method. We conclude that for this test case, both the space and space-time adaptation methods achieve significantly lower error level for a given amount of CPU time compared to using uniform meshes. Moreover, when using mesh adaptation, there does not seem to be much difference between the constant and adapted time-step methods. For both methods, either a choice of time step, or a choice of TOL_T will give a comparable result for a given TOL_S. Note that for all computational methods, the effectivity index ei^S is close to 1, varying between 0.5 and 2.

It should be noted that one of the advantages of using the space-time adapted method is that it is more automated. From Table 3.8, the choice $TOL_T \approx TOL_S$ consistently gives a good result. This observation suggests a strategy for how to proceed for general problems. Choosing $TOL_T = TOL_S$, start with a small time step. If the time estimator is too large, start over with a smaller time step. Otherwise, if the estimator is small, the time step will automatically increase to an appropriate value, and the rest of the computation proceeds. When solving with a constant time step, it is not always clear how the step should be chosen. A further remark is that since the nonlinear system (3.3.16) is solved with Newton's method, increasing the time step may increase the number of Newton iterations required for each time, and therefore, the CPU time savings for using a larger time step may not be as great as expected. Another approach would be to try a linearized version of (3.3.16) where the reaction term is evaluated explicitly at u_h^{n-2} and u_h^{n-1}.

TOL_S	τ	$\lvert e_h\rvert_{1,\Omega}$	η^S	ei^S	CPU(sec)	steps
0.5	4.00×10^{-4}	0.0232	0.0415	1.79	11.3	25
0.5	2.00×10^{-4}	0.0196	0.0385	1.96	11.5	50
0.5	1.00×10^{-4}	0.0226	0.0389	1.72	12.7	100
0.5	5.00×10^{-5}	0.0228	0.0389	1.71	14.9	200
0.5	2.50×10^{-5}	0.0235	0.0399	1.70	19.7	400
0.25	4.00×10^{-4}	0.0194	0.0225	1.16	21.5	25
0.25	2.00×10^{-4}	0.0112	0.0213	1.91	24.2	50
0.25	1.00×10^{-4}	0.0117	0.0210	1.79	25.1	100
0.25	5.00×10^{-5}	0.0132	0.0212	1.61	30.6	200
0.25	2.50×10^{-5}	0.0142	0.0214	1.50	46.9	400
0.25	1.25×10^{-5}	0.0144	0.0211	1.47	78.2	800
0.25	6.25×10^{-6}	0.0127	0.0213	1.68	129.7	1600
0.125	4.00×10^{-4}	0.0241	0.0118	0.49	60.6	25
0.125	2.00×10^{-4}	0.0073	0.0112	1.55	58.3	50
0.125	1.00×10^{-4}	0.0055	0.0109	1.98	78.3	100
0.125	5.00×10^{-5}	0.0058	0.0108	1.88	100.8	200
0.125	2.50×10^{-5}	0.0059	0.0109	1.85	161.2	400
0.125	1.25×10^{-5}	0.0059	0.0108	1.83	267.2	800

Table 3.7: Test case 2: error and total CPU time for adapted meshes with constant time step.

TOL_S	TOL_T	$\lvert e_h\rvert_{1,\Omega}$	η^S	ei^S	CPU(sec)	steps
0.5	3	0.0430	0.0444	1.03	12.2	21
0.5	1.5	0.0210	0.0422	2.00	11.8	29
0.5	0.74	0.0207	0.0415	2.01	12.0	40
0.5	0.375	0.0218	0.0415	1.90	11.3	56
0.5	0.1875	0.0251	0.0408	1.62	11.8	76
0.25	0.75	0.0152	0.0230	1.51	21.8	38
0.25	0.375	0.0109	0.0223	2.06	22.6	55
0.25	0.1875	0.0119	0.0220	1.85	23.7	79
0.25	0.09735	0.0133	0.0222	1.67	25.0	111
0.25	0.046875	0.0146	0.0218	1.50	29.3	155
0.25	0.0234375	0.0131	0.0217	1.65	34.4	220
0.125	0.375	0.0126	0.0119	0.94	65.9	42
0.125	0.1875	0.0076	0.0116	1.52	65.4	58
0.125	0.09375	0.0057	0.0114	2.00	69.9	82
0.125	0.046875	0.0056	0.0112	2.00	75.8	113
0.125	0.0234375	0.0058	0.0112	1.93	89.9	160
0.125	0.01171875	0.0059	0.0111	1.90	107.4	223

Table 3.8: Test case 2: error and total CPU time for adapted meshes with adapted time step.

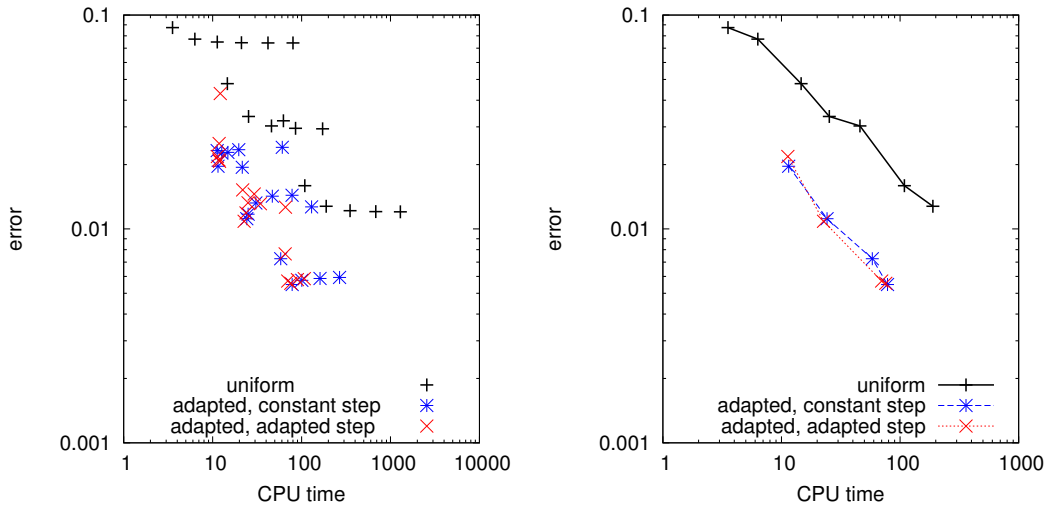


Figure 3.8: Test case 2: CPU time vs. error for all computations (left) and best points (right).

3.5 Conclusion

This chapter introduced an error estimator for the energy norm of the error for nonlinear reaction-diffusion equations, approximated with piecewise linear finite elements in space, and the BDF2 method in time. The estimator is shown to be reliable under certain conditions of the reaction term, and it is verified numerically that the effectivity index is close to 1. A space-time adaptation method is proposed to simultaneously control the estimators for the space and time error. The mesh is adapted using a metric to control the anisotropic nature of the error. The adaptation method is applied to a traveling wave with constant diffusion and a solution with variable diffusion. It was found that the space-time method is at least as efficient in terms of achieving a global level of error in a given CPU time compared to applying mesh adaptation with a constant time step. Looking ahead, the method can be applied to more complex situations, such systems of equations exhibiting multiscale behaviour, where it is reasonable to expect that a combined space-time approach will lead to improved efficiency.

Chapter 4

Application to electrophysiological models

4.1 Introduction

In this chapter we apply the general theoretical and adaptive framework of the previous chapter to the monodomain model used in cardiac electrophysiology.

A variety of adaptive techniques have been applied to the monodomain and bidomain models. The majority of this work has been dedicated to isotropic meshes. A heuristic method is employed in [66], where mesh elements containing the wave-front are successively refined, based on the observation that the variation of the solution is low outside this region. A similar heuristic method is employed in [100], where elements are refined based on the magnitude of the gradient of the transmembrane potential, which again, is expected to occur primarily in the wave front. Additionally, the time step is adapted based on the variation of certain ionic currents, specific to the Luo-Rudy I model. Both [66] and [100] report an increase in computational efficiency compared to uniform refinement, however, without a clear theoretical framework it is not clear how widely their methods can be applied. Towards this direction, in [28] the mesh and time step are adapted based on estimating the local truncation error for a finite difference scheme using a Richardson extrapolation, and a similar technique is applied to a finite element method in [96], while in [77], the interpolation error is approximated for trilinear elements. In [46], the authors use a hierarchical error estimator, which approximates the residual in a higher-order space, for a multilevel finite element discretization in space and a Rosenbrock time-stepping scheme. This work is applied to computations on a realistic heart geometry, simulating fibrillation dynamics in [35]. The theoretical foundation of the method can be found in [64]. A different approach is taken in [4], where they apply a p-adaptive method. The error estimator is based on an approximation of the error in space only, between the semidiscretization in time and the full discretization. An advantage of their method

is a relatively quick reassembly of the matrices involved, since the mesh connectivity is preserved. While the adaptive approach taken in this thesis means that we cannot avoid the issue of matrix reassembly, the general methodology we take is to not adapt the mesh too often.

Work on adaptive methods in an anisotropic setting applied to electrophysiology has only begun recently. The first work in that direction can be found in [10]. The mesh adaptation is based on a simple hierarchical estimator, constructed from gradient and Hessian recovery techniques. Results are presented for 2D spiral waves, where the elements of the mesh are aligned for the minimization of the gradient of the error, capturing the anisotropic features of the solution. Results in 3D are presented in [11], where the authors apply a Riemannian metric adaptive technique using Hessian recovery, and extend their results to 3D scroll waves in [12]. In these works, however, mesh adaptation is performed after every time step. The potential gains in CPU time using the adaptive mesh could be offset by the increased overhead required to perform the adaptation steps and to recompute the matrices to solve the system. In [91], the adaptation step is performed after fixed intervals, and a speedup of up 11.2 compared to the uniform method is observed. However, a significant percentage of the total computation time is still spent adapting the mesh, up to 79%. A parallel version was presented in [90]. In [85], a similar method was used while the portion of time spent adapting the mesh was reported to be about 25% of the total time when adapting every 10 time steps.

The work in this chapter continues in the vein of anisotropic adaptive methods. Unlike the anisotropic methods used previously, the adaptation is driven by an error estimator computed from the residual of the system involved. The general framework uses the anisotropic interpolation estimates for piecewise linear elements found in [44] and [45], which is combined with a gradient recovery operator to obtain an a posteriori error estimator as done in [80] and [72]. As in [80], mesh adaptation is only performed when the estimated error is above or below a certain threshold, therefore systematically avoiding the issue of too frequent adaptation found in previous anisotropic methods in electrophysiology. We find that it is necessary to treat the ODE variables of the system differently as they do not benefit from parabolic smoothing. We propose a modified estimator that does not make use of a residual. In some previous work, for instance in [91], [66], [100], the adaptation takes into account only the variation of the transmembrane potential, likely based on the observation that it varies more rapidly than the ODE variables. Here we illustrate numerically that all variables should be taken into account, especially when simulating a heartbeat with realistic duration scales. We use the fully implicit backward difference formula of second order (BDF2) for the discretization in time. The error due to the time discretization is approximated with an extension of the estimator from [2] to the nonlinear setting and with a variable time step. A space-time adaptation algorithm is employed.

The rest of the chapter will proceed as follows. In Section 4.2 we recall the monodomain model. In Section 4.3 we give the full discretization of the problem and derive the error estimates. The theoretical applicability of the residual estimate will be discussed for both the FitzHugh-Nagumo and the Mitchell-Schaeffer ionic models. In addition to a residual estimator, we consider a simplified estimator only based on the recovered gradient for the recovery variable. Section 4.4 presents the numerical results. After a brief discussion of the adaptive algorithm, two numerical test cases are presented. The first, with the FitzHugh-Nagumo model, is used in order to verify the reliability of the estimators and the mesh adaptation algorithm. The second, with the Mitchell-Schaeffer model, models a heartbeat with realistic phase time scales and is suitable to a space-time adaptation algorithm.

4.2 Model problem

In this chapter, we keep the notation introduced in the preceding chapter. In particular, let $V = H^1(\Omega), H = L^2(\Omega)$ and let V_h be the space of piecewise linear continuous functions. For u_0, $w_0 \in H$, let $u \in \mathcal{W}(V, V')$, $w \in \mathcal{W}(H, H')$ be the solution to the following initial value problem:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \Delta u + F(u,w) = 0, & \text{in } (0,T)\times\Omega, \\ \dfrac{\partial w}{\partial t} + G(u,w) = 0, & \text{in } (0,T)\times\Omega, \\ \nabla u \cdot n = 0, & \text{on } (0,T)\times\partial\Omega, \\ u(0) = u_0, \\ w(0) = w_0, \end{cases} \tag{4.2.1}$$

where F, $G : \mathbb{R}^2 \to \mathbb{R}$ are continuous. Moreover, we assume that F, G satisfy one of the following conditions for every bounded domain $\mathcal{D} \subseteq \mathbb{R}^2$,

C4.2.1 there exists $\alpha_{\mathcal{D}} > 0$ such that for $x = (x_1, x_2)$, $y = (y_1, y_2) \in \mathcal{D}$

$$(F(x) - F(y))(x_1 - y_1) + (G(x) - G(y))(x_2 - y_2) \geq -\alpha_{\mathcal{D}} \|x - y\|_2^2,$$

C4.2.2 there exists $\alpha_{\mathcal{D}} > 0$ such that for x, $y \in \mathcal{D}$

$$\max\{|F(x) - F(y)|, |G(x) - G(y)|\} \leq \alpha_{\mathcal{D}} \|x - y\|_2.$$

To improve the estimates, we also consider a stronger form of (C4.2.1)

C4.2.3 there exist positive constants $\alpha_{\mathcal{D}}$, $\beta_{\mathcal{D}}$ such that for $x = (x_1, x_2)$, $y = (y_1, y_2) \in \mathcal{D}$

$$\begin{aligned}(F(x) - F(y))(x_1 - y_1) + (G(x) - G(y))(x_2 - y_2) \\ \geq -\alpha_{\mathcal{D}}(x_1 - y_1)^2 + \beta_{\mathcal{D}}(x_2 - y_2)^2.\end{aligned}$$

The domain $\mathcal{D}$ will be assumed implicitly assumed, and we will denote $\alpha_{\mathcal{D}}$ by α and $\beta_{\mathcal{D}}$ by β in (C4.2.1), (C4.2.2) and (C4.2.3). In particular, we will assume the solution, and approximate solutions, are uniformly bounded in $L^\infty(\Omega)$. Some details on the boundedness assumption follow in Section 4.3.

There are several well-posedness results for (4.2.1). In [20], a weak solution is proven to exist globally in time provided the reaction terms satisfy mild growth conditions, which apply for instance to the FitzHugh-Nagumo and Aliev-Panfilov models. Uniqueness is proven for the FitzHugh-Nagumo model. Additionally, local existence of more regular solutions is proven provided further regularity of the reaction terms and the initial data. In [19] existence is proven for a regularized version of the Mitchell-Schaeffer model. Note that the results in the references above apply to the bidomain problem, for which the monodomain problem is a simplification.

The error estimates in Section 4.3 require additionally that w belongs to $L^2(0,T;V)$. We can obtain more regular solutions by considering so-called strong solutions of (4.2.1) in the setting of [55]. For instance, if we modify the definition of the spaces Z and Z^α in [20, Section 4] to use the space $B = V \cap L^\infty(\Omega)$, then on a maximal interval $[0, \tau_{\max})$, there exists a unique strong solution in the sense of [20, Definition 18] provided the function $(u,w) \in Z^\alpha \mapsto (F(u,w), G(u,w)) \in Z$ is locally Lipschitz continuous and $(u_0, w_0) \in Z^\alpha$. In particular, we have $w \in C([0, \tau_{max}); V)$. For the solution to exist globally, we require that the local Lipschitz condition be replaced by a global one. For F, this can be achieved provided there exists a priori bounds on u, w in the $L^\infty(\Omega)$ norm. Such bounds will be discussed for specific ionic models in Section 4.3 due to the existence of invariant rectangles. For G this is not quite enough since the norm on V involves the gradient. However, it suffices to assume that G and its first derivatives are globally Lipschitz continuous, which is satisfied for the ionic models considered in this chapter.

4.3 Discretization and error estimator

We use the notation from Section 3.3.2. For the full discretization of (4.2.1), we apply continuous piecewise linear finite element approximation for the space discretization and fully implicit BDF2 for the time discretization. The initial conditions are interpolated $u_h^0 = I_h(u_0), w_h^0 = I_h(w_0)$. Recall that $\partial_n^G u_h$ for $n \geq 2$ denotes the the second-order accurate discrete time derivatives used for the BDF2 methods. Then for $n \geq 2$, u_h^n, w_h^n are obtained by solving the variational problem

$$\begin{cases} \left(\partial_n^G u_h,\, \phi_h\right)_\Omega + \left(\nabla u_h^n,\, \nabla \phi_h\right)_\Omega + \left(F(u_h^n, w_h^n),\, \phi_h\right)_\Omega = 0, \\ \left(\partial_n^G w_h,\, \psi_h\right)_\Omega + \left(G(u_h^n, w_h^n),\, \psi_h\right)_\Omega = 0, \end{cases} \tag{4.3.1}$$

for all $\phi_h,\, \psi_h \in V_h$. The solution for $n = 1$ is obtained by the backward Euler method.

Recall that $u_{h\Delta t}$ and $\tilde{u}_{h\Delta t}$ respectively denote the piecewise linear and quadratic in time reconstructions. Define the element residuals

$$R^U_{K,n} = \frac{\partial \tilde{u}_{h\Delta t}}{\partial t} - \Delta u_{h\Delta t} + F(\tilde{u}_h, \tilde{w}_h),$$
$$R^W_{K,n} = \frac{\partial \tilde{w}_{h\Delta t}}{\partial t} + G(\tilde{u}_h, \tilde{w}_h),$$

and edge residual

$$r^U_{K,n} = [\nabla u_{h\Delta t}].$$

Define $e_u = u - u_{h\Delta t}$, $\tilde{e}_u = u - \tilde{u}_{h\Delta t}$ and similarly define e_w and $\tilde{e}_w$. Define the local space estimators

$$\tilde{\eta}^{S,U}_{K,n}(e_u) = \left(\int_{t_{n-1}}^{t_n} \left(\|R^U_{K,n}\|_{0,K} + \left(\frac{h_K}{\lambda_{1,K}\lambda_{2,K}} \right)^{1/2} \|r^U_{K,n}\|_{0,\partial K} \right) \tilde{\omega}_K(\tilde{e}_u)\, \mathrm{d}t \right)^{1/2},$$
$$\tilde{\eta}^{S,W}_{n,K}(e_w) = \left(\int_{t_{n-1}}^{t_n} \|R^W_{K,n}\|_{0,K} \tilde{\omega}_K(\tilde{e}_w)\, \mathrm{d}t \right)^{1/2}, \tag{4.3.2}$$

and global space estimators

$$\tilde{\eta}^{S,U}_n = \left(\sum_K (\tilde{\eta}^{S,U}_{K,n})^2 \right)^{1/2}, \qquad \tilde{\eta}^{S,W}_n = \left(\sum_K (\tilde{\eta}^{S,W}_{K,n})^2 \right)^{1/2}. \tag{4.3.3}$$

Define the time estimators

$$\eta^{T,U}_n = \left((\eta^{1,T,U}_n)^2 + (\eta^{2,T,U}_n)^2 + (\eta^{3,T,U}_n)^2 \right)^{1/2},$$
$$\eta^{T,W}_n = \left((\eta^{2,T,W}_n)^2 + (\eta^{3,T,W}_n)^2 \right)^{1/2}, \tag{4.3.4}$$

where

$$\eta^{1,T,U}_n = \left(\frac{1}{120} \tau_n^5 \left| \partial_n^2 u_h \right|^2_{1,\Omega} \right)^{1/2}, \qquad \eta^{2,T,U}_n = \left(p_n \left\| \partial_n^3 u_h \right\|^2_{0,\Omega} \right)^{1/2}$$
$$\eta^{3,T,U}_n = \left(\int_{t_{n-1}}^{t_n} \|F(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}) - I_1(F(u_h, w_h))\|^2_{0,\Omega}\, \mathrm{d}t \right)^{1/2}, \tag{4.3.5}$$

where p_n is defined as in (3.3.17). The time estimators for W are defined similarly. Note that there is no corresponding estimator for $\eta^{1,T,U}$ for w since there is no Laplacian operator. Also, note that there is no analogous term corresponding to $\eta^{2,T}$ from (3.3.25). This is due to the fact that we are using the time derivatives of the quadratic reconstructions $\frac{\partial \tilde{u}_h}{\partial t}$ and $\frac{\partial \tilde{w}_h}{\partial t}$ in the element residuals, instead of retaining only the constant part $\partial_n^G u_h$ and $\partial_n^G w_h$ as is done in Chapter 3. The following lemma is an easy extension of Lemma 3.3.5.

Lemma 4.3.1. *For $v_h \in V_h$ and $t \in (t_{n-1}, t_n]$ with $n \geq 3$*

$$\left(\frac{\partial \tilde{u}_{h\Delta t}}{\partial t},\, v_h\right)_\Omega + (\nabla u_{h\Delta t},\, \nabla v_h)_\Omega = -\left(I_1(F(u_h, w_h)),\, v_h\right)_\Omega - Q_n(t)\left(\partial_n^3 u_h,\, v_h\right)_\Omega,$$
$$\left(\frac{\partial \tilde{w}_{h\Delta t}}{\partial t},\, v_h\right)_\Omega = -\left(I_1(G(u_h, w_h)),\, v_h\right)_\Omega - Q_n(t)\left(\partial_n^3 w_h,\, v_h\right)_\Omega,$$

where $Q_n(t) = \frac{\tau_{n-1}(\tau_n+\tau_{n-1}+\tau_{n-2})}{6\tau_n}(t - t_n)$.

Recall the definition of the global and local energy norm for $v \in L^2(0, T; V)$

$$|||v||| = \left(\int_0^T |v|_{1,\Omega}^2 \, \mathrm{d}t\right)^{1/2}, \qquad |||v|||_n = \left(\int_{t_{n-1}}^{t_n} |v|_{1,\Omega}^2 \, \mathrm{d}t\right)^{1/2}.$$

For convenience, we also write $|||v|||_\infty$ and $|||v|||_{\infty,n}$ to respectively denote the $L^\infty(0, T; H)$ and $L^\infty(t_{n-1}, t_n; H)$ norms. Recall as in Chapter 3 that $I_h : H^1(\Omega) \to V_h$ denotes a Scott-Zhang interpolation operator.

Theorem 4.3.2. *Suppose that (C4.2.1) or (C4.2.2) is satisfied. There exists a constant $C_{\hat{K}} > 0$ independent of the mesh and the step size and $C(\alpha) > 0$ depending on α such that, for $n \geq 3$,*

$$\begin{aligned}
&|||e_u|||_{\infty,n}^2 + |||e_w|||_{\infty,n}^2 + |||e_u|||_n^2 \\
&\leq C_{\hat{K}} e^{C(\alpha)\tau_n}\left(\|e_u(t_{n-1})\|_{0,\Omega}^2 + \|e_w(t_{n-1})\|_{0,\Omega}^2 + (\tilde{\eta}_n^{S,U})^2 + (\tilde{\eta}_n^{S,W})^2 + (\eta_n^{T,U})^2 + (\eta_n^{T,W})^2\right).
\end{aligned} \tag{4.3.6}$$

If (C4.2.3) is satisfied, then there exists $C_{AN} > 0$ depending on the superconvergence assumption (3.3.14) such that, for $n \geq 3$,

$$\begin{aligned}
&|||e_u|||_{\infty,n}^2 + |||e_w|||_{\infty,n}^2 + \left(1 - \alpha\frac{C_{AN}}{N_T}\right)|||e_u|||_n^2 + \frac{\beta}{2}\|e_w\|_{L^2(t_{n-1},t_n;H)}^2 \\
&\leq C_{\hat{K}}\left(\|e_u(t_{n-1})\|_{0,\Omega}^2 + \|e_w(t_{n-1})\|_{0,\Omega}^2 + (\tilde{\eta}_n^{S,U})^2 + (\tilde{\eta}_n^{S,W})^2 + (\eta_n^{T,U})^2 + (\eta_n^{T,W})^2\right).
\end{aligned} \tag{4.3.7}$$

Proof:

By inspecting the proof of Theorem 3.3.6 up to (3.3.20), we get for the first equation

$$\begin{aligned}
&\left(\frac{\partial \tilde{e}_u}{\partial t},\, \tilde{e}_u\right)_\Omega + (\nabla e_u,\, \nabla \tilde{e}_u)_\Omega \\
&= -\left(\frac{\partial \tilde{u}_{h\Delta t}}{\partial t} + F(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}),\, \tilde{e}_u - I_h(\tilde{e}_u)\right)_\Omega - (\nabla u_{h\Delta t},\, \nabla(\tilde{e}_u - I_h(\tilde{e}_u)))_\Omega
\end{aligned}$$

$$- Q_n(t)\left(\partial_n^3 u_h,\, I_h(\tilde{e}_u)\right)_\Omega - \left(F(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}) - I_1(F(u_h, w_h)),\, I_h(\tilde{e}_u)\right)_\Omega$$
$$- \left(F(u,w) - F(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}),\, \tilde{e}_u\right)_\Omega . \tag{4.3.8}$$

Therefore, applying Young's inequality and (3.3.21), there exist positive constants C_1, C_2 depending on the interpolation operator I_h such that

$$\begin{aligned}
&\frac{d}{dt}\|e_u\|_{0,\Omega}^2 + |e_u|_{1,\Omega}^2 + |\tilde{e}_u|_{1,\Omega}^2 \\
&\qquad \le C_1 \left(\sum_K \left(\|R_{K,n}^U\|_{0,K} + \left(\frac{h_K}{\lambda_{1,K}\lambda_{2,K}}\right)^{1/2} \|r_{K,n}^U\|_{0,\partial K} \right) \tilde{\omega}_K(\tilde{e}_u) \right. \\
&\qquad + \frac{1}{4}(t-t_n)^2(t-t_{n-1})^2 \left|\partial_n^2 u_h\right|_{1,\Omega}^2 + \left\|Q_n(t)\partial_n^3 u_h\right\|_{0,\Omega}^2 \\
&\qquad \left. + \|F(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}) - I_1(F(u_h, w_h))\|_{0,\Omega}^2 \right) \\
&\qquad + C_2\|\tilde{e}_u\|_{0,\Omega}^2 - \left(F(u,w) - F(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}),\, \tilde{e}_u\right)_\Omega ,
\end{aligned} \tag{4.3.9}$$

Similarly, for the second equation we get

$$\begin{aligned}
\left(\frac{\partial \tilde{e}_w}{\partial t},\, \tilde{e}_w\right)_\Omega &= -\left(\frac{\partial \tilde{w}_{h\Delta t}}{\partial t} + G(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}),\, \tilde{e}_w - I_h(\tilde{e}_w)\right)_\Omega \\
&\quad - Q_n(t)\left(\partial_n^3 w_h,\, I_h(\tilde{e}_w)\right)_\Omega - \left(G(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}) - I_1(G(u_h, w_h)),\, I_h(\tilde{e}_w)\right)_\Omega \\
&\quad - \left(G(u,w) - G(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}),\, \tilde{e}_w\right)_\Omega .
\end{aligned} \tag{4.3.10}$$

so there exist constants C_3, C_4, also depending on I_h, such that

$$\begin{aligned}
\frac{d}{dt}\|e_w\|_{0,\Omega}^2 &\le C_3 \left(\sum_K \|R_{K,n}^W\|_{0,K}\omega_K(\tilde{e}_w) + \left\|Q_n(t)\partial_n^3 w_h\right\|_{0,\Omega}^2 \right. \\
&\qquad \left. + \|G(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}) - I_1(G(u_h, w_h))\|_{0,\Omega}^2 \right) \\
&\qquad + C_4\|\tilde{e}_w\|_{0,\Omega}^2 - \left(G(u,w) - G(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}),\, \tilde{e}_w\right)_\Omega .
\end{aligned} \tag{4.3.11}$$

For convenience, we respectively denote by $S_1(t)$ and $S_2(t)$ the right sides of (4.3.9) and (4.3.11). If we assume either (C4.2.1) or (C4.2.2), then

$$\begin{aligned}
&C_2\|\tilde{e}_u\|_{0,\Omega}^2 + C_4\|\tilde{e}_w\|_{0,\Omega}^2 \\
&- \left(F(u,w) - F(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}),\, \tilde{e}_u\right)_\Omega - \left(G(u,w) - G(\tilde{u}_{h\Delta t}, \tilde{w}_{h\Delta t}),\, \tilde{e}_w\right)_\Omega \\
&\le (C(\alpha))(\|\tilde{e}_u\|_{0,\Omega}^2 + \|\tilde{e}_w\|_{0,\Omega}^2),
\end{aligned}$$

where $C(\alpha) = \max\{C_2, C_4\} + \alpha$. Taking the sum of (4.3.9) and (4.3.11), and applying Gronwall's inequality we obtain for a.e. $t \in (t_{n-1}, t_n]$

$$\|e_u(t)\|_{0,\Omega}^2 + \|e_w(t)\|_{0,\Omega}^2 + \int_{t_{n-1}}^t e^{C(\alpha)(s-t_{n-1})} |e_u(s)|_{1,\Omega}^2 \, \mathrm{d}s$$

$$\leq \|e_u(t_{n-1})\|_{0,\Omega}^2 + \|e_w(t_{n-1})\|_{0,\Omega}^2 + \int_{t_{n-1}}^{t} e^{C(\alpha)(s-t_{n-1})}(S_1(s)+S_2(s))\,\mathrm{d}s$$
$$\leq \|e_u(t_{n-1})\|_{0,\Omega}^2 + \|e_w(t_{n-1})\|_{0,\Omega}^2 + \int_{t_{n-1}}^{t_n} e^{C(\alpha)(t-t_{n-1})}(S_1(t)+S_2(t))\,\mathrm{d}t.$$

Since t is arbitrary, we have

$$|||e_u|||_{\infty,n}^2 + |||e_u|||_{\infty,n}^2 + \int_{t_{n-1}}^{t_n} e^{C(\alpha)(t-t_{n-1})}\,|e_h(t)|_{1,\Omega}^2\,\mathrm{d}t$$
$$\leq 2\left(\|e_u(t_{n-1})\|_{0,\Omega}^2 + \|e_w(t_{n-1})\|_{0,\Omega}^2 + \int_{t_{n-1}}^{t_n} e^{C(\alpha)(t-t_{n-1})}(S_1(t)+S_2(t))\,\mathrm{d}t\right),$$

and conclude (4.3.6). On the other hand, suppose (C4.2.3) holds. Then

$$C_2\|\tilde{e}_u\|_{0,\Omega}^2 + C_4\|\tilde{e}_w\|_{0,\Omega}^2$$
$$-\,(F(u,w)-F(\tilde{u}_{h\Delta t},\tilde{w}_{h\Delta t}),\,\tilde{e}_u)_\Omega - (G(u,w)-G(\tilde{u}_{h\Delta t},\tilde{w}_{h\Delta t}),\,\tilde{e}_w)_\Omega$$
$$\leq (C_2+\alpha)\|\tilde{e}_u\|_{0,\Omega}^2 + (C_4-\beta)\|\tilde{e}_w\|_{0,\Omega}^2.$$

From the proof of Theorem 3.3.6, we note that the constant C_4 can be made arbitrarily small, so in particular, we can take $C_4 < \frac{\beta}{2}$. Then (4.3.7) follows after applying (3.3.14) and integrating over t. ∎

Note that the estimators (4.3.3) used in Theorem 4.3.2 depend on the unknown solutions u and w. Therefore, as is normally done in the literature, we replace the value of ∇u and ∇w in (4.3.2) with recovered versions $\Pi_h(u_h)$ and $\Pi_h(w_h)$, where $\Pi_h : V_h \to V_h \oplus V_h$ is a superconvergent gradient recovery operator. More in detail, define

$$G_K(u_h) = \left(\int_K \left(\frac{\partial u_h}{\partial x_i} - \Pi(u_h)_i\right)\left(\frac{\partial u_h}{\partial x_j} - \Pi(u_h)_j\right)\mathrm{d}x\right)_{i,j}, \tag{4.3.12}$$

$$\omega_K(u_h) = (\lambda_{1,K}^2 r_{1,K}^T G_K(u_h) r_{1,K} + \lambda_{2,K}^2 r_{2,K}^T G_K(u_h) r_{2,K})^{1/2}, \tag{4.3.13}$$

and

$$\eta_{K,n}^{S,U} = \left(\int_{t_{n-1}}^{t_n}\left(\|R_{K,n}^U\|_{0,K} + \left(\frac{h_K}{\lambda_{1,K}\lambda_{2,K}}\right)^{1/2}\|r_{K,n}^U\|_{0,\partial K}\right)\omega_K(u_h)\,\mathrm{d}t\right)^{1/2}, \tag{4.3.14}$$

with similar definition for $\eta_{K,n}^{S,W}$. In the remainder, we use the gradient recovery version of the estimators. In this chapter we found the simplified Zienkiewicz-Zhu estimator

from [87] was sufficient for our purposes. Also note that as in Chapters 2 and 3, the definition of $G_K(u_h)$ that we used in practice did not involve an integral on the patch Δ_K, but only on the element K.

In what follows, we look at specific ionic models. For convenience, we have the following lemma.

Lemma 4.3.3. *Suppose there exist positive constants μ_1, μ_2, μ_3 such the following inequalities hold uniformly on the domain $\mathcal{D}$:*

$$\frac{\partial F}{\partial x_1} \geq -\mu_1, \quad \left|\frac{\partial F}{\partial x_2}\right| + \left|\frac{\partial G}{\partial x_1}\right| \leq \mu_2, \quad \frac{\partial G}{\partial x_2} \geq \mu_3. \tag{4.3.15}$$

Then (C4.2.3) holds with $\alpha = \mu_1 + \frac{\mu_2^2}{2\mu_3}$, and $\beta = \frac{\mu_3}{2}$.

Proof:

For any $x, y \in \mathcal{D}$, by the mean value theorem there exists ξ_1, ξ_2 on the line segment between x and y such that

$$\begin{aligned}
&(F(x) - F(y))(x_1 - y_1) + (G(x) - G(y))(x_2 - y_2) \\
&= \frac{\partial F}{\partial x_1}(\xi_1)(x_1 - y_1)^2 + \left(\frac{\partial F}{\partial x_2}(\xi_1) + \frac{\partial G}{\partial x_1}(\xi_2)\right)(x_1 - y_1)(x_2 - y_2) + \frac{\partial G}{\partial x_2}(\xi_2)(x_2 - y_2)^2.
\end{aligned}$$

Applying (4.3.15) and Young's inequality, for arbitrary $\gamma > 0$

$$\begin{aligned}
&(F(x) - F(y),\, G(x) - G(y)) \cdot (x - y) \\
&\geq -\mu_1(x_1 - y_1)^2 - \mu_2|x_1 - y_1||x_2 - y_2| + \mu_3(x_2 - y_2)^2 \\
&\geq -\left(\mu_1 + \frac{\mu_2^2}{2\gamma}\right)(x_1 - y_1)^2 + \left(\mu_3 - \frac{\gamma}{2}\right)(x_2 - y_2)^2.
\end{aligned}$$

The result follows choosing $\gamma = \mu_3$. ∎

4.3.1 FitzHugh-Nagumo model

$$F(u, w) = f_1(u) + w = u(u - a)(u - 1) + w, \tag{4.3.16}$$
$$G(u, w) = -\epsilon(\kappa u - w), \tag{4.3.17}$$

and $0 < a < 1$, $\epsilon, \kappa > 0$. It is clear that F, G are locally Lipschitz continuous. Invariant rectangles of arbitrary size exist for the model, see [89], [31], so the solution remains bounded if $u_0, w_0 \in L^\infty(\Omega)$. Furthermore, letting $\mu > 0$ be such that $f_1'(x) \geq -\mu$, then applying Lemma 4.3.3 we obtain (C4.2.3) with $\alpha = \mu + \frac{(1+\epsilon\kappa)^2}{2\epsilon}$ and $\beta = \frac{\epsilon}{2}$.

4.3.2 Mitchell-Schaeffer model

$$F(u,w) = \frac{1}{\tau_{in}} w u^2 (u-1) + \frac{1}{\tau_{out}} u \tag{4.3.18}$$

$$G(u,w) = \begin{cases} \frac{1}{\tau_{open}}(w-1), & u < u_{gate}, \\ \frac{1}{\tau_{close}} w, & u \geq u_{gate}, \end{cases} \tag{4.3.19}$$

where τ_{in}, τ_{out}, τ_{open}, τ_{close}, u_{gate} are positive constants, with $0 < u_{gate} < 1$. The reaction term G is discontinuous on the line $u = u_{gate}$, and F, G do not satisfy (C4.2.1) or (C4.2.2) for a domain $\mathcal{D}$ crossing this line. We will use the following smoothed version:

$$\begin{aligned} G(u,w) &= \frac{1}{\tau_u}((1-s)(w-1) + sw), \\ \tau_u &= \tau_{open} + (\tau_{close} - \tau_{open})s, \\ s(u,\kappa,u_{gate}) &= \frac{1}{2}(1 + \tanh(\kappa(u - u_{gate}))), \end{aligned}$$

where $\kappa > 0$. Both F, G are now C^1 so that (C4.2.2) holds. As for the FitzHugh-Nagumo model, the solutions remain bounded provided the initial conditions are bounded. Moreover, applying the maximum principle from [31], for arbitrary $\epsilon > 0$, the region $\{(u,\, w) : -\epsilon \leq u \leq 1,\, 0 \leq w \leq 1\}$ is invariant, and we conclude that the region $\{(u,\, w) : 0 \leq u \leq 1,\, 0 \leq w \leq 1\}$ is invariant as well.

We now show that (C4.2.3) also holds. For the first reaction term,

$$\frac{\partial F}{\partial x_1} = \frac{1}{\tau_{in}} x_2 (3x_1^2 - 2x_1) + \frac{1}{\tau_{out}}.$$

For a typical heartbeat, (u, w) is in $[0,1]^2$ so we may take $\mu_1 = \frac{1}{3\tau_{in}} - \frac{1}{\tau_{out}}$ from the bound on w. From the bound on u we get

$$\left| \frac{\partial F}{\partial x_2} \right| \leq \max_{x_1 \in [0,1]} \frac{1}{\tau_{in}} |x_1^2 (x_1 - 1)| = \frac{1}{\tau_{in}} \frac{4}{27}.$$

For the second term,

$$\frac{\partial G}{\partial x_2} = \frac{1}{\tau_u} \geq \min\left\{ \frac{1}{\tau_{open}}, \frac{1}{\tau_{close}} \right\}.$$

Finally,

$$\frac{\partial G}{\partial x_1} = \kappa \operatorname{sech}^2(\kappa(x_1 - u_{gate})) \frac{1}{\tau_u^2} \left(\tau_u - (x_2 + s - 1)(\tau_{close} - \tau_{open}) \right),$$

so assuming that x_2 remains in $[0,1]$,

$$\left|\frac{\partial G}{\partial x_1}\right| \leq \kappa \frac{\tau_u + |\tau_{close} - \tau_{open}|}{\tau_u^2} \leq \kappa \frac{\tau_{max} + |\tau_{close} - \tau_{open}|}{\tau_{min}^2},$$

where $\tau_{min,max} = \min, \max\{\tau_{open}, \tau_{close}\}$. To summarize, from Lemma 4.3.3 we obtain (C4.2.3) with $\alpha = \frac{1}{3}\tau_{in}^{-1} - \tau_{out}^{-1} + \frac{1}{2}\left(\frac{4}{27}\tau_{in}^{-1} + \kappa\frac{\tau_{max}+|\tau_{close}-\tau_{open}|}{\tau_{min}^2}\right)^2 \tau_{max}$ and $\beta = \frac{1}{2\tau_{max}}$. Under the assumption that $\tau_{in} \ll \tau_{out} \ll \tau_{open}, \tau_{close}$, we have $\alpha = O\left(\tau_{in}^{-2} + \kappa^2 \frac{\tau_{max}^3}{\tau_{min}^2}\right)$. Note that when taking the limit $\kappa \to \infty$ to approach the original discontinuous model, the constant α blows up. The proof above relies on a uniform bound for $\frac{\partial G}{\partial x_1}$, which clearly does not exist for the discontinuous model, and therefore a different approach would have to be taken.

4.3.3 A modified estimator for the recovery variable

We discuss some technical limitations for the use of the estimators introduced in Section 4.3 for use mesh adaptation, in particular for the ODE variable w. To simplify the discussion, we consider a simple ODE model:

$$\begin{cases} \dfrac{\partial w}{\partial t} + \mu w = f, \\ w(0) = w_0. \end{cases} \tag{4.3.20}$$

where $\mu \geq 0$ is a constant and $f \in L^2(0,T;H)$. Applying the same arguments as in Theorem 4.3.2 it can be shown that the error satisfies

$$\begin{aligned} &\|e_w(t_n)\|_{0,\Omega}^2 + \mu \int_{t_{n-1}}^{t_n} \|e_w(t)\|_{0,\Omega}^2 \, \mathrm{d}t \\ &\leq \|e_w(t_{n-1})\|_{0,\Omega}^2 + C_{\hat{K}} \left((\eta_n^{S,W})^2 + (\eta_n^{T,W})^2\right), \end{aligned} \tag{4.3.21}$$

where the residual defined on K is $R_{K,n}^W = \frac{\partial \tilde{w}_h}{\partial t} + \mu \tilde{w}_h - f$. If we are in the special situation that f is in $L^2(0,T;V_h)$, then $R_{K,n}^W$ is as well. Therefore, by Galerkin orthogonality, $R_{K,n}^W(t_n) = 0$ and from this it is easy to conclude that $R_{K,n}^W(t) \to 0$ uniformly in t as $\tau_n \to 0$. By contrast, note that for parabolic problems, the differential operator A rarely satisfies $A(V_h) \subseteq V_h$. Moreover, since the time estimator $\eta_n^{T,W}$ is $O(\tau_n^{5/2})$, for τ_n small enough, the dominant term on the right side of (4.3.21) is the initial error $\|e_w(t_{n-1})\|_{0,\Omega}^2$. This term is highly sensitive to the interpolation error at each mesh adaptation step. More specifically, if π_h^n is the interpolation operator for the new mesh obtained at time t_n, and $w_h^{n-1,old}$ is the solution computed

on the previous mesh, then we have $w_h^{n-1} = \pi_h^n(w_h^{n-1,old})$. Then from the relation $e_w(t_{n-1}) = w(t_{n-1}) - w_h^{n-1,old} + w_h^{n-1,old} - \pi_h^n(w_h^{n-1,old})$ we get

$$\|e_w(t_{n-1})\|_{0,\Omega} \leq \|e_w^{old}(t_{n-1})\|_{0,\Omega} + \|w_h^{n-1,old} - \pi_h(w_h^{n-1,old})\|_{0,\Omega}. \tag{4.3.22}$$

If the interpolation error is not taken into account when adapting the mesh, the last term on the right will spoil the control of the error. For instance, in regions of the domain where the solution varies rapidly, the interpolation error remains large independent of the time step. At the same time, the element residual converges to 0 uniformly as the time step is decreased.

An even worse situation can occur if the source term f does not belong to the finite element space. In this situation, we might expect the element residual $R_{K,n}^W$ to be large only in regions where f varies rapidly, and is far locally, in the L^2 sense on the element K, from the finite element space. As we will demonstrate shortly with a numerical example, the variation in f does not necessarily correspond with the local variation of the solution, an observation which has significant implications on the effectiveness of the estimator for mesh adaptation purposes.

We propose the use of a modified estimator of the space error for w. By Young's inequality, we note that

$$\begin{aligned}\eta_{K,n}^{S,W} &\leq \left(\frac{1}{2}\int_{t_{n-1}}^{t_n} \|R_{K,n}^W\|_{0,K}^2 \,\mathrm{d}t + \frac{1}{2}\int_{t_{n-1}}^{t_n} \omega_K^2 \,\mathrm{d}t\right)^{1/2} \\ &\leq \left(\int_{t_{n-1}}^{t_n} \|R_{K,n}^W\|_{0,K}^2 \,\mathrm{d}t\right)^{1/2} + \left(\int_{t_{n-1}}^{t_n} \omega_K^2 \,\mathrm{d}t\right)^{1/2},\end{aligned}$$

which splits the estimator into the residual, and the gradient recovery part. We therefore consider the following local estimator for w

$$\omega_{K,n}^{S,W} = \left(\int_{t_{n-1}}^{t_n} \omega_K^2 \,\mathrm{d}t\right)^{1/2}, \tag{4.3.23}$$

and global estimator

$$\omega_n^{S,W} = \left(\sum_K (\omega_{K,n}^{S,W})^2\right)^{1/2}, \quad \omega^{S,W} = \left(\sum_n (\omega_n^{S,W})^2\right)^{1/2}. \tag{4.3.24}$$

Comparing the estimators $\eta_{K,n}^{S,W}$ with $\omega_{K,n}^{S,W}$, we see that the value of the former will depend highly on the value of the element residual in a given region. The residual may indicate new regions where the error accumulation will occur, but it may still dampen the estimator in regions of sharp gradient. The latter estimator, on the other hand, is problem-independent, its value only depending on the local features of the

approximate solution. Heuristically, we might expect that this estimator will be large in regions where the initial interpolation error (4.3.22) is large.

We illustrate the above issues with the residual estimator, and relative merits of the simplified estimator, with a numerical example. Let $\Omega = (0, 100) \times (0, 100)$, $T = 100$, and let w solve (4.3.20) with $f = -0.016875 \tanh(x - 0.25t - 50)$, $\mu = 0.01$ and initial condition $w(0) = 0$. The solution of this equation mimics the behaviour of the recovery variable for the FitzHugh-Nagumo model. In particular, the unknown u in (4.3.17) is replaced by $\tanh(x - 0.25t - 50)$, which represents a traveling wave moving to the right with fixed speed. The initial condition is chosen so that the initial interpolation error in (4.3.21) will be zero, and therefore, the only important term on the right side for the error in space is the residual term.

The solution is computed on a uniform mesh with 6400 elements and constant time step $\tau = 1$. The exact error is estimated by computing a reference solution on a much finer mesh with 1638400 elements, obtained after 4 subdivisions of the coarser mesh. The results of the error computation are shown in Figure 4.1. To highlight the local properties of the estimator, in addition to the plot of the estimator $\eta^{S,W}$ in the usual L^2 in time sense, as highlighted in (4.3.21), we plotted the estimators $\eta^{S,W}(t)$ and $\omega^{S,W}(t)$ as functions of time. The plot on the left shows the evolution of the estimators and the exact error over time. We clearly have the following upper bounds for the error at time t:

$$\|e_w(t)\|_{0,\Omega} \leq \left(\int_0^t (\eta^{S,W})^2 \, \mathrm{d}t \right)^{1/2}. \tag{4.3.25}$$

For larger t, this value tends to overestimate the true error. This is not too surprising, since we are estimating an error which is L^∞ in time with an estimator which accumulates in an L^2 sense.

To get a better picture of what is going on, we look at the local distribution of the error over the elements of the mesh. Figure 4.1 (right) plots these values over the line $y = 50$ at time $t = 100$ (left y-axis), as well as the superposition of the function $\tanh(x - 0.25t - 50)$ (right y-axis). In particular, the estimator $\eta^{S,W}(t)$ only detects error in regions where source term varies rapidly, while near $x = 45$, where the source term is close to zero, the error is underestimated by several orders of magnitude. Therefore, while the estimator is reliable in the sense that (4.3.25) holds, the estimator cannot be localized in space and time. We can't expect to reliably estimate the local value of the error $\|e_w(t)\|_{0,K}$ without taking into account the value of the estimator at all previous times, so the estimator is impractical for mesh adaptation. Note that from the left plot in the figure, the estimator, taken pointwise in time, does appear to give an accurate estimate of the error in the sense that

$$\|e_w(t)\|_{0,\Omega} \leq C\eta^{S,W}(t),$$

for some constant C which is close to 1. However, given the above observations, such an upper bound is not something we expect in general. Finally, note that the estimator $\omega^{S,W}(t)$ on the other hand remains within an order of magnitude of the exact error throughout the domain, and therefore gives a local representation of the error.

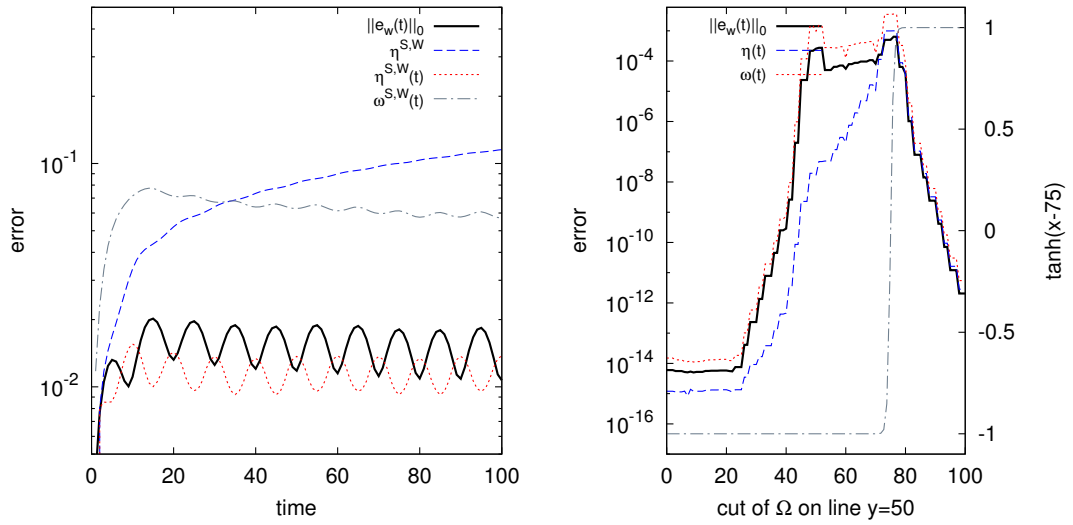


Figure 4.1: Left: plot of the total error over time. Right: plot of the error (left y-axis) and source term (right y-axis) over the line $y = 50$ at time $t = 100$.

All mesh adaptation done in Section 4.4 for problem (4.2.1) makes use of the estimator (4.3.23) for the variable w, while the estimator (4.3.14) is used for u. The reliability of the estimator is verified numerically in Section 4.4.1, as well as the time step independence property. Further theoretical issues of the residual estimator is discussed in Section 4.4.2. Recall that for elliptic and parabolic problems, the normal jump r_K tends to dominate the residual [26], and therefore the size of the residual is less crucial.

4.4 Numerical results

4.4.1 Test case 1

Take the domain $\Omega = (0, 100) \times (0, 100)$, $T = 350$. Solve with FitzHugh-Nagumo model with $\epsilon = 0.01$, $\kappa = 0.16875$, $a = 0.25$, and initial condition

$$u_0 = 0.5 - \frac{1}{\pi} \arctan\left(\sqrt{x^2 + y^2} - 200\right), \qquad w_0 = 0.$$

At time $t = 0$ the wave is activated in the lower left corner of the domain, and a circular wave action potential results moving away from the origin.

time-step dependence of estimator

Before applying the adaptive algorithm, we compare the residual and non-residual estimators introduced in Section 4.3 on a fixed uniform mesh, with $h = 5$ (1600 elements), and with a varying time step. Given $s \in [0, \tau]$, we approximate the estimator at different points of the interval in the following way:

$$res_W(s) = \tau^{1/2} \left(\sum_n \sum_K \| R^W_{K,n}(t_{n-1} + s) \|_{0,K} \omega_K(\tilde{e}_w(t_{n-1} + s)) \right)^{1/2},$$

and define $res_U(s)$ similarly. For comparison, we also consider $res_W^{BDF1}(s)$ where the time derivative in the residual is replaced by the time derivative corresponding to the piecewise linear reconstruction. From Table 4.1 we observe only minimal variation for different values s and time step τ in the estimator for u. However, for w, the definition appears to be crucial. The residual at $s = \tau$ changes very little, while the values for $s < \tau$ decrease to the value at $s = \tau$ as τ is decreased. An explanation for this trend is given in Section 4.4.2, the essential idea being a difference between parabolic PDEs and ODEs. At the same time, the estimator $\omega^{S,W}$ does not change with the time step, and as will be seen in what follows, is quite close to the actual error, as observed in Section 4.3.3.

τ	$res_U^{BDF1}(\tau)$	$res_U(0)$	$res_U(\tau/2)$	$res_U(\tau)$	
2	45.2	46.5	46.5	46.5	
1	46.1	46.7	46.8	46.8	
0.5	46.5	46.8	46.8	46.9	
0.25	46.7	46.9	46.6	46.9	
0.125	46.8	46.9	46.9	46.9	
τ	$res_W^{BDF1}(\tau)$	$res_W(0)$	$res_W(\tau/2)$	$res_W(\tau)$	$\omega^{S,W}$
2	0.184	0.0815	0.0715	0.0199	0.907
1	0.132	0.0453	0.0402	0.0192	0.910
0.5	0.0937	0.0279	0.0258	0.0189	0.911
0.25	0.0664	0.0214	0.0208	0.0189	0.911
0.125	0.0472	0.0195	0.0193	0.0189	0.911

Table 4.1: Test case 1: residual estimators for uniform mesh $h = 5$.

Adaptive algorithm

We apply mesh adaptation only. Choose positive constants TOL_S^U, TOL_S^W. The goal is to control the relative error:

$$0.25\ TOL_S^U \leq \frac{\eta_n^{S,U}}{|||u_h|||_n} \leq TOL_S^U, \tag{4.4.1}$$

$$0.25\ TOL_S^W \leq \frac{\omega_n^{S,W}}{|||w_h|||_n} \leq 0.65\ TOL_S^W. \tag{4.4.2}$$

The mesh is adapted if either of the upper bounds in (4.4.1) or (4.4.2) are violated, or if both of the lower bounds are violated. The meshes are adapted using the same method as in Chapter 3. That is, for each variable, metrics $\mathcal{M}(u)_k$ and $\mathcal{M}(w)_k$ are constructed for the time steps t_{n-2}, t_{n-1}, t_n according to the algorithm given in Section 3.3.2. Then we take the average $\mathcal{M} = \frac{1}{6}\left(\sum_{s=0}^{2}\mathcal{M}(u)_{n-s} + \sum_{s=0}^{2}\mathcal{M}(w)_{n-s}\right)$. One expects that the estimators $\eta^{S,U}$ and $\omega^{S,W}$ are respectively of order 1 and 2, so we refine the tolerances to maintain $\frac{TOL_S^U}{\sqrt{TOL_S^W}}$ to be a constant ratio. See Table 4.3 for numerical verification of this assumption for uniform meshes.

In Figure 4.2 we show an example solution and meshes using the adaptive method. The mesh elements are heavily concentrated in the depolarization and repolarization regions of the transmembrane potential, while there are few elements in the region ahead of the wave-front, where both variables are nearly constant. Additionally, the regions corresponding to the plateau and recovery are reasonably well refined, capturing the slow variation of the recovery variable. Figure 4.3 shows a zoom on the wave-front illustrating that the mesh fits both variables well.

Computation of the effectivity index

We estimate the following effectivity indices

$$ei^{S,U} = \frac{\eta^{S.U}}{|||e_u|||}, \quad ei^{S,W} = \frac{\omega^{S.W}}{|||e_w|||_\infty},$$
$$ei^{ZZ,U} = \frac{\eta^{ZZ.U}}{|||e_u|||}, \quad ei^{ZZ,W} = \frac{\eta^{ZZ.W}}{|||e_w|||},$$

where

$$\eta^{ZZ,U} = \left(\int_0^T \|\nabla u_h - ZZ(u_h)\|_{1,\Omega}^2\, \mathrm{d}t\right)^{1/2}$$

is the estimated error for the Zienkiewicz-Zhu recovered gradient (see [87]). Since we do not have an exact solution, to assess the robustness of the estimators we compute a reference solution. In Chapter 3 it was noted that computing a reference

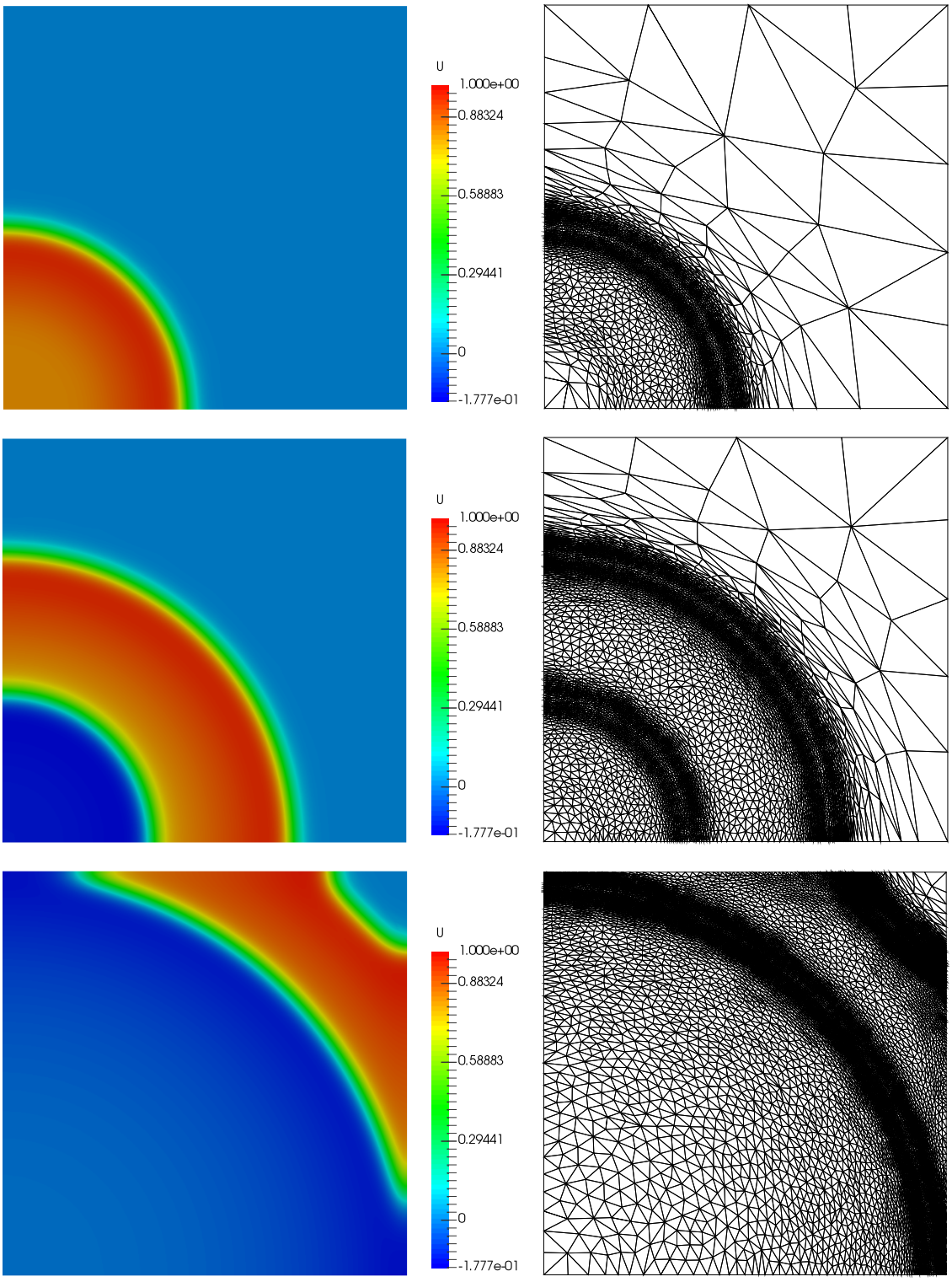


Figure 4.2: Test case 1: transmembrane potential u (left) and corresponding adapted mesh (right) when applying the adaptive method with $TOL_S^U = 0.125$, $TOL_S^W = 0.0125$ and $\tau = 0.5$ at $t = 87.5$ (top), $t = 175$ (centre) and $t = 350$ (bottom).

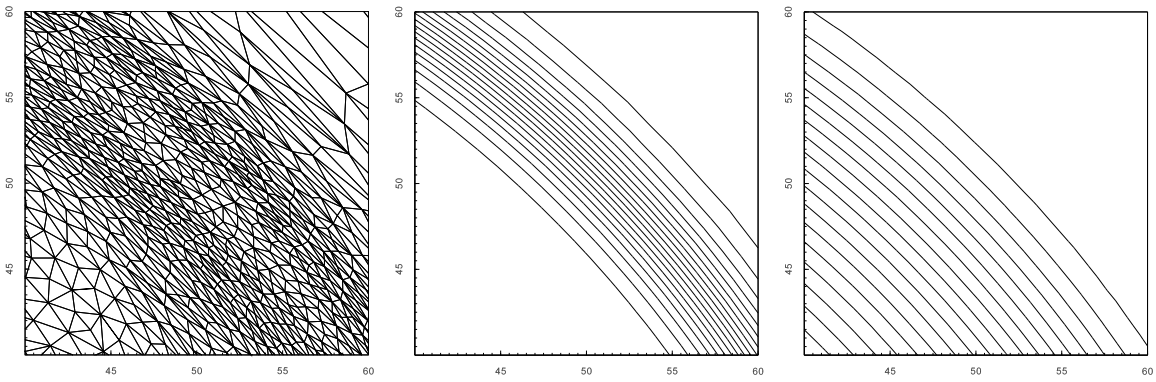


Figure 4.3: Test case 1: zoom of the mesh at wave-front at time $t = 175$ (left), and contours of u (centre) and w (right).

solution with uniform meshes would be impractical. This issue was additionally observed in [11] when solving the monodomain problem. As before, we compute a reference solution using the adapted method. The reference solution is computed with $TOL_S^U = 0.015625$, $TOL_S^W = 0.0015625$, and $\tau = 0.0625$, with meshes ranging from 110000 elements at $t = 0$ to 540000 elements when the wave hits the boundary (around $t = 260$). The accuracy of error computation is checked by comparing the results with a somewhat coarser adapted solution, with $TOL_S^U = 0.03125$, $TOL_S^W = 0.003125$, resulting in meshes with approximately 4 times fewer elements. Table 4.2 records the relative differences for some of the computations. That is, let u_h^{Ref}, denote the reference solution, and $u_h^{\text{Ref},C}$ the coarser solution. If we want to estimate the exact error for a given approximate solution u_H, then we compute $e^F = \|u_h^{\text{Ref}} - u_H\|$ and $e^C = \|u_h^{\text{Ref},C} - u_H\|$ and record the value $100\frac{|e^F - e^C|}{e^F}$ in Table 4.2. For the uniform mesh computations, the error calculations appears to be quite reliable, with the error computations differing by less than 0.1%. The adapted mesh computation with $TOL_S^U = 0.25$ and $TOL_S^W = 0.2$ also appears reliable with the computations differing by at most 0.15%, while the computations with $TOL_S^U = 0.125$ and $TOL_S^W = 0.05$ are at the limit of how far the reference solution can be trusted, and it is likely only one or two digits of the estimated error is correct. Figure 4.4 plots the solution at the point $(0.5, 0.5)$, giving qualitative evidence that the uniform mesh computations are converging to the reference solution as h and τ are decreased.

Some results for uniform meshes, computing the error with this reference solution, are summarized in Table 4.3, and for adapted meshes in Table 4.4. The effectivity indices $ei^{S,U}$ and $ei^{S,W}$ are generally close to 1. Note the effectivity index $ei^{S,U}$ tends to increase, on average as we decrease h for the uniform mesh and both indices $ei^{S,U}$ and $ei^{S,W}$ tend to increase as we decrease TOL_S^U and TOL_S^W for the adapted mesh computations. The tendency can be explained by the plots of the error in Figure 4.7 for uniform meshes, and Figure 4.8 for adapted meshes. Here, for the energy

h	τ	$\|\|\|e_u\|\|\| - diff\%$	$\|\|\|e_w\|\|\|_\infty - diff\%$	
1.25	1	0.0864082	0.0282279	
1.25	0.5	0.0682789	0.00152349	
1.25	0.25	0.063556	0.00366051	
TOL_S^U	TOL_S^W	τ	$\|\|\|e_u\|\|\| - diff\%$	$\|\|\|e_w\|\|\|_\infty - diff\%$
0.25	0.2	1	0.0343926	0.0509442
0.25	0.2	0.5	0.0451661	0.105481
0.25	0.2	0.25	0.0759341	0.153237
0.125	0.05	1	0.284668	0.11999
0.125	0.05	0.5	0.391476	1.22847
0.125	0.05	0.25	0.448963	0.579438

Table 4.2: Test case 1: relative difference (in %) when computing the error using the reference solution and when using a coarser adapted solution, by a factor of 4.

norm and error estimators, we plot at time t_n the sum of the error up to time t_n. That is, for the estimator η, we plot $(\sum_{k=1}^{n} \tau \eta_k^2)^{1/2}$. For the L^2 error computation we have just plotted the L^2 error at t_n. We see that the increase in $ei^{S,U}$ just reflects the fact that the estimator more closely captures the growth of the error as the tolerance decreases. Moreover, we can see that for both u and w, the ZZ estimator very accurately predicts the energy norm of the error, and suggests that the effectivity indices $ei^{ZZ,U/W}$ approach 1. For the L^2 error, we note that there is a slight tendency of the estimator $\omega^{S,W}$ to overestimate the error for the adapted mesh computations. While Table 4.4 suggests that this overestimation becomes worse as the tolerance is decreased, Figure 4.8 suggests that the overall rate of growth is closer to that of the exact error for smaller tolerance.

Efficiency of the adaptive method

From Tables 4.3 and 4.4 we can get some idea of the efficiency of the adaptive method. First we observe, however, that for the uniform mesh computations, the smallest time step does not generally result in the lowest error. For fixed h, the error decreases as τ is decreased to an optimal value τ_{opt}, and then the error increases as the time step is decreased further. From Table 4.3, we infer $\tau_{opt} = 8$ for $h = 5$, $\tau_{opt} = 5.656$ for $h = 2.5$ and $\tau_{opt} = 2$ for $h = 1.25$. Also, in Table 4.4, the lowest error for $TOL_S^U = 0.5, TOL_S^W = 0.2$ occurs with the step $\tau = 4$. Recall that we are solving a traveling wave solution, and for large time step and coarse mesh the speed of the wave is generally wrong. The wave front for the approximate solution gets ahead of or lags behind the actual wave front, and the effect of this displacement is that the exact error varies significantly depending on the time step. Therefore, we content ourselves with only comparing the most accurate solutions, for which the convergence of the approximate solution should be closer to the asymptotic rate. The lowest error for the

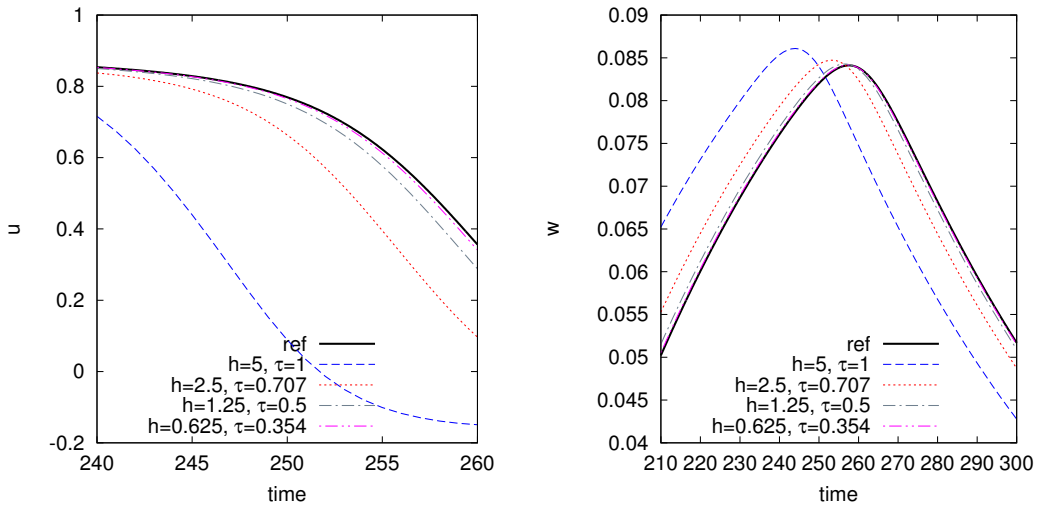


Figure 4.4: Test case 1: comparison of reference solution with solutions on uniform meshes, plotted at point $(x, y) = (0.5, 0.5)$ of u (left) and w (right).

uniform mesh occurs with $h = 1.25$ and $\tau = 2$. With $TOL_S^U = 0.125, TOL_S^W = 0.05$ and $\tau = 1$, the adapted method achieves an error which is 14% lower for u in 33% less CPU time and with 40% fewer elements. The error for w is 18% higher for the the adapted mesh solution. However, looking at the surrounding values in Table 4.3, the low value of the error is not typical, and appears to occur almost by accident. The trend for the for surrounding values in Table 4.4 seems more regular, decreasing as the time step is decreased. In Figures 4.5 and 4.6 we plot the error for CPU time and number of elements. For the uniform mesh computation, we used the same τ_{opt} noted earlier, while for the adapted mesh, we chose τ at $4, 2$ and 0.5 respectively for TOL_S^U at $0.5, 0.25$ and 0.125. We see the trend is we can expect a decrease in error by about 50% for a given level of CPU time and by about $50 - 80\%$ for a given number of elements, indicating savings in memory.

4.4.2 Test case 2

Take the same domain and initial condition as test case 1 with $T = 500$, but with the smoothed Mitchell-Schaeffer model with smoothing parameter $\kappa = 100$. The parameters are taken from [86]: $\tau_{in} = 0.315$, $\tau_{out} = 5.556$, $\tau_{open} = 94.942$, $\tau_{close} = 168.5$, $u_{gate} = 0.13$, diffusion coefficient is 3.949. The solution is a more realistic representation of a heartbeat in terms of the duration of the phases of the action potential. The action potential is activated in the lower left corner of the domain and the wave radiates away from the origin until about time $t = 60$ when the entire domain is depolarized. A long plateau follows in which the gate w closes and the inward and

h	τ	$\|\|\|e_u\|\|\|$	$\|\|\|e_w\|\|\|_\infty$	$\eta^{S,U}$	$\omega^{S,W}$	$ei^{S,U}$	$ei^{S,W}$	CPU(s)	N_T
5	8	77.4282	1.3315	40.5122	0.8459	0.5232	0.6353	26	1600
5	5.656	77.6913	1.2806	42.1245	0.8777	0.5422	0.6853	34	1600
5	4	80.9626	1.3696	43.4789	0.8953	0.5370	0.6537	47	1600
5	2.828	83.7481	1.4722	44.4672	0.9033	0.5310	0.6136	60	1600
5	2	85.5144	1.5480	45.1927	0.9072	0.5285	0.5860	80	1600
5	1.414	86.4901	1.5953	45.7251	0.9095	0.5287	0.5701	106	1600
5	1	87.0392	1.6225	46.0677	0.9101	0.5293	0.5609	149	1600
5	0.707	87.3052	1.6374	46.3265	0.9108	0.5306	0.5563	196	1600
5	0.5	87.4675	1.6452	46.4819	0.9107	0.5314	0.5535	278	1600
5	0.354	87.5272	1.6492	46.6012	0.9108	0.5324	0.5523	391	1600
5	0.25	87.5697	1.6513	46.6814	0.9108	0.5331	0.5516	543	1600
2.5	8	31.3545	0.5285	21.3164	0.2255	0.6799	0.4268	121	6400
2.5	5.656	22.0597	0.2759	22.4235	0.2354	1.0165	0.8530	156	6400
2.5	4	23.2969	0.2402	23.3803	0.2412	1.0036	1.0043	210	6400
2.5	2.828	30.9476	0.3398	24.0348	0.2438	0.7766	0.7177	283	6400
2.5	2	37.8576	0.4279	24.4838	0.2452	0.6467	0.5730	377	6400
2.5	1.414	42.2648	0.4857	24.7991	0.2460	0.5868	0.5065	492	6400
2.5	1	44.7831	0.5194	24.9872	0.2462	0.5580	0.4741	612	6400
2.5	0.707	46.1200	0.5379	25.1309	0.2465	0.5449	0.4583	866	6400
2.5	0.5	46.8454	0.5477	25.2099	0.2465	0.5382	0.4500	1163	6400
2.5	0.354	47.1964	0.5528	25.2733	0.2465	0.5355	0.4459	1681	6400
2.5	0.25	47.3887	0.5555	25.3147	0.2465	0.5342	0.4438	2438	6400
1.25	4	20.4654	0.2138	11.6837	0.0607	0.5709	0.2841	761	25600
1.25	2.828	11.0160	0.0966	12.0312	0.0614	1.0922	0.6355	1023	25600
1.25	2	6.9944	0.0384	12.2691	0.0617	1.7541	1.6074	1636	25600
1.25	1.414	9.3211	0.0784	12.4358	0.0619	1.3342	0.7900	2205	25600
1.25	1	11.8854	0.1118	12.5336	0.0620	1.0545	0.5543	3056	25600
1.25	0.707	13.4654	0.1308	12.6086	0.0620	0.9364	0.4744	3629	25600
1.25	0.5	14.3438	0.1409	12.6485	0.0620	0.8818	0.4400	5015	25600
1.25	0.354	14.7991	0.1462	12.6808	0.0620	0.8569	0.4241	7732	25600
1.25	0.25	15.0408	0.1490	12.7017	0.0620	0.8445	0.4162	9444	25600

Table 4.3: Test case 1: results for uniform mesh computations.

outward currents are roughly balanced. At about $t = 290$ the outward current begins to dominate and the region is completely repolarized by about $t = 360$. As u drops below u_{gate}, the gate slowly opens. See Figure 4.9 for the solution at different phases, and Figure 4.10 for the complete action potential.

On the residual for the recovery variable

For the monodomain problem, the variables u and w do not decouple, so the arguments relating to problem (4.3.20) do not directly apply. However, the equation for w in the Mitchell-Schaeffer model switches between two equations of this form: with $\mu = \tau_{open}^{-1}$, $f = \tau_{open}^{-1}$ for $u < u_{gate}$ and $\mu = \tau_{close}^{-1}$, $f = 0$ for $u \geq u_{gate}$. During the action potential, this switching occurs when u varies quickly during the depolarization and

TOL_S^U, TOL_S^W	τ	$\|\|\|e_u\|\|\|$	$\|\|\|e_w\|\|\|_\infty$	$\eta^{S,U}$	$\omega^{S,W}$	$ei^{S,U}$	$ei^{S,W}$	CPU(s)	$maxN_T$
0.5, 0.2	8	28.3658	0.4947	19.2791	0.5957	0.6797	1.2041	43	2104
0.5, 0.2	5.656	21.3598	0.2549	19.1085	0.5348	0.8946	2.0984	47	2431
0.5, 0.2	4	16.8330	0.1470	18.9243	0.4938	1.1242	3.3596	61	2470
0.5, 0.2	2.828	18.4506	0.1769	18.9116	0.4519	1.0250	2.5545	75	2718
0.5, 0.2	2	24.2087	0.2791	18.6489	0.4342	0.7703	1.5560	90	2751
0.5, 0.2	1.414	24.8318	0.2714	18.2091	0.4160	0.7333	1.5329	116	2843
0.5, 0.2	1	31.2314	0.3429	18.1779	0.4128	0.5820	1.2039	134	2708
0.5, 0.2	0.5	31.0808	0.3378	18.2355	0.4072	0.5867	1.2056	256	2925
0.5, 0.2	0.25	31.4417	0.3406	17.9535	0.3891	0.5710	1.1424	490	3005
0.25, 0.2	8	29.7775	0.4942	15.7462	0.3773	0.5288	0.7634	78	3747
0.25, 0.2	5.656	30.8307	0.3408	15.4878	0.3074	0.5023	0.9018	90	4466
0.25, 0.2	4	24.6326	0.2605	14.7702	0.2764	0.5996	1.0610	107	4251
0.25, 0.2	2.828	16.1720	0.1468	14.2122	0.2571	0.8788	1.7516	122	4325
0.25, 0.2	2	10.3628	0.0574	13.7931	0.2470	1.3310	4.3022	147	4165
0.25, 0.2	1.414	10.2044	0.0683	13.7948	0.2388	1.3518	3.4965	196	4324
0.25, 0.2	1	11.0217	0.0808	13.4485	0.2269	1.2202	2.8071	268	4409
0.25, 0.2	0.5	12.6683	0.1147	13.1701	0.2210	1.0396	1.9271	388	4432
0.25, 0.2	0.25	13.2568	0.1347	13.0424	0.2107	0.9838	1.5636	745	4706
0.125, 0.05	4	29.7520	0.3389	8.1244	0.0734	0.2731	0.2165	498	15003
0.125, 0.05	2.828	20.9759	0.2267	7.6320	0.0645	0.3638	0.2845	570	14433
0.125, 0.05	2	13.4546	0.1382	7.3431	0.0596	0.5458	0.4313	670	14906
0.125, 0.05	1.414	8.3719	0.0765	7.0998	0.0579	0.8480	0.7566	866	14894
0.125, 0.05	1	5.9626	0.0455	6.8849	0.0556	1.1547	1.2217	1086	15109
0.125, 0.05	0.5	4.6691	0.0184	6.6940	0.0537	1.4337	2.9192	1709	15768
0.125, 0.05	0.25	4.6680	0.0175	6.5266	0.0528	1.3982	3.0196	3237	15441

Table 4.4: Test case 1: results for adapted mesh computations.

repolarization phases. These phases occupy a relatively short duration. During the plateau and recovery phases, w varies almost independently of u. From Figure 4.11 we observe that the element residual $R^W_{K,n}$ is only significant when $u \approx u_{gate}$, which occurs on the wave-front, and as a result, the estimator $\eta^{S,W}$ is close to 0 elsewhere. On the other hand, the estimator $\omega^{S,W}$ detects a significant error contribution when $(x^2 + y^2)^{1/2} \approx 20$. This can be understood from Figure 4.9, top row left and centre. One expects a relatively large contribution to the interpolation error, as w transitions from a constant value. In this region the interpolation error is not seen by the residual estimator. This is in agreement with the discussion in Section 4.3.3.

Space-time adaptation

We plot the estimators in time in Figure 4.12. In particular, we can see the effect of the phases of the action potential on the error. The estimators for u are only large during the polarization phases, while the estimators for w generally decrease outside the polarization phases, though at a slower rate. Moreover, the time error for both variables is low during the plateau and recovery phases. Since these phases

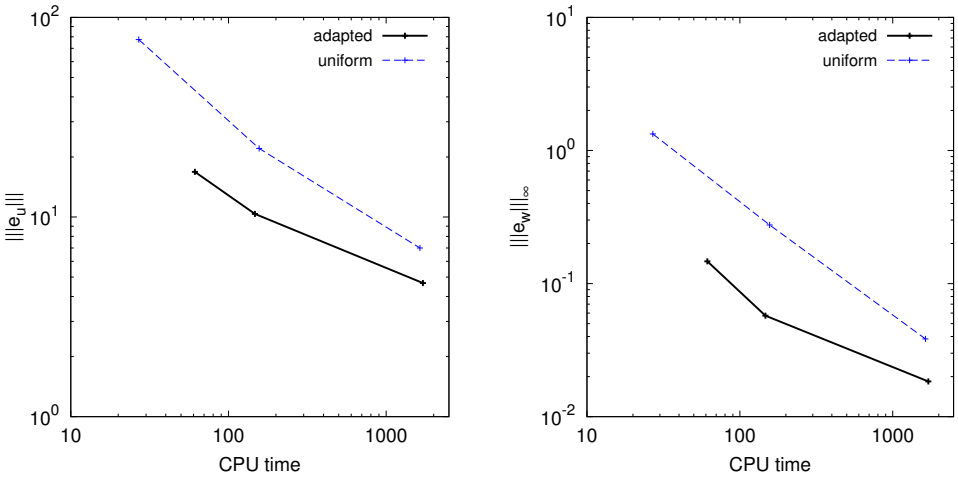


Figure 4.5: Test case 1: error vs. CPU time for uniform and adapted mesh computations.

comprise the majority of the time duration, we suspect a possible gain in efficiency when applying an adaptive time step.

The method for the time-step adaptation is similar to that used in Chapter 3. Choose positive constants TOL_T^U, TOL_T^W. The time step is decreased by a factor of 2/3 if

$$\frac{\eta_n^{T,U}}{|||u_h|||_n} > 1.5TOL_T^U, \quad \text{or} \quad \frac{\eta_n^{T,W}}{|||w_h|||_n} > 1.5TOL_T^W, \tag{4.4.3}$$

and the time step is increased by a factor of 3/2 if

$$\frac{\eta_n^{T,U}}{|||u_h|||_n} < 0.5TOL_T^U, \quad \text{and} \quad \frac{\eta_n^{T,W}}{|||w_h|||_n} < 0.5TOL_T^W. \tag{4.4.4}$$

We perform a complete space-time adaptive solution with the following parameters: $TOL_S^U = 0.0625$, $TOL_S^W = 0.0625$, $TOL_T^U = 0.035875$ and $TOL_T^W = 0.0075$. From Figure 4.13 we see a variation in the time step of two order of magnitude, with the smallest steps ranging from 0.05 to 0.1 in the depolarization and repolarization phases, and the largest steps ranging from about 1 to 4 during the plateau and recovery phases. The control of the error in space is illustrated in Figure 4.14, and we see that most of the adaptation occurs during the depolarization and repolarization phases. From the subplot on the plot on the intervals $[0, 60]$ and $[290, 360]$, note that the decision to adapt during the depolarization phase is given by condition (4.4.1), u being the fast variable, while it is given by (4.4.2) during the repolarization phase. Examples of adapted meshes for these phases are shown in Figure 4.15. The mesh

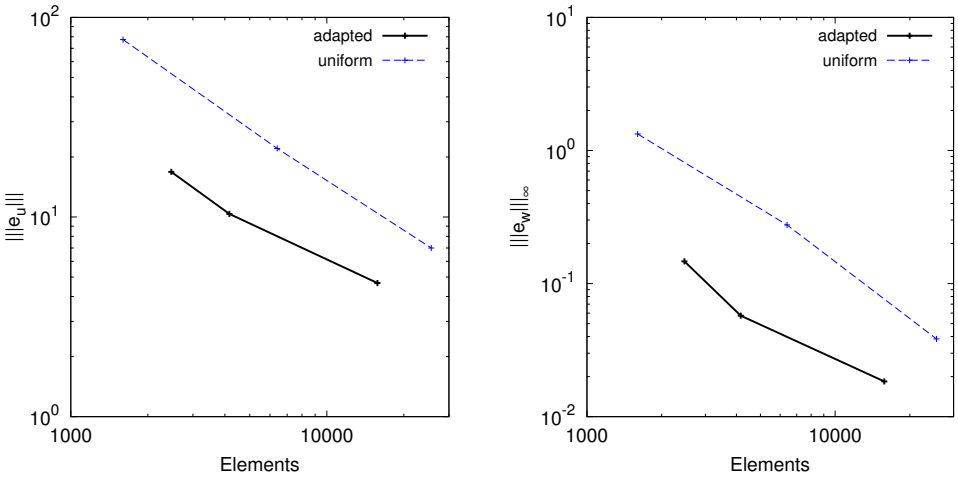


Figure 4.6: Test case 1: Test case 1: error vs. number of elements for uniform and adapted mesh computations.

for the depolarization phase is quite similar to those for the FitzHugh-Nagumo model in Figure 4.2, and with a small layer of refinement near $\sqrt{x^2+y^2} = 20$, fitting the observation made for Figure 4.11. We remark that if the mesh were to be adapted only to the variable u, as is done for instance in [91], the the mesh would only be refined in the wave front, and the slow variation in w would be not be captured properly. This in turn would eventually spoil the quality of approximation for the subsequent phases. The mesh for the repolarization phase is generally more diffuse, with the refinement of the action potential downstroke requiring fewer elements than the wave front, resulting in a mesh with 7000 elements. This is likely due to the slow variation of w.

As a final note, the efficiency of applying the space-time adaptation algorithm for this example requires a thorough study. This could for instance be done by computing a reference solution as is done for Test case 4.4.1 and in Chapter 3, and therefore approximating the exact error. However, due to the scale of this problem, a different approach may be required.

4.5 Conclusion

In this chapter we introduced an anisotropic residual error estimator for the monodomain equation, discretized with P_1 finite elements in space, and the variable step BDF2 method in time. We proved that the estimator gives an upper bound for the energy norm of the error for the first variable, and the $L^\infty(0,T;L^2(\Omega))$ norm of the error for the second variable. It was found that the estimator for the second variable could

not be used in practice for mesh adaptation purposes, due to a fundamental difference in behaviour between parabolic PDEs and ODEs. Instead, a simplified estimator was proposed for the ODE variable, which is based on interpolation estimates combined with a gradient recovery operator. Numerical computations are carried out, confirming the reliability of the estimator, and demonstrating improved efficiency gained from the error estimator guided mesh adaptation method. A space-time adaptation method is applied to a problem exhibiting large variation in time scales. While the results appear promising, more work is required to accurately assess the efficiency of the algorithm.

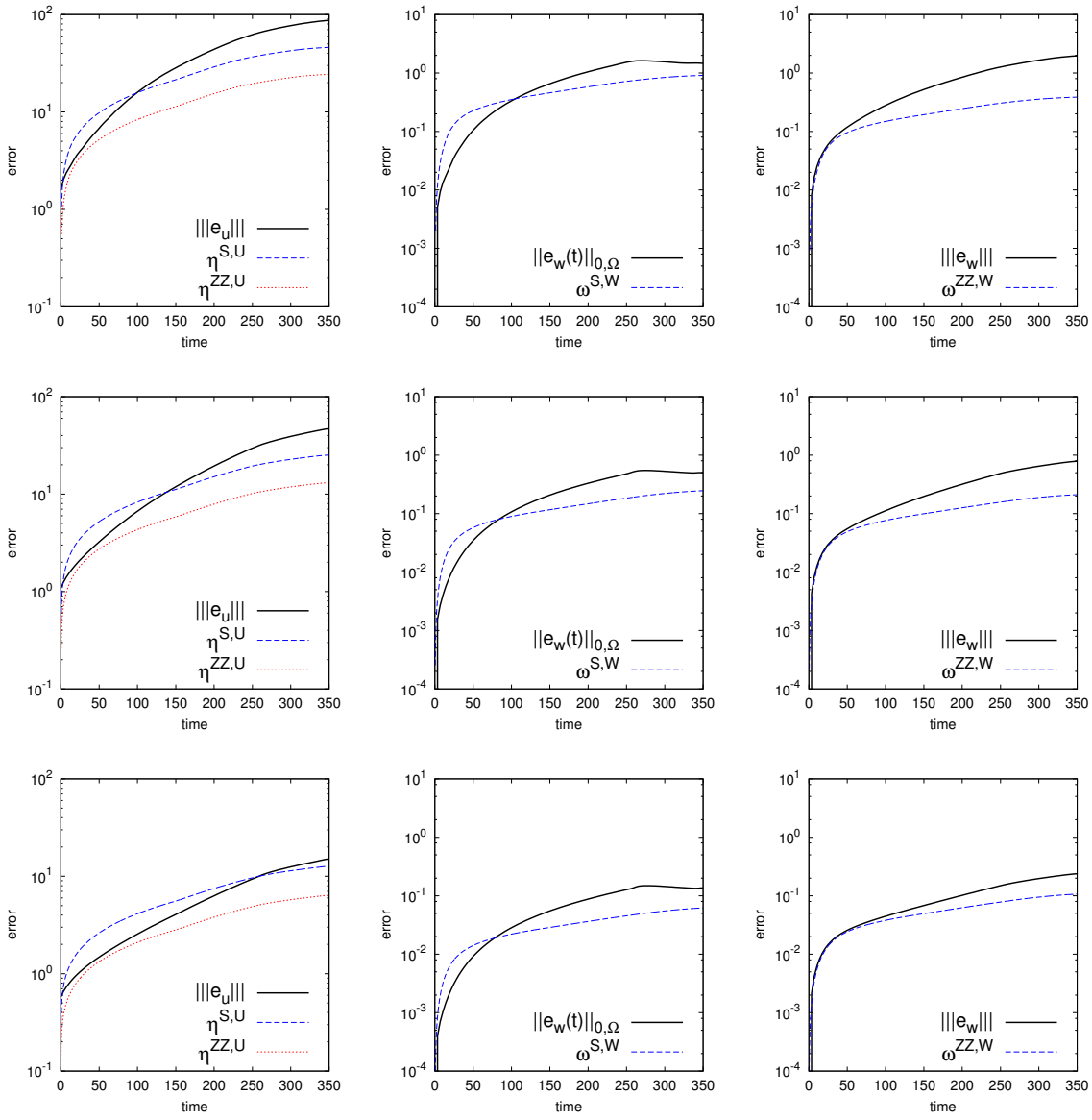


Figure 4.7: Test case 1: estimators and exact error solving with uniform meshes with $h = 5, \tau = 1$ (top), $h = 2.5, \tau = 0.5$ (centre), $h = 1.25, \tau = 0.25$ (bottom).

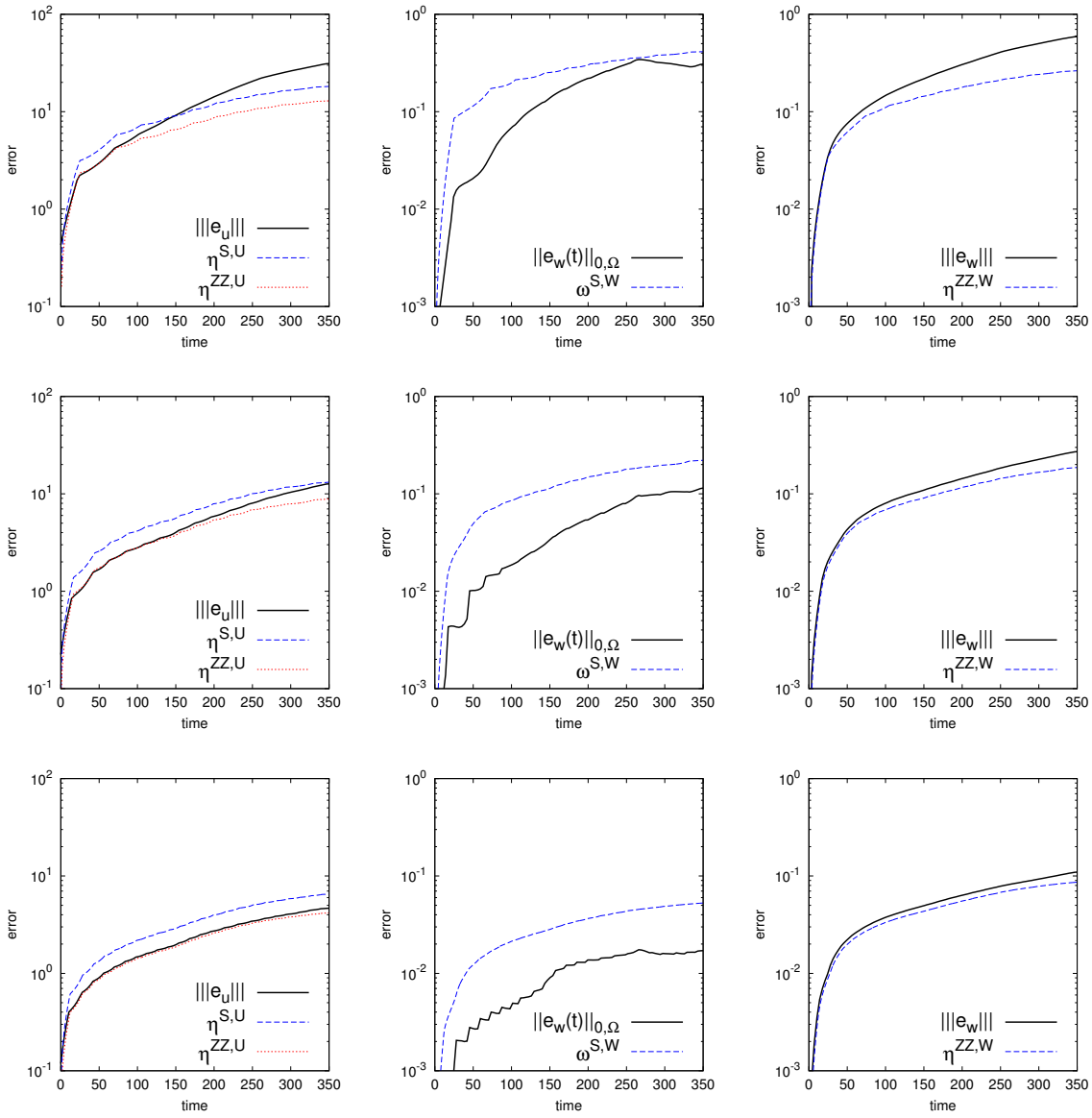


Figure 4.8: Test case 1: estimators and exact error when applying mesh adaptation with $TOL_{S,U} = 0.5, TOL_{S,W} = 0.2, \tau = 1$ (top), $TOL_{S,U} = 0.25, TOL_{S,W} = 0.2, \tau = 0.5$ (centre), $TOL_{S,U} = 0.125, TOL_{S,W} = 0.05, \tau = 0.25$ (bottom).

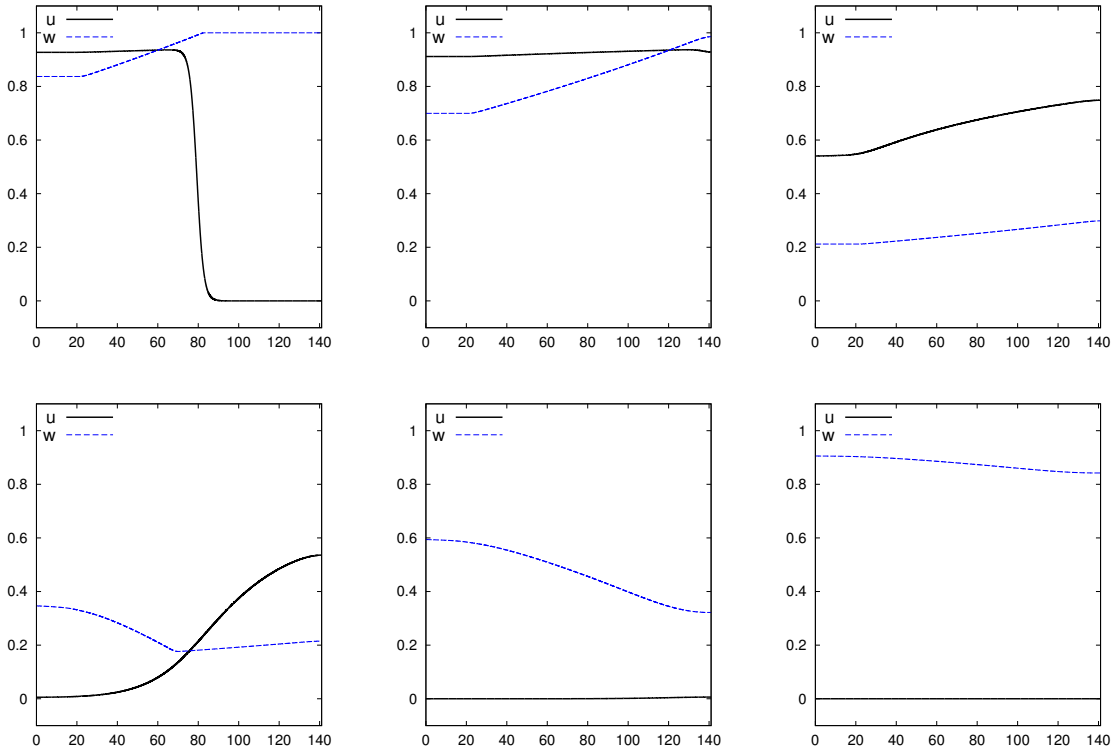


Figure 4.9: Test case 2: plot of u, w over the line $x = y$ as a function of arc-length at different points in time. During the depolarization at $t = 30$ (top left), start of the plateau at $t = 60$ (top centre), start of the repolarization at $t = 254$ (top right), part way through the repolarization at $t = 316$ (bottom left), beginning of the recovery phase at $t = 360$ (bottom centre), and end of the simulation at $t = 500$ (bottom right).

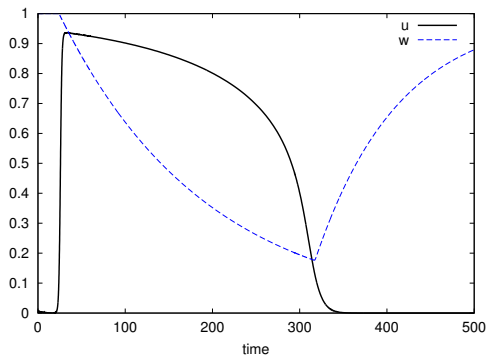


Figure 4.10: Test case 2: plot in time of the solution at the point $(50, 50)$.

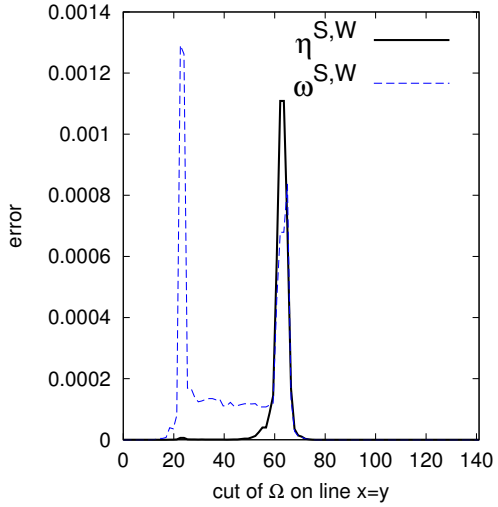


Figure 4.11: Test case 2: plot of the estimators for w on the line $x = y$ at time $t = 20$. Solution computed on a uniform mesh with 6400 elements and a constant time step 0.25.

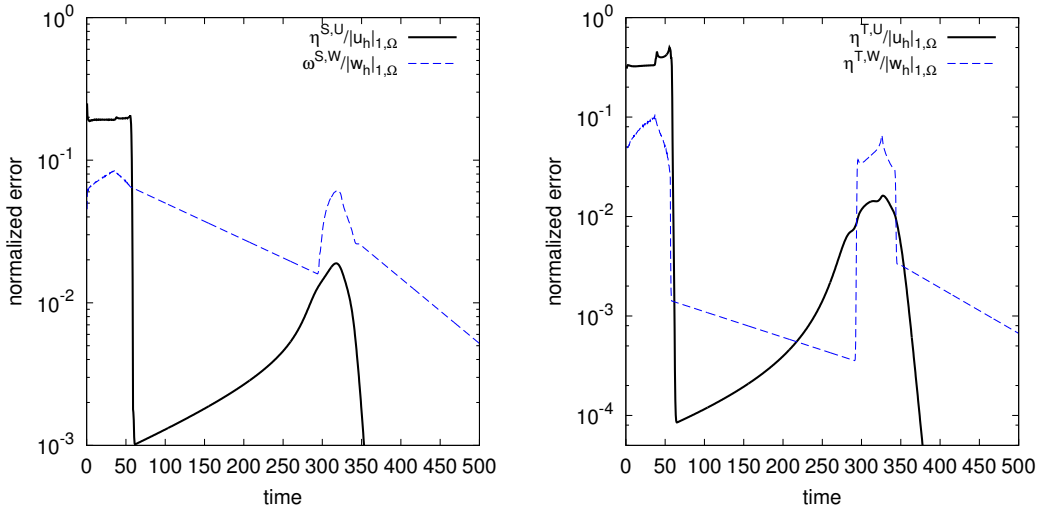


Figure 4.12: Test case 2: evolution of the normalized space error (left) and time error (right) computed on a uniform mesh with $h = 2.5$ and step $\tau = 0.5$.

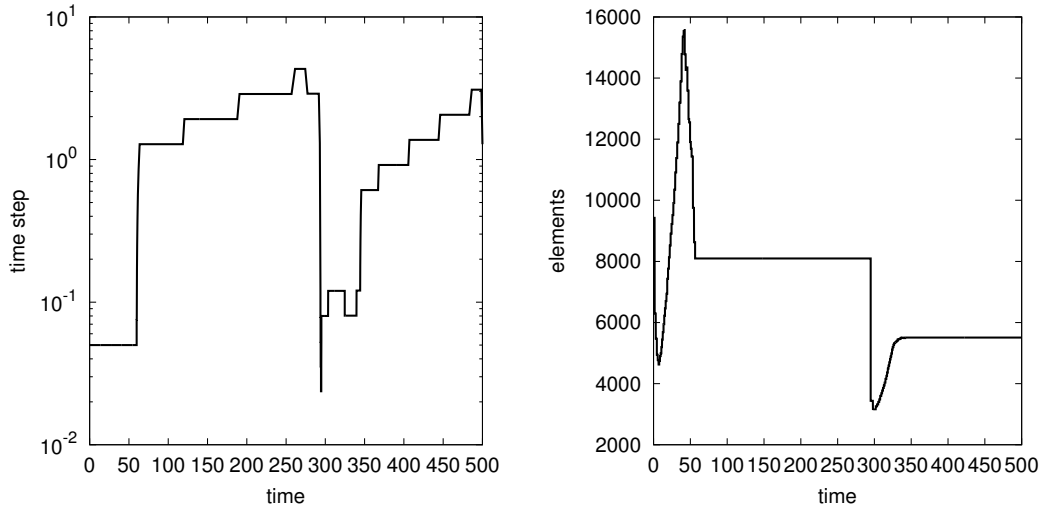


Figure 4.13: Test case 2: evolution of the time step (left) and the number of elements (right).

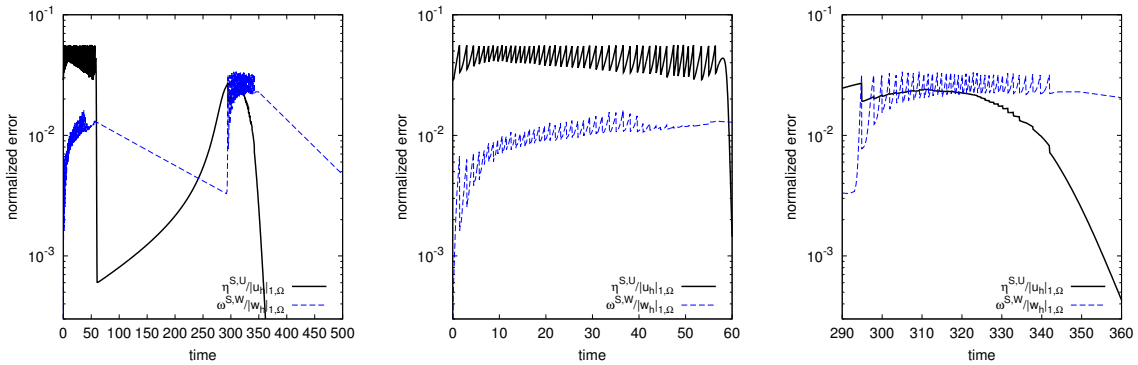


Figure 4.14: Test case 2: evolution of the space error for space-time adapted solution.

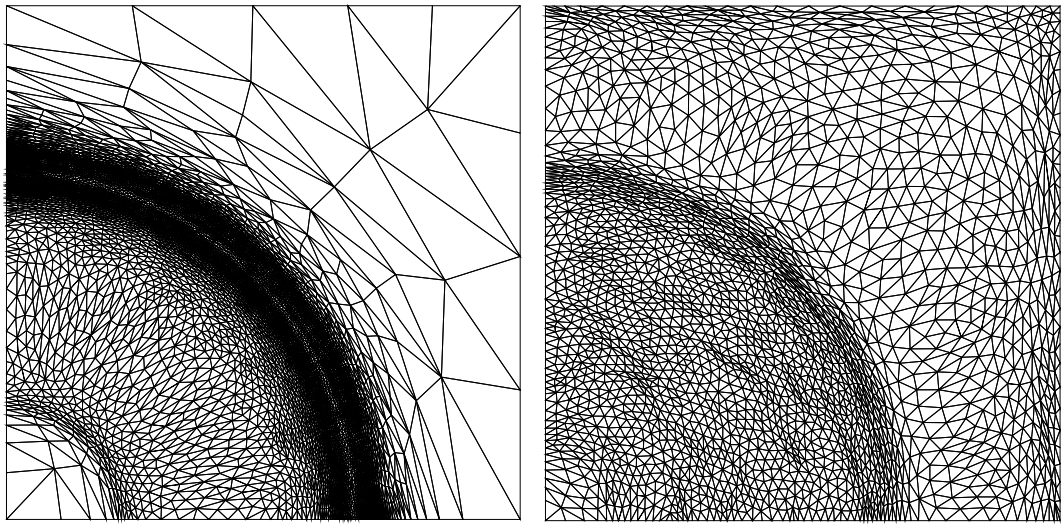


Figure 4.15: Test case 2: adapted mesh at $t = 30$ (left) during the depolarization phase and at $t = 316$ (right) during the repolarization phase.

Chapter 5

Conclusions

In this thesis we considered adaptive methods for solving elliptic and parabolic equations with the use of explicit anisotropic, residual-based a posteriori error estimators. We focussed on algorithmic aspects of the implementation of a new element-based method and the implementation of a space-time adaptation algorithm, as well as the introduction of such techniques to the field of cardiac electrophysiology.

5.1 Contributions

5.1.1 Element-based adaptation

An element-based adaptation method was implemented for an anisotropic residual error estimator. As far as we know, we are the first propose and investigate local criteria to drive local mesh modifications for this estimator. The usual approach taken in the literature is to convert the estimator into a metric and apply an averaging technique. The element-based approach allows direct control of the estimator, instead of indirectly through the recovered metric. The method is shown to converge in the sense that the total number of local operations decreases to a minimal value, and it is shown that the resulting error is reasonably well equidistibuted over the elements of the mesh. Compared to other anisotropic methods, it was found to result in marginally lower error in the H^1-seminorm and similar error in the L^2-norm for a given number of degrees of freedom. However, in terms of CPU time efficiency, the current method is still far off from the existing methods, particularly the metric based methods.

5.1.2 An L^2-norm error estimator

A new error estimator for the L^2-norm of the error for elliptic problems was introduced in Chapter 2. The estimator makes use of an existing anisotropic residual estimator

for the H^1-seminorm and exploits a mesh-dependent relation between the error in the respective norms. The estimator is shown to be equivalent to the exact L^2-norm of the error under reasonable mesh-dependent assumptions. This equivalence is verified numerically. An element-based mesh adaptation method was employed to control the L^2-norm, by combining it with the estimator for the H^1-seminorm, similar to the method used for a hierarchical estimator in [17]. As expected, the method is found to give better control of the L^2-norm than with the estimator for the H^1-seminorm.

Future work could involve the use of the estimator for metric adaptation, thereby exploiting the efficiency of such methods.

5.1.3 Space-time adaptation for reaction-diffusion systems

An a posteriori error estimator was developed for scalar reaction-diffusion problems as well as the monodomain problem, which consists of a scalar parabolic problem coupled with an ODE. The estimator consists of an anisotropic residual estimator for the space discretization, as well as an estimator for the BDF2 time discretization. As far as we know this is the first time the estimator for the BDF2 discretization from [2] has been extended to a PDE problem. We believe this is the first anisotropic residual estimator that has been developed for the monodomain problem. For both the scalar and monodomain problem, it was shown that the error estimator estimator gives an upper bound for the error. The proof of the upper bound relies on a convergence hypothesis, modified from one originally used in [24], and we have provided partial results on its validity. In the case of the ODE variable for the monodomain model, we also considered a simplified estimator that does not use the residual. Numerically, we verify that the estimated error is equivalent to the exact error for the scalar problem, while for the monodomain problem, the equivalence of the error is verified for the PDE variable, as well as the ODE variable with the modified non-residual estimator. A space-time adaptation algorithm is successfully applied for both models, and in the case of the scalar equation, the method is found to be at least as efficient in terms of CPU time compared to a mesh adaptation method computed with a nearly optimal constant time step, and more efficient than the standard uniform method. Moreover, the use of the time-step adaptation prevents one from having to guess an optimal time step. We applied the space-time adaptation algorithm to a physiologically relevant example, and preliminary computations suggests that there could be a potential gain in efficiency over long time scales.

5.2 Future work

5.2.1 Improvements to the element-based method

In the future, it would be desirable to apply the element-based adaptation method to more complex problems, such as time-dependent problems. However, the practicality of such an application is still far off, as the current method performs poorly in terms of CPU time compared to the existing methods. A good deal of work needs to go into optimizing the method. The primary difficulty stems from recomputing the estimator after performing mesh modification. The discontinuous fields that go into the a posteriori estimator, which include the singular valued decomposition, the gradient, and the normal jump of the derivative, need to be recomputed from the reinterpolated solution. Methods to either simplify or avoid such computations should be considered. A secondary issue is the non-local nature of the estimator, namely that the error on an element depends on a contribution from adjacent elements in the normal jump. We have to consider patches larger than what would be required to perform just the local operation, suggesting that changes to the local data structures should be considered. Finally, the computation of the residual can be expensive, and the possibility of removing this term from the computation needs further investigation.

5.2.2 Efficiency of the space-time adaptive method

Further work is required to assess the efficiency of the space-time adaptation method. For the scalar equation, there was no noticeable increase in CPU time efficiency compared to computation with space adaptation only. It is likely that the problems considered for the scalar equation did not exhibit significant significantly complex multiscale temporal behaviour to benefit from the use of a space-time adaptive procedure. On the other hand, a proper assessment of the efficiency for long time scale examples, such as the Mitchell-Schaeffer test case used in this thesis, would require a considerable amount of CPU resources. Time could be reduced, for instance, by applying higher-order methods in space and domain decomposition methods when computing the reference solution. Additionally, for the monodomain, the efficiency of the mesh-only adaptation needs further investigation.

5.2.3 Theoretical and practical considerations for the time estimator

The control of the interpolation error between adapted meshes proved to be crucial. The Lagrange interpolation operator is clearly not accurate enough to apply the full time estimator for small time steps, and a modification is required to continue. More accurate interpolation operators, which are specifically designed to interpolate finite

element functions between different meshes, should be considered in the space-time adaptation algorithm. Furthermore, in the context of the monodomain problem, the effect of the interpolation error on the quality of the residual error estimator for the ODE needs to be investigated.

5.2.4 Theoretical issues with the recovery variable

In practice, we were not able to use the residual estimator for the recovery variable of the monodomain model. Part of the issue is the dominance of the interpolation error, which appears to be an issue related to ODE problems that lack a spatial Laplacian operator. The residual seems to primarily capture variation in the PDE variable. To use the estimator, we were required to "remove" the residual as a direct multiplier. At the same time, the apparent reliability of the modified estimator also needs further investigation.

5.2.5 Extensions to more complex settings

There are many directions in which this research may be extended. In this thesis, we only considered very simple geometries and action potentials with circular wave fronts. A first step would be to apply the adaptive method to more realistic curved 2D and 3D geometries, as well as simulations of pathological conditions such as spiral and scroll waves for fibrillation dynamics. It should be reasonably straightforward to extend the theoretical results to the more complex bidomain problem, which is considered a more physically realistic model. Additionally, in this thesis we only considered simple two-variable ionic models. A challenging problem, although still far off, would be to extend the results to more complex models, for instance the ten Tuscher-Noble-Noble-Panfilov model [94], with twelve gating variables and four ionic concentrations. The primary difficulty of implementing an adaptive strategy for the model appears to be identifying which variables are essential, and which can be ignored or lumped. Finally, the adaptive framework should be considered with different discretizations. In time, it should be straightforward to consider a semi-implicit BDF2 method, which avoids the need to solve with Newton's method, and may prove more efficient. In space, the extension to P_2 finite elements should be considered by employing appropriate anisotropic interpolation estimates. In cardiac electrophysiology, it has already been observed [10], that P_2 finite elements lead to improved efficiency over P_1.

Bibliography

[1] M. Ainsworth and J. Oden. A posteriori error estimation in finite element analysis. *Comput. Methods Appl. Mech. Engrg.*, 142:1–88, 1997.

[2] G. Akrivis and P. Chatzipantelidis. A posteriori error estimates for the two-step backward differentiation formula method for parabolic equations. *SIAM J. Numer. Anal.*, 48(1):109–132, 2010.

[3] C. J. Arthurs, M. J. Bishop, and D. Kay. Efficient simulation of cardiac electrical propagation using high order finite elements. *Journal of computational physics*, 231(10):3946–3962, 2012.

[4] C. J. Arthurs, M. J. Bishop, and D. Kay. Efficient simulation of cardiac electrical propagation using high-order finite elements II: Adaptive p-version. *J. Comput. Phys.*, 253:443–470, 2013.

[5] I. Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM J. Numer. Anal.*, 13(2):214–226, 1976.

[6] I. Babuska and A. Miller. A feedback finite element method with a posteriori error estimation: Part 1. the finite element method and some basic properties of the a posteriori error estimator. *Comput. Methods Appl. Mech. Engrg.*, 61(1):1–40, 1987.

[7] I. Babuska and W. Rheinboldt. Error estimates for adaptive finite element method computations. *Int. J. Numer. Methods Engrg.*, 12:1597–1615, 1978.

[8] I. Babuska and W. Rheinboldt. A posteriori error estimatimates for the finite element method. *Int. J. Numer. Methods Engrg.*, 12:1597–1615, 1978.

[9] J. Becker. A second order backward difference method with variable steps for a parabolic problem. *BIT*, 38(4):644–662, 1998.

[10] Y. Belhamadia. A time-dependent adaptive remeshing for electrical waves of the heart. *IEEE Transactions on Biomedical Engineering*, 55(2):443–452, Feb 2008.

[11] Y. Belhamadia, A. Fortin, and Y. Bourgault. Towards accurate numerical method for monodomain models using a realistic heart geometry. *Mathematical Biosciences*, 220:89–101, 2009.

[12] Y. Belhamadia, A. Fortin, and Y. Bourgault. On the performance of anisotropic mesh adaptation for scroll wave turbulence dynamics in reaction-diffusion systems. *J. Comput. Appl. Math.*, 271:233–246, 2014.

[13] Y. Belhamadia, A. Fortin, and E. Chamberland. Anisotropic mesh adaptation for the solution of the stefan problem. *J. Comput. Phys.*, 194(1):233–255, 2004.

[14] M. Bieterman and I. Babuška. The finite element method for parabolic equations. II. A posteriori error estimation and adaptive approach. *Numer. Math.*, 40(3):373–406, 1982.

[15] E. Boey, Y. Bourgault, and T. Giordano. Anisotropic residual based a posteriori mesh adaptation in 2D: element based approach. *J. Appl. Comput. Math.* submitted May 9, 2016, Ms. Ref. No.: CAM-D-16-00959.

[16] R. Bois. *Adaptation de maillages anisotropes par un estimateur d'erreur hierarchique*. PhD thesis, Universite Laval, 2012.

[17] R. Bois, M. Fortin, and A. Fortin. A fully anisotropic mesh adaptation method based on a hierarchical error estimator. *Comput. Methods Appl. Mech. Engrg.*, 209-212:12–27, 2012.

[18] H. Borouchaki, P. George, F. Hecht, P. Laug, and E. Saltel. Delaunay mesh generation governed by metric specifications. Part I. Algorithms. *Finite Elem. Anal. Des.*, 25(1-2):61–83, 1997.

[19] M. Boulakia, M. A. Fernández, J.-F. Gerbeau, and N. Zemzemi. A coupled system of PDEs and ODEs arising in electrocardiograms modeling. *Applied Mathematics Research eXpress*, 2008:abn002, 2008.

[20] Y. Bourgault, Y. Coudière, and C. Pierre. Existence and uniqueness of the solution for the bidomain model used in cardiac electrophysiology. *Nonlinear Anal.: Real World Appl.*, 10:458–482, 2009.

[21] Y. Bourgault and M. Picasso. Anisotropic error estimates and space adaptivity for a semidiscrete finite element approximation of the transient transport equation. *SIAM J. Sci. Comput.*, 35(2):A1192–A1211, 2013.

[22] Y. Bourgault, M. Picasso, F. Alauzet, and A. Loseille. On the use of anisotropic a posteriori error estimates for the adaptative solution of 3D inviscid compressible flows. *Int. J. Numer. Meth. Fluids*, 59:47–74, 2009.

[23] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008.

[24] E. Burman and M. Picasso. Anisotropic, adaptive finite elements for the computation of a solutal dentrite. *Interfaces and Free Boundaries*, 5:103–127, 2003.

[25] W. Cao. Anisotropic measures of third order derivatives and the quadratic interpolation error on triangular elements. *SIAM J. Sci. Comput.*, 29(2):756–781 (electronic), 2007.

[26] C. Carstensen and R. Verfürth. Edge residuals dominate a posteriori error estimates for low order finite element methods. *SIAM J. Numer. Anal.*, 27:1571–1587, 1999.

[27] Z. Chen and G. Ji. Sharp L^1 a posteriori error analysis for nonlinear convection-diffusion problems. *Math. Comp.*, 75(253):43–71, 2006.

[28] E. M. Cherry, H. S. Greenside, and C. S. Henriquez. Efficient simulation of three-dimensional anisotropic cardiac tissue using an adaptive mesh refinement method. *Chaos*, 13(3):853–865, 2003.

[29] P. Clément. Approximation by finite element functions using local regularization. *RAIRO Analyse Numérique*, 9(R-2):77–84, 1975.

[30] P. Colli Franzone and G. Savaré. Degenerate evolution systems modeling the cardiac electric field at micro- and macroscopic level. In *Evolution equations, semigroups and functional analysis (Milano, 2000)*, volume 50 of *Progr. Nonlinear Differential Equations Appl.*, pages 49–78. Birkhäuser, Basel, 2002.

[31] Y. Coudière and C. Pierre. Stability and convergence of a finite volume method for two systems of reaction-diffusion equations in electro-cardiology. *Nonlinear analysis: real world applications*, 7(4):916–935, 2006.

[32] F. Dassi, S. Perotto, and L. Formaggia. A priori anisotropic mesh adaptation on implicitly defined surfaces. *SIAM J. Sci. Comput.*, 37(6):A2758–A2782, 2015.

[33] E. D'Azevedo. Optimal triangular mesh generation by coordinate transformation. *SIAM J. Sci. Stat. Comput.*, 12(4):755–786, 1991.

[34] E. D'Azevedo and R. Simpson. On optimal interpolation triangle incidences. *SIAM J. Sci. Stat. Comput.*, 10(6):1063–1075, 1989.

[35] P. Deuflhard, B. Erdmann, R. Roitzsch, and G. T. Lines. Adaptive finite element simulation of ventricular fibrillation dynamics. *Comput. Vis. Sci.*, 12(5):201–205, 2009.

[36] C. Dobrzynski. *Adaptation de maillage anisotrope 3D et application à l'aérothermique des bâtiments*. PhD thesis, Université Pierre et Marie Curie, 2005.

[37] S. Eckert, H. Baaser, D. Gross, and O. Scherf. A bdf2 integration method with step size control for elasto-plasticity. *Computational Mechanics*, 34(5):377–386, 2004.

[38] E. Emmrich. Stability and error of the variable two-step BDF for semilinear parabolic problems. *J. Appl. Math. & Computing*, 19(1-2):33–55, 2005.

[39] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems. IV. Nonlinear problems. *SIAM J. Numer. Anal.*, 32(6):1729–1749, 1995.

[40] A. Ern and J.-L. Guermond. *Theory and practice of finite elements*, volume 159 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2004.

[41] D. Estep, M. Larson, and R. Williams. Estimating the error of numerical solutions of systems of reaction-diffusion equations. *Mem. Amer. Math. Soc.*, 146(696):vii+109 pages, 2000.

[42] M. Ethier and Y. Bourgault. Semi-implicit time-discretization schemes for the bidomain model. *SIAM J. Numer. Anal.*, 46(5):2443–2468, 2008.

[43] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445, 1961.

[44] L. Formaggia and S. Perotto. New anisotropic a priori error estimates. *Numer. Math.*, 89:641–667, 2001.

[45] L. Formaggia and S. Perotto. Anisotropic error estimates for elliptic problems. *Numer. Math.*, 94:67–92, 2003.

[46] P. C. Franzone, P. Deuflhard, B. Erdmann, J. Lang, and L. F. Pavarino. Adaptivity in space and time for reaction-diffusion systems in electrocardiology. *SIAM J. Sci. Comput.*, 28(3):942–962, 2006.

[47] P. C. Franzone, L. Guerri, and S. Tentoni. Mathematical modeling of the excitation process in myocardial tissue: influence of fiber rotation on wavefront propagation and potential field. *Mathematical Biosciences*, 101(2):155–235, 1990.

[48] P. Frey and P. George. *Mesh Generation: Application to Finite Elements*. Wiley, Hoboken, NJ, 2nd edition, 2008.

[49] P. George. Gamanic3d, adaptive anisotropic tetrahedral mesh generator. Technical note, INRIA, Paris, 2003.

[50] GIREF. http://giref.ulaval.ca/.

[51] G. Golub and C. V. Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[52] P. Grindrod. *The theory and applications of reaction-diffusion equations: patterns and waves*. Clarendon Press, 1996.

[53] W. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M. Vallet. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent cfd. part I: General principles. *Int. J. Numer. Meth. Fluids*, 32:725–744, 2000.

[54] F. Hecht. BAMG: Bidimensional anisotropic mesh generator. http://www.ann.jussieu.fr/hecht/ftp/bamg/bamg.pdf.

[55] D. Henry. *Geometric theory of semilinear parabolic equations*, volume 840 of *Lecture notes in mathematics*. Springer-Verlag, 1981.

[56] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

[57] W. Huang, L. Kamenski, and J. Lang. A new anisotropic mesh adaptation method based upon hierarchical a posteriori error estimates. *J. Comput. Phys.*, 229(6):2179–2198, 2010.

[58] L. Kamenski and W. Huang. How a nonconvergent recovered Hessian works in mesh adaptation. *SIAM J. Numer. Anal.*, 52(4):1692–1708, 2014.

[59] J. Keener and J. Sneyd. *Mathematical physiology. Vol. I: Cellular physiology*, volume 8/ of *Interdisciplinary Applied Mathematics*. Springer, New York, 2nd edition, 2009.

[60] J. Keener and J. Sneyd. *Mathematical physiology. Vol. II: Systems physiology*, volume 8/ of *Interdisciplinary Applied Mathematics*. Springer, New York, 2nd edition, 2009.

[61] J. P. Keener. A geometrical theory for spiral waves in excitable media. *SIAM J. Appl. Math.*, 46(6):1039–1056, 1986.

[62] G. Kunert. *A posteriori error estimation for anisotropic tetrahedral and triangular finite element meshes*. PhD thesis, TU Chemnitz, 1999.

[63] G. Kunert and R. Verfürth. Edge residuals dominate a posteriori error estimates for linear finite element methods on anisotropic triangular and tetrahedral meshes. *Numer. Math.*, 86(2):283–303, 2000.

[64] J. Lang. *Adaptive multilevel solution of nonlinear parabolic PDE systems: theory, algorithm, and applications*, volume 16 of *Lect. Notes Comput. Sci. Eng.* Springer-Verlag, 2000.

[65] P. Laug and H. Bourochaki. BL2D-V2 : mailleur bidimensionnel adaptatif. Technical Report RT-0275, INRIA, Paris, 2003.

[66] G. T. Lines, P. Grøttum, and A. Tveito. Modeling the electrical activity of the heart: a bidomain model of the ventricles embedded in a torso. *Comput. Vis. Sci.*, 5(4):195–213, 2003.

[67] J. Lions. *Quelques methodes de resolution des problemes aux limites non lineaires*. Dunod, 1969.

[68] A. Loseille and F. Alauzet. Continuous mesh framework part I: Well-posed continuous interpolation error. *SIAM J. Numer. Anal.*, 49(1):38–60, 2011.

[69] A. Loseille and F. Alauzet. Continuous mesh framework part II: Validations and applications. *SIAM J. Numer. Anal.*, 49(1):61–86, 2011.

[70] A. Lozinski, M. Picasso, and V. Prachittham. An anisotropic error estimator for the Crank-Nicolson method: applications to a parabolic problem. *SIAM J. Sci. Comput.*, 31(4):2757–2783, 2009.

[71] M. Marion. Finite-dimensional attractors associated with partly dissipative reaction-diffusion systems. *SIAM J. Math. Anal.*, 20(4):816–844, 1989.

[72] S. Micheletti and S. Perotto. Reliability and efficiency of an anisotropic Zienkiewicz-Zhu error estimator. *Comput. Methods Appl. Mech. Engrg.*, 195:799–835, 2006.

[73] S. Micheletti and S. Perotto. Space-time adaptation for purely difusive problems in an anisotropic framework. *Int. J. Numer. Anal. Mod.*, 7(1):125–155, 2010.

[74] S. Micheletti, S. Perotto, and M. Picasso. Stabilized finite elements on anisotropic meshes: a priori error estimates for the advection-diffusion and the Stokes problems. *SIAM J. Numer. Anal.*, 41(3):1131–1162 (electronic), 2003.

[75] C. C. Mitchell and D. G. Schaeffer. A two-current model for the dynamics of cardiac membrane. *Bulletin of mathematical biology*, 65(5):767–793, 2003.

[76] W. Mitchell. A collection of 2D elliptic problems for testing adaptive algorithms. Technical report, NISTIR 7668, 2010.

[77] P. K. Moore. An adaptive finite element method for parabolic differential systems: some algorithmic considerations in solving in three space dimensions. *SIAM J. Sci. Comput.*, 21(4):1567–1586 (electronic), 1999/00.

[78] J. Pascal. A fully automatic adaptive isotropic surface remeshing procedure. 2001.

[79] L. F. Pavarino and P. Colli Franzone. Parallel solution of cardiac reaction-diffusion models. In *Domain decomposition methods in science and engineering*, volume 40 of *Lect. Notes Comput. Sci. Eng.*, pages 669–676. Springer, Berlin, 2005.

[80] M. Picasso. An anisotropic error indicator based on Zienkiewicz-Zhu error estimator: Application to elliptic and parabolic problems. *SIAM J. Sci. Comput.*, 24(4):1328–1355, 2003.

[81] M. Picasso. Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives. *Comput. Methods Appl. Mech. Engrg.*, 916:14–23, 2006.

[82] M. Picasso, F. Alauzet, H. Borouchaki, and P.-L. George. A numerical study of some Hessian recovery techniques on isotropic and anisotropic meshes. *SIAM J. Sci. Comput.*, 33(3):1058–1076, 2011.

[83] V. Prachittham, M. Picasso, and M. A. M. Gijs. Adaptive finite elements with large aspect ratio for mass transport in electroosmosis and pressure-driven microflows. *Internat. J. Numer. Methods Fluids*, 63(9):1005–1030, 2010.

[84] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Number 23 in Springer Series in Computational Mathematics. Springer-Verlag, 1997.

[85] M. Rioux. *Numerical computations of action potentials for the heart-torso coupling problem*. PhD thesis, University of Ottawa, 2012.

[86] M. Rioux and Y. Bourgault. A predictive method allowing the use of a single ionic model in numerical cardiac electrophysiology. *ESAIM Math. Model. Numer. Anal.*, 47(4):987–1016, 2013.

[87] R. Rodriguez. Some remarks on Zienkiewicz-Zhu estimator. *Numer. Methods Partial Different. Equat.*, 10:625–635, 1994.

[88] L. R. Scott and S. Zhang. Finite element interpolation of non-smooth of functions satisfying boundary conditions. *Math. Comp.*, 54:483–493, 1990.

[89] J. Smoller. *Shock waves and reaction-diffusion equations*, volume 258 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, New York, 2nd edition, 1994.

[90] J. Southern, G. Gorman, M. Piggott, and P. Farrell. Parallel anisotropic mesh adaptivity with dynamic load balancing for cardiac electrophysiology. *Journal of Computational Science*, 3(12):8 – 16, 2012.

[91] J. Southern, G. Gorman, M. Piggott, P. Farrell, M. Bernabeu, and J. Pitt-Francis. Anisotropic mesh adaptivity for cardiac electrophysiology. *Procedia Computer Science*, 1(1):935 – 944, 2010.

[92] J. Sundnes, G. T. Lines, X. Cai, B. F. Nielsen, K.-A. Mardal, and A. Tveito. *Computing the electrical activity in the heart*, volume 1 of *Monographs in Computational Science and Engineering*. Springer-Verlag, Berlin, 2006.

[93] R. Temam. *Infinite-Dimensional Dynamical Systemt in Mechanics and Physics*. Number 68 in Applied Mathematical Sciences. Springer-Verlag, 1988.

[94] K. H. W. J. ten Tusscher, D. Noble, P. J. Noble, and A. V. Panfilov. A model for human ventricular tissue. *American Journal of Physiology - Heart and Circulatory Physiology*, 286(4):H1573–H1589, 2004.

[95] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems*, volume 25 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2nd edition, 2006.

[96] J. A. Trangenstein and C. Kim. Operator splitting and adaptive mesh refinement for the Luo-Rudy I model. *J. Comput. Phys.*, 196(2):645–679, 2004.

[97] M.-G. Vallet, C.-M. Manole, J. Dompierre, S. Dufour, and F. Guibault. Numerical comparison of some Hessian recovery techniques. *Internat. J. Numer. Methods Engrg.*, 72(8):987–1007, 2007.

[98] R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. *Journal of Computational and Applied Mathematics*, 50(1):67–83, 1994.

[99] V. Volpert. *Elliptic partial differential equations. Vol. 2*, volume 104 of *Monographs in Mathematics*. Birkhäuser/Springer Basel AG, Basel, 2014. Reaction-diffusion equations.

[100] P. J. Whiteley. Physiology driven adaptivity for the numerical solution of the bidomain equations. *Annals of Biomedical Engineering*, 35(9):1510–1520, 2007.

[101] Z. Zhang and A. Naga. A new finite element gradient recovery method: Superconvergence property. *SIAM J. Sci. Comput.*, 26(4):1192–1213, 2005.

[102] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Internat. J. Numer. Methods Engrg.*, 24(2):337–357, 1987.