Computer Assisted Methods in Engineering and Science, **24**: 113–126, 2017.

A remark on material parameter identification using finite elements based on constitutive models of evolutionary-type

Stefan Hartmann
Institute of Applied Mechanics
Clausthal University of Technology
Adolph-Roemer-Str. 2a, 38678 Clausthal-Zellerfeld, Germany
e-mail: stefan.hartmann@tu-clausthal.de

In this paper, we show that sensitivity analysis in connection with material parameter identification problems – using implicit finite elements of quasi-static problems on the basis of evolutionary-type constitutive equations – is related to simultaneous sensitivity equations and internal numerical differentiation. Thus, this study mainly focuses on investigating how these approaches are connected to the solution procedures based on finite elements. In addition, we discuss how to consider reaction forces in the sensitivity analysis, as this aspect is often neglected despite the fact that experimental results often involve force data.

Keywords: sensitivity analysis, material parameter identification, DAE-systems, finite elements, constitutive models of evolutionary-type.

Nomenclature

Abbreviations

DAE	–	differential-algebraic equations,
ODE	–	ordinary differential equations,
END	–	external numerical differentiation,
IND	–	internal numerical differentiation,
SSE	–	simultaneous sensitivity eqns.,
MLNA	–	multilevel-Newton algorithm.

Scalars, geometrical vectors and tensors

$f \in \mathbb{R}$	–	objective function in least-square method,
$\rho \in \mathbb{R}$	–	density,
$t \in \mathbb{R}$	–	time,
$w^{e(j)}$	–	weighting factor of Gaussian integration,
$\overrightarrow{k} \in \mathbb{V}$	–	acceleration of gravity,
$\overrightarrow{u} \in \mathbb{V}$	–	displacement vector,
$\overrightarrow{x} \in \mathbb{V}$	–	position vector to material point,
$\mathbf{E} \in \mathbb{L}(\mathbb{V})$	–	strain tensor,
$\mathbf{h} \in \mathbb{L}(\mathbb{V})$	–	stress defining equations,
$\mathbf{T} \in \mathbb{L}(\mathbb{V})$	–	stress tensor.

Dimensional quantities

n_{d} – no. of experimental data points at one time,
n_{D} – no. of all experimental data points at one time,
n^{e}_{GP} – no. of Gauss-point in element e,
n_{κ} – no. of material parameters,
n_{p} – no. of prescribed nodal displacements,
n_{q} – no. of local internal variables,
n_{Q} – no. of all internal variables in an entire finite element discretization,
n_{T} – no. times in an experiments, where the data is measured,
N – no. times in an finite element simulation, where the time integrator evaluates the DAE-system,
n_{u} – no. of unknown nodal displacements,
$\widetilde{n}_{\mathrm{u}}$ – no. of nodal displacements compared with experimental data.

Column vectors and matrices

$\mathbf{B}^{e(j)} \in \mathbb{R}^{6\times n^{e}_{\mathrm{u}}}$ – strain displacement matrix,
$\boldsymbol{C}_{\mathrm{c}} \in \mathbb{R}^{n_{\mathrm{p}}}$ – constraint equations for given nodal displacements,
$\boldsymbol{d}_k \in \mathbb{R}^{n_{\mathrm{d}}}$ – experimental displacement at time t_k,
$\boldsymbol{d} \in \mathbb{R}^{n_{\mathrm{D}}}$ – all experimental data of all discrete times,
$\boldsymbol{D}_{\mathrm{u}} \in \mathbb{R}^{n_{\mathrm{u}}\times n_{\kappa}}$ – sensitivity of unknown displacements w.r.t. material parameters,
$\boldsymbol{D}_{\mathrm{q}} \in \mathbb{R}^{n_{\mathrm{Q}}\times n_{\kappa}}$ – sensitivity of internal variables w.r.t. material parameters,
$\boldsymbol{D}_{\lambda} \in \mathbb{R}^{n_{\mathrm{p}}\times n_{\kappa}}$ – sensitivity of Lagrange-multipliers,
$\boldsymbol{e}_i \in \mathbb{R}^{n_{\mathrm{d}}}$ – unit vector containing only zeros, at position i there is a 1,
$\overline{\boldsymbol{e}}_i \in \mathbb{R}^{n_{\kappa}}$ – unit vector containing only zeros, at position i there is a 1,
$\mathbf{E}^{e(j)} \in \mathbb{R}^{6}$ – symmetric part of the strain tensor in Voigt notation in element e at Gauss-point (j),
$\boldsymbol{F} \in \mathbb{R}^{n_{\kappa}}$ – non-linear function to be solved to obtain material parameters (derivative of goal function f w.r.t. material parameters $\boldsymbol{\kappa}$),
$\overline{\boldsymbol{F}} \in \mathbb{R}^{n_{\mathrm{u}}+2n_{\mathrm{p}}+n_{\mathrm{Q}}}$ – DAE-system if reaction force computations is performed,
$\boldsymbol{g} \in \mathbb{R}^{n_{\mathrm{u}}}$ – discretized principle of virtual displacements,
$\boldsymbol{G}_{\mathrm{u}} \in \mathbb{R}^{n_{\mathrm{u}}\times n_{\mathrm{u}}}$ – derivative of discretized principle of virtual displacements w.r.t. unknown nodal displacements,
$\boldsymbol{G}_{\mathrm{q}} \in \mathbb{R}^{n_{\mathrm{u}}\times n_{\mathrm{Q}}}$ – derivative of discretized principle of virtual displacements w.r.t. internal variables,
$\boldsymbol{G}_{\kappa} \in \mathbb{R}^{n_{\mathrm{u}}\times n_{\kappa}}$ – partial derivative of discretized principle of virtual displacements w.r.t. material parameters,
$\mathbf{J}^{e(j)} \in \mathbb{R}^{3\times 3}$ – Jacobi-matrix of the coordinate transformation between reference element coordinates and the global coordinates,
$\mathbf{k}^{e} \in \mathbb{R}^{n^{e}_{\mathrm{u}}\times n^{e}_{\mathrm{u}}}$ – element stiffness matrix,
$\overline{\boldsymbol{K}}_{\mathrm{pu}} \in \mathbb{R}^{n_{\mathrm{p}}\times n_{\mathrm{u}}}$ – tangential stiffness matrix assigned to prescribed displacements,
$\boldsymbol{l} \in \mathbb{R}^{n_{\mathrm{Q}}}$ – integration step for internal variables,
$\boldsymbol{L}_{\kappa} \in \mathbb{R}^{n_{\mathrm{Q}}\times n_{\kappa}}$ – derivative of integration step for internal variables w.r.t. material parameters,
$\boldsymbol{L}_{\mathrm{u}} \in \mathbb{R}^{n_{\mathrm{Q}}\times n_{\mathrm{u}}}$ – derivative of integration step for internal variables w.r.t. unknown nodal displacements,
$\boldsymbol{L}_{\mathrm{q}} \in \mathbb{R}^{n_{\mathrm{Q}}\times n_{\mathrm{Q}}}$ – derivative of integration step for internal variables w.r.t. internal variables,
$\boldsymbol{M} \in \mathbb{R}^{(n_{\mathrm{u}}+n_{\mathrm{p}})\times n_{\mathrm{p}}}$ – assigns all unknown nodal displacements $\boldsymbol{u}_{\mathrm{a}}$ to the known displacements $\overline{\boldsymbol{u}} \in \mathbb{R}^{n_{\mathrm{p}}}$,

$\widetilde{\boldsymbol{M}} \in \mathbb{R}^{\widetilde{n}_\mathrm{u} \times n_\mathrm{u}}$ – assignment matrix of all nodal displacements to only that part, which is compared to experimental data,
$\overline{\boldsymbol{p}} \in \mathbb{R}^{n_\mathrm{u}}$ – known equivalent nodal force vector,
$\mathbf{q}^{e(j)} \in \mathbb{R}^{n_\mathrm{q}}$ – vector of internal variables at a Gauss-point,
$\boldsymbol{q} \in \mathbb{R}^{n_\mathrm{Q}}$ – unknown internal variables of all spatial integration points,
$\boldsymbol{Q}_\mathrm{u} \in \mathbb{R}^{n_\mathrm{Q} \times n_\mathrm{u}}$ – derivative of internal variables w.r.t. unknown nodal displacements,
$\boldsymbol{Q}_\kappa \in \mathbb{R}^{n_\mathrm{Q} \times n_\kappa}$ – derivative of internal variables w.r.t. material parameters,
$\mathbf{r} \in \mathbb{R}^{n_\mathrm{q}}$ – evolution equations for internal variables,
$\boldsymbol{r} \in \mathbb{R}^{n_\mathrm{Q}}$ – all evolution equations for all unknowns in the structure,
$\widetilde{\boldsymbol{r}} \in \mathbb{R}^{n_\mathrm{D}}$ – residual vector of least-square method,
$\boldsymbol{R}_\kappa \in \mathbb{R}^{n_\mathrm{Q} \times n_\kappa}$ – derivative of differential part of DAE-system w.r.t. material parameters,
$\boldsymbol{R}_\mathrm{u} \in \mathbb{R}^{n_\mathrm{Q} \times n_\mathrm{u}}$ – derivative of differential part of DAE-system w.r.t. unknown nodal displacements,
$\boldsymbol{R}_\mathrm{q} \in \mathbb{R}^{n_\mathrm{Q} \times n_\mathrm{Q}}$ – derivative of differential part of DAE-system w.r.t. internal variables,
$\boldsymbol{s} \in \mathbb{R}^{n_\mathrm{D}}$ – simulation data compared to experimental data,
$\mathbf{T} \in \mathbb{R}^{6}$ – symmetric part of stress tensor in Voigt notation,
$\boldsymbol{u} \in \mathbb{R}^{n_\mathrm{u}}$ – unknown nodal displacements,
$\overline{\boldsymbol{u}} \in \mathbb{R}^{n_\mathrm{p}}$ – known (prescribed) nodal displacements,
$\boldsymbol{u}_\mathrm{a} \in \mathbb{R}^{n_\mathrm{u}+n_\mathrm{p}}$ – all nodal displacements of the entire spatial discretization,
$\widehat{\boldsymbol{u}} \in \mathbb{R}^{n_\mathrm{p}}$ – unknown nodal displacements assigned to those DOF, where displacements are prescribed,
$\mathbf{u}^e \in \mathbb{R}^{n_\mathrm{u}^e}$ – element nodal displacements,
$\widetilde{\boldsymbol{u}} \in \mathbb{R}^{\widetilde{n}_\mathrm{u}}$ – discrete displacements compared to experimental data,
$\boldsymbol{Z}^e \in \mathbb{R}^{n_\mathrm{u}^e \times n_\mathrm{u}}$ – assembling (incidence) matrix assigned to unknown displacement degrees of freedom,
$\overline{\boldsymbol{Z}}^e \in \mathbb{R}^{n_\mathrm{u}^e \times n_\mathrm{p}}$ – assembling (incidence) matrix assigned to known displacement degrees of freedom,
$\boldsymbol{Z}_\mathrm{q}^{e(j)} \in \mathbb{R}^{n_\mathrm{q} \times n_\mathrm{Q}}$ – incidence matrix for assigning internal variables to a Gauss-point,
$\boldsymbol{\kappa} \in \mathbb{R}^{n_\kappa}$ – vector of material parameters,
$\boldsymbol{\lambda} \in \mathbb{R}^{n_\mathrm{p}}$ – Lagrange multipliers (negative reaction forces),
$\boldsymbol{\xi}^{(j)} \in \mathbb{R}^{3}$ – local coordinates of Gauss-points,
$\boldsymbol{1}$ – identity matrix,
$\boldsymbol{0}$ – zero matrix or vector.

1. Introduction

Material parameter identification can either be done for homogeneous deformations or inhomogeneous displacement fields in experiments with solid materials. If deformation is inhomogeneous, we have to solve the entire boundary-value problem. In such a case, it is common to make use of the finite element method. In [36], this was demonstrated for elastic materials with inclusion. The procedures for inelastic materials were discussed in [3, 25, 26]. The works of Mahnken have mainly contributed to material parameter identification in the context of finite element simulations of problems solid mechanics. Further contributions were provided in [4, 7, 19–21, 33], in the ones incorporating gradient-free schemes, in [15] and [17, 18]. Specific applications such as, for example, indentation tests, can be found [6, 23, 31]. Most of these approaches treat constitutive equations of evolutionary-type, such as models of viscoelasticity, viscoplasticity or rate-independent plasticity. In the field of heat equation, similar approaches are used, see, for example, [1, 2].

By contrast, specific considerations with regard to sensitivity analysis, i.e., the parameter identification on the basis of ordinary differential equations, differential-algebraic equations, and even

partial differential equations, were developed in numerical mathematics, for an overview see [34]. These apparently independent approaches, used in engineering and mathematical communities, are, however, connected, which is one of the essential aspects to be shown in this investigation. To this end, the interpretation of implicit finite element simulations – which are based on constitutive models of viscoelasticity, viscoplasticity, or even rate-independent plasticity – as the solution of differential-algebraic equations is applied, see [9]. Here, we restrict ourselves to applications drawing on the classical Backward-Euler method to integrate evolution equations of internal variables at Gauss-point level of finite element programs. Extensions to high-order time-integration schemes such as, for example, diagonally-implicit Runge-Kutta methods, are straightforward.

Since it is common to consider resulting forces in experiments, a sensitivity analysis of the reaction force computation on the basis of finite elements has to be considered as well. Here, the consistent approach presented in [14] is transferred to the sensitivity analysis of differential-algebraic equations (DAEs).

The structure of the paper is as follows. First, we recap the underlying problem. After that, we turn to numerical schemes of simultaneous sensitivity equations, internal numerical differentiation, and external numerical differentiation using a forward differential formula. Finally, an extension to reaction force computations is considered.

The notation in use is defined in the following manner: geometrical vectors are symbolized by $\overrightarrow{a}$ and second-order tensors $\mathbf{A}$ by bold-faced Roman letters. Furthermore, we introduce matrices at global level of finite element algorithms symbolized by bold-faced italic letters $\boldsymbol{A}$ and matrices on local (element) level using bold-faced Roman letters $\mathbf{A}$.

2. Sensitivity analysis

Commonly, a model has to be calibrated at experimental data $\boldsymbol{d} \in \mathbb{R}^{n_\mathrm{D}}$. The model depends on a set of parameters $\boldsymbol{\kappa} \in \mathbb{R}^{n_\kappa}$, which in our case are material parameters occurring in constitutive (material) equations. We consider models for which the internal variables are given by ordinary differential equations (ODEs) or by differential-algebraic equations (DAEs). The latter is the case if rate-independent yield function-based equations are chosen so that a yield condition has to be fulfilled, see [9, 37]. Pure ODEs appear – sometimes also with case distinctions controlled by loading conditions or Kuhn-Tucker conditions implying non-smooth functions – in models of viscoelasticity and viscoplasticity. Since the resulting problem is not essentially changed when considering ODEs instead of more general DAEs in the sense of optimization, we only treat models of evolutionary-type based on ODEs. In this respect, continuous differentiability of some constitutive models with respect to material parameters is violated.

In the following, we consider small strain problems to avoid lengthy presentation. The extension to large strains is straightforward if the tensors are operating on the reference configuration [27, 29]. The basic problems are quasi-static and isothermal boundary-value problems of the form

$$\begin{aligned} &\operatorname{div}\mathbf{T} + \rho\,\overrightarrow{k} = \overrightarrow{0}, \\ &\mathbf{T} = \mathbf{h}(\mathbf{E}, \mathbf{q}), \\ &\dot{\mathbf{q}}(t) = \mathbf{r}(\mathbf{E}, \mathbf{q}). \end{aligned} \tag{1}$$

$\mathbf{T}$ is the stress tensor, $\mathbf{E} = (\operatorname{grad}\overrightarrow{u} + \operatorname{grad}^T\overrightarrow{u})$ symbolizes the strain tensor, $\overrightarrow{u}(\overrightarrow{x}, t)$ the displacement vector, t is time, ρ is density and $\overrightarrow{k}$ the acceleration of gravity. $\mathbf{q} \in \mathbb{R}^{n_\mathrm{q}}$ contains all internal variables describing the non-linear hardening behavior of the material. Applying the method of vertical lines, which implies the spatial discretization in space as a first step, yields, for finite element approximations, the following DAE-system:

$$\begin{aligned} &\boldsymbol{g}(t, \boldsymbol{u}, \boldsymbol{q}) = \boldsymbol{0}, \\ &\dot{\boldsymbol{q}}(t) = \boldsymbol{r}(t, \boldsymbol{u}, \boldsymbol{q}), \end{aligned} \tag{2}$$

with the unknown nodal displacements $\boldsymbol{u}(t) \in \mathbb{R}^{n_\mathrm{u}}$ and the unknown internal variables $\boldsymbol{q}(t) \in \mathbb{R}^{n_\mathrm{Q}}$ of all spatial integration points, see [9, 10, 38]. Frequently, the initial conditions are $\boldsymbol{u}(0) = \boldsymbol{0}$ and $\boldsymbol{q}(0) = \boldsymbol{q}_0$ at time $t = 0$, where the DAE-system has to have consistent initial conditions $\boldsymbol{g}(0, \boldsymbol{0}, \boldsymbol{q}_0) = \boldsymbol{0}$. Equation $(2)_1$ describes the discretized weak form (principle of virtual displacements),

$$\boldsymbol{g}(t, \boldsymbol{u}, \boldsymbol{q}) = \sum_{e=1}^{n_\mathrm{el}} \boldsymbol{Z}^{eT} \left\{ \sum_{j=1}^{n^e_\mathrm{GP}} w^{e(j)} \mathbf{B}^{e(j)^T} \mathbf{h}(\mathbf{E}^{e(j)}, \mathbf{q}^{e(j)}) \det \mathbf{J}^{e(j)} \right\} - \overline{\boldsymbol{p}}(t), \tag{3}$$

where $\boldsymbol{Z}^e \in \mathbb{R}^{n^e_\mathrm{u} \times n_\mathrm{u}}$ symbolizes the assembling matrix, which is not programmed but defined in order to describe the assembling procedure. n^e_u is the number of element nodal displacement degrees of freedom, $w^{e(j)}$ the weighting factors of the Gauss-integration in an element, n^e_GP are the number of Gauss-points within one element, and $\mathbf{B}^{e(j)} \in \mathbb{R}^{6 \times n^e_\mathrm{u}}$ defines the strain-displacement matrix of element e evaluated at the j-th Gauss-point, $j = 1, \ldots, n^e_\mathrm{GP}$. The latter has the local coordinates $\boldsymbol{\xi}^{(j)} \in \mathbb{R}^3$ (we only consider three-dimensional continua). Furthermore, $\mathbf{J}^{e(j)} \in \mathbb{R}^{3 \times 3}$ symbolizes the Jacobi-matrix of the coordinate transformation between reference element coordinates and global coordinates, and $\overline{\boldsymbol{p}}(t) \in \mathbb{R}^{n_\mathrm{u}}$ defines the given equivalent nodal force vector. The symmetric part of the stress tensor $(1)_2$ is transferred into a vector $\mathbf{T} \in \mathbb{R}^6$, $\mathbf{T} = \mathbf{h}(\mathbf{E}^{e(j)}, \mathbf{q}^{e(j)})$, which is evaluated at Gauss-point $\boldsymbol{\xi}^{(j)}$, and depends on the strains

$$\mathbf{E}^{e(j)} = \mathbf{B}^{e(j)} \mathbf{u}^e = \mathbf{B}^{e(j)} \left\{ \boldsymbol{Z}^e \boldsymbol{u} + \overline{\boldsymbol{Z}}^e \overline{\boldsymbol{u}}(t) \right\}, \qquad \mathbf{E}^{e(j)} \in \mathbb{R}^6 \tag{4}$$

with the element nodal displacements $\mathbf{u}^e \in \mathbb{R}^{n^e_\mathrm{u}}$ depending on the unknown and known nodal displacements $\boldsymbol{u} \in \mathbb{R}^{n_\mathrm{u}}$ and $\overline{\boldsymbol{u}} \in \mathbb{R}^{n_\mathrm{p}}$, $\mathbf{u}^e = \boldsymbol{Z}^e \boldsymbol{u} + \overline{\boldsymbol{Z}}^e \overline{\boldsymbol{u}}(t)$. $\overline{\boldsymbol{u}}(t)$ are given functions of the time t. For more information regarding the chosen notation, see [32]. The key point of the interpretation of finite elements applied to constitutive models with internal variables as solution procedure to solve DAEs stems from a formal assembling of all internal variables $\mathbf{q}^{e(j)} \in \mathbb{R}^{n_\mathrm{q}}$ into a large vector

$$\boldsymbol{q}(t) = \sum_{e=1}^{n_\mathrm{el}} \sum_{j=1}^{n^e_\mathrm{GP}} \boldsymbol{Z}^{e(j)^T}_\mathrm{q} \mathbf{q}^{e(j)}(t), \qquad \text{or} \quad \mathbf{q}^{e(j)}(t) = \boldsymbol{Z}^{e(j)}_\mathrm{q} \boldsymbol{q}(t), \tag{5}$$

which also implies the assembling of all evolution equations of all internal variables $(1)_3$,

$$\dot{\boldsymbol{q}}(t) = \sum_{e=1}^{n_\mathrm{el}} \sum_{j=1}^{n^e_\mathrm{GP}} \boldsymbol{Z}^{e(j)^T}_\mathrm{q} \dot{\mathbf{q}}^{e(j)}(t) = \boldsymbol{r}(t, \boldsymbol{u}, \boldsymbol{q}) \qquad \text{with} \qquad \boldsymbol{r}(t, \boldsymbol{u}, \boldsymbol{q}) = \sum_{e=1}^{n_\mathrm{el}} \sum_{j=1}^{n^e_\mathrm{GP}} \boldsymbol{Z}^{e(j)^T}_\mathrm{q} \mathbf{r}(\mathbf{E}^{e(j)}, \mathbf{q}^{e(j)}). \tag{6}$$

The internal variables assigned to a Gauss-point are assembled by the matrix $\boldsymbol{Z}^{e(j)}_\mathrm{q} \in \mathbb{R}^{n_\mathrm{q} \times n_\mathrm{Q}}$ into $\boldsymbol{q} \in \mathbb{R}^{n_\mathrm{Q}}$ (even here, $\boldsymbol{Z}^{e(j)}_\mathrm{q}$ is not programmed but describes the storage/extracting scheme into/out of the storage $\boldsymbol{q}$). Since the constitutive equations $(1)_{2,3}$ depend on the material parameters $\boldsymbol{\kappa} \in \mathbb{R}^{n_\kappa}$, the entire DAE-system (2) and the solution vectors $\boldsymbol{u}(\boldsymbol{\kappa})$ as well as $\boldsymbol{q}(\boldsymbol{\kappa})$ depend on the parameters $\boldsymbol{\kappa}$,

$$\begin{aligned} &\boldsymbol{g}(t, \boldsymbol{u}(\boldsymbol{\kappa}), \boldsymbol{q}(\boldsymbol{\kappa}), \boldsymbol{\kappa}) = \boldsymbol{0}, \\ &\dot{\boldsymbol{q}}(t, \boldsymbol{\kappa}) = \boldsymbol{r}(t, \boldsymbol{u}(\boldsymbol{\kappa}), \boldsymbol{q}(\boldsymbol{\kappa}), \boldsymbol{\kappa}), \end{aligned} \tag{7}$$

where the initial conditions of the internal variables might also depend on the material parameters, $\boldsymbol{q}_0(\boldsymbol{\kappa})$, for example, in isotropic hardening models in elastoplasticity, where the initial yield stress has to be determined. With regard to identifying these material parameters, we assume the following: we have experimental data $\boldsymbol{d}_i \in \mathbb{R}^{n_\mathrm{d}}$ at time t_i, $i = 0, \ldots, n_\mathrm{T}$. The data might be given by digital image correlation systems, see, for instance [3, 35]. First, only a subset of the nodal displacements $\boldsymbol{u} \in \mathbb{R}^{n_\mathrm{u}}$ can be compared, $\widetilde{\boldsymbol{u}} = \widetilde{\boldsymbol{M}} \boldsymbol{u}$, $\widetilde{\boldsymbol{u}} \in \mathbb{R}^{\widetilde{n}_\mathrm{u}}$, i.e., $\widetilde{n}_\mathrm{u} = n_\mathrm{d}$. The matrix $\widetilde{\boldsymbol{M}}$ extracts only the necessary nodal displacements. Second, there must be a spatial interpolation of the experimental data $\boldsymbol{d}_i$ to

places, where the numerical data $\widetilde{\boldsymbol{u}}(t_i)$ is provided (or the other way round). Third, an interpolation of the evaluation time in the experiments and that of the time-discretized DAE-system (2) has to be carried out. For the sake of simplicity, we assume that the evaluation times and the spatial evaluation points are identical.

Let t_n, $n = 0, \ldots, N$, be the evaluation times of the time discretization scheme (and, accordingly, of the experimental data), i.e. $\boldsymbol{d}^T = \{\boldsymbol{d}_0^T \boldsymbol{d}_1^T \ldots \boldsymbol{d}_N^T\}$ forms the experimental data vector and $\boldsymbol{s}^T(\boldsymbol{\kappa}) = \{\widetilde{\boldsymbol{u}}_0^T(\boldsymbol{\kappa})\widetilde{\boldsymbol{u}}_1^T(\boldsymbol{\kappa}) \ldots \widetilde{\boldsymbol{u}}_N^T(\boldsymbol{\kappa})\}$ the simulation data, $\boldsymbol{d} \in \mathbb{R}^{n_\mathrm{D}}$ and $\boldsymbol{s} \in \mathbb{R}^{n_\mathrm{D}}$ with $n_\mathrm{D} = (N+1)n_\mathrm{d}$. In a least-square sense, the square of the residual $\widetilde{\boldsymbol{r}}(\boldsymbol{\kappa}) = \boldsymbol{s}(\boldsymbol{\kappa}) - \boldsymbol{d}$ has to be minimized

$$f(\boldsymbol{\kappa}) = \frac{1}{2}\widetilde{\boldsymbol{r}}^T(\boldsymbol{\kappa})\widetilde{\boldsymbol{r}}(\boldsymbol{\kappa}) = \frac{1}{2}\{\boldsymbol{s}(\boldsymbol{\kappa}) - \boldsymbol{d}\}^T\{\boldsymbol{s}(\boldsymbol{\kappa}) - \boldsymbol{d}\} \rightarrow \min. \tag{8}$$

A necessary condition for a minimum at $\boldsymbol{\kappa} = \boldsymbol{\kappa}^*$ requires a vanishing derivative (find the root of a non-linear system)

$$\boldsymbol{F}(\boldsymbol{\kappa}^*) = \left.\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\kappa}}\right|_{\boldsymbol{\kappa}=\boldsymbol{\kappa}^*} = \boldsymbol{D}^T(\boldsymbol{\kappa}^*)\{\boldsymbol{s}(\boldsymbol{\kappa}^*) - \boldsymbol{d}\} = \boldsymbol{0}, \tag{9}$$

with $\boldsymbol{D}(\boldsymbol{\kappa}) := \mathrm{d}\widetilde{\boldsymbol{r}}(\boldsymbol{\kappa})/\mathrm{d}\boldsymbol{\kappa} = \mathrm{d}\boldsymbol{s}(\boldsymbol{\kappa})/\mathrm{d}\boldsymbol{\kappa}$, $\boldsymbol{D} \in \mathbb{R}^{n_\mathrm{D} \times n_\kappa}$. Gauss-Newton-like methods require the derivatives (sensitivities)

$$\boldsymbol{D}(\boldsymbol{\kappa}) = \frac{\mathrm{d}\widetilde{\boldsymbol{r}}(\boldsymbol{\kappa})}{\mathrm{d}\boldsymbol{\kappa}} = \frac{\mathrm{d}\boldsymbol{s}(\boldsymbol{\kappa})}{\mathrm{d}\boldsymbol{\kappa}} = \begin{bmatrix} \dfrac{\mathrm{d}\widetilde{\boldsymbol{u}}_0(\boldsymbol{\kappa})}{\mathrm{d}\boldsymbol{\kappa}} \\ \vdots \\ \dfrac{\mathrm{d}\widetilde{\boldsymbol{u}}_N(\boldsymbol{\kappa})}{\mathrm{d}\boldsymbol{\kappa}} \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{M}}\dfrac{\mathrm{d}\boldsymbol{u}_0(\boldsymbol{\kappa})}{\mathrm{d}\boldsymbol{\kappa}} \\ \vdots \\ \widetilde{\boldsymbol{M}}\dfrac{\mathrm{d}\boldsymbol{u}_N(\boldsymbol{\kappa})}{\mathrm{d}\boldsymbol{\kappa}} \end{bmatrix}, \tag{10}$$

$\boldsymbol{D} \in \mathbb{R}^{n_\mathrm{D} \times n_\kappa}$, see, for example, [22, 28, 34]. There are three main approaches to determine the sensitivity $\boldsymbol{D}$: *simultaneous sensitivity equations* (SSE), *internal numerical differentiation* (IND), and *external numerical differentiation* (END). In the following, these methods are compared within the context of the finite element method using constitutive equations of evolutionary-type.

2.1. Simultaneous simulation of sensitivities

One possibility to determine the sensitivities (10), is to apply the derivative of the DAE-system (7) with respect to the parameters $\boldsymbol{\kappa}$, see [8, 24], leading to the linear matrix DAE-system

$$\begin{aligned} \boldsymbol{0} &= \boldsymbol{G}_\mathrm{u}\boldsymbol{D}_\mathrm{u} + \boldsymbol{G}_\mathrm{q}\boldsymbol{D}_\mathrm{q} + \boldsymbol{G}_\kappa, \\ \dot{\boldsymbol{D}}_\mathrm{q} &= \boldsymbol{R}_\mathrm{u}\boldsymbol{D}_\mathrm{u} + \boldsymbol{R}_\mathrm{q}\boldsymbol{D}_\mathrm{q} + \boldsymbol{R}_\kappa. \end{aligned} \tag{11}$$

In Eq. $(11)_2$, the time derivative and the derivative with respect to the parameters $\boldsymbol{\kappa}$ are exchanged. In this respect, we have to determine the sensitivities

$$\boldsymbol{D}_\mathrm{u} := \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\kappa}}, \qquad \boldsymbol{D}_\mathrm{u} \in \mathbb{R}^{n_\mathrm{u} \times n_\kappa}, \qquad \boldsymbol{D}_\mathrm{q} := \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{\kappa}}, \qquad \boldsymbol{D}_\mathrm{q} \in \mathbb{R}^{n_\mathrm{Q} \times n_\kappa} \tag{12}$$

by solving DAE-system (11). In Eq. (11), we introduced the abbreviations

$$\boldsymbol{G}_\mathrm{u} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}, \qquad \boldsymbol{G}_\mathrm{q} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}, \qquad \boldsymbol{G}_\kappa = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\kappa}}, \qquad \boldsymbol{R}_\mathrm{u} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}, \qquad \boldsymbol{R}_\mathrm{q} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}}, \qquad \boldsymbol{R}_\kappa = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\kappa}}. \tag{13}$$

Since the initial conditions of the nodal displacements do not depend on the parameters $\boldsymbol{\kappa}$, the initial conditions for the sensitivities are zero as well, $\boldsymbol{D}_\mathrm{u}(0) = \partial \boldsymbol{u}(0, \boldsymbol{\kappa})/\partial \boldsymbol{\kappa} = \boldsymbol{0}$. However, the initial sensitivities of the internal variables must be provided (if they depend on the parameters),

$\boldsymbol{D}_{\mathrm{q}}(0) = \partial \boldsymbol{q}(0,\boldsymbol{\kappa})/\partial \boldsymbol{\kappa} = \mathrm{d}\boldsymbol{q}_0/\mathrm{d}\boldsymbol{\kappa}$. The combined system (2) and (11) contain $n_{\mathrm{u}}+n_{\mathrm{Q}}+(n_{\mathrm{u}}+n_{\mathrm{Q}})\times n_\kappa = (n_{\mathrm{u}}+n_{\mathrm{Q}})(1+n_\kappa)$ equations, where – depending on the spatial discretization – $\boldsymbol{q}$ alone might contain several million entries. Thus, not all approaches are efficient. Both DAE-systems (2) and (11) could be computed with different time integrators. Since the system (11) depends on the results of DAE-system (2) (one-side coupling), we can either use the same time-steps to provide $\boldsymbol{u}$ and $\boldsymbol{q}$ to (11), or an interpolation technique is required to transfer $\boldsymbol{u}$ and $\boldsymbol{q}$ to the evaluation times of the solver, which computes (11). This, however, is out of the scope of this article.

Commonly, we assume that the number of material parameters is small ($n_{\mathrm{q}} < 30$). The question whether these parameters can be identified solely by experiments is not addressed in this presentation. More information on that topic can be found in [13].

If we follow the classical finite element approach, where a Backward-Euler time-integration method is applied to the constitutive equations at Gauss-point level, we have to follow an overall Backward-Euler scheme to solve the DAE-system (2), see [9]. Additionally, we apply the same time integrator to DAE-system (11) leading to

$$\begin{aligned}
&\boldsymbol{g}(t_{n+1},\boldsymbol{u}_{n+1},\boldsymbol{q}_{n+1}) = \boldsymbol{0},\\
&\boldsymbol{I}(t_{n+1},\boldsymbol{u}_{n+1},\boldsymbol{q}_{n+1}) = \boldsymbol{0},\\
&\boldsymbol{G}_{\mathrm{u}n+1}\boldsymbol{D}_{\mathrm{u}n+1} + \boldsymbol{G}_{\mathrm{q}n+1}\boldsymbol{D}_{\mathrm{q}n+1} + \boldsymbol{G}_{\kappa n+1} = \boldsymbol{0},\\
&\boldsymbol{D}_{\mathrm{q}n+1} - \boldsymbol{D}_{\mathrm{q}n} - \Delta t_n\big(\boldsymbol{R}_{\mathrm{u}n+1}\boldsymbol{D}_{\mathrm{u}n+1} + \boldsymbol{R}_{\mathrm{q}n+1}\boldsymbol{D}_{\mathrm{q}n+1} + \boldsymbol{R}_{\kappa n+1}\big) = \boldsymbol{0}.
\end{aligned} \tag{14}$$

Here, we abbreviated Eq. $(14)_2$ by

$$\boldsymbol{I}(t_{n+1},\boldsymbol{u}_{n+1},\boldsymbol{q}_{n+1}) = \boldsymbol{q}_{n+1} - \boldsymbol{q}_n - \Delta t_n \boldsymbol{r}(t_{n+1},\boldsymbol{u}_{n+1},\boldsymbol{q}_{n+1}). \tag{15}$$

The first two non-linear systems, Eqs. $(14)_{1,2}$, are decoupled from Eqs. $(14)_{3,4}$. Thus, specific procedures can be applied. Here, Eqs. $(14)_{1,2}$ must be solved first, and the results have to be inserted into Eqs. $(14)_{3,4}$. The first choice would be to apply the Newton-Raphson method to the coupled system $(14)_{1,2}$. In [12], it is mentioned that it is common to apply the multilevel-Newton

Table 1. Multilevel-Newton algorithm.

Given: starting vector estimation $\boldsymbol{u}^{(0)}$, $\boldsymbol{q}^{(0)}$ and Δt_n, t_{n+1}		
Repeat	$m = 0,\ldots$	
	local (element) level	
	given: $\boldsymbol{u}^{(m)}$	
	local integration step	
	$\boldsymbol{I}(\boldsymbol{u}^{(m)},\boldsymbol{q}^{(m+1)}) = \boldsymbol{0}$	$\rightsquigarrow \boldsymbol{q}^{(m+1)}$
	consistent linearization ($\boldsymbol{y} = \big(\boldsymbol{u}^{(m)},\boldsymbol{q}^{(m+1)}\big)$)	
	$\left.\frac{\partial \boldsymbol{I}}{\partial \boldsymbol{q}}\right\|_{\boldsymbol{y}} \left.\frac{\mathrm{d}\widehat{\boldsymbol{q}}}{\mathrm{d}\boldsymbol{u}}\right\|_{\boldsymbol{y}} = -\left.\frac{\partial \boldsymbol{I}}{\partial \boldsymbol{u}}\right\|_{\boldsymbol{y}}$	$\rightsquigarrow \left.\frac{\mathrm{d}\widehat{\boldsymbol{q}}}{\mathrm{d}\boldsymbol{u}}\right\|_{\boldsymbol{y}}$
	global level	
	solve linear system of equations	
	$\left[\left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right\|_{\boldsymbol{y}} + \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}\right\|_{\boldsymbol{y}} \left.\frac{\mathrm{d}\widehat{\boldsymbol{q}}}{\mathrm{d}\boldsymbol{u}}\right\|_{\boldsymbol{y}}\right]\Delta \boldsymbol{u} = -\boldsymbol{g}(\boldsymbol{u}^{(m)},\boldsymbol{q}^{(m+1)})$	$\rightsquigarrow \Delta \boldsymbol{u}$
	update of global variables	
	$\boldsymbol{u}^{(m+1)} \leftarrow \boldsymbol{u}^{(m)} + \Delta \boldsymbol{u}$	$\rightsquigarrow \boldsymbol{u}^{(m+1)}$
until	the convergence criterion is fulfilled	

algorithm (MLNA) to finite elements if Eq. $(14)_2$ is non-linear in the internal variables $\boldsymbol{q}_{n+1}$. This scheme is based on the implicit function theorem, see [16, 30], leading to the classical structure of implicit finite element programs. In Table 1 the method is assembled, which has a certain appeal if at Gauss-point level the internal variable computation can be solved very efficiently, see examples in [12].

2.2. Internal numerical differentiation

In the case of internal numerical differentiation, the entire time integration step, e.g., in Runge-Kutta-type methods, both the stage quantities and stage derivatives as well as the starting values is assumed to be dependent on the parameters $\boldsymbol{\kappa}$, [5, 34]. The stage quantities and derivatives represent further vectorial values evaluated at the integration points of the time integrator. The starting value is a vector of quantities computed in previous steps, see, for example, [11]. Applying the ε-embedding method, see [11], in combination with the Backward-Euler method to the DAE-system (7) yields

$$\begin{aligned} &\boldsymbol{g}(t_{n+1}, \boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{\kappa}) = \boldsymbol{0}, \\ &\widetilde{\boldsymbol{I}}(t_{n+1}, \boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_n(\boldsymbol{\kappa}), \boldsymbol{\kappa}) = \boldsymbol{0}, \end{aligned} \tag{16}$$

with

$$\widetilde{\boldsymbol{I}}\big(t_{n+1}, \boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_n(\boldsymbol{\kappa}), \boldsymbol{\kappa}\big) = \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}) - \boldsymbol{q}_n(\boldsymbol{\kappa}) - \Delta t_n \boldsymbol{r}\big(t_{n+1}, \boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{\kappa}\big), \tag{17}$$

see Eq. (15). Here, the starting value $\boldsymbol{q}_n$ also depends on the parameters $\boldsymbol{\kappa}$.

2.2.1. Sensitivities based on multilevel-Newton algorithm

The first approach is similar to the ideas presented in [3, 25, 26]. To show this, the implicit function theorem has to be applied, i.e., we assume that a function $\widehat{\boldsymbol{q}}(\boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{\kappa})$ exists, which we insert into the equilibrium equations $(16)_1$,

$$\boldsymbol{g}(\boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \widehat{\boldsymbol{q}}(\boldsymbol{u}(\boldsymbol{\kappa}), \boldsymbol{\kappa}), \boldsymbol{\kappa}) = \boldsymbol{0}, \tag{18}$$

see also [19]. The derivative with respect to the parameters $\boldsymbol{\kappa}$ yields the linear system (after some rearrangements)

$$[\boldsymbol{G}_{\mathrm{u}n+1} + \boldsymbol{G}_{\mathrm{q}n+1}\boldsymbol{Q}_{\mathrm{u}n+1}]\,\boldsymbol{D}_{\mathrm{u}n+1} = -\boldsymbol{G}_{\kappa n+1} - \boldsymbol{G}_{\mathrm{q}n+1}\boldsymbol{Q}_{\kappa n+1}, \tag{19}$$

with

$$\boldsymbol{Q}_{\mathrm{u}n+1} = \frac{\partial \widehat{\boldsymbol{q}}}{\partial \boldsymbol{u}_{n+1}} \qquad \text{and} \qquad \boldsymbol{Q}_{\kappa n+1} = \frac{\partial \widehat{\boldsymbol{q}}}{\partial \boldsymbol{\kappa}} \tag{20}$$

where the coefficient matrix on the left-hand side is exactly the same as in the MLNA (global level), see Table 1, because the tangential stiffness matrix reads explicitly

$$\boldsymbol{G}_{\mathrm{u}n+1} + \boldsymbol{G}_{\mathrm{q}n+1}\boldsymbol{Q}_{\mathrm{u}n+1} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}_{n+1}} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}_{n+1}} \frac{\partial \widehat{\boldsymbol{q}}}{\partial \boldsymbol{u}_{n+1}}.$$

In other words, if the MLNA fulfills the convergence criterion, i.e., $\boldsymbol{g} = \boldsymbol{0}$ and $\boldsymbol{I} = \boldsymbol{0}$, the last tangential stiffness matrix can be used to solve the linear system (19) with several right-hand sides. In the case of an LU-decomposition – in a finite element simulation using a direct solver – this implies only additional back-substitutions. The required quantity, namely $\boldsymbol{Q}_{\kappa n+1}$, see Eq. $(20)_2$, can be provided

on the so-called "local level", since the internal variables are only formally assembled, i.e., they are decoupled from spatial integration point to spatial integration point. In order to show this, we insert $\widehat{\boldsymbol{q}}(\boldsymbol{u}_{n+1}(\boldsymbol{\kappa}),\boldsymbol{\kappa})$ into the integration step of the internal variables $(14)_2$ leading to

$$\widetilde{\boldsymbol{I}}(\boldsymbol{u}_{n+1}(\boldsymbol{\kappa}),\widehat{\boldsymbol{q}}(\boldsymbol{u}_{n+1}(\boldsymbol{\kappa}),\boldsymbol{\kappa}),\boldsymbol{q}_n(\boldsymbol{\kappa}),\boldsymbol{\kappa}) = \boldsymbol{0}. \tag{21}$$

The derivative with respect to the parameters yields

$$\boldsymbol{L}_{\mathrm{u}n+1}\boldsymbol{D}_{\mathrm{u}n+1} + \boldsymbol{L}_{\mathrm{q}n+1}\boldsymbol{Q}_{\mathrm{u}n+1}\boldsymbol{D}_{\mathrm{u}n+1} + \boldsymbol{L}_{\mathrm{q}n+1}\boldsymbol{Q}_{\kappa n+1} + \boldsymbol{L}_{\mathrm{q}n}\boldsymbol{Q}_{\kappa n} + \boldsymbol{L}_{\kappa n+1} = \boldsymbol{0},$$

with

$$\boldsymbol{L}_{\mathrm{u}n+1} = \frac{\partial\widetilde{\boldsymbol{I}}}{\partial\boldsymbol{u}_{n+1}}, \qquad \boldsymbol{L}_{\mathrm{q}n+1} = \frac{\partial\widetilde{\boldsymbol{I}}}{\partial\boldsymbol{q}_{n+1}}, \qquad \boldsymbol{L}_{\mathrm{q}n} = \frac{\partial\widetilde{\boldsymbol{I}}}{\partial\boldsymbol{q}_{n}} = -\boldsymbol{1}, \qquad \text{and} \qquad \boldsymbol{L}_{\kappa n+1} = \frac{\partial\widetilde{\boldsymbol{I}}}{\partial\boldsymbol{\kappa}}. \tag{22}$$

Since

$$\boldsymbol{L}_{\mathrm{u}n+1} + \boldsymbol{L}_{\mathrm{q}n+1}\boldsymbol{Q}_{\mathrm{u}n+1} = \boldsymbol{0}, \tag{23}$$

see local level of MLNA of Table 1, we arrive at the linear system

$$\boldsymbol{L}_{\mathrm{q}n+1}\boldsymbol{Q}_{\kappa n+1} = \boldsymbol{Q}_{\kappa n} - \boldsymbol{L}_{\kappa n+1}. \tag{24}$$

If we insert Eqs. (23) and (24) into the system (19), we arrive at

$$\left[\boldsymbol{G}_{\mathrm{u}n+1} + \boldsymbol{G}_{\mathrm{q}n+1}\boldsymbol{L}^{-1}_{\mathrm{q}n+1}\boldsymbol{L}_{\mathrm{u}n+1}\right]\boldsymbol{D}_{\mathrm{u}n+1} = -\boldsymbol{G}_{\kappa n+1} + \boldsymbol{G}_{\mathrm{q}n+1}\boldsymbol{L}^{-1}_{\mathrm{q}n+1}\left[-\boldsymbol{Q}_{\kappa n} + \boldsymbol{L}_{\kappa n+1}\right]. \tag{25}$$

Here, the matrix $\boldsymbol{L}_{\mathrm{q}n+1} = \partial\widetilde{\boldsymbol{I}}/\partial\boldsymbol{q}_{n+1}$ is already known from the last iteration (local level in the MLNA). Since the internal variables $\boldsymbol{q}_{n+1}$ were only assembled formally, see Eq. (5), small linear systems with n_κ right-hand sides at Gauss-point level of dimension n_q have to be solved,

$$\frac{\partial\widetilde{\mathbf{I}}}{\partial\mathbf{q}_{n+1}}\frac{\partial\widehat{\mathbf{q}}}{\partial\boldsymbol{\kappa}} = -\frac{\mathrm{d}\mathbf{q}_n}{\mathrm{d}\boldsymbol{\kappa}} - \frac{\partial\widetilde{\mathbf{I}}}{\partial\boldsymbol{\kappa}} \tag{26}$$

with $\widetilde{\mathbf{I}} \in \mathbb{R}^{n_\mathrm{q}}$ and $\mathbf{q}_n \in \mathbb{R}^{n_\mathrm{q}}$ evaluated at time t_{n+1} for element e and Gauss-point $\boldsymbol{\xi}^{(j)}$.

Now, the results of the aforementioned discussion become clear: the finite element approach of computing the sensitivities on Gauss-point level (24), $\boldsymbol{D}_{\mathrm{q}n+1} = \partial\widehat{\boldsymbol{q}}/\partial\boldsymbol{\kappa}$, and solving the linear system with global tangential stiffness matrix (19) at the end of the MLNA yields the sensitivities $\boldsymbol{D}_{\mathrm{u}n+1}$, which are required in Eq. (10). The sensitivities $\partial\widehat{\mathbf{q}}/\partial\boldsymbol{\kappa}$ computed in Eq. (26) must be stored for the next time-step, in which they represent the quantity $\mathrm{d}\mathbf{q}_n/\mathrm{d}\boldsymbol{\kappa}$ for the next time-step. The total amount of storage is $n_\mathrm{el} \times n^e_\mathrm{GP} \times n_\mathrm{q} \times n_\kappa$.

2.2.2. Sensitivities based on chain rule

Alternatively, we can also apply the chain rule to system (16), since the equations have to be fulfilled for all $\boldsymbol{\kappa}$,

$$\begin{aligned}
&\boldsymbol{G}_{\mathrm{u}n+1}\boldsymbol{D}_{\mathrm{u}n+1} + \boldsymbol{G}_{\mathrm{q}n+1}\boldsymbol{D}_{\mathrm{q}n+1} + \boldsymbol{G}_{\kappa n+1} = \boldsymbol{0},\\
&\boldsymbol{L}_{\mathrm{u}n+1}\boldsymbol{D}_{\mathrm{u}n+1} + \boldsymbol{L}_{\mathrm{q}n+1}\boldsymbol{D}_{\mathrm{q}n+1} + \boldsymbol{L}_{\mathrm{q}n}\boldsymbol{D}_{\mathrm{q}n} + \boldsymbol{L}_{\kappa n+1} = \boldsymbol{0}.
\end{aligned} \tag{27}$$

If we eliminate the sensitivities $\boldsymbol{D}_{\mathrm{q}n+1}$ with respect to the internal variables from Eq. $(27)_2$, and exploit Eq. $(22)_3$,

$$\boldsymbol{D}_{\mathrm{q}n+1} = -\boldsymbol{L}^{-1}_{\mathrm{q}n+1}\left[\boldsymbol{L}_{\mathrm{u}n+1}\boldsymbol{D}_{\mathrm{u}n+1} - \boldsymbol{L}_{\mathrm{q}n}\boldsymbol{D}_{\mathrm{q}n} + \boldsymbol{L}_{\kappa n+1}\right], \tag{28}$$

and insert this expression into Eq. $(27)_1$, we obtain, after rearrangement, the linear system

$$\left[\boldsymbol{G}_{\mathrm{u}n+1} - \boldsymbol{G}_{\mathrm{q}n+1}\boldsymbol{L}^{-1}_{\mathrm{q}n+1}\boldsymbol{L}_{\mathrm{u}n+1}\right]\boldsymbol{D}_{\mathrm{u}n+1} = -\boldsymbol{G}_{\kappa n+1} + \boldsymbol{G}_{\mathrm{q}n+1}\boldsymbol{L}^{-1}_{\mathrm{q}n+1}\left[-\boldsymbol{D}_{\mathrm{q}n} + \boldsymbol{L}_{\kappa n+1}\right]. \tag{29}$$

This expression determines the sensitivities $\boldsymbol{D}_{\mathrm{u}n+1}$ and is equivalent to Eq. (25), i.e., both approaches are identical (here, $\boldsymbol{Q}_{\kappa n} = \boldsymbol{D}_{\mathrm{q}n}$ holds).

2.3. External numerical differentiation

In solid mechanics, it is more common to apply external numerical differentiation using numerical differentiation, particularly a forward differential formula. In this regard, the sensitivities (10) are approximated by

$$\frac{\mathrm{d}\widetilde{\boldsymbol{u}}_{n+1}}{\mathrm{d}\boldsymbol{\kappa}} \approx \sum_{i=1}^{n_\mathrm{d}} \sum_{j=1}^{n_\kappa} \frac{\widetilde{\boldsymbol{u}}_{n+1}(\boldsymbol{\kappa} + \Delta\kappa_j \overline{\boldsymbol{e}}_j) - \widetilde{\boldsymbol{u}}_{n+1}(\boldsymbol{\kappa})}{\Delta\kappa_j} \boldsymbol{e}_i \overline{\boldsymbol{e}}_j^T, \tag{30}$$

with the "unit vectors" $\boldsymbol{e}_i \in \mathbb{R}^{n_\mathrm{d}}$ (all entries are zero except for one having a 1 in row i) and $\overline{\boldsymbol{e}}_j \in \mathbb{R}^{n_\kappa}$. The vectors $\widetilde{\boldsymbol{u}}_{n+1}(\boldsymbol{\kappa})$ are determined by solving the DAE-system $(14)_{1,2}$. Here, the results $\boldsymbol{u}_{n+1}$, $n = 0, \ldots, N-1$, must be stored with sufficient accuracy. Additionally, the entire finite element program has to be run again n_κ-times with perturbed material parameters $\boldsymbol{\kappa} + \Delta\kappa_j \overline{\boldsymbol{e}}_j$, $j = 1, \ldots, n_\kappa$. Even here the resulting vectors of the required nodal displacements $\widetilde{\boldsymbol{u}}_{n+1}(\boldsymbol{\kappa} + \Delta\kappa_j \overline{\boldsymbol{e}}_j)$ have to be stored. If high accuracy requirements are necessary, the external numerical differentiation might lead to an insufficient iteration scheme to obtain a local minimum.

2.4. Equivalence of SSE and IND

We can show that – when applying the same time integrator, in this case the Backward-Euler method, to Eqs. (2) and (11) – the non-linear system (14) (SSE) is equivalent to IND presented in Subsec. 2.2. To this end, Eq. $(14)_4$ is rearranged, leading to

$$[\boldsymbol{1} - \Delta t_n \boldsymbol{R}_{\mathrm{q}n+1}]\, \boldsymbol{D}_{\mathrm{q}n+1} = \boldsymbol{D}_{\mathrm{q}n}\big[\boldsymbol{R}_{\mathrm{u}n+1}\boldsymbol{D}_{\mathrm{u}n+1} + \boldsymbol{R}_{\kappa n+1}\big]. \tag{31}$$

Since

$$\boldsymbol{L}_{\mathrm{q}n+1} = \boldsymbol{1} - \Delta t_n \boldsymbol{R}_{\mathrm{q}n+1}, \quad \boldsymbol{D}_{\mathrm{q}n} = \boldsymbol{Q}_{\kappa n}, \quad \boldsymbol{L}_{\mathrm{u}n+1} = -\Delta t_n \boldsymbol{R}_{\mathrm{u}n+1}, \quad \text{and} \quad \boldsymbol{L}_{\kappa n+1} = -\Delta t_n \boldsymbol{R}_{\kappa n+1}, \tag{32}$$

hold, Eq. (31) is identical to Eq. (29). Thus, all the schemes mentioned above are equivalent.

3. Sensitivities in reaction force computations

To the best knowledge of the author, there is no literature on sensitivity determination in reaction force computations. Our focus, however, lies in clear presentation of the displacement and reaction force sensitivities in finite element computations, since forces are often measured in experiments as well, and, accordingly, commonly occur in the parameter identification process. Thus, we refer to [14], where the reaction force computation is explained in detail. Classical formulations, which are based on the principle of virtual displacements, do not explicitly contain reaction forces. In this context, the Lagrange-multiplier method is the method of choice. It leads to the DAE-system

$$\overline{\boldsymbol{F}}(t, \boldsymbol{y}(t), \dot{\boldsymbol{y}}(t)) := \left\{ \begin{array}{c} \boldsymbol{g}_\mathrm{a}(t, \boldsymbol{u}_\mathrm{a}(t), \boldsymbol{q}(t)) + \boldsymbol{M}\boldsymbol{\lambda}(t) \\ \boldsymbol{C}_c(t, \boldsymbol{u}_\mathrm{a}(t)) \\ \dot{\boldsymbol{q}}(t) - \boldsymbol{r}(\boldsymbol{u}_\mathrm{a}(t), \boldsymbol{q}(t)) \end{array} \right\} = \boldsymbol{0} \tag{33}$$

with

$$\boldsymbol{y}(t) := \left\{ \begin{array}{c} \boldsymbol{u}_\mathrm{a}(t) \\ \boldsymbol{\lambda}(t) \\ \boldsymbol{q}(t) \end{array} \right\} \quad \text{and the initial conditions} \quad \boldsymbol{y}(t_0) := \left\{ \begin{array}{c} \boldsymbol{u}_\mathrm{a}(t_0) \\ \boldsymbol{\lambda}(t_0) \\ \boldsymbol{q}(t_0) \end{array} \right\} = \left\{ \begin{array}{c} \boldsymbol{u}_{\mathrm{a}0} \\ \boldsymbol{\lambda}_0 \\ \boldsymbol{q}_0 \end{array} \right\} := \boldsymbol{y}_0. \tag{34}$$

In the following, the vector of all nodal displacements $\boldsymbol{u}_{\mathrm{a}} \in \mathbb{R}^{n_{\mathrm{u}}+n_{\mathrm{p}}}$, which is separated into degrees of freedom that are unknown, $\boldsymbol{u} \in \mathbb{R}^{n_{\mathrm{u}}}$, and degrees of freedom where displacements are prescribed as $\widehat{\boldsymbol{u}} \in \mathbb{R}^{n_{\mathrm{p}}}$, $\boldsymbol{u}_{\mathrm{a}}^T = \{\boldsymbol{u}^T, \widehat{\boldsymbol{u}}^T\}$ represents the unknown. The constraint condition is given by

$$\boldsymbol{C}_{\mathrm{c}}(\boldsymbol{u}_{\mathrm{a}}(\boldsymbol{\kappa})) = \widehat{\boldsymbol{u}}(\boldsymbol{\kappa}) - \overline{\boldsymbol{u}} = \boldsymbol{M}^T \boldsymbol{u}_{\mathrm{a}}(\boldsymbol{\kappa}) - \overline{\boldsymbol{u}} = \boldsymbol{0}, \qquad \text{with} \qquad \boldsymbol{M} = \begin{bmatrix} \boldsymbol{0}_{n_{\mathrm{u}} \times n_{\mathrm{p}}} \\ \boldsymbol{1}_{n_{\mathrm{p}}} \end{bmatrix}. \tag{35}$$

$\boldsymbol{M} \in \mathbb{R}^{(n_{\mathrm{u}}+n_{\mathrm{p}}) \times n_{\mathrm{p}}}$ assigns all unknown nodal displacements $\boldsymbol{u}_{\mathrm{a}}$ to the known displacements $\overline{\boldsymbol{u}} \in \mathbb{R}^{n_{\mathrm{p}}}$.

The algebraic part of the DAE-system also contains

$$\boldsymbol{g}_{\mathrm{a}}(t, \boldsymbol{u}_{\mathrm{a}}, \boldsymbol{q}) = \begin{Bmatrix} \boldsymbol{g}(\boldsymbol{y}) \\ \overline{\boldsymbol{g}}(\boldsymbol{y}) \end{Bmatrix} = \begin{Bmatrix} \displaystyle\sum_{e=1}^{n_{\mathrm{el}}} \boldsymbol{Z}^{eT} \left\{ \sum_{j=1}^{n_{\mathrm{GP}}^e} w^{e(j)} \mathbf{B}^{e(j)^T} \, \mathbf{h}\left(\mathbf{E}^{e(j)}, \mathbf{q}^{e(j)}\right) \det \mathbf{J}^{e(j)} \right\} - \overline{\boldsymbol{p}}(t) \\ \displaystyle\sum_{e=1}^{n_{\mathrm{el}}} \overline{\boldsymbol{Z}}^{eT} \left\{ \sum_{j=1}^{n_{\mathrm{GP}}^e} w^{e(j)} \mathbf{B}^{e(j)^T} \, \mathbf{h}\left(\mathbf{E}^{e(j)}, \mathbf{q}^{e(j)}\right) \det \mathbf{J}^{e(j)} \right\} \end{Bmatrix}, \tag{36}$$

where the matrices $\boldsymbol{Z}^e \in \mathbb{R}^{n_{\mathrm{u}}^e \times n_{\mathrm{u}}}$ and $\overline{\boldsymbol{Z}}^e \in \mathbb{R}^{n_{\mathrm{u}}^e \times n_{\mathrm{p}}}$, see Eq. (4), assemble the element contributions into a large vector $\boldsymbol{g}_{\mathrm{a}} \in \mathbb{R}^{n_{\mathrm{u}}+n_{\mathrm{p}}}$. The differential part of the DAE-system given by the evolution equations of the internal variables is not affected.

Applying the Backward-Euler method yields

$$\begin{aligned} &\boldsymbol{g}\,(t_{n+1}, \boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \widehat{\boldsymbol{u}}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{\kappa}) = \boldsymbol{0}, \\ &\overline{\boldsymbol{g}}\left(\boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \widehat{\boldsymbol{u}}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{\kappa}\right) + \boldsymbol{\lambda}_{n+1}(\boldsymbol{\kappa}) = \boldsymbol{0}, \\ &\boldsymbol{C}_c\left(t_{n+1}, \widehat{\boldsymbol{u}}_{n+1}(\boldsymbol{\kappa})\right) = \boldsymbol{0}, \\ &\boldsymbol{I}\left(\boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \widehat{\boldsymbol{u}}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_n(\boldsymbol{\kappa}), \boldsymbol{\kappa}\right) = \boldsymbol{0}. \end{aligned} \tag{37}$$

Since $\overline{\boldsymbol{u}} \in \mathbb{R}^{n_{\mathrm{p}}}$ are the prescribed and known displacements, $\widehat{\boldsymbol{u}}$ cannot be dependent on the material parameters $\boldsymbol{\kappa}$ if a solution is found (fulfillment of the constraint). As a result of applying the Backward-Euler/MLNA procedure, where the constraints (algebraic part of DAE-system) are fulfilled after each time-step, the system (37) degenerates to

$$\begin{aligned} &\boldsymbol{g}\,(t_{n+1}, \boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{\kappa}) = \boldsymbol{0}, \\ &\overline{\boldsymbol{g}}\,(t_{n+1}, \boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{\kappa}) + \boldsymbol{\lambda}_{n+1}(\boldsymbol{\kappa}) = \boldsymbol{0}, \\ &\boldsymbol{I}\,(\boldsymbol{u}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_{n+1}(\boldsymbol{\kappa}), \boldsymbol{q}_n(\boldsymbol{\kappa}), \boldsymbol{\kappa}) = \boldsymbol{0}, \end{aligned} \tag{38}$$

i.e., we observe, see Eqs. $(38)_1$ and $(38)_3$, the same system as in (16). Additionally, an equation for the Lagrange multiplier (reaction force) is provided, Eq. $(38)_2$. Based on Eqs. $(38_1$ and $(38)_3$, it is possible to determine the sensitivities $\boldsymbol{D}_{\mathrm{u}n+1}$ and $\boldsymbol{D}_{\mathrm{q}n+1}$ as before, and they can then be inserted into the sensitivity of the Lagrange-multiplier equation

$$\boldsymbol{D}_{\lambda n+1} = -\overline{\boldsymbol{G}}_{\mathrm{u}n+1} \boldsymbol{D}_{\mathrm{u}n+1} - \overline{\boldsymbol{G}}_{\mathrm{q}n+1} \boldsymbol{D}_{\mathrm{q}n+1} - \overline{\boldsymbol{G}}_{\kappa n+1} \tag{39}$$

with

$$\boldsymbol{D}_{\lambda n+1} = \frac{\mathrm{d}\boldsymbol{\lambda}_{n+1}}{\mathrm{d}\boldsymbol{\kappa}} \in \mathbb{R}^{n_{\mathrm{p}} \times n_{\kappa}}, \qquad \overline{\boldsymbol{G}}_{\mathrm{u}n+1} = \frac{\partial \overline{\boldsymbol{g}}}{\partial \boldsymbol{u}_{n+1}} \in \mathbb{R}^{n_{\mathrm{p}} \times n_{\mathrm{u}}},$$

$$\overline{\boldsymbol{G}}_{\mathrm{q}n+1} = \frac{\partial \overline{\boldsymbol{g}}}{\partial \boldsymbol{q}_{n+1}} \in \mathbb{R}^{n_{\mathrm{p}} \times n_{\mathrm{Q}}}, \qquad \overline{\boldsymbol{G}}_{\kappa n+1} = \frac{\partial \overline{\boldsymbol{g}}}{\partial \boldsymbol{\kappa}} \in \mathbb{R}^{n_{\mathrm{p}} \times n_{\kappa}}.$$

If we insert the sensitivity (28), we arrive at

$$\boldsymbol{D}_{\lambda n+1} = -\Big[\overline{\boldsymbol{G}}_{\mathrm{u}n+1} - \overline{\boldsymbol{G}}_{\mathrm{q}n+1}\boldsymbol{L}^{-1}_{\mathrm{q}n+1}\boldsymbol{L}_{\mathrm{u}n+1}\Big]\boldsymbol{D}_{\mathrm{u}n+1} + \overline{\boldsymbol{G}}_{\mathrm{q}n+1}\boldsymbol{L}^{-1}_{\mathrm{q}n+1}\big[\boldsymbol{D}_{\mathrm{q}n} - \boldsymbol{L}_{\kappa n+1}\big] - \overline{\boldsymbol{G}}_{\kappa n+1}, \tag{40}$$

where the first large matrix represents a particular tangential stiffness matrix,

$$\overline{\boldsymbol{K}}_{\mathrm{pu}} = \sum_{e=1}^{n_{\mathrm{el}}} \overline{\boldsymbol{Z}}^{eT}\mathbf{k}^{e}\boldsymbol{Z}^{e}, \tag{41}$$

compiled by parts of the element stiffness matrix $\mathbf{k}^e$. In other words, there is essentially no additional work to provide the sensitivities for reactional forces since the sensitivities of the displacements are computed before.

4. Conclusions

In this paper, we were able to show that – in the context of finite elements and where the constitutive equations are of evolutionary type – material parameter identification procedures, which require the sensitivity matrix, can be interpreted as internal numerical differentiation scheme developed in numerical mathematics. Moreover, the internal numerical differentiation is equivalent to the simultaneous simulation of sensitivities for the same time integrator. We were also able to show that internal numerical differentiation can be interpreted by two schemes: one that is based on the implicit function theorem (close to the multilevel-Newton algorithm), and another where static condensation (Newton-Schur step) is exploited. The entire formal procedure proposed becomes possible because the finite element discretization is interpreted as the solution of differential-algebraic equations after the spatial discretization. As the first approach, we chose the Backward-Euler method in combination with the multilevel-Newton algorithm (as in the classical implicit finite elements). A side-product is also the sensitivity of the reaction forces, which requires only additional matrix-matrix products of quantities already provided in the sensitivity analysis. Thus, we obtained a closed and systematic framework – making it possible to also consider other high-order time integration schemes applied to DAE-systems.

Acknowledgements

The author would like to gratefully acknowledge the financial support provided by the German Research Foundation DFG (grant no. HA 2024/12-1) supporting the original idea of investigating identification on the basis of full field measurement data.

References

[1] W.P. Adamczyk, T. Kruczek, G. Moskal, R.A. Bialecki. Nondestructive technique of measuring heat conductivity of thermal barrier coatings. *International Journal of Heat and Mass Transfer*, **111**: 442–450, 2017.

[2] W.P. Adamczyk, Z. Ostrowski. Retrieving thermal conductivity of the solid sample using reduced order model inverse approach. *International Journal of Numerical Methods for Heat & Fluid Flow*, **27**: 729–739, 2017.

[3] K. Andresen, S. Dannemeyer, H. Friebe, R. Mahnken, R. Ritter, E. Stein. Parameteridentifikation für ein plastisches Stoffgesetz mit FE-Methoden und Rasterverfahren [in German]. *Bauingenieur*, **71**: 21–31, 1996.

[4] U. Benedix, U.-J. Görke, R. Kreißig, S. Kretzschmar. Local and global analysis of inhomogeneous displacement fields for the identification of material parameters. In S.V. Hoa, W.P. De Wilde, W.R. Blain [Eds.], *Computer Methods in Composite Materials VI (CADCOMP 98)*, pp. 159–168, 1998.

[5] H.G. Bock. Recent advances in parameter identification techniques for ODEs. In C.W. Gear, T. Vu, P. Deuflhard, E. Hairer [Eds.], *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, number 2 in Progress in Scientific Computing, pp. 95–121, Birkhäuser, Basel, 1983.

[6] Z. Chen, S. Diebels. Nanoindentation of soft polymers: Modeling, experiments and parameter identification. *Technische Mechanik*, **34**: 166–189, 2014.

[7] S. Cooreman, D. Lecompte, H. Sol, J. Vantomme, D. Debruyne. Elasto-plastic material parameter identification by inverse methods: Calculation of the sensitivity matrix. *International Journal of Solids and Structures*, **44**: 4329–4341, 2007.

[8] A.M. Dunker. The decoupled direct method for calculating sensitivity coefficients in chemical kinetics. *The Journal of Chemical Physics*, 81, 1984.

[9] P. Ellsiepen, S. Hartmann. Remarks on the interpretation of current non-linear finite-element-analyses as differential-algebraic equations. *International Journal for Numerical Methods in Engineering*, **51**: 679–707, 2001.

[10] P. Fritzen. *Numerische Behandlung nichtlinearer Probleme der Elastizitäts- und Plastizitätstheorie*. Doctoral thesis, Department of Mathematics, University of Darmstadt, 1997.

[11] E. Hairer, G. Wanner. *Solving Ordinary Differential Equations II*. Springer, Berlin, 2nd revised edition, 1996.

[12] S. Hartmann. A remark on the application of the Newton-Raphson method in non-linear finite element analysis. *Computational Mechanics*, **36**(2): 100–116, 2005.

[13] S. Hartmann, R.R. Gilbert. Identifiability of material parameters in solid mechanics. *Published online Archive of Applied Mechanics*, 2017.

[14] S. Hartmann, K.J. Quint, A.-W. Hamkar. Displacement control in time-adaptive non-linear finite-element analysis. *ZAMM Journal of Applied Mathematics and Mechanics*, **88**(5): 342–364, 2008.

[15] S. Hartmann, T. Tschöpe, L. Schreiber, P. Haupt. Large deformations of a carbon black-filled rubber. Experiment, optical measurement and parameter identification using finite elements. *European Journal of Mechanics, Series A/Solids*, **22**: 309–324, 2003.

[16] W. Hoyer, J.W. Schmidt. Newton-type decomposition methods for equations arising in network analysis. *ZAMM Zeitschrift für Angewandte Mathematik und Mechanik*, **64**: 397–405, 1984.

[17] N. Huber, C. Tsakmakis. Determination of constitutive properties from spherical indentation data using neural networks, Part I: plasticity with nonlinear and kinematic hardening. *Journal of the Mechanics and Physics of Solids*, **47**: 1589–1607, 1999.

[18] N. Huber, C. Tsakmakis. Determination of constitutive properties from spherical indentation data using neural networks, Part II: the case of pure kinematic hardening in plasticity laws. *Journal of the Mechanics and Physics of Solids*, **47**: 1569–1588, 1999.

[19] S. Krämer. *Einfluss von Unsicherheiten in Materialparametern auf Finite-Elemente Simulationen* [in German]. PhD Thesis, report no. 5/2016, Institute of Applied Mechanics, Clausthal University of Technology, Clausthal-Zellerfeld, 2016.

[20] R. Kreißig. Auswertung inhomogener Verschiebungsfelder zur Identifikation der Parameter elastisch-plastischer Deformationsgesetze [in German]. *Forschung im Ingenieurwesen*, **64**: 99–109, 1998.

[21] R. Kreissig, U. Benedix, U.-J. Goerke. Statistical aspects of the identification of material parameters for elasto-plastic models. *Archive of Applied Mechanics*, **71**: 123–134, 2001.

[22] C.L. Lawson, R.J. Hanson. *Solving least squares problems*. SIAM Society for Industrial and Applied Mathematics, Philadelphia, 1995.

[23] H. Lee, J.H. Lee, G.M. Pharr. A numerical approach to spherical indentation techniques for material property evaluation. *Journal of the Mechanics and Physics of Solids*, **53**: 2037–2069, 2005.

[24] J.R. Leis, M.A. Kramer. The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations. *ACM Transactions on Mathematical Software*, **14**: 45–60, 1988.

[25] R. Mahnken, E. Stein. A unified approach for parameter identification of inelastic material models in the frame of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, **136**: 225–258, 1996.

[26] R. Mahnken, E. Stein. Parameter identification for finite deformation elasto-plasticity in principal directions. *Computer Methods in Applied Mechanics and Engineering*, **147**: 17–39, 1997.

[27] T. Netz, A.-W. Hamkar, S. Hartmann. High-order quasi-static finite element computations in space and time with application to finite strain viscoelasticity. *Computers and Mathematics with Applications*, **66**: 441–459, 2013.

[28] J. Nocedal, S.J. Wright. *Numerical Optimization*. Springer, New York, 1999.

[29] K.J. Quint. *Thermomechanically coupled processes for functionally graded materials: experiments, modelling, and finite element analysis using high-order DIRK-methods*. PhD Thesis, report no. 2/2012, Institute of Applied Mechanics, Clausthal University of Technology, Clausthal-Zellerfeld, 2012.

[30] N.B.G. Rabbat, A.L. Sangiovanni-Vincentelli, H.Y. Hsieh. A multilevel Newton algorithm with macromodeling and latency for the analysis of large-scale nonlinear circuits in the time domain. *IEEE Transactions on Circuits and Systems*, **26**: 733–740, 1979.

[31] G. Rauchs, J. Bardon, D. Georges. Identification of the material parameters of a viscous hyperelastic constitutive law from spherical indentation tests of rubber and validation by tensile tests. *Mechanics of Materials*, **42**: 961–973, 2010.

[32] S. Rothe, P. Erbts, A. Düster, S. Hartmann. Monolithic and partitioned coupling schemes for thermo-viscoplasticity. *Computer Methods in Applied Mechanics and Engineering*, **293**: 375–410, 2015.

[33] G. Scheday. *Theorie und Numerik der Parameteridentifikation von Materialmodellen der finiten Elastizität und Inelastizität auf der Grundlage optischer Feldmessmethoden* [in German]. PhD Thesis, Report No. I-11 (2003), University of Stuttgart (Germany), Institute of Mechanics, 2003.
[34] K. Schittkowski. *Numerical Data Fitting in Dynamical Systems*. Kluwer Academic Publ., Dordrecht, 2002.
[35] S. Schmaltz, K. Willner. Material parameter identification utilizing optical full-field strain measurement and digital image correlation. *Journal of the Japanese Society for Experimental Mechanics*, **13**: s120–s125, 2013.
[36] D.S. Schnur, N. Zabaras. An inverse method for determining elastic material properties and a material interface. *International Journal for Numerical Methods in Engineering*, **33**: 2039–2057, 1992.
[37] P. Shi, I. Babuška. Analysis and computation of a cyclic plasticity model by aid of Ddassl. *Computational Mechanics*, **19**: 380–385, 1997.
[38] J. Wittekindt. *Die numerische Lösung von Anfangs-Randwertproblemen zur Beschreibung inelastischen Werkstoffverhaltens* [in German]. Doctoral Thesis, Department of Mathematics, University of Darmstadt, 1991.