DISSERTATION

Reliable goal-oriented adaptive FEM

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften unter der Leitung von

Univ.-Prof. Dr. Dirk Praetorius
E101 – Institut für Analysis und Scientific Computing, TU Wien

eingereicht an der Technischen Universität Wien
Fakultät für Mathematik und Geoinformation
von

Dipl.-Ing. Michael Innerberger, BSc BSc
Matrikelnummer: 01225448

Diese Dissertation haben begutachtet:

Prof. Dr. Roland Becker
Laboratoire de mathématiques et de leurs applications, Université de Pau et des Pays de l'Adour

Prof. Dr. Dirk Praetorius
Institut für Analysis und Scientific Computing, TU Wien

Prof. Dr. Rob Stevenson
Korteweg–de Vries Instituut voor Wiskunde, Universiteit van Amsterdam

Wien, am 19. April 2022

Kurzfassung

Diese Arbeit betrachtet zielorientierte adaptive Finite Elemente Methoden (GOAFEM, engl. *goal-oriented adaptive finite element method*). Diese versuchen, eine von der Lösung einer partiellen Differentialgleichung (PDE, engl. *partial differential equation*) abgeleiteten Zielgröße zu approximieren. Trotz der praktischen Relevanz von GOAFEM ist mathematische Forschung dazu rar. Insbesondere ist die existierende Forschung zu optimaler GOAFEM im Wesentlichen auf lineare elliptische PDEs mit linearen Zielen beschränkt. In dieser Arbeit erweitern wir existierende Resultate zu GOAFEM in Richtung praktisch relevanterer Szenarien und entwerfen Algorithmen, die zuverlässig die Zielgröße mit hoher (oder sogar optimaler) Effizienz approximieren.

Zuerst wird ein kurzer Überblick über existierende Resultate zu optimaler GOAFEM gegeben, bevor wir erstmals GOAFEM für lineare elliptische PDEs mit quadratischem Ziel betrachten. Wir stellen einen adaptiven Algorithmus vor, der ein linearisiertes duales Problem für Fehlerschätzung und Markierung verwendet und mit dem auftretenden Linearisierungsfehler umgehen kann. Wir beweisen Konvergenz dieses Algorithmus für jedes quadratische Ziel und, darüber hinaus, Konvergenz mit optimalen algebraischen Raten, sofern die Fréchet-Ableitung des Ziels kompakt ist.

Als nächstes untersuchen wir die Optimalität von GOAFEM für lineare elliptische PDEs mit linearem Ziel, wobei die primale und duale Lösung durch einen (inexakten) iterativen Löser berechnet werden. Wir beobachten, dass die diskrete Zielgröße in diesem Fall korrigiert werden muss und beweisen (lineare) Konvergenz des korrigierten Zielfehlers unter der einzigen Annahme, dass der Löser kontraktiv ist. Weiters präsentieren wir Kriterien basierend auf *a posteriori* Fehlerschätzern für den Diskretisierungsfehler und den algebraischen Fehler, um den iterativen Löser in jedem Schritt des adaptiven Algorithmus zu terminieren. Falls die involvierten Parameter hinreichend klein sind, ist der resultierende adaptive Algorithmus optimal in Bezug auf die Anzahl der Freiheitsgrade und sogar die gesamte Rechenzeit, die auch den Aufwand für das iterative Lösen inkludiert.

Als Anwendung für GOAFEM betrachten wir danach Parameterschätzprobleme für lineare elliptische PDEs, die von einer endlichen Anzahl an Parametern abhängen. Die Parameter werden durch Vergleich von Experimentaldaten und numerischen Simulationen berechnet, welche mittels GOAFEM durchgeführt werden können, indem die Parameter als Zielgröße betrachtet werden. Wir beweisen eine neuartige *a priori* Abschätzung für den Parameterfehler, basierend auf PDEs, die vom primalen und dualen Problem abhängen. Indem diese Abschätzung als Basis einer Abschätzung durch gewöhnliche *a posteriori* Residualschätzer für Energiefehler fungiert, können wir einen adaptiven Algorithmus entwerfen, bei dem die Konvergenzrate der *a posteriori* Abschätzung der des Parameterfehlers entspricht.

In allen Teilen der Arbeit präsentieren wir numerische Belege für unsere theoretischen Ergebnisse. Der letzte Teil der Arbeit gilt daher Implementierungsaspekten numerischer Experimente für GOAFEM, wobei wir eine objektorientierte Matlab-Implementierung im Detail beschreiben. Diese Bibliothek implementiert FEM höherer Ordnung für elliptische PDEs zweiter Ordnung, wobei die Koeffizienten sehr allgemein gewählt werden können, um auch die meisten Fälle, die typischerweise bei der iterativen Linearisierung nichtlinearer PDEs auftreten, abzudecken. Insbesondere umfasst der Code alle angegebenen numerischen Experimente.

Abstract

This thesis considers goal-oriented adaptive finite element methods (GOAFEM), which aim to approximate some quantity of interest, the goal value, derived from the solution of a partial differential equation (PDE). Despite the practical relevance of GOAFEM, mathematical research on it is scarce and, in particular, existing research on optimal algorithms for GOAFEM is essentially limited to linear elliptic PDEs with linear goals. In this thesis we extend existing results of GOAFEM towards practically more relevant cases and design algorithms that reliably approximate the goal value at high (or even optimal) efficiency.

First, we give a brief overview of the existing results on optimal GOAFEM before we consider, for the first time, GOAFEM for linear elliptic PDEs with quadratic goal. We propose an adaptive algorithm that uses a linearized dual problem for error estimation and marking, and deals with the arising linearization error. We prove convergence of this algorithm for every quadratic goal and even convergence with optimal algebraic rates in the case that the Fréchet derivative of the goal is compact.

Next, we investigate optimality results of GOAFEM for linear elliptic PDEs with linear goal, where the primal and dual problem are solved by an (inexact) iterative solver. We observe that the discrete goal value needs to be corrected in this case, and prove (linear) convergence of the corrected goal error under the sole assumption that the solver is contractive. Furthermore, we present criteria to stop the iterative solver on each step of the adaptive algorithm based on *a posteriori* error estimates of both discretization error and algebraic error. If the involved parameters are sufficiently small, the resulting adaptive algorithm is optimal with respect to the number of degrees of freedom and even with respect to the total computational cost, which also includes the cost of the iterative solver.

As an application of GOAFEM, we then consider parameter estimation problems for linear elliptic PDEs that depend on a finite number of parameters. These parameters are inferred by comparing experimental data to numerical simulations, which, by regarding the parameters as a goal value, can be performed by GOAFEM. We prove a novel *a priori* estimate for the error in the parameters, based on a set of PDEs corresponding to primal and dual problem. Using this estimate as the basis of an estimate by usual *a posteriori* residual estimators for the energy error, we are able to design an adaptive algorithm, where the *a posteriori* bound matches the rate of convergence of the parameter error.

Throughout, we give numerical evidence to support our theoretical findings. The last part of this thesis is dedicated to implementational aspects of numerical experiments for GOAFEM, where we give the details of an object oriented implementation in Matlab. The library implements higher-order FEM for second-order elliptic PDEs where the coefficients can be quite general, covering also most cases typically arising from the iterative linearization of nonlinear PDEs. In particular, the code covers all presented numerical experiments.

Danksagung

Zuallererst möchte ich Prof. Dirk Praetorius danken, von dem ich in den letzten dreieinhalb Jahren unglaublich viel lernen durfte. Als Betreuer meiner Dissertation hat er mir immer die nötige Anleitung, aber auch genügend Freiheiten gegeben, um mich fachlich und persönlich weiterentwickeln zu können. Er hat sich immer um eine kritische, aber stets freundschaftliche Haltung bemüht, die die Zusammenarbeit äußerst produktiv und angenehm gemacht hat.

Für die Begutachtung meiner Arbeit danke ich an dieser Stelle zum einen Prof. Rob Stevenson von der Universität Amsterdam; zum anderen Prof. Roland Becker, der mich einige Zeit an der Universität Pau willkommen geheißen hat, wo ich während etlicher anregender Gespräche und Unternehmungen viel Neues kennenlernen durfte.

Mein Dank gilt auch allen ehemaligen und aktuellen Arbeitskollegen: Max Bernkopf, Simon Brandstetter, Max Brunner, Markus Faustmann, Michael Feischl, Giovanni Di Fratta, Gregor Gantner, Alexander Haberl, Ani Miraçi, Carl Pfeiler, Alexander Rieder, Michele Ruggeri, Andrea Scaglioni, Stefan Schimanko, Ursula Schweigler, Bernhard Stiftner, und Julian Streitberger. Viele von ihnen waren von Beginn meines Doktoratsstudiums an dabei und haben mir den Einstieg durch ihren fachlichen Rat oft wesentlich erleichtert. Die Zusammenarbeit mit allen, aber auch die unzähligen interessanten Gespräche über Gott und die Welt während so mancher Kaffeepause haben maßgeblich dazu beigetragen, dass ich immer gerne in die Arbeit gegangen bin.

Mit dem Abschluss des Doktorats geht für mich eine unvergessliche Studienzeit zu Ende. Ich bedanke mich ganz herzlich bei allen Studienkollegen und Freunden, die mich auf dem Weg begleitet haben und ohne die ich mein Studium nicht annähernd so erfolgreich hätte abschließen können. Sie haben diese Zeit zu etwas ganz Besonderem gemacht.

Schließlich danke ich auch meinen Eltern, die mir in allen Lebenslagen Stütze und Vorbild waren und mich in all meinen Entscheidungen bestärkt haben; nicht zuletzt auch dafür, dass sie mir mein Studium überhaupt ermöglicht haben.

Mehr als alles andere hat mich in den vergangenen sieben Jahren jedoch die Unterstützung meiner Freundin Magdalena getragen. Danke für die wunderbare gemeinsame Zeit, für dein Verständnis und die Geduld, die du allen meinen kleinen und großen Problemen entgegenbringst, und für deine bedingungslose Liebe.

Ich danke dem *österreichischen Wissenschaftsfonds (FWF)*, der meine Arbeit über das Doktoratskolleg *Dissipation and dispersion in nonlinear PDEs* (grant W1245) und den SFB *Taming complexity in partial differential systems* (grant SFB F65) finanziert hat, sowie dem *Institut Français d'Autriche*, welches mir einen Auslandsaufenthalt an der Universität Pau ermöglicht hat.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 19. April 2022

Michael Innerberger

Contents

1 Introduction **1**

1.1 The finite element method . . . 2
1.1.1 Model problem . . . 2
1.1.2 Discretization of the domain: meshes . . . 3
1.1.3 Discretization of the equation . . . 5
1.1.4 Goal-oriented FEM . . . 6
1.2 A goal-oriented adaptive FEM algorithm . . . 7
1.2.1 Mesh refinement: newest vertex bisection . . . 7
1.2.2 A posteriori error estimation . . . 9
1.2.3 Marking . . . 10
1.2.4 Adaptive algorithm . . . 12
1.3 Optimal convergence of GOAFEM . . . 13
1.3.1 Necessary abstract properties . . . 14
1.3.2 Rate optimality . . . 16
1.3.3 Main steps of proof . . . 18
1.4 Outline of thesis . . . 24
1.5 Other scientific contributions . . . 27
1.5.1 Instance optimal GOAFEM . . . 27
1.5.2 GOAFEM for semilinear problems . . . 27
1.5.3 Weak-strong uniqueness for solutions of the LLG equation . . . 28
1.5.4 Exact diagonalization of time-dependent Hamiltonians . . . 28
1.5.5 Impact ionization in solar-cell models . . . 29

2 Optimal convergence rates for goal-oriented FEM with quadratic goal functional **31**

2.1 Introduction . . . 31
2.2 Adaptive algorithm & main result . . . 33
2.2.1 Variational formulation . . . 33
2.2.2 Finite element method . . . 34
2.2.3 Linearization of the goal functional . . . 34
2.2.4 Mesh-refinement . . . 34
2.2.5 Error estimators . . . 35
2.2.6 Adaptive algorithm . . . 36
2.2.7 Alternative adaptive algorithm . . . 38
2.2.8 Extension of analysis to compactly perturbed elliptic problems . . . 40
2.3 Numerical experiments . . . 41
2.3.1 Weighted L^2-norm . . . 41
2.3.2 Nonlinear convection . . . 42
2.3.3 Force evaluation . . . 42

2.3.4 Discussion of numerical experiments . . . 43
2.4 Auxiliary results . . . 45
2.4.1 Axioms of adaptivity . . . 45
2.4.2 Quasi-orthogonality . . . 47
2.5 Proof of plain convergence of Algorithm 2A and 2B . . . 49
2.5.1 Algorithm 2A . . . 49
2.5.2 Algorithm 2B . . . 51
2.6 Proof of Theorem 2.2 . . . 51
2.6.1 Linear convergence . . . 51
2.6.2 Optimal rates . . . 53
2.7 Proof of Theorem 2.5 . . . 55

3 Goal-oriented adaptive finite element methods with optimal computational complexity 57
3.1 Introduction . . . 57
3.2 Goal-oriented adaptive finite element method . . . 59
3.2.1 Variational formulation . . . 59
3.2.2 Finite element discretization and solution . . . 60
3.2.3 Discrete goal quantity . . . 60
3.2.4 Mesh refinement . . . 61
3.2.5 Estimator properties . . . 61
3.2.6 Marking strategy . . . 62
3.2.7 Adaptive algorithm . . . 63
3.3 Main results . . . 65
3.3.1 Linear convergence with optimal rates . . . 65
3.3.2 Alternative termination criteria for iterative solver . . . 67
3.4 Numerical examples . . . 69
3.4.1 Singularity in goal functional only . . . 70
3.4.2 Geometrical singularity . . . 73
3.5 Proof of Theorem 3.5 . . . 73
3.6 Proof of Theorem 3.7 (optimal rates) . . . 80

4 Adaptive FEM for parameter-errors in elliptic linear-quadratic parameter estimation problems 83
4.1 Introduction . . . 83
4.2 Parameter estimation problem . . . 85
4.2.1 Problem formulation . . . 85
4.2.2 Solution components . . . 86
4.2.3 Least squares system and solution . . . 86
4.2.4 FEM discretization . . . 87
4.2.5 Co-state components . . . 88
4.3 Adaptive algorithm and main results . . . 88
4.3.1 A priori estimate . . . 88
4.3.2 Mesh refinement . . . 89
4.3.3 A posteriori error estimation . . . 89
4.3.4 Adaptive algorithm . . . 91

4.3.5 Convergence of Algorithm 4A . . . 92
4.4 Proof of Theorem 4.5 . . . 93
4.4.1 Auxiliary a priori bounds . . . 93
4.4.2 Error bound for parameter error . . . 96
4.5 Proof of Theorems 4.13 and 4.14 . . . 99
4.5.1 Linear convergence . . . 99
4.5.2 Proof of optimal rates . . . 100
4.6 Numerical examples . . . 102
4.6.1 Single parameter and measurement . . . 102
4.6.2 Multiple parameters and measurements with perturbation . . . 103

5 MooAFEM: An object oriented Matlab code for higher-order (nonlinear) adaptive FEM 107
5.1 Introduction . . . 107
5.2 Adaptive algorithm and importance of OOP . . . 109
5.2.1 Necessity of OOP in MATLAB FEM . . . 111
5.3 Code structure . . . 113
5.3.1 Module geometry . . . 113
5.3.2 Module integration . . . 116
5.3.3 Module FEM . . . 118
5.4 Data structures . . . 119
5.4.1 Mesh . . . 119
5.4.2 Mesh construction . . . 120
5.4.3 Array layout . . . 120
5.4.4 Efficient linear algebra . . . 121
5.5 Examples . . . 122
5.5.1 Higher order AFEM with known solution . . . 122
5.5.2 Goal-oriented AFEM with discontinuous data . . . 125
5.5.3 Iterative solution of nonlinear equations . . . 127

Bibliography 131

Curriculum vitae 137

1 Introduction

> *As Henri Poincaré once remarked, "solution of a mathematical problem" is a phrase of indefinite meaning. Pure mathematicians sometimes are satisfied with showing that the non-existence of a solution implies a logical contradiction, while engineers might consider a numerical result as the only reasonable goal. Such one sided views seem to reflect human limitations rather than objective values. In itself mathematics is an indivisible organism uniting theoretical contemplation and active application.*
>
> — Richard Courant, Address to the meeting of the AMS, 1941 [Cou43]

Mathematics has an exceptional position among all scientific disciplines in that it is fundamentally different from all other fields: instead of making use of the *scientific method*, which seeks to falsify theories through comparison with reality, i.e., by experiment, mathematics can *prove* its theories without ever making a connection to the "real" world.

Yet, mathematics has indispensable implications on our daily lives: The models and theories of most scientific fields are formulated in the language of mathematics, be it engineering, medicine, or social science. Once this translation is performed, the abstract construct of mathematics often allows to make predictions or even draw conclusions within those theories. In return, applications often provide intuition that directs and drives mathematical progress.

It is exactly this mutual support of mathematics and applications that Richard Courant identifies as the inherent nature of mathematics in the speech quoted at the beginning of this chapter. He proceeds in this address to describe and highlight the importance of *Variational methods for the solution of problems of equilibrium and vibrations*, which are, in our modern nomenclature, Galerkin methods. Since the time of this speech, Galerkin methods have become the arguably most widespread tool for numerical simulation, thus further tightening the bond of mathematics and applications. This thesis is concerned with one particular class of Galerkin methods: goal oriented adaptive finite element methods (GOAFEM).

The starting point of any FEM computation is a partial differential equation (PDE) that is often motivated by a model in physics or engineering, although other disciplines like economics, medicine, and social sciences are also relying on PDE models to a greater extent in recent years. These models have become so complex that solving the underlying PDE is not possible by hand anymore and therefore numerical solution by FEM is required. This is done by discretizing the underlying physical domain into finitely many subdomains, the *elements*, on which the solution is approximated by simple functions, e.g., piecewise polynomials.

In applications, however, one is often not interested in the solution as a whole but only some derived quantity of interest, the *goal (value)*, e.g., the flux over the boundary or the mass in only a part of the domain. Therefore, it is reasonable to choose the FEM discretization such that the goal is as accurate as possible. Unfortunately, it is not clear in advance what parts of the solution are most critical for choosing the discretization. As a consequence, *adaptive* algorithms have emerged, which iteratively refine the discretization to make best use of computational resources for approximating the goal value.

Adaptive FEM algorithms are of the following general form:

(1.1)

While the precise meaning of the building blocks is described in the further course of this chapter and dependent on the specific problem one wants to solve, we stress here that GOAFEM is an iterative process: A tentative solution is computed (SOLVE), from which estimates of the accuracy of the goal are inferred (ESTIMATE). Subsequently, the latter are used to change the underlying discretization in order to improve the solution quality (MARK and REFINE). These steps are repeated until a prescribed accuracy is reached, or computational resources are exhausted.

The naturally arising question is whether these iterative computations on successively refined discretizations are actually more expensive than doing the computation once on a sufficiently fine uniform discretization. The aim of this thesis is to design GOAFEM algorithms that negate this question in a reliable way: under some assumptions, they provide a guaranteed upper bound of the goal error and drive down this bound in an optimal way, i.e., by making best use of available computational resources. So far, this has only been done for some problems of mainly academic interest. In the following, we first give a precise mathematical introduction on the concepts mentioned so far and, subsequently, extend the existing scientific literature on GOAFEM towards more practically relevant cases.

1.1 The finite element method

We first give a brief introduction to the overarching model problem of this thesis and its discretization by the finite element method, which constitutes the step SOLVE from the adaptive loop (1.1). For more details, the reader is referred to a standard reference on finite elements, e.g., [BS08; EG04].

1.1.1 Model problem

Let $\Omega \subset \mathbb{R}^d$ be an open domain with polygonal Lipschitz boundary $\partial\Omega$ for $d \geq 2$. On this domain, we consider the *linear second-order elliptic PDE*

$$-\operatorname{div} \boldsymbol{A}\nabla u + \boldsymbol{b} \cdot \nabla u + cu = f + \operatorname{div} \boldsymbol{f} \qquad \text{in } \Omega, \tag{1.2a}$$

$$u = 0 \qquad \text{on } \partial\Omega. \tag{1.2b}$$

We suppose that the diffusion matrix $\boldsymbol{A}(x) \in \mathbb{R}^{d\times d}_{\mathrm{sym}}$ is symmetric and that $\boldsymbol{A} \in [L^\infty(\Omega)]^{d\times d}$ is uniformly positive definite. Furthermore, we assume that the convection vector $\boldsymbol{b}(x) \in \mathbb{R}^d$ and the reaction coefficient $c(x) \in \mathbb{R}$ are essentially bounded, i.e., $\boldsymbol{b} \in [L^\infty(\Omega)]^d$ and $c \in L^\infty(\Omega)$. Finally, we require that the given data are square integrable, i.e., $f \in L^2(\Omega)$ and $\boldsymbol{f} \in [L^2(\Omega)]^d$, where the divergence in (1.2a) is understood in a weak sense.

Remark 1.1. *The PDE (1.2) is the least general equation that covers all model problems of the subsequent chapters. Where necessary or favorable for the presentation, the model is restricted accordingly. However, some results apply to more general problems than (1.2). Both cases are noted at the beginning of the respective chapters.*

The natural space to look for solutions of (1.2) is the Sobolev space

$$H^1(\Omega) := \{v \in L^2(\Omega) \,|\, \nabla v \in [L^2(\Omega)]^d \text{ in a weak sense}\},$$

which is a Hilbert spaces when equipped with the scalar product

$$\langle v\,,\, w\rangle_{H^1(\Omega)} := \int_\Omega \langle v\,,\, w\rangle + \langle \nabla v\,,\, \nabla w\rangle \,\mathrm{d}x \quad \text{for all } v, w \in H^1(\Omega),$$

where $\langle \cdot\,,\, \cdot\rangle$ is the Euclidian scalar product on $\mathbb{R}^n$ for some $n \in \mathbb{N}$ which is clear from context. The corresponding norm reads

$$\|v\|^2_{H^1(\Omega)} := \|v\|^2_{L^2(\Omega)} + \|\nabla v\|^2_{L^2(\Omega)} \quad \text{for all } v \in H^1(\Omega).$$

Functions $v \in H^1(\Omega)$ admit a trace in $L^2(\partial\Omega)$ [Gri11]. Thus, the boundary condition (1.2b) can be incorporated in the ansatz space, i.e.,

$$H^1_0(\Omega) := \{v \in H^1(\Omega) \,|\, v = 0 \text{ on } \partial\Omega\}.$$

We finally define the topological dual space $H^{-1}(\Omega) := \big(H^1_0(\Omega)\big)'$.

The weak formulation of (1.2), which is obtained by multiplication of (1.2) with a test function $v \in H^1_0(\Omega)$ and integration by parts, then reads: Find $u \in H^1_0(\Omega)$ such that

$$a(u, v) := \int_\Omega \boldsymbol{A}\nabla u \cdot \nabla v + \boldsymbol{b} \cdot \nabla u\, v + cuv \,\mathrm{d}x = \int_\Omega fv - \boldsymbol{f} \cdot \nabla v \,\mathrm{d}x =: F(v) \quad \text{for all } v \in H^1_0(\Omega). \tag{1.3}$$

As we are not concerned with solvability aspects in this work, we assume that the coefficients are chosen in such a way that (1.3) fits into the setting of the Lax–Milgram lemma, i.e., there exist constants $C_{\mathrm{cnt}}, C_{\mathrm{ell}} > 0$ such that

$$C_{\mathrm{ell}}\,\|v\|^2_{H^1(\Omega)} \le a(v, v) \quad \text{and} \quad a(v, w) \le C_{\mathrm{cnt}}\,\|v\|_{H^1(\Omega)}\|w\|_{H^1(\Omega)} \qquad \text{for all } v, w \in H^1_0(\Omega); \tag{1.4}$$

see, e.g., [Eva10, Section 6.2]. Under these assumptions, (1.3) admits a unique solution. We further note that the so-called *energy norm*

$$|\!|\!| v |\!|\!|^2 := \int_\Omega \boldsymbol{A}\nabla v \cdot \nabla v \,\mathrm{d}x$$

induced by the principal part of $a(\cdot, \cdot)$ is an equivalent norm on $H^1_0(\Omega)$.

1.1.2 Discretization of the domain: meshes

As noted at the beginning of this chapter, it is often not possible to analytically find a solution to (1.3). It is therefore necessary to approximate solutions numerically, e.g., by a Galerkin method. Common to all of them is the principle of solving (1.3) on a *finite dimensional* subspace $\mathcal{X}_H \subseteq H^1_0(\Omega)$. This basic idea was apparently already known to Leonhard Euler in the eighteenth century, but emerged as an independent method through the seminal works of Walter Ritz and Boris Galerkin at the beginning of the twentieth century [GW12]. Among the most widely used implementations of this general principle are Krylov and spectral methods, as well as boundary and finite element methods. This thesis is concerned with the latter, the modern understanding of which was fundamentally shaped by Richard Courant, who contributed the introductory quote of this thesis.

The finite element method consists of dividing the computational domain Ω into smaller pieces, the *elements*, on which local approximations to the solution are constructed from polynomials. The collection of all elements is called *mesh* [EG04].

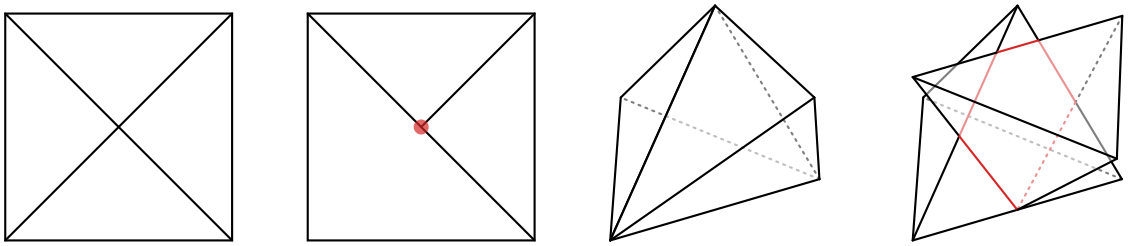


Figure 1.1: Example of conforming 2D, non-conforming 2D, conforming 3D, and non-conforming 3D triangulation (left to right). The mismatched parts are highlighted in red.

Definition 1.2: Mesh

Let $\Omega \subseteq \mathbb{R}^d$ be an open domain. A mesh is a finite set $\mathcal{T}_H$ of closed, connected sets in $\mathbb{R}^d$, called elements*, such that*

- *the d-dimensional Lebesgue measure is positive, $|T| > 0$ for all $T \in \mathcal{T}_H$;*
- *the elements are non-overlapping, $|T \cap T'| = 0$ for all $T, T' \in \mathcal{T}_H$ with $T \neq T'$;*
- *Ω is covered by $\mathcal{T}_H$, $\bigcup_{T \in \mathcal{T}_H} T = \overline{\Omega}$.*

An example of an element is a d-dimensional simplex $T = \operatorname{conv}\{z_0, z_1, \ldots, z_d\}$, which is the convex hull of $d+1$ points $z_0, z_1, \ldots, z_d \in \mathbb{R}^d$, i.e., a triangle in two and a tetrahedron in three dimensions. For $0 \leq n \leq d$, an n-dimensional subsimplex is the convex hull of a subset $z_{i_0}, z_{i_1}, \ldots, z_{i_n}$. The boundary of a simplex consists of $(d-1)$-dimensional subsimplices, which we call *faces*; one dimensional subsimplices are called *edges*. In this thesis, we exclusively consider simplicial meshes. We note, however, that also meshes of quadrilaterals [BN10], general polytopes (e.g., in the virtual element method [BBC+13]), and even non-polytopal elements (e.g., in isogeometric analysis [CHB09]) are commonly used. Furthermore, we restrict ourselves to conforming meshes; see Figure 1.1 for a visualization.

Definition 1.3: Conforming (simplicial) mesh

We say that a simplicial mesh is conforming *if, for $T, T' \in \mathcal{T}_H$, the intersection $T \cap T'$ is either empty or a subsimplex of both T and T'. In particular, no hanging nodes (0-dimensional subsimplices) must occur.*

We associate to each element $T \in \mathcal{T}_H$ its local size

$$h_T := |T|^{1/d}$$

and say that a mesh is γ-*shape regular* for some $\gamma > 0$ if

$$\max_{T \in \mathcal{T}_H} \frac{\operatorname{diam}(T)^d}{|T|} < \gamma < \infty. \tag{1.5}$$

1.1.3 Discretization of the equation

By means of the discretization of the underlying computational domain Ω, the finite element method discretizes also the space of functions, in which solutions to (1.3) are sought. To this end, we define the space of piecewise polynomials of degree $p \in \mathbb{N}$ by

$$\mathcal{S}^p(\mathcal{T}_H) := \left\{ v \in H^1(\Omega) \,\middle|\, v|_T \text{ is a polynomial of degree } p \text{ for all } T \in \mathcal{T}_H \right\}$$

and set $\mathcal{S}^p_0(\mathcal{T}_H) := \mathcal{S}^p(\mathcal{T}_H) \cap H^1_0(\Omega)$. We sometimes abbreviate $\mathcal{X}_H := \mathcal{S}^p_0(\mathcal{T}_H)$ as well as $\mathcal{X} := H^1_0(\Omega)$. The discretization of the weak formulation (1.3) then reads:

$$\text{Find } u \in \mathcal{X}_H \text{ such that} \qquad a(u_H, v_H) = F(v_H) \quad \text{for all } v_H \in \mathcal{X}_H. \tag{1.6}$$

By using a basis $(\varphi_k)_{k=1}^N$ of $\mathcal{S}^p_0(\mathcal{T}_H)$, where $N := \dim\left(\mathcal{S}^p_0(\mathcal{T}_H)\right)$ is the number of *degrees of freedom*, this equation reduces to the linear system

$$Ax = b, \tag{1.7}$$

where

$$A_{ij} = a(\varphi_j, \varphi_i) \quad \text{and} \quad b_i = F(\varphi_i) \qquad \text{for all } 1 \le i, j \le N.$$

The solution vector $x \in \mathbb{R}^N$ then yields the coefficients of the discrete solution of (1.6), i.e.,

$$u_H = \sum_{k=1}^{N} x_i \varphi_i. \tag{1.8}$$

Because we supppose that the coefficients of (1.2) are such that the Lax–Milgram theory can be applied, problem (1.6) (as well as the linear system (1.7)) admits a unique solution. There further holds the following quasi-best approximation result [EG04, Lemma 2.28].

Theorem 1.4: Céa lemma

Let $u \in H^1_0(\Omega)$ be the solution of (1.2) and $u_H \in \mathcal{X}_H$ be its FEM-approximation, i.e., the solution of (1.6). Then, there holds

$$\|u - u_H\|_{H^1(\Omega)} \le \frac{C_{\mathrm{cnt}}}{C_{\mathrm{ell}}} \inf_{v_H \in \mathcal{X}_H} \|u - v_H\|_{H^1(\Omega)}. \tag{1.9}$$

Usually, one further estimates the infimum in (1.9) by approximation properties of the space $\mathcal{X}_H$. This gives rise to *a priori* error estimates of the form

$$|||u - u_H||| \lesssim H^\alpha, \tag{1.10}$$

where $\alpha > 0$ is the *rate* of approximation, H is the global mesh width of $\mathcal{T}_H$, i.e.,

$$H := \max_{T \in \mathcal{T}_H} h_T,$$

and $a \lesssim b$ for $a, b \in \mathbb{R}$ means that there exists a constant $C > 0$ such that $a \le C\, b$; in particular, we suppose that the hidden constant is independent of H. In the following, if there holds $a \lesssim b \lesssim a$, we abbreviate this by $a \simeq b$. The estimate (1.10) implies convergence of the finite element approximation $u_H \to u$ if $H \to 0$, i.e., if the elements of the used mesh become uniformly small.

1.1.4 Goal-oriented FEM

While standard FEM tries to approximate the whole solution in the energy norm and aims to minimize the error in the energy norm (1.10), one is often only interested in a small part of the information provided by the solution u. This interesting part is modeled by a so-called (linear) *goal functional* $G \in H^{-1}(\Omega)$ such that the quantity of interest becomes $G(u)$, also called *goal value* or just goal. The goal can be approximated by means of the discrete FEM solution via $G(u_H)$. The overall aim then is to minimize the *goal error* $|G(u) - G(u_H)|$, the FEM becomes *goal-oriented*.

To estimate the goal error, one can use boundedness of the goal functional to obtain

$$|G(u) - G(u_H)| \lesssim \|G\|_{H^{-1}(\Omega)} \|\!| u - u_H |\!\|. \tag{1.11}$$

Then, the error estimate (1.10) guarantees convergence of the goal error if the mesh width H tends to zero. This, however, does not take into account that only the derived quantity $G(u)$ rather than the solution u as a whole needs to be approximated.

To make use of this information, one considers the so-called *dual* problem:

$$\text{Find } z \in H^1_0(\Omega) \text{ such that} \qquad a(v, z) = G(v) \quad \text{for all } v \in H^1_0(\Omega). \tag{1.12}$$

For the remainder of this chapter, we suppose that the goal functional has essentially the same structure as the right-hand side of the model problem (1.3), i.e., there exist functions $g \in L^2(\Omega)$ and $\boldsymbol{g} \in [L^2(\Omega)]^d$ such that

$$G(v) = \int_\Omega g v - \boldsymbol{g} \cdot \nabla v \,\mathrm{d}x \quad \text{for all } v \in H^1_0(\Omega). \tag{1.13}$$

In explicit terms, the weak formulation of the dual problem then reads

$$\int_\Omega \nabla z \cdot \boldsymbol{A} \nabla v - \boldsymbol{b} \cdot \nabla z \, v + (c - \operatorname{div} \boldsymbol{b}) z v \,\mathrm{d}x = \int_\Omega g v - \boldsymbol{g} \cdot \nabla v \,\mathrm{d}x \quad \text{for all } v \in H^1_0(\Omega). \tag{1.14}$$

Note that the difference to (1.3), besides the obvious change of right-hand side, is the order of the arguments in the bilinear form $a(\cdot, \cdot)$. In this context, problem (1.3) is sometimes also called *primal problem*. On some mesh $\mathcal{T}_H$, one can approximate the dual solution z by some finite element function $z_H \in \mathcal{X}_H$ that satisfies

$$a(v_H, z_H) = G(v_H) \quad \text{for all } v_H \in \mathcal{X}_H. \tag{1.15}$$

A further ingredient of a suitable goal error estimate is the Galerkin orthogonality for the primal problem,

$$a(u - u_H, v_H) = 0 \quad \text{for all } v_H \in \mathcal{X}_H, \tag{1.16}$$

which can be seen by subtracting (1.6) from (1.3). Combining all previous equalities yields

$$G(u) - G(u_H) \overset{(1.12)}{=} a(u - u_H, z) \overset{(1.16)}{=} a(u - u_H, z - z_H).$$

Finally, continuity (1.4) of the bilinear form yields

$$|G(u) - G(u_H)| \lesssim \|\!| u - u_H |\!\| \, \|\!| z - z_H |\!\|. \tag{1.17}$$

If the primal and the dual problem can be approximated as in (1.10) with a rate $\alpha > 0$ and $\beta > 0$, respectively, the goal error can be approximated at a rate $\alpha + \beta$, i.e.,

$$|G(u) - G(u_H)| \lesssim H^{\alpha+\beta}. \tag{1.18}$$

Thus, if one is not interested in the overall solution u but only the goal $G(u)$, one can expect faster convergence of the approximation in this case.

1.2 A goal-oriented adaptive FEM algorithm

In this section, we go into more detail about the remaining components of the adaptive loop (1.1), i.e., ESTIMATE, MARK, and REFINE, with the final aim of formulating a concrete GOAFEM algorithm. The idea of employing an adaptive loop to accelerate FEM computations already goes back to [BV84]. While our modern understanding of adaptivity for energy norm errors (1.10) was shaped by the seminal contributions [Dör96; MNS00], goal-oriented adaptivity is much younger, albeit being often more important in applications. The idea of abandoning the naive approach suggested by (1.11) and, instead, exploiting the dual problem (1.12) was first used in [BR01; BR03; EEHJ95; GS02], mainly in the course of the so-called *dual-weighted residual (DWR)* method. GOAFEMs that employ residual error estimation for energy norms, which enables extensive convergence results, were only introduced several years later in the seminal works [BET11; MS09]. We present here the most recent viewpoint.

1.2.1 Mesh refinement: newest vertex bisection

We first tend to (local) mesh refinement, i.e., the process of obtaining a new mesh from subdividing a number of elements in the old mesh. Since we do not permit hanging nodes, generally more elements than those singled out for refinement have to be refined to recover conformity. To do this in a way that does not refine too many elements and yet does not generate badly shaped elements is a non-trivial task. From this viewpoint, the most suitable refinement algorithm for adaptive FEM is *newest vertex bisection* (NVB) [KPP13; Mau95; Ste08; Tra97]. We only state here the most simple case that is not trivial, $d = 2$, in order to not overload the presentation with technicalities. Afterwards, we remark on extensions to general dimensions.

Algorithm in two dimensions

Our presentation for this case follows [KPP13]. We associate to every element $T \in \mathcal{T}_H$ one of its edges, called *refinement edge* and denoted by $\mathrm{re}(T)$. If the element $T = \mathrm{conv}\{z_0, z_1, z_2\}$ with $\mathrm{re}(T) = \mathrm{conv}\{z_1, z_2\}$ is refined, it is split into two simplices by bisecting the refinement edge as follows: A new node $z = (z_1 + z_2)/2$ is introduced and the new elements are

$$T_1 = \mathrm{conv}\{z_0, z_1, z\} \quad \text{and} \quad T_2 = \mathrm{conv}\{z_0, z, z_2\}$$

with refinement edges $\mathrm{re}(T_1) = \mathrm{conv}\{z_0, z_1\}$ and $\mathrm{re}(T_2) = \mathrm{conv}\{z_0, z_2\}$, respectively. These are the edges opposite to the *newest vertex*; hence, the name. Further refinement of the child elements is carried out by the same procedure. The whole algorithm reads as follows.

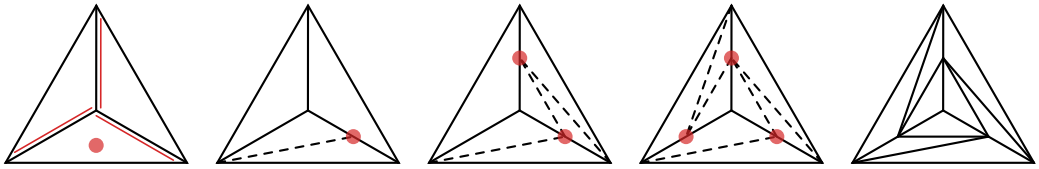


Figure 1.2: Example of the closure step, Algorithm 1A(ii). In the leftmost mesh, the refinement edges and the marked element are highlighted. Algorithm 1A(ii) successively marks edges such that no hanging nodes are present (middle); tentative new edges are dashed. The resulting rightmost mesh has no hanging nodes and all marked elements are bisected at least once.

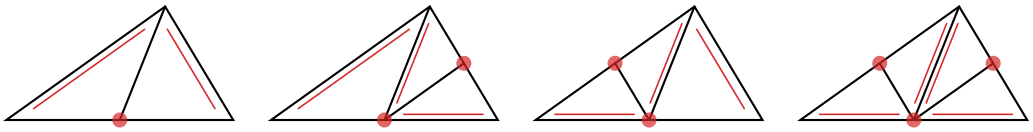


Figure 1.3: All possible bisections of one element that can occur during a call of Algorithm 1A. The refinement edge of the parent triangle is assumed to be the bottom edge. Marked edges, i.e., edges in $\mathcal{U}_H$, and refinement edges of the children are highlighted in red.

Algorithm 1A: Newest vertex bisection

Input: Mesh $\mathcal{T}_H$ and set of marked elements $\mathcal{M}_H \subseteq \mathcal{T}_H$

(i) Define $\mathcal{U}_H := \emptyset$ and $\mathcal{C}_H := \{\mathrm{re}(T) \,|\, T \in \mathcal{M}_H\}$

(ii) While $\mathcal{C}_H \neq \emptyset$ do

(1) $\mathcal{U}_H := \mathcal{U}_H \cup \mathcal{C}_H$

(2) $\mathcal{C}_H := \big\{\mathrm{re}(T) \,\big|\, T \in \mathcal{T}_H : \exists E \in \mathcal{U}_H \text{ with } E \subset T\big\} \setminus \mathcal{U}_H$

(iii) Refine elements such that all edges in $\mathcal{U}_H$ are bisected according to Figure 1.3

Output: Refined mesh $\mathcal{T}_h$

We write $\mathcal{T}_h = \texttt{refine}(\mathcal{T}_H, \mathcal{M}_H)$. The inner loop, Algorithm 1A(ii), is called *closure step* and ensures that no hanging nodes are left in the refined mesh. Therein, edge refinement is propagated through the mesh by successively marking refinement edges of elements with hanging nodes to obtain $\mathcal{C}_H$; see Figure 1.2. After that, all elements are refined at once according to their edges marked for bisection in $\mathcal{U}_H$. We note that Algorithm 1A is guaranteed to terminate for all configurations of refinement edges in the initial mesh [KPP13].

We further write $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$ if $\mathcal{T}_h$ can be obtained from $\mathcal{T}_H$ by finitely many steps of NVB, i.e., there exist $\mathcal{T}_i$ and $\mathcal{M}_i \subseteq \mathcal{T}_i$ for $i = 1, \dots, N-1 \in \mathbb{N}$ with $\mathcal{T}_1 = \mathcal{T}_H$, $\mathcal{T}_N = \mathcal{T}_h$, and $\mathcal{T}_{i+1} = \texttt{refine}(\mathcal{T}_i, \mathcal{M}_i)$. Henceforth, we fix an initial mesh $\mathcal{T}_0$ of Ω and abbreviate $\mathbb{T} := \mathbb{T}(\mathcal{T}_0)$.

Generalization to higher dimensions

For higher dimensions, NVB is somewhat more involved than Algorithm 1A. In particular, in this case it is not straightforward how to choose the refinement edge of the child elements, since there is more than one edge opposite to the newest vertex. The strategy proposed in [Ste08] suggests that each simplex is assigned an additional integer *tag*, which increases when passing to child elements. The refinement edge is then chosen according to the value of this tag.

Furthermore, it is not yet known how to generalize the fundamental structure of Algorithm 1A (first propagating bisected edges through the mesh and then refining elements based on marked edges) to higher dimensions: For Algorithm 1A(iii), all possible refinement patterns of one element have to be known. In two dimensions, these are depicted in Figure 1.3; In higher dimensions, the number of possible refinements grows exponentially such that even for $d = 3$ no complete list analogous to Figure 1.3 is known. Hence, the NVB algorithm for higher dimensions is formulated as recursive algorithm that sequentially bisects suitable patches of elements [Ste08]. To guarantee the existence of such bisectable patches, the choice of refinement edges on the intitial triangulation cannot be arbitrary but has to satisfy some preconditions [BDD04].

We stress that, although NVB is more involved in dimensions $d \geq 3$, it is well-analyzed if the above modifications are taken into account; see [Ste08].

1.2.2 A posteriori error estimation

To estimate the local error of the approximations $u_H \approx u$ and $z_H \approx z$ (and, thus, by (1.17) also of $G(u_H)$), we employ so-called *residual error estimators* [Ver13]. These are based on an element-wise decomposition of the weak form; e.g., for the primal problem (1.3), we have that

$$0 = a(u_H, v_H) - F(v_H) = \sum_{T \in \mathcal{T}_H} \int_T \big[\boldsymbol{A}\nabla u_H \cdot \nabla v_H + \boldsymbol{b} \cdot \nabla u_H \, v_H + c u_H v_H - f v_H + \boldsymbol{f} \cdot \nabla v_H\big] \,\mathrm{d}x.$$

Since $u_H \in \mathcal{S}^p_0(\mathcal{T}_H)$ means that $u_H|_T$ is a polynomial and therefore smooth for every element $T \in \mathcal{T}_H$, we can use element-wise integration by parts to obtain that

$$0 = \sum_{T \in \mathcal{T}_H} \int_T \big[-\operatorname{div}(\boldsymbol{A}\nabla u_H + \boldsymbol{f}) + \boldsymbol{b} \cdot \nabla u_H + c u_H - f\big] v_H \,\mathrm{d}x + \sum_{T \in \mathcal{T}_H} \int_{\partial T \cap \Omega} \big[\boldsymbol{A}\nabla u_H + \boldsymbol{f}\big] \cdot \boldsymbol{n} \, v_H \,\mathrm{d}s,$$

where $\boldsymbol{n}$ is the outwards-facing unit normal vector on ∂T. Summarizing the contributions over element boundaries for each face and introducing the normal jump $[\![(\cdot)\cdot\boldsymbol{n}]\!]$ over faces, this motivates to introduce, for $v_H \in \mathcal{S}^p_0(\mathcal{T}_H)$ and $T \in \mathcal{T}_H$, the *primal indicator*

$$\begin{aligned}\eta_H(T, v_H)^2 := h_T^2 \, \| \operatorname{div}(\boldsymbol{A}\nabla v_H + \boldsymbol{f}) - \boldsymbol{b} \cdot \nabla v_H - c v_H + f \|^2_{L^2(T)} \\ + h_T \, \| [\![(\boldsymbol{A}\nabla v_H + \boldsymbol{f}) \cdot \boldsymbol{n}]\!] \|^2_{L^2(\partial T \cap \Omega)}.\end{aligned} \tag{1.19a}$$

Analogously, the weak dual problem (1.14) leads to the *dual indicator*

$$\begin{aligned}\zeta_H(T, v_H)^2 := h_T^2 \, \| \operatorname{div}(\boldsymbol{A}\nabla v_H + \boldsymbol{g}) + \boldsymbol{b} \cdot \nabla v_H + (\operatorname{div} \boldsymbol{b} - c) v_H + g \|^2_{L^2(T)} \\ + h_T \, \| [\![(\boldsymbol{A}\nabla v_H + \boldsymbol{g}) \cdot \boldsymbol{n}]\!] \|^2_{L^2(\partial T \cap \Omega)}.\end{aligned} \tag{1.19b}$$

Note that, in order for the L^2-norms in (1.19) to be well-defined, we need the additional assumptions that $\boldsymbol{b} \in H(\mathrm{div};\Omega)$, and $\boldsymbol{f}|_T, \boldsymbol{g}|_T \in H(\mathrm{div};T)$ as well as $\boldsymbol{f}|_{\partial T}, \boldsymbol{g}|_{\partial T} \in L^2(\partial T)$ for all $T \in \mathcal{T}_H$. This can be ensured, e.g., by choosing $\boldsymbol{b}$ constant and $\boldsymbol{f}, \boldsymbol{g}$ as $\mathcal{T}_0$-piece-wise polynomials.

On a subset $\mathcal{U}_H \subseteq \mathcal{T}_H$, we define the estimator to be the quadratic sum of the element contributions, i.e.,

$$\eta_H(\mathcal{U}_H, v_H)^2 := \sum_{T\in\mathcal{U}_H} \eta_H(T, v_H)^2, \qquad \zeta_H(\mathcal{U}_H, v_H)^2 := \sum_{T\in\mathcal{U}_H} \zeta_H(T, v_H)^2.$$

We further abbreviate $\eta_H(v_H) := \eta_H(\mathcal{T}_H, v_H)$, $\eta_H(\mathcal{U}_H) := \eta_H(\mathcal{U}_H, u_H)$, and $\eta_H := \eta_H(\mathcal{T}_H)$, as well as the corresponding quantities for ζ_H and z_H.

The error estimators serve as an *a posteriori* error bound for the error in the energy norm, i.e.,

$$\|\!|u - u_H|\!\| \lesssim \eta_H \quad \text{and} \quad \|\!|z - z_H|\!\| \lesssim \zeta_H \qquad \text{for all } \mathcal{T}_H \in \mathbb{T}, \tag{1.20}$$

where the hidden constant is, in particular, independent of the mesh $\mathcal{T}_H$. This estimate is also called *reliability* and plays a vital role in the convergence analysis further below. By means of (1.17), it also leads to an *a posteriori* estimate for the goal error:

$$|G(u) - G(u_H)| \lesssim \eta_H\, \zeta_H. \tag{1.21}$$

We stress that the quantities (1.19) and, hence, also the upper bound for the goal error can be computed easily once the Galerkin solutions u_H and z_H are known.

The advantage of residual error estimators lies in their exquisite analytical properties (see Section 1.3). However, alternative approaches such as, most notably, the DWR method, which uses the primal *a posteriori* estimator η_H with local weights based on the approximate dual solution z_H to drive adaptive algorithms, also yield good results in practice without extensive convergence analysis; in particular, for nonlinear equations [BR01].

1.2.3 Marking

Elements in a mesh are marked according to their error indicators (1.19). The most common marking strategy in the mathematical literature concerning AFEM is the *bulk chasing criterion* introduced by Dörfler [Dör96]. In case of only primal error indicators and for a marking parameter $0 < \theta \leq 1$, it asks for a set $\mathcal{M}_H \subseteq \mathcal{T}_H$ of (quasi-)minimal cardinality such that

$$\theta\, \eta_H^2 \leq \eta_H(\mathcal{M}_H)^2, \tag{1.22}$$

i.e., the set $\mathcal{M}_H$ is responsible for at least the fraction θ of the error estimator on the whole mesh. This criterion is also known as *Dörfler marking* throughout the literature. The phrase *quasi-minimal cardinality* means that, with a fixed constant $C_{\mathrm{mark}} \geq 1$, there holds

$$\#\mathcal{M}_H \leq C_{\mathrm{mark}}\, \min\{\#\mathcal{U}_H \,|\, \mathcal{U}_H \subseteq \mathcal{T}_H \text{ with } \theta\, \eta_H^2 \leq \eta_H(\mathcal{U}_H)^2\},$$

where we denote by $\#\mathcal{M}_H$ the cardinality of the (finite) set $\mathcal{M}_H$. Note that this set is non-empty, i.e., $\mathcal{M}_H \neq \emptyset$, but not necessarily unique. The set $\mathcal{M}_H$ from (1.22) with arbitrary $C_{\mathrm{mark}} \geq 1$ can generally determined by sorting in almost linear complexity; an algorithm based on a binning technique can carry out marking in linear time, although with the restriction $C_{\mathrm{mark}} = 2$ [Ste07];

recently, an algorithm based on quick-selection has been proposed, which achieves linear complexity even for $C_{\mathrm{mark}} = 1$ [PP20].

Although there are other marking strategies present in the literature [DKS16; IP21; KS16; MSV08; Sie11], the Dörfler marking criterion has received the most attention because of its striking analytical features, which we exploit in the optimality analysis down below; in particular, for standard AFEM, Dörfler marking is sufficient (and in some sense even necessary, see Proposition 1.18 below) as a marking step to guarantee optimal convergence rates.

For GOAFEM, however, also the dual indicators have to be taken into account. The existing GOAFEM marking algorithms all comprise a step to combine the information of primal and dual indicators as well as a marking step, though they differ in the order of these steps. The first proposed marking strategy for GOAFEM appeared in the seminal paper [MS09] and carries out separate Dörfler marking before combining the information.

Algorithm 1B: MS marking

Input: Indicators $\eta_H(T)$ and $\zeta_H(T)$ for all $T \in \mathcal{T}_H$, marking parameter $0 < \theta \leq 1$

(i) Use Dörfler marking (1.22) for η_H and ζ_H to obtain sets $\mathcal{M}_H^u$ and $\mathcal{M}_H^z$, respectively

(ii) Choose $\mathcal{M}_H := \arg\min\{\#\mathcal{M}_H^u, \#\mathcal{M}_H^z\}$

Output: Set of marked elements $\mathcal{M}_H$

A variant, which proves more effective in numerical experiments was proposed in [FPZ16].

Algorithm 1C: FPZ marking

Input: Indicators $\eta_H(T)$ and $\zeta_H(T)$ for all $T \in \mathcal{T}_H$, marking parameter $0 < \theta \leq 1$

(i) Use Dörfler marking (1.22) for η_H and ζ_H to obtain sets $\mathcal{M}_H^u$ and $\mathcal{M}_H^z$, respectively

(ii) Set $n := \min\{\#\mathcal{M}_H^u, \#\mathcal{M}_H^z\}$

(iii) Choose $\overline{\mathcal{M}}_H^u \subseteq \mathcal{M}_H^u$ and $\overline{\mathcal{M}}_H^z \subseteq \mathcal{M}_H^z$ with $\#\overline{\mathcal{M}}_H^u = \#\overline{\mathcal{M}}_H^z = n$

(iv) Set $\mathcal{M}_H := \overline{\mathcal{M}}_H^u \cup \overline{\mathcal{M}}_H^z$

Output: Set of marked elements $\mathcal{M}_H$

The heuristic reason why Algorithm 1C seems to perform better in practice than Algorithm 1B is that the latter must choose between marking for primal or dual problem. The former, however, additionally marks elements with large indicator for the respective other problem. Thus, the number of elements is expected to grow faster with Algorithm 1C without deteriorating convergence rates.

Such a consideration is also the starting point of the following marking strategy from [BET11], which reverses the order of marking and combination step. The idea is to first weight the primal and dual indicators by the respective other indicators to obtain a new weighted error estimator, for which Dörfler marking is employed.

Algorithm 1D: BET marking

Input: Indicators $\eta_H(T)$ and $\zeta_H(T)$ for all $T \in \mathcal{T}_H$, marking parameter $0 < \theta \leq 1$

(i) Define the weighted error estimator $\varrho_H(T)^2 := \eta_H(T)^2\zeta_H^2 + \eta_H^2\zeta_H(T)^2$.

(ii) Use Dörfler marking (1.22) for ϱ_H to obtain the set of marked elements $\mathcal{M}_H$

Output: Set of marked elements $\mathcal{M}_H$

For a very simple model problem, it is shown in [BET11] that Algorithm 1D performs better than Algorithm 1B in the sense that it leads to a larger contraction per step. This advantage is also observed in practice for more general problems. However, the "price" to pay for this is that the marking parameter must be chosen smaller to ensure optimality; see Theorem 1.10 below. This is not a mere theoretical artifact: e.g., for point evaluations in BEM computations, lower convergence rates for large marking parameters θ have been observed in [FGH+16].

Remark 1.5. *The work [BET11] offers an interesting point of view on Algorithms 1B–1D: First, observe that, for $\eta_H > 0$, the Dörfler marking criterion (1.22) can also be written as*

$$\theta \leq \frac{\eta_H(\mathcal{M}_H)^2}{\eta_H^2}.$$

The assumption $\eta_H > 0$ is not at all restrictive, since $\eta_H = 0$ and reliability (1.20) already imply $|||u - u_H||| = 0$. Then, under the assumption $\eta_H, \zeta_H > 0$, one can reformulate Algorithm 1B as finding $\mathcal{M}_H \subseteq \mathcal{T}_H$ with minimal cardinality such that

$$\theta \leq \max\left\{\frac{\eta_H(\mathcal{M}_H)^2}{\eta_H^2}, \frac{\zeta_H(\mathcal{M}_H)^2}{\zeta_H^2}\right\}. \tag{1.23}$$

Analogously, Algorithm 1D can be reformulated (actually, this is the original formulation) as

$$\theta \leq \frac{1}{2}\left(\frac{\eta_H(\mathcal{M}_H)^2}{\eta_H^2} + \frac{\zeta_H(\mathcal{M}_H)^2}{\zeta_H^2}\right).$$

Hence, both algorithms are just different realizations of some mean between the primal and the dual estimator ratio. This creates a unified theoretical frame for both algorithms. From this viewpoint, also Algorithm 1C takes the form (1.23) with $C_{\mathrm{mark}} = 2$, although the additional elements are not chosen arbitrarily, but in a clever way.

1.2.4 Adaptive algorithm

Finally, we are able to write down the full adaptive algorithm for model problem (1.2) with linear goal functional (1.13), which concretizes the abstract adaptive loop (1.1).

Algorithm 1E: GOAFEM

Input: Initial mesh $\mathcal{T}_0$, marking parameter $0 < \theta \leq 1$
Loop: For all $\ell = 0, 1, \ldots$ do

Solve Solve (1.3) and (1.14) on $\mathcal{T}_\ell$ to obtain u_ℓ and z_ℓ, respectively

ESTIMATE Compute refinement indicators $\eta_\ell(T)$ and $\zeta_\ell(T)$ from (1.19) for all $T \in \mathcal{T}_\ell$

MARK Obtain marked elements $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ from Algorithm 1B, 1C, or 1D

REFINE Compute $\mathcal{T}_{\ell+1} := \texttt{refine}(\mathcal{T}_\ell, \mathcal{M}_\ell)$ by NVB (Algorithm 1A)

Output: Solutions u_ℓ, z_ℓ, approximate goal values $G(u_\ell)$, and goal error estimates $\eta_\ell\zeta_\ell$ for all $\ell \in \mathbb{N}_0$

We note that the adaptive algorithm for GOAFEM is very similar to the one of standard (i.e., not goal-oriented) AFEM. Apart from the lack of a dual solution z_ℓ and the corresponding refinement indicator $\zeta_\ell(T)$, the main difference is the marking step, which usually is (1.22) for AFEM.

Of course, Algorithm 1E cannot be iterated indefinitely in practice. Usually, one stops as soon as all computational resources are depleted or the upper bound for the goal error $\eta_\ell\zeta_\ell$ falls below some small given threshold $\tau > 0$. For the latter stopping criterion to work for every threshold, one needs at least convergence $\lim_{\ell\to\infty} \eta_\ell\zeta_\ell = 0$. We go into detail about convergence of Algorithm 1E in the following section, where a much stronger convergence result is shown.

Remark 1.6. *Because of the upper bound of the goal error presented in (1.11), instead of Algorithm 1E one can use a standard AFEM, e.g., as described in [CFPP14], that is driven only by the primal estimator η_ℓ or the dual estimator ζ_ℓ. Also, by using the Young inequality on (1.21), one can use a standard AFEM driven by the product space estimator $(\eta_\ell^2 + \zeta_\ell^2)^{1/2}$ as presented in Algorithm 2B. However, both approaches lead to sub-optimal rates compared to that of Algorithm 1E; see Chapter 2 and [MS09] for details.*

1.3 Optimal convergence of GOAFEM

The standard *a priori* error estimates (1.10) and (1.18) involve the *global* mesh width. From these, convergence follows only if the mesh becomes successively smaller in a uniform way, since this guarantees a decrease of the global mesh width, i.e., $H \to 0$. However, most adaptive FEM algorithms do not guarantee that H gets arbitrarily small. Instead, the convergence analysis has to rely on the *a posteriori* estimates (1.20) and (1.17) as well as other tools that are introduced in this section.

These tools presented here were originally developed for standard AFEM; see, e.g., [BDD04; CKNS08; Dör96; MNS00; Ste07] for some seminal contributions, or [CFPP14] for a survey within an abstract framework. For GOAFEM, results are much more scarce. In the works [BET11; MS09], the foundation was laid by transferring the AFEM analysis to the goal-oriented setting for some simple model problems. The works [FGH+16; FPZ16] (and [HP16], where only linear convergence is shown) then treated more general problems, but still lagged behind the literature on standard AFEM at that time. Nonlinear problems were, so far, only considered in [HPZ15; XHYM22], but no optimality result is proven therein.

We begin by stating here the optimality result of GOAFEM for (1.2), which goes way beyond mere convergence. Subsequently, we outline the main steps of its proof as given in [CFPP14; FPZ16], which is important for the understanding of the following chapters.

1.3.1 Necessary abstract properties

Although we are only concerned with standard residual error estimators as well as newest vertex bisection in this thesis, we state the abstract properties necessary for the convergence analysis in the spirit of [CFPP14; FPZ16]. Note that, differing from the presentation in the cited works, we divide the properties into three categories: those which involve the estimator, those which involve the mesh, and one which depends only on the PDE problem. This is due to simplifications that arise from our model problem being less general than the one in [CFPP14]. In the following, we state these properties in order.

Estimator properties

Let $\mathcal{T}_H \in \mathbb{T}$ and $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$. The error estimators (1.19) satisfy the following set of properties termed *(estimator) axioms* with constants $C_{\mathrm{stab}}, C_{\mathrm{rel}}, C_{\mathrm{drel}} > 0$ and $0 < q_{\mathrm{red}} < 1$ [CFPP14; FPZ16; Ver13]. We stress that these constants are, in particular, independent of the meshes $\mathcal{T}_H$ and $\mathcal{T}_h$.

(A1) Stability: For all $v_h \in \mathcal{X}_h$, $v_H \in \mathcal{X}_H$, and $\mathcal{U}_H \subseteq \mathcal{T}_h \cap \mathcal{T}_H$, it holds that

$$\big|\eta_h(\mathcal{U}_H, v_h) - \eta_H(\mathcal{U}_H, v_H)\big| + \big|\zeta_h(\mathcal{U}_H, v_h) - \zeta_H(\mathcal{U}_H, v_H)\big| \leq C_{\mathrm{stab}} \,|\!|\!| v_h - v_H |\!|\!|.$$

(A2) Reduction: For all $v_H \in \mathcal{X}_H$, it holds that

$$\eta_h(\mathcal{T}_h \backslash \mathcal{T}_H, v_H) \leq q_{\mathrm{red}}\, \eta_H(\mathcal{T}_H \backslash \mathcal{T}_h, v_H) \quad \text{and} \quad \zeta_h(\mathcal{T}_h \backslash \mathcal{T}_H, v_H) \leq q_{\mathrm{red}}\, \zeta_H(\mathcal{T}_H \backslash \mathcal{T}_h, v_H).$$

(A3) Reliability: The Galerkin solutions $u_H, z_H \in \mathcal{X}_H$ satisfy that

$$|\!|\!| u - u_H |\!|\!| \leq C_{\mathrm{rel}}\, \eta_H(u_H) \quad \text{and} \quad |\!|\!| z - z_H |\!|\!| \leq C_{\mathrm{rel}}\, \zeta_H(z_H).$$

(A4) Discrete reliability: The Galerkin solutions $u_H, z_H \in \mathcal{X}_H$ and $u_h, z_h \in \mathcal{X}_h$ satisfy that

$$|\!|\!| u_h - u_H |\!|\!| \leq C_{\mathrm{drel}}\, \eta_H(\mathcal{T}_H \backslash \mathcal{T}_h, u_H) \quad \text{and} \quad |\!|\!| z_h - z_H |\!|\!| \leq C_{\mathrm{drel}}\, \zeta_H(\mathcal{T}_H \backslash \mathcal{T}_h, z_H).$$

Stability (A1) states that, on elements which are not refined during the transition from $\mathcal{T}_H$ to $\mathcal{T}_h$, the refinement indicators are uniformly Lipschitz continuous in the function argument. Reduction (A2), on the other hand, states that, on refined elements, the error estimator uniformly contracts. These two properties were first introduced in [CKNS08]. Furthermore, (discrete) reliability (A3)–(A4) states that the approximation error of the Galerkin solution can be uniformly bounded by the error estimator (on refined elements). While reliability is a fundamental property in *a posteriori* error estimation [AO93; Ver13], discrete reliability was first introduced in [Ste07].

Remark 1.7. *Note that the estimator axioms (A1)–(A4) are slightly more general than what is needed for the convergence analysis below. In particular, property (A3) is redundant, since it follows from the a priori approximation property (1.10) and (A4); see [CFPP14, Lemma 3.4]. However, (linear) convergence can be proved without (A4) so that it is advantageous to state the property (A3), which is usually easier to prove, on its own.*

Mesh refinement properties

Newest vertex bisection from Section 1.2.1 for $d \geq 2$ guarantees the following properties [BDD04; CKNS08; Ste07; Ste08]. Here, for two meshes $\mathcal{T}, \mathcal{T}' \in \mathbb{T}$, the *coarsest common refinement* $\mathcal{T} \oplus \mathcal{T}'$ is defined as $\mathcal{T}_\star \in \mathbb{T}(\mathcal{T}) \cap \mathbb{T}(\mathcal{T}')$ with minimal cardinality $\#\mathcal{T}_\star$.

(R1) Child estimate: For all $\mathcal{T}_H \in \mathbb{T}$ and all $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, it holds that

$$\#(\mathcal{T}_H \backslash \mathcal{T}_h) + \#\mathcal{T}_H \leq \#\mathcal{T}_h.$$

(R2) Overlay estimate: For all $\mathcal{T}_H \in \mathbb{T}$ and all $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, it holds that

$$\#(\mathcal{T}_H \oplus \mathcal{T}_h) \leq \#\mathcal{T}_H + \#\mathcal{T}_h - \#\mathcal{T}_0.$$

(R3) Closure estimate: There exists a constant $C_{\mathrm{cls}} > 0$ such that, for the sequence $(\mathcal{T}_\ell)_{\ell \in \mathbb{N}_0}$ generated by Algorithm 1E, it holds that

$$\#\mathcal{T}_\ell - \#\mathcal{T}_0 \leq C_{\mathrm{cls}} \sum_{k=0}^{\ell-1} \#\mathcal{M}_k \quad \text{for all } \ell \in \mathbb{N}_0.$$

These properties, in essence, limit the number of elements that are generated by refinement, the coarsest common refinement (also called *overlay*), and mesh closure (i.e., additionally refining elements to ensure conformity), respectively. So far, newest vertex bisection is the only known refinement strategy for simplices that satisfies (R1)–(R3) in any dimension $d \geq 2$.

Remark 1.8. *The overlay estimate* (R2) *essentially goes back to [Ste07]. The closure estimate* (R3)*, on the other hand, was first proved for NVB in [BDD04] for $d = 2$ and later generalized to $d \geq 3$ by [Ste08]. Both proofs rely on an admissibility condition for $\mathcal{T}_0$, which was later removed for $d = 2$ in [KPP13].*

Remark 1.9. *Note that, even though uniform γ-shape regularity* (1.5) *of the family $\mathbb{T}$ for a fixed constant $\gamma > 0$ is an important property of newest vertex bisection, it does not occur in the list above. However, γ-shape regularity enters in the proofs of the estimator axioms* (A1)–(A4) *for the residual error estimators* (1.19)*. It is therefore also necessary in our analysis.*

Quasi-orthogonality

The last property for the convergence proof below is the so-called *quasi-orthogonality*.

(QO) Quasi-orthogonality: Let $(\mathcal{T}_\ell)_{\ell \in \mathbb{N}_0}$ be the sequence of meshes generated by Algorithm 1E and $n \in \mathbb{N}_0$. For all $0 < \varepsilon < 1$, there exists $\ell_0 \in \mathbb{N}_0$ such that, for all $\ell \geq \ell_0$ and all $n \in \mathbb{N}_0$,

$$\begin{aligned} \|\!| u - u_{\ell+n} |\!\|^2 + \|\!| u_{\ell+n} - u_\ell |\!\|^2 &\leq \frac{1}{1-\varepsilon} \|\!| u - u_\ell |\!\|^2, \\ \|\!| z - z_{\ell+n} |\!\|^2 + \|\!| z_{\ell+n} - z_\ell |\!\|^2 &\leq \frac{1}{1-\varepsilon} \|\!| z - z_\ell |\!\|^2. \end{aligned}$$

We note that, in prospect of the situation in the following chapters, we formulate a more general version of (QO) than is actually needed in the setting of this introductory chapter. In fact, if we suppose that $a(\cdot,\cdot)$ is symmetric and $|||v|||^2 = a(v,v)$, from the Galerkin orthogonalities (1.16) and $u_{\ell+n} - u_\ell, z_{\ell+n} - z_\ell \in \mathcal{T}_\ell$, there follow the Pythagoras identities

$$|||u - u_{\ell+n}|||^2 + |||u_{\ell+n} - u_\ell|||^2 = |||u - u_\ell|||^2 \quad \text{and} \quad |||z - z_{\ell+n}|||^2 + |||z_{\ell+n} - z_\ell|||^2 = |||z - z_\ell|||^2.$$

These are actual orthogonalities and, in particular, imply (QO) with $\ell_0 = 0$.

1.3.2 Rate optimality

To state the convergence result of goal-oriented adaptive FEM, we need additional notation.

Approximation classes

For $N \in \mathbb{N}_0$, we denote the finite set of all meshes that have at most N elements more than $\mathcal{T}_0$ by

$$\mathbb{T}(N) := \big\{\mathcal{T} \in \mathbb{T} \,\big|\, \#\mathcal{T} - \#\mathcal{T}_0 \le N\big\}.$$

For $s, t > 0$, we then introduce the so-called *rate-approximability* for the primal and dual problem:

$$\|u\|_{\mathbb{A}_s} := \sup_{N\in\mathbb{N}_0} (N+1)^s \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(N)} \eta_{\mathrm{opt}}(u_{\mathrm{opt}}) \in [0,\infty], \tag{1.24a}$$

$$\|z\|_{\mathbb{A}_t} := \sup_{N\in\mathbb{N}_0} (N+1)^t \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(N)} \zeta_{\mathrm{opt}}(z_{\mathrm{opt}}) \in [0,\infty]. \tag{1.24b}$$

A short explanation on the definitions (1.24) is in order. In both expressions, the minimum finds the mesh with lowest error estimator among all meshes below a certain number of elements. The corresponding (minimal) error estimator is then multiplied by a prefactor that grows with an algebraic rate s and t, respectively, over which the supremum is taken. This means that, if the error estimators on sequences of optimal meshes fall (at least) with rate s and t with respect to the number of elements, the rate-approximabilities $\|u\|_{\mathbb{A}_s}$ and $\|z\|_{\mathbb{A}_t}$ are finite. Thus, the rate-approximabilities measure the ability of the model problem to be numerically approximated by FEM on the family $\mathbb{T}$ of meshes. The set of all functions that have a finite (primal or dual) rate-approximability is sometimes called *approximation class*.

It is also easy to see that the converse is true [GP22]: Consider, e.g., the primal problem and let $s > 0$ with $\|u\|_{\mathbb{A}_s} < \infty$. Then, for each $\ell \in \mathbb{N}_0$, let $\mathcal{T}_\ell \in \mathbb{T}(\ell)$ be the mesh such that $\eta_\ell = \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(\ell)} \eta_{\mathrm{opt}}(u_{\mathrm{opt}})$. From the equivalence $\#\mathcal{T}_\ell \simeq \#\mathcal{T}_\ell - \#\mathcal{T}_0 + 1$ and the definition of the rate-approximability, we see that the estimator indeed falls with rate s along the sequence $\mathcal{T}_\ell$, i.e., with a constant $C(s) > 0$ it holds that

$$\eta_\ell \le C(s)\,(\#\mathcal{T}_\ell)^{-s} \quad \text{for all } \ell \in \mathbb{N}_0. \tag{1.25}$$

Statement of optimality

With these approximation classes, we can state the main result of GOAFEM: convergence of Algorithm 1E with optimal algebraic rates. The first result of this kind for GOAFEM goes back to [MS09], where a Poisson problem is considered. To cover our general model problem (1.2), we

present the corresponding result from [FPZ16]. Recall that the mesh refinement in our GOAFEM algorithm satisfies (R1)–(R3) and the error estimators satisfy (A1)–(A4) as well as (QO) for both the primal and the dual problem.

Theorem 1.10: Optimal convergence rates of Algorithm 1E [FPZ16, Theorem 13]

Suppose (R1)–(R3) *and* (A1)–(A4) *as well as* (QO). *Let* $(\mathcal{T}_\ell)_{\ell\in\mathbb{N}_0}$ *be the sequence of meshes generated by Algorithm 1E with one of the following marking strategies:*

- $0 < \theta < \theta_{\mathrm{opt}} := (1 + C_{\mathrm{stab}}^2 C_{\mathrm{rel}}^2)^{-1}$ *and either Algorithm 1B or Algorithm 1C;*
- $0 < \theta < \theta_{\mathrm{opt}}/2$ *and Algorithm 1D.*

Then, there exist constants $C_{\mathrm{opt}} > 0$ *and* $0 < q_{\mathrm{opt}} < 1$ *such that, for all* $s, t > 0$ *with* $\|u\|_{\mathbb{A}_s} + \|z\|_{\mathbb{A}_t} < \infty$, *there holds*

$$|G(u) - G(u_H)| \lesssim \eta_\ell \zeta_\ell \le \frac{C_{\mathrm{opt}}^{1+s+t}}{\big(1 - q_{\mathrm{opt}}^{1/(s+t)}\big)^{s+t}} \, \|u\|_{\mathbb{A}_s} \, \|z\|_{\mathbb{A}_t} \, \big(\#\mathcal{T}_\ell - \#\mathcal{T}_0\big)^{-(s+t)}, \tag{1.26}$$

i.e., Algorithm 1E asymptotically drives down the estimator product with any possible algebraic rate. The constant q_{opt} *depends only on* (A1)–(A4), (QO), *and* θ, *whereas the constant* C_{opt} *additionally depends on* C_{cls} *from* (R3), *and* C_{mark}.

Theorem 1.10 states that Algorithm 1E is optimal in the following sense: As remarked in (1.25), the finiteness of the rate-approximabilities $\|u\|_{\mathbb{A}_s} + \|z\|_{\mathbb{A}_t} < \infty$ implies the existence of sequences $(\mathcal{T}_{\ell,u})_{\ell\in\mathbb{N}_0}, (\mathcal{T}_{\ell,z})_{\ell\in\mathbb{N}_0} \subset \mathbb{T}$ such that

$$\eta_{\ell,u} \lesssim (\#\mathcal{T}_{\ell,u})^{-s} \quad \text{and} \quad \zeta_{\ell,z} \lesssim (\#\mathcal{T}_{\ell,z})^{-t} \qquad \text{for all } \ell \in \mathbb{N}_0. \tag{1.27}$$

Thus, the *a posteriori* goal error estimate (1.21) suggests that the best possible decay of the goal error along some optimal sequence of meshes $(\mathcal{T}_{\ell,\mathrm{opt}})_{\ell\in\mathbb{N}_0} \subset \mathbb{T}$ is

$$|G(u) - G(u_{\ell,\mathrm{opt}})| \lesssim (\#\mathcal{T}_{\ell,\mathrm{opt}})^{-(s+t)} \quad \text{for all } \ell \in \mathbb{N}_0. \tag{1.28}$$

We stress that this is a purely heuristic argument, since (1.27) does not necessarily imply (1.28), for the optimal sequences in (1.27) do not need to coincide. However, Theorem 1.10 states that already the sequence generated by Algorithm 1E satisfies (1.28). Thus, if a rate is possibly attainable, Algorithm 1E drives down the goal error with that rate. This is the meaning of *rate optimality*.

Almost optimal rates of single estimators

Finally, we repeat here an observation from [FPZ16], which states that, although Algorithm 1E is rate optimal with repect to the goal error, i.e., it drives down the estimator product with every possible rate, it is only almost rate optimal with respect to the single primal and dual error estimator.

Corollary 1.11 ([FPZ16, Corollary 14])**.** *Assume that*

$$\overline{s} := \sup\{s > 0 \,|\, \|u\|_{\mathbb{A}_s} < \infty\} < \infty \quad \textit{and} \quad \overline{t} := \sup\{t > 0 \,|\, \|z\|_{\mathbb{A}_t} < \infty\} < \infty.$$

Then, for any $0 < s < \overline{s}$ *and* $0 < t < \overline{t}$ *there exist subsequences* $(\ell_k^u)_{k\in\mathbb{N}_0}, (\ell_k^z)_{k\in\mathbb{N}_0} \subseteq \mathbb{N}$ *such that*

$$\eta_{\ell_k^u} \lesssim (\#\mathcal{T}_{\ell_k^u})^{-s} \quad \text{and} \quad \zeta_{\ell_k^z} \lesssim (\#\mathcal{T}_{\ell_k^z})^{-t} \qquad \text{for all } k \in \mathbb{N}_0.$$

The hidden constants depend on C_{opt}, q_{opt}, *and additionally on* $\overline{s} - s > 0$ *and* $\overline{t} - t > 0$, *respectively.*

The non-attainability of (1.28) by mere construction of overlays, as well as the statement of Corollary 1.11 make it clear that Algorithm 1E is a non-trivial extension of the theory for standard AFEM.

1.3.3 Main steps of proof

We proceed in this section to outline the proof of Theorem 1.10. Although this is not original research, the benefits of retracing the main steps of the proof far outweigh those of a shorter introduction. First, Theorem 1.10 as well as its proof act as a baseline for the results presented in Chapters 2–4, where the framework of the present chapter is substantially extended. Second, certain subsets of the assumptions needed for Theorem 1.10 already yield plain convergence or linear convergence of the underlying GOAFEM, which are interesting results on their own. These subsets will become clear in the course of this section. Third, our model problem admits a certain amount of simplification and relaxation of assumptions for the results from [FPZ16], which are stated but somewhat hidden in [CFPP14]; these are also collected here.

Plain convergence

We start by outlining the steps that lead to convergence of Algorithm 1E with respect to the estimator product, which, by (1.21), is an upper bound for the goal error. The first ingredient is a quasi-contraction property of error estimators that holds as soon as Dörfler marking is employed, which goes essentially back to [CKNS08].

Proposition 1.12 (Generalized estimator reduction [FPZ16, Lemma 9])**.** *Suppose* (A1)–(A2). *Let* $\mathcal{T}_H \in \mathbb{T}$ *and* $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$. *Then, for all* $\delta > 0$, *there holds*

$$\eta_h^2 \le (1+\delta)\,\eta_H^2 + C_{\mathrm{stab}}(1+\delta^{-1})\,|||u_h - u_H|||^2. \tag{1.29}$$

Let additionally $0 < \theta \le 1$ *and* $\mathcal{T}_h \in \mathbb{T}\big(\texttt{refine}(\mathcal{T}_H, \mathcal{M}_H)\big)$, *where* $\mathcal{M}_H \subseteq \mathcal{T}_H$ *satisfies the Dörfler marking* (1.22). *Then, there exist constants* $C > 0$ *and* $0 < q < 1$ *such that*

$$\eta_h^2 \le q\,\eta_H^2 + C\,|||u_h - u_H|||^2. \tag{1.30}$$

The constants C, q *depend only on* (A1) *and* (A2) *as well as on* θ.

We further observe that the estimator cannot grow too much on refined meshes.

Proposition 1.13 (Quasi-monotonicity [FPZ16, Lemma 6])**.** *Suppose* (A1)–(A3). *Then, there exists a constant* $C_{\mathrm{mon}} > 0$ *such that, for all* $\mathcal{T}_H \in \mathbb{T}$ *and* $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, *there holds*

$$\eta_h \le C_{\mathrm{mon}}\,\eta_H. \tag{1.31}$$

The constant C_{mon} depends only on (A1)–(A3) as well as $C_{\mathrm{Céa}}$.

Proof. Starting from (1.29) with $\delta = 1$, the triangle inequality and the quasi-best approximation result (1.9) yield that

$$\eta_h^2 \le 2\,\eta_H^2 + 2C_{\mathrm{stab}}\,|||u_h - u_H|||^2 \le 2\,\eta_H^2 + 4C_{\mathrm{stab}}(1 + C_{\mathrm{Céa}})\,|||u - u_H|||^2.$$

Finally, reliability (A3) concludes the proof with $C_{\mathrm{mon}} = 2 + 4C_{\mathrm{stab}}C_{\mathrm{rel}}(1 + C_{\mathrm{Céa}})$. □

For standard AFEM, Propositions 1.12 and 1.13 already imply convergence of the error estimator, since there holds Dörfler marking in every step of the adaptive algorithm. For GOAFEM, on the other hand, Dörfler marking for primal and dual estimator in every step is not guaranteed; in fact, it cannot hold in general because of the minimality constraint on the cardinality of the set of marked elements. However, the following result, which can be seen immediately from the representation of the marking strategies in Remark 1.5, states that Dörfler marking holds at least for one of the estimators (with possibly modified marking parameter).

Proposition 1.14 (Dörfler marking for primal or dual estimator)**.** *Let $0 < \theta \le 1$.*

(i) *If $\mathcal{M}_H$ is the output of Algorithm 1B or 1C, there holds*

$$\theta\,\eta_H^2 \le \eta_H(\mathcal{M}_H)^2 \quad \text{or} \quad \theta\,\zeta_H^2 \le \zeta_H(\mathcal{M}_H)^2.$$

(ii) *If $\mathcal{M}_H$ is the output of Algorithm 1D, there holds*

$$\frac{\theta}{2}\,\eta_H^2 \le \eta_H(\mathcal{M}_H)^2 \quad \text{or} \quad \frac{\theta}{2}\,\zeta_H^2 \le \zeta_H(\mathcal{M}_H)^2.$$

(iii) *If $\eta_H\zeta_H \neq 0$ and $\mathcal{M}_H \subseteq \mathcal{T}_H$ satisfies*

$$\theta\,\eta_H^2 \le \eta_H(\mathcal{M}_H)^2 \quad \text{or} \quad \theta\,\zeta_H^2 \le \zeta_H(\mathcal{M}_H)^2,$$

there holds

$$\theta \le \max\left\{\frac{\eta_H(\mathcal{M}_H)^2}{\eta_H^2}, \frac{\zeta_H(\mathcal{M}_H)^2}{\zeta_H^2}\right\} \quad \text{and} \quad \frac{\theta}{2} \le \frac{1}{2}\left(\frac{\eta_H(\mathcal{M}_H)^2}{\eta_H^2} + \frac{\zeta_H(\mathcal{M}_H)^2}{\zeta_H^2}\right).$$

These results already imply plain convergence, i.e., convergence without any guaranteed rates, of the estimator product and, hence, also of the the goal error via (1.21).

Theorem 1.15: Plain convergence [GP22]

Suppose (A1)–(A3). Let $0 < \theta \le 1$ and let $(\mathcal{T}_\ell)_{\ell\in\mathbb{N}_0} \subseteq \mathbb{T}$ be the sequence of meshes generated by Algorithm 1E. Then, there holds plain convergence of the goal error, i.e.,

$$|G(u) - G(u_\ell)| \lesssim \eta_\ell\,\zeta_\ell \longrightarrow 0 \quad \text{as } \ell \to \infty. \tag{1.32}$$

The hidden constant depends only on C_{cnt} and (A1)–(A3).

Sketch of proof. Choose $\delta > 0$ in Proposition 1.12 small enough such that $q' := (1+\delta)q < 1$. Because of Proposition 1.14, the estimator product satisfies

$$\eta_{\ell+1}\,\zeta_{\ell+1} \le q'\,\eta_\ell\,\zeta_\ell + R_\ell,$$

with some remainder term

$$R_\ell \lesssim \|\!|u_{\ell+1} - u_\ell|\!|\,\zeta_\ell + \eta_\ell\,|\!|\!|z_{\ell+1} - z_\ell|\!|\!| + |\!|\!|u_{\ell+1} - u_\ell|\!|\!|\,|\!|\!|z_{\ell+1} - z_\ell|\!|\!|.$$

This upper bound is a null sequence because there holds *a priori* convergence $|\!|\!|u_{\ell+1} - u_\ell|\!|\!| \to 0$ as $\ell \to \infty$ due to the Céa Lemma (Theorem 1.4) as well as boundedness of the estimators, $\eta_\ell \le C_{\mathrm{mon}}\,\eta_0$, due to quasi-monotonicity (Proposition 1.13). We thus have $R_\ell \to 0$ as $\ell \to \infty$. As the estimator product is a contraction up to a null-sequence, convergence follows from basic calculus. □

Linear convergence

The last section shows that plain convergence only needs (A1)–(A3). For the "next level" of convergence, linear convergence, quasi-orthogonality (QO) enters the picture. Before we can make use of this additional assumption,

The next result states that primal as well as dual estimator contract each time Dörfler marking is employed for the respective problem.

Proposition 1.16 (Generalized linear convergence [FPZ16, Proposition 10])**.** *Suppose* (A1)–(A3) *and* (QO). *Let* $(\mathcal{T}_\ell)_{\ell\in\mathbb{N}_0} \subseteq \mathbb{T}$ *be the sequence of meshes generated by Algorithm 1E and let* $0 < \theta \le 1$. *Then, there exist constants* $C_{\mathrm{lin}} > 0$ *and* $0 < q_{\mathrm{lin}} < 1$ *such that there holds that*

$$\eta_{\ell+n} \le C_{\mathrm{lin}}\,q_{\mathrm{lin}}^{k_\eta}\,\eta_\ell \quad \text{and} \quad \zeta_{\ell+n} \le C_{\mathrm{lin}}\,q_{\mathrm{lin}}^{k_\zeta}\,\zeta_\ell \qquad \text{for all } \ell, n \in \mathbb{N}_0, \tag{1.33}$$

where k_ω *for* $\omega \in \{\eta, \zeta\}$ *is defined as the number of steps where Dörfler marking is satisfied for* ω *within* n *consecutive steps, i.e.,*

$$k_\omega := \#\big\{0 \le k \le n \,\big|\, \theta\,\omega_{\ell+k}^2 \le \omega_{\ell+k}(\mathcal{M}_{\ell+k})^2\big\} \quad \text{and} \quad k_\eta + k_\zeta \ge n.$$

The constants $C_{\mathrm{lin}}, q_{\mathrm{lin}}$ *depend only on* (A1)–(A3) *as well as on* ℓ_0 *from* (QO) *and* θ.

Sketch of proof. Let $(\ell_k)_{k=0}^{k_\eta}$ be the set of indices such that Dörfler marking occurs for η_{ℓ_k}. Since $\mathcal{T}_{\ell_{k+1}} \in \mathbb{T}(\mathcal{T}_{\ell_k+1})$, we can relate two successive estimators η_{ℓ_k} and $\eta_{\ell_{k+1}}$ by (1.30). We then sum this over all $k = 0, \dots, k_\eta$, absorb the estimators on the left-hand side, and use quasi-orthogonality (QO) on the sum over the energy norms. With a constant $C > 1$, this yields

$$\sum_{k=k_\eta-j}^{k_\eta} \eta_{\ell_{k+1}}^2 \le C\,\eta_{\ell_{k-j}}^2 \quad \text{for all } 0 \le j \le k_\eta.$$

Basic calculus shows linear convergence (1.33) between the ℓ_0-th and the ℓ_{k_η}-th mesh. We conclude by relating the ℓ_0-th to the ℓ-th and the ℓ_{k_η}-th to the $(\ell+n)$-th mesh by quasi-monotonicity (1.31). □

Finally, we can combine generalized linear convergences for the primal and the dual estimator to obtain full linear convergence.

Theorem 1.17: Linear convergence [FPZ16, Theorem 12]

Suppose (A1)–(A3) *and* (QO). *Let* $(\mathcal{T}_\ell)_{\ell\in\mathbb{N}_0} \subseteq \mathbb{T}$ *be the sequence of meshes generated by Algorithm 1E and let* $0 < \theta \leq 1$. *Then, for all* $\ell, n \in \mathbb{N}_0$ *there holds that*

$$\eta_{\ell+n}\,\zeta_{\ell+n} \leq C_{\mathrm{lin}}^2\, q_{\mathrm{lin}}^n\, \eta_\ell\,\zeta_\ell \tag{1.34}$$

with constants $C_{\mathrm{lin}} > 0$ *and* $0 < q_{\mathrm{lin}} < 1$ *from Proposition 1.16.*

Sketch of proof. From Proposition 1.16, we infer that there holds (1.33). Proposition 1.14 then implies that $k_\eta + k_\zeta \geq n$. Multiplying both inequalities in (1.33) finally yields (1.34). □

Proof of optimal rates Theorem 1.10

For the first building block of the proof of Theorem 1.10, we finally need the last of the estimator axioms, (A4). This first result states that any sufficient reduction of the error estimator already implies that Dörfler marking (1.22) holds. In that sense, Dörfler marking is the optimal marking criterion for AFEM. The first statement of this result can be found in [Ste07], where a sufficient contraction of the energy errors is assumed. The presented version, where contraction of the error estimators is assumed, was first presented in [CFPP14].

Proposition 1.18 (Optimality of Dörfler marking [FPZ16, Lemma 7])**.** *Suppose* (A1) *and* (A4). *Let* $0 < \theta < \theta_{\mathrm{opt}} := (1 + C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2)$. *There exists a constant* $0 < \kappa_{\mathrm{opt}} < 1$ *such that, for all* $\mathcal{T}_H \in \mathbb{T}$ *and* $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, *there holds*

$$\eta_h^2 \leq \kappa_{\mathrm{opt}}\,\eta_H^2 \quad\Longrightarrow\quad \theta\,\eta_H^2 \leq \eta_H(\mathcal{T}_H\backslash\mathcal{T}_h)^2, \tag{1.35a}$$

$$\zeta_h^2 \leq \kappa_{\mathrm{opt}}\,\zeta_H^2 \quad\Longrightarrow\quad \theta\,\zeta_H^2 \leq \zeta_H(\mathcal{T}_H\backslash\mathcal{T}_h)^2. \tag{1.35b}$$

The constant κ_{opt} *depends only on stability* (A1), *discrete reliability* (A4), *and* θ.

Sketch of proof. First, we split η_H^2 into refined and non-refined elements. For the non-refined elements, we use stability (A1), discrete reliability (A4), and the assumption $\eta_h^2 \leq \kappa_{\mathrm{opt}}\,\eta_H^2$. This yields the estimate

$$\big[1 - (1+\delta^{-1})\,\kappa_{\mathrm{opt}}\big]\eta_H^2 \leq \big(1 + (1+\delta)C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2\big)\,\eta_H(\mathcal{T}_H\backslash\mathcal{T}_h)^2 \quad \text{for all } \delta > 0.$$

Dividing by the prefactor of the right-hand side, we chose $\delta > 0$ small enough such that the resulting prefactor of the left-hand side is smaller than θ. This shows (1.35). □

Except from linear convergence of the error product, Theorem 1.17, every building block presented so far applies to the primal and dual problem separately. The next result, which states that there always exists a refinement with sufficient estimator reduction but not too many elements, makes an assertion about both problems together. It was first introduced in [Ste07] for standard AFEM and later generalized to GOAFEM in [MS09]. For the sake of presentation, we state here a version from [FGH+16]. Here, also mesh refinement enters the picture via the overlay estimate (R2).

Proposition 1.19 (Comparison lemma [FGH+16, Lemma 14]). *Suppose* (A1)–(A3) *and* (R2). *Let* $0 < \kappa < 1$ *and let* $(\mathcal{T}_\ell)_{\ell\in\mathbb{N}_0} \subseteq \mathbb{T}$ *be the sequence of meshes generated by Algorithm 1E. There exist constants* $C, C' > 0$ *such that the following holds: For every* $\ell \in \mathbb{N}_0$, *there exists a refinement* $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_\ell)$ *such that, for all* $s, t > 0$ *with* $\|u\|_{\mathbb{A}_s} + \|z\|_{\mathbb{A}_t} < \infty$, *there holds*

$$\eta_h^2\,\zeta_h^2 \le \kappa\,\eta_\ell^2\,\zeta_\ell^2, \tag{1.36}$$

$$\#\mathcal{T}_h - \#\mathcal{T}_\ell \le C'\,\big(C\,\|u\|_{\mathbb{A}_s}\|z\|_{\mathbb{A}_t}\big)^{1/(s+t)}\,\big(\eta_\ell\,\zeta_\ell\big)^{-1/(s+t)}. \tag{1.37}$$

The constants C, C' *only depend on* C_{mon}, $\mathcal{T}_0$, *and* (R2).

Sketch of proof. Define $\varepsilon := C_{\mathrm{mon}}^{-2}\,\kappa^{1/2}\,\eta_\ell\zeta_\ell$ and choose $N \in \mathbb{N}_0$ minimal such that $\|u\|_{\mathbb{A}_s}\|z\|_{\mathbb{A}_t} \le \varepsilon\,(N+1)^{s+t}$. Then, choose

$$\mathcal{T}_\omega = \arg\min\big\{\omega_H \,\big|\, \mathcal{T}_H \in \mathbb{T}(N)\big\} \quad \text{for } \omega \in \{\eta, \zeta\}.$$

Furthermore, define the overlays $\mathcal{T}_\varepsilon := \mathcal{T}_\eta \oplus \mathcal{T}_\zeta$ and $\mathcal{T}_h := \mathcal{T}_\varepsilon \oplus \mathcal{T}_\ell$. Then, the overlay estimate (R2) and minimality of N show that

$$\#\mathcal{T}_h - \#\mathcal{T}_\ell \lesssim N \lesssim (\eta_\ell\zeta_\ell)^{-1/(s+t)},$$

which gives (1.37). Moreover, quasi-monotonicity (1.31), the definition of the approximation classes (1.24), and the choice of N yield

$$\eta_h\,\zeta_h \le C_{\mathrm{mon}}^2\,(N+1)^{-(s+t)}\,\|u\|_{\mathbb{A}_s}\|z\|_{\mathbb{A}_t} \le C_{\mathrm{mon}}^2\varepsilon = \kappa^{1/2}\,\eta_\ell\zeta_\ell.$$

This shows (1.36) and thus concludes the proof. □

We have now stated every piece of the puzzle that is the proof of Theorem 1.10. The pieces neatly interlock each other; the dependency tree is shown in Figure 1.4.

Sketch of proof of Theorem 1.10. First, note that the assumptions imply $0 < \theta < \theta_{\mathrm{opt}}$ for Algorithms 1B–1C as well as $0 < 2\theta < \theta_{\mathrm{opt}}$ for Algorithm (1D). The proof consists of four steps in which the constant $0 < \kappa_{\mathrm{opt}} < 1$ from Proposition 1.18 corresponds to θ and 2θ for the different marking strategies, respectively.

Step 1: The mesh $\mathcal{T}_h$ from Proposition 1.19 is obtained by Dörfler marking.
Let $\kappa = \kappa_{\mathrm{opt}}^2$ in Proposition 1.19 with corresponding mesh $\mathcal{T}_h$. Then, it follows that

$$\eta_h^2\zeta_h^2 \le \kappa_{\mathrm{opt}}^2\,\eta_\ell^2\zeta_\ell^2 \qquad\Longrightarrow\qquad \Big[\eta_h^2 \le \kappa_{\mathrm{opt}}\,\eta_\ell^2 \quad\text{or}\quad \zeta_h^2 \le \kappa_{\mathrm{opt}}\,\zeta_\ell^2\Big].$$

We know from Proposition 1.18 that this implies Dörfler marking (1.22) with the set $\mathcal{T}_h\backslash\mathcal{T}_\ell$ either for the primal or for the dual problem with corresponding marking parameter. From Proposition 1.14, we then infer that the set $\mathcal{T}_\ell\backslash\mathcal{T}_h$ satisfies the used marking strategy from Algorithm 1B–1D, i.e.,

$$\theta \le \max\left\{\frac{\eta_\ell(\mathcal{T}_\ell\backslash\mathcal{T}_h)^2}{\eta_\ell^2}, \frac{\zeta_\ell(\mathcal{T}_\ell\backslash\mathcal{T}_h)^2}{\zeta_\ell^2}\right\} \quad\text{and}\quad \theta \le \frac{1}{2}\left(\frac{\eta_\ell(\mathcal{T}_\ell\backslash\mathcal{T}_h)^2}{\eta_\ell^2} + \frac{\zeta_\ell(\mathcal{T}_\ell\backslash\mathcal{T}_h)^2}{\zeta_\ell^2}\right), \tag{1.38}$$

respectively.

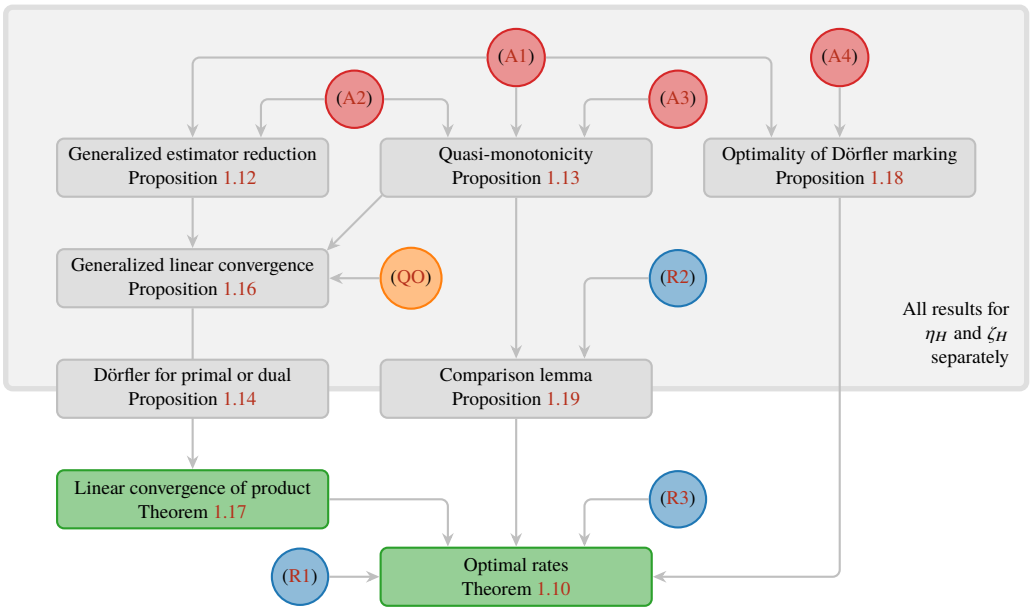


Figure 1.4: Basic structure of the proof of Theorem 1.10. Implications are indicated by arrows. Intermediate results in the large gray box are proven for primal and dual problem separately. The combining result is Proposition 1.14, which states that, in each step, Dörfler marking holds either for the primal or dual error estimator.

Step 2: Relate the sets $\mathcal{M}_\ell$ and $\mathcal{T}_\ell\backslash\mathcal{T}_h$ by quasi-minimal cardinality.
Recall that the marking strategies from Algorithms 1B–1D require quasi-minimal cardinality of the set of marked edges $\mathcal{M}_\ell$. Thus, (1.38) and the child estimate (R1) imply that

$$\#\mathcal{M}_\ell \le C_{\mathrm{mark}}\,\#(\mathcal{T}_\ell\backslash\mathcal{T}_h) \overset{\text{(R1)}}{\le} C_{\mathrm{mark}}\,\big(\#\mathcal{T}_h - \#\mathcal{T}_\ell\big).$$

Step 3: Use the mesh closure estimate to bound $\#\mathcal{T}_\ell - \#\mathcal{T}_0$ for all $\ell \in \mathbb{N}_0$.
From the mesh closure estimate (R3) and the comparison lemma Proposition 1.19, we see that, for all $\ell \in \mathbb{N}_0$,

$$\#\mathcal{T}_\ell - \#\mathcal{T}_0 \overset{\text{(R3)}}{\lesssim} \sum_{k=0}^{\ell-1} \#\mathcal{M}_k \overset{\text{(1.37)}}{\lesssim} \sum_{k=0}^{\ell-1} \big(\eta_k\,\zeta_k\big)^{-1/(s+t)}.$$

Step 4: Use linear convergence to bound the sum by the ℓ-th element.
Linear convergence Theorem 1.17 allows to bound the sum in the last expression to obtain

$$\#\mathcal{T}_\ell - \#\mathcal{T}_0 \lesssim \sum_{k=0}^{\ell-1} \big(\eta_k\,\zeta_k\big)^{-1/(s+t)} \overset{\text{(1.34)}}{\lesssim} \big(\eta_\ell\,\zeta_\ell\big)^{-1/(s+t)} \sum_{k=0}^{\ell-1} q_{\mathrm{lin}}^{k/(s+t)} \lesssim \big(\eta_\ell\,\zeta_\ell\big)^{-1/(s+t)}.$$

This immediately gives (1.26). □

1.4 Outline of thesis

The remainder of this thesis is concerned with extension, application, and numerical simulation of the setting presented in this introductory chapter. It can be roughly divided into two parts:

I. Chapters 2–4 are concerned with analytical aspects of GOAFEM.

II. Chapter 5 is concerned with implementational aspects in the high-level programming language MATLAB.

Each of the following chapters is dedicated to a specific research question that I, together with collaborators, have tackled during my PhD studies.

Chapter 2: GOAFEM with quadratic goal

The introductory framework of the present chapter assumes that the equation (1.2) as well as the goal functional (1.13) is linear. It is a natural extension to abandon the linearity assumption on either one of them. While nonlinearities in the equation have been analyzed in the context of GOAFEM [HPZ15; XHYM22] (but without convergence analysis in the sense of Section 1.3), nonlinearities in the goal functional have not been analyzed before in a clear mathematical framework. In particular, existing literature on GOAFEM with optimal rates was only concerned with linear problems as well as linear goal functionals [BET11; FGH+16; FPZ16; MS09].

In this chapter, we consider a goal functional that has a quadratic structure, i.e., there exists a bounded linear operator $\mathcal{K}\colon H^1_0(\Omega) \to H^{-1}(\Omega)$ such that

$$G(v) := \big\langle \mathcal{K}v\,,\, v\big\rangle_{H^{-1}\times H^1_0} \quad \text{for all } v \in H^1_0(\Omega).$$

This allows to consider goal functionals such as a (weighted) L^2-norm. In this context, many building blocks of the theory presented in the introduction have to be modified or completely replaced. In particular, the dual problem (1.12) would feature a nonlinear right-hand side and, hence, has to be reformulated using a suitable linearization of the goal functional G. This has also severe impacts on error estimation and marking.

We devise a GOAFEM algorithm that takes into account all the problems arising in this setting and prove that there holds plain convergence for every operator $\mathcal{K}$. Furthermore, under the additional assumption that $\mathcal{K}$ is compact, we show linear convergence of our algorithm and, hence, convergence with optimal algebraic rates, analogously to Theorem 1.10.

Chapter 3: GOAFEM with iterative solver

The present introduction does not go into detail about the numerical solution of the arising linear system (1.7). While, in principle, solving linear systems numerically is a well-trodden path, it is worthwhile to further investigate this topic in the context of (GO)AFEM because of two reasons. First, since solving the linear system is usually executed with finite precision arithmetic, only approximations to u_H, z_H are available for error estimation. It is *a priori* not clear that the analysis shown so far is robust with respect to these approximations. Second, and more importantly, if the linear system (1.7) is solved iteratively (as is the case in most applications with sufficiently many

degrees of freedom), the question arises when to stop this iterative solver without spoiling (optimal) convergence.

This last question is of particular importance in adaptive FEM algorithms, since it does not make sense to iterate the solution to machine precision if the mesh under consideration does not allow for a sufficiently good approximation in the first place. We note that inexact solution and the associated computational cost have already been considered in the seminal works [Ste07] for AFEM and [MS09] for GOAFEM. Therein, the authors consider a generic iterative solver (in the routine GALSOLVE) which improves the energy error of an initial guess by a factor $0 < \tau < 1$ at $\mathcal{O}(|\log(\tau)|)$ times linear cost. For instance, this is satisfied for contractive solvers like an optimally preconditioned CG method [CNX12] or an optimal geometric multigrid solver [WZ17]. However, the algorithms from [MS09; Ste07] require sufficiently small parameters (θ for mesh-refinement and ω for the solver accuracy) and linear convergence as well as optimal computational cost is then guaranteed for the final solver iterates only. We note that convergence for arbitrary θ but sufficiently small ω can be shown by a perturbation argument following, e.g., the works [CKNS08] for AFEM and [FPZ16] for GOAFEM.

In this chapter, we use ideas for AFEM from [GHPS18; GHPS21] and combine them with ideas from [FPZ16] for GOAFEM with exact solver. Essentially adapting the solver stopping criterion from [MS09; Ste07], we design a GOAFEM algorithm that also takes into account the iterative solution of the linear system (1.7) and steers the iterative solver such that the errors of the solver and the discretization are equilibrated. Under the natural assumption that the iterative solver is contractive, we show that the results presented in this introduction also hold true in the case of inexact solutions. More precisely, we even show full linear convergence, i.e., linear convergence of the estimator product—independently of the algorithmic decision for either mesh-refinement or solver step. In particular, this holds for arbitrary adaptivity parameters (θ for mesh-refinement and λ for the solver termination). Moreover, we note that this full linear convergence also implies that rates with respect to the number of elements coincide with the rates with respect to the cumulative computational costs.

As a consequence of the preceding arguments, if the adaptivity parameters are chosen sufficiently small, this implies that the proposed GOAFEM strategy leads to optimal rates (in the sense of Theorem 1.10) with respect to the overall computational costs for the full sequence of computed discrete solutions. This extends and transfers the results from [MS09] from the Poisson model problem to symmetric second-order linear elliptic PDEs and, in particular, avoids another loop for the adaptive approximation of the given loads as in [MS09].

Finally, we note that the idealized GOAFEM algorithm from [MS09] (using exact discrete solutions) relies on one mesh sequence only, while the practical GOAFEM algorithm from [MS09] (using an optimal iterative solver) can lead to separate (yet nested) meshes for primal and dual problem. One further advantage of the presented GOAFEM strategy might be that approximate primal and dual solutions are always computed on the same mesh (and therefore with the same solver), as is the case for the idealized algorithm from [MS09] and the succeeding algorithm from [FPZ16].

Chapter 4: Parameter estimation

In this chapter, we look at a non-straightforward application of goal-oriented FEM, namely estimation of a finite set of parameters from experimentally obtained data in an elliptic linear-quadratic

setting. Many PDE models from applications such as physics, engineering, and social sciences, among others, depend on parameters that adjust the abstract model to a real setting. Often, however, it is not clear how to choose the parameters as they do not have a physical quantity attached to them that can be measured directly. Thus, the parameters need to be inferred via indirect measurements, i.e., measuring derived quantities that can be modeled by a number of measurement functionals G_i.

We use the tools from goal-oriented FEM to infer parameters by comparing the real measurements with values obtained by a FEM simulation in a least-squares sense. This can be viewed as a special case of a finite dimensional optimal control problem, for which adaptive FEM algorithms based on residual *a posteriori* estimators, which depend on the model as well as the measurement functionals, are available [BM11; GY17; GYZ16; LC17]. However, the error estimators in these works show either a suboptimal rate in the error of the control (the parameters), or require strong regularity assumptions to the domain Ω.

Our analysis is instead based on [BV04; BV05], which consider dual weighted residual estimators but do not prove convergence of their methods. This allows us to assess the quality of the parameter estimate by *a priori* error bounds that match the rate of convergence of the parameter error without additional regularity assumptions. Based on this bound, we further use suitable residual estimators to devise an adaptive algorithm that drives down an analogous *a posteriori* bound for the parameter error with optimal rate in the sense of Theorem 1.10.

Chapter 5: Code

In addition to the mathematical analysis of the investigated algorithm or method, an integral part of numerical research is carrying out experiments to test the method. Software for such purposes should ideally possess three qualities: First, it should be accessible, i.e., all parts can be understood with little effort, in order for the researcher to understand which algorithms are implemented and, hence, used in the experiment. This is a key aspect in making research transparent and reproducible. Second, the software needs to be flexible to allow for modifications, since the method under investigation may not have been conceived before. Third, it should be efficient to enable large problem sizes and repeated computations, e.g., for parameter scans.

Existing codes, from the viewpoint of numerical analysts, often lack at least one of the listed qualities; see, e.g., [ABD+21; ABH+15; BBD+21; BHL+21; Che09; FGS15; FPW11; Sch14] for a representative sample. In this chapter, we present our own research code MooAFEM for (adaptive) FEM simulations in 2D, that is written exclusively in the high-level mathematical scripting language MATLAB. By use of object oriented programming, the library is partitioned into efficient modules with well-defined interactions, that allow for easy extensions.

MooAFEM is designed to deal with all the numerical experiments presented in this thesis: it can solve FEM discretizations of linear problems of the form (1.2) (and even more general ones). In doing so, it allows for very general coefficients; in particular, FEM functions can be used, which enables solution of nonlinear problems through iterative linearization techniques (as, e.g., used in [AW15; HPW21]). We explain the basic syntax of our library as well as the main design principles that underlie it. With some well-chosen numerical examples (where we give the implementation as well as the results), we show that it exhibits a good balance between the desired qualities.

1.5 Other scientific contributions

This last section of the introduction is dedicated to give a brief overview over the research questions that I have worked on during my PhD studies, but are not included in this thesis. Only research that is already published or submitted for publication is mentioned here. We note that the questions outlined in Sections 1.5.1–1.5.2 are also concerned with GOAFEM, but are not included in this thesis. This is due to both not fitting well into the specific framework of this thesis. As the first one is not concerned with rate optimality as introduced in Section 1.3, and the second one deals mainly with regularity issues of the linearized dual problem of a semilinear PDE.

1.5.1 Instance optimal GOAFEM

M. Innerberger and D. Praetorius. Instance-optimal goal-oriented adaptivity. *Comput. Methods Appl. Math.*, 21(1):109–126, 2021. DOI: 10.1515/cmam-2019-0115

Apart from rate optimality, which is the topic of this thesis, another notion of optimality exists: *instance optimality*. Instead of asymptotically achieving convergence with every possible rate, instance optimality aims for local optimality in the sense that the sequence of meshes $(\mathcal{T}_\ell)_{\ell\in\mathbb{N}_0}$ generated by an instance optimal algorithm is, up to a constant, the best sequence possible, i.e., there exist constants $C, C' > 0$ such that

$$|||u - u_\ell||| \leq C \inf\big\{|||u - u_h||| \,\big|\, \mathcal{T}_h \in \mathbb{T}(C'\#\mathcal{T}_\ell)\big\} \quad \text{for all } \ell \in \mathbb{N}_0.$$

Thus, instance optimality is a stronger statement than rate optimality.

Instance optimality was proved in [BDD04], one of the earliest works on AFEM, but was soon abandoned for rate optimality because it required a computationally expensive coarsening routine in every step in addition to refinement. It was only recently that a competitive instance optimal AFEM algorithm could be designed. The two works that achieved this are concerned with lowest order FEM ($p = 1$) for the Poisson model problem [DKS16] and lowest order non-conforming FEM for the Stokes equation [KS16].

In our work, we extend the framework of [DKS16] to conforming FEM of general polynomial order and, most importantly, design an instance optimal algorithm for GOAFEM. The most striking feature of such an algorithm is that, unlike Corollary 1.11, it already implies instance optimality of primal and dual problem separately. Thus, our algorithm allows for a trivial extension to drive down multiple goal functionals simultaneously in an instance optimal fashion.

1.5.2 GOAFEM for semilinear problems

R. Becker, M. Brunner, M. Innerberger, J. M. Melenk, and D. Praetorius. Goal-oriented adaptive finite element method for semilinear elliptic PDEs, 2021. arXiv: 2112.06687

As mentioned in the outline of Chapter 2, a logical extension of the linear theory presented in the introduction is to consider nonlinear problems. In contrast to Chapter 2, where a linear equation with nonlinear goal is considered, in this work we examine a semilinear equation together with a linear goal. The earlier works on GOAFEM for semilinear equations [HPZ15; XHYM22] prove only plain convergence of their methods, while at the same time imposing severe assumptions in the form of *a priori* L^∞-bounds on the (discrete) primal solution.

In this setting, the natural formulation of the dual problem involves the exact primal solution, which is not known in general. To circumvent this, we propose a *practical* dual problem that can be solved numerically. This leads to a goal-error estimate very similar to (2.12). Thus, we use some of the ideas presented in Chapter 2 to design error estimators and marking strategies. Drawing from this body of knowledge, in this work we can improve upon existing results in two ways. First, our work is the first one which proves convergence with optimal rates for GOAFEM for a nonlinear problem. Second, our analysis does not feature any assumed L^∞-bounds, but only assumptions on the growth of the nonlinear part of the operator.

1.5.3 Weak-strong uniqueness for solutions of the LLG equation

G. Di Fratta, M. Innerberger, and D. Praetorius. Weak-strong uniqueness for the Landau–Lifshitz–Gilbert equation in micromagnetics. *Nonlinear Anal. Real World Appl.*, 55:103122, 2020. DOI: 10.1016/j.nonrwa.2020.103122

The theory of computational micromagnetism is governed by the so-called Landau–Lifshitz–Gilbert (LLG) equation, which, for a magnetization $\boldsymbol{m} \in [H^1(\Omega)]^3$ on a bounded Lipschitz domain $\Omega \subset \mathbb{R}^3$, is of the form

$$\partial_t \boldsymbol{m} = \alpha\, \boldsymbol{m} \times \partial_t \boldsymbol{m} - \boldsymbol{m} \times \big[\Delta \boldsymbol{m} + \boldsymbol{\pi}(\boldsymbol{m})\big] \quad \text{with} \quad |\boldsymbol{m}|^2 = 1 \text{ a.e. in } \Omega.$$

The main difficulties in analyis as well as numerics for this equation are the nonlinear structure of the operator and the non-convex side constraint. To add to the problem, the *lower order terms* $\boldsymbol{\pi}$ can comprise non-local terms for which, e.g., the quasistatic Maxwell equations on the unbounded domain $\mathbb{R}^3\backslash\Omega$ have to be solved, and non-symmetric terms of the form $\boldsymbol{m} \cdot \operatorname{curl}(\boldsymbol{m})$. It is, however, exactly these terms that give rise to the most interesting phenomena; e.g., chiral magnetic Skyrmions that have a myriad of promising applications in microelectronics.

For the LLG equation with general lower order terms, existence of strong solutions can only be shown local in time, but in this case there holds uniqueness [FT17]. Weak solutions on the other hand exist globally in time, but are shown to be non-unique in some situations [AS92]. In this work, we show a weak-strong uniqueness principle: If a strong solution $\boldsymbol{m}_1$ and a weak solution $\boldsymbol{m}_2$ co-exist in some time interval $(0, T)$, there already holds $\boldsymbol{m}_1 = \boldsymbol{m}_2$ in $(0, T)$. In particular, this shows that weak solutions are locally unique in the interval of existence of a strong solution.

1.5.4 Exact diagonalization of time-dependent Hamiltonians

M. Innerberger, P. Worm, P. Prauhart, and A. Kauch. Electron-light interaction in nonequilibrium: exact diagonalization for time-dependent Hubbard Hamiltonians. *Eur. Phys. J. Plus*, 135:922, 2020. DOI: 10.1140/epjp/s13360-020-00919-2

Atomistic models in solid state physics, e.g., the Hubbard model, often lead to systems of differential equations

$$\partial_t x = Ax \quad \text{with } x \in \mathbb{C}^N, A \in \mathbb{C}^{N\times N} \tag{1.39}$$

involving a huge number of unknowns $N \in \mathbb{N}$. In the case of the Hubbard model, this number depends on the specific configuration under consideration. Under some realistic assumptions, e.g., that only neighboring atoms can interact, it can be as large as $N = 4^n$, where $n \in \mathbb{N}$ is the number of atoms. However, the system matrix A is usually sparse as well as Hermitian, i.e., $A = \overline{A}^\intercal$.

Computing the time evolution of the nonequilibrium system (1.39) without further simplifications is known as *exact diagonalization*.

In this work, we give an efficient implementation of the solution of (1.39) by means of Krylov subspace exponential integrators based on the Lanczos algorithm. In particular, only matrix-vector multiplications with the system matrix A are performed within the algorithm. This suggests to use the compressed sparse row (CSR) format to store only the non-zero entries of A. Our implementation permits computations of up to $n = 12$ atoms on standard consumer hardware within reasonable time. Several tests are conducted to assess the accuracy of the numerical solution with respect to physically relevant invariants.

1.5.5 Impact ionization in solar-cell models

A. Kauch, P. Worm, P. Prauhart, M. Innerberger, C. Watzenböck, and K. Held. Enhancement of impact ionization in Hubbard clusters by disorder and next-nearest-neighbor hopping. *Phys. Rev. B*, 102(24):245125, 2020. DOI: 10.1103/PhysRevB.102.245125

Silicon based solar cells can only harvest a specific amount of energy equal to their internal energy gap from every incoming photon. The energy of photons which do not reach this barrier, as well as excess energy from ones which surpass it, is not converted to electrical energy but dissipated as heat. Taking into account the spectral distribution of solar radiation, this consideration leads to the Shockley–Queisser limit of 34% efficiency of traditional solar cells [SQ61].

This limit can theoretically be surpassed if *impact ionization* takes place, i.e., the excess energy of photons is partially reused if it surpasses the energy gap; see, e.g., [Man10]. In this work, we model solar cells by small clusters within the well-known Hubbard model and perform exact diagonalization using the methods presented in [IWPK20]. We find impact ionization in all geometries that are at least two-dimensional. Furthermore, we find that this effect even grows larger with increasing perturbations in the geometry or the interaction between atoms due to a larger amount of different possible energy states in the lattice.

2 Optimal convergence rates for goal-oriented FEM with quadratic goal functional

Sections 2.2–2.7 of this chapter are taken from:
R. Becker, M. Innerberger, and D. Praetorius. Optimal convergence rates for goal-oriented FEM with quadratic goal functional. *Comput. Methods Appl. Math.*, 21(2):267–288, 2021.
DOI: 10.1515/cmam-2020-0044

2.1 Introduction

In this, in terms of content, first chapter, we relax the well-analyzed setting of the general introduction on GOAFEM given in Chapter 1 and drop one small but essential assumption: the linearity of the goal functional. To this end, we consider the linear elliptic PDE

$$-\operatorname{div} \boldsymbol{A}\nabla u + \boldsymbol{b}\cdot\nabla u + cu = f + \operatorname{div} \boldsymbol{f} \qquad \text{in } \Omega, \tag{2.1a}$$

$$u = 0 \qquad \text{on } \partial\Omega, \tag{2.1b}$$

where the weak formulation and corresponding discretizations will be made precise below. We further suppose, as in the introduction, that this problem fits into the setting of the Lax–Milgram lemma, i.e., there holds (1.4), and thus admits a unique solution (we also briefly comment on the case that only a Gårding inequality is satisfied in Section 2.2.8).

Differing from (1.13), we suppose that the goal functional $G\colon H^1_0(\Omega) \to \mathbb{R}$ is a quadratic form

$$G(v) := \langle \mathcal{K}v\,,\, v\rangle_{H^{-1}\times H^1_0} \quad \text{for all } v \in H^1_0(\Omega), \tag{2.2}$$

with a bounded linear operator $\mathcal{K}\colon H^1_0(\Omega) \to H^{-1}(\Omega)$ and the dual pairing $\langle\cdot\,,\,\cdot\rangle_{H^{-1}\times H^1_0}$ of $H^1_0(\Omega)$ and its dual space $H^{-1}(\Omega)$. Possible applications for goal functionals of the form (2.2) are, e.g., the weighted L^2-norm $G(u) = \int_\Omega g\,u^2\,\mathrm{d}x$ or some nonlinear convection term $G(u) = \int_\Omega u\,\boldsymbol{g}\cdot\nabla u\,\mathrm{d}x$ for given weights $g \in L^\infty(\Omega)$ and $\boldsymbol{g} \in [L^\infty(\Omega)]^d$.

In this chapter, we propose a GOAFEM algorithm in the spirit of Algorithm 1E and analyze if and how the results from Section 1.3 carry over to this new setting. The relaxation of the linearity assumption has far reaching consequences both for the design of a GOAFEM algorithm as well as its convergence analysis.

Consequences for GOAFEM algorithm

The change to a quadratic goal affects the steps Solve, Estimate, and Mark of Algorithm 1E.

First, it is not clear how to formulate the dual problem (1.12) in this context, since the right-hand side $G(v)$ is nonlinear. Instead, a suitable well-posed linearization of this problem has to be solved

in every step to obtain a dual solution; see (2.9) below. The right-hand side of the presented linearized dual problem is the Fréchet-derivative of G around the discrete primal solution u_ℓ and, hence, changes in every step of the GOAFEM algorithm. In particular, this implies a dependence of the (discrete) dual solution z_ℓ on the discrete primal solution, which is denoted by $z_\ell[u_\ell]$.

Second, this linearization introduces an error that renders the crucial goal error estimate (1.21) invalid. Instead, the substitute

$$|G(u) - G(u_\ell)| \lesssim \eta_\ell(u_\ell) \left[\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2 \right]^{1/2}, \tag{2.3}$$

which is shown below, must be used to estimate the goal error from the information provided by a suitable adaption of the *a posteriori* estimators (1.19). The quantity $\left[\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\right]^{1/2}$ is called combined (or product) estimator.

Finally, since the marking strategies introduced in Section 1.2.3 are tailored to the structure of the goal error estimate (1.21), marking for GOAFEM in the presence of a quadratic goal must be tailored to (2.3), accordingly. For this, we present two different approaches:

1. From comparing the goal error estimates (1.21) and (2.3), the most natural approach is to substitute the dual estimator by the combined estimator in the GOAFEM marking strategies presented so far.

2. Since the primal estimator is dominated by the combined estimator, another approach is to only mark elements for the combined estimator based on the Dörfler criterion (1.22). This is computationally less expensive but might also lead to a possibly reduced rate of convergence.

Consequences for analysis

For such profound changes in the structure of the GOAFEM algorithm, the necessary adjustments in the analysis are relatively mild. The idea is to show that there hold the estimator axioms (A1)–(A4) as well as quasi-orthogonality (QO) for the combined estimator and retrace the steps from Section 1.3. For the estimator axioms, this does not involve much effort and immediately implies plain convergence of the goal error, i.e.,

$$|G(u) - G(u_\ell)| \to 0 \quad \text{as} \quad \ell \to \infty. \tag{2.4}$$

As is evident from Figure 1.4, quasi-orthogonality (QO) is necessary to show linear convergence and, hence, convergence with optimal rates. In the introduction, to show (quasi-)orthogonality we use the Galerkin orthogonality (1.16), which, in the present setting, does not hold anymore since the dual problem changes on every level of the adaptive algorithm. To recover quasi-orthogonality, we further assume that $\mathcal{K}$ is compact. Under this additional assumption, convergence with optimal rates, Theorem 1.10, can be shown for both proposed algorithms with slightly adjusted convergence rates that result from the setting.

Chapter outline

In Section 2.2, we go into more detail about the finite element discretization and state the two goal-oriented adaptive algorithms outlined above (Algorithm 2A and 2B). Furthermore, we state the main results in this section. For every linear bounded operator $\mathcal{K}$, Proposition 2.1 and Proposition 2.4

state plain convergence of Algorithm 2A and Algorithm 2B, respectively. Under the additional assumption that the operator $\mathcal{K}$ is compact, Theorem 2.2 yields linear convergence with optimal rates for Algorithm 2A. For the computationally less expensive Algorithm 2B, Theorem 2.5 yields convergence with almost optimal rates in the sense that the rates are, in general, lower that that of Algorithm 2A. We further provide some numerical experiments in Section 2.3 for the examples stated above as well as an example with bounded but not compact $\mathcal{K}$. Finally, the main theorems are proved in Sections 2.4–2.7, where the most important auxiliary result is Lemma 2.13, quasi-orthogonality for the combined quantities.

2.2 Adaptive algorithm & main result

2.2.1 Variational formulation

Define the bilinear form

$$a(u,v) := \int_\Omega \boldsymbol{A}\nabla u \cdot \nabla v \,\mathrm{d}x + \int_\Omega \boldsymbol{b}\cdot\nabla u\, v \,\mathrm{d}x + \int_\Omega cuv\,\mathrm{d}x. \tag{2.5}$$

We suppose that $a(\cdot,\cdot)$ fits into the setting of the Lax–Milgram lemma, i.e., $a(\cdot,\cdot)$ is continuous and elliptic on $H^1_0(\Omega)$. While continuity

$$a(u,v) \le C_{\mathrm{cnt}}\,\|u\|_{H^1(\Omega)}\|v\|_{H^1(\Omega)} \quad \text{for all } u,v\in H^1_0(\Omega) \tag{2.6}$$

follows from the assumptions made with $C_{\mathrm{cnt}} = \|\boldsymbol{A}\|_{L^\infty(\Omega)} + \|\boldsymbol{b}\|_{L^\infty(\Omega)} + \|c\|_{L^\infty(\Omega)}$, the ellipticity

$$a(u,u) \ge C_{\mathrm{ell}}\,\|u\|^2_{H^1(\Omega)} \quad \text{for all } u\in H^1_0(\Omega) \tag{2.7}$$

requires additional assumptions on the coefficients, e.g.,

$$\inf_{x\in\Omega}\inf_{\boldsymbol{y}\in\mathbb{R}^d\setminus\{0\}} \frac{\boldsymbol{y}\cdot\boldsymbol{A}(x)\boldsymbol{y}}{|\boldsymbol{y}|^2} > 0 \quad\text{and}\quad \boldsymbol{b}\in\boldsymbol{H}(\mathrm{div};\Omega) \quad\text{with}\quad \inf_{x\in\Omega}\Big(\frac{1}{2}\,\mathrm{div}\,\boldsymbol{b}(x) + c(x)\Big) \ge 0.$$

The weak formulation of (2.1) reads

$$a(u,v) = F(v) := \int_\Omega fv\,\mathrm{d}x - \int_\Omega \boldsymbol{f}\cdot\nabla v\,\mathrm{d}x \quad \text{for all } v\in H^1_0(\Omega). \tag{2.8}$$

According to the Lax–Milgram lemma, (2.8) admits a unique solution $u\in H^1_0(\Omega)$. Given $w\in H^1_0(\Omega)$, the same argument applies and proves that the (linearized) dual problem

$$a(v,z[w]) = b(v,w) + b(w,v) \quad \text{for all } v\in H^1_0(\Omega) \tag{2.9}$$

admits a unique solution $z[w]\in H^1_0(\Omega)$, where we abbreviate the notation by use of $b(v,w) := \langle\mathcal{K}v,\, w\rangle_{H^{-1}\times H^1_0}$. We note that $b(\cdot,\cdot)$ is, in particular, a continuous bilinear form on $H^1_0(\Omega)$. Throughout, we denote by $|||v|||^2 := \int_\Omega \boldsymbol{A}\nabla v\cdot\nabla v\,\mathrm{d}x$ the energy norm induced by the principal part of $a(\cdot,\cdot)$, which is an equivalent norm on $H^1_0(\Omega)$. Finally, we stress that all main results also apply to the case that $a(\cdot,\cdot)$ satisfies only a Gårding inequality (instead of the strong ellipticity (2.7)) as long as the weak formulations (2.8) and (2.9) are well-posed; see Section 2.2.8 below.

2.2.2 Finite element method

For a conforming triangulation $\mathcal{T}_H$ of Ω into compact simplices and a polynomial degree $p \geq 1$, we consider the conforming finite element space

$$\mathcal{X}_H := \{v_H \in H^1_0(\Omega) \,|\, \forall T \in \mathcal{T}_H \quad v_H|_T \text{ is a polynomial of degree } \leq p\}. \tag{2.10}$$

We approximate $u \approx u_H \in \mathcal{X}_H$ and $z[w] \approx z_H[w] \in \mathcal{X}_H$. More precisely, the Lax–Milgram lemma yields the existence and uniqueness of discrete FEM solutions $u_H, z_H[w] \in \mathcal{X}_H$ of

$$a(u_H, v_H) = F(v_H) \text{ and } a(v_H, z_H[w]) = b(v_H, w) + b(w, v_H) \text{ for all } v_H \in \mathcal{X}_H. \tag{2.11}$$

2.2.3 Linearization of the goal functional

To control the goal error $|G(u) - G(u_H)|$, we employ the dual problem. Note that

$$\begin{aligned} b(u - u_H, u - u_H) &= b(u, u) - b(u_H, u) - b(u, u_H) + b(u_H, u_H) \\ &= \big[G(u) - G(u_H)\big] - \big[b(u_H, u) + b(u, u_H) - 2b(u_H, u_H)\big] \\ &= \big[G(u) - G(u_H)\big] - \big[b(u_H, u - u_H) + b(u - u_H, u_H)\big]. \end{aligned}$$

With the dual problem and the Galerkin orthogonality, we rewrite the second bracket as

$$b(u - u_H, u_H) + b(u_H, u - u_H) \overset{(2.9)}{=} a(u - u_H, z[u_H]) = a(u - u_H, z[u_H] - z_H[u_H]).$$

With continuity of the bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$, we thus obtain that

$$\begin{aligned} \big|G(u) - G(u_H)\big| &= \big|a(u - u_H, z[u_H] - z_H[u_H]) + b(u - u_H, u - u_H)\big| \\ &\lesssim |\!|\!| u - u_H |\!|\!| \, \big[\, |\!|\!| z[u_H] - z_H[u_H] |\!|\!| + |\!|\!| u - u_H |\!|\!| \,\big]. \end{aligned} \tag{2.12}$$

2.2.4 Mesh-refinement

Let $\mathcal{T}_0$ be a given conforming triangulation of Ω. We suppose that the mesh-refinement is a deterministic and fixed strategy, e.g., newest vertex bisection [Ste08]. For each triangulation $\mathcal{T}_H$ and marked elements $\mathcal{M}_H \subseteq \mathcal{T}_H$, let $\mathcal{T}_h := \texttt{refine}(\mathcal{T}_H, \mathcal{M}_H)$ be the coarsest triangulation, where all $T \in \mathcal{M}_H$ have been refined, i.e., $\mathcal{M}_H \subseteq \mathcal{T}_H \backslash \mathcal{T}_h$. We write $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, if $\mathcal{T}_h$ results from $\mathcal{T}_H$ by finitely many steps of refinement. To abbreviate notation, let $\mathbb{T} := \mathbb{T}(\mathcal{T}_0)$.

We further suppose that each refined element has at least two sons, i.e.,

$$\#(\mathcal{T}_H \backslash \mathcal{T}_h) + \#\mathcal{T}_H \leq \#\mathcal{T}_h \quad \text{for all } \mathcal{T}_H \in \mathbb{T} \text{ and all } \mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H), \tag{2.13}$$

and that the refinement rule satisfies the mesh-closure estimate

$$\#\mathcal{T}_\ell - \#\mathcal{T}_0 \leq C_{\mathrm{mesh}} \sum_{j=0}^{\ell-1} \#\mathcal{M}_j \quad \text{for all } \ell \in \mathbb{N}, \tag{2.14}$$

where $C_{\mathrm{mesh}} > 0$ depends only on $\mathcal{T}_0$. This has first been proved for 2D newest vertex bisection in [BDD04] and has later been generalized to arbitrary dimension $d \geq 2$ in [Ste08]. While both works require an additional admissibility assumption on $\mathcal{T}_0$, this has been proved unnecessary at

least for 2D in [KPP13]. Finally, it has been proved in [CKNS08; Ste07] that newest vertex bisection ensures the overlay estimate, i.e., for all triangulations $\mathcal{T}_H, \mathcal{T}_h \in \mathbb{T}$, there exists a common refinement $\mathcal{T}_H \oplus \mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H) \cap \mathbb{T}(\mathcal{T}_h)$ which satisfies that

$$\#(\mathcal{T}_H \oplus \mathcal{T}_h) \le \#\mathcal{T}_H + \#\mathcal{T}_h - \#\mathcal{T}_0. \tag{2.15}$$

For meshes with first-order hanging nodes, (2.13)–(2.15) are analyzed in [BN10], while T-splines and hierarchical splines for isogeometric analysis are considered in [Mor16; MP15] and [BGMP16; GHP17], respectively.

2.2.5 Error estimators

For $\mathcal{T}_H \in \mathbb{T}$ and $v_H \in \mathcal{X}_H$, let

$$\eta_H(T, v_H) \ge 0 \quad \text{and} \quad \zeta_H(T, v_H) \ge 0 \quad \text{for all } T \in \mathcal{T}_H$$

be given refinement indicators. For $\mathcal{U}_H \subseteq \mathcal{T}_H$, let

$$\eta_H(\mathcal{U}_H, v_H) := \Big(\sum_{T \in \mathcal{U}_H} \eta_H(T, v_H)^2 \Big)^{1/2} \quad \text{and} \quad \zeta_H(\mathcal{U}_H, v_H) := \Big(\sum_{T \in \mathcal{U}_H} \zeta_H(T, v_H)^2 \Big)^{1/2}.$$

To abbreviate notation, let $\eta_H(v_H) := \eta_H(\mathcal{T}_H, v_H)$ and $\zeta_H(v_H) := \zeta_H(\mathcal{T}_H, v_H)$.

We suppose that the estimators η_H and ζ_H satisfy the following *axioms of adaptivity* from [CFPP14]: There exist constants $C_{\mathrm{stab}}, C_{\mathrm{rel}}, C_{\mathrm{drel}} > 0$ and $0 < q_{\mathrm{red}} < 1$ such that for all $\mathcal{T}_H \in \mathbb{T}$ and all $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, the following assumptions are satisfied:

(A1) stability: For all $v_h \in \mathcal{X}_h$, $v_H \in \mathcal{X}_H$, and $\mathcal{U}_H \subseteq \mathcal{T}_h \cap \mathcal{T}_H$, it holds that

$$\big|\eta_h(\mathcal{U}_H, v_h) - \eta_H(\mathcal{U}_H, v_H)\big| + \big|\zeta_h(\mathcal{U}_H, v_h) - \zeta_H(\mathcal{U}_H, v_H)\big| \le C_{\mathrm{stab}} \, |\!|\!| v_h - v_H |\!|\!|.$$

(A2) reduction: For all $v_H \in \mathcal{X}_H$, it holds that

$$\eta_h(\mathcal{T}_h \backslash \mathcal{T}_H, v_H) \le q_{\mathrm{red}} \, \eta_H(\mathcal{T}_H \backslash \mathcal{T}_h, v_H),$$
$$\zeta_h(\mathcal{T}_h \backslash \mathcal{T}_H, v_H) \le q_{\mathrm{red}} \, \zeta_H(\mathcal{T}_H \backslash \mathcal{T}_h, v_H).$$

(A3) reliability: For all $w \in H^1_0(\Omega)$, the Galerkin solutions $u_H, z_H[w] \in \mathcal{X}_H$ to (2.11) satisfy that

$$|\!|\!| u - u_H |\!|\!| \le C_{\mathrm{rel}} \, \eta_H(u_H),$$
$$|\!|\!| z[w] - z_H[w] |\!|\!| \le C_{\mathrm{rel}} \, \zeta_H(z_H[w]).$$

(A4) discrete reliability: For all $w \in H^1_0(\Omega)$, the Galerkin solutions $u_H, z_H[w] \in \mathcal{X}_H$ and $u_h, z_h[w] \in \mathcal{X}_h$ to (2.11) satisfy that

$$|\!|\!| u_h - u_H |\!|\!| \le C_{\mathrm{drel}} \, \eta_H(\mathcal{T}_H \backslash \mathcal{T}_h, u_H),$$
$$|\!|\!| z_h[w] - z_H[w] |\!|\!| \le C_{\mathrm{drel}} \, \zeta_H(\mathcal{T}_H \backslash \mathcal{T}_h, z_H[w]).$$

We note that the axioms (A1)–(A4) are satisfied for, e.g., standard residual error estimators. Given $w \in H^1_0(\Omega)$, the mapping $v \mapsto b(v,w) + b(w,v)$ is linear and continuous by assumption. Hence, the Riesz theorem from functional analysis guarantees the existence (and uniqueness) of $g[w] \in H^1_0(\Omega)$ such that

$$(g[w], v)_{H^1} := \int_\Omega g[w] v \,\mathrm{d}x + \int_\Omega \nabla g[w] \cdot \nabla v \,\mathrm{d}x = b(v,w) + b(w,v) \text{ for all } v \in H^1_0(\Omega). \tag{2.16}$$

With $\boldsymbol{g}[w] = -\nabla g[w]$, we thus get that

$$b(v,w) + b(w,v) = \int_\Omega g[w] v \,\mathrm{d}x - \int_\Omega \boldsymbol{g}[w] \cdot \nabla v \,\mathrm{d}x \quad \text{for all } v \in H^1_0(\Omega), \tag{2.17}$$

i.e., the right-hand sides of the primal problem (2.8) and the (linearized) dual problem (2.9) take the same form. With this[1], the residual error estimators read for $v_H \in \mathcal{X}_H$ as

$$\begin{aligned}
\eta_H(T, v_H)^2 &:= h_T^2 \| -\operatorname{div}(\boldsymbol{A}\nabla v_H + \boldsymbol{f}) + \boldsymbol{b}\cdot\nabla v_H + c v_H - f\|^2_{L^2(T)} \\
&\qquad + h_T \|[\![(\boldsymbol{A}\nabla v_H + \boldsymbol{f})\cdot \boldsymbol{n}]\!]\|^2_{L^2(\partial T\cap\Omega)}, \\
\zeta_H(T, v_H)^2 &:= h_T^2 \| -\operatorname{div}(\boldsymbol{A}\nabla v_H + \boldsymbol{g}[u_H]) - \boldsymbol{b}\cdot\nabla v_H + (c - \operatorname{div}\boldsymbol{b}) v_H - g[u_H]\|^2_{L^2(T)} \\
&\qquad + h_T \|[\![(\boldsymbol{A}\nabla v_H + \boldsymbol{g}[u_H])\cdot \boldsymbol{n}]\!]\|^2_{L^2(\partial T\cap\Omega)},
\end{aligned}$$

where $[\![\cdot]\!]$ denotes the jump across faces and $\boldsymbol{n}$ is the outwards-facing unit normal vector. We stress that our experiments below directly provide $g[w] \in L^2(\Omega)$ and $\boldsymbol{g}[w] \in [L^2(\Omega)]^d$ satisfying the representation (2.17), so that there is, in fact, no need to solve (2.16).

2.2.6 Adaptive algorithm

We consider the following adaptive algorithm, which adapts the marking strategy proposed in [FPZ16].

Algorithm 2A

Input: Adaptivity parameters $0 < \theta \le 1$ and $C_{\mathrm{mark}} \ge 1$, initial mesh $\mathcal{T}_0$.
Loop: For all $\ell = 0, 1, 2, \dots$, perform the following steps (i)–(v):

(i) Compute the discrete solutions $u_\ell, z_\ell[u_\ell] \in \mathcal{X}_\ell$ to (2.11).

(ii) Compute the refinement indicators $\eta_\ell(T, u_\ell)$ and $\zeta_\ell(T, z_\ell[u_\ell])$ for all $T \in \mathcal{T}_\ell$.

(iii) Determine sets $\overline{\mathcal{M}}^u_\ell, \overline{\mathcal{M}}^{uz}_\ell \subseteq \mathcal{T}_\ell$ of up to the multiplicative constant C_{mark} minimal

[1] Recall the strong form of the primal problem

$$-\operatorname{div}\boldsymbol{A}\nabla u + \boldsymbol{b}\cdot\nabla u + cu = f + \operatorname{div}\boldsymbol{f} \quad \text{in } \Omega$$

and note that the corresponding (linearized) strong form of the dual problem reads

$$-\operatorname{div}\boldsymbol{A}\nabla z - \boldsymbol{b}\cdot\nabla z + (c - \operatorname{div}\boldsymbol{b}) z = g[w] + \operatorname{div}\boldsymbol{g}[w] \quad \text{in } \Omega.$$

cardinality such that

$$\theta\,\eta_\ell(u_\ell)^2 \le \eta_\ell(\overline{\mathcal{M}}_\ell^u, u_\ell)^2, \tag{2.18a}$$

$$\theta\,\big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big] \le \big[\,\eta_\ell(\overline{\mathcal{M}}_\ell^{uz}, u_\ell)^2 + \zeta_\ell(\overline{\mathcal{M}}_\ell^{uz}, z_\ell[u_\ell])^2\,\big]. \tag{2.18b}$$

(iv) Let $\mathcal{M}_\ell^u \subseteq \overline{\mathcal{M}}_\ell^u$ and $\mathcal{M}_\ell^{uz} \subseteq \overline{\mathcal{M}}_\ell^{uz}$ with $\#\mathcal{M}_\ell^u = \#\mathcal{M}_\ell^{uz} = \min\{\,\#\overline{\mathcal{M}}_\ell^u\,,\,\#\overline{\mathcal{M}}_\ell^{uz}\,\}$.

(v) Define $\mathcal{M}_\ell := \mathcal{M}_\ell^u \cup \mathcal{M}_\ell^{uz}$ and generate $\mathcal{T}_{\ell+1} := \texttt{refine}(\mathcal{T}_\ell, \mathcal{M}_\ell)$.

Output: Sequence of triangulations $\mathcal{T}_\ell$ with corresponding discrete solutions u_ℓ and $z_\ell[u_\ell]$ as well as error estimators $\eta_\ell(u_\ell)$ and $\zeta_\ell(z_\ell[u_\ell])$.

With Algorithm 2B below, we give and examine an alternative adaptive algorithm that is seemingly cheaper in computational costs.

Our first result states that Algorithm 2A indeed leads to convergence.

Proposition 2.1. *For any bounded linear operator $\mathcal{K}\colon H_0^1(\Omega) \to H^{-1}(\Omega)$, there hold the following statements* (i)–(ii)*:*

(i) *There exists a constant $C'_{\mathrm{rel}} > 0$ such that*

$$\big|G(u) - G(u_H)\big| \le C'_{\mathrm{rel}}\,\eta_H(u_H)\,\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big]^{1/2} \quad \text{for all } \mathcal{T}_H \in \mathbb{T}. \tag{2.19}$$

(ii) *For all $0 < \theta \le 1$ and $1 < C_{\mathrm{mark}} \le \infty$, Algorithm 2A leads to convergence*

$$|G(u) - G(u_\ell)| \le C'_{\mathrm{rel}}\eta_\ell(u_\ell)\,\big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big]^{1/2} \longrightarrow 0 \quad \text{as } \ell \to \infty. \tag{2.20}$$

The constant C'_{rel} depends only on the constants from (A1)–(A3), the bilinear form $a(\cdot,\cdot)$ and the boundedness of $\mathcal{K}$.

To formulate our main result on optimal convergence rates, we need some additional notation. For $N \in \mathbb{N}_0$, let $\mathbb{T}_N := \{\mathcal{T} \in \mathbb{T} \mid \#\mathcal{T} - \#\mathcal{T}_0 \le N\}$ denote the (finite) set of all refinements of $\mathcal{T}_0$, which have at most N elements more than $\mathcal{T}_0$. For $s, t > 0$, we define

$$\|u\|_{\mathbb{A}_s} := \sup_{N\in\mathbb{N}_0} \Big((N+1)^s \min_{\mathcal{T}_H\in\mathbb{T}_N} \eta_H(u_H)\Big) \in \mathbb{R}_{\ge 0} \cup \{\infty\},$$

$$\|z[u]\|_{\mathbb{A}_t} := \sup_{N\in\mathbb{N}_0} \Big((N+1)^t \min_{\mathcal{T}_H\in\mathbb{T}_N} \zeta_H(z_H[u])\Big) \in \mathbb{R}_{\ge 0} \cup \{\infty\}.$$

In explicit terms, e.g., $\|u\|_{\mathbb{A}_s} < \infty$ means that an algebraic convergence rate $O(N^{-s})$ for the error estimator η_ℓ is possible, if the optimal triangulations are chosen.

The following theorem concludes the main results of the present work:

Theorem 2.2

For any compact operator $\mathcal{K} : H_0^1(\Omega) \to H^{-1}(\Omega)$, there even hold the following statements (i)–(ii)*, which improve Proposition 2.1*(ii)*:*

(i) *For all $0 < \theta \le 1$ and $C_{\mathrm{mark}} \ge 1$, there exists $\ell_0 \in \mathbb{N}_0$, $C_{\mathrm{lin}} > 0$, and $0 < q_{\mathrm{lin}} < 1$ such*

that Algorithm 2A guarantees that, for all $\ell, n \in \mathbb{N}_0$ with $n \geq \ell \geq \ell_0$,

$$\eta_n(u_n)\,\big[\eta_n(u_n)^2 + \zeta_n(z_n[u_n])^2\big]^{1/2} \leq C_{\mathrm{lin}} q_{\mathrm{lin}}^{n-\ell}\,\eta_\ell(u_\ell)\,\big[\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\big]^{1/2}. \tag{2.21}$$

(ii) *There exist $C_{\mathrm{opt}} > 0$ and $\ell_0 \in \mathbb{N}_0$ such that Algorithm 2A guarantees that, for all $0 < \theta < \theta_{\mathrm{opt}} := (1 + C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2)^{-1}$, for all $s, t > 0$ with $\|u\|_{\mathbb{A}_s} + \|z[u]\|_{\mathbb{A}_t} < \infty$, and all $\ell \in \mathbb{N}_0$ with $\ell \geq \ell_0$, it holds that*

$$\eta_\ell(u_\ell)\big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big]^{1/2} \leq C_{\mathrm{opt}}\,\|u\|_{\mathbb{A}_s}(\|u\|_{\mathbb{A}_s} + \|z\|_{\mathbb{A}_t})\,(\#\mathcal{T}_\ell - \#\mathcal{T}_0)^{-\alpha}, \tag{2.22}$$

where $\alpha := \min\{2s, s+t\}$.

The constants C_{lin}, q_{lin}, and ℓ_0 depend only on θ, q_{red}, C_{stab}, C_{rel}, the bilinear form $a(\cdot,\cdot)$, and the compact operator $\mathcal{K}$. The constant C_{opt} depends only on θ, C_{mesh}, C_{mark}, C_{lin}, q_{lin}, ℓ_0, and (A1)–(A4).

Remark 2.3. (i) *We note that, according to the considered dual problem (2.9), the goal functional (2.2) is linearized around u_ℓ in each step of the adaptive algorithm. Hence, we must enforce that the linearization error satisfies that $|||z_\ell[u] - z_\ell[u_\ell]||| \to 0$ as $\ell \to \infty$. This is guaranteed by Proposition 2.1(ii) and Theorem 2.2(i), since both factors of the product involve the primal error estimator $\eta_\ell(u_\ell)$.*

(ii) *For a linear goal functional and hence $z_\ell[u] = z_\ell[u_\ell]$, the work [FPZ16] considers plain ζ_ℓ^2 (instead of $\eta_\ell^2 + \zeta_\ell^2$) for the Dörfler marking (2.18b) and then proves a convergence behavior $|G(u) - G(u_\ell)| \lesssim \eta_\ell \zeta_\ell = \mathcal{O}\big((\#\mathcal{T}_\ell)^{-\alpha}\big)$ for the estimator product, where $\alpha = s+t$ with $s > 0$ being the optimal rate for the primal problem and $t > 0$ being the optimal rate for the dual problem. Instead, Algorithm 2A will only lead to $\mathcal{O}\big((\#\mathcal{T}_\ell)^{-\alpha}\big)$, where $\alpha = \min\{2s, s+t\}$; see Theorem 2.2(ii).*

(iii) *The marking strategy proposed in [BET11], where Dörfler marking is carried out for the weighted estimator*

$$\rho_H(T, u_H, z_H[u_H])^2 := \eta_H(T, u_H)^2 \zeta_H(z_H[u_H])^2 + \eta_H(u_H)^2 \zeta_H(T, z_H[u_H])^2, \tag{2.23}$$

might be unable to ensure convergence of the linearization error $|||z_\ell[u] - z_\ell[u_\ell]|||$, since in every step Dörfler marking is implied for either $\eta_H(u_H)$ or $\zeta_H(z_H[u_H])$; cf. [FPZ16]. If one instead considers

$$\begin{aligned}\varrho_H(T, u_H, z_H[u_H])^2 := \eta_H(T, u_H)^2\big[\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\big] \\ + \eta_H(u_H)^2\big[\eta_H(T, u_H)^2 + \zeta_H(T, z_H[u_H])^2\big],\end{aligned} \tag{2.24}$$

the present results and the analysis in [FPZ16] make it clear that this strategy implies convergence with rate $\min\{2s, s+t\}$. Details are omitted.

2.2.7 Alternative adaptive algorithm

From the upper bound (2.19) in Proposition 2.1(i), we can further estimate the goal error by

$$\big|G(u) - G(u_H)\big| \leq C'_{\mathrm{rel}}\,\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big] \quad \text{for all } \mathcal{T}_H \in \mathbb{T}.$$

This suggests the following algorithm, which marks elements solely based on the combined estimator.

Algorithm 2B

Input: Adaptivity parameters $0 < \theta \leq 1$ and $C_{\mathrm{mark}} \geq 1$, initial mesh $\mathcal{T}_0$.
Loop: For all $\ell = 0, 1, 2, \ldots$, perform the following steps (i)–(iv):

(i) Compute the discrete solutions $u_\ell, z_\ell[u_\ell] \in \mathcal{X}_\ell$ to (2.11).

(ii) Compute the refinement indicators $\eta_\ell(T, u_\ell)$ and $\zeta_\ell(T, z_\ell[u_\ell])$ for all $T \in \mathcal{T}_\ell$.

(iii) Determine a set $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ of up to the multiplicative constant C_{mark} minimal cardinality such that

$$\theta \left[\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2 \right] \leq \left[\eta_\ell(\mathcal{M}_\ell, u_\ell)^2 + \zeta_\ell(\mathcal{M}_\ell, z_\ell[u_\ell])^2 \right]. \tag{2.25}$$

(iv) Generate $\mathcal{T}_{\ell+1} := \texttt{refine}(\mathcal{T}_\ell, \mathcal{M}_\ell)$.

Output: Sequence of triangulations $\mathcal{T}_\ell$ with corresponding discrete solutions u_ℓ and $z_\ell[u_\ell]$ as well as error estimators $\eta_\ell(u_\ell)$ and $\zeta_\ell(z_\ell[u_\ell])$.

First, we note that Algorithm 2B also leads to convergence.

Proposition 2.4. *For any bounded linear operator $\mathcal{K}$, there hold the following statements* (i)–(ii)*:*
(i) *There exists a constant $C'_{\mathrm{rel}} > 0$ such that*

$$\big|G(u) - G(u_H)\big| \leq C'_{\mathrm{rel}} \left[\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2 \right] \quad \text{for all } \mathcal{T}_H \in \mathbb{T}. \tag{2.26}$$

(ii) *For all $0 < \theta \leq 1$ and $1 < C_{\mathrm{mark}} \leq \infty$, Algorithm 2B leads to convergence*

$$|G(u) - G(u_\ell)| \leq C'_{\mathrm{rel}} \left[\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2 \right] \longrightarrow 0 \quad \text{as } \ell \to \infty. \tag{2.27}$$

The constant C'_{rel} depends only on the constants from (A1)–(A3), the bilinear form $a(\cdot,\cdot)$, and the boundedness of $\mathcal{K}$.

The following theorem proves linear convergence of Algorithm 2B with almost optimal convergence rate, where we note that $\beta \leq \alpha$ for the rates in (2.22) and (2.29). By abuse of notation we use the same constants as in Theorem 2.2.

Theorem 2.5

For any compact operator $\mathcal{K}$, there even hold the following statements (i)–(ii)*, which improve Proposition 2.4(ii):*
(i) *For all $0 < \theta \leq 1$ and $C_{\mathrm{mark}} \geq 1$, there exists $\ell_0 \in \mathbb{N}_0$, $C_{\mathrm{lin}} > 0$, and $0 < q_{\mathrm{lin}} < 1$ such that Algorithm 2B guarantees that, for all $\ell, n \in \mathbb{N}_0$ with $n \geq \ell \geq \ell_0$,*

$$\left[\eta_n(u_n)^2 + \zeta_n(z_n[u_n])^2 \right] \leq C_{\mathrm{lin}} q_{\mathrm{lin}}^{n-\ell} \left[\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2 \right]. \tag{2.28}$$

(ii) *There exist $C_{\mathrm{opt}} > 0$ and $\ell_0 \in \mathbb{N}_0$ such that Algorithm 2A guarantees that, for all $0 < \theta < \theta_{\mathrm{opt}} := (1 + C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2)^{-1}$, for all $s, t > 0$ with $\|u\|_{\mathbb{A}_s} + \|z[u]\|_{\mathbb{A}_t} < \infty$, and all $\ell \in \mathbb{N}_0$*

with $\ell \geq \ell_0$, it holds that

$$\big[\, \eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2 \,\big] \leq C_{\mathrm{opt}} \, (\|u\|^2_{\mathbb{A}_s} + \|z\|^2_{\mathbb{A}_t}) \, (\#\mathcal{T}_\ell - \#\mathcal{T}_0)^{-\beta}, \tag{2.29}$$

where $\beta := \min\{2s, 2t\}$.

The constants C_{lin}, q_{lin}, and ℓ_0 depend only on θ, q_{red}, C_{stab}, C_{rel}, the bilinear form $a(\cdot,\cdot)$, and the compact operator $\mathcal{K}$. The constant C_{opt} depends only on θ, C_{mesh}, C_{mark}, and (A1)–(A4).

Note that Algorithm 2B has slightly lower computational costs than Algorithm 2A, but achieves only a lower rate in general. However, if there holds $s \leq t$ both algorithms achieve rate $2s$.

2.2.8 Extension of analysis to compactly perturbed elliptic problems

For the ease of presentation, we have restricted ourselves to the case that the bilinear form $a(\cdot,\cdot)$ from (2.5) is continuous (2.6) and elliptic (2.7). Actually, it suffices to assume that $a(\cdot,\cdot)$ is continuous and that the energy norm $|||\cdot|||$ induced by the principal part is an equivalent norm on $H^1_0(\Omega)$, e.g., by assuming that $\boldsymbol{A} \in L^\infty(\Omega)$ is uniformly positive definite. Then, $a(\cdot,\cdot)$ is elliptic up to some compact perturbation (and hence satisfies a Gårding inequality). A prominent example for this problem class is the Helmholtz problem.

We have to assume that the primal formulation (2.8) is well-posed, i.e., for all $w \in H^1_0(\Omega)$ it holds that

$$\big[\, a(w,v) = 0 \text{ for all } v \in H^1_0(\Omega) \,\big] \quad \Longrightarrow \quad w = 0.$$

Then, the Fredholm alternative and standard functional analysis imply that the primal formulation (2.8) as well as the dual formulation (2.9) admit unique solutions. Moreover, as soon as $\mathcal{T}_H$ is sufficiently fine, also the FEM problems (2.11) admit unique solutions and, more importantly, the discrete inf-sup constants are uniformly bounded from below; see, e.g., [BHP17, Section 2].

As noted in [BHP17], such an analytical setting requires only two minor modifications of adaptive algorithms:

(a) *Step* (i) *in Algorithm 2A or Algorithm 2B:* If the discrete solutions u_ℓ and $z_\ell[u_\ell]$ exist (and, hence, are also unique), then we proceed as before. If either u_ℓ or $z_\ell[u_\ell]$ does not exist, then the mesh $\mathcal{T}_{\ell+1}$ is obtained by uniform refinement of $\mathcal{T}_\ell$, i.e., $\mathcal{M}_\ell := \mathcal{T}_\ell$.

(b) *Step* (iv) *of Algorithm 2A or step* (iii) *of Algorithm 2B:* Having determined a set of marked elements $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$, we select a superset $\mathcal{M}_\ell^\# \supseteq \mathcal{M}_\ell$ with $\#\mathcal{M}_\ell^\# \leq 2\#\mathcal{M}_\ell$ as well as $\mathcal{M}_\ell^\# \cap \{T \in \mathcal{T}_\ell \,|\, |T| \geq |T'| \text{ for all } T' \in \mathcal{T}_\ell\} \neq \emptyset$ and define the refined mesh $\mathcal{T}_{\ell+1} := \texttt{refine}(\mathcal{T}_\ell, \mathcal{M}_\ell^\#)$ via the extended set of marked elements.

It is observed in [BHP17] that uniform refinement caused by the modification (a) can only occur finitely many times. Moreover, the modification (b) ensures that $H^1_0(\Omega) = \overline{\bigcup_{\ell=0}^\infty \mathcal{X}_\ell}$ so that the adaptive algorithm indeed converges to the right limit. For standard adaptive FEM, it is shown in [BHP17] that this procedure still leads to optimal convergence rates. We note that the arguments from [BHP17] obviously extend to the present goal-oriented adaptive FEM.

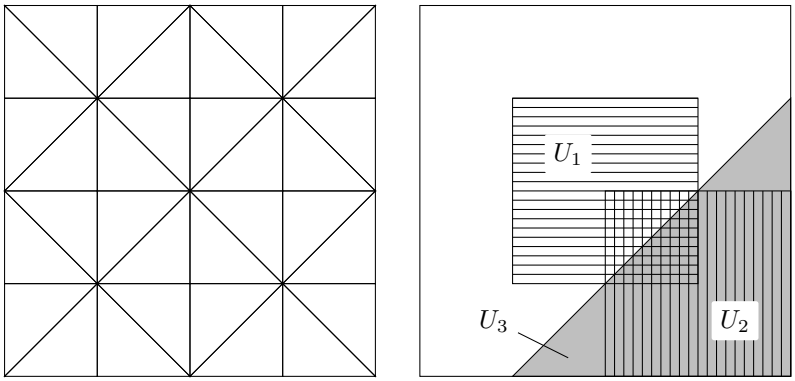


Figure 2.1: Initial mesh (left) and sets U_1, U_2, U_3 (right) on the unit square $\Omega = (0,1)^2$.

2.3 Numerical experiments

In this section, we underline our theoretical findings by some numerical examples. As starting point of all examples, we use equation (2.1) with $\boldsymbol{A} = I$, $\boldsymbol{b} = \boldsymbol{0}$, and $c = 0$ on the unit square $\Omega = (0,1)^2$. The initial mesh $\mathcal{T}_0$ on Ω is obtained from certain uniform refinements from the mesh shown in Figure 2.1. All examples are computed with conforming finite elements of order $p = 1$ and $p = 2$, as outlined in Section 2.2.2.

In the following, we consider the marking strategies of Algorithm 2A and Algorithm 2B (denoted by A and B, respectively), as well as the marking strategies outlined in Remark 2.3(iii), i.e., Dörfler marking for (2.23) and (2.24), which will be denoted by BET1 and BET2, respectively. If not stated otherwise, the marking parameter is $\theta = 0.5$ for all experiments.

2.3.1 Weighted L^2-norm

Suppose some weight function $\lambda \in L^\infty(\Omega)$ with $\lambda \geq 0$ a.e., whose regions of discontinuity are resolved by the initial mesh $\mathcal{T}_0$ (i.e., g is continuous in the interior of every element of $\mathcal{T}_0$). Then, we consider the weighted L^2-norm

$$G(u) = \int_\Omega \lambda(x) u(x)^2 \,\mathrm{d}x = \langle \lambda u,\, u\rangle_{H^{-1}\times H^1_0} = \|\lambda^{1/2} u\|^2_{L^2(\Omega)} \tag{2.30}$$

as goal functional. We note that $b(v,w) = \langle \lambda v,\, w\rangle_{H^{-1}\times H^1_0}$ and hence (2.17) holds with $g[w] = 2\lambda w$ and $\boldsymbol{g}[w] = 0$. Moreover, we observe that $\mathcal{K}u = \lambda u \in L^2(\Omega) \hookrightarrow H^{-1}(\Omega)$, where the embedding is compact, so that the goal functional from (2.30) fits in the setting of Theorem 2.2 and Theorem 2.5. We choose

$$\lambda(x) = \begin{cases} 1, & x \in U_1, \\ 0, & x \notin U_1, \end{cases}$$

with $U_1 = (0.25, 0.75)^2$. This functional is evaluated at the solution of equation (2.1) with $f = 2x(x-1) + 2y(y-1)$ and $\boldsymbol{f} = \boldsymbol{0}$. The solution of this equation, as well as the value of the goal functional, can be computed analytically to be $u = xy(1-x)(1-y)$ and $G(u) = \int_{U_1} u^2 \,\mathrm{d}x = \frac{41209}{58982400}$, respectively. The numerical results are visualized in Figure 2.2.

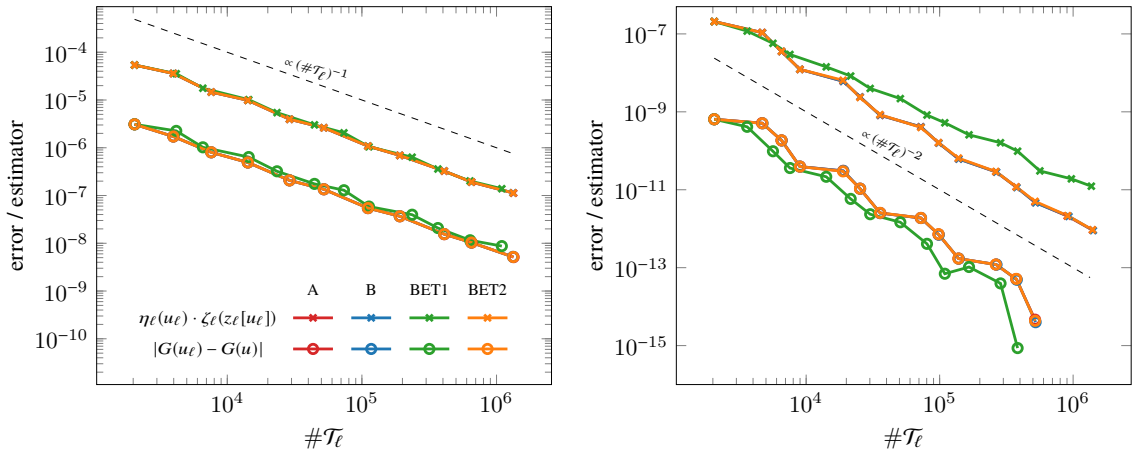


Figure 2.2: Convergence rates of estimator product and goal error for the problem setting in Section 2.3.1 with $p = 1$ (left) and $p = 2$ (right). Note that the lines marked with A, B, and BET2 are almost identical.

2.3.2 Nonlinear convection

Suppose that $\boldsymbol{\lambda} \in [L^\infty(\Omega)]^2$ is some vector field, whose regions of discontinuity are resolved by the initial mesh $\mathcal{T}_0$. As goal functional we consider the nonlinear convection term

$$G(u) = \int_\Omega u(x)\boldsymbol{\lambda}(x)\cdot\nabla u(x)\,\mathrm{d}x = \langle \boldsymbol{\lambda}\cdot\nabla u,\, u\rangle_{H^{-1}\times H^1_0}. \tag{2.31}$$

We note that $b(v,w) = \langle \boldsymbol{\lambda}\cdot\nabla v,\, w\rangle_{H^{-1}\times H^1_0}$ and hence (2.17) holds with $g[w] = \boldsymbol{\lambda}\cdot\nabla w$ and $\boldsymbol{g}[w] = -w\boldsymbol{\lambda}$. Moreover, we observe that $\mathcal{K}u = \boldsymbol{\lambda}\cdot\nabla u \in L^2(\Omega) \hookrightarrow H^{-1}(\Omega)$, where the embedding is compact, so that the goal functional from (2.31) fits in the setting of Theorem 2.2 and Theorem 2.5.

We compute the solutions to the primal and the dual problem for $f = 0$,

$$\boldsymbol{f}(x) = \begin{cases} \frac{1}{\sqrt{2}}(-1,1) & \text{if } x\in U_3,\\ 0 & \text{else,}\end{cases} \quad\text{and}\quad \boldsymbol{\lambda} = \frac{\sigma}{\sqrt{2}}\begin{pmatrix}-1\\1\end{pmatrix} \text{ with } \sigma = \begin{cases}1 & \text{if } x\in U_2,\\ -1 & \text{else.}\end{cases}$$

The sets $U_3 := \{x\in\Omega \,|\, x_1 - x_2 \geq 0.25\}$ and $U_2 := (0.5,1)\times(0,0.5)$ are shown in Figure 2.1. The numerical results are visualized in Figure 2.3. Note that the primal problem in this case exhibits a singularity which is not induced by the geometry and thus is not present in the dual problem.

2.3.3 Force evaluation

Let $\varepsilon > 0$ and let ψ be a cut-off function that satisfies

$$\psi(x) = 1 \text{ if } x\in U_1 \quad\text{and}\quad \psi(x) = 0 \text{ if } \operatorname{dist}(x,U_1) > \varepsilon.$$

For a given direction $\boldsymbol{\chi}\in\mathbb{R}^2$, consider a goal functional of the form

$$G(u) := \int_\Omega \nabla\psi\cdot(\nabla u\otimes\nabla u - \tfrac{1}{2}|\nabla u|^2 I)\boldsymbol{\chi}\,\mathrm{d}x. \tag{2.32}$$

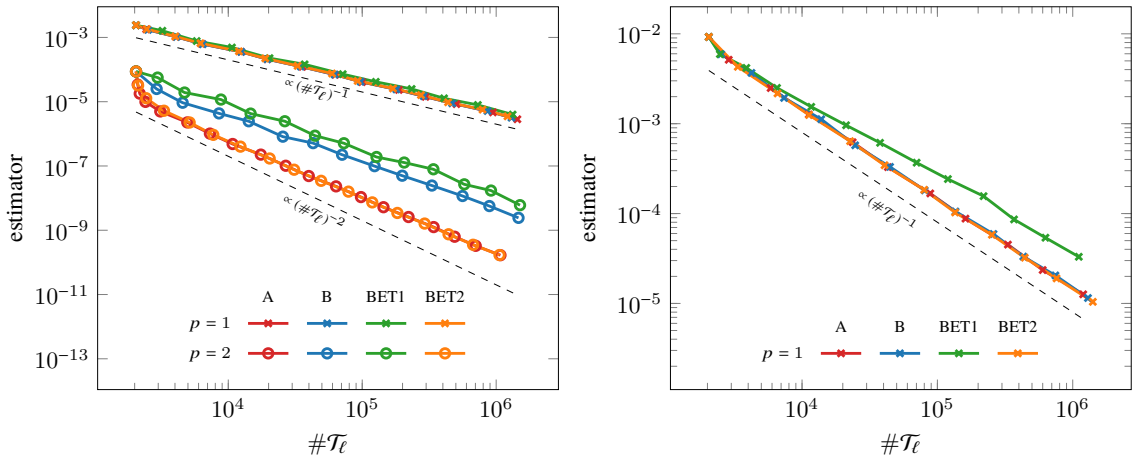


Figure 2.3: Convergence rates of estimator product for the problem setting in Section 2.3.2 (left) and Section 2.3.3 (right).

This approximates the electrostatic force which is exerted by an electric potential u on a charged body occupying the domain U_1 in direction $\boldsymbol{\chi}$ (the part of the integrand in brackets is the so-called Maxwell stress tensor). We note that

$$b(v, w) = \int_\Omega \nabla\psi \cdot \big(\nabla v \otimes \nabla w - \tfrac{1}{2}(\nabla v \cdot \nabla w) I\big)\boldsymbol{\chi} \,\mathrm{d}x$$

and hence (2.17) holds with $g[w] = 0$ and $\boldsymbol{g}[w] = (\nabla\psi \cdot \boldsymbol{\chi})\nabla w - (\nabla\psi \cdot \nabla w)\boldsymbol{\chi} - (\boldsymbol{\chi} \cdot \nabla w)\nabla\psi$. We stress that the goal functional from (2.32) does not fit in the setting of Theorem 2.2 and Theorem 2.5, since the corresponding operator $\mathcal{K}$ is not compact. Hence, we cannot guarantee optimal rates for our Algorithms 2A and 2B. However, Proposition 2.1 and Proposition 2.4 still guarantee convergence of our algorithms.

For our experiments, we choose $\boldsymbol{\chi} = \frac{1}{\sqrt{2}}(1, 1)^\top$, $f = 1$, and $\boldsymbol{f} = \boldsymbol{0}$. Furthermore, we choose ψ to be in $\mathcal{X}_0$ for $p = 1$, i.e., ψ is piecewise linear, and ε is chosen such that ψ falls off to 0 exactly within one layer of elements around U_1 in $\mathcal{T}_0$.

The results can be seen in Figure 2.3.

2.3.4 Discussion of numerical experiments

We clearly see from Figures 2.2–2.3 that our Algorithm 2A and BET2 outperform BET1 and sometimes even Algorithm 2B. From Figure 2.4, where we plot estimator product (and, if available, goal error) for different parameters $\theta = 0.1, 0.2, \dots, 1.0$, we see that this behavior does not depend on the marking parameter θ, generally speaking. It is striking that the strategy BET1 with $\theta < 1$ fails to drive down the estimator product at the same speed as uniform refinement. This is likely due to the fact that the linearization error $|||z_\ell[u] - z_\ell[u_\ell]|||$ is disregarded; see Remark 2.3.

In Figure 2.5, we plot the cumulative costs

$$\sum_{\ell \in S[\tau]} \#\mathcal{T}_\ell, \quad \text{with} \quad S[\tau] := \{\ell \in \mathbb{N} \,|\, \mathrm{error}_\ell \geq \tau\}, \tag{2.33}$$

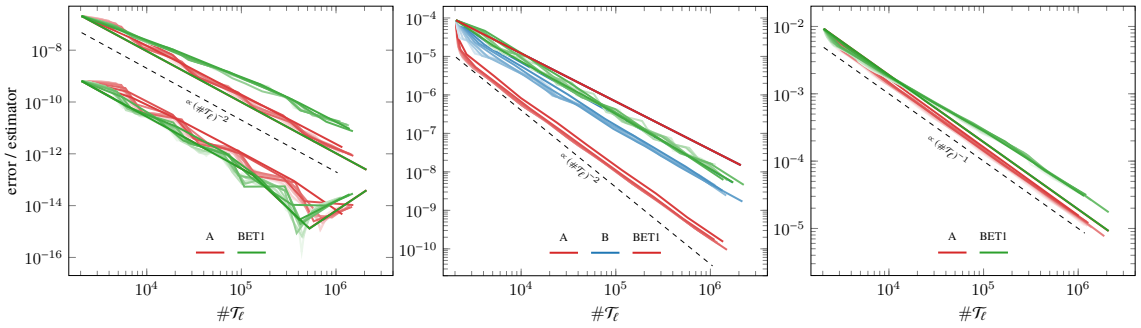


Figure 2.4: Variation of θ from 0.1 (light) to 1.0 (dark) in steps of 0.1. Left: Setting from Section 2.3.1 with $p = 2$, where the upper lines represent the estimator product and the lower ones the goal error. Middle: Estimator product for the setting from Section 2.3.2 with $p = 2$. Right: Estimator product for the setting from Section 2.3.3 with $p = 1$.

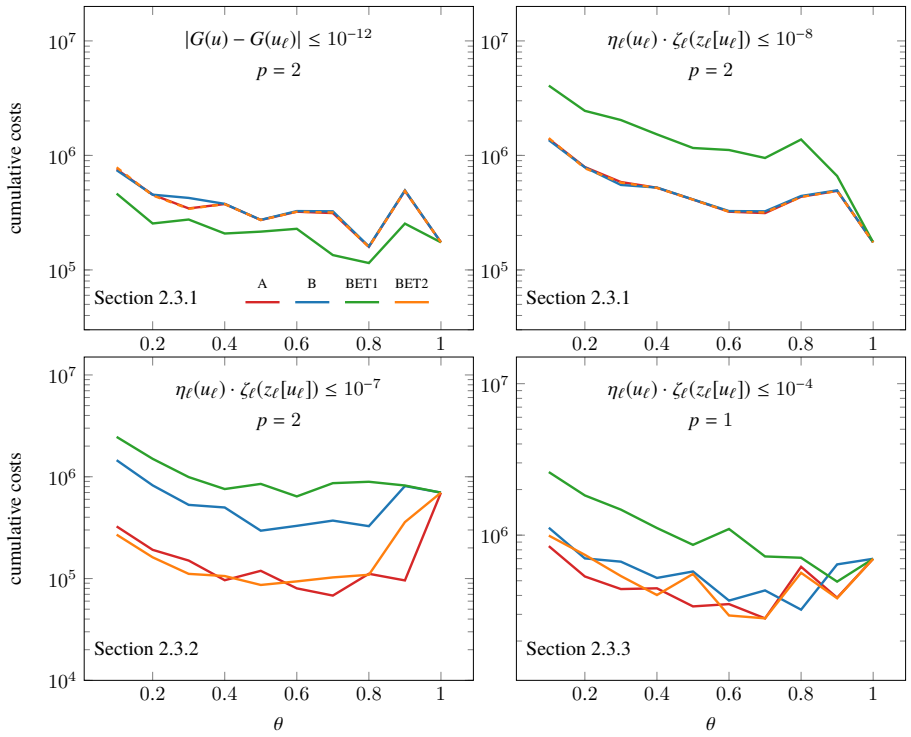


Figure 2.5: Cumulative costs (2.33) for estimator product and goal error for the setting of Section 2.3.1 (top), for the estimator product for the setting of Section 2.3.2 (bottom left), and for the estimator product for the setting of Section 2.3.3 (bottom right). The parameters θ are chosen uniformly in $[0.1, 1]$ with stepsize 0.1.

where error_ℓ is either the estimator product $\eta_\ell(u_\ell)\zeta_\ell(z_\ell[u_\ell])$, or the goal error $|G(u)-G(u_\ell)|$ in the ℓ-th step of the adaptive algorithm. We see that for the setting from Section 2.3.1, where no singularity occurs, optimal costs are achieved by uniform refinement, as is expected. For the goal error, which is not known in general, the strategy BET1 performs better than our Algorithms 2A and 2B. However, for the estimator product, which is the relevant quantity in most applications (since the error is unknown), it is inferior. In the other settings, where there is a singularity, our Algorithms 2A and 2B achieve their minimal cost around the value 0.7 for the marking parameter θ.

2.4 Auxiliary results

2.4.1 Axioms of adaptivity

Clearly, $z[w]$ and $z_H[w]$ depend linearly on w (since $\mathcal{K}$ is linear and hence $b(\cdot,\cdot)$ is bilinear). Moreover, we have the following stability estimates.

Lemma 2.6. *For all $w \in H^1_0(\Omega)$ and all $\mathcal{T}_H \in \mathbb{T}(\mathcal{T}_0)$, it holds that*

$$C_1^{-1}\,|||z_H[w]||| \le |||z[w]||| \le C_2\,|||w|||, \tag{2.34}$$

where $C_1 > 0$ depends only on $a(\cdot,\cdot)$, while $C_2 > 0$ depends additionally on the boundedness of $\mathcal{K}$.

Proof. The definition of the dual problem shows that

$$|||z[w]|||^2 \lesssim a(z[w],z[w]) \overset{(2.9)}{=} b(z[w],w) + b(w,z[w]) \lesssim |||w|||\,|||z[w]|||$$

and hence $|||z[w]||| \lesssim |||w|||$. Moreover, the stability of the Galerkin method yields that $|||z_H[w]||| \lesssim |||z[w]|||$. This concludes the proof. □

Next, we show that the combined estimator for the primal and dual problem satisfies the assumptions (A1)–(A4), where particular emphasis is put on (A3)–(A4). For the ease of presentation (and by abuse of notation), we use the same constants as for the original properties (A1)–(A4), even though they now depend additionally on the bilinear form $a(\cdot,\cdot)$ and the boundedness of $\mathcal{K}$.

Proposition 2.7. *Suppose (A1)–(A4) for η_H and ζ_H. Let $\mathcal{T}_H \in \mathbb{T}$ and $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$. Then, (A1)–(A4) hold also for the combined estimator $\big[\eta_H(\cdot)^2 + \zeta_H(\cdot)^2\big]^{1/2}$:*

(A1) *For all $v_h, w_h \in \mathcal{X}_h$, $v_H, w_H \in \mathcal{X}_H$, and $\mathcal{U}_H \subseteq \mathcal{T}_h \cap \mathcal{T}_H$, it holds that*

$$\Big|\big[\eta_h(\mathcal{U}_H,v_h)^2 + \zeta_h(\mathcal{U}_H,w_h)^2\big]^{1/2} - \big[\eta_H(\mathcal{U}_H,v_H)^2 + \zeta_H(\mathcal{U}_H,w_H)^2\big]^{1/2}\Big| \le C_{\text{stab}}\big[|||v_h - v_H||| + |||w_h - w_H|||\big].$$

(A2) *For all $v_H, w_H \in \mathcal{X}_H$, it holds that*

$$\big[\eta_h(\mathcal{T}_h\backslash\mathcal{T}_H,v_H)^2 + \zeta_h(\mathcal{T}_h\backslash\mathcal{T}_H,w_H)^2\big]^{1/2} \le q_{\text{red}}\big[\eta_H(\mathcal{T}_H\backslash\mathcal{T}_h,v_H)^2 + \zeta_H(\mathcal{T}_H\backslash\mathcal{T}_h,w_H)^2\big]^{1/2}.$$

(A3) *The Galerkin solutions* $u_H, z_H[u_H] \in \mathcal{X}_H$ *to* (2.11) *satisfy that*

$$|||u - u_H||| + |||z[u] - z_H[u_H]||| + |||z[u_H] - z_H[u_H]||| \le C_{\mathrm{rel}} \left[\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\right]^{1/2}.$$

(A4) *The Galerkin solutions* $u_H, z_H[u_H] \in \mathcal{X}_H$ *and* $u_h, z_h[u_h] \in \mathcal{X}_h$ *to* (2.11) *satisfy that*

$$|||u_h - u_H||| + |||z_h[u_h] - z_H[u_H]||| \le C_{\mathrm{drel}} \left[\eta_H(\mathcal{T}_H\backslash\mathcal{T}_h, u_H)^2 + \zeta_H(\mathcal{T}_H\backslash\mathcal{T}_h, z_H[u_H])^2\right]^{1/2}.$$

Proof. By the triangle inequality and $[a^2 + b^2]^{1/2} \le a + b$, (A1) follows from stability of η_H and ζ_H. Reduction (A2) follows directly from the corresponding properties of η_H and ζ_H. For (A3), we see with Lemma 2.6 that

$$|||z[u] - z_H[u_H]||| \le |||z[u] - z[u_H]||| + |||z[u_H] - z_H[u_H]||| \overset{(2.34)}{\lesssim} |||u - u_H||| + |||z[u_H] - z_H[u_H]|||.$$

Hence, (A3) follows from reliability of η_H and ζ_H. Discrete reliability (A4) follows from the same arguments. □

In the following, we recall some basic results of [CFPP14].

Lemma 2.8 (quasi-monotonicity of estimators [CFPP14, Lemma 3.6])**.** *Let* $w \in H^1_0(\Omega)$. *Let* $\mathcal{T}_H \in \mathbb{T}$ *and* $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$. *The properties* (A1)–(A3) *together with the Céa lemma guarantee that*

$$\eta_h(u_h)^2 \le C_{\mathrm{mon}}\, \eta_H(u_H)^2 \quad \text{as well as} \quad \zeta_h(z_h[w])^2 \le C_{\mathrm{mon}}\, \zeta_H(z_H[w])^2. \tag{2.35}$$

Moreover, the properties (A1)–(A3) *for the combined error estimator together with the Céa lemma show that*

$$\eta_h(u_h)^2 + \zeta_h(z_h[u_h])^2 \le C_{\mathrm{mon}} \left[\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\right]. \tag{2.36}$$

The constant $C_{\mathrm{mon}} > 0$ *depends only on the properties* (A1)–(A3) *and on the bilinear form* $a(\cdot,\cdot)$ *and the boundedness of* $\mathcal{K}$. □

Lemma 2.9 (generalized estimator reduction [CFPP14, Lemma 4.7])**.** *Let* $\mathcal{T}_H \in \mathbb{T}$ *and* $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$. *Let* $v_H \in \mathcal{X}_H$, $v_h \in \mathcal{X}_h$, *and* $\delta > 0$. *Then,*

- $\eta_h(v_h)^2 \le (1+\delta)\left[\eta_H(v_H)^2 - (1 - q_{\mathrm{red}}^2)\,\eta_H(\mathcal{T}_H\backslash\mathcal{T}_h, v_H)^2\right] + (1+\delta^{-1})\, C_{\mathrm{stab}}^2\, |||v_h - v_H|||^2$,
- $\zeta_h(v_h)^2 \le (1+\delta)\left[\zeta_H(v_H)^2 - (1 - q_{\mathrm{red}}^2)\,\zeta_H(\mathcal{T}_H\backslash\mathcal{T}_h, v_H)^2\right] + (1+\delta^{-1})\, C_{\mathrm{stab}}^2\, |||v_h - v_H|||^2$.

If, for instance, $\theta\, \eta_H(u_H)^2 \le \eta_H(\mathcal{T}_H\backslash\mathcal{T}_h, u_H)^2$ *with* $0 < \theta \le 1$, *then it follows that*

$$\eta_h(u_h)^2 \le q\, \eta_H(u_H)^2 + C\, |||u_h - u_H|||^2. \tag{2.37}$$

In this case, it holds that $0 < q := (1+\delta)\left[1 - (1 - q_{\mathrm{red}}^2)\,\theta\right] < 1$ *and* $C := (1+\delta^{-1})\, C_{\mathrm{stab}}^2$ *with* $\delta > 0$ *being sufficiently small.* □

Lemma 2.10 (optimality of Dörfler marking [CFPP14, Proposition 4.12])**.** *Suppose stability* (A1) *and discrete reliability* (A4)*. For all* $0 < \theta < \theta_{\mathrm{opt}} := (1 + C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2)^{-1}$*, there exists some* $0 < \kappa_{\mathrm{opt}} < 1$ *such that for all* $\mathcal{T}_\ell \in \mathbb{T}$ *and all* $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_\ell)$*, it holds that*

$$\eta_h(u_h)^2 \le \kappa_{\mathrm{opt}}\, \eta_\ell(u_\ell)^2 \quad \Longrightarrow \quad \theta\, \eta_\ell(u_\ell)^2 \le \eta_\ell(\mathcal{T}_\ell \backslash \mathcal{T}_h, u_\ell)^2, \tag{2.38}$$

$$\begin{aligned} &\big[\eta_h(u_h)^2 + \zeta_h(z_h[u_h])^2\big] \le \kappa_{\mathrm{opt}} \big[\eta_\ell^2 + \zeta_\ell(z_\ell[u_\ell])^2\big] \\ &\quad \Longrightarrow \quad \theta \big[\eta_\ell^2 + \zeta_\ell(z_\ell[u_\ell])^2\big] \le \big[\eta_\ell(\mathcal{T}_\ell \backslash \mathcal{T}_h, u_\ell)^2 + \zeta_\ell(\mathcal{T}_\ell \backslash \mathcal{T}_h, z_\ell[u_\ell])^2\big]. \end{aligned} \tag{2.39}$$

□

2.4.2 Quasi-orthogonality

To prove linear convergence in the spirit of [CKNS08], we imitate the approach from [BHP17]. One crucial ingredient are appropriate quasi-orthogonalities. To this end, our proofs exploit the observation of [FFP14] that, for any compact operator $C\colon H^1_0(\Omega) \to H^{-1}(\Omega)$, convergence $\|u - u_\ell\|_{H^1(\Omega)} \to 0$ plus Galerkin orthogonality for the nested discrete spaces $\mathcal{X}_\ell \subseteq \mathcal{X}_{\ell+1} \subset H^1_0(\Omega)$ for all $\ell \in \mathbb{N}_0$ even yields that $\|C(u - u_\ell)\|_{H^{-1}(\Omega)}/\|u - u_\ell\|_{H^1(\Omega)} \to 0$ as $\ell \to \infty$. The latter is also the key argument for the following two lemmas.

Lemma 2.11 (quasi-orthogonality for primal problem [BHP17, Lemma 18])**.** *Suppose that* $|||u - u_\ell||| \to 0$ *as* $\ell \to \infty$*. Then, for all* $0 < \varepsilon < 1$*, there exists* $\ell_0 \in \mathbb{N}$ *such that, for all* $\ell \ge \ell_0$ *and all* $n \in \mathbb{N}_0$*,*

$$|||u - u_{\ell+n}|||^2 + |||u_{\ell+n} - u_\ell|||^2 \le \frac{1}{1-\varepsilon}\, |||u - u_\ell|||^2. \tag{2.40}$$

□

The same result holds for the dual problem, if the algorithm ensures convergence $|||z[u] - z_\ell[u]||| \to 0$ as $\ell \to \infty$.

Lemma 2.12 (quasi-orthogonality for *exact* dual problem [BHP17, Lemma 18])**.** *Suppose that* $|||z[u] - z_\ell[u]||| \to 0$ *as* $\ell \to \infty$*. Then, for all* $0 < \varepsilon < 1$*, there exists* $\ell_0 \in \mathbb{N}$ *such that, for all* $\ell \ge \ell_0$ *and all* $n \in \mathbb{N}_0$*,*

$$|||z[u] - z_{\ell+n}[u]|||^2 + |||z_{\ell+n}[u] - z_\ell[u]|||^2 \le \frac{1}{1-\varepsilon}\, |||z[u] - z_\ell[u]|||^2. \tag{2.41}$$

□

Lemma 2.13 (combined quasi-orthogonality for *inexact* dual problem)**.** *Suppose that* $|||u - u_\ell||| + |||z[u] - z_\ell[u_\ell]||| \to 0$ *as* $\ell \to \infty$*. Then, for all* $0 < \delta < 1$*, there exists* $\ell_0 \in \mathbb{N}$ *such that, for all* $\ell \ge \ell_0$ *and all* $n \in \mathbb{N}_0$*,*

$$\begin{aligned} &\big[\, |||u - u_{\ell+n}|||^2 + |||z[u] - z_{\ell+n}[u_{\ell+n}]|||^2 \,\big] + \big[\, |||u_{\ell+n} - u_\ell||| + |||z_{\ell+n}[u_{\ell+n}] - z_\ell[u_\ell]|||^2 \,\big] \\ &\qquad \le \frac{1}{1-\delta} \big[\, |||u - u_\ell|||^2 + |||z[u] - z_\ell[u_\ell]|||^2 \,\big]. \end{aligned} \tag{2.42}$$

Proof. According to Lemma 2.6, it holds that

$$|||z[u] - z_\ell[u]||| \le |||z[u] - z_\ell[u_\ell]||| + |||z_\ell[u] - z_\ell[u_\ell]||| \overset{(2.34)}{\lesssim} |||z[u] - z_\ell[u_\ell]||| + |||u - u_\ell||| \xrightarrow{\ell\to\infty} 0.$$

Hence, we may exploit the conclusions of Lemma 2.11 and Lemma 2.12. For arbitrary $\alpha > 0$, the Young inequality guarantees that

$$\begin{aligned}
\|\!|z[u] - z_{\ell+n}[u_{\ell+n}]|\!\|^2 &\le (1+\alpha)\,\|\!|z[u] - z_{\ell+n}[u]|\!\|^2 + (1+\alpha^{-1})\,\|\!|z_{\ell+n}[u] - z_{\ell+n}[u_{\ell+n}]|\!\|^2,\\
\|\!|z_{\ell+n}[u_{\ell+n}] - z_\ell[u_\ell]|\!\|^2 &\le (1+\alpha)\,\|\!|z_{\ell+n}[u] - z_\ell[u]|\!\|^2 + (1+\alpha^{-1})^2\,\|\!|z_\ell[u] - z_\ell[u_\ell]|\!\|^2\\
&\qquad + (1+\alpha)(1+\alpha^{-1})\,\|\!|z_{\ell+n}[u] - z_{\ell+n}[u_{\ell+n}]|\!\|^2,\\
\|\!|z[u] - z_\ell[u_\ell]|\!\|^2 &\le (1+\alpha)\,\|\!|z[u] - z_\ell[u_\ell]|\!\|^2 + (1+\alpha^{-1})\,\|\!|z_\ell[u] - z_\ell[u_\ell]|\!\|^2.
\end{aligned}$$

Together with Lemma 2.12, this leads to

$$\begin{aligned}
&\|\!|z[u] - z_{\ell+n}[u_{\ell+n}]|\!\|^2 + \|\!|z_{\ell+n}[u_{\ell+n}] - z_\ell[u_\ell]|\!\|^2\\
&\quad\le (1+\alpha)\,\big[\,\|\!|z[u] - z_{\ell+n}[u]|\!\|^2 + \|\!|z_{\ell+n}[u] - z_\ell[u]|\!\|^2\,\big]\\
&\qquad + (2+\alpha)(1+\alpha^{-1})\,\|\!|z_{\ell+n}[u] - z_{\ell+n}[u_{\ell+n}]|\!\|^2 + (1+\alpha^{-1})^2\,\|\!|z_\ell[u] - z_\ell[u_\ell]|\!\|^2\\
&\quad\overset{(2.41)}{\le} \frac{1+\alpha}{1-\varepsilon}\,\|\!|z[u] - z_\ell[u]|\!\|^2 + (1+\alpha^{-1})^2\,\|\!|z_\ell[u] - z_\ell[u_\ell]|\!\|^2\\
&\qquad + (2+\alpha)(1+\alpha^{-1})\,\|\!|z_{\ell+n}[u] - z_{\ell+n}[u_{\ell+n}]|\!\|^2\\
&\quad\le \frac{(1+\alpha)^2}{1-\varepsilon}\,\|\!|z[u] - z_\ell[u_\ell]|\!\|^2 + \Big[(1+\alpha^{-1})^2 + \frac{(1+\alpha^{-1})(1+\alpha)}{1-\varepsilon}\Big]\,\|\!|z_\ell[u] - z_\ell[u_\ell]|\!\|^2\\
&\qquad + (2+\alpha)(1+\alpha^{-1})\,\|\!|z_{\ell+n}[u] - z_{\ell+n}[u_{\ell+n}]|\!\|^2
\end{aligned}\tag{2.43}$$

for all $0 < \varepsilon < 1$ and all $\ell \ge \ell_0$, where $\ell_0 \in \mathbb{N}_0$ depends only on ε. With the (compact) adjoint $\mathcal{K}' \colon H^1_0(\Omega) \to H^{-1}(\Omega)$ of $\mathcal{K}$, we note that

$$\begin{aligned}
\|\!|z_\ell[u] - z_\ell[u_\ell]|\!\|^2 = \|\!|z_\ell[u - u_\ell]|\!\|^2 &\lesssim a(z_\ell[u-u_\ell], z_\ell[u-u_\ell])\\
&= b(z_\ell[u-u_\ell], u-u_\ell) + b(u-u_\ell, z_\ell[u-u_\ell])\\
&= \langle \mathcal{K}(z_\ell[u-u_\ell]),\, u-u_\ell\rangle_{H^{-1}\times H^1_0} + \langle \mathcal{K}(u-u_\ell),\, z_\ell[u-u_\ell]\rangle_{H^{-1}\times H^1_0}\\
&= \langle \mathcal{K}'(u-u_\ell),\, z_\ell[u-u_\ell]\rangle_{H^{-1}\times H^1_0} + \langle \mathcal{K}(u-u_\ell),\, z_\ell[u-u_\ell]\rangle_{H^{-1}\times H^1_0}.\\
&\overset{(2.34)}{\lesssim} \big[\,\|\mathcal{K}'(u-u_\ell)\|_{H^{-1}(\Omega)} + \|\mathcal{K}(u-u_\ell)\|_{H^{-1}(\Omega)}\,\big]\,\|\!|u-u_\ell|\!\|.
\end{aligned}$$

Since $\mathcal{K}$ and $\mathcal{K}'$ are compact operators (according to the Schauder theorem), it follows from [FFP14, Lemma 3.5] (see also [BHP17, Lemma 17]) that

$$\big[\,\|\mathcal{K}'(u-u_\ell)\|_{H^{-1}(\Omega)} + \|\mathcal{K}(u-u_\ell)\|_{H^{-1}(\Omega)}\,\big] \le \widetilde{\kappa}_\ell\,\|\!|u-u_\ell|\!\| \quad\text{with}\quad 0 \le \widetilde{\kappa}_\ell \xrightarrow{\ell\to\infty} 0.$$

Combining the two last estimates, we see that

$$\|\!|z_\ell[u] - z_\ell[u_\ell]|\!\|^2 \le \kappa_\ell\,\|\!|u-u_\ell|\!\|^2 \quad\text{for all } \ell \in \mathbb{N}_0,\ \text{where } 0 \le \kappa_\ell \xrightarrow{\ell\to\infty} 0.\tag{2.44}$$

Plugging (2.44) into (2.43), we thus have shown that

$$\begin{aligned}
&\|\!|z[u] - z_{\ell+n}[u_{\ell+n}]|\!\|^2 + \|\!|z_{\ell+n}[u_{\ell+n}] - z_\ell[u_\ell]|\!\|^2\\
&\quad\le \frac{(1+\alpha)^2}{1-\varepsilon}\,\|\!|z[u] - z_\ell[u_\ell]|\!\|^2 + \Big[(1+\alpha^{-1})^2 + \frac{(1+\alpha^{-1})(1+\alpha)}{1-\varepsilon}\Big]\,\kappa_\ell\,\|\!|u-u_\ell|\!\|^2\\
&\qquad + (2+\alpha)(1+\alpha^{-1})\,\kappa_{\ell+n}\,\|\!|u-u_{\ell+n}|\!\|^2
\end{aligned}$$

for all $0 < \varepsilon < 1$, all $\alpha > 0$, and all $\ell \geq \ell_0$, where $\ell_0 \in \mathbb{N}_0$ depends only on ε. We combine this estimate with that of Lemma 2.11. This leads to

$$\begin{aligned}&\big[\, |||u - u_{\ell+n}|||^2 + |||z[u] - z_{\ell+n}[u_{\ell+n}]|||^2 \,\big] + \big[\, |||u_{\ell+n} - u_\ell||| + |||z_{\ell+n}[u_{\ell+n}] - z_\ell[u_\ell]|||^2 \,\big] \\ &\quad \leq C(\alpha,\varepsilon,\ell)\,\big[\, |||u - u_\ell|||^2 + |||z[u] - z_\ell[u_\ell]|||^2 \,\big] + (2+\alpha)(1+\alpha^{-1})\,\kappa_{\ell+n}\,|||u - u_{\ell+n}|||^2,\end{aligned}$$

where

$$C(\alpha,\varepsilon,\ell) := \max\left\{\frac{(1+\alpha)^2}{1-\varepsilon},\, \frac{1}{1-\varepsilon} + \left[(1+\alpha^{-1})^2 + \frac{(1+\alpha^{-1})(1+\alpha)}{1-\varepsilon}\right]\kappa_\ell\right\}$$

for all $0 < \varepsilon < 1$, all $\alpha > 0$, and all $\ell \geq \ell_0$, where $\ell_0 \in \mathbb{N}_0$ depends only on ε. For arbitrary $0 < \alpha, \beta, \varepsilon < 1$, there exists $\ell_0' \in \mathbb{N}_0$ such that for all $\ell \geq \ell_0'$, it holds that

$$(2+\alpha)(1+\alpha^{-1})\kappa_\ell \leq \beta$$

as well as

$$\frac{1}{1-\varepsilon} + \left[(1+\alpha^{-1})^2 + \frac{(1+\alpha^{-1})(1+\alpha)}{1-\varepsilon}\right]\kappa_\ell \leq \frac{(1+\alpha)^2}{1-\varepsilon}.$$

Hence, we are led to

$$\begin{aligned}&\big[\, |||u - u_{\ell+n}|||^2 + |||z[u] - z_{\ell+n}[u_{\ell+n}]|||^2 \,\big] + \big[\, |||u_{\ell+n} - u_\ell||| + |||z_{\ell+n}[u_{\ell+n}] - z_\ell[u_\ell]|||^2 \,\big] \\ &\quad \leq \frac{(1+\alpha)^2}{(1-\varepsilon)(1-\beta)}\,\big[\, |||u - u_\ell|||^2 + |||z[u] - z_\ell[u_\ell]|||^2 \,\big].\end{aligned} \tag{2.45}$$

Given $0 < \delta < 1$, we first fix $\alpha > 0$ such that $(1+\alpha)^2 < \frac{1}{1-\delta}$. Then, we choose $0 < \varepsilon, \beta < 1$ such that $\frac{(1+\alpha)^2}{(1-\varepsilon)(1-\beta)} \leq \frac{1}{1-\delta}$. The choices of ε and β also provide some index $\ell_0 \in \mathbb{N}_0$ such that estimate (2.45) holds for all $\ell \geq \ell_0$. This concludes the proof. □

2.5 Proof of plain convergence of Algorithm 2A and 2B

2.5.1 Algorithm 2A

First, we prove the upper bound for the goal error.

***Proof of Proposition 2.1*(i).** It holds that

$$\begin{aligned}\big|G(u) - G(u_H)\big| &\overset{(2.12)}{\lesssim} |||u - u_H|||\,\big[\, |||z[u_H] - z_H[u_H]||| + |||u - u_H||| \,\big] \\ &\overset{(A3)}{\lesssim} \eta_H(u_H)\,\big[\, \zeta_H(z_H[u_H]) + \eta_H(u_H) \,\big].\end{aligned}$$

The hidden constants depend only on the boundedness of $a(\cdot,\cdot)$ and $\mathcal{K}$, and on the constant C_{rel} from (A3). According to the Young inequality, this concludes the proof. □

Since Algorithm 2A linearizes the dual problem around the known discrete solution (i.e., it employs $z_\ell[u_\ell]$ instead of the non-computable $z_\ell[u]$), a first important observation is that Algorithm 2A ensures convergence for the primal solution. In particular, the following proposition allows to apply the quasi-orthogonalities from Section 2.4.2.

Proposition 2.14 (plain convergence of errors and error estimators)**.** *Suppose* (A1)–(A3)*. Then, for any choice of marking parameters* $0 < \theta \le 1$ *and* $C_{\mathrm{mark}} \ge 1$*, Algorithm 2A and Algorithm 2B guarantee that*

- $\|\!|u - u_\ell|\!\| + \eta_\ell(u_\ell) \to 0$ *if* $\#\{k \in \mathbb{N}_0 \mid \mathcal{M}_k$ *satisfies* (2.18a)$\} = \infty$,
- $\|\!|u - u_\ell|\!\| + \eta_\ell(u_\ell) + \|\!|z[u] - z_\ell[u_\ell]|\!\| + \|\!|z[u] - z_\ell[u]|\!\| + \zeta_\ell(z_\ell[u_\ell]) \to 0$ *if* $\#\{k \in \mathbb{N}_0 \mid \mathcal{M}_k$ *satisfies* (2.18b)$\} = \infty$,

as $\ell \to \infty$*. Moreover, at least one of these two cases is met.*

Proof. Since the discrete spaces are nested, it follows from the Céa lemma that there exists $u_\infty \in H^1_0(\Omega)$ such that

$$\|\!|u_\infty - u_\ell|\!\| \xrightarrow{\ell\to\infty} 0; \tag{2.46}$$

see, e.g., [AFP12; MSV08] or even the early work [BV84]. More precisely, u_∞ is the Galerkin approximation of u with respect to the "discrete limit space" $\mathcal{X}_\infty := \overline{\bigcup_{\ell=0}^\infty \mathcal{X}_\ell}$, where the closure is taken in $H^1_0(\Omega)$. Analogously, there exists $z_\infty \in H^1_0(\Omega)$ such that

$$\|\!|z_\infty - z_\ell[u_\infty]|\!\| \xrightarrow{\ell\to\infty} 0.$$

Together with (2.46) and Lemma 2.6, this also proves that

$$\begin{aligned}\|\!|z_\infty - z_\ell[u_\ell]|\!\| &\le \|\!|z_\infty - z_\ell[u_\infty]|\!\| + \|\!|z_\ell[u_\infty] - z_\ell[u_\ell]|\!\| \\ &\lesssim \|\!|z_\infty - z_\ell[u_\infty]|\!\| + \|\!|u_\infty - u_\ell|\!\| \xrightarrow{\ell\to\infty} 0.\end{aligned}$$

In the following, we aim to show that, in particular, $u = u_\infty$. To this end, the proof considers two cases:

⟨1⟩ There exists a subsequence $(\mathcal{T}_{\ell_k})_{k\in\mathbb{N}_0}$ such that $\mathcal{M}_{\ell_k}$ satisfies (2.18a) for all $k \in \mathbb{N}_0$,

⟨2⟩ There exists a subsequence $(\mathcal{T}_{\ell_k})_{k\in\mathbb{N}_0}$ such that $\mathcal{M}_{\ell_k}$ satisfies (2.18b) for all $k \in \mathbb{N}_0$.

Clearly, (at least) one of these subsequences is well-defined (i.e., there are infinitely many steps of the respective marking).

Case ⟨1⟩. According to Lemma 2.9, there exists $0 < q < 1$ and $C > 0$ such that

$$\eta_{\ell_{k+1}}(u_{\ell_{k+1}})^2 \overset{(2.37)}{\le} q\,\eta_{\ell_k}(u_{\ell_k})^2 + C\,\|\!|u_{\ell_{k+1}} - u_{\ell_k}|\!\|^2 \quad \text{for all } k \in \mathbb{N}_0.$$

With (2.46), the last estimate proves that the estimator subsequence is contractive up to some zero sequence. Therefore, it follows from basic calculus and reliability (A3) that

$$\|\!|u - u_{\ell_k}|\!\| \overset{(A3)}{\lesssim} \eta_{\ell_k}(u_{\ell_k}) \xrightarrow{k\to\infty} 0;$$

see, e.g., [AFP12, Lemma 2.3]. In particular, this proves that $u = u_\infty$ and hence $\|\!|u - u_\ell|\!\| \to 0$ as $\ell \to \infty$. Moreover, according to quasi-monotonicity (Lemma 2.8), the convergence of the subsequence $\eta_{\ell_k}(u_{\ell_k}) \to 0$ even yields that $\eta_\ell(u_\ell) \to 0$ as $\ell \to \infty$.

Case ⟨2⟩. We repeat the arguments from case ⟨1⟩. Instead of $\eta_H(u_H)^2$, we consider the combined estimator $\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2$. For all $k \in \mathbb{N}_0$, this leads to

$$\begin{aligned}\eta_{\ell_{k+1}}(u_{\ell_{k+1}})^2 + \zeta_{\ell_{k+1}}(z_{\ell_{k+1}}[u_{\ell_{k+1}}])^2 &\le q\,\big[\eta_{\ell_k}(u_{\ell_k})^2 + \zeta_{\ell_k}(z_{\ell_k}[u_{\ell_k}])^2\big] \\ &\quad + C\,\big[\,|||u_{\ell_{k+1}} - u_{\ell_k}|||^2 + |||z_{\ell_{k+1}}[u_{\ell_{k+1}}] - z_{\ell_k}[u_{\ell_k}]|||^2\,\big].\end{aligned}$$

As before, basic calculus reveals that

$$|||u - u_{\ell_k}|||^2 + |||z[u] - z_{\ell_k}[u_{\ell_k}]|||^2 \overset{(A3)}{\lesssim} \eta_{\ell_k}(u_{\ell_k})^2 + \zeta_{\ell_k}(z_{\ell_k}[u_{\ell_k}])^2 \xrightarrow{k\to\infty} 0.$$

In this case, we thus see that $u = u_\infty$ and $z[u] = z_\infty$ as well as estimator convergence $\eta_\ell(u_\ell) + \zeta_\ell(z_\ell[u_\ell]) \to 0$ as $\ell \to \infty$ (following now from Lemma 2.8). In any case, this concludes the proof. □

***Proof of Proposition 2.1*(ii).** Recall from Proposition 2.1(i) that

$$\big|G(u) - G(u_\ell)\big| \overset{(2.19)}{\lesssim} \eta_\ell(u_\ell)\,\big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big]^{1/2}.$$

Suppose $\#\{k \in \mathbb{N}_0 \,|\, \mathcal{M}_\ell \text{ satisfies (2.18a)}\} = \infty$. According to Proposition 2.14, it holds that $\eta_\ell(u_\ell) \to 0$ as $\ell \to \infty$. According to Lemma 2.8, it holds that $\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2 \lesssim \eta_0(u_0)^2 + \zeta_0(z_0[u_0])^2 < \infty$. Thus, the right-hand side of (2.20) vanishes. In the case $\#\{k \in \mathbb{N}_0 \,|\, \mathcal{M}_\ell \text{ satisfies (2.18b)}\} = \infty$ one can argue analogously. This concludes the proof. □

2.5.2 Algorithm 2B

First, we prove the upper bound.

***Proof of Proposition 2.4*(i).** From Proposition 2.1(i) it follows that

$$\begin{aligned}\big|G(u) - G(u_H)\big| &\overset{(2.19)}{\le} C'_{\mathrm{rel}}\,\eta_H(u_H)\,\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big]^{1/2} \\ &\le C'_{\mathrm{rel}}\,\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big].\end{aligned}$$

This proves the claim. □

***Proof of Proposition 2.4*(ii).** Recall from Proposition 2.4(i) that

$$\big|G(u) - G(u_\ell)\big| \overset{(2.19)}{\lesssim} \big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big].$$

Note that the marking step (2.25) of Algorithm 2B implies, in particular, that $\#\{k \in \mathbb{N}_0 \,|\, \mathcal{M}_\ell \text{ satisfies (2.18b)}\} = \infty$. According to Proposition 2.14, it holds that $\eta_\ell(u_\ell) + \zeta_\ell(z_\ell[u_\ell]) \to 0$ as $\ell \to \infty$. Thus, the right-hand side of (2.27) vanishes. This concludes the proof. □

2.6 Proof of Theorem 2.2

2.6.1 Linear convergence

Based on estimator reduction (Lemma 2.9) and quasi-orthogonality (Lemma 2.11, Lemma 2.13), we are in the position to address linear convergence.

Proposition 2.15 (generalized contraction)**.** *Suppose* (A1)–(A3). *Then, there exist constants* $\gamma > 0$ *and* $0 < q_{\mathrm{ctr}} < 1$ *such that the quasi-errors*

$$\Delta_H^u := |||u - u_H|||^2 + \gamma\, \eta_H(u_H)^2 \quad \text{and} \quad \Delta_H^z := |||z[u] - z_H[u_H]|||^2 + \gamma\, \zeta_H(z_H[u_H])^2, \tag{2.47}$$

defined for all $\mathcal{T}_H \in \mathbb{T}$, *satisfy the following contraction properties: There exists an index* $\ell_0 \geq 0$ *such that, for all* $\ell \in \mathbb{N}_0$ *with* $\ell \geq \ell_0$ *and all* $n \in \mathbb{N}$, *it holds that*

- $\Delta_{\ell+n}^u \leq q_{\mathrm{ctr}}\, \Delta_\ell^u$ *provided that* $\mathcal{M}_\ell$ *satisfies* (2.18a);
- $\big[\, \Delta_{\ell+n}^u + \Delta_{\ell+n}^z \,\big] \leq q_{\mathrm{ctr}} \big[\, \Delta_\ell^u + \Delta_\ell^z \,\big]$ *provided that* $\mathcal{M}_\ell$ *satisfies* (2.18b);

The constants γ *and* q_{ctr} *depend only on* θ, q_{red}, C_{stab}, *and* C_{rel}, *and the index* ℓ_0, *which depends on* γ *and* q_{ctr}, *is essentially provided by Lemma 2.11 and Lemma 2.13.*

Proof. Let $0 < \varepsilon, \delta, \gamma < 1$ be free parameters, which will be fixed later.

Step 1. Consider the case that $\#\{k \in \mathbb{N}_0 \,|\, \mathcal{M}_k \text{ satisfies (2.18a)}\} = \infty$ and that $\mathcal{M}_\ell$ satisfies (2.18a). From the generalized estimator reduction (Lemma 2.9), we get that

$$\eta_{\ell+n}(u_{\ell+n})^2 \overset{(2.9)}{\leq} q\, \eta_\ell(u_\ell)^2 + C\, |||u_{\ell+n} - u_\ell|||^2,$$

where $0 < q < 1$ depends only on θ and q_{red}, while $C > 0$ depends additionally on C_{stab}. Together with the quasi-orthogonality (Lemma 2.11), we see that

$$\begin{aligned}\Delta_{\ell+n}^u &= |||u - u_{\ell+n}|||^2 + \gamma\, \eta_\ell(u_{\ell+n})^2 \\ &\leq \frac{1}{1-\varepsilon}\, |||u - u_\ell|||^2 + \gamma q\, \eta_\ell(u_\ell)^2 + (\gamma C - 1)\, |||u_{\ell+n} - u_\ell|||^2.\end{aligned}$$

The choice of γ must enforce that $\gamma C \leq 1$. Together with reliability, we are then led to

$$\Delta_{\ell+n}^u \overset{(A3)}{\leq} \left[\, \frac{1}{1-\varepsilon} - \gamma\delta \right] |||u - u_\ell|||^2 + \gamma \left[\, q + \delta C_{\mathrm{rel}}^2 \,\right] \eta_\ell(u_\ell)^2.$$

The choice of $\delta > 0$ must guarantee that $q + \delta C_{\mathrm{rel}}^2 < 1$. Finally, the choice of $\varepsilon > 0$ must guarantee that $(1-\varepsilon)^{-1} - \gamma\delta < 1$. Then, we see that

$$\Delta_{\ell+n}^u \leq q_{\mathrm{ctr}}\, \Delta_\ell, \quad \text{where} \quad q_{\mathrm{ctr}} := \max\{(1-\varepsilon)^{-1} - \gamma\delta,\, q + \delta C_{\mathrm{rel}}^2\} < 1.$$

Step 2. Consider the case that $\#\{k \in \mathbb{N}_0 \,|\, \mathcal{M}_k \text{ satisfies (2.18b)}\} = \infty$ and that $\mathcal{M}_\ell$ satisfies (2.18b). The same arguments apply (now based on the combined quasi-orthogonality from Lemma 2.13).

Step 3. If $\ell_0^u := \#\{k \in \mathbb{N}_0 \,|\, \mathcal{M}_k \text{ satisfies (2.18a)}\} < \infty$, choose $\ell_0 > \ell_0^u$ as well as the free parameters according to Step 2.

Step 4. If $\ell_0^{uz} := \#\{k \in \mathbb{N}_0 \,|\, \mathcal{M}_k \text{ satisfies (2.18b)}\} < \infty$, choose $\ell_0 > \ell_0^{uz}$ as well as the free parameters according to Step 1.

Step 5. Finally, note that Step 3 and Step 4 are exclusive, since $\mathbb{N}_0 = \{k \in \mathbb{N}_0 \,|\, \mathcal{M}_k \text{ satisfies (2.18a)}\} \cup \{k \in \mathbb{N}_0 \,|\, \mathcal{M}_k \text{ satisfies (2.18b)}\}$. This concludes the proof. □

***Proof of Theorem 2.2*(i).** Recall the quasi-errors from (2.47). We first prove that $\Delta_H^u \simeq \eta_H(u_H)^2$ as well as $\big[\,\Delta_H^u + \Delta_H^z\,\big] \simeq \big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big]$. To see this, note that

$$\gamma\,\eta_H(u_H)^2 \le \Delta_H^u \overset{\text{(A3)}}{\le} (C_{\mathrm{rel}}^2 + \gamma)\,\eta_H(u_H)^2$$

as well as

$$\gamma\,\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big] \le \Delta_H^u + \Delta_H^z \overset{\text{(A3)}}{\le} (2\,C_{\mathrm{rel}}^2 + \gamma)\,\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big].$$

Let $\ell \ge \ell_0$. In n steps, the adaptive algorithm satisfies k times $\overline{\mathcal{M}}_\ell^u \subseteq \mathcal{M}_\ell$ and (at least) $n-k$ times $\overline{\mathcal{M}}_\ell^{uz} \subseteq \mathcal{M}_\ell$. From Proposition 2.15, we hence infer that

$$\eta_{\ell+n}(u_{\ell+n})^2 \le (C_{\mathrm{rel}}^2 + \gamma)\gamma^{-1}\,q_{\mathrm{ctr}}^k\,\eta_\ell(u_\ell)^2$$

as well as

$$\big[\,\eta_{\ell+n}(u_{\ell+n})^2 + \zeta_{\ell+n}(z_{\ell+n}[u_{\ell+n}])^2\big] \le (2\,C_{\mathrm{rel}}^2 + \gamma)\gamma^{-1}\,q_{\mathrm{ctr}}^{n-k}\,\big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big].$$

Multiplying these two estimates, we conclude the proof with $q_{\mathrm{lin}} = q_{\mathrm{ctr}}^{1/2}$ and $C_{\mathrm{lin}} = (2\,C_{\mathrm{rel}}^2+\gamma)/\gamma$. □

2.6.2 Optimal rates

Linear convergence, together with the following lemma, finally proves optimal rates for Algorithm 2A.

Lemma 2.16. *Suppose (A1)–(A4). For all $0 < \theta < \theta_{\mathrm{opt}} = (1 + C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2)^{-1}$, there exists $C_{\mathrm{aux}} > 0$ such that the following holds: For all $\mathcal{T}_H \in \mathbb{T}$, there exists some $\mathcal{R}_H \subseteq \mathcal{T}_H$ such that for all $s,t > 0$ with $\|u\|_{\mathbb{A}_s} + \|z[u]\|_{\mathbb{A}_t} < \infty$ and $\alpha = \min\{2s, s+t\}$, it holds that*

$$\#\mathcal{R}_H \le 2\,\big[\,C_{\mathrm{aux}}\,\|u\|_{\mathbb{A}_s}(\|u\|_{\mathbb{A}_s} + \|z[u]\|_{\mathbb{A}_t})\,\big]^{1/\alpha}\Big(\eta_H(u_H)\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big]^{1/2}\Big)^{-1/\alpha} \tag{2.48}$$

as well as the Dörfler marking (2.18), i.e.,

$$\begin{aligned} \theta\eta_H(u_H)^2 &\le \eta_H(\#\mathcal{R}_H, u_H)^2 \quad \text{or} \\ \theta\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big] &\le \eta_H(\#\mathcal{R}_H, u_H)^2 + \zeta_H(\#\mathcal{R}_H, z_H[u_H])^2. \end{aligned} \tag{2.49}$$

The constant C_{aux} depends only on θ and (A1)–(A4).

Proof. Adopt the notation of Lemma 2.10. According to Lemma 2.6, stability (A1) and reliability (A3), it holds that

$$\zeta_H(z_H[u_H]) \le C\big[\eta_H(u_H) + \zeta_H(z_H[u])\big] \text{ and } \zeta_H(z_H[u]) \le C\big[\eta_H(u_H) + \zeta_H(z_H[u_H])\big]$$

with $C := \max\{1, C_{\mathrm{stab}}C_{\mathrm{rel}}C_1C_2\}$. The quasi-monotonicity of the estimators (Lemma 2.8) and $\big[\,\eta_h(u_h)^2 + \zeta_h(z_h[u_h])^2\,\big]^{1/2} \le (C+1)\big[\,\eta_h(u_h) + \zeta_h(z_h[u])\,\big]$ yield that

$$\begin{aligned} \varepsilon &:= (C+1)^{-1}C_{\mathrm{mon}}^{-1}\kappa_{\mathrm{opt}}\,\eta_H(u_H)\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big]^{1/2} \\ &\le (C+1)^{-1}\kappa_{\mathrm{opt}}\,\eta_0(u_0)\big[\,\eta_0(u_0)^2 + \zeta_0(z_0[u_0])^2\,\big]^{1/2} \\ &< \eta_0(u_0)(\eta_0(u_0) + \zeta_0(z_0[u])) \le \|u\|_{\mathbb{A}_s}(\|u\|_{\mathbb{A}_s} + \|z[u]\|_{\mathbb{A}_t}) < \infty. \end{aligned}$$

Choose the minimal $N \in \mathbb{N}_0$ such that

$$\|u\|_{\mathbb{A}_s}(\|u\|_{\mathbb{A}_s} + \|z[u]\|_{\mathbb{A}_t}) \le \varepsilon\,(N+1)^{\alpha}.$$

From the choice of ε and the previous estimate, it follows that $N > 0$. Choose $\mathcal{T}_{\varepsilon_1}, \mathcal{T}_{\varepsilon_2} \in \mathbb{T}_N$ with $\eta_{\varepsilon_1}(u_{\varepsilon_1}) = \min_{\mathcal{T}_h \in \mathbb{T}_N} \eta_h(u_h)$ and $\zeta_{\varepsilon_2}(z_{\varepsilon_2}[u]) = \min_{\mathcal{T}_h \in \mathbb{T}_N} \zeta_h(z_h[u])$. Define $\mathcal{T}_\varepsilon := \mathcal{T}_{\varepsilon_1} \oplus \mathcal{T}_{\varepsilon_2}$ and $\mathcal{T}_h := \mathcal{T}_\varepsilon \oplus \mathcal{T}_H$. Then, Lemma 2.8, the definition of the approximation classes, and the choice of N and $\alpha = \min\{2s, s+t\}$ give that

$$\begin{aligned}
\eta_h(u_h)\big[\,\eta_h(u_h)^2 + \zeta_h(z_h[u_h])^2\,\big]^{1/2} &\le C_{\mathrm{mon}}\eta_{\varepsilon_1}(u_{\varepsilon_1})\big[\,\eta_{\varepsilon_1}(u_{\varepsilon_1})^2 + \zeta_{\varepsilon_2}(z_{\varepsilon_2}[u_{\varepsilon_2}])^2\,\big]^{1/2}\\
&\le (C+1)C_{\mathrm{mon}}\eta_{\varepsilon_1}(u_{\varepsilon_1})\,\big[\,\eta_{\varepsilon_1}(u_{\varepsilon_1}) + \zeta_{\varepsilon_2}(z_{\varepsilon_2}[u])\,\big]\\
&\le (C+1)C_{\mathrm{mon}}\big((N+1)^{-(2s)}\|u\|_{\mathbb{A}_s}^2 + (N+1)^{-(s+t)}\|u\|_{\mathbb{A}_s}\|z[u]\|_{\mathbb{A}_t}\big)\\
&\le (C+1)C_{\mathrm{mon}}(N+1)^{-\alpha}\|u\|_{\mathbb{A}_s}(\|u\|_{\mathbb{A}_s} + \|z[u]\|_{\mathbb{A}_t})\\
&\le (C+1)C_{\mathrm{mon}}\varepsilon = \kappa_{\mathrm{opt}}\,\eta_H(u_H)\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big]^{1/2}.
\end{aligned}$$

This implies that $\big[\,\eta_h(u_h)^2 + \zeta_h(z_h[u_h])^2\,\big] \le \kappa_{\mathrm{opt}}\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big]$ or $\eta_h(u_h)^2 \le \kappa_{\mathrm{opt}}\,\eta_H(u_H)^2$. Lemma 2.10 hence proves (2.49) with $\mathcal{R}_H := \mathcal{T}_H \backslash \mathcal{T}_h$. It remains to derive (2.48). To that end, define

$$\widetilde{C} := \big[\,\|u\|_{\mathbb{A}_s}(\|u\|_{\mathbb{A}_s} + \|z[u]\|_{\mathbb{A}_t})\,C_{\mathrm{mon}}\kappa_{\mathrm{opt}}^{-1}\,\big]^{1/\alpha}.$$

Then, minimality of $N \in \mathbb{N}_0$ and $N > 0$ yield that

$$N < \big[\,\|u\|_{\mathbb{A}_s}(\|u\|_{\mathbb{A}_s} + \|z[u]\|_{\mathbb{A}_t})\,\big]^{1/\alpha}\varepsilon^{-1/\alpha} = \widetilde{C}\left(\eta_H(u_H)\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big]^{1/2}\right)^{-1/\alpha}.$$

According to the choice of $\mathcal{T}_h$ and $\mathcal{R}_H$, the overlay estimate (2.15) yields that

$$\begin{aligned}
\#\mathcal{R}_H = \#(\mathcal{T}_H\backslash\mathcal{T}_h) &\overset{(2.13)}{\le} \#\mathcal{T}_h - \#\mathcal{T}_H \overset{(2.15)}{\le} \#\mathcal{T}_\varepsilon - \#\mathcal{T}_0 \overset{(2.15)}{\le} \#\mathcal{T}_{\varepsilon_1} + \#\mathcal{T}_{\varepsilon_2} - 2\,\#\mathcal{T}_0 \le 2N\\
&< 2\widetilde{C}\left(\eta_H(u_H)\big[\,\eta_H(u_H)^2 + \zeta_H(z_H[u_H])^2\,\big]^{1/2}\right)^{-1/\alpha}.
\end{aligned} \tag{2.50}$$

Overall, we conclude (2.48) with $C_{\mathrm{aux}} = C_{\mathrm{mon}}/\kappa_{\mathrm{opt}}$. □

Proof of Theorem 2.2(ii). According to (2.49) of Lemma 2.16 and the marking strategy in Algorithm 2A, for all $j \in \mathbb{N}_0$, it holds that

$$\#\mathcal{M}_j \le 2\,\min\{\#\overline{\mathcal{M}}_j^{u}, \#\overline{\mathcal{M}}_j^{uz}\} \le 2C_{\mathrm{mark}}\,\#\mathcal{R}_j.$$

With $\alpha = \min\{2s, s+t\} > 0$, estimate (2.48) of Lemma 2.16 implies that

$$\#\mathcal{M}_j \le 4\,C_{\mathrm{mark}}\big[\,C_{\mathrm{aux}}\,\|u\|_{\mathbb{A}_s}(\|u\|_{\mathbb{A}_s} + \|z\|_{\mathbb{A}_t})\,\big]^{1/\alpha}\left(\eta_j(u_j)\big[\,\eta_j(u_j)^2 + \zeta_j(z_j[u_j])^2\,\big]^{1/2}\right)^{-1/\alpha}.$$

With the mesh-closure estimate (2.14), we obtain that

$$\#\mathcal{T}_\ell - \#\mathcal{T}_0 \overset{(2.14)}{\le} C_{\mathrm{mesh}}\sum_{j=0}^{\ell-1}\#\mathcal{M}_j = C_{\mathrm{mesh}}\Big(\sum_{j=0}^{\ell_0-1}\#\mathcal{M}_j + \sum_{j=\ell_0}^{\ell-1}\#\mathcal{M}_j\Big).$$

Using that $\#\mathcal{M}_j \le \#\mathcal{T}_{j+1} - \#\mathcal{T}_j$, we get that

$$\begin{aligned}\#\mathcal{T}_\ell - \#\mathcal{T}_0 &\le C_{\mathrm{mesh}}\Big(\#\mathcal{T}_{\ell_0} - \#\mathcal{T}_0 + \sum_{j=\ell_0}^{\ell-1} \#\mathcal{M}_j\Big) \le C_{\mathrm{mesh}}\big(\#\mathcal{T}_{\ell_0} - \#\mathcal{T}_0 + 1\big) \sum_{j=\ell_0}^{\ell-1} \#\mathcal{M}_j \\ &\lesssim \sum_{j=\ell_0}^{\ell-1} \Big(\eta_j(u_j)\big[\,\eta_H(u_j)^2 + \zeta_j(z_j[u_j])^2\,\big]^{1/2}\Big)^{-1/\alpha}.\end{aligned} \tag{2.51}$$

Linear convergence (2.21) implies that

$$\eta_\ell(u_\ell)\big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big]^{1/2} \le C_{\mathrm{lin}}\, q_{\mathrm{lin}}^{\ell-j}\, \eta_j(u_j)\big[\,\eta_j(u_j)^2 + \zeta_j(z_j[u_j])^2\,\big]^{1/2}$$

for all $0 \le j \le \ell$ and hence

$$\begin{aligned}&\Big(\eta_j(u_j)\big[\,\eta_H(u_j)^2 + \zeta_j(z_j[u_j])^2\,\big]^{1/2}\Big)^{-1/\alpha} \\ &\qquad\le C_{\mathrm{lin}}^{1/\alpha}\, q_{\mathrm{lin}}^{(\ell-j)/\alpha}\Big(\eta_\ell(u_\ell)\big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big]^{1/2}\Big)^{-1/\alpha}.\end{aligned}$$

With $0 < q := q_{\mathrm{lin}}^{1/\alpha} < 1$, the geometric series applies and yields that

$$\begin{aligned}&\sum_{j=\ell_0}^{\ell-1} \Big(\eta_j(u_j)\big[\,\eta_H(u_j)^2 + \zeta_j(z_j[u_j])^2\,\big]^{1/2}\Big)^{-1/\alpha} \\ &\qquad\le C_{\mathrm{lin}}^{1/\alpha}\Big(\eta_\ell(u_\ell)\big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big]^{1/2}\Big)^{-1/\alpha} \sum_{j=\ell_0}^{\ell-1} q^{\ell-j} \\ &\qquad\le \frac{C_{\mathrm{lin}}^{1/\alpha}}{1 - q_{\mathrm{lin}}^{1/\alpha}}\Big(\eta_\ell(u_\ell)\big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big]^{1/2}\Big)^{-1/\alpha}.\end{aligned}$$

Combining this with (2.51), we obtain that

$$\begin{aligned}\#\mathcal{T}_\ell - \#\mathcal{T}_0 \le 4\,\frac{C_{\mathrm{mesh}} C_{\mathrm{mark}}}{1 - q_{\mathrm{lin}}^{1/\alpha}}\big(\#\mathcal{T}_{\ell_0} - \#\mathcal{T}_0 + 1\big)\,\big[\,C_{\mathrm{lin}} C_{\mathrm{aux}}\,\|u\|_{\mathbb{A}_s}\big(\|u\|_{\mathbb{A}_s} + \|z\|_{\mathbb{A}_t}\big)\,\big]^{1/\alpha} \\ \Big(\eta_\ell(u_\ell)\big[\,\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2\,\big]^{1/2}\Big)^{-1/\alpha}.\end{aligned}$$

Altogether, we conclude (2.22) with $C_{\mathrm{opt}} = \widetilde{C}_{\mathrm{opt}}^{1+\alpha}\big(\#\mathcal{T}_{\ell_0} - \#\mathcal{T}_0 + 1\big)^{1+\alpha}/(1 - q_{\mathrm{lin}}^{1/\alpha})^\alpha$ and $\widetilde{C}_{\mathrm{opt}} := \max\{C_{\mathrm{lin}} C_{\mathrm{aux}}, 4\, C_{\mathrm{mesh}} C_{\mathrm{mark}}\}$. □

2.7 Proof of Theorem 2.5

In contrast to the corresponding results for Algorithm 2A, the proof of Theorem 2.5 (for Algorithm 2B) follows essentially from the abstract setting of [CFPP14].

Proof of Theorem 2.5. (i) Note that (2.25) coincides with (2.18b). Hence, Proposition 2.15 can be applied and results in

$$\big[\, \Delta_{\ell+n}^u + \Delta_{\ell+n}^z \,\big] \le q_{\mathrm{ctr}} \big[\, \Delta_\ell^u + \Delta_\ell^z \,\big] \quad \text{for all } \ell, n \in \mathbb{N}_0 \text{ with } \ell \ge \ell_0.$$

For $n = 1$, we conclude contraction and hence linear convergence

$$\begin{aligned}\big[\, \eta_k(u_k)^2 + \zeta_k(z_k[u_k])^2 \,\big] &\simeq \big[\, \Delta_k^u + \Delta_k^z \,\big] \\ &\le q_{\mathrm{ctr}}^{k-\ell} \big[\, \Delta_\ell^u + \Delta_\ell^z \,\big] \simeq q_{\mathrm{ctr}}^{k-\ell} \big[\, \eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2 \,\big] \quad \text{for all } k \ge \ell \ge \ell_0.\end{aligned}$$

(ii) Since Proposition 2.7 shows that there hold (A1)–(A4) even for the combined estimator, [CFPP14, Theorem 4.1(ii)] guarantees convergence with optimal rates according to the approximation class

$$\|(u, z[u])\|_{\mathbb{A}_\beta} := \sup_{N\in\mathbb{N}_0} \Big((N+1)^\beta \min_{\mathcal{T}_H \in \mathbb{T}_N} \big[\eta_H(u)^2 + \zeta_H(z_H[u])^2 \big]^{1/2} \Big) \in \mathbb{R}_{\ge 0} \cup \{\infty\}$$

with $\beta > 0$. In particular, there exists a constant $\widetilde{C}_{\mathrm{opt}} > 0$ such that

$$\sup_{\ell\in\mathbb{N}_0} \frac{[\eta_\ell(u_\ell)^2 + \zeta_\ell(z_\ell[u_\ell])^2]^{1/2}}{(\#\mathcal{T}_\ell - \#\mathcal{T}_0 + 1)^{-\beta}} \le \widetilde{C}_{\mathrm{opt}} \, \|(u, z[u])\|_{\mathbb{A}_\beta} \tag{2.52}$$

for all $\beta > 0$. For $N \in \mathbb{N}_0$ choose $\mathcal{T}_{H^u}, \mathcal{T}_{H^z} \in \mathbb{T}_N$ such that

$$\eta_{H^u}(u_{H^u}) = \min_{\mathcal{T}_\star\in\mathbb{T}_N} \eta_\star(u_\star), \quad \zeta_{H^z}(z_{H^z}[u]) = \min_{\mathcal{T}_\star\in\mathbb{T}_N} \zeta_\star(z_\star[u]). \tag{2.53}$$

Then, for the overlay $\mathcal{T}_H := \mathcal{T}_{H^u} \oplus \mathcal{T}_{H^z}$, it holds that

$$\#\mathcal{T}_H - \#\mathcal{T}_0 \le \#\mathcal{T}_{H^u} + \#\mathcal{T}_{H^z} - 2\#\mathcal{T}_0 \le 2N$$

and thus $\mathcal{T}_H \in \mathbb{T}_{2N}$. From the minimality assumption in (2.53), we infer that

$$\begin{aligned}(2N+1)^\beta \big[\, \eta_H(u_H)^2 + \zeta_H(z_H[u])^2 \,\big]^{1/2} &\le 2^\beta \big[\, (N+1)^\beta \eta_H(u_H) + (N+1)^\beta \zeta_H(z_H[u]) \,\big] \\ &\overset{(2.35)}{\le} 2^\beta \big[\, C_{\mathrm{mon}}(N+1)^\beta \eta_{H^u}(u_{H^u}) + C_{\mathrm{mon}}(N+1)^\beta \zeta_{H^z}(z_{H^z}[u]) \,\big] \\ &\overset{(2.53)}{\le} 2^\beta C_{\mathrm{mon}} \big[\, \|u\|_{\mathbb{A}_\beta} + \|z\|_{\mathbb{A}_\beta} \,\big].\end{aligned}$$

Hence, we have $\|(u, z[u])\|_{\mathbb{A}_\beta} \lesssim \|u\|_{\mathbb{A}_s} + \|z\|_{\mathbb{A}_t}$ for $\beta \le \min\{s, t\}$. From this and (2.52), we obtain (2.29) by squaring, thus doubling the rate, i.e. $\beta = 2\alpha$. □

3 Goal-oriented adaptive finite element methods with optimal computational complexity

Sections 3.2–3.6 of this chapter are taken from:
R. Becker, G. Gantner, M. Innerberger, and D. Praetorius. Goal-oriented adaptive finite element methods with optimal computational complexity, 2021. arXiv: 2101.11407

3.1 Introduction

In the general introduction in Chapter 1, we have assumed that the discrete FEM equations (1.6) and (1.15) can be solved exactly. This viewpoint is convenient from an analysis point of view, since it allows to concentrate on the essential features of the algorithm under investigation. In practice, however, the discrete FEM equations correspond to linear systems of the form

$$Ax^\star = b, \quad \text{with } x^\star, b \in \mathbb{R}^N, A \in \mathbb{R}^{N\times N}, \tag{3.1}$$

for some $N \in \mathbb{N}$. Usually, N is large and the system matrix A is sparse (because of the local support of FEM basis functions). Such systems can hardly be solved exactly on a computer, which operates with finite precision arithmetic. Rather, we obtain an approximate solution $x \approx x^\star$ of (3.1), which corresponds to a FEM solution u_H on a mesh $\mathcal{T}_H$ by (1.8). To make the notation more explicit in this chapter, we denote the exact FEM solution corresponding to $x^\star$ by $u_H^\star$.

The solution of linear systems can be computed directly, e.g., by Gaussian elimination or factorization, where the error $\|x^\star - x\|$ is only caused by rounding errors of the involved arithmetical operations; or iteratively, e.g., by the conjugate gradient method or a multigrid method, where additionally an *iteration error* occurs: An iterative solver computes the solution of (3.1) by starting with an initial guess $x^0 \in \mathbb{R}^N$ and then iteratively updating this guess by a (preferably inexpensive) computation to obtain the next iterates $x^1, x^2, \dots$ and so on. The iteration is stopped before the exact solution $x^\star$ is reached. We denote FEM functions corresponding to iterates of an iterative solver by the same upper indices.

For large linear systems, direct methods become prohibitively expensive, leaving only iterative methods to choose from. This claim is also supported by the numerical experiments in Chapter 5, where runtimes for different tasks of actual AFEM computations are shown; solving (3.1) directly is by far the most costly task and its runtime grows super-linearly. In this chapter, we therefore consider iterative solvers like the preconditioned CG method with optimal multilevel additive Schwarz preconditioner [CNX12] or the geometric multigrid method [WZ17] for the solution of the primal and dual solution $u_H^\star, z_H^\star$ in the Solve step of Algorithm 1E. These impose the additional restriction that A is symmetric positive definite. Hence, we drop the non-symmetric convection

term from (1.2) to arrive at

$$\begin{aligned} -\operatorname{div} \boldsymbol{A}\nabla u^\star + cu^\star &= f + \operatorname{div} \boldsymbol{f} && \text{in } \Omega, \\ u^\star &= 0 && \text{on } \Gamma := \partial\Omega \end{aligned} \tag{3.2}$$

as this chapter's model problem, where the star is added also for the continuous problem to be consistent in the notation. With the goal functional (1.13), the sought goal value then reads

$$G(u^\star) = \int_\Omega \left(gu^\star - \boldsymbol{g} \cdot \nabla u^\star\right) \mathrm{d}x. \tag{3.3}$$

We replace the Solve step of Algorithm 1E by an iterative solver. The two main questions of this chapter are when to stop this solver and what consequences does this have for the optimality analysis outlined in Section 1.3.

Stopping the iterative solver

For standard AFEM (for possibly nonlinear problems) driven by residual error estimators of the energy norm error, the works [GHPS18; GHPS21] have shown that it suffices to solve the linear system up to an accuracy that is comparable with the achievable accuracy on the current mesh. Since neither of these accuracies is known, they use surrogates: For a stopping parameter $\lambda > 0$ they stop the iterative solver as soon as

$$|||u_H^k - u_H^{k-1}||| \leq \lambda\, \eta_H(u_H^k), \tag{3.4}$$

where $u_H^k \approx u_H^\star$ are the discrete approximations arising from the iterative solver (i.e., u_H^k corresponds to the coefficient vector x^k). Note that both sides of this inequality only involve available iterates and, hence, are computable. We remark that the solver stopping criterion (3.4) is essentially similar to the one used in [MS09; Ste07].

For GOAFEM, the linear system of the primal and dual problem have to be solved in every step and a stopping criterion needs to take both into account. One possibility is to stop the solver separately for primal and dual problem as soon as the respective stopping criterion is met. With Algorithm 3A below, we formulate a GOAFEM algorithm that takes into account iterative solution of primal and dual problem, where stopping the iterative solver is done for both problems separately according to (3.4).

Since, for efficiency reasons, primal and dual solution are often iterated simultaneously, one might not want to treat both problems separately. We suggest further stopping criteria reflecting this desire in Section 3.3.2 and note that our analysis holds for all presented criteria.

Optimal computational cost

Not having the exact discrete solutions $u_H^\star, z_H^\star$ available influences the goal error estimate. Similarly to Chapter 2, the error of the available approximate goal value $|G(u^\star) - G(u_H^k)|$ cannot be estimated by a product of energy norms as in (1.17) and an additional term appears in this upper bound. In contrast to Chapter 2, however, the additional term can be computed. Using this term as a correction, we arrive at a discrete quantity of interest $G_H(u_H^k, z_H^k)$ that acts as an approximate goal value and by means of which an analogue of (1.17) can be recovered.

With the discrete quantity of interest, we are able to prove optimality in the sense of Section 1.3 under two additional assumptions: the iterative solver is contractive (see (3.10) below) and the stopping parameter $\lambda > 0$ is sufficiently small. This also gives a positive answer to the question if the optimality analysis presented in the introduction is robust with respect to numerical errors.

Under the specified assumptions even a stronger notion of optimality can be shown. We not only prove that the error $|G(u^\star) - G_\ell(u_\ell^k, z_\ell^k)|$ of the final iterates u_ℓ^k, z_ℓ^k decays with optimal rate with respect to $\#\mathcal{T}_\ell$, but also that the error $|G(u^\star) - G_\ell(u_\ell^k, z_\ell^k)|$ of *all* iterates decays even with optimal rate with respect to the *overall computational cost*. The latter concept also takes into account how computationally expensive it is to compute the solutions on every level, i.e., the number of solver steps made; see (3.27) below.

Chapter outline

We first present the details of our GOAFEM algorithm (Algorithm 3A) in Section 3.2: the finite element discretization, the precise assumptions for the iterative solver, the marking strategy, and the error estimators. In Section 3.3 we state our main results: The first one is Theorem 3.5, full linear convergence for arbitrary stopping parameter λ, i.e., there holds linear convergence for a suitable quasi-error quantity for every range of steps of the adaptive algorithm, regardless of the type of step (solver steps or mesh refinement). This provides the key argument for proving Theorem 3.7, convergence with optimal rates with respect to the total computational cost if the adaptivity parameters are sufficiently small. To complete this section, we comment on alternative termination criteria for the iterative solver. After we give some numerical experiments to underline our analysis in Section 3.4, we finally prove our main Theorems in Section 3.5 and Section 3.6, respectively.

3.2 Goal-oriented adaptive finite element method

3.2.1 Variational formulation

Defining the symmetric bilinear form

$$a(u, v) := \int_\Omega \boldsymbol{A}\nabla u \cdot \nabla v \,\mathrm{d}x + \int_\Omega cuv \,\mathrm{d}x, \tag{3.5}$$

we suppose that $a(\cdot,\cdot)$ is continuous and elliptic on $H^1_0(\Omega)$ and thus fits into the setting of the Lax–Milgram lemma, i.e., there exist constants $0 < C_{\mathrm{ell}} \le C_{\mathrm{cnt}} < \infty$ such that

$$C_{\mathrm{ell}}\|u\|^2_{H^1_0(\Omega)} \le a(u,u) \quad \text{and} \quad a(u,v) \le C_{\mathrm{cnt}}\|u\|_{H^1_0(\Omega)}\|v\|_{H^1_0(\Omega)} \quad \text{for all } u, v \in H^1_0(\Omega).$$

In particular, $a(\cdot,\cdot)$ is a scalar product that yields an equivalent norm $|||v|||^2 := a(v,v)$ on $H^1_0(\Omega)$. The weak formulation of (3.2) reads

$$a(u^\star, v) = F(v) := \int_\Omega \big(fv \,\mathrm{d}x - \boldsymbol{f}\cdot\nabla v\big)\,\mathrm{d}x \quad \text{for all } v \in H^1_0(\Omega). \tag{3.6}$$

The Lax–Milgram lemma proves existence and uniqueness of the solution $u^\star \in H^1_0(\Omega)$ of (3.6). The same argument applies and proves that the dual problem

$$a(v, z^\star) = G(v) \quad \text{for all } v \in H^1_0(\Omega) \tag{3.7}$$

admits a unique solution $z^\star \in H^1_0(\Omega)$, where the linear goal functional $G \in H^{-1}(\Omega) := H^1_0(\Omega)'$ is defined by (3.3).

Remark 3.1. *For ease of presentation, we restrict our model problem (3.2) to homogeneous Dirichlet boundary conditions. We note, however, that for mixed homogeneous Dirichlet and inhomogeneous Neumann boundary conditions our main results hold true with the obvious modifications. In particular, with the partition $\partial\Omega = \overline{\Gamma}_D \cup \overline{\Gamma}_N$ into Dirichlet boundary Γ_D with $|\Gamma_D| > 0$ and Neumann boundary Γ_N, the space $H^1_0(\Omega)$ (and its discretization) has to be replaced by $H^1_D(\Omega) := \{v \in H^1(\Omega) \,|\, v|_{\Gamma_D} = 0$ in the sense of traces$\}$ and the Neumann data has to be given in $L^2(\Gamma_N)$. Furthermore, the coefficient $\boldsymbol{f}$ must vanish in a neighborhood of Γ_N to go from the strong form (3.2) to the weak form (3.6) via integration by parts.*

3.2.2 Finite element discretization and solution

For a conforming triangulation $\mathcal{T}_H$ of Ω into compact simplices and a polynomial degree $p \geq 1$, let

$$\mathcal{X}_H := \{v_H \in H^1_0(\Omega) \,|\, \forall T \in \mathcal{T}_H \quad v_H|_T \text{ is a polynomial of degree } \leq p\}. \tag{3.8}$$

To obtain conforming finite element approximations $u^\star \approx u_H \in \mathcal{X}_H$ and $z^\star \approx z_H \in \mathcal{X}_H$, we consider the Galerkin discretizations of (3.6)–(3.7). First, we note that the Lax–Milgram lemma yields the existence and uniqueness of *exact* discrete solutions $u^\star_H, z^\star_H \in \mathcal{X}_H$, i.e., there holds that

$$a(u^\star_H, v_H) = F(v_H) \quad \text{and} \quad a(v_H, z^\star_H) = G(v_H) \quad \text{for all } v_H \in \mathcal{X}_H. \tag{3.9}$$

In practice, the discrete systems (3.9) are rarely solved exactly (or up to machine precision). Instead, a suitable iterative solver is employed, which yields *approximate* discrete solutions $u^m_H, z^n_H \in \mathcal{X}_H$. We suppose that this iterative solver is contractive, i.e., for all $m, n \in \mathbb{N}$, it holds that

$$\|\!| u^\star_H - u^m_H |\!\| \leq q_{\mathrm{ctr}} \|\!| u^\star_H - u^{m-1}_H |\!\| \quad \text{and} \quad \|\!| z^\star_H - z^n_H |\!\| \leq q_{\mathrm{ctr}} \|\!| z^\star_H - z^{n-1}_H |\!\|, \tag{3.10}$$

where $0 < q_{\mathrm{ctr}} < 1$ is a generic constant and, in particular, independent of $\mathcal{X}_H$. Assumption (3.10) is satisfied, e.g., for an optimally preconditioned conjugate gradient (PCG) method (see [CNX12]) or geometric multigrid solvers (see [WZ17]); see also the discussion in [GHPS21].

3.2.3 Discrete goal quantity

To approximate $G(u^\star)$, we proceed as in [GS02]: For any $u_H, z_H \in \mathcal{X}_H$, it holds that

$$\begin{aligned} G(u^\star) - G(u_H) = G(u^\star - u_H) &\overset{(3.7)}{=} a(u^\star - u_H, z^\star) = a(u^\star - u_H, z^\star - z_H) + a(u^\star - u_H, z_H) \\ &\overset{(3.6)}{=} a(u^\star - u_H, z^\star - z_H) + \big[F(z_H) - a(u_H, z_H)\big]. \end{aligned}$$

Defining the *discrete quantity of interest*

$$G_H(u_H, z_H) := G(u_H) + \big[F(z_H) - a(u_H, z_H)\big], \tag{3.11}$$

the goal error can be controlled by means of the Cauchy–Schwarz inequality

$$\big|G(u^\star) - G_H(u_H, z_H)\big| \le \big|a(u^\star - u_H, z^\star - z_H)\big| \le |||u^\star - u_H||| \, |||z^\star - z_H|||. \tag{3.12}$$

We note that the additional term in (3.11) is the residual of the discrete primal problem (3.9) evaluated at an arbitrary function $z_H \in \mathcal{X}_H$ and hence $G(u_H^\star) = G_H(u_H^\star, z_H)$.

In the following, we design an adaptive algorithm that provides a computable upper bound to (3.12) which tends to zero at optimal algebraic rate with respect to the number of elements $\#\mathcal{T}_H$ as well as with respect to the total computational cost.

3.2.4 Mesh refinement

Let $\mathcal{T}_0$ be a given conforming triangulation of Ω. We suppose that the mesh-refinement is a deterministic and fixed strategy, e.g., newest vertex bisection [Ste08]. For each conforming triangulation $\mathcal{T}_H$ and marked elements $\mathcal{M}_H \subseteq \mathcal{T}_H$, let $\mathcal{T}_h := \texttt{refine}(\mathcal{T}_H, \mathcal{M}_H)$ be the coarsest conforming triangulation, where all $T \in \mathcal{M}_H$ have been refined, i.e., $\mathcal{M}_H \subseteq \mathcal{T}_H \backslash \mathcal{T}_h$. We write $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, if $\mathcal{T}_h$ results from $\mathcal{T}_H$ by finitely many steps of refinement. To abbreviate notation, let $\mathbb{T} := \mathbb{T}(\mathcal{T}_0)$. We note that the order on $\mathbb{T}$ is respected by the finite element spaces, i.e., $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$ implies that $\mathcal{X}_H \subseteq \mathcal{X}_h$.

We further suppose that each refined element has at least two sons, i.e.,

$$\#(\mathcal{T}_H \backslash \mathcal{T}_h) + \#\mathcal{T}_H \le \#\mathcal{T}_h \quad \text{for all } \mathcal{T}_H \in \mathbb{T} \text{ and all } \mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H), \tag{3.13}$$

and that the refinement rule satisfies the mesh-closure estimate

$$\#\mathcal{T}_\ell - \#\mathcal{T}_0 \le C_{\mathrm{cls}} \sum_{j=0}^{\ell-1} \#\mathcal{M}_j \quad \text{for all } \ell \in \mathbb{N}, \tag{3.14}$$

where $C_{\mathrm{cls}} > 0$ depends only on $\mathcal{T}_0$. For newest vertex bisection, this has been proved under an additional admissibility assumption on $\mathcal{T}_0$ in [BDD04; Ste08] and for 2D even without any additional assumption in [KPP13]. Finally, we suppose that the overlay estimate holds, i.e., for all triangulations $\mathcal{T}_H, \mathcal{T}_h \in \mathbb{T}$, there exists a common refinement $\mathcal{T}_H \oplus \mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H) \cap \mathbb{T}(\mathcal{T}_h)$ which satisfies that

$$\#(\mathcal{T}_H \oplus \mathcal{T}_h) \le \#\mathcal{T}_H + \#\mathcal{T}_h - \#\mathcal{T}_0, \tag{3.15}$$

which has been proved in [CKNS08; Ste07] for newest vertex bisection.

3.2.5 Estimator properties

For $\mathcal{T}_H \in \mathbb{T}$ and $v_H \in \mathcal{X}_H$, let

$$\eta_H(T, v_H) \ge 0 \quad \text{and} \quad \zeta_H(T, v_H) \ge 0 \quad \text{for all } T \in \mathcal{T}_H$$

be given refinement indicators. For $\mu_H \in \{\eta_H, \zeta_H\}$, we use the usual convention that

$$\mu_H(v_H) := \mu_H(\mathcal{T}_H, v_H), \quad \text{where} \quad \mu_H(\mathcal{U}_H, v_H) = \Big(\sum_{T \in \mathcal{U}_H} \mu_H(T, v_H)^2 \Big)^{1/2} \tag{3.16}$$

for all $v_H \in \mathcal{X}_H$ and all $\mathcal{U}_H \subseteq \mathcal{T}_H$.

We suppose that the estimators η_H and ζ_H satisfy the so-called *axioms of adaptivity* (which are designed for, but not restricted to, weighted-residual error estimators) from [CFPP14]: There exist constants $C_{\mathrm{stab}}, C_{\mathrm{rel}}, C_{\mathrm{drel}} > 0$ and $0 < q_{\mathrm{red}} < 1$ such that for all $\mathcal{T}_H \in \mathbb{T}(\mathcal{T}_0)$ and all $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, the following assumptions are satisfied:

(A1) Stability: For all $v_h \in \mathcal{X}_h$, $v_H \in \mathcal{X}_H$, and $\mathcal{U}_H \subseteq \mathcal{T}_h \cap \mathcal{T}_H$, it holds that

$$\big|\eta_h(\mathcal{U}_H, v_h) - \eta_H(\mathcal{U}_H, v_H)\big| + \big|\zeta_h(\mathcal{U}_H, v_h) - \zeta_H(\mathcal{U}_H, v_H)\big| \le C_{\mathrm{stab}} \,|\!|\!| v_h - v_H |\!|\!|.$$

(A2) Reduction: For all $v_H \in \mathcal{X}_H$, it holds that

$$\eta_h(\mathcal{T}_h \backslash \mathcal{T}_H, v_H) \le q_{\mathrm{red}}\, \eta_H(\mathcal{T}_H \backslash \mathcal{T}_h, v_H) \quad \text{and} \quad \zeta_h(\mathcal{T}_h \backslash \mathcal{T}_H, v_H) \le q_{\mathrm{red}}\, \zeta_H(\mathcal{T}_H \backslash \mathcal{T}_h, v_H).$$

(A3) Reliability: The Galerkin solutions $u_H^\star, z_H^\star \in \mathcal{X}_H$ to (3.9) satisfy that

$$|\!|\!| u^\star - u_H^\star |\!|\!| \le C_{\mathrm{rel}}\, \eta_H(u_H^\star) \quad \text{and} \quad |\!|\!| z^\star - z_H^\star |\!|\!| \le C_{\mathrm{rel}}\, \zeta_H(z_H^\star).$$

(A4) Discrete reliability: The Galerkin solutions $u_H^\star, z_H^\star \in \mathcal{X}_H$ and $u_h^\star, z_h^\star \in \mathcal{X}_h$ to (3.9) satisfy that

$$|\!|\!| u_h^\star - u_H^\star |\!|\!| \le C_{\mathrm{drel}}\, \eta_H(\mathcal{T}_H \backslash \mathcal{T}_h, u_H^\star) \quad \text{and} \quad |\!|\!| z_h^\star - z_H^\star |\!|\!| \le C_{\mathrm{drel}}\, \zeta_H(\mathcal{T}_H \backslash \mathcal{T}_h, z_H^\star).$$

By assumptions (A1) and (A3), we can estimate for every discrete function $w_H \in \mathcal{X}_H$ the errors in the energy norm of the primal and the dual problem by

$$|\!|\!| u^\star - w_H |\!|\!| \le C \left[\eta_H(w_H) + |\!|\!| u_H^\star - w_H |\!|\!|\right] \quad \text{and} \quad |\!|\!| z^\star - w_H |\!|\!| \le C \left[\zeta_H(w_H) + |\!|\!| z_H^\star - w_H |\!|\!|\right],$$

respectively, where $C = \max\{C_{\mathrm{rel}}, C_{\mathrm{rel}} C_{\mathrm{stab}} + 1\} > 0$. Together with (3.12), we then obtain that the goal error for approximations $u_H^m \approx u_H^\star$ and $z_H^n \approx z_H^\star$ in $\mathcal{X}_H$ is bounded by

$$\big|G(u^\star) - G_H(u_H^m, z_H^n)\big| \le C^2 \left[\, \eta_H(u_H^m) + |\!|\!| u_H^\star - u_H^m |\!|\!| \,\right] \left[\, \zeta_H(z_H^n) + |\!|\!| z_H^\star - z_H^n |\!|\!| \,\right]. \tag{3.17}$$

In the following sections, we provide building blocks for our adaptive algorithm that allow to control the arising estimators (by a suitable marking strategy) as well as the arising norms in the upper bound of (3.17) (by an appropriate stopping criterion for the iterative solver).

3.2.6 Marking strategy

We suppose that the refinement indicators $\eta_H(T, u_H^m)$ and $\zeta_H(T, z_H^n)$ for some $m, n \in \mathbb{N}$ are used to mark a subset $\mathcal{M}_H \subseteq \mathcal{T}_H$ of elements for refinement, which, for fixed marking parameter $0 < \theta \le 1$, satisfies that

$$2\theta \eta_H(u_H^m)^2 \zeta_H(z_H^n)^2 \le \eta_H(\mathcal{M}_H, u_H^m)^2 \zeta_H(z_H^n)^2 + \zeta_H(\mathcal{M}_H, z_H^n)^2 \eta_H(u_H^m)^2. \tag{3.18}$$

Remark 3.2. *Given $0 < \vartheta \le 1$, possible choices of marking strategies satisfying assumption (3.18) are the following:*

(a) *The strategy proposed in [BET11] defines the weighted estimator*

$$\rho_H(T, u_H^m, z_H^n)^2 := \eta_H(T, u_H^m)^2 \zeta_H(z_H^n)^2 + \eta_H(u_H^m)^2 \zeta_H(T, z_H^n)^2$$

and then determines a set $\mathcal{M}_H \subseteq \mathcal{T}_H$ *such that*

$$\vartheta\, \rho_H(u_H^m, z_H^n) \le \rho_H(\mathcal{M}_H, u_H^m, z_H^n) \tag{3.19}$$

which is the Dörfler marking criterion introduced in [Dör96] and well-known in the context of AFEM analysis; see, e.g., [CFPP14]. This strategy satisfies (3.18) *with* $\theta = \vartheta^2$.

(b) *The strategy proposed in [MS09] determines sets* $\overline{\mathcal{M}}_H^u, \overline{\mathcal{M}}_H^z \subseteq \mathcal{T}_H$ *such that*

$$\vartheta\, \eta_H(u_H^m) \le \eta_\ell(\overline{\mathcal{M}}_H^u, u_H^m) \quad \text{and} \quad \vartheta\, \zeta_H(z_H^n) \le \zeta_H(\overline{\mathcal{M}}_H^z, z_H^n) \tag{3.20}$$

and then chooses $\mathcal{M}_H := \arg\min\{ \#\overline{\mathcal{M}}_H^u, \#\overline{\mathcal{M}}_H^z \}$. *This strategy satisfies* (3.18) *with* $\theta = \vartheta^2/2$.

(c) *A more aggressive variant of* (b) *was proposed in [FPZ16]: Let* $\overline{\mathcal{M}}_H^u$ *and* $\overline{\mathcal{M}}_H^z$ *as above. Then, choose* $\mathcal{M}_H^u \subseteq \overline{\mathcal{M}}_H^u$ *and* $\mathcal{M}_H^z \subseteq \overline{\mathcal{M}}_H^z$ *with* $\#\mathcal{M}_H^u = \#\mathcal{M}_H^z = \min\{ \#\overline{\mathcal{M}}_H^u, \#\overline{\mathcal{M}}_H^z \}$. *Finally, define* $\mathcal{M}_H := \mathcal{M}_H^u \cup \mathcal{M}_H^z$. *Again, this strategy satisfies* (3.18) *with* $\theta = \vartheta^2/2$.

Note that our main results of Theorem 3.5 and 3.7 below hold true for all presented marking criteria (a)–(c). *For our numerical experiments, we focus on criterion* (a)*, which empirically tends to achieve slightly better performance in practice.*

3.2.7 Adaptive algorithm

Any adaptive algorithm strives to drive down the bound in (3.17). However, the errors of the iterative solver, $\|\!| u_H^\star - u_H^m |\!\|$ and $\|\!| z_H^\star - z_H^n |\!\|$, cannot be computed in general since the exact discrete solutions $u_H^\star, z_H^\star \in \mathcal{X}_H$ to (3.9) are unknown and will not be computed. Thus, we note that (3.10) and the triangle inequality prove that

$$(1 - q_{\mathrm{ctr}})\, \|\!| u_H^\star - u_H^{m-1} |\!\| \le \|\!| u_H^m - u_H^{m-1} |\!\| \le (1 + q_{\mathrm{ctr}})\, \|\!| u_H^\star - u_H^{m-1} |\!\| \tag{3.21a}$$

as well as

$$(1 - q_{\mathrm{ctr}})\, \|\!| z_H^\star - z_H^{n-1} |\!\| \le \|\!| z_H^n - z_H^{n-1} |\!\| \le (1 + q_{\mathrm{ctr}})\, \|\!| z_H^\star - z_H^{n-1} |\!\|. \tag{3.21b}$$

With $C_{\mathrm{goal}} = \max\{C_{\mathrm{rel}}, C_{\mathrm{rel}} C_{\mathrm{stab}} + 1\} \big(1 + q_{\mathrm{ctr}}/(1 - q_{\mathrm{ctr}})\big)$, (3.17) leads to

$$\big| G(u^\star) - G_H(u_H^m, z_H^n) \big| \le C_{\mathrm{goal}}^2 \big[\eta_H(u_H^m) + \|\!| u_H^m - u_H^{m-1} |\!\| \big] \big[\zeta_H(z_H^n) + \|\!| z_H^n - z_H^{n-1} |\!\| \big], \tag{3.22}$$

which is a computable upper bound to the goal error if $m, n \ge 1$. Moreover, given some $\lambda_{\mathrm{ctr}} > 0$, this motivates to stop the iterative solvers as soon as

$$\|\!| u_H^m - u_H^{m-1} |\!\| \le \lambda_{\mathrm{ctr}}\, \eta_H(u_H^m) \quad \text{and} \quad \|\!| z_H^n - z_H^{n-1} |\!\| \le \lambda_{\mathrm{ctr}}\, \zeta_H(z_H^n)$$

to equibalance the contributions of the upper bound in (3.22). Overall, we thus consider the following adaptive algorithm.

Algorithm 3A

Let $u_0^0, z_0^0 \in \mathcal{X}_0$ be initial guesses. Let $0 < \theta \leq 1$ as well as $\lambda_{\mathrm{ctr}} > 0$ be arbitrary but fixed marking parameters. For all $\ell = 0, 1, 2, \dots$, perform the following steps (i)–(vi):

(i) Employ (at least one step of) the iterative solver to compute iterates $u_\ell^1, \dots, u_\ell^m$ and $z_\ell^1, \dots, z_\ell^n$ together with the corresponding refinement indicators $\eta_\ell(T, u_\ell^k)$ and $\zeta_\ell(T, z_\ell^k)$ for all $T \in \mathcal{T}_\ell$, until

$$\|\!| u_\ell^m - u_\ell^{m-1} |\!\| \leq \lambda_{\mathrm{ctr}}\, \eta_\ell(u_\ell^m) \quad \text{and} \quad \|\!| z_\ell^n - z_\ell^{n-1} |\!\| \leq \lambda_{\mathrm{ctr}}\, \zeta_\ell(z_\ell^n). \tag{3.23}$$

(ii) Define $\underline{m}(\ell) := m$ and $\underline{n}(\ell) := n$.

(iii) If $\eta_\ell(u_\ell^m) = 0$ or $\zeta_\ell(z_\ell^m) = 0$, then define $\underline{\ell} := \ell$ and terminate.

(iv) Otherwise, find a set $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ such that the marking criterion (3.18) is satisfied.

(v) Generate $\mathcal{T}_{\ell+1} := \texttt{refine}(\mathcal{T}_\ell, \mathcal{M}_\ell)$.

(vi) Define the initial guesses $u_{\ell+1}^0 := u_\ell^m$ and $z_{\ell+1}^0 := z_\ell^n$ for the iterative solver.

Remark 3.3. *Theorem 3.5 below proves (linear) convergence for any choice of the marking parameters $0 < \theta \leq 1$ and $\lambda_{\mathrm{ctr}} > 0$, and for any of the marking strategies from Remark 3.2. Theorem 3.7 below proves optimal convergence rates (with respect to the number of elements and the total computational cost) if both parameters are sufficiently small (see (3.33) for the precise condition) and if the set $\mathcal{M}_\ell$ is constructed by one of the strategies from Remark 3.2, where the respective sets have quasi-minimal cardinality.*

Remark 3.4. *Note that Algorithm 3A(i) requires to evaluate the error estimator after each solver step. Clearly, it would be favorable to replace $\eta_\ell(u_\ell^m)$ (resp. $\zeta_\ell(z_\ell^n)$) by $\eta_\ell(u_\ell^0)$ (resp. $\zeta_\ell(z_\ell^0)$) in (3.23). Arguing as in [DFGP19, Lemma 8], this allows to prove convergence of the adaptive strategy, but full linear convergence (Theorem 3.5 below) and optimal convergence rates (Theorem 3.7 below) are exptected to fail.*

For each adaptive level ℓ, Algorithm 3A performs at least one solver step to compute u_ℓ^m as well as one solver step to compute z_ℓ^n. By definition, $\underline{m}(\ell) \geq 1$ is the solver step, for which the discrete solution $u_\ell^{\underline{m}(\ell)}$ is accepted (to contribute to the set of marked elements $\mathcal{M}_\ell$). Analogously, $\underline{n}(\ell) \geq 1$ is the solver step, for which the discrete solution $z_\ell^{\underline{n}(\ell)}$ is accepted (to contribute to $\mathcal{M}_\ell$). If the iterative solver for either the primal or the dual problem fails to terminate for some level $\ell \in \mathbb{N}_0$, i.e., (3.23) cannot be achieved for finite m, or n, we define $\underline{m}(\ell) := \infty$, or $\underline{n}(\ell) := \infty$, respectively, and $\underline{\ell} := \ell$. With $\underline{k}(\ell) := \max\{\underline{m}(\ell), \underline{n}(\ell)\}$, we define

$$\begin{aligned} u_\ell^k &:= u_\ell^{\underline{m}(\ell)} \quad \text{for all } k \in \mathbb{N} \text{ with } \underline{m}(\ell) < k \leq \underline{k}(\ell), \\ z_\ell^k &:= z_\ell^{\underline{n}(\ell)} \quad \text{for all } k \in \mathbb{N} \text{ with } \underline{n}(\ell) < k \leq \underline{k}(\ell). \end{aligned} \tag{3.24}$$

For ease of presentation, we omit the ℓ-dependence of the indices for final iterates $\underline{m}(\ell)$, $\underline{n}(\ell)$, and $\underline{k}(\ell)$ in the following if they appear as upper indices and write, e.g., $u_\ell^{\underline{m}} := u_\ell^{\underline{m}(\ell)}$ and $u_\ell^{\underline{m}-1} := u_\ell^{\underline{m}(\ell)-1}$.

If Algorithm 3A does not terminate in step (iii) for some $\ell \in \mathbb{N}$, then we define $\underline{\ell} := \infty$. To formulate the convergence of Algorithm 3A, we define the ordered set

$$\mathcal{Q} := \{(\ell, k) \in \mathbb{N}_0^2 \,|\, \ell \le \underline{\ell} \text{ and } 1 \le k \le \underline{k}(\ell)\}, \quad \text{where} \quad |(\ell, k)| := k + \sum_{j=0}^{\ell-1} \underline{k}(j). \tag{3.25}$$

Note that $|(\ell, k)|$ is proportional to the overall number of solver steps to compute the estimator product $\eta_\ell(u_\ell^k)\zeta_\ell(z_\ell^k)$. Additionally, we sometimes require the notation

$$\mathcal{Q}_0 := \{(\ell, k) \in \mathbb{N}_0^2 \,|\, \ell \le \underline{\ell} \text{ and } 0 \le k \le \underline{k}(\ell)\} = \mathcal{Q} \cup \{(\ell, 0) \in \mathbb{N}_0^2 \,|\, \ell \le \underline{\ell}\}. \tag{3.26}$$

To estimate the work necessary to compute a pair $(u_\ell^k, z_\ell^k) \in \mathcal{X}_\ell \times \mathcal{X}_\ell$, we make the following assumptions which are usually satisfied in practice:

- The iterates u_ℓ^k and z_ℓ^k are computed in parallel and each step of the solver in Algorithm 3A(i) can be done in linear complexity $\mathcal{O}(\#\mathcal{T}_\ell)$;
- Computation of all indicators $\eta_\ell(T, u_\ell^k)$ and $\zeta_\ell(T, z_\ell^k)$ for $T \in \mathcal{T}_\ell$ requires $\mathcal{O}(\#\mathcal{T}_\ell)$ steps;
- The marking in Algorithm 3A(iv) can be performed at linear cost $\mathcal{O}(\#\mathcal{T}_\ell)$ (according to [Ste07] this can be done for the strategies outlined in Remark 3.2 with $\mathcal{M}_\ell$ having almost minimal cardinality; moreover, we refer to a recent own algorithm in [PP20] with linear cost even for $\mathcal{M}_\ell$ having minimal cardinality);
- We have linear cost $\mathcal{O}(\#\mathcal{T}_\ell)$ to generate the new mesh $\mathcal{T}_{\ell+1}$.

Since a step $(\ell, k) \in \mathcal{Q}$ of Algorithm 3A depends on the full history of preceding steps, the total work spent to compute $(u_\ell^k, z_\ell^k) \in \mathcal{X}_\ell \times \mathcal{X}_\ell$ is then of order

$$\texttt{work}(\ell, k) := \sum_{\substack{(\ell', k') \in \mathcal{Q} \\ |(\ell', k')| \le |(\ell, k)|}} \#\mathcal{T}_{\ell'} \quad \text{for all } (\ell, k) \in \mathcal{Q}. \tag{3.27}$$

Finally, we note that Algorithm 3A(vi) employs *nested iteration* to obtain the initial guesses $u_{\ell+1}^0, z_{\ell+1}^0$ of the solver from the final iterates u_ℓ^m, z_ℓ^n for the mesh $\mathcal{T}_\ell$. According to (3.22), this allows for *a posteriori* error control for all indices $(\ell, k) \in \mathcal{Q}_0 \setminus \{(0, 0)\}$ beyond the initial step.

3.3 Main results

3.3.1 Linear convergence with optimal rates

Our first main result states linear convergence of the quasi-error product

$$\Lambda_\ell^k := \big[\, |||u_\ell^\star - u_\ell^k||| + \eta_\ell(u_\ell^k) \,\big] \big[\, |||z_\ell^\star - z_\ell^k||| + \zeta_\ell(z_\ell^k) \,\big] \quad \text{for all } (\ell, k) \in \mathcal{Q}_0 \tag{3.28}$$

for every choice of the stopping parameter $\lambda_{\mathrm{ctr}} > 0$. Recall from (3.17) that the quasi-error product is an upper bound for the error $|G(u^\star) - G_\ell(u_\ell^k, z_\ell^k)|$. Moreover, if $k = \underline{k}(\ell)$, then (3.21) and (3.23) give that $\Lambda_\ell^k \simeq \eta_\ell(u_\ell^k)\zeta_\ell(z_\ell^k)$.

Theorem 3.5

Suppose (A1)–(A3). *Suppose that* $0 < \theta \le 1$ *and* $\lambda_{\mathrm{ctr}} > 0$. *Then, Algorithm 3A satisfies linear convergence in the sense of*

$$\Lambda_{\ell'}^{k'} \le C_{\mathrm{lin}} q_{\mathrm{lin}}^{|(\ell',k')|-|(\ell,k)|}\, \Lambda_\ell^k \quad \text{for all } (\ell,k),(\ell',k') \in \mathcal{Q} \cup \{(0,0)\} \text{ with } |(\ell',k')| \ge |(\ell,k)|. \tag{3.29}$$

The constants $C_{\mathrm{lin}} > 0$ *and* $0 < q_{\mathrm{lin}} < 1$ *depend only on* C_{stab}, q_{red}, C_{rel}, q_{ctr}, *and the (arbitrary) adaptivity parameters* $0 < \theta \le 1$ *and* $\lambda_{\mathrm{ctr}} > 0$.

Full linear convergence implies that convergence rates with respect to degrees of freedom and with respect to total computational cost are equivalent. From this point of view, full linear convergence indeed turns out to be the core argument for optimal complexity.

Corollary 3.6. *Recall the definition of the total computational cost* $\mathtt{work}(\ell,k)$ *from* (3.27). *Let* $r > 0$ *and* $C_r := \sup_{(\ell,k)\in\mathcal{Q}} (\#\mathcal{T}_\ell - \#\mathcal{T}_0 + 1)^r \Lambda_\ell^k \in [0,\infty]$. *Then, under the assumptions of Theorem 3.5, it holds that*

$$C_r \le \sup_{(\ell,k)\in\mathcal{Q}} (\#\mathcal{T}_\ell)^r\, \Lambda_\ell^k \le \sup_{(\ell,k)\in\mathcal{Q}} \mathtt{work}(\ell,k)^r\, \Lambda_\ell^k \le C_{\mathrm{rate}}\, C_r, \tag{3.30}$$

where the constant $C_{\mathrm{rate}} > 0$ *depends only on* r, $\#\mathcal{T}_0$, *and on the constants* $q_{\mathrm{lin}}, C_{\mathrm{lin}}$ *from Theorem 3.5.*

Proof. The first two estimates in (3.30) are obvious. It remains to prove the last estimate in (3.30). To this end, note that it follows from the definition of C_r that

$$\#\mathcal{T}_\ell - \#\mathcal{T}_0 + 1 \le \big(\Lambda_\ell^k\big)^{-1/r}\, C_r^{1/r} \quad \text{for all } (\ell,k) \in \mathcal{Q}.$$

Moreover, elementary algebra yields that

$$\#\mathcal{T}_{\ell'} \le \#\mathcal{T}_0 (\#\mathcal{T}_{\ell'} - \#\mathcal{T}_0 + 1) \quad \text{for all } (\ell',0) \in \mathcal{Q}_0.$$

For $(\ell,k) \in \mathcal{Q}$, Theorem 3.5 and the geometric series thus show that

$$\begin{aligned}
\mathtt{work}(\ell,k) &\overset{(3.27)}{=} \sum_{\substack{(\ell',k')\in\mathcal{Q}\\ |(\ell',k')|\le|(\ell,k)|}} \#\mathcal{T}_{\ell'} \le \#\mathcal{T}_0 \sum_{\substack{(\ell',k')\in\mathcal{Q}\\ |(\ell',k')|\le|(\ell,k)|}} (\#\mathcal{T}_{\ell'} - \#\mathcal{T}_0 + 1) \\
&\le \#\mathcal{T}_0 C_r^{1/r} \sum_{\substack{(\ell',k')\in\mathcal{Q}\\ |(\ell',k')|\le|(\ell,k)|}} \big(\Lambda_{\ell'}^{k'}\big)^{-1/r} \le \#\mathcal{T}_0 C_r^{1/r} C_{\mathrm{lin}}^{1/r} \frac{1}{1 - q_{\mathrm{lin}}^{1/r}} \big(\Lambda_\ell^k\big)^{-1/r}.
\end{aligned}$$

With $C_{\mathrm{rate}} := (\#\mathcal{T}_0)^r C_{\mathrm{lin}}\, 1/(1 - q_{\mathrm{lin}}^{1/r})^r$, this gives that

$$\mathtt{work}(\ell,k)^r \Lambda_\ell^k \le C_{\mathrm{rate}} C_r \quad \text{for all } (\ell,k) \in \mathcal{Q}.$$

This shows the final inequality in (3.30) and thus concludes the proof. □

If θ and λ_{ctr} are small enough, we are able to show that linear convergence from Theorem 3.5 even guarantees optimal rates with respect to both the number of unknowns $\#\mathcal{T}_\ell$ and the total cost $\mathtt{work}(\ell, k)$. Given $N \in \mathbb{N}_0$, let $\mathbb{T}(N)$ be the set of all $\mathcal{T}_H \in \mathbb{T}$ with $\#\mathcal{T}_H - \#\mathcal{T}_0 \le N$. With

$$\|u^\star\|_{\mathbb{A}_r} := \sup_{N\in\mathbb{N}_0} (N+1)^r \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(N)} \eta_{\mathrm{opt}}(u^\star_{\mathrm{opt}}) \in [0,\infty] \tag{3.31a}$$

and

$$\|z^\star\|_{\mathbb{A}_r} := \sup_{N\in\mathbb{N}_0} (N+1)^r \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(N)} \zeta_{\mathrm{opt}}(z^\star_{\mathrm{opt}}) \in [0,\infty] \tag{3.31b}$$

for all $r > 0$, there holds the following result.

Theorem 3.7

Recall the definition of the total computational cost $\mathtt{work}(\ell, k)$ *from* (3.27). *Suppose the mesh properties* (3.13)–(3.15) *as well as the axioms* (A1)–(A4). *Define*

$$\theta_\star := \frac{1}{1 + C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2} \quad \text{and} \quad \lambda_\star := \frac{1-q_{\mathrm{ctr}}}{q_{\mathrm{ctr}} C_{\mathrm{stab}}}. \tag{3.32}$$

Let both adaptivity parameters $0 < \theta \le 1$ *and* $0 < \lambda_{\mathrm{ctr}} < \lambda_\star$ *be sufficiently small such that*

$$0 < \left(\frac{\sqrt{2\theta} + \lambda_{\mathrm{ctr}}/\lambda_\star}{1 - \lambda_{\mathrm{ctr}}/\lambda_\star}\right)^2 < \theta_\star. \tag{3.33}$$

Let $1 \le C_{\mathrm{mark}} < \infty$. *Suppose that the set of marked elements* $\mathcal{M}_\ell$ *in Algorithm 3A*(iv) *is constructed by one of the strategies from Remark 3.2*(a)–(c), *where the sets in* (3.19) *and* (3.20) *have up to the factor* C_{mark} *minimal cardinality. Let* $s, t > 0$ *with* $\|u^\star\|_{\mathbb{A}_s} + \|z^\star\|_{\mathbb{A}_t} < \infty$. *Then, there exists a constant* $C_{\mathrm{opt}} > 0$ *such that*

$$\sup_{(\ell,k)\in\mathcal{Q}} \mathtt{work}(\ell,k)^{s+t} \Lambda_\ell^k \le C_{\mathrm{opt}} \max\{\|u^\star\|_{\mathbb{A}_s} \|z^\star\|_{\mathbb{A}_t}, \Lambda_0^0\}. \tag{3.34}$$

The constant C_{opt} *depends only on* C_{cls}, C_{stab}, q_{red}, C_{rel}, C_{drel}, q_{ctr}, C_{mark}, θ, λ_{ctr}, $\#\mathcal{T}_0$, s, *and* t.

Remark 3.8. *The constraint* (3.33) *is enforced by our analysis of the marking strategy from Remark 3.2*(a), *while the marking strategies from Remark 3.2*(b)–(c) *allow to relax the condition to*

$$0 < \left(\frac{\sqrt{\theta} + \lambda_{\mathrm{ctr}}/\lambda_\star}{1 - \lambda_{\mathrm{ctr}}/\lambda_\star}\right)^2 < \theta_\star. \tag{3.35}$$

3.3.2 Alternative termination criteria for iterative solver

The above formulations of Algorithm 3A stops the iterative solver for u_ℓ^m and the iterative solver for z_ℓ^n independently of each other as soon as the respective termination criteria in (3.23) are satisfied. In this section, we briefly discuss two alternative termination criteria:

Stronger termination: The current proof of linear convergence (and of the subsequent proof of optimal convergence) does only exploit that u_ℓ^k and z_ℓ^k satisfy the stopping criterion and the previous

iterates do not (cf. Lemma 3.9(iii)). This can also be ensured by the following modification of Algorithm 3A(i):

(i) Employ the iterative solver to compute iterates $u_\ell^1, \dots, u_\ell^k$ and $z_\ell^1, \dots, z_\ell^k$ together with the corresponding refinement indicators $\eta_\ell(T, u_\ell^k)$ and $\zeta_\ell(T, z_\ell^k)$ for all $T \in \mathcal{T}_\ell$, until

$$\|\!| u_\ell^k - u_\ell^{k-1} |\!\| \le \lambda_{\mathrm{ctr}}\, \eta_\ell(u_\ell^k) \quad \text{and} \quad \|\!| z_\ell^k - z_\ell^{k-1} |\!\| \le \lambda_{\mathrm{ctr}}\, \zeta_\ell(z_\ell^k). \tag{3.36}$$

Note that this will lead to more solver steps, since now $k = \underline{k}(\ell)$ (if it exists) is the smallest index for which the stopping criterion holds simultaneously for both u_ℓ^k and z_ℓ^k.

Inspecting the proof of Lemma 3.9 below, we see that all results hold verbatim also for this stopping criterion. Thus, we conclude linear and optimal convergence (in the sense of Theorem 3.5 and Theorem 3.7) also in this case.

Natural termination: The following stopping criterion (which is somehow the most natural candidate) also leads to linear convergence: Let $\underline{m}(\ell), \underline{n}(\ell) \in \mathbb{N}$ be minimal with (3.23). If either of them do not exist, we set again $\underline{m}(\ell) = \infty$, or $\underline{n}(\ell) = \infty$, respectively. Define $\underline{k}(\ell) := \max\{\underline{m}(\ell), \underline{n}(\ell)\}$. Then, employ the iterative solver $\underline{k}(\ell)$ times for both the primal and the dual problem, i.e., the solver provides iterates u_ℓ^k and z_ℓ^k until both stopping criteria in (3.23) have been satisfied once (which avoids the artificial definition (3.24)). For instance, if $\underline{m}(\ell) < \underline{n}(\ell) = \underline{k}(\ell) < \infty$, we continue to iterate for the primal problem until $u_\ell^{\underline{k}}$ is obtained (or never stop the iteration if $\underline{n}(\ell) = \underline{k}(\ell) = \infty$). If $\lambda_{\mathrm{ctr}} > 0$ is sufficiently small such that $1 - \frac{q_{\mathrm{ctr}}}{1-q_{\mathrm{ctr}}}\, C_{\mathrm{stab}}\, (1 + q_{\mathrm{ctr}})\lambda_{\mathrm{ctr}} > 0$, then we can define

$$\lambda_{\mathrm{ctr}} \le \lambda'_{\mathrm{ctr}} := \max\left\{1, \frac{(1 + q_{\mathrm{ctr}})q_{\mathrm{ctr}}}{(1 - q_{\mathrm{ctr}})\big(1 - \frac{q_{\mathrm{ctr}}}{1-q_{\mathrm{ctr}}}\, C_{\mathrm{stab}}\, (1 + q_{\mathrm{ctr}})\lambda_{\mathrm{ctr}}\big)}\right\} \lambda_{\mathrm{ctr}} < \infty,$$

and we can guarantee the stopping condition (3.23) with the larger constant λ'_{ctr}, i.e.,

$$\|\!| u_\ell^{\underline{k}} - u_\ell^{\underline{k}-1} |\!\| \le \lambda'_{\mathrm{ctr}}\, \eta_\ell(u_\ell^{\underline{k}}) \quad \text{and} \quad \|\!| z_\ell^{\underline{k}} - z_\ell^{\underline{k}-1} |\!\| \le \lambda'_{\mathrm{ctr}}\, \zeta_\ell(z_\ell^{\underline{k}}); \tag{3.37}$$

see the proof below. Again, we notice that then the assumptions of Lemma 3.9 below are met. Hence, we conclude linear convergence (in the sense of Theorem 3.5) also for this stopping criterion. Moreover, optimal rates in the sense of Theorem 3.7 hold if λ_{ctr} in (3.33) is replaced by λ'_{ctr}.

Proof of (3.37). Without loss of generality, let us assume that $\underline{m}(\ell) < \underline{k}(\ell) = \underline{n}(\ell) < \infty$. First, we have that

$$\|\!| u_\ell^{\underline{k}} - u_\ell^{\underline{m}} |\!\| \le \|\!| u_\ell^\star - u_\ell^{\underline{k}} |\!\| + \|\!| u_\ell^\star - u_\ell^{\underline{m}} |\!\| \le (1 + q_{\mathrm{ctr}}^{\underline{k}(\ell)-\underline{m}(\ell)}) \|\!| u_\ell^\star - u_\ell^{\underline{m}} |\!\|.$$

Then, using the fact that $u_\ell^{\underline{m}}$ satisfies the stopping criterion in (3.23) and stability (A1), we get that

$$\begin{aligned} \|\!| u_\ell^\star - u_\ell^{\underline{m}} |\!\| &\overset{(3.21)}{\le} \frac{q_{\mathrm{ctr}}}{1 - q_{\mathrm{ctr}}} \|\!| u_\ell^{\underline{m}} - u_\ell^{\underline{m}-1} |\!\| \overset{(3.23)}{\le} \frac{q_{\mathrm{ctr}}\lambda_{\mathrm{ctr}}}{1 - q_{\mathrm{ctr}}} \eta_\ell(u_\ell^{\underline{m}}) \overset{(A1)}{\le} \frac{q_{\mathrm{ctr}}\lambda_{\mathrm{ctr}}}{1 - q_{\mathrm{ctr}}} \Big(\eta_\ell(u_\ell^{\underline{k}}) + C_{\mathrm{stab}} \|\!| u_\ell^{\underline{k}} - u_\ell^{\underline{m}} |\!\|\Big) \\ &\le \frac{q_{\mathrm{ctr}}\lambda_{\mathrm{ctr}}}{1 - q_{\mathrm{ctr}}} \Big(\eta_\ell(u_\ell^{\underline{k}}) + C_{\mathrm{stab}}(1 + q_{\mathrm{ctr}}^{\underline{k}(\ell)-\underline{m}(\ell)}) \|\!| u_\ell^\star - u_\ell^{\underline{m}} |\!\|\Big). \end{aligned}$$

For $\lambda_{\mathrm{ctr}} < (1 - q_{\mathrm{ctr}})/[C_{\mathrm{stab}} q_{\mathrm{ctr}}(1 + q_{\mathrm{ctr}}^{\underline{k}(\ell)-\underline{m}(\ell)})]$ we can absorb the last term to obtain

$$\|\!| u_\ell^\star - u_\ell^{\underline{m}} |\!\| \le \frac{q_{\mathrm{ctr}}}{1 - q_{\mathrm{ctr}}} \Big(1 - \frac{C_{\mathrm{stab}} q_{\mathrm{ctr}}}{1 - q_{\mathrm{ctr}}} (1 + q_{\mathrm{ctr}}^{\underline{k}(\ell)-\underline{m}(\ell)})\lambda_{\mathrm{ctr}}\Big)^{-1} \lambda_{\mathrm{ctr}} \eta_\ell(u_\ell^{\underline{k}}).$$

Finally, we observe that

$$|||u_\ell^k - u_\ell^{k-1}||| \le (1+q_{\mathrm{ctr}})|||u_\ell^\star - u_\ell^{k-1}||| \le (1+q_{\mathrm{ctr}})q_{\mathrm{ctr}}^{k-\underline{m}-1}|||u_\ell^\star - u_\ell^{\underline{m}}|||.$$

Combining the last two estimates we obtain that

$$|||u_\ell^k - u_\ell^{k-1}||| \le \frac{(1+q_{\mathrm{ctr}})q_{\mathrm{ctr}}^{\underline{k}(\ell)-\underline{m}(\ell)}}{(1-q_{\mathrm{ctr}})\big(1-\frac{q_{\mathrm{ctr}}}{1-q_{\mathrm{ctr}}}C_{\mathrm{stab}}(1+q_{\mathrm{ctr}}^{\underline{k}(\ell)-\underline{m}(\ell)})\lambda_{\mathrm{ctr}}\big)}\,\lambda_{\mathrm{ctr}}\,\eta_\ell(u_\ell^{\underline{k}}).$$

Hence, (3.37) follows with $q_{\mathrm{ctr}}^{\underline{k}(\ell)-\underline{m}(\ell)} \le q_{\mathrm{ctr}}$ and $|||z_\ell^{\underline{k}} - z_\ell^{k-1}||| \le \lambda_{\mathrm{ctr}}\,\zeta_\ell(z_\ell^{\underline{k}}) \le \lambda'_{\mathrm{ctr}}\,\zeta_\ell(z_\ell^{\underline{k}})$. □

3.4 Numerical examples

In this section, we consider two numerical examples which solve the equation

$$\begin{aligned} -\Delta u^\star &= f && \text{in } \Omega,\\ u^\star &= 0 && \text{on } \Gamma_D,\\ \nabla u^\star \cdot \boldsymbol{n} &= \phi && \text{on } \Gamma_N, \end{aligned} \tag{3.38}$$

where $\phi \in L^2(\Gamma_N)$ and $\boldsymbol{n}$ is the element-wise outwards facing unit normal vector. We refer the reader to Remark 3.1 for a comment on the applicability of our results to this model problem. We further suppose that the goal functional is a slight variant of the one proposed in [MS09], i.e.,

$$G(v) = -\int_\omega \nabla v \cdot \boldsymbol{g}\,\mathrm{d}x \qquad \text{for } v \in H^1_D(\Omega), \tag{3.39}$$

with a subset $\omega \subseteq \Omega$ and a fixed direction $\boldsymbol{g}(x) = \boldsymbol{g}_0 \in \mathbb{R}^2$. Moreover, for error estimation, we employ standard residual error estimators, which in our case, for all $(\ell,k) \in \mathcal{Q}$ and all $T \in \mathcal{T}_\ell$, read

$$\begin{aligned} \eta_\ell(T,u_\ell^k)^2 &:= h_T^2\|\Delta u_\ell^k + f\|^2_{L^2(T)} + h_T\|[\![\nabla u_\ell^k \cdot \boldsymbol{n}]\!]\|^2_{L^2(\partial T\cap\Omega)} + h_T\|\nabla u_\ell^k \cdot \boldsymbol{n} - \phi\|^2_{L^2(\partial T\cap\Gamma_N)},\\ \zeta_\ell(T,z_\ell^k)^2 &:= h_T^2\|\operatorname{div}(\nabla z_\ell^k + \boldsymbol{g})\|^2_{L^2(T)} + h_T\|[\![(\nabla z_\ell^k + \boldsymbol{g})\cdot\boldsymbol{n}]\!]\|^2_{L^2(\partial T\cap\Omega)}, \end{aligned}$$

where $h_T = |T|^{1/2}$ is the local mesh-width and $[\![\cdot]\!]$ denotes the jump across interior edges. It is well-known [CFPP14; FPZ16] that η_ℓ and ζ_ℓ satisfy the assumptions (A1)–(A4). The examples are chosen to showcase the performance of the proposed GOAFEM algorithm for different types of singularities.

Throughout this section, we solve (3.38) as well as the corresponding dual problem numerically using Algorithm 3A, where we make the following choices:

- We solve the problems on the lowest order finite element space, i.e., with polynomial degree $p = 1$.
- As initial values, we use $u_0^0 = z_0^0 = 0$.
- To solve the arising linear systems, we use a preconditioned conjugate gradient (PCG) method with an optimal additive Schwarz preconditioner. We refer to [CNX12; Sch21] for details and, in particular, the proof that this iterative solver satisfies (3.10).

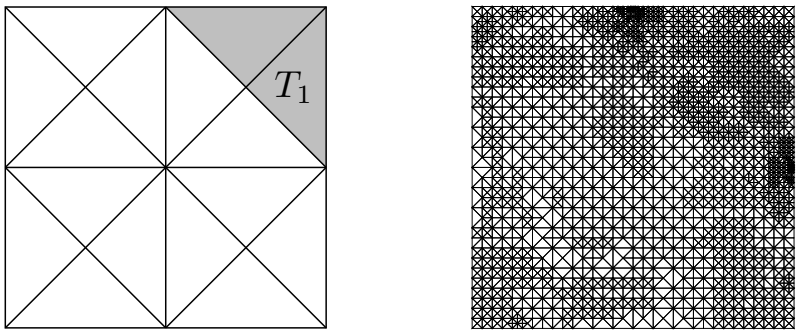


Figure 3.1: Left: Initial mesh $\mathcal{T}_0$. The shaded area is the set T_1 from Section (3.4.1). Right: Mesh after 14 iterations of Algorithm 3A with $\#\mathcal{T}_{14} = 4157$.

- We use the marking criterion from Remark 3.2(a) and choose $\mathcal{M}_\ell$ such that it has minimal cardinality.
- Unless mentioned otherwise, we use $\vartheta = 0.5$ and $\lambda_{\mathrm{ctr}} = 10^{-5}$.

3.4.1 Singularity in goal functional only

In our first example, the primal problem is (3.38) with $f = 2x_1(1-x_1) + 2x_2(1-x_2)$ on the unit square $\Omega = (0,1)^2$, and $\Gamma_D = \partial\Omega$ (and thus, $\Gamma_N = \emptyset$). For this problem, the exact solution reads

$$u^\star(x) = x_1 x_2 (1-x_1)(1-x_2).$$

The goal functional is (3.39) with $\omega = T_1 := \{x \in \Omega \,|\, x_1 + x_2 \geq 3/2\}$ and $\boldsymbol{g}_0 = (-1, 0)$. The exact goal value can be computed analytically to be

$$G(u^\star) = \int_{T_1} \frac{\partial u^\star}{\partial x_1} \,\mathrm{d}x = 11/960.$$

The initial mesh $\mathcal{T}_0$ as well as a visualization of the set T_1 can be seen in Figure 3.1.

For this setting, we compare our iterative solver to a conjugate gradient method without preconditioner in Figure 3.2, where we plot the computable upper bound from (3.22),

$$\Xi_\ell^k := \big[\eta_\ell(u_\ell^k) + |||u_\ell^k - u_\ell^{k-1}|||\big]\big[\zeta_\ell(z_\ell^k) + |||z_\ell^k - z_\ell^{k-1}|||\big] \quad \text{for all } (\ell, k) \in \mathcal{Q},$$

over $\texttt{work}(\ell, k)$ for all iterates $(\ell, k) \in \mathcal{Q}$ and the estimator product for the final iterates $\eta_\ell(u_\ell^k)\zeta_\ell(z_\ell^k)$ over $\#\mathcal{T}_\ell$. We stress that, for $(\ell, k) \in \mathcal{Q}$, the computable upper bound Ξ_ℓ^k and the quasi-error product Λ_ℓ^k from (3.28) are related by $\Lambda_\ell^k \lesssim \Xi_\ell^k \lesssim \Lambda_\ell^{k-1}$ so that linear convergence (3.29) with optimal rates (3.34) of Λ_ℓ^k also yields linear convergence with optimal rates of Ξ_ℓ^k. Since in our experiments $\lambda_{\mathrm{ctr}} = 10^{-5}$ is small, it is plausible to assume that the final estimates on every level approximate the exact solutions sufficiently well in the sense of estimator products, i.e., $\eta_\ell(u_\ell^k)\zeta_\ell(z_\ell^k) \approx \eta_\ell(u_\ell^\star)\zeta_\ell(z_\ell^\star)$ (cf. Lemma 3.12 below) for which [FPZ16] proves optimal convergence rates with respect to $\#\mathcal{T}_\ell$. Indeed, we see optimal rates for $\eta_\ell(u_\ell^k)\zeta_\ell(z_\ell^k)$ with respect to $\#\mathcal{T}_\ell$ for both solvers in Figure 3.2. However, the non-preconditioned CG method fails to satisfy uniform contraction (3.10) and thus

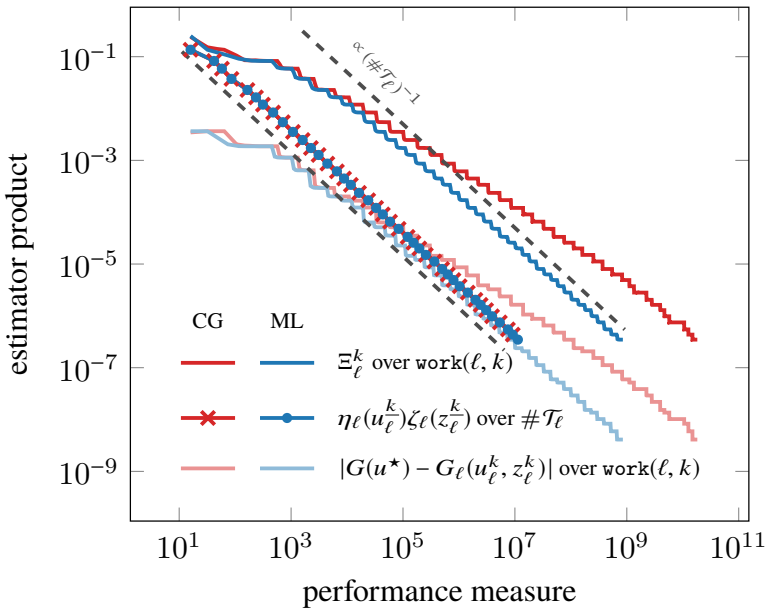


Figure 3.2: Comparison between iterative solvers for the problem from Section 3.4.1. A conjugate gradient method without preconditioner (CG) leads to optimal rates with respect to $\#\mathcal{T}_\ell$ for the final iterates where $k = \underline{k}(\ell)$, but not with respect to $\mathtt{work}(\ell, k)$ for every $(\ell, k) \in \mathcal{Q}$. Our choice of the iterative solver (ML) achieves optimal rates with respect to both measures.

Theorem 3.7 cannot be applied. In fact, Figure 3.2 shows that this method fails to drive down Ξ_ℓ^k with optimal rates with respect to $\mathtt{work}(\ell, k)$ (cf. (3.27)), as opposed to the optimally preconditioned PCG method.

Furthermore, we plot in Figure 3.3 different error measures over $\mathtt{work}(\ell, k)$ for every iterate $(\ell, k) \in \mathcal{Q}$. This shows that the corrector term

$$a(u_\ell^k, z_\ell^k) - F(z_\ell^k) \tag{3.40}$$

(which is the residual of u_ℓ^k evaluated at the dual solution z_ℓ^k) in the definition of the discrete goal functional (3.11) is indeed necessary. We see that throughout the iteration, the goal value $G(u_\ell^k)$ highly oscillates and, for large values of λ_{ctr}, even shows a different rate than the Ξ_ℓ^k over $\mathtt{work}(\ell, k)$. In general, we thus cannot expect the quantity Ξ_ℓ^k to bound the uncorrected goal-error $|G(u^\star) - G(u_\ell^k)|$.

For the discrete goal, the corrector term compensates the oscillations of the goal functional, such that their sum decreases with the same rate as Ξ_ℓ^k, as predicted by (3.22). Smaller values of λ_{ctr} imply that on every level ℓ the approximate solutions u_ℓ^k, z_ℓ^k are computed more accurately, such that the corrector term becomes smaller and the effect on the rate of the goal value becomes negligible.

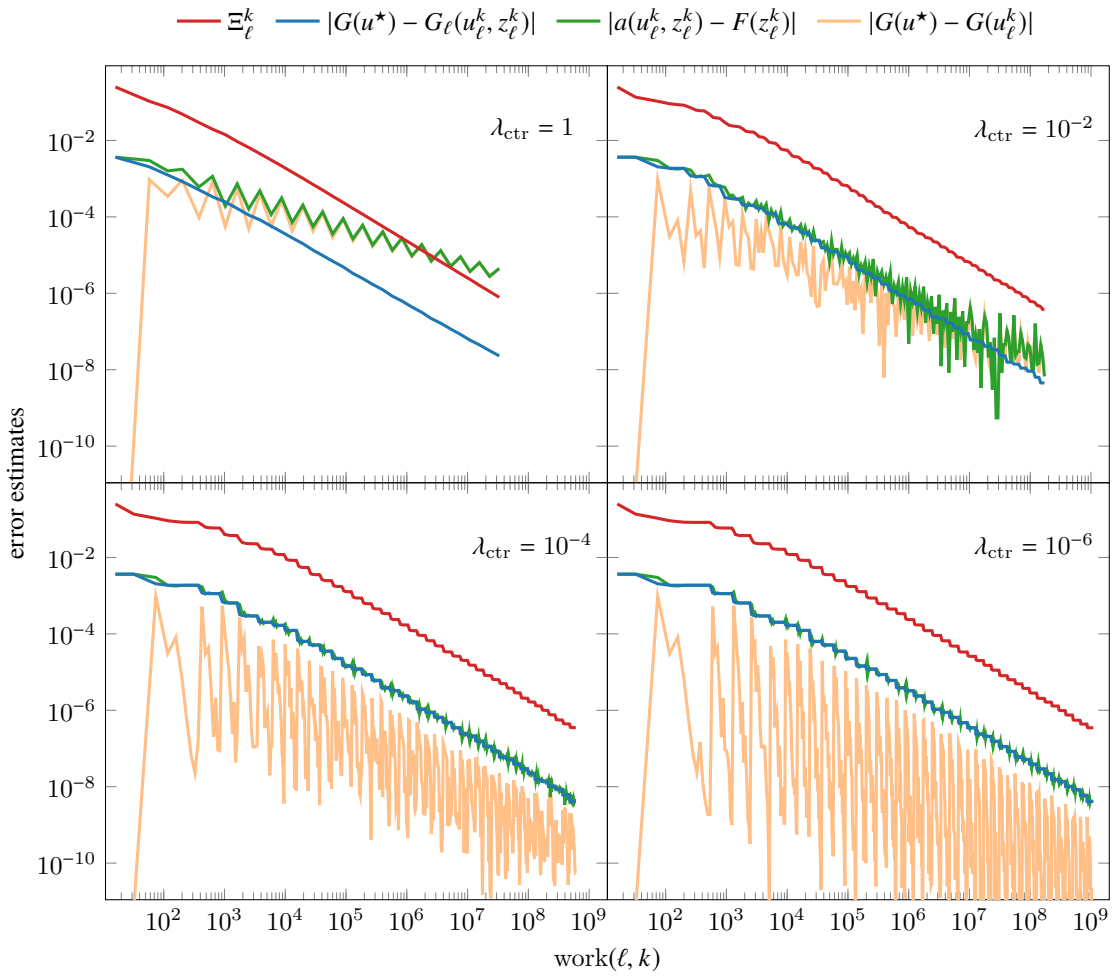


Figure 3.3: Comparison between Ξ_ℓ^k, discrete goal $G_\ell(u_\ell^k, z_\ell^k)$, primal residual evaluated at the dual solution z_ℓ^k, and direct evaluation of goal functional $G(u_\ell^k)$ for every iterate $(\ell, k) \in \mathcal{Q}$ and different values of $\lambda_{\mathrm{ctr}} \in \{1, 10^{-2}, 10^{-4}, 10^{-6}\}$. The primal residual evaluated at the dual solution z_ℓ^k is the difference between goal and discrete goal; see (3.11).

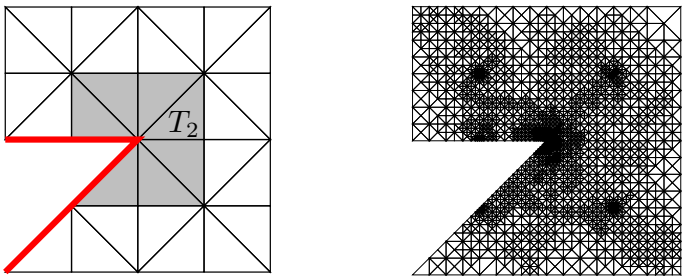


Figure 3.4: Left: Initial mesh $\mathcal{T}_0$. The shaded area is the set T_2 from Section (3.4.2) and the Dirichlet boundary at the re-entrant corner is marked in red. Right: Mesh after 13 iterations of Algorithm 3A with $\#\mathcal{T}_{13} = 4534$.

3.4.2 Geometrical singularity

Our second example is the classical example of a geometric singularity on the so-called Z-shape $\Omega = (-1,1)^2 \setminus \operatorname{conv}\{(-1,-1),(0,0),(-1,0)\}$, where Γ_D is only the re-entrant corner (cf. Figure 3.4). The primal problem is (3.38) with $f = 0$ and $\phi = \nabla u^\star \cdot \boldsymbol{n}$, where the exact solution in polar coordinates $r(x)$ and $\varphi(x)$ of $x \in \mathbb{R}^2$ is prescribed as

$$u^\star(x) = r(x)^{4/7} \sin(\tfrac{4}{7}\varphi(x) + \tfrac{3\pi}{7}).$$

The goal functional is (3.39) with $\omega = T_2 := (0.5, 0.5)^2 \cap \Omega$ and $\boldsymbol{g}_0 = (-1,-1)$ and can be computed directly via numerical integration to be

$$G(u^\star) = \int_{T_2} \left(\frac{\partial u^\star}{\partial x_1} + \frac{\partial u^\star}{\partial x_2} \right) \mathrm{d}x \approx 0.82962247157810.$$

In Figure 3.4, the initial triangulation $\mathcal{T}_0$ as well as the mesh after several iterations of Algorithm 3A can be seen. The adaptive algorithm resolves the singularity at the re-entrant corner, as well as critical points of the goal functional, which are at the corners of T_2.

Figure 3.5 shows the rate of the estimator product $\eta_\ell(u_\ell^k)\zeta_\ell(z_\ell^k)$ of the final iterates over $\#\mathcal{T}_\ell$ as well as the rate of Ξ_ℓ^k over $\texttt{work}(\ell,k)$ for all $(\ell,k) \in \mathcal{Q}$.

3.5 Proof of Theorem 3.5

The following core lemma extends one of the key observations of [GHPS21] to the present setting, where we stress that the nonlinear product structure of Δ_ℓ^k leads to technical challenges which go much beyond [GHPS21].

> **Lemma 3.9.** *Suppose* (A1)–(A3). *Then, there exist constants* $\mu, C_{\mathrm{aux}} > 0$, *and* $0 < q_{\mathrm{aux}} < 1$, *and some scalar sequence* $(R_\ell)_{\ell\in\mathbb{N}_0} \subset \mathbb{R}$ *such that the quasi-error product*
>
> $$\Delta_\ell^k := \big[\, |||u_\ell^\star - u_\ell^k||| + \mu\, \eta_\ell(u_\ell^k) \,\big] \big[\, |||z_\ell^\star - z_\ell^k||| + \mu\, \zeta_\ell(z_\ell^k) \,\big] \quad \text{for all } (\ell,k) \in \mathcal{Q}_0$$
>
> *satisfies the following statements* (i)–(v)*:*

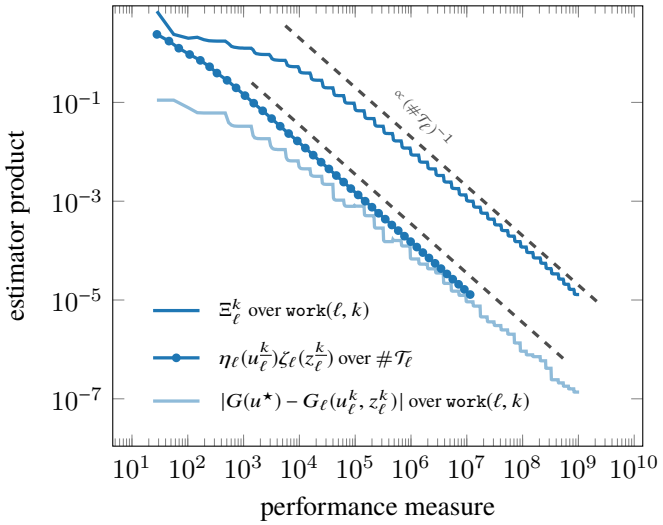


Figure 3.5: Rates of the estimator product for final iterates over $\#\mathcal{T}_\ell$ and Ξ_ℓ^k as well as goal error over $\mathtt{work}(\ell, k)$ for all $(\ell, k) \in \mathcal{Q}$.

(i) $\Delta_\ell^k \le \Delta_\ell^j$ *for all* $0 \le j \le k \le \underline{k}(\ell)$.

(ii) $\Delta_\ell^{k-1} \le C_{\mathrm{aux}}\, \Delta_\ell^k$ *if* $\underline{k}(\ell) < \infty$.

(iii) $\Delta_\ell^k \le q_{\mathrm{aux}}\, \Delta_\ell^{k-1}$ *for all* $0 < k < \underline{k}(\ell)$.

(iv) $\Delta_{\ell+1}^0 \le q_{\mathrm{aux}}\, \Delta_\ell^{k-1} + R_\ell$ *for all* $0 < \ell < \underline{\ell}$.

(v) $\sum_{\ell=\ell'}^{\underline{\ell}-1} R_\ell^2 \le C_{\mathrm{aux}}(\Delta_\ell^{k-1})^2$ *for all* $0 \le \ell' < \underline{\ell} - 1$.

The constants μ, C_{aux}, *and* q_{aux} *depend only on* C_{stab}, q_{red}, C_{rel}, *and* q_{ctr} *as well as on the (arbitrary) adaptivity parameters* $0 < \theta \le 1$ *and* $\lambda_{\mathrm{ctr}} > 0$.

For the following proofs, we define

$$\alpha_\ell^k := |||u_\ell^\star - u_\ell^k|||, \qquad x_\ell^\star := |||u_{\ell+1}^\star - u_\ell^\star|||,$$
$$\beta_\ell^k := |||z_\ell^\star - z_\ell^k|||, \qquad y_\ell^\star := |||z_{\ell+1}^\star - z_\ell^\star|||,$$

such that the quasi-error product reads $\Delta_\ell^k = \big[\alpha_\ell^k + \mu\,\eta_\ell(u_\ell^k)\big]\big[\beta_\ell^k + \mu\,\zeta_\ell(z_\ell^k)\big]$ with a free parameter $\mu > 0$ which will be fixed below.

Proof of Lemma 3.9(i). Recall from (3.24) that $u_\ell^k = u_\ell^{\underline{m}}$ for all $\underline{m}(\ell) < k \le \underline{k}(\ell)$. Thus, we have that

$$\alpha_\ell^k + \mu\,\eta_\ell(u_\ell^k) = \alpha_\ell^{\underline{m}} + \mu\,\eta_\ell(u_\ell^{\underline{m}}) \quad \text{for all } \underline{m}(\ell) < k \le \underline{k}(\ell).$$

For $0 < k < \underline{m}(\ell)$, on the other hand, the solution u_ℓ^k is obtained by one step of the iterative solver.

From stability (A1) and solver contraction (3.10), we have for all $0 \le j < k \le \underline{m}(\ell)$ that

$$\begin{aligned}\alpha_\ell^k + \mu\,\eta_\ell(u_\ell^k) &\overset{\text{(A1)}}{\le} \alpha_\ell^k + \mu\,\big[\eta_\ell(u_\ell^j) + C_{\mathrm{stab}}\|\!|u_\ell^k - u_\ell^j|\!\|\big] \\ &\overset{(3.10)}{\le} \big(q_{\mathrm{ctr}}^{k-j} + \mu C_{\mathrm{stab}}(1 + q_{\mathrm{ctr}}^{k-j})\big)\alpha_\ell^j + \mu\,\eta_\ell(u_\ell^j) \le \big(q_{\mathrm{ctr}} + 2\mu C_{\mathrm{stab}}\big)\alpha_\ell^j + \mu\,\eta_\ell(u_\ell^j).\end{aligned}$$

If μ is chosen small enough such that $q_{\mathrm{ctr}} + 2\mu C_{\mathrm{stab}} \le 1$, together with the trivial case $j = k$, the last two equations show that

$$\alpha_\ell^k + \mu\,\eta_\ell(u_\ell^k) \le \alpha_\ell^j + \mu\,\eta_\ell(u_\ell^j) \quad \text{for all } 0 \le j \le k \le \underline{k}(\ell).$$

The same argument shows that

$$\beta_\ell^k + \mu\,\zeta_\ell(z_\ell^k) \le \beta_\ell^j + \mu\,\zeta_\ell(z_\ell^j). \quad \text{for all } 0 \le j \le k \le \underline{k}(\ell). \tag{3.41}$$

Multiplication of the last two estimates shows the assertion. □

Proof of Lemma 3.9(ii). Recall that for the index $\underline{k}(\ell)$ there holds (3.23). From the triangle inequality, we thus get for the primal estimator that

$$\alpha_\ell^{k-1} = \|\!|u_\ell^\star - u_\ell^{k-1}|\!\| \le \|\!|u_\ell^\star - u_\ell^k|\!\| + \|\!|u_\ell^k - u_\ell^{k-1}|\!\| \overset{(3.23)}{\le} \alpha_\ell^k + \lambda_{\mathrm{ctr}}\,\eta_\ell(u_\ell^k).$$

Furthermore, stability (A1) leads to

$$\eta_\ell(u_\ell^{k-1}) \overset{\text{(A1)}}{\le} \eta_\ell(u_\ell^k) + C_{\mathrm{stab}}\,\|\!|u_\ell^k - u_\ell^{k-1}|\!\| \overset{(3.23)}{\le} \big(1 + \lambda_{\mathrm{ctr}} C_{\mathrm{stab}}\big)\eta_\ell(u_\ell^k).$$

Combining the last two estimates, we see that

$$\alpha_\ell^{k-1} + \mu\,\eta_\ell(u_\ell^{k-1}) \le \big(1 + \lambda_{\mathrm{ctr}}(C_{\mathrm{stab}} + \mu^{-1})\big)\,\big[\,\alpha_\ell^k + \mu\,\eta_\ell(u_\ell^k)\,\big].$$

Together with the analogous estimate for $\beta_\ell^{k-1} + \mu\,\zeta_\ell(z_\ell^{k-1})$, we conclude the proof with $C_{\mathrm{aux}} = \big(1 + \lambda_{\mathrm{ctr}}(C_{\mathrm{stab}} + \mu^{-1})\big)^2$. □

Proof of Lemma 3.9(iii). Without loss of generality, suppose that $\underline{k}(\ell) = \underline{m}(\ell)$ and thus $\|\!|u_\ell^k - u_\ell^{k-1}|\!\| > \lambda_{\mathrm{ctr}}\,\eta_\ell(u_\ell^k)$. Then, this yields that

$$\eta_\ell(u_\ell^k) < \lambda_{\mathrm{ctr}}^{-1}\,\|\!|u_\ell^k - u_\ell^{k-1}|\!\| \overset{(3.21)}{\le} \lambda_{\mathrm{ctr}}^{-1}\,(1 + q_{\mathrm{ctr}})\,\alpha_\ell^{k-1} \quad \text{for all } 0 < k < \underline{k}(\ell).$$

With contraction of the solver (3.10), this leads to

$$\alpha_\ell^k + \mu\,\eta_\ell(u_\ell^k) \le q_{\mathrm{ctr}}\alpha_\ell^{k-1} + \mu\lambda_{\mathrm{ctr}}^{-1}(1 + q_{\mathrm{ctr}})\,\alpha_\ell^{k-1} \quad \text{for all } 0 < k < \underline{k}(\ell).$$

From (3.41) for μ small enough, we see that $\beta_\ell^k + \mu\,\zeta_\ell(z_\ell^k) \le \beta_\ell^{k-1} + \mu\,\zeta_\ell(z_\ell^{k-1})$. Together with the previous estimate, this shows that

$$\Delta_\ell^k \le \big(q_{\mathrm{ctr}} + \mu\lambda_{\mathrm{ctr}}^{-1}(1 + q_{\mathrm{ctr}})\big)\Delta_\ell^{k-1}. \tag{3.42}$$

Up to the choice of μ, this concludes the proof. □

Proof of Lemma 3.9(iv). First, we note that $\eta_\ell(u_\ell^k)\zeta_\ell(z_\ell^k) \neq 0$, according to Algorithm 3A(iii) and the assumption that $\ell < \underline{\ell}$. From reduction of the solver (3.10) and nested iteration, we get that

$$\begin{aligned} \alpha_{\ell+1}^0 &= \|\!| u_{\ell+1}^\star - u_\ell^k |\!\| \le \|\!| u_{\ell+1}^\star - u_\ell^\star |\!\| + q_{\mathrm{ctr}} \|\!| u_\ell^\star - u_\ell^{k-1} |\!\| = x_\ell^\star + q_{\mathrm{ctr}}\, \alpha_\ell^{k-1}, \\ \beta_{\ell+1}^0 &= \|\!| z_{\ell+1}^\star - z_\ell^k |\!\| \le \|\!| z_{\ell+1}^\star - z_\ell^\star |\!\| + q_{\mathrm{ctr}} \|\!| z_\ell^\star - z_\ell^{k-1} |\!\| = y_\ell^\star + q_{\mathrm{ctr}}\, \beta_\ell^{k-1} \end{aligned} \tag{3.43}$$

and thus

$$\alpha_{\ell+1}^0 \beta_{\ell+1}^0 \le q_{\mathrm{ctr}}^2\, \alpha_\ell^{k-1} \beta_\ell^{k-1} + q_{\mathrm{ctr}}(\alpha_\ell^{k-1} y_\ell^\star + \beta_\ell^{k-1} x_\ell^\star) + x_\ell^\star y_\ell^\star. \tag{3.44}$$

For the estimator terms, we have with stability (A1) and reduction (A2) that

$$\begin{aligned} \eta_{\ell+1}(u_{\ell+1}^0)^2 = \eta_{\ell+1}(u_\ell^k)^2 &= \eta_{\ell+1}(\mathcal{T}_{\ell+1} \cap \mathcal{T}_\ell, u_\ell^k)^2 + \eta_{\ell+1}(\mathcal{T}_{\ell+1} \setminus \mathcal{T}_\ell, u_\ell^k)^2 \\ &\le \eta_\ell(\mathcal{T}_{\ell+1} \cap \mathcal{T}_\ell, u_\ell^k)^2 + q_{\mathrm{red}}^2\, \eta_\ell(\mathcal{T}_\ell \setminus \mathcal{T}_{\ell+1}, u_\ell^k)^2 \\ &= \eta_\ell(u_\ell^k)^2 - (1 - q_{\mathrm{red}}^2)\, \eta_\ell(\mathcal{T}_\ell \setminus \mathcal{T}_{\ell+1}, u_\ell^k)^2. \end{aligned}$$

On the one hand, with $C_1 := C_{\mathrm{stab}}(1 + q_{\mathrm{red}})$, this implies that

$$\eta_{\ell+1}(u_{\ell+1}^0) \le \eta_\ell(u_\ell^k) \overset{\text{(A1)}}{\le} \eta_\ell(u_\ell^{k-1}) + C_{\mathrm{stab}} \|\!| u_\ell^k - u_\ell^{k-1} |\!\| \overset{(3.21)}{\le} \eta_\ell(u_\ell^{k-1}) + C_1\, \alpha_\ell^{k-1}. \tag{3.45}$$

On the other hand, with $0 < q_\theta := 1 - (1 - q_{\mathrm{red}}^2)\theta < 1$, we get that

$$\frac{\eta_{\ell+1}(u_{\ell+1}^0)^2}{\eta_\ell(u_\ell^k)^2} \le q_\theta + (1 - q_{\mathrm{red}}^2)\Big[\theta - \frac{\eta_\ell(\mathcal{T}_\ell \setminus \mathcal{T}_{\ell+1}, u_\ell^k)^2}{\eta_\ell(u_\ell^k)^2}\Big]. \tag{3.46}$$

Using (3.46), the corresponding estimate for the dual estimator, and the Young inequality, we obtain that

$$\frac{\eta_{\ell+1}(u_{\ell+1}^0)}{\eta_\ell(u_\ell^k)} \frac{\zeta_{\ell+1}(z_{\ell+1}^0)}{\zeta_\ell(z_\ell^k)} \le q_\theta + \frac{(1 - q_{\mathrm{red}}^2)}{2}\Big[2\theta - \frac{\eta_\ell(\mathcal{T}_\ell \setminus \mathcal{T}_{\ell+1}, u_\ell^k)^2}{\eta_\ell(u_\ell^k)^2} - \frac{\zeta_\ell(\mathcal{T}_\ell \setminus \mathcal{T}_{\ell+1}, z_\ell^k)^2}{\zeta_\ell(z_\ell^k)^2}\Big].$$

The marking criterion (3.18), which is applicable due to $\ell < \underline{\ell}$, estimates the term in brackets by zero. Thus stability (A1) leads to

$$\begin{aligned} &\eta_{\ell+1}(u_{\ell+1}^0)\zeta_{\ell+1}(z_{\ell+1}^0) \le q_\theta\, \eta_\ell(u_\ell^k)\zeta_\ell(z_\ell^k) \\ &\quad \overset{\text{(A1)}}{\le} q_\theta \big[\eta_\ell(u_\ell^{k-1}) + C_{\mathrm{stab}} \|\!| u_\ell^k - u_\ell^{k-1} |\!\|\big]\big[\zeta_\ell(z_\ell^{k-1}) + C_{\mathrm{stab}} \|\!| z_\ell^k - z_\ell^{k-1} |\!\|\big] \\ &\quad \overset{(3.21)}{\le} q_\theta\, \eta_\ell(u_\ell^{k-1})\zeta_\ell(z_\ell^{k-1}) + q_\theta C_1 \big[\eta_\ell(u_\ell^{k-1})\beta_\ell^{k-1} + \zeta_\ell(z_\ell^{k-1})\alpha_\ell^{k-1}\big] + C_1^2\, \alpha_\ell^{k-1}\beta_\ell^{k-1}. \end{aligned} \tag{3.47}$$

For the mixed terms in $\Delta_{\ell+1}^0$, we have with (3.43) and (3.45) that

$$\begin{aligned} \eta_{\ell+1}(u_{\ell+1}^0)\beta_{\ell+1}^0 &\le \big[\eta_\ell(u_\ell^{k-1}) + C_1\, \alpha_\ell^{k-1}\big]\big[y_\ell^\star + q_{\mathrm{ctr}}\, \beta_\ell^{k-1}\big] \\ &= q_{\mathrm{ctr}}\, \eta_\ell(u_\ell^{k-1})\beta_\ell^{k-1} + \eta_\ell(u_\ell^{k-1}) y_\ell^\star + C_1\, \alpha_\ell^{k-1} y_\ell^\star + C_1 q_{\mathrm{ctr}}\, \alpha_\ell^{k-1}\beta_\ell^{k-1}. \end{aligned} \tag{3.48}$$

Analogously, we see that

$$\zeta_{\ell+1}(z_{\ell+1}^0)\alpha_{\ell+1}^0 \le q_{\mathrm{ctr}}\, \zeta_\ell(z_\ell^{k-1})\alpha_\ell^{k-1} + \zeta_\ell(z_\ell^{k-1}) x_\ell^\star + C_1\, \beta_\ell^{k-1} x_\ell^\star + C_1 q_{\mathrm{ctr}}\, \alpha_\ell^{k-1}\beta_\ell^{k-1}. \tag{3.49}$$

Combining (3.44) and (3.47)–(3.49), we get that

$$\begin{aligned}
\Delta_{\ell+1}^0 &= \alpha_{\ell+1}^0\beta_{\ell+1}^0 + \mu\left[\eta_{\ell+1}(u_{\ell+1}^0)\beta_{\ell+1}^0 + \zeta_{\ell+1}(z_{\ell+1}^0)\alpha_{\ell+1}^0\right] + \mu^2\,\eta_{\ell+1}(u_{\ell+1}^0)\zeta_{\ell+1}(z_{\ell+1}^0)\\
&\le q_{\mathrm{ctr}}^2\,\alpha_\ell^{k-1}\beta_\ell^{k-1} + q_{\mathrm{ctr}}(\alpha_\ell^{k-1}y_\ell^\star + \beta_\ell^{k-1}x_\ell^\star) + x_\ell^\star y_\ell^\star\\
&\quad + \mu\left[q_{\mathrm{ctr}}\,\eta_\ell(u_\ell^{k-1})\beta_\ell^{k-1} + \eta_\ell(u_\ell^{k-1})y_\ell^\star + C_1\,\alpha_\ell^{k-1}y_\ell^\star + C_1 q_{\mathrm{ctr}}\,\alpha_\ell^{k-1}\beta_\ell^{k-1}\right]\\
&\quad + \mu\left[q_{\mathrm{ctr}}\,\zeta_\ell(z_\ell^{k-1})\alpha_\ell^{k-1} + \zeta_\ell(z_\ell^{k-1})x_\ell^\star + C_1\,\beta_\ell^{k-1}x_\ell^\star + C_1 q_{\mathrm{ctr}}\,\alpha_\ell^{k-1}\beta_\ell^{k-1}\right]\\
&\quad + \mu^2\left[q_\theta\,\eta_\ell(u_\ell^{k-1})\zeta_\ell(z_\ell^{k-1}) + q_\theta C_1\big(\eta_\ell(u_\ell^{k-1})\beta_\ell^{k-1} + \zeta_\ell(z_\ell^{k-1})\alpha_\ell^{k-1}\big) + C_1^2\,\alpha_\ell^{k-1}\beta_\ell^{k-1}\right].
\end{aligned}$$

Rearranging the terms, we obtain that

$$\begin{aligned}
\Delta_{\ell+1}^0 \le{}& \big(q_{\mathrm{ctr}}^2 + 2\mu q_{\mathrm{ctr}}C_1 + \mu^2C_1^2\big)\,\alpha_\ell^{k-1}\beta_\ell^{k-1}\\
&+ \mu\,\big(q_{\mathrm{ctr}} + \mu q_\theta C_1\big)\left[\eta_\ell(u_\ell^{k-1})\beta_\ell^{k-1} + \zeta_\ell(z_\ell^{k-1})\alpha_\ell^{k-1}\right]\\
&+ \mu^2\,q_\theta\,\eta_\ell(u_\ell^{k-1})\zeta_\ell(z_\ell^{k-1}) + R_\ell,
\end{aligned}\tag{3.50}$$

where the remainder term is defined as

$$R_\ell := \mu\left[\eta_\ell(u_\ell^{k-1})y_\ell^\star + \zeta_\ell(z_\ell^{k-1})x_\ell^\star\right] + (q_{\mathrm{ctr}} + \mu C_1)\big[\alpha_\ell^{k-1}y_\ell^\star + \beta_\ell^{k-1}x_\ell^\star\big] + x_\ell^\star y_\ell^\star.\tag{3.51}$$

Up to the choice of μ, this concludes the proof. □

Proof of Lemma 3.9 (choosing μ). For Lemma 3.9(i), we choose μ small enough such that $q_{\mathrm{ctr}} + 2\mu C_{\mathrm{stab}} \le 1$. From (3.42) and (3.50) in the proofs of Lemma 3.9(iii)–(iv), we see that we additionally require

$$q_{\mathrm{ctr}} + \mu\lambda_{\mathrm{ctr}}^{-1}(1 + q_{\mathrm{ctr}}) < 1, \qquad q_{\mathrm{ctr}}^2 + 2\mu q_{\mathrm{ctr}}C_1 + \mu^2C_1^2 < 1, \quad \text{and} \quad q_{\mathrm{ctr}} + \mu q_\theta C_1 < 1.\tag{3.52}$$

Choosing μ small enough, we satisfy all estimates. We define $q_{\mathrm{aux}} < 1$ as the maximum of all terms in (3.52) and q_θ. □

Proof of Lemma 3.9(v). First, we note that from stability (A1) it follows that

$$\eta_\ell(u_\ell^{k-1}) \lesssim \eta_\ell(u_\ell^\star) + \alpha_\ell^{k-1} \quad \text{and} \quad \eta_\ell(u_\ell^\star)\zeta_\ell(z_\ell^\star) \lesssim \Delta_\ell^j \text{ for all } 0 \le j \le \underline{k}.\tag{3.53}$$

Furthermore, Galerkin orthogonality and reliability (A3) imply that, for all $n \in \mathbb{N}$ with $\ell' + n < \underline{\ell}$,

$$\sum_{\ell=\ell'}^{\ell'+n}(y_\ell^\star)^2 = \sum_{\ell=\ell'}^{\ell'+n}|||z_{\ell+1}^\star - z_\ell^\star|||^2 = |||z_{\ell'+n+1}^\star - z_{\ell'}^\star|||^2 \le |||z^\star - z_{\ell'}^\star|||^2 \overset{(A3)}{\lesssim} \zeta_{\ell'}(z_{\ell'}^\star)^2.\tag{3.54}$$

With (3.53) and (3.54) for $n = 1$, we can bound the remainder term from (3.51) by

$$R_\ell \lesssim \eta_\ell(u_\ell^\star)y_\ell^\star + \zeta_\ell(z_\ell^\star)x_\ell^\star + \alpha_\ell^{k-1}y_\ell^\star + \beta_\ell^{k-1}x_\ell^\star.$$

Next, let us recall from [CFPP14, Lemma 3.6] the quasi-monotonicity of the estimator, which follows from (A1)–(A3) and the Céa lemma, i.e., for all $\ell' \le \ell < \underline{\ell}$,

$$\eta_\ell(u_\ell^\star) \le \eta_{\ell'}(u_{\ell'}^\star) + C_{\mathrm{stab}}\,|||u_\ell^\star - u_{\ell'}^\star||| \le \eta_{\ell'}(u_{\ell'}^\star) + C_{\mathrm{stab}}\,|||u^\star - u_{\ell'}^\star||| \lesssim \eta_{\ell'}(u_{\ell'}^\star).\tag{3.55}$$

For $\eta_\ell(u_\ell^\star)y_\ell$, we get by summation for all $0 \le j \le \underline{k}(\ell')$ and all $n \in \mathbb{N}$ with $\ell' + n < \underline{\ell}$ that

$$\sum_{\ell=\ell'}^{\ell'+n} \eta_\ell(u_\ell^\star)^2 (y_\ell^\star)^2 \overset{(3.55)}{\lesssim} \eta_{\ell'}(u_{\ell'}^\star)^2 \sum_{\ell=\ell'}^{\ell'+n} (y_\ell^\star)^2 \overset{(3.54)}{\lesssim} \eta_{\ell'}(u_{\ell'}^\star)^2 \zeta_{\ell'}(z_{\ell'}^\star)^2 \overset{(3.53)}{\lesssim} (\Delta_{\ell'}^j)^2.$$

Analogously, we see that

$$\sum_{\ell=\ell'}^{\ell'+n} (x_\ell^\star)^2 \lesssim \eta_{\ell'}(u_{\ell'}^\star)^2 \quad \text{as well as} \quad \sum_{\ell=\ell'}^{\ell'+n} \zeta_\ell(z_\ell^\star)^2 (x_\ell^\star)^2 \lesssim (\Delta_{\ell'}^j)^2. \tag{3.56}$$

We proceed with $\alpha_\ell^{k-1} y_\ell^\star$. From (3.43) and the Young inequality with $\delta > 0$, we see for $0 < \ell' \le \ell < \underline{\ell}$ that

$$(\alpha_\ell^{k-1})^2 \le (\alpha_\ell^0)^2 \overset{(3.43)}{\le} (1 + \delta^{-1}) (x_{\ell-1}^\star)^2 + q_{\mathrm{ctr}}(1 + \delta) (\alpha_{\ell-1}^{k-1})^2.$$

For δ small enough such that $q_2 := q_{\mathrm{ctr}}(1 + \delta) < 1$ and all for $0 \le \ell \le \ell' < \underline{\ell}$, the geometric series proves that

$$(\alpha_\ell^{k-1})^2 \le (1 + \delta^{-1}) \sum_{j=\ell'}^{\ell-1} (x_j^\star)^2 + (\alpha_\ell^{k-1})^2 \sum_{j=0}^{\infty} q_2^j \overset{(3.56)}{\lesssim} \eta_{\ell'}(u_{\ell'}^\star)^2 + (\alpha_{\ell'}^{k-1})^2$$

and thus

$$\sum_{\ell=\ell'}^{\ell'+n} (\alpha_\ell^{k-1})^2 (y_\ell^\star)^2 \le \big[\eta_{\ell'}(u_{\ell'}^\star)^2 + (\alpha_{\ell'}^{k-1})^2\big] \sum_{\ell=\ell'}^{\ell'+n} (y_\ell^\star)^2 \overset{(3.54)}{\lesssim} \big[\eta_{\ell'}(u_{\ell'}^\star)^2 + (\alpha_{\ell'}^{k-1})^2\big] \zeta_{\ell'}(z_{\ell'}^\star)^2 \lesssim (\Delta_{\ell'}^{k-1})^2.$$

Analogously, we see that $\sum_{\ell=\ell'}^{\ell'+n} (\beta_\ell^{k-1})^2 (x_\ell^\star)^2 \lesssim (\Delta_{\ell'}^{k-1})^2$. Combining all estimates with

$$R_\ell^2 \lesssim \eta_\ell(u_\ell^\star)^2 (y_\ell^\star)^2 + \zeta_\ell(z_\ell^\star)^2 (x_\ell^\star)^2 + (\alpha_\ell^{k-1})^2 (y_\ell^\star)^2 + (\beta_\ell^{k-1})^2 (x_\ell^\star)^2,$$

we conclude the proof. □

With the foregoing auxiliary result, we are in the position to prove linear convergence.

Proof of Theorem 3.5. Let $(\ell, k) \in \mathcal{Q}$. We recall the quasi-error products

$$\begin{aligned} \Lambda_\ell^k &= \big[\, |||u_\ell^\star - u_\ell^k||| + \eta_\ell(u_\ell^k) \,\big] \big[\, |||z_\ell^\star - z_\ell^k||| + \zeta_\ell(z_\ell^k) \,\big], \\ \Delta_\ell^k &= \big[\, |||u_\ell^\star - u_\ell^k||| + \mu\, \eta_\ell(u_\ell^k) \,\big] \big[\, |||z_\ell^\star - z_\ell^k||| + \mu\, \zeta_\ell(z_\ell^k) \,\big] \end{aligned}$$

from Theorem 3.5 and Lemma 3.9, respectively. Note that

$$\Lambda_\ell^k \le \Delta_\ell^k \le \mu^2\, \Lambda_\ell^k \quad \text{if } \mu \ge 1, \qquad \Delta_\ell^k \le \Lambda_\ell^k \le \mu^{-2}\, \Delta_\ell^k \quad \text{if } \mu < 1,$$

which yields the equivalence

$$\min\{1, \mu^2\}\, \Lambda_\ell^k \le \Delta_\ell^k \le \max\{1, \mu^2\}\, \Lambda_\ell^k. \tag{3.57}$$

We first show linear convergence of Δ_ℓ^k. By Lemma 3.9(i), we can absorb the term $\Delta_{\ell'}^{\underline{k}} \le \Delta_{\ell'}^{k-1}$ for all ℓ'. Paying attention to the possible case $k = \underline{k}(\ell)$, this allows us to estimate

$$\sum_{\substack{(\ell',k')\in\mathcal{Q}\\ |(\ell',k')|\ge|(\ell,k)|}} (\Delta_{\ell'}^{k'})^2 \lesssim (\Delta_\ell^k)^2 + \sum_{k'=k}^{\underline{k}(\ell)-1} (\Delta_\ell^{k'})^2 + \sum_{\ell'=\ell+1}^{\underline{\ell}} \sum_{k'=0}^{\underline{k}(\ell')-1} (\Delta_{\ell'}^{k'})^2.$$

Lemma 3.9(iii) shows uniform reduction of the quasi-error on every level. This yields that

$$\sum_{\substack{(\ell',k')\in\mathcal{Q}\\ |(\ell',k')|\ge|(\ell,k)|}} (\Delta_{\ell'}^{k'})^2 \lesssim (\Delta_\ell^k)^2 \sum_{k'=k}^{\underline{k}(\ell)} q_{\mathrm{aux}}^{2(k'-k)} + \sum_{\ell'=\ell+1}^{\underline{\ell}} (\Delta_{\ell'}^0)^2 \sum_{k'=0}^{\underline{k}(\ell')-1} q_{\mathrm{aux}}^{2k'} \lesssim (\Delta_\ell^k)^2 + \sum_{\ell'=\ell+1}^{\underline{\ell}} (\Delta_{\ell'}^0)^2.$$

To estimate the sum over all levels, we use that, for the refinement step, Lemma 3.9(iv) shows contraction up to a remainder term. The Young inequality with $\delta > 0$ and Lemma 3.9(i) then prove that

$$\begin{aligned}(\Delta_{\ell'}^0)^2 &\le q_{\mathrm{aux}}^2(1+\delta)\,(\Delta_{\ell'-1}^{\underline{k}-1})^2 + (1+\delta^{-1})\,R_{\ell'-1}^2\\ &\le q_{\mathrm{aux}}^2(1+\delta)\,(\Delta_{\ell'-1}^0)^2 + (1+\delta^{-1})\,R_{\ell'-1}^2 \quad \text{for all } 0<\ell'\le\underline{\ell}.\end{aligned}$$

Choosing δ small enough such that $q := q_{\mathrm{aux}}^2(1+\delta) < 1$, we obtain from repeatedly applying the previous estimates that

$$(\Delta_{\ell'}^0)^2 \le q^{\ell'-\ell}\,(\Delta_\ell^{\underline{k}-1})^2 + (1+\delta^{-1}) \sum_{n=\ell}^{\ell'-1} q^{(\ell'-1)-n}\,R_n^2 \quad \text{for all } 0\le\ell<\ell'\le\underline{\ell}.$$

Using this estimate and a change of summation indices, the geometric series and Lemma 3.9(v) uniformly bound the sum over all levels by

$$\begin{aligned}\sum_{\ell'=\ell+1}^{\underline{\ell}} (\Delta_{\ell'}^0)^2 &\lesssim \sum_{\ell'=\ell+1}^{\underline{\ell}} \Big[q^{\ell'-\ell}\,(\Delta_\ell^{\underline{k}-1})^2 + \sum_{n=\ell}^{\ell'-1} q^{(\ell'-1)-n}\,R_n^2\Big]\\ &\lesssim (\Delta_\ell^{\underline{k}-1})^2 + \sum_{n=\ell}^{\underline{\ell}-1} R_n^2 \sum_{i=0}^{\infty} q^i \lesssim (\Delta_\ell^{\underline{k}-1})^2 + \sum_{n=\ell}^{\underline{\ell}-1} R_n^2 \overset{\text{(v)}}{\lesssim} (\Delta_\ell^{\underline{k}-1})^2.\end{aligned}$$

Combining the estimates above, we obtain that

$$\sum_{\substack{(\ell',k')\in\mathcal{Q}\\ |(\ell',k')|\ge|(\ell,k)|}} (\Delta_{\ell'}^{k'})^2 \lesssim (\Delta_\ell^k)^2 + \sum_{\ell'=\ell+1}^{\underline{\ell}} (\Delta_{\ell'}^0)^2 \lesssim (\Delta_\ell^k)^2 + (\Delta_\ell^{\underline{k}-1})^2.$$

In the case $k < \underline{k}(\ell)$, Lemma 3.9(i) proves that

$$\sum_{\substack{(\ell',k')\in\mathcal{Q}\\ |(\ell',k')|\ge|(\ell,k)|}} (\Delta_{\ell'}^{k'})^2 \le C\,(\Delta_\ell^k)^2.$$

In the case $k = \underline{k}(\ell)$, this follows with Lemma 3.9(ii). In either case, the constant $C > 0$ depends only on C_{aux} and q_{aux} from Lemma 3.9. Basic calculus then provides the existence of $C'_{\mathrm{lin}} := (1+C)^{1/2} > 1$ and $0 < q_{\mathrm{lin}} := (1-C^{-1})^{-1/2} < 1$ such that

$$\Delta_{\ell'}^{k'} \le C'_{\mathrm{lin}} q_{\mathrm{lin}}^{|(\ell',k')|-|(\ell,k)|}\, \Delta_\ell^k \quad \text{for all } (\ell,k),(\ell',k') \in \mathcal{Q} \text{ with } (\ell',k') \ge (\ell,k);$$

see [CFPP14, Lemma 4.9]. Finally, the claim of Theorem 3.5 follows from (3.57) with $C_{\mathrm{lin}} = \max\{\mu^{-2},\mu^2\}\, C'_{\mathrm{lin}}$. □

3.6 Proof of Theorem 3.7 (optimal rates)

We recall the following comparison lemma from [FGH+16]. While [FGH+16] is concerned with point errors in boundary element computations, we stress that the proof of [FGH+16, Lemma 14] works on a completely abstract level and thus is applicable here as well.

Lemma 3.10 ([FGH+16, Lemma 14])**.** *The overlay estimate* (3.15) *and the axioms* (A1)–(A2) *and* (A4) *yield the existence of a constant $C_1 > 0$ such that, given $0 < \kappa < 1$, each mesh $\mathcal{T}_H \in \mathbb{T}$ admits some refinement $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$ such that for all $s,t > 0$, it holds that*

$$\eta_h(u_h^\star)^2 \zeta_h(z_h^\star)^2 \le \kappa^2 \eta_H(u_H^\star)^2 \zeta_H(z_H^\star)^2, \tag{3.58a}$$

$$\#\mathcal{T}_h - \#\mathcal{T}_H \le 2\big(C_1\kappa^{-1}\|u^\star\|_{\mathbb{A}_s}\|z^\star\|_{\mathbb{A}_t}\big)^{1/(s+t)}\big(\eta_H(u_H^\star)\zeta(z_H^\star)\big)^{1/(s+t)}. \tag{3.58b}$$

The constant C_1 depends only on C_{stab}, q_{red}, and C_{drel}. □

Note that (3.58a) immediately implies that

$$\eta_h(u_h)^2 \le \kappa \eta_H(u_H^\star)^2 \quad \text{or} \quad \zeta_h(z_h^\star)^2 \le \kappa \zeta_H(z_H^\star)^2. \tag{3.59}$$

We will employ this lemma in combination with the so-called optimality of Dörfler marking from [CFPP14].

Lemma 3.11 ([CFPP14, Proposition 4.12])**.** *Under* (A1) *and* (A4)*, for all $0 < \Theta' < 1/(1 + C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2)$, there exists $0 < \kappa_{\Theta'} < 1$ such that for all $\mathcal{T}_H \in \mathbb{T}$ and all $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$,* (3.59) *with $\kappa = \kappa_{\Theta'}$ implies that*

$$\Theta'\eta_H(u_H^\star)^2 \le \eta_H(\mathcal{T}_H \setminus \mathcal{T}_h, u_H^\star)^2 \quad \text{or} \quad \Theta'\zeta_H(z_H^\star)^2 \le \zeta_H(\mathcal{T}_H \setminus \mathcal{T}_h, z_H^\star)^2. \tag{3.60}$$

The constant $\kappa_{\Theta'}$ depends only on C_{stab}, C_{drel}, and Θ'. □

The next lemma is already implicitly found in [GHPS18]. It shows that, if $\lambda_{\mathrm{ctr}} > 0$ is sufficiently small, then Dörfler marking for the exact discrete solution implicitly implies Dörfler marking for the approximate discrete solution. This will turn out to be the key observation to prove optimal convergence rates. We include the proof for the convenience of the reader.

Lemma 3.12. *Suppose* (A1)–(A3)*. Let $0 < \Theta \le 1$ and $0 < \lambda_{\mathrm{ctr}} < \lambda_\star := (1-q_{\mathrm{ctr}})/(q_{\mathrm{ctr}}C_{\mathrm{stab}})$. Define $\Theta' := \big(\frac{\sqrt{\Theta}+\lambda_{\mathrm{ctr}}/\lambda_\star}{1-\lambda_{\mathrm{ctr}}/\lambda_\star}\big)^2$. Then, as soon as the iterative solver terminates* (3.23)*, there hold the following statements* (i)–(iv) *for all $0 \le \ell < \underline{\ell}$ and all $\mathcal{U}_\ell \subseteq \mathcal{T}_\ell$:*

(i) $(1-\lambda_{\mathrm{ctr}}/\lambda_\star)\,\eta_\ell(u_\ell^m) \le \eta_\ell(u_\ell^\star) \le (1+\lambda_{\mathrm{ctr}}/\lambda_\star)\,\eta_\ell(u_\ell^m)$.

(ii) $\Theta\,\eta_\ell(u_\ell^m)^2 \le \eta_\ell(\mathcal{U}_\ell, u_\ell^m)^2$ *provided that* $\Theta'\,\eta_\ell(u_\ell^\star)^2 \le \eta_\ell(\mathcal{U}_\ell, u_\ell^\star)^2$.

(iii) $(1-\lambda_{\mathrm{ctr}}/\lambda_\star)\,\zeta_\ell(z_\ell^n) \le \zeta_\ell(z_\ell^\star) \le (1+\lambda_{\mathrm{ctr}}/\lambda_\star)\,\zeta_\ell(z_\ell^n)$.

(iv) $\Theta\,\zeta_\ell(z_\ell^n) \le \zeta_\ell(\mathcal{U}_\ell, z_\ell^n)$ *provided that* $\Theta'\,\zeta_\ell(z_\ell^\star)^2 \le \zeta_\ell(\mathcal{U}_\ell, z_\ell^\star)^2$.

Proof. It holds that

$$\eta_\ell(\mathcal{U}_\ell, u_\ell^\star) \overset{\text{(A1)}}{\le} \eta_\ell(\mathcal{U}_\ell, u_\ell^m) + C_{\mathrm{stab}}\,|||u_\ell^\star - u_\ell^m||| \overset{(3.21)}{\le} \eta_\ell(\mathcal{U}_\ell, u_\ell^m) + C_{\mathrm{stab}}\,\frac{q_{\mathrm{ctr}}}{1-q_{\mathrm{ctr}}}\,|||u_\ell^m - u_\ell^{m-1}|||$$
$$\overset{(3.23)}{\le} \eta_\ell(\mathcal{U}_\ell, u_\ell^m) + C_{\mathrm{stab}}\,\frac{q_{\mathrm{ctr}}}{1-q_{\mathrm{ctr}}}\,\lambda_{\mathrm{ctr}}\,\eta_\ell(u_\ell^m) \quad = \quad \eta_\ell(\mathcal{U}_\ell, u_\ell^m) + \frac{\lambda_{\mathrm{ctr}}}{\lambda_\star}\,\eta_\ell(u_\ell^m).$$

The same argument proves that

$$\eta_\ell(\mathcal{U}_\ell, u_\ell^m) \le \eta_\ell(\mathcal{U}_\ell, u_\ell^\star) + \frac{\lambda_{\mathrm{ctr}}}{\lambda_\star}\,\eta_\ell(u_\ell^m).$$

For $\mathcal{U}_\ell = \mathcal{T}_\ell$, the latter two estimates lead to

$$(1-\lambda_{\mathrm{ctr}}/\lambda_\star)\,\eta_\ell(u_\ell^m) \le \eta_\ell(u_\ell^\star) \le (1+\lambda_{\mathrm{ctr}}/\lambda_\star)\,\eta_\ell(u_\ell^m).$$

This concludes the proof of (i). To see (ii), we use the assumption

$$(1-\lambda_{\mathrm{ctr}}/\lambda_\star)\,\sqrt{\Theta'}\,\eta_\ell(u_\ell^m) \overset{\text{(i)}}{\le} \sqrt{\Theta'}\,\eta_\ell(u_\ell^\star) \le \eta_\ell(\mathcal{U}_\ell, u_\ell^\star) \le \eta_\ell(\mathcal{U}_\ell, u_\ell^m) + \frac{\lambda_{\mathrm{ctr}}}{\lambda_\star}\,\eta_\ell(u_\ell^m).$$

Noting that $\sqrt{\Theta} = (1-\lambda_{\mathrm{ctr}}/\lambda_\star)\,\sqrt{\Theta'} - \lambda_{\mathrm{ctr}}/\lambda_\star$, this concludes the proof of (ii). The remaining claims (iii)–(iv) follow verbatim. □

Proof of Theorem 3.7. By Corollary 3.6, it is sufficient to prove that

$$C_{s+t} = \sup_{(\ell,k)\in\mathcal{Q}} \big(\#\mathcal{T}_\ell - \#\mathcal{T}_0 + 1\big)^{s+t}\,\Lambda_\ell^k \lesssim \max\{\|u^\star\|_{\mathbb{A}_s}\|z^\star\|_{\mathbb{A}_t}, \Lambda_0^0\}.$$

We prove this inequality in two steps.

Step 1: In this step, we bound the number of marked elements $\#\mathcal{M}_{\ell'}$ for arbitrary $0 \le \ell' < \underline{\ell}$. Let $\Theta > 0$ and corresponding Θ' from Lemma 3.12 such that

$$\Theta' = \left(\frac{\sqrt{\Theta} + \lambda_{\mathrm{ctr}}/\lambda_\star}{1-\lambda_{\mathrm{ctr}}/\lambda_\star}\right)^2 < \frac{1}{1 + C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2}. \tag{3.61}$$

Let $\mathcal{T}_{h(\ell')} \in \mathbb{T}(\mathcal{T}_{\ell'})$ be the corresponding mesh as in Lemma 3.10. With Lemma 3.11, this yields that

$$\Theta'\eta_{\ell'}(u_{\ell'}^\star)^2 \le \eta_{\ell'}(\mathcal{T}_{\ell'} \setminus \mathcal{T}_{h(\ell')}, u_{\ell'}^\star)^2 \quad \text{or} \quad \Theta'\zeta_{\ell'}(z_{\ell'}^\star)^2 \le \zeta_{\ell'}(\mathcal{T}_{\ell'} \setminus \mathcal{T}_{h(\ell')}, z_{\ell'}^\star)^2.$$

Lemma 3.12 with $\mathcal{U}_{\ell'} = \mathcal{T}_{\ell'} \setminus \mathcal{T}_{h(\ell')}$ shows that

$$\Theta\eta_{\ell'}(u_{\ell'}^m)^2 \le \eta_{\ell'}(\mathcal{T}_{\ell'} \setminus \mathcal{T}_{h(\ell')}, u_{\ell'}^\star)^2 \quad \text{or} \quad \Theta\zeta_{\ell'}(z_{\ell'}^n)^2 \le \zeta_{\ell'}(\mathcal{T}_{\ell'} \setminus \mathcal{T}_{h(\ell')}, z_{\ell'}^\star)^2. \tag{3.62}$$

We consider the marking strategies from Remark 3.2 separately.

For strategy (a), we have with $\Theta := 2\theta$ and assumption (3.33) that (3.61) is satisfied. Hence, (3.62) implies that there holds (3.18), i.e.,

$$2\theta\eta_{\ell'}(u_{\ell'}^m)^2\zeta_{\ell'}(z_{\ell'}^n)^2 \le \eta_{\ell'}(\mathcal{T}_{\ell'}\setminus\mathcal{T}_{h(\ell')}, u_{\ell'}^m)^2\zeta_{\ell'}(z_{\ell'}^n)^2 + \eta_{\ell'}(u_{\ell'}^m)^2\zeta_{\ell'}(\mathcal{T}_{\ell'}\setminus\mathcal{T}_{h(\ell')}, z_{\ell'}^n)^2.$$

By assumption of Theorem 3.7, $\mathcal{M}_{\ell'}$ is essentially minimal with (3.18). We infer that

$$\#\mathcal{M}_{\ell'} \le C_{\mathrm{mark}}\#(\mathcal{T}_{\ell'}\setminus\mathcal{T}_{h(\ell')}) \overset{(3.13)}{\lesssim} \#\mathcal{T}_{h(\ell')} - \#\mathcal{T}_{\ell'}. \tag{3.63}$$

For the strategies (b)–(c), we set $\Theta = \theta$ and note that assumption (3.33) (as well as the weaker assumption (3.35)) imply (3.61), and hence (3.62). Again, by assumption of Theorem 3.7, $\mathcal{M}_\ell$ is chosen essentially minimal (with an additional factor two for the strategy (c)) such that (3.62) holds. For all three strategies, we therefore conclude that

$$\begin{aligned}\#\mathcal{M}_{\ell'} \lesssim \#\mathcal{T}_{h(\ell')} - \#\mathcal{T}_{\ell'} &\overset{(3.58b)}{\lesssim} \big(\|u^\star\|_{\mathbb{A}_s}\|z^\star\|_{\mathbb{A}_t}\big)^{1/(s+t)}\big(\eta_{\ell'}(u_{\ell'}^\star)\zeta_{\ell'}(z_{\ell'}^\star)\big)^{-1/(s+t)}\\ &\overset{\text{Lem.3.12}}{\lesssim} \big(\|u^\star\|_{\mathbb{A}_s}\|z^\star\|_{\mathbb{A}_t}\big)^{1/(s+t)}\big(\eta_{\ell'}(u_{\ell'}^m)\zeta_{\ell'}(z_{\ell'}^n)\big)^{-1/(s+t)}.\end{aligned}$$

Recall that (3.21) and (3.23) give that $\eta_{\ell'}(u_{\ell'}^k)\zeta_{\ell'}(z_{\ell'}^k) \simeq \Lambda_{\ell'}^k$. This finally shows that

$$\#\mathcal{M}_{\ell'} \lesssim \big(\|u^\star\|_{\mathbb{A}_s}\|z^\star\|_{\mathbb{A}_t}\big)^{1/(s+t)}(\Lambda_{\ell'}^k)^{-1/(s+t)}.$$

Step 2: Let $(\ell,k)\in\mathcal{Q}$. First, we consider $\ell > 0$ and thus $\#\mathcal{T}_\ell > \#\mathcal{T}_0$. The closure estimate and Step 1 prove that

$$\begin{aligned}\#\mathcal{T}_\ell - \#\mathcal{T}_0 + 1 \simeq \#\mathcal{T}_\ell - \#\mathcal{T}_0 &\overset{(3.14)}{\lesssim} \sum_{\ell'=0}^{\ell-1}\#\mathcal{M}_{\ell'} \lesssim \big(\|u^\star\|_{\mathbb{A}_s}\|z^\star\|_{\mathbb{A}_t}\big)^{1/(s+t)}\sum_{\ell'=0}^{\ell-1}(\Lambda_{\ell'}^k)^{-1/(s+t)}\\ &\le \big(\|u^\star\|_{\mathbb{A}_s}\|z^\star\|_{\mathbb{A}_t}\big)^{1/(s+t)}\sum_{\substack{(\ell',k')\in\mathcal{Q}\\ |(\ell',k')|\ge|(\ell,k)|}}(\Lambda_{\ell'}^{k'})^{-1/(s+t)}.\end{aligned}$$

Linear convergence of Theorem 3.5, further shows that

$$\begin{aligned}\#\mathcal{T}_\ell - \#\mathcal{T}_0 + 1 &\lesssim \big(\|u^\star\|_{\mathbb{A}_s}\|z^\star\|_{\mathbb{A}_t}\big)^{1/(s+t)}C_{\mathrm{lin}}^{1/(s+t)}(\Lambda_\ell^k)^{-1/(s+t)}\sum_{\substack{(\ell',k')\in\mathcal{Q}\\ |(\ell',k')|\ge|(\ell,k)|}}(q_{\mathrm{lin}}^{1/(s+t)})^{|(\ell,k)|-|(\ell',k')|}\\ &\le \big(\|u^\star\|_{\mathbb{A}_s}\|z^\star\|_{\mathbb{A}_t}\big)^{1/(s+t)}\frac{C_{\mathrm{lin}}^{1/(s+t)}}{1-q_{\mathrm{lin}}^{1/(s+t)}}C_{\mathrm{lin}}^{1/(s+t)}(\Lambda_\ell^k)^{-1/(s+t)}.\end{aligned}$$

Rearranging this estimate, we see that

$$(\#\mathcal{T}_\ell - \#\mathcal{T}_0 + 1)^{s+t}\Lambda_\ell^k \lesssim \|u^\star\|_{\mathbb{A}_s}\|z^\star\|_{\mathbb{A}_t} \quad \text{for all } (\ell,k)\in\mathcal{Q} \text{ with } \ell > 0.$$

It remains to consider $\ell = 0$. By Theorem 3.5, we have that

$$(\#\mathcal{T}_\ell - \#\mathcal{T}_0 + 1)^{s+t}\Lambda_\ell^k = \Lambda_0^k \lesssim \Lambda_0^0 \quad \text{for all } (\ell,k)\in\mathcal{Q} \text{ with } \ell = 0.$$

This concludes the proof. □

4 Adaptive FEM for parameter-errors in elliptic linear-quadratic parameter estimation problems

Sections 4.2–4.6 of this chapter are taken from:
R. Becker, M. Innerberger, and D. Praetorius. Adaptive FEM for parameter-errors in elliptic linear-quadratic parameter estimation problems, 2021. arXiv: 2111.03627

4.1 Introduction

While the last two chapters are concerned with analytical aspects of fundamental extensions to the setting from Chapter 1, the present chapter applies this setting to a practical example: parameter estimation for PDEs in a linear-quadratic setting (this refers to the equation and the optimization problem, respectively). Many models arising from applications depend on (a finite number of) parameters, which adjust the general model to a particular experimental situation, e.g., a specific material. For the sake of presentation, we focus here on a particular basic model problem,

$$-\operatorname{div}(\boldsymbol{A}\nabla u(\boldsymbol{p})) = f(\boldsymbol{p}) \quad \text{in } \Omega, \qquad u(\boldsymbol{p}) = 0 \quad \text{on } \partial\Omega, \tag{4.1}$$

where the right-hand side $f(\boldsymbol{p}) \in H^{-1}(\Omega)$ and, hence, also the solution $u(\boldsymbol{p})$ depend linearly on some finite dimensional parameter $\boldsymbol{p} \in Q \subseteq \mathbb{R}^{n_Q}$ for fixed $n_Q \in \mathbb{N}$. We suppose that set Q of admissible parameters is convex and closed, and print parameter vectors in boldface for better readability.

In practice, the parameter $\boldsymbol{p} \in Q$ is usually unknown and has to be determined by experiments. However, the individual parameters hardly ever correspond to independently and directly measurable physical quantities like the spatial dimensions of an object. Instead, the parameter can be inferred by indirect measurement: We suppose that some experimental measurements, modeled by a vector-valued measurement operator $\boldsymbol{G}\colon H^1_0(\Omega) \to C := \mathbb{R}^{n_C}$ for fixed $n_C \in \mathbb{N}$, are given to obtain exact simulated values $\boldsymbol{G}(u(\boldsymbol{p}^\star))$. The simulated values are then compared to experimentally obtained ones $\boldsymbol{G}^\star \in C$, and a parameter $\boldsymbol{p}^\star \in Q$ is chosen by a least squares computation. Simulation can be done by FEM on a mesh $\mathcal{T}_H$, which leads to an approximation $\boldsymbol{p}^\star_H \in Q$ of the real parameter $\boldsymbol{p}^\star$.

This problem allows for different viewpoints: First, since the unknown parameter $\boldsymbol{p}^\star$ is a quantity derived from the solution of a PDE (model problem (4.1)), it can be viewed as a *goal value*, on which a GOAFEM algorithm can be based. The goal error then is the parameter error $\|\boldsymbol{p}^\star - \boldsymbol{p}^\star_H\|_Q$, where we denote the Euclidean norm on Q by $\|\cdot\|_Q$ (analogous for C). Second, the parameter estimation problem can be viewed as a special case of an optimal control problem with finite dimensional control (the parameter $\boldsymbol{p}$). In this context, the solution $u(\boldsymbol{p})$ is called *state* and the solution $z(\boldsymbol{p})$ of a suitable dual problem introduced below, is called *co-state*. Thus, we use the ideas of existing literature on (GO)AFEM for optimal control problems.

There, adaptivity is usually driven by residual based *a posteriori* estimators for energy norms by means of the estimate ([BM11; GY17])

$$\|\boldsymbol{p}^\star - \boldsymbol{p}_H^\star\|_Q + |||u(\boldsymbol{p}^\star) - u_H(\boldsymbol{p}_H^\star)||| + |||z(\boldsymbol{p}^\star) - z_H(\boldsymbol{p}_H^\star)||| \lesssim \eta_H(u_H(\boldsymbol{p}_H^\star)) + \zeta_H(z_H(\boldsymbol{p}_H^\star)), \quad (4.2)$$

and optimality is shown with respect to the upper bound. However, the convergence rate of the (co-)state energy error $|||u(\boldsymbol{p}^\star) - u_H(\boldsymbol{p}_H^\star)||| + |||z(\boldsymbol{p}^\star) - z_H(\boldsymbol{p}_H^\star)|||$ is typically of lower order than that of the parameter error; see, e.g., Figure 4.2. Thus, the upper bound in (4.2) is not sharp. A partial remedy investigated by [GYZ16; LC17] employs L^2-norm error estimators in the spirit of [DS11]. This approach indeed leads to a sharp upper bound, but requires strong regularity assumptions to the co-state problem, thus essentially allowing only for convex domains. For a full remedy for the special case that is treated in this chaper, we completely replace (4.2).

Improved a priori estimates

For the GOAFEM presented in Chapter 1, the goal error estimate (1.21) is directly derived from the *a priori* estimate (1.17) by (A3). The first main part of this chapter is therefore dedicated to find a suitable *a priori* estimate for the parameter error $\|\boldsymbol{p}^\star - \boldsymbol{p}_H^\star\|$. To this end, we look into ideas from the works [BV04; BV05], which propose a GOAFEM algorithm where the convergence rate of the *a posteriori* estimators experimentally matches that of the parameter error. Since these works are concerned with dual-weighted residual error estimators, they lack a rigorous convergence analysis for their GOAFEM algorithms. However, they provide the key ideas to derive the desired *a priori* estimate; see Theorem 4.5 below. We note that instead of the sum structure of (4.2), the upper bound of this estimate has a product structure similar to (1.17):

$$\|\boldsymbol{p}^\star - \boldsymbol{p}\|_Q \lesssim \Big[\sum_{i=0}^{n_Q} |||u_i - u_{H,i}|||^2\Big]^{1/2} \Big[\sum_{j=1}^{n_C} \|z_j - z_{H,j}\|^2\Big]^{1/2} \quad (4.3)$$

where the terms on the right-hand side estimate the contributions of the separate components of parameters p_i and measurements $G_j^\star$.

Optimal GOAFEM for the parameter estimation problem

Once the *a priori* estimate is in place, we can again use (A3) to obtain an *a posteriori* estimate, which can, in turn, be used to drive a GOAFEM algorithm. However, estimate (4.3) has a non-standard product-of-sums structure. It is therefore not completely straight-forward to design a GOAFEM algorithm that can be shown to converge with optimal rates.

To remedy this, we turn to the idea of product (or combined) estimators used in Chapter 2 and generalize this concept to an arbitrary number of components. They are subsequently used to summarize all component estimators of the state and co-state sum, respectively, ending up with the well-known product structure (1.21). From there, optimality results follow analogously to Section 1.3.

Chapter outline

We begin by providing details of the problem formulation and its numerical solution by FEM in Section 4.2. This is cast into an adaptive algorithm in Section 4.3, where also our main results are

stated: Theorem 4.5 precisely formulates the *a priori* estimate (4.3) for the parameter error, which is an interesting result on its own; linear convergence with optimal algebraic rates of our GOAFEM is then stated in Theorems 4.13 and 4.14. The subsequent Sections 4.4 and 4.5 are dedicated to the proofs of our main results. To conclude, we underline our analysis with numerical experiments in Section 4.6.

Finally, we note that, in this chapter, for a function $J\colon \mathcal{H}_1 \to \mathcal{H}_2$ between Hilbert spaces $\mathcal{H}_1, \mathcal{H}_2$, we denote the gradient and Hessian of J at $v \in \mathcal{H}_1$ by $J'[v]\colon \mathcal{H}_1 \to \mathcal{H}_2$ and $J''[v]\colon \mathcal{H}_1 \times \mathcal{H}_1 \to \mathcal{H}_2$, respectively.

4.2 Parameter estimation problem

4.2.1 Problem formulation

We consider the linear elliptic PDE problem (4.1) in weak formulation: For $\boldsymbol{p} \in \mathcal{Q}$, find $u(\boldsymbol{p}) \in \mathcal{X} := H^1_0(\Omega)$ such that

$$a(u(\boldsymbol{p}), v) := \int_\Omega \boldsymbol{A}\nabla u(\boldsymbol{p}) \cdot \nabla v \,\mathrm{d}x = F_0(v) + b(\boldsymbol{p}, v) \quad \text{for all } v \in \mathcal{X}. \tag{4.4}$$

We suppose that $\boldsymbol{A}$ is a piecewise constant and positive definite matrix and that there exist $f_i \in L^2(\Omega)$ as well as $\boldsymbol{f}_i \in [L^2(\Omega)]^d$ for $i = 0, \dots, n_{\mathcal{Q}}$ such that

$$F_0(v) = \int_\Omega f_0 v - \boldsymbol{f}_0 \cdot \nabla v \,\mathrm{d}x \quad \text{and} \quad b(\boldsymbol{p}, v) = \sum_{i=1}^{n_{\mathcal{Q}}} p_i \int_\Omega f_i v - \boldsymbol{f}_i \cdot \nabla v \,\mathrm{d}x \quad \text{for all } v \in \mathcal{X}.$$

In particular, this implies that $b(\cdot,\cdot)\colon \mathcal{Q} \times \mathcal{X} \to \mathbb{R}$ is linear in both arguments and that, for all $\boldsymbol{p} \in \mathcal{Q}$, $F_0, b(\boldsymbol{p}, \cdot) \in \mathcal{X}' = H^{-1}(\Omega)$ are linear and continuous functionals on $\mathcal{X}$. Under these assumptions, the Lax–Milgram theory implies that problem (4.4) has a unique solution $u(\boldsymbol{p}) \in \mathcal{X}$, which is called *state (variable)*, for all parameters $\boldsymbol{p} \in \mathcal{Q}$.

Remark 4.1. *We note that our analysis below readily extends to more general problems, where*

$$a(u(\boldsymbol{p}), v) = \int_\Omega \boldsymbol{A}\nabla u(\boldsymbol{p}) \cdot \nabla v + \boldsymbol{b} \cdot \nabla u(\boldsymbol{p}) v + c\, u(\boldsymbol{p}) v \,\mathrm{d}x,$$

and to mixed (inhomogeneous) Dirichlet / Neumann boundary data. For such problems, the obvious adaptions are to be made for solution theory and error estimation.

We suppose that, for all $i = 1, \dots, n_C$, the components $G_i\colon \mathcal{X} \to \mathbb{R}$ of the measurement operator $\boldsymbol{G}\colon \mathcal{X} \to C$ take the form

$$G_i(v) = \int_\Omega g_i v - \boldsymbol{g}_i \cdot \nabla v \,\mathrm{d}x \quad \text{for all } v \in \mathcal{X},$$

for some given $g_i \in L^2(\Omega)$ and $\boldsymbol{g}_i \in [L^2(\Omega)]^d$.

We seek a parameter $\boldsymbol{p}^\star \in \mathcal{Q}$ such that the modeled measurements $\boldsymbol{G}(u(\boldsymbol{p}^\star))$ match the real measurements $\boldsymbol{G}^\star$ in some sense. To this end, we define the residual with respect to the parameter $\boldsymbol{p} \in \mathcal{Q}$ as

$$\boldsymbol{r}(\boldsymbol{p}) := \boldsymbol{G}(u(\boldsymbol{p})) - \boldsymbol{G}^\star \in C.$$

Allowing for an additional (regularization) constant $\alpha \geq 0$, we obtain the sought parameter as solution of the *parameter problem*

$$J(\boldsymbol{p}) := \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{p})\|_C^2 + \frac{\alpha}{2}\|\boldsymbol{p}\|_Q^2 \to \min \quad \text{in } Q, \tag{4.5}$$

with the (regularized) least-squares functional J. Note that J is quadratic due to linearity of the residual $\boldsymbol{r}$ and α can be chosen such that J is strictly convex; see (4.10) below. With such a choice of α, since Q is a closed and convex set, there exists a unique minimizer of (4.5), which we call $\boldsymbol{p}^\star$. In particular, the corresponding state $u(\boldsymbol{p}^\star) \in \mathcal{X}$ also exists and is unique in this case.

Remark 4.2. *While the linear-quadratic parameter estimation problem (4.4)–(4.5) is interesting on its own account, we note that such problems also appear as linearization of nonlinear parameter estimation problems, e.g., in the course of a Gauss–Newton iteration. Nonlinear parameter estimation problems in the context of adaptive iterative linearized algorithms as presented, e.g., in [HPW21] will be the subject of future work.*

4.2.2 Solution components

Due to the linearity of (4.4) with respect to the parameter $\boldsymbol{p}$, we can decompose the solution $u(\boldsymbol{p})$ into components. To this end, we denote by $\boldsymbol{e}_i \in \mathbb{R}^{n_Q}$ the i-th unit vector and define

$$\begin{aligned} u_0 \in \mathcal{X}: &\qquad a(u_0, v) = F_0(v) &&\text{for all } v \in \mathcal{X}, \\ u_i \in \mathcal{X}: &\qquad a(u_i, v) = b(\boldsymbol{e}_i, v) &&\text{for all } v \in \mathcal{X}, \text{ and all } i = 1, \dots, n_Q. \end{aligned} \tag{4.6}$$

Considering the solution $u(\boldsymbol{p})$ to (4.4) as a mapping $u\colon Q \to \mathcal{X}$, we can compute the derivative $u'\colon Q \to L(Q, \mathcal{X})$ with respect to the parameter to see that, for every $\boldsymbol{p}, \boldsymbol{q} \in Q$, the function $u'(\boldsymbol{p}) := u'[\boldsymbol{q}](\boldsymbol{p})$ is independent of the linearization point $\boldsymbol{q}$ and solves

$$a(u'(\boldsymbol{p}), v) = b(\boldsymbol{p}, v) \qquad \text{for all } v \in \mathcal{X}. \tag{4.7}$$

Due to linearity of $b(\cdot,\cdot)$ in both arguments and $\boldsymbol{p} = \sum_{i=1}^{n_Q} p_i \boldsymbol{e}_i$, we have that

$$u(\boldsymbol{p}) = u_0 + \sum_{i=1}^{n_Q} p_i u_i, \qquad u'(\boldsymbol{p}) = \sum_{i=1}^{n_Q} p_i u_i. \tag{4.8}$$

4.2.3 Least squares system and solution

By the assumptions on Q and quadraticity of $J\colon Q \to \mathbb{R}$, the problem (4.5) is a convex optimization problem. The first-order necessary condition for the minimizer $\boldsymbol{p}^\star \in Q$ of (4.5) is

$$J'[\boldsymbol{p}^\star](\boldsymbol{p} - \boldsymbol{p}^\star) \geq 0 \quad \text{for all } \boldsymbol{p} \in Q. \tag{4.9}$$

Since the least squares functional is quadratic, the Hessian $J'' \in \mathbb{R}^{n_Q \times n_Q}$ is constant and, in particular, independent of the evaluation point. We assume that there exists a constant $\kappa > 0$ such that the second-order sufficient condition

$$J''(\boldsymbol{p}, \boldsymbol{p}) \geq \kappa \|\boldsymbol{p}\|_Q^2 \quad \text{for all } \boldsymbol{p} \in Q \tag{4.10}$$

holds, i.e., that every solution to (4.9) is indeed a minimizer; see the general text [NW06] for existence and uniqueness of minimizers and Remark 4.3(ii) below for the role of κ.

From linearity of the measurement functional $\boldsymbol{G}\colon \mathcal{X} \to C$, we infer that $\boldsymbol{G}'[v](u'(\boldsymbol{q})) = \boldsymbol{G}(u'(\boldsymbol{q}))$ for all $v \in \mathcal{X}$ and $\boldsymbol{q} \in Q$. Thus, it holds that

$$\begin{aligned} J'[\boldsymbol{p}^\star](\boldsymbol{q}) &= \big\langle \boldsymbol{r}(\boldsymbol{p}^\star),\, \boldsymbol{r}'[\boldsymbol{p}^\star](\boldsymbol{q}) \big\rangle_C + \alpha \langle \boldsymbol{p}^\star,\, \boldsymbol{q} \rangle_Q \\ &= \big\langle \boldsymbol{G}(u(\boldsymbol{p}^\star)) - \boldsymbol{G}^\star,\, \boldsymbol{G}(u'(\boldsymbol{q})) \big\rangle_C + \alpha \langle \boldsymbol{p}^\star,\, \boldsymbol{q} \rangle_Q . \end{aligned} \tag{4.11}$$

Defining $\boldsymbol{B} \in \mathbb{R}^{n_Q \times n_C}$ by $\boldsymbol{B}_{ij} := G_j(u_i)$, we can use linearity in the last equation to obtain that

$$\begin{aligned} J'[\boldsymbol{p}^\star](\boldsymbol{q}) &= \sum_{i,j=1}^{n_Q} \sum_{k=1}^{n_C} \Big[\big(G_k(u_0) + p_i^\star G_k(u_i) - G_k^\star\big)\big(q_j G_k(u_j)\big) \Big] + \alpha \langle \boldsymbol{p}^\star,\, \boldsymbol{q} \rangle_Q \\ &= \boldsymbol{q}^\top (\boldsymbol{B}\boldsymbol{B}^\top + \alpha \boldsymbol{I}) \boldsymbol{p}^\star + \boldsymbol{q}^\top \boldsymbol{B}(\boldsymbol{G}(u_0) - \boldsymbol{G}^\star). \end{aligned} \tag{4.12}$$

From this representation, a solution to (4.9) can be computed, heeding the constraints induced by Q; see [NW06] for a comprehensive treatment of algorithms for such (possibly non-linear) numerical optimization problems.

Remark 4.3. (i) *For the unconstrained case $Q = \mathbb{R}^{n_Q}$, solving the optimality condition (4.9) simplifies to solving the (linear) least-squares system*

$$(\boldsymbol{B}\boldsymbol{B}^\top + \alpha \boldsymbol{I}) \boldsymbol{p}^\star = \boldsymbol{B}(\boldsymbol{G}^\star - \boldsymbol{G}(u_0)).$$

(ii) *From the explicit representation*

$$J''(\boldsymbol{p}, \boldsymbol{p}) = \boldsymbol{p}^\top \big(\boldsymbol{B}\boldsymbol{B}^\top + \alpha \boldsymbol{I}\big) \boldsymbol{p}, \tag{4.13}$$

one infers two sufficient conditions such that (4.10) holds: The first one is that the system matrix $\boldsymbol{B}\boldsymbol{B}^\top$ has full rank, in which case $\alpha = 0$ is an admissible choice. In particular, this requires $n_C \geq n_Q$, i.e., more measurements than parameters. The second condition is that $\alpha > 0$.

4.2.4 FEM discretization

For a conforming triangulation $\mathcal{T}_H$ of $\Omega \subset \mathbb{R}^d$ into compact simplices and a polynomial degree $k \geq 1$, let

$$\mathcal{X}_H := \{ v_H \in H_0^1(\Omega) \,|\, \forall T \in \mathcal{T}_H \colon \quad v_H|_T \text{ is a polynomial of degree } \leq k \}.$$

To obtain a conforming finite element approximation $u(\boldsymbol{p}) \approx u_H(\boldsymbol{p}) \in \mathcal{X}_H$ for $\boldsymbol{p} \in Q$, we consider the Galerkin discretization of (4.4), which reads: Find $u_H(\boldsymbol{p})$ such that

$$a(u_H(\boldsymbol{p}), v_H) = F_0(v_H) + b(\boldsymbol{p}, v_H) \qquad \text{for all } v_H \in \mathcal{X}_H .$$

Moreover, the functions $u_{H,0}, u_{H,i} \in \mathcal{X}_H$ are defined analogously to (4.6), which is why (4.8) holds accordingly.

On $\mathcal{T}_H$, an approximation of the continuous parameter $\boldsymbol{p}^\star$ can be obtained by minimizing a discretized version of the least-squares functional (4.5):

$$J_H(\boldsymbol{p}) := \frac{1}{2} \|\boldsymbol{r}_H(\boldsymbol{p})\|_C^2 + \frac{\alpha}{2} \|\boldsymbol{p}\|_Q^2 \quad \text{with} \quad \boldsymbol{r}_H(\boldsymbol{p}) := \boldsymbol{G}(u_H(\boldsymbol{p})) - \boldsymbol{G}^\star. \tag{4.14}$$

The minimizer $\boldsymbol{p}_H^\star \in Q$ of the discrete least-squares functional satisfies

$$J_H'[\boldsymbol{p}_H^\star](\boldsymbol{p} - \boldsymbol{p}_H^\star) \geq 0 \quad \text{for all } \boldsymbol{p} \in Q. \tag{4.15}$$

We note that a discrete representation of (4.14) can be derived in complete analogy to (4.12), with $\boldsymbol{B}_{H,ij} := G_j(u_{H,i})$.

4.2.5 Co-state components

In the following adaptive algorithm and its analysis we also need information about the measurement operators. To this end, we introduce the *co-state components*

$$z_j \in \mathcal{X}: \qquad a(v, z_j) = G_j(v) \qquad \text{for all } v \in \mathcal{X},\ j = 1, \dots, n_C \tag{4.16}$$

and their discretizations $z_{H,j} \in \mathcal{X}_H$.

To give a concise presentation of our analysis, we further define for $\boldsymbol{p} \in Q$ the functions

$$z(\boldsymbol{p}) := \sum_{j=1}^{n_C} r_{H,j}(\boldsymbol{p}) z_j \quad \text{and} \quad z_H(\boldsymbol{p}) := \sum_{j=1}^{n_C} r_{H,j}(\boldsymbol{p}) z_{H,j}$$

and note that they satisfy

$$\begin{aligned} a(v, z(\boldsymbol{p})) &= \big\langle \boldsymbol{r}_H(\boldsymbol{p}),\, \boldsymbol{G}(v) \big\rangle_C \quad \text{for all } v \in \mathcal{X}, \\ a(v_H, z_H(\boldsymbol{p})) &= \big\langle \boldsymbol{r}_H(\boldsymbol{p}),\, \boldsymbol{G}(v_H) \big\rangle_C \quad \text{for all } v_H \in \mathcal{X}_H. \end{aligned} \tag{4.17}$$

Taking the derivative of the last equations with respect to the parameter, we obtain

$$\begin{aligned} a(v, z'(\boldsymbol{p})) &= \big\langle \boldsymbol{G}(u_H'(\boldsymbol{p})),\, \boldsymbol{G}(v) \big\rangle_C \quad \text{for all } v \in \mathcal{X}, \\ a(v_H, z_H'(\boldsymbol{p})) &= \big\langle \boldsymbol{G}(u_H'(\boldsymbol{p})),\, \boldsymbol{G}(v_H) \big\rangle_C \quad \text{for all } v_H \in \mathcal{X}_H \end{aligned} \tag{4.18}$$

and note that the following identities hold:

$$z_H'(\boldsymbol{p}) = \sum_{j=1}^{n_C} G_j\big(u_H'(\boldsymbol{p})\big) z_j \quad \text{and} \quad z_H'(\boldsymbol{p}) = \sum_{j=1}^{n_C} G_j\big(u_H'(\boldsymbol{p})\big) z_{H,j}. \tag{4.19}$$

Remark 4.4. *By considering the co-state components $z_{H,j}$ from (4.16), the matrix entries $\boldsymbol{B}_{H,ij}$ can be computed by the n_Q state components $u_{H,i}$, or the n_C co-state components $z_{H,j}$ via*

$$\boldsymbol{B}_{H,ij} = G_j(u_{H,i}) \overset{(4.16)}{=} a(u_{H,i}, z_j) = a(u_{H,i}, z_{H,j}) \overset{(4.6)}{=} b(\boldsymbol{e}_i, z_{H,j}). \tag{4.20}$$

Thus, for assembling $\boldsymbol{B}_H$, one can decide between computing state or co-state components. For our adaptive algorithm below, however, we need the state as well as the co-state components anyway to compute the necessary a posteriori estimators.

4.3 Adaptive algorithm and main results

4.3.1 A priori estimate

Our first main result is an *a priori* estimate for the parameter error. To the best of our knowledge, this is a novel result. Its proof is given in Section 4.4 below.

Theorem 4.5

There exists a constant $C_Q > 0$ such that

$$\|\boldsymbol{p}^\star - \boldsymbol{p}_H^\star\|_Q \le C_Q \Big[\sum_{i=0}^{n_Q} |||u_i - u_{H,i}|||^2\Big]^{1/2} \Big[\sum_{j=1}^{n_C} \|z_j - z_{H,j}\|^2\Big]^{1/2} \quad \text{for all } \mathcal{T}_H \in \mathbb{T}. \tag{4.21}$$

The constant C_Q depends only on Q, Ω, $\boldsymbol{G}^\star$, κ, α, and the data $\boldsymbol{A}$, f_i, g_j, $\boldsymbol{f}_i$, and $\boldsymbol{g}_j$.

Remark 4.6. *The proof of the estimate (4.21) relies heavily on the linear dependence of the primal and dual solution on the parameter, as well as the finite dimension of Q. Problems which depend on the parameter in a nonlinear fashion are usually solved iteratively, such that the linearization in each step again depends linearly on the parameter. Thus, a result analogous to Theorem 4.5 might also hold in this case. For optimal control problems with $\dim Q = \infty$, however, it is not clear how to replace the components u_i and z_j as well as their discretizations in (4.21).*

The estimate (4.21) is the fundamental result that allows to design the adaptive algorithm that is presented in the following sections.

4.3.2 Mesh refinement

Let $\mathcal{T}_0$ be a given conforming triangulation of Ω which is admissible for $d \ge 3$ in the sense of [Ste08]. For mesh refinement, we employ newest vertex bisection; see [BDD04; KPP13; Ste08]. For each conforming triangulation $\mathcal{T}_H$ and marked elements $\mathcal{M}_H \subseteq \mathcal{T}_H$, let $\mathcal{T}_h := \texttt{refine}(\mathcal{T}_H, \mathcal{M}_H)$ be the coarsest conforming triangulation where all $T \in \mathcal{M}_H$ have been refined, i.e., $\mathcal{M}_H \subseteq \mathcal{T}_H \backslash \mathcal{T}_h$. We write $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, if $\mathcal{T}_h$ results from $\mathcal{T}_H$ by finitely many steps of refinement. To abbreviate notation, let $\mathbb{T} := \mathbb{T}(\mathcal{T}_0)$. We note that there holds nestedness of finite element spaces, i.e., $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$ implies that $\mathcal{X}_H \subseteq \mathcal{X}_h$.

We note that newest vertex bisection generates a family of shape-regular meshes, i.e., there exists a constant $\gamma > 0$ such that

$$\sup_{\mathcal{T}\in\mathbb{T}} \max_{T\in\mathcal{T}} \frac{\operatorname{diam}(T)^d}{|T|} \le \gamma < \infty.$$

4.3.3 A posteriori error estimation

We consider standard residual error estimators, i.e., for $\mathcal{T}_H \in \mathbb{T}$, $T \in \mathcal{T}_H$, and $v_H \in \mathcal{X}_H$ we define

$$\eta_{H,i}(T, v_H)^2 := h_T^2 \big\| f_i + \operatorname{div}(\boldsymbol{f}_i + \boldsymbol{A}\nabla v_H)\big\|_{L^2(T)}^2 + h_T \big\| [\![(\boldsymbol{f}_i + \boldsymbol{A}\nabla v_H) \cdot \boldsymbol{n}]\!] \big\|_{L^2(\partial T\cap\Omega)}^2,$$
$$\zeta_{H,j}(T, v_H)^2 := h_T^2 \big\| g_j + \operatorname{div}(\boldsymbol{g}_j + \boldsymbol{A}\nabla v_H)\big\|_{L^2(T)}^2 + h_T \big\| [\![(\boldsymbol{g}_j + \boldsymbol{A}\nabla v_H) \cdot \boldsymbol{n}]\!] \big\|_{L^2(\partial T\cap\Omega)}^2,$$

for all $0 \le i \le n_Q$ and $1 \le j \le n_C$, where $h_T := |T|^{1/d}$ and $[\![\cdot]\!]$ is the jump across element boundaries. For a subset $\mathcal{U}_H \subseteq \mathcal{T}_H$, we further define

$$\eta_{H,i}(\mathcal{U}_H, v_H)^2 := \sum_{T\in\mathcal{U}_H} \eta_{H,i}(T, v_H)^2, \qquad \zeta_{H,j}(\mathcal{U}_H, v_H)^2 := \sum_{T\in\mathcal{U}_H} \zeta_{H,j}(T, v_H)^2, \tag{4.22}$$

and we abbreviate

$$\begin{aligned} &\eta_{H,i}(v_H) := \eta_{H,i}(\mathcal{T}_H, v_H), && \eta_{H,i}(\mathcal{U}_H) := \eta_{H,i}(\mathcal{U}_H, u_{H,i}), && \eta_{H,i} := \eta_{H,i}(\mathcal{T}_H), \\ &\zeta_{H,j}(v_H) := \zeta_{H,j}(\mathcal{T}_H, v_H), && \zeta_{H,j}(\mathcal{U}_H) := \zeta_{H,j}(\mathcal{U}_H, z_{H,j}), && \zeta_{H,j} := \zeta_{H,j}(\mathcal{T}_H). \end{aligned} \tag{4.23}$$

It is well-known that, in our setting, these estimators satisfy the so-called *axioms of adaptivity*; see, e.g., [CFPP14].

Proposition 4.7. *There exist constants $C_{\mathrm{stab}}, C_{\mathrm{rel}}, C_{\mathrm{drel}} > 0$ and $0 < q_{\mathrm{red}} < 1$ such that for all $\mathcal{T}_H \in \mathbb{T}(\mathcal{T}_0)$, $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, $0 \le i \le n_Q$, and $1 \le j \le n_C$ the residual error estimators satisfy the following properties:*

(A1) Stability: For all $v_h \in \mathcal{X}_h$, $v_H \in \mathcal{X}_H$, and $\mathcal{U}_H \subseteq \mathcal{T}_h \cap \mathcal{T}_H$, it holds that

$$\big|\eta_{h,i}(\mathcal{U}_H, v_h) - \eta_{H,i}(\mathcal{U}_H, v_H)\big| + \big|\zeta_{h,j}(\mathcal{U}_H, v_h) - \zeta_{H,j}(\mathcal{U}_H, v_H)\big| \le C_{\mathrm{stab}} \,|\!|\!| v_h - v_H |\!|\!|.$$

(A2) Reduction: For all $v_H \in \mathcal{X}_H$, it holds that

$$\eta_{h,i}(\mathcal{T}_h \backslash \mathcal{T}_H, v_H) \le q_{\mathrm{red}}\, \eta_{H,i}(\mathcal{T}_H \backslash \mathcal{T}_h, v_H) \quad \text{and} \quad \zeta_{h,j}(\mathcal{T}_h \backslash \mathcal{T}_H, v_H) \le q_{\mathrm{red}}\, \zeta_{H,j}(\mathcal{T}_H \backslash \mathcal{T}_h, v_H).$$

(A3) Reliability: The state and co-state components $u_{H,i}, z_{H,j} \in \mathcal{X}_H$ satisfy that

$$|\!|\!| u_i - u_{H,i} |\!|\!| \le C_{\mathrm{rel}}\, \eta_{H,i} \quad \text{and} \quad |\!|\!| z_j - z_{H,j} |\!|\!| \le C_{\mathrm{rel}}\, \zeta_{H,j}.$$

(A4) Discrete reliability: The components $u_{H,i}, z_{H,j} \in \mathcal{X}_H$ and $u_{h,i}, z_{h,j} \in \mathcal{X}_h$ satisfy

$$|\!|\!| u_{h,i} - u_{H,i} |\!|\!| \le C_{\mathrm{drel}}\, \eta_{H,i}(\mathcal{T}_H \backslash \mathcal{T}_h) \quad \text{and} \quad |\!|\!| z_{h,j} - z_{H,j} |\!|\!| \le C_{\mathrm{drel}}\, \zeta_{H,j}(\mathcal{T}_H \backslash \mathcal{T}_h). \qquad \square$$

Together with our *a priori* estimate Theorem 4.5, reliability (A3) immediately implies that a suitable combination of error estimators of the components is indeed an upper bound to the parameter error.

Theorem 4.8

With the constant $C_Q > 0$ from Theorem 4.5, there holds

$$\|\boldsymbol{p}^\star - \boldsymbol{p}_H^\star\|_Q \le C_{\mathrm{rel}}^2\, C_Q \Big[\sum_{i=0}^{n_Q} \eta_{H,i}^2\Big]^{1/2} \Big[\sum_{j=1}^{n_C} \zeta_{H,j}^2\Big]^{1/2} \quad \text{for all } \mathcal{T}_H \in \mathbb{T}. \tag{4.24}$$

Remark 4.9. *The usual (residual) a posteriori estimate for optimal control problems in the literature is an upper bound for the sum*

$$\|\boldsymbol{p}^\star - \boldsymbol{p}_H^\star\|_Q + |\!|\!| u(\boldsymbol{p}_H^\star) - u_H(\boldsymbol{p}_H^\star) |\!|\!| + |\!|\!| z(\boldsymbol{p}_H^\star) - z_H(\boldsymbol{p}_H^\star) |\!|\!|.$$

Since the error in the parameter can be expected to be of higher order than that of the state and the co-state variable, such an estimate is sub-optimal with respect to the parameter error. Our estimate in (4.42) neatly exploits the finite dimension of Q to gain two advantages:

- *it shows an improved rate of convergence compared to the usual estimator for the sum above (see the numerical experiments in Section 4.6);*
- *it does not use $\boldsymbol{p}_H^\star$, so that its (possibly costly) computation can be avoided, if one is only interested in an a posteriori estimate. For instance, one can improve the mesh by use of Algorithm 4A below until the upper bound in (4.24) is sufficiently small. Then, $\boldsymbol{p}_H^\star$ is only computed once for the final mesh.*

4.3.4 Adaptive algorithm

Once Theorem 4.8 provides an error estimator for the parameter error on a mesh $\mathcal{T}_H \in \mathbb{T}$ involving only (co-)state components, we can define weighted refinement indicators in the spirit of [BET11]:

$$\varrho_H(T)^2 := \Big[\sum_{i=0}^{n_Q} \eta_{H,i}^2\Big]\Big[\sum_{j=1}^{n_C} \zeta_{H,j}(T)^2\Big] + \Big[\sum_{i=0}^{n_Q} \eta_{H,i}(T)^2\Big]\Big[\sum_{j=1}^{n_C} \zeta_{H,j}^2\Big]. \tag{4.25}$$

For a subset $\mathcal{U}_H \subseteq \mathcal{T}_H$, we again define $\varrho_H(\mathcal{U}_H)^2 := \sum_{T\in\mathcal{U}_H} \varrho_H(T)^2$ and $\varrho_H := \varrho_H(\mathcal{T}_H)$.

This allows us to devise an adaptive algorithm for the parameter estimation problem.

Algorithm 4A

Input: Initial triangulation $\mathcal{T}_0$, marking parameter $\theta \in (0, 1]$.
For $\ell = 0, 1, 2, \ldots$, do

(i) Compute $u_{\ell,i}$ and $z_{\ell,j}$ for $i = 0, \ldots, n_Q$ and $j = 1, \ldots, n_C$.

(ii) Compute refinement indicators $\varrho_\ell(T)$.

(iii) Find a minimal set $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ of elements such that $\varrho_\ell(\mathcal{M}_\ell)^2 \geq \theta\varrho_\ell^2$.

(iv) Compute $\mathcal{T}_{\ell+1} := \texttt{refine}(\mathcal{T}_\ell, \mathcal{M}_\ell)$.

Output: Sequence of triangulations $(\mathcal{T}_\ell)_\ell$, components $(u_{\ell,i})_\ell, (z_{\ell,j})_\ell$, and corresponding error estimators.

Remark 4.10. *We note that the proposed weighted error estimator (4.25) satisfies*

$$\varrho_H^2 = 2\Big[\sum_{i=0}^{n_Q} \eta_{H,i}^2\Big]\Big[\sum_{j=1}^{n_C} \zeta_{H,j}^2\Big], \tag{4.26}$$

which is essentially the square of the upper bound of the parameter error (4.24) from Theorem 4.8. Thus, by proving convergence (with optimal rates) of Algorithm 4A with respect to the weighted estimator ϱ_H, we can draw the same conclusion for the parameter error.

Remark 4.11. *Note that computing the parameter $\boldsymbol{p}_H^\star$ from (4.15) can involve high computational effort, depending on the precise constraints imposed by $\mathcal{Q}$. For this reason, we stress that Algorithm 4A relies only on the unweighted (co-)state components $u_{H,i}$ and $z_{H,j}$ for error estimation and refinement in each step. Therefore, $\boldsymbol{p}_H^\star$ will only be computed once, namely when the upper bound in (4.24) for $\mathcal{T}_H = \mathcal{T}_\ell$ is sufficiently small and, hence, Algorithm 4A is terminated for this step ℓ.*

Remark 4.12. *Instead of the weighted marking strategy in Algorithm 4A*(iii)*, the marking strategies from [FPZ16] and the seminal work [MS09] can also be used in our analysis below. Both strategies first find minimal sets* $\mathcal{M}_\ell^u$ *and* $\mathcal{M}_\ell^z$ *that satisfy*

$$\theta \sum_{i=0}^{n_Q} \eta_{\ell,i}^2 \le \sum_{i=0}^{n_Q} \eta_{\ell,i}(\mathcal{M}_\ell^u)^2 \quad \text{and} \quad \theta \sum_{j=1}^{n_C} \zeta_{\ell,j}^2 \le \sum_{j=1}^{n_C} \zeta_{\ell,j}(\mathcal{M}_\ell^z)^2,$$

respectively. The set of marked elements is then defined as suitable subset $\mathcal{M}_\ell \subseteq \mathcal{M}_\ell^u \cup \mathcal{M}_\ell^z$ *with* $\#\mathcal{M}_\ell \le C_{\mathrm{mark}} \min\{\#\mathcal{M}_\ell^u, \#\mathcal{M}_\ell^z\}$*, where* $C_{\mathrm{mark}} = 1$ *in [MS09] and* $C_{\mathrm{mark}} = 2$ *in [FPZ16]. In particular, the minimality assumption to* $\mathcal{M}_\ell$*,* $\mathcal{M}_\ell^u$*, and* $\mathcal{M}_\ell^z$ *can be weakened to only hold up to an arbitrary but fixed factor greater than* 1*; see [FPZ16].*

4.3.5 Convergence of Algorithm 4A

Our second main result concerns linear convergence of the error estimator.

Theorem 4.13

Suppose $0 < \theta \le 1$*. Then, Algorithm 4A satisfies linear convergence*

$$\varrho_{\ell+n} \le C_{\mathrm{lin}} q_{\mathrm{lin}}^n \varrho_\ell \quad \text{for all } \ell, n \in \mathbb{N}_0. \tag{4.27}$$

The constants $C_{\mathrm{lin}} > 0$ *and* $0 < q_{\mathrm{lin}} < 1$ *depend only on* C_{stab}*,* q_{red}*,* C_{rel}*, and the (arbitrary) adaptivity parameter* $0 < \theta \le 1$*.*

For our last main result, linear convergence of the error estimator ϱ_ℓ with optimal algebraic rates, we introduce so-called approximation classes. Given $N \in \mathbb{N}_0$, let $\mathbb{T}(N)$ be the set of all $\mathcal{T}_H \in \mathbb{T}$ with $\#\mathcal{T}_H - \#\mathcal{T}_0 \le N$. For all $r > 0$, we define

$$\begin{aligned} \|u_i\|_{\mathbb{A}_r} &:= \sup_{N \in \mathbb{N}_0} (N+1)^r \min_{\mathcal{T}_{\mathrm{opt}} \in \mathbb{T}(N)} \eta_{\mathrm{opt},i}(u_{\mathrm{opt},i}) \in [0,\infty], \quad \text{for } 0 \le i \le n_Q, \\ \|z_j\|_{\mathbb{A}_r} &:= \sup_{N \in \mathbb{N}_0} (N+1)^r \min_{\mathcal{T}_{\mathrm{opt}} \in \mathbb{T}(N)} \zeta_{\mathrm{opt},j}(z_{\mathrm{opt},j}) \in [0,\infty], \quad \text{for } 1 \le j \le n_C. \end{aligned} \tag{4.28}$$

By definition, e.g., $\|u_i\|_{\mathbb{A}_r} < \infty$ yields that $\eta_{\mathrm{opt},i}(u_{\mathrm{opt},i})$ decays at least with algebraic rate $r > 0$ along a sequence of optimal meshes. The following theorem states that any possible overall rate for the upper bound of (4.24) will indeed be realized by Algorithm 4A.

Theorem 4.14

Let $0 < \theta < \theta_{\mathrm{opt}} := (1 + C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2)^{-1}$*. Let* $s_i, t_j > 0$ *with*

$$\sum_{i=0}^{n_Q} \|u_i\|_{\mathbb{A}_{s_i}}^2 + \sum_{j=1}^{n_C} \|z_j\|_{\mathbb{A}_{t_j}}^2 < \infty.$$

Then, there exists a constant $C_{\mathrm{opt}} > 0$ such that for all $\ell \in \mathbb{N}_0$ there holds that

$$\varrho_\ell \le C_{\mathrm{opt}} \Big[\sum_{i=0}^{n_Q} \|u_i\|^2_{\mathbb{A}_{s_i}} \Big]^{1/2} \Big[\sum_{j=1}^{n_C} \|z_j\|^2_{\mathbb{A}_{t_j}} \Big]^{1/2} \big(\#\mathcal{T}_\ell - \#\mathcal{T}_0\big)^{-\beta}, \tag{4.29}$$

where $\beta := \min\{s_i \,|\, 0 \le i \le n_Q\} + \min\{t_j \,|\, 1 \le j \le n_C\}$. The constant C_{opt} depends only on C_{cls}, C_{stab}, q_{red}, C_{rel}, C_{drel}, θ, n_Q, n_C, s_i, and t_j.

Remark 4.15. *Note that our adaptive algorithm drives down only the upper bound of the parameter error with optimal rates and not the parameter error itself. However, in general, one cannot expect to obtain a rate higher than $\min\{s_i \,|\, 0 \le i \le n_Q\} + \min\{t_j \,|\, 1 \le j \le n_C\}$ for the parameter error. This is due to the fact that the assembly of the matrix entries $\boldsymbol{B}_{\ell,ij}$, which are needed to compute $\boldsymbol{p}^\star_\ell$, uses all state or co-state components and thus its accuracy is constricted by their respective minimal rate.*

Remark 4.16. *Theorem 4.14 holds true for all mesh refinement strategies as long as there hold the* son estimate

$$\#(\mathcal{T}_H \backslash \mathcal{T}_h) + \#\mathcal{T}_H \le \#\mathcal{T}_h \quad \text{for all } \mathcal{T}_H \in \mathbb{T} \text{ and all } \mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H), \tag{4.30}$$

the overlay estimate

$$\#(\mathcal{T}_H \oplus \mathcal{T}_h) \le \#\mathcal{T}_H + \#\mathcal{T}_h - \#\mathcal{T}_0, \quad \text{for all } \mathcal{T}_H \in \mathbb{T} \text{ and all } \mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H), \tag{4.31}$$

and the closure estimate *(with constant $C_{\mathrm{cls}} > 0$)*

$$\#\mathcal{T}_\ell - \#\mathcal{T}_0 \le C_{\mathrm{cls}} \sum_{j=0}^{\ell-1} \#\mathcal{M}_j \quad \text{for all } \ell \in \mathbb{N}, \tag{4.32}$$

as well as the axioms (A1)–(A4); see [CFPP14]. In this sense, the present analysis is indeed independent of newest vertex bisection.

4.4 Proof of Theorem 4.5

4.4.1 Auxiliary a priori bounds

We start by stating some well-known *a priori* estimates, which are used throughout the subsequent analysis.

Lemma 4.17. *There exists a constant $C_{\boldsymbol{A}} > 0$ such that, for all $\mathcal{T}_H \in \mathbb{T}$,*

$$\begin{aligned} |||u_{H,i}||| \le |||u_i||| &\le C_{\boldsymbol{A}} \big[\|f_i\|_{L^2(\Omega)} + \|\boldsymbol{f}_i\|_{L^2(\Omega)}\big] \quad \text{for all } 0 \le i \le n_Q, \\ |||z_{H,j}||| \le |||z_j||| &\le C_{\boldsymbol{A}} \big[\|g_j\|_{L^2(\Omega)} + \|\boldsymbol{g}_j\|_{L^2(\Omega)}\big] \quad \text{for all } 1 \le j \le n_C. \end{aligned} \tag{4.33}$$

The constant $C_{\boldsymbol{A}}$ depends only on $\boldsymbol{A}$ and Ω. □

Since $u'(\boldsymbol{p})$, $z'(\boldsymbol{p})$, and their discrete counterparts depend linearly on the parameter, we can use the *a priori* bounds to split parameters and component errors.

Lemma 4.18. *There exists a constant $C > 0$ such that, for all $\boldsymbol{p} \in Q$,*

$$\begin{aligned} |||u'(\boldsymbol{p}) - u'_H(\boldsymbol{p})||| &\leq \|\boldsymbol{p}\|_Q \Big[\sum_{i=1}^{n_Q} |||u_i - u_{H,i}|||^2 \Big]^{1/2}, \\ |||z'(\boldsymbol{p}) - z'_H(\boldsymbol{p})||| &\leq C\, \|\boldsymbol{p}\|_Q \Big[\sum_{j=1}^{n_C} |||z_j - z_{H,j}|||^2 \Big]^{1/2}. \end{aligned} \tag{4.34}$$

The constant C depends only on Ω as well as the data $\boldsymbol{A}$, f_i, g_j, $\boldsymbol{f}_i$, and $\boldsymbol{g}_j$.

Proof. Let $\boldsymbol{p} \in Q$. With the representation (4.8), the triangle inequality, and the Cauchy–Schwarz inequality, we obtain the first inequality of (4.34) by

$$|||u'(\boldsymbol{p}) - u'_H(\boldsymbol{p})||| \overset{(4.8)}{\leq} \sum_{i=1}^{n_Q} |\boldsymbol{p}_i| |||u_i - u_{H,i}||| \leq \|\boldsymbol{p}\|_Q \Big[\sum_{i=1}^{n_Q} |||u_i - u_{H,i}|||^2 \Big]^{1/2}.$$

The same arguments, together with a priori bounds for $u_{H,i}$, can be used to show that

$$|||u'_H(\boldsymbol{p})||| \overset{(4.8)}{\leq} \|\boldsymbol{p}\|_Q \Big[\sum_{i=1}^{n_Q} |||u_{H,i}|||^2 \Big]^{1/2} \overset{(4.33)}{\leq} \|\boldsymbol{p}\|_Q C_A \Big[\sum_{i=1}^{n_Q} \big(\|f_i\|_{L^2(\Omega)} + \|\boldsymbol{f}_i\|_{L^2(\Omega)} \big)^2 \Big]^{1/2}. \tag{4.35}$$

For the second inequality in (4.34), we can again employ the triangle inequality. Together with continuity of the measurement functionals G_j, which depend only on g_j and $\boldsymbol{g}_j$, and (4.35) this leads to

$$\begin{aligned} |||z'(\boldsymbol{p}) - z'_H(\boldsymbol{p})||| &\overset{(4.19)}{\leq} \sum_{j=1}^{n_C} \big| G_j\big(u'_H(\boldsymbol{p})\big) \big|\, |||z_j - z_{H,j}||| \\ &\leq \big\| \boldsymbol{G}\big(u'_H(\boldsymbol{p})\big) \big\|_C \Big[\sum_{j=1}^{n_C} |||z_j - z_{H,j}|||^2 \Big]^{1/2} \overset{(4.35)}{\lesssim} \|\boldsymbol{p}\|_Q \Big[\sum_{j=1}^{n_C} |||z_j - z_{H,j}|||^2 \Big]^{1/2}. \end{aligned}$$

This concludes the proof. ||| □

Lemma 4.19. *There exists a constant $C_\star > 0$ such that, for all $\mathcal{T}_H \in \mathbb{T}$,*

$$\|\boldsymbol{p}^\star\|_Q \leq C_\star \quad \text{and} \quad \|\boldsymbol{p}^\star_H\|_Q \leq C_\star. \tag{4.36}$$

The constant $C_\star$ depends only on α, $\boldsymbol{G}^\star$, Ω, κ, $\min\{\|\boldsymbol{q}\|_Q \,|\, \boldsymbol{q} \in Q\}$, and the data A, f_i, g_j, $\boldsymbol{f}_i$, and $\boldsymbol{g}_j$.

Proof. We first define $C := \min\{\|\boldsymbol{q}\|_Q \,|\, \boldsymbol{q} \in Q\}$ and choose $\boldsymbol{p} \in Q$ such that $\|\boldsymbol{p}\|_Q = C$ (such a choice exists since Q is closed and convex). From the first and second order optimality condition, (4.9) and (4.10), and the explicit form of J' in (4.12), we have that

$$\begin{aligned} \kappa \|\boldsymbol{p}^\star\|_Q^2 &\overset{(4.10)}{\leq} J''(\boldsymbol{p}^\star, \boldsymbol{p}^\star) \overset{(4.13)}{=} (\boldsymbol{p}^\star)^\intercal \big(\boldsymbol{B}\boldsymbol{B}^\intercal + \alpha \boldsymbol{I} \big) \boldsymbol{p}^\star \overset{(4.12)}{=} J'[\boldsymbol{p}^\star](\boldsymbol{p}^\star) - (\boldsymbol{p}^\star)^\intercal \boldsymbol{B}\big(\boldsymbol{G}(u_0) - \boldsymbol{G}^\star \big) \\ &\overset{(4.9)}{\leq} J'[\boldsymbol{p}^\star](\boldsymbol{p}) - (\boldsymbol{p}^\star)^\intercal \boldsymbol{B}\big(\boldsymbol{G}(u_0) - \boldsymbol{G}^\star \big) \\ &\overset{(4.12)}{=} (\boldsymbol{p}^\star)^\intercal \big(\boldsymbol{B}\boldsymbol{B}^\intercal + \alpha \boldsymbol{I} \big) \boldsymbol{p} + (\boldsymbol{p} - \boldsymbol{p}^\star)^\intercal \boldsymbol{B}\big(\boldsymbol{G}(u_0) - \boldsymbol{G}^\star \big). \end{aligned}$$

We denote by $\|\cdot\|_{L(Q)}$ the natural matrix norm induced by the Euclidean norm $\|\cdot\|_Q$, i.e., the spectral norm. Using the Cauchy–Schwarz inequality together with $\|\boldsymbol{p}\|_Q = C$, we see that

$$\kappa \|\boldsymbol{p}^\star\|_Q^2 \leq \|\boldsymbol{p}^\star\|_Q \big(C \|\boldsymbol{B}\boldsymbol{B}^\intercal + \alpha \boldsymbol{I}\|_{L(Q)} + \|\boldsymbol{B}(\boldsymbol{G}(u_0) - \boldsymbol{G}^\star)\|_Q\big) + C \|\boldsymbol{B}(\boldsymbol{G}(u_0) - \boldsymbol{G}^\star)\|_Q.$$

In the case $\|\boldsymbol{p}^\star\|_Q < 1$, there already holds the first inequality of (4.36) with $C_\star = 1$. In the case $\|\boldsymbol{p}^\star\|_Q \geq 1$, we can divide by $\kappa\|\boldsymbol{p}^\star\| \geq \kappa > 0$ and further estimate the last inequality by

$$\|\boldsymbol{p}^\star\|_Q \leq \kappa^{-1}\big(C \|\boldsymbol{B}\boldsymbol{B}^\intercal + \alpha \boldsymbol{I}\|_{L(Q)} + (1 + C)\|\boldsymbol{B}(\boldsymbol{G}(u_0) + \boldsymbol{G}^\star)\|_Q\big). \tag{4.37}$$

From the definition of $\boldsymbol{B}$, we have that, for all $1 \leq i \leq n_Q$ and $1 \leq j \leq n_C$,

$$|\boldsymbol{B}_{ij}| = |G_j(u_i)| = a(u_i, z_j) \overset{(4.33)}{\leq} C_{\boldsymbol{A}}^2 \big[\|f_i\|_{L^2(\Omega)} + \|\boldsymbol{f}_i\|_{L^2(\Omega)}\big]\big[\|g_j\|_{L^2(\Omega)} + \|\boldsymbol{g}_j\|_{L^2(\Omega)}\big].$$

We can thus estimate the Frobenius-norm $\|\cdot\|_F$ of B by

$$\|B\|_F^2 = \sum_{i=1}^{n_Q}\sum_{j=1}^{n_C} |B_{ij}|^2 \leq C_{\boldsymbol{A}}^4 \sum_{i=1}^{n_Q}\sum_{j=1}^{n_C} \big[\|f_i\|_{L^2(\Omega)} + \|\boldsymbol{f}_i\|_{L^2(\Omega)}\big]^2\big[\|g_j\|_{L^2(\Omega)} + \|\boldsymbol{g}_j\|_{L^2(\Omega)}\big]^2.$$

Using these estimates together with

$$\|\boldsymbol{B}\boldsymbol{B}^\intercal + \alpha \boldsymbol{I}\|_{L(Q)} \leq \|B\|_F^2 + \alpha \quad \text{and} \quad \|\boldsymbol{B}(\boldsymbol{G}(u_0) - \boldsymbol{G}^\star)\|_Q \leq \|\boldsymbol{B}\|_F \|\boldsymbol{G}(u_0) - \boldsymbol{G}^\star\|_C$$

to bound the matrix norms in (4.37), we show the continuous estimate in (4.36). Since the estimate of $\boldsymbol{B}_{ij}$ holds in the discrete case as well, the discrete estimate in (4.36) follows analogously. □

The next lemma shows an estimate similar to Lemma 4.18, but without the additional factor $\|\boldsymbol{p}\|_Q$. Note that such an additional factor cannot be expected since neither $u(\boldsymbol{p})$ nor $z(\boldsymbol{p})$ are linear in the parameter.

Lemma 4.20. *There exists a constant $C > 0$ such that, for all $\mathcal{T}_H \in \mathbb{T}$ and $\boldsymbol{p} \in \{\boldsymbol{p}^\star, \boldsymbol{p}_H^\star\}$,*

$$\begin{aligned} |\!|\!| u(\boldsymbol{p}) - u_H(\boldsymbol{p}) |\!|\!| &\leq C\Big[\sum_{i=0}^{n_Q} |\!|\!| u_i - u_{H,i} |\!|\!|^2\Big]^{1/2}, \\ |\!|\!| z(\boldsymbol{p}) - z_H(\boldsymbol{p}) |\!|\!| &\leq C\Big[\sum_{j=1}^{n_C} |\!|\!| z_j - z_{H,j} |\!|\!|^2\Big]^{1/2}. \end{aligned} \tag{4.38}$$

The constant C depends only on $\boldsymbol{G}^\star$, the data $\boldsymbol{A}$, f_i, g_j, $\boldsymbol{f}_i$, and $\boldsymbol{g}_j$, and the constants $C_{\boldsymbol{A}}$ from Lemma 4.17 and $C_\star$ from Lemma 4.19.

Proof. The estimate for the difference in the state u follows along the same lines as in the proof of Lemma 4.18, where the parameter can be estimated by (4.36). For the difference in the co-state, we note that (4.33) and $\boldsymbol{p} \in \{\boldsymbol{p}^\star, \boldsymbol{p}_H^\star\}$ imply that

$$|\!|\!| u_H(\boldsymbol{p}) |\!|\!| \leq C < \infty, \tag{4.39}$$

where the constant $C > 0$ depends only on C_A from Lemma 4.17, the data f_i and $\boldsymbol{f}_i$, and $C_\star$ from Lemma 4.19. Thus, with continuity of the measurement functionals G_j, which depend only on g_j, $\boldsymbol{g}_j$, we have that

$$\begin{aligned}
\|\!| z(\boldsymbol{p}) - z_H(\boldsymbol{p}) |\!\| &\leq \sum_{j=1}^{n_C} \left| G_j\big(u_H(\boldsymbol{p})\big) - G_j^\star \right| \|\!| z_j - z_{H,j} |\!\| \\
&\leq \left\| \boldsymbol{G}\big(u_H(\boldsymbol{p})\big) - \boldsymbol{G}^\star \right\|_C \Big[\sum_{j=1}^{n_C} \|\!| z_j - z_{H,j} |\!\|^2 \Big]^{1/2} \\
&\lesssim \big(\|\!| u_H(\boldsymbol{p}) |\!\| + \|\boldsymbol{G}^\star\|_C \big) \Big[\sum_{j=1}^{n_C} \|\!| z_j - z_{H,j} |\!\|^2 \Big]^{1/2} \overset{(4.39)}{\lesssim} \Big[\sum_{j=1}^{n_C} \|\!| z_j - z_{H,j} |\!\|^2 \Big]^{1/2}.
\end{aligned}$$

This concludes the proof. □

4.4.2 Error bound for parameter error

The next lemma estimates the error in the derivative of the least-squares functional.

Lemma 4.21. *For all $q, p \in Q$, it holds that*

$$\begin{aligned}
|J'[\boldsymbol{p}](\boldsymbol{q}) - J_H'[\boldsymbol{p}](\boldsymbol{q})| \leq{}& \|\!| u(\boldsymbol{p}) - u_H(\boldsymbol{p}) |\!\| \|\!| z'(\boldsymbol{q}) - z_H'(\boldsymbol{q}) |\!\| \\
&+ \|\!| u'(\boldsymbol{q}) - u_H'(\boldsymbol{q}) |\!\| \|\!| z(\boldsymbol{p}) - z_H(\boldsymbol{p}) |\!\| \\
&+ \|\!| u(\boldsymbol{p}) - u_H(\boldsymbol{p}) |\!\| \|\!| u'(\boldsymbol{q}) - u_H'(\boldsymbol{q}) |\!\| \sum_{j=1}^{n_C} \|\!| z_j - z_{H,j} |\!\|^2.
\end{aligned} \tag{4.40}$$

Proof. Let $\boldsymbol{q}, \boldsymbol{p} \in Q$. Note that $\boldsymbol{r}'(\boldsymbol{q}) = \boldsymbol{G}(u'(\boldsymbol{q}))$ and $\boldsymbol{r}_H'(\boldsymbol{q}) = \boldsymbol{G}(u_H'(\boldsymbol{q}))$. Therefore, we have that

$$\begin{aligned}
J'[\boldsymbol{p}](\boldsymbol{q}) - J_H'[\boldsymbol{p}](\boldsymbol{q}) &\overset{(4.11)}{=} \big\langle \boldsymbol{r}(\boldsymbol{p}),\, \boldsymbol{r}'(\boldsymbol{q}) \big\rangle - \big\langle \boldsymbol{r}_H(\boldsymbol{p}),\, \boldsymbol{r}_H'(\boldsymbol{q}) \big\rangle \\
&= \big\langle \boldsymbol{r}(\boldsymbol{p}) - \boldsymbol{r}_H(\boldsymbol{p}),\, \boldsymbol{r}'(\boldsymbol{q}) \big\rangle + \big\langle \boldsymbol{r}_H(\boldsymbol{p}),\, \boldsymbol{r}'(\boldsymbol{q}) - \boldsymbol{r}_H'(\boldsymbol{q}) \big\rangle \\
&= \Big\langle \boldsymbol{r}(\boldsymbol{p}) - \boldsymbol{r}_H(\boldsymbol{p}),\, \boldsymbol{G}(u'(\boldsymbol{q})) \Big\rangle + \Big\langle \boldsymbol{r}_H(\boldsymbol{p}),\, \boldsymbol{G}\big(u'(\boldsymbol{q}) - u_H'(\boldsymbol{q})\big) \Big\rangle.
\end{aligned} \tag{4.41}$$

The first term in (4.41) can be reformulated by inserting the term $\boldsymbol{G}(u_H'(\boldsymbol{q}))$, the definition of $z'(\boldsymbol{q})$, and the Galerkin orthogonality. This yields that

$$\begin{aligned}
&\big\langle \boldsymbol{r}(\boldsymbol{p}) - \boldsymbol{r}_H(\boldsymbol{p}),\, \boldsymbol{G}(u'(\boldsymbol{q})) \big\rangle \\
&\quad = \Big\langle \boldsymbol{G}\big(u(\boldsymbol{p}) - u_H(\boldsymbol{p})\big),\, \boldsymbol{G}\big(u'(\boldsymbol{q}) - u_H'(\boldsymbol{q})\big) \Big\rangle + \Big\langle \boldsymbol{G}\big(u(\boldsymbol{p}) - u_H(\boldsymbol{p})\big),\, \boldsymbol{G}\big(u_H'(\boldsymbol{q})\big) \Big\rangle \\
&\quad \overset{(4.18)}{=} \Big\langle \boldsymbol{G}\big(u(\boldsymbol{p}) - u_H(\boldsymbol{p})\big),\, \boldsymbol{G}\big(u'(\boldsymbol{q}) - u_H'(\boldsymbol{q})\big) \Big\rangle + a\big(u(\boldsymbol{p}) - u_H(\boldsymbol{p}), z'(\boldsymbol{q})\big) \\
&\quad = \Big\langle \boldsymbol{G}\big(u(\boldsymbol{p}) - u_H(\boldsymbol{p})\big),\, \boldsymbol{G}\big(u'(\boldsymbol{q}) - u_H'(\boldsymbol{q})\big) \Big\rangle + a\big(u(\boldsymbol{p}) - u_H(\boldsymbol{p}), z'(\boldsymbol{q}) - z_H'(\boldsymbol{q})\big).
\end{aligned}$$

We employ the definition of the co-state components z_j and the Galerkin orthogonalities to obtain

that

$$\begin{aligned}\left\langle \boldsymbol{G}\big(u(\boldsymbol{p}) - u_H(\boldsymbol{p})\big),\, \boldsymbol{G}\big(u'(\boldsymbol{q}) - u'_H(\boldsymbol{q})\big)\right\rangle &= \sum_{j=1}^{n_C} G_j\big(u(\boldsymbol{p}) - u_H(\boldsymbol{p})\big)\, G_j\big(u'(\boldsymbol{q}) - u'_H(\boldsymbol{q})\big)\\ &\overset{(4.16)}{=} \sum_{j=1}^{n_C} a\big(u(\boldsymbol{p}) - u_H(\boldsymbol{p}), z_j\big)\, a\big(u'(\boldsymbol{q}) - u'_H(\boldsymbol{q}), z_j\big)\\ &= \sum_{j=1}^{n_C} a\big(u(\boldsymbol{p}) - u_H(\boldsymbol{p}), z_j - z_{H,j}\big)\, a\big(u'(\boldsymbol{q}) - u'_H(\boldsymbol{q}), z_j - z_{H,j}\big).\end{aligned}$$

For the second term in (4.41), we use the definition of $z(\boldsymbol{q})$ and the Galerkin orthogonality to obtain that

$$\begin{aligned}\Big\langle \boldsymbol{r}_H(\boldsymbol{p}),\, \boldsymbol{G}\big(u'(\boldsymbol{q}) - u'_H(\boldsymbol{q})\big)\Big\rangle &\overset{(4.17)}{=} a\big(u'(\boldsymbol{q}) - u'_H(\boldsymbol{q}), z(\boldsymbol{p})\big)\\ &= a\big(u'(\boldsymbol{q}) - u'_H(\boldsymbol{q}), z(\boldsymbol{p}) - z_H(\boldsymbol{p})\big).\end{aligned}$$

Finally, the claim (4.40) results from combining above identities and using the Cauchy–Schwarz inequality. □

Finally, we combine the last auxiliary results to obtain an estimate for the error in the parameter.

Lemma 4.22. *There exists a constant $C > 0$ such that the approximation error in the parameter can be estimated by*

$$\begin{aligned}\|\boldsymbol{p}^\star - \boldsymbol{p}_H^\star\|_Q \le C\Big[&\|\!| u(\boldsymbol{p}_H^\star) - u_H(\boldsymbol{p}_H^\star) |\!\|\Big[\sum_{j=1}^{n_C} |\!\| z_j - z_{H,j} |\!\|^2\Big]^{1/2}\\ &\times \Big(1 + \Big[\sum_{j=1}^{n_C} |\!\| z_j - z_{H,j} |\!\|^2\Big]^{1/2}\Big[\sum_{i=1}^{n_Q} |\!\| u_i - u_{H,i} |\!\|^2\Big]^{1/2}\Big)\\ &+ |\!\| z(\boldsymbol{p}_H^\star) - z_H(\boldsymbol{p}_H^\star) |\!\|\Big[\sum_{i=1}^{n_Q} |\!\| u_i - u_{H,i} |\!\|^2\Big]^{1/2}\Big].\end{aligned} \tag{4.42}$$

The constant $C > 0$ depends only on Ω and κ as well as the data $\boldsymbol{A}$, f_i, g_j, $\boldsymbol{f}_i$, and $\boldsymbol{g}_j$.

Proof. In the following, let $\boldsymbol{q} := \boldsymbol{p}_H^\star - \boldsymbol{p}^\star$. Note that the second order optimality condition (4.10) holds for all $\boldsymbol{q} \in \mathbb{R}^{n_Q}$, since J'' is independent of its linearization point. Therefore, we have that

$$\kappa\|\boldsymbol{q}\|_Q^2 \overset{(4.10)}{\le} J''(\boldsymbol{q}, \boldsymbol{q}) \overset{(4.12),(4.13)}{=} J'[\boldsymbol{p}_H^\star](\boldsymbol{q}) - J'[\boldsymbol{p}^\star](\boldsymbol{q}),$$

since $J'[\cdot](\boldsymbol{q})$ is affine. With the continuous and discrete first order optimality conditions, (4.9) and (4.15), respectively, we see that

$$\begin{aligned}\kappa\|\boldsymbol{q}\|_Q^2 &\le \big(J'[\boldsymbol{p}_H^\star](\boldsymbol{q}) - J'_H[\boldsymbol{p}_H^\star](\boldsymbol{q})\big) + \big(J'_H[\boldsymbol{p}_H^\star](\boldsymbol{q}) - J'[\boldsymbol{p}^\star](\boldsymbol{q})\big)\\ &\le J'[\boldsymbol{p}_H^\star](\boldsymbol{q}) - J'_H[\boldsymbol{p}_H^\star](\boldsymbol{q}).\end{aligned}$$

This last expression can be further bounded by Lemma 4.21:

$$\kappa\|\boldsymbol{q}\|_Q^2 \overset{(4.40)}{\leq} |||u(\boldsymbol{p}_H^\star) - u_H(\boldsymbol{p}_H^\star)||| \, |||z'(\boldsymbol{q}) - z_H'(\boldsymbol{q})||| + |||u'(\boldsymbol{q}) - u_H'(\boldsymbol{q})||| \, |||z(\boldsymbol{p}_H^\star) - z_H(\boldsymbol{p}_H^\star)|||$$
$$+ |||u(\boldsymbol{p}_H^\star) - u_H(\boldsymbol{p}_H^\star)||| \, |||u'(\boldsymbol{q}) - u_H'(\boldsymbol{q})||| \sum_{j=1}^{n_C} |||z_j - z_{H,j}|||^2.$$

From the right-hand side, a factor $\|\boldsymbol{q}\|_Q$ can be split off from the $\boldsymbol{q}$-dependent terms by Lemma 4.18 to obtain that

$$\kappa\|\boldsymbol{q}\|_Q^2 \overset{(4.34)}{\lesssim} |||u(\boldsymbol{p}_H^\star) - u_H(\boldsymbol{p}_H^\star)||| \, \|\boldsymbol{q}\|_Q \Big[\sum_{j=1}^{n_C} |||z_j - z_{H,j}|||^2 \Big]^{1/2}$$
$$+ \|\boldsymbol{q}\|_Q \Big[\sum_{i=1}^{n_Q} |||u_i - u_{H,i}|||^2 \Big]^{1/2} |||z(\boldsymbol{p}_H^\star) - z_H(\boldsymbol{p}_H^\star)|||$$
$$+ |||u(\boldsymbol{p}_H^\star) - u_H(\boldsymbol{p}_H^\star)||| \, \|\boldsymbol{q}\|_Q \Big[\sum_{i=1}^{n_Q} |||u_i - u_{H,i}|||^2 \Big]^{1/2} \sum_{j=1}^{n_C} |||z_j - z_{H,j}|||^2.$$

The claim follows by division through $\kappa\|\boldsymbol{q}\|_Q = \kappa\|\boldsymbol{p}^\star - \boldsymbol{p}_H^\star\|_Q$. □

From (4.42), we can absorb higher order terms to deduce an upper bound for the parameter error, which is the assertion of Theorem 4.5.

Proof of Theorem 4.5. Note that due to the stability estimate (4.33) there holds for all $0 \leq i \leq n_Q$ and all $1 \leq j \leq n_C$ that

$$|||u_i - u_{H,i}||| \leq 2C_{\boldsymbol{A}} \big[\|f_i\|_{L^2(\Omega)} + \|\boldsymbol{f}_i\|_{L^2(\Omega)} \big], \quad |||z_j - z_{H,j}||| \leq 2C_{\boldsymbol{A}} \big[\|g_j\|_{L^2(\Omega)} + \|\boldsymbol{g}_j\|_{L^2(\Omega)} \big].$$

Using these estimates on the factor in (4.42), we obtain that

$$1 + \Big[\sum_{j=1}^{n_C} |||z_j - z_{H,j}|||^2 \Big]^{1/2} \Big[\sum_{i=1}^{n_Q} |||u_i - u_{H,i}|||^2 \Big]^{1/2} \lesssim 1, \tag{4.43}$$

where the hidden constant depends only on $C_{\boldsymbol{A}}$ and the data f_i, g_j, $\boldsymbol{f}_i$, and $\boldsymbol{g}_j$. Hence, (4.42) reads as

$$\|\boldsymbol{p}^\star - \boldsymbol{p}_H^\star\|_Q \lesssim |||u(\boldsymbol{p}_H^\star) - u_H(\boldsymbol{p}_H^\star)||| \Big[\sum_{j=1}^{n_C} |||z_j - z_{H,j}|||^2 \Big]^{1/2}$$
$$+ |||z(\boldsymbol{p}_H^\star) - z_H(\boldsymbol{p}_H^\star)||| \Big[\sum_{i=1}^{n_Q} |||u_i - u_{H,i}|||^2 \Big]^{1/2}.$$

Lemma 4.20 allows to bound the energy norms to finally obtain (4.21). This concludes the proof. □

Remark 4.23. *Note that our adaptive Algorithm 4A guarantees convergence*

$$\Big[\sum_{j=1}^{n_C} |||z_j - z_{H,j}|||^2 \Big]^{1/2} \Big[\sum_{i=0}^{n_Q} |||u_i - u_{H,i}|||^2 \Big]^{1/2} \to 0 \quad \text{as } \ell \to \infty.$$

In particular, this implies that the estimate (4.43) *is too pessimistic, as the estimated term asymptotically tends to* 1.

4.5 Proof of Theorems 4.13 and 4.14

4.5.1 Linear convergence

We aim to employ the analysis of goal-oriented AFEM done by [FGH+16; FPZ16] to prove convergence rates with optimal algebraic rates for the error estimator. To this end, we first note that the axioms presented in Proposition 4.7 also hold for the sums of state and co-state components, respectively. For convenience of the reader, we state the main intermediate results for proving Theorems 4.13 and 4.14.

In view of Algorithm 4A, we define estimators $\widetilde{\eta}_H$ on $\mathcal{X}_H^{n_Q+1}$ and $\widetilde{\zeta}_H$ on $\mathcal{X}_H^{n_C}$ by

$$\begin{aligned}\widetilde{\eta}_H(T,\boldsymbol{v}_H)^2 &:= \sum_{i=0}^{n_Q} \eta_{H,i}(T,v_{H,i})^2 \quad \text{for all } \boldsymbol{v}_H \in \mathcal{X}_H^{n_Q+1},\\ \widetilde{\zeta}_H(T,\boldsymbol{w}_H)^2 &:= \sum_{j=1}^{n_C} \zeta_{H,j}(T,w_{H,j})^2 \quad \text{for all } \boldsymbol{w}_H \in \mathcal{X}_H^{n_C}\end{aligned} \tag{4.44}$$

such that there holds

$$\varrho_H(T)^2 = 2\,\widetilde{\eta}_H(T,\boldsymbol{u}_H)^2\widetilde{\zeta}_H(T,\boldsymbol{z}_H)^2, \tag{4.45}$$

where we set $\boldsymbol{u}_H = (u_{H,i})_{i=0}^{n_Q} \in \mathcal{X}_H^{n_Q+1}$ and $\boldsymbol{z}_H = (z_{H,j})_{j=1}^{n_C} \in \mathcal{X}_H^{n_C}$. We employ the same notation, e.g., $\widetilde{\eta}_H(\mathcal{U}_H,\boldsymbol{v}_H)$ or $\widetilde{\zeta}_H(\boldsymbol{w}_H)$, as for $\eta_{H,i}$ and $\zeta_{H,j}$ in (4.22)–(4.23). Moreover, we equip the spaces $\mathcal{X}^{n_Q+1}$ and $\mathcal{X}^{n_C}$ with the norms

$$|||\boldsymbol{v}|||_{n_Q} := \Big[\sum_{i=0}^{n_Q} |||v_i|||^2\Big]^{1/2} \text{ for all } \boldsymbol{v} \in \mathcal{X}^{n_Q+1}, \quad |||\boldsymbol{w}|||_{n_C} := \Big[\sum_{j=1}^{n_C} |||w_j|||^2\Big]^{1/2} \text{ for all } \boldsymbol{w} \in \mathcal{X}^{n_C},$$

and note that $\mathcal{X}_H^{n_Q+1} \subseteq \mathcal{X}^{n_Q+1}$ as well as $\mathcal{X}_H^{n_C} \subseteq \mathcal{X}^{n_C}$ for all $\mathcal{T}_H \in \mathbb{T}$. Then, the properties (A1)–(A4) from Proposition 4.7 also hold for $\widetilde{\eta}_H$ and $\widetilde{\zeta}_H$:

Proposition 4.24. *Let $C_{\mathrm{stab}}, C_{\mathrm{rel}}, C_{\mathrm{drel}} > 0$, and $0 < q_{\mathrm{red}} < 1$ be the constants from Proposition 4.7. Then, for all $\mathcal{T}_H \in \mathbb{T}(\mathcal{T}_0)$, and all $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$, there hold the following properties:*

(A1$^+$) Stability: For all $\boldsymbol{v}_h \in \mathcal{X}_h^{n_Q+1}$, $\boldsymbol{v}_H \in \mathcal{X}_H^{n_Q+1}$, $\boldsymbol{w}_h \in \mathcal{X}_h^{n_C}$, $\boldsymbol{w}_H \in \mathcal{X}_H^{n_C}$, and $\mathcal{U}_H \subseteq \mathcal{T}_h \cap \mathcal{T}_H$, it holds that

$$\begin{aligned}\big|\widetilde{\eta}_h(\mathcal{U}_H,\boldsymbol{v}_h) - \widetilde{\eta}_H(\mathcal{U}_H,\boldsymbol{v}_H)\big| &\le C_{\mathrm{stab}}\,|||\boldsymbol{v}_h - \boldsymbol{v}_H|||_{n_Q},\\ \big|\widetilde{\zeta}_h(\mathcal{U}_H,\boldsymbol{w}_h) - \widetilde{\zeta}_H(\mathcal{U}_H,\boldsymbol{w}_H)\big| &\le C_{\mathrm{stab}}\,|||\boldsymbol{w}_h - \boldsymbol{w}_H|||_{n_C}.\end{aligned}$$

(A2$^+$) Reduction: For all $\boldsymbol{v}_H \in \mathcal{X}_H^{n_Q+1}$ and $\boldsymbol{w}_H \in \mathcal{X}_H^{n_C}$, it holds that

$$\widetilde{\eta}_h(\mathcal{T}_h\backslash\mathcal{T}_H,\boldsymbol{v}_H) \le q_{\mathrm{red}}\,\widetilde{\eta}_H(\mathcal{T}_H\backslash\mathcal{T}_h,\boldsymbol{v}_H), \quad \widetilde{\zeta}_h(\mathcal{T}_h\backslash\mathcal{T}_H,\boldsymbol{w}_H) \le q_{\mathrm{red}}\,\widetilde{\zeta}_H(\mathcal{T}_H\backslash\mathcal{T}_h,\boldsymbol{w}_H).$$

(A3$^+$) Reliability: The state and co-state components $\boldsymbol{u}_H = (u_{H,i})_{i=0}^{n_Q} \in \mathcal{X}_H^{n_Q+1}$ and $\boldsymbol{z}_H = (z_{H,j})_{j=1}^{n_C} \in \mathcal{X}_H^{n_C}$ satisfy that

$$|||\boldsymbol{u} - \boldsymbol{u}_H|||_{n_Q} \le C_{\mathrm{rel}}\,\widetilde{\eta}_H \quad \text{and} \quad |||\boldsymbol{z} - \boldsymbol{z}_H|||_{n_C} \le C_{\mathrm{rel}}\,\widetilde{\zeta}_H.$$

(A4$^+$) Discrete reliability: The state and co-state components $\boldsymbol{u}_H \in \mathcal{X}_H^{n_Q+1}$, $\boldsymbol{u}_h \in \mathcal{X}_h^{n_Q+1}$, $z_H \in \mathcal{X}_H^{n_C}$, and $z_h \in \mathcal{X}_h^{n_C}$ satisfy that

$$\|\boldsymbol{u}_h - \boldsymbol{u}_H\|_{n_Q} \le C_{\mathrm{drel}}\,\widetilde{\eta}_H(\mathcal{T}_H\backslash\mathcal{T}_h), \quad \|z_h - z_H\|_{n_C} \le C_{\mathrm{drel}}\,\widetilde{\zeta}_H(\mathcal{T}_H\backslash\mathcal{T}_h).$$

Proof. For stability (A1$^+$) of the state, the inverse triangle inequality proves that

$$\begin{aligned}
\big|\widetilde{\eta}_h(\mathcal{U}_H, \boldsymbol{v}_h) - \widetilde{\eta}_H(\mathcal{U}_H, \boldsymbol{v}_H)\big| &= \Big|\Big[\sum_{i=0}^{n_Q} \eta_{h,i}(\mathcal{U}_H, v_{h,i})^2\Big]^{1/2} - \Big[\sum_{i=0}^{n_Q} \eta_{H,i}(\mathcal{U}_H, v_{H,i})^2\Big]^{1/2}\Big| \\
&\le \Big|\sum_{i=0}^{n_Q} \big[\eta_{h,i}(\mathcal{U}_H, v_{h,i}) - \eta_{H,i}(\mathcal{U}_H, v_{H,i})\big]^2\Big|^{1/2} \\
&\overset{(A1)}{\le} C_{\mathrm{stab}} \Big|\sum_{i=0}^{n_Q} \|v_{h,i} - v_{H,i}\|^2\Big|^{1/2} = C_{\mathrm{stab}}\,\|\boldsymbol{v}_h - \boldsymbol{v}_H\|_{n_Q}.
\end{aligned}$$

The estimate for the co-state follows analogously. Finally, the properties (A2$^+$)–(A4$^+$) follow directly from the corresponding properties from Proposition 4.7. □

Since the problems for the state and co-state components, (4.6) and (4.16), fit into the Lax–Milgram setting, there hold the Pythagoras identities

$$\begin{aligned}
\|\boldsymbol{u} - \boldsymbol{u}_{\ell+n}\|_{n_Q}^2 + \|\boldsymbol{u}_{\ell+n} - \boldsymbol{u}_\ell\|_{n_Q}^2 &= \|\boldsymbol{u} - \boldsymbol{u}_\ell\|_{n_Q}^2, \\
\|z - z_{\ell+n}\|_{n_C}^2 + \|z_{\ell+n} - z_\ell\|_{n_C}^2 &= \|z - z_\ell\|_{n_C}^2,
\end{aligned}$$

for all $\ell, n \in \mathbb{N}_0$. Overall, we get the following Proposition as an immediate consequence from, e.g., [FPZ16, Theorem 12]. From this, Theorem 4.13 follows readily with (4.45).

Proposition 4.25. *Suppose* (A1)–(A3) *and* $0 < \theta \le 1$. *Then, Algorithm 4A guarantees linear convergence*

$$\widetilde{\eta}_{\ell+n}\widetilde{\zeta}_{\ell+n} \le C_{\mathrm{lin}}\,q_{\mathrm{lin}}^n\,\widetilde{\eta}_\ell\widetilde{\zeta}_\ell \quad \text{for all } \ell, n \in \mathbb{N}_0. \tag{4.46}$$

The constants $C_{\mathrm{lin}} > 0$ *and* $0 < q_{\mathrm{lin}} < 1$ *depend only on* C_{stab}, q_{red}, C_{rel}, *and the (arbitrary) adaptivity parameter* $0 < \theta \le 1$. □

4.5.2 Proof of optimal rates

For all $r > 0$, we define the (combined) approximation classes

$$\begin{aligned}
\|\boldsymbol{u}\|_{\mathbb{A}_r} &:= \sup_{N\in\mathbb{N}_0} (N+1)^r \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(N)} \widetilde{\eta}_{\mathrm{opt}}(\boldsymbol{u}_{\mathrm{opt}}) \in [0,\infty], \\
\|z\|_{\mathbb{A}_r} &:= \sup_{N\in\mathbb{N}_0} (N+1)^r \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(N)} \widetilde{\zeta}_{\mathrm{opt}}(z_{\mathrm{opt}}) \in [0,\infty].
\end{aligned} \tag{4.47}$$

For these, we get the following result from, e.g., [FPZ16, Theorem 13].

Proposition 4.26. *Let* $0 < \theta < \theta_{\mathrm{opt}} := (1 + C_{\mathrm{stab}}^2 C_{\mathrm{drel}}^2)^{-1}$. *Suppose that the set of marked elements* $\mathcal{M}_\ell$ *in Algorithm 4A*(iv) *has minimal cardinality. Let* $s, t > 0$ *with* $\|\boldsymbol{u}\|_{\mathbb{A}_s} + \|z\|_{\mathbb{A}_t} < \infty$. *Then, there exists a constant* $\widetilde{C}_{\mathrm{opt}} > 0$ *such that*

$$\widetilde{\eta}_\ell\widetilde{\zeta}_\ell \le \widetilde{C}_{\mathrm{opt}}\,\|\boldsymbol{u}\|_{\mathbb{A}_s}\|z\|_{\mathbb{A}_t}\big(\#\mathcal{T}_\ell - \#\mathcal{T}_0\big)^{-(s+t)} \quad \text{for all } \ell \in \mathbb{N}_0. \tag{4.48}$$

The constant $\widetilde{C}_{\mathrm{opt}}$ depends only on C_{stab}, q_{red}, C_{rel}, C_{drel}, C_{mark}, θ, s, t, and the properties (4.30)–(4.32) of the mesh-refinement. □

Finally, Theorem 4.14 follows from Proposition 4.26 by relating the different approximation classes used in both results.

Proof of Theorem 4.14. Let $r > 0$. We show that $\sum_{i=0}^{n_Q} \|u_i\|_{\mathbb{A}_r}^2 \simeq \|\boldsymbol{u}\|_{\mathbb{A}_r}^2$. From the definitions (4.28) and (4.47) of the approximation classes, we immediately see that, for all $i = 0, \dots, n_Q$,

$$\begin{aligned}\|u_i\|_{\mathbb{A}_r} &= \sup_{N\in\mathbb{N}_0} (N+1)^r \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(N)} \eta_{\mathrm{opt},i}(u_{\mathrm{opt},i}) \\ &\le \sup_{N\in\mathbb{N}_0} (N+1)^r \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(N)} \Big[\sum_{i=0}^{n_Q} \eta_{\mathrm{opt},i}(u_{\mathrm{opt},i})^2\Big]^{1/2} = \|\boldsymbol{u}\|_{\mathbb{A}_r}.\end{aligned}$$

Summing the last estimate for all $i = 0, \dots, n_Q$, we obtain that

$$\frac{1}{n_Q+1} \sum_{i=0}^{n_Q} \|u_i\|_{\mathbb{A}_r}^2 \le \|\boldsymbol{u}\|_{\mathbb{A}_r}^2. \tag{4.49}$$

For the converse estimate, we fix $N \in \mathbb{N}$ and define $K := \lfloor N/(n_Q+1) \rfloor$. Let further be $\mathcal{T}_k \in \mathcal{T}(K)$ for $k = 0, \dots, n_Q$ such that

$$\eta_{k,k}(u_{k,k}) = \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(K)} \eta_{\mathrm{opt},k}(u_{\mathrm{opt},k}).$$

With the overlay estimate (4.31), we have that

$$\#\bigoplus_{k=0}^{n_Q} \mathcal{T}_k = \Big[\sum_{k=0}^{n_Q} \#\mathcal{T}_k\Big] - n_Q \#\mathcal{T}_0 = \Big[\sum_{k=0}^{n_Q} (\#\mathcal{T}_k - \#\mathcal{T}_0)\Big] + \#\mathcal{T}_0 = \Big[\sum_{k=0}^{n_Q} K\Big] + \#\mathcal{T}_0 = N + \#\mathcal{T}_0.$$

Therefore, it holds that $\mathcal{T}_\triangle := \bigoplus_{k=0}^{n_Q} \mathcal{T}_k \in \mathbb{T}(N)$. From this, we infer that

$$\min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(N)} \Big[\sum_{i=0}^{n_Q} \eta_{\mathrm{opt},i}(u_{\mathrm{opt},i})^2\Big]^{1/2} \le \Big[\sum_{i=0}^{n_Q} \eta_{\triangle,i}(u_{\triangle,i})^2\Big]^{1/2} \overset{(4.27)}{\le} C_{\mathrm{lin}} \Big[\sum_{i=0}^{n_Q} \eta_{i,i}(u_{i,i})^2\Big]^{1/2}.$$

Multiplying this by $(N+1)^r$, we obtain that

$$\begin{aligned}(N+1)^r \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(N)} \Big[\sum_{i=0}^{n_Q} \eta_{\mathrm{opt},i}(u_{\mathrm{opt},i})^2\Big]^{1/2} &\le C_{\mathrm{lin}} (N+1)^r \Big[\sum_{i=0}^{n_Q} \eta_{i,i}(u_{i,i})^2\Big]^{1/2} \\ &= C_{\mathrm{lin}} \Big(\frac{N+1}{K+1}\Big)^r (K+1)^r \Big[\sum_{i=0}^{n_Q} \eta_{i,i}(u_{i,i})^2\Big]^{1/2} \\ &= C_{\mathrm{lin}} \Big(\frac{N+1}{K+1}\Big)^r (K+1)^r \Big[\sum_{i=0}^{n_Q} \min_{\mathcal{T}_{\mathrm{opt}}\in\mathbb{T}(K)} \eta_{\mathrm{opt},i}(u_{\mathrm{opt},i})^2\Big]^{1/2} \\ &\le C_{\mathrm{lin}} \Big(\frac{N+1}{K+1}\Big)^r \Big[\sum_{i=0}^{n_Q} \|u_i\|_{\mathbb{A}_r}^2\Big]^{1/2}.\end{aligned}$$

Taking the supremum over all $N \in \mathbb{N}$ of the last estimate and using $(N+1)/(K+1) \leq n_Q + 2$, we finally arrive at

$$\|\boldsymbol{u}\|_{\boldsymbol{A}_r} \leq C_{\mathrm{lin}}\,(n_Q+2)^r \Big[\sum_{i=0}^{n_Q} \|u_i\|^2_{\boldsymbol{A}_r}\Big]^{1/2}. \tag{4.50}$$

Thus, by combining (4.49)–(4.50), we have that

$$\frac{1}{n_Q+1}\sum_{i=0}^{n_Q} \|u_i\|^2_{\boldsymbol{A}_s} \leq \|\boldsymbol{u}\|^2_{\boldsymbol{A}_s} \leq C^2_{\mathrm{lin}}(n_Q+2)^{2s}\sum_{i=0}^{n_Q} \|u_i\|^2_{\boldsymbol{A}_s}.$$

Clearly, it thus holds $\|\boldsymbol{u}\|_{\boldsymbol{A}_s} < \infty$ if and only if $s = \min\{s_i \,|\, 0 \leq i \leq n_Q\}$. Analogously, it follows that

$$\frac{1}{n_C}\sum_{j=1}^{n_C} \|z_j\|^2_{\boldsymbol{A}_t} \leq \|\boldsymbol{z}\|^2_{\boldsymbol{A}_t} \leq C^2_{\mathrm{lin}}\,(n_C+1)^{2t}\sum_{j=1}^{n_C} \|z_j\|^2_{\boldsymbol{A}_t},$$

and $\|\boldsymbol{z}\|_{\boldsymbol{A}_t} < \infty$ if and only if $t = \min\{t_j \,|\, 1 \leq j \leq n_C\}$.

Finally, combining the last statements with the statement of Proposition 4.26, we conclude the proof with $C_{\mathrm{opt}} := C^2_{\mathrm{lin}}(n_Q+2)^{s+1/2}(n_C+1)^{t+1/2}\widetilde{C}_{\mathrm{opt}}$. □

4.6 Numerical examples

For the following examples, we consider the initial mesh $\mathcal{T}_0$ of $\Omega := (0,1)^2 \subseteq \mathbb{R}^2$ shown in Figure 4.1 with the sets

$$\begin{aligned} T_1 &:= \{x \in \mathbb{R}^2 \,|\, x_1 + x_2 > 3/2\} \cap \Omega,\\ T_2 &:= \{x \in \mathbb{R}^2 \,|\, x_1 + x_2 < 1/2\} \cap \Omega,\\ T_3 &:= \{x \in \mathbb{R}^2 \,|\, \max\{x_1, x_2\} < 1/4\} \cap \Omega. \end{aligned}$$

With the characteristic function χ_ω of a measurable subset $\omega \subseteq \Omega$, we define

$$\begin{aligned} f_1 &:= 4x_1(1-x_1) + 4x_2(1-x_2), \quad f_2 := 5\pi^2 \sin(\pi x_1)\sin(2\pi x_2),\\ \boldsymbol{g}_1 &:= (1,0)^\intercal \chi_{T_1}, \quad \boldsymbol{g}_2 := (-1,0)^\intercal \chi_{T_2}, \quad g_3 := \chi_{T_3}, \end{aligned}$$

as well as $f_0 = g_1 = g_2 = 0$ and $\boldsymbol{f}_0 = \boldsymbol{f}_1 = \boldsymbol{f}_2 = \boldsymbol{g}_3 = \boldsymbol{0}$. In all our experiments, we set $\alpha = 0$. Since all $f_i, g_j, \boldsymbol{f}_i, \boldsymbol{g}_j$ are linearly independent, the matrix $\boldsymbol{B}^\intercal\boldsymbol{B}$ has full rank. In particular, condition (4.10) is satisfied; see Remark 4.3. As marking parameter, we use $\theta = 0.5$.

4.6.1 Single parameter and measurement

For our first experiment, we consider the following parametrized discrete PDE problem with parameter $\boldsymbol{p} \in \mathcal{Q} := \mathbb{R}$: Find $u_H(\boldsymbol{p}) \in \mathcal{X}_H$ such that

$$a(u_H(\boldsymbol{p}), v_H) := \int_\Omega \nabla u_H(\boldsymbol{p}) \cdot \nabla v_H \,\mathrm{d}x = p_1 \int_\Omega f_1 v_H \,\mathrm{d}x =: b(\boldsymbol{p}, v) \quad \text{for all } v_H \in \mathcal{X}_H. \tag{4.51}$$

The exact (continuous) solution of this problem is known to be

$$u(\boldsymbol{p}) = p_1 u_1 = p_1 x_1 x_2 (1-x_1)(1-x_2).$$

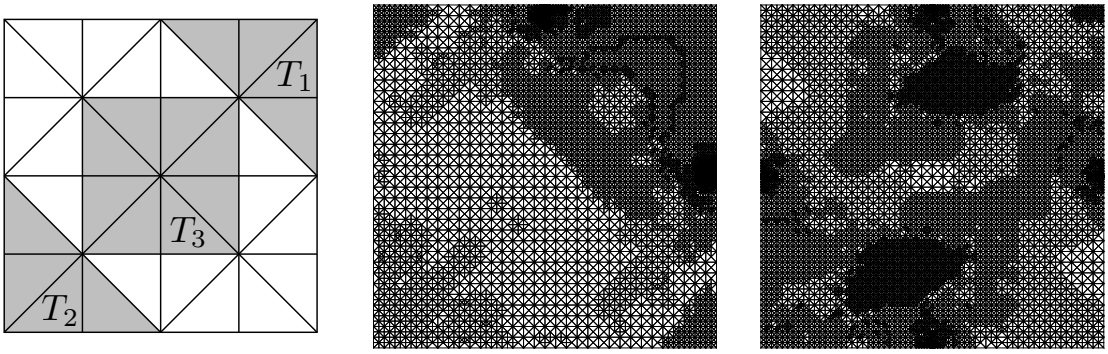


Figure 4.1: Left: Initial mesh $\mathcal{T}_0$ of the unit square $(0,1)^2$ for numerical experiments. Middle: Mesh with $\#\mathcal{T}_{10} = 11096$ elements for the setting of Section 4.6.1. Right: Mesh with $\#\mathcal{T}_9 = 16080$ elements for the setting of Section 4.6.2.

We further suppose that we have one measurement (corresponding to $\boldsymbol{p}^\star = 1$)

$$\boldsymbol{G}^\star = \frac{11}{960} = \boldsymbol{G}(u(\boldsymbol{p}^\star)) := -\int_\Omega \boldsymbol{g}_1 \cdot \nabla u(\boldsymbol{p}^\star)\,\mathrm{d}x = -\int_{T_1} \frac{\partial u(\boldsymbol{p}^\star)}{\partial x_1}\,\mathrm{d}x.$$

We compute an approximation to $\boldsymbol{p}^\star$ by two methods:

- The approximations $\boldsymbol{p}_\ell^\star$ are obtained by our adaptive Algorithm 4A, which is driven by the estimator ϱ_ℓ from (4.25).
- The approximations $\overline{\boldsymbol{p}}_\ell^\star$ are obtained by our adaptive algorithm, where ϱ_ℓ is substituted by $\overline{\varrho}_\ell(\overline{\boldsymbol{p}}_\ell^\star) := \eta_\ell(\overline{\boldsymbol{p}}_\ell^\star) + \zeta_\ell(\overline{\boldsymbol{p}}_\ell^\star)$, which is the prevalent error estimator from the existing literature on AFEM for optimal control problems [BM11; GY17]. Here, for $\boldsymbol{p} \in Q$, $\eta_H(\boldsymbol{p})$ and $\zeta_H(\boldsymbol{p})$ are the residual error estimators of the energy errors of $u_H(\boldsymbol{p})$ from (4.4) and $z_H(\boldsymbol{p})$ from (4.17), respectively.

The results can be seen in Figure 4.2. We see that the classical estimator $\overline{\varrho}_\ell(\overline{\boldsymbol{p}}_\ell^\star)$ drastically underestimates the rate of the parameter error, whereas our estimator ϱ_ℓ matches it perfectly. Also, the parameter error of our approach is uniformly better by some (small) multiplicative factor. However, this effect is negligible for large $\#\mathcal{T}_\ell$.

4.6.2 Multiple parameters and measurements with perturbation

For our second experiment, we consider the following parametrized discrete PDE problem with parameter $\boldsymbol{p} \in Q := \mathbb{R}^2$: Find $u_H(\boldsymbol{p}) \in \mathcal{X}_H$ such that

$$\begin{aligned} a(u_H(\boldsymbol{p}), v_H) &:= \int_\Omega \nabla u_H(\boldsymbol{p}) \cdot \nabla v_H \,\mathrm{d}x \\ &= p_1 \int_\Omega f_1 v_H \,\mathrm{d}x + p_2 \int_\Omega f_2 v_H \,\mathrm{d}x =: b(\boldsymbol{p}, v) \quad \text{for all } v_H \in \mathcal{X}_H. \end{aligned} \tag{4.52}$$

The exact (continuous) solution of this problem is known to be

$$u(\boldsymbol{p}) = p_1 u_1 + p_2 u_2 = p_1 x_1 x_2 (1 - x_1)(1 - x_2) + p_2 \sin(\pi x_1) \sin(2\pi x_2).$$

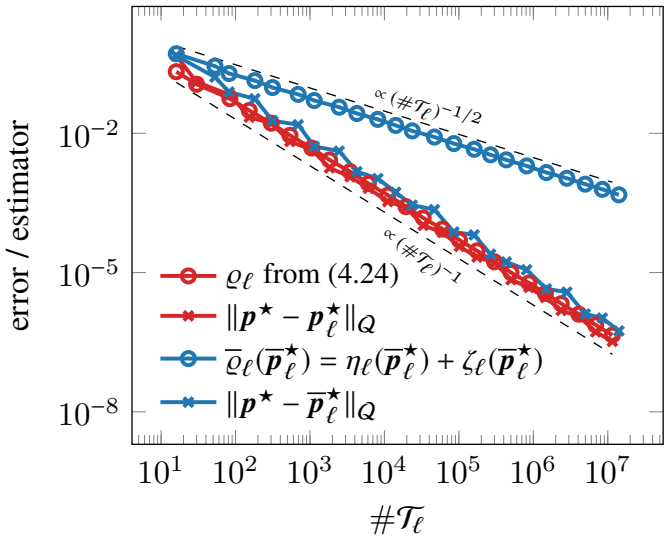


Figure 4.2: Results for the problem from Section 4.6.1. The $\boldsymbol{p}_\ell^\star$ are computed by our adaptive Algorithm 4A driven by the estimator ϱ_ℓ; the $\overline{\boldsymbol{p}}_\ell^\star$ are computed by an algorithm driven by $\overline{\varrho}_\ell(\overline{\boldsymbol{p}}_\ell^\star) = \eta_\ell(\overline{\boldsymbol{p}}_\ell^\star) + \zeta_\ell(\overline{\boldsymbol{p}}_\ell^\star)$.

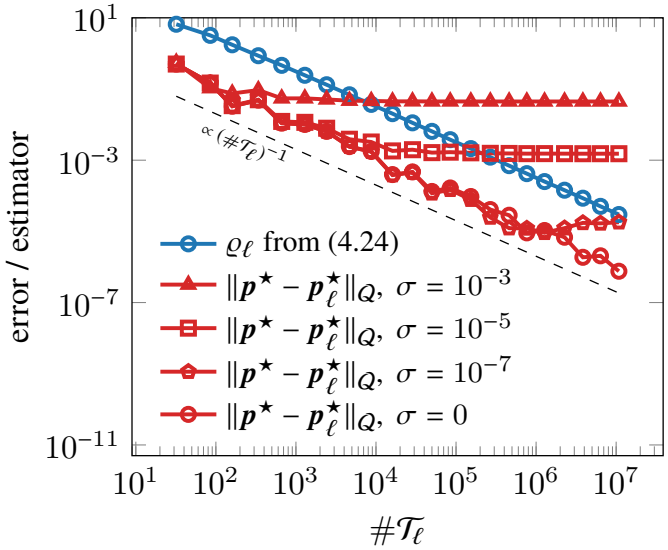


Figure 4.3: Results for the problem from Section 4.6.2. Differently marked lines are obtained by perturbing the true measurements by Gaussian noise with standard deviation σ.

We further suppose that we have three *exact* measurements (corresponding to the exact parameter $\boldsymbol{p}^\star = (2, 1/2)^\intercal$)

$$\begin{aligned}\overline{\boldsymbol{G}}^\star &= \Big(\frac{11\pi + 160}{480\pi}, \frac{11\pi - 160}{480\pi}, \frac{121}{4608}\Big)^\intercal = \big(G_1(u(\boldsymbol{p}^\star)), G_2(u(\boldsymbol{p}^\star)), G_3(u(\boldsymbol{p}^\star))\big)^\intercal \\ &:= \Big(-\int_\Omega \boldsymbol{g}_1 \cdot \nabla u(\boldsymbol{p}^\star)\,\mathrm{d}x, -\int_\Omega \boldsymbol{g}_2 \cdot \nabla u(\boldsymbol{p}^\star)\,\mathrm{d}x, \int_\Omega g_3 u(\boldsymbol{p}^\star)\,\mathrm{d}x\Big)^\intercal \\ &= \Big(-\int_{T_1} \frac{\partial u(\boldsymbol{p}^\star)}{\partial x_1}\,\mathrm{d}x, \int_{T_2} \frac{\partial u(\boldsymbol{p}^\star)}{\partial x_1}\,\mathrm{d}x, \int_{T_3} u(\boldsymbol{p}^\star)\,\mathrm{d}x\Big)^\intercal.\end{aligned}$$

These exact measurements are perturbed by Gaussian random noise $X_k \sim N(0, \sigma^2)$ for $i = 1, 2, 3$ and for some standard deviation $\sigma \geq 0$, such that $\boldsymbol{G}^\star = \overline{\boldsymbol{G}}^\star + (X_1, X_2, X_3)^\intercal$. Note that, for non-vanishing perturbation, $\boldsymbol{G}^\star$ does not necessarily coincide with $\boldsymbol{G}(u(\boldsymbol{p}^\star))$ anymore. Likewise, the sequence $(\boldsymbol{p}_\ell^\star)_{\ell\in\mathbb{N}}$ does not converge to the given parameter $\boldsymbol{p}^\star$ but rather to some $\overline{\boldsymbol{p}}^\star \in Q$ which is the least-squares solution to (4.5) on the continuous level with the perturbed measurements $\boldsymbol{G}^\star$.

We compute $\boldsymbol{p}_\ell^\star$ by Algorithm 4A with different levels of perturbation. The results can be seen in Figure 4.3. We see that, for the different levels of perturbation, the parameter error cannot fall beyond a threshold that depends on the standard deviation σ of the perturbation. This is to be expected, since inference of parameters from experiments is limited by the accuracy of the measurement. Our estimator, however, is independent of the discrete parameter and measurements and, hence, continues to converge independently of the perturbation. In particular, the matrix $\boldsymbol{B}_\ell$ on the finest level of the algorithm can be stored and reused to compute a refined parameter value if a new set of measurements with improved accuracy becomes available. This is not the case for the sum estimator $\overline{\varrho}_\ell(\overline{\boldsymbol{p}}_\ell^\star)$ from the last section, since it explicitly depends on the parameter estimates.

5 MooAFEM: An object oriented Matlab code for higher-order (nonlinear) adaptive FEM

Sections 5.2–5.5 of this chapter are taken from:
M. Innerberger and D. Praetorius. MooAFEM: An object oriented Matlab code for higher-order (nonlinear) adaptive FEM, 2022. arXiv: 2203.01845

5.1 Introduction

This last chapter is dedicated to the implementation of the numerical methods presented in chapters 1–4. This is an integral part of research on numerical analysis, yet often treated superficially. The implementation efforts for the problems of this thesis culminated in the development of our own research code, the MATLAB object oriented AFEM library MooAFEM. We give here an overview over its structure and capabilities.

A reasonably general class of problems to consider for research in numerical analysis is the following: With notation and assumptions from Chapter 1, we assume the (polygonal) boundary of the domain Ω to consist of *Robin*, *Neumann*, and *Dirichlet* boundary, i.e., $\partial\Omega = \overline{\Gamma}_R \cup \overline{\Gamma}_N \cup \overline{\Gamma}_D$ where $\Gamma_R, \Gamma_N, \Gamma_D$ are pairwise disjoint and relatively open. With this decomposition of the boundary, we consider the model problem

$$
\begin{aligned}
-\operatorname{div} \boldsymbol{A}\nabla u + \boldsymbol{b}\cdot\nabla u + cu &= f - \operatorname{div} \boldsymbol{f} && \text{in } \Omega, && \text{(5.1a)}\\
\alpha\, u + \boldsymbol{A}\nabla u\cdot\boldsymbol{n} &= \gamma && \text{on } \Gamma_R, && \text{(5.1b)}\\
\boldsymbol{A}\nabla u\cdot\boldsymbol{n} &= \phi && \text{on } \Gamma_N, && \text{(5.1c)}\\
u &= 0 && \text{on } \Gamma_D, && \text{(5.1d)}
\end{aligned}
$$

where we note that we do not assume $\boldsymbol{A}(x) \in \mathbb{R}^{d\times d}$ to be symmetric here, and that additional functions $\alpha, \gamma \in L^2(\Gamma_R)$, and $\phi \in L^2(\Gamma_N)$ are given. This, in particular, encompasses the linear model problems from Chapters 3–4, and even the nonlinear setting of Chapter 2, where we linearize the nonlinear goal to obtain the computable linearized dual problem (2.9). We stress that this is not specific to the situation presented here, but to all nonlinear problems: In order to solve them on a computer they are linearized, leading to one or more problems of the form (5.1) to be solved.

Implementationally, to cover all cases mentioned above, it must be possible ot let the coefficients in (5.1) depend on the spatial variable $x \in \Omega$ as well as on FEM functions, even for higher-order FEM discretizations. Existing software that is able to tackle this problem can be divided into two categories, which turn out to be both unsatisfactory for our purpose of thoroughly validating our theoretical results.

Large general-purpose libraries

The first category is made up of widely known and well established FEM libraries like deal.II, FEniCS, Dune, and NGSolve; see [ABD+21; ABH+15; BBD+21; Sch14]. They have enormous codebases that are often conglomerates of parts in different (compiled) languages like C/C++, Fortran, or assembly for efficiency, which is often their most pronounced and important feature. Most of them employ object oriented programming (OOP) to structure these vast amounts of source code and keep it maintainable. For these reasons, they are usually not very accessible, in the sense that the average user can obtain a near complete understanding of most code parts; however, an unproportional share of the blame for this is usually assigned to OOP, entailing its bad reputation of overengineered and convoluted code.

Whereas the assessment on efficiency and accessibility is usually quite clear, it is more complicated for flexibility. In recent years, some FEM packages have introduced easy-to-use interfaces in a scripting language like Python, which often greatly simplifies implementation of a specific problem if it is in the scope of the library. However, extending the library code to implement new methods that are not already provided is still tedious; see, e.g., [BHL+21] for an extension of FEniCS to AFEM. This is often further complicated by the scripting language interface, since this adds another obfuscating layer to the library.

Small specialized research codes

The other category are small in-house research codes that are often specialized to only compute one specific method or problem type and written in a high level scripting language like Python or MATLAB [Che09; FGS15; FPW11]. Therefore, they are usually very accessible in the above sense, which is an essential feature for numerical analysis.

However, researchers are often content if their code additionally is either flexible or efficient, which implies increased computation time or implementation effort for future projects. In this chapter, we argue on the example of MooAFEM that, despite its bad reputation, OOP allows to have FEM software that combines all desired properties: By use of well-known object oriented design principles [GHJV95] and coding best-practices, our MooAFEM package is

- accessible, since it uses a consistent and descriptive naming scheme together with a lean class hierarchy that closely resembles the mathematical structure of FEM and is well documented;
- flexible, since the classes are loosely coupled and cooperate only through well-defined interfaces, which allow for easy extension; modification is also facilitated by an extensive collection of unit tests, which helps catching errors early on;
- efficient, since all actual computations are internally done on large chunks of data by built-in MATLAB routines, maximizing the gain through parallelization, and frequently used data (like the underlying mesh structure) is cached for inexpensive reuse.

Our MooAFEM package thus aims to support the use of OOP and the reuse of code in AFEM research in order to keep implementation and computation time small, while still having reliable and well-tested code.

Chapter outline

The first part of this chapter, Section 5.2, is dedicated to list the advantages of object oriented design for FEM in MATLAB. These advantages are showcased on a schematic algorithm for solving a simple Poisson problem with our MooAFEM package. The remainder of the chapter then deals with implementation of the package: Section 5.3 explains the three modules that make up MooAFEM and goes into various levels of detail for all contained classes. Section 5.4 lays out some finer details of the implementation, such as the underlying memory layout of the data structures. In the concluding part of this thesis, Section 5.5, we finally showcase what MooAFEM is capable of: higher-order AFEM and GOAFEM for linear problems as well as iterative linearization of nonlinear equations, all of which are needed in the preceding chapters.

5.2 Adaptive algorithm and importance of OOP

To solve equation (5.1), we employ FEM with underlying triangulation $\mathcal{T}_H$ of Ω. To this triangulation, we associate the FEM space $\mathcal{S}^p(\mathcal{T}_H) := \mathcal{P}^p(\mathcal{T}_H) \cap H^1(\Omega)$ with

$$\mathcal{P}^p(\mathcal{T}_H) := \big\{ v \in L^2(\Omega) \,\big|\, v|_T \text{ is a polynomial of degree } p \text{ for all } T \in \mathcal{T}_H \big\}.$$

With $H^1_D(\Omega) := \{v \in H^1(\Omega) \,|\, v|_{\Gamma_D} = 0\}$, we set $\mathcal{S}^p_D(\mathcal{T}_H) := \mathcal{S}^p(\mathcal{T}_H) \cap H^1_D(\Omega)$. The discrete weak form of (5.1) then reads: Find $u \in \mathcal{S}^p_D(\mathcal{T}_H)$ such that

$$\begin{aligned} a(u_H, v_H) &:= \int_\Omega \boldsymbol{A}\nabla u_H \cdot \nabla v_H + \boldsymbol{b} \cdot \nabla u_H v_H + c\, u_H v_H \,\mathrm{d}x + \int_{\Gamma_R} \alpha\, u_H v_H \,\mathrm{d}s \\ &= \int_\Omega f v_H + \boldsymbol{f} \cdot \nabla v_H \,\mathrm{d}x + \int_{\Gamma_N} \phi\, v_H \,\mathrm{d}s + \int_{\Gamma_R} \gamma\, v_H \,\mathrm{d}s =: F(v_H) \quad \text{for all } v_H \in \mathcal{S}^p_D(\mathcal{T}_H). \end{aligned} \tag{5.2}$$

Our software is intended to solve equation (5.2) by adaptive finite element methods (AFEM), an abstract form of which is presented in the following [BR03; Ste07].

Algorithm 5A

Input: Initial triangulation $\mathcal{T}_0$ of Ω
Loop: For $\ell = 0, 1, \ldots$ do

(i) Solve equation (5.2) to obtain u_ℓ

(ii) Estimate the error by computing refinement indicators $\eta_\ell(T)$ for all $T \in \mathcal{T}_\ell$

(iii) Mark elements $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ based on η_ℓ

(iv) Refine marked elements to obtain $\mathcal{T}_{\ell+1} := \texttt{refine}(\mathcal{T}_\ell, \mathcal{M}_\ell)$

Output: Sequence of solutions u_ℓ

A realization of the abstract adaptive Algorithm 5A is shown in the code snippet in Listing 5.1 below. It computes the lowest-order FEM solution of the Poisson equation $-\Delta u = 1$ on the unit square $\Omega := (0, 1)^2$ with homogeneous Dirichlet data $u = 0$ on the boundary $\Gamma_D := \partial\Omega$.

```
1  mesh = Mesh.loadFromGeometry('unitsquare');
2  fes = FeSpace(mesh, LowestOrderH1Fe);
3  u = FeFunction(fes);
4  blf = BilinearForm(fes);
5  blf.a = Constant(mesh, 1);
6  lf = LinearForm(fes);
7  lf.f = Constant(mesh, 1);
8
9  while mesh.nElements < 1e6
10   A = assemble(blf);
11   F = assemble(lf);
12   freeDofs = getFreeDofs(fes);
13   u.setFreeData(A(freeDofs,freeDofs) \ F(freeDofs));
14
15   hT = sqrt(getAffineTransformation(mesh).area);
16   qrEdge = QuadratureRule.ofOrder(1, '1D');
17   qrTri = QuadratureRule.ofOrder(1, '2D');
18   volumeRes = integrateElement(CompositeFunction(@(x) x.^2, lf.f), qrTri);
19   edgeRes = integrateNormalJump(Gradient(u), qrEdge, @(j) j.^2, {}, ':');
20   edgeRes(mesh.boundaries{:}) = 0;
21   eta2 = hT.^2.*volumeRes + hT.*sum(edgeRes(mesh.element2edges), Dim.Vector);
22
23   marked = markDoerflerSorting(eta2, 0.5);
24   mesh.refineLocally(marked, 'NVB');
25 end
```

Listing 5.1: Adaptive P1-FEM for Poisson problem $-\Delta u = 1$ on $\Omega = (0,1)^2$ subject to $u = 0$ on $\partial\Omega$.

In Listing 5.1, lines 1–7 are setup code that initializes all necessary data structures. Lines 10–13 solve equation (5.2), i.e., they realize Algorithm 5A(i). For Algorithm 5A(ii), the error is then estimated in lines 15-21 by local contributions of the residual *a posteriori* error estimator [Ver13]

$$\eta_\ell(T)^2 := |T|\,\|f\|^2_{L^2(T)} + |T|^{1/2}\|[\![\nabla u \cdot \boldsymbol{n}]\!]\|^2_{L^2(\partial T\cap\Omega)} \quad \text{for all } T \in \mathcal{T}_\ell.$$

In line 23, Algorithm 5A(iii) is executed by the so-called *Dörfler* marking criterion [Dör96]

$$\theta \sum_{T\in\mathcal{T}_\ell} \eta_\ell(T)^2 \le \sum_{T\in\mathcal{M}_\ell} \eta_\ell(T)^2 \quad \text{with } \theta = 0.5.$$

Finally, line 24 corresponds to Algorithm 5A(iv) and uses *newest vertex bisection* (NVB) [Ste08] to refine (at least) the marked elements.

In the following, we assume that there is a fixed initial triangulation $\mathcal{T}_0$ of Ω. We write $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$ if $\mathcal{T}_h$ is obtained from $\mathcal{T}_H$ by a finite number of refinement steps, i.e., there exist $\mathcal{T}_1, \dots, \mathcal{T}_n$ and $\mathcal{M}_i \subseteq \mathcal{T}_i$ with $\mathcal{T}_1 = \mathcal{T}_H$, $\mathcal{T}_n = \mathcal{T}_h$, and $\mathcal{T}_{i+1} := \texttt{refine}(\mathcal{T}_i)$ for all $i = 1, \dots, n-1$. Furthermore, we abbreviate $\mathbb{T} := \mathbb{T}(\mathcal{T}_0)$. Finally, we denote the set of edges in the mesh $\mathcal{T}_H \in \mathbb{T}$ by $\mathcal{E}_H$ and the set of vertices by $\mathcal{V}_H$.

Remark 5.1. *We note that all other coefficients from* (5.1) *can be set just as easily as in line 6–7 of Listing 5.1. The corresponding members of* `blf` *and* `lf` *are the following:*

- `blf.a`, `blf.b`, `blf.c`, `lf.f`, *and* `lf.fvec` *for the data of* (5.1a)*;*
- `blf.robin` *and* `lf.robin` *for the data of* (5.1b)*;*
- `lf.neumann` *for the data of* (5.1c).

The types of functions that can be used are described in Section 5.3 below. There, also quadrature rules are presented, which can be assigned to the corresponding members `qra`, `qrb`, `qrc`, *and* `qrRobin` *for* `blf` *as well as* `qrf`, `qrfvec`, `qrRobin`, *and* `qrNeumann` *for* `lf`.

5.2.1 Necessity of OOP in MATLAB FEM

The use of OOP is not mandatory in MATLAB, but it facilitates code that is powerful yet concise and flexible. In particular, the code from Listing 5.1 relies heavily on OOP due to some peculiarities of the MATLAB programming language. Most notably, the referencing mechanics of MATLAB differ greatly from languages like C/C++, Java, or Python, where referencing variables is either done by default, or explicitly. We proceed by outlining the language specific topics that are necessary to understand how MooAFEM works.

Function argument passing

MATLAB has a lazy copy policy for function arguments[1]: Arguments are generally passed by reference, but copied if they are modified within the function. However, no local copy is made of variables that are assigned to themselves by returning data:

```
1 function z = foo(x, y, z)
2   y(1) = x;
3   z = y + z;
4 end
5 z = foo(x, y, z)
```

In this example, `x` is passed by reference, `y` is copied because it is modified in line 2 (which is equivalent to passing the argument by value), and `z` is modified but not copied since the output value is again assigned to `z` in line 5.

FEM computations often re-use data throughout many sub-tasks; e.g., element areas are used in assembly of FEM systems, *a posteriori* error estimation, and even interpolation, via integration routines. With the passing mechanics outlined above, there are two options to approach this issue: First, data can be recomputed in each of the sub-tasks, yielding a clear public API, but causing superfluous operations. Second, the data can be precomputed explicitly and held in memory. In this case, the data management has to be done on the highest level of code by the user, or computations have to be grouped to respect data availability; both lead to a public API that is error-prone and inflexible. It is therefore highly desirable to have proper call by reference mechanics, which, in MATLAB, are only available through OOP.

Value vs. handle classes

Value classes cannot change their state[2]. This is the default for objects in MATLAB. The reason for this is the argument passing mechanism described above. In fact, except for pathological cases, the method invocation `obj.method(...)` and the function call `method(obj, ...)` are equivalent[3]. Hence, the copy-on-modify mechanics for function arguments also apply to instances of (value) classes. To change fields of already existing objects, a method must return an object, which overwrites the original object. However, classes that are derived from the abstract `handle` class can overcome the limitations of value classes in the sense that they allow for modifications of state through methods:

[1] https://mathworks.com/help/matlab/matlab_prog/avoid-unnecessary-copies-of-data.html
[2] https://mathworks.com/help/matlab/matlab_oop/comparing-handle-and-value-classes.html
[3] https://mathworks.com/help/matlab/matlab_oop/method-invocation.html

```
classdef MyClass
  % ...
  function obj = modify(obj, value)
    obj.field = value;
  end
end
obj = MyClass();
obj = modify(obj, 1);
```

```
classdef MyClass < handle
  % ...
  function modify(obj, value)
    obj.field = value
  end
end
obj = MyClass()
obj.modify(1)
```

Both code snippets result in `obj.field` being equal to one.

Also, handle classes can inherently be referenced: assigning an instance of a handle class to a variable does not copy the underlying data, but only assigns a reference. Finally, handle classes have native support for the *observer pattern*, which is one of the central design elements of our code; see Section 5.3.1 for details.

In order to communicate clearly, where methods can possibly alter the state of an object, we adhere to a coding best-practice called *command query separation* throughout documentation and examples: Commands, i.e., methods that alter the internal state of the calling object, are called with dot-notation `obj.method(...)` and never return data; queries, i.e., methods that do not alter the state of the calling object but may return data, are called with function call-notation `method(obj, ...)`.

Emulating statically typed languages

One of the disadvantages of dynamically typed languages like MATLAB is the lack of automatic type checks and function overloading. By using classes to represent (even trivial) data, this behavior can be emulated to a certain degree[4].

Type checks can be automated by function argument validation, which was introduced recently in MATLAB. This is achieved by an optional `arguments`-block after the function head that performs some checks on all input arguments of that function. In particular, it can check class, dimension, and values of the input. This greatly improves usability and error mitigation.

Since method invocation `obj.method(...)` and function call `method(obj, ...)` are virtually equivalent, function dispatch in MATLAB goes by the first argument of a function. This emulates function overloading at least in the first argument; e.g., in Listing 5.1 (solving the Poisson problem), one can readily use `plot(mesh)` and `plot(u)` to plot the mesh and the FEM solution. While this might be seen as syntactic sugar, it also greatly aids debugging.

Vectorization

One of the key features of MATLAB are efficiently vectorized built-in linear algebra operations. The usual local FEM formulation (i.e., on single elements and edges), however, does not allow for performance improvements through vectorization and parallelization, which are most pronounced if used with sufficiently large arrays to compensate for possible overhead. It is therefore desirable to defer actual computation as long as possible to make optimal use of MATLAB built-in routines.

Our code provides several well-defined interfaces very close to the natural (local) formulation of FEM which can be used to extend the functionality; see Section 5.3.3. The referencing mechanics of

[4] https://martinfowler.com/bliki/ValueObject.html

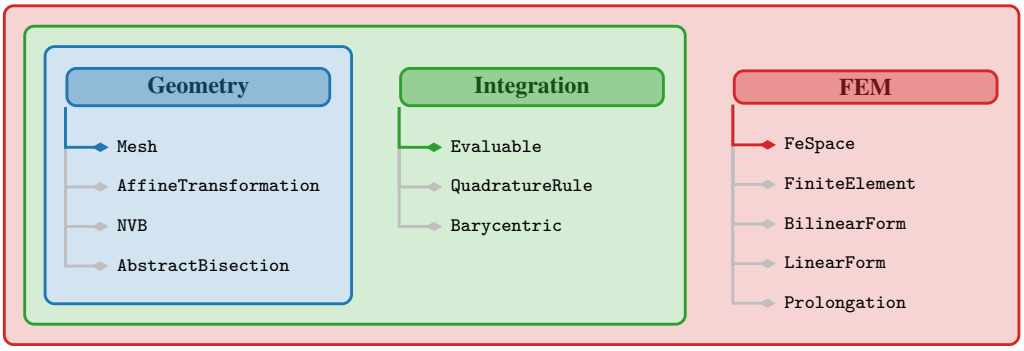


Figure 5.1: Overall structure of the presented software package. Shown are all classes of the software package, subdivided into three modules. The most important class of each submodule is at the top of the respective list.

handle classes then allow the internal generation of global data from this local code by pre-existing routines and passing it to the built-in routines.

5.3 Code structure

Some FEM packages have a huge code-base and, necessarily, a cleverly designed class hierarchy that may require significant effort to understand in many cases. Since one of our aims is that our code is easy to modify and extend, we strive for a relatively small code-base that nevertheless covers the most widespread demands of academic FEM software. Thus, the core of our software package is made up of only twelve classes and interfaces that can be roughly divided into three modules: Geometry, integration, and FEM. This partition is shown in Figure 5.1.

What follows is a description of the separate core classes as well as their inter relationship to one another. Following the partition of the code, our presentation is divided into three parts.

5.3.1 Module geometry

The geometry module can be used entirely on its own. It handles mesh generation as well as local mesh refinement.

Mesh representation

As the underlying mesh is the cornerstone in any adaptive FEM algorithm, the `Mesh` class is the central building block of MooAFEM. It consists of all data that is important for the digital representation of a 2D triangulation: coordinates, edges, elements, connectivity of edges and elements, edge orientation, and boundary information. The precise data structures for storing this information are described in Section 5.4 below.

Here, we focus on the role of the class in the code compound. Most other classes need a `Mesh` to function properly and, hence, store a reference to a suitable instance. Validity of data is strongly coupled to the underlying mesh: as soon as the mesh changes, derived data (e.g., geometric information as well as the data used in FEM computations) may be invalid. It is therefore of vital

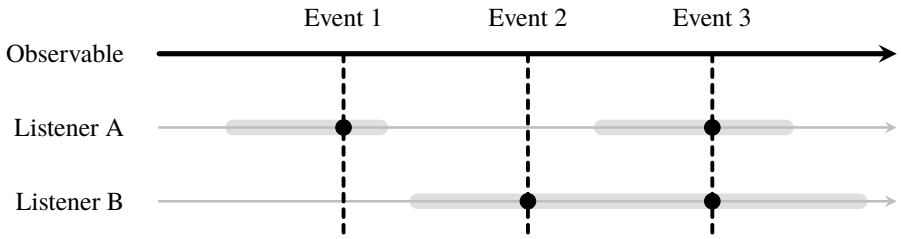


Figure 5.2: Schematic functionality of the observer pattern. The timelines represent the lifetimes of the observable object and the listeners. Two listeners are temporarily registered to receive events (highlighted in gray). If the listeners are registered during the broadcast of an event (dashed lines), some internal reaction is triggered (bold dots). For each additional observable or event type of the same observable, a separate graph is needed.

importance that changes in the mesh are made public to every object that stores a reference to it, contrary to the usual flow of information in object oriented code. Also, objects that do not explicitly depend on a mesh may need to act if the mesh changes. A well-known remedy for this issue is presented in the next section.

Observer pattern

The observer pattern [GHJV95] is used to broadcast *events* from a central object (the *observable*) to other objects (the observers, termed *listeners* in MATLAB), which can react to the event in a predefined way and which can, at runtime, register and de-register to receive such events[5]; see Figure 5.2. The events signal, e.g., a change of state of the observable. All classes that derive from `handle` in MATLAB automatically implement the interfaces necessary to be the source of events. In particular, our `Mesh` class broadcasts events that signal a change in the mesh (e.g., after refinement) as well as completed computation of bisection data to signal imminent refinement; this is covered in the next two sections.

Refinement

Mesh refinement is implementationally divided into bisection of single elements $T \in \mathcal{T}_H$ and coordination of bisections on the whole mesh $\mathcal{T}_H$ to obtain $\mathcal{T}_h = \texttt{refine}(\mathcal{T}_H, \mathcal{M}_H)$. In the class `AbstractBisection`, possible bisections of a single triangle $T \in \mathcal{T}_H$ are encoded. Subclasses of this class manage the generation of all `Mesh` data structures (see Section 5.4 below) in the passage from $T \in \mathcal{T}_H$ to its children $\{T' \in \mathcal{T}_h \mid T' \subseteq T\}$ in $\mathcal{T}_h$. The subclasses that are currently implemented are shown in Figure 5.3 and build on the implementation of [FPW11].

Local bisections are combined in mesh refinement routines derived from newest vertex bisection (NVB) [KPP13; Ste08], in the course of which all elements $T \in \mathcal{T}_H$ are assigned a subclass of `AbstractBisection`. We follow the iterative algorithm given in [KPP13] for NVB, which terminates regardless of the mesh $\mathcal{T}_H$ under consideration. Given a subset $\mathcal{M}_H \subseteq \mathcal{T}_H$ of marked elements, the abstract scheme is the following:

[5] https://mathworks.com/help/matlab/matlab_oop/learning-to-use-events-and-listeners.html

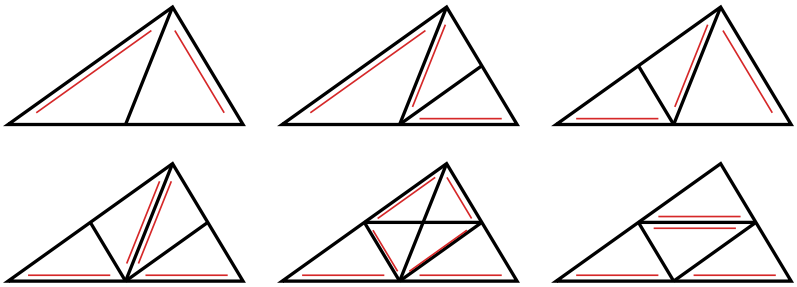


Figure 5.3: Implemented bisection methods (top left to bottom right): `Bisec1`, `Bisec12`, `Bisec13`, `Bisec123`, `Bisec5`, and `BisecRed`. The refinement edge of the parent triangle is the bottom line, those of the children are highlighted by parallel lines.

(i) Determine all edges in $\mathcal{E}_H$ that have to be bisected in order to bisect all elements in $\mathcal{M}_H$ by a given bisection rule.

(ii) Compute the *mesh closure*, i.e., determine all edges that *additionally* have to be refined to eliminate hanging vertices.

(iii) For every element in $\mathcal{T}_H$ determine which bisection method to employ, based on the marked edges.

(iv) Execute bisection of all elements according to their assigned bisection methods.

The first three steps are executed by subclasses of `NVB`, the fourth step is carried out by the mesh itself. The rationale behind this splitting is that, before the fourth step, all necessary information to carry out mesh refinement is already known; thus, this provides a natural hook for other classes to harness this information, e.g., prolongation operators or multi-grid solvers.

Several realizations of the abstract scheme presented above are implemented in our software package: `NVB1`, `NVB3` (= `NVB`), `NVB5`, and `RGB`, which are outlined in [FPW11], as well as `NVBEdge`, which is an edge-driven refinement strategy used in [DKS16; IP21].

Additional geometric information

For triangular meshes, all elements can be obtained by affine transformations of the so-called reference triangle $T_{\mathrm{ref}} := \mathrm{conv}\{(0,0),(1,0),(0,1)\}$, i.e., for all $T \in \mathcal{T}_H$, there exists an affine diffeomorphism $F_T : T_{\mathrm{ref}} \to T$. Many computations in FEM need additional geometric information based on this diffeomorphism. In particular, the transposed inverse $\mathrm{D}\, F_T^{-\intercal}$ and the determinant $\det \mathrm{D}\, F_T = 2|T|$ of its derivative are of utmost importance in integration and assembly routines. Together with the length $\mathrm{d}s = |E|$ of each edge $E \in \mathcal{E}_H$ and its unit-normal vector $\boldsymbol{n}_E$, these data are stored in instances of `AffineTransformation`, which take a `Mesh` to construct.

For performance reasons, instances of `AffineTransformation` are requested from the mesh and are cached, i.e., computed at the first request, then stored as a reference within the mesh object and returned if further requests occur.

5.3.2 Module integration

Integration is a crucial part of FEM assembly and postprocessing (e.g., *a posteriori* error estimation). The module defines two classes that encapsulate data for numerical integration (also termed *quadrature*) routines and one to encapsulate function evaluation with a unified interface. This module can be used only in conjunction with the geometry module.

Barycentric coordinates

We denote all evaluation points and quadrature nodes in barycentric coordinates. On a triangle $T = \operatorname{conv}\{z^{(1)}, z^{(2)}, z^{(3)}\}$ and a point $x \in T$, we denote by $\lambda \in [0,1]^3$ the barycentric coordinates of x, determined by the equations

$$\sum_{i=1}^{3} \lambda_i = 1 \quad \text{and} \quad \sum_{i=1}^{3} \lambda_i z^{(i)} = x.$$

If the triangle T is non-degenerate, λ is unique. In MooAFEM, 2D barycentric coordinates are implemented in the class `Barycentric2D`, which is derived from the abstract class `Barycentric`. For convenience, this class stores a collection of barycentric coordinates.

The concept of barycentric coordinates can be generalized to d-simplices with $d \geq 1$ such that $\lambda \in [0,1]^{d+1}$. In particular, in the case $d = 1$ we have $\lambda \in [0,1]^2$, which is used for evaluation points and quadrature rules on edges and implemented in the class `Barycentric1D`. For any point $x \in T_{\mathrm{ref}} = \operatorname{conv}\{(0,0),(1,0),(0,1)\}$ in the reference triangle, the barycentric coordinates are $\lambda = (1 - x_1 - x_2, x_1, x_2)$; for any point $x \in E_{\mathrm{ref}} = [0,1]$ on the reference edge, they are $\lambda = (x, 1-x)$.

Denoting all function evaluation points in barycentric coordinates allows for triangle independent representation. This is reflected by `Evaluable.eval` below, which is the core of our vectorization efforts and paramount for the efficiency of MooAFEM.

Quadrature data

To approximate the integral of a (possibly vector-valued) function f on a triangle $T \in \mathcal{T}_H$, we employ numerical quadrature:

$$\int_T f(x)\,\mathrm{d}x \approx |T| \sum_{k=1}^{N} \omega_k f\big(x(\lambda^{(k)}, T)\big), \tag{5.3}$$

where $(\lambda^{(k)})_{k=1}^N$ is a collection of barycentric coordinates, $x(\lambda^{(k)}, T)$ is the Cartesian coordinate corresponding to the k-th barycentric coordinate on T, and $(\omega_k)_{k=1}^N$ are weights with $\sum_{k=1}^N \omega_k = 1$.

Barycentric coordinates and weights make up a `QuadratureRule` object. Quadrature rules can either be constructed explicitly by giving barycentric coordinates and weights, or by the static method

```
qr = QuadratureRule.ofOrder(order, [dim]);
```

The optional string argument `dim` is used to distinguish between `'1D'` and `'2D'` quadrature rules, the default being `'2D'`. For 1D, suitable Gauss-rules are implemented. For 2D, symmetric quadrature rules up to order 5 are implemented [ZCL09]. Higher order quadrature rules on triangles use tensorized Gauss-rules on $[0,1]^2$ and the Duffy transform $\Phi\colon [0,1]^2 \to T_{\mathrm{ref}}, \Phi(s,t) = \big(s, t(1-s)\big)$; [Duf82].

Function evaluation

The core of the integration module is a wrapper for functions, the abstract `Evaluable` class:

```
classdef Evaluable < handle
  properties (Abstract, GetAccess='public', SetAccess='protected')
    mesh
  end
  methods (Abstract, Access='public')
    eval(obj, bary, idx)
  end
end
```

The abstract method `eval` is intended to evaluate the function at all points $x(\lambda^{(k)}, T)$ for all barycentric coordinates $\lambda^{(k)}$ in `bary` and all elements given by the index set `idx`.

By programming all routines (e.g., integration, plotting, finite element assembly) only to this interface, one can wrap virtually anything in a suitable subclass of `Evaluable` and readily use the predefined routines. The most important classes that implement this interface are:

- `Constant`: Efficiently wraps constant functions in the `Evaluable` interface.
- `MeshFunction`: General functions $f : \Omega \to \mathbb{R}^n$ for some $n \in \mathbb{N}$.
- `FeFunction`: FEM functions, e.g., $u \in \mathcal{S}^p(\mathcal{T}_H)$.
- `Gradient`, `Hessian`: Element-wise gradient ∇u and Hessian $\nabla^2 u$ for FEM functions.
- `CompositeFunction`: Arbitrarily combine any of the above; see the following explanation.

In particular, all of the above can be used as coefficients in (5.1).

Especially powerful is the subclass `CompositeFunction`, which uses the composite pattern (see, e.g., [GHJV95]):

```
f = CompositeFunction(funcHandle, funcArgument1, ..., funcArgumentN)
```

This class takes a function handle and one `Evaluable` for every argument of the function handle. E.g., xu^2 can be implemented by

```
f = CompositeFunction(@(x,u) x.*u.^2, ...
      MeshFunction(mesh, @(x) x(1,:,:)), FeFunction(fes));
```

If `f` is evaluated, first all arguments are evaluated, then the resulting data is processed by the function handle. By polymorphism, the `Evaluable` arguments can be of any subclass. This allows to define complex functions that still can be evaluated efficiently, since evaluation of the function handle is deferred until the data for all requested elements and quadrature nodes is available. In this way, the vectorization capabilities of MATLAB are used to full extent.

Quadrature routines

There are several routines for quadrature implemented in our MooAFEM package:

- `integrateElement`: Integration on elements; cf. (5.3).
- `integrateEdge`: An analogue over edges. This can only be used for subclasses of `Evaluable` that implement the method `evalEdge`. Edge evaluation is not well-defined for some functions that are not continuous across edges (e.g., element-wise polynomials).

- `integrateJump`, `integrateNormalJump`: Integrate $[\![\cdot]\!]$ and $[\![(\cdot)\cdot\boldsymbol{n}]\!]$ over edges, respectively.

All quadrature routines take an `Evaluable`, a `QuadratureRule`, and an optional set of indices that corresponds to a subset of elements or edges on which the quadrature should be evaluated. The routines handling (normal) jumps take additional arguments: a function handle, a cell array of `Evaluable`, and edge indices.

```
int = integrateJump(f, qr, funcHandle, funcArg, idx)
```

This is used for post-processing the jump with the first argument of the function handle being reserved for the jump; i.e., the routine works roughly as follows: First, compute the jump by `jump = [[f]]` (or `jump = [[f * n]]`). Then, all additional `Evaluables` are evaluated on the edges indicated by `idx`, where also the value of the jump is updated by the function handle:

```
val = {evalEdge(funcArg{1}, idx), ..., evalEdge(funcArg{N}, idx)};
jump(idx) = functionHandle(jump(idx), val{1}, ..., val{N});
```

Multiple such triplets `funcHandle`, `funcArg`, `idx` for post-processing are allowed and sequentially applied as in the above listing, one after another.

5.3.3 Module FEM

This last module uses the classes presented in the last sections to conveniently represent FEM functions and efficiently assemble FEM data.

Finite element spaces

Finite elements are usually defined on the reference triangle T_{ref}, everything else follows from using the affine transformation F_T for every $T \in \mathcal{T}_H$. This viewpoint is represented accordingly in our code. The abstract class `FiniteElement` asks to implement evaluation of all basis functions (as well as their gradient and their Hessian) on T_{ref}. Furthermore, evaluation on a reference edge (if applicable) and the connectivity of the degrees of freedom (DOFs), i.e., how the finite element couples across edges and vertices, have to be specified.

The class `FeSpace` combines the local information of finite elements with the global geometry of the mesh. It takes a `Mesh` and a `FiniteElement` to assemble lists of global DOFs per element and edge, as well as free DOFs, i.e., DOFs that do not lie on Γ_D.

So far, Lagrange finite elements of arbitrary order are implemented; both $H^1(\Omega)$-conforming (i.e., $\mathcal{S}^p(\mathcal{T}_H) = \mathcal{P}^p(\mathcal{T}_H) \cap H^1(\Omega)$) and $L^2(\Omega)$-conforming (i.e., $\mathcal{P}^p(\mathcal{T}_H)$). The underlying implementation uses Bernstein–Bézier polynomials [AAD11]. For lowest-order finite elements (both continuous and discontinuous), additional optimized implementations are available.

FEM system assembly

Let $(\varphi_k)_{k=1}^N$ be a basis of $\mathcal{S}^p(\mathcal{T}_H)$. Responsible for the assembly of the data

$$A_{ij} := \int_\Omega \boldsymbol{A}\nabla\varphi_j \cdot \nabla\varphi_i + \boldsymbol{b} \cdot \nabla\varphi_j \varphi_i + c\,\varphi_j\varphi_i \,\mathrm{d}x + \int_{\Gamma_R} \alpha\,\varphi_j\varphi_i \,\mathrm{d}s, \tag{5.4a}$$

$$F_i := \int_\Omega f\varphi_i + \boldsymbol{f} \cdot \nabla\varphi_i \,\mathrm{d}x + \int_{\Gamma_N} \phi\,\varphi_i \,\mathrm{d}s + \int_{\Gamma_R} \gamma\,\varphi_i \,\mathrm{d}s \tag{5.4b}$$

are the classes `BilinearForm` and `LinearForm`, respectively. Both classes have fields for their respective coefficients and quadrature rules for each of the terms in (5.4); see Remark 5.1. The data in (5.4) are then obtained by calling the `assemble` methods of both classes.

Note that `Evaluable` is a handle class. Thus, no data must be copied to set (bi-)linear form coefficients. Furthermore, in situations where the coefficients change frequently, e.g., in the presence of iterative solvers for nonlinear PDEs, the coefficients of the (bi-)linear form can change between two consecutive calls of `assemble`. This results in much cleaner code since one does not need to re-set the coefficients.

Prolongation

It is often necessary to prolongate FEM functions (e.g., $u_H \in \mathcal{S}^p(\mathcal{T}_H)$) to a refined mesh (i.e., $\mathcal{S}^p(\mathcal{T}_h)$ for $\mathcal{T}_h \in \mathbb{T}(\mathcal{T}_H)$). This is handled by subclasses of the abstract class `Prolongation`. The implemented prolongation methods are `LoFeProlongation` and `FeProlongation` for lowest-order and general (in particular, higher-order) FEM functions, respectively. Note that the latter is not tailored to a specific finite element and, hence, its computational effort is slightly higher than that of the former. The syntax of prolongation of a function `u = FeFunction(fes)` on a finite element space `fes` from a coarse to a refined mesh is as follows:

```
P = FeProlongation(fes);
mesh.refineLocally(marked);
u.setData(P.prolongate(u));
```

The call to `P.prolongate` performs a matrix-vector multiplication with the (sparse) prolongation matrix `P.matrix`, which is set automatically whenever the mesh is refined, due to the events sent by the mesh; see Section 5.3.1 and the examples in Section 5.5.

5.4 Data structures

5.4.1 Mesh

The `Mesh` class stores information about coordinates, edges, and elements. In the following, let $n_\mathcal{V}, n_\mathcal{E}, n_\mathcal{T} \in \mathbb{N}$ denote the number of vertices, edges, and elements, respectively. The class has the following fields:

- `coordinates` ($2 \times n_\mathcal{V}$): Coordinates of mesh vertices. The entries `coordinates(1,i)` and `coordinates(2,i)` store the x- and y-coordinates of the i-th vertex, respectively. The order of the coordinates is provided by the user.

- `edges` ($2 \times n_\mathcal{E}$): Indices of vertices of all edges in the mesh. The i-th edge of the mesh starts at vertex `edges(1,i)` and ends at vertex `edges(2,i)`. The order is determined automatically from information provided in `elements`. Boundary edges are oriented such that the domain lies on its left; inner edges cannot be assigned a meaningful orientation and, therefore, it is chosen randomly.

- `elements` ($3 \times n_\mathcal{T}$): Indices of vertices of which elements are comprised. The i-th element is spanned by the vertices with indices `elements(:,i)`, where the order is counter-clockwise. The order of elements is provided by the user.

- `element2edges` ($3 \times n_{\mathcal{T}}$): Indices of edges which are contained in elements. The i-th element contains edges with indices `element2edges(:,i)`. The j-th edge `element2edges(j,i)` of the i-th element is the one between the vertices with indices `elements(j,i)` and `elements(mod(j+1,3)+1, i)` (but not necessarily in that order). This information is determined automatically.
- `boundaries` (cell array): Indices of all edges that form a specific part of the boundary. The cell `boundaries{i}` is a vector of edge indices that form the i-th boundary (if present). The boundary parts are provided by the user (see below), but the association with edge indices is done automatically.

The orientation of the normal vector from `AffineTransformation` follows the orientation of the edges: it points to the right of the edge. In particular, this means that the normal vectors on boundary edges point out of the domain.

5.4.2 Mesh construction

An instance of the `Mesh` class can be constructed by

```
mesh = Mesh(coordinates, elements, bndEdges);
```

Here, `coordinates` and `elements` have to be given as they are stored in the class (see above). The cell array `bndEdges` describes the boundary parts, where the i-th edge of the k-th boundary part lies between the vertices with indices `bndEdges{k}(1,i)` and `bndEdges{k}(2,i)`. This is necessary because the user does not know the internal edge numbering before construction. Special attention has to be payed to the correct orientation of the elements (counter-clockwise) and the edges on the boundary (domain on their left), because this is not checked by the constructor.

The arrays required by the constructor can be assembled and passed from a MATLAB script, or loaded from comma separated value files by the static method

```
mesh = Mesh.loadFromGeometry('<name>');
```

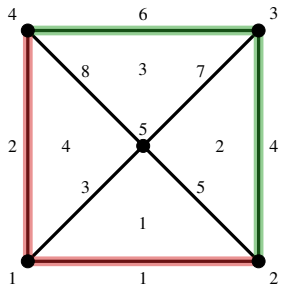
These files must be placed in a subdirectory `lib/mesh/@Mesh/geometries/<name>` and be named `coordinates.dat`, `elements.dat`, and `boundary<n>.dat`, where boundary parts are enumerated by $n \in \mathbb{N}$.

Mesh construction and data structures are showcased in Figure 5.4. Note that the orientation of the user-provided edges in `bndEdges` is preserved by the automatically generated field `edges`.

5.4.3 Array layout

The array layout is chosen such that the first three dimensions of arrays correspond to modeling concepts of finite elements; see Figure 5.5:

- **1. dimension (columns)**: This corresponds to the components of vector- or matrix-valued data. Matrices are stored column-wise.
- **2. dimension (rows)**: This corresponds to the different units of the mesh, i.e., elements, edges, or vertices.
- **3. dimension (pages)**: This corresponds to different barycentric coordinates, e.g., for numerical quadrature.



n	1	2	3	4	5	6	7	8
coordinates	0.0	1.0	1.0	0.0	0.5			
	0.0	0.0	1.0	1.0	0.5			
edges	1	4	1	2	2	3	3	4
	2	1	5	3	5	4	5	5
elements	1	2	3	4				
	2	3	4	1				
	5	5	5	5				
element2edges	1	4	6	2				
	5	7	8	3				
	3	5	7	8				

n	1	2
bndEdges{1}	1	4
	2	1
bndEdges{2}	2	3
	3	4
boundaries{1}	1	2
boundaries{2}	4	6

Figure 5.4: Example mesh on the unit square $(0,1)^2 \subset \mathbb{R}^2$ as well as corresponding data structures. Boundary part 1 (e.g., Γ_D) is marked in red, boundary part 2 (e.g., Γ_N) is marked in green.

The rationale behind this order is twofold. First, objects on the same element often need to be multiplied together. Hence, it is of advantage if the data representing these objects are continuous in memory. This is achieved by arranging them along the columns of the array, since MATLAB stores arrays in column-major order. Second, arrays that extend into the third dimension are somewhat clumsy to work with and hard to debug for programmers. Since evaluations on multiple barycentric coordinates occur mostly internally (e.g., for quadrature rule or plotting), arranging different barycentric coordinates along the third dimension minimizes exposure of the user to more-than-two dimensional arrays. Within MooAFEM, one can use the enumeration class `Dim` to access these dimensions by `Dim.Vector`, `Dim.Elements`, and `Dim.QuadratureNodes`, respectively.

As an illustrative example, consider the call

```
f = MeshFunction(mesh, @(x) x);
val = eval(f, bary);
```

which evaluates $f\colon \Omega \to \mathbb{R}^2\colon x \mapsto x$ on a collection `bary` of barycentric coordinates and a given mesh element-wise. The value stored in `val(i,j,k)` corresponds to $x_i(\lambda^{(k)}, T_j)$, i.e., the i-th component of `f` evaluated at the k-th barycentric coordinate on the j-th element. The matrix valued function

$$f\colon \Omega \to \mathbb{R}^{2\times 2}\colon x \mapsto \begin{pmatrix} 1x_1 & 3x_1 \\ 2x_1 & 4x_1 \end{pmatrix}$$

can be implemented by `f = MeshFunction(mesh, @(x) [1;2;3;4].*x(1,:,:))`. The corresponding evaluation `val(i,j,k)` is equal to $i \cdot x_1(\lambda^{(k)}, T_j)$, since matrices are stored column-wise. See also Figure 5.5 for a sketch of this memory layout.

5.4.4 Efficient linear algebra

According to the last section, 3D arrays are essentially interpreted as collections of matrices stored column-wise. To efficiently execute matrix operations within this memory layout, the function

```
C = vectorProduct(A, B, sizeA, sizeB);
```

is used. It computes the product of two 3D arrays `A` and `B`, where the first dimension is interpreted as matrix with given size `sizeA` and `sizeB`, respectively. In particular, for all admissible $i, j \in \mathbb{N}$, the output of above call satisfies

```
C(:,i,j) = reshape(A(:,i,j), sizeA) * reshape(B(:,i,j), sizeB);
```

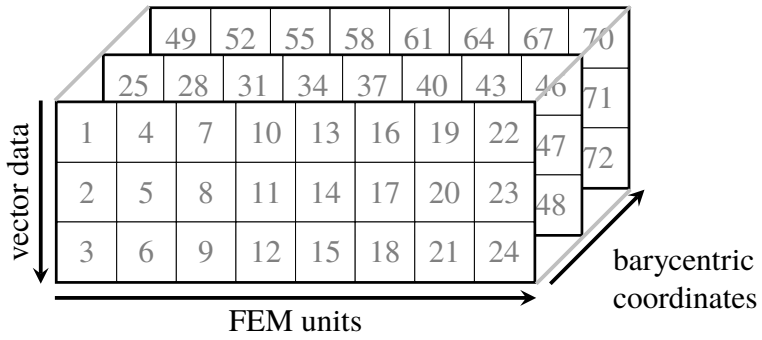


Figure 5.5: Illustration of the memory layout chosen in our implementation. The numbers indicate the order in which the items are stored in memory.

If either `sizeA` or `sizeB` is a column-vector, the corresponding factor in the above listing is transposed. For an example, consider the vector $\mathtt{v} := (1,2,3,4,5,6)^\intercal$, which, in our memory model, can be interpreted as

$$\mathtt{[2,3]}: \quad A := \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} \quad \text{or} \quad \mathtt{[3,2]}: \quad B := \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}.$$

Clearly there holds $A^\intercal \neq B$. Therefore, transposing the size is necessary to indicate transposition of the corresponding factor in the matrix product:

$$\begin{aligned} AB &= \mathtt{vectorProduct(v, v, [2,3], [3,2])}, \\ AA^\intercal &= \mathtt{vectorProduct(v, v, [2,3], [2,3]')}, \\ B^\intercal B &= \mathtt{vectorProduct(v, v, [3,2]', [3,2])}. \end{aligned}$$

The function `vectorProduct` thus enables all possible matrix operations within our memory layout in a convenient, yet very efficient way. In fact, this routine can also deal with arrays of arbitrary dimension, where extension of singleton dimensions is done automatically, as is common in MATLAB. Per default, the sizes are chosen such that the dot product

```
C(:,i,j) = A(:,i,j)' * B(:,i,j);
```

is computed, which is the most common application; this is also reflected in the name.

5.5 Examples

In this section, we discuss several extensions of the basic AFEM algorithm that is implemented in Listing 5.1. We do not claim that MooAFEM can deal with all FEM applications out of the box, but are convinced that our code structure makes extensions and modifications relatively easy.

5.5.1 Higher order AFEM with known solution

As a first example we consider the L-shape $\Omega := (-1,1)^2 \backslash \big([0,1] \times [-1,0]\big)$ with boundary parts $\Gamma_R = \emptyset$, $\Gamma_N := \big([0,1] \times \{0\}\big) \cup \big(\{0\} \times [-1,0]\big)$, and $\Gamma_D := \partial\Omega \backslash \Gamma_N$. With $\big(r(x), \varphi(x)\big)$ being the

polar coordinates of $x \in \mathbb{R}^2$, we prescribe the exact solution

$$u(x) := r(x)^{2/3} \sin(2\pi/3)$$

and note that it solves

$$-\Delta u = 0 \ \text{ in } \Omega, \quad \nabla u \cdot \boldsymbol{n} =: \phi \ \text{ on } \Gamma_N, \quad u = 0 \ \text{ on } \Gamma_D. \tag{5.5}$$

To solve (5.5) numerically with MooAFEM, only minor adjustments of the code from Listing 5.1 are necessary. First, obviously, the correct mesh must be loaded via

```
mesh = Mesh.loadFromGeometry('Lshape');
```

This geometry has two predefined boundary parts: The first (boundary index 1) is at the re-entrant corner (Γ_D), the second (boundary index 2) is everything else (Γ_N). Next, the finite element space has to be chosen accordingly with some $p \in \mathbb{N}$:

```
fes = FeSpace(mesh, HigherOrderH1Fe(p), 'dirichlet', 1);
```

This loads an implementation of $\mathcal{S}^p_D(\mathcal{T}_0)$. No further adjustments regarding the implementation of higher-order finite elements are necessary.

The next changes concern the coefficients of the linear form `lf`. Our problem (5.5) includes only the Neumann part of the right-hand side from (5.2), which can be implemented by the following listing.

```
lf.neumann = MeshFunction(mesh, @exactSolutionNeumannData);
lf.bndNeumann = 2;
function y = exactSolutionNeumannData(x)
  x1 = x(1,:,:);
  x2 = x(2,:,:);
  % determine boundary parts
  right  = (x1 > 0) & (abs(x1) > abs(x2));
  left   = (x1 < 0) & (abs(x1) > abs(x2));
  top    = (x2 > 0) & (abs(x1) < abs(x2));
  bottom = (x2 < 0) & (abs(x1) < abs(x2));
  % compute d/dn u
  [phi, r] = cart2pol(x(1,:,:), x(2,:,:));
  Cr = 2/3 * r.^(-4/3);
  Cphi = 2/3 * (phi + 2*pi*(phi < 0));
  dudx = Cr .* (x1.*sin(Cphi) - x2.*cos(Cphi));
  dudy = Cr .* (x2.*sin(Cphi) + x1.*cos(Cphi));
  y = zeros(size(x1));
  y(right)  = dudx(right);
  y(left)   =-dudx(left);
  y(top)    = dudy(top);
  y(bottom) =-dudy(bottom);
end
```

Here, the main part is the implementation of the Neumann derivative $\nabla u \cdot \boldsymbol{n}$, rather than making the function available to the assembly routines. Finally, we need to set quadrature rules of sufficiently high order for the corresponding terms of the (bi-)linear form:

```
blf.qra = QuadratureRule.ofOrder(max(2*p-2, 1));
lf.qrNeumann = QuadratureRule.ofOrder(2*p, '1D');
```

With these preparatory steps, the FEM system is solved by lines 10–13 of Listing 5.1.

Finally, the *a posteriori* indicators have to be adjusted to the current setting:

$$\eta_H(T)^2 = h_T^2 \|\Delta u_H\|^2_{L^2(T)} + h_T \big[\|[\![\nabla u_H \cdot \boldsymbol{n}]\!]\|^2_{L^2(\partial T \cap \Omega)} + \|(\nabla u_H - \nabla u) \cdot \boldsymbol{n}\|^2_{L^2(\partial T \cap \Gamma_N)} \big]. \tag{5.6}$$

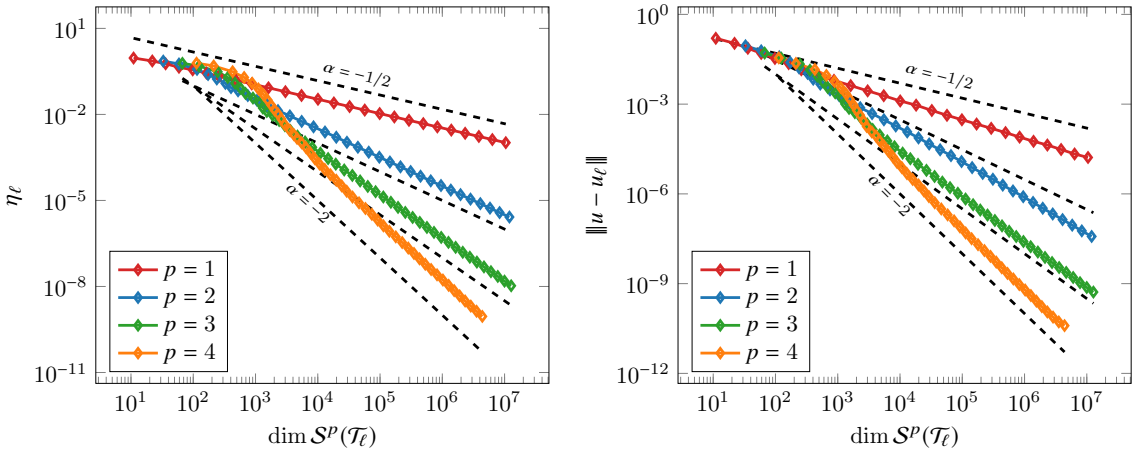


Figure 5.6: Error estimator η_ℓ (left) and energy error $|||u - u_\ell|||$ (right) over number of DOFs for problem (5.5) with different polynomial orders p.

Recall from Section 5.4.3 that matrices are stored column-wise in the first dimension of 3D arrays. Thus, the L^2-norm of the volume term can be computed by

```
f = CompositeFunction(@(D2u) (D2u(1,:,:) + D2u(4,:,:)).^2, Hessian(u));
qr = QuadratureRule.ofOrder(max(2*(p-2), 1));
volumeRes = integrateElement(f, qr);
```

The edgewise L^2-norms are a bit more involved. This is handled by

```
qr = QuadratureRule.ofOrder(p, '1D');
edgeRes = integrateNormalJump(Gradient(u), qr, ...
  @(j) zeros(size(j)), {}, mesh.boundaries{1}, ...
  @(j,phi) j-phi, {lf.neumann}, mesh.boundaries{2}, ...
  @(j) j.^2, {}, ':');
```

The syntax of `integrateNormalJump` is explained in Section 5.3.2. First, the jump $[\![\nabla u_H \cdot \boldsymbol{n}]\!]$ is computed on every edge. Then, since the edges on Γ_D (boundary index 1) do not contribute to the error estimator, the corresponding contributions are set to zero. Furthermore, the term $\nabla u \cdot \boldsymbol{n}$, which is stored in `lf.neumann`, is subtracted on Γ_N (boundary index 2). Finally, every edge contribution is squared to obtain the edgewise L^2-norms of (5.6).

The remainder of the code is analogous to the one presented in Listing 5.1. Note that virtually all changes are due to the different model problem and not for implementational reasons. Figure 5.6 shows the results obtained for $p = 1, 2, 3, 4$. Note that, from an implementational point of view, the polynomial degree $p \in \mathbb{N}$ can be chosen arbitrarily high. Computation times for the different parts of the adaptive algorithm are shown in Figure 5.7. In both the lowest and the higher order case, most time is spent for solution of the linear system. In the higher order case, one clearly sees that solving with the MATLAB backslash operator has more than linear complexity.

As a final note, the exact error of the finite element solution u_H can be easily computed by the following code snippet. Recall that `A` is the finite element matrix of the Laplacian.

```
uex = FeFunction(fes);
uex.setData(nodalInterpolation(MeshFunction(mesh, @exactSolution), fes));
deltaU = u.data - uex.data;
H1Error = sqrt(deltaU * A * deltaU');
```

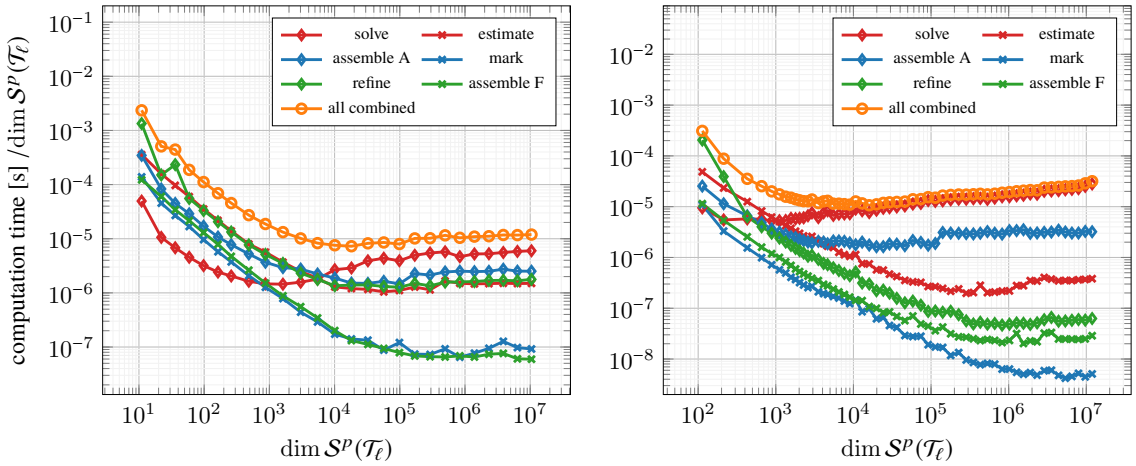


Figure 5.7: Computation time per DOF over number of DOFs for the different parts of the AFEM algorithm for problem (5.5) with polynomial degree $p = 1$ (left) and $p = 4$ (right).

```
5
6  function y = exactSolution(x)
7    [phi, r] = cart2pol(x(1,:,:), x(2,:,:));
8    phi = phi + 2*pi*(phi < 0);
9    y = r.^(2/3) .* sin(2/3 * phi);
10 end
```

5.5.2 Goal-oriented AFEM with discontinuous data

With $\Omega := (0,1)^2$ and $\Gamma_D := \partial\Omega$, we consider an example from [MS09]:

$$-\Delta u = -\operatorname{div} \boldsymbol{f} \ \text{ in } \Omega, \quad u = 0 \ \text{ on } \Gamma_D, \quad \text{where } \boldsymbol{f}(x) := \begin{cases} (1,0) & \text{if } x_1 + x_2 < 1/2, \\ (0,0) & \text{else.} \end{cases} \tag{5.7}$$

For most FEM software, discontinuous coefficients or data demand some caution: for quadrature nodes that lie on the discontinuity, evaluation is not well-defined. A first solution is to make the initial triangulation $\mathcal{T}_0$ of Ω resolve the regions of discontinuity. In our case, this can be achieved by uniform refinement using the RGB-strategy:

```
1 mesh = Mesh.loadFromGeometry('unitsquare');
2 mesh.refineUniform(1, 'RGB');
```

This is also needed for residual error estimators, since they are comprised of element-wise L^2-norms of $\operatorname{div} \boldsymbol{f}$ (which vanishes if the discontinuity is resolved by the mesh and is not defined otherwise).

A second problem is the jump term $[\![\cdot]\!]$ in the error estimators, since this is evaluated on edges, where the discontinuity now lies. This can be solved by interpolating the data to a non-continuous FEM space. To obtain vector-valued data, we first interpolate the non-continuous first component and then compose this with the vanishing second component, according to the memory layout presented in Section 5.4.3:

```
1 ncFes = FeSpace(mesh, LowestOrderL2Fe);
2 w = FeFunction(ncFes);
```

```
chiT = MeshFunction(mesh, @(x) sum(x, Dim.Vector) < 1/2);
w.setData(nodalInterpolation(chiT, ncFes));
lfF = LinearForm(fes);
lfF.fvec = CompositeFunction(@(w) [w;zeros(size(w))], w);
```

The nodal interpolation in the listing above only sets the data for `w` on the initial mesh $\mathcal{T}_0$. To have `w` available on refined meshes, we can repeat this interpolation process after every mesh refinement. A more efficient method is to use the prolongation class `P = LoFeProlongation(fes)` that is tailored specifically to lowest order L^2- and H^1-elements; see Section 5.3.3. Data for prolongation is computed automatically whenever the mesh is refined; see Section 5.3.1. After updating `w` by `w.setData(P.prolongate(w))`, the next call of `assemble(lfF)` already yields the updated right-hand side, since the coefficient `lfF.fvec` stores a reference to `w`.

In *goal-oriented* adaptive FEM (GOAFEM), we are interested in the *goal value* $G(u)$ for a linear functional

$$G\colon H^1_D(\Omega) \to \mathbb{R}, \quad G(v) = \int_\Omega \boldsymbol{g} \cdot \nabla v \,\mathrm{d}x \quad \text{with} \quad \boldsymbol{g}(x) := \begin{cases} (-1,0) & \text{if } x_1 + x_2 > 3/2, \\ (0,0) & \text{else.} \end{cases}$$

Approximating the goal value $G(u)$ is often more interesting in applications than approximating the solution u as a whole. To efficiently approximate the goal value in the spirit of Algorithm 5A, one introduces the so-called *dual* problem

$$-\Delta z = -\operatorname{div} \boldsymbol{g} \ \text{ in } \Omega, \quad z = 0 \ \text{ on } \Gamma_D, \tag{5.8}$$

which can be implemented and solved analogously to (5.7). Since solving is often the most time consuming part of AFEM, we can do this in parallel for (5.7) and (5.8):

```
rhs = [assemble(lfF), assemble(lfG)];
uz = A(freeDofs,freeDofs) \ rhs(freeDofs,:);
u.setFreeData(uz(:,1));
z.setFreeData(uz(:,2));
```

After solving (5.7) and (5.8) by FEM on a triangulation $\mathcal{T}_H$ to obtain the discrete solutions u_H and z_H, respectively, one can compute the *a posteriori* residual error estimators

$$\eta_H(T)^2 = h_T^2 \|\Delta u_H\|^2_{L^2(T)} + h_T \|[\![(\nabla u_H - \boldsymbol{f}) \cdot \boldsymbol{n}]\!]\|^2_{L^2(\partial T \cap \Omega)},$$
$$\zeta_H(T)^2 = h_T^2 \|\Delta z_H\|^2_{L^2(T)} + h_T \|[\![(\nabla z_H - \boldsymbol{g}) \cdot \boldsymbol{n}]\!]\|^2_{L^2(\partial T \cap \Omega)}$$

analogously to (5.6). The error in the goal functional is controlled by the estimator product

$$|G(u) - G(u_H)| \lesssim \Big[\sum_{T \in \mathcal{T}_H} \eta_H(T)^2\Big]^{1/2} \Big[\sum_{T \in \mathcal{T}_H} \zeta_H(T)^2\Big]^{1/2}, \tag{5.9}$$

for which different marking criteria have been analyzed [FPZ16]. Thus, the remaining implementation comprises only minor modifications of Listing 5.1. The upper bounds of this last equation for different polynomial orders p can be seen in Figure 5.8 and the resulting meshes for $p = 1, 3$ are shown in Figure 5.9.

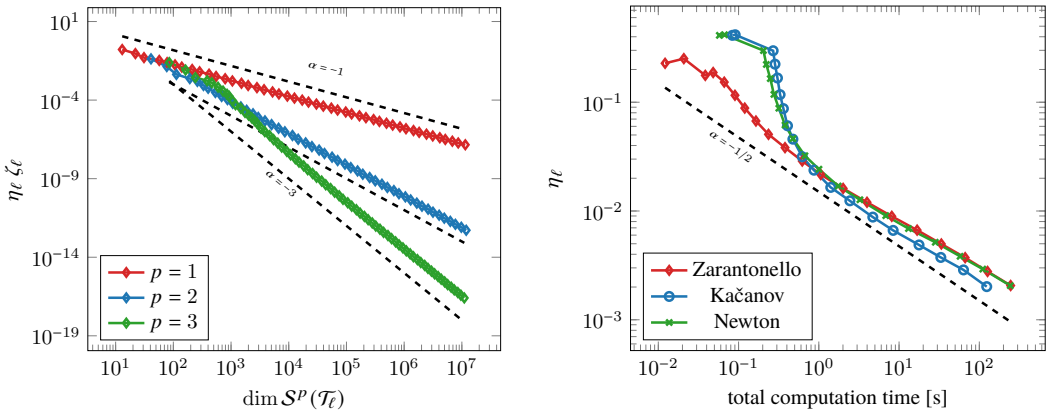


Figure 5.8: Left: Estimator for the goal error (5.9) over number of DOFs for problem (5.7) from Section 5.5.2 with different polynomial orders p. Right: Error estimator over total computation time for the linearization methods from Section 5.5.3.

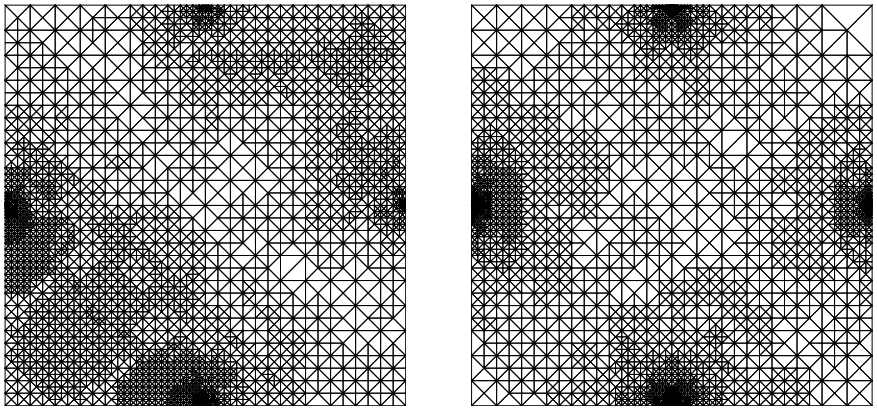


Figure 5.9: Meshes generated from the GOAFEM algorithm from Section 5.5.2 with polynomial orders $p = 1$ (left) and $p = 3$ (right).

5.5.3 Iterative solution of nonlinear equations

In this last example, we consider the L-shape $\Omega := (-1,1)^2 \backslash \big([0,1]\times[-1,0]\big)$ with Dirichlet boundary $\Gamma_D := \partial\Omega$. On this domain, we consider the quasi-linear problem

$$-\operatorname{div}\Big(\mu\big(|\nabla u|^2\big)\nabla u\Big) = 1 \ \text{ in } \Omega, \quad u = 0 \ \text{ on } \Gamma_D, \quad \text{with} \quad \mu(t) = 1 + \exp(-t). \tag{5.10}$$

This is a variation of an example given in [HPW21], where also adaptive iterative linearization techniques (AILFEM) for this class of problems are presented. With a given initial guess $u^0 \in H^1_D(\Omega)$, we consider the following linearizations:

(i) **Zarantonello iteration**: Let $\delta > 0$ be sufficiently small. Given $u^n \in H^1_D(\Omega)$, the next iterate $u^{n+1} \in H^1_D(\Omega)$ reads $u^{n+1} := u^n + \delta v$, where $v \in H^1_D(\Omega)$ solves

$$-\Delta v = \operatorname{div}\Big(\mu\big(|\nabla u^n|^2\big)\nabla u^n\Big) + 1.$$

(ii) **Kačanov iteration**: Given $u^n \in H^1_D(\Omega)$, the next iterate $u^{n+1} \in H^1_D(\Omega)$ solves

$$-\operatorname{div}\Big(\mu\big(|\nabla u^n|^2\big)\nabla u^{n+1}\Big) = 1.$$

(iii) **Newton iteration**: Given $u^n \in H^1_D(\Omega)$, the next iterate $u^{n+1} \in H^1_D(\Omega)$ reads $u^{n+1} := u^n + v$, where $v \in H^1_D(\Omega)$ solves

$$-\operatorname{div}\Big(\mu(|\nabla u^n|^2)\nabla v + 2\mu'(|\nabla u^n|^2)\big(\nabla u^n \otimes \nabla u^n\big)\nabla v\Big) = \operatorname{div}\Big(\mu\big(|\nabla u^n|^2\big)\nabla u^n\Big) + 1.$$

All iterations (i)–(iii) feature coefficients that depend in a nonlinear fashion on the previous iterate u^n. However, their implementation is relatively simple, owing to the uniform evaluation mechanics of the `Evaluable` interface, from which also `FeFunction` is derived. Assuming that, for some triangulation $\mathcal{T}_H$ of Ω, the previous iterates u^n_H correspond to the `FeFunction` instance `u`, the following code snippet acts as template for all three iterations with $u^0_H = 0$:

```
% set coefficients of blf & lf
u = FeFunction(fes);
u.setData(0);
v = FeFunction(fes);
freeDofs = getFreeDofs(fes);
while true
  A = assemble(blf);
  F = assemble(lf);
  % solve linear systems and update data of u
end
```

The two steps in this template that are merely outlined in a comment differ for each method. They are described in the following listing, separated by comments:

```
% --- Zarantonello: setup
blf.a = Constant(mesh, 1);
lf.f = Constant(mesh, 1);
lf.fvec = CompositeFunction(@(p) -mu(vectorProduct(p, p)) .* p, Gradient(u));
% --- Zarantonello: update
v.setFreeData(A(freeDofs,freeDofs) \ F(freeDofs));
u.setData(u.data + delta*v.data);
% --- Kacanov: setup
blf.a = CompositeFunction(@(p) mu(vectorProduct(p, p)), Gradient(u));
lf.f = Constant(mesh, 1);
% --- Kacanov: update
u.setFreeData(A(freeDofs,freeDofs) \ F(freeDofs));
% --- Newton: setup
blf.a = CompositeFunction(@(p) mu(vectorProduct(p, p)) .* [1;0;0;1] ...
  + 2*muPrime(vectorProduct(p, p)).*vectorProduct(p, p, [2,1], [2,1]'), ...
  Gradient(u));
lf.f = Constant(mesh, 1);
lf.fvec = CompositeFunction(@(p) -mu(vectorProduct(p, p)) .* p, Gradient(u));
% --- Newton: update
v.setFreeData(A(freeDofs,freeDofs) \ F(freeDofs));
u.setData(u.data + v.data);
% --- Additional functions
mu = @(t) 1 + exp(-t);
muPrime = @(t) -exp(-t);
```

As explained in Section 5.4.4, with `p = Gradient(u)`, the two calls of `vectorProduct` in the Newton bilinear form represent

$$|\nabla u^n_H|^2 = \texttt{vectorProduct(p, p)},$$
$$\nabla u^n_H \otimes \nabla u^n_H = \texttt{vectorProduct(p, p, [2,1], [2,1]')}.$$

To get an adaptive algorithm in the spirit of Algorithm 5A for lowest order FEM, i.e., $p = 1$, error estimation is done by

$$\eta_H(T)^2 := h_T^2 \|1\|_{L^2(T)}^2 + h_T \|[\![\mu(|\nabla u_H^n|^2)\nabla u_H^n \cdot \boldsymbol{n}]\!]\|_{L^2(\partial T \cap \Omega)}^2,$$

which is analogous to (5.6). Finally, we remark that [HPW21] suggests to use $u_0^0 = 0 \in \mathcal{S}_D^1(\mathcal{T}_0)$ only on the coarsest level and then to proceed by nested iteration

$$u_{\ell+1}^0 := u_\ell^{\underline{n}(\ell)} \in \mathcal{S}_D^1(\mathcal{T}_{\ell+1}) \quad \text{for all } \ell \in \mathbb{N},$$

where $\underline{n}(\ell)$ is the last iteration on the previous level $\mathcal{T}_\ell$, i.e., $u_\ell^{\underline{n}(\ell)}$ is the final iterate on $\mathcal{T}_\ell$. For lowest-order $H_D^1(\Omega)$-conforming FEM, this can be done by the prolongation class `LoFeProlongation`; see Section 5.3.3.

A numerical comparison of the three presented iterative linearization methods can be seen in Figure 5.8.

Bibliography

[AAD11] M. Ainsworth, G. Andriamaro, and O. Davydov. Bernstein-Bézier finite elements of arbitrary order and optimal assembly procedures. *SIAM J. Sci. Comput.*, 33(6):3087–3109, 2011. DOI: 10.1137/11082539X.

[ABD+21] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells. The deal.II finite element library: Design, features, and insights. *Comput. Math. Appl.*, 81:407–422, 2021. DOI: 10.1016/j.camwa.2020.02.022.

[ABH+15] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The FEniCS Project Version 1.5. *Arch. Numer. Software*, Vol 3, 2015. DOI: 10.11588/ans.2015.100.20553.

[AFP12] M. Aurada, S. Ferraz-Leite, and D. Praetorius. Estimator reduction and convergence of adaptive BEM. *Appl. Numer. Math.*, 62(6):787–801, 2012. DOI: 10.1016/j.apnum.2011.06.014.

[AO93] M. Ainsworth and J. T. Oden. A posteriori error estimators for second order elliptic systems. I. Theoretical foundations and a posteriori error analysis. *Comput. Math. Appl.*, 25(2):101–113, 1993. DOI: 10.1016/0898-1221(93)90227-M.

[AS92] F. Alouges and A. Soyeur. On global weak solutions for Landau-Lifshitz equations: existence and nonuniqueness. *Nonlinear Anal.*, 18(11):1071–1084, 1992. DOI: 10.1016/0362-546X(92)90196-L.

[AW15] M. Amrein and T. P. Wihler. Fully adaptive Newton-Galerkin methods for semilinear elliptic partial differential equations. *SIAM J. Sci. Comput.*, 37(4):A1637–A1657, 2015. DOI: 10.1137/140983537.

[BBC+13] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo. Basic principles of virtual element methods. *Math. Models Methods Appl. Sci.*, 23(1):199–214, 2013. DOI: 10.1142/S0218202512500492.

[BBD+21] P. Bastian, M. Blatt, A. Dedner, N.-A. Dreier, C. Engwer, R. Fritze, C. Gräser, C. Grüninger, D. Kempf, R. Klöfkorn, M. Ohlberger, and O. Sander. The Dune framework: Basic concepts and recent developments. *Comput. Math. Appl.*, 81:75–112, 2021. DOI: 10.1016/j.camwa.2020.06.007.

[BBI+21] R. Becker, M. Brunner, M. Innerberger, J. M. Melenk, and D. Praetorius. Goal-oriented adaptive finite element method for semilinear elliptic PDEs, 2021. arXiv: 2112.06687.

[BDD04] P. Binev, W. Dahmen, and R. DeVore. Adaptive finite element methods with convergence rates. *Numer. Math.*, 97(2):219–268, 2004. DOI: 10.1007/s00211-003-0492-7.

[BET11] R. Becker, E. Estecahandy, and D. Trujillo. Weighted marking for goal-oriented adaptive finite element methods. *SIAM J. Numer. Anal.*, 49(6):2451–2469, 2011. DOI: 10.1137/100794298.

[BGIP21] R. Becker, G. Gantner, M. Innerberger, and D. Praetorius. Goal-oriented adaptive finite element methods with optimal computational complexity, 2021. arXiv: 2101.11407.

[BGMP16] A. Buffa, C. Giannelli, P. Morgenstern, and D. Peterseim. Complexity of hierarchical refinement for a class of admissible mesh configurations. *Comput. Aided Geom. Design*, 47:83–92, 2016. DOI: 10.1016/j.cagd.2016.04.003.

[BHL+21] R. Bulle, J. S. Hale, A. Lozinski, S. P. A. Bordas, and F. Chouly. Hierarchical a posteriori error estimation of Bank-Weiser type in the FEniCS Project, 2021. arXiv: 2102.04360.

[BHP17] A. Bespalov, A. Haberl, and D. Praetorius. Adaptive FEM with coarse initial mesh guarantees optimal convergence rates for compactly perturbed elliptic problems. *Comput. Methods Appl. Mech. Engrg.*, 317:318–340, 2017. DOI: 10.1016/j.cma.2016.12.014.

[BIP21a] R. Becker, M. Innerberger, and D. Praetorius. Adaptive FEM for parameter-errors in elliptic linear-quadratic parameter estimation problems, 2021. arXiv: 2111.03627.

[BIP21b] R. Becker, M. Innerberger, and D. Praetorius. Optimal convergence rates for goal-oriented FEM with quadratic goal functional. *Comput. Methods Appl. Math.*, 21(2):267–288, 2021. DOI: 10.1515/cmam-2020-0044.

[BM11] R. Becker and S. Mao. Quasi-optimality of an adaptive finite element method for an optimal control problem. *Comput. Methods Appl. Math.*, 11(2):107–128, 2011. DOI: 10.2478/cmam-2011-0006.

[BN10] A. Bonito and R. H. Nochetto. Quasi-optimal convergence rate of an adaptive discontinuous Galerkin method. *SIAM J. Numer. Anal.*, 48(2):734–771, 2010. DOI: 10.1137/08072838X.

[BR01] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numer.*, 10:1–102, 2001. DOI: 10.1017/S0962492901000010.

[BR03] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Springer, Basel, 2003. DOI: 10.1007/978-3-0348-7605-6.

[BS08] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*. Springer, New York, third edition, 2008. DOI: 10.1007/978-0-387-75934-0.

[BV04] R. Becker and B. Vexler. A posteriori error estimation for finite element discretization of parameter identification problems. *Numer. Math.*, 96(3):435–459, 2004. DOI: 10.1007/s00211-003-0482-9.

[BV05] R. Becker and B. Vexler. Mesh refinement and numerical sensitivity analysis for parameter calibration of partial differential equations. *J. Comput. Phys.*, 206(1):95–110, 2005. DOI: 10.1016/j.jcp.2004.12.018.

[BV84] I. Babuška and M. Vogelius. Feedback and adaptive finite element solution of one-dimensional boundary value problems. *Numer. Math.*, 44(1):75–102, 1984. DOI: 10.1007/BF01389757.

[CFPP14] C. Carstensen, M. Feischl, M. Page, and D. Praetorius. Axioms of adaptivity. *Comput. Math. Appl.*, 67(6):1195–1253, 2014. DOI: 10.1016/j.camwa.2013.12.003.

[CHB09] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric analysis*. John Wiley & Sons, Chichester, 2009. DOI: 10.1002/9780470749081.

[Che09] L. Chen. *i*FEM: an integrated finite element methods package in MATLAB. Technical report, 2009. URL: https://github.com/lyc102/ifem.

[CKNS08] J. M. Cascon, C. Kreuzer, R. H. Nochetto, and K. G. Siebert. Quasi-optimal convergence rate for an adaptive finite element method. *SIAM J. Numer. Anal.*, 46(5):2524–2550, 2008. DOI: 10.1137/07069047X.

[CNX12] L. Chen, R. H. Nochetto, and J. Xu. Optimal multilevel methods for graded bisection grids. *Numer. Math.*, 120(1):1–34, 2012. DOI: 10.1007/s00211-011-0401-4.

[Cou43] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc.*, 49:1–23, 1943. DOI: 10.1090/S0002-9904-1943-07818-4.

[DFGP19] G. Di Fratta, T. Führer, G. Gantner, and D. Praetorius. Adaptive Uzawa algorithm for the Stokes equation. *ESAIM Math. Model. Numer. Anal.*, 53(6):1841–1870, 2019. DOI: 10.1051/m2an/2019039.

[DIP20] G. Di Fratta, M. Innerberger, and D. Praetorius. Weak-strong uniqueness for the Landau–Lifshitz–Gilbert equation in micromagnetics. *Nonlinear Anal. Real World Appl.*, 55:103122, 2020. DOI: 10.1016/j.nonrwa.2020.103122.

[DKS16] L. Diening, C. Kreuzer, and R. Stevenson. Instance optimality of the adaptive maximum strategy. *Found. Comput. Math.*, 16(1):33–68, 2016. DOI: 10.1007/s10208-014-9236-6.

[Dör96] W. Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM J. Numer. Anal.*, 33(3):1106–1124, 1996. DOI: 10.1137/0733054.

[DS11] A. Demlow and R. Stevenson. Convergence and quasi-optimality of an adaptive finite element method for controlling L_2 errors. *Numer. Math.*, 117(2):185–218, 2011. DOI: 10.1007/s00211-010-0349-9.

[Duf82] M. G. Duffy. Quadrature over a pyramid or cube of integrands with a singularity at a vertex. *SIAM J. Numer. Anal.*, 19(6):1260–1262, 1982. DOI: 10.1137/0719090.

[EEHJ95] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. *Acta Numer.*, 4:105–158, 1995. DOI: 10.1017/S0962492900002531.

[EG04] A. Ern and J.-L. Guermond. *Theory and practice of finite elements*. Springer, New York, 2004. DOI: 10.1007/978-1-4757-4355-5.

[Eva10] L. C. Evans. *Partial differential equations*. American Mathematical Society, Providence, second edition, 2010. DOI: 10.1090/gsm/019.

[FFP14] M. Feischl, T. Führer, and D. Praetorius. Adaptive FEM with optimal convergence rates for a certain class of nonsymmetric and possibly nonlinear problems. *SIAM J. Numer. Anal.*, 52(2):601–625, 2014. DOI: 10.1137/120897225.

[FGH+16] M. Feischl, G. Gantner, A. Haberl, D. Praetorius, and T. Führer. Adaptive boundary element methods for optimal convergence of point errors. *Numer. Math.*, 132(3):541–567, 2016. DOI: 10.1007/s00211-015-0727-4.

[FGS15] Z. Fu, L. F. Gatica, and F.-j. Sayas. Algorithm 949: MATLAB Tools for HDG in Three Dimensions. *ACM Trans. Math. Software*, 41(3):1–21, 2015. DOI: 10.1145/2658992.

[FPW11] S. Funken, D. Praetorius, and P. Wissgott. Efficient implementation of adaptive P1-FEM in Matlab. *Comput. Methods Appl. Math.*, 11(4):460–490, 2011. DOI: 10.2478/cmam-2011-0026.

[FPZ16] M. Feischl, D. Praetorius, and K. G. van der Zee. An abstract analysis of optimal goal-oriented adaptivity. *SIAM J. Numer. Anal.*, 54(3):1423–1448, 2016. DOI: 10.1137/15M1021982.

[FT17] M. Feischl and T. Tran. Existence of regular solutions of the Landau-Lifshitz-Gilbert equation in 3D with natural boundary conditions. *SIAM J. Math. Anal.*, 49(6):4470–4490, 2017. DOI: 10.1137/16M1103427.

[GHJV95] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of reusable object-oriented software*. Addison-Wesley Reading, Massachusetts, 1995.

[GHP17] G. Gantner, D. Haberlik, and D. Praetorius. Adaptive IGAFEM with optimal convergence rates: hierarchical B-splines. *Math. Models Methods Appl. Sci.*, 27(14):2631–2674, 2017. DOI: 10.1142/S0218202517500543.

[GHPS18] G. Gantner, A. Haberl, D. Praetorius, and B. Stiftner. Rate optimal adaptive FEM with inexact solver for nonlinear operators. *IMA J. Numer. Anal.*, 38(4):1797–1831, 2018. DOI: 10.1093/imanum/drx050.

[GHPS21] G. Gantner, A. Haberl, D. Praetorius, and S. Schimanko. Rate optimality of adaptive finite element methods with respect to overall computational costs. *Math. Comp.*, 90(331):2011–2040, 2021. DOI: 10.1090/mcom/3654.

[GP22] G. Gantner and D. Praetorius. *Adaptive Finite Element Methods: Convergence & optimal convergence rates*. In preparation, 2022.

[Gri11] P. Grisvard. *Elliptic problems in nonsmooth domains*. Society for Industrial and Applied Mathematics, Philadelphia, 2011. DOI: 10.1137/1.9781611972030.ch1.

[GS02] M. B. Giles and E. Süli. Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality. *Acta Numer.*, 11:145–236, 2002. DOI: 10.1017/S096249290200003X.

[GW12] M. J. Gander and G. Wanner. From Euler, Ritz, and Galerkin to modern computing. *SIAM Rev.*, 54(4):627–666, 2012. DOI: 10.1137/100804036.

[GY17] W. Gong and N. Yan. Adaptive finite element method for elliptic optimal control problems: convergence and optimality. *Numer. Math.*, 135(4):1121–1170, 2017. DOI: 10.1007/s00211-016-0827-9.

[GYZ16] W. Gong, N. Yan, and Z. Zhou. Convergence of L^2-norm based adaptive finite element method for elliptic optimal control problems, 2016. arXiv: 1608.08699.

[HP16] M. Holst and S. Pollock. Convergence of goal-oriented adaptive finite element methods for nonsymmetric problems. *Numer. Methods Partial Differential Equations*, 32(2):479–509, 2016. DOI: 10.1002/num.22002.

[HPW21] P. Heid, D. Praetorius, and T. P. Wihler. Energy contraction and optimal convergence of adaptive iterative linearized finite element methods. *Comput. Methods Appl. Math.*, 21(2):407–422, 2021. DOI: 10.1515/cmam-2021-0025.

[HPZ15] M. Holst, S. Pollock, and Y. Zhu. Convergence of goal-oriented adaptive finite element methods for semilinear problems. *Comput. Vis. Sci.*, 17(1):43–63, 2015. DOI: 10.1007/s00791-015-0243-1.

[IP21] M. Innerberger and D. Praetorius. Instance-optimal goal-oriented adaptivity. *Comput. Methods Appl. Math.*, 21(1):109–126, 2021. DOI: 10.1515/cmam-2019-0115.

[IP22] M. Innerberger and D. Praetorius. MooAFEM: An object oriented Matlab code for higher-order (nonlinear) adaptive FEM, 2022. arXiv: 2203.01845.

[IWPK20] M. Innerberger, P. Worm, P. Prauhart, and A. Kauch. Electron-light interaction in nonequilibrium: exact diagonalization for time-dependent Hubbard Hamiltonians. *Eur. Phys. J. Plus*, 135:922, 2020. DOI: 10.1140/epjp/s13360-020-00919-2.

[KPP13] M. Karkulik, D. Pavlicek, and D. Praetorius. On 2D newest vertex bisection: optimality of mesh-closure and H^1-stability of L_2-projection. *Constr. Approx.*, 38(2):213–234, 2013. DOI: 10.1007/s00365-013-9192-4.

[KS16] C. Kreuzer and M. Schedensack. Instance optimal Crouzeix-Raviart adaptive finite element methods for the Poisson and Stokes problems. *IMA J. Numer. Anal.*, 36(2):593–617, 2016. DOI: 10.1093/imanum/drv019.

[KWP+20] A. Kauch, P. Worm, P. Prauhart, M. Innerberger, C. Watzenböck, and K. Held. Enhancement of impact ionization in Hubbard clusters by disorder and next-nearest-neighbor hopping. *Phys. Rev. B*, 102(24):245125, 2020. DOI: 10.1103/PhysRevB.102.245125.

[LC17] H. Leng and Y. Chen. Convergence and quasi-optimality of an adaptive finite element method for optimal control problems on L^2 errors. *J. Sci. Comput.*, 73(1):438–458, 2017. DOI: 10.1007/s10915-017-0425-8.

[Man10] E. Manousakis. Photovoltaic effect for narrow-gap Mott insulators. *Phys. Rev. B*, 82:125109, 2010. DOI: 10.1103/PhysRevB.82.125109.

[Mau95] J. M. Maubach. Local bisection refinement for n-simplicial grids generated by reflection. *SIAM J. Sci. Comput.*, 16(1):210–227, 1995. DOI: 10.1137/0916014.

[MNS00] P. Morin, R. H. Nochetto, and K. G. Siebert. Data oscillation and convergence of adaptive FEM. *SIAM J. Numer. Anal.*, 38(2):466–488, 2000. DOI: 10.1137/S0036142999360044.

[Mor16] P. Morgenstern. Globally structured three-dimensional analysis-suitable T-splines: definition, linear independence and m-graded local refinement. *SIAM J. Numer. Anal.*, 54(4):2163–2186, 2016. DOI: 10.1137/15M102229X.

[MP15] P. Morgenstern and D. Peterseim. Analysis-suitable adaptive T-mesh refinement with linear complexity. *Comput. Aided Geom. Design*, 34:50–66, 2015. DOI: 10.1016/j.cagd.2015.02.003.

[MS09] M. S. Mommer and R. Stevenson. A goal-oriented adaptive finite element method with convergence rates. *SIAM J. Numer. Anal.*, 47(2):861–886, 2009. DOI: 10.1137/060675666.

[MSV08] P. Morin, K. G. Siebert, and A. Veeser. A basic convergence result for conforming adaptive finite elements. *Math. Models Methods Appl. Sci.*, 18(5):707–737, 2008. DOI: 10.1142/S0218202508002838.

[NW06] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, New York, second edition, 2006. DOI: 10.1007/978-0-387-40065-5.

[PP20] C.-M. Pfeiler and D. Praetorius. Dörfler marking with minimal cardinality is a linear complexity problem. *Math. Comp.*, 89(326):2735–2752, 2020. DOI: 10.1090/mcom/3553.

[Sch14] J. Schöberl. C++11 implementation of finite elements in NGSolve. ASC Report No. 30, 2014. URL: https://www.asc.tuwien.ac.at/preprint/2014/asc30x2014.pdf.

[Sch21] S. Schimanko. *On rate-optimal adaptive algorithms with inexact solvers*. PhD thesis, TU Wien, Institute of Analysis and Scientific Computing, 2021.

[Sie11] K. G. Siebert. A convergence proof for adaptive finite elements without lower bound. *IMA J. Numer. Anal.*, 31(3):947–970, 2011. DOI: 10.1093/imanum/drq001.

[SQ61] W. Shockley and H. J. Queisser. Detailed Balance Limit of Efficiency of p-n Junction Solar Cells. *J. Appl. Phys.*, 32:510–519, 1961. DOI: 10.1063/1.1736034.

[Ste07] R. Stevenson. Optimality of a standard adaptive finite element method. *Found. Comput. Math.*, 7(2):245–269, 2007. DOI: 10.1007/s10208-005-0183-0.

[Ste08] R. Stevenson. The completion of locally refined simplicial partitions created by bisection. *Math. Comp.*, 77(261):227–241, 2008. DOI: 10.1090/S0025-5718-07-01959-X.

[Tra97] C. T. Traxler. An algorithm for adaptive mesh refinement in n dimensions. *Computing*, 59(2):115–137, 1997. DOI: 10.1007/BF02684475.

[Ver13] R. Verfürth. *A posteriori error estimation techniques for finite element methods*. Oxford University Press, Oxford, 2013. DOI: 10.1093/acprof:oso/9780199679423.001.0001.

[WZ17] J. Wu and H. Zheng. Uniform convergence of multigrid methods for adaptive meshes. *Appl. Numer. Math.*, 113:109–123, 2017. DOI: 10.1016/j.apnum.2016.11.005.

[XHYM22] F. Xu, Q. Huang, H. Yang, and H. Ma. Multilevel correction goal-oriented adaptive finite element method for semilinear elliptic equations. *Appl. Numer. Math.*, 172:224–241, 2022. DOI: 10.1016/j.apnum.2021.10.001.

[ZCL09] L. Zhang, T. Cui, and H. Liu. A set of symmetric quadrature rules on triangles and tetrahedra. *J. Comput. Math.*, 27(1):89–96, 2009.

Curriculum Vitae

Name: Michael Innerberger

Date of birth: November 15, 1992 in Salzburg, Austria

Citizenship: Austria

E-mail: michael.innerberger@asc.tuwien.ac.at

Homepage: https://www.asc.tuwien.ac.at/~minnerberger/

Education

10/2018–04/2022 (expected) **PhD studies in Technical Mathematics**, TU Wien, Austria.
Member of Doctoral program *Dissipation and dispersion in nonlinear PDEs*.
Supervisor: Prof. Dirk Praetorius.

01/2016–10/2018 **Master's degree in Technical Mathematics**, TU Wien, Austria.
Passed with distinction, Master thesis:
On instance optimality of adaptive 2D FEM, Supervisor: Prof. Dirk Praetorius.

02/2014–02/2018 **Bachelor's degree in Technical Physics**, TU Wien, Austria.
Passed with distinction, Bachelor thesis supervisor: Prof. Karsten Held.

10/2012–01/2016 **Bachelor's degree in Technical Mathematics**, TU Wien, Austria.
Passed with distinction, Bachelor thesis supervisor: Prof. Anton Arnold.

09/2011–05/2012 **Military service**, Austria.

06/2011 **AHS Matura**, PG Borromäum, Salzburg, Austria.

Research experience

since 10/2018 **Project assistent**, TU Wien, Austria.
Institute of Analysis and Scientific Computing, work group on *Numerics of PDEs*.

2021 **Research stay**, Université de Pau, France.

2018 **Research internship**, Jülich Supercomputing Centre, FZ Jülich, Germany.
Distributed memory parallelization of a fast multipole method in C++.

2017 **Research internship**, Austrian Institute of Technology, Austria.
Co-simulation for HVAC building models.

Teaching experience

2020 & 2021 **Co-supervision of Bachelor theses**, TU Wien, Austria.
Conforming bisection of simplicial triangulations in 3D, in progress.
An elementary proof for convergence of adaptive FEM, 2020.

2021 **Teaching assistant**, TU Wien, Austria.
A posteriori error estimation and adaptive FEM, exercise class.

2019 & 2020 **Teaching assistant**, TU Wien, Austria.
Numerical solution of differential equations, exercise class.

2019 **Teaching assistant**, TU Wien, Austria.
Numerics of partial differential equations: stationary problems, exercise class.

2015–2018 **Teaching assistant**, TU Wien, Austria.
Introduction to programming for technical mathematics, exercise class.

Scholarships and Awards

2021 **BGF scholarship** of the French government.

2017–2018 **Scholarship**, TU Wien, Institute of Analysis and Scientific Computing, Austria.
Master's thesis funded by the Austrian Science Fund (FWF) under grant P27005
Optimal adaptivity for BEM and FEM-BEM coupling.

2014–2017 **Scholarship of the Scholarship Foundation** of TU Wien.
Granted annually for excellent achievements in the preceding academic year.

2013–2017 **Merit-based scholarship** of the faculty of Mathematics of TU Wien.
Granted annually for excellent achievements in the preceding academic year.

2011 **Dr. Hans Riegel Award** for outstanding High School Thesis.
On the problem of squaring the circle.

Scientific talks

2020 **14th World Congress in Computational Mechanics and ECCOMAS Congress (WCCM-ECCOMAS 2020)**, Paris, France (online). *Optimal convergence rates for goal-oriented FEM with quadratic goal functional*.

2019 **Reliable Methods of Mathematical Modeling (RMMM 2019)**, TU Wien, Austria. *Instance-optimal goal-oriented adaptivity*.

Own scientific publications

2022 M. Innerberger and D. Praetorius. MooAFEM: An object oriented Matlab code for higher-order (nonlinear) adaptive FEM, 2022. arXiv: 2203.01845

2021 R. Becker, M. Brunner, M. Innerberger, J. M. Melenk, and D. Praetorius. Goal-oriented adaptive finite element method for semilinear elliptic PDEs, 2021. arXiv: 2112.06687

2021 R. Becker, M. Innerberger, and D. Praetorius. Adaptive FEM for parameter-errors in elliptic linear-quadratic parameter estimation problems, 2021. arXiv: 2111.03627, accepted for publication in *SIAM J. Numer. Anal.*

2021 R. Becker, G. Gantner, M. Innerberger, and D. Praetorius. Goal-oriented adaptive finite element methods with optimal computational complexity, 2021. arXiv: 2101.11407

2021 R. Becker, M. Innerberger, and D. Praetorius. Optimal convergence rates for goal-oriented FEM with quadratic goal functional. *Comput. Methods Appl. Math.*, 21(2):267–288, 2021. DOI: 10.1515/cmam-2020-0044

2021 M. Innerberger and D. Praetorius. Instance-optimal goal-oriented adaptivity. *Comput. Methods Appl. Math.*, 21(1):109–126, 2021. DOI: 10.1515/cmam-2019-0115

2020 A. Kauch, P. Worm, P. Prauhart, M. Innerberger, C. Watzenböck, and K. Held. Enhancement of impact ionization in Hubbard clusters by disorder and next-nearest-neighbor hopping. *Phys. Rev. B*, 102(24):245125, 2020. DOI: 10.1103/PhysRevB.102.245125

2020 M. Innerberger, P. Worm, P. Prauhart, and A. Kauch. Electron-light interaction in nonequilibrium: exact diagonalization for time-dependent Hubbard Hamiltonians. *Eur. Phys. J. Plus*, 135:922, 2020. DOI: 10.1140/epjp/s13360-020-00919-2

2020 G. Di Fratta, M. Innerberger, and D. Praetorius. Weak-strong uniqueness for the Landau–Lifshitz–Gilbert equation in micromagnetics. *Nonlinear Anal. Real World Appl.*, 55:103122, 2020. DOI: 10.1016/j.nonrwa.2020.103122