Inaugural - Dissertation

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

Ruprecht-Karls-Universität

Heidelberg

vorgelegt von

Diplommathematiker Hartmut Ulrich Kapp

aus Stuttgart

2000

Tag der mündlichen Prüfung: 13.12.2001

Adaptive Finite Element Methods

for Optimization

in Partial Differential Equations

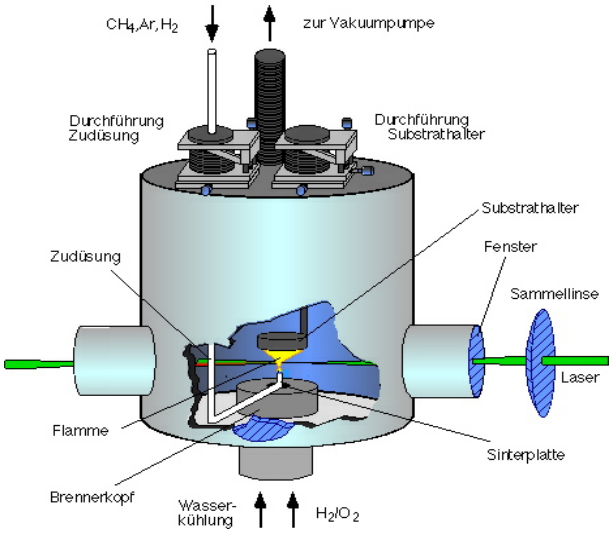
Gutachter: Professor Dr. Rolf Rannacher
Professor Dr. Hans Georg Bock

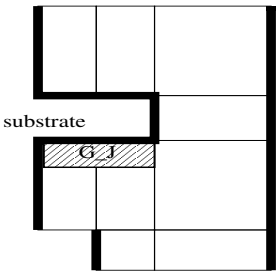
Contents

Introduction

Optimization has various applications in natural sciences and economy. The development of appropriate models for (real life) problems can be a challenging and difficult task. But the application of this model may lead to further problems. Missing model parameter data has to be determined (parameter estimation). Some data should be chosen in order to get optimal solutions like a minimal drag or a maximal output (optimal control or process optimization).

This thesis is a part of a project within the Sonderforschungsbereich 359 'Reactive Flow, Diffusion and Transport' with research by several work groups. In this project one final aim is to optimize the following Chemical Vapour Deposition (CVD) experiment:





The experiment leads to artificial production of diamond on a substrate ('Substrathalter' in above experiment Figure). The methyl radical CH_3 is ascribed playing a crucial role in the production of the diamond. So one possible optimization criterion is that the concentration of CH_3 should be maximized in an appropriate region G_J:

$$\max \quad \frac{1}{G_J} \int_{G_J} C_{CH_3}.$$

The problem can be given either in Cartesian coordinates or in cylindrical polar coordinates.

Contributions from mathematics, computer science, physics and chemistry are necessary in order to solve this problem modeled by reactive flow. The aim of this thesis is to provide mathematical techniques which will enable to solve the optimization problem. Therefore, the following systematic structure for developing appropriate optimization methods was chosen:

- Optimization governed by the Poisson equation. The optimization problem is derived in a very detailed way in Sections 1.1, 1.2, and 1.3. Numerical results are given in Section 1.6.

- Optimization governed by the linear state equation $-\Delta u + u = 0$ as an exemplary problem. The developed techniques will be explained. The problem formulation and the numerical results are presented in Section 2.13.

- Optimization governed by the nonlinear state equation $-\Delta u + u^3 - u = f$. It is used in Ginzburg-Landau models in superconductivity for semiconductors. Boundary control problems on Neumann boundaries on several domains will be considered. The optimization criteria are retrieval of prescribed solutions either on the domain or on parts of it (distributed or boundary observation). The optimization problem formulation is given in Section 4.2. Numerical results will be presented in Sections 4.4, 4.5, 2.12, 4.6, and 4.7.

- Optimization governed by the Navier-Stokes equations modeling flow ('laminar flow around an object'). Boundary control problems on Dirichlet boundaries on more complicated domains will be solved. The optimization criteria are minimal drag coefficients on boundaries of an object in the domain. The problem formulation is in Section 5.5. The numerical results can be found in Section 5.7.

- Optimization governed by the Navier-Stokes equations modeling flow with temperature by the Boussinesq model. Boundary control problems on Dirichlet boundaries on more complicated domains will be solved. The optimization criteria are maximal temperature in a certain region. The problem formulation and the numerical results are presented in Section 5.8.

In these examples, optimization may lead to unexpected solutions e.g. for the state equation. We have to keep in mind that the optimization is based on the presented mathematical models and may not necessarily be for the original physical optimization problems. Furthermore, the real sensitivities in optimization problems can also lead to solutions which are different from expected solutions ('we can learn from optimization').

Various methods have been developed to solve optimization problems. Two main streams are 'black-box optimization' and 'simultaneous optimization' leading to a coupled system. Black-box optimization takes a given simulation with the possibility to choose some model data. The simulation is the black box. The optimization process changes the model data such that the simulation fulfills in some way prescribed criteria. The simultaneous optimization approach tries to solve the whole problem in one equation system. The simulation is a more or less integrated part of the system.

The presented approach utilizes the classical Lagrangian framework for reformulating the optimal control problem as a boundary value problem for stationary solutions of the associated first-order necessary optimality conditions. By differentiation of the continuous Lagrangian functional, the first order necessary conditions of a constraint optimization problem are derived. This leads to a *coupled system* for the equations of the variables. In each step, the whole equation system is solved (*simultaneous optimization*). A standard finite element method is used for discretizing this saddle-point problem which then results in finite dimensional problems. As long as the discretization procedure uses a pure Galerkin approach, the discrete problem actually corresponds to a formulation of the original minimization problem on the discrete state space. Since discretization in partial differential equations is expensive, at least for challenging applications, the question of how this "model

reduction" affects the quality of the optimization result is crucial for a cost-efficient computation. The need for a posteriori error control is therefore evident.

The discretization of the state equation generally leads to approximate solutions, which are *not admissible* in the strict sense for the original continuous constrained optimization problem. If numerical computation with controlled accuracy should be performed, the notion of an "admissible solution" must be substituted by an error estimate for the state equation. Of course, the distance between the numerical and the exact solution should be measured with respect to the specific needs of the optimization problem, i.e. its effect on the functional to be minimized. This asks for a sensitivity analysis of the optimization problem with respect to perturbations in the state equation, particularly perturbations resulting from discretization. In this sense, the a posteriori error estimation aims to control the error due to replacing the infinite dimensional problem by its finite dimensional analogue. The crucial question is now which quality measure is appropriate for controlling the discretization error. In general, forcing this error to be small uniformly in the whole computational domain, as is often required in ODE and DAE models, is not feasible for partial differential equations. Therefore, it seems to be necessary to develop control of the discretization error in accordance with the sensitivity properties of the optimization problem.

Little research has been done on adaptivity and error estimation for optimization problems governed by partial differential equations. Habitually, this adaptivity is based on criteria considering simulation. One main idea of this thesis is that the adaptivity is obtained by error estimation criteria really originating in the optimization problem. The equation system for the error estimation problem is derived analytically. The scaling of the terms of the error estimator is done naturally by analytically derived weights. These weights involve dual solutions of the optimization problem. They describe the dependence of the error on variations of the local residuals, i.e. on the local mesh size. In general, the developed a posteriori error estimate has to be approximated by numerically solving the dual problem. This results in a feed-back process for generating successively more and more accurate error bounds and solution-adapted meshes. In applying this approach to saddle-point problems arising from optimal control problems, a natural choice for the error functional results. It is the (discretization) error in the cost functional. Applying this technique, the mesh refinement reflects the optimization problem. Some numerical results illustrate the main features of the adaptive algorithm particularly in comparison to more conventional methods based on global error control for the state equation.

The developed methods merge concepts from numerics of partial differential equations, (a posteriori) error estimation theory and nonlinear optimization. The error estimation theory is valid for the case of nonlinear state equation and nonlinear cost functional. Good analytic criteria for model reduction or discretization in optimization with partial differential equations are derived based on the theory for dual-weighted error estimators for numerical solutions of partial differential equations developed by R. Becker and R. Rannacher. Small discrete optimization problems result with good accuracy with respect to the optimization problem. This model reduction process is driven by developed dual-weighted error estimators. The aim was to develop a general method which can be applied to various families of optimization problems. Good numerical results are obtained for the presented optimization problems. Furthermore, the value of the developed weighted error estimator

can be a good estimator for error in the discrete optimization problems. An efficient and simple method for error estimation results. Adaptivity is obtained with very low additional costs. This results from a new interpretation of the (discrete) Lagrangian multiplier. It is used for the error estimation as dual solution for the primal variables. A mutual weighting for u and λ in the error estimator is derived analytically. The chosen formulation of the optimization problem including boundaries leads to a special property of the developed dual-weighted error estimator for optimization: By these weights depending on dual solutions and λ, a local control of sensitivities in the optimization problem is provided. An automatic and natural choice of the scaling in the error estimator results. In the original error estimation theory, these weights enable local stability control and local error propagation (which is of course also valid for the presented error estimator). In Section 2.5 it will be shown that the approach is not restricted to optimization problems. A general nonlinear error estimator theory is derived. An efficient method for error estimation in several applications is presented.

It may be seen as a drawback that in this approach the accuracy in the discretization of the state equation is only controlled with respect to its effect on the cost functional. This can lead to discrete models which approximate the original optimization problem with minimal cost but the obtained discrete states and controls are "admissible" only in a very weak sense, possibly insufficient for particular applications. If satisfaction of the state equation is desired in a stronger sense, the method can be combined with traditional "energy-error control" or with other necessary criteria of the problem.

The approach to discretization is relevant for good numerical solutions of systems with partial differential equations. Using the wrong discretization may lead to discrete solutions which are very different from the original continuous solution. This was observed for the presented optimization problems, especially with Navier-Stokes equations in Chapter 5. Criteria for good accuracy should be based on the whole optimization problem.

To avoid misunderstandings easily arising in this field connecting error estimation and optimization, there will be the following notation: The *dual problem* is the problem stated for solving the error estimation problem. Whereas the *adjoint problem* is the problem arising from the Lagrangian approach to solve the optimization problem.

The optimization problems may not fulfill Hadamard's postulates of well-posedness. For this reason, regularization methods are applied. Possible reasons for ill-posed problems are:

- no solution in the strict sense for all admissible data,
- solutions might not be unique for all admissible data,
- solutions might not depend continuously on the data.

By the discretization, non-uniqueness of the obtained discrete (numerical) solutions can be introduced.

Due to a new technique for Dirichlet boundary control (DBC), the regularization parameter α for the optimization problems governed by incompressible Navier-Stokes equations could be reduced from around 80 to 10^{-5} or even lower. By means of this technique, the computed control q is less restricted by the given regularization profile.

The error estimator is derived from the full analytical Fréchet differentiations of the optimization problem resulting in the first order necessary conditions. These full analytical systems will be given for optimization governed by Poisson equation, Ginzburg-Landau models in superconductivity and Navier-Stokes equations. These equation systems will be solved by applying a Newton-type method. To get a better search direction in the Newton method, a globalization method which exploits the second order condition of the optimization problem is developed. The second order information can also be used to determine if the stationary point is a (local) maximum, minimum or saddle point.

The numerical solution methods will be described in Chapter 6. The solver is based on a GMRES method with multi-grid preconditioning. The robustness of the solver is obtained from GMRES. The acceleration of the convergence rate results from multi-grid techniques.

Due to the simultaneous optimization approach, a multi saddle point structure results. This leads to the requirement of an appropriate preconditioner and other special numerical solution methods. The multi-grid techniques have to be adapted for optimization problems. The developed methods lead to a convergence even for the 'pure' Newton method.

The implementation of the optimization code is based on the DEAL library ([8]). This library was developed to compute numerical solutions of partial differential equations. In this project, optimization was added. Now, optimization features like globalization and special numerical solution methods are provided. A new class in C++ for special boundary handling was designed including boundary control and boundary observation. There is a distinction in Neumann and Dirichlet boundaries. It should be noted that boundary handling is more difficult for optimization problems than for normal partial differential equation simulations. For example the choice of priority of boundary conditions is stricter because adjoint solution, control and observation have additionally to be considered.

Several codes have been developed in order to solve the optimization problems as mentioned. They all use a similar basic structure, but have different applications and features. Numerical results for iteration to the limit on each discretization level (codes 'bkr', 'of' and 'oft') and for a less rigorous diagonal version (code 'rhoptcon') are provided.

This thesis is organized in the following way: In Chapter 1 the new approach for solving optimization problems governed by partial differential equations will be presented in detail. In Chapter 2 the developed error estimation technique is derived and explained. Chapter 3 contains considerations on globalization methods for the presented optimization problems. Chapter 4 gives results for the optimization problems governed by the nonlinear Ginzburg-Landau equations. Chapter 5 considers the optimization problems governed by Navier-Stokes equations. In Chapter 6 basic ideas on the developed numerical solutions methods are contained. And in appendix A results obtained with black box optimization are given.

Notations

Ω: domain
Γ_Q, Γ_C: control boundary
Γ_O: observation boundary
Γ_o: outflow boundary
Γ_w: wall boundary
Γ_S: substrate boundary
Γ_s: symmetry boundary (cylindrical polar coordinates)
Γ_F: fix inflow boundary
Γ_J: region of evaluation of cost functional
$\mathbf{T}_h$: triangulation (of the domain Ω)

V: Hilbert space (for state and co-state variables)
Q: Hilbert space (for control variables)
H: Hilbert space (for observations)
$H^1(\Omega)$: first-order Sobolev Hilbert-space on Ω (in the standard notation)
$L^2(\Gamma)$: Lebesgue Hilbert-space on $\Gamma \subset \Omega$
$(.,.)_\Omega$: L^2 dual products over Ω
$(.,.)_{\partial\Omega}$: L^2 dual products over $\partial\Omega$
$(.,.)$: dual product
K^*: adjoint of K $(\forall x, y : (Kx, y) = (x, K^*y))$
obs: part of domain, where objective function is evaluated ('observe') (Ω or Γ_O)
$c : V \to H$ (bounded linear) observation operator

q: boundary control variable ($\in Q$ or $\in L^2(\Gamma_C)$)
u: solution of the state equations ($\in V$ or $\in H^1(\Omega)$)
$v = (u, w)$: velocities ($\in V$ or $\in H^1(\Omega)^2$)
p: pressure ($\in V$ or $\in L_2(\Omega)/\mathbb{R}$)
t: temperature ($\in V$ or $\in H^1(\Omega)$)
λ: Lagrangian multiplier ($\in V'$ or $\in H^1(\Omega)'$)

L: Lagrangian function
H: Hessian matrix (second order differentiation of L)
J: cost functional
$J(.), E_h$: (value of) error functional
F: simulation or model or forward problem
$\omega(z)$: dual weights (in the error estimator)
c_D: drag coefficient
α: regularization factor
κ: percentage of refinement in adaptive step (fixed fraction strategy)

Chapter 1

Basic principles for optimization with PDE models

In this chapter, a new approach for solving optimization problems governed by partial differential equations will be presented in Section 1.1. The developed method will be explained for an exemplary optimization problem which is governed by the Poisson equation in the following Sections. In Section 1.8, relations to other approaches in optimization theory for problems governed by partial differential equations is given. Differentiation and stabilization play an important role in the presented approaches and will be analyzed in a general way for the given cases in Sections 1.9 and 1.10.

1.1 A new approach for solving an optimization problem governed by PDE

Let V, W and Q be Hilbert spaces. Although the following approach is rather general, it will be presented in the general framework of models containing partial differential equations. The following type of optimization problems will be considered in the sequel for $u \in V$ and $q \in Q$:

$$\min_{u,q} \quad J(u,q), \tag{1.1}$$

$$\text{s.t.} \qquad F(u,q) \;=\; 0. \tag{1.2}$$

In this dissertation, the optimization variable q will denote a boundary control variable. The *primal solution* which corresponds to the solution of the simulation will be denoted by the state variable u. The *objective function* or *cost function* J is defined as:

$$J: \; V \;\times\; Q \;\rightarrow \mathbb{R}. \tag{1.3}$$

For convex cost functionals J, it is shown in [34] that the presented optimization problems are well-defined. As indicated, the functional J can be evaluated on the whole domain Ω (*distributed observation*) or on the boundary of Ω or parts of it (*boundary observation*). The equality conditions F will always contain a simulation from partial differential equations:

$$F: V \times Q \;\rightarrow\; W \;\supseteq\; V'. \tag{1.4}$$

For the constraints, inequalities will not be considered.

For nonlinear state equations contained in F there may be a non uniqueness of the solutions. The developed theory is still valid in that case (see [34, p. 1004, remark]). In this case, there can be several stationary points (e.g. local minima).

In the first descriptive step, the problems considered have the form

$$J(u,q) \;\rightarrow\; \min!, \qquad A(u) = f + B(q), \tag{1.5}$$

where A is an elliptic differential operator, B an impact operator and J is the cost functional. The constraints are in this case $F(u,q) := A(u) - f - B(q)$.

The developed approach is based on the weak formulation of system by requiring

$$(F(u,q),\phi) \;=\; 0 \quad \forall \phi \in \; V.$$

The Lagrangian formalism is applied in order to solve the constraint optimization problem. The *Lagrangian function* is introduced

$$L(u,q,\lambda) := J(u,q) + (\lambda, F(u,q)) \tag{1.6}$$

involving a *Lagrangian multiplier* $\lambda \in H^{-1}(\Omega)$ required by the definition of the dual product $(.,.)$. It should be mentioned that $H^{-1}(\Omega) \cong H^1(\Omega)$ as shown in [17, p. 54, (2.4.6)]. This fact will also be important for the correct choice of the test spaces in the weak formulation. Stationary points of L are sought which are candidates for optimal solutions,

$$\frac{\partial L(u,q,\lambda)}{\partial (u,q,\lambda)} \;=\; 0.$$

This is a boundary value problem for triples $\{u,q,\lambda\} \;\in\; H^1(\Omega) \times L^2(\Gamma_C) \times H^1(\Omega)$,

$$(J'_u(u,q),\psi) \;+\; (\lambda, F'_u(u,q)\psi) \;=\; 0 \quad \forall \psi \;\in\; H^1(\Omega), \tag{1.7}$$

$$(J'_q(u,q),\chi) \;+\; (\lambda, F'_q(u,q)\chi) \;=\; 0 \quad \forall \chi \;\in\; L^2(\Gamma_C), \tag{1.8}$$

$$(F(u,q),\phi) \;=\; 0 \quad \forall \phi \;\in\; H^1(\Omega). \tag{1.9}$$

To get the solution of this equation system, a Newton type method on the continuous level is applied. Denoting by $H(u,q,\lambda)$ the Hessian matrix of $L(u,q,\lambda)$, each Newton step amounts to solving a linear system

$$H(u,q,\lambda)(\delta u, \delta q, \delta\lambda) \;=\; -\frac{\partial L(u,q,\lambda)}{\partial (u,q,\lambda)}, \tag{1.10}$$

for the increments $\{\delta u, \delta q, \delta\lambda\}$ of $\{u,q,\lambda\}$. The right hand side of (1.10) will further on be called *Newton residual*. There are different ways to solve the linear systems occurring in the Newton method. In order to reduce costs, we evaluate the product $H(u,q,\lambda)(\delta u, \delta q, \delta\lambda)$ as the second-order differentiation in the direction of the increments $\{\delta u,\ \delta q,\ \delta\lambda\}$.

This system has the structure of a saddle point problem which, because of its indefiniteness, requires special care in the numerical solution. In the present examples, the differentials $J'_u,\ J'_q,\ F'_u,\ F'_q, H$ are derived analytically (see Section 1.9).

For the considered optimal control problems, one of the main properties is the type of boundary control, i.e. Neumann or Dirichlet boundary control depending if there is a

Neumann (NBC) or a Dirichlet (DBC) boundary condition on the control boundary. The functional $F : V \times Q \to V'$ takes a different form for (NBC) or (DBC). There will be different boundary integrals in the derived equation systems depending on the type of the control boundary.

1.2 Exemplary optimization problem: Poisson equation as simulation

The goal of this and the following sections is to give an introduction to some basic principles underlying the applied methods. The first and exemplary optimization problem will be governed by the Poisson equation:

$$A(u) := -\Delta u = f \qquad \text{in } \Omega \tag{1.11}$$

In the presented applications, the state variables u are taken in $H^1(\Omega)$ and the Lagrangian multiplier λ in its dual space $H^1(\Omega)'$. The Poisson equation will be considered with different boundary values for the control boundary Γ_C. The boundary control variables q are in $L^2(\Gamma_C)$. The cost functional can be evaluated on the observation boundary Γ_O or on subdomains $\Omega_0 \subset \Omega$. The equation system is formulated in Cartesian coordinates. The following two types of boundary conditions are considered with $\Gamma_w = \partial\Omega \setminus (\Gamma_C \cup \Gamma_O)$:

Neumann boundary control **(NBC)**:

$$\begin{aligned} \partial_n u &= q && \text{on } \Gamma_C, \\ u &= 0 && \text{on } \Gamma_w, \\ \partial_n u &= 0 && \text{on } \Gamma_O. \end{aligned} \tag{1.12}$$

Dirichlet boundary control **(DBC)**:

$$\begin{aligned} u &= q && \text{on } \Gamma_C, \\ u &= 0 && \text{on } \Gamma_w, \\ \partial_n u &= 0 && \text{on } \Gamma_O. \end{aligned} \tag{1.13}$$

The state equation (1.11) and the boundary conditions (1.12) or (1.13) will be the constraints F.

For simplicity, the exemplary case of a rectangular domain Ω in Figure 1.1 is chosen. The mathematical theory which will be developed in the following sections is independent of the special choice of the presented domain.

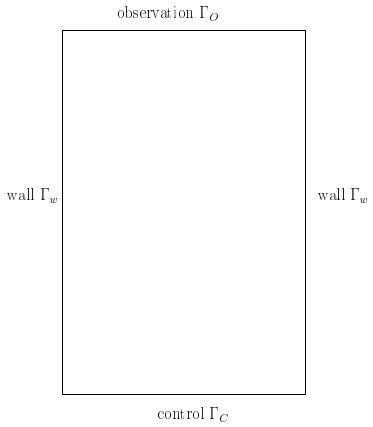


Figure 1.1: Domain for Cartesian coordinates

For (DBC), after considering the boundary conditions, the weak formulation of the simulation takes the form with appropriate test functions ϕ:

$$(F(u,q),\phi) \;=\; (\nabla u, \nabla\phi)_\Omega \;-\; (\partial_n q, \phi)_{\Gamma_C} \;-\; (f,\phi)_\Omega = 0 \quad \forall\phi.$$

Keep in mind that $u = q$ on Γ_C, which will be important in the formulation of the equation system which will be solved.

For (NBC), the boundary control condition $\partial_n u = q$ on Γ_C are considered in the term $(-\partial_n u, \phi)_{\partial\Omega}$. We get for (NBC) the following weak formulation of the simulation with appropriate test functions ϕ:

$$(F(u,q),\phi) \;=\; (\nabla u, \nabla\phi)_\Omega \;-\; (q, \phi)_{\Gamma_C} \;-\; (f,\phi)_\Omega = 0 \quad \forall\phi.$$

The following optimal control problem for the Poisson equation is considered: For a prescribed profile u_d the boundary control variable q is sought to minimize the distance between u and u_d (*Dirichlet observation*). This profile may be given on sub-domains $\Omega_0 \subset \Omega$ or on parts of the boundary Γ_O. The corresponding objective function $J : \; H^1(\Omega) \times L^2(\Gamma_C) \to \mathbb{R}$ is

$$J(u,q) = \frac{1}{2} \; \|u - u_d\|^2_{obs}.$$

The index '*obs*' indicates an evaluation only in that part of the domain, where the objective function is evaluated ('observe'). In the considered cases this is a sub-domain $\Omega_0 \subset \Omega$ or an observation boundary Γ_O.

To enhance the stability of the optimization problem, the objective function is augmented by a regularization term. In Section 1.8, a short motivation for regularization in optimization problems will be given. For (NBC), the following regularization is used (see [41] and [34]):

$$J(u,q) = \frac{1}{2} \; \|u - u_d\|^2_{obs} \;+\; \frac{\alpha}{2} \; \|q - q_0\|^2_{\Gamma_C}, \tag{1.14}$$

where q_0 is a suitable reference value. For (DBC), Gunzburger and Hou propose in [34] the following regularization:

$$J(u,q) = \frac{1}{2} \; \|u - u_d\|^2_{obs} \;+\; \frac{\alpha}{2} \; \|q - q_0\|^2_{\Gamma_C} \;+\; \frac{\alpha}{2} \; \|\nabla_s q\|^2_{\Gamma_C}, \tag{1.15}$$

where ∇_s denotes the surface gradient. With this latter regularization, the control q can be taken in $H^{\frac{1}{2}}(\Gamma_C)$. For more theoretical details see [34, p. 1033]. This regularization changes the setting since the original optimization problem is not solved. There are theoretical considerations (for details see [41]) as well as practical experiences which indicate that in this case, calculations are also possible without regularization.

For this case it has been shown in [34] and [41], that the corresponding optimization method is well-posed.

1.3 Equation systems for optimization with Poisson equation

The optimization problem of Section 1.2 leads to the following Lagrangian function with the notation of equation (1.5):

$$L(u,q,\lambda) = \|u-u_d\|^2 + \frac{1}{2}n(q,q) + A(u,\lambda) - (f,\lambda) - B(q,\lambda)$$

The operator $n(.,.)$ denotes the regularization of the cost functional. For nonlinear A, the term $A(u,\lambda)$ can be replaced by $A(u)(\lambda)$. The general setting will be described in a more detailed way in Section 2.2. This operator includes the state equations with boundary conditions. The boundary control operator $B(q,\lambda)$ is either (q,λ) for (NBC) or $(\partial_n q,\lambda)$ for (DBC). The first order necessary conditions of the constrained optimization problem are obtained by differentiation w.r.t. the variables u, q, λ.

For (NBC), the first order necessary conditions are:

$$\begin{aligned}
(u-u_d,\psi)_{obs} \;+\; (\nabla\psi,\nabla\lambda)_\Omega &= 0 \;\forall\psi\in H^1(\Omega), && (1.16)\\
\alpha(q,\chi)_{\Gamma_C} \;-\; \alpha(q_0,\chi)_{\Gamma_C} \;-\; (\chi,\lambda)_{\Gamma_C} &= 0 \;\forall\chi\in L^2(\Gamma_C), && (1.17)\\
(\nabla u,\nabla\phi)_\Omega \;-\; (f,\phi)_\Omega \;-\; (q,\phi)_{\Gamma_C} &= 0 \;\forall\phi\in H^1(\Omega). && (1.18)
\end{aligned}$$

For this equation system, the following form on the left hand side of (1.10) for (NBC) is obtained:

$$H(u,q,\lambda)(\delta u,\delta q,\delta\lambda)(\psi,\chi,\phi) = \begin{pmatrix} (\delta u,\psi)_{obs} \;+\; (\nabla\psi,\nabla\delta\lambda)_\Omega \\ \alpha(\delta q,\chi)_{\Gamma_C} \;-\; (\chi,\delta\lambda)_{\Gamma_C} \\ (\nabla\delta u,\nabla\phi)_\Omega \;+\; (\delta q,\phi)_{\Gamma_C} \end{pmatrix}. \qquad (1.19)$$

It should be pointed out that for (NBC) there are no differentials needed in the equation on the boundary Γ_C resulting from the differentiation w.r.t. q. This will be different for (DBC), and there will be the problem of choosing the correct formulation of the differentials on Γ_C in order to get a numerically stable solution process.

For (DBC), the boundary condition $u = q$ has to be considered in the formulation of the equation system. In order to get symmetry of the equation system, the term $(\partial_n\chi,\lambda)_{\Gamma_C}$ is transformed to $-(\chi,\partial_n\lambda)_{\Gamma_C}$ by partial integration.

$$\begin{aligned}
(u-u_d,\psi)_{obs} \;+\; (\nabla\psi,\nabla\lambda)_\Omega &= 0 \;\forall\psi\in H^1(\Omega), && (1.20)\\
\alpha(q-q_0,\chi)_{\Gamma_C} + \alpha(\nabla_s q,\nabla_s\chi)_{\Gamma_C} - (\partial_n\lambda,\chi)_{\Gamma_C} &= 0 \;\forall\chi\in L^2(\Gamma_C), && (1.21)\\
(\nabla u,\nabla\phi)_\Omega \;-\; (\partial_n u,\phi)_{\partial\Omega} \;-\; (f,\phi)_\Omega &= 0 \;\forall\phi\in H^1(\Omega). && (1.22)
\end{aligned}$$

Applying the Newton method on the continuous level, the following form of the left hand side of (1.10) results for (DBC):

$$H(u,q,\lambda)(\delta u,\delta q,\delta\lambda)(\psi,\chi,\phi) = \begin{pmatrix} (\delta u,\psi)_{obs} \;+\; (\nabla\psi,\nabla\delta\lambda)_\Omega \\ \alpha(\delta q,\chi)_{\Gamma_C} + \alpha(\nabla_s\delta q,\nabla_s\chi)_{\Gamma_C} - (\partial_n\delta\lambda,\chi)_{\Gamma_C} \\ (\nabla\delta u,\nabla\phi)_\Omega \;-\; (\partial_n\delta u,\phi)_{\partial\Omega} \end{pmatrix}. \tag{1.23}$$

In the stated equations, the control q is obtained by a weak equation system on the boundary. Whereas the boundary conditions for the variables u and λ are a mixture of strong boundary conditions and weak boundary conditions obtained by the equation system derived above.

It should be mentioned that there are compatibility conditions valid for the control q by its relation with the state variable u. Only those q are allowed which lead to a u fulfilling its state equations.

So far, only the first order necessary conditions of an optimization problem with constraints are considered. The second order condition for optimization problems will be used later on to develop trust region-like modified Newton methods in Chapter 3.

1.4 Choice of boundary conditions

In this section, a theoretical derivation of the boundary conditions for the Lagrangian multiplier should be given. Later on, the Lagrangian multiplier λ will be in relation with the dual solution arising from the error estimation problem. A detailed derivation of these equation systems will be given in Chapter 2. The duality can easily be seen by the following fact: λ is the solution of the equations attained by the differentiation of the Lagrangian function w.r.t. u. And u is obtained by the differentiation of the Lagrangian function w.r.t. λ.

The notation of the boundaries is the same as in the last section. The example in this chapter is the Laplace equation

$$-\Delta u \;=\; 0 \qquad \text{in } \Omega.$$

For Neumann boundary control (NBC), the derivation leads to the same boundary conditions for u and λ on the boundaries. For the Dirichlet boundary control (DBC), only the observation and control boundary will be considered. We have the following boundary conditions for u:

$$u \;=\; q \qquad \text{on } \Gamma_C, \qquad \partial_n u \;=\; 0 \qquad \text{on } \Gamma_O.$$

For the other boundaries, the boundary conditions are normally obvious and are in general the same boundary conditions as u. They have a natural Dirichlet or natural Neumann boundary condition.

Let the cost functional $J(u,q)$ be

$$J(u,q) = \frac{1}{2}\|u-u_d\|^2\Gamma_O + \frac{\alpha}{2}\|q\|^2_{\Gamma_C} + \frac{\beta}{2}\|\nabla_s q\|^2_{\Gamma_C} + \frac{\gamma}{2}\|q\|^2_{H^{1/2}(\Gamma_C)} + \frac{\theta}{2}\|\nabla u\|^2_\Omega$$

In this formulation, several regularization methods are contained. The appropriate space for q depends now on the chosen regularization. For $\beta = 0$, $q \in H^{1/2}(\Gamma_C)$ with $q|_{\partial\Gamma_C} = 0$, the boundary condition for u which is also valid for q. This means that q must be at least in the space of the traces of H^1-functions. For $\beta \neq 0$, $q \in H^1(\Gamma_C)$ again with $q|_{\partial\Gamma_C} = 0$.

Let now $L := J(u,q) + (\nabla\lambda, \nabla u)_\Omega$ be the Lagrangian function of the optimization problem of the previous section (without explicit boundary conditions). The variable u is in $q + H^1(\Gamma;\Omega)$ and λ is in $H^1(\Gamma;\Omega)$, the dual space. Differentiation of L w.r.t. λ leads to:

$$\frac{\partial L}{\partial \lambda} = (\nabla\delta\lambda, \nabla u)_\Omega = 0 \quad \forall\delta\lambda \qquad \Rightarrow \quad \Delta u = 0 \qquad \partial_n u|_{\Gamma_O} = 0.$$

The above boundary conditions for u can be stated.

Differentiation of L w.r.t. u leads then to the equation system for λ:

$$\begin{aligned}\frac{\partial L}{\partial u} &= (u - u_d, \delta u)_{\Gamma_O} + (\nabla\lambda, \nabla\delta u)_\Omega = 0 \quad \forall\delta u \\ &\Rightarrow \quad -\Delta\lambda = 0 \qquad \partial_n\lambda|_{\Gamma_O} = u - u_d.\end{aligned}$$

The latter boundary integral results from partial integration. The boundary condition on the control boundary Γ_C is obtained by the error condition from the dual problem $u - u_h = 0$ with the discrete variable u_h of u. This means that there is no error on Γ_C. Otherwise, also the Galerkin orthogonality (see Sections 2.2 and 2.4) would not be true. Hence, the following equations result for λ:

$$-\Delta\lambda = 0 \qquad \lambda|_{\Gamma_C} = 0 \qquad \partial_n\lambda|_{\Gamma_O} = u - u_d.$$

For $\theta \neq 0$, the term $\frac{\theta}{2}(\nabla u, \nabla\psi)_\Omega$ can be eliminated by partial integration. The resulting terms $(\Delta u, \delta u)_\Omega$ and $(\partial_n u, \delta u)_{\partial\Omega}$ are equal to 0.

The derivation of L by the control q underlines the above choice for some critical boundary conditions:

$$\frac{\partial L}{\partial q} = \frac{d}{dt} L(u, q + t\chi, \lambda)|_{t=0}.$$

Let ϕ be the harmonic prolongation of χ on Ω, i.e. $\phi|_{\Gamma_C} = \chi$, $\Delta\phi = 0$. It is not used explicitely, but for theorectical reasons it must be defined. Hence,

$$\begin{aligned}\frac{d}{dt}L(u, q + t\phi, \lambda)|_{t=0} &= \frac{1}{2}\frac{d}{dt}J(u + t\phi, q + t\chi) + (\nabla\lambda, \nabla(u + t\phi))_\Omega, \\ &\qquad \text{with } u + t\phi \in (q + t\chi) + H_0^1 \\ &= (u - u_d, \phi)_{\Gamma_O} + \alpha(q,\phi)_{\Gamma_C} + \beta(\nabla_s q, \nabla_s\phi)_{\Gamma_C} + \gamma(q,\phi)_{\Gamma_C} \\ &\quad + \theta(\nabla\phi, \nabla u)_\Omega + (\nabla\lambda, \nabla\phi)_\Omega \\ &= (u - u_d, \phi)_{\Gamma_O} + \alpha(q,\phi)_{\Gamma_C} + \beta(\nabla_s q, \nabla_s\phi)_{\Gamma_C} + \gamma(q,\phi)_{\Gamma_C} \\ &\quad + \theta(\nabla\phi, \nabla u)_\Omega - (\Delta\lambda, \phi)_\Omega + (\partial_n\lambda, \phi)_{\Gamma_C} + (\partial_n\lambda, \phi)_{\Gamma_O} = 0 \quad \forall\phi \\ \Rightarrow\ & u - u_d = \partial_n\lambda \quad \text{on} \quad \Gamma_O \\ & \text{for } \alpha = \beta = \gamma = \theta = 0: \qquad \partial_n\lambda = 0 \quad \text{on} \quad \Gamma_C.\end{aligned}$$

The critical reformulation of the equation is done by partial integration. The last equation is equation (1.21) for (DBC).

On the boundary Γ_C, we can now state the following equation:

$$-\alpha q + \beta \Delta_s q - \gamma \Delta_s^{1/2} q + \theta \partial_n u + \partial_n \lambda|_{\Gamma_C} = 0. \tag{1.24}$$

For $\gamma = \theta = 0$, the following equation is obtained:

$$-\alpha q + \beta \Delta_s q + \partial_n \lambda|_{\Gamma_C} = 0. \tag{1.25}$$

This is the equation which results from the regularization of the previous section.

Whereas for $\gamma = \beta = 0$, the following equation is obtained:

$$-\alpha q + \theta \partial_n u + \partial_n \lambda|_{\Gamma_C} = 0. \tag{1.26}$$

The latter equation with the differential of u was taken for (DBC). It enabled good computations.

By the above derivation, u and λ have always Dirichlet or Neumann boundary conditions on the same boundaries. If u has an inhomogeneous Dirichlet boundary condition, then λ has a homogeneous Dirichlet boundary condition. Hence, the same test functions can be used for u and λ. These test functions have the same boundary conditions as u and λ. They can also be interpreted as test functions of each other. This is important for a good formulation of the error equations. The problem with weak and strong formulations of boundary conditions is eliminated by this choice.

The boundary condition $u - u_d = 0$ on Γ_O is not true for all cases. The boundary condition $\partial_n u|_{\Gamma_O} = 0$ need not be fulfilled for u_d. The data u_d is a given profile. Otherwise, the choice $u - u_d = 0$ on Γ_O seems to be correct.

The boundary condition $u - u_h = 0$ on Γ_C need not be valid. There can be some additional errors in the equation system like the interpolation error or the linearization error. By the above derivation, it is clear that there is a Dirichlet boundary condition on Γ_C. The question is which (Dirichlet) value should be assigned. Neglecting the other errors, the choice $u - u_h = 0$ on Γ_C seems to be correct.

If there is no regularization and $u \equiv u_d$ on Γ_O, then $\lambda = 0$ in the optimal solution. (Also $L = 0$).

The boundary conditions for the increments are described in Section 6.4.

1.5 Galerkin method

The Galerkin finite element discretization of the saddle-point problem resulting from the Lagrangian formulation of the optimization problem uses subspaces $V_h \subset V$ of piecewise polynomial functions defined on regular decompositions $\mathbf{T}_h = \cup_{T \in \mathbf{T}_h} \{T\}$ of the domain Ω into cells T (triangles or quadrilaterals); see Brenner and Scott [17]. The applied discretization is based on standard finite element Galerkin techniques.

The following well-known (and generalized) precis of the Ritz-Galerkin method given in [17] should be indicated. Let V be the continuous space in which we solve the continuous problem. In the weak formulation, the solution u of the optimization problem can be characterized by finding

$$u \in V \quad \text{such that} \quad a(u, v) = (f, v) \quad \forall v \in V. \tag{1.27}$$

Let $V_h, W_h \subset V$ be any (finite dimensional) subspaces of the continuous space V. Then the discrete scheme for approximating (1.27) can be stated as

$$u_h \in V_h \quad \text{such that} \quad a(u_h, v) = (f, v) \quad \forall v \in W_h. \tag{1.28}$$

The Ritz-Galerkin method ($V_h = W_h$) can be characterized by solving

$$KU = F. \tag{1.29}$$

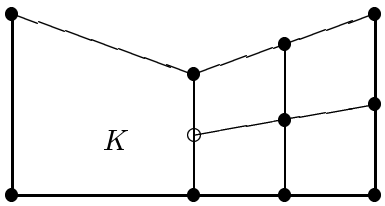
The matrix $K = (K_{ij})$ and the vectors $U = (U_j)$ and $F = (F_i)$ can be stated in the following way: $u_h = \sum_{j=1}^{n} U_j \phi_j$, $K_{ij} = a(\phi_j, \phi_i)$ and $F_i = (f, \phi_i)$. $\{\phi_i | 1 \leq i \leq n\}$ is a basis of V_h and n is the dimension of V_h.

1.6 Numerical results for the Poisson equation for cartesian coordinates

In this section, some numerical results for the equation systems in Cartesian coordinates derived above will be presented.

The resulting linear systems in the Newton iteration will be solved by an adaptive finite element scheme.

The discretization of this equation system is based on a finite element Galerkin method with Q^1-elements. The meshes fulfill the usual regularity conditions. Hanging nodes are allowed and facilitate local mesh refinement, but at most one hanging node per edge:



The corresponding degrees of freedom are eliminated by interpolation in order to keep the discretization conforming. For the state and adjoint variables, piecewise polynomial shape functions are taken. For the control variables, the traces of the above shape functions on Γ_C are used. This choice is not necessary but simplifies notation. In order to avoid unnecessary complications due to curved boundaries, the domain Ω is supposed to be polygonal. The discretization is realized using the DEAL library ([8]).

There is a crucial difference between (NBC)-systems and (DBC)-systems. (NBC)-systems for the presented case do not need differentiation on the control boundary Γ_C. The calculation of the boundary control values is therefore easier. One needs a possibility to handle boundary integrals.

(DBC)-systems have not only the problem mentioned in Section 1.4 that the values from the control q to the state variable u have to be assigned. The appropriate choice of the test spaces results from this fact. Additionally, there is differentiation information needed on the control boundary Γ_C. With the boundary conditions from Section 1.4, one

can not avoid to take information from the integration on the domain next to the boundary to get the necessary differentiation information. The differentiation values on the boundary Γ_C are computed for the presented version of (DBC) by the cell (in this case a rectangle) next to the boundary element, or, in fact, the finite element which contains the boundary element as described in Section 6.6. These values are transformed to the boundary (choice of convenience).

Due to the calculation of q and the boundary conditions for u on Γ_C leading to a change of u by q, convergence in one Newton iteration is not necessarily obtained. For this consideration, the control q can be taken as a 'perturbation', decelerating the solution (and convergence) process.

In the test problem in Figure 1.1, the observation is $u_d = 0$ on Γ_O. Thus one solution would be with a state equation which is $u = 0$ in Ω. But also other solution which are harmonic functions with $u = 0$ on Γ_O and fulfilling the boundary conditions $u = 0$ on the wall Γ_w are possible. The starting value for u is 10 and for q is 4. Therefore, one expects a convergence to 0 of u and q. Especially q should be a paraboloid because of the strong effect of the boundary conditions $u = 0$ on Γ_w. This behavior was observed. There were other starting values tested for q (0 and -5), both leading to similar results.

The following table shows numerical results. In this numerical examples, α is equal to 1. The control boundary type, observation boundary type, number of cells, Newton residual and Newton increment are denoted by 'control', 'obs', '#cells', 'n_res' and 'n_incr', respectively:

control	obs	#cells	n_res	n_incr	CPU-seconds
(DBC)	D	256	$2 * 10^{-6}$	$5.4 * 10^{-6}$	20.3
(DBC)	N	256	10^{-8}	$1.1 * 10^{-8}$	1.5
(NBC)	D	quadratic convergence			
		256	$2 * 10^{-7}$	$3 * 10^{-7}$	25
		1024	$5.7 * 10^{-7}$	$2 * 10^{-6}$	92.8
(NBC)	N	quadratic convergence			
		256	$2 * 10^{-7}$	$3 * 10^{-7}$	29
		1024	$5.7 * 10^{-7}$	$2 * 10^{-6}$	103

For the presented version of (DBC), the solutions are obtained faster than for (NBC) in all test cases. Especially (DBC) with Neumann observations, the solution is found in only 4 Newton iterations. The optimization problems governed by the Poisson equation can be solved in a satisfactory way even on rather coarse meshes. The obtained residuals and increments are sufficiently small.

As we will see later on, the solutions of the optimization problem depend of the value of the regularization factor α. Big α lead to faster convergence, but do also change the original optimization problem and therefore the attained solution in a stronger way. See Section 2.13 for numerical examples.

1.7 Optimization for the Poisson equation in cylindrical polar coordinates

In this section, the equation system for optimization in *cylindrical polar coordinates* for scalar solution of the Poisson equation u is considered. As mentioned in the introduction, important future applications like the CVD experiment can also be formulated in cylindrical polar coordinates. There is an additional integral $\int_\Omega \frac{u_r}{r^2}\ \phi\ d\Omega$ for the Poisson equation comparing the previous formulation in Cartesian coordinates with the cylindrical polar coordinates. Note that the following integrals are the same: $\int_\Omega \frac{u_r}{r^2}\ \phi\ d\Omega = \int_\Omega \frac{u_r}{r}\ \phi\ dr\ dz$. The theoretical derivation of this additional integral and general information on cylindrical polar coordinates can be found in [5], [54] or [63].

Due to the formulation in cylindrical polar coordinates, there are different boundary conditions than in the Cartesian case. Additionally, there is a symmetry boundary Γ_s, which is the axis of rotation (see Figure 1.2).

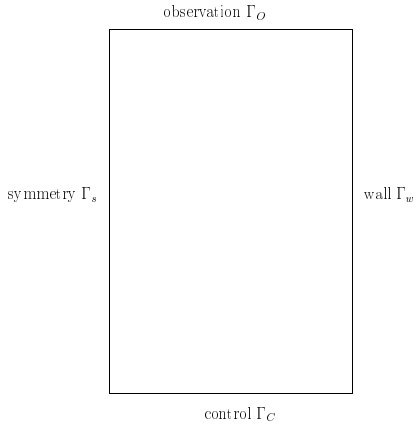


Figure 1.2: Domain for cylindrical polar coordinates

The following boundary conditions are stated for u and λ for the presented problem in cylindrical polar coordinates and domain as in Figure 1.2:

$$
\begin{array}{lrcrcll}
\textbf{(NBC)}: & \partial_n u & = & q & \text{and} & \partial_n \lambda = 0 & \text{on } \Gamma_C, \\
& u & = & 0 & \text{and} & \lambda = 0 & \text{on } \Gamma_w, \\
& \partial_n u & = & 0 & \text{and} & \partial_n \lambda = 0 & \text{on } \Gamma_s, \\
& \partial_n u & = & 0 & \text{and} & \partial_n \lambda = 0 & \text{on } \Gamma_O.
\end{array}
$$

For the Dirichlet boundary control, we have a Dirichlet boundary on Γ_C:

$$
\begin{array}{lrcrcll}
\textbf{(DBC)}: & u & = & q & \text{and} & \lambda = 0 & \text{on } \Gamma_C, \\
& u & = & 0 & \text{and} & \lambda = 0 & \text{on } \Gamma_w, \\
& \partial_n u & = & 0 & \text{and} & \partial_n \lambda = 0 & \text{on } \Gamma_s, \\
& \partial_n u & = & 0 & \text{and} & \partial_n \lambda = 0 & \text{on } \Gamma_O.
\end{array}
$$

For (NBC), the weak formulation of the Poisson equation can be stated as:

$$(F(u,q),\cdot) \;= (\nabla u, \nabla\cdot)_\Omega + (u_r r^{-1},\cdot)_\Omega - (q,\cdot)_{\Gamma_C} - (f,\cdot)_\Omega.$$

Therefore, the first order necessary condition of this optimization problem reads:

$$\begin{aligned}(u,\psi)_{obs} - (u_d,\psi)_{obs} + (\nabla\psi,\nabla\lambda)_\Omega + (\lambda_r r^{-1},\psi_r)_\Omega - (\partial_n\lambda,\psi)_{(\partial\Omega\setminus\Gamma_C)} &= 0,\\ \alpha(q,\chi)_{\Gamma_C} - \alpha(q_0,\chi)_{\Gamma_C} - (\chi,\lambda)_{\Gamma_C} &= 0,\\ (\nabla u,\nabla\phi)_\Omega + (u_r r^{-1},\phi_r)_\Omega - (f,\phi)_\Omega - (q,\phi)_{\Gamma_C} - (\partial_n u,\phi)_{(\partial\Omega\setminus\Gamma_C)} &= 0.\end{aligned}$$

The left hand side of (1.10) for (NBC) is:

$$\begin{pmatrix} (\delta u,\psi)_{obs} + (\nabla\psi,\nabla\delta\lambda)_\Omega + (\delta\lambda_r r^{-1},\psi_r)_\Omega - (\partial_n\delta\lambda,\psi)_{(\partial\Omega\setminus\Gamma_C)} \\ \alpha(\delta q,\chi)_{\Gamma_C} - (\chi,\delta\lambda)_{\Gamma_C} \\ (\nabla\delta u,\nabla\phi)_\Omega + (\delta u_r r^{-1},\phi_r)_\Omega - (\delta q,\phi)_{\Gamma_C} - (\partial_n\delta u,\phi)_{(\partial\Omega\setminus\Gamma_C)} \end{pmatrix}.$$

For (NBC) in the test case of Figure 1.2, convergence is stated.

Also for (DBC), the difference to the formulation in Cartesian coordinates is the integral $\int_\Omega u_r r^{-2}\ \phi\ d\Omega$ and the additional boundary condition on the boundary for $r = 0$ (symmetry). The weak formulation of the Poisson equation can be stated as:

$$(F(u,q),\cdot) \;=\; (\nabla u,\nabla\cdot)_\Omega \;+\; (u_r r^{-1},\cdot)_\Omega \;-\; (\partial_n u,\cdot)_{\partial\Omega} \;-\; (f,\cdot)_\Omega.$$

The first order necessary condition of this optimization problem for (DBC) reads:

$$\begin{aligned}(u,\psi)_{obs} - (u_d,\psi)_{obs} + (\nabla\psi,\nabla\lambda)_\Omega + (\lambda_r r^{-1},\psi_r)_\Omega - (\partial_n\lambda,\psi)_{(\partial\Omega\setminus\Gamma_C)} &= 0,\\ \alpha(q,\chi)_{\Gamma_C} - \alpha(q_0,\chi)_{\Gamma_C} - (\partial_n\lambda,\chi)_{\Gamma_C} &= 0,\\ (\nabla u,\nabla\phi)_\Omega + (u_r r^{-1},\phi_r)_\Omega - (\partial_n u,\phi)_{(\partial\Omega\setminus\Gamma_C)} - (f,\phi)_\Omega &= 0.\end{aligned}$$

The left hand side of (1.10) for (DBC) is:

$$\begin{pmatrix} (\delta u,\psi)_{obs} + (\nabla\psi,\nabla\delta\lambda)_\Omega + (\delta\lambda_r r^{-1},\psi_r)_\Omega - (\partial_n\delta\lambda,\psi)_{(\partial\Omega\setminus\Gamma_C)} \\ \alpha(\delta q,\chi)_{\Gamma_C} - (\partial_n\delta\lambda,\chi)_{\Gamma_C} \\ (\nabla\delta u,\nabla\phi)_\Omega + (\delta u_r r^{-1},\phi_r)_\Omega - (\partial_n\delta u,\phi)_{(\partial\Omega\setminus\Gamma_C)} \end{pmatrix}.$$

For (DBC) in the test case of Figure 1.2, convergence can be stated. If the solution of the optimization problem is taken as starting value for the iterations, the code terminates immediately detecting that the optimum is already obtained.

1.8 Optimization theory with PDE simulation

In this section, a fundamental outline on optimization theory of systems governed by partial differential equations with respect to the presented optimization problems should be given. The outline is mainly based on a paper of Gunzburger and Hou [34] and on a book of Lions [48].

For optimal control theory of systems governed by elliptic partial differential equations, the general and abstract derivation of the equations of an optimal control problem can be found in [48, chapter II]. For the control, there is a distinction between distributed

control and boundary control. *Distributed control* means that the control is distributed over the domain Ω. An alternative definition would be that the 'control is effected through a source term in the governing partial differential equations'([34]). Whereas *boundary control* means that the control is a function defined only on (a part of) the boundary of Ω. The same definitions are applied for the observations. The hardest case is boundary control and boundary observation. For distributed observation, the observation represents the canonical injection $c : H^1(\Omega) \rightarrow L^2(\Omega)$. Whereas for boundary observation, the observation leads to a trace operator $c : H^1(\Omega) \rightarrow L^2(\Gamma_O)$. But also the sensitivities of the control from the observations seem to be better. For distributed observation, the transmission of model information from 'obs' defined on the whole domain to Γ_C is clearly easier. Furthermore, there is a distinction between Neumann and Dirichlet problems, both implying different difficulties. Some stated difficulties in the formulas derived above are not given in Lions [48]. Also pointwise control and observation are considered ([48, section 5.4]). Furthermore, existence results for optimal controls are proved ([48, section 7]).

The derivation of the optimal control theory for systems governed by parabolic and hyperbolic partial differential equations can also be found in [48].

An introduction in optimization for flow problems can be found in [33].

In [34], an abstract framework for the analysis and approximation of a class of nonlinear optimization problems is given. Both constraints and objective functional can be nonlinear. Existence results of optimal solutions and of the Lagrangian multipliers are given. By this, an optimization system is derived which leads to the optimal states and controls. The approximation is done by finite element methods as in this thesis. Two applications are Ginzburg-Landau equations of superconductivity and the Navier-Stokes equations for incompressible, viscous flow. Both will be analyzed later on. A main step is that there must be the existence of a solution for the simulation. Then the existence of a solution for the optimization problem can be attained. But the restrictions for the solutions of the simulation will somehow occur in the solution of the optimization problem.

In many cases of optimization with partial differential equation models, regularization terms for the optimization problem are necessary. Regularization techniques usually are applied in order to get stability of the *optimization* problem. Additionally, if there is no existence of a solution of the optimization problem in the classical sense, regularization methods are used to obtain well-posedness. The regularization terms are originally not introduced for discretization reasons. A general introduction in regularization techniques for optimization problems, mainly for inverse problems, can be found in [26]: 'In general terms, regularization is the approximation of an ill-posed problem by a family of neighboring well-posed problems'. 'All that a regularization method can do is to recover partial information about the solution as stably as possible. The "art" of applying regularization methods will always be to find the right compromise between accuracy and stability'. Many concrete examples are given therein. Some very elaborated regularization methods like the Tikhonov regularization (minimizing Tikhonov functional $x \rightarrow \|Tx - y^\delta\|^2 + \alpha\|x\|^2$) do not seem to be applicable for the presented context in the moment for the evaluation of the operators seems to become too complicated and too costly. Therefore, the regularization methods published in [34] have been used. Depending on the type of control, there are several regularization methods: for distributed control on page 1017 ($\int_\Omega q^2 d\Omega$), for (NBC) on page 1024 ($\int_\Gamma q^2 d\Gamma$) and for (DBC) on page 1032 ($\int_\Gamma (|\nabla_s q|^2 + |q|^2) d\Gamma$ whereas ∇_s denotes

the surface gradient). These regularization methods are motivated by theoretical optimization criteria, e.g. proofs on existence of the optimization problems. The main reasons for regularization are:

- enhance the stability of the optimization problem
- avoid ill-posedness of the optimization problem
- improve conditioning
- enable rigorous mathematical analysis under less restrictive assumptions
- enable control on the optimization variable which guarantees solvability and convergence of approximations

The regularization changes the original optimization problem leading to (a family of) better-posed problems. Nevertheless, the regularizations can influence the solution of the optimization problem in a strong way. See Section 2.13 for numerical examples. Above regularization techniques were used with an application-dependent regularization factor (not necessarily equal to 1).

1.9 Possible choices for the differentiation operators

The presented equation systems (1.10) are obtained by applying the exact Fréchet differentiation on the continuous level. More precisely, it is the formal differentiation on the differential equation level. For optimization with partial differential equations this strategy seems more appropriate than other techniques like external numerical differentiation (END) or internal numerical differentiation (IND) ([15]). For optimization with partial differential equations, the discretization can lead to a large number of discrete variables. Therefore, the function evaluations are very expensive. Additionally, high accuracy is needed in the solution process. Numerical differentiation techniques lead to additional errors, which are much higher for huge systems arising from the discretization of the solutions of the partial differential equations and the whole optimization problem. Furthermore, in the context of error estimation for solutions of optimization problems with simulations from partial differential equations, the error by END or IND would be an additional error to be considered.

There are also alternative computation strategies for the Hessian matrix. One example are the approximations BFGS or DFP updates (see [31]). The Fréchet differentiation has the advantage that the analytical system for the Newton method is taken. Again, the exact system is considered, not an approximate one, reducing the total error of the system. Nevertheless, one advantage of the BFGS formula would be that, under certain conditions, to Hessian matrices are positive definite. Additionally, recent research in [47], [53] indicates that the optimal solution attained by exact Hessian matrices are closer to the continuous one or the results in experiments (at least for ODE and DAE models).

A good survey on a comparison on different methods to compute the sensitivities for optimization with partial differential equation models, mainly for flow problems, is given in [3]. It also supports the chosen approach which should even be better then methods using automatic differentiation techniques. The latter comparison was done by J.R. Appel and M.D. Gunzburger with ADIFOR.

1.10 Stabilization of the optimization problem

In this thesis, by 'stabilization' a stability of the *discretization* should be obtained. The presented method uses the stabilization of the simulation for the whole optimization problem. For the stabilization of the Navier-Stokes equations, a Schur complement technique using the LBB condition and the Rayleigh quotient to reduce the effect of the saddle point structure is described in [6, p. 53]. The zero entry on the diagonal of the matrix is replaced to get a stable formulation of the simulation. By symmetry, this method can as described be applied to the dual solution. The advantage of this approach is that a stable simulation is sufficient for the stabilization of the primal and dual problem. With additional techniques for the 'pure' optimization part, a solution for the optimization problem can be attained.

Another method would be to construct the stabilization for the dual solution separately. This would lead to additional effort for the additional stabilization. For this method, expert knowledge for the formulation of stabilization of the dual solution would be required.

Depending on the optimization problem, the diagonal of the matrix of the differential of second order of the Lagrangian function can have several zero entries. $\frac{\partial^2 L}{\partial \lambda^2}$ is always equal to 0 because the Lagrangian multiplier is only linear in our equation systems. $\frac{\partial^2 L}{\partial q^2}$ depends on the regularization, especially on the regularization factor α in the chosen regularization. $\frac{\partial^2 L}{\partial u^2}$ depends strongly on the kind of optimization problem. For example, in the case of parameter estimation problems, this diagonal entry can also tend to 0 (and should do so for the global minimum, if the data is not perturbed). Also for the zero entries resulting from the optimization part, a stabilization for example by Schur complement methods for the appearing saddle points would be needed. This could lead to a better convergence and behavior of the equation system. The whole problem could also be stabilized without considering special parts like the simulation separately. Anyway, it would need much effort for every new optimization problem and also for every new formulation of it. To avoid a new derivation of the stabilization for each optimization problem, the presented technique based on the stabilization of the simulation was developed.

Furthermore, stabilization depends always on the norm in which it is considered. For example, if the considered problem is stable in the L^2-norm ($(u,v)_{L^2} = \int_\Omega u(x)v(x)dx$), it need not be stable in the H^1-norm ($(u,v)_{H^1} = \int_\Omega u(x)v(x)dx \; + \; \int_\Omega \nabla u(x)\nabla v(x)dx$). The additional integral with the differentials of the functions may lead to an unstable behavior. This effectuates various discussions on appropriate norms for optimization problems. One problem is that the calculation of the residual and the increment in the H^1-norm is not easy for the determination of the differentials of the residuals and increments is not obvious.

Chapter 2

Error estimation and adaptivity in optimization with partial differential equations

Error estimation for optimization problems differs from error estimation for a classical simulation (forward solution). For an optimization problem, both the cost functional and the control (for optimal control problems) have to be considered. The error estimator in [13] is extended to optimization problems. First steps and comparison with some heuristic error estimators can be found in [9]. The theoretical approach for the linear case and some of the presented results are published in [10] and [11]. A *residual-based a posteriori error estimator* will be developed for the Lagrangian approach of a nonlinear optimization problem (exploiting the structure given by the first order necessary conditions). Duality arguments are applied to get information on the global error propagation. This approach enables to bypass the problem of the determination of the (global) stability error constant arising in error estimation. Additionally for local mesh refinement, the local information from the weights seems more appropriate then the global information from the stability constant. Furthermore these local weights enable a local sensitivity control of the optimization problem. For the developed approach, the dual solutions are directly connected to the Lagrangian multiplier technique in optimization. There are two dual problems: One dual problem corresponds to the adjoint problem in the optimization approach. The second dual problem enables error estimation of a given functional. The computed state and co-state variables can be used as sensitivity factors multiplying the local cell residuals in the error estimator. An other essential new feature is a natural choice of the error functional by which the quality of the discretization of the optimization problem is measured. The presented approach also gives an automatic and natural choice of the scaling of the terms (especially those arising from boundary control and boundary observation) in the developed error estimator.

The approach to adaptivity in optimization problems will be developed within a general setting in order to abstract from inessential technicalities. Numerical results will be given in the following chapters for optimization problems with a partial differential equation-simulation from superconductivity (*Ginzburg-Landau* equations) and from flow problems (*Navier-*

Stokes equations). The adaptive mesh refinement for the finite element discretization is driven by the developed error estimator. This new error estimator will be compared to a simple energy error estimator for the state equations.

One main advantage of the presented error estimation theory is that no additional dual problem has to be built. The resulting adaptive mesh refinement is therefore almost with no additional costs (see Section 2.2).

The presented error estimation techniques have two principal points: The first aim is mesh design for economical computation (*qualitative error estimation*). The solution of the discrete system should be as close as possible to the solution of the underlying continuous problem. This can be achieved with the least number of discretization elements which is possible for a given accuracy. Or, for a given quantity of discretization elements, the to the continuous solution closest discrete solution should be obtained (in the measure given by the optimization problem).

The second aim is to know how close the solution of the discrete system is to the solution of the underlying continuous system (*quantitative error estimation*). This gives an evaluation of the quality of the solution of the discrete optimization problem by the value of the error estimator. The duality arguments are especially valuable for this application of the error estimator, because the error constants can so far not analytically be determined for all cases in a sharp sense. The effectivity index I_{eff} will be used for classification of the obtained values of the error estimators.

The developed error estimator 'measures' the error between the solution of the continuous optimization problem and the solution of the discrete optimization problem (discretization error). For nonlinear problems additionally the linearization error may be important. This does not directly mean that it 'measures' the error between the computed discrete solution and given data like observations. This latter error can be seen in the proposed dual solutions (which are an important part of the developed residual based error estimator).

This difference is not only of theoretical interest. If the calculations are done on a too coarse grid, the resulting numerical solutions may be very different from the underlying continuous solution. This will be stated in chapters 4 and 5. The accuracy of the numerical solution depends on its proposed reliability.

It will be shown that the concepts of error estimation theory for optimization problems are also valid for the nonlinear case. A generalized version for the case of arbitrary functionals will be derived in section 2.5.

The indirect approach to solve an optimization problem can be viewed as more appropriate for the presented methods in error estimation than the direct approach. The indirect approach seems to be closer to the idea of approximating the underlying continuous problem by discrete problems. But if the (Newton) iteration is done to the limit on each discretization level, this difference disappears by the reasoning in section 2.2.

The discretization of the equation system may also be viewed as a perturbation of the continuous equation system. So the classical theorems for perturbation theory as e.g. in Bock [14] and Lions [48] could be applied.

2.1 Interpretation of λ

This section is dedicated to the interpretation of the Lagrange multiplier introduced in the formulation of the Lagrangian approach. The standard interpretation (see [49]) should be indicated as well as the interpretation in the context of error estimation for optimization problems.

The gradient of the constraints is a linear combination of the gradient of the objective function by the Lagrange multipliers from the first order necessary conditions of an optimization problem for regular points, $\nabla J(x^*) = -\lambda^t \nabla F(x^*)$.

From sensitivity analysis, the Lagrange multipliers associated with a constrained minimization problem can be understood as prices, similar to the prices associated with constraints in linear programming. In the nonlinear case the Lagrange multipliers are associated with the particular solution point and correspond to incremental or marginal prices, that is, prices associated with small variations in the constraint requirements. They are the incremental prices of the constraint requirements measured in units of the objective function ($\nabla_c J(x(c))|_{c=0} = -\lambda^t$).

In general, the Lagrangian multipliers are not considered to be functions. In this context, a 'natural' interpretation of λ as a dual solution in the optimization problem is obtained. In this case, the Lagrangian multipliers are functions.

Each Lagrangian multiplier is associated to one constraint. Some of these constraints are equations from the simulation. The solutions of these equations are primal variables. The equation for calculating this type of Lagrangian multiplier is obtained by differentiation of the Lagrangian function w.r.t. a primal variable. This points out that we get the sensitivities for the solutions of the belonging equation by the Lagrangian multiplier. Strictly speaking, each Lagrangian multiplier which belongs to a state equation gives the sensitivity of this state equation with respect to the cost functional J. So also the sensitivity of the primal variable derived as solution of this state equation is obtained.

To compute the values for a Lagrangian multiplier implies that we use the values of other Lagrangian multipliers. Therefore, if the values of one Lagrangian multiplier become too large, this can have an effect on the other Lagrangian multipliers. This was one problem observed during the research for this thesis, especially in computations with cylindrical polar coordinates.

A general way to calculate the Lagrangian multiplier is $\lambda^T = J_{x_1} F_{x_1}^{-1}$, whereas J is objective function and F are the constraints as above. F is a vector and F_{x_1} is the regular part of the matrix F_x with x_1 a sub-vector of x. This formula can be derived from the proof of the necessary conditions for an optimal point. This formula was already used in the last preceding paragraph. The differentiation w.r.t. a primal variable leads to a regular part in the matrix F_x. Strictly speaking, this formula is only valid in the optimal point x^* by definition.

Normally, for each variable in the optimization problem (u, q, λ) a dual variable (z_u, z_λ, z_q) would have to be introduced. This leads to the double amount of variables. In the presented approach, the Lagrangian multiplier can be viewed as the dual solution of the solution of the equation for which the Lagrangian multiplier is introduced ($\lambda - \lambda_h = z_u$). The formulas are given in section 2.6 with the derivation of the weighted error estimator. This reduces the equation system for error estimation with an optimization problem remarkably. There

are no additional variables and no additional equation systems have to be solved. For the computation, the solution of the simulation and the Lagrangian multiplier can be adopted for the error estimation. So there are not much additional costs for the error estimation with the presented approach.

As already indicated, λ can be considered as dual solution of the optimization problem. The derivation of this fact will be given below with the derivation of the weighted error estimator. One motivation is that $\frac{\partial L}{\partial \lambda} = 0$ leads to the equation system for the primal variables u, whereas $\frac{\partial L}{\partial u} = 0$ gives the equations system for the Lagrangian multipliers.

One main interpretation in the context of error estimation is that the Lagrangian multiplier enables to measure the local error propagation. This means that the in section 2.6 derived weighted error estimator the weights $\omega(z)$ describe the dependence of the error functional $J(e)$ on variations of the residuals $\rho(u, \lambda, q)$. Therefore, for each cell T of the triangulation $\mathbf{T}_h$ the relation $\frac{\partial J(e)}{\partial \rho_T(u,\lambda,q)} \approx h_T^k \omega_T(z)$ with k depending on the finite element, the error indicator and the error functional $J(e)$ can be stated, as generally indicated in [7] and [51]. In section 2.6 the relation between the primal variables (u, λ, q) and dual variables $z = (z_u, z_\lambda, z_q)$ will be stated with the already mentioned $\lambda - \lambda_h = z_u$. A motivation for this is that the dual problem is an inverse problem and its solution is the backward solution. The error e is here the error from the difference between the solutions of the continuous and the discrete optimization problem. It is mainly the discretization error, but also the linearization error may come into play.

In the error estimation approach, the Lagrangian multiplier arises in its continuous formulation. For the used error estimator, the dual problem is replaced by the linearized dual problem. For the evaluation of the error estimator, the discrete values are taken as described in section 2.6. Therefore, problems may appear for computations on coarse grids where the used discrete Lagrangian multiplier can be 'far away' from the continuous Lagrangian multiplier. The sensitivity for the accuracy of the calculation of the Lagrangian multiplier is given in the developed calculus by $u - u_h$. The local accuracy check of the calculation of the Lagrangian multiplier is also driven by the weighted error estimator developed in section 2.6, as shown in section 2.6.2.

2.2 General model formulation for nonlinear problems

A linear version of this section can be found in [10]. The following abstract setting for optimal control will be considered: Let Q, V and H be Hilbert spaces for the control variable $q \in Q$, the state variable $u \in V$, and given observations $u_d \in H$. The inner product and norm of H are $(\cdot, \cdot)$ and $\|\cdot\|$, respectively. The state equation is given in the form

$$a(u, \phi) + b(q, \phi) = (f, \phi) \quad \forall \phi \in V, \tag{2.1}$$

where the semi-linear form $a(\cdot, \cdot)$ (linear in its second argument) represents an (elliptic) operator and the bilinear form $b(\cdot, \cdot)$ expresses the action of the control. The goal is to minimize the cost functional

$$J(u(q), q) = \tfrac{1}{2}\|cu(q) - u_d\|^2 + \tfrac{1}{2}n(q, q), \tag{2.2}$$

where $c : V \to H$ is a linear bounded observation operator. It is assumed that each $q \in Q$ defines a unique solution $u = u(q) \in V$ of (2.1) and that the resulting functional $J(\cdot)$ has the appropriate continuity and coercivity properties in order to apply the calculus of variations. For presented applications, this guarantees the existence of a unique solution of the optimal control problem and the classical regularity theory for elliptic equations applies (see, e.g., [48]). For nonlinear state equations there may be a non uniqueness of the solutions. The developed theory is still valid (see [34, p. 1004]). In this case, there can be several stationary points (e.g. local minima). The operator $n(\cdot,\cdot)$ will denote the regularization of the cost functional. It is mainly determined by the control, for example to achieve the necessary coercivity properties for the optimization problem. For simplicity, we suppose that $a(\cdot,\cdot)$ and $n(\cdot,\cdot)$ induce norms denoted by $\|\cdot\|_a$ and $\|\cdot\|_n$ on the spaces V and Q, respectively, which will be used in the following.

Introducing a Lagrangian parameter $\lambda \in V$ and the corresponding Lagrangian function $\mathcal{L}(u, q, \lambda)$, the first order necessary conditions (Euler-Lagrange equations) of the optimization problem reads

$$\begin{aligned} a'(u; v, \lambda) + (cu - u_d, cv) &= 0 \quad \forall v \in V, \\ a(u; \mu) + b(q, \mu) &= (f, \mu) \quad \forall \mu \in V, \\ -b(r, \lambda) + n(q, r) &= 0 \quad \forall r \in Q. \end{aligned} \tag{2.3}$$

The first equation results from $\frac{\partial \mathcal{L}}{\partial u} = 0$, the second from $\frac{\partial \mathcal{L}}{\partial \lambda} = 0$ and the third from $\frac{\partial \mathcal{L}}{\partial q} = 0$ as already described in section 1.1. The simulation equation is $a(u, \mu) + b(q, \mu) = (f, \mu)$. The operator $b(., \mu)$ results from the optimal control. It is generally analyzed in sections 1.1 and 1.2. For (NBC) it is just $(., \mu)_{\Gamma_Q}$. Whereas for (DBC), the strong boundary condition $u = q$ on Γ_Q is stated. The operator $n(., r)$ represents the regularization of the objective function J. The nonlinearity can be in the operator $a(.,.)$ (which is linear in the second argument, the test function) and in the objective function J.

This system leads to a non-symmetric saddle point structure

$$\begin{aligned} (cu, cv) + a'(u; v, \lambda) &= (u_d, cv) \quad \forall v \in V, \\ a(u; \mu) + b(q, \mu) &= (f, \mu) \quad \forall \mu \in V, \\ b(r, \lambda) - n(q, r) &= 0 \quad \forall r \in Q. \end{aligned} \tag{2.4}$$

For the resulting Hessian matrix (see sections 1.1 and 6.2), a symmetric saddle point structure is obtained.

For applications with *linear* operator $a(\cdot,\cdot)$, the symmetric saddle point structure is already obtained for the first order necessary condition, because $a'(u; v, \lambda) = a(\lambda; v)$.

For all linear problems and problems with special nonlinearities like in the Ginzburg-Landau equations (chapter 4) or in the Navier-Stokes equations (chapter 5), the following matrix form can be stated. Introducing operators A, A', B, C, N which represent the corresponding bilinear or nonlinear forms, system (2.4) can also be written in matrix form as

$$\begin{bmatrix} C & A' & 0 \\ A & 0 & B \\ 0 & B^T & -N \end{bmatrix} \begin{bmatrix} u \\ \lambda \\ q \end{bmatrix} = \begin{bmatrix} u_d \\ f \\ 0 \end{bmatrix}. \tag{2.5}$$

The matrix will be in the following denoted by M.

To illustrate the following theoretical considerations, the (linear) example of the first chapter will be used. For simplicity, (NBC) is taken. Let $\Omega \subset \mathbb{R}^2$ be an open bounded domain with Lipschitz boundary $\partial\Omega$ which is decomposed into a Dirichlet part Γ_D and a control part Γ_Q on which the control acts,

$$\begin{aligned} &-\Delta u = f \quad \text{in } \Omega, \\ &u = 0 \ \text{ on } \Gamma_D, \quad \partial_n u = q \ \text{ on } \Gamma_Q. \end{aligned} \tag{2.6}$$

The observations are given on a part Γ_O of the boundary and the associated cost functional is

$$J(u,q) = \tfrac{1}{2}\|u - u_d\|^2_{\Gamma_O} + \tfrac{1}{2}\alpha\|q\|^2_{\Gamma_Q}. \tag{2.7}$$

with a regularization parameter $\alpha \geq 0$. In this case the natural function spaces are $V = \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_D\}$, $H = L^2(\Gamma_O)$ and $Q = L^2(\Gamma_Q)$, whereas the operator c corresponds to the trace operator. V is the first-order Sobolev Hilbert-space over Ω. H and Q are the usual Lebesgue Hilbert-spaces over Γ_O and Γ_Q. The bilinear forms $a(\cdot,\cdot)$, $b(\cdot,\cdot)$ and $n(\cdot,\cdot)$ are given by

$$a(u,v) = (\nabla u, \nabla v)_\Omega + (u,v)_\Omega, \quad b(q,v) = (q,v)_{\Gamma_Q}, \quad n(q,r) = \alpha(q,r)_{\Gamma_Q},$$

where $(\cdot,\cdot)_\Sigma$ denotes the L^2-inner product on Σ.

The main issue of the rest of this section will be the discussion of the direct and indirect approach for solving an optimization problem in the presented context of partial differential equations with adaptive finite element Galerkin discretization (see Section 1.5).

The *indirect* approach takes the continuous formulation of the optimization problem. The discretization is done *after* the equation system (1.10) is derived. Whereas in the *direct* approach the discretization is done *before these equations are derived*. The latter means that equation (1.10) is derived for the discrete system. Therefore, the optimization problem is the one of the discrete formulation. In both cases, the same original optimization problem is considered. Only the discretization is done at another point of the derivation of the equations of the optimization system. So it can be questioned, which influence this has on the solution of the whole optimization problem and its numerical solution.

The applied discretization is based on standard finite element Galerkin techniques. So the difference between the direct and indirect Galerkin methods has to be analyzed.

In the direct Galerkin approach, an operator a_h and a matrix K_h is derived from the discretized optimization problem (see Section 1.5). For the indirect approach, this operator and this matrix are denoted by a and K, respectively. For the Galerkin approach, it is obvious that there is no difference to the equations of the indirect Galerkin approach, i.e. $a(u_h, v) = a_h(u_h, v)$ and $f = f_h$. Concerning the finite element formulation, also the same equations result because, again, $K = K_h$.

From the stated equations, it is obvious that there is no difference between the direct and the indirect Galerkin approach. The resulting equations are the same for the two approaches. This is even valid for the finite element formulation.

A difference between the direct and the indirect approach can arise in the numerical evaluation on the finite elements. On this level, it is possible to notice a difference between

the continuous operator of the indirect approach and the discrete operator of the direct approach.

2.3 A priori error estimate

This section is dedicated to state a priori error estimates for the presented applications. These are related to the operator $A(.)$ introduced in the last section. $A(.)$ will be linear, resulting from the Ginzburg-Landau equations (chapter 4) or from the Navier-Stokes equations (chapter 5). The following reasoning can be developed for the general linear case for $A(.)$. If $A(.)$ is nonlinear, the estimates have to be derived for certain classes of problems or even for only one problem, depending on the difficulty of the problem. Until now, there is no general proof in the nonlinear case known to the author.

For simplicity of notation, we introduce the space $X = V \times V \times Q$, with elements of the form $x = \{u, \lambda, q\}$ which is equipped with the product-space norm

$$\|x\|_X := \left(\|u\|_V^2 + \|\lambda\|_V^2 + \|q\|_Q^2 \right)^{1/2}.$$

Furthermore, $M(\cdot,\cdot)$ on X representing the first order necessary conditions of the optimization problem is defined by

$$\begin{aligned} &M(x,y) = M(\{u,\lambda,q\},\{v,\mu,r\}) := \\ &(cu, c\mu) + a(u,v) - b(q,v) + a'(u;\mu,\lambda) - b(r,\lambda) - n(q,r). \end{aligned}$$

Using this notation, system (2.4) can be written in compact form as

$$M(x,y) = F(y) \quad \forall y \in X, \tag{2.8}$$

with the linear functional $F(\cdot)$ defined by

$$F(y) = F(\{v,\mu,r\}) := (u_d, c\mu) + (f,v).$$

For a *linear* operator $a(.)$, the following a priori error estimates can be derived as described in [10]. In order to simplify the analysis, we impose the following conditions,

$$|M(x,y)| \le c_M \|x\|_X \|y\|_X, \tag{2.9}$$

$$|b(r,v)| \le c_b \|r\|_n \|v\|_a. \tag{2.10}$$

The second condition, which relies on the regularization term $n(\cdot,\cdot)$ (requiring that $\alpha > 0$), is rather strong. It can be substituted by an 'inf-sup'-condition for $b(\cdot,\cdot)$ under which the regularization could be omitted. $M(\cdot,\cdot)$ satisfies the following stability condition:

Proposition 2.3.1. *Under the assumptions (2.9) and (2.10) there exists a constant γ such that*

$$\inf_{x\in X} \sup_{y\in X} \frac{M(x,y)}{\|x\|_X \|y\|_X} \ge \gamma > 0. \tag{2.11}$$

Proof. For any fixed $x = \{u, \lambda, q\}$, we choose the test triple $y = \{v, \mu, r\} := \{u, \lambda, -q\}$, in order to obtain

$$\begin{aligned} M(x,y) &= \|cu\|^2 + \|u\|_a^2 + \|\lambda\|_a^2 + \|q\|_n^2 - b(q,\lambda) - b(q,u) \\ &\geq \|cu\|^2 + \|u\|_a^2 + \|\lambda\|_a^2 + \|q\|_n^2 - \frac{1}{4}\|q\|_n^2 - \frac{3}{4}\|\lambda\|_a^2 - \frac{1}{4}\|q\|_n^2 - \frac{3}{4}\|u\|_a^2 \\ &\geq \|cu\|^2 + \frac{1}{4}\|u\|_a^2 + \frac{1}{4}\|\lambda\|_a^2 + \frac{1}{2}\|q\|_n^2. \end{aligned}$$

We conclude the asserted estimate by noting that $\|y\| = \|x\|$. □

We consider the discretization of the variational equation (2.8) by a standard Galerkin method using trial spaces $X_h := V_h \times V_h \times Q_h \subset X$. For each $x \in X$, there shall exist an "interpolation" $i_h x \in X_h$, such that $\|x - i_h x\|_X \to 0\ (h \to 0)$. The discrete problem reads

$$x_h \in X_h : \quad M(x_h, y_h) = F(y_h) \quad \forall y_h \in X_h. \tag{2.12}$$

This discretization is automatically stable since a discrete analogue of (2.11) is fulfilled by the same argument as used above,

$$\inf_{x_h \in X_h} \sup_{y_h \in X_h} \frac{M(x_h, y_h)}{\|x_h\|_X \|y_h\|_X} \geq \gamma > 0. \tag{2.13}$$

Combining equations (2.12) and (2.8), we get the Galerkin orthogonality

$$M(x - x_h, y_h) = 0, \quad y_h \in X_h. \tag{2.14}$$

This leads us to the following abstract a priori error estimate.

Proposition 2.3.2. *For the Galerkin approximation on spaces $X_h \subset X$, there holds*

$$\begin{aligned} &\|u - u_h\|_a + \|\lambda - \lambda_h\|_a + \|q - q_h\|_n \\ &\qquad \leq c \Big(\inf_{v_h \in V_h} \|u - v_h\|_a + \inf_{v_h \in V_h} \|\lambda - v_h\|_a + \inf_{p_h \in Q_h} \|q - p_h\|_n \Big). \end{aligned} \tag{2.15}$$

Proof. The stability estimate (2.13) implies that

$$\gamma \|i_h x - x_h\| \leq \sup_{y_h \in X_h} \frac{M(i_h x - x_h, y_h)}{\|y_h\|_X} = \sup_{y_h \in X_h} \frac{M(i_h x - x, y_h)}{\|y_h\|_X} \leq c_M \|i_h x - x\|_X .$$

Here, we have used the Galerkin relation (2.14) and the continuity estimate (2.9). □

Of course, more precise error estimates can be given using refined arguments, which exploit the structure of the underlying problem. For instance, it would be interesting to equip the space Q with a different norm than the one induced by $n(\cdot,\cdot)$ in order to get robustness with respect to the regularization. This affords replacing (2.10) by an appropriate (weaker) inf-sup-condition like

$$\inf_{q \in Q} \left\{ \sup_{v \in V} \frac{b(q,v)}{\|v\|_a} \right\} \geq \kappa > 0.$$

It should be noted that for the model example with boundary control and boundary observations given above the conditions (2.9) and (2.10) are satisfied.

For *nonlinear* A, theoretical results already stated in [34] are used. In fact, the searched a priori error estimates can be found therein. A general theorem for a priori error estimates is theorem 3.5 on page 1013. Here some abstract results are stated for a certain class of optimization problems. For the Ginzburg-Landau equations the a priori error estimates are given in theorem 4.7 on page 1030. And for the Navier-Stokes equations they can be found in theorem 4.10 on page 1041.

2.4 Motivation: Poisson equation

The a posteriori error estimator for optimization problem which will be derived in the following sections can be motivated starting from already known facts in error estimation for the Poisson equation ([12],[13]). Taking the Poisson equation as an optimization problem, will lead to the classical a posteriori error estimator for solutions of partial differential equations.

The motivation is for the Poisson problem

$$-\Delta u \;=\; f \quad \text{in } \Omega, \qquad u|_{\partial\Omega} = 0.$$

Let $(.,.)_\Omega, \|.\|_\Omega$ denote the L^2-inner product and norm. Then the variational form seeks

$$u \in V := H_0^1(\Omega) \quad \text{such that} \quad (\nabla u, \nabla\phi)_\Omega = (f,\phi)_\Omega \;\; \forall \phi \in V.$$

Let $\mathbf{T}_h$ be a triangulation of Ω. We define the subspace $V_h \subset V$ as

$$V_h = \{\phi \in V : \; \phi|_K \in Q_1(K) \; \forall K \in \mathbf{T}_h\}.$$

This leads to the following finite element Galerkin approximation for the above variational form:

$$u_h \in V_h : \quad (\nabla u_h, \nabla\phi_h)_\Omega = (f,\phi_h)_\Omega \;\; \forall \phi_h \in V_h.$$

Defining the error $e = u - u_h$, the Galerkin orthogonality can be stated (which is an essential feature):

$$(\nabla e, \nabla\phi_h)_\Omega = 0 \;\; \forall \phi_h \in V_h.$$

The a posteriori error estimation is done with respect to the (linear) functional output

$$|J(u) - J(u_h)| \leq \mathrm{TOL}.$$

The corresponding dual problem is:

$$z \in V : \quad (\nabla\phi, \nabla z)_\Omega = J(\phi) \;\; \forall \phi \in V. \tag{2.16}$$

Taking the error representation for the test function equal to the error ($\phi = e$) and using the Galerkin orthogonality and cell wise integration by parts leads to

$$\begin{aligned} J(e) &= (\nabla e, \nabla z)_\Omega = (\nabla e, \nabla(z - z_h))_\Omega = (f, z - z_h)_\Omega - (\nabla u_h, \nabla(z - z_h))_\Omega \\ &= \sum_{K \in \mathbf{T}_h} \{(f + \Delta u, z - z_h)_K - \frac{1}{2}([\partial_n u_h], z - z_h)_{\partial K}\} \qquad (2.17) \\ &\leq \sum_{K \in \mathbf{T}_h} \{\|f + \Delta u\|_K \|z - z_h\|_K + \frac{1}{2}\|[\partial_n u_h]\|_{\partial K} \|z - z_h\|_{\partial K}\} \\ &\approx \sum_{K \in \mathbf{T}_h} \{\|f + \Delta u\|_K \|z - z_h\|_K + \frac{1}{2}\rho_K(u_h)\omega_K(z)\}. \end{aligned}$$

when taking only the dominant terms after the inequality. The jump over the cell K is denoted by [.]. The cell residuals

$$\rho_K(u_h) = \frac{1}{2} h_K^{\frac{1}{2}} \|[\partial_n u_h]\|_{\partial K}$$

can be interpreted as *smoothness measure* of the solution. Whereas the weights

$$\omega_K(z) \;\leq\; C_i h_K \|\nabla^2 z\|_K \;\approx\; h_K^{\frac{1}{2}} \|[\partial_n z_h]\|_{\partial K}$$

represent *sensitivity factors of* $J(e)$. The full a posteriori error estimate would be by equation (2.17):

$$|J(e)| \leq \eta(u_h) := \sum_{K \in \mathbf{T}_h} h_K^2 \left\{\rho_K^{(u)} \omega_K^{(z)} + \rho_{\partial K}^{(u)} \omega_{\partial K}^{(z)}\right\}, \qquad (2.18)$$

with the cell residuals and weights

$$\begin{aligned} \rho_K^{(u)} &:= h_K^{-1} \|f + \Delta u_h\|_K, & \rho_{\partial K}^{(u)} &:= \tfrac{1}{2} h_K^{-3/2} \|n \cdot [\nabla u_h]\|_{\partial K \setminus \partial\Omega}, \\ \omega_K^{(z)} &:= h_K^{-1} \|z - z_h\|_K, & \omega_{\partial K}^{(z)} &:= h_K^{-1/2} \|z - z_h\|_{\partial K \setminus \partial\Omega}. \end{aligned}$$

In view of the local approximation properties of finite elements, there holds

$$\omega_K^{(z)} + \omega_{\partial K}^{(z)} \leq c_I h_K^2 \max_K |\nabla^2 z|\,. \qquad (2.19)$$

In practice the weights $\omega_K^{(z)}$, $\omega_{\partial K}^{(z)}$ have to be determined computationally. Let $z_h \in H_h$ be the Galerkin approximation of z defined by

$$(\nabla \phi_h, \nabla z_h)_\Omega = J(\phi_h) \quad \forall \phi_h \in V_h. \qquad (2.20)$$

In view of the estimate (2.19), we can approximate

$$\omega_K^{(z)} + \omega_{\partial K}^{(z)} \approx c_I h_K^2 \max_K |\nabla_h^2 z_h|, \qquad (2.21)$$

where $\nabla_h z_h$ is a suitable difference quotient approximating $\nabla^2 z$. The interpolation constant is usually in the range $c_I \approx 0.1$ to 1 and can be determined by calibration. Alternatively, we may construct from $z_h \in H_h$ a patch wise bi-quadratic extrapolation $I_h^2 z_h$ and replace $z - \phi_h$ in the weights by $I_h^2 z_h - z_h$. This gives an approximation which is free of any interpolation constant. The quality of these approximations for the model problem has been analyzed in [13].

This derivation can be reformulated as an optimization problem. The results can be seen as a motivation for the following sections on error estimation for optimization problems. Let the weak formulation of the Poisson equation be the functional to be minimized (i.e. the cost functional of the optimization problem):

$$\min_{u \in V = H_0^1(\Omega)} F(u) := \frac{1}{2}\|\nabla u\|_\Omega^2 - (f,u)_\Omega.$$

Considering the error $e := u - u_h$ results

$$\begin{aligned} F(u) - F(u_h) &= \frac{1}{2}\|\nabla u\|_\Omega^2 - (f,u)_\Omega - \frac{1}{2}\|\nabla u_h\|_\Omega^2 + (f,u_h)_\Omega \\ &= -\frac{1}{2}\|\nabla u\|_\Omega^2 - \frac{1}{2}\|\nabla u_h\|_\Omega^2 + (\nabla u, \nabla u_h)_\Omega = -\frac{1}{2}\|\nabla e\|_\Omega^2. \end{aligned}$$

In this case, energy-error control means error control with respect to the 'cost functional' F. Exploiting the Galerkin orthogonality for the first equation leads to a motivation for the error functional

$$G(e) := -\frac{1}{2}\|\nabla e\|_\Omega^2 = -\frac{1}{2}(\nabla e, \nabla u)_\Omega = F(u) - F(u_h).$$

By the definition of the dual problem (2.16), we get

$$(\nabla z, \nabla \phi)_\Omega = -\frac{1}{2}(\nabla e, \nabla \phi)_\Omega$$

leading to the dual solution

$$z = -\frac{1}{2}e.$$

The general a posteriori error estimate (2.18) takes the particular form

$$\|\nabla e\|_\Omega^2 \le \sum_{K \in \mathbf{T}_h} h_K^2 \{\rho_K^{(u)} \omega_K^{(u)} + \rho_{\partial K}^{(u)} \omega_{\partial K}^{(u)}\}, \tag{2.22}$$

with the weights $\omega_K^{(u)} = h_K^{-1}\|u - \phi_h\|_K$ and $\omega_{\partial K}^{(u)} = h_K^{-1/2}\|u - \phi_h\|_{\partial K \setminus \partial\Omega}$. Then, using the local approximation estimate

$$\inf_{\phi_h \in H_h} \Big(\sum_{K \in \mathbf{T}_h} \{h_K^{-2}\|u - \phi_h\|_K^2 + h_K^{-1}\|u - \phi_h\|_{\partial K}^2\} \Big)^{1/2} \le c_I \|\nabla e\|_\Omega, \tag{2.23}$$

it can be concluded from (2.22) that

$$\|\nabla e\|_\Omega^2 \le c_I \Big(\sum_{K \in \mathbf{T}_h} h_K^4 \{\rho_K^{(u)2} + \rho_{\partial K}^{(u)2}\} \Big)^{1/2} \|\nabla e\|_\Omega.$$

This implies the standard residual–based energy-norm a posteriori error estimate (see, e.g. Verfürth [62]):

$$|F(u) - F(u_h)| = \tfrac{1}{2}\|\nabla e\|_\Omega^2 \leq \tfrac{1}{2}c_I^2 \sum_{K\in\mathbf{T}_h} h_K^4 \left\{\rho_K^{(u)2} + \rho_{\partial K}^{(u)2}\right\}. \tag{2.24}$$

Below, it will be shown that the peculiar relation $z = -\frac{1}{2}e$ for the dual solution corresponding to the "energy functional" $F(\cdot)$ follows from a general principle which can be used also for the discretization of the optimal control problem described above.

2.5 General approach to a posteriori error analysis

In this section, two abstract and general approaches to a posteriori error analysis are presented. Applying these approaches to optimization problems leads to the presented dual-weighted a posteriori error estimator. Both approaches are not restricted to optimization problems. Starting from a (possibly nonlinear) functional, the described mechanism can be applied. The first version has been published in [11]. For linear or quadratic problems, the error estimate can be exact. Otherwise, there is an additional error. This error will be given in the second version in this section by the remainder R (see [10]).

Let $L(u)$ be a twice differentiable functional on some Hilbert space V, e.g. the energy functional related to the Poisson problem or the Lagrangian functional defined for the Ginzburg-Landau model. For its first and second differentials at u, the notation $L'(u;\cdot)$ and $L''(u;\cdot,\cdot)$, respectively, is used. Notice that $L''(u;\cdot,\cdot)$ is symmetric. Stationary points $u \in V$ of $L(\cdot)$ are searched,

$$L'(u;\phi) = 0 \quad \forall \phi \in V. \tag{2.25}$$

For an optimization problem, this equation is the first order necessary condition of the underlying original continuous optimization problem, i.e. the equation which has to be solved. Corresponding approximations $u_h \in V_h$ are defined in finite dimensional subspaces $V_h \subset V$ by the Galerkin equations

$$L'(u_h;\phi_h) = 0 \quad \forall \phi_h \in V_h. \tag{2.26}$$

Let $J(\cdot)$ be a functional chosen for measuring the error $e = u - u_h$. Then,

$$J(u) - J(u_h) = \int_0^1 J'(u_h + te; e)\, dt, \tag{2.27}$$

$$L'(u;\cdot) - L'(u_h;\cdot) = \int_0^1 L''(u_h + te; e, \cdot)\, dt, \tag{2.28}$$

leads to consider the "dual problem"

$$\int_0^1 L''(u_h + te; \phi, z)\, dt = \int_0^1 J'(u_h + te; \phi)\, dt \quad \forall \phi \in V, \tag{2.29}$$

which is assumed to have a solution $z \in V$. Then, taking $\phi = e$ in (2.29) and using the Galerkin equation (2.26) results in the error identity

$$J(u) - J(u_h) = L'(u; z) - L'(u_h; z) = -L'(u_h; z - \phi_h), \tag{2.30}$$

with arbitrary $\phi_h \in V_h$. In general, this error representation cannot be evaluated since the left-hand side as well as the right-hand side in the dual problem (2.29) depend on the unknown continuous solution u. The simplest way of approximation is to replace u by u_h, which yields the perturbed dual problem

$$L''(u_h; \phi, \tilde{z}) = J'(u_h; \phi) \quad \forall \phi \in V. \tag{2.31}$$

Controlling the effect of this perturbation on the accuracy of the resulting error estimate may be a delicate task and depends strongly on the particular problem under consideration. Experiences from different types of applications (e.g. the Navier-Stokes equations) indicate that this problem is not critical as long as the solution to be computed is stable. The crucial problem is the approximation of the perturbed dual solution by solving a discrete dual problem

$$L''(u_h; \phi_h, \tilde{z}_h) = J'(u_h; \phi_h) \quad \forall \phi_h \in V_h. \tag{2.32}$$

So far, the derivation of the error representation (2.30) did not use that the variational equation (2.25) stems from an "energy functional". In fact it can be used for much more general situations; see the surveys given in Eriksson, et al. [27], and in [51]. It seems natural to control the error $e = u - u_h$ with respect to the given "energy" functional $L(\cdot)$. Observing that $L'(u; \phi) = 0$, it follows by integration by parts that

$$\int_0^1 L'(u_h + te; \phi)\, dt = -\int_0^1 L''(u_h + te; e, \phi) t\, dt = -\int_0^1 L''(u_h + te; \phi, e) t\, dt.$$

Hence, in this case the dual problem (2.29) takes the special form

$$\int_0^1 L''(u_h + te; \phi, z)\, dt = -\int_0^1 L''(u_h + te; \phi, e) t\, dt \quad \forall \phi \in V. \tag{2.33}$$

If the functional $L(\cdot)$ is quadratic or in the general case by linearization $u \to u_h$, the following perturbed dual problem is obtained

$$L''(u_h; \phi, \tilde{z}) = -\tfrac{1}{2} L''(u_h; \phi, e) \quad \forall \phi \in V, \tag{2.34}$$

with the solution $\tilde{z} = -\frac{1}{2} e$. The resulting a posteriori error estimate has the form

$$|L(u) - L(u_h)| \approx \inf_{\phi_h \in V_h} |L'(u_h; \tilde{z} - \phi_h)| = \inf_{\phi_h \in V_h} \tfrac{1}{2} |L'(u_h; \tilde{u} - \phi_h)|. \tag{2.35}$$

In the ideal case of a quadratic functional $L(\cdot)$ linearization is not required and this error bound becomes exact. Here, again the quantity

$$\tilde{z} - \phi_h = -\tfrac{1}{2} e - \phi_h = \tfrac{1}{2}(u - \psi_h)$$

has to be approximated as described above by using the computed solution $u_h \in H_h$.

It should be emphasized that in this particular case the evaluation of the a posteriori error estimate with respect to the "energy functional" does not require the explicit solution of the dual problem. This abstract reasoning can be taken as guide-line for systematically deriving a posteriori error estimates in concrete situations, for example for optimization problems in the following chapters.

Remark 2.5.1. *The factor $-\frac{1}{2}$ results from the difference between the first and the second order of differentiation. Alternatively, it can be found in the Taylor approximation in equation (2.51). It could be eliminated by a multiplication of the cost functional with factor 2. The standard notation in optimization theory is the first one.*

Following the formalism of this and the previous chapter, the estimation of the error $e = \{e_u, e_\lambda, e_q\}$ with respect to the Lagrangian functional $L(\cdot)$ is searched. The corresponding linearized dual problem

$$L''(u_h; \phi, \tilde{z}) = -\tfrac{1}{2} L''(u_h; \phi, e) \quad \forall \phi \in V, \tag{2.36}$$

then has the solution $\tilde{z} = -\frac{1}{2}\{e_u, e_\lambda, e_q\}$. Hence, this dual problem has not to be built (nor extra work for solving it has to be spent). The following result is also true for nonlinear state equations as for example shown in chapter 4.

Theorem 2.5.1. *For the finite element discretization of the variational equation (1.7) - (1.9) for the considered optimization problem, there holds the a posteriori error relation*

$$|J(u,q) - J(u_h,q_h)| \;\le\; \eta_\omega(u_h,\lambda_h,q_h) = \sum_{K\in\mathbf{T}_h} \eta_K(u_h,\lambda_h,q_h), \tag{2.37}$$

with the local error indicators

$$\eta_K(u_h,\lambda_h,q_h) := \rho_K^{(u)}\,\omega_K^{(\lambda)} + \rho_{\partial K}^{(u)}\,\omega_{\partial K}^{(\lambda)} + \rho_K^{(\lambda)}\,\omega_K^{(u)} + \rho_{\partial K}^{(\lambda)}\,\omega_{\partial K}^{(u)} + \rho_{\partial K}^{(q)}\,\omega_{\partial K}^{(q)}\,.$$

and the cell-wise residuals and weights

$$\begin{aligned}
\rho_K^{(u)} &:= \|R_h^{(u)}\|_K, & \omega_K^{(\lambda)} &:= \|\lambda - i_h\lambda\|_K,\\
\rho_{\partial K}^{(u)} &:= \|r_h^{(u)}\|_{\partial K}, & \omega_{\partial K}^{(\lambda)} &:= \|\lambda - i_h\lambda\|_{\partial K},\\
\rho_K^{(\lambda)} &:= \|R_h^{(\lambda)}\|_K, & \omega_K^{(u)} &:= \|u - i_h u\|_K,\\
\rho_{\partial K}^{(\lambda)} &:= \|r_h^{(\lambda)}\|_{\partial K}, & \omega_{\partial K}^{(u)} &:= \|u - i_h u\|_{\partial K},\\
\rho_{\partial K}^{(q)} &:= \|r_h^{(q)}\|_{\partial K\cap\Gamma_C}, & \omega_{\partial K}^{(q)} &:= \|q - j_h q\|_{\partial K\cap\Gamma_C},
\end{aligned}$$

The "cell residuals" $R_h^{(u)}$, $R_h^{(\lambda)}$, and the "edge residuals" $r_h^{(u)}$, $r_h^{(\lambda)}$, $r_h^{(q)}$, are on cells K and cell edges Γ defined by

$$R_{h|K}^{(u)} := -\Delta u_h + u_h - f, \quad R_{h|K}^{(\lambda)} := -\Delta\lambda_h + \lambda_h, \qquad r_{h|\Gamma}^{(q)} := \alpha q_h - \lambda_h, \;\; \text{if } \Gamma\subset\Gamma_C,$$

$$r_{h|\Gamma}^{(u)} := \begin{cases} \frac{1}{2}h_\Gamma^{-1/2}[\partial_n\phi_h], & \text{if } \Gamma\subset\partial K\setminus\partial\Omega,\\ h_\Gamma^{-1/2}\partial_n u_h, & \text{if } \Gamma\subset\partial\Omega\setminus\Gamma_C,\\ h_\Gamma^{-1/2}(\partial_n u_h - q_h), & \text{if } \Gamma\subset\Gamma_C, \end{cases} \qquad r_{h|\Gamma}^{(\lambda)} := \begin{cases} \frac{1}{2}h_\Gamma^{-1/2}[\partial_n\phi_h], & \text{if } \Gamma\subset\partial K\setminus\partial\Omega,\\ h_\Gamma^{-1/2}\partial_n\lambda_h, & \text{if } \Gamma\subset\partial\Omega\setminus\Gamma_O,\\ h_\Gamma^{-1/2}(c_0 - u_h + \partial_n\lambda_h), & \text{if } \Gamma\subset\Gamma_O. \end{cases}$$

Here, $[\partial_n \phi_h]$ denotes the jump of the normal derivative of ϕ_h across the inter-element edges Γ, the boundary components Γ_C, Γ_O are the control and observation boundary, respectively, and i_h, j_h denote some local interpolation operators into the finite element spaces.

If the Lagrangian functional $L(\cdot)$ is quadratic this error relation yields a true upper bound.

Proof. In the present case, there holds

$$\begin{aligned} L(v) - L(v_h) &= J(u,q) + (\nabla u, \nabla\lambda)_\Omega - (f,\lambda)_\Omega - (q,\lambda)_{\Gamma_C} \\ &\quad - J(u_h,q_h) - (\nabla u_h, \nabla\lambda_h)_\Omega + (f,\lambda_h)_\Omega + (q_h,\lambda_h)_{\Gamma_C} \\ &= J(u,q) - J(u_h,q_h), \end{aligned}$$

since $\{u,\lambda,q\}$ and $\{u_h,\lambda_h,q_h\}$ satisfy the continuous and discrete version of equation (1.9). Hence, error control with respect to the Lagrangian functional $L(\cdot)$ and the cost functional $J(\cdot)$ is equivalent. Now, the general error identity (2.35) implies that

$$|J(u,q) - J(u_h,q_h)| \approx \inf_{\phi_h \in V_h} |L'(v_h; v - \phi_h)|, \tag{2.38}$$

where $v_h = \{u_h,\lambda_h,q_h\}$ and $v = \{u,\lambda,q\}$. Notice that this relation is an identity if the functional $J(\cdot)$ is quadratic. From the discrete version of (1.7) - (1.9), it results that

$$\begin{aligned} L'(v_h; v - \phi_h) &= (u_h - u_O, u - \psi_h)_{\Gamma_O} + (\nabla(u - \psi_h), \nabla\lambda_h)_\Omega + (u - \psi_h, \lambda_h)_\Omega \\ &\quad + (\nabla u_h, \nabla(\lambda - \pi_h))_\Omega - (f, \lambda - \pi_h)_\Omega - (q_h, \lambda - \pi_h)_{\Gamma_C} \\ &\quad + (\lambda_h - \alpha q_h, q - \chi_h)_{\Gamma_C}. \end{aligned}$$

Splitting the global integrals into the contributions from each single cell $K \in \mathbf{T}_h$ and each cell edge $\Gamma \subset \partial\Omega$, respectively, and integrating locally by parts yields

$$\begin{aligned} L'(v_h; v - \phi_h) &= \sum_{\Gamma\subset\Gamma_O} (u_h - u_O + \partial_n\lambda_h, u - \psi_h)_\Gamma + \sum_{\Gamma\subset\partial\Omega\setminus\Gamma_O} (\partial_n\lambda_h, u - \psi_h)_\Gamma \\ &\quad + \sum_{\Gamma\subset\Gamma_C} (\partial_n u_h - q_h, \lambda - \pi_h)_\Gamma + \sum_{\Gamma\subset\partial\Omega\setminus\Gamma_C} (\partial_n u_h, \lambda - \pi_h)_\Gamma \\ &\quad + \sum_{\Gamma\subset\Gamma_C} (\lambda_h - \alpha q_h, q - \chi_h)_\Gamma \\ &\quad + \sum_{K\in\mathbf{T}_h} \left\{(-\Delta u_h - f, \lambda - \pi_h)_K + \tfrac{1}{2}(n\cdot[\nabla u_h], \lambda - \pi_h)_{\partial K\setminus\partial\Omega}\right\} \\ &\quad + \sum_{K\in\mathbf{T}_h} \left\{(u - \psi_h, -\Delta\lambda_h + \lambda_h)_K + \tfrac{1}{2}(u - \psi_h, n\cdot[\nabla\lambda_h])_{\partial K\setminus\partial\Omega}\right\}. \end{aligned}$$

From this the asserted relation follows by applying the Hölder inequality. □

The linearization error can be given in an explicit form. In the rest of this section this will be shown.

The a posteriori error estimation in the case of a nonlinear state equation follows the same pattern as in the linear case. First, an abstract result is stated.

Proposition 2.5.1. *For the Galerkin finite element approximation of the abstract model problem (2.3) with nonlinear state equation and quadratic cost functional there holds*

$$J(u,q) - J(u_h,q_h) = \tfrac{1}{2}\nabla\mathcal{L}(x_h)(x - i_h x) + R(x,x_h)\,, \tag{2.39}$$

where the remainder term $R(x,x_h)$ *can be estimated by*

$$|R(x,x_h)| \;\leq\; \sup_{\hat{x}\in[x_h,x]} |\nabla^3\mathcal{L}(\hat{x})(x-x_h,x-x_h,x-x_h)|. \tag{2.40}$$

Proof. The Galerkin orthogonality relation now reads

$$\nabla^2\mathcal{L}(\overline{xx_h})(x-x_h,\phi_h) = \nabla\mathcal{L}(x)(\phi_h) - \nabla\mathcal{L}(x_h)(\phi_h) = 0, \quad \phi_h \in X_h, \tag{2.41}$$

with the abbreviating notation

$$\mathcal{L}(\overline{xx_h}) := \int_0^1 \mathcal{L}(x + t(x_h - x))\,dt.$$

Since the solutions u and u_h satisfy the corresponding state equations there holds again

$$J(u,q) - J(u_h,q_h) = \mathcal{L}(x) - \mathcal{L}(x_h).$$

By Taylor expansion, there holds

$$\begin{aligned}\mathcal{L}(x) - \mathcal{L}(x_h) = \nabla\mathcal{L}(x)(x-x_h) &- \tfrac{1}{2}\nabla^2\mathcal{L}(x)(x-x_h,x-x_h)\\ &+\tfrac{1}{6}\nabla^3\mathcal{L}(\tilde{x})(x-x_h,x-x_h,x-x_h),\end{aligned}$$

where $\tilde{x}$ lies between x and x_h. Since x is a stationary point of $\mathcal{L}$, the first term on the right vanishes. In order to relate the second term to the Galerkin relation (2.41), again Taylor expansion is used:

$$\begin{aligned}\nabla^2\mathcal{L}(x)(x-x_h,x-x_h) = \nabla^2\mathcal{L}(\overline{xx_h})&(x-x_h,x-x_h)\\ &+\nabla^3\mathcal{L}(\hat{x})(x-x_h,x-x_h,x-x_h),\end{aligned}$$

where $\hat{x}$ is another point between x and x_h. In view of the identity

$$\nabla^2\mathcal{L}(\overline{xx_h})(x-x_h,\cdot) = \nabla\mathcal{L}(x)(\cdot) - \nabla\mathcal{L}(x_h)(\cdot) = -\nabla\mathcal{L}(x_h)(\cdot),$$

and the Galerkin relation (2.41), it can be concluded that

$$\begin{aligned}\mathcal{L}(x) - \mathcal{L}(x_h) &= -\tfrac{1}{2}\nabla^2\mathcal{L}(\overline{xx_h})(x-x_h,x-x_h) + R(x,x_h)\\ &= -\tfrac{1}{2}\nabla^2\mathcal{L}(\overline{xx_h})(x-x_h,x-x_h-\phi_h) + R(x,x_h)\\ &= \tfrac{1}{2}\nabla\mathcal{L}(x_h)(x-x_h-\phi_h) + R(x,x_h),\end{aligned}$$

with an arbitrary $\phi_h \in X_h$, and the remainder term

$$R(x,x_h) = \nabla^3\mathcal{L}(\hat{x})(x-x_h,x-x_h,x-x_h) + \tfrac{1}{6}\nabla^3\mathcal{L}(\tilde{x})(x-x_h,x-x_h,x-x_h).$$

Taking here $\phi_h = i_h x - x_h$, eventually results in

$$\mathcal{L}(x) - \mathcal{L}(x_h) = \tfrac{1}{2}\nabla\mathcal{L}(x_h)(x - i_h x) + R(x,x_h),$$

which completes the proof. □

It should be noted that, if the cost functional $J(\cdot)$ is quadratic and the control form $b(\cdot,\cdot)$ bilinear, then the only non-zero terms in $\nabla^3\mathcal{L}$ are

$$\frac{\partial^3\mathcal{L}}{\partial\lambda\partial^2 u}(x) = a''(u)(\cdot,\cdot,\cdot), \qquad \frac{\partial^3\mathcal{L}}{\partial^3 u}(x) = a'''(u)(\cdot,\cdot,\cdot,\lambda).$$

Further, if additionally the state equation is linear, then the remainder term $R(x,x_h)$ vanishes.

This abstract result will be applied for a nonlinear problem of optimal control in the "Ginzburg-Landau model" of superconductivity in semiconductors in chapter 4. It has the same structure as the model problem considered above,

$$\begin{aligned} &-\Delta u + s(u) = f \quad \text{in } \Omega, \\ &\partial_n u = 0 \ \text{ on } \Gamma_N, \quad \partial_n u = q \ \text{ on } \Gamma_C, \end{aligned} \tag{2.42}$$

with the nonlinearity $s(u) := u^3 - u$, and the quadratic cost functional

$$J(u,q) = \tfrac{1}{2}\|u - c_0\|^2_{\Gamma_O} + \tfrac{\alpha}{2}\|q\|^2_{\Gamma_C}.$$

The corresponding first-order necessary condition (2.3) uses the notation

$$a(u)(v) = (\nabla u, \nabla v)_\Omega + (s(u), v)_\Omega, \quad b(q,v) = (q,v)_{\Gamma_C}, \quad n(q,r) = \alpha(q,r)_{\Gamma_C},$$

and is approximated by the Galerkin finite element approximation of the scheme (2.3). The well-posedness of this optimization problem, the existence of the adjoint variable λ, as well as a priori error estimates for its discretization have been discussed by Gunzburger and Hou [34]. From Proposition 2.5.1, we conclude the following a posteriori result.

Proposition 2.5.2. *For error control with respect to the cost functional J, there holds the weighted* a posteriori *error estimate*

$$|J(u,q) - J(u_h,q_h)| \ \le \ \eta_\omega(u_h,\lambda_h,q_h) + R(\{u,\lambda,q\},\{u_h,\lambda_h,q_h\}), \tag{2.43}$$

where the local error indicators $\eta_K(u_h,\lambda_h,q_h)$ in the linearized error estimator

$$\eta_\omega(u_h,\lambda_h,q_h) := \sum_{K\in\mathbf{T}_h} \eta_K(u_h,\lambda_h,q_h) \tag{2.44}$$

are defined as in the linear case, here with the "cell residuals"

$$\begin{aligned} R^{(u)}_{h|K} &:= -\Delta u_h + s(u_h) - f, \quad R^{(\lambda)}_{h|K} := -\Delta\lambda_h + s'(u_h)\lambda_h, \\ &\qquad\qquad r^{(q)}_{h|\Gamma} := \alpha q_h - \lambda_h, \ \ \textit{if } \Gamma \subset \Gamma_C. \end{aligned} \tag{2.45}$$

For the remainder term, there holds the a priori estimate

$$\big|R(\{u,\lambda,q\},\{u_h,\lambda_h,q_h\})\big| \ \le \ 6\int_\Omega \Big\{ \max\{|u|,|u_h|\}|u-u_h|^3 + |u-u_h|^2|\lambda-\lambda_h|\Big\}\,dx. \tag{2.46}$$

As in the linear case, the weights are evaluated numerically using the approximations $\{u_h, \lambda_h, q_h\}$, but now the weighted error estimator contains an additional linearization error represented by the remainder R. Theory as well as practical experience show that, in the present case, this additional error is of higher order on well-adapted meshes and can therefore be neglected. In fact, assuming sufficient smoothness of the solution $\{u, \lambda, q\}$, there holds

$$\left|R(\{u, \lambda, q\}, \{u_h, \lambda_h, q_h\})\right| \leq c(u, u_h)\, h_{\max}^6, \tag{2.47}$$

with the maximum step size $h_{\max}$ of the mesh. The proof of this order-optimal estimate employing techniques from L^∞-error analysis of finite elements could be given by known techniques which are beyond the topic of this thesis. In view of this observation, the remainder term in the a posteriori error estimate (2.43) is neglected and base the mesh adaptation on its main part $\eta_\omega(u_h, \lambda_h, q_h)$.

The discrete problems of (2.3) are solved by a quasi-Newton iteration which is derived from a corresponding scheme formulated on the continuous level. On each discrete level the Newton iteration is carried to the limit before the error estimator is applied for mesh refinement. The results of this process may significantly differ from those obtained if each Newton step is discretized separately mixing iteration and discretization errors together; see the publication [9] for the latter approach.

2.6 Derivation of the dual weighted error estimator for optimization problems

In the last section, the dual-weighted error estimator was derived. In section, some additional remarks for the case of optimization problems should be given.

As already stated, the primal optimization problem in the weak formulation reads

$$y \in X := V \times V' \times Q: \; M(y, v) = F(v) \quad \forall v \in X, \tag{2.48}$$

leading to a primal solution $y = (u, \lambda, q)$ of the optimization problem. Following the general theory for dual problems, a dual problem fitting to the error estimation problem $y - y_h = (u - u_h, \lambda - \lambda_h, q - q_h)$ can be constructed. Let $G(.)$ be a general *linear* error functional $G(\cdot) = \{G_u(\cdot), G_\lambda(\cdot), G_q(\cdot)\}$ defined on X. $G(.)$ is linear in the error $y - y_h$, but not necessarily in the variables y. In order to obtain an *a posteriori* error estimator for $G(y - y_h)$, the following corresponding dual problem has to be considered:

$$z \in X: \quad M^t(z, x) = G(x) \quad \forall x \in X \tag{2.49}$$

with the dual solution $z = (z_u, z_\lambda, z_q)$ of the optimization problem. In equation (2.49), x is an arbitrary test function of X. It will later on be set to the error $y - y_h$.

For the special case of $G(x) = J(y) - J(y_h)$ with J being the objective functional of the optimization problem, the following approach can be derived: The Lagrangian functional $\mathcal{L}(y) = J(u, q) + \langle Au - Bq, \lambda \rangle$ is stationary at the continuous solution $y = \{u, \lambda, q\}$ and the discrete solution y_h. This leads for the difference between the continuous and the discrete

version to

$$\begin{aligned} J(u,q) - J(u_h,q_h) &= \mathcal{L}(u,\lambda,q) - \mathcal{L}(u_h,\lambda_h,q_h) && (2.50)\\ &= \nabla\mathcal{L}(y)(y-y_h) + \tfrac{1}{2}\nabla^2\mathcal{L}(y)(y-y_h)^2 + O(\|y-y_h\|_X^3) \\ &= \tfrac{1}{2}\nabla^2\mathcal{L}(y)(y-y_h)^2 + O(\|y-y_h\|_X^3). && (2.51)\end{aligned}$$

$A(.)$ can be nonlinear. The continuous and discrete state equations must be equal to 0 for equation (2.50). In equation (2.51), the first order necessary condition for the Lagrange function $\mathcal{L}$ is used. Until now, only the primal optimization problem has been exploited.

With the choice $G(x) = J(y) - J(y_h)$, an interpretation of equation (2.51) by the dual problem can be found, leading to an error estimate. The above matrix M is the matrix of the first order necessary conditions. Thus it contains the first order differential of $\mathcal{L}$. This matrix M is not symmetric for nonlinear A. But the second order differential of $\mathcal{L}$ is symmetric. So the matrix $\nabla^2\mathcal{L}$ can also be interpreted as an equation by the dual problem M^t. Taking $G(x) = J(y) - J(y_h)$, an equation for the estimation of the error functional $G(.)$ was derived. The variable x is now the error $y - y_h$, so this test function is taken.

Now, also the relation between the primal and the dual solutions can be derived. The dual variable of λ is $z_\lambda = u - u_h$ because u, u_h appear in the error functional $J(u,q) - J(u_h,q_h)$ and the considered (dual) problem is M^t and not M. By the same argument, $z_u = \lambda - \lambda_h$ can be stated. This means that the dual variable z can be expressed in terms of the primal variable y. Hence no extra dual variables need not be generated for the computation. This reduces the system on the half of the variables of the whole system for adaptivity. And error estimation is almost "for free".

Remark 2.6.1. *In theorem 2.5.1 an error estimate similar to the linear will be derived for the general case. The derived estimate is an approximation and will be an upper bound in (the given) case of a quadratic cost functional.*

Remark 2.6.2. *In the* a posteriori *error estimate (2.37), the residual of the state equation is weighted by terms involving the Lagrangian multiplier λ from the original equation* (2.4)*. This has a natural interpretation as it is well-known from sensitivity analysis that the Lagrangian multiplier measures the influence of perturbations on the cost functional. Since discretization can be interpreted as a special perturbation, the appearance of λ in the estimator is not surprising. The special form of the weights involving the interpolation $i_h z$ is a characteristic feature of the Galerkin discretization (orthogonality of residuals with respect to the test space).*

Remark 2.6.3. *The* a posteriori *error estimate (2.37) is derived from the first-order optimality condition which is a system of partial differential equations. An interpretation in terms of the original minimization problem can be very illuminative. Indeed, the discretization of the state equation leads to numerical solutions which are not admissible (in the strict sense) for the original constrained minimization problem. The situation can be summarized as follows: Let $s : Q \to V$ denote the (linear) solution operator which associates the state variable to a given control function. The optimal control then minimizes the functional $j(q) := J(s(q), q)$ without constraints over the space Q. Since the discretization changes the state equation, not only the space of possible controls is changed, but also the functional. Denoting by $s_h : Q \to V_h$ the discrete solution operator, the discrete optimal*

control q_h minimizes the functional $j_h(q) := J(s_h(q), q)$ over the space Q_h. If numerical computation is performed, the notion of "admissible" solution has to be substituted by an error estimate for the state equation. Of course, the distance between the numerical and the continuous state should be measured with respect to the specific needs of the optimization problem, i.e., the influence on the functional to be minimized. This is exactly what the a posteriori *estimator derived above is designed for.*

2.6.1 Error functional for optimization problems

In the general approach in section 2.6, the error functional $G(.)$ is not forced to be related to the cost functional $J(.)$ of the optimization problem. For special optimization problems, there may exist good error functionals which are not related to the cost functional $J(.)$. The principal point of this approach is that a *general approach* for error estimation of optimization problems should be developed. Furthermore, the presented error functional results from the analytic derivation of the dual-weighted error estimate.

The sometimes for optimization problems applied strategy that the cost functional is mainly used as regularization to get a well-stated state equation is not considered. For this section, the cost functional $J(.)$ is really the function which gives the quality of the obtained solution of the continuous and discretized optimization problem. The error functional $G(.)$ shows the quality of the discrete solution of the (discretized) optimization problem. These two qualities are very close. So it can be considered as a natural choice to search for a relation between $G(.)$ and $J(.)$. By the analytical derivation of the weighted error estimator in equation (2.51), this natural choice can be stated. The proposed relation is that *the error functional $G(.)$ shows the effect of the discretization on the cost functional $J(.)$*. As already stated, the discretization can be interpreted as a perturbation of the optimization problem. For this interpretation, $G(.)$ is the difference of the continuous and discretized cost functional $G(x, x_h) := J(x) - J(x_h)$, where x is the continuous solution (u, λ, q).

If the global minimum fits to the data or measurements u_d (e.g. no perturbations of the measurements) and this global minimum is the solution of the optimization problem, then $u - u_d = 0$. Therefore, $J(x) - J(x_h) = \frac{1}{2}(u - u_d)^2 - \frac{1}{2}(u_h - u_d)^2 = \frac{1}{2}(u_h - u_d)^2$. This is equivalent to the right hand side of the equation $\frac{\partial L}{\partial u} = 0$ in first order necessary condition of the optimization problem, which is the equation for the determination of the dual solution λ. Hence, in some cases, $G(x, x_h) = \frac{\partial J}{\partial x}|_{x_h}$ can be taken. It is an integral on the observation boundary or on the domain Ω.

By the equations of the error estimator and numerical results, it can be observed that the sensitivities of the optimization problem show up by this approach. The dependencies of the observation, of the control and of the primal solutions can be stated.

It should be noted that the term 'error functional' is not the one really known in traditional error estimation theory. Here, the error functional is defined as the difference of two solutions. A term like 'output functional' could be more appropriate for this fact. Nevertheless, the traditional term is taken in order to simplify the understanding of the developed theory.

2.6.2 Interpretation of weighted error estimator for optimization problems

In this section, some criteria for heuristic error estimation for optimization problems with partial differential equation models should be discussed. It will be shown that these are fulfilled for the analytically derived weighted error estimator in section 2.6.

The main idea is that the *important properties of the original and underlying continuous optimization problem must be valid in the discrete optimization problem in a certain accuracy.* Therefore, the following question arises: *Where placing how many discretization elements in the discrete optimization problem?* This is a multidimensional problem depending on the domain Ω.

The following points can be seen as heuristic criteria for a good discrete optimization problem in the indicated context:

1. The *evaluation of cost functional* J. J provides the criterion for the quality of the solution of optimization problem. It is the function which is to be minimized. By section 2.6.1, J leads to a choice of error functional for adaptivity which enables to measure this quality.

2. The *sensitivities with respect to* J in optimization problems. These sensitivities arise for example by the first order necessary conditions of the constraint optimization problem. They are derivations of the Lagrangian function. So the optimization process includes automatically the sensitivities. By standard interpretation (see section 2.1), the Lagrangian multipliers λ show sensitivities in optimization problem. Furthermore, sensitivities can also be motivated by variation of (input) data. Taking the whole calculation as a black box, changes in the input data can cause other output values like for the cost functional J.

3. *Local control of these sensitivities* with respect to J. The effect of perturbation at discrete points on evaluation of J is studied. If this perturbation has a big effect on the evaluation, a high sensitivity is stated. Hence, a *higher evaluation accuracy* is necessary. In other words: An inappropriate discretization will automatically lead to large values of the weights, which in turn will induce local mesh refinement. This has also a contribution to the question: where placing the discretization elements?

These points can be stated for the developed error estimator. The evaluation of J is included by the choice of the error functional $G = J - J_h$.

The sensitivity analysis is contained by the derivation of the weighted error estimator from the first order necessary conditions of the constraint optimization problem. Furthermore, the Lagrangian multipliers λ play an important role in the weighted error estimator.

The local control of the sensitivities is done by the weights arising from the duality. These weights have for optimization two interpretations: The standard interpretation is a control of the local stability (see section 2.9). It is used for treating the error propagation. The new interpretation is the local control of the sensitivities in the optimization problem in combination with the duality theory from optimization. The new property arises from the merge of the weights and the interpretation of the Lagrangian multiplier λ. A motivation

is the greater value of the weights in case of a bad discretization. The appropriate term in the weighted error estimator is then more important.

Another point could be that the discrete stationary point is as close as possible to the stationary point of the underlying continuous problem. By the above reasoning, this is only partially true. In regions, which are not important for the optimization problem, the two stationary points can be far away from each other for some applications.

For the above mentioned interpretation, an appropriate formulation of the optimization problem is necessary. For other formulations, the application of the weighted a posteriori error estimation theory in [12] and [13] may not be as straight forward as presented. From the presented formulation, all dependencies and scalings in the error estimator are somehow natural. In this formulation, also the additional parts for optimization instead of only the forward solution are contained. This means, that the considered mechanisms and dependencies of the optimization problem can be found in the first order necessary conditions of the constrained optimization problem. This includes also the applied regularization methods. Whereas some optimization features like the globalization part can not be found here, see chapter 3.

2.7 Heuristic error indicators

The presented heuristic error estimates are developed for the applications of the Poisson equation (chapter 1, $s(.) = 0$) and of the Ginzburg-Landau equations (chapter 4).

In the following error indicator, the equation for the Lagrangian multiplier is not considered. It can therefore be interpreted as error indicator with 'frozen' λ. The general approach in [10] led to the following a posteriori error estimate

$$|J(u) - J(u_h)| \approx | < J'_u(u), u - u_h > | \ \leq \ \eta_{weight}(u_h, q_h), \tag{2.52}$$

where

$$\eta_{weight}(u_h, q_h) := \sum_{K \in \mathbf{T}_h} \Big\{ \rho_K(u_h, q_h)\ \omega_K(z_h) \ + \rho_{\partial K}(u_h, q_h)\ \omega_{\partial K}(z_h) \Big\} \tag{2.53}$$

with the residual terms

$$\begin{aligned} \rho_K(u_h, q_h) &= \|f + \Delta u_h - s(u_h)\|_K + \frac{1}{2}\ h_K^{-\frac{1}{2}} \|[\partial_n u_h]\|_{\partial K}, \\ \rho_{\partial K}(u_h, q_h) &= \|q_h + \partial_n u_h\|_{\partial K \cap \Gamma_2}, \end{aligned}$$

and the weights

$$\omega_K(z_h) = C_i h_K^2 \|D_h^2 \lambda_h\|_K, \quad \omega_{\partial K}(z_h) = C_i h_{\Gamma_2}^{3/2} \|D_h^2 \lambda_h\|_{\partial K \cap \Gamma_2}.$$

By $[\partial_n u_h]$ the jump of $\partial_n u_h$ across the element boundary is denoted. This error indicator is derived in [10]. It will be named *'opt1'* in the numerical tests presented below.

The following error indicator contains the co-state equations for λ. But not all boundary integrals of the optimization problem are contained. The above mentioned (heuristically

motivated) alternative for an error indicator for optimization problems by augmenting the residual terms in (2.53) can be obtained as follows:

$$\rho_K(u_h, q_h) \;=\; \|\Delta u_h - s(u_h) + f\|_K + \|\Delta\lambda_h + s'(u)\lambda\|_K,$$

while the boundary terms $\rho_{\partial K}$ are kept unchanged. This error indicator contains the residuals of the full equation system (1.7)-(1.9) and will be named *'opt2'* in the numerical tests presented below.

The weighted a posteriori error indicators and estimators will be compared against a more traditional *energy error indicator* which links the mesh adaptation to the local residuals of the computed solution with respect to the equation of state alone (for a survey of this type of error indicators see, e.g., [62]). In this case there is no duality information used. Furthermore, the optimization problem is not considered in an appropriate way. Such an error indicator has the form

$$\eta_{energy}(u_h, q_h) \;=\; \Big(\sum_{K\in\mathbf{T}_h} h_K^2 \rho_K(u_h)^2 \Big)^{\frac{1}{2}}, \quad \rho_K(u_h) := h_K^{-\frac{1}{2}} \|[\partial_n u_h]\|_{\partial K}.$$

The residual terms are omitted because the jump terms will dominate the residual terms ([13]).

There is just a control of the error in the "energy norm" of the state equation alone. Alternatively, the energy error indicator can be formulated by

$$\eta_E(u_h) := c_I \sum_{K\in\mathbf{T}_h} h_K^3 \, \rho_{\partial K}^{(u)2} + c_I \sum_{\Gamma\subset\partial\Omega} h_\Gamma^3 \, \rho_\Gamma^{(u)2}, \tag{2.54}$$

with the cell residuals $\rho_{\partial K}^{(u)}$ and $\rho_\Gamma^{(u)}$. This version also includes the boundary terms of the problem (and not only cell boundaries). Furthermore, incorporating error control for the adjoint equation results in

$$\eta_E(u_h, \lambda_h) := c_I \sum_{K\in\mathbf{T}_h} h_K^3 \{\rho_{\partial K}^{(u)2} + \rho_{\partial K}^{(\lambda)2}\} + c_I \sum_{\Gamma\subset\partial\Omega} h_\Gamma^3 \{\rho_\Gamma^{(u)2} + \rho_\Gamma^{(\lambda)2}\}. \tag{2.55}$$

Both ad-hoc criteria aim at satisfying the state equation and the adjoint equation uniformly with good accuracy. However, this concept seems questionable since it does not take into account the sensitivity of the cost functional with respect to the local perturbations introduced by discretization. Capturing these dependencies is the particular feature of the presented approach.

In some cases it may be interesting to consider a combination of the weighted error estimator and the energy error estimator. An application can be if one needs a certain exactness of the state equation. This combination can be done as follows: $\eta_{\omega,E}(u_h, \lambda_h, q_h) := \eta_\omega(u_h, \lambda_h, q_h) + \beta\eta_E(u_h)$, with a suitable weighting factor $\beta \geq 0$.

2.8 Algorithmic realization

The main issue in the solution approach is the appropriate design of the computational mesh by using adaptivity. To get close to the continuous solution of the optimization problem, calculations on fine grids have to be done. There are various ways to design the grid

for the calculations. One possibility is to take *equidistant grids* leading to very expensive calculations. To reduce the costs of the calculation, one can use *adaptively constructed* meshes, i.e., the meshes are refined only where it is necessary for achieving sufficient accuracy. In this case, the computations are done on a series of locally refined meshes. The criterion for mesh refinement has to be chosen in accordance with the particular needs of the problem considered. For the presented approach, the mesh refinement is based on a posteriori error estimates for the discrete solution derived by duality arguments.

In designing the solution method for the optimization problem (see section 2.2), the method tries to stay as long as possible within the context of the continuous formulation. Accordingly, the Newton iteration for solving the boundary value problem (1.7)-(1.9) is applied on the continuous level while discretization takes place independently for each linear sub-step. This approach fits better with the presented concept of mesh adaptivity which is based on a computational sensitivity analysis for the continuous problem.

There are several possibilities for combining adaptive discretization with the optimization process. The approach tries to stay close to the continuous problem in order to exploit the inherent partial differential equation structure. Alternatively, one could discretize at first and then use optimization strategies for the discrete problem. In this case, there would be less difficulties in determining appropriate search directions for the discrete Newton method. In the continuous approach it may happen that the search directions obtained after discretization are not very good for the underlying continuous Newton iteration. However, this potential difficulty seems to be less critical in the present situation provided that good globalization strategies are used if necessary, particularly on coarser meshes.

Since in the presented approach Newton iteration and discretization is nested it is not so clear when to apply mesh adaptivity at best. The theory is oriented at the discretization error for the boundary value problem (1.7)-(1.9). Two possibilities for mesh adaptivity have been tried: 1) The adaptive mesh refinement is done after at most a maximal number of Newton iterations on each discretization level. The Newton system is therefore not necessarily in the limit of the Newton iteration on the discretization levels when refinement is done. In order to save costs in the model computations, optimization and the mesh adaptation process is mixed, yet rigorous justification is lacking. Furthermore, an acceleration of the convergence can be stated for some examples. The results presented below indicate that this *'diagonal' iteration* is sufficiently robust and efficient. For the Ginzburg-Landau equations presented in chapter 4 this version is implemented in the code 'rhoptcon'.

2) The refinement is always done in the limit of the Newton iteration on the discretization levels. In this case, the theory can be applied in a more rigorous way. The error estimates are then really estimates and not only heuristic error indicators. For the Ginzburg-Landau equations this version is implemented in the code 'bkr'.

The adaptive mesh refinement itself is organized as follows: From the global error estimator, local 'error indicators' are extracted by which the mesh adaptation is driven:

$$\eta := \sum_{T \in \mathbf{T}_h} \eta_T, \tag{2.56}$$

with certain cell indicators, e.g. $\eta_T := h_T^3 \rho(u_h)_{\partial T} \rho(\lambda_h)_{\partial T}$. We aim at achieving a prescribed tolerance TOL for the quantity $J(u)$ and the number of mesh cells N which measures the complexity of the computational model. Usually the admissible complexity is

constrained by some maximum value $N_{\max}$. Here, a version of the so called *fixed fraction strategy* as described in [13] is adopted. In each adaptation cycle the mesh cells are ordered in accordance to the size of the value of their local error indicators. Then, those elements with the largest values in the ordered list are refined until a certain percentage of the total error bound (say 30%) is reached. This leads to a gradual refinement of the (too) coarse starting grid.

For good quantitative error estimation, the value of the (weighted) error estimator is used as stopping criterion for the adaptive mesh refinement process. The stopping value depends strongly on the application. Main ideas are accuracy of measurements, formulation of the cost functional, exactness of the solution of the code and last but not least the model itself.

On each mesh, the Euler-Lagrange equations are discretized by the Galerkin finite element method as described above using piecewise bilinear shape functions for both the state and adjoint variables u and λ, while the traces on Γ_C of the bilinear shape functions form the control space Q_h. Then, the resulting discrete systems are solved iteratively and new meshes are generated on the basis of a posteriori error estimators. In all cases, the weights are evaluated by using difference approximation as described in the previous section with interpolation constants set to appropriate values like $C_I = 0.1$.

2.9 Comparison of different error indicators

In this section, a comparison of several applied error indicators should be given.

An important feature of the presented approach in error estimation is that the error functional has to be taken such that the mesh refinement is organized in accordance to the particular needs of the optimization process. In contrast to the standard energy-error estimator commonly used in static elliptic problems, error control in optimization problems has to follow different strategies. The most natural choice appears, as already described in section 2.6.1, to relate the error functional for driving the mesh refinement with the cost functional of the optimization problem.

The numerical tests presented below confirm that the philosophy underlying the presented approach to adaptivity in optimization is valid: The discretization of the problem should be adapted in accordance to the sensitivity of the optimization problem and not merely to the accuracy requirements of the partial differential equation (equation of state). Consequently it may happen that some of the constraints do not need to be fulfilled with high accuracy in some parts of the domain while still allowing a good approximation of the optimization process.

The energy error estimator from section 2.7 just considers the state equation. It will be therefore not appropriate for optimization problems in which the state equation is not the sole important determining criterion.

If the diagonal version described in section 2.8 is applied, an additional error is estimated: The error arising from the Newton method. This results from the fact that we are not in the limit of the Newton iteration.

Traditional *a posteriori* error estimates like the one derived in [13]

$$\|\nabla^{1-r} e\| \;\leq\; C_s C_i \Big(\sum_{T \in \mathbf{T}_h} h_T^{2r} \rho_T^2 \Big) \tag{2.57}$$

have a bound in the energy norm for $r = 0$ and in the L_2-norm for $r = 1$. It depends on the interpolation constant C_i and the stability constant C_s. C_i is usually of size $0.1 \leq C_i \leq 1$. C_s measures the stability properties of the dual problem $z \in X : M^t(z, y) = G(y)\ \forall y \in X$ in terms of the global a priori estimate $\|\nabla^{1+r} z\| \leq C_s \|\nabla^{1-r} e\|$. The *a posteriori* error estimate (2.57) contains information about the mechanism of error propagation only through the global stability constant C_s. To overcome this deficiency, the local weights ω_T have been introduced as factors to the local residuals ρ_T. They have been proposed by R. Becker and R. Rannacher for *a posteriori* error estimators for the forward solution in [12]. These weights contain all information about the local approximation properties of the spaces X_h, as well as the local stability properties of the underlying continuous problem. The stability constant C_s is replaced by this weights. So the mechanism of error propagation is now also captured by local information. Hence the approach gets independent of C_s which is difficult to compute for advanced applications, especially in a sharp sense. In general, C_s has to be determined by numerical computation.

The weighted residual based a posteriori error estimates developed by R. Becker and R. Rannacher (see for example in [12] and [13]) take the following special form:

$$|G(e)| \quad \leq \quad C_i \sum_{K \in T_h} \rho_K \; \omega_K(z_h).$$

The residual terms ρ_K are 'weighted' with the local weights $\omega_K(z_h)$ mentioned above. They replace the global stability constant C_s. For local mesh refinement, the local information from the weights seems more appropriate then the global stability constant C_s. The error estimators are used to construct good grids. They give criteria for a good discretization. The estimation of the (special) error functional $G(e)$ is a consequence of this.

In the literature exist one error estimation approach for optimization problems known to the author. The approach in [34] is of theoretical importance. It provides a priori error estimates. Therefore, only estimation of the variables of the optimization problem is given. No error functional is provided. So not the whole optimization problem is considered. Furthermore, no local stability control and no local sensitivity control is provided. Only an abstract error constant independent of the mesh size h is given. There are no numerical computations done (which are not possible for the estimates containing (unknown) continuous information).

It should be mentioned that it is theoretically possible with the presented approach to compute for each continuous variable an own grid representing the special properties just of this variable. This leads to split error estimates for the variables. The present version of the approach calculates one grid which represents all important information of the whole optimization problem on one grid which is the bases for the grids of all variables. The grids for the boundary variables like the control are just a part of this base grid.

The error due to the linearization of the nonlinear optimization problem can cause additional strategies for mesh refinement. One solution is the development of hybrid error estimates first reducing the linearization error and then the discretization error.

2.10 Comparison to model reduction approaches

The presented approach can be interpreted in the well-known scheme of model reduction as described in [44].

Nonlinear continuous optimization problems like the one stated above generally cannot be solved analytically. They have to be approximated by discretization. For the presented approach, the model reduction is discretization of the problem. This means the reduction of an originally infinite dimensional problem to a finite dimensional problem which can be solved on the computer. The question is, how can this reduced model be constructed as appropriate as possible? The criteria are cheap solution and good accuracy.

The model reduction has to follow criteria which respect the original optimization problem particularly the sensitivities of the cost functional J. Accordingly, the strategy for arranging the computational mesh should take into account these sensitivities. Further the important properties of the underlying continuous model must be preserved with a certain accuracy. This leads to the heuristic motivation of section 2.6.2.

Considering the structure of the above optimization problem, criteria like the following can be derived: The value of the cost functional J shows the quality of an approximate solution of the optimization problem. Hence, the quality of the evaluation of the cost functional J is one possible heuristic criterion. Other criteria result from the sensitivities inherent to the optimization problem which are represented by the Lagrangian multiplier λ. Further, the effect of variations of the control function on the state variable should be included. Therefore, local control of all these sensitivities is necessary. A big effect means a high sensitivity. So a higher evaluation accuracy is necessary. This leads again to the question of how to turn these qualitative arguments into quantitative criteria for mesh arrangement.

The presented analytic approach in sections 2.6 and 2.5 leads to a model reduction for the system. These criteria should not only provide information about "where the mesh cells have to be placed", but also quantitative information about "how many mesh cells have to be placed in certain areas". The too coarse model is refined (or enriched) gradually by the error estimator until the discrete problem is close enough to the original continuous model.

2.11 Quantitative error estimation

This section should provide a measure to compare the error of the discrete system (in comparison with the underlying continuous problem) with the value of the error indicator. This is called *quantitative error estimation.*

The *effectivity index* $I_{\text{eff}} := \frac{\text{error}}{\eta(u_h, q_h, \lambda_h)}$ provides the measure mentioned above. In the literature, also the inverse can be found as effectivity index. The presented version seems better for the presented problems because $\eta(u_h, q_h, \lambda_h) >$ error. Asymptotic sharpness is stated for $\lim_{TOL \to 0} I_{\text{eff}} = 1$. If the effectivity index I_{eff} is close to 1, the value of the error indicator is close to the error of the discrete system. If $I_{\text{eff}} < 1$, the error is overestimated ($\eta(u_h, q_h, \lambda_h) >$ error). This shape of the quotient I_{eff} is used for the error can more easily be equal to 0 than the value of the applied error indicators (see for example upper bound in error inequality and parameter estimation problems).

If there is a good connection between error of the discrete system and value of the error indicator, the latter can be used as a stopping criterion the the grid adaptivity process. The special choice of the stopping value depends on the application. It has very often a connection with (the evaluation of) the cost functional.

2.12 Example: A "forward" test case

In this section, the difference between traditional "energy error control" and our functional-oriented "dual-weighted error control" by considering the following linear primal test example should be illustrated:

$$-\Delta u + u = 0 \quad \text{on } \Omega, \tag{2.58}$$
$$\partial_n u = q \ \text{ on } \Gamma_C, \quad \partial_n u = 0 \ \text{ on } \partial\Omega \setminus \Gamma_C.$$

The domain Ω for the test in this section is the following Configuration 2. The two configurations differ in the choice of the observation boundary. The two configurations will be used in the next section for the numerical tests of the error indicators for an optimization problem. The numerical results are obtained with the code 'bkr' (which refines in the limit of the Newton iteration of the discretization levels).

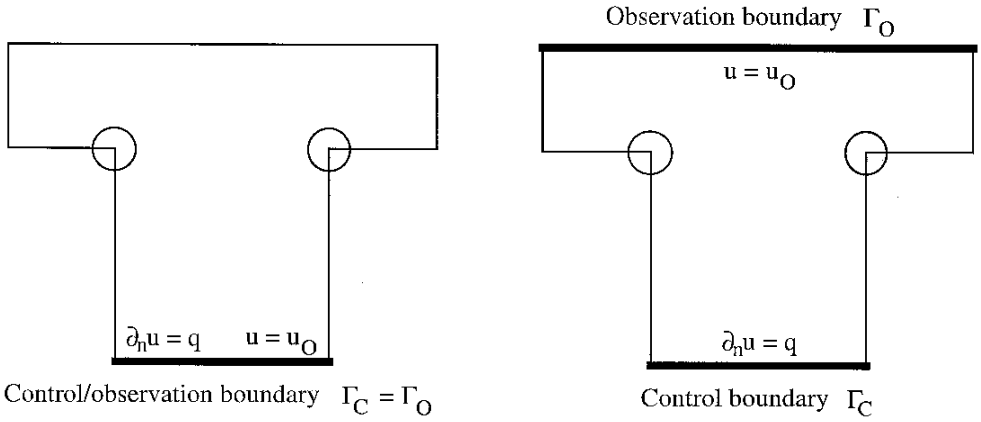


Figure: Configuration of the boundary control model problem on a T-domain (Ginzburg-Landau model): Configuration 1 (left), Configuration 2 (right).

The boundary control is frozen as $q \equiv 0.0503455$ (taken from an optimization result). The corresponding discrete equations read

$$(\nabla u_h, \nabla \psi_h)_\Omega + (u_h, \psi_h)_\Omega = (q, \psi_h)_{\Gamma_C} \quad \forall \psi_h \in V_h. \tag{2.59}$$

The error $e = u - u_h$ with respect to the *quadratic* observation functional

$$J(u) = \tfrac{1}{2}\|u - u_O\|^2_{\Gamma_O}$$

should be controlled.

The corresponding dual solution $z \in V$ is obtained by solving the corresponding system (4.3) - (4.5) with frozen boundary function q and linearized right-hand side $J'(u_h;\psi)$. The resulting a posteriori error bound is

$$|J(u) - J(u_h)| \le \eta_\omega(u_h) := \sum_{T\in\mathbf{T}_h} h_T^2\{\rho_T^{(u)}\omega_T^{(z)} + \rho_{\partial T}^{(u)}\omega_{\partial T}^{(z)}\} + \sum_{\Gamma\in\partial\Omega} h_\Gamma^2\rho_\Gamma^{(u)}\omega_\Gamma^{(z)}, \qquad (2.60)$$

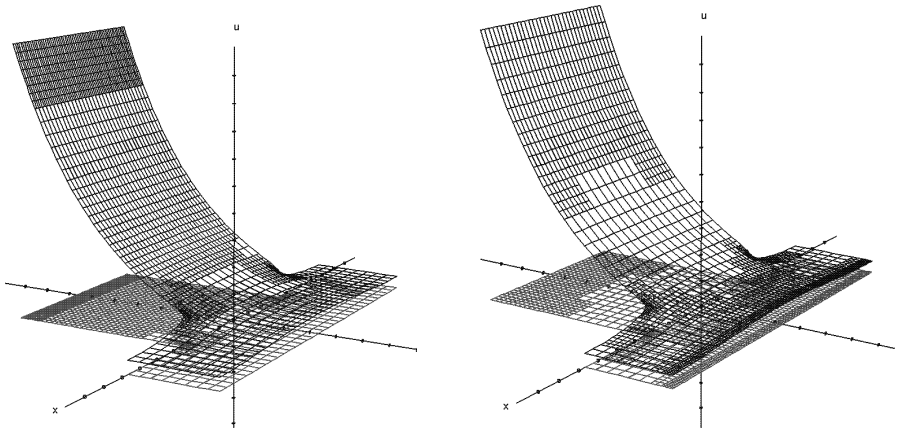
with cell residuals and weights defined as above. The asymptotic correctness of this error estimator is demonstrated in the following table. It shows effectivity index I_{eff} of the dual–weighted error estimator $\eta_\omega(u_h)$ applied to the linear primal model problem, i.e. $E(u_h) := |J(u) - J(u_h)|$.

N	1376	5840	22544	57104	84368
$E(u_h)$	1.64e-05	4.17e-06	1.01e-06	3.5e-07	2.49e-07
I_{eff}	0.81	0.91	0.92	0.95	0.88

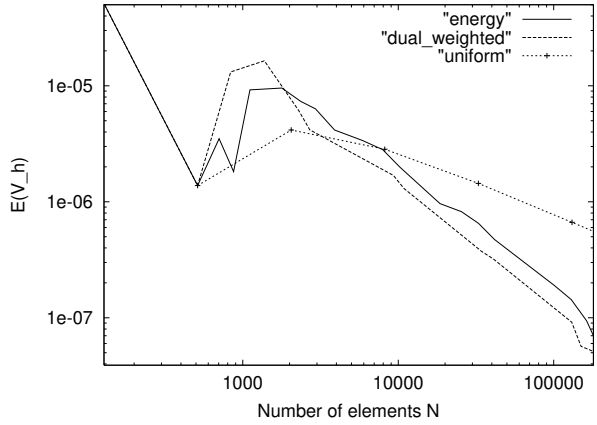
The dual–weighted error estimator should be compared with the traditional energy-norm error estimator which in this case reads as follows:

$$\|\nabla e\|_\Omega^2 \le \eta_E(u_h) := c_I \sum_{T\in\mathbf{T}_h} h_T^4\{\rho_T^{(u)2} + \rho_{\partial T}^{(u)2}\} + c_I \sum_{\Gamma\in\partial\Omega} h_\Gamma^4\rho_\Gamma^{(u)2}, \qquad (2.61)$$

with the notation as introduced above. Clearly, small $\|\nabla e\|_\Omega$ implies small $E(u_h)$, but not vice versa. Hence, mesh adaptation based on the energy-error estimator may result in overly refined meshes. This is clearly seen in the following figures. They contain results on meshes obtained by the error estimators $\eta_E(u_h)$ (left) and $\eta_\omega(u_h)$ (right) with $N \sim 5000$ cells in both cases. The graph of the solution is strongly scaled up.



The efficiency of the computed meshes generated by the error estimators $\eta_E(u_h)$ (solid line) and $\eta_\omega(u_h)$ (dashed line), and by uniform refinement (crosses) is shown in the following figure. The values are in *log/log* scale.



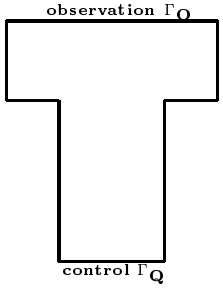
The reference value $J(e)_{\text{ref}}$=1.990239068196715 is calculated on a very fine grid. It can be stated that this value seems to be good also if it is compared to the values given by the *equidistant refinement* in the following table. By the figure comparing the efficiency of the computed meshes, the meshes by the equidistant refinement are the worse for finer grids. From the computed data, also for these meshes a gradual convergence to the reference value is obvious.

N	8192	32768	131072	524288
$J(e)$	1.99023622	1.99023762	1.99023840	1.99023876
$E(u_h)$	2.8e-06	1.4e-06	6.7e-07	3.0e-07

Obviously, the energy-error estimator puts too much emphasis on refining at the reentrant corners which is obviously less important for achieving good accuracy along the observation boundary Γ_O. In contrast to that, the dual–weighted error estimator provides a better balance between resolving the corner singularities and the neighborhood of Γ_O. This results in a higher mesh economy as shown by the corresponding error plots in the above figure. This demonstrates the value of capturing the sensitivities inherent to the problem under consideration. This effect will become even more pronounced in solving the optimal control problem.

2.13 Example: A linear test case

A first example of the theory developed above should be provided by the following optimization problem with linear state equation. It has already been considered in [10]. In this section some additional numerical data will also be provided. In this example the control acts along the lower boundary Γ_C, whereas the observation is taken along the upper boundary Γ_O.



$$\begin{aligned} -\Delta u \;+\; u &= 0 \qquad \text{in } \Omega, \\ \partial_n u &= 0 \qquad \text{on } \partial\Omega \setminus \Gamma_Q, \\ \partial_n u &= q \qquad \text{on } \Gamma_Q. \end{aligned} \tag{2.62}$$

The cost functional is chosen as

$$J(u,q) := \tfrac{1}{2}\|u - c_0\|^2_{\Gamma_O} + \tfrac{\alpha}{2}\|q\|^2_{\Gamma_C},$$

with $c_0 \equiv 1$ and $\alpha = 1$. In this case, the regularization term $\frac{\alpha}{2}\|q\|^2_{\Gamma_C}$ may be viewed as part of the cost functional with its own physical meaning. Computations on a series of locally refined meshes are performed. On each mesh, the system of the first-order necessary condition is discretized by the Galerkin finite element method described above. The resulting discrete saddle-point problems are solved iteratively by a GMRES method with multi-grid pre-conditioning. The adaptive mesh refinement is based on an a posteriori error estimator already described in the previous sections. The weights in the error estimator (2.37) are evaluated with an interpolation constant set to $C_I = 0.1$. The mesh refinement uses the "Fixed-Fraction Strategy" described above.

Table 2.1 shows the quality of the error estimator (2.37) for quantitative error control. The *effectivity index* is defined by $I_{eff} := E_h/\eta_h$, where $E_h := |J(u,q) - J(u_h,q_h)|$ is the error in the cost functional and $\eta_h := \eta(u_h,q_h)$ the value of the error estimator used. The reference value is obtained on a mesh with more than $200,000$ cells. We compare the weighted error estimator with a simple ad hoc approach based on the already presented standard energy-error estimator for the state equation. Figure 2.1 shows the computed "optimal" states over the meshes generated by the two different error estimators.
The two meshes are quite different: The energy-error estimator over-emphasizes the steep gradients near the control boundary and it leaves the mesh too coarse along the observation boundary. The more selective *weighted* error estimator concentrates the mesh cells where they are needed for the optimization process. The quantitative effects on the mesh efficiency of these two different refinement criteria is shown in Figure 2.2 (E_h versus N in log/log-scale).

Finally, how the approximation $\{u_h, \lambda_h, q_h\}$ obtained by the weighted error estimator (2.37) actually satisfies the state equation is checked; for this the *global* energy-error estimator is taken as quality measure. Table 2.2 shows a comparison of the two sequences of meshes generated by the weighted error estimator $\eta_\omega = \eta_\omega(u_h,\lambda_h,q_h)$ ("ω-meshes") and the energy-error estimator $\eta_E = \eta_E(u_h)$ ("E-meshes"). The first and second columns contain the values of η_ω and η_E on ω-meshes, while the third and fourth columns contain the values of η_ω and η_E on E-meshes.
The energy-norm error bound η_E for the state equation on the ω-meshes is slightly larger than on the E-meshes. This is not surprising since the ω-meshes are not so much refined in the regions where the state variable has a steep gradient. The cells are rather concentrated along the control and observation boundaries which seems to be more effective for the optimization process. Indeed, the approximate solution $\{u_h, \lambda_h, q_h\}$ obtained by the weighted

error estimator η_ω achieves a much smaller value (factor ~ 0.1) of the cost functional. However, for other data, e.g., $c_0 = cos(2x)$ and $\alpha = 0.0001$, the discrepancy between the two kinds of meshes with respect to the satisfaction of the state equation may be more significant.

Regularization can influence the solution of the optimization system. If the regularization factor α is chosen too big, the solution of the optimization problem can be (very) different from the solution of the original optimization problem. In other words: The solution of the optimization problem can be dominated by the regularization for big α. For the optimization problem of this section an example should be given with the following data: The domain is the same as in the previous calculations. The starting values for u is 100, for λ is 0.1 and for q is 0. The observation is $\cos(3x)$ and the regularization profile is $q_0 = 0$. Figure 2.3 shows the primal solution and the Lagrangian multiplier obtained with regularization factor $\alpha = 0.01$. Whereas Figure 2.4 shows the same with $\alpha = 0.00001$. These results show that the solution with $\alpha = 0.00001$ seems to be close to the solution of the original optimization problem. The solution with $\alpha = 0.01$ clearly shows the (too) strong influence of the regularization especially on the control function. The special structure of the boundary regularization may cause the effect that in some cases this influence of the regularization on the grid refinement is almost without transition (see section 5.7).

Table 2.1: Linear test (Configuration 1): Efficiency of the weighted error estimator.

N	320	1376	4616	11816	23624	48716
E_h	$1.0e-3$	$3.5e-4$	$3.2e-5$	$1.6e-5$	$6.4e-6$	$2.8e-6$
I_{eff}	1.1	0.7	0.7	1.0	0.8	0.7

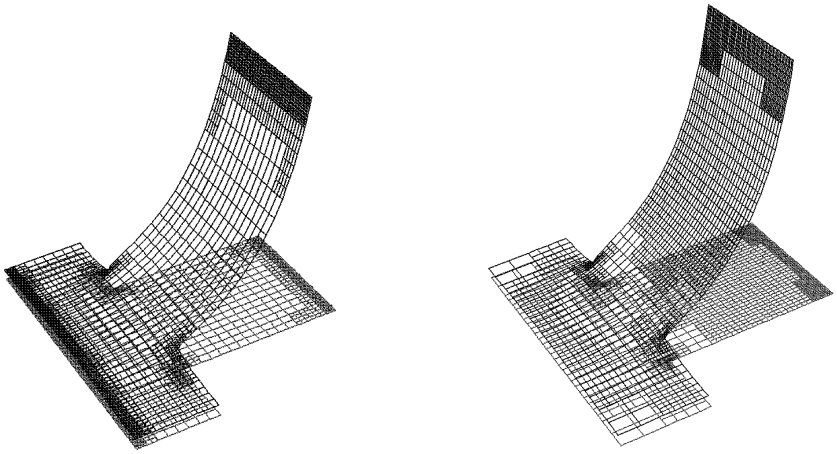


Figure 2.1: Linear test: Comparison of discrete solutions obtained by the weighted error estimator (left, $N \sim 1600$ cells) and the energy-error estimator (right, $N \sim 1700$ cells).

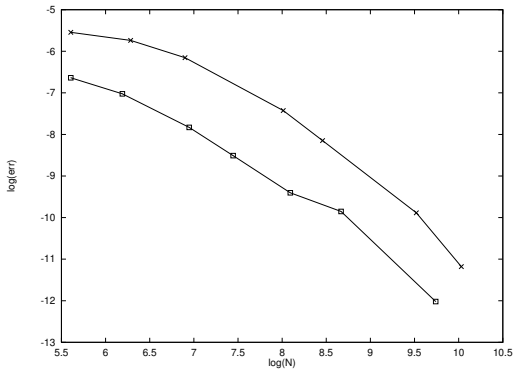


Figure 2.2: Linear test (Configuration 1): Comparison of the efficiency of the meshes generated by the the weighted error estimator (symbol $\square$) and the energy -error estimator (symbol $\times$) in log / log scale.

Table 2.2: Linear test (Configuration 1): Values of the two error estimators η_ω and η_E obtained on "ω-meshes" and on "E-meshes".

$N \approx$	η_ω on ω-meshes	η_E on ω-meshes	η_ω on E-meshes	η_E on E-meshes
140	0.0040205	0.0193270	0.0043245	0.0162589
300	0.0022030	0.0157156	0.0026536	0.0112183
750	0.0008330	0.0092718	0.0020437	0.0074801
3700	0.0001660	0.0049598	0.0004870	0.0034197
11000	0.0000532	0.0026208	0.0002199	0.0019036
21000	0.0000317	0.0020740	0.0001189	0.0014285
28000	0.0000239	0.0016294	0.0001088	0.0012403
48000	0.0000108	0.0013373	0.0000722	0.0009399
145000	0.0000037	0.0006950	0.0000328	0.0005466

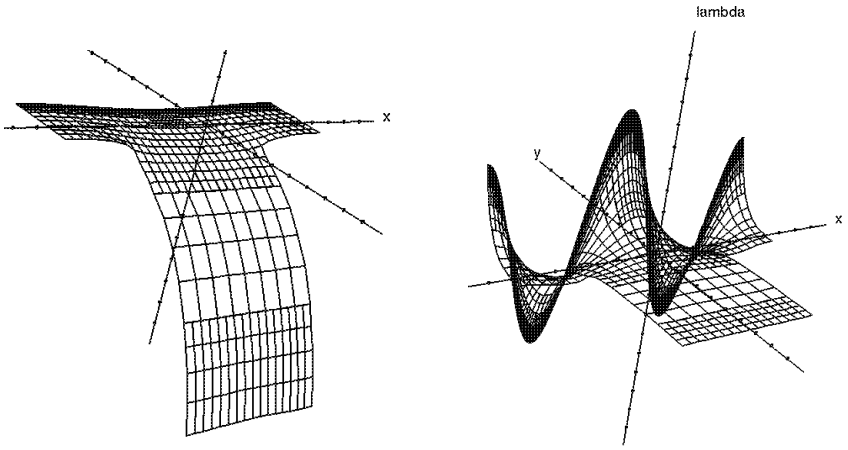


Figure 2.3: Linear test: Discrete primal solutions (left) and Lagrangian multiplier (right) obtained by the weighted error estimator with $\alpha = 0.01$.

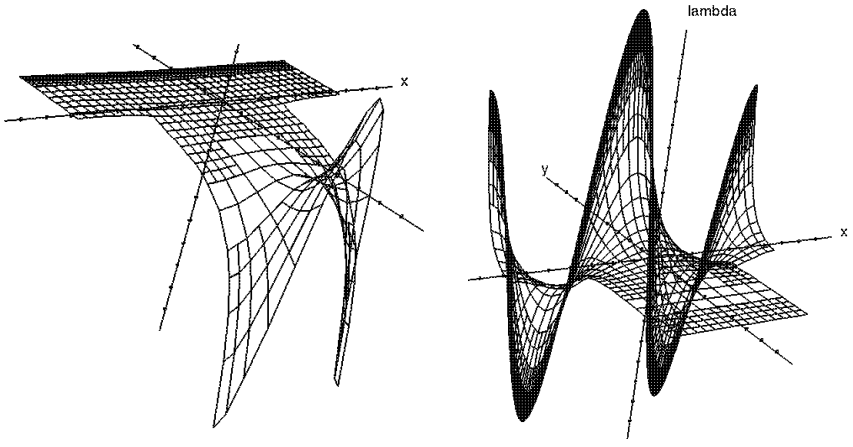


Figure 2.4: Linear test: Discrete primal solutions (left) and Lagrangian multiplier (right) obtained by the weighted error estimator with $\alpha = 0.00001$.

Chapter 3

Globalization techniques

In our context, one main problem involved with the application of a Newton method is that its convergence depends on the starting values. The robustness of the developed Newton method can not be assured. Much effort is spent in optimization theory to develop appropriate methods in order to improve the range of convergence. These techniques are known as globalization methods. In the presented context, the globalization must work with adaptivity and error estimation, especially the original underlying continuous optimization problem should still play a decisive role. Furthermore it should not be too costly. Function and differentiation evaluations are very expensive for partial differential equations, especially for large and coupled systems as in the presented case.

The principal goal of this thesis is to develop optimization and adaptivity techniques for partial differential equations. This is the first step. Globalization will be the second step. Some standard methods have been tested for the presented applications. Certain new developments are sketched. Promising results will be given in sections 4.5 and 4.7 and chapter 5. Further research would be necessary to develop as tuned strategies as for optimization governed by ODE and DAE systems. However, this is beyond the topic of this thesis.

Two main streams can be sketched as globalization techniques: reducing the step size and changing the search direction of the Newton method. Various techniques and mixtures have been developed in the last decades. The application of these methods on optimization governed by partial differential equations is not yet satisfactory solved, especially, if error estimation is included. For the search direction, a principal problem is the difference between the direction from the discrete optimization problems and the direction from the underlying continuous optimization problem. The computations are for the discrete problems but the continuous optimization problem originally has to be solved.

A modified Newton method will be developed in section 3.3. The second order conditions of a constraint optimization problem are exploited for a correction of the search direction of the Newton method. A check of these second order conditions and the determination of the stationary point as minimum, maximum or saddle point results additionally by this technique as shown in section 3.3.

It is well-known that Newton methods with full step length lead to quadratic convergence rates in the neighborhood of the solutions in case of convergence. Normally, the globalized methods do not reach such an convergence rate. For example, gradient methods

just have linear convergence. Quasi-Newton methods like the presented technique normally reach normally super-linear convergence.

3.1 Damped Newton methods

Damped Newton methods reduce the step length of the Newton increment by a fixed factor until the value of the residual is reduced with respect to the former iteration. This cheap method is not elaborated but it may help in many cases. It is applied in the developed codes as additional means if the full step length leads to divergence.

3.2 Line search methods

Line search is one standard method for a good reduction of the step length of the search direction of the Newton method. An often applied version is the Armijo-Goldstein principle ([31, p. 100]). By this method, normally a larger range of convergence can be achieved. For ODE and DAE systems, line search methods are applied successfully. For the presented context of adaptivity in partial differential equations with finite element discretization, line search methods may face other types of problems. In section 4.5, good results with a slight modification of the Armijo-Goldstein line search will be presented. In some cases, the values of the Newton residuals are bad and the resulting primal and dual solutions are not satisfactory. An explanation may be the fact that step length reduction has a different effect on each of the cells. A good reduction for one cell can be a bad reduction for an other one. The numerical results indicate that these methods in the present formulation are not appropriate for the presented nonlinear applications and solution methods (see chapter 6).

Additionally, a cyclic behavior (Maratos effect) and divergence was stated for some test cases (examples from the optimization problems presented in chapter 4). For this reason, also a *watch-dog* line search method was tested ([19], [38] and [39]). It prevents standard problems like a cyclic behavior of the iterations by its special algorithmic structure and relaxed criteria for the trial step-length acceptance. Backtrack capabilities allow the program to abandon a nonproductive correction step and recover a base step. Furthermore, it uses second-order correction methods. Some algorithms even apply bypass conditions: Use second-order correction methods when they will improve performance; otherwise do not use them. One main problem for optimization governed by partial differential equations is that the storage of the base steps needs much memory - especially on fine grids. Numerical results showed that a simplified version of this technique does not lead to an appropriate globalization method for optimization governed by partial differential equations. The reasons are again that the step length reduction has a different effect on each cell.

One possibility to use these methods is to apply them in the beginning until the stationary point is 'close enough'. Then, a pure Newton method is used. This method was already tested successfully.

3.3 Trust region like modified Newton method

A very simple and efficient method was developed together with G.H. Bock and J. Schlöder to adapt the search direction of the Newton method. Additional information is attained by exploiting the second order conditions of a constraint optimization problem:
The Hessian matrix $H = \nabla^2 L(x^*, q, \lambda)$ must be positive semidefinite in a minimum. For equality constraints $F(x^*) = 0$ in the optimal solution x^* this means:

$$P^T \ \nabla^2 L(x^*, q, \lambda) \ P \quad \geq \ 0 \quad \forall P \in T(x^*),$$

with

$$T(z) := \{P \mid \nabla F(z)^T P = 0\}.$$

As above L denotes the Lagrangian function. If the second order sufficient conditions of a constraint optimization problem

$$P^T \ \nabla^2 L(x^*, q, \lambda) \ P \quad > \ 0 \quad \forall P \in T(x^*)$$

are fulfilled, then x^* is a strict local minimum.

The above condition ensures that the stationary point is a minimum. If this condition is not fulfilled, the method may converge to a maximum, a saddle point or may even lead to divergence.

Proposition 3.3.1. *If the Hessian matrix is positive definite, the presented Newton method converges to a minimum.*

Proof. Let ϕ be the following merit function:

$$\phi(t) \ := \ L(x_k + t \triangle x_k).$$

Then derivation w.r.t. parameter t leads to:

$$\phi'(t)|_{t=0} \ = \ \nabla L(x_k)^T \triangle x_k \ = \ -\nabla L(x_k)^T \ \nabla^2 L(x_k)^{-1} \ \nabla L(x_k) \quad < 0$$

if the symmetric Hessian matrix is positive definite (which means that also its inverse matrix $\nabla^2 L(x_k)^{-1}$ is positive definite because the eigenvalues of the inverse matrix are the inverse eigenvalues of the original matrix). The latter equality results from the Newton step. This means that a Newton method leads to a descent direction for a positive definite Hessian matrix. □

This idea leads to the application of the Levenberg-Marquardt approach ([31]). The Hessian matrix is updated with a non-negative multiple of the identity matrix:

$$(\nabla^2 L(x_k) + \beta_k I) x_k = \nabla L(x_k).$$

A traditional choice for the Levenberg-Marquardt parameter β_k is the absolute value of the smallest eigenvalue of the Hessian matrix (if this smallest eigenvalue is negative). Then the updated Hessian matrix is positive definite. The computation of this eigenvalue can be very expensive especially for large systems often arising in adaptive finite element methods.

Therefore, the Levenberg-Marquardt parameter β_k is computed by the following approach: If the Hessian matrix is positive definite, then

$$\triangle x_k \ \nabla^2 L(x_k)^{-1} \ \triangle x_k > 0.$$

Otherwise, this value is ≤ 0. In the latter case, the absolute value of it is taken as Levenberg-Marquardt parameter:

$$\beta_k \ := \ |\triangle x_k \ \nabla^2 L(x_k)^{-1} \ \triangle x_k| + \gamma_k \qquad \text{if} \qquad \triangle x_k \ \nabla^2 L(x_k)^{-1} \ \triangle x_k \leq 0.$$

Small γ_k effectuate a better numerical behavior as calculation without γ_k. For optimization governed by Ginzburg-Landau models in superconductivity and by Navier-Stokes equations, $\gamma_k = 0.001$ was chosen.

Again the updated Hessian matrix is positive definite but with much less costs. If the original Hessian matrix is already positive definite, the pure Newton method is applied. So a relatively cheap globalization method based on second order information is derived. This modified Newton method is a Quasi-Newton method.

Several additional advantages are obtained by this technique: It can be determined by the update factor if the stationary point is a minimum, a maximum or a saddle point.

For this modified Newton method, promising results will be presented in section 4.7 and in chapter 5. In general, Quasi-Newton methods lead to a deceleration of the convergence compared with pure Newton methods in a neighborhood of the solution. This can be derived analytically and was also confirmed by numerical examples (see [31, 4.5.2.3.]).

There is a connection between trust region, modified Newton, Levenberg-Marquardt and regularization methods. In [26], section 11.2, this connection is sketched: 'The key idea of any Newton type method consists in repeatedly linearizing the operator equation $F(x) = y$ around some approximate solution x_k^δ, and then solving the linearized problem ... for x_{k+1}^δ '

$$F'(x_k^\delta)(x_{k+1}^\delta - x_k^\delta) = y^\delta - F(x_k^\delta). \tag{3.1}$$

The Levenberg-Marquardt method

$$x_{k+1}^\delta = x_k^\delta + (F'(x_k^\delta)^* F'(x_k^\delta) + \beta_k I)^{-1} F'(x_k^\delta)^* (y^\delta - F(x_k^\delta)) \tag{3.2}$$

can be interpreted as a (nonlinear) Tikhonov regularization (by linearizing F):

$$\|y^\delta - F(x_k^\delta) - F'(x_k^\delta)(x_{k+1}^\delta - x_k^\delta)\|^2 + \beta_k \|x_{k+1}^\delta - x_k^\delta\|^2 \tag{3.3}$$

when minimizing this quadratic functional for x_{k+1}^δ. The appropriate choice of the regularization parameters β_k in (3.2) is a crucial question. The original idea behind the Levenberg-Marquardt approach is to minimize $\|y^\delta - F(x_k^\delta)\|$ within a trust region $\|x - x_k^\delta\| \leq h_k$. This gives a relation to trust region methods.

The presented modified Newton method can by the same kind of argument also be interpreted as a regularization method. An alternative and very less expensive way for choosing the regularization parameters β_k is given. It is just an other regularized approximation of the solution of the above optimization problem (3.1).

Chapter 4

Optimization for nonlinear Ginzburg-Landau models

In this chapter, the above developed theory will be applied to the model of nonlinear Ginzburg-Landau equations describing superconductivity in semiconductors. The control will always be a Neumann boundary control (NBC). The observations are both distributed or boundary observations. The systems are derived by analytical differentiation as described in section 1.9. The obtained numerical results, especially those with the dual-weighted error estimator, are the first challenging application for the developed theory. The important results can mainly be found in the last sections. Many results of this chapter have already been published in [9], [10], [11] and [44].

4.1 Superconductivity

The purpose of this section is a motivation for the presented equations for superconductivity. Only some basic facts will be given. Further details should be searched in literature for this is beyond the purpose of this thesis.

Superconductivity was discovered 1911 by H. Kamerlingh Onnes in Leiden. It is defined as *'electrical resistance of various metals disappears completely in a small temperature range at a critical temperature T_c'*([60]). It is a characteristic of metals like mercury, lead, tin. There are two principal points connected with superconductivity: The first aim is perfect conductivity, which means that the magnetic field is excluded from a superconductor. The second aim is perfect diamagnetism discovered by Meissner and Ochsenfeld. The latter means that the magnetic field is expelled from an originally normal sample. Applications are high-current transmission lines and high field magnets.

There are three main models: The London equations, the BCS theory and the Ginzburg-Landau theory. A description of these theories can be found in [60].

'A superconductor with a perfect Meissner effect ... is the ideal superconductor with a constant density of superconducting charge-carriers and an excluded magnetic field' ([24]). In the case of a perfect Meissner effect, the Ginzburg-Landau equations reduce to the considered system version (see [24]): They result in a model of partial differential equations defining a complex pseudo-wave function u. Neglecting internal magnetic fields, the

simplified Ginzburg-Landau model takes the form (for details see [60]):

$$\begin{aligned} -\Delta u \; + \; s(u) \quad &= \quad f \qquad \text{in } \Omega, \\ \partial_n u \quad &= \quad 0 \qquad \text{on } \partial\Omega \backslash \Gamma_C, \\ \partial_n u \quad &= \quad q \qquad \text{on } \Gamma_C. \end{aligned} \tag{4.1}$$

The nonlinearity $s(u)$ may be chosen for example as $u^3 - u$ and the right hand side is usually $f = 0$.

The Neumann boundary control q can be interpreted as external magnetic fields which have an impact on the domain Ω.

4.2 General optimization problem

The weak formulation of system (4.1) in which a state variable $u \in H^1(\Omega)$ and a control function $q \in L^2(\Gamma_C)$ is determined by requiring

$$(F(u,q),\phi) \;=\; 0 \quad \forall \phi \in \; H^1(\Omega).$$

Here, the functional $F: \; H^1(\Omega) \; \times \; L^2(\Gamma_C) \; \rightarrow H^1(\Omega)'$ is defined by

$$(F(u,q),\cdot) \;=\; (\nabla u, \nabla \cdot)_\Omega \; + \; (s(u),\cdot)_\Omega \; - \; (f,\cdot)_\Omega \; - \; (q,\cdot)_{\Gamma_C},$$

where $(.,.)_\Omega$ and $(.,.)_{\Gamma_C}$ denote the L^2 inner products over Ω and Γ_C, respectively.

We consider an optimal control problem for the simplified Ginzburg-Landau model. For a prescribed profile u_d the boundary control variable q is sought to minimize the distance between u and u_d. This profile may be given on the whole domain or on parts of its boundary. The corresponding objective function $J: \; H^1(\Omega) \; \times \; L^2(\Gamma_C) \; \rightarrow \mathbb{R}$ is

$$J(u,q) = \frac{1}{2} \; \|u - u_d\|^2_{obs}.$$

The index '*obs*' indicates an evaluation only in that part of the domain, where we evaluate the objective function ('observe'). In this case the control variable q may be viewed as modeling the effect of an external magnetic field.

To enhance the stability of the optimization problem, we augment the objective function by a regularization term,

$$J(u,q) = \frac{1}{2} \; \|u - u_d\|^2_{obs} \; + \; \frac{\alpha}{2} \; \|q - q_0\|^2_{\Gamma_C}, \tag{4.2}$$

where q_0 is a suitable reference value. Besides avoiding ill-posedness and improving conditioning, regularization makes rigorous mathematical analysis possible under less restrictive assumptions. Particularly, in the context of partial differential equations the regularization gives control on the optimization variable which guarantees solvability and convergence of approximations (see section 1.8). As we can see, this regularization changes our setting as we do not solve the original optimization problem (see e.g. Figures 2.3 and 2.4). There are theoretical considerations (for details see [41]) as well as practical experiences which indicate that calculations are also possible without regularization in this case.

Since the calculations were stable, there was no need to use a stabilization besides the regularization in the cost functional.

The optimization problem is well-posed by [41] and [34].

The first order necessary conditions for this (NBC) optimization problem is

$$\begin{aligned}
(u,\psi)_{obs} \;-\; (u_d,\psi)_{obs} \;+\; (\nabla\psi,\nabla\lambda)_\Omega \;+\; (s'(u)\;\psi,\lambda)_\Omega \;&=\; 0, && (4.3)\\
\alpha(q,\chi)_{\Gamma_2} \;-\; \alpha(q_0,\chi)_{\Gamma_2} - \;(\chi,\lambda)_{\Gamma_2} \;&=\; 0, && (4.4)\\
(\nabla u,\nabla\phi)_\Omega \;+\; (s(u),\phi)_\Omega \;-\; (f,\phi)_\Omega \;-\; (q,\phi)_{\Gamma_2} \;&=\; 0. && (4.5)
\end{aligned}$$

For the discrete optimization problem, the first order necessary conditions for this (NBC) optimization problem reads

$$\begin{aligned}
(u_h,\psi_h)_{obs} \;-\; (u_d,\psi_h)_{obs} \;+\; (\nabla\psi_h,\nabla\lambda_h)_\Omega \;+\; (s'(u_h)\;\psi_h,\lambda_h)_\Omega \;&=\; 0, && (4.6)\\
\alpha(q_h,\chi_h)_{\Gamma_2} \;-\; \alpha(q_0,\chi_h)_{\Gamma_2} - \;(\chi_h,\lambda_h)_{\Gamma_2} \;&=\; 0, && (4.7)\\
(\nabla u_h,\nabla\phi_h)_\Omega \;+\; (s(u_h),\phi_h)_\Omega \;-\; (f_h,\phi_h)_\Omega \;-\; (q_h,\phi_h)_{\Gamma_2} \;&=\; 0. && (4.8)
\end{aligned}$$

The resulting discrete solutions are the calculated solutions.

The left hand side in the Newton method (1.10) is

$$\begin{pmatrix}
(\delta u,\psi)_{obs} \;+\; (s''(u)\psi\delta u,\lambda)_\Omega \;+\; (\nabla\psi,\nabla\delta\lambda)_\Omega \;+\; (s'(u)\psi,\delta\lambda)_\Omega \\
\alpha(\delta q,\chi)_{\Gamma_2} \;-\; (\chi,\delta\lambda)_{\Gamma_2} \\
(\nabla\delta u,\nabla\phi)_\Omega \;+\; (s'(u)\delta u,\phi)_\Omega \;-\; (\delta q,\phi)_{\Gamma_2}
\end{pmatrix}. \qquad (4.9)$$

The discretization of this equation system is done by a finite element Galerkin method with Q^1-elements. The meshes fulfill the usual regularity conditions. 'Hanging nodes' are allowed and facilitate local mesh refinement, but at most one 'hanging node' per edge. For the state and adjoint variables, piecewise polynomial (linear or bilinear) shape functions are taken. For the control variables, the traces of the above shape functions on Γ_C are used. The discretization is realized by using the DEAL library ([8]).

4.3 Weighted a posteriori error estimator

Proposition 4.3.1. *For control of the given cost functional $J(\cdot)$, there holds the weighted* a posteriori *error estimate*

$$\begin{aligned}
|J(u,q) - J(u_h,q_h)| \le \sum_{\Gamma\subset\partial\Omega} h_\Gamma^2 \{\rho_\Gamma^{(\lambda)}\omega_\Gamma^{(u)} + \rho_\Gamma^{(u)}\omega_\Gamma^{(\lambda)}\} + \sum_{\Gamma\subset\Gamma_C} h_\Gamma^2 \rho_\Gamma^{(q)}\omega_\Gamma^{(q)} \qquad (4.10)\\
+ \sum_{T\in\mathbf{T}_h} h_T^2 \{\rho_T^{(u)}\omega_T^{(\lambda)} + \rho_{\partial T}^{(u)}\omega_{\partial T}^{(\lambda)} + \rho_T^{(\lambda)}\omega_T^{(u)} + \rho_{\partial T}^{(\lambda)}\omega_{\partial T}^{(u)}\},
\end{aligned}$$

with the cell residuals and weights

$$\begin{aligned}
\rho_\Gamma^{(\lambda)} &= h_\Gamma^{-3/2}\|u_h - u_O + \partial_n\lambda_h\|_\Gamma, \quad \Gamma\subset\Gamma_O, & \omega_\Gamma^{(u)} &= h_\Gamma^{-1/2}\|u-\psi_h\|_\Gamma,\\
\rho_\Gamma^{(\lambda)} &= h_\Gamma^{-3/2}\|\partial_n\lambda_h\|_\Gamma, \quad \Gamma\subset\partial\Omega\backslash\Gamma_O, & &\\
\rho_\Gamma^{(u)} &= h_\Gamma^{-3/2}\|\partial_n u_h - q_h\|_\Gamma, \quad \Gamma\in\Gamma_C, & \omega_\Gamma^{(\lambda)} &= h_\Gamma^{-1/2}\|\lambda-\pi_h\|_\Gamma,\\
\rho_\Gamma^{(u)} &= h_\Gamma^{-3/2}\|\partial_n u_h\|_\Gamma, \quad \Gamma\subset\partial\Omega\backslash\Gamma_C, & &\\
\rho_\Gamma^{(q)} &= h_\Gamma^{-3/2}\|\lambda_h - \alpha q_h\|_\Gamma, & \omega_\Gamma^{(q)} &= h_\Gamma^{-1/2}\|q-\chi_h\|_\Gamma,
\end{aligned}$$

$$\begin{aligned}
\rho_T^{(u)} &= h_T^{-1}\|\Delta u_h - s(u_h) + f\|_T, & \omega_T^{(\lambda)} &= h_T^{-1}\|\lambda-\pi_h\|_T,\\
\rho_{\partial T}^{(u)} &= \tfrac{1}{2}h_T^{-3/2}\|n\cdot[\nabla u_h]\|_{\partial T\backslash\partial\Omega}, & \omega_{\partial T}^{(\lambda)} &= h_T^{-1/2}\|\lambda-\pi_h\|_{\partial T\backslash\partial\Omega},\\
\rho_T^{(\lambda)} &= h_T^{-1}\|\Delta\lambda_h - s'(u_h)\lambda_h\|_T, & \omega_T^{(u)} &= h_T^{-1}\|u-\psi_h\|_T,\\
\rho_{\partial T}^{(\lambda)} &= \tfrac{1}{2}h_T^{-3/2}\|n\cdot[\nabla\lambda_h]\|_{\partial T\backslash\partial\Omega}, & \omega_{\partial T}^{(u)} &= h_T^{-1/2}\|u-\psi_h\|_{\partial T\backslash\partial\Omega}.
\end{aligned}$$

Proof. In the present case, there holds

$$\begin{aligned}
L(v) - L(v_h) &= J(u,q) + (\nabla u,\nabla\lambda)_\Omega + (s(u)-f,\lambda)_\Omega - (q,\lambda)_{\Gamma_C}\\
&\quad - J(u_h,q_h) - (\nabla u_h,\nabla\lambda_h)_\Omega - (s(u_h)-f,\lambda_h)_\Omega + (q_h,\lambda_h)_{\Gamma_C}\\
&= J(u,q) - J(u_h,q_h),
\end{aligned}$$

since $\{u,\lambda,q\}$ and $\{u_h,\lambda_h,q_h\}$ satisfy the equations (4.5) and (4.8), respectively. Hence, error control with respect to the Lagrangian functional $L(\cdot)$ and the cost functional $J(\cdot)$ is equivalent. Now, the general error identity (2.35) implies that

$$|J(u,q) - J(u_h,q_h)| = \inf_{\phi_h\in V_h}|L'(v_h;v-\phi_h)|, \tag{4.11}$$

where $v_h = \{u_h,\lambda_h,q_h\}$ and $v = \{u,\lambda,q\}$. From (4.6) - (4.8), we see that

$$\begin{aligned}
L'(v_h;v-\phi_h) &= (u_h - u_O, u-\psi_h)_{\Gamma_O} + (\nabla(u-\psi_h),\nabla\lambda_h)_\Omega\\
&\quad + (u-\psi_h, s'(u_h)\lambda_h)_\Omega \quad + (\nabla u_h,\nabla(\lambda-\pi_h))_\Omega + (s(u_h)-f,\lambda-\pi_h)_\Omega\\
&\quad - (q_h,\lambda-\pi_h)_{\Gamma_C} \quad + (\lambda_h-\alpha q_h, q-\chi_h)_{\Gamma_C}.
\end{aligned}$$

Splitting the global integrals into the contributions from each single cell $T\in\mathbf{T}_h$ and each cell edge $\Gamma\subset\partial\Omega$, respectively, and integrating locally by parts yields

$$\begin{aligned}
L'(v_h;v-\phi_h) &= \sum_{\Gamma\subset\Gamma_O}(u_h-u_O+\partial_n\lambda_h, u-\psi_h)_\Gamma + \sum_{\Gamma\subset\partial\Omega\backslash\Gamma_O}(\partial_n\lambda_h, u-\psi_h)_\Gamma\\
&\quad + \sum_{\Gamma\subset\Gamma_C}(\partial_n u_h - q_h,\lambda-\pi_h)_\Gamma + \sum_{\Gamma\subset\partial\Omega\backslash\Gamma_C}(\partial_n u_h,\lambda-\pi_h)_\Gamma\\
&\quad + \sum_{\Gamma\subset\Gamma_C}(\lambda_h-\alpha q_h, q-\chi_h)_\Gamma\\
&\quad + \sum_{T\in\mathbf{T}_h}\big\{(-\Delta u_h + s(u_h) - f,\lambda-\pi_h)_T + \tfrac{1}{2}(n\cdot[\nabla u_h],\lambda-\pi_h)_{\partial T\backslash\partial\Omega}\big\}\\
&\quad + \sum_{T\in\mathbf{T}_h}\big\{(u-\psi_h, -\Delta\lambda_h + s'(u_h)\lambda_h)_T + \tfrac{1}{2}(u-\psi_h, n\cdot[\nabla\lambda_h])_{\partial T\backslash\partial\Omega}\big\}.
\end{aligned}$$

From this the asserted relation follows by applying the Hölder inequality. □

4.4 Comparison to other approaches

In this section, some numerical results will be presented, which have been obtained with the code 'rhoptcon'. The adaptive mesh refinement is done after some Newton iterations on the discretization level. The Newton system is therefore not necessarily in the limit of the Newton iteration. The main purpose is to demonstrate that our solution approach is capable to reproduce solutions of certain test problems obtained by other authors.

The presented application to superconductivity was already considered in a paper of Ito and Kunisch [41]. These authors applied an augmented Lagrangian approach for stabilizing the saddle point problem. This was discretized by the usual five-point difference operator on an equidistant grid without adaptivity. From [41] we recall the following test configuration ('Run1' in [41]):

$$\begin{aligned}
&\Omega = [0,1] \times [0,2], \qquad \Gamma_C \;=\; \partial\Omega, \qquad obs \;=\; \Omega, \\
&s = u^3 - u, \qquad f = 0, \\
&J(u,q) \;=\; \frac{1}{2} \int_\Omega |u - u_d|^2 \;+\; \frac{\alpha}{2} \int_{\Gamma_C} (q - q_0)^2, \\
&u_0 = 3 \;,\; \lambda_0 = 0 \;,\; q_0 = 0 \;,\; u_d = 3.
\end{aligned}$$

The regularization factor has the value $\alpha = 10^{-3}$. We note that in this case, the code also allows calculations without regularization. If we start with $u_0 = 3$ as proposed in [41], the solver immediately terminates since we are too close to an optimum. Therefore, we chose other starting values for the comparison. In the tables of this section the following notation is used:

- The starting value for u is 'u_0'.
- The regularization parameter is 'α'.
- '#iter' is the number of Newton iterations required to reach a certain prescribed level for the the norm of the algebraic Newton residual. We note that on the discrete level our iteration corresponds only to an approximate Newton method.
- The value of the objective function for the computed approximation is '$J(u_h, q_h)$'.
- The Newton residual and the Newton increment both measured in the discrete Euclidean norm are denoted by '$\text{res}_{\text{Newton}}$' and '$\text{incr}_{\text{Newton}}$', respectively.
- The calculation time in CPU seconds is 'time'.
- L is the maximal number of Newton steps which are performed between two refinement cycles.

The following tables show the results of some calculations by our code for the present test case. In the first test, we fix the number of Newton iterations to $L = 8$ and compare the effect of varying u_0 and α.

u_0	α	#iter	$J(u_h, q_h)$	$\text{res}_{\text{Newton}}$	$\text{incr}_{\text{Newton}}$	time
100.	0.0	48	6.26262e-6	0.0020208	0.0019185	1554
100.	0.01	42	6.28279e-6	0.0019931	0.0017035	≈800
0.0	0.001	47	0.0743429	0.0050597	0.0051517	≈2500

The values obtained for the objective function show that, in the first two cases, we reach a global minimum while in the third case apparently only a local minimum is obtained. The global minimum and the corresponding mesh is shown in Figure 1.

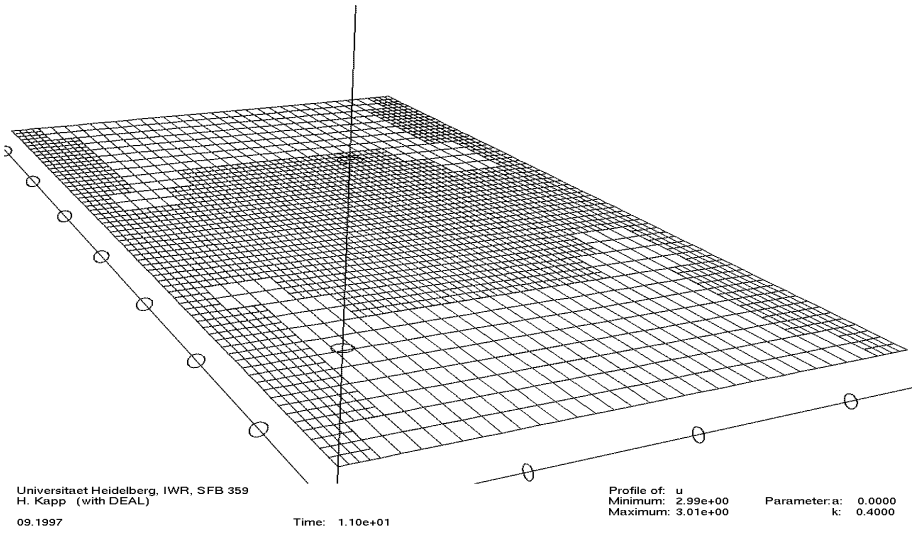


Fig 1: Run1 - close to the global minimum

If only $L = 4$ Newton steps are performed between two refinement cycles, we get the following result:

u_0	α	#iter	$J(u_h, q_h)$	$\text{res}_{\text{Newton}}$	$\text{incr}_{\text{Newton}}$	time
0.0	0.001	23	0.0739137	0.0058241	0.0056670	≈650

This corresponds to a local minimum shown in Figure 2 which was obtained in [41]. This test demonstrates that mesh adaptivity may have a strong influence on the optimization process. Further, we see that in the case of convergence the pure Newton method can give very good results even without regularization as demonstrated in the following table:

u_0	α	#iter	$J(u_h, q_h)$	$\text{res}_{\text{Newton}}$	$\text{incr}_{\text{Newton}}$	time
100.	0.0	15	2.73916	0.001741	5.82511e-8	61
150.	0.0	16	2.73916	0.0017416	4.73266e-8	56
-100.	0.0	32	2.73917	0.0001957	5.22385e-8	569
-7.0	0.0	14	2.73915	0.0003929	1.79801e-8	105
-17007.	0.0	47	2.73916	3.64664e-5	1.55469e-8	34060

In this case, $L = 4$ Newton steps are performed between mesh adaptation cycles. However, for all starting values, we do not reach a global minimum. The obtained local minimum is shown in Figure 3.

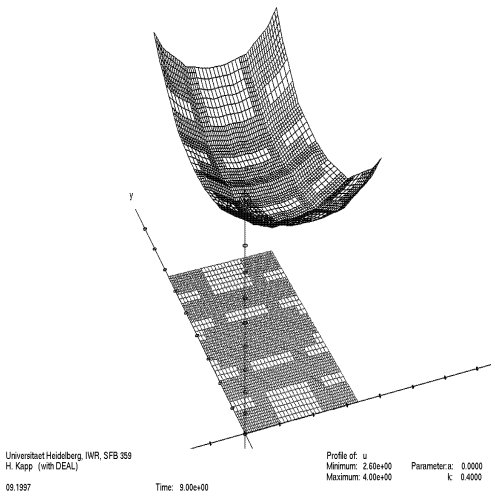


Fig 2: Run1 - a local minimum

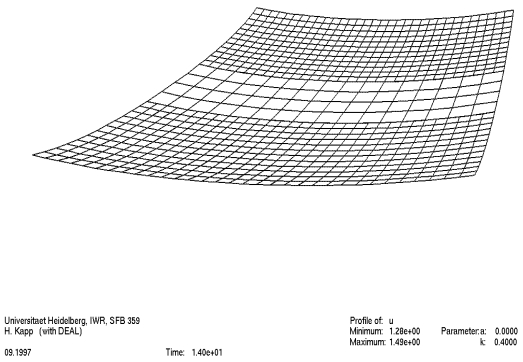


Fig 3: Run1 - pure Newton method

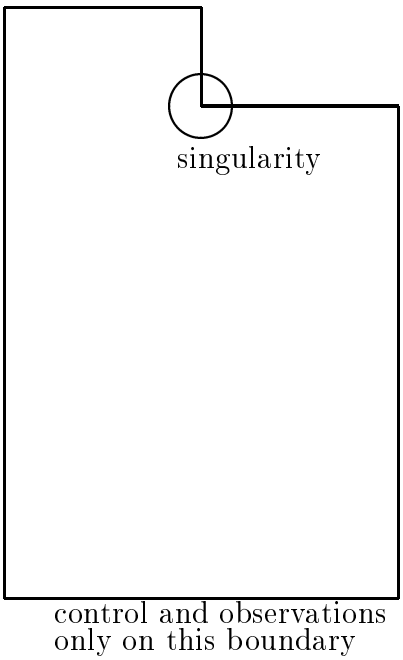
Finally, we consider a non homogeneous right hand side, $f = 100^3 - 100$. Using again $L = 4$ Newton steps per refinement cycle the following results are obtained, showing that again only a local minimum is reached:

u_0	α	#iter	$J(u_h, q_h)$	$\text{res}_{\text{Newton}}$	$\text{incr}_{\text{Newton}}$	time
0.0	0.001	24	0.0739041	0.0056094	0.0057108	753
0.0	0.0	24	0.0739037	0.0056095	0.0057108	753

For the other test cases in [41], our code has produced similar results. We omit further details. For these first tests the mesh adaptation has been driven by the simple energy error indicator mentioned above. In the next section we will compare this approach against our new weighted error estimator.

4.5 Numerical results for heuristic error estimators

In this section, we compare the performance of the different error indicators 'opt1', 'opt2', and 'energy' defined in chapter 2 at the following test problem. The presented two versions of the error indicator 'opt1', which will be named 'opt1_2' and 'opt1', differ only in the presence of the residual term in $\rho_K(u_h, q_h)$ in addition to the normal-jump terms. For the error estimator 'opt1', we consider only the jump terms. This difference is motivated by the observation that, for linear finite elements in the case of smooth f, 'the contribution of the normal jump terms asymptotically dominates that of the domain residuals, and the latter may therefore be neglected' (see [13]). The results are obtained with the code 'rhoptcon', so the refinement is not necessarily done in the limit of the Newton iterations on each discretization level.



$$\Gamma_C \;=\; obs \;\subset\; \partial\Omega$$

$$s = u^3 - u$$

$$f = 0$$

$$J(u,q) \;=\; \tfrac{1}{2}\int_{\Gamma_C}\left\{|u-u_d|^2 + \alpha(q-q_0)^2\right\}ds$$

$$u_0 = 5\ ,\ \lambda_0 = 0\ ,\ q_0 = 0$$

$$\alpha = 0$$

$$u_d(x,y) \;=\; \begin{cases} \frac{1}{|y-0.5|}, \\ \quad \text{for } \ y < 0.45 \ \text{ or } y \geq 0.55, \\ -(y-0.5)^2 + 25.00025, \\ \quad \text{for } \ 0.45 \leq y < 0.55\ . \end{cases}$$

Using the energy error indicator, the adaptive mesh refinement leads to the result (grid and solution) shown in Figure 4, while those obtained with 'opt1' and 'opt2' are shown in Figures 5 and 6, respectively. Comparing the grids, we see that the energy indicator tends to over refine at the corner singularity which is insignificant for the optimization process while the other two indicators correctly cause refinement along the control/observation boundary. Of course, all three estimators yield stronger refinement in the areas of larger variations of the state variable.

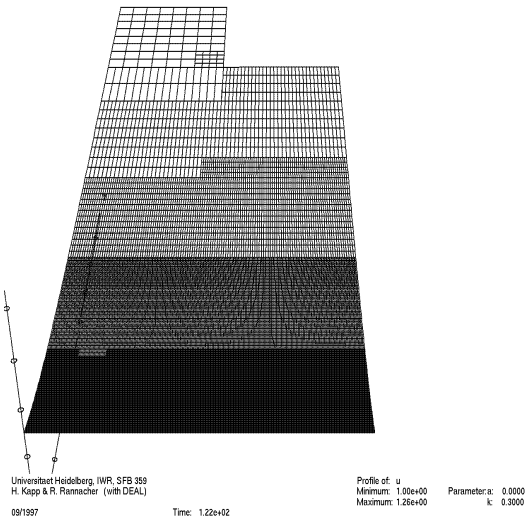


Fig 4: Energy error indicator

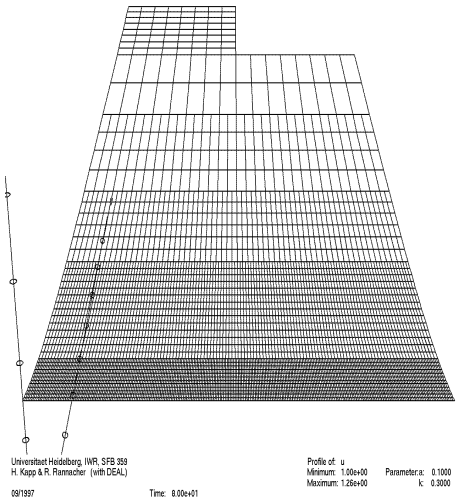


Fig 5: Error indicator opt1

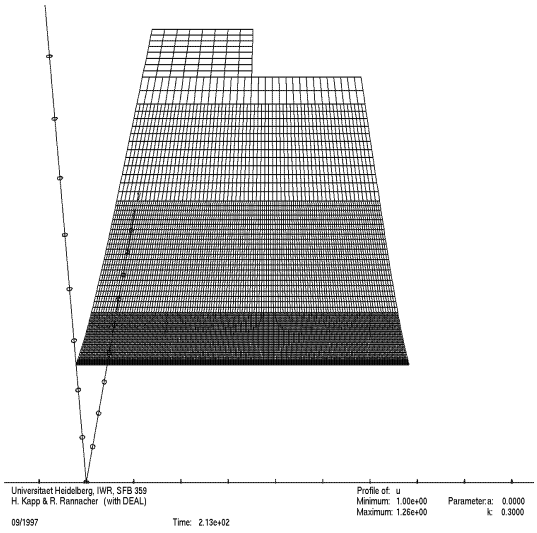


Fig 6: Error indicator opt2

Figures 7 and 8 show the corresponding Lagrangian multiplier λ and control variable q for the indicator 'opt1', respectively. Clearly, q fulfills the condition that the trace of the Lagrangian multiplier must be equal to the control.

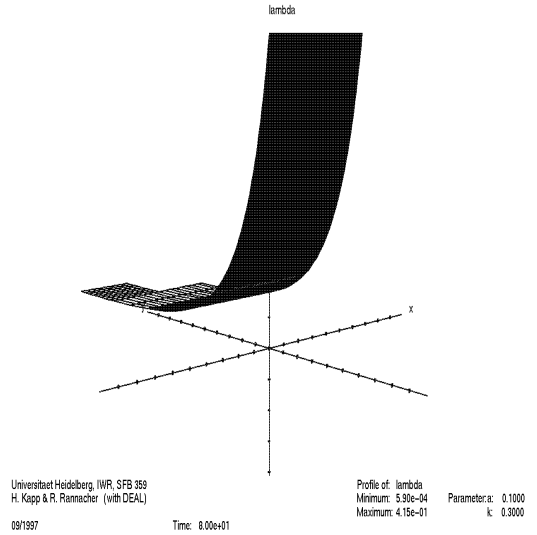


Fig 7: Adjoint variable λ

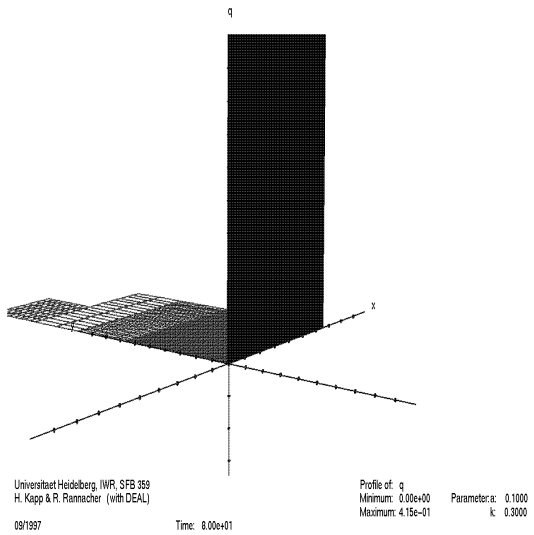


Fig 8: Control q

The following Figures 9 and 10 show the convergence history for the several error indicators and the corresponding values of the objective function of our solution method in

the case that the Newton method is used *without globalization*.

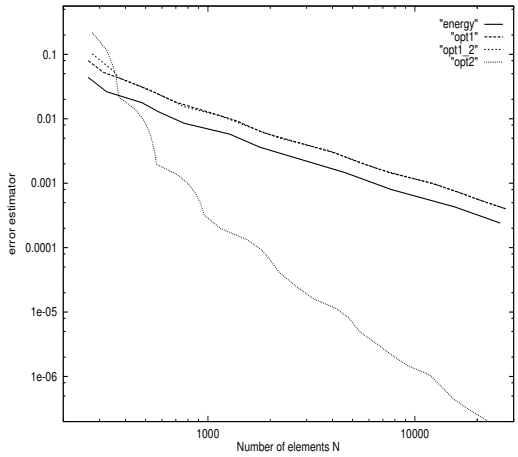


Fig 9: Values error indicators

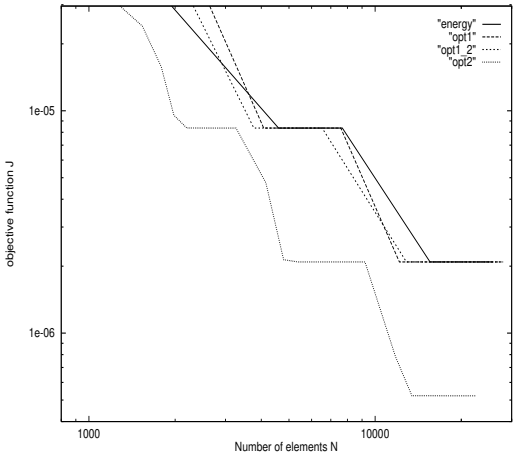


Fig 10: Values objective function

The results for 'opt1' and 'energy' are very close to each other since both indicators use essentially smoothness information, while 'opt2' also measures the Newton iteration error which apparently yields much better results. The algebraic Newton residual shows a similar behavior for all three indicators, see Figure 11.

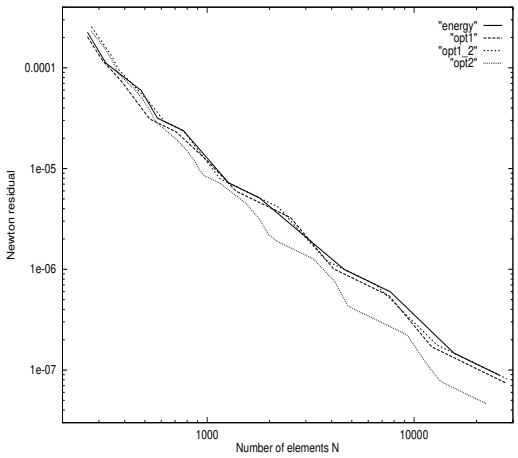


Fig 11: Discrete Newton residual

Using a modified Armijo-Goldstein line search strategy in the Newton iteration, we observe quite different convergence behavior for error indicators and corresponding values for the objective function as shown in Figures 12 and 13.

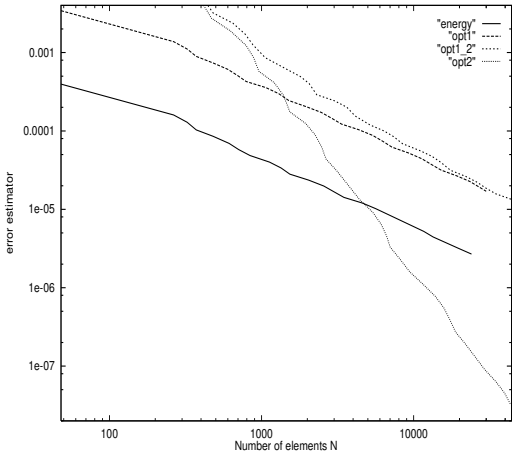


Fig 12: Values error indicators

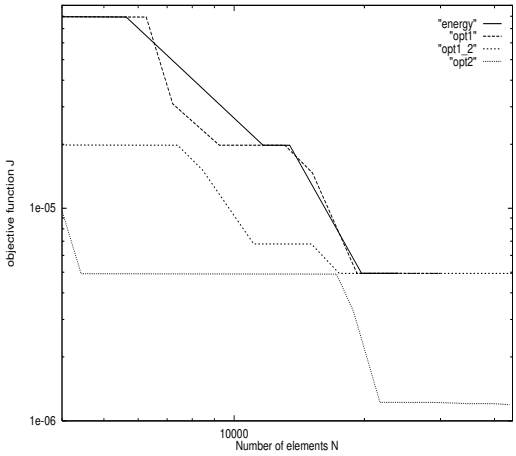


Fig 13: Values objective function

Again the Newton residual and Newton increment exhibit a similar development for all three indicators shown in Figures 14 and 15.

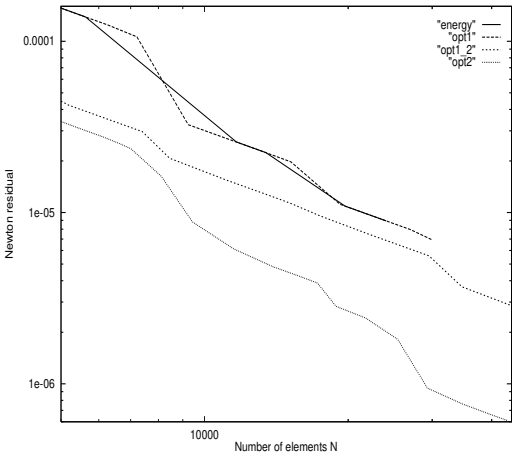


Fig 14: Discrete Newton residual

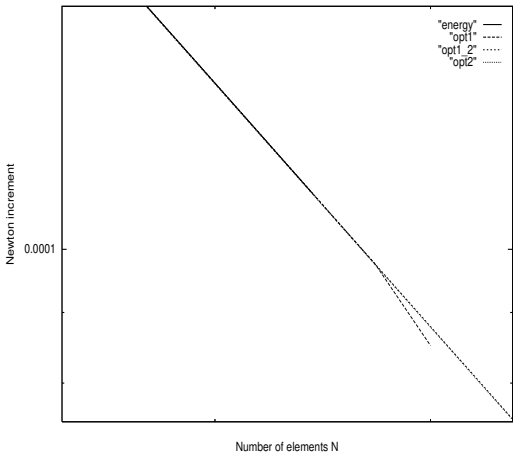


Fig 15: Discrete Newton increment

We summarize that in both cases, plain Newton and globalized Newton, the values of the error indicators and the corresponding values of the objective function show a similar development. However, for the plain Newton method, the values of the objective function are smaller. This is due to the fact that the plain Newton method has a higher convergence rate than the globalized Newton method. This effect can also be seen by the values for the Newton residual. For the globalized Newton method, we have a larger difference between 'opt2' and the other two error indicators than for the plain Newton method. This reflects the fact that 'opt2' also measures the size of the Newton residual which causes differences if we are still far away from the limit point.

4.6 Numerical results for weighted error estimator

The equations of state are again the *nonlinear* Ginzburg-Landau equations

$$\begin{aligned} -\Delta u + s(u) &= f \quad \text{on } \Omega, \\ \partial_n u &= q \ \text{ on } \Gamma_C, \quad \partial_n u = 0 \ \text{ on } \partial\Omega \setminus \Gamma_C, \end{aligned} \tag{4.12}$$

with $s(u) = u^3 - u$ and right hand side $f = 0$. The corresponding first-order necessary condition to be solved are the equations (4.3)-(4.5).

By the nonlinear situation, the derivation of the dual-weighted error estimator introduces an additional linearization error in the duality argument. Theory as well as practical experience show that, in the present case, this additional error is of higher order on well-adapted meshes and can therefore be neglected. The *a posteriori* error estimate derived in (4.10) is applied. The discretization is the same as in the previous sections combined with linearization by a Newton iteration. We note that the Newton iteration is always carried to the limit before the error estimator is applied for mesh refinement. The computations are done with the code 'bkr'. The results of this process may significantly differ from those obtained if discretization and iteration error are mixed together (see the preceding sections 4.4 and 4.5).

We again compare the dual-weighted error estimator with a simple ad hoc energy-error estimator. We consider two different choices for the boundaries of control and observation. First, the extreme situation is taken that control and observation are on the same boundary, $\Gamma_C = \Gamma_O$ (lower boundary of the T-shaped domain). This is Configuration 1 on page 56. In this case, we have the main parts of the optimization problem at one boundary. Hence, we do not expect any need for induced stronger mesh refinement 'far away' from this boundary if we only want to deal with the optimization problem. In the second case, we take the control and the observation on opposite boundaries, $\Gamma_C \cap \Gamma_O = \emptyset$ (lower and upper boundary of the T-shaped domain). This is Configuration 2 on page 56. In this case, we expect better results for the energy-error estimator because the information must pass from the control to the observation boundary and the corner singularities will have a stronger effect on the mesh refinement. Therefore, the simulation will play a more decisive role for mesh refinement.

Test case 1: The observations for Configuration 1 are taken as $u_d(x) = \sin(0.19x)$. The following table shows the quality of the dual-weighted error estimator for quantitative error control for this first nonlinear test case for $\alpha = 0$. This means that there is no regularization and the original optimization problem is solved.

N	596	1616	5084	8648	15512
E_h	2.56e-04	2.38e-04	8.22e-05	4.21e-05	3.99e-05
I_{eff}	0.34	0.81	0.46	0.29	0.43

The reference value $J(u,q)$ for the objective function is computed on a refined mesh with about 131000 cells. Due to the special choice $\Gamma_C = \Gamma_O$, the dual solution equals zero almost everywhere away from Γ_C, and the error indicators $\rho_\Gamma^\lambda, \Gamma \subset \Gamma_O$ in (4.10) dominate all the other terms in the estimator. The dual-weighted error estimator considers only the neighborhood of the control boundary, whereas the energy-error estimators reflect too

much the singularity in primal solution at the reentrant corners (see Figure 4.2). The distributions of the values for the error estimators $\eta_E(u_h)$, $\eta_E(u_h, \lambda_h)$ and $\eta_w(u_h, \lambda_h, q_h)$ are given in Figure 4.1. It should be mentioned that the coupling between the control and the observation is by computation on the whole domain Ω and not only by assignment on the boundary $\Gamma_C = \Gamma_O$.
In Figure 4.3, we compare the efficiency of the meshes generated by the two estimators in the first nonlinear case with $\alpha = 0$. We see that in this "extreme" boundary layer example, we can approximate the solution of the optimization problem on a grid with much less cells using the dual-weighted error estimator. In this example it is possible to get the same accuracy with the dual-weighted error estimator on 3500 elements compared to the energy estimator on 100000 elements. This means that the heuristic energy-error indicator produce inefficient meshes in this example.

Which values do the several terms of the dual-weighted error estimator have? For test case 1, the main part of the optimization problem is focused on the 'optimization boundary' $\Gamma_C = \Gamma_O$. It seems therefore consequential that the dominant integrals lie on this boundary. The following table shows the detailed information for the dual-weighted error estimator. The whole value of the integrals on the cells are split in their several parts. The notation is the one of proposition 4.3.1:

N	1616	15512	81536
$\sum \rho_w$	0.0032855	0.000827338	0.000421932
$\sum \rho_T$	9.1e-11	1.1e-10	1 e-10
$\sum \rho_\Gamma^\lambda, \Gamma \subset \Gamma_O$	0.00328542	0.000827337	0.000421932
$\sum \rho_\Gamma^u, \Gamma \in \Gamma_C$	8.2e-08	6.3e-10	7.2e-11
$\sum \rho_\Gamma^q$	2.7e-10	1.1e-10	5.3e-11

This is an explanation for the big gain which can be achieved by the dual-weighted error estimator. The heuristic energy-error indicator does not use this important information on the boundary in an appropriate way and therefore it leads to inefficient meshes. Such an extreme behavior can normally be expected if the *boundary indicators are dominant over the interior indicators.*
Test case 2: For Configuration 2, the observations are taken as $u_d \equiv 1$ and the regularization factor α is set to 0.1 or 1. Now, depending on the nonlinearity $s(u)$, there exist several stationary points of $\mathcal{L}(u, q, \lambda)$. By varying the starting values for the Newton iteration, these solutions can be approximated. One solution corresponds to constant $u \equiv u_d$, which is actually the global minimum. For this stationary point, we get an objective function value equal to zero (up to round-off error). Accordingly, we match these observations with our numerical solution already on a rather coarse mesh with $N = 512$ cells. The corresponding Newton residual and Newton increment are both converged to zero. We do not show the trivial results of these computations.

The other two obtained stationary points are symmetric to each other with respect to the plane $\{x = 0\}$ in this case. These computed solutions correspond to a local minimum and a local maximum by second order information of the optimization problem. The following table shows the quality of the dual-weighted error estimator $\eta_w(u_h, \lambda_h, q_h)$ for error control of one of these stationary points (the second one) for test case 2 with $\alpha = 0.1$. The reference value 0.04888934625... for the objective function is obtained on an adaptive

mesh with $N = 545216$ cells corresponding more than 10^6 unknowns.

N	512	15368	27800	57632	197408
E_h	9.29e-05	8.14e-07	4.86e-07	2.31e-07	4.58e-08
I_{eff}	1.32	0.56	0.35	0.42	0.32

The numerical results demonstrate the correct qualitative behavior of the dual-weighted error estimator. The effectivity index indicates also a relatively good quantitative accuracy (with interpolation constant $C_I = 0.1$), although the values produced are still too big. This defect is caused by taking the absolute signs under the sums thereby suppressing possible error cancellation. Furthermore, the error E_h is very small for $\alpha = 0.1$. In the case $\alpha = 1$, we get better results as shown in the following table for the second stationary point.

N	512	8120	25544	42608	126284
E_h	2.08e-03	4.35e-05	9.26e-06	5.95e-06	8.94e-07
I_{eff}	0.52	0.73	0.88	1.21	0.98

Next, Figure 4.4 shows the distribution of local cell indicators for the three error estimators in the critical case $\alpha = 0.1$. Figure 4.5 shows the corresponding computed discrete solutions. Obviously, the dual-weighted error estimator induces a much stronger refinement along the observation and control boundaries which seems more relevant for the optimization process than resolving the corner singularities. Whereas the energy-error estimator emphasizes the corner singularities.

In Figure 4.6, a faster convergence to the solution of the continuous problem with the dual-weighted error estimator can be stated. Especially interesting is that the values of $\eta_E(u_h, \lambda_h)$ are worse than those of $\eta_E(u_h)$. Normally, one would expect a better behavior (as in Figure 4.3) because of the additional dual information which is used. This shows that the energy-error indicators are just based on heuristic criteria. The observed jumps in the plotted results can be explained by the hanging nodes. There are some hanging nodes introduced at critical points. This deteriorates the obtained results.

Finally, for the third stationary point, we indicate the efficiency of the generated meshes. As seen from the effectivity index in the following table for the third stationary point, the quantitative behavior of the dual-weighted estimator is very good in this case (with interpolation constant $C_I = 0.1$).

N	1784	4544	15452	29096	77096
E_h	1.663e-05	6.02e-06	1.54e-06	7.43e-07	2.73e-07
I_{eff}	0.91	0.97	0.97	0.82	0.84

For test case 2, the dominant parts of the error estimator are different from those in test case 1. The total contribution of the interior cell indicators is dominant over that of the boundary indicators. The following table will show this split in several parts of the values of the cell indicators for the second stationary point. The notation is again the one of proposition 4.3.1:

N	8120	77096	283016
$\sum \rho_w$	0.000566207	5.162e-05	1.289e-05
$\sum \rho_T$	0.000498468	4.857e-05	1.181e-05
$\sum \rho_\Gamma^\lambda, \Gamma \subset \Gamma_O$	3.3e-05	1.50e-06	5.3e-07
$\sum \rho_\Gamma^u, \Gamma \in \Gamma_C$	3.4e-05	1.54e-06	5.5e-07
$\sum \rho_\Gamma^q$	9.8e-15	2 e-14	1.3e-13

This gives another explanation why the gain in test case 2 is not so big as for test case 1. The interior local cell indicators play a much more important role than in test case 1. This means that the error is much bigger in the interior than on the boundaries. Hence the energy-error indicators can get the true error of the optimization problem in a better way than in test case 1.

It should be mentioned that calculations with up to 1.4 million variables on 700 000 grid points have been done for the presented application with this code 'bkr'.

By the above results it can be concluded that it is possible to derive good criteria in an analytic way for model reduction in optimization with the presented Ginzburg-Landau model. The model reduction process is driven by the error indicators leading to small discrete optimization models. The qualitative error estimation with the dual-weighted error estimator enables to get good numerical results. The important properties of the original continuous optimization problem are preserved by reduced models in a certain accuracy.

The quantitative error estimation is also successful. This is not instantly clear because the coupling of the different equations and scalings can lead to many problems. By the numerical results, the value of the developed dual-weighted error estimator is a good estimator for the error in the discrete optimization problem. The error control is solely based on the computed primal solution of the variables u_h, q_h, λ_h and is therefore relatively cheap.

4.7 Modified Newton method

The modified Newton method of section 3.3 will be applied to the problem of section 4.6. Globalization methods may decelerate the convergence in comparison with convergence of the pure Newton method. But one advantage is a larger range of convergence of the globalized Newton method. Furthermore, this modified Newton method leads only to stationary points which are minima as shown below. The starting values of the variables decide which minimum is obtained.

By the second order information it can be stated that in the calculations of section 4.6, test case 2, the global minimum is really a minimum, the second stationary point is a local maximum and the third stationary point is a local minimum (which could have been imagined by the fact that the latter two saddle points lie symmetrically to each other).

Several tests were done with the configuration of section 4.6, test case 2. The initial values of u were changed in order to test the range of convergence. Results are shown in the tables below. The following notation is used:

- N: the number of cells,

- 'JJ': the value of the cost functional in the limit of the Newton iteration on the discrete level,
- '#corr': the number of necessary updates by the modified Newton methods (corrections),
- 'max TRfactor': the maximal value of a factor for the update of the Hessian for the modified Newton method on this discrete level,
- 'min TRfactor': the minimal value of a factor for the update of the Hessian for the modified Newton method on this discrete level,
- 'CPU seconds TR': the CPU seconds of the modified Newton method necessary to reach the limit,
- 'CPU seconds pure': the CPU seconds of the pure Newton method necessary to reach the limit.

The regularization factor in all tests is $\alpha = 1$.

Test1: Starting value of u in the iterations is -200. Both methods lead to the third stationary point which is a local minimum. The starting value of the cost functional is JJ=90395.7:

N	128	632	1160	1784
JJ	8.015748	7.997655	7.997659	7.997657
#corr	9	0	1	0
max TRfactor	$3 * 10^7$	0	0.001	0
min TRfactor	25.1	0	0.001	0
CPU seconds TR	124.55	314.21	713.35	894.98
CPU seconds pure	59.64	196.36	417.94	564.31

Test2: Starting value of u in the iterations is 1100. The modified Newton method leads to the global minimum. The pure Newton method leads to the local maximum. Therefore a comparison of the CPU seconds is omitted. The pure Newton method needs more iterations due to the more difficult structure of the stationary point. The starting value of the cost functional is JJ=107526224.8.

Test3: Even for a starting value $u = 951100000000$ in the iterations, the modified Newton method leads to the global minimum. The starting value of the cost functional is JJ=$8.04 * 10^{23}$.

The following table shows numerical results for test2 and test3 for the last discretization level in each test case:

test	test2	test3
N	512	728
JJ	$7 * 10^{-11}$	$8.36 * 10^{-28}$
#corr	1	2
max TRfactor	0.368645	2.02635
min TRfactor	0.368645	0.135268
CPU seconds TR	297.67	1568.95

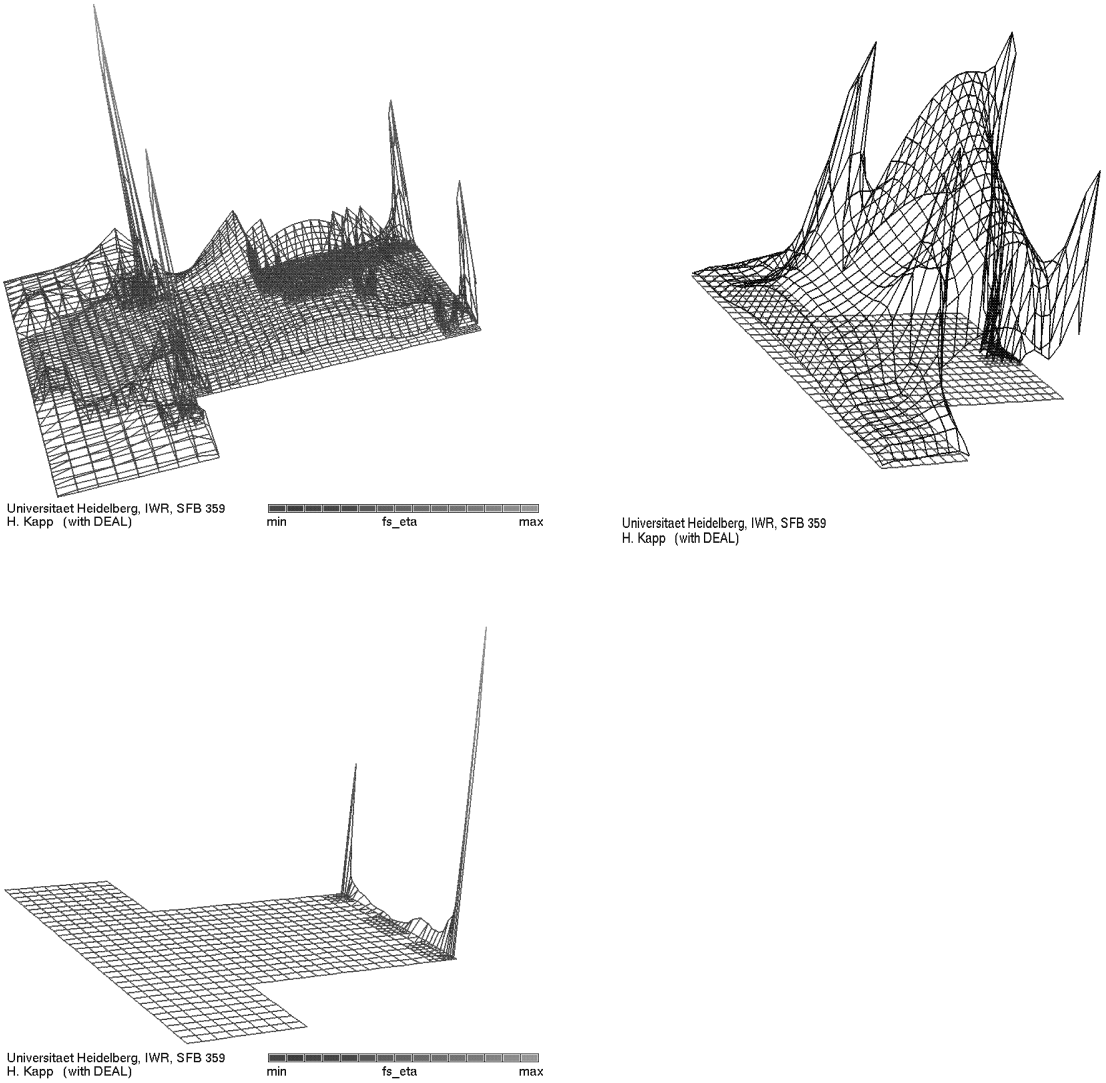


Figure 4.1: Nonlinear test 1 ($\alpha = 0$): Distributions of local error indicators in the energy-error estimator $\eta_E(u_h)$ (left), the energy-error estimator $\eta_E(u_h, \lambda_h)$ (right) and the dual-weighted error estimator $\eta_w(u_h, \lambda_h, q_h)$ (bottom).

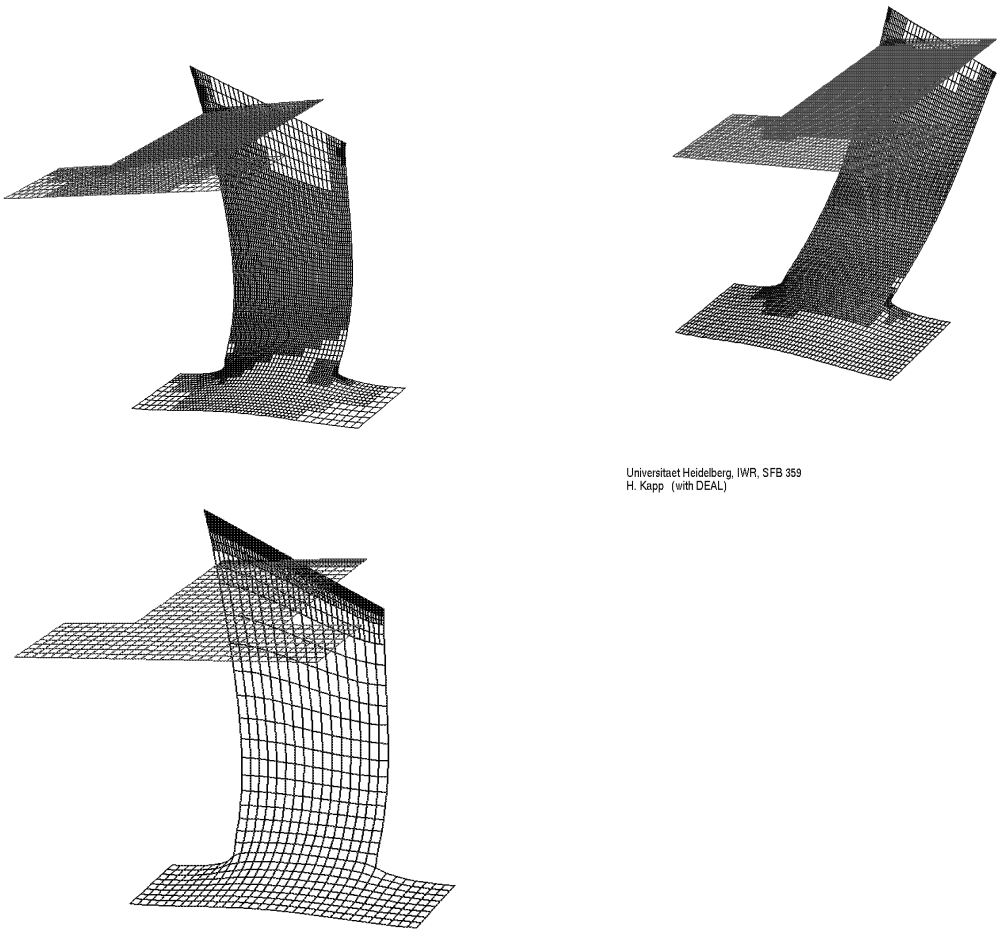


Figure 4.2: Nonlinear test 1 ($\alpha = 0$): Comparison of discrete solutions obtained by the energy-error estimator $\eta_E(u_h)$ (left $N \sim 4800$ cells), the energy-error estimator $\eta_E(u_h, \lambda_h)$ (right $N \sim 5700$ cells) and the dual-weighted error estimator $\eta_w(u_h, \lambda_h, q_h)$ (bottom $N \sim 5000$ cells).

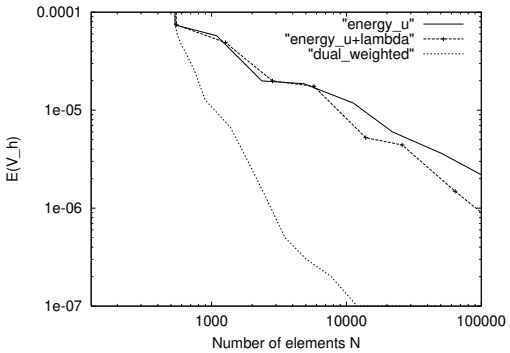


Figure 4.3: Nonlinear test 1 ($\alpha = 0$): Comparison of efficiency of meshes generated by the error estimators $\eta_E(u_h)$ (solid line), $\eta_E(u_h, \lambda_h)$ (crosses) and $\eta_w(u_h, \lambda_h, q_h)$ (dashed line) in log / log scale.

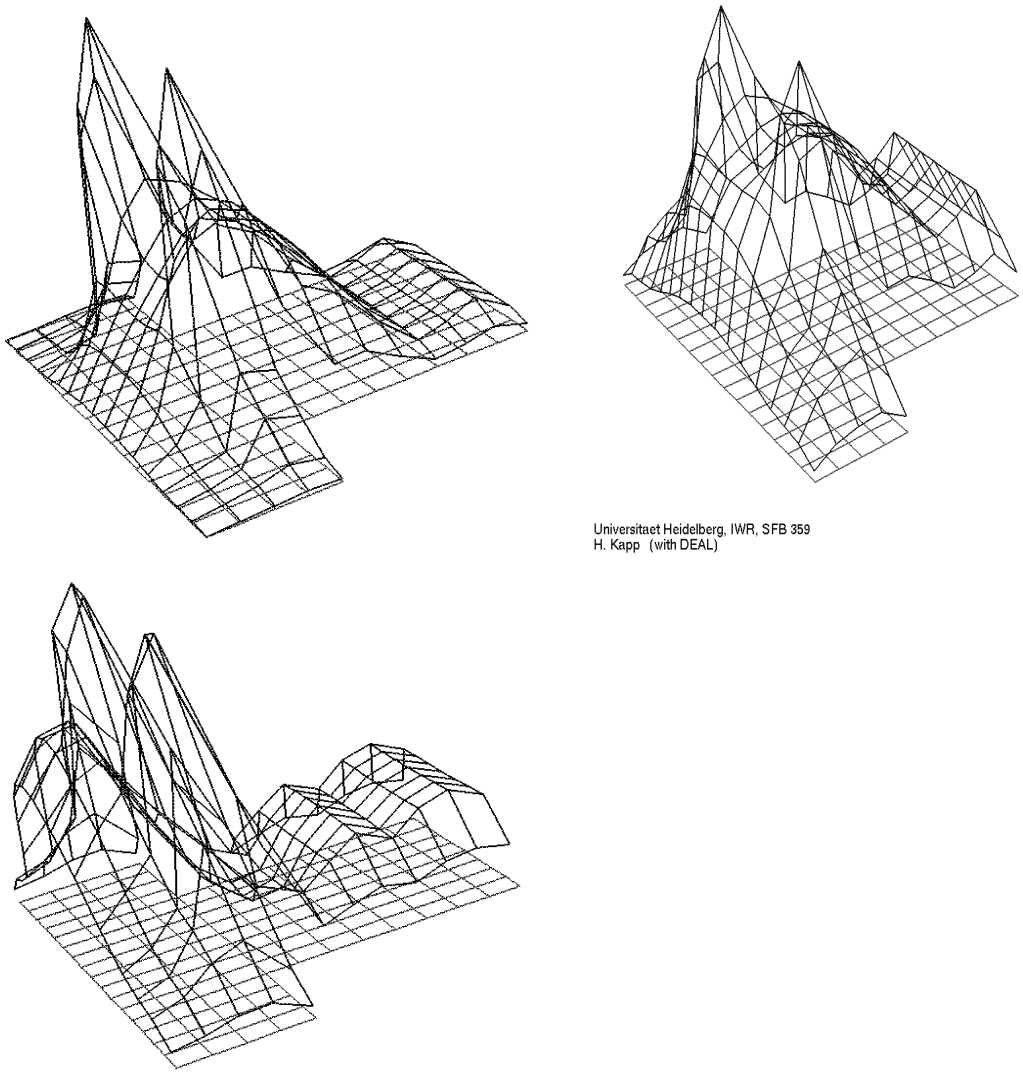


Figure 4.4: Nonlinear test 2: Distributions of local cell indicators in the energy-error estimator $\eta_E(u_h)$ (left), the energy-error estimator $\eta_E(u_h, \lambda_h)$ (right) and the dual-weighted error estimator $\eta_w(u_h, \lambda_h, q_h)$ (bottom).

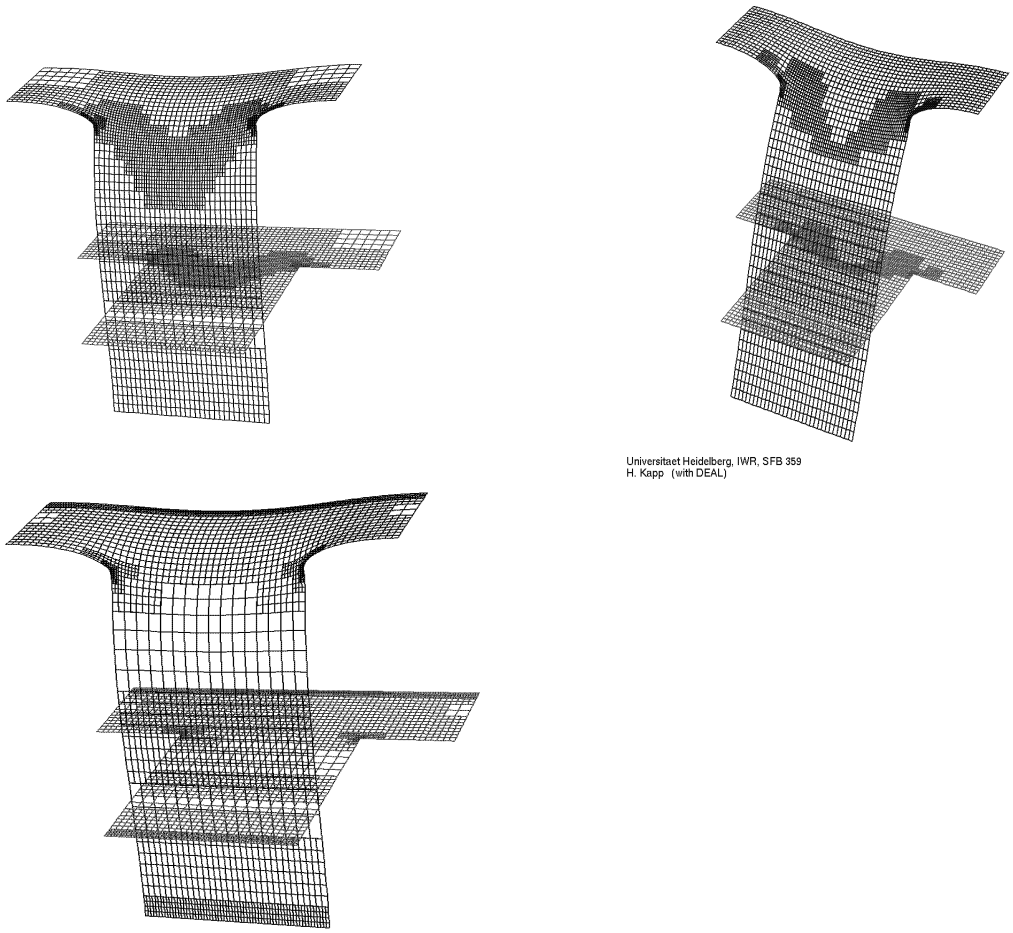


Figure 4.5: Nonlinear test 2: Comparison of discrete solutions obtained by the energy-error estimator $\eta_E(u_h)$ (left, $N \sim 3300$ cells), the energy-error estimator $\eta_E(u_h, \lambda_h)$ (right, $N \sim 3100$ cells) and the dual-weighted error estimator $\eta_w(u_h, \lambda_h, q_h)$ (bottom, $N \sim 3000$ cells).

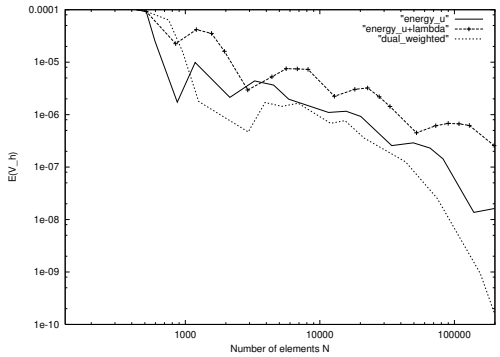


Figure 4.6: Configuration 2: Comparison of efficiency of meshes generated by the error indicators $\eta_E(u_h)$ (solid line), $\eta_E(u_h, \lambda_h)$ (crosses), and $\eta_\omega(u_h, \lambda_h, q_h)$ (dashed line) in log / log scale for 2nd stationary point.