

# A framework for the adaptive finite element solution of large inverse problems. I. Basic techniques

**Wolfgang Bangerth**

Center for Subsurface Modeling, Institute for Computational Engineering and Sciences, University of Texas at Austin; Austin TX 78712, USA

E-mail: [bangerth@ices.utexas.edu](mailto:bangerth@ices.utexas.edu)

**Abstract.** Since problems involving the estimation of distributed coefficients in partial differential equations are numerically very challenging, efficient methods are indispensable. In this first part of a series, we will introduce a framework for the efficient solution of such problems. This comprises the use of adaptive finite element schemes, efficient solvers for the large linear systems arising from discretization, and methods to treat additional information in the form of inequality constraints on the parameter sought. The methods to be developed will be based on an *all-at-once* approach, in which the inverse problem is solved through a Lagrangian formulation. In order to allow for discretizations that are adaptively refined as nonlinear iterations proceed, all algorithms are formulated in a continuous function-space setting. Numerical examples will demonstrate the applicability of the method for problems with several million unknowns and more than 10,000 parameters.

This article also defines the notation and basic methods used in subsequent parts to develop a posteriori error estimates, upon which optimal discretizations will be based.

Submitted to: *Inverse Problems*

## 1. Introduction

Parameter estimation methods are important tools in cases where quantities we would like to know cannot be measured directly, but where only measurements of related quantities, so-called observables, are available. This relation between parameter and observable is usually called the *equation of state*. The goal of parameter estimation is then to recover the unknown quantity from measurements of observables, using the state equation.

If the state equation is a differential equation, such parameter estimation problems are commonly referred to as *inverse problems*. These problems have a vast number of applications. For example, identification of the underground structure (e.g. the elastic properties, density, electric or magnetic permeabilities of the earth) from measurements at the surface or in boreholes, or of the groundwater permeability of a soil from measurements of the hydraulic head fall in this class. Likewise, detecting cracks

in materials, computer tomography, and electrical impedance tomography in medical imaging can be cast as inverse problems.

In particular, we will consider cases where we make *many experiments* to identify the parameters. Here, by an experiment we mean subjecting the physical system to a certain forcing and measuring its response. For example, in computer tomography, a single experiment would be characterized by irradiating a body from a given angle and measuring the transmitted part of the radiation; the multiple experiment situation is characterized by using data from various incidence angles and trying to find a set of parameters that matches *all* the measurements *at the same time* (joint inversion). Likewise, in geophysics, a single experiment would be placing a seismic source somewhere and measuring reflection data at various receiver position; the multiple experiment case is taking into account data for more than one source position. We may also want to include other types of data on the same region, say magnetotelluric or gravimetry data for a joint, multi-physics inversion scenario.

This series of papers is devoted to the development of efficient techniques for the solution of such inverse problems where the state equation is a partial differential equation (henceforth abbreviated as PDE), and the parameters to be determined are one or several distributed functions. It is well-known that the numerical solution of PDE constrained inverse or optimization problems is significantly more challenging than that of a PDE alone, since the optimization procedure is usually iterative and in each iteration may need the numerical solution of a large number of partial differential equations. In some applications, several ten or hundred thousand solutions of linearized PDEs are required to solve the inverse problem, and each PDE may be discretized by up to several hundred thousand unknowns.

It is thus obvious that efficiency of solution is a major concern. In this and following papers, we will devise adaptive finite element techniques that significantly reduce the numerical effort needed to solve such problems. Adaptive methods are now commonly accepted as necessary ingredients of present finite element solvers for partial differential equations. However, they have not yet found their way into the numerical solution of distributed inverse problems, and are only slowly adopted in the solution of PDE constrained optimization, see for example [10, 9, 27]. In this paper, we lay the necessary algorithmic and mathematical foundations for a framework in which adaptive meshing is an integral part. It will be used to derive error estimates in later parts of the series, which we will then in turn use to drive automatic mesh refinement.

The goal of the present work is thus to develop a general framework for the efficient solution of PDE constrained inverse problems. The main ingredients will be:

- *formulation of all algorithms in function spaces*, i.e. before rather than after discretization, since this gives us more flexibility in discretizing as iterations proceed, and resolves all scaling issues once and for all;
- the use of *adaptive finite element techniques* with mesh refinement based on a posteriori error estimates;

- the use of *different meshes for the discretization of different quantities*, for example of state variables and of parameters, in order to reflect their respective properties;
- the use of *Newton-type methods for the outer (nonlinear) iteration*, and of efficient linear solvers for the Newton steps;
- the use of approaches that allow for the *parallelization* of work, yielding subproblems that are equivalent to only forward and adjoint problems;
- the *inclusion of pointwise bounds on the parameters* into the solution process.

Except for the derivation of error estimates, which will be discussed in a subsequent part of this series, we will discuss all these building blocks, and will show that these techniques allow us to solve problems of the size outlined above. It should be stressed that the formulation of algorithms in function spaces is indispensable if we want to use successively finer discretizations, since otherwise quantities computed on one grid would not be comparable to one after mesh refinement.

We envision that the techniques to be presented are used for relatively complex problems. We will thus state them in a general setting, and in this paper present their application to a simple model case for brevity of exposition. However, the algorithms will be designed for efficiency, and will not make use of the relative simplicity of the model problem. More complex examples will then be treated in following parts of this series.

Solving large-scale multiple-experiment inverse problems requires algorithms on several levels, all of which have to be tailored to high efficiency. In this article, we will review the building blocks of a framework for this:

- formulation as a Lagrangian optimization problem with PDE constraints (Section 2); a model problem is given in Section 3;
- outer nonlinear solution by a Gauss-Newton method posed in function spaces (Section 4);
- discretization of each Newton step by finite elements on independent meshes (Section 5);
- linear Schur complement solvers for the resulting discrete systems (Section 6);
- methods to incorporate bound constraints on the parameters sought (Section 7)

In Section 8, we will present numerical examples. We will draw conclusions in the final section.

In subsequent papers of this series, the derivation of mesh refinement criteria based on a posteriori error estimates for energy-type and general output-type quantities will be discussed, and the application of all these techniques to more complex problems will be demonstrated.

## 2. General formulation and notation

Let us begin by introducing some abstract notation, which we will use for the derivation of the entire scheme. This, above all, concerns the set of parameters, state equations,

measurements, regularization, and the introduction of an abstract Lagrangian.

Let us note that some of the formulas below will become cumbersome to read because of the number of indices. To understand their meaning, it is often helpful to imagine we had only a single experiment (for example, only one incidence angle in tomography, or only one source position in seismic imaging). In this case, drop the index  $i$  on first reading, as well as all summations over  $i$ .

### State equations

Let the general setting of the problems we consider be as follows: assume we have a physical situation that is described by a number  $i = 1, \dots, N$  of (independent) partial differential equations posed on a domain  $\Omega \subset \mathbb{R}^d$ :

$$\mathcal{A}^i[q] u^i = f^i \quad \text{in } \Omega, \quad (1)$$

$$\mathcal{B}^i[q] u^i = h^i \quad \text{on } \Gamma_N^i \subset \partial\Omega, \quad (2)$$

$$u^i = g^i \quad \text{on } \Gamma_D^i = \partial\Omega \setminus \Gamma_N^i, \quad (3)$$

where  $\mathcal{A}^i[q]$  are partial differential operators that depend on a set of a priori unknown distributed coefficients  $q = q(\mathbf{x}) \in \mathcal{Q}$ ,  $\mathbf{x} \in \Omega$ , and  $\mathcal{B}^i[q]$  are boundary operators that may also depend on the coefficients.  $f^i, h^i$  and  $g^i$  are right hand sides and boundary values, respectively, of which we assume that they are known, and that do not depend on the coefficients  $q$ . The functions  $u^i$  are the solutions of the partial differential equations, and are assumed to be from spaces  $V_g^i = \{\varphi \in V^i : \varphi|_{\Gamma_D^i} = g^i\}$ . These solutions can be vector valued, but we assume that solutions  $u^i, u^j$  of different equations  $i \neq j$  do not couple except for their dependence on the same set of parameters  $q$ . We also assume that the solutions to each of the differential equations is unique for every given set of parameters  $q$  in a subset  $\mathcal{Q}_{ad} \subset \mathcal{Q}$  of physically meaningful values, for example  $\mathcal{Q}_{ad} = \{q \in L^\infty(\Omega) : q_0 \leq q(\mathbf{x}) \leq q_1\}$ .

Typical cases we have in mind would be a Laplace-type equation when we are considering electrical impedance tomography or gravimetry inversion, or Helmholtz or wave equations in case we are looking at inverting seismic or magnetotelluric data. The set of parameters  $q$  would, in these cases, be electrical conductivities, densities, or elasticity coefficients. If we have done several measurements for the identification of the same parameters, but with different source terms, at different frequencies, or in an otherwise different setting, then we will identify each of these experiments with one index  $i = 1, \dots, N$ . The operators  $\mathcal{A}^i$  may be the same for several experiments, but if we use different physical effects (for example gravimetry and seismic data) to identify the same parameters – the so-called *multi-physics inversion* case – then they will be different.

The formulation above may easily be extended also to the case of time-dependent problems, but we omit this for brevity. Likewise, the case that the parameters are a finite number of scalar values is included as a simple special case, but will not be the main focus of our work.

For treatment in a Lagrangian setting in function spaces as well as for discretization by finite elements, it is necessary to formulate the state equations (1)–(3) in a variational form. For this we assume that the solutions  $u^i \in V_g^i$  are solutions of the following variational equalities:

$$A^i(q; u^i)(\varphi^i) = 0 \quad \forall \varphi^i \in V_0^i, \quad (4)$$

where  $V_0^i = \{\varphi^i \in V^i : \varphi^i|_{\Gamma_D^i} = 0\}$ . The semilinear form  $A^i : \mathcal{Q} \times V_g^i \times V_0^i \rightarrow \mathbb{R}$  may be nonlinear in its first set of arguments, but is linear in the test function, and includes the actions of the domain and boundary operators  $\mathcal{A}^i$  and  $\mathcal{B}^i$ , as well as the inhomogeneous terms. We assume that the  $A^i$  are differentiable. Again, extensions to more complicated problems are possible, but are omitted for brevity.

Below, we will introduce a model problem for the Laplace equation. In this particular case,  $\mathcal{A}^i[q]u^i = -\nabla \cdot (q\nabla u^i)$ ,  $\mathcal{B}^i[q]u^i = q\partial_n u^i$ , and

$$A(q; u^i)(\varphi^i) = (q\nabla u^i, \nabla \varphi^i)_\Omega - (f, \varphi^i)_\Omega - (g, \varphi^i)_{\Gamma_N}.$$

### Measurements

In order to determine the unknown quantities  $q$ , we have measurements of (parts of) each of the state variables  $u^i$ , or of derived quantities. For example, we might have measurements of the value at certain points, averages on subdomains, or gradients. Let us denote the space of measurements of the  $i$ th state variable by  $\mathcal{Z}^i$ , and let  $M^i : V_g^i \rightarrow \mathcal{Z}^i$  be the measurement operator, i.e. the operator that extracts from the state solution  $u^i$  the measurement values.

Reconstruction of the coefficients  $q$  will then be accomplished by finding those coefficients, for which the expected measurements  $M^i u^i$  match the actual measurements  $z^i \in \mathcal{Z}^i$  best. We will measure this comparison using a convex, differentiable functional  $m : \mathcal{Z}^i \rightarrow \mathbb{R}^+$  with  $m(0) = 0$ . In many cases,  $m$  will simply be an  $L^2$  norm on  $\mathcal{Z}^i$ , but more general functionals are allowed, for example to suppress the effects of non-Gaussian noise.

Examples of common measurement situations for scalar valued solutions are:

- *$L^2$  measurements of values:* If measurements on a set  $\Sigma \subset \Omega$  are available, then  $M^i$  is the embedding operator from  $V_g^i$  into  $\mathcal{Z}^i = L^2(\Sigma)$ , and the quantity to be minimized is

$$m^i(M^i u^i - z^i) = \frac{1}{2} \|u^i - z^i\|_{L^2(\Sigma)}^2.$$

As special cases,  $\Sigma$  can be the whole domain  $\Omega$ , or its boundary  $\partial\Omega$ . The case of distributed measurements occurs in situations where a measuring device can be moved around to every point of  $\Sigma$ , for example a laser scanning a membrane, or imaging radiation on a photographic film.

- *Point measurements:* If we have  $S$  measurements of  $u(\mathbf{x})$  at positions  $x_s \in \Omega$ ,  $s = 1, \dots, S$ , then  $\mathcal{Z}^i = \mathbb{R}^S$ , and  $(M^i u^i)_s = u(\mathbf{x}_s)$ . If we take again a quadratic norm

on  $\mathcal{Z}^i$ , then for example

$$m^i(M^i u^i - z^i) = \frac{1}{2} \sum_{s=1}^S |u^i(\mathbf{x}_s) - z_s^i|^2$$

is a possible choice. The case of point measurements is frequent in applications where a small number of stationary measurement devices is used, for example seismometers in seismic data assimilation.

- *Gradient measurements:* If we only have measurements of the gradient, then  $M^i = \nabla$ ,  $\mathcal{Z}^i = (L^2)^d$ , and we can for example choose

$$m^i(M^i u^i - z^i) = \frac{1}{2} \|\nabla u^i - z^i\|_{L^2(\Omega)}^2.$$

An example for this is measuring the gravitational pull of a body, which is the gradient of the gravimetric potential.

Other choices are of course possible, and are usually dictated by the type of available measurements.

We will in general assume that the  $M^i$  are linear, but there are applications where this is not the case. For example, in some applications only statistical correlations of  $u^i$  are known, or a power spectrum. Extending the algorithms below to nonlinear  $M^i$  is straightforward, but we omit this for brevity.

### *Regularization*

Since inverse problems are often ill-posed, regularization is needed to suppress unwanted features in solutions  $q$ . In this work, we include it by using a Tikhonov regularization term involving a convex differentiable regularization functional  $r : \mathcal{Q} \rightarrow \mathbb{R}^+$ , see for example [26].  $r$  is usually chosen to be the square of a quadratic seminorm on  $\mathcal{Q}$ , for example  $r(q) = \frac{1}{2} \|\nabla^t(q - \bar{q})\|_{L^2(\Omega)}^2$  with an a priori guess  $\bar{q}$  and some  $t \in \mathbb{N}^+$ . Other popular choices are smoothed versions of bounded variation seminorms [20, 17, 19]. As above, the type of regularization is usually dictated by the application and insight into physical and unphysical features of solutions. It may also be chosen to give the misfit functional a proper meaning, for example in the point value case where we need a continuous solution. An additional regularization functional may be used if different distributed parameters, e.g. the two Lamé constants, are assumed to share structural properties such as locations of discontinuities (see, e.g., [23]).

### *Characterization of solutions*

The goal of the inverse problem is to find that set of physical parameters  $q \in \mathcal{Q}_{ad}$  for which the observations  $M^i u^i$  of the resulting state variables would match the actual observations  $z^i$  best. We formulate this as the following constrained minimization problem over  $u^i \in V_g^i, q \in \mathcal{Q}_{ad}$ :

$$\text{minimize} \quad J(\{u^i\}, q) = \sum_{i=1}^N \sigma^i m^i(M^i u^i - z^i) + \beta r(q) \quad (5)$$

$$\text{such that } A^i(q; u^i)(\varphi^i) = 0 \quad \forall \varphi^i \in V_0^i, \quad 1 \leq i \leq N.$$

Here,  $\sigma^i > 0$  are factors weighting the relative importance of individual measurements, and  $\beta > 0$  is a regularization parameter. As the choice of these constants is a topic of its own, we assume their values as given within the scope of this work.

In order to characterize solutions to (5), let us subsume the individual solutions  $u^i$  to a vector  $\mathbf{u}$ , and likewise denote the spaces  $V_g^i, V_0^i$  by the components of spaces  $\mathbf{V}_g, \mathbf{V}_0$ . Furthermore, we introduce a set of Lagrange multipliers  $\boldsymbol{\lambda} \in \mathbf{V}_0$ , and denote the joint set of variables by  $x = \{\mathbf{u}, \boldsymbol{\lambda}, q\} \in \mathcal{X}_g = \mathbf{V}_g \times \mathbf{V}_0 \times \mathcal{Q}$ .

We assume that solutions of problem (5) can be characterized as stationary points of the following Lagrangian  $L : \mathcal{X}_g \rightarrow \mathbb{R}$ , which couples the functional  $J : \mathbf{V}_g \times \mathcal{Q} \rightarrow \mathbb{R}^+$  defined above to the state equation constraints through Lagrange multipliers  $\lambda^i \in V_0^i$ :

$$L(x) = J(\mathbf{u}, q) + \sum_{i=1}^N A^i(q; u^i)(\lambda^i). \quad (6)$$

Solutions are then stationary points of this Lagrangian, i.e. the optimality conditions read in abstract form

$$L_x(x)(y) = 0 \quad \forall y \in \mathcal{X}_0, \quad (7)$$

where the semilinear form  $L_x : \mathcal{X}_g \times \mathcal{X}_0 \rightarrow \mathbb{R}$  is the differential of the Lagrangian  $L$ , and  $\mathcal{X}_0 = \mathbf{V}_0 \times \mathbf{V}_0 \times \mathcal{Q}$ . Indicating derivatives of functionals with respect to their arguments by subscripts, we can expand (7) to yield the following set of nonlinear equations:

$$L_{\lambda^i}(x; \varphi^i) \equiv A^i(q; u^i)(\varphi^i) = 0 \quad \forall \varphi^i \in V_0^i, \quad (8)$$

$$L_{u^i}(x; \psi^i) \equiv \sigma^i m_u^i (M^i u^i - z^i)(\psi^i) + A_{u^i}^i(q; u^i)(\psi^i, \lambda^i) = 0 \quad \forall \psi^i \in V_0^i, \quad (9)$$

$$L_q(x; \chi) \equiv \beta r_q(q)(\chi) + \sum_{i=1}^N A_q^i(q; u^i)(\chi, \lambda^i) = 0 \quad \forall \chi \in \partial \mathcal{Q}. \quad (10)$$

The first set of equations denotes the state equations for  $i = 1, \dots, N$ ; the second the adjoint equations defining the Lagrange multipliers  $\lambda^i$ ; finally, the third is the control equation holding for all functions from the tangent space  $\partial \mathcal{Q}$  to  $\mathcal{Q}$  at the solution  $q$ . Note that we have included the (linear) constraint of Dirichlet boundary values into the function space where we look for solutions, rather than the Lagrange functional. We do so, since we can simply set an initial iterate in the solution of this nonlinear system to these boundary values. Further updates to this iterate are then sought from the vector space with zero boundary conditions.

The question whether the solution of the original constrained optimization problem (5) can indeed be characterized by the stationarity of a Lagrangian, (7), depends crucially on the exact form of the various functionals involved, and the function spaces  $\mathbf{V}_g$  and  $\mathcal{Q}$  and cannot be answered in general. It holds for a large number of practical applications (see, for example, [8]), and we assume that this augmentability holds also for the cases discussed in this work.

### 3. A model problem

As a simple model problem which we will use to give the abstract results of this work a more concrete form, we will consider the following situation: assume we intend to identify the coefficient  $q$  in the (single) elliptic PDE

$$-\nabla \cdot (q\nabla u) = f \quad \text{in } \Omega, \quad u = g \quad \text{on } \partial\Omega, \quad (11)$$

and that measurements are the values of the solution  $u$  everywhere in  $\Omega$ , i.e. we choose  $m(Mu - z) = m(u - z) = \frac{1}{2}\|u - z\|_{L^2(\Omega)}^2$ . This situation can be considered as a mathematical description of a membrane with variable stiffness  $q(\mathbf{x})$  which we try to identify by subjecting it to a known force  $f$  and clamping it at the boundary with boundary values  $g$ . This results in displacements of which we obtain measurements  $z$  everywhere. For simplicity, we have here chosen to consider only one experiment, i.e.  $N = 1$ .

For this situation,  $V_g = \{u \in H^1 : u|_{\partial\Omega} = g\}$ ,  $\mathcal{Q} \subset L^\infty$ . Choosing  $\sigma = 1$ , the Lagrange functional then has the form

$$L(x) = m(u - z) + \beta r(q) + (q\nabla u, \nabla \lambda) - (f, \lambda).$$

With this, the optimality conditions (8)–(10) read in weak form

$$(q\nabla u, \nabla \varphi) = (f, \varphi), \quad (12)$$

$$(q\nabla \psi, \nabla \lambda) = -(u - z, \psi), \quad (13)$$

$$\beta r_q(q; \chi) = -(\nabla u, \nabla \lambda), \quad (14)$$

and have to hold for all test functions  $\{\varphi, \psi, \chi\} \in V_0^1 \times V_0^1 \times \mathcal{Q} = H_0^1 \times H_0^1 \times \mathcal{Q}$ . The first of these is the state equation, the second the adjoint one.

### 4. Nonlinear solvers

The stationarity conditions (7) form a set of nonlinear partial differential equations that has to be solved iteratively, for example using Newton's method, or a variant thereof. In this section, we will formulate the Gauss-Newton method in function spaces. The discretization of each step by adaptive finite elements will then be presented in the next section, followed by a discussion of solvers for the resulting linear systems.

Since there is no need to compute the initial nonlinear steps on a very fine grid when we are still far away from the solution, we will want to use successively finer meshes as we approach the solution. In order to make quantities computed on different meshes comparable, all of the following algorithms will be formulated in a continuous setting, and only then be discretized. This also answers once and for all questions about the correct scaling of weighting matrices in misfit and regularization functionals, as discussed for example in [2], even if we choose locally refined grids, as they will appear naturally upon discretization. In this section, we indicate a Gauss-Newton procedure, i.e. determination of search direction and step length, in infinite dimensional spaces, and discuss in the next section its discretization by a finite element scheme. It is

worth noting that the number of possible methods for solving such problems is vast [24, 1, 31, 28, 22, 29, 16].

However, we believe that the Gauss-Newton method is particularly suited since it allows for scalable algorithms even with large numbers of experiments, and large numbers of degrees of freedom both in the discretization of the state equations as well as of the parameter. Comparing this method to a pure Newton method, it allows for the use of more efficient linear solvers for the discretized problems, see Section 6. In addition, the Gauss-Newton method has been shown to have better stability properties for parameter estimation problems than the Newton method, see [15, 14].

### Search directions

Given a present iterate  $x_k = \{\mathbf{u}_k, \boldsymbol{\lambda}_k, q_k\} \in \mathcal{X}$ , the first task of any method is to compute a search direction  $\delta x_k = \{\delta \mathbf{u}_k, \delta \boldsymbol{\lambda}_k, \delta q_k\} \in \mathcal{X}_{\delta g}$ , in which we seek the next iterate  $x_{k+1}$ . The Dirichlet boundary values  $\delta g$  of this update are chosen as  $\delta u_k^i|_{\Gamma_D} = g^i - u_k^i|_{\Gamma_D}$ ,  $\delta \lambda_k^i|_{\Gamma_D} = 0$ , bringing us to the exact boundary values if we take a full step.

In order to determine the search direction, we use the Gauss-Newton method. In the present case, we determine updates  $\{\delta \mathbf{u}_k, \delta q_k\}$  by minimizing the following quadratic approximation to  $J(\cdot, \cdot)$  with linearized constraints:

$$\begin{aligned} \min_{\delta \mathbf{u}_k, \delta q_k} & J(\mathbf{u}_k, q_k) + J_u(\mathbf{u}_k, q_k)(\delta \mathbf{u}_k) + J_q(\mathbf{u}_k, q_k)(\delta q_k) \\ & + \frac{1}{2} J_{uu}(\mathbf{u}_k, q_k)(\delta \mathbf{u}_k, \delta \mathbf{u}_k) + \frac{1}{2} J_{qq}(\mathbf{u}_k, q_k)(\delta q_k, \delta q_k) \\ \text{s.t.} & A^i(q_k; u_k^i)(\varphi^i) + A_u^i(q_k; u_k^i)(\delta u_k^i, \varphi^i) + A_q^i(q_k; u_k^i)(\delta q_k, \varphi^i) = 0, \end{aligned} \quad (15)$$

where the linearized constraints are understood to hold for  $1 \leq i \leq N$  and for all test functions  $\varphi^i \in V_0^i$ . The solution of this problem provides us with updates  $\delta \mathbf{u}_k, \delta q_k$  for the state variables and the parameters. The updates for the Lagrange multiplier  $\delta \boldsymbol{\lambda}_k$  are not determined by the Gauss-Newton step at first. However, the solution of (15) is characterized by coupling the linearized constraints to the quadratic function. It turns out that we can get updates  $\delta \boldsymbol{\lambda}_k$  for the original problem, by using  $\boldsymbol{\lambda}_k + \delta \boldsymbol{\lambda}_k$  as Lagrange multiplier for the constraint of the Gauss-Newton step (15). Bringing the terms with  $\boldsymbol{\lambda}_k$  to the right hand side, the updates are then characterized by the following system of linear equations:

$$\begin{aligned} \sigma^i m_{uu}^i (M^i u_k^i - z^i)(\delta u_k^i, \varphi^i) + A_u^i(q_k; u_k^i)(\varphi^i, \delta \lambda_k^i) &= -L_{u^i}(x_k)(\varphi^i) \\ A_u^i(q_k; u_k^i)(\delta u_k^i, \psi^i) + A_q^i(q_k; u_k^i)(\delta q_k^i, \psi^i) &= -L_{\lambda^i}(x_k)(\psi^i) \\ \sum_i A_q^i(q_k; u_k^i)(\chi, \delta \lambda_k^i) + \beta r_{qq}(q_k)(\delta q_k, \chi) &= -L_q(x_k)(\chi), \end{aligned} \quad (16)$$

for all test functions  $\{\varphi^i, \psi^i, \chi\}$ . The right hand side of these equations are the negative gradient of the original Lagrangian, given already in the optimality condition (8)–(10).

The matrix structure of this system will be given in the next section, showing a number of nice properties. Note that the equations determining the updates for the  $i$ th experiment decouple from all other experiments, except for the last equation.

This will allow us to solve them mostly separated, and in particular it allows for simple parallelization by placing the description of different experiments onto different machines. Furthermore, the first and second equations can be solved sequentially.

In order to illustrate these equations, we state their form for the model problem of Section 3. In this case, the above system reads

$$\begin{aligned} (\delta u_k, \varphi) + (\nabla \delta \lambda_k, q_k \nabla \varphi) &= -L_u(x_k)(\varphi), \\ (\nabla \psi, q_k \nabla \delta u_k) + (\nabla \psi, \delta q_k \nabla u_k) &= -L_\lambda(x_k)(\psi), \\ (\nabla \delta \lambda_k, \chi \nabla u_k) + \beta r_{qq}(q_k)(\delta q_k, \chi) &= -L_q(x_k)(\chi), \end{aligned}$$

with the right hand sides being the gradient of the Lagrangian given in Section 3.

In general, this continuous Gauss-Newton direction will not be computable analytically. We will therefore approximate it by a finite element function  $\delta x_{k,h}$ , as discussed in the next section.

As a final remark, let us note that the pure Newton's method would read

$$L_{xx}(x_k)(\delta x_k, y) = -L_x(x_k)(y) \quad \forall y \in \mathcal{X}_0, \quad (17)$$

where  $L_{xx}(x_k)(\cdot, \cdot)$  denotes the bilinear form of second variational derivatives of the Lagrangian  $L$  at position  $x_k$ . The Gauss-Newton method can alternatively be obtained from this by simply dropping all terms that are proportional to the Lagrange multiplier  $\lambda_k$ . The reasoning for this is that we expect the Lagrange multiplier to be small for small-noise problems, a fact that is warranted by looking at (9): the Lagrange multiplier has to satisfy an equation with a term proportional to  $Mu - z$  as right hand side; thus, assuming stability of the (linear) adjoint operator, it will be small if the residual is small at the solution.

### Step lengths

Once we have a search direction  $\delta x_k$ , we have to decide how far to go in this direction starting at  $x_k$  to obtain the next iterate  $x_{k+1} = x_k + \alpha_k \delta x_k$ . In constrained optimization, a merit function including the minimization functional  $J(\cdot)$  as well as the violation of the constraints is usually used. One particular problem here is the infinite dimensional nature of the state equation constraint, with the residual of the state equation being in the dual space,  $V'_0$ , of  $V_0$  (which, for the model problem, is  $H^{-1}$ ). This places some restrictions on the types of merit functions that can practically be used.

In order to avoid these problems, we simply use the norm of the residual of the optimality condition (7) on the dual space of  $\mathcal{X}_0$  as merit function:

$$p(\alpha) = \frac{1}{2} \|L_x(x_k + \alpha \delta x_k)(\cdot)\|_{\mathcal{X}'_0}^2 \equiv \frac{1}{2} \sup_{y \in \mathcal{X}_0} \frac{[L_x(x_k + \alpha \delta x_k)(y)]^2}{\|y\|_{\mathcal{X}_0}^2}.$$

We will show in the next section that we can give a simple-to-compute lower bound for this norm using the discretization we already employ for the computation of the search direction.

The following lemma shows that this merit function is actually useful:

**Lemma 1** *The merit function  $p(\alpha)$  is valid, i.e. Newton directions are directions of descent,  $p'(0) < 0$ . Furthermore, if  $x_k$  is a solution of the parameter estimation problem, then  $p(0) = 0$ . Finally, in the vicinity of the solution, full steps are taken, i.e.  $\alpha = 1$  minimizes  $p$ .*

*Proof.* We prove the lemma for the special case of only one experiment ( $N = 1$ ) and that  $\mathcal{X} = H_0^1 \times H_0^1 \times L^2$ , i.e. the situation of the model example. In this case, there is a representation  $g_u(x_k + \alpha\delta x_k) = L_u(x_k + \alpha\delta x_k)(\cdot) \in H^{-1}$ ,  $g_\lambda(x_k + \alpha\delta x_k) = L_\lambda(x_k + \alpha\delta x_k)(\cdot) \in H^{-1}$ , and  $g_q(x_k + \alpha\delta x_k) = L_q(x_k + \alpha\delta x_k)(\cdot) \in L^2$ . The dual norm of  $L_x$  can then be written as

$$\|L_x\|_{\mathcal{X}'_0}^2 = \langle g_u, (-\Delta)^{-1}g_u \rangle + \langle g_\lambda, (-\Delta)^{-1}g_\lambda \rangle + (g_q, g_q),$$

where  $(-\Delta)^{-1} : H^{-1} \rightarrow H_0^1$ , and  $\langle \cdot, \cdot \rangle$  indicates the duality pairing between  $H^{-1}$  and  $H_0^1$ . Then,

$$p'(0) = \langle g_{uu}(\delta u_k), (-\Delta)^{-1}g_u \rangle + \langle g_{\lambda\lambda}(\delta \lambda_k), (-\Delta)^{-1}g_\lambda \rangle + (g_{qq}(\delta q_k), g_q),$$

where  $g_{ux}(\delta x_k)$  is the derivative of  $g_u$  in direction  $\delta x_k$ , i.e. the functional of second derivatives of  $L$ . However, by definition of the Newton direction, (17), this is equal to the negative gradient, i.e.

$$p'(0) = -\|L_x(x_k)\|_{\mathcal{X}'_0}^2 = -2p(0) < 0.$$

Thus, Newton directions are directions of descent for this merit function.

The second part of the lemma is obvious by noting the optimality condition (7). The last part follows by noting that near the solution the Lagrangian (and thus the function  $p(\alpha)$ ) is well approximated by a quadratic function if the various functionals involved in the Lagrangian are sufficiently smooth.

Although the lemma technically does not cover the case of Gauss-Newton search directions, all numerical experiments indicate that the results also hold for this case, at least in the case of small noise.

## 5. Discretization

In order to compute finite-dimensional approximations to the solution of the parameter estimation problem, we have to discretize both the state and adjoint variables, as well as the parameters. Here, we will choose finite element schemes to do so, and we will use subsequently finer meshes on subsequent Gauss-Newton iterations, to make the initial iterations cheaper. In each iteration, we define finite element spaces  $\mathcal{X}_h \subset \mathcal{X}$  over triangulations in the usual way. For these grids and spaces, we assume the following requirements:

- *Nesting:* The mesh  $\mathbb{T}_k^i$  to be used in the discretization of the  $i$ th state equation in step  $k$  must be obtainable from  $\mathbb{T}_{k-1}^i$  by hierarchic coarsening and refinement. This greatly simplifies operations like evaluation of the right hand side of the Newton direction equation,  $L_x(x_k)(\cdot)$  for all discrete test functions, but also the update operation  $x_{k+1} = x_k + \alpha_k \delta x_{k,h}$ .

- *State vs. parameter meshes:* Independently of the meshes  $\mathbb{T}_k^i$  used for the discretization state equations, a mesh  $\mathbb{T}_k^q$  will be used to discretize the parameters  $q$  on step  $k$ . This reflects that the regions of missing regularity of parameters and state variables need not necessarily coincide. We may also use different discretization spaces for parameters and state/adjoint variables, for example spaces of discontinuous functions for quantities like density or elasticity coefficients. We will require that each of the ‘state meshes’  $\mathbb{T}_k^i$  can be obtained by hierarchical refinement from the ‘parameter mesh’  $\mathbb{T}_k^q$ .

Although obvious, both the choice of independent grids for state and parameter meshes, as well as the adaptive choice of subsequently finer meshes, have apparently not been used in the literature to the author’s best knowledge. Both techniques offer the prospect of greatly reducing the amount of numerical work. We will see that with the requirements on the meshes above, the additional work associated with using different meshes is very small.

Here, the basic idea in choosing subsequently finer meshes is that as we are still far away from the solution, we can work on coarser meshes for approximating the Newton steps, and only go over to finer ones as we approach the solution. Most of the initial steps will therefore contribute only negligibly to the total cost of the solution process, both because they are relatively small problems, but also because a smaller size reduces the ill-posedness, making the iterative solution of the linear problems faster. In addition, choosing coarse meshes for the first iterates also stabilizes the problem on the initial steps, when we are still far away from the solution, effectively acting as an additional regularization.

Choosing different ‘state’ and ‘parameter meshes’ also allows for problems where the parameters do not require high resolution, or only in certain areas of the domain, while the state equation requires this. A typical problem would be high-frequency tomography, where the coefficient sought might be a function that is constant in large parts of the domain, while the high-frequency oscillations of state and adjoint variables require a fine grid everywhere where these variables have a significant value. Again, being able to reduce the number of parameters compared to the number of state variables, and localizing them at places where enough information is available for their reconstruction, reduces the numerical work and improves conditioning of the subproblems.

In the next few paragraphs, we will briefly describe the process of discretizing the equations for the search directions and the choice of the step length. We will then give a brief note on the criteria for generating the meshes on which we discretize.

### *Search directions*

By choosing a finite dimensional subspace  $\mathcal{X}_h = V_h \times V_h \times \mathcal{Q}_h \subset \mathcal{X}$ , we obtain a discrete counterpart for equation (16) describing the Gauss-Newton search direction:

$$\begin{aligned} \sigma^i m_{uu}^i (M^i u_k^i - z^i) (\delta u_{k,h}^i, \varphi_h^i) + A_u^i(q_k; u_k^i) (\varphi_h^i, \delta \lambda_{k,h}^i) &= -L_{u^i}(x_{k,h})(\varphi_h^i), \\ A_u^i(q_k; u_k^i) (\delta u_{k,h}^i, \psi_h^i) + A_q^i(q_k; u_k^i) (\delta q_{k,h}^i, \psi_h^i) &= -L_{\lambda^i}(x_{k,h})(\psi_h^i) \end{aligned} \quad (18)$$

$$\sum_i A_q^i(q_k; u_k^i)(\chi_h, \delta\lambda_{k,h}^i) + \beta r_{qq}(q_k)(\delta q_{k,h}, \chi_h) = -L_q(x_{k,h})(\chi_h),$$

Note that if the discretization we use does not involve stabilization terms, then discretization and differentiation commute, so (18) can either be viewed as the discretization of the continuous Gauss-Newton step (16), or as one step for a discretized version (on a fixed mesh) of the Gauss-Newton step problem (15). We prefer the first view, since it allows more intuitively to change the meshes between iterations.

Choosing a basis for the space  $\mathcal{X}_h$ , we can write (18) in matrix form as follows:

$$\begin{pmatrix} \mathbf{M} & \mathbf{A}^T & \mathbf{0} \\ \mathbf{A} & \mathbf{0} & \mathbf{C} \\ \mathbf{0} & \mathbf{C}^T & \beta\mathbf{R} \end{pmatrix} \begin{pmatrix} \delta\mathbf{u}_{k,h} \\ \delta\boldsymbol{\lambda}_{k,h} \\ \delta q_{k,h} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_u \\ \mathbf{F}_\lambda \\ \mathbf{F}_q \end{pmatrix}. \quad (19)$$

Since the individual state equations and variables do not couple across experiments,  $\mathbf{M} = \text{diag}(\mathbf{M}^i)$  and  $\mathbf{A} = \text{diag}(\mathbf{A}^i)$  are block diagonal matrices, with the diagonal blocks stemming from second derivatives of the misfit functionals, and of the tangential operators of the state equations, respectively. They are equal to

$$(\mathbf{M}^i)_{kl} = m_{uu}^i(M^i u_k^i - z^i)(\varphi_k^i, \varphi_l^i), \quad (\mathbf{A}^i)_{kl} = A_u^i(x_k)(\varphi_l^i, \varphi_k^i),$$

where  $\varphi_l^i$  are test functions for the discretization of the  $i$ th state equation. Likewise,  $\mathbf{C} = [\mathbf{C}^1, \dots, \mathbf{C}^N]$  is defined by

$$(\mathbf{C}^i)_{kl} = A_q^i(x_k)(\chi_l^q, \varphi_k^i),$$

with  $\chi_l^q$  being discrete test functions for the parameters  $q$ , and  $(\mathbf{R})_{kl} = r_{qq}(q_k)(\chi_k^q, \chi_l^q)$ .

The evaluation of  $\mathbf{C}^i$  may be difficult since it involves shape functions from different meshes and finite element spaces. However, since we have required that  $\mathbb{T}_k^i$  can be obtained from  $\mathbb{T}_k^q$  by hierarchical refinement, we can represent each shape function  $\chi_k^q$  on the parameter mesh as a sum over respective shape functions  $\chi_s^i$  on each of the state meshes:  $\chi_k^q = \sum_s \mathbf{X}_{ks}^i \chi_s^i$ . Thus,  $\mathbf{C}^i = \hat{\mathbf{C}}^i \mathbf{X}^i$ , with  $\hat{\mathbf{C}}^i$  built with shape functions from only one grid. The matrix  $\mathbf{X}^i$  is fairly simple to generate because of the hierarchical structure of the meshes. In general, it is worthwhile to store  $\hat{\mathbf{C}}^i$  and  $\mathbf{X}^i$  separately, rather than forming  $\mathbf{C}^i$  explicitly.

Note that if we had wanted to use the full Newton search direction instead of the Gauss-Newton one, then the top right and bottom left blocks of the matrix in (19) would also be nonzero. In addition, if the state equations were nonlinear in  $u^i$  or  $q^i$ , additional terms would have to be added to the  $\mathbf{M}$  and  $\mathbf{R}$  blocks.

### Step lengths

Since step length selection is only a tool for seeking the exact solution, we may be content with approximating the merit function  $p(\alpha)$  introduced in Section 4. To this end, we use a lower bound  $\underline{p}(\alpha)$  for  $p(\alpha)$ , by restricting the set of possible test functions to the discrete space  $\mathcal{X}_h$  which we are already using for the discretization of the search direction:

$$\underline{p}(\alpha) = \frac{1}{2} \sup_{y \in \mathcal{X}_h} \frac{[L_x(x_k + \alpha \delta x_k)(y)]^2}{\|y\|_{\mathcal{X}_0}^2} \leq \frac{1}{2} \|L_x(x_k + \alpha \delta x_k)(\cdot)\|_{\mathcal{X}_0'}^2 = p(\alpha).$$

By selecting a basis of  $\mathcal{X}_h$ ,  $\underline{p}(\alpha)$  can be computed by linear algebra. For example, for the single experiment case ( $N = 1$ ) and if  $\mathcal{X} = H_0^1 \times H_0^1 \times L^2$ , we have that

$$\underline{p}(\alpha) = \frac{1}{2} [\langle g_u(\alpha), Y_1^{-1} g_u(\alpha) \rangle + \langle g_\lambda(\alpha), Y_1^{-1} g_\lambda(\alpha) \rangle + \langle g_q(\alpha), Y_0^{-1} g_q(\alpha) \rangle],$$

where  $(Y_0)_{kl} = (\chi_k, \chi_l)$ ,  $(Y_1)_{kl} = (\nabla\varphi_k, \nabla\varphi_l)$  are mass and Laplace matrices, respectively. The gradient vectors are  $(g_u)_l = L_u(x_k + \alpha\delta x_k)(\varphi_l)$ , and correspondingly for  $g_\lambda$  and  $g_q$ . Here,  $\varphi_l$  are again basis functions from the discrete approximation space to the state and adjoint variable, and  $\chi_l$  for the parameters.

The evaluation of  $\underline{p}(\alpha)$  therefore requires only two inversions of  $Y_1$  per experiment, and of one mass matrix for the parameters. Setting up the gradient vectors reuses operations that are also available from the generation of the linear system in each Gauss-Newton step. With this merit function, the computation of a step length is then done using the usual methods (see, e.g., [30]). Compared to the effort required for the solution of the Newton step, the effort for line search is usually negligible, in particular since not many evaluations of  $\underline{p}$  will be necessary. However, numerical experiments indicate that the effort can be further reduced by approximating  $Y_0, Y_1$ , for example by only considering their diagonal elements; however, the main feature of correctly scaling degrees of freedom corresponding to cells of different size should be preserved in schemes with adaptively chosen meshes.

### *Mesh refinement*

As noted, we may choose different grids for each Newton step. For practical purposes, these meshes should share a minimum of characteristics, as described above, but apart from that their generation is arbitrary. For the numerical examples presented in this paper, the meshes are kept constant for a number of nonlinear iterations until we are satisfied with the progress of the nonlinear iterations. The meshes are then refined using simple smoothness indicators to refine both the meshes for the state and adjoint variables, as well as for the parameter. In subsequent parts of this series, we will derive error estimates for the inverse problem, and generate refinement criteria from them. Also, the process of evaluating the progress of iterations on a grid will be discussed in more detail.

## **6. Linear solvers**

The linear system (19) is hardly solvable as is, except for the most simple problems. This is due to its size, which is twice the number of variables in each discrete forward problem, summed up over all experiments. Furthermore, it is indefinite and often extremely ill-conditioned (see [4]): if the state equations are the simple Poisson equations of the model problem, then the condition number of the matrix grows with the mesh size  $h$  as  $\mathcal{O}(h^{-4})$  (for a misfit functional  $m(\varphi) = \frac{1}{2}\|\nabla\varphi\|^2$ ) or even  $\mathcal{O}(h^{-6})$  (for  $m(\varphi) = \frac{1}{2}\|\varphi\|^2$ )! This, and the size of the problem makes both solution by direct as well as iterative

methods very hard or impossible for the large problems arising if many experiments are involved; for discussions of various approaches for such problems see, e.g., [18, 21, 3, 13].

However, we can re-state the system of equations by block elimination and use of the sub-structure of the individual blocks to obtain the following Schur complement formulation:

$$\mathbf{S} \delta q_{k,h} = \mathbf{F}_q - \sum_{i=1}^N \mathbf{C}^{iT} \mathbf{A}^{i-T} (\mathbf{F}_u^i - \mathbf{M}^i \mathbf{A}^{i-1} \mathbf{F}_\lambda^i), \quad (20)$$

$$\mathbf{A}^i \delta u_{k,h}^i = \mathbf{F}_\lambda^i - \mathbf{C}^i \delta q_{k,h}, \quad (21)$$

$$\mathbf{A}^{iT} \delta \lambda_{k,h}^i = \mathbf{F}_u^i - \mathbf{M}^i \delta q_{k,h}, \quad (22)$$

where  $\mathbf{S}$  denotes the Schur complement

$$\mathbf{S} = \left( \beta \mathbf{R} + \sum_{i=1}^N \mathbf{C}^{iT} \mathbf{A}^{i-T} \mathbf{M}^i \mathbf{A}^{i-1} \mathbf{C}^i \right). \quad (23)$$

These equations are much simpler to solve, mainly for their size and their structure: For the second and third equations, which are linearized forward and adjoint problems, efficient solvers are usually available. Since the equations for the individual experiments are separated, they can also be solved in parallel. The system in the first equation, (20), is small, its size being equal to the number  $\#\delta q_{k,h}$  of discretized parameters  $\delta q_{k,h}$ , which is *much* smaller than the total number of degrees of freedom and in particular independent of the number of experiments. Furthermore,  $\mathbf{S}$  has some nice properties:

**Lemma 2** *The Schur complement matrix  $\mathbf{S}$  is symmetric and positive definite if at least one of the matrices  $\mathbf{R}$  or  $\mathbf{M}$  as defined above is positive definite.*

The proof of this lemma is trivial, noting that both  $\mathbf{R}$  and  $\mathbf{M}$  stem from second derivatives of convex functionals, and are thus already at least positive semidefinite.

Regarding the actual solution of this equation, we note that one may build up an actual representation of the matrix  $\mathbf{S}$ . This would involve  $\#\delta q_{k,h}$  linearized forward and backward solutions for each experiment, which is usually too expensive as we anticipate  $\#\delta q_{k,h}$  to be in the range of thousands, or even more. However, by consequence of the lemma, we can use well-known and fast iterative methods for the solution of this equation, such as the Conjugate Gradient method. In each matrix-vector multiplication we would then have to perform one linearized forward and one backward step for each experiment. The other operations are comparably cheap.

Of crucial importance for the speed of convergence of the CG method is the condition number of the Schur complement matrix  $\mathbf{S}$ . Numerical experiments have shown that, in contrast to the original matrix (19), the condition number only grows by  $\mathcal{O}(h^{-2})$  (for  $m(\varphi) = \frac{1}{2} \|\nabla \varphi\|^2$ ) or  $\mathcal{O}(h^{-4})$  (for  $m(\varphi) = \frac{1}{2} \|\varphi\|^2$ ), i.e. by two orders of  $h$  less than the full matrix [4]. Furthermore and even more importantly, the condition number improves if more experiments are treated, i.e.  $N$  is higher, corresponding to the fact that more information reduces the ill-posedness of the problem. In particular, the condition number of the Schur complement matrix is not greater than the maximal

condition number of its building blocks (assuming that both  $\mathbf{R}$  and  $\mathbf{C}^{iT} \mathbf{A}^{i-T} \mathbf{M}^i \mathbf{A}^{i-1} \mathbf{C}^i$  are regular):

**Lemma 3** *For the condition number  $\kappa(\mathbf{S})$  of the Schur complement matrix, there holds*

$$\kappa(\mathbf{S}) \leq \max \left\{ \kappa(\mathbf{R}), \max_{1 \leq i \leq N} \kappa \left( \mathbf{C}^{iT} \mathbf{A}^{i-T} \mathbf{M}^i \mathbf{A}^{i-1} \mathbf{C}^i \right) \right\}.$$

The proof is simple using Rayleigh quotients for upper and lower eigenvalues and some basic inequalities; it is thus omitted. The extension where  $\mathbf{R}$  is only semidefinite is also easily obtained. In practice, the condition number  $\kappa(\mathbf{S})$  of the joint inversion matrix is often significantly smaller than that of the single experiment inversion matrices.

Finally, the CG method allows us to terminate the iteration relatively early, since high accuracy is not required in the computation of search directions and due to the fact that the right hand side of the equation is the gradient, and thus presumably already not too far away from the Gauss-Newton direction. Experience shows that for typical cases, a good solution can be obtained with much less than  $\# \delta q_{k,h}$  steps.

The solution of the Schur complement equation can be accelerated by preconditioning the matrix. Since one will not usually build up the matrix, a preconditioner cannot make use of the individual matrix elements. However, preconditioners based on the past history (i.e. previous Newton steps) are possible, for example through the use of updating formulas known from quasi-Newton methods. This is based on the fact that as we approach the solution, also the quadratic models which we use in the Gauss-Newton steps converge. Thus, we can use the information we have gained from matrix-vector multiplications in previous Newton iterations to precondition the present matrix. Such methods are presently under investigation. For other approaches see, for example, [32, 13].

Another advantage of the Schur complement formulation is that it is simple to parallelize (see [4]), which is particularly advantageous if the number of experiments is large, or the discretization of each of the experiments requires significant computing time or memory. In this case, matrix-vector multiplications with the Schur complement matrix are easily performed in parallel due to the sum structure of  $\mathbf{S}$ , and the remaining two equations defining the updates for the state and adjoint variables are independent anyway.

As a final note we remark that, as is well known, the full Newton matrix emerging from formulation (17) does not allow an as simple Schur complement representation, which furthermore is more expensive to form and is not necessarily positive definite.

## 7. Bound constraints

In the previous sections, we have described an efficient scheme for the discretization and solution of the inverse problem (5). However, in practical applications, one often has more information on the parameter than the one included in that formulation. For example, lower and upper bounds  $q_0 \leq q(\mathbf{x}) \leq q_1$  on parameters may be known,

possibly only in parts of the domain, or with spatially dependent bounds. The inequality constraint is understood as acting on each component of the set of parameters  $q$ . Such inequalities typically denote prior physical knowledge about the material properties we would like to identify, but even if such knowledge is absent, we will often want to impose constraints of the form  $q \geq q_0 > 0$ , if as in the model problem  $q$  appears as a coefficient in an elliptic operator.

In this section, we will extend the scheme developed above to incorporate such bounds, and we will show that the inclusion of these bounds comes at essentially no additional cost, since it only reuses information that is already there. On the contrary, as it reduces the size of the problems, it makes its solution faster. We would also like to stress that the approach does not make use of the actual form of state equations, or misfit or regularization functionals; it is therefore possible to implement it in a very generic way inside the Newton solver. The approach is based on the same ideas that active set methods use (see, e.g., [30]) and is similar to the gradient projection–reduced Newton method [33]. However, since we consider problems with several thousands or more parameters, some parts of the algorithm have to be devised differently. In particular, the determination of the active set has to happen on the continuous level, to make sure that different scalings due to local mesh refinement does not negatively affect the accuracy with which we can detect parameters at their bounds. For related approaches to constrained optimization problems in partial differential equations, we refer to [34, 12, 27].

### Basic idea

Since the method to be introduced is simple to extend to the more general case, let us describe the basic idea here for the special case that  $q$  is only one scalar parameter function, and that we only have lower bounds,  $q_0 \leq q(\mathbf{x})$ . The approach is then: before each step, identify those regions where the parameters are already at their bounds and we expect their values to move out of the feasible region. Let us denote this part of the domain, the so-called *active set*, by  $\mathcal{I} = \{\mathbf{x} \in \Omega : q_k(\mathbf{x}) = q_0, \delta q_k(\mathbf{x}) \text{ presumably } < 0\}$ . After discretization,  $\mathcal{I}$  will usually be the union of a number of cells from  $\mathbb{T}_k^q$ .

We then have to answer two questions: how do we identify  $\mathcal{I}$ , and once we have found it what do we do with the parameter degrees of freedom inside  $\mathcal{I}$ ? Let us start with the second question: In order to prevent these parameters from moving further outside, we simply set the respective updates to zero, and for this augment the definition (15) of the Gauss-Newton step by a corresponding equality condition:

$$\begin{aligned} \min_{\delta \mathbf{u}_k, \delta q_k} & J(\mathbf{u}_k, q_k) + J_u(\mathbf{u}_k, q_k)(\delta \mathbf{u}_k) + J_q(\mathbf{u}_k, q_k)(\delta q_k) \\ & + J_{uu}(\mathbf{u}_k, q_k)(\delta \mathbf{u}_k, \delta \mathbf{u}_k) + J_{qq}(\mathbf{u}_k, q_k)(\delta q_k, \delta q_k) \\ \text{s.t.} & A^i(q_k; u_k^i)(\varphi^i) + A_u^i(q_k; u_k^i)(\delta u_k^i, \varphi^i) + A_q^i(q_k; u_k^i)(\delta q_k, \varphi^i) = 0, \\ & (\delta q_k, \xi)_{\mathcal{I}} = 0, \end{aligned} \tag{24}$$

where the last constraint is to hold for all test functions  $\xi \in L^2(\mathcal{I})$ .

The optimality conditions for this minimization problem are then equal to the original ones stated in (16), except that the last equation has to be replaced by

$$\sum_i A_q^i(q_k; u_k^i)(\chi, \delta\lambda_k^i) + \beta r_{qq}(q_k)(\delta q_k, \chi) + (\mu, \chi)_{\mathcal{I}} = -L_q(x_k)(\chi), \quad (25)$$

where  $\mu$  is the Lagrange multiplier corresponding to the constraint  $\delta q_k|_{\mathcal{I}} = 0$ .

These equations can be discretized in the same way as before. In particular, we take the same space  $\mathcal{Q}_h$  for the discrete Lagrange multiplier  $\mu$  as for  $\delta q_k$ . After performing the same block elimination procedure we used for (19), we then get as matrix the following system to compute the Lagrange multipliers and parameter updates:

$$\begin{pmatrix} \mathbf{S} & \mathbf{B}_{\mathcal{I}}^T \\ \mathbf{B}_{\mathcal{I}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta q_{k,h} \\ \mu_h \end{pmatrix} = \begin{pmatrix} \mathbf{F}_{red} \\ 0 \end{pmatrix}, \quad (26)$$

with the reduced right hand side  $\mathbf{F}_{red}$  as in (20):

$$\mathbf{F}_{red} = \mathbf{F}_q - \sum_{i=1}^N \mathbf{C}^{iT} \mathbf{A}^{i-T} (\mathbf{F}_u^i - \mathbf{M}^i \mathbf{A}^{i-1} \mathbf{F}_{\lambda}^i).$$

The equations identifying  $\delta \mathbf{u}_{k,h}$  and  $\delta \boldsymbol{\lambda}_{k,h}$  are exactly as in (21) and (22), and are solved once  $\delta q_{k,h}$  is available.

The matrix  $\mathbf{B}_{\mathcal{I}}$  appearing in (26) is of mass matrix type. If we denote by  $\mathcal{I}_h$  the set of indices of those basis functions in  $\mathcal{Q}_h$  with a support that intersects  $\mathcal{I}$ , and  $\mathcal{I}_h(k)$  its  $k$ th element, then  $\mathbf{B}_{\mathcal{I}}$  is of size  $\#\mathcal{I}_h \times \#\delta q_{k,h}$ , and  $(\mathbf{B}_{\mathcal{I}})_{kl} = (\chi_{\mathcal{I}_h(k)}, \chi_l)_{\mathcal{I}}$ . In this way, the last row of the system,  $\mathbf{B}_{\mathcal{I}} \delta q_{k,h} = 0$ , simply sets parameter updates in the selected region to zero.

Let us now denote by  $\mathbf{Q}$  the projector onto the feasible set for  $\delta q_{k,h}$ , i.e. it is a rectangular matrix of size  $(\#\delta q_{k,h} - \#\mathcal{I}_h) \times \#\delta q_{k,h}$ , where we have a row for each degree of freedom  $i \notin \mathcal{I}_h$  with a 1 at position  $i$ . Then we have  $\mathbf{Q} \mathbf{B}_{\mathcal{I}}^T = 0$ , and we know that  $\delta q_{k,h} = \mathbf{Q}^T \mathbf{Q} \delta q_k$ . Elementary calculations then yield that the updates we seek satisfy

$$\begin{bmatrix} \mathbf{Q} \mathbf{S} \mathbf{Q}^T \end{bmatrix} (\mathbf{Q} \delta q_{k,h}) = \mathbf{Q} \mathbf{F}_{red}, \quad \mathbf{B}_{\mathcal{I}} \delta q_{k,h} = \mathbf{0},$$

which are conditions for disjoint subsets of parameter degrees of freedom. Besides being smaller, the reduced Schur complement  $\mathbf{Q} \mathbf{S} \mathbf{Q}^T$  inherits the following desirable properties from  $\mathbf{S}$ :

**Lemma 4** *The reduced Schur complement  $\mathbf{S}_{red} = \mathbf{Q} \mathbf{S} \mathbf{Q}^T$  is symmetric and positive definite. Its condition number satisfies  $\kappa(\mathbf{S}_{red}) \leq \kappa(\mathbf{S})$ .*

While symmetry is obvious, we inherit (positive) definiteness from  $\mathbf{S}$  by the fact that the matrix  $\mathbf{Q}$  has by construction full row rank. For the proof of the condition number estimate, let  $N^q = \#\delta q_{k,h}$ ,  $N_{red}^q = N^q - \#\mathcal{I}_h$ ; then we have for the maximal eigenvalue of  $\mathbf{S}_{red}$ :

$$\begin{aligned} \Lambda(\mathbf{S}_{red}) &= \max_{\mathbf{v} \in \mathbb{R}^{N_{red}^q}, \|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{S}_{red} \mathbf{v} = \max_{\mathbf{w} \in \mathbb{R}^{N^q}, \|\mathbf{w}\|=1, \mathbf{w}|_{\mathcal{I}_h} = 0} \mathbf{w}^T \mathbf{S} \mathbf{w} \\ &\leq \max_{\mathbf{w} \in \mathbb{R}^{N^q}, \|\mathbf{w}\|=1} \mathbf{w}^T \mathbf{S} \mathbf{w} = \Lambda(\mathbf{S}). \end{aligned}$$

Similarly, we get for the smallest eigenvalue  $\lambda(\mathbf{S}_{red}) \geq \lambda(\mathbf{S})$ , which completes the proof.

In practice,  $\mathbf{S}_{red}$  needs not be built up for use in a conjugate gradient method. Since application of  $\mathbf{Q}$  is essentially for free, the inversion of  $\mathbf{Q}\mathbf{S}\mathbf{Q}^T$  for the constrained updates is at most as expensive as that of  $\mathbf{S}$  for the unconstrained ones, and possibly cheaper if the condition number is indeed smaller. It is worth noting that treating constrained nodes in this way does not imply knowledge of the actual problem under consideration. Also, conversely, in the implementation of solvers for the state equations, one does not have to care about these constraints; this would, for example, be the case if positivity of a parameter is enforced by replacing  $q$  by  $\exp(q)$ .

### *Determination of the active set*

There remains the question of how to determine the set of parameter updates we want to constrain to zero. For this, let us for a moment consider  $\mathcal{I}$  as an unknown set that is implicitly determined by the fact that the constraint is active there at the solution. The idea of active set methods is then the following: from (25), we see that at the optimum there holds  $(\mu, \chi)_{\mathcal{I}} = -L_q(x)(\chi)$  for all test functions  $\chi$ . Outside  $\mathcal{I}$ ,  $\mu$  should be zero, and optimization theory tells us that it must be negative inside. If we have not yet found the solution, these properties do not hold exactly, but as we approach the solution, the updates  $\delta\boldsymbol{\lambda}_k, \delta q_k$  become small and we can use the identity to get an approximation  $\tilde{\mu}_k$  to the Lagrange multiplier defined on all of  $\Omega$ . If we discretize it using the same space  $\mathcal{Q}_h$  as for the parameters, then we get

$$(\tilde{\mu}_{k,h}, \chi_h) = -L_q(x_{k,h})(\chi_h) \quad \forall \chi_h \in \mathcal{Q}_h.$$

We will then use  $\tilde{\mu}_{k,h}$  as an indicator whether a point lies inside the set where the constraint on  $q$  is active and define

$$\mathcal{I}_h = \{\mathbf{x} \in \Omega : q_{k,h}(\mathbf{x}) = q_0, \tilde{\mu}_{k,h}(\mathbf{x}) \leq -\varepsilon\},$$

with a small positive number  $\varepsilon$ . With the so fixed set  $\mathcal{I}_h$ , the algorithm proceeds as above. Since  $-L_q(x_{k,h})(\chi_h)$  is already available as the right hand side of the discretized Gauss-Newton step, computing  $\tilde{\mu}_{k,h}$  amounts to only the inversion of the mass matrix resulting from the left hand side term  $(\mu_{k,h}, \chi_h)$ . This is particularly cheap if  $\mathcal{Q}_h$  is made up of discontinuous shape functions.

Numerical experiments indicate that it is necessary to set up this scheme in a function space first, and only discretize afterwards. Enforcing bounds only after discretization would amount to replacing the mass matrix by the identity matrix. This would then lead to the elements of the Lagrange multiplier  $\tilde{\mu}_{k,h}$  having a size that scales with the size of the cell they are defined on, preventing us from comparing their size with a fixed number  $\varepsilon$  in the definition of the set  $\mathcal{I}_h$ .

At the end of this section, let us state two comments. First, it is important to note again that we need not resort to the actual form of the problem under consideration, and that those parts of the program implementing the state equations do not need to know about the method used to impose the bounds. Second, the techniques shown

above are easily extendable to the case, where we do not have a constraint on each of the parameters,  $q_0 \leq q(\mathbf{x}) \leq q_1$ , but rather a constraint on a linear combination of the parameters, i.e.  $q_0 \leq Tq(\mathbf{x}) \leq q_1$ , with  $T$  a matrix. A typical constraint of this form would be that in elasticity,  $\lambda + 2\mu \geq 0$  must hold, where  $\lambda$  and  $\mu$  are the Lamé constants describing the compressible and shear moduli of a material.

## 8. Numerical examples

In this last section, let us give some examples of computations that have been performed with an implementation of the framework laid out above. All examples will consider recovering the spatially dependent coefficient  $q$  in a Laplace-type operator  $-\nabla \cdot (q\nabla)$  from measurements of the state variable. In one or two space dimensions, this is a model of a bar or membrane of variable stiffness that is subjected to a known force; the stiffness coefficient is then identified by measuring the deflection at every point. Similar applications arise also in groundwater management, where the hydraulic head satisfies a Poisson equation with  $q$  being the water permeability. Other applications are, for example, electrical impedance tomography in medical imaging or non-destructive material testing. We will consider more complicated problems, for example involving Helmholtz-type equations with high wave numbers, as appearing in seismic imaging, in later parts of this series (see also [4]).

In all cases, mesh adaptation is performed by simply looking at the smoothness of the solution. We refine the mesh for primal and dual variables (which are continuous) where the jumps of the gradients across cell interfaces,  $[\nabla u_h]$ , is large, for which we use the following criterion on cell  $K$ :

$$\eta_K^2 = h_K \int_{\partial K} |\mathbf{n} \cdot [\nabla u_h]|^2 dx.$$

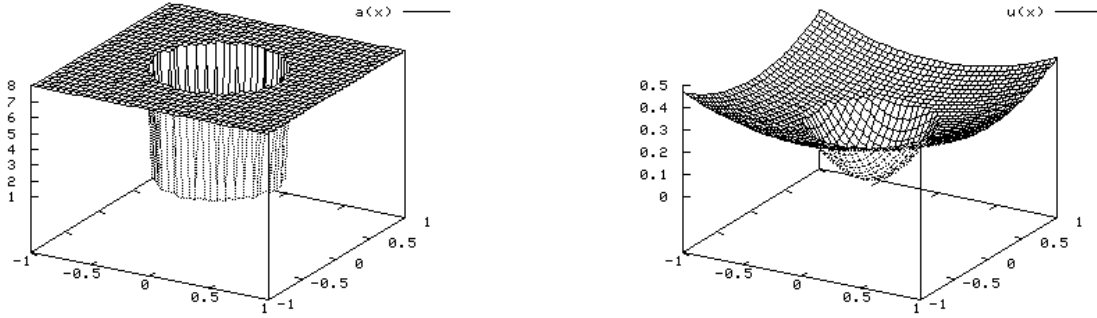
This quantity measures the size of the second derivatives of the numerical solution, scaled by appropriate powers of the local mesh width  $h_K$ . It was originally proposed as an a posteriori error estimate for the Laplace equation by Kelly et al. [25]. The coefficient will be discretized using discontinuous finite elements on a mesh that is refined by considering a scaled finite difference approximation of the gradient.

The program used here is built on the open source finite element library `deal.II` [5, 6, 7] and runs on multiprocessor machines or clusters of computers.

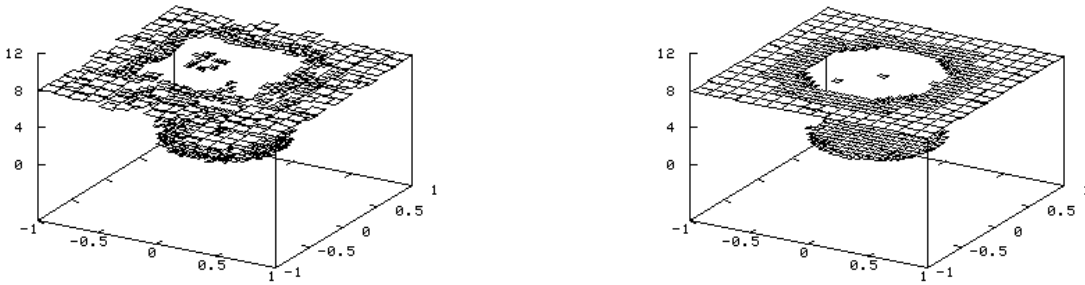
### 8.1. Example 1: A single experiment

In this section, we will consider identification of a discontinuous coefficient from a single global measurement of the solution of a Laplace equation. Thus, we have  $N = 1$  and

$$\begin{aligned} A(q; u)(\varphi) &= (q\nabla u, \nabla \varphi) - (f, \varphi), \\ m(Mu - z) &= \frac{1}{2} \|u - z\|_{\Omega}^2. \end{aligned}$$



**Figure 1.** Example 1: Exact coefficient  $q^*$  (left) and displacement  $u^*$  (right).



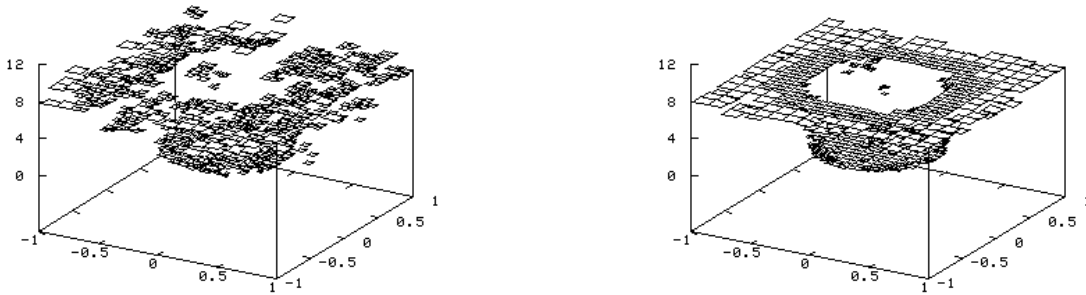
**Figure 2.** Example 1: Recovered coefficient with no noise, on grids  $\mathbb{T}^q$  with 800-900 degrees of freedom. Left: No bounds on  $q$  imposed. Right:  $1 \leq q \leq 8$  imposed.

The coefficient  $q^*$  we try to identify on a square domain is given by

$$q^*(\mathbf{x}) = \begin{cases} 1 & \text{for } |\mathbf{x}| < \frac{1}{2}, \\ 8 & \text{else.} \end{cases}$$

Note that the circular jump in the coefficient is not aligned with the mesh cells, so that we will need mesh refinement to resolve it properly. With  $f = 2d, d = \dim\Omega$ , the solution of the Laplace equation is  $u^*(\mathbf{x}) = |\mathbf{x}|^2$  inside  $|\mathbf{x}| < \frac{1}{2}$ , and  $u^*(\mathbf{x}) = \frac{1}{8}|\mathbf{x}|^2 + \frac{7}{32}$  otherwise.  $u^*$  and  $q^*$  are shown in Figure 1. Measurement data is generated by choosing  $z(\mathbf{x}) = u^*(\mathbf{x}) + \varepsilon(\mathbf{x})$ , where  $\varepsilon$  is random noise with a fixed amplitude  $\|\varepsilon\|_\infty$ .

For the case of no noise, i.e. measurements can be made everywhere without error, Figure 2 shows the discretization grid for and the values of the identified parameter after some refinement steps, at the left with no bounds on  $q$  imposed and at the right with tight bounds  $1 \leq q \leq 8$ . The latter case is typical if one knows that a body is composed of two different materials, but their interface is unknown. In both cases, the accuracy of reconstruction is good, and it is clear that adding bound information stabilizes the process. No regularization is used for this experiment.



**Figure 3.** Example 1: Same as Fig. 2, but with 2% noise in the measurement.

On the other hand, if noise is present, Figure 3 shows the identified coefficient without and with bounds on the parameter imposed. The noise level is  $\|\varepsilon\|_\infty/\|z\|_\infty \approx 0.02$ . Again, no regularization is used, and it is obvious that the additional information of bounds on the parameter improves the result significantly (quantitative results are given as part of the next section). Of course, adding a regularization term would also be a possible way to get a better reconstruction. However, since devising regularization functionals for discontinuous coefficients is difficult and often involves Bounded Variation-type seminorms with their well-known difficulties (see, e.g., [20, 17, 19]), we will not venture further in this direction and rather consider noise suppression by multiple measurements in the next section.

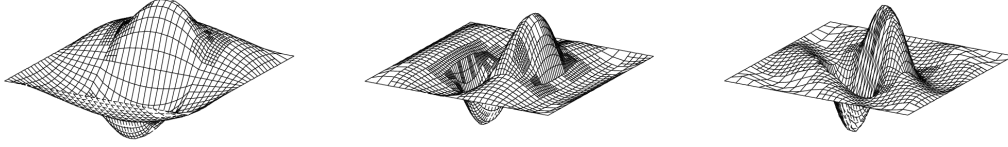
### 8.2. Example 2: Multiple experiments

Let us consider the same situation as in the previous section, but this time use multiple measurements  $z^i$  of the data and for each measurement use a different forcing  $f^i$ . Thus, for each experiment  $1 \leq i \leq N$ , we have

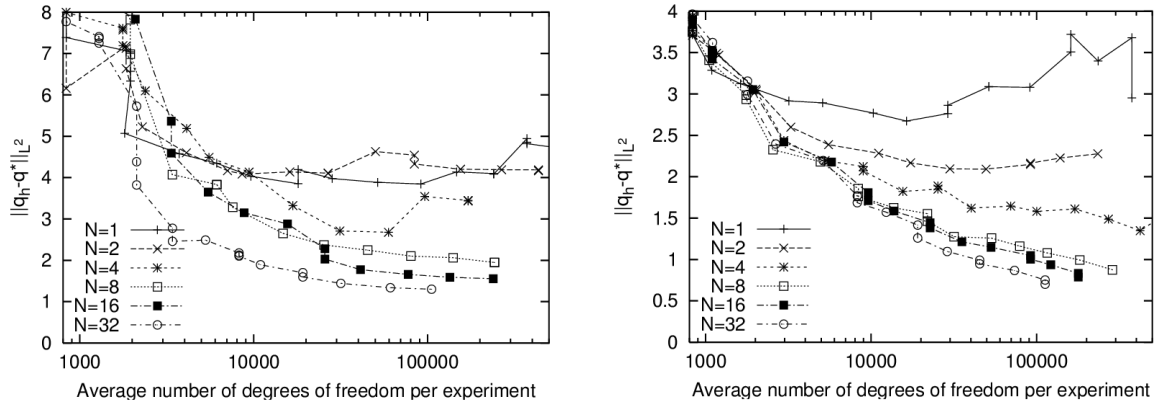
$$\begin{aligned} A^i(q; u^i)(\varphi) &= (q \nabla u^i, \nabla \varphi) - (f^i, \varphi), \\ m^i(M^i u^i - z^i) &= \frac{1}{2} \|u^i - z^i\|_\Omega^2. \end{aligned} \quad (27)$$

Our hope is that if each of these measurements is noisy, we can still recover the correct coefficient well if we only measure often enough. Since the measurements have independent noise, measuring more than once already yields a gain if the experimental setup is identical, i.e. when the right hand sides  $f^i$  are the same. However, simple considerations indicate that  $q$  is not identifiable at points where  $\nabla u^*$  is zero, see for example [8]; we thus gain more if we use different forcing functions  $f^i$  (with different sets of points where  $\nabla u^i = 0$ ) in different experiments, and numerical studies also show that this significantly increases the obtained accuracy, see [4].

In addition to the constant right hand side  $f^1(\mathbf{x}) = 2d$  and matching boundary conditions already used in the last section and shown in Fig. 1, we use as forcing terms



**Figure 4.** Example 2: Solutions of the state equations for experiments  $i = 2, 6, 12$ .



**Figure 5.** Error  $\|q_h - q^*\|_{L^2(\Omega)}$  in the reconstructed coefficient as a function of the number  $N$  of experiments used in the reconstruction and the average number of degrees of freedom used in the discretization of each experiment. Left: No bounds imposed. Right:  $1 \leq q \leq 8$  imposed. 2% noise in both cases. Note the different scales.

$f^i : \mathbb{R}^d \mapsto \mathbb{R}$ ,  $2 \leq i \leq N$  for the different state equations (27) the functions

$$f^i(\mathbf{x}) = \pi^2 \mathbf{k}_i^2 \sin(\pi \mathbf{k}_i \cdot \mathbf{x}),$$

where the vectors  $\mathbf{k}_i$  are the first  $N$  elements of the integer lattice  $\mathbb{N}^d = \{(0, 0), (1, 0), (1, 1), (0, 1), \dots\}$  when ordered by their  $l_2$ -norm and after eliminating collinear pairs in favor of the element of smaller magnitude. Numerical solutions for these right hand sides are shown in Fig. 4 for  $i = 2, 6, 12$ . Measurements  $z^i$  were obtained by numerically solving the state equations with the exact coefficient using a discretization of higher order than the one used in the reconstruction algorithm; random noise was then added to this numerical solution.

Fig. 5 shows a quantitative comparison of the reconstruction error  $\|q_h - q^*\|_{L^2(\Omega)}$ , as we increase the number of experiments used for the reconstruction, and as Newton iterations proceed on successively finer grids. In most cases, we only perform one Newton iteration on each grid, but if we are not satisfied with the progress on this grid, more than one iteration will be done; in this case, curves in the charts have vertically stacked data points, i.e. more than one data point for the same number of cells. The finest discretizations had 3-400,000 degrees of freedom for the discretization of state and

adjoint variable in each experiment (i.e., up to a total of about 4 million unknowns in the examples shown), and about 10,000 degrees of freedom for the discretization of the parameter  $q_h$ ; this then also was the size of the Schur complement matrix to be solved in each step, which takes between 100 and 200 CG iterations to solve to a relative accuracy of  $10^{-3}$ . We only show the case of non-zero noise level, since otherwise the number of experiments was not relevant for the reconstruction error.

From the charts, it is obvious that imposing bounds helps significantly, and that using more measurements is sufficient to strongly reduce the effects of noise. Also note that if noise is present, there is a limit for the amount of information that can be obtained and that further refining meshes deteriorates the result afterwards (see the erratic and growing behavior of curves for small  $N$  for high number of degrees of freedom; the identified parameter then deteriorates by high frequency oscillations). Since the numerical effort required to solve the problem grows roughly linear with the number of experiments, using more experiments may actually be cheaper in many cases, as the discretization of each of them requires significantly less degrees of freedom to achieve a given accuracy for the reconstructed parameter than if we used less experiments.

## 9. Conclusions and outlook

In this paper, we have presented a framework for the solution of large-scale multiple-experiment inverse problems. Its main features are:

- formulation in function spaces, allowing for different discretizations in subsequent steps of Newton-type nonlinear solvers;
- discretization by adaptive finite element schemes, with different meshes for state and adjoint variables on the one hand, and the parameters sought on the other;
- inclusion of bound constraints with a simple but efficient active set strategy;
- choice of a formulation that allows for efficient parallelization.

This framework has then been applied to some examples showing that inclusion of bounds can stabilize the identification of a coefficient from noisy data, as well as the (obvious) fact that measuring more than once can reduce the effects of noise. For these numerical examples, only a rather simple criterion was used to drive mesh refinement.

The framework as laid out above will be used as the basis for further articles in this series. In particular, while the entire framework is tailored to the use of different meshes, we have not shown how to construct them in a systematic way. Thus, in the second part, we will derive error estimates for the finite element discretization with respect to the minimization functional  $J(\mathbf{u}, q)$ , i.e. estimates for the quantity  $J(\mathbf{u}, q) - J(\mathbf{u}_h, q_h)$ , and use them to drive adaptive mesh refinement. We will also discuss how to couple mesh refinement strategies with the outer Newton-type iteration, and how to derive estimates for arbitrary functionals of the solution. This allows to specify, for example, that we are interested in the values of the coefficient only in part of the domain, and drive mesh adaptivity for this particular goal. These techniques will again be demonstrated using

numerical examples involving the identification of a coefficient from the solution of a Laplace equation.

In a final part, all these techniques will be applied to wave propagation problems involving the Helmholtz equation.

**Acknowledgments.** Part of this work has been done while at the Institute of Applied Mathematics, University of Heidelberg; this work has been supported by the Graduiertenkolleg “Modellierung und Wissenschaftliches Rechnen in Mathematik und Naturwissenschaften”. Recent work on this project is financed by a Postdoctoral Research Fellowship by the Institute for Computational Engineering and Sciences, University of Texas at Austin. Support by both institutions is greatly acknowledged. Computational resources have also been provided by the CFDlab at the University of Texas at Austin.

The author would like to thank Rolf Rannacher for his support and encouragement for this work.

## Bibliography

- [1] R. Acar. Identification of coefficients in elliptic equations. *SIAM J. Control Optim.*, 31:1221–1244, 1993.
- [2] U. M. Ascher and E. Haber. Grid refinement and scaling for distributed parameter estimation problems. *Inverse Problems*, 17:571–590, 2001.
- [3] U. M. Ascher and E. Haber. A multigrid method for distributed parameter estimation problems. Technical report, University of British Columbia, 2002.
- [4] W. Bangerth. *Adaptive Finite Element Methods for the Identification of Distributed Coefficients in Partial Differential Equations*. PhD thesis, University of Heidelberg, 2002.
- [5] W. Bangerth, R. Hartmann, and G. Kanschat. *deal.II Differential Equations Analysis Library, Technical Reference*, 2004. <http://www.dealii.org/>.
- [6] W. Bangerth and G. Kanschat. Concepts for object-oriented finite element software – the `deal.II` library. Preprint 99-43, SFB 359, Universität Heidelberg, October 1999.
- [7] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkhäuser Verlag, 2003.
- [8] H. T. Banks and K. Kunisch. *Estimation Techniques for Distributed Parameter Systems*. Birkhäuser, Basel–Boston–Berlin, 1989.
- [9] R. Becker. Adaptive finite elements for optimal control problems. Habilitation thesis, University of Heidelberg, 2001.
- [10] R. Becker, H. Kapp, and R. Rannacher. Adaptive finite element methods for optimal control of partial differential equations: Basic concept. *SIAM J. Contr. Optim.*, 39:113–132, 2000.
- [11] R. Becker and B. Vexler. Mesh adaptation for parameter identification problems. Proceedings of ENUMATH 2001, 2002. submitted.
- [12] M. Bergounioux, K. Ito, and K. Kunisch. Primal-dual strategy for constrained optimal control problems. *SIAM J. Control Optim.*, 37:1176–1194, 1999.
- [13] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver. submitted, 2003.
- [14] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.
- [15] H. G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, volume 183 of *Bonner Mathematische Schriften*. University of Bonn, Bonn, 1987.
- [16] J. Carter and C. Romero. Using genetic algorithms to invert numerical simulations. In *ECMOR*

- VIII: 8th European Conference on the Mathematics of Oil Recovery, Freiberg, Germany, pages E–45. European Association of Geoscientists and Engineers (EAGE), 2002.
- [17] Z. Chen and J. Zou. An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems. *SIAM J. Control Optim.*, 37:892–910, 1999.
- [18] M. J. Daydé, J.-Y. L’Excellent, and N. I. M. Gould. Element-by-element preconditioners for large partially separable optimization problems. *SIAM J. Sci. Comp.*, 18:1767–1787, 1997.
- [19] F. Dibos and G. Koepfler. Global total variation minimization. *SIAM J. Numer. Anal.*, 37:646–664, 2000.
- [20] S. Gutman. Identification of discontinuous parameters in flow equations. *SIAM J. Control Optimization*, 28(5):1049–1060, 1990.
- [21] E. Haber and U. M. Ascher. Preconditioned all-at-once methods for large, sparse parameter estimation problems. *Inverse Problems*, 17:1847–1864, 2001.
- [22] E. Haber, U. M. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems*, 16:1263–1280, 2000.
- [23] E. Haber and D. Oldenburg. Joint inversion: a structural approach. *Inverse Problems*, 13:63–77, 1997.
- [24] K. Ito, M. Kroller, and K. Kunisch. A numerical study of an augmented Lagrangian method for the estimation of parameters in elliptic systems. *SIAM J. Sci. Stat. Comput.*, 12:884–910, 1991.
- [25] D. W. Kelly, J. P. de S. R. Gago, O. C. Zienkiewicz, and I. Babuška. A posteriori error analysis and adaptive processes in the finite element method: Part I—error analysis. *Int. J. Num. Meth. Engrg.*, 19:1593–1619, 1983.
- [26] C. Kravaris and J. H. Seinfeld. Identification of parameters in distributed parameter systems by regularization. *SIAM J. Control Optim.*, 23:217–241, 1985.
- [27] R. Li, W. Liu, H. Ma, and T. Tang. Adaptive finite element approximation for distributed elliptic optimal control problems. *SIAM J. Control Optim.*, 41:1321–1349, 2002.
- [28] R. Luce and S. Perez. Parameter identification for an elliptic partial differential equation with distributed noisy data. *Inverse Problems*, 15:291–307, 1999.
- [29] X.-Q. Ma. A constrained global inversion method using an over-parameterised scheme: Application to poststack seismic data. *Geophysics Online*, August 2000.
- [30] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999.
- [31] M. K. Sen, A. Datta-Gupta, P. L. Stoffa, L. W. Lake, and G. A. Pope. Stochastic reservoir modeling using simulated annealing and genetic algorithms. *SPE Formation Evaluation*, pages 49–55, March 1995.
- [32] C. R. Vogel. Sparse matrix computations arising in distributed parameter identification. *SIAM J. Matrix Anal. Appl.*, 20:1027–1037, 1999.
- [33] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002.
- [34] W. H. Yu. Necessary conditions for optimality in the identification of elliptic systems with parameter constraints. *J. Optimization Theory Appl.*, 88:725–742, 1996.