TECHNISCHE UNIVERSITÄT CHEMNITZ

Torsten Hein Marcus Meyer

Identification of material parameters in linear elasticity - some numerical results

CSC/07-09

Chemnitz Scientific Computing Preprints

Impressum:

Chemnitz Scientific Computing Preprints — ISSN 1864-0087
(1995–2005: Preprintreihe des Chemnitzer SFB393)

Herausgeber:
Professuren für
Numerische und Angewandte Mathematik
an der Fakultät für Mathematik
der Technischen Universität Chemnitz

Postanschrift:
TU Chemnitz, Fakultät für Mathematik
09107 Chemnitz
Sitz:
Reichenhainer Str. 41, 09126 Chemnitz

http://www.tu-chemnitz.de/mathematik/csc/

TECHNISCHE UNIVERSITÄT CHEMNITZ

Chemnitz Scientific Computing Preprints

Torsten Hein Marcus Meyer

Identification of material parameters in linear elasticity - some numerical results

CSC/07-09

Abstract

In this paper we present some numerical results concerning the identification of material parameters in linear elasticity by dealing with small deformations. On the basis of a precise example different aspects of the parameter estimation problem are considered. We deal with practical questions such as the experimental design for obtaining sufficient data for recovering the unknown parameters as well as questions of treating the corresponding inverse problems numerically. Two algorithms for solving these problems can be introduced and extensive numerical case studies are presented and discussed.

CSC/07-09 ISSN 1864-0087 November 2007

Contents

Author's addresses:

Torsten Hein
Marcus Meyer
TU Chemnitz
Fakultät für Mathematik
Reichenhainer Str. 41
D-09107 Chemnitz

http://www.tu-chemnitz.de/mathematik/num_analysis/
http://www.tu-chemnitz.de/mathematik/inverse_probleme/

1 Introduction

In this paper the numerical studies of [4] are continued. We examine an inverse problem for small deformations in linear elasticity. For one material the deformation law can be described by the two so-called Lamé constants λ and μ which represents material properties. If we have a body consisting of different types of material we have two parameters for each material. If we do not know these parameters we can perform mechanical experiments in order to get some information depending on these parameters. This could be done in the following way:

- For a given force we measure the deformation of the body on some points on the boundary.
- We provide a certain displacement on one part of the boundary and consider the deformation on some points on other parts of the boundary. Additionally we can measure the distributed load which is necessary to obtain the desired deformation.

Thereby we have to regard that the measurements usually are afflicted with some errors. So the parameter identification problem under consideration deals with the following question: can we estimate the unknown parameters stable from the given (noisy) data? Considering this problems we want to give answers to several topics:

- What kind of data do we need so that we can identify the parameter uniquely?
- How many measurement points should be used for the identification and where should they be located?
- How do we solve the inverse problem numerically?
- Does it be necessary or advantageously to introduce some regularization strategies for stabilizing the numerical algorithms?

In our simulation we restrict our numerical experiments to deformations with plain stress, i.e. the stress into the third space coordinate is assumed to be zero, so that the calculations can be reduced to a problem on a two-dimensional geometry.

The paper is organized as follows: In section 2 we shortly present the analysis of small deformations in linear elasticity with plain stress. Thereby the weak formulation for solving the corresponding equations by a Finite-elements approach is specified. For the treatment of the corresponding inverse problems derivatives with respect to the unknown parameters are necessary. Those were derived also in section 2. Section 3 is devoted to the Finite elements discretization for the numerical solution of the underlying differential equations. In section 4 two types of (discretized) inverse problems for the estimation of the unknown parameters are formulated. Moreover, one unregularized least-squares minimization problem and a multi-parameter regularization approach for solving these inverse problems were introduced. Additionally, a modification of the multi-parameter regularization approach is presented. In the final section 5 first the test problem under consideration is described. The following detailed numerical studies presented in this section should give some answers to the questions formulated above.

2 Weak formulation

Let $\Omega \subset \mathbb{R}^2$ be a bounded domain which describes the geometry of the (two-dimensional) plain body under consideration. We consider small mechanical deformations with plain stress, see e.g. [3], i.e. there occur no stresses in the third dimension. Then a deformation $\varphi : \Omega \longrightarrow \mathbb{R}^2$ can be written as

$$\varphi(x) := x + u(x), \qquad x = (x_1, x_2)^T \in \Omega,$$

where $u(x) = (u_1(x_1, x_2), u_2(x_1, x_2))^T$ is the displacement by ignoring deformations into the third dimension. Moreover, we define the (two-dimensional) strain tensor

$$\varepsilon(u) := \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} \\ \varepsilon_{21} & \varepsilon_{22} \end{pmatrix} \qquad \text{with} \qquad \varepsilon_{ij} := \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \qquad i, j = 1, 2.$$

and the symmetric stress tensor

$$\sigma(u) := \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}$$

i.e. we have $\sigma_{12} = \sigma_{21}$. The correlation between σ and ε is given by the linear deformation law

$$\begin{aligned} \vec{\sigma}(u) := \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{pmatrix} &= \frac{E}{1-\nu^2}\begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{pmatrix}\begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{pmatrix} \\ &= \begin{pmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & 2\mu \end{pmatrix}\begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{pmatrix} \\ &=: C(\lambda, \mu)\vec{\varepsilon}(u) \end{aligned}$$

with

$$\lambda := \frac{E\nu}{1-\nu^2}, \; \mu := \frac{E}{2(1+\nu)} \quad \text{and} \quad \vec{\varepsilon}(u) := \left(\frac{\partial u_1}{\partial x_1}, \frac{\partial u_2}{\partial x_2}, \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right)^T.$$

Ignoring volume forces, the equations of equilibrium are given by

$$\begin{aligned} \operatorname{div}\sigma(u) &= 0 && \text{on } \Omega, \\ u &= g_D && \text{on } \Gamma_D, \\ \sigma(u)\cdot\vec{n} &= g_N && \text{on } \Gamma_N. \end{aligned} \tag{1}$$

Thereby Γ_D denotes the part of Γ with given Dirichlet boundary condition, whereas we have Neumann boundary conditions on Γ_N. As usual notation, $\vec{n}$ describes the outer normal vector on Γ_N.

The Dirichlet boundary condition reads as

$$u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} g_D^{(1)} \\ g_D^{(2)} \end{pmatrix} =: g_D \quad \text{on } \Gamma_D,$$

with functions $g_D^{(i)} \in L^2(\Gamma_D)$, $i = 1, 2$. It can be also replaced by a one-dimensional condition

$$u_i = g_D^{(i)} \quad \text{on } \Gamma_D \qquad \text{for } i = 1 \text{ or } i = 2.$$

The Neumann boundary condition we can rewrite with $\vec{n} = (n_1, n_2)^T$ as

$$\sigma(u)\cdot\vec{n} = \begin{pmatrix} \sigma_{11}n_1 + \sigma_{12}n_2 \\ \sigma_{21}n_1 + \sigma_{22}n_2 \end{pmatrix} = \begin{pmatrix} g_N^{(1)} \\ g_N^{(2)} \end{pmatrix} =: g_N$$

with two functions $g_N^{(i)} \in L^2(\Gamma_N)$, $i = 1, 2$.

In order to derive the weak formulation we define the space of test functions

$$\mathbb{V}_0 := \left\{ v \in \left[H^1(\Omega)\right]^2 \,:\, v|_{\Gamma_D} = 0 \right\}$$

and the space of ansatz functions

$$\mathbb{V}_D := \left\{ u \in \left[H^1(\Omega)\right]^2 \,:\, u|_{\Gamma_D} = g_D \right\}.$$

Then the bilinear form $a(\cdot,\cdot;\lambda,\mu) : \mathbb{V}_D \times \mathbb{V}_0 \longrightarrow \mathbb{R}$ is given by

$$a(u, v; \lambda, \mu) := \int\limits_{\Omega} \vec{\varepsilon}(u)^T C(\lambda,\mu)\vec{\varepsilon}(v)\, dx. \tag{2}$$

We now call $u \in [H^1(\Omega)]^2$ a weak solution of (1) if

$$\begin{cases} a(u, v; \lambda, \mu) = \displaystyle\int\limits_{\Gamma_N} g_N \cdot v\, dx, & \forall\, v \in \mathbb{V}_0, \\ u \in \mathbb{V}_D. \end{cases} \tag{3}$$

We define the forward operator $F : \mathcal{D}(F) \subset \mathcal{X} \longrightarrow \mathcal{Y}$ as

$$F(p) := u$$

with $p = (\lambda, \mu)^T$ and $u \in \mathbb{V}_D$ is the corresponding solution of (3). The choice of the spaces $\mathcal{X}$ and $\mathcal{Y}$ we discuss later.

We consider derivatives. Let be $u_0 := F(p_0)$ with $p_0 = (\lambda_0, \mu_0)^T \in \mathcal{D}(F)$. Furthermore we set $\Delta p := (0, \Delta\mu)^T$ with $p_0 + \Delta p \in \mathcal{D}(F)$ and define $p_\tau := p_0 + \tau\,\Delta p$ as well as $u_\tau := F(p_\tau)$ for $\tau \in (0, 1]$. We introduce

$$w^{(\mu)} := \lim_{\tau \to 0} \frac{1}{\tau} \left(F(p_\tau) - F(p_0) \right).$$

Obviously we have $w^{(\mu)} \in \mathbb{V}_0$ and

$$\begin{aligned} 0 &= \int\limits_\Omega \vec{\varepsilon}(u_\tau)^T C(\lambda_0, \mu_0 + \tau\Delta\mu)\vec{\varepsilon}(v)\, dx - \int\limits_\Omega \vec{\varepsilon}(u_0)^T C(\lambda_0, \mu_0)\vec{\varepsilon}(v)\, dx \\ &= \int\limits_\Omega \vec{\varepsilon}(u_\tau - u_0)^T C(\lambda_0, \mu_0)\vec{\varepsilon}(v)\, dx + \tau \int\limits_\Omega \vec{\varepsilon}(u_\tau)^T C(0, \Delta\mu)\vec{\varepsilon}(v)\, dx \end{aligned}$$

for all test functions $v \in \mathbb{V}_0$. Dividing by τ, taking the limit $\tau \to 0$ and noticing that $\vec{\varepsilon}(u_\tau) \to \vec{\varepsilon}(u_0)$ we conclude, that we can determine $w^{(\mu)}$ as solution of

$$\left\{ \begin{array}{ll} a(w^{(\mu)}, v; \lambda_0, \mu_0) = -a(u_0, v; 0, \Delta\mu), & \forall\, v \in \mathbb{V}_0, \\ w^{(\mu)} \in \mathbb{V}_0. & \end{array} \right. \tag{4}$$

Analogous, with $p_\tau := p_0 + \tau\,\Delta p$, $\Delta p := (\Delta\lambda, 0)^T$ fulfilling $p_0 + \Delta p \in \mathcal{D}(F)$ and

$$w^{(\lambda)} := \lim_{\tau \to 0} \frac{1}{\tau} \left(F(p_\tau) - F(p_0) \right),$$

we find, that $w^{(\lambda)} \in [H^1(\Omega)]^2$ satisfies

$$\left\{ \begin{array}{ll} a(w^{(\lambda)}, v; \lambda_0, \mu_0) = -a(u_0, v; \Delta\lambda, 0), & \forall\, v \in \mathbb{V}_0, \\ w^{(\lambda)} \in \mathbb{V}_0. & \end{array} \right. \tag{5}$$

Hence the derivative $F'(p_0) : \mathcal{X} \longrightarrow \mathcal{Y}$ is given as

$$F'(p_0) \begin{pmatrix} \Delta\lambda \\ \Delta\mu \end{pmatrix} := w^{(\lambda)} + w^{(\mu)}, \qquad \begin{pmatrix} \Delta\lambda \\ \Delta\mu \end{pmatrix} \in \mathcal{X}. \tag{6}$$

Finally we introduce another mapping. Let $\tilde{\Gamma}_D \subseteq \Gamma_D$ be a part of the Dirichlet boundary. Then we define $G : \mathcal{D}(F) \subset \mathcal{X} \longrightarrow \mathbb{R}^2$ as $G(p) := (g_1, g_2)^T$ where

$$\begin{pmatrix} \tilde{g}_N^{(1)} \\ \tilde{g}_N^{(2)} \end{pmatrix} := \sigma(u) \cdot \vec{n}|_{\Gamma_D}, \; g_i := \int\limits_{\tilde{\Gamma}_D} \tilde{g}_N^{(i)}\, ds, \; i = 1, 2. \tag{7}$$

We give an alternative way for calculating G, which will be used later for numerical calculations. Let, for given p, $u \in \mathbb{V}_D$ be a solution of (3). For simplicity, we assume, that $\tilde{\Gamma}_D = \Gamma_D$. The given displacement is obtained by a uniquely determined force $\tilde{g}_N = (\tilde{g}_N^{(1)}, \tilde{g}_N^{(2)})^T$ on Γ_D. Hence we can rewrite (1) by adding the boundary condition

$$\sigma(u) \cdot \vec{n} = \tilde{g}_N \quad \text{on } \Gamma_D.$$

Note, that we cannot remove the Dirichlet boundary condition completely, since then a (weak) solution of (1) is determined uniquely only up to a constant. The weak formulation now is given as finding the function $u \in \mathbb{V}_D$ which satisfies

$$a(u, v; \lambda, \mu) = \int\limits_{\Gamma_N} g_N \cdot v \, dx + \int\limits_{\Gamma_D} \tilde{g}_N \cdot v \, dx, \qquad \forall\, v \in \left[H^1(\Omega)\right]^2.$$

We choose two test functions $v_1, v_2 \in [H^1(\Omega)]^2$ with

$$v_1|_{\Gamma_D} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \text{and} \qquad v_2|_{\Gamma_D} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{8}$$

Using (7) we obviously have

$$g_i = a(u, v_i; \lambda, \mu) - \int\limits_{\Gamma_N} g_N^{(i)} \, ds, \qquad i = 1, 2. \tag{9}$$

Note, that there are no further restrictions on the test functions v_1 and v_2. The derivative $G'(p_0) : \mathcal{X} \longrightarrow \mathbb{R}^2$ is given by $G'(p_0)\Delta p = (g_1', g_2')^T \in \mathbb{R}^2$ with $p_0 := (\lambda_0, \mu_0)^T$ and $\Delta p := (\Delta\lambda, \Delta\mu)^T$ as

$$g_i' := a(F'(p_0)\Delta p, v_i; \lambda_0, \mu_0) + a(F(p_0), v_i; \Delta\lambda, \Delta\mu), \qquad i = 1, 2,$$

where the functions v_i, $i = 1, 2$, again satisfy (8).

3 Finite element discretization

Let $\mathcal{T}$ be a triangulation of the domain Ω with n nodes P_i, $1 \leq i \leq n$. Moreover, $\varphi_1, \ldots, \varphi_n$ denote the corresponding ansatz functions with $\varphi_i(P_j) = \delta_{ij}$, $1 \leq i, j \leq n$. Then we can define the Finite-element space

$$\mathbb{V}^{(2n)} := \operatorname{span}\{\tilde{\varphi}_1, \ldots, \tilde{\varphi}_{2n}\}$$

with

$$\tilde{\varphi}_i := \begin{pmatrix} \varphi_i \\ 0 \end{pmatrix} \quad \text{and} \quad \tilde{\varphi}_{n+i} := \begin{pmatrix} 0 \\ \varphi_i \end{pmatrix}, \qquad 1 \leq i \leq n.$$

The spaces $\mathbb{V}_0^{(2n)}$ and $\mathbb{V}_D^{(2n)}$ are the subspaces of elements of $\mathbb{V}^{(2n)}$ which satisfy the corresponding Dirichlet boundary conditions. We also refer to [2] for the numerical treatment of small deformations in linear elasticity in the three-dimensional case.

Moreover $\underline{u} := (u_1^{(1)}, \dots, u_n^{(1)}, u_1^{(2)}, \dots, u_n^{(2)})^T \in \mathbb{R}^{2n}$ represents the approximation of $u \in \mathbb{V}_D^{(2n)}$, i.e.

$$u \approx \sum_{i=1}^{n} \left(u_i^{(1)} \tilde{\varphi}_i + u_i^{(2)} \tilde{\varphi}_{n+i} \right) = \begin{pmatrix} \sum\limits_{i=1}^{n} u_i^{(1)} \varphi_i \\ \sum\limits_{i=1}^{n} u_i^{(2)} \varphi_i \end{pmatrix}.$$

We introduce discretization of the parameter p. We set $\mathcal{X} := [\operatorname{span}\{\psi_1, \dots, \psi_m\}]^2$ and assume

$$p = \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} \sum\limits_{i=1}^{m} \lambda_i \psi_i \\ \sum\limits_{i=1}^{m} \mu_i \psi_i \end{pmatrix}.$$

Then the function p can be considered as vector $\underline{p} := (\lambda_1, \dots, \lambda_m, \mu_1, \dots, \mu_m)^T \in \mathbb{R}^{2m}$.

Furthermore we define the matrix $\underline{K}(\underline{p}) = (k_{ij}) \in \mathbb{R}^{2n \times 2n}$, which is given by

$$k_{ij} = a(\tilde{\varphi}_i, \tilde{\varphi}_j; \lambda, \mu), \qquad 1 \le i, j \le 2n.$$

For the discretization of (3) we calculate the vector $\underline{f} = (f_1, \dots, f_{2n})^T \in \mathbb{R}^{2n}$ with

$$f_i := \int\limits_{\Gamma_N} g_N \cdot \tilde{\varphi}_i \, ds, \qquad 1 \le i \le 2n.$$

Then we obtain an approximate solution $\underline{u}$ of (3) by solving the problem

$$\frac{1}{2} \underline{u}^T \underline{K}(\underline{p}) \underline{u} - \underline{f}^T \underline{u} \to \min \qquad \text{subject to} \quad \underline{u} \in \mathbb{V}_D^{(2n)}. \tag{10}$$

Remark 3.1 *The condition $\underline{u} \in \mathbb{V}_D^{(2n)}$ can be easily reformulated as equation $\underline{H}\,\underline{u} = \underline{g}_D$. Then, by using a Langrange approach, we can find a solution $\underline{u}$ of (10) by solving the linear system*

$$\begin{array}{lclcl} \underline{K}(\underline{p})\underline{u} & + & \underline{H}^T \underline{\nu} & = & \underline{f} \\ \underline{H}\,\underline{u} & & & = & \underline{g}_D. \end{array}$$

Thereby $\underline{\nu} \in \mathbb{R}^s$ denotes the corresponding Lagrange multiplier. The dimension s depends on the number of the nodes laying on the Dirichlet boundary Γ_D.

We give an alternative representation of the matrix $\underline{K}(\underline{p})$. Therefore we introduce the matrices $\underline{K}^{(k)} = (k_{ij}^{(k)}) \in \mathbb{R}^{2n\times 2n}$, $1 \le k \le 2m$, with

$$k_{ij}^{(k)} := a(\tilde{\varphi}_i, \tilde{\varphi}_j; \psi_k, 0), \;\; k_{ij}^{(k+m)} := a(\tilde{\varphi}_i, \tilde{\varphi}_j; 0, \psi_k), \qquad 1 \le i, j \le 2n, \; 1 \le k \le m.$$

Then

$$\underline{K}(\underline{p}) = \sum_{k=1}^{m} \lambda_k \underline{K}^{(k)} + \mu_k \underline{K}^{(m+k)} \tag{11}$$

holds. The formula (11) has the advantage, that we can easily derive a discretization of the derivatives of the operators F and G.

The discretization $\underline{F} : \mathcal{D}(\underline{F}) \subset \mathbb{R}^{2m} \longrightarrow \mathbb{R}^{2n}$ of the operator F is now given as

$$\underline{F}(\underline{p}) := \underline{u}, \qquad \underline{u} \in \mathcal{D}(\underline{F}),$$

where, for given $\underline{p}$, $\underline{u}$ is the corresponding solution of (10).

For the discretization $\underline{G} : \mathbb{R}^{2m} \longrightarrow \mathbb{R}^2$ of the operator G we use the representation (9). We define the vector $\underline{q} = (q_1, \dots, q_n)^T \in \mathbb{R}^n$ with

$$q_i := \begin{cases} 1, & P_i \in \tilde{\Gamma}_D, \\ 0, & \text{else.} \end{cases}$$

Then we introduce the functions

$$v_1 := \begin{pmatrix} \sum_{i=1}^{n} q_i \varphi_i \\ 0 \end{pmatrix} = \sum_{i=1}^{n} q_i \tilde{\varphi}_i \qquad \text{and} \qquad v_2 := \begin{pmatrix} 0 \\ \sum_{i=1}^{n} q_i \varphi_i \end{pmatrix} = \sum_{i=1}^{n} q_i \tilde{\varphi}_{n+i}$$

which satisfy (8). Then, by using the notation of the previous section, we have

$$\begin{aligned} g_1 &= a(u, v_1; \lambda, \mu) - \int_{\Gamma_N} g_N \cdot v_1 \, ds \\ &= \sum_{i=1}^{n} q_i a(u, \tilde{\varphi}_i; \lambda, \mu) - \sum_{i=1}^{n} q_i \int_{\Gamma_N} g_N \cdot \tilde{\varphi}_i \, ds \\ &= (\underline{q}^T \;\; 0) \left(\underline{K}(\underline{p}) \underline{u} - \underline{f} \right), \end{aligned}$$

where $(\underline{q}^T \;\; 0)^T \in \mathbb{R}^{2n}$. The same consideration holds for v_2. Hence we obtain $\underline{G}(\underline{p})$ by easily calculating

$$\underline{G}(\underline{p}) = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} := \begin{pmatrix} \underline{q}^T & 0 \\ 0 & \underline{q}^T \end{pmatrix} \left(\underline{K}(\underline{p}) \underline{u} - \underline{f} \right).$$

Let $\underline{p}_0 \in \mathcal{D}(\underline{F})$ be given and $\underline{u}_0 := \underline{F}(\underline{p}_0)$. We calculate the vectors $\underline{w}_1, \dots, \underline{w}_{2m} \in \mathbb{R}^{(2n)}$ as solutions of

$$\frac{1}{2}\underline{w}^T \underline{K}(\underline{p}_0)\underline{w} + \underline{u}_0^T \underline{K}^{(i)}\underline{w} \to \min \qquad \text{subject to} \quad \underline{w} \in \mathbb{V}_0^{(2n)}, \qquad 1 \le i \le 2m. \tag{12}$$

Then $\underline{F}'(\underline{p}_0) := (\underline{w}_1, \dots, \underline{w}_{2m}) \in \mathbb{R}^{2n \times 2m}$ is the discretization of $F'(p_0)$.

The discrete derivative $\underline{G}'(\underline{p}_0) := (\underline{g}_1, \dots, \underline{g}_{2m}) \in \mathbb{R}^{2 \times 2m}$ is given by

$$\underline{g}_i := \begin{pmatrix} \underline{q}^T & 0 \\ 0 & \underline{q}^T \end{pmatrix} \left(\underline{K}(\underline{p}_0)\underline{w}_i + \underline{K}^{(i)}\underline{u}_0 \right), \qquad 1 \le i \le 2m.$$

4 The inverse problem

Let $\underline{y}_{data} \in \mathbb{R}^{\tilde{M}}$ denote a (noisy) observation of $\underline{y} := \underline{Q}\,\underline{u}$, where $\underline{u} := \underline{F}(\underline{p})$ and $\underline{Q} \in \mathbb{R}^{\tilde{M} \times 2n}$ denotes the corresponding projection matrix. Furthermore, let $P_{i_1}, \dots, P_{i_M}$ denote the nodes of the mesh $\mathcal{T}$, where the displacement is measured. We differentiate between 3 cases:

a) $\tilde{M} = 2M$ and $\underline{y} := (u_{i_1}^{(1)}, \dots, u_{i_M}^{(1)}, u_{i_1}^{(2)}, \dots, u_{i_M}^{(2)})^T$, which is the case, that the displacement is given in both directions x_1 and x_2,

b) $\tilde{M} = M$ and $\underline{y} := (u_{i_1}^{(1)}, \dots, u_{i_M}^{(1)})^T$, where we only consider displacements in x_1-direction and finally

c) $\tilde{M} = M$ and $\underline{y} := (u_{i_1}^{(2)}, \dots, u_{i_M}^{(2)})^T$, where displacements in x_2-direction are given.

The second data $\underline{z}_{data}$ is a (noisy) observation of $\underline{z} := \underline{\tilde{G}}(\underline{p})$, where $\underline{\tilde{G}}$ is a (possible) slight modification of $\underline{G}$. Again we discuss 3 possibilities:

i) $\underline{z} \in \mathbb{R}^2$ and $\underline{\tilde{G}} = \underline{G}$, i.e. the distributed load is given in both directions x_1 and x_2,

ii) $\underline{z} = z \in \mathbb{R}$ and $\underline{\tilde{G}}(\underline{p}) := g_1$, where $\underline{G}(\underline{p}) = (g_1, g_2)^T$, and alternatively

iii) $\underline{z} = z \in \mathbb{R}$ and $\underline{\tilde{G}}(\underline{p}) := g_2$.

Now we can define two types of inverse problems. In the first case we exploit only the displacement $\underline{y}_{data}$. The distributed load $\underline{z}_{data}$ does not enter the calculations. We define the appropriate problem.

Definition 4.1 (Discrete inverse problem I - [IP-1]) *Let $\underline{y}_{data}$ be the given data. We search for a parameter $\underline{p} \in \mathcal{D}(\underline{F})$, such that*

$$\underline{Q}\,\underline{F}(\underline{p}) = \underline{y}_{data}. \tag{13}$$

On the other hand, for identification problems it is always helpful to use as many as possible information about the unknown parameters. In our context we use both data $\underline{y}_{data}$ and $\underline{z}_{data}$ for evaluating the parameter $\underline{p}$. This leads to the following problem.

Definition 4.2 (Discrete inverse problem II - [IP-2]) *Let $\underline{y}_{data}$ and $\underline{z}_{data}$ be the given data. We search for a parameter $\underline{p} \in \mathcal{D}(\underline{F})$, such that*

$$\underline{Q}\,\underline{F}(\underline{p}) = \underline{y}_{data} \quad \textit{and} \quad \underline{\tilde{G}}(\underline{p}) = \underline{z}_{data}. \tag{14}$$

Remark 4.1 *It is reasonable to deal both problems. By considering Case II and III of Definition 5.1 in the next section, it can be easily seen that the parameter $\underline{p}$ cannot be identified uniquely by measuring only the displacement $\underline{y}_{data}$. Assume, that parameter $\underline{p}$ satisfies (13). Then $c\underline{p}$ is also a solution of (13) for each constant $c > 0$ with $c\underline{p} \in \mathcal{D}(\underline{F})$. In fact, numerical tests show, that for arbitrary $\underline{p} \in \mathcal{D}(\underline{F})$ the matrix $\underline{F}'(\underline{p})^T \underline{Q}^T \underline{Q}\,\underline{F}'(\underline{p})$ is (almost) singular with one eigenvalue (close to) zero. On the other hand, $\underline{G}(c\underline{p}) = c\underline{G}(\underline{p})$, so that we can overcome this nonuniqueness by examining the problem [IP-2].*

4.1 Solving the problem without regularization

We consider the second case [IP-2]. For solving this equations approximately, we deal with the minimizing problem

$$J_{ls}(\underline{p}) := \left\| \underline{Q}\,\underline{F}(\underline{p}) - \underline{y}_{data} \right\|^2 + \left\| \underline{\tilde{G}}(\underline{p}) - \underline{z}_{data} \right\|^2 \to \min . \tag{15}$$

In particular, both measurements have the same weight in the objective functional J_{ls}. This might cause some difficulties when only noisy data is given. Therefore an alternative approach is given in the next section.

For solving (15) numerically we apply a Gauß-Newton method, see e.g. [9, chapter 10]. For given $\underline{p}_k$ we calculate a search direction $\underline{d}_k$ as solution of the normal equation

$$\underline{H}_k \underline{d} = \underline{F}'(\underline{p}_k)^T \underline{Q}^T \left(\underline{y}_{data} - \underline{Q}\,\underline{F}(\underline{p}_k) \right) + \underline{\tilde{G}}'(\underline{p}_k)^T \left(\underline{z}_{data} - \underline{\tilde{G}}(\underline{p}_k) \right) \tag{16}$$

with

$$\underline{H}_k := \underline{F}'(\underline{p}_k)^T \underline{Q}^T \underline{Q}\,\underline{F}'(\underline{p}_k) + \underline{\tilde{G}}'(\underline{p}_k)^T \underline{\tilde{G}}'(\underline{p}_k) \tag{17}$$

and update

$$\underline{p}_{k+1} := \underline{p}_k + \gamma_k \underline{d}_k \quad \text{such that} \quad J_{ls}(\underline{p}_{k+1}) < J_{ls}(\underline{p}_k). \tag{18}$$

The numerical results are presented in section 5.

4.2 Noisy data and multi-parameter regularization approaches

Let us now assume, that we do not know the exact data $\underline{y}_{data}$ and $\underline{z}_{data}$. We have only noisy data $\underline{y}^{\delta}_{data}$ and $\underline{z}^{\delta}_{data}$ with given error estimate $\|\underline{y}^{\delta}_{data} - \underline{y}_{data}\| \leq \delta_1$ and $\|\underline{z}^{\delta}_{data} - \underline{z}_{data}\| \leq \delta_2$ with two constants $\delta_1 \geq 0$ and $\delta_2 \geq 0$. Therefore we deal with a multi-parameter regularization approach for solving the problem [IP-2] approximately. Let $J : \mathcal{D}(\underline{F}) \longrightarrow \mathbb{R}$ be a nonnegative objective functional. Then we consider the constrained minimization problem

$$J(\underline{p}) \to \min \qquad \text{subject to} \qquad \left\{ \begin{array}{lcl} \|Q\underline{F}(\underline{p}) - \underline{y}^{\delta}_{data}\| & \leq & \delta_1, \\ \|\underline{\tilde{G}}(\underline{p}) - \underline{z}^{\delta}_{data}\| & \leq & \delta_2. \end{array} \right. \tag{19}$$

For the analytical background of such problems we refer to [6], see also [5] for some newer results. Devoted to the noisy data a solution of (19) is denoted now by $\underline{p}^{\delta}$. If the problem is not ill-posed we also can think on problems of the form

$$\|Q\underline{F}(\underline{p}) - \underline{y}^{\delta}_{data}\|^2 \to \min \qquad \text{subject to} \qquad \|\underline{\tilde{G}}(\underline{p}) - \underline{z}^{\delta}_{data}\| \leq \delta_2, \tag{20}$$

or

$$\|\underline{\tilde{G}}(\underline{p}) - \underline{z}^{\delta}_{data}\|^2 \to \min \qquad \text{subject to} \qquad \|Q\underline{F}(\underline{p}) - \underline{y}^{\delta}_{data}\| \leq \delta_1, \tag{21}$$

where we do not introduce an additionally objective functional $J(\underline{p})$.

We can apply Lagrangian techniques for solving the problem (19). For the analytical background of Lagrangian techniques we advise to [1], see also [8] for some aspects of their numerical realization. For simplicity we assume that the side constraints are fulfilled both with equality. If not, we have to apply an active set strategy additionally. Moreover we set

$$J(\underline{p}) := \frac{1}{2}\|\underline{p} - \underline{p}^*\|^2$$

for a given a priori guess $\underline{p}^* \in \mathbb{R}^{2m}$. The Lagrangian functional is now given as

$$L(\underline{p}, \lambda_1, \lambda_2) := \frac{1}{2}\|\underline{p}-\underline{p}^*\|^2+\lambda_1 \left(\|Q\underline{F}(\underline{p}) - \underline{y}_{data}\| - \delta_1\right)+\lambda_2 \left(\|\underline{\tilde{G}}(\underline{p}) - \underline{z}_{data}\| - \delta_2\right).$$

We set

$$h_1(\underline{p}) := \|Q\underline{F}(\underline{p}) - \underline{y}_{data}\| - \delta_1 \quad \text{and} \quad h_2(\underline{p}) := \|\underline{\tilde{G}}(\underline{p}) - \underline{z}_{data}\| - \delta_2.$$

We apply a quadratic programming approach for solving (19). Let the iterate $(\underline{p}^{\delta}_k, \lambda_{1,k}, \lambda_{2,k})$ be given. Then we set

$$\underline{p}^{\delta}_{k+1} := \underline{p}^{\delta}_k + \underline{d}_k \quad \text{and} \quad \lambda_{i,k+1} := \lambda_i,\ i = 1, 2,$$

where $\underline{d}_k$ is the solution of the quadratic problem

$$\frac{1}{2}\underline{d}^T\underline{H}_k\underline{d} + J'(\underline{p}_k^\delta)^T\underline{d} \to \min \qquad \text{subject to} \qquad h_i(\underline{p}_k^\delta) + h_i'(\underline{p}_k^\delta)^T\underline{d} = 0, \quad i = 1,2. \tag{22}$$

with Lagrangian multiplier λ_1 and λ_2. Thereby $\underline{H}_k$ is an approximation of the Hessian of the functional $L(\underline{p}, \lambda_1, \lambda_2)$ at the element $(\underline{p}_k^\delta, \lambda_{1,k}, \lambda_{2,k})$. We have

$$\begin{aligned}
J'(\underline{p}) &= \underline{p} - \underline{p}^*, \qquad J''(\underline{p}) \equiv \underline{I}, \\
h_1'(\underline{p}) &= \frac{1}{\|\underline{Q}\underline{F}(\underline{p}) - \underline{y}_{data}^\delta\|}\underline{F}'(\underline{p})^T\underline{Q}^T\left(\underline{Q}\underline{F}(\underline{p}) - \underline{y}_{data}^\delta\right) \qquad \text{and} \\
h_2'(\underline{p}) &= \frac{1}{\|\underline{\tilde{G}}(\underline{p}) - \underline{z}_{data}^\delta\|}\underline{\tilde{G}}'(\underline{p})^T\left(\underline{\tilde{G}}(\underline{p}) - \underline{z}_{data}^\delta\right),
\end{aligned}$$

by assuming that the residuals are not zero. The matrix $\underline{I}$ denotes the identity matrix. The second derivatives of the side constraints we approximate by

$$\begin{aligned}
h_1''(\underline{p}) &\approx \frac{\underline{F}'(\underline{p})^T\underline{Q}^T\underline{Q}\underline{F}'(\underline{p})}{\|\underline{Q}\underline{F}(\underline{p}) - \underline{y}_{data}^\delta\|} \qquad \text{and} \\
h_2''(\underline{p}) &\approx \frac{\underline{\tilde{G}}'(\underline{p})^T\underline{\tilde{G}}'(\underline{p})}{\|\underline{\tilde{G}}(\underline{p}) - \underline{z}_{data}^\delta\|}.
\end{aligned}$$

Then we set

$$\underline{H}_k := \underline{I} + \lambda_{1,k}\frac{\underline{F}'(\underline{p}_k^\delta)^T\underline{Q}^T\underline{Q}\underline{F}'(\underline{p}_k^\delta)}{\|\underline{Q}\underline{F}(\underline{p}_k^\delta) - \underline{y}_{data}^\delta\|} + \lambda_{2,k}\frac{\underline{\tilde{G}}'(\underline{p}_k^\delta)^T\underline{\tilde{G}}'(\underline{p}_k^\delta)}{\|\underline{\tilde{G}}(\underline{p}_k^\delta) - \underline{z}_{data}^\delta\|}.$$

The solution of the KKT-system of (22) is given as solution of the equation

$$\begin{pmatrix} \underline{H}_k & h_1'(\underline{p}_k^\delta) & h_2'(\underline{p}_k^\delta) \\ h_1'(\underline{p}_k^\delta)^T & 0 & 0 \\ h_2'(\underline{p}_k^\delta) & 0 & 0 \end{pmatrix}\begin{pmatrix} \underline{d} \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = -\begin{pmatrix} \underline{p}_k^\delta - \underline{p}^* \\ h_1(\underline{p}_k^\delta) \\ h_2(\underline{p}_k^\delta) \end{pmatrix}. \tag{23}$$

The problems (20) and (21) can be treated similarly.

We finally want to mention another modification. With the same notation as above we consider the quadratic problem

$$\frac{1}{2}\underline{d}^T\underline{H}_k\underline{d} \to \min \qquad \text{subject to} \qquad h_i(\underline{p}_k^\delta) + h_i'(\underline{p}_k^\delta)^T\underline{d} = 0, \quad i = 1,2, \tag{24}$$

which we can obtain by setting $\underline{p}^* = \underline{p}_k^\delta$ in each iteration. The interpretation of the idea is simple. It is a Levenberg-Marquardt scheme for solving the problem:

$$\text{find any} \quad \underline{p} = \underline{p}^\delta \in \mathcal{D}(\underline{F}) \quad \text{with} \quad \|\underline{Q}\underline{F}(\underline{p}) - \underline{y}_{data}^\delta\| \le \delta_1 \quad \text{and} \quad \|\underline{\tilde{G}}(\underline{p}) - \underline{z}_{data}^\delta\| \le \delta_2. \tag{25}$$

In particular, if there occur no ill-posedness phenomena in solving [IP-2] since we have only a little number of parameters to be identified the idea (24) should be preferred. The convergence of the iteration is expected to be faster.

There is also another motivation of considering the modified iteration (24) in the case of noisy data. By solving least-square problems for given noisy data the following effect can be often observed: the convergence of the iterates becomes very slow, when the residual is already as small as or smaller than the given noise level. The noisy data causes trouble to find an optimal parameter which fits the given data almost exactly. On the other hand, there is no need to fit the noisy data as best as possible. We can stop the iteration as soon as the residual has reached the size of the given noise level. This idea often safes much numerical costs. Moreover, since we have multiple data, we cannot guarantee that a minimizer of (15) is also a solution of (25). Hence it is advantageously to solve (25) instead of (15) since we can avoid the final minimization procedure of (15) by stopping the iteration (24) when the accuracy of $\underline{F}(\underline{p}_k^\delta)$ and $\underline{\tilde{G}}(\underline{p}_k^\delta)$ with respect to the given data achieve the corresponding noise levels.

5 Numerical examples

In the following section we investigate the properties of the inverse problems [IP-1] and [IP-2] by detailed numerical studies. For the solution of the inverse problems the Gauß-Newton method without applying any regularization approach is used. With this strategy we are able to find out, if ill-posedness phenomena appear.

5.1 Formulation of a test problem

First we define a test problem with a fixed L-shaped geometry and varying boundary conditions. The domain $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ is given in Figure 1.

We assume that the parameter p is constant on Ω_i, $i = 1, 2, 3$. That means it can be represented as a vector $\underline{p} \in \mathbb{R}^6$. As default value we set

$$\underline{p}^\dagger = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} := \begin{pmatrix} 69 \\ 82 \\ 108 \\ 81 \\ 96 \\ 127 \end{pmatrix}, \tag{26}$$

which fits approximately the corresponding parameters of steel. The ansatz func-

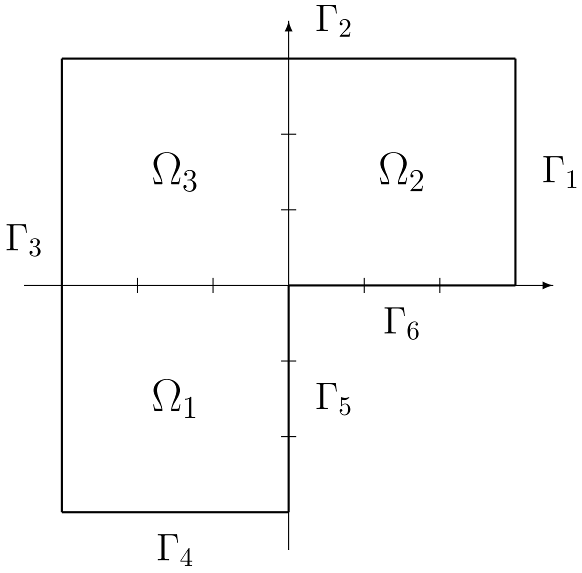

Figure 1: body Ω

tions ψ_i are determined as

$$\psi_i(\xi) := \chi_{\Omega_i}(\xi), \quad \xi \in \Omega,\ i = 1, 2, 3\ .$$

We consider three different types of boundary conditions.

Definition 5.1 (Boundary conditions)

- *Case I: given force g_N on Γ_4, i.e.*

$$\begin{aligned} u &= 0 && \textit{on}\,\Gamma_1, \\ \sigma(u)\cdot\vec{n} &= \begin{pmatrix} 0 \\ 20 \end{pmatrix} && \textit{on}\,\Gamma_4. \end{aligned}$$

- *Case II: given displacement in x_2-direction on Γ_4, i.e.*

$$\begin{aligned} u &= 0 && \textit{on}\,\Gamma_1, \\ u_2 &= 1 && \textit{on}\,\Gamma_4. \end{aligned}$$

- *Case III: given displacement on Γ_4, i.e.*

$$\begin{aligned} u &= 0 && \textit{on}\,\Gamma_1, \\ u &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} && \textit{on}\,\Gamma_4. \end{aligned}$$

The corresponding deformed bodies for the boundary conditions I-III are displayed in the figures 2-4. Note that for all the three cases the body Ω is fixed on Γ_1.

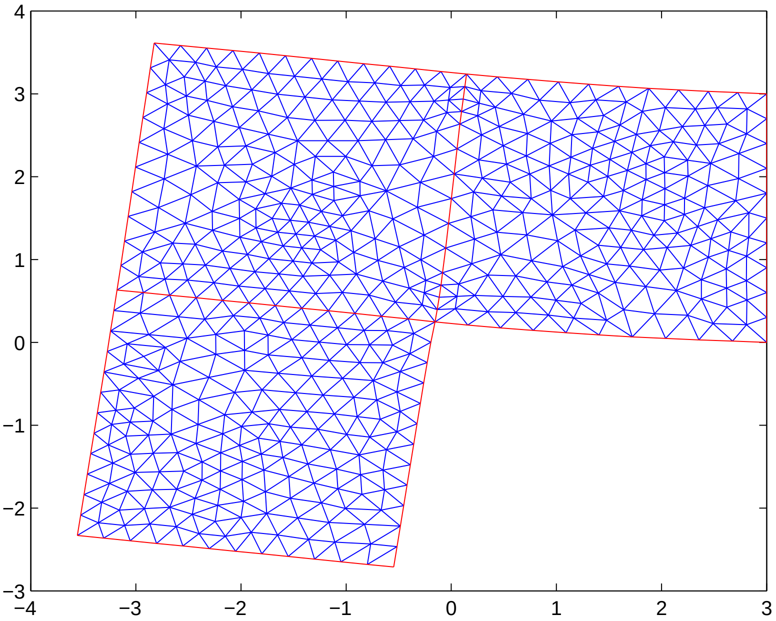

Figure 2: deformed mesh I

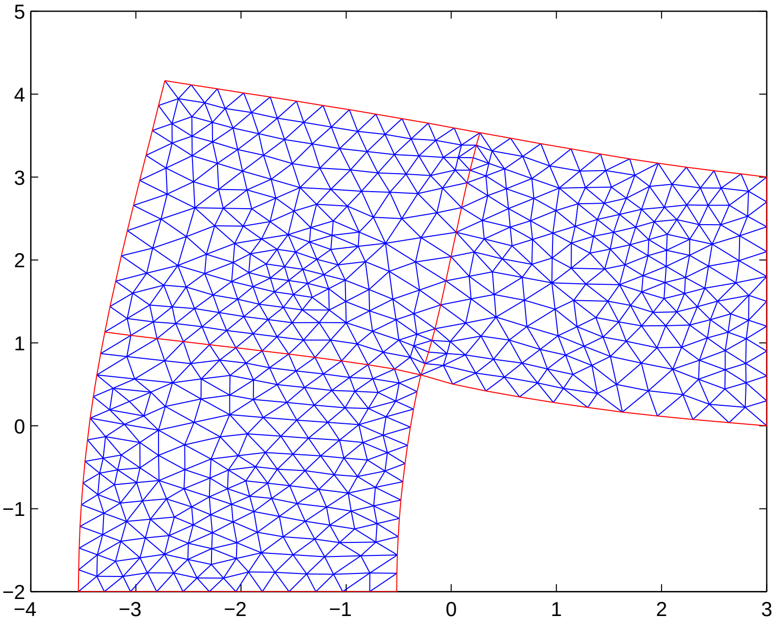

Figure 3: deformed mesh II

While solving the inverse problem [IP-1] or [IP-2], our aim is to identify the exact material parameter $\underline{p}^{\dagger}$ from given data. We discuss several measuring methods for specifying the data $\underline{y}_{data}$ and $\underline{z}_{data}$. With respect to the measured displacement

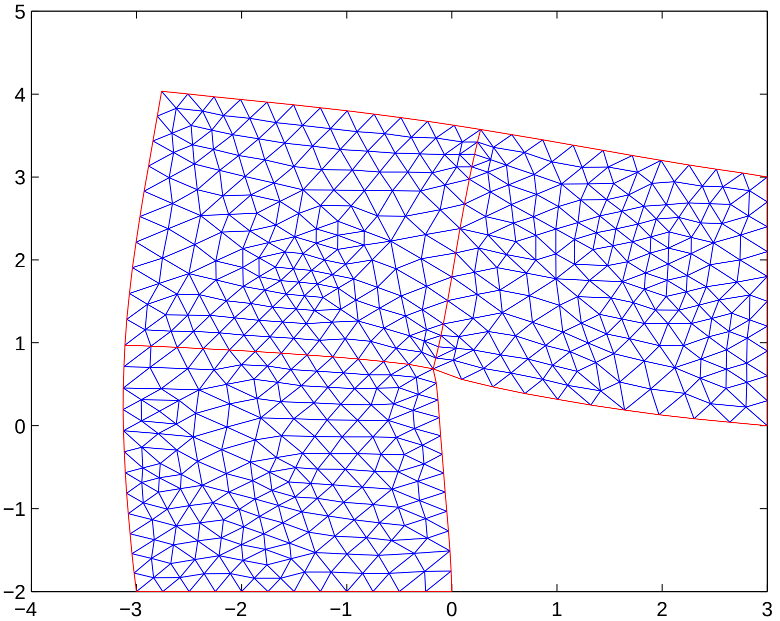

Figure 4: deformed mesh III

$\underline{y}_{data}$ we focus on variant c) as mentioned in the preceding section. That means, only displacements in the x_2-direction are considered. Since it is a single number we use the notation z_{data} (instead of $\underline{z}_{data}$) in the following. Different numbers M of measuring points are placed on the boundary subsets Γ_2 and Γ_3. In the numerical tests we check out four different constellations of measuring $\underline{y}_{data}$. For the first three cases there are only displacements measured on Γ_2, whereas the number of points M is varied from 12 to 24 and 48. Additional, as a fourth possibility, we set 12 points on Γ_2 and 12 points on Γ_3. Thus we get displacement data on two borders of the domain Ω.

Definition 5.2 (Measurement design) *The four measuring methods:*

$$12 \textit{ measuring points on } \Gamma_2 \tag{27}$$

$$24 \textit{ measuring points on } \Gamma_2 \tag{28}$$

$$48 \textit{ measuring points on } \Gamma_2 \tag{29}$$

$$12 \textit{ points on } \Gamma_2 \textit{ and } 12 \textit{ points on } \Gamma_3 \tag{30}$$

Figure 5 shows the arrangement of measuring points for the methods (27)-(30).

Additional to the displacement data, there is a possibility to get further information by measuring the force that induces the deformation. Namely the given displacement on Γ_4 in the boundary conditions II and III corresponds to a distributed load operating on Γ_4. The operator $\underline{\tilde{G}}$ or $\underline{G}$ assigns a force $\underline{z}$ being equivalent to these distributed load to a given parameter $\underline{p}$. Thereby we get the

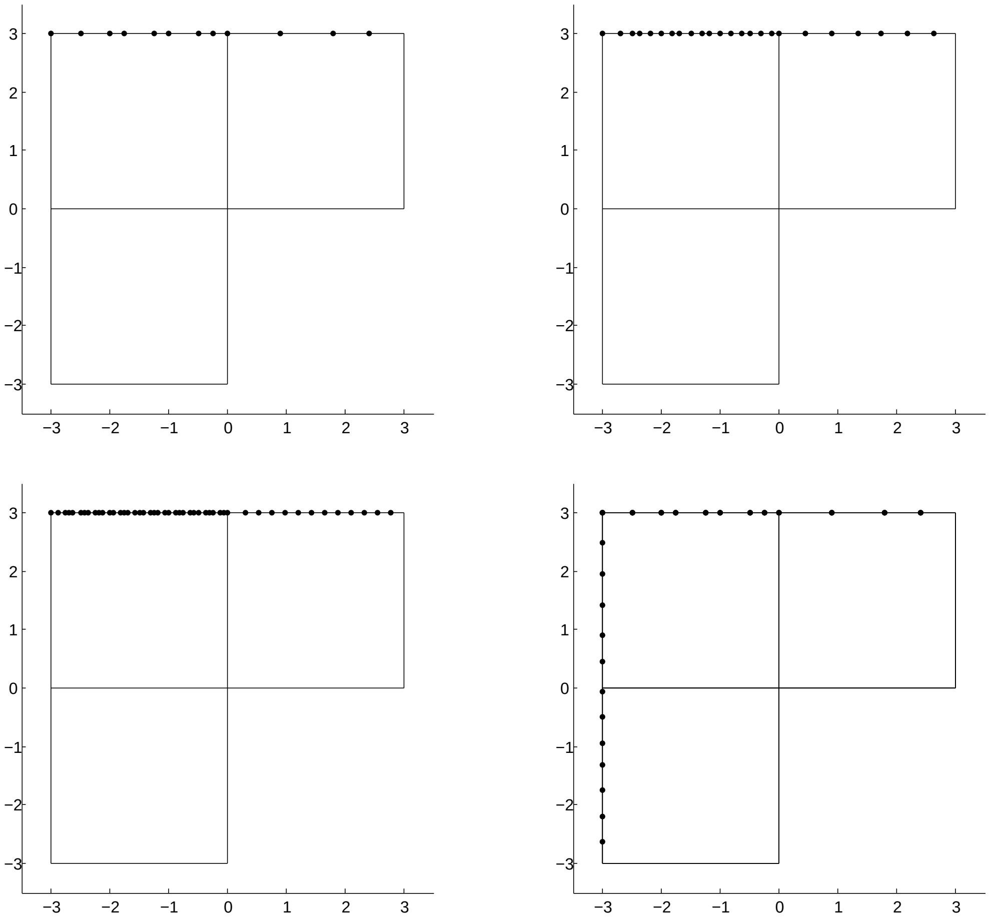

Figure 5: Measuring points for displacement data $\underline{y}_{data}$

second data $\underline{z}_{data}$. In the following computations method iii) as mentioned above is used. That means, only loads in x_2-direction are considered. For boundary condition I there is no sense in measuring the load, because it is implemented in the boundary condition. From this consideration we treat the identification of $\underline{p}^\dagger$ with boundary condition I as an inverse problem [IP-1]. The boundary conditions II and III refer to the inverse problem [IP-2].

5.2 Numerical implementation

For numerical computations we use MATLAB R2007, whereas the solution of the partial differential equation (1) is done by exploiting the PARTIAL-DIFFERENTIAL-EQUATION-TOOLBOX [7]. All computing times mentioned below were revealed under a LINUX setup on the CASE-computers of the department of mathematics at the TU Chemnitz. The time unit is seconds.

The data $\underline{y}_{data}$ and z_{data} we provide by solving the direct problem for given parameter $\underline{p}^\dagger$. In this context a finite element mesh with 132801 nodes is used to discretize the domain Ω. By this strategy we get exact data without any perturbation. In order to deal with noisy observations, the exact data is disturbed with given noise levels δ_1 and δ_2. The number δ_1 specifies the relative noise level of $\underline{y}_{data}$ and analogous δ_2 refers to the relative measurement error of z_{data}. We generate a random vector $\underline{e} \in \mathbb{R}^{\tilde{M}}$ with $N(0,1)$-distributed entries and compute the noisy displacement data as

$$\underline{y}^\delta_{data} := \underline{y}_{data} + \delta_1 \frac{\|\underline{y}_{data}\|_2}{\|\underline{e}\|_2} \underline{e} \; .$$

The disturbed load data is defined by

$$z^\delta_{data} := z_{data} + \delta_2 z_{data} \; .$$

As mentioned above we use the Gauß-Newton iteration (16)-(18) for solving the least-square problem (15). If no other value is referred to, the starting parameter of the iteration is fixed as

$$\underline{p}_0 := (100, 100, 100, 100, 100, 100)^T \; .$$

The iteration is terminated at stopping index K_0, if one of the three following stopping criteria is fulfilled for the first time. The first way to stop is, that the norm of the search direction is very small up to a given tolerance:

$$\left\|\underline{d}_{K_0}\right\|_2 \le 10^{-6} \; .$$

As a second termination rule we set the stopping index, if the residual norm is close enough to zero. That means for [IP-1]

$$\left\| \underline{Q}\underline{F}(\underline{p}_{K_0}) - \underline{y}^{\delta}_{data} \right\|_2 \leq 10^{-6}$$

and for [IP-2]

$$\left\| \begin{pmatrix} \underline{Q}\underline{F}(\underline{p}_{K_0}) \\ \underline{\tilde{G}}(\underline{p}_{K_0}) \end{pmatrix} - \begin{pmatrix} \underline{y}^{\delta}_{data} \\ z^{\delta}_{data} \end{pmatrix} \right\|_2 \leq 10^{-6} \ .$$

Finally the iteration is stopped if the iteration index exceeds a maximum number of iterations:

$$K_0 = k_{max} = 50 \ .$$

As a last detail for the numerical tests we have to verify that the problem remains elliptic and bounded during the whole iteration process. The ellipticity of the differential equation is warranted by installing a lower bound for the iterates $\underline{p}_k$. Therefore we define a positive constant $\varepsilon > 0$ and compute in every iteration the projection

$$\underline{\tilde{p}}_k := \max \left\{ \underline{p}_k, \underline{\varepsilon} \right\}$$

with a vector $\underline{\varepsilon} := (\varepsilon, \dots, \varepsilon)^T \in \mathbb{R}^6$. After this the iteration is continued with $\underline{\tilde{p}}_k$ instead of $\underline{p}_k$. In the same way we introduce an upper bound for the parameter $\underline{p}_k$. With a positive number ε_2, whereas $\varepsilon < \varepsilon_2 < \infty$ holds, we define

$$\underline{\hat{p}}_k := \min \left\{ \underline{p}_k, \underline{\varepsilon}_2 \right\} \ .$$

As default values we set

$$\varepsilon := 0.001 \quad \text{and} \quad \varepsilon_2 := 10^6 \ .$$

5.3 Numerical results

In the following part of the section we present some selected results of the numerical studies. The results listed in the tables below include numerical costs corresponding to the iteration number K_0 and the computing time. The quality of the results is specified by the relative error of the computed parameter $\underline{p}^{\delta}$ versus the exact parameter $\underline{p}^{\dagger}$, namely

$$\frac{\left\| \underline{p}^{\delta} - \underline{p}^{\dagger} \right\|_2}{\left\| \underline{p}^{\dagger} \right\|_2} \ .$$

In this context the index δ indicates the usage of noisy data.

a) Accuracy depending on the mesh size

In the beginning we check, what discretization level is necessary in order to get good results without exceeding the numerical costs. The first part of table 1 shows, that for exact data the quality of results increases by refining the mesh up to exactness of the identified parameter. Note that the accurate computation of $\underline{p}^\dagger$ for 132801 nodes is enforced by the fact, that the data is produced with the same mesh. Respecting the second part of table 1 we conclude, that the 33337-nodes mesh fits the needed requirements best. For noisy data we reveal finer meshes not being appropriate, because the enlargement of computation times is disproportionate to the improvement of the results. As one can see for the boundary conditions II and III the discretization error for 33337 nodes is much smaller than the error caused by the noise level. From this reason we restrict the tests and present the following results only for the 33337-nodes mesh. We mention that no adaptive meshing is used. In this paper we want to focus on the inverse problem. Further research may improve the finite element solution strategy included in the problem and lead to more efficient mesh structures.

Noise		**Nodes**	**Bound. Cond. I**			**Bound. Cond. II**			**Bound. Cond. III**		
δ_1	δ_2	n	K_0	Time	$\frac{\Vert\underline{p}^\delta-\underline{p}^\dagger\Vert_2}{\Vert\underline{p}^\dagger\Vert_2}$	K_0	Time	$\frac{\Vert\underline{p}^\delta-\underline{p}^\dagger\Vert_2}{\Vert\underline{p}^\dagger\Vert_2}$	K_0	Time	$\frac{\Vert\underline{p}^\delta-\underline{p}^\dagger\Vert_2}{\Vert\underline{p}^\dagger\Vert_2}$
0	0	561	500[1]	64.9	0.4543	7	2.5	0.0553	36	6.4	0.2163
0	0	2153	500[1]	316.9	0.4537	7	6.8	0.0399	9	8.1	0.1229
0	0	8433	8	31.3	0.3707	7	28.1	0.0211	7	28.1	0.0449
0	0	33377	7	139.2	0.1983	6	122.8	0.0078	7	138.2	0.0134
0	0	132801	6	702.4	0.0000	6	698.5	0.0000	6	704.0	0.0000
10^{-4}	10^{-4}	561	50	8.2	0.4546	7	2.5	0.0677	72[1]	11.6	0.2480
10^{-4}	10^{-4}	2153	50	34.0	0.4538	6	6.3	0.0544	9	8.4	0.1733
10^{-4}	10^{-4}	8433	50	160.3	0.4128	6	24.5	0.0368	8	31.4	0.0867
10^{-4}	10^{-4}	33377	10	185.9	0.3732	6	122.0	0.0248	7	138.2	0.0519
10^{-4}	10^{-4}	132801	11	1165.3	0.3359	6	697.8	0.0184	7	795.7	0.0374

Table 1: Least-squares solutions for varying discretization levels with exact and noisy data ($M = 12$ measuring points on Γ_2)

b) Influence of the noise level

Our next considerations concern the influence of the noise levels δ_1 and δ_2 on the reached accuracy of $\underline{p}^\delta$. Table 2 lists results for several noise levels. We derive that the computing errors increase if the noise levels rise. Besides this expected behavior the iteration remains stable. So we cannot detect any instability due to some ill-posedness of the problem. Because of the small number of unknown parameters, that have to be identified, a regularization by discretization comes into effect. For boundary condition I we see that large errors in $\underline{p}^\delta$ occur while using measuring method (27). Later we will inspect this problem deeper. In the

[1] For some computations the maximum number of iterations is extended up to $k_{max} := 500$.

case of boundary condition II and III quite good approximations of the material parameter are reached for $\delta_1 = \delta_2 = 10^{-4}$. The relative deviation of the computed parameter lies between 2% and 5%. Increasing the noise level to 10^{-3} causes a loss of accuracy, that is no more acceptable.

In the middle part of table 2 the value of δ_2 was changed while δ_1 remained fixed. One can realize from the listed results, that the value of δ_2 has almost no effect on the accuracy of $\underline{p}^\delta$, if it is not too large. A conspicuous loss of quality is pointed out not before $\delta_2 > 10^{-2}$. From this follows for [IP-2] that the measured force data z^δ_{data} needs not to be as precise as the displacement data $\underline{y}^\delta_{data}$. The main part of the computing error is determined by the displacement noise level δ_1. As the last row of table 2 shows, the above mentioned bad results for $\delta_1 = 10^{-3}$ cannot be improved by increasing the accuracy of z^δ_{data}.

Noise		**Bound. Cond. I**			**Bound. Cond. II**			**Bound. Cond. III**		
δ_1	δ_2	K_0	Time	$\frac{\|\underline{p}^\delta - \underline{p}^\dagger\|_2}{\|\underline{p}^\dagger\|_2}$	K_0	Time	$\frac{\|\underline{p}^\delta - \underline{p}^\dagger\|_2}{\|\underline{p}^\dagger\|_2}$	K_0	Time	$\frac{\|\underline{p}^\delta - \underline{p}^\dagger\|_2}{\|\underline{p}^\dagger\|_2}$
0	0	7	139.2	0.1983	6	122.8	0.0078	7	138.2	0.0134
10^{-4}	10^{-4}	10	185.9	0.3732	6	122.0	0.0248	7	138.2	0.0519
10^{-3}	10^{-3}	50	819.1	0.4583	6	122.9	0.1990	50	816.4	0.6364
10^{-4}	10^{-4}	-	-	-	6	122.0	0.0248	7	138.2	0.0519
10^{-4}	10^{-3}	-	-	-	6	122.7	0.0249	7	138.1	0.0523
10^{-4}	10^{-2}	-	-	-	6	122.5	0.0273	7	138.1	0.0562
10^{-4}	10^{-1}	-	-	-	6	122.7	0.1044	8	145.1	0.1289
10^{-3}	10^{-3}	-	-	-	6	122.9	0.1990	50	816.4	0.6364
10^{-3}	0	-	-	-	6	122.6	0.1987	50	817.8	0.6354

Table 2: Impact of different noise levels on the least-squares solutions ($M = 12$ measuring points on Γ_2, $n = 33377$ nodes)

c) Changing the measurement design

As we learned from the analyzes above, for measuring method (27) in combination with boundary condition I and noise levels $\delta_1 > 10^{-4}$ we achieve unsatisfying results referring to the accuracy. We are interested in, if the measurement design plays a role in this context. For improving results we use two ways. The first idea is just raising the number of measuring points. As a second way we include measurement information containing Γ_3. For details of the several measurements see definition 5.2 and figure 5.

In table 3 computed results for differing measuring methods and noise levels are listed. In the case of exact data we derive that simple enlarging of the number M has almost no effect. In contrast with this the quality of results concerning boundary condition I is considerably magnified by using method (30). That means, if we include displacement data covering information about Ω_1, we get fine approximations of $\underline{p}^\dagger$ by solving [IP-1], too. This result remains true for noisy data. Even if the noise level is 10^{-3}, the computation error does not exceed 7%

while applying boundary condition I. Continuing with boundary condition II we see, that the behavior of results differs a little. With a noise level of 10^{-4} the measuring methods do not evolve significant variances of accuracy. If the noise is increased up to 10^{-3}, measuring method (29) marks to be much more precise than all the other. While using boundary condition III in combination with noisy data, the numerical tests show, that increasing the number of measuring points improves the quality of results. Applying method (30) produces a comparable accuracy of parameter values. Altogether it is sticking out, that in the case of enlarged noise the quality of results can be improved by appropriate measurement design.

Meas. Pts.	**Noise**	**Bound. Cond. I**			**Bound. Cond. II**			**Bound. Cond. III**		
M	$\delta_1 = \delta_2$	K_0	Time	$\frac{\|p^\delta - p^\dagger\|_2}{\|p^\dagger\|_2}$	K_0	Time	$\frac{\|p^\delta - p^\dagger\|_2}{\|p^\dagger\|_2}$	K_0	Time	$\frac{\|p^\delta - p^\dagger\|_2}{\|p^\dagger\|_2}$
12 on Γ_2	0	7	139.2	0.1983	6	122.8	0.0078	7	138.2	0.0134
24 on Γ_2	0	7	139.4	0.2201	6	124.0	0.0084	8	154.7	0.0119
48 on Γ_2	0	7	141.2	0.2364	6	124.9	0.0088	8	156.7	0.0122
12 Γ_2/12 Γ_3	0	5	109.3	0.0014	7	139.9	0.0080	5	107.3	0.0017
12 on Γ_2	10^{-4}	10	185.9	0.3732	6	122.0	0.0248	7	138.2	0.0519
24 on Γ_2	10^{-4}	50	819.9	0.4540	6	123.7	0.0300	9	170.7	0.0327
48 on Γ_2	10^{-4}	-	-	-	7	141.0	0.0122	8	156.7	0.0051
12 Γ_2/12 Γ_3	10^{-4}	6	123.9	0.0061	7	139.6	0.0198	6	123.6	0.0069
12 on Γ_2	10^{-3}	50	819.1	0.4583	6	122.9	0.1990	50	816.4	0.6364
24 on Γ_2	10^{-3}	-	-	-	7	138.3	0.2837	50	810.5	0.2310
48 on Γ_2	10^{-3}	-	-	-	7	141.3	0.0461	15	266.9	0.0473
12 Γ_2/12 Γ_3	10^{-3}	7	139.6	0.0699	8	155.7	0.1956	6	123.6	0.0582

Table 3: Results for least-squares solutions by varying measuring methods (33377 nodes)

Reverting to the fact, that without using displacement measurement on Γ_3, the solution of [IP-1] contains large errors, we analyze the concerning results in detail. See table 4 for the error levels evolved in each subdomain of Ω. At this the computing errors restricted on the subdomains Ω_i, $i = 1, 2, 3$, are defined as

$$\Delta p_{\Omega_i} := \frac{\left\| \begin{pmatrix} \lambda_i \\ \mu_i \end{pmatrix} - \begin{pmatrix} \lambda_i^\delta \\ \mu_i^\delta \end{pmatrix} \right\|_2}{\left\| \begin{pmatrix} \lambda_i \\ \mu_i \end{pmatrix} \right\|_2}, \quad i = 1, 2, 3$$

with μ_i and λ_i being the exact values from (26). The computed parameters are denoted as μ_i^δ and λ_i^δ. We clearly recognize, that for boundary condition I the main part of error trouble happens in Ω_1, while measuring methods (27)-(29) are used. In contrast to this, for method (30) the parameter values in Ω_1 are as accurate as in the other subdomains. This indicates, that for every combination of boundary condition and measuring method we have to check whether the solution of the inverse problem is unique or not. In practical problems this may be an important fact for experimental design.

Meas. Pts.	**Bound. Cond. I**			**Bound. Cond. II**			**Bound. Cond. III**		
M	Δp_{Ω_1}	Δp_{Ω_2}	Δp_{Ω_3}	Δp_{Ω_1}	Δp_{Ω_2}	Δp_{Ω_3}	Δp_{Ω_1}	Δp_{Ω_2}	Δp_{Ω_3}
12 on Γ_2	0.8227	0.0152	0.0044	0.0446	0.0265	0.0033	0.0519	0.0162	0.0955
24 on Γ_2	1.0000	0.0392	0.0062	0.0441	0.0413	0.0043	0.0111	0.0601	0.0392
48 on Γ_2	0.9928	0.0157	0.0031	0.0229	0.0117	0.0008	0.0033	0.0085	0.0021
12 Γ_2/12 Γ_3	0.0085	0.0078	0.0032	0.0280	0.0278	0.0034	0.0014	0.0126	0.0009

Table 4: Computed errors of least-squares solutions seperated for each subdomain (33377 nodes, $\delta_1 = \delta_2 = 10^{-4}$)

In the same context, it plays an important role, whether the identification problem is modelled as [IP-1] or [IP-2]. For this we solved [IP-1] with boundary condition II and III. The difference to [IP-2] then is, that no force information on boundary is available. As we see in table 5, for boundary condition II and III omitting force measurement results in large errors. This holds for all measuring methods (27)-(30). The reason is, that without using additional force data, the parameter identification problem has not a unique solution. Due to a loss of information, the parameters cannot be identified. Hence the statements of remark 4.1 are certified.

Meas. Pts.	**Bound. Cond. II**			**Bound. Cond. III**		
M	K_0	Time	$\frac{\|\underline{p}^\delta - \underline{p}^\dagger\|_2}{\|\underline{p}^\dagger\|_2}$	K_0	Time	$\frac{\|\underline{p}^\delta - \underline{p}^\dagger\|_2}{\|\underline{p}^\dagger\|_2}$
12 on Γ_2	7	139.6	0.2150	7	139.2	0.2088
24 on Γ_2	7	138.6	0.2148	7	138.5	0.2094
48 on Γ_2	7	141.1	0.2147	7	140.6	0.2094
12 Γ_2/12 Γ_3	7	139.7	0.2155	6	123.6	0.2125

Table 5: Results for least-squares solutions without using force data z_{data} (33377 nodes, $\delta_1 = \delta_2 = 0$)

d) Robustness with respect to the initial guess

The next part of the numerical studies deals with the robustness of the Gauß-Newton iteration (16)-(18) w.r.t. the choice of initial guess $\underline{p}_0$, lower (ε) and upper (ε_2) bounds of the unknown parameters. At first varying iteration starting points were tested by substituting $\tilde{\underline{p}}_0$ for $\underline{p}_0$. Table 6 shows the corresponding results considering measuring method (27) and (30). Note that for method (27), (28) and (29) the robustness results are exactly the same.

In case of boundary condition II and III the bandwidth of adequate starting solutions is between the lower bound and about eight times (bound. cond. II) or two times (bound. cond. III) the exact parameter. This holds for all measuring methods in the same way. As a consequence the solution of [IP-2] is more robust for boundary condition II. If the initial guess $\tilde{\underline{p}}_0$ is too far away from the exact solution $\underline{p}^\dagger$ several trouble can happen depending on the chosen bounds. For wide bounds the problem becomes ill conditioned and the resulting numerical output is not useful. In table 6 the entry 'badly scaled' refers to this problem. Because badly scaling happens for the default values of ε and ε_2, we tried to extinguish that problem by variation of bounds. But constricting the lower and upper bounds correlates with the effect, that the iterates $\underline{p}_k$ tend to hang in the bounds. That means despite the modification of bounds no feasible search direction is found. For example in case of measuring method (27) with $\tilde{\underline{p}}_0 = 7.5\underline{p}_0$ the restriction of bounds $\varepsilon = 1$ and $\varepsilon_2 = 1000$ only corrects the ill conditioning, but the inaccurateness of results is more than 600%. The same situation is achieved for measuring method (30).

Due to the above results concerning the attainable accuracy, for boundary condition I it makes only sense to check the robustness, if measuring method (30) is used. From the results we discover, that in this setting the initial guess can be chosen in a very wide range. In particular all starting values between the default lower and upper bounds lead to an accurate solution.

Meas. Pts.	**Start.**	**Bound. Cond. I**			**Bound. Cond. II**			**Bound. Cond. III**		
M	$\tilde{\underline{p}}_0$	K_0	Time	$\frac{\|\underline{p}^\delta-\underline{p}^\dagger\|_2}{\|\underline{p}^\dagger\|_2}$	K_0	Time	$\frac{\|\underline{p}^\delta-\underline{p}^\dagger\|_2}{\|\underline{p}^\dagger\|_2}$	K_0	Time	$\frac{\|\underline{p}^\delta-\underline{p}^\dagger\|_2}{\|\underline{p}^\dagger\|_2}$
12 on Γ_2	$0.01\underline{p}_0$	-	-	-	7	138.9	0.0248	9	170.1	0.0519
12 on Γ_2	$0.10\underline{p}_0$	-	-	-	7	138.9	0.0248	9	170.4	0.0519
12 on Γ_2	$1.00\underline{p}_0$	-	-	-	6	122.0	0.0248	7	138.2	0.0519
12 on Γ_2	$2.00\underline{p}_0$	-	-	-	-	-	-	9	170.0	0.0519
12 on Γ_2	$3.00\underline{p}_0$	-	-	-	-	-	-	badly scaled		
12 on Γ_2	$5.00\underline{p}_0$	-	-	-	7	139.2	0.0248	badly scaled		
12 on Γ_2	$7.50\underline{p}_0$	-	-	-	8	154.9	0.0248	badly scaled		
12 on Γ_2	$9.00\underline{p}_0$	-	-	-	badly scaled			badly scaled		
12 Γ_2/12 Γ_3	$0.01\underline{p}_0$	12	218.2	0.0061	8	155.2	0.0198	6	123.2	0.0069
12 Γ_2/12 Γ_3	$0.10\underline{p}_0$	9	170.9	0.0061	7	139.5	0.0198	6	123.4	0.0069
12 Γ_2/12 Γ_3	$1.00\underline{p}_0$	6	123.9	0.0061	7	139.6	0.0198	6	123.6	0.0069
12 Γ_2/12 Γ_3	$2.00\underline{p}_0$	-	-	-	-	-	-	7	139.1	0.0069
12 Γ_2/12 Γ_3	$3.00\underline{p}_0$	-	-	-	-	-	-	badly scaled		
12 Γ_2/12 Γ_3	$5.00\underline{p}_0$	23	387.8	0.0061	9	171.3	0.0198	badly scaled		
12 Γ_2/12 Γ_3	$7.50\underline{p}_0$	23	389.1	0.0061	badly scaled			badly scaled		
12 Γ_2/12 Γ_3	$10^6\underline{p}_0$	23	392.5	0.0061	badly scaled			badly scaled		

Table 6: Robustness of the Gauß-Newton iteration with respect to the choice of the starting parameter (33377 nodes, $\delta_1 = \delta_2 = 10^{-4}$)

e) Multi-parameter regularization approaches

Even there are no ill-posedness effects it makes sense to test multi-parameter regularization approaches to show that they work in principle. For the boundary condition I we do not apply these methods since only the displacement is given.

So we only consider the conditions II and III.

We choose the measurement design with 24 measurement points on the upper boundary and consider two different noise levels $\delta_1 = 10^{-4}$ and $\delta_1 = 10^{-3}$ for the error in the measured displacement. In both cases we used $\delta_2 = 10^{-3}$ for the error in the second data. Then the multiparameter regularization (19) was applied. The results can be found in table 7. The comparable results for the standard least-square-minimization (15) were presented in table 3. As we see, in three of four situations we have higher numerical costs since the iteration numbers are increasing for $\delta_1 = 10^{-4}$ with both boundary conditions and for $\delta_1 = 10^{-3}$ for boundary condition II. But for $\delta_1 = 10^{-3}$ and boundary condition III we achieved a reduction of the iteration number. This shows the stabilization effect of the multi-parameter regularization approach (19): finding a minimizer of (19) is numerically more robust than solving the least-square problem (15). As we can also see, the accuracy of the estimated parameter $\underline{p}^\delta$ could be improved.

Noise		**Bound. Cond. II**			**Bound. Cond. III**		
δ_1	δ_2	K_0	Time	$\frac{\\|\underline{p}^\delta - \underline{p}^\dagger\\|_2}{\\|\underline{p}^\dagger\\|_2}$	K_0	Time	$\frac{\\|\underline{p}^\delta - \underline{p}^\dagger\\|_2}{\\|\underline{p}^\dagger\\|_2}$
10^{-4}	10^{-3}	13	230.4	0.0088	15	262.8	0.0144
10^{-3}	10^{-3}	14	246.7	0.0206	16	278.7	0.0286

Table 7: Multi-parameter regularization for different noise levels ($M = 24$ measuring points on Γ_2, $n = 33377$ nodes)

Since we do not have ill-posedness effects we finally consider the modified multi-parameter regularization iteration (24). As written above, the iteration stops as soon as we have found a solution of the problem (25). This avoides the final minimization procedure of (15) which might become numerically expensive for noisy data as seen in the results of table 3 for $\delta_1 = 10^{-3}$. The corresponding results are presented in table 8 with the same experimental design as in the descriptions above. So we can see that there is a decrease of the iteration number in all four calculations and a slight improvement of the accuracy of the estimated parameter $\underline{p}^\delta$. These calculations show, that for considering noisy data it makes numerically sense to replace the least-square-minimization (15) by multi-parameter

Noise		**Bound. Cond. II**			**Bound. Cond. III**		
δ_1	δ_2	K_0	Time	$\frac{\\|\underline{p}^\delta - \underline{p}^\dagger\\|_2}{\\|\underline{p}^\dagger\\|_2}$	K_0	Time	$\frac{\\|\underline{p}^\delta - \underline{p}^\dagger\\|_2}{\\|\underline{p}^\dagger\\|_2}$
10^{-4}	10^{-3}	9	168.8	0.0057	11	200.8	0.0121
10^{-3}	10^{-3}	10	184.3	0.0150	10	184.8	0.0280

Table 8: Modified multi-parameter regularization for different noise levels ($M = 24$ measuring points on Γ_2, $n = 33377$ nodes)

regularization algorithms even for problems without ill-posedness effects. If no ill-posedness effects occur the modification (24) should be preferred.

References

[1] Bertsekas, D.P.: *Constrained Optimization and Lagrange Multiplier Methods.* Athena Scientific (1996).

[2] Braess, D.: *Finite Elemente.* Springer, Berlin (2003).

[3] Kreissig, R.; Benedix, U.: *Höhere technische Mechanik.* Springer Wien New York (2002).

[4] Hein, T.; Meyer, M.: *Simultane Identifikation voneinander unabhängiger Materialparameter – numerische Studien.* TU Chemnitz, CSC Preprint 07-05 (2007).

[5] Hein, T.: *Multi-parameter regularization – convergence and convergence rates results.* Preprint 2007-17, TU Chemnitz, Fakultät für Mathematik (2007).

[6] Hofmann, B.: *Regularization for Applied Inverse and Ill-Posed Problems.* Teubner Verlag, Leipzig (1986).

[7] The MathWorks, Inc.:*Partial Differential Equation Toolbox Users's Guide.* Software-Dokumentation, www.mathworks.com (2006).

[8] Nocedal, J.; Wright, S.J.: *Numerical Optimization.* Springer, New York (1999).

[9] Schwetlick, H.: *Numerische Lösung nichtlinearer Gleichungen.* VEB Deutscher Verlag der Wissenschaften, Berlin (1979).

Some titles in this CSC and the former SFB393 preprint series:

06-01 T. Eibner, J. M. Melenk. p-FEM quadrature error analysis on tetrahedra. October 2006.

06-02 P. Benner, H. Faßbender. On the solution of the rational matrix equation $X = Q + LX^{-1}L^{\top}$. September 2006.

06-03 P. Benner, H. Mena, J. Saak. On the Parameter Selection Problem in the Newton-ADI Iteration for Large Scale Riccati Equations. October 2006.

06-04 J. M. Badía, P. Benner, R. Mayo, E. S. Quintana-Ortí, G. Quintana-Ortí, A. Remón. Balanced Truncation Model Reduction of Large and Sparse Generalized Linear Systems. November 2006.

07-01 U. Baur, P. Benner. Gramian-Based Model Reduction for Data-Sparse Systems. February 2007.

07-02 A. Meyer. Grundgleichungen und adaptive Finite-Elemente-Simulation bei ''Großen Deformationen". Februar 2007.

07-03 P. Steinhorst. Rotationssymmetrie für piezoelektrische Probleme. Februar 2007.

07-04 S. Beuchler, T. Eibner, U. Langer. Primal and Dual Interface Concentrated Iterative Substructuring Methods. April 2007.

07-05 T. Hein, M. Meyer. Simultane Identifikation voneinander unabhängiger Materialparameter - numerische Studien. Juni 2007.

07-06 A. Bucher, U.-J. Görke, P. Steinhorst, R. Kreißig, A. Meyer. Ein Beitrag zur adaptiven gemischten Finite-Elemente-Formulierung der nahezu inkompressiblen Elastizität bei großen Verzerrungen. September 2007.

07-07 U.-J. Görke, A. Bucher, R. Kreißig Zur Numerik der inversen Aufgabe für gemischte (u/p) Formulierungen am Beispiel der nahezu inkompressiblen Elastizität bei großen Verzerrungen. October 2007.

07-08 A. Meyer, P. Steinhorst. Betrachtungen zur Spektraläquivalenz für das Schurkomplement im Bramble-Pasciak-CG bei piezoelektrischen Problemen. October 2007.

The complete list of CSC and SFB393 preprints is available via
`http://www.tu-chemnitz.de/mathematik/csc/`

Chemnitz Scientific Computing Preprints – ISSN 1864-0087