PRACE IPPT · IFTR REPORTS 7/2006

Piotr Kowalczyk

SENSITIVITY ANALYSIS IN FINITE ELEMENT COMPUTATIONS OF ELASTO-PLASTICITY

INSTYTUT PODSTAWOWYCH PROBLEMÓW TECHNIKI
POLSKIEJ AKADEMII NAUK
WARSZAWA 2006

ISSN 0208-5658

Redaktor Naczelny:
doc. dr hab. Zbigniew Kotulski

Recenzent:
prof. dr hab. Włodzimierz Sosnowski

Praca wpłynęła do Redakcji 8 maja 2006 r.

Praca habilitacyjna

Instytut Podstawowych Problemów Techniki PAN
Nakład 100 egz.   Ark. wyd. 15,5
Oddano do druku w sierpniu 2006 r.

Druk i oprawa: Drukarnia Braci Grodzickich, Piaseczno, ul. Geodetów 47a

Contents

Chapter 1

Introduction

Until early nineties of the 20th century, computational engineering problems were commonly understood as tasks in which input parameters (like geometry, material and sizing properties, initial state variables, etc.) were defined and assigned certain numerical values, while the response — space-time distributions of state variables and various functionals thereof — were sought for as solutions of certain initial-boundary value problems in their either analytical forms or space-time discretized at various levels. Highly advanced methods were (and continue to be) developed in which the wide use of computers has turned out to be indispensable for any sophisticated assessment of realistic structural behaviour. Particular difficulties have been encountered and gradually overcome in the case of problems with complex geometries, non-classical (unilateral) boundary conditions and complicated, strongly nonlinear and path-dependent material behaviour.

Among many methodologies of approximate solution of nonlinear problems of mechanics, the finite element method (FEM) is widely recognized as most effective and universal tool. Its success has been proved in industrial practice — it seems hardly imaginable now that design process of any structure would not include extensive numerical simulations of its mechanical behaviour performed with the use of a nonlinear finite element code. Fundamentals and application power of the method, developed since early seventies, have been presented in a number of monographs and textbooks, like e.g. [8, 25, 46, 49, 61, 94, 105, 141], as well as in manuals to specific commercial implementations, e.g. [1]. It is stressed that abilities of the methodology include both statical and dynamical analysis, various types of geometric and material nonlinearities, complex boundary condition types, solid–fluid mechanical interactions, as well as couplings with electrical, thermal and magnetic phenomena.

Plastic phenomena occurring in many structural and geotechnical materials have always belonged to most challenging tasks in computational engineering.

Strongly nonlinear, path- and frequently also rate- and temperature-dependent character of plastic deformation combined with geometric nonlinearities and discontinuous changes in material behaviour at limit points between elastic and inelastic régimes require advanced mathematical models and appropriate computational solution algorithms. Starting from works of Tresca [123], Huber [47] and von Mises [86] who had laid fundamentals for modern theory of plasticity, large progress has been made in both understanding the micromechanical nature of plastic flow and in its mathematical modelling in the macro scale. Formulation of mathematical theory of slip-induced plasticity in metal crystals [5, 43, 45] was accompanied by development of models including large deformations [75, 76, 93], anisotropy [42, 50, 89], porosity and void nucleation [24, 34, 126], nonlinear hardening and cyclic effects [19, 26, 79], and viscoplastic phenomena [2, 39, 102, 103, 127]. For granular materials (rocks, soils) and metallic foams, where plasticity is mainly friction-induced, a variety of non-associated and associated flow models were developed [16, 28, 30, 109]. The citations above refer to only milestone publications and are by no means intended to be exhaustive — extensive reviews of constitutive models of plasticity can be found e.g. in monographs [44, 57, 81, 139].

Elasto-plastic and elasto-viscoplastic constitutive relations are flow equations and are thus formulated in terms of rates of state variables. In numerical applications they require appropriate integration in time. The standard approach employs the ‘elastic predictor–plastic corrector’ scheme, in which trial elastic finite increment of stress is, whenever violating the yield condition, corrected by a plastic counterpart determined from approximate time integration of the flow equation. Such return-mapping algorithms were first proposed in the form of radial return schemes for cylindrical flow surfaces in the space of principal stress components (e.g. Huber–Mises yield condition), cf. [131], and [74] (account for strain hardening). Return mapping strategies for more general plasticity models were developed in e.g. [53, 116] (plane stress state), [110, 112, 114] (general smooth and non-smooth yield surfaces), [87, 104] (for yield surfaces with corners), [14, 15] (for the Cam-clay plasticity model), [103, 130] (for viscoplastic models), and [52] (for granular materials). A critical issue regarding efficiency of iterative solution schemes in such algorithms is determination of the consistent tangent stiffness operator [111, 115, 116]. This frequently appears to be a complex task, e.g., in a number of large-deformation formulations, evaluation of such an operator requires unique differentiation of tensor spectral decomposition routines, including cases of multiple eigenvalues. The monograph [113] contains an exhaustive review of return-mapping algorithms and discussion of related computational issues, including consistent differentiation.

In some practical steady plastic flow applications (forging, extrusion), the constitutive relations do not need to be time-integrated — instead, they are spatially discretized in the rate form and only state variable rates are of interest as the analysis response, cf. [140] for fundamentals and [117] for practical applications of the 'flow approach' in computational plasticity.

In the recent two decades, we have been witnessing emergence of a very fruitful branch in computational mechanics — the parameter *sensitivity analysis* (SA). Mathematical fundamentals of the method were presented in numerous articles and books; let us only mention here milestone monographs [22, 32, 37, 41, 65] and collections [66, 108]. The aim of SA is to find the relationship between parameters defining the system at hand and the system response. The main interest of the analyst is here not just to quantify the response for a given set of input parameters, but also to evaluate influence of possible variations of the parameters on the corresponding change in the response. In practice this means determination of response gradients with respect to the system parameters, although higher order derivatives may also be a subject of interest [27, 38, 40], e.g. in the critical state analysis [90, 91].

Depending on the nature of input parameters and their variations considered in the sensitivity analysis, we may distinguish between different types of SA in mechanical problems. Probably the most frequently encountered one in the literature is the *design sensitivity analysis* (DSA) in which the system input parameters are understood as the *design parameters*, i.e. the quantities whose particular numerical values as well as their changes depend on arbitrary decisions of the designer. Another branch of SA is the *imperfection sensitivity analysis* (ISA) in which variations of the system input parameters are independent of the designer's will and have usually random character. Mathematical essence in both cases is the same — the sensitivity analysis consists in determination of first- or higher-order gradients of response functionals with respect to a certain set of input scalar quantities. Thus, the names are frequently exchanged and the term 'design variation' happens to be used to describe imperfection, while 'design parameters' may as well denote e.g. probability distribution parameters (means, standard deviations, etc.) of random input quantities. In this thesis, 'design parameters' will be referred to as scalar input parameters of any type defining the system under consideration. Consequently, the term DSA will be considered equivalent to SA in general.

The area of applications of sensitivity analysis is wide and still increasing. Computational power of contemporary computer hardware and maturity of mathematical formulations allow to obtain response sensitivity information for large-scale nonlinear mechanical problems as numerically cheap add-on to the primary analysis results. It is thus widely believed that any robust computer simulation of a mechanical system will be considered incomplete as long as it is not complemented with studies on response sensitivity with respect to input parameter variations. Natural areas of applications for DSA are the *parametric 'what-if' studies* at the stage of structural design, and the *gradient-based design optimization* in which the 'best' set of design parameter values is sought for according to certain optimality criteria [37]. *System identification* problems in which the unknown values of design parameters are to be found, given the response values, are another area of applications of DSA, conceptually very similar to the design optimization. ISA may again find its applications in system identification, but it can also be employed in *error analysis*, *reliability-based optimization*, as well as in *stochastic analysis* of the system response.

This study is devoted to methods of sensitivity analysis in computational problems of elasto-plasticity. Evaluation of sensitivity gradients for solutions embedding a so wide spectrum of nonlinear phenomena is a tempting challenge for researchers and, on the other hand, it responds to increasing needs of industry, where elasto-plastic behaviour of materials often plays the crucial role from the point of view of the safety, economy and ergonomy of the final product. Main attention is focused on the analytical methods of sensitivity, i.e. on determination of analytical response gradients with respect to design parameters (contrary to their approximate determination with finite-difference formulae at small design perturbations).

Publications on sensitivity analysis in these classes of problems had been appearing from late 80's. DSA for elasto-plasticity described with the independent deformation theory was discussed in [9, 10]. For path-dependent problems, which include most of the elasto-plastic constitutive theories, Ryu et al. [107] were the first to point out the most important consequence of path-dependence for sensitivity analysis — that sensitivity gradients are path-dependent, too, i.e. at each time instant they depend on the response itself as well as on its sensitivity at previous time instants. Computational aspects of DSA for small-strain elasto-plasticity were then discussed in [13, 73, 77, 85, 92, 95, 106, 128, 132, 135] (rate-independent models), and in [17, 35, 62–65, 67, 129] (rate-dependent models). Reduced dimension (plane stress) formulation was presented in [68]. Formulation applying to a general rate-type constitutive model of soil was discussed in [31]. Important issues specific to the analysed class of problems, discussed in the publications, include the crucial role of the algorithmic constitutive tangent stiffness operator (consistent with the time integration scheme) [31, 62, 64, 67, 68, 128, 129] and the problem of discontinuity of sensitivity gradients [67, 68, 77]. For large deformation elasto-plastic models, sensitivity analysis was discussed in [118] (rigid-plastic approach only), [51, 96, 124, 125] for one-dimensional case (trusses), [21, 59, 65, 72, 78, 134] for 3-d models based on additive elastic-plastic strain decomposition, and finally in [6, 7, 23, 58, 60, 71, 80, 119, 133] for models based on multiplicative elastic-plastic decomposition of the deformation gradient. Formulations of sensitivity analysis in the velocity description, including the flow approach concept, can be found in [3, 4, 29, 54, 65, 82, 136, 137].

The references above include both statical, dynamical and structural stability formulations, applicable in the formalism of the finite as well as the boundary element method. At the level of constitutive formulations there are no significant differences between these areas of applications. Besides, they include formulations of both sizing and shape parameter sensitivity analysis. Despite different nature and way of description of both the types of input parameters, the shape and non-shape sensitivity can be apparently treated in a uniform manner, upon introduction of the reference volume concept [62].

1.1. Scope of the thesis

The scope of the thesis is limited to first-order parameter sensitivity analysis in geometrically linear and nonlinear problems of computational isothermal elasto-plasticity of metals, including elasto-viscoplasticity. The contents is in the main part a compilation of the author's original developments performed in the years 1994–2006 at the Institute of Fundamental Technological Research in Warsaw, partially in cooperation with colleagues from the Department of Computational Science, and published in [64, 65, 67, 68, 71–73, 132, 133]. Attention will be focused on static and quasi-static processes, however, extensions towards dynamic computations will be discussed, too. Issues related to structural stability and vibration analysis, leading to determination of the system's eigenvalues and eigenvectors, will not be considered.

All the initial-boundary problems considered in the thesis will be discussed in terms of the approximate solution techniques based on the finite-element discretization. This will not affect generality of the discussion as long as the sizing and material parameter sensitivity is considered. In the case of shape parameter sensitivity, the notion of the design-independent reference configuration will be strictly related to the definition of the parent configuration of an isoparametric finite element. Applicability of the resulting algorithms within the frame of other approximate solution techniques may thus not be straightforward for this case.

The contents of the thesis is organized as follows. Chapter 2 presents the general idea of sensitivity analysis of nonlinear mechanical systems. It starts with a general formulation of the primary problem in both continuum and discrete variables, and with definition of two basic notions of sensitivity analysis, i.e. design variables and performance functionals. Two fundamental approaches to the sensitivity analysis, continuum and discrete, are distinguished and described, with discussion of their advantages and drawbacks, and with indication of certain intermediate ways. Further, the discrete approach is discussed in detail, with two alternative solution strategies, direct differentiation method (DDM) and adjoint system method (ASM). The analytical, semi-analytical and finite-difference methods of determination of sensitivity gradients are presented and reviewed. Finally, the key issue of elasto-plastic response differentiability is risen and its consequences regarding possible discontinuity of the sensitivity solution are discussed.

Formulation of DSA requires deep and detailed understanding of the primary problem formulation. In Chapter 3, the primary problem of static deformation of an elasto-plastic body is presented in detail. It includes constitutive formulations of elasto-plasticity and elasto-viscoplasticity in both small and large deformations, and general finite element equations of the global equilibrium problem. Even though the material virtually does not contain original contributions of the author, it seems necessary to place it in the thesis, as the following formulations of sensitivity in Chapter 4 extensively recall the notation and details of the formulation of the primary problem.

In Chapter 4 the sensitivity formulation associated with the primary problem is derived and presented. This is the main original part of the thesis. Attention is focused on computational aspects of DSA, including, among others, the role of the consistent, algorithmic tangent operator. The problem of shape sensitivity is also addressed — it is demonstrated that there is actually no fundamental difference between the DSA formulation for shape and non-shape design parameters. Further, the equations appearing in the incremental constitutive procedures of elasticity and elasto-plasticity, cf. Section 3.3, are differentiated with respect to design-dependent input parameters in order to determine partial and total design derivatives of response state fields necessary in the global sensitivity formulation. Linearity of the sensitivity formulation at a time step is underlined, even in presence of highly nonlinear plastic consistency equations in the constitutive model. Resulting formulations are presented in the form of closed-form algorithms, readily implementable in computer programs. A number of computational examples illustrate the presented computational algorithms and inspire discussion.

Chapter 5 contains extension of the considerations towards dynamic analysis of elasto-plastic structures. Both primary and associated sensitivity equations are presented in the formalism of the finite element method, for both implicit and explicit solution strategies. Our attention is mainly focused on explicit algorithms of dynamic analysis, first because this approach is most popular in practical engineering applications due to its simplicity and efficiency, and also because the sensitivity analysis in the case of explicit dynamic formulation follows a different strategy than in the case of statics or implicit dynamics. The mathematical derivations are again illustrated with computational examples.

Chapter 6 contains remarks on implementation of sensitivity algorithms in finite element codes. Practical difficulties (a large amount of additional coding) and advantages (automated character of the sensitivity formulae) are pointed out. The issue of automatic code generation at various levels of formulation is discussed.

In Chapter 7, conclusions are formulated and directions of desired future research are outlined.

1.2. Remarks on notation

In view of a large variety of mathematical objects appearing in the text, the notation may be confusing to the reader. Limited choice of available alphabet letters, type faces and other symbols makes it virtually impossible to distinguish so many different classes of variables, operators, indices, etc., in a clear, unambiguous manner. The following general remarks are intended to minimize possible misunderstandings by explaining rules governing the notation assumed by the author. Many of explanations given below are formulated in more detail further in the text, wherever a particular notation is introduced, it seems yet worthy to gather them in this place so that the reader can easily locate them without tedious tracking the entire book.

Coordinates

Everywhere in the text, the Cartesian coordinate system is only used, defined by the coordinates $\boldsymbol{x} = \{x_i\}$, $i = 1, 2, 3$. Time is denoted by t and the coordinates of a moving material point in the time instant t are denoted by $\boldsymbol{x}^t$. Coordinates of the point in the reference configuration are denoted by $\boldsymbol{X}$ (cf. Section 3.1.1). Differentiation of any field $a(\boldsymbol{X}, t)$ with respect to the reference coordinates X_i is denoted by $a_{,i}$ while with respect to time — with the dot accent $\dot{a}$.

The notation $\boldsymbol{\xi} = \{\xi_i\}$, $i = 1, 2, 3$, is used for dimensionless coordinates of the parent isoparametric finite element as well as for coordinates of the parent design-independent configuration in shape sensitivity (cf. Section 4.1.2). Differentiation of any field $a(\boldsymbol{\xi})$ with respect to ξ_j is denoted by $a_{;j}$.

Scalars, vectors and tensors in the 3D space

Scalars are denoted by normal face italic letters (latin or greek, small or capital).

Vectors in the absolute notation are denoted by small bold face italic letters (e.g. $\boldsymbol{u}$, $\boldsymbol{f}$) while in the index notation — with corresponding normal face letters (e.g. u_i , f_i) with subscripts taking the values 1,2,3.

Second rank tensors in the absolute notation are denoted by bold face italic letters (latin or greek, small or capital, e.g. $\boldsymbol{F}$, $\boldsymbol{\varepsilon}$, $\boldsymbol{s}$) while in the index notation — with corresponding normal face letters (e.g. F_{ij} , ε_{ij} , s_{ij}) with subscripts taking the values 1,2,3. In the large deformation description, tensors defined in the reference configuration (Lagrangean) are distinguished by capital letters (e.g. the second Piola–Kirchhoff-type stress tensors $\boldsymbol{T}$, $\boldsymbol{A}$), contrary to those defined in the current, moving configuration (spatial) denoted by small letters (e.g. the Kirchhoff-type stress tensors $\boldsymbol{\tau}$, $\boldsymbol{\alpha}$). In the small deformation description, tensors are denoted by small letters (e.g. Cauchy stress $\boldsymbol{\sigma}$, linear strain $\boldsymbol{\varepsilon}$).

Fourth rank tensors in the absolute notation are denoted by bold face gothic letters (e.g. the constitutive stiffness tensor $\boldsymbol{\mathfrak{C}}$) while in the index notation — with corresponding normal face gothic letters (e.g. $\mathfrak{C}_{ijkl}$) with subscripts taking the values 1,2,3. Similarly as above, in the large deformation description, capital gothic letters indicate Lagrangean tensors (e.g. $\boldsymbol{\mathfrak{C}}$) while small letters — spatial tensors (e.g. $\boldsymbol{\mathfrak{c}}$). An exception is the linearized, small-deformation constitutive tensor denoted by a calligraphic letter $\boldsymbol{\mathcal{C}}$.

In the index notation, the Gauss summation convention is applied. A simple product is denoted e.g. by $\boldsymbol{v}\boldsymbol{w} = v_i w_i$ or $\boldsymbol{\sigma}\boldsymbol{n} = \{\sigma_{ij} n_j\}$. The full product of tensors is denoted by a colon : , e.g. $\boldsymbol{\sigma} : \boldsymbol{\varepsilon} = \sigma_{ij}\varepsilon_{ij}$ or $\boldsymbol{\mathfrak{C}} : \boldsymbol{E} = [\mathfrak{C}_{ijkl} E_{kl}]$.

FE vector/matrix representation of tensors

In Sections 3.3.3, and 4.2.3, where the reduced-dimension formulations (1D bar, plane stress) are discussed, the so called finite element vector/matrix notation is used in which symmetric 2nd and 4th rank tensors are represented by 6 × 1 column arrays (vectors) and 6 × 6 matrices, respectively. Detailed explanation is given in the beginning of Section 3.3.3. Such matrices are denoted with bold face upright letters (e.g. $\boldsymbol{\sigma}$, $\mathbf{C}$) corresponding to their italic or gothic counterparts (e.g. $\boldsymbol{\sigma}$, $\boldsymbol{\mathfrak{C}}$) in the absolute tensor notation.

Global discrete formulation arrays

Global arrays in discrete formulations (e.g. global vectors and matrices in the finite element formulation) are denoted with sans-serif upright letters, bold face in the absolute notation (e.g. $\boldsymbol{\mathsf{q}}$, $\boldsymbol{\mathsf{K}}$) or normal face in the index notation (e.g.

q_α , $\mathsf{K}_{\alpha\beta}$). To avoid misunderstanding, the indices defining elements of such arrays (and running from 1 to the global number of discrete parameters of interpolation functions) are denoted with small greek letters, contrary to small latin letters, reserved either for spatial coordinate indices or for subsequent time instant numbers. Arrays with mixed types of indices, like e.g. shape functions $\mathsf{\Phi}_{i\alpha}$ or the geometric array $\mathsf{B}_{ij\alpha}$, are never used in the absolute notation. Indices being the node counters in the finite element mesh are denoted by capital latin letters (A, B, ...)

Other arrays

Arrays of variables of different types are denoted by small bold face latin upright letters. Examples are $\mathbf{h}$ (array of scalar design parameters), $\mathbf{G}$ (array of scalar design performances), $\mathbf{p}$ (array of input fields), $\mathbf{z}$ (array of scalar and tensorial constitutive state parameters). The use of the same type face as in the case of the (mentioned before) f.e. vector/matrix representations of symmetric tensors should not be confusing since in both cases we have to do with simple arrays of numbers (or of other objects).

Generally, in the notation of arrays, curly braces $\{\cdots\}$ denote a column-aligned 1D array (vector) of the displayed components (usually separated with commas for better legibility). Brackets $[\cdots]$ denote either a row-aligned 1D array (vector), or a two- or more-dimensional array (matrix) of the displayed components.

Differentiation of a quantity with respect to components of a column vector of parameters yields a row vector of derivatives. E.g., if $A = A(\boldsymbol{\mathsf{q}})$ and $\boldsymbol{\mathsf{q}} = \{\mathsf{q}_1, \mathsf{q}_2, \ldots\}$ then $\frac{\mathrm{d}A}{\mathrm{d}\boldsymbol{\mathsf{q}}} = [\frac{\mathrm{d}A}{\mathrm{d}\mathsf{q}_1}, \frac{\mathrm{d}A}{\mathrm{d}\mathsf{q}_2}, \ldots]$. If $\mathbf{A}(\boldsymbol{\mathsf{q}})$ is a column array of functions of $\boldsymbol{\mathsf{q}}$, then the derivative $\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}\boldsymbol{\mathsf{q}}}$ is a matrix whose subsequent columns are arrays $\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}\mathsf{q}_i}$. Hence, according to the standard notation of the vector/matrix calculus, we can write $\mathrm{d}(\cdot) = \frac{\mathrm{d}(\cdot)}{\mathrm{d}\boldsymbol{\mathsf{q}}}\,\mathrm{d}\boldsymbol{\mathsf{q}}$, independently of the type (scalar, array, tensor, etc.) of $(\cdot)$.

Design derivatives

The design derivative of any function $(\cdot)$, i.e. the row-aligned array of its derivatives with respect to components h_i of $\mathbf{h}$, is denoted by $\mathrm{d}_{\mathbf{h}}(\cdot)$. The explicit design derivative, i.e. the partial design derivative expressing dependence of $(\cdot)$ on $\mathbf{h}$ only via those arguments of $(\cdot)$ whose design-dependence is explicitly known, is denoted by $\partial_{\mathbf{h}}(\cdot)$ (cf. Section 2.1.3). The notation $\mathrm{d}_{\mathbf{h}!b}(\cdot)$ stands for the partial design derivative computed at the argument b (of any type) kept constant, i.e.

$\mathrm{d}_{\mathbf{h}}(\cdot) = \mathrm{d}_{\mathbf{h}!b}(\cdot) + \frac{\partial(\cdot)}{\partial b}\,\mathrm{d}_{\mathbf{h}}b$ (cf. Section 4.1). The notation $\mathrm{d}^{\diamond}_{\mathbf{h}}(\cdot) = [\mathrm{d}_{\mathbf{h}}(\cdot)]\,(\cdot)^{-1}$ denotes a 'normalized' design derivative (deprived of the units of $(\cdot)$)

Other notation

The hat ˆ over a symbol denotes a known value of the input variable.

The tilde ˜ has several meanings. In the transit between continnum and discrete equations of structural mechanics, it indicates variables of intermediate formulations, e.g. discretized in space but continuous in time (cf. Section 2.1). In the global equilibrium formulation, it denotes generic measure of stress and the associated work-conjugate strain variation (i.e. different particular pairs of tensorial work-conjugate stress and strain variation measures can be substituted under $(\tilde{\sigma}, \delta\tilde{\varepsilon})$, cf. Section 3.1.2). Elsewhere, the tilde indicates the stress-free deformed configuration in the kinematical model of large-deformation elastoplasticity based on the multiplicative split of the gradient deformation (3.66). In the kinematical description of the enhanced assumed strain elements (Section 3.4.4), the tilde denotes a value averaged over the entire element volume.

The bar ¯ over a symbol has several meanings, too. In description of the deformation state it denotes the isochoric part of a tensor describing deformation (like deformation gradient $\bar{\boldsymbol{F}} = J^{-\frac{1}{3}}\boldsymbol{F}$, Cauchy–Green deformation tensors $\bar{\boldsymbol{C}}$, $\bar{\boldsymbol{b}}$, etc., cf. Eqs. (3.33)–(3.35). In the kinematical description of the enhanced assumed strain elements (Section 3.4.4) it denotes the enhanced measure of such a tensor (e.g. $\bar{\boldsymbol{F}} = (\tilde{J}/J)\boldsymbol{F}$). The two meanings of this notation used with the same types of tensors may lead to confusion, however, both are so well grounded in the tradition of the subject literature that it seems perhaps even more confusing to assign another notation to any of them.

$\mathcal{L}_{\boldsymbol{v}}(\cdot)$ denotes the objective Lie derivative of a spatial tensor $(\cdot)$ (cf. e.g. Eq. (3.19)). $\mathrm{d}^{\circ}(\cdot)$ stands for the analogous objective differential of the tensor (i.e. computed with the same formula as the Lie derivative, with time derivatives replaced by differentials, Eq. (3.97)).

The superscripts $'$ and $''$ at a function symbol denote single and double differentiation of the function with respect to its argument. This usually refers to certain plastic constitutive functions (hardening functions) where the argument of the differentiation is the equivalent plastic strain $\bar{e}^{\mathrm{p}}$.

In discussion of reduced-dimension formulations, the transversal components of stress and strain are denoted with the subscript $\perp$ (e.g. $\sigma_{\perp}$) while the longitudinal (or in-plane) components — with the subscript $\|$ (e.g. $\varepsilon_{\|}$).

Chapter 2

Concepts of sensitivity analysis of nonlinear systems

This chapter presents the general idea of sensitivity analysis of nonlinear mechanical systems. First, the primary problem is briefly formulated, and the two basic notions of sensitivity analysis, i.e. design variables and performance functionals, are introduced. Then, two fundamental approaches to the sensitivity analysis, continuum and discrete, are distinguished and described, with discussion of their advantages and drawbacks, and with indication of certain intermediate ways. Further, the discrete approach is discussed in detail, with two alternative solution strategies, DDM and ASM. The analytical, semi-analytical and finite-difference methods of determination of sensitivity gradients are presented and reviewed. Finally, the key issue of elasto-plastic response differentiability is risen and its consequences regarding possible discontinuity of the sensitivity solution are discussed.

2.1. General formulation of the problem

The sensitivity of a structural system response to variations of its parameters is one of the most important problems necessary for proper understanding of the system performance. In fact, it is now widely acknowledged that any reliable approach to practical structural engineering problems should provide the analyst with an assessment of not just the response itself, but also of its parameter sensitivity.

2.1.1. Primary problem

Before we refer in our considerations to analysis of response sensitivity, let us first identify the primary problem, i.e. the problem of structural mechanics. This is the problem whose solution is of the analyst's primary interest and which is going to be the subject of sensitivity analysis.

A structural system with defined geometry, boundary conditions, loads, etc., must obey physical laws which are expressed in the mathematical form as a system of partial differential equations with appropriate initial and boundary conditions,

$$\mathcal{F}[\mathbf{r}(\boldsymbol{x},t),\mathbf{p}(\boldsymbol{x},t)]=\mathbf{0}. \tag{2.1}$$

$\mathcal{F}$ denotes a differential operator acting on the unknown space-time fields, generally denoted by $\mathbf{r}$ and referred to as *response*, and on the known fields $\mathbf{p}$ being the problem *input*. Depending on the particular problem nature, the response $\mathbf{r}$ may include displacements, stresses, strains, constitutive state parameters, temperature, pressure, etc. (all understood as space-time distributions), while such quantities as geometry, material properties, prescribed loads and kinematical boundary conditions constitute the input fields $\mathbf{p}$.

Solution of the problem (2.1) is typically sought for in its approximate form with the use of generally understood numerical methods. Without coming into details of such methods and without introducing particular formalisms of e.g. the finite element method, or the boundary element method, etc., we can generally say that their essence is to transform the *distributed parameter system* in which both the given input $\mathbf{p}$ and the unknown response $\mathbf{r}$ are fields on a continuous domain $(\boldsymbol{x},t)$ onto a *discrete parameter system*

$$\mathsf{F}(\mathsf{q},\mathsf{p})=\mathbf{0} \tag{2.2}$$

in which p and q are arrays of discrete values while F is an array of nonlinear functions of q and p. The unknown components of q and the known components of p play the role of parameters that define the approximate response $\mathbf{r}(\boldsymbol{x},t)$ and the approximate input fields $\mathbf{p}(\boldsymbol{x},t)$ according to the scheme

$$\mathbf{p}(\boldsymbol{x},t)\approx\hat{\mathbf{p}}(\mathsf{p},\boldsymbol{x},t),\qquad \mathbf{r}(\boldsymbol{x},t)\approx\hat{\mathbf{r}}(\mathsf{q},\boldsymbol{x},t), \tag{2.3}$$

in which $\hat{\mathbf{p}}$ and $\hat{\mathbf{r}}$ are explicit functions of their arguments provided by the particular numerical method used.

If the lengths of the arrays F and q are the same, Eq. (2.2) is a nonlinear algebraic system of equations that can be solved with respect to q. In practice, q is frequently obtained as a solution of a series of smaller systems of equations involving separate parts of q, usually associated with subsequent discrete time instants.

Transformation from (2.1) into (2.2) requires discretization of the structural domain in both space and time. Typically, the space and time domains are discretized independently, which makes possible intermediate, distributed-discrete formulations situated between (2.1) and (2.2). An example of such a formulation is

$$\tilde{\mathsf{F}}[\tilde{\mathsf{q}}(t), \tilde{\mathsf{p}}(t)] = \mathbf{0}\,, \tag{2.4}$$

in which the array of space-discrete parameter functions of time $\tilde{\mathsf{q}}(t)$ is the unknown, while the space-discrete transient input is defined as an array $\tilde{\mathsf{p}}(t)$. The operator $\tilde{\mathsf{F}}$ denotes an array of differential operators acting on arrays of the mentioned time functions.

The formulation (2.4) is just one of possible examples; we can also mention e.g. different levels of time-discretization (for global system equations and for local constitutive equations) in which the interpolation/integration rules do not necessary need to be the same. Generally, the pass from distributed to discrete

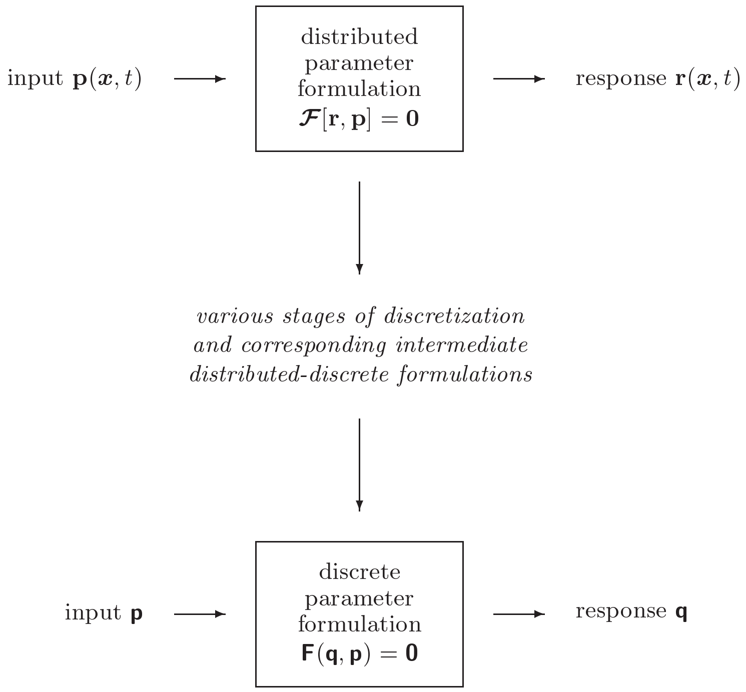

input $\mathbf{p}(\boldsymbol{x}, t)$
distributed parameter formulation $\mathcal{F}[\mathbf{r}, \mathbf{p}] = \mathbf{0}$
response $\mathbf{r}(\boldsymbol{x}, t)$
various stages of discretization and corresponding intermediate distributed-discrete formulations
input p
discrete parameter formulation $\mathsf{F}(\mathsf{q}, \mathsf{p}) = \mathbf{0}$
response q

Figure 2.1. Primary problem of structural mechanics — distributed vs. discrete parameter formulations

parameter formulations can be graphically depicted with the scheme presented in Fig. 2.1.

2.1.2. Performance functionals and design parameters

In practice, analysts are usually not interested in the entire response fields $\mathbf{r}$ being the solution of Eq. (2.1). For them, the term *response* denotes rather a number of certain particular quantities characterizing the structural response from the engineering point of view. The quantities, referred to as *response functionals* or *performance functionals*, are denoted by $\mathcal{G}[\mathbf{r}(\boldsymbol{x},t),\mathbf{p}(\boldsymbol{x},t)]$. They are explicitly defined functionals of the given space-time fields. Their values may be scalar-, vector-, or tensor-type, however, without much loss of generality, we can limit ourselves in this study to only scalar ones. Let us denote by N_p the total number of performance functionals.

Thus, having solved the system (2.1), we obtain the 'raw' distributed response $\mathbf{r}(\boldsymbol{x},t)$ that can be utilized to quantify the system performance $\boldsymbol{\mathcal{G}}$, understood as a column-aligned array of functionals

$$\boldsymbol{\mathcal{G}}[\mathbf{r},\mathbf{p}] = \{\, \mathcal{G}_1[\mathbf{r},\mathbf{p}]\,,\, \mathcal{G}_2[\mathbf{r},\mathbf{p}]\,,\,\ldots\,,\, \mathcal{G}_{N_p}[\mathbf{r},\mathbf{p}]\,\}. \tag{2.5}$$

Analogously, the raw discrete response $\mathbf{q}$ obtained as a solution of (2.2) can be used to determine an approximate value of the performance $\boldsymbol{\mathcal{G}}$ that becomes in this formulation an array $\mathbf{G}$ of N_p explicit scalar functions

$$\mathbf{G}(\mathbf{q},\mathbf{p}) = \{\, G_1(\mathbf{q},\mathbf{p})\,,\, G_2(\mathbf{q},\mathbf{p})\,,\,\ldots\,,\, G_{N_p}(\mathbf{q},\mathbf{p})\,\}. \tag{2.6}$$

Let us now define by

$$\mathbf{h} = \{\, h_1\,,\, h_2\,,\,\ldots\,,\, h_{N_d}\,\} \tag{2.7}$$

a column-aligned array of scalar parameters called *design parameters* that affect in an explicitly known way the input fields $\mathbf{p}(\boldsymbol{x},t)$. Thus, we may write

$$\mathbf{p} = \mathbf{p}(\boldsymbol{x},t;\mathbf{h}). \tag{2.8}$$

In many cases the dependence of $\mathbf{p}$ on $\mathbf{h}$ may have a trivial, Boolean form — if, e.g., the Young modulus distribution, constituting a part of the input $\mathbf{p}$, is uniform ($E(\boldsymbol{x},t) = \text{const} = E$) and a design parameter h_1 is identified with its value E, then the derivative $\partial\mathbf{p}/\partial h_1$ takes the uniform value 1 for the Young modulus and 0 for all the other input fields (unless they somehow depend on E, like e.g. the shear modulus G). However, e.g. shape parameters (i.e. design parameters defining geometry of the structure, like coordinates of characteristic points, curvature radii, etc.) may affect many different input fields in a quite complex manner. Thus, the general form (2.8) of input design-dependence will be retained in the following considerations.

The name *design parameters* suggests that values of the parameters h_i , $i = 1, \ldots, N_d$, are all subject to the designer's choice. Recalling explanations given in Chapter 1, this notion will be given a broader sense in this dissertation — h_i will be generally understood as parameters of any nature that may affect the input fields. Particularly, parameters characterizing geometric or material imperfections may also enter the set **h**. Since mathematical formulation of such dependence is the same for all types of scalar parameters, this naming convention should not affect the generality of the following derivations.

In view of the above definitions, we may modify the diagram of Fig. 2.1 and present the primary formulations as "black-boxes" that determine the system performance based on the given design, see Fig. 2.2. Explicit relations between distributed or discrete input and **h** as well as those between performance measures and the distributed or discrete response enter the boxes together with the equation systems (2.1) or (2.2).

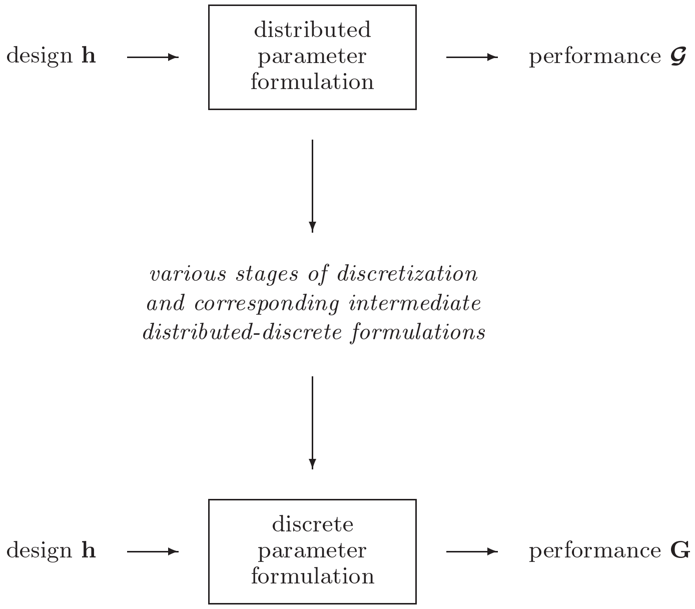

design $\mathbf{h}$
distributed parameter formulation
performance $\boldsymbol{\mathcal{G}}$
various stages of discretization and corresponding intermediate distributed-discrete formulations
design $\mathbf{h}$
discrete parameter formulation
performance $\mathbf{G}$

Figure 2.2. Primary problem of structural mechanics — distributed vs. discrete parameter formulations (modified)

2.1.3. Sensitivity problem

If $\mathbf{p} = \mathbf{p}(\boldsymbol{x}, t; \mathbf{h})$ and $\mathbf{r}$ is the solution of the system (2.1), it is obvious that $\mathbf{r} = \mathbf{r}(\boldsymbol{x}, t; \mathbf{h})$, too, i.e. Eq. (2.1) becomes

$$\boldsymbol{\mathcal{F}}[\mathbf{r}(\boldsymbol{x}, t; \mathbf{h}), \mathbf{p}(\boldsymbol{x}, t; \mathbf{h})] = \mathbf{0}. \tag{2.9}$$

In this case, however, the relationship between $\mathbf{r}$ and $\mathbf{h}$ is by no means explicit. For any arbitrary value of $\mathbf{h} = \hat{\mathbf{h}}$, the corresponding field $\mathbf{r}(\boldsymbol{x}, t; \hat{\mathbf{h}})$ can be determined with the methods discussed above, but the general relationship $\mathbf{r}(\boldsymbol{x}, t; \mathbf{h})$ could only be sought for in either a tabularized form, by subsequently solving the system (2.9) at various values of $\mathbf{h}$, or in an approximate analytical form, as a Taylor series expansion around a known solution for $\mathbf{h} = \hat{\mathbf{h}}$,

$$\begin{aligned}\mathbf{r}(\boldsymbol{x}, t; \mathbf{h}) \;=\; & \mathbf{r}(\boldsymbol{x}, t; \hat{\mathbf{h}}) + \mathrm{d}_{\mathbf{h}}\mathbf{r}(\boldsymbol{x}, t; \hat{\mathbf{h}})\,(\mathbf{h} - \hat{\mathbf{h}}) \\ & + \tfrac{1}{2}\,(\mathbf{h} - \hat{\mathbf{h}})^{\mathrm{T}}\,\mathrm{d}^2_{\mathbf{h}}\mathbf{r}(\boldsymbol{x}, t; \hat{\mathbf{h}})\,(\mathbf{h} - \hat{\mathbf{h}}) + \cdots ,\end{aligned} \tag{2.10}$$

where the following notation has been introduced,

$$\mathrm{d}_{\mathbf{h}}(\cdot) = \frac{\mathrm{d}(\cdot)}{\mathrm{d}\mathbf{h}} \;=\; \left[\frac{\partial(\cdot)}{\partial h_1}, \frac{\partial(\cdot)}{\partial h_2}, \ldots, \frac{\partial(\cdot)}{\partial h_{N_d}}\right]. \tag{2.11a}$$

$$\mathrm{d}^2_{\mathbf{h}}(\cdot) = \frac{\mathrm{d}^2(\cdot)}{\mathrm{d}\mathbf{h}^2} = \begin{bmatrix} \frac{\partial^2(\cdot)}{\partial h_1^2} & \frac{\partial^2(\cdot)}{\partial h_1 \partial h_2} & \cdots & \frac{\partial^2(\cdot)}{\partial h_1 \partial h_{N_d}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2(\cdot)}{\partial h_{N_d} \partial h_1} & \frac{\partial^2(\cdot)}{\partial h_{N_d} \partial h_2} & \cdots & \frac{\partial^2(\cdot)}{\partial h_{N_d}^2} \end{bmatrix}. \tag{2.11b}$$

The formulation (2.10) is only possible if the above ordinary derivatives exist. Here we sustain this *a priori* assumption, making also reference to Section 2.2.4 where the extension towards discontinuous (directional) design derivatives is discussed.

Expressing a design-dependent performance functional $\mathcal{G}[\mathbf{r}(\boldsymbol{x}, t; \mathbf{h}), \mathbf{p}(\boldsymbol{x}, t; \mathbf{h})]$ as an implicit function of design, $\mathcal{G}(\mathbf{h})$, we can consequently approximate its value either by interpolation of tabularized results for certain values of $\mathbf{h}$, or by employing an approximate form analogous to (2.10),

$$\mathcal{G}(\mathbf{h}) = \mathcal{G}(\hat{\mathbf{h}}) + \mathrm{d}_{\mathbf{h}}\mathcal{G}(\hat{\mathbf{h}})\,(\mathbf{h} - \hat{\mathbf{h}}) + \tfrac{1}{2}\,(\mathbf{h} - \hat{\mathbf{h}})^{\mathrm{T}}\,\mathrm{d}^2_{\mathbf{h}}\mathcal{G}(\hat{\mathbf{h}})\,(\mathbf{h} - \hat{\mathbf{h}}) + \cdots . \tag{2.12}$$

In both the series, (2.10) and (2.12), all the derivatives with respect to $\mathbf{h}$ are unknown.

The essence of design sensitivity analysis (DSA) of a distributed parameter system (2.9) is to determine all the performance gradients with respect to design $\mathbf{h}$ that are necessary to approximate changes of $\boldsymbol{\mathcal{G}}$, Eq. (2.5), corresponding to small variations of $\mathbf{h}$ by employing a Taylor series expansion (2.12) around a known solution at a given *primary design* $\mathbf{h}$. The gradients will be referred to as *design sensitivities*, cf. classical definitions of e.g. [41, 65]. Depending on the order of the gradients we may talk about first- and higher-order sensitivity analysis (e.g. sensitivities of $\mathcal{G}$ employed in Eq. (2.12) are displayed up to the second order). In this dissertation, attention will only be focused on the first-order sensitivity analysis methods.

Determination of the first order design sensitivities of the performance functionals, $\mathrm{d}_{\mathbf{h}}\boldsymbol{\mathcal{G}}$, at given design parameters $\mathbf{h}$ requires

- solution of the distributed parameter sensitivity problem, complementary to the problem (2.1),

$$\boldsymbol{\mathcal{D}}[\mathrm{d}_{\mathbf{h}}\mathbf{r}(\boldsymbol{x},t;\mathbf{h}),\mathbf{r}(\boldsymbol{x},t;\mathbf{h}),\partial_{\mathbf{h}}\mathbf{p}(\boldsymbol{x},t;\mathbf{h}),\mathbf{p}(\boldsymbol{x},t;\mathbf{h})]=\mathbf{0}. \tag{2.13}$$

  in which the unknown field is the sensitivity $\mathrm{d}_{\mathbf{h}}\mathbf{r}(\boldsymbol{x},t;\mathbf{h})$, and whose particular formulation depends on a particular form of the primary problem operator $\boldsymbol{\mathcal{F}}$, and

- determination of the performance sensitivity functionals

$$\mathrm{d}_{\mathbf{h}}\boldsymbol{\mathcal{G}}[\mathrm{d}_{\mathbf{h}}\mathbf{r}(\boldsymbol{x},t;\mathbf{h}),\mathbf{r}(\boldsymbol{x},t;\mathbf{h}),\partial_{\mathbf{h}}\mathbf{p}(\boldsymbol{x},t;\mathbf{h}),\mathbf{p}(\boldsymbol{x},t;\mathbf{h})] \tag{2.14}$$

  whose particular forms again depend on the particular forms of $\mathcal{G}_i$.

In the above formulation, it is assumed that the primary response solution $\mathbf{r}(\boldsymbol{x},t;\mathbf{h})$ and values of the functionals $\boldsymbol{\mathcal{G}}$ are known. It is once again stressed that design-dependence of the input $\mathbf{p}$ is explicit and thus the design derivatives of $\mathbf{p}$ can also be explicitly determined. To underline this fact, the notation $\partial_{\mathbf{h}}(\cdot)$ for *explicit design derivative* is introduced and used in Eq. (2.13) for $\mathbf{p}$ (so that known and unknown design derivatives can be easily distinguished in the formulation).

Looking at the formulae (2.13)–(2.14) it is easy to notice that their combination may lead to a formulation in which $\mathrm{d}_{\mathbf{h}}\boldsymbol{\mathcal{G}}$ is determined directly as a solution of a differential equations system, without intermediate determination of the field $\mathrm{d}_{\mathbf{h}}\mathbf{r}(\boldsymbol{x},t;\mathbf{h})$. Such a formulation, if available in a closed form, is referred to as the *adjoint system formulation* of the DSA.

Similarly as the primary problem of structural mechanics, the sensitivity problem (2.13) generally requires numerical methods in order to obtain a solution, which is obviously approximate. Again, a variety of methods remains to the analyst's choice, all of them finally leading to a discrete parameter sensitivity problem that can be expressed in the following general form

$$\mathsf{D}(\mathrm{d}_{\mathbf{h}}\mathsf{q}, \mathsf{q}, \partial_{\mathbf{h}}\mathsf{p}, \mathsf{p}) = \mathbf{0}\,. \tag{2.15}$$

Here, D is an array of nonlinear functions of their arguments, $\mathrm{d}_{\mathbf{h}}\mathsf{q}$, q, $\partial_{\mathbf{h}}\mathsf{p}$ and p, which are arrays of parameters defining, for an assumed interpolation/integration scheme, the approximate fields $\mathrm{d}_{\mathbf{h}}\mathbf{r}(\boldsymbol{x},t;\mathbf{h})$, $\mathbf{r}(\boldsymbol{x},t;\mathbf{h})$, $\partial_{\mathbf{h}}\mathbf{p}(\boldsymbol{x},t;\mathbf{h})$ and $\mathbf{p}(\boldsymbol{x},t;\mathbf{h})$, respectively. Assuming q to be known from the primary problem solution, Eq. (2.15) constitutes an algebraic system of equations that can be solved against the only unknown array $\mathrm{d}_{\mathbf{h}}\mathsf{q}$. Sensitivities of the performance measures $\mathbf{G}(\mathsf{q},\mathsf{p})$, expressed in the discrete formulation as analytical functions of the arrays q and p, can be then obtained from the chain rule[1]

$$\mathrm{d}_{\mathbf{h}}\mathbf{G} = \frac{\partial \mathbf{G}}{\partial \mathsf{q}}\,\mathrm{d}_{\mathbf{h}}\mathsf{q} + \frac{\partial \mathbf{G}}{\partial \mathsf{p}}\,\partial_{\mathbf{h}}\mathsf{p}, \tag{2.16a}$$

or, utilizing explicit dependence of $\mathbf{G}$ on p, and thus also on $\mathbf{h}$, and introducing the notation $\partial_{\mathsf{q}}\mathbf{G} = \partial\mathbf{G}/\partial\mathsf{q}$,

$$\mathrm{d}_{\mathbf{h}}\mathbf{G} = \partial_{\mathsf{q}}\mathbf{G}\,\mathrm{d}_{\mathbf{h}}\mathsf{q} + \partial_{\mathbf{h}}\mathbf{G}. \tag{2.16b}$$

The performance sensitivity $\mathrm{d}_{\mathbf{h}}\mathbf{G}$ is an array of $N_p \times N_d$ functions of q and p,

$$\mathrm{d}_{\mathbf{h}}\mathbf{G} = \begin{bmatrix} \partial G_1(\mathsf{q}(\mathbf{h}),\mathsf{p}(\mathbf{h}))/\partial h_1 & \cdots & \partial G_1(\mathsf{q}(\mathbf{h}),\mathsf{p}(\mathbf{h}))/\partial h_{N_d} \\ \vdots & & \vdots \\ \partial G_{N_p}(\mathsf{q}(\mathbf{h}),\mathsf{p}(\mathbf{h}))/\partial h_1 & \cdots & \partial G_{N_p}(\mathsf{q}(\mathbf{h}),\mathsf{p}(\mathbf{h}))/\partial h_{N_d} \end{bmatrix}.$$

The numerical method employed to transform the distributed parameter sensitivity problem (2.13) to the discrete parameter form (2.15) does not need to be the same as the one applied in the primary analysis formulation. In particular, space and time discretizations, interpolation functions, integration rules, as well as many other numerical details, may differ in both the cases. Since accuracy and stability requirements for the sensitivity solution may be different than those for the primary problems, it seems natural to conclude that e.g. the space/time discretization should be defined separately in both the problems.

On the other hand, it is obviously possible, and often quite advantageous, to use exactly the same numerical approach to discrete solution of both the primary and the sensitivity problem. It can be easily verified that, in such a case, the

[1] According to our previously introduced convention, arrays like $\mathrm{d}_{\mathbf{h}}\mathbf{G}$, $(\partial\mathbf{G}/\partial\mathsf{q})$, $(\partial\mathbf{G}/\partial\mathsf{p})$ are understood as matrices whose subsequent columns are derivatives of $\mathbf{G}$ with respect to subsequent components h_i , q_α and p_α , respectively.

formulation (2.15) can be derived directly from design differentiation of the discrete primary problem (2.2),

$$\mathbf{F}(\mathbf{q}(\mathbf{h}), \mathbf{p}(\mathbf{h})) = \mathbf{0}\,. \tag{2.17}$$

Since the operator $\mathbf{F}$ is merely an explicit nonlinear function of its displayed arguments, its differentiation with respect to $\mathbf{h}$ yields the following equation

$$\frac{\partial \mathbf{F}(\mathbf{q}, \mathbf{p})}{\partial \mathbf{q}}\, \mathrm{d}_{\mathbf{h}}\mathbf{q} = -\frac{\partial \mathbf{F}(\mathbf{q}, \mathbf{p})}{\partial \mathbf{p}}\, \partial_{\mathbf{h}}\mathbf{p}\,, \tag{2.18a}$$

which is a particular form of Eq. (2.15). Utilizing again explicit dependence of $\mathbf{F}$ on $\mathbf{p}$, and introducing the notation $\mathbf{K} = -\partial \mathbf{F}/\partial \mathbf{q}$, it can be rewritten as

$$\mathbf{K}\, \mathrm{d}_{\mathbf{h}}\mathbf{q} = \partial_{\mathbf{h}}\mathbf{F}\,. \tag{2.18b}$$

At a known primary solution $\mathbf{q}(\mathbf{h})$ for a given $\mathbf{h}$, the above equation is a linear system of algebraic equations with respect to the unknown array $\mathrm{d}_{\mathbf{h}}\mathbf{q}$, with the square coefficient matrix $\mathbf{K}$. More strictly, it is a series of N_d systems of equations, one per each design parameter h_i, with the common coefficient matrix $\mathbf{K}$ and different right-hand sides.

Combining Eqs. (2.16) and (2.18), we may express the sensitivity of performance functionals as

$$\mathrm{d}_{\mathbf{h}}\mathbf{G} = \left[-\frac{\partial \mathbf{G}}{\partial \mathbf{q}} \left[\frac{\partial \mathbf{F}}{\partial \mathbf{q}} \right]^{-1} \frac{\partial \mathbf{F}}{\partial \mathbf{p}} + \frac{\partial \mathbf{G}}{\partial \mathbf{p}} \right] \partial_{\mathbf{h}}\mathbf{p} = \partial_{\mathbf{q}}\mathbf{G}\, \mathbf{K}^{-1}\, \partial_{\mathbf{h}}\mathbf{F} + \partial_{\mathbf{h}}\mathbf{G}\,. \tag{2.19}$$

Resuming, in the formulation of a discrete parameter sensitivity problem complementary to the primary problem expressed in the continuous form as (2.9) and in the discrete form as (2.17), two alternative ways can be considered:

(i) derivation of a distributed parameter sensitivity problem in the form (2.13) followed by application of a numerical approach (generally different than the one employed in discretization of (2.9)) leading to the general formulation (2.15)–(2.16),

(ii) direct design differentiation of the discrete parameter primary problem (2.17) leading to the formulation (2.19), being a particular, detailed form of (2.15)–(2.16).

The advantage of (i) is the mentioned possibility of adapting the chosen numerical schemes to requirements of stability and accuracy separately for each of the two problems. Its drawback, however, is the not always easy derivation of the sensitivity problem, especially in the case of complex forms of the operator $\boldsymbol{\mathcal{F}}$, and the necessity of separate numerical implementation of the two problems. The advantage of (ii) is simplicity of practical derivation and implementation, frequently available by employing more or less automatized methods. Additionally, results of sensitivity analysis performed with the method (ii) appear to be easily verifiable numerically — in fact, up to the accuracy of numerical roundings, they are the limit values of approximate finite-difference design derivatives of the discrete primary results of (2.17) determined for perturbed designs $\mathbf{h} \pm \Delta\mathbf{h}$, with $\Delta\mathbf{h} \to \mathbf{0}$. Checking this convergence may thus be a good test on correctness of the problem formulation and its numerical implementation. This obviously may not be the case when using the approach (i) where such convergence cannot be guaranteed.

The two ways defined above are in fact not the only approaches to sensitivity problem formulation. One can imagine a number of intermediate formulations combining partially features of the two extreme ones. Recalling e.g. the primary formulation (2.4), being one of intermediate steps between the distributed and discrete parameter systems, we can easily imagine a complementary sensitivity formulation

$$\tilde{\mathbf{D}}[\mathrm{d}_{\mathbf{h}}\tilde{\mathbf{q}}(t;\mathbf{h}), \tilde{\mathbf{q}}(t;\mathbf{h}), \mathrm{d}_{\mathbf{h}}\tilde{\mathbf{p}}(t;\mathbf{h}), \tilde{\mathbf{p}}(t;\mathbf{h})] = \mathbf{0}\,, \tag{2.20}$$

employing the same spatial discretization, and transform it to the form (2.15) with the use of an approximate time-integration scheme, not necessarily the same as the one employed in the primary analysis, when passing from Eq. (2.1) to Eq. (2.2).

Various approaches to sensitivity analysis discussed above and their relations with the approaches to the primary problem solution methods can be graphically summarized as in Fig. 2.3. The approach referred to as (i) in the above considerations corresponds in the diagram to the most upper and right limit path of design-differentiation and discretization sequence while the approach (ii) — to the opposite, most left and lower path. All possible intermediate strategies are situated within the area between them.

In further considerations, our attention will be focused on the latter approach in which the fully discretized primary problem is the basis for the sensitivity problem formulation. Being aware of all its drawbacks mentioned in this section, we underline once again its advantages:

- more simple derivation,
- straightforward numerical implementation and computation cost,

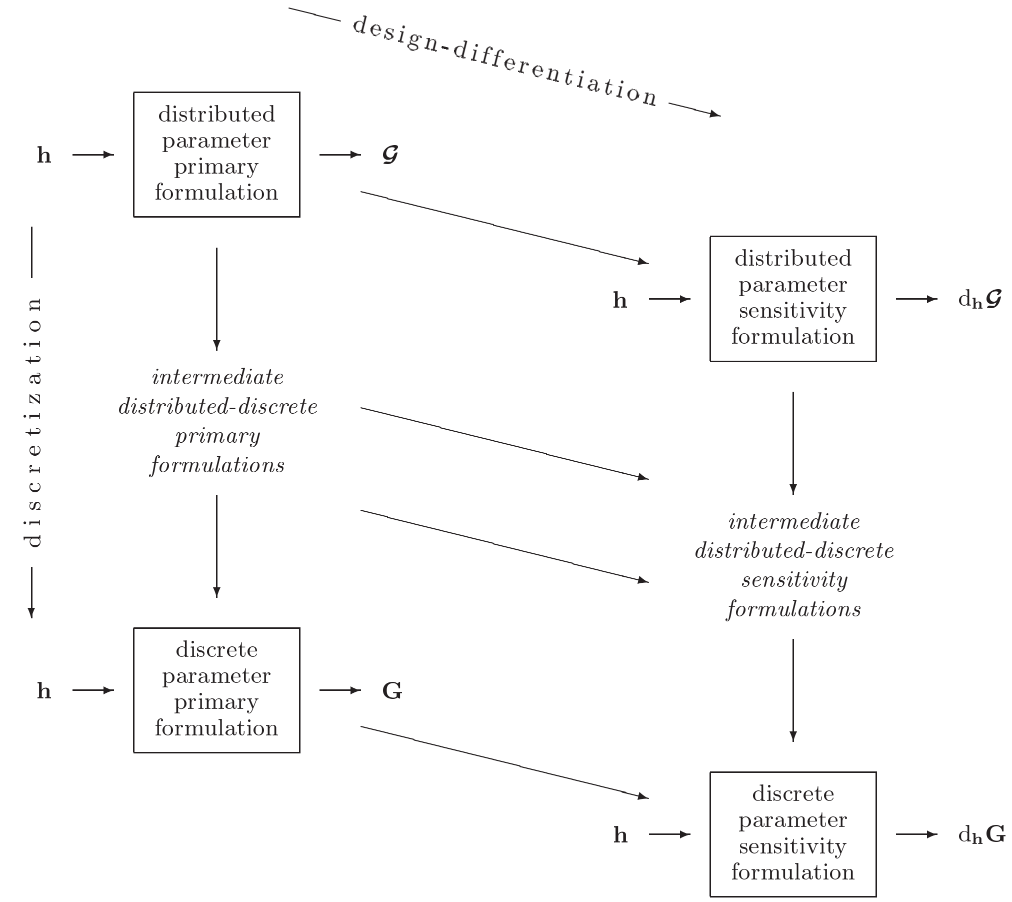

design-differentiation
discretization
h
distributed parameter primary formulation
$\mathcal{G}$
intermediate distributed-discrete primary formulations
h
discrete parameter primary formulation
G
h
distributed parameter sensitivity formulation
$\mathrm{d}_{\mathbf{h}}\mathcal{G}$
intermediate distributed-discrete sensitivity formulations
h
discrete parameter sensitivity formulation
$\mathrm{d}_{\mathbf{h}}\mathbf{G}$

Figure 2.3. Distributed vs. discrete parameter formulations of primary and sensitivity analysis problems of structural mechanics

- easy and reliable verification of results,

to which we can add one more, not mentioned before,

- possibility of general formulation for different types of primary problems.

In fact, it must be noted that different types of the differential operator $\mathcal{F}[\mathbf{r},\mathbf{p}]$ and the functionals $\mathcal{G}[\mathbf{r},\mathbf{p}]$ require different specific mathematical methods in formulation of the distributed-parameter sensitivity problem (2.13)–(2.14). The intermediate distributed-discrete formulations do not remove this disadvantage. In the fully discrete formulation, however, the expressions $\mathbf{F}(\mathbf{q},\mathbf{p})$ and $\mathbf{G}(\mathbf{q},\mathbf{p})$

are no more differential operators and functionals — they become merely functions of their discrete design-dependent arguments. Thus, no matter what their particular forms are, the sensitivity problem can be written down in a unified way in terms of their derivatives with respect to the arguments, or, employing the chain rule, in terms of their design derivatives, cf. Eq. (2.19). Such formulations will be subject to detailed investigation in the sequel of this thesis.

2.2. Solution methods

2.2.1. Primary problem: Newton iteration

Let us recall the fundamental primary problem of structural mechanics in the discrete-parameter form (2.2)

$$\mathbf{F}(\mathbf{q},\mathbf{p}) = \mathbf{0}. \tag{2.21}$$

It is stressed that no assumptions have been made at this stage about a particular numerical method used to derive the above form. The discrete array $\mathbf{q}$ is thus not associated with e.g. nodal displacements or local values of any other fields; it is only assumed that $\mathbf{q}$ contains parameters that uniquely define, upon assumed interpolation and integration schemes, all the space-time state fields constituting the approximate structural response. Analogously, the given array $\mathbf{p}$ uniquely defines all the approximate fields that constitute the problem input data.

Since Eq. (2.21) is a nonlinear system of algebraic equations with respect to $\mathbf{q}$, the general solution strategy utilizes the Newton iteration method

$$\begin{tikzcd} {} \arrow[r] & \mathbf{K}^{(i)}\,\delta\mathbf{q} = \mathbf{F}(\mathbf{q}^{(i)},\mathbf{p}) \arrow[r] & \mathbf{q}^{(i+1)} = \mathbf{q}^{(i)} + \delta\mathbf{q} \arrow[d, dash] \\ {} \arrow[u, dash] & i := i+1 \arrow[l, dash] & {} \arrow[l] \end{tikzcd} \tag{2.22}$$

in which the notation $\mathbf{K} = -\partial\mathbf{F}/\partial\mathbf{q}$ introduced in Section 2.1.3 has been employed. The iteration loop contains a system of linear equations, solved with respect to subsequent solution correctors $\delta\mathbf{q}$ until convergence is achieved. It is assumed that the initial predictor $\mathbf{q}^{(0)}$ is chosen so as to ensure convergence of the procedure.

Only in a limited number of cases (e.g. linear elasticity) the formulation (2.21) is linear with respect to $\mathbf{q}$. In these cases, the formulation (2.22) is reduced to a one-time solution of the linear equation system

$$\mathbf{K}\mathbf{q} = \mathbf{F}(\mathbf{0},\mathbf{p})\,.$$

The size of the array $\mathbf{q}$ and the corresponding number of equations may be extremely large in realistic transient large-scale problems. Thus, the solution strategy is frequently modified in order to decrease the problem size by utilizing certain features of assumed approximation schemes. For example, introduction of a unique time discretization $(t_0\,, t_1\,, \ldots\,, t_n\,, \ldots)$ in the entire structure volume and association of certain parts of $\mathbf{q}$ with particular discrete time instances leads to the following division of the column-aligned response array,

$$\mathbf{q} = \{\,\mathbf{q}_0\,, \mathbf{q}_1\,, \ldots\,, \mathbf{q}_n\,, \ldots\}. \tag{2.23}$$

If the numerical formulation allows to also associate particular equations in the array $\mathbf{F}$ with subsequent time instants

$$\mathbf{F} = \{\,\mathbf{F}_0\,, \mathbf{F}_1\,, \ldots\,, \mathbf{F}_n\,, \ldots\}, \tag{2.24}$$

the system (2.21) can be divided into a series of (generally coupled) systems of equations

$$\mathbf{F}_n(\mathbf{q},\mathbf{p}) = \mathbf{0}, \qquad n = 0, 1, \ldots \tag{2.25}$$

with respect to unknown arrays $\mathbf{q}_n$,

If the problem is path-independent (e.g. large-deformation elasticity), then

$$\mathbf{F}_n = \mathbf{F}_n(\mathbf{q}_n, \mathbf{p}). \tag{2.26}$$

Thus, the coefficient matrix $\mathbf{K}$ has the following form,

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & \cdots & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{K}_2 & \cdots & \mathbf{0} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{K}_n & \cdots \\ \vdots & \vdots & & \vdots & \ddots \end{bmatrix}, \qquad \mathbf{K}_n = -\frac{\partial \mathbf{F}_n}{\partial \mathbf{q}_n}\,. \tag{2.27}$$

The systems for different time instants are then entirely uncoupled and can be separately solved for each $\mathbf{q}_n$. The Newton loop (2.22) at each time instant t_n assumes the form

$$\begin{tikzcd} \mathbf{K}_n^{(i)}\,\delta\mathbf{q}_n = \mathbf{F}_n(\mathbf{q}_n^{(i)}, \mathbf{p}) \arrow[rr] & & \mathbf{q}_n^{(i+1)} = \mathbf{q}_n^{(i)} + \delta\mathbf{q}_n \arrow[dl] \\ & i := i + 1 \arrow[ul] & \end{tikzcd} \tag{2.28}$$

Although it seems natural to proceed with the solutions in the sequence of increasing time instants, there is no formal restriction against any other order of solving the systems for different instants t_n in this case.

For path-dependent problems, the response at t_n depends on the response in previous time instants, i.e.

$$\mathbf{F}_n = \mathbf{F}_n(\mathbf{q}_n, \mathbf{q}_{n-1}, \dots, \mathbf{q}_1, \mathbf{p}). \tag{2.29}$$

Thus, the coefficient matrix $\mathbf{K}$ has the following form,

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{0} & \cdots & \mathbf{0} & \cdots \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \cdots & \mathbf{0} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \\ \mathbf{K}_{n1} & \mathbf{K}_{n2} & \cdots & \mathbf{K}_{nn} & \cdots \\ \vdots & \vdots & & \vdots & \ddots \end{bmatrix}, \qquad \mathbf{K}_{nm} = -\frac{\partial \mathbf{F}_n}{\partial \mathbf{q}_m}. \tag{2.30}$$

The systems for different time instants can still be separately solved for each $\mathbf{q}_n$, however, the solutions associated with the previous time instants $(t_m < t_n)$ are necessary to proceed with computations in the n-th step:

$$\begin{tikzcd} \mathbf{K}^{(i)}_{nn}\,\delta\mathbf{q}_n = \mathbf{F}_n(\mathbf{q}^{(i)}_n, \mathbf{q}_{n-1}, \dots, \mathbf{q}_1, \mathbf{p}) \arrow[rr] & & \mathbf{q}^{(i+1)}_n = \mathbf{q}^{(i)}_n + \delta\mathbf{q}_n \arrow[dl] \\ & i := i+1 \arrow[ul] & \end{tikzcd} \tag{2.31}$$

(as it can be seen, explicit determination of the matrices $\mathbf{K}_{nm}$, $m < n$, is not necessary). The solution strategy requires in this case that the systems are solved in the sequence of increasing time instants t_n and that results of the previous time instants are stored during the analysis. This increasing from step to step amount of storage can be frequently limited by expressing $\mathbf{F}_n$ in an equivalent form as

$$\mathbf{F}_n = \mathbf{F}_n(\mathbf{q}_n, \mathbf{s}_{n-1}, \mathbf{p}) \tag{2.32a}$$

where $\mathbf{s}_{n-1}$ is an array of discrete state variables, implicitly dependent on the history $(\mathbf{q}_{n-1}, \dots, \mathbf{q}_1)$, but explicitly defined by a recursive update formula

$$\mathbf{s}_n = \mathbf{s}_n(\mathbf{q}_n, \mathbf{s}_{n-1}, \mathbf{p}). \tag{2.32b}$$

This formulation allows to only store and update the array $\mathbf{s}$ (whose size is constant) instead of the entire history of $\mathbf{q}$ (whose size gradually increases during analysis).

It is noteworthy that, at the n-th time instant computations of a path-dependent problem, there is in fact no reason to conceptually distinguish between the input array $\mathbf{p}$ and the already known response arrays $\mathbf{q}_1, \dots, \mathbf{q}_{n-1}$ or their functions $\mathbf{s}_{n-1}$, which become input data to the current step computations in the same sense as $\mathbf{p}$. We can thus define an ‘extended’ input array

$$\mathbf{p}_n = \{\mathbf{p}, \mathbf{q}_1, \dots, \mathbf{q}_{n-1}\} \qquad \text{or} \qquad \mathbf{p}_n = \{\mathbf{p}, \mathbf{s}_{n-1}\} \tag{2.33}$$

and formulate the current step problem as

$$\mathbf{F}_n(\mathbf{q}_n, \mathbf{p}_n) = \mathbf{0}. \tag{2.34}$$

The corresponding iteration loop takes then the form

$$\begin{tikzcd} \mathbf{K}_{nn}^{(i)}\,\delta\mathbf{q}_n = \mathbf{F}_n(\mathbf{q}_n^{(i)}, \mathbf{p}_n) \arrow[r] & \mathbf{q}_n^{(i+1)} = \mathbf{q}_n^{(i)} + \delta\mathbf{q}_n \arrow[dl, bend left] \\ i := i + 1 \arrow[u, bend left] & \end{tikzcd} \tag{2.35}$$

Clearly, the path-independent formulation appears to be a special case of the above, with $\mathbf{p}_n = \mathbf{p}$.

2.2.2. Sensitivity problem: DDM and ASM

Let us recall Eqs. (2.18b) and (2.19) that allow to express the design gradients (sensitivities) of performances $\mathbf{G}$ as

$$\mathrm{d}_{\mathbf{h}}\mathbf{G} = \partial_{\mathbf{q}}\mathbf{G}\ \underbrace{\mathbf{K}^{-1}\,\partial_{\mathbf{h}}\mathbf{F}}_{\mathrm{d}_{\mathbf{h}}\mathbf{q}} + \partial_{\mathbf{h}}\mathbf{G}\,. \tag{2.36}$$

Let us also recall that the derivatives $\partial_{\mathbf{h}}\mathbf{F}$ and $\partial_{\mathbf{h}}\mathbf{G}$ appearing in this formula describe the *explicit* design dependence of $\mathbf{F}$ and $\mathbf{G}$, respectively, i.e. they include entire design-dependence through the input $\mathbf{p}(\mathbf{h})$ but *not* through the response $\mathbf{q}(\mathbf{h})$.

One of natural ways to determine the solution of performance design sensitivity analysis (DSA), further referred to as *direct differentiation method* (DDM), consists in determination of the response sensitivities $\mathrm{d}_{\mathbf{h}}\mathbf{q}$ from Eq. (2.18b)

$$\mathbf{K}\,\mathrm{d}_{\mathbf{h}}\mathbf{q} = \partial_{\mathbf{h}}\mathbf{F} \tag{2.37}$$

and substitution to Eq. (2.36).

As it was mentioned, Eq. (2.18b) constitutes a series of linear systems of N_d algebraic equations that must be solved against each particular design derivative $\mathrm{d}\mathbf{q}/\mathrm{d}h_i$ with corresponding right-hand side vectors $\partial\mathbf{F}/\partial h_i = (\partial\mathbf{F}/\partial\mathbf{p})(\partial\mathbf{p}/\partial h_i)$. The coefficient matrix is the same in each case and equal to the matrix computed in the Newton iteration loop (2.22) during the primary analysis. More strictly, it is equal to the matrix computed in the last iteration, when the solution predictor $\mathbf{q}^{(i)}$ best approximates the accurate solution $\mathbf{q}$.

There are three important consequences of the above observations. First, it is obvious that the sensitivity computations must be performed *after* the primary analysis computations. Otherwise it is not possible to determine the coefficient matrix $\mathbf{K}$, being generally a function of $\mathbf{q}$. Only in a linear case, DSA may go in parallel with the primary analysis, or even precede it.

The second consequence is related to the issue of numerical efficiency. Decomposition of the coefficient matrix in a large system of equations is usually the most time-consuming procedure in practical computations of structural mechanics. Having the already decomposed (during the primary analysis) matrix $\mathbf{K}$, we are able to perform the sensitivity analysis at a usually very low cost compared to the primary problem (however, proportional to the number of design variables)[1]. All that is necessary, is a series of back-substitutions of the decomposed matrix against N_d sensitivity right-hand side vectors $\partial_{\mathbf{h}}\mathbf{F}$.

The third consequence, already mentioned in Section 2.1.3, is the fact that, no matter how complex and nonlinear the primary problem is, the sensitivity problem is linear[2] and thus requires no iteration procedure. This strengthens even more the conclusion about efficiency of the DSA formulation vs. that of the primary analysis. Computational examples in Section 4.3 will demonstrate this advantage of the presented analytical method of sensitivity analysis.

In the case of path-independent transient problems, when the matrix $\mathbf{K}$ has the form (2.27), the sensitivity solution may be sought for separately at each time step,

$$\mathbf{K}_n \,\mathrm{d}_{\mathbf{h}}\mathbf{q}_n = \partial_{\mathbf{h}}\mathbf{F}_n \,, \tag{2.38}$$

right after completion of the primary analysis for this step (i.e. before the decomposed matrix $\mathbf{K}_n$ is removed from the computer memory).

In the case of path-dependent problems, cf. Eqs. (2.29), (2.30), we have, for the n-th time instant,

$$\mathbf{K}_{nn} \,\mathrm{d}_{\mathbf{h}}\mathbf{q}_n = \hat{\partial}_{\mathbf{h}}\mathbf{F}_n \,, \tag{2.39}$$

[1] Except for formulations with a diagonal matrix (e.g. explicit dynamics) which do not require decomposition and thus numerical costs of sensitivity analysis (per one design variable) are comparable to (or even higher than) those of the primary analysis, see Chapter 5 for more details.

[2] In path dependent problems, when the time-decoupled solution strategy (2.34) is utilized, we can actually only talk about linearity of DSA at a single time step (see Eqs. (2.39)–(2.40)) which, however, does not weaken the following conclusion about efficiency.

where the explicit design derivative $\hat{\partial}_{\mathbf{h}}\mathsf{F}_n = \mathrm{d}_{\mathbf{h}}\mathsf{F}_n|_{\text{frozen } \mathsf{q}_n}$ can be expressed as

$$\hat{\partial}_{\mathbf{h}}\mathsf{F}_n = \partial_{\mathbf{h}}\mathsf{F}_n - \sum_{m=1}^{n-1} \mathsf{K}_{nm}\, \mathrm{d}_{\mathbf{h}}\mathsf{q}_m \tag{2.40a}$$

or, in a formulation (2.32) with state variables s,

$$\hat{\partial}_{\mathbf{h}}\mathsf{F}_n = \partial_{\mathbf{h}}\mathsf{F}_n + \frac{\partial \mathsf{F}_n}{\partial \mathsf{s}_{n-1}}\, \mathrm{d}_{\mathbf{h}}\mathsf{s}_{n-1}\,. \tag{2.40b}$$

The sensitivity solution is again sought for right after completion of the primary analysis for this step, and it additionally requires the 'historical' sensitivity solutions $\mathrm{d}_{\mathbf{h}}\mathsf{q}_1\,, \ldots\,, \mathrm{d}_{\mathbf{h}}\mathsf{q}_{n-1}$ that have to be stored along with the primary solutions from preceding time steps, or the sensitivity $\mathrm{d}_{\mathbf{h}}\mathsf{s}_{n-1}$ that has to be stored and finally updated according to the formula

$$\mathrm{d}_{\mathbf{h}}\mathsf{s}_n = \frac{\partial \mathsf{s}_n}{\partial \mathsf{q}_n}\, \mathrm{d}_{\mathbf{h}}\mathsf{q}_n + \frac{\partial \mathsf{s}_n}{\partial \mathsf{s}_{n-1}}\, \mathrm{d}_{\mathbf{h}}\mathsf{s}_{n-1} + \frac{\partial \mathsf{s}_n}{\partial \mathsf{p}}\, \partial_{\mathbf{h}}\mathsf{p}\,. \tag{2.41}$$

(Distinguishing between explicit and implicit design derivatives of quantities related to preceding time steps may seem somewhat artificial in the above equations — in fact, all once computed *total* response design derivatives at $t < t_n$ become *explicit* input design derivatives to the formulations for $t \geq t_n$, according to the definition of 'extended' input p_n (2.33).)

An alternative approach to evaluation of the performance design sensitivities $\mathrm{d}_{\mathbf{h}}\mathbf{G}$, further referred to as *adjoint system method* (ASM), consists in determination of terms in Eq. (2.19) in the following, different order,

$$\mathrm{d}_{\mathbf{h}}\mathbf{G} = \underbrace{\partial_{\mathsf{q}}\mathbf{G}\,\mathsf{K}^{-1}}_{\boldsymbol{\lambda}^{\mathrm{T}}}\, \partial_{\mathbf{h}}\mathsf{F} + \partial_{\mathbf{h}}\mathbf{G}\,, \tag{2.42}$$

i.e. first solving the linear system of equations against the *adjoint response* array $\boldsymbol{\lambda}$,

$$\mathsf{K}^{\mathrm{T}}\boldsymbol{\lambda} = \partial_{\mathsf{q}}\mathbf{G}^{\mathrm{T}}, \tag{2.43}$$

and then substituting the result to Eq. (2.42).

The system (2.43) is in fact a series of N_p systems of equations with the same coefficient matrix K^{T} and with different right-hand sides. Thus, $\boldsymbol{\lambda}$ is an array of N_p solutions of the system.

From the numerical point of view, ASM and DDM are nearly the same. The same arrays ($\partial_{\mathsf{q}}\mathbf{G}$, $\partial_{\mathbf{h}}\mathsf{F}$, $\partial_{\mathbf{h}}\mathbf{G}$, K^{-1}) have to be assembled and similar operations are performed on them. In both the cases we have to do with linear systems of DSA equations, utilizing the coefficient matrix decomposed in the primary analysis (and then only transposed in the case of ASM). Our conclusions about efficiency drawn for DDM remain true for ASM, too. The only remarkable difference is the number of the back-substitution operations necessary to solve the systems (2.37) and (2.43), i.e. N_d and N_p, respectively. Thus, the main argument considered when choosing between DDM and ASM seems to be the computation cost: if $N_p > N_d$ then DDM is more favourable than ASM, and oppositely.

It appears, however, that this choice may be determined by other circumstances, too. Let us consider the case of transient problems with separation of equations and unknowns as in Eqs. (2.23)–(2.24). Analogously, we can write

$$\boldsymbol{\lambda} = \{\, \boldsymbol{\lambda}_0\,, \boldsymbol{\lambda}_1\,, \,\ldots\,, \boldsymbol{\lambda}_n\,, \,\ldots\}.$$

In the case of path-independent formulations, the coefficient matrix in the system (2.43) has the form (cf. Eq. (2.27))

$$\mathbf{K}^{\mathrm{T}} = \begin{bmatrix} \mathbf{K}_1^{\mathrm{T}} & \mathbf{0} & \cdots & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{K}_2^{\mathrm{T}} & \cdots & \mathbf{0} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{K}_n^{\mathrm{T}} & \cdots \\ \vdots & \vdots & & \vdots & \ddots \end{bmatrix} \tag{2.44}$$

and the solution is obtained in exactly the same way as in the case of DDM, i.e. the sensitivity sub-systems of equations

$$\mathbf{K}_n^{\mathrm{T}}\, \boldsymbol{\lambda}_n = \partial_{\mathbf{q}_n} \mathbf{G}^{\mathrm{T}},$$

have to be solved right after each time step primary computations.

In the case of path-dependent formulations, however, the analogy with DDM is not that straightforward. The coefficient matrix in the ASM system of equations (2.43) has the form (cf. Eq. (2.30))

$$\mathbf{K}^{\mathrm{T}} = \begin{bmatrix} \mathbf{K}_{11}^{\mathrm{T}} & \mathbf{K}_{21}^{\mathrm{T}} & \cdots & \mathbf{K}_{n1}^{\mathrm{T}} & \cdots \\ \mathbf{0} & \mathbf{K}_{22}^{\mathrm{T}} & \cdots & \mathbf{K}_{n2}^{\mathrm{T}} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{K}_{nn}^{\mathrm{T}} & \cdots \\ \vdots & \vdots & & \vdots & \ddots \end{bmatrix}.$$

This again allows to decouple the sub-systems of equations for different time instances t_n, but their solutions must proceed in the opposite sequence, starting from the last time instance towards t_0. This is obviously possible, but all advantages of DSA regarding efficiency are totally lost in this strategy. All arrays would have to be either once again assembled and all matrices $\mathbf{K}_{nn}$ decomposed (which is extremely time-consuming), or they would have to be stored after each time step of primary computations until their end (which requires unreasonably large computer storage resources). Thus, in path-dependent problems, ASM cannot be considered a suitable method of design sensitivity analysis. In our further detailed considerations related to sensitivity analysis in elasto-plastic computations, DDM will be the only approach discussed.

Let us finally mention, that the performance functionals in path dependent problems are frequently directly defined in terms of the state variables $\mathbf{s}$,

$$\mathbf{G} = \mathbf{G}(\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_n, \ldots, \mathbf{p}), \tag{2.45}$$

which in view of (2.32b) is a particular, recursive example of the general form $\mathbf{G} = \mathbf{G}(\mathbf{q}, \mathbf{p})$. In this case, Eq. (2.36) must be replaced by a series of update equations performed subsequently at each $t = t_n$,

$$\mathrm{d}_{\mathbf{h}}\mathbf{G}_n = \mathrm{d}_{\mathbf{h}}\mathbf{G}_{n-1} + \frac{\partial \mathbf{G}}{\partial \mathbf{s}_n}\, \mathrm{d}_{\mathbf{h}}\mathbf{s}_n \tag{2.46}$$

with the 'initial' setting $\mathrm{d}_{\mathbf{h}}\mathbf{G}_0 \equiv \partial_{\mathbf{h}}\mathbf{G} = (\partial \mathbf{G}/\partial \mathbf{p})\, \partial_{\mathbf{h}}\mathbf{p}$ and with the 'total' value of $\mathrm{d}_{\mathbf{h}}\mathbf{G}$ obtained only at the final time instant. Obviously, this update scheme makes only sense in DDM, where $\mathrm{d}_{\mathbf{h}}\mathbf{s}_n$ are directly computed at each $t = t_n$.

2.2.3. Analytical, semi-analytical and finite-difference sensitivity analysis

The methods of sensitivity analysis presented in the previous sections of this chapter are called analytical. The reason for this is the analytical form of the performance design derivatives sought for, both in the cases of distributed and discrete parameter formulations. In the latter case, the solution requires determination of arrays of certain analytical explicit design derivatives and then the unknown arrays of analytical sensitivities are evaluated from the solution of an algebraic system of equations.

The main advantage of the analytical formulations is their efficiency. DSA appears to be a linear problem (at least at a single time step) that does not require decomposition of any large coefficient matrix. There is a price to pay for this, though. The programming effort necessary to implement the methods in a numerical code is significant, frequently as extensive as implementation of the primary algorithm itself. Assembling arrays like $\partial_{\mathbf{h}}\mathsf{F}$ in Eq. (2.19) requires several derivations and writing a substantial amount of code. Thus, a justified question arises whether approximation of performance sensitivity with a finite difference formulae

$$\mathrm{d}_{\mathbf{h}}\mathbf{G} = \left[\frac{\partial \mathbf{G}}{\partial h_i}\right] \approx \left[\frac{\mathbf{G}(h_i+\Delta h_i) - \mathbf{G}(h_i)}{\Delta h_i}\right] \qquad \text{(forward f.-d. scheme)} \qquad (2.47a)$$

$$\mathrm{d}_{\mathbf{h}}\mathbf{G} = \left[\frac{\partial \mathbf{G}}{\partial h_i}\right] \approx \left[\frac{\mathbf{G}(h_i) - \mathbf{G}(h_i-\Delta h_i)}{\Delta h_i}\right] \qquad \text{(backward f.-d. scheme)} \quad (2.47b)$$

or, more accurately,

$$\mathrm{d}_{\mathbf{h}}\mathbf{G} \approx \left[\frac{\mathbf{G}(h_i+\Delta h_i) - \mathbf{G}(h_i-\Delta h_i)}{2\Delta h_i}\right] \qquad \text{(central f.-d. scheme)} \qquad (2.47c)$$

(for small perturbations Δh_i) would not be a better choice in practice. Assuming the methods and numerical procedures for the primary analysis are available on hand, the solutions for perturbed design can be readily evaluated by repeated execution of the primary analysis code.

There are two arguments against this choice. First is efficiency. Evaluation of the finite difference design sensitivities (2.47) requires one or two more runs of the primary analysis. Thus, the numerical cost of the DSA (per each design variable) compared to that of the primary analysis is 1 or 2, depending on whether the forward or central finite-difference scheme is applied. Recalling considerations from Section 2.2.2 about the coefficient matrix decomposition, we can conclude that this ratio should be only a small fraction of unity in the analytical method. And indeed it is, as will be shown later in computational examples. Thus, the finite difference approach to DSA is commonly criticized as numerically inefficient.

Another argument refers to accuracy, particularly to the choice of the perturbation Δh_i for each design parameter. The expressions (2.47) can be proved to converge to the analytical solution with $\Delta h_i \to 0$ which suggests that a sufficiently small perturbation should ensure good approximation. However, numerical roundings in the machine arithmetics require that the perturbation cannot be too small, too. Fitting the range of suitable perturbation sizes may be a problematic issue in practice.

Let us consider a simple computational example of a two-bar symmetric statically indeterminate truss (Fig. 2.4). Initial length of the bars is l_0, the distance a is a given data, and the vertical position h of the central node is a 'design'

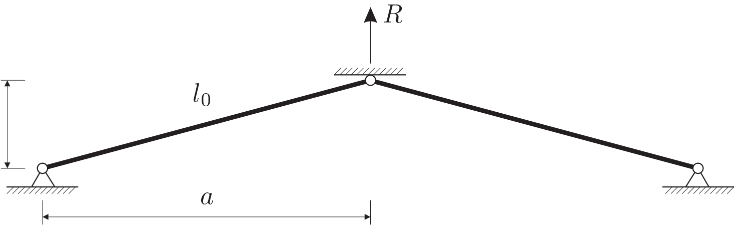


Figure 2.4. A statically indeterminate two-bar truss

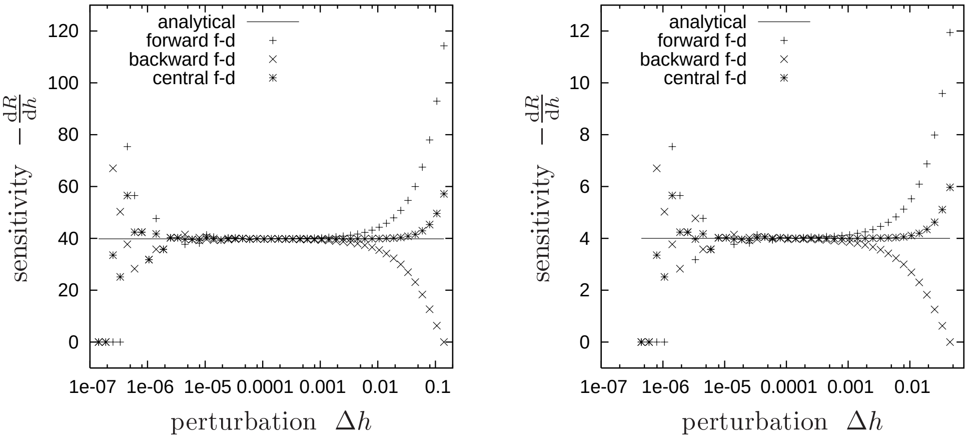


Figure 2.5. Results of sensitivity analysis at $h = \hat{h}$; left: for $a = 0.99\, l_0$ ($\hat{h} = 0.141\, l_0$), right: for $a = 0.999\, l_0$ ($\hat{h} = 0.045\, l_0$)

parameter that affects the value of the initial reaction force $R(h)$. If the elastic stiffness coefficient of each bar is k, then the reaction force is

$$R(h) = \frac{2k[l(h) - l_0]h}{l(h)}\,, \qquad l(h) = \sqrt{a^2 + h^2}\,,$$

and it takes the zero value at $h = \hat{h} = \sqrt{l_0^2 - a^2}$ which is the primary design value. (Perturbations of this value may be considered imperfections of the structure.)

Figure 2.5 presents approximate values of the reaction sensitivity with respect to h for two different initial geometries of the truss, computed with a numerical code with the finite-difference method and for various perturbation sizes. In the first test, for $\hat{h} = 0.141\, l_0$, the suitable range for Δh is about $[5\cdot 10^{-6}, 5\cdot 10^{-3}]$ for the forward and backward f.-d. schemes and is slightly broader for the central f.-d. scheme. It is thus quite a wide range but it may become much narrower when geometry of the structure is modified, see the second test with $\hat{h} = 0.045\, l_0$ in which the suitable range for Δh decreases to $[1 \cdot 10^{-5}, 2 \cdot 10^{-3}]$, i.e. by one order of magnitude. One can imagine structures and circumstances for which this range may be even narrower or not exist at all. The left limit of the range depends on the machine arithmetics precision, so it may be obviously moved far left by using double- instead of single-precision floating point number representation. But even then, an arbitrary choice of the perturbation size suffers from uncertainty about its correct placement in the suitable range, as there are no strict universal rules to estimate the range's limits. To make sure that an appropriate design perturbation has been chosen, the analyst should in fact perform in each case a number of numerical tests with different orders of magnitude of Δh and check whether reasonable results are obtained. This increases the numerical cost of finite-difference sensitivity analysis to even more than the above reported value of 1 or 2.

There is another approach to determination of sensitivity gradients that allows to enjoy several advantages of both analytical and finite-difference methods and avoid some of their drawbacks. This approach, called *semi-analytical method*, cf. [20, 36], consists in the use of finite difference formulae to only determine partial design derivatives $\partial_{\mathbf{h}}\mathsf{F}$ and $\partial_{\mathbf{h}}\mathsf{G}$ in Eq. (2.19) and then to proceed with the solution by either DDM or ASM as described in Section 2.2.2. In the forward finite difference scheme, we will have e.g.

$$\partial_{\mathbf{h}}\mathsf{F} = \left[\frac{\mathrm{d}\mathsf{F}}{\mathrm{d}h_i}\bigg|_{\text{frozen } \mathsf{q}} \right] \approx \left[\frac{\mathsf{F}(\mathsf{q}(\mathbf{h}), \mathsf{p}(\mathbf{h}{+}\Delta_i\mathbf{h})) - \mathsf{F}(\mathsf{q}(\mathbf{h}), \mathsf{p}(\mathbf{h}))}{\Delta h_i} \right]$$

where

$$\mathbf{h} + \Delta_i\mathbf{h} = \{\, h_1\,,\, h_2\,,\, \dots\,,\, h_{i-1}\,,\, h_i{+}\Delta h_i\,,\, h_{i+1}\,,\, \dots\,,\, h_{N_d}\,\}.$$

The semi-analytical sensitivity methods perfectly preserve efficiency of the analytical methods. Moreover, the implementation cost is very low compared to that of analytical DSA. These advantages make the methods very interesting for researchers and attract attention of software producers. Implementation of DSA in e.g. ABAQUS finite element system [1] is, in the current version of the program, entirely based on the semi-analytical approach.

On the other hand, all the above critical considerations regarding uncertainty about the proper choice of perturbation amount in the finite-difference approximations refer to the semi-analytical methods as well. Moreover, accuracy of the semi-analytical results, especially in the case of shape sensitivity, is reported to be additionally affected by details of numerical methods used in the discrete primary problem formulation, like number of elements [98, 101], or by specific features of structure deformation, like large rigid body motion [55]. There are several publications dealing with improvements of the latter drawback by introducing various correction factors and modifications to the finite-difference schemes,cf. e.g. [97, 98], or by computing certain selected terms in detailed expressions for the explicit derivatives $\partial_{\mathbf{h}}\mathsf{F}$ (like e.g. those related to rigid body motion) in an exact analytical way while all the other terms by finite-differences, cf. e.g. [55, 100]. All such improvements, however, result in increase in the necessary numerical implementation cost, which puts into question the fundamental advantage of the semi-analytical method.

2.2.4. Differentiability of response and uniqueness of the sensitivity solution

In the discussion of previous sections in this chapter the assumption has been made that all design gradients, implicit or explicit, exist, i.e. that all the functions and fields are differentiable with respect to design $\mathbf{h}$. It is known from practice that this frequently may not be the case, and the main reason is the non-differentiability of F with respect to the response q. An example can be analysis with contact boundary conditions or elasto-plastic analysis with instantaneous change of stiffness at the yield limit. This may imply non-differentiability of q with respect to design $\mathbf{h}$, too. Considering a primary solution at the very transition point between e.g. elastic and plastic range and assuming a small perturbation Δh_i, we may find the solution for perturbed design falling into the purely elastic range while the opposite perturbation $-\Delta h_i$ may drive the solution into the plastic range, i.e. beyond the yield point. In such a case, the two corresponding response perturbations would be generally different, not just in sign (as it would be in the case of smooth dependence) but in amount and direction, too.

Consider for example a uniaxially stretched bar, cf. Fig. 2.6, loaded with a stretching force $P(t) = \lambda t$ with λ given. The length and cross-sectional area of the bar are denoted by l and A, respectively. At a certain time instant $t = t_{\mathrm{y}}$, the

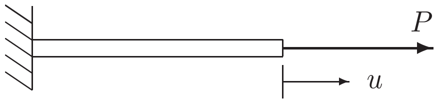


Figure 2.6. Uniaxially stretched elasto-plastic bar

yield limit is encountered and plastic flow continues for $t > t_\mathrm{y}$. Obviously, the bar stiffness changes abruptly at t_y and so may the response design derivatives. Even a rough analysis of the problem with a simplified 'double-modulus' elastic constitutive equation

$$\sigma = \begin{cases} E\,\varepsilon & \text{for } \varepsilon \leq \frac{\sigma_\mathrm{y}}{E}\,, \\ E_T\,\varepsilon + \sigma_\mathrm{y}\left(1 - \frac{E_T}{E}\right) & \text{otherwise,} \end{cases} \tag{2.48}$$

so that the displacement

$$u(t) = \begin{cases} \frac{P(t)\,l}{EA} & \text{for } t \leq t_\mathrm{y}, \ \text{ i.e. } P(t) \leq \sigma_\mathrm{y} A\,, \\ \frac{P(t)\,l}{E_T A} - \sigma_\mathrm{y} l \left(\frac{1}{E_T} - \frac{1}{E}\right) & \text{otherwise,} \end{cases} \tag{2.49}$$

shows that the displacement sensitivities $\mathrm{d}u/\mathrm{d}h_i$ with respect to e.g. the bar cross-sectional area $h_1 = A$ or initial yield limit $h_2 = \sigma_\mathrm{y}$, considered as functions of time, exhibit a discontinuous jump at the yield limit point t_y, see Fig. 2.7. In other words, the design derivatives, understood as

$$\frac{\mathrm{d}u}{\mathrm{d}h_i} = \lim_{\Delta h_i \to \pm 0} \frac{u(h_i{+}\Delta h_i) - u(h_i)}{\Delta h_i}$$

have at t_y two different limit values depending on the sign of Δh_i. Thus, in order to compute in such a case the design variation of u with respect to a given infinitesimal variation δh_i, one should compute the expression $(\mathrm{d}u/\mathrm{d}h_i)\,\delta h_i$ by choosing one of the two limits of $\mathrm{d}u/\mathrm{d}h_i$ in accordance with the sign of δh_i considered. The left-hand side limit (that is actually output as a result at t_y if the yield condition is strictly enforced — note the weak inequality in condition $(2.49)_1$) describes properly the sensitivity with respect to only such variations δh_i for which the perturbed structure, loaded with the same force $P(t{=}t_\mathrm{y})$, is still elastic at t_y (in both of the considered cases of $h_1 = A$ and $h_2 = \sigma_\mathrm{y}$, this takes place for $\delta h_i > 0$). However, if only t infinitesimally exceeds t_y, one obtains the sensitivity result that is valid for the remaining design variations. This can be written as

$$\delta u(t{=}t_\mathrm{y}) = \begin{cases} \left.\dfrac{\partial u}{\partial h_1}\right|_{t_\mathrm{y}^-} \delta h_1 + \left.\dfrac{\partial u}{\partial h_2}\right|_{t_\mathrm{y}^-} \delta h_2 & \text{for such } \delta h_i \text{ that } \sigma(h_i{+}\delta h_i) \leq \sigma_\mathrm{y}\,, \\ \left.\dfrac{\partial u}{\partial h_1}\right|_{t_\mathrm{y}^+} \delta h_1 + \left.\dfrac{\partial u}{\partial h_2}\right|_{t_\mathrm{y}^+} \delta h_2 & \text{otherwise,} \end{cases} \tag{2.50}$$

with $(\partial u/\partial h_i)|_{t_\mathrm{y}^-}$ and $(\partial u/\partial h_i)|_{t_\mathrm{y}^+}$ denoting left- and right-hand limits of $\partial u/\partial h_i$ at the discontinuity point.

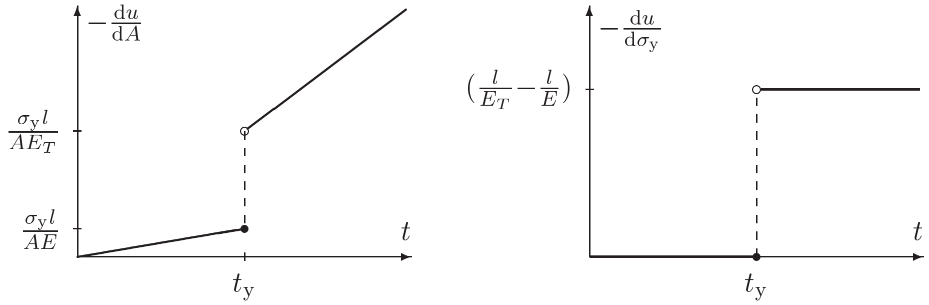


Figure 2.7. Elasto-plastic bar. Displacement sensitivities

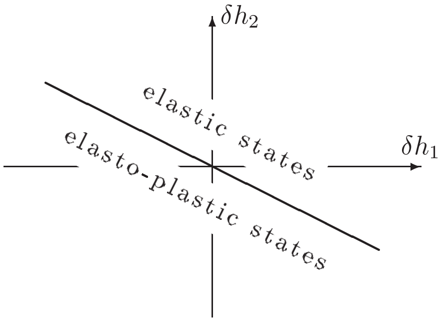


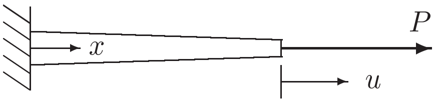
Figure 2.8. Elasto-plastic bar. Space of design variations at $t = t_{\mathrm{y}}$

Investigating Eq. (2.49) we can realize that the space of possible (infinitesimal) variations δh_i at $t = t_{\mathrm{y}}$ can be divided into two half-spaces corresponding to the cases of Eq. $(2.50)_1$ (elastic perturbed states) and $(2.50)_2$ (elasto-plastic perturbed states), see Fig. 2.8. This observation allows to employ the notion of Gâteaux differential (directional derivative) to describe design sensitivity at the discontinuous transition point. Generalizing the conclusion onto 2D and 3D problems with an arbitrary number of design parameters, we can state that the considerations regarding sensitivity computations in the preceding sections of this chapter remain valid also for such a case of non-differentiable response, upon the assumption that the design sensitivity $\mathrm{d}_{\mathbf{h}}(\cdot)$ is understood as a directional derivative in the Gâteaux sense. Detailed formulae have to be, however, reformulated in view of the fact that the frequently utilized chain rule of differentiation does not generally apply to Gâteaux differentials [41]. Another question that has to be risen is about the conditions upon which particular variations are associated with particular limit values of sensitivity.

These questions will not be discussed in this thesis, though, since their practical importance appears to be minor. There are two reasons for this.

First, in computational practice, it is very unlikely that we ever have to determine sensitivity at the transition point. To the contrary, in real cases of space- and time-discretized systems, probability of hitting the very limit point at the end of a typical time step (i.e. the precise fulfilment of the yield point condition $\bar{\sigma} = \sigma_{\mathrm{y}}$) is extremely low. Thus, numerical formulations based on a notion of regular design derivative are generally sufficient for sensitivity analysis, even in problems with locally non-differentiable response. Besides, even in an unlikely case of meeting such a point in computations, the 'regular' formulation is able to yield a solution which corresponds to one of the two limit values of the directional design derivative. Negligence of the other limit value is not a really fatal error, though. All we risk in such an instance, is that we are not able to correctly predict the response perturbation for a certain class of design perturbations, because we use the incorrect value of the sensitivity gradient. Note, however, that all 'good' sensitivity solutions obtained for neighbour points enjoying differentiable response may also be criticized for bearing the same kind of error, too. This is because the solution $\mathrm{d}_{\mathbf{h}}\mathbf{G}$, allowing to predict the response variation $\delta\mathbf{G}$ due to an infinitesimal design variation $\delta\mathbf{h}$, is usually employed in practice to predict finite response perturbations $\Delta\mathbf{G}$ corresponding to finite perturbations $\Delta\mathbf{h}$. If we occasionally choose such Δh that 'drives' the perturbed structure onto the other side of the transition point, such prediction will lead to erroneous prediction of the response perturbation. This may be considered a drawback of the method but, since it is its inherent feature, it cannot be overcome and the user have to be simply aware of such possible problems. Thus, the above mentioned problem of possible wrong choice of the design sensitivity limit value a the very transition point appears to be just another exemplification of this general weakness of the method.

Second, occurrences of gradient discontinuities are much less frequent and their magnitudes much lower than it might be suggested by the example considered in this section. Indeed, this is an extreme case, hardly ever met in reality, when the entire structure switches from elastic into plastic régime at one time instance. In the continuum mechanics such transition typically occurs gradually, so that at each time instant only isolated lines or surfaces are in the state of transition from elastic to plastic range. It is easy to verify, on a simple example, that the reported discontinuities in sensitivity do not occur in such a case. Let us consider a similar example of an uniaxially stretched bar whose cross-section area changes linearly along its length, Fig. 2.9. Applying the same proportional load, $P(t) = \lambda t$, we can predict that the solution will be elastic



$$A(x) = A_0(1-\tfrac{x}{l}) + A_1\tfrac{x}{l}$$

Figure 2.9. Uniaxially stretched elasto-plastic bar with variable cross-section

until $t = t_{\mathrm{y}1}$ corresponding to $P = \sigma_{\mathrm{y}} A_1$ and then, until $t = t_{\mathrm{y}2}$ corresponding to $P = \sigma_{\mathrm{y}} A_0$, the transition point will gradually move from right to left, taking for different values of P the location $\frac{x}{l} = \frac{\sigma_{\mathrm{y}} A_0 - P}{\sigma_{\mathrm{y}}(A_0 - A_1)}$. It is easy to verify that

$$u(t) = \begin{cases} \frac{P(t)\,l}{E(A_0-A_1)} \ln \frac{A_0}{A_1} & \text{for } t \leq t_{\mathrm{y}1} \\ \frac{P(t)\,l}{A_0-A_1} \left(\frac{1}{E_T} \ln \frac{P(t)}{\sigma_{\mathrm{y}} A_1} - \frac{1}{E} \ln \frac{P(t)}{\sigma_{\mathrm{y}} A_0} \right) - \frac{P(t)-\sigma_{\mathrm{y}} A_1}{A_0-A_1}\, l \left(\frac{1}{E_T} - \frac{1}{E} \right) & \\ & \text{for } t_{\mathrm{y}1} < t \leq t_{\mathrm{y}2} \\ \frac{P(t)\,l}{E_T(A_0-A_1)} \ln \frac{A_0}{A_1} - \sigma_{\mathrm{y}}\, l \left(\frac{1}{E_T} - \frac{1}{E} \right) & \text{otherwise.} \end{cases}$$

The above solution appears to be continuous and differentiable with respect to all input parameters.

This conclusion lends itself to easy generalizations towards 2D and 3D continuum elasto-plastic formulations. It must be noted, however, that in discrete parameter formulations, the transition from elastic into plastic range does not occur continuously. Actually, there are numerous but small non-differentiable 'jumps' associated with elasto-plastic transitions occurring at certain discrete material points. We might imagine them by replacing the bar in Fig. 2.9 with a bar consisting of a series of constant-cross-section short bars ordered monotonically from the thickest $A = A_0$ to the thinnest $A = A_1$ and connected to each other. Nevertheless, in realistic engineering computations, the discontinuities in sensitivity solution appear to be really small and possible errors limited, which will be shown in computational examples in Section 4.3. Such numerous but small discontinuities are also likely to appear in analyses of discrete-member structures, like trusses [77], not discussed in this study.

These conclusions allow to formulate the statement that possible discontinuities in sensitivity solutions do not usually require special treatment in practical numerical formulations. In other words, despite the lack of full mathematical justification, the formulation of design sensitivity analysis in terms of 'regular' definition of derivative is satisfactory for most non-trivial problems of elasto-plasticity, including those in which local non-differentiability of response is encountered.

Chapter 3

Elasto-plastic static equilibrium problem formulation

Formulation of DSA requires deep and detailed understanding of the primary problem formulation. The following chapter presents in a general form the primary problem of static deformation of an elasto-plastic body. Presented are constitutive equations of elasto-plasticity and elasto-viscoplasticity in both small and large deformations, and general finite element equations of the global path-dependent equilibrium problem. The contents of the chapter is merely a recapitulation of known theoretical and numerical formulations for problems of nonlinear mechanics with special attention paid to elasto-plasticity. Even though it virtually does not contain original contributions of the author, it seems indispensable to place the material in the thesis, as the following formulations of sensitivity in Chapter 4 extensively recall the notation and details of the formulation of the primary problem.

3.1. Continuum variational formulation of nonlinear statics

3.1.1. Geometry and boundary conditions

Let us consider a deformable body occupying in the initial undeformed configuration $\mathcal{C}^0$ the volume Ω^0 limited by the closed boundary surface $\partial\Omega^0$ (Fig. 3.1). Starting from the initial time instant $t = 0$, the body remains in a quasi-static[1] motion, assuming at each time instant t the configuration $\mathcal{C}^t$ and occupying then the volume Ω^t. Let us define the *reference configuration* $\mathcal{C}^r$ in which the body occupies the volume Ω^r and each material point of the body coincides with a point in the three-dimensional space defined by the Cartesian coordinates $\boldsymbol{X} = [X_i]$, $i = 1, 2, 3$. These coordinates define thus uniquely the material

[1] i.e. slow enough to neglect any inertial forces dependent on acceleration

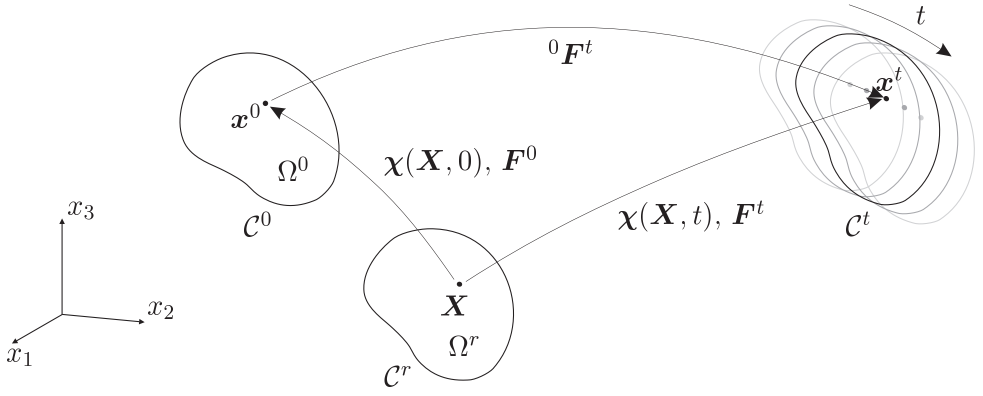


Figure 3.1. Configurations of a deformable body in motion

point. The reference configuration can be chosen arbitrarily — in the most natural case it is equivalent to the initial configuration $\mathcal{C}^0$, but it may also be defined as coincident with another, deformed configuration of the body or, possibly, as an abstract configuration, not associated with any stage of the body's deformation.

Instantaneous position of a material point $\boldsymbol{x}^t$ at any time instant t is defined by a unique transformation of the reference coordinates,

$$\boldsymbol{x}^t = \chi(\boldsymbol{X}, t) \tag{3.1}$$

Let us define, at any time instant t, the displacement field

$$\boldsymbol{u}^t \equiv \boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{x}^t - \boldsymbol{X} = \chi(\boldsymbol{X}, t) - \boldsymbol{X} \tag{3.2}$$

and the deformation gradient with its determinant

$$\boldsymbol{F}^t \equiv \boldsymbol{F}(\boldsymbol{X}, t) = \boldsymbol{\nabla}\boldsymbol{x}^t = \boldsymbol{I} + \boldsymbol{\nabla}\boldsymbol{u}^t\,, \qquad J^t = \det \boldsymbol{F}^t \tag{3.3}$$

where $\boldsymbol{\nabla}(\cdot) = \mathrm{d}(\cdot)/\mathrm{d}\boldsymbol{X} = [(\cdot)_{,i}]$ denotes the gradient operator in $\mathcal{C}^r$ and $\boldsymbol{I} = [\delta_{ij}]$ is the 2nd rank identity tensor. The above definitions may be generalized to time increments — the relative displacement and deformation gradient at a material point between the time instants $t = t_1$ and $t = t_2$ read

$$\begin{aligned} {}^{t_1}\boldsymbol{u}^{t_2} &= \boldsymbol{x}^{t_2} - \boldsymbol{x}^{t_1} = \boldsymbol{u}^{t_2} - \boldsymbol{u}^{t_1}, && (3.4)\\ {}^{t_1}\boldsymbol{F}^{t_2} &= \frac{\mathrm{d}\boldsymbol{x}^{t_2}}{\mathrm{d}\boldsymbol{x}^{t_1}} = \boldsymbol{F}^{t_2}(\boldsymbol{F}^{t_1})^{-1}, \qquad {}^{t_1}J^{t_2} = J^{t_2}/J^{t_1}\,. && (3.5) \end{aligned}$$

Unless stated otherwise, the reference configuration $\mathcal{C}^r$ in the following considerations will be identified with the initial configuration $\mathcal{C}^0$, which implies in particular that ${}^{0}\boldsymbol{u}^t \equiv \boldsymbol{u}^t$ and ${}^{0}\boldsymbol{F}^t \equiv \boldsymbol{F}^t$ for each time instant t.

External loads are applied as known transient distributions of volume forces $\hat{\boldsymbol{f}}(\boldsymbol{X},t)$ acting within Ω and surface stresses $\hat{\boldsymbol{t}}(\boldsymbol{X},t)$ acting on the part of the external surface $\partial\Omega_\sigma$. Both $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{t}}$ are related to unit volume and surface in the reference configuration $\mathcal{C}^r$, respectively. The remaining part of the external surface, $\partial\Omega_u$ ($\partial\Omega = \partial\Omega_\sigma \cup \partial\Omega_u$), is where prescribed displacements $\hat{\boldsymbol{u}}(\boldsymbol{X},t)$ are enforced. The division of $\partial\Omega$ into $\partial\Omega_\sigma$ and $\partial\Omega_u$ remains unchanged throughout the history of motion, i.e. $\partial\Omega^t_\sigma$ and $\partial\Omega^t_u$ become transient projections of points belonging to $\partial\Omega^r_\sigma$ and $\partial\Omega^r_u$, respectively, according to transformation (3.1).

Other mixed types of boundary conditions (e.g. surface contact conditions) will not be discussed, in order to preserve clarity of the presentation.

3.1.2. Virtual work equation

Without recalling the local partial differential equations governing the quasi-static equilibrium of a nonlinear continuum, let us write down the corresponding variational formulation. At each time instant t, the virtual work equation,

$$\int_{\Omega^r} \tilde{\boldsymbol{\sigma}} : \delta\tilde{\boldsymbol{\varepsilon}} \,\mathrm{d}\Omega = \int_{\Omega^r} \hat{\boldsymbol{f}}\,\delta\boldsymbol{u}\,\mathrm{d}\Omega + \int_{\partial\Omega^r} \hat{\boldsymbol{t}}\,\delta\boldsymbol{u}\,\mathrm{d}(\partial\Omega), \tag{3.6}$$

is required to hold for each kinematically admissible displacement field variation $\delta\boldsymbol{u}$, where $\delta\tilde{\boldsymbol{\varepsilon}}$ is a strain variation tensor associated with $\delta\boldsymbol{u}$, and $\tilde{\boldsymbol{\sigma}}$ is a stress tensor work-conjugate to $\delta\tilde{\boldsymbol{\varepsilon}}$. Even if $\hat{\boldsymbol{t}}$ is only defined at $\partial\Omega^r_\sigma$, the last integral in Eq. (3.6), taken over the entire boundary $\partial\Omega^r = \partial\Omega^r_\sigma \cup \partial\Omega^r_u$, is formally correct since, by the definition of kinematical admissibility, $\delta\boldsymbol{u} \equiv \boldsymbol{0}$ at $\partial\Omega_u$.

There is a variety of work-conjugate pairs $(\tilde{\boldsymbol{\sigma}}, \delta\tilde{\boldsymbol{\varepsilon}})$ that may be used in Eq. (3.6). For small deformations ($\boldsymbol{F} \approx \boldsymbol{I}$) these are the true Cauchy stress $\boldsymbol{\sigma}$ and the Cauchy strain variation $\delta\boldsymbol{\varepsilon}$,

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}, \qquad \delta\tilde{\boldsymbol{\varepsilon}} = \delta\boldsymbol{\varepsilon} = \tfrac{1}{2}\left[\boldsymbol{\nabla}\delta\boldsymbol{u} + (\boldsymbol{\nabla}\delta\boldsymbol{u})^{\mathrm{T}}\right] \tag{3.7a}$$

which corresponds to the definition of the linearized Cauchy strain,

$$\boldsymbol{\varepsilon} = \tfrac{1}{2}[\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^{\mathrm{T}}]. \tag{3.7b}$$

For large deformations, the following work-conjugate pairs will be considered in the subsequent derivations:

- The 2nd Piola–Kirchhoff stress tensor $\boldsymbol{T}$,

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{T} = J\,\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-\mathrm{T}}\,, \tag{3.8a}$$

and the Green strain variation $\delta\boldsymbol{E}$,

$$\delta\tilde{\boldsymbol{\varepsilon}} = \delta\boldsymbol{E} = \tfrac{1}{2}(\delta\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F} + \boldsymbol{F}^{\mathrm{T}}\delta\boldsymbol{F}), \tag{3.8b}$$

which corresponds to the following definition of the Green strain tensor itself,

$$\boldsymbol{E} = \tfrac{1}{2}(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F} - \boldsymbol{I}). \tag{3.8c}$$

- The Kirchhoff stress tensor $\boldsymbol{\tau}$,

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\tau} = J\,\boldsymbol{\sigma} = \boldsymbol{F}\boldsymbol{T}\boldsymbol{F}^{\mathrm{T}}\,, \tag{3.9a}$$

  and the variation $\delta\boldsymbol{\epsilon}$,

$$\delta\tilde{\boldsymbol{\varepsilon}} = \delta\boldsymbol{\epsilon} = \tfrac{1}{2}\left[\delta\boldsymbol{F}\,\boldsymbol{F}^{-1} + (\delta\boldsymbol{F}\,\boldsymbol{F}^{-1})^{\mathrm{T}}\right], \tag{3.9b}$$

  which, however, does *not* correspond to any closed-form definition of a strain tensor $\boldsymbol{\epsilon}$.

In all the three cases given above, both the stress $\tilde{\boldsymbol{\sigma}}$ and the strain variation $\delta\tilde{\boldsymbol{\varepsilon}}$ are symmetric, and the latter can be generally expressed as

$$\delta\tilde{\boldsymbol{\varepsilon}} = \tfrac{1}{2}\left(\delta\tilde{\boldsymbol{\varepsilon}}^{\mathrm{u}} + (\delta\tilde{\boldsymbol{\varepsilon}}^{\mathrm{u}})^{\mathrm{T}}\right), \qquad \delta\tilde{\boldsymbol{\varepsilon}}^{\mathrm{u}} = \begin{cases} \delta\boldsymbol{\varepsilon}^{\mathrm{u}} = \boldsymbol{\nabla}\delta\boldsymbol{u} & \text{for } \tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}, \\ \delta\boldsymbol{E}^{\mathrm{u}} = \boldsymbol{F}^{\mathrm{T}}\delta\boldsymbol{F} & \text{for } \tilde{\boldsymbol{\sigma}} = \boldsymbol{T}, \\ \delta\boldsymbol{\epsilon}^{\mathrm{u}} = \delta\boldsymbol{F}\,\boldsymbol{F}^{-1} & \text{for } \tilde{\boldsymbol{\sigma}} = \boldsymbol{\tau}. \end{cases} \tag{3.10}$$

Thus, the integrand in the l.h.s. of Eq. (3.6) can be equivalently written as

$$\tilde{\boldsymbol{\sigma}} : \delta\tilde{\boldsymbol{\varepsilon}} = \tilde{\boldsymbol{\sigma}} : \delta\tilde{\boldsymbol{\varepsilon}}^{\mathrm{u}} \tag{3.11}$$

which will be taken advantage of in later sections.

3.2. Constitutive relations

3.2.1. General formulations

The virtual work formulation presented in Section 3.1.2 can be employed to find the unknown transient fields $\boldsymbol{u}(\boldsymbol{X},\boldsymbol{\tau})$ and $\tilde{\boldsymbol{\sigma}}(\boldsymbol{X},\boldsymbol{\tau})$ provided that constitutive relations between stress $\tilde{\boldsymbol{\sigma}}$, deformation $\boldsymbol{F}$, and possibly a set of other (scalar or tensor) constitutive state variables $\mathbf{z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots\}$ and/or their rates $\dot{\mathbf{z}} = \{\dot{\boldsymbol{z}}_1, \dot{\boldsymbol{z}}_2, \ldots\}$ are given. A general form of such relations is

$$\boldsymbol{g}_j(\boldsymbol{F}, \dot{\boldsymbol{F}}, \tilde{\boldsymbol{\sigma}}, \dot{\tilde{\boldsymbol{\sigma}}}, \mathbf{z}, \dot{\mathbf{z}}) = \mathbf{0} \tag{3.12}$$

in which certain requirements of objectivity and symmetry must be additionally fulfilled.

In the case of elasticity, there are no additional constitutive state variables $\boldsymbol{z}_i$ ($\mathbf{z} = \emptyset$) and the relationship between stress and strain tensors is expressed with the use of a scalar function of the deformation called the elastic strain potential W,

$$\boldsymbol{T} = \frac{\mathrm{d}W}{\mathrm{d}\boldsymbol{E}}\,, \tag{3.13}$$

which for small deformations can be expressed as

$$\boldsymbol{\sigma} = \frac{\mathrm{d}\mathcal{W}}{\mathrm{d}\boldsymbol{\varepsilon}}\,, \tag{3.14}$$

where $\mathcal{W}$ is a linearized (at $\boldsymbol{F} \approx \boldsymbol{I}$) approximation of W. The virtual work equation (3.6) with such a constitutive model can be independently solved at each time instant t.

In the case of elasto-plasticity, Eq. (3.12) may assume the following, general rate-type form

$$\dot{\tilde{\boldsymbol{\sigma}}} = \dot{\tilde{\boldsymbol{\sigma}}}(\dot{\boldsymbol{F}}, \boldsymbol{F}, \mathbf{z}), \qquad \dot{\boldsymbol{z}}_i = \dot{\boldsymbol{z}}_i(\dot{\boldsymbol{F}}, \boldsymbol{F}, \mathbf{z}). \tag{3.15}$$

In this formulation, the constitutive equations (3.15) must be integrated in time, starting from $t = 0$ before substitution to Eq. (3.6), which finally yields the solution $(\boldsymbol{u}, \tilde{\boldsymbol{\sigma}}, \mathbf{z})$ as history-dependent transient fields. This obviously means that additional initial conditions must be defined for all the state fields and their rates at $t = 0$.

Relationship between rates of stress and strain is assumed linear, i.e., Eq. $(3.15)_1$ can be expressed for large deformations as

$$\dot{\boldsymbol{T}} = \mathfrak{C}^{\mathrm{tg}} : \dot{\boldsymbol{E}} + \boldsymbol{H} \tag{3.16}$$

where $\mathfrak{C}^{\mathrm{tg}} = \mathfrak{C}^{\mathrm{tg}}(\boldsymbol{F}, \mathbf{z})$ is the 4-th rank constitutive tangent stiffness tensor and $\boldsymbol{H} = \boldsymbol{H}(\boldsymbol{F}, \mathbf{z})$. Defining tensors of deformation velocity $\boldsymbol{l}$ and deformation rate $\boldsymbol{d}$ as

$$\boldsymbol{l} = \dot{\boldsymbol{F}}\boldsymbol{F}^{-1}\,, \qquad \boldsymbol{d} = \tfrac{1}{2}(\boldsymbol{l} + \boldsymbol{l}^{\mathrm{T}}), \tag{3.17}$$

(note analogy between the rate $\boldsymbol{d}$ and the variation $\delta\boldsymbol{\epsilon}$ in Eq. (3.9b)), we can express Eq. (3.16) in spatial tensor measures as

$$\mathcal{L}_v\boldsymbol{\tau} = \mathfrak{c}^{\mathrm{tg}} : \boldsymbol{d} + \boldsymbol{\eta} \tag{3.18}$$

where $\boldsymbol{\eta} = \boldsymbol{F}\boldsymbol{H}\boldsymbol{F}^{\mathrm{T}}$, the Lie derivative $\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\tau}$ is defined as

$$\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\tau} \equiv \boldsymbol{F}\dot{\boldsymbol{T}}\boldsymbol{F}^{\mathrm{T}} = \dot{\boldsymbol{\tau}} - \boldsymbol{l}\boldsymbol{\tau} - \boldsymbol{\tau}\boldsymbol{l}^{\mathrm{T}}, \tag{3.19}$$

and the 4th rank stiffness tensor $\mathfrak{c}^{\mathrm{tg}}$ can be expressed as

$$\mathfrak{c}^{\mathrm{tg}}_{ijkl} = F_{ip}F_{jq}F_{kr}F_{ls}\mathfrak{C}^{\mathrm{tg}}_{pqrs}\,. \tag{3.20}$$

Clearly, for small deformations ($\boldsymbol{F} \approx \boldsymbol{I}$), both Eqs. (3.16) and (3.18) reduce to

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{\mathcal{C}}^{\mathrm{tg}} : \dot{\boldsymbol{\varepsilon}} + \boldsymbol{\mathcal{H}} \tag{3.21}$$

where $\boldsymbol{\mathcal{C}}^{\mathrm{tg}}$ is a limit of both $\mathfrak{C}^{\mathrm{tg}}$ and $\mathfrak{c}^{\mathrm{tg}}$ at $\boldsymbol{F} \to \boldsymbol{I}$ while $\boldsymbol{\mathcal{H}}$ is a corresponding limit of both $\boldsymbol{H}$ and $\boldsymbol{\eta}$.

Differentiating in time Eqs. (3.13)–(3.14), we realize that the elasto-plastic rate-type constitutive equations (3.16)–(3.21) include pure elasticity as a special case, with $\boldsymbol{H} = \boldsymbol{\eta} = \boldsymbol{\mathcal{H}} = \boldsymbol{0}$ and

$$\mathfrak{C}^{\mathrm{tg}} = \mathfrak{C}^{\mathrm{e}} \equiv \frac{\mathrm{d}^2 W}{\mathrm{d}\boldsymbol{E}^2}\,, \qquad \boldsymbol{\mathcal{C}}^{\mathrm{tg}} = \boldsymbol{\mathcal{C}}^{\mathrm{e}} \equiv \frac{\mathrm{d}^2 \mathcal{W}}{\mathrm{d}\boldsymbol{\varepsilon}^2}\,. \tag{3.22}$$

With these settings, Eqs. (3.16), (3.18), and (3.21), become

$$\dot{\boldsymbol{T}} = \mathfrak{C}^{\mathrm{e}} : \dot{\boldsymbol{E}}, \qquad \mathcal{L}_{\boldsymbol{v}}\boldsymbol{\tau} = \mathfrak{c}^{\mathrm{e}} : \boldsymbol{d}, \qquad \dot{\boldsymbol{\sigma}} = \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \dot{\boldsymbol{\varepsilon}}, \tag{3.23}$$

with $\mathfrak{c}^{\mathrm{e}}_{ijkl} = F_{ip}F_{jq}F_{kr}F_{ls}\mathfrak{C}^{\mathrm{e}}_{pqrs}\,.$

3.2.2. Elasticity

Constitutive equations of the type (3.13) or (3.14) are called *hyperelastic*. Definition of the hyperelastic constitutive equation consists in providing a particular form of the elastic strain potential W. Two forms of W are considered in the thesis: (i) the *linear elastic strain potential* and (ii) the *modified neo-Hookean strain potential.*

The *linear elastic strain potential* has in the finite deformation range the form

$$W = \tfrac{1}{2}\,\boldsymbol{E} : \mathfrak{C}^{\mathrm{e}} : \boldsymbol{E}\,, \qquad \text{i.e.} \quad W = \tfrac{1}{2}\,E_{ij}\,\mathfrak{C}^{\mathrm{e}}_{ijkl}\,E_{kl} \tag{3.24}$$

where $\mathfrak{C}^{\mathrm{e}}$ is a constant tensor, here identical with the constitutive stiffness tensor $\mathfrak{C}^{\mathrm{tg}}$ defined in Eq. (3.22). Indeed,

$$\boldsymbol{T} = \mathfrak{C}^{\mathrm{e}} : \boldsymbol{E}\,, \qquad \dot{\boldsymbol{T}} = \mathfrak{C}^{\mathrm{e}} : \dot{\boldsymbol{E}}\,. \tag{3.25}$$

In the case of isotropy,

$$\mathfrak{C}^{\mathrm{e}} \;=\; K\,\boldsymbol{I} \otimes \boldsymbol{I} + 2G\left(\mathfrak{I} - \tfrac{1}{3}\boldsymbol{I} \otimes \boldsymbol{I}\right)\,, \tag{3.26a}$$

$$\mathfrak{C}^{\mathrm{e}}_{ijkl} \;=\; \left(K - \tfrac{2}{3}G\right)\delta_{ij}\delta_{kl} + G(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})\,, \tag{3.26b}$$

where $\boldsymbol{\mathfrak{I}}$ is the fourth rank identity tensor and K, G are the isotropic bulk and shear moduli, respectively, related to the engineering elastic constants E, ν as

$$G = \frac{E}{2(1+\nu)}\,, \qquad K = \frac{E}{3(1-2\nu)}\,. \tag{3.27}$$

Thus

$$\boldsymbol{T} = K(\mathrm{tr}\boldsymbol{E})\,\boldsymbol{I} + 2G\,\mathrm{dev}\boldsymbol{E}\,, \qquad \dot{\boldsymbol{T}} = K(\mathrm{tr}\dot{\boldsymbol{E}})\,\boldsymbol{I} + 2G\,\mathrm{dev}\dot{\boldsymbol{E}} \tag{3.28}$$

where the deviator $\mathrm{dev}\boldsymbol{E} = \big(\boldsymbol{\mathfrak{I}} - \frac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\big) : \boldsymbol{E}$.

For small deformations, Eqs. (3.24)–(3.25) hold for the tensors $\boldsymbol{T}$ and $\boldsymbol{E}$ replaced by $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$, respectively, e.g.,

$$\mathcal{W} = \tfrac{1}{2}\,\boldsymbol{\varepsilon} : \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \boldsymbol{\varepsilon}\,, \qquad \boldsymbol{\sigma} = \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \boldsymbol{\varepsilon}\,, \qquad \dot{\boldsymbol{\sigma}} = \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \dot{\boldsymbol{\varepsilon}} \tag{3.29}$$

(where the 'linearized' tensor $\boldsymbol{\mathcal{C}}^{\mathrm{e}}$ in this case exactly equals $\boldsymbol{\mathfrak{C}}^{\mathrm{e}}$), and, particularly, for isotropy (3.26)

$$\boldsymbol{\sigma} = K(\mathrm{tr}\boldsymbol{\varepsilon})\,\boldsymbol{I} + 2G\,\mathrm{dev}\boldsymbol{\varepsilon}\,, \qquad \dot{\boldsymbol{\sigma}} = K(\mathrm{tr}\dot{\boldsymbol{\varepsilon}})\,\boldsymbol{I} + 2G\,\mathrm{dev}\dot{\boldsymbol{\varepsilon}}\,. \tag{3.30}$$

Another isotropic form of the elastic strain potential, suitable for the finite-deformation elasto-plastic formulations, is the *modified neo-Hookean potential*, cf. [110]. Let us recall definitions of the right and left Cauchy–Green deformation tensors,

$$\boldsymbol{C} = \boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}\,, \qquad \boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^{\mathrm{T}}\,, \tag{3.31}$$

so that, cf. Eqs. (3.8c) and (3.13),

$$\boldsymbol{E} = \tfrac{1}{2}(\boldsymbol{C} - \boldsymbol{I}), \qquad \boldsymbol{T} = 2\,\frac{\mathrm{d}W}{\mathrm{d}\boldsymbol{C}}\,. \tag{3.32}$$

Let us also split the deformation gradient $\boldsymbol{F}$ into the volumetric and isochoric part,

$$\boldsymbol{F} = \boldsymbol{F}^{\mathrm{v}}\,\bar{\boldsymbol{F}}\,, \tag{3.33}$$

where

$$\boldsymbol{F}^{\mathrm{v}} = J^{\frac{1}{3}}\boldsymbol{I}\,, \quad \det\boldsymbol{F}^{\mathrm{v}} = J, \qquad \bar{\boldsymbol{F}} = J^{-\frac{1}{3}}\boldsymbol{F}\,, \quad \det\bar{\boldsymbol{F}} = 1\,, \tag{3.34}$$

so that isochoric deformation tensors can be defined as

$$\bar{\boldsymbol{C}} = \bar{\boldsymbol{F}}^{\mathrm{T}}\bar{\boldsymbol{F}} = J^{-\frac{2}{3}}\boldsymbol{C}\,, \qquad \bar{\boldsymbol{b}} = \bar{\boldsymbol{F}}\bar{\boldsymbol{F}}^{\mathrm{T}} = J^{-\frac{2}{3}}\boldsymbol{b}\,. \tag{3.35}$$

The elastic strain potential is assumed in the form

$$W = \tfrac{1}{2}K\left[\tfrac{1}{2}(J^2{-}1) - \ln J\right] + \tfrac{1}{2}G\left(\mathrm{tr}\bar{\boldsymbol{b}} - 3\right) \tag{3.36}$$

in which the volumetric and isochoric modes of deformation have been explicitly separated. Since $\mathrm{tr}\bar{\boldsymbol{b}} = \mathrm{tr}\bar{\boldsymbol{C}} = J^{-\frac{2}{3}}\,\mathrm{tr}\boldsymbol{C}$ and $\mathrm{d}J = J\,\mathrm{tr}(\mathrm{d}\boldsymbol{F}\boldsymbol{F}^{-1}) = \frac{1}{2}\,J\boldsymbol{C}^{-1} : \boldsymbol{C}$, we can differentiate W with respect to $\boldsymbol{C}$, cf. Eq. $(3.32)_2$, and obtain

$$\boldsymbol{T} = \tfrac{1}{2}K(J^2{-}1)\,\boldsymbol{C}^{-1} + G\,\boldsymbol{F}^{-1}\,\mathrm{dev}\bar{\boldsymbol{b}}\,\boldsymbol{F}^{-\mathrm{T}}, \tag{3.37}$$

which can be written in a more compact form in terms of the Kirchhoff stress tensor as

$$\boldsymbol{\tau} = \tfrac{1}{2}K(J^2{-}1)\,\boldsymbol{I} + G\,\mathrm{dev}\bar{\boldsymbol{b}}\,. \tag{3.38}$$

The rate-type form of the constitutive equation (3.38) can be written in a convenient way in terms of the Lie derivative of $\boldsymbol{\tau}$, cf. Eq. $(3.23)_2$

$$\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\tau} = \mathfrak{c}^{\mathrm{e}} : \boldsymbol{d} \tag{3.39}$$

where

$$\begin{aligned}\mathfrak{c}^{\mathrm{e}} \;=\; & K\left[J^2\,\boldsymbol{I}\otimes\boldsymbol{I} - (J^2{-}1)\,\boldsymbol{\mathfrak{I}}\right] \\ & + \tfrac{2}{3}G\left[(\mathrm{tr}\bar{\boldsymbol{b}})\big(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\big) - \big(\mathrm{dev}(\bar{\boldsymbol{b}})\otimes\boldsymbol{I} + \boldsymbol{I}\otimes\mathrm{dev}(\bar{\boldsymbol{b}})\big)\right].\end{aligned} \tag{3.40}$$

Assumption of small deformations ($\boldsymbol{F} \approx \boldsymbol{I}$, $J \approx 1$) makes the neo-Hookean model converge to linear isotropic elasticity, cf. Eq. (3.30).

3.2.3. Elasto-plasticity at small deformations

The displacement field $\boldsymbol{u}$ in this formulation is assumed to be a sum of the elastic and inelastic parts

$$\boldsymbol{u} = \boldsymbol{u}^{\mathrm{e}} + \boldsymbol{u}^{\mathrm{p}}. \tag{3.41}$$

Defining, in analogy to Eq. (3.7b) $\boldsymbol{\varepsilon}^{\mathrm{e}} = \frac{1}{2}[\boldsymbol{\nabla}\boldsymbol{u}^{\mathrm{e}} + (\boldsymbol{\nabla}\boldsymbol{u}^{\mathrm{e}})^{\mathrm{T}}]$ and $\boldsymbol{\varepsilon}^{\mathrm{p}} = \frac{1}{2}[\boldsymbol{\nabla}\boldsymbol{u}^{\mathrm{p}} + (\boldsymbol{\nabla}\boldsymbol{u}^{\mathrm{p}})^{\mathrm{T}}]$, we can express the material linearized strain rate $\dot{\boldsymbol{\varepsilon}}$ as

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{\mathrm{e}} + \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} \tag{3.42}$$

where the first, elastic component remains in the linear relation to the stress rate, cf. Eq. $(3.23)_3$,

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \dot{\boldsymbol{\varepsilon}}^{\mathrm{e}} \tag{3.43}$$

while the second is either zero (elastic deformation) or obeys the rate-type flow rule (elasto-plastic deformation). Purely elastic deformation occurs whenever

$$f(\boldsymbol{\sigma}, \boldsymbol{\alpha}, \bar{e}^{\mathrm{p}}) < 0 \tag{3.44}$$

where f denotes the yield function in the form

$$f(\boldsymbol{\sigma}, \boldsymbol{\alpha}, \bar{e}^{\mathrm{p}}) = \bar{\sigma}(\boldsymbol{\sigma}, \boldsymbol{\alpha}) - \kappa(\bar{e}^{\mathrm{p}}) \tag{3.45}$$

while the Ziegler–Prager's back stress $\boldsymbol{\alpha}$ and equivalent plastic strain $\bar{e}^{\mathrm{p}}$ are internal plastic state parameters. The isotropic hardening function $\kappa(\bar{e}^{\mathrm{p}})$ is a known material function taking the value of σ_{y} (initial yield limit) at $\bar{e}^{\mathrm{p}} = 0$, and

$$\bar{\sigma} = \sqrt{\tfrac{3}{2}}\, \|\boldsymbol{s}\| = \sqrt{\tfrac{3}{2}\, \boldsymbol{s} : \boldsymbol{s}}\,, \qquad \boldsymbol{s} = \operatorname{dev}(\boldsymbol{\sigma} - \boldsymbol{\alpha}), \tag{3.46}$$

is the Huber–Mises equivalent stress, being an invariant of the deviatoric 'relative' stress $\boldsymbol{s}$.

If $f \geq 0$ then plastic flow occurs. The associative flow rule, conforming with most metal plasticity observations, defines the plastic strain rate as proportional to the normal direction to the current yield surface f in the stress space,

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} = \dot{\bar{e}}^{\mathrm{p}} \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\sigma}} = \sqrt{\frac{3}{2}}\, \dot{\bar{e}}^{\mathrm{p}}\, \boldsymbol{n}\,, \qquad \boldsymbol{n} = \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|} = \sqrt{\frac{3}{2}} \frac{\boldsymbol{s}}{\bar{\sigma}}\,. \tag{3.47}$$

The above rule implies that the plastic strain $\boldsymbol{\varepsilon}^{\mathrm{p}}$ is deviatoric (isochoric) which again agrees with observations.

Evolution of the back stress $\boldsymbol{\alpha}$, defining the translation of the yield surface in the stress space, is described by another associative rule [138],

$$\dot{\boldsymbol{\alpha}} = \dot{\mu}(\boldsymbol{\sigma} - \boldsymbol{\alpha}) \tag{3.48}$$

which, in view of Eq. (3.47), may also be written as

$$\operatorname{dev} \dot{\boldsymbol{\alpha}} = \frac{2}{3} H' \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} \tag{3.49}$$

where $H' \equiv \mathrm{d}H/\mathrm{d}\bar{e}^{\mathrm{p}}$ and the kinematic hardening function $H(\bar{e}^{\mathrm{p}})$ is another known material function. Note that only the deviatoric part of the back stress $\boldsymbol{\alpha}$ is in fact the state variable — its spherical part does not enter the formulation and is beyond our interest.

What still remains to be defined in the constitutive model, is the evolution rule of the equivalent plastic strain $\bar{e}^{\mathrm{p}}$, being the measure of the advance in plastic deformation. There is a variety of such rules proposed in the literature. A number of them will be presented below.

Rate-independent model

In this model of plastic flow, the yield function f is assumed to never exceed zero, and the evolution of $\bar{e}^{\mathrm{p}}$ results directly from the consistency condition

$$\dot{f} = 0. \tag{3.50}$$

Taking the time derivative of Eqs. (3.45)–(3.46) and employing (3.47), (3.49), we obtain

$$\dot{f} = \frac{3}{2\bar{\sigma}}\,\boldsymbol{s} : (\dot{\boldsymbol{\sigma}} - \dot{\boldsymbol{\alpha}}) - \kappa'\,\dot{\bar{e}}^{\mathrm{p}} = \frac{3}{2\bar{\sigma}}\,\boldsymbol{s} : \dot{\boldsymbol{\sigma}} - (\kappa' + H')\dot{\bar{e}}^{\mathrm{p}}, \tag{3.51}$$

which, when set to 0, yields

$$\dot{\bar{e}}^{\mathrm{p}} = \frac{3\,\boldsymbol{s} : \dot{\boldsymbol{\sigma}}}{2\bar{\sigma}(\kappa'{+}H')} = \sqrt{\frac{3}{2}}\,\frac{\boldsymbol{n} : \dot{\boldsymbol{\sigma}}}{\kappa'{+}H'}\,. \tag{3.52}$$

Substituting it into Eq. (3.47) and employing Eqs. (3.42)–(3.43) we get

$$\dot{\bar{e}}^{\mathrm{p}} = \sqrt{\frac{2}{3}}\,\frac{\boldsymbol{n} : \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \dot{\boldsymbol{\varepsilon}}}{h + \boldsymbol{n} : \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \boldsymbol{n}}\,, \qquad h = \tfrac{2}{3}(\kappa'{+}H')\,, \tag{3.53}$$

which finally allows to write the rate-type constitutive equation as, cf. Eq. (3.21),

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{\mathcal{C}}^{\mathrm{e}} : (\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}}) = \boldsymbol{\mathcal{C}}^{\mathrm{tg}} : \dot{\boldsymbol{\varepsilon}}\,, \tag{3.54}$$

where the elasto-plastic constitutive tangent tensor $\boldsymbol{\mathcal{C}}^{\mathrm{tg}}$ is expressed as

$$\boldsymbol{\mathcal{C}}^{\mathrm{tg}} = \boldsymbol{\mathcal{C}}^{\mathrm{e}} - \frac{(\boldsymbol{\mathcal{C}}^{\mathrm{e}} : \boldsymbol{n}) \otimes (\boldsymbol{n} : \boldsymbol{\mathcal{C}}^{\mathrm{e}})}{h + \boldsymbol{n} : \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \boldsymbol{n}} \tag{3.55a}$$

or

$$\mathcal{C}^{\mathrm{tg}}_{ijkl} = \mathcal{C}^{\mathrm{e}}_{ijkl} - \frac{\mathcal{C}^{\mathrm{e}}_{ijmn} n_{mn} n_{pq} \mathcal{C}^{\mathrm{e}}_{pqkl}}{h + n_{rs}\mathcal{C}^{\mathrm{e}}_{rstu} n_{tu}}\,. \tag{3.55b}$$

In the case of isotropic elasticity, i.e., if $\boldsymbol{\mathcal{C}}^{\mathrm{e}}$ is expressed with Eq. (3.26), this can be reduced to

$$\boldsymbol{\mathcal{C}}^{\mathrm{tg}} = \boldsymbol{\mathcal{C}}^{\mathrm{e}} - 2G\gamma\,\boldsymbol{n} \otimes \boldsymbol{n}\,, \qquad \gamma = \frac{3G}{\kappa'{+}H'{+}3G}\,. \tag{3.56}$$

Rate-dependent (elasto-viscoplastic) models

Here, the states corresponding to $f > 0$ are allowed and the evolution equation for $\bar{e}^{\mathrm{p}}$ is provided as an additional condition in the general form

$$\dot{\bar{e}}^{\mathrm{p}} = g(\bar{\sigma}, \bar{e}^{\mathrm{p}})\,. \tag{3.57}$$

In this case the rate-type constitutive equation can be expressed as, cf. Eq. (3.21),

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{\mathcal{C}}^{\mathrm{tg}} : \dot{\boldsymbol{\varepsilon}} + \boldsymbol{\mathcal{H}}\,, \qquad \boldsymbol{\mathcal{C}}^{\mathrm{tg}} = \boldsymbol{\mathcal{C}}^{\mathrm{e}}, \qquad \boldsymbol{\mathcal{H}} = -\sqrt{\tfrac{3}{2}}\, g(\bar{\sigma}, \bar{e}^{\mathrm{p}})\, \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \boldsymbol{n}\,. \tag{3.58}$$

There is a variety of particular forms of the function $g(\bar{\sigma}, \bar{e}^{\mathrm{p}})$ used in engineering practice. In this study, the following two will be discussed as examples:

- The *overstress* (Perzyna) model [102],

  $$g(\bar{\sigma}, \bar{e}^{\mathrm{p}}) = \frac{1}{\mu}\left(\frac{f(\bar{\sigma}, \bar{e}^{\mathrm{p}})}{\kappa(\bar{e}^{\mathrm{p}})}\right)^{m}, \qquad \text{for} \quad f \geq 0, \quad \text{otherwise} \quad g = 0, \tag{3.59}$$

  where $f(\bar{\sigma}, \bar{e}^{\mathrm{p}})$ is the yield function given by Eq. (3.45),

  $$f(\bar{\sigma}, \bar{e}^{\mathrm{p}}) = \bar{\sigma} - \kappa(\bar{e}^{\mathrm{p}}), \tag{3.60}$$

  and μ, m are material constants.

- The *power-law strain and strain-rate hardening* model [103],

  $$g(\bar{\sigma}, \bar{e}^{\mathrm{p}}) = \dot{\varepsilon}_0 \left[\frac{\bar{\sigma}}{\bar{g}(\bar{e}^{\mathrm{p}})}\right]^{\frac{1}{m}}, \qquad \bar{g}(\bar{e}^{\mathrm{p}}) = E\varepsilon_0 \left(1 + \frac{\bar{e}^{\mathrm{p}}}{\varepsilon_0}\right)^{\frac{1}{n}}, \tag{3.61}$$

  where E is the Young modulus and $\dot{\varepsilon}_0$, ε_0, m, n are material constants. The original formulation of this viscoplastic model did not include kinematic hardening ($\boldsymbol{s} \equiv \mathrm{dev}\boldsymbol{\sigma}$), however, there are no formal restrictions against the generalization $(3.46)_2$. Note that there is no yield limit in this case and the plastic flow may occur at any stress state, i.e. $\kappa \equiv 0$ should be assumed in the definition of the yield function f, so the condition (3.44) never applies.

Other forms of the viscoplastic evolution function (3.57) were also given by Hart [39], Anand [2], Chaboche [19], and others. Some of them require in fact some generalization of the model described above, introducing e.g. additional state variables [2] or combining the evolution equation of the back-stress $\boldsymbol{\alpha}$ with that of $\bar{e}^{\mathrm{p}}$ [19].

If the yield condition is not fulfilled (i.e. if $f < 0$, in material models to which it concerns), then the evolution equation reduces in both rate-independent and rate-dependent models to $\dot{\bar{e}}^{\mathrm{p}} = 0$ and Eqs. (3.54)–(3.58) are replaced with the elastic constitutive model $(3.23)_3$, $\dot{\boldsymbol{\sigma}} = \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \dot{\boldsymbol{\varepsilon}}$.

The constitutive model presented above is a particular form of the general constitutive formulation (3.15) expressed in terms of Cauchy stress and strain measures, with the set of internal state parameters

$$\mathbf{z} = \{\operatorname{dev} \boldsymbol{\sigma}, \operatorname{dev} \boldsymbol{\alpha}, \bar{e}^{\mathrm{p}}\} . \tag{3.62}$$

In the specific case of $H \equiv 0$ we can set $\boldsymbol{\alpha} \equiv \mathbf{0}$ and remove it from the set of state parameters $\mathbf{z}$ — the material does not exhibit kinematic hardening in such a case.

3.2.4. Elasto-plasticity at large deformations

The constitutive theory presented in Section 3.2.3 may be easily extended to large deformation case by simply substituting the linearized strains $\boldsymbol{\varepsilon}$ by the Green strain $\boldsymbol{E}$, the Cauchy stresses $\boldsymbol{\sigma}$, $\boldsymbol{\alpha}$ by their 2nd Piola–Kirchhoff counterparts $\boldsymbol{T}$, $\boldsymbol{A}$, and retaining all the other equations in the unchanged form. Although, in view of the definition of the Green strain (3.8c), there may be problems with consistent definitions of the elastic and plastic strains $\boldsymbol{E}^{\mathrm{e}}$, $\boldsymbol{E}^{\mathrm{p}}$ in the so modified additive equation (3.42),

$$\dot{\boldsymbol{E}} = \dot{\boldsymbol{E}}^{\mathrm{e}} + \dot{\boldsymbol{E}}^{\mathrm{p}}, \tag{3.63}$$

the simple concept is mathematically correct and it has been advocated by Casey [18] for the cases of small elastic/moderate plastic strains (and vice versa) as well as of small strains/moderate rotations (and vice versa). It is particularly suitable for purely elastic problems with large deformations and with the strain energy potential defined in terms of the Green strain, $W(\boldsymbol{E})$. Its great advantage is the ability to reuse the small deformation constitutive equations without changes, after only redefinition of selected terms to include nonlinear geometry.

The rate-type constitutive equation takes in such a case the form (3.16), with

$$\mathfrak{C}^{\mathrm{tg}} = \mathfrak{C}^{\mathrm{e}} - \frac{(\mathfrak{C}^{\mathrm{e}} : \boldsymbol{N}) \otimes (\boldsymbol{N} : \mathfrak{C}^{\mathrm{e}})}{h + \boldsymbol{N} : \mathfrak{C}^{\mathrm{e}} : \boldsymbol{N}}\,, \qquad \boldsymbol{H} = \boldsymbol{0}, \tag{3.64}$$

for the rate-independent model, cf. Eq. (3.55), or

$$\mathfrak{C}^{\mathrm{tg}} = \mathfrak{C}^{\mathrm{e}}, \qquad \boldsymbol{H} = -\sqrt{\tfrac{3}{2}}\, g(\bar{\sigma}, \bar{e}^{\mathrm{p}})\, \mathfrak{C}^{\mathrm{e}} : \boldsymbol{N}\,. \tag{3.65}$$

for rate-dependent models, cf. Eq. (3.58), where $\boldsymbol{N} = \boldsymbol{S}/\|\boldsymbol{S}\|$, $\boldsymbol{S} = \operatorname{dev}(\boldsymbol{T}-\boldsymbol{A})$.

In the general case of large elasto-plastic deformations, however, this approach cannot be employed as it does not comply with observations regarding the physical nature of elasto-plasticity. Actually, reversible elastic deformations of metal crystals are superimposed on irreversible plastic deformations due to dislocation motion in the crystals. This can be described by the multiplicative decomposition of the deformation gradient into the elastic and plastic parts, first proposed by Lee [75],

$$\boldsymbol{F} = \boldsymbol{F}^{\mathrm{e}} \boldsymbol{F}^{\mathrm{p}}\,, \qquad \boldsymbol{F}^{\mathrm{e}} = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\tilde{\boldsymbol{x}}}\,, \qquad \boldsymbol{F}^{\mathrm{p}} = \frac{\mathrm{d}\tilde{\boldsymbol{x}}}{\mathrm{d}\boldsymbol{X}}\,, \tag{3.66}$$

which corresponds to the kinematical scheme depicted in Fig. 3.2 (with the initial configuration considered as the reference configuration). Here, $\tilde{\mathcal{C}}^t$ is a hypothetical intermediate stress-free configuration that moves along with the deformed body.

The split (3.66) is not unique because the stress-free configuration may undergo arbitrary rigid-body rotations. Thus, detailed constitutive formulations

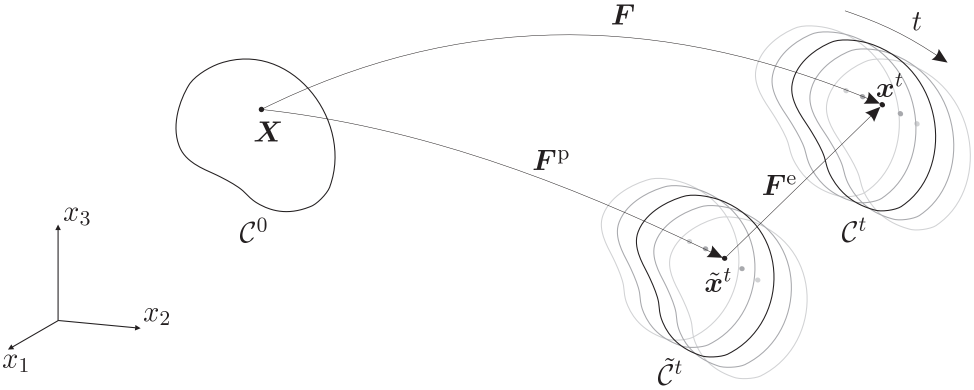


Figure 3.2. Configurations of a body in elasto-plastic deformation

based on the concept differ from each other in the way the rotation is assigned to elastic or plastic deformation, and in the consequent definitions of back- and/or forward-rotated measures of certain tensorial quantities used in the equations. Defining, in analogy to Eq. (3.17), elastic and plastic deformation velocities and symmetric deformation rates,

$$\boldsymbol{l}^{\mathrm{e}} = \dot{\boldsymbol{F}}^{\mathrm{e}}(\boldsymbol{F}^{\mathrm{e}})^{-1}\,, \qquad \boldsymbol{d}^{\mathrm{e}} = \tfrac{1}{2}[\boldsymbol{l}^{\mathrm{e}} + (\boldsymbol{l}^{\mathrm{e}})^{\mathrm{T}}]\,, \tag{3.67a}$$

$$\boldsymbol{l}^{\mathrm{p}} = \dot{\boldsymbol{F}}^{\mathrm{p}}(\boldsymbol{F}^{\mathrm{p}})^{-1}\,, \qquad \boldsymbol{d}^{\mathrm{p}} = \tfrac{1}{2}[\boldsymbol{l}^{\mathrm{p}} + (\boldsymbol{l}^{\mathrm{p}})^{\mathrm{T}}]\,, \tag{3.67b}$$

$$\tilde{\boldsymbol{l}}^{\mathrm{p}} = \boldsymbol{F}^{\mathrm{e}}\boldsymbol{l}^{\mathrm{p}}(\boldsymbol{F}^{\mathrm{e}})^{-1}\,, \qquad \tilde{\boldsymbol{d}}^{\mathrm{p}} = \tfrac{1}{2}[\tilde{\boldsymbol{l}}^{\mathrm{p}} + (\tilde{\boldsymbol{l}}^{\mathrm{p}})^{\mathrm{T}}]\,, \tag{3.67c}$$

where $\tilde{\boldsymbol{l}}^{\mathrm{p}}$ and $\tilde{\boldsymbol{d}}^{\mathrm{p}}$ should be understood as push-forwarded (to the current configuration $\mathcal{C}^t$) measures of $\boldsymbol{l}^{\mathrm{p}}$ and $\boldsymbol{d}^{\mathrm{p}}$ (defined in $\tilde{\mathcal{C}}^t$), we can write [93]

$$\boldsymbol{l} = \boldsymbol{l}^{\mathrm{e}} + \tilde{\boldsymbol{l}}^{\mathrm{p}}\,, \qquad \boldsymbol{d} = \boldsymbol{d}^{\mathrm{e}} + \tilde{\boldsymbol{d}}^{\mathrm{p}}\,. \tag{3.68}$$

This additive split of the deformation rates enables formulations that again reuse the equations of small-deformation elasto-plasticity presented in the previous section. However, the equations have to be now formulated and integrated in spatial description, i.e. in a deformed, moving configuration $\mathcal{C}^t$, which introduces numerous difficulties. Since the work-conjugated stress tensor to the deformation rate $\boldsymbol{d}$ is the Kirchhoff stress tensor, it is this tensor that is postulated to replace the linearized Cauchy stress $\boldsymbol{\sigma}$ in the constitutive equations. This implies necessity of using an objective time derivative in the elastic rate-type constitutive equation, e.g., c.f. Eq. $(3.23)_2$,

$$\mathcal{L}_{\boldsymbol{v}^{\mathrm{e}}}\boldsymbol{\tau} \equiv \dot{\boldsymbol{\tau}} - \boldsymbol{l}^{\mathrm{e}}\boldsymbol{\tau} - \boldsymbol{\tau}(\boldsymbol{l}^{\mathrm{e}})^{\mathrm{T}} = \mathfrak{c}^{\mathrm{e}} : \boldsymbol{d}^{\mathrm{e}}\,, \qquad \mathfrak{c}^{\mathrm{e}}_{ijkl} = F^{\mathrm{e}}_{ip}F^{\mathrm{e}}_{jq}F^{\mathrm{e}}_{kr}F^{\mathrm{e}}_{ls}\mathfrak{C}^{\mathrm{e}}_{pqrs}\,. \tag{3.69}$$

This equation does not conform to the small-deformation equation (3.43) and thus it has to be integrated in time in a different manner. A common way to treat this issue is to formulate this equation in a hypoelastic form

$$\overset{\triangledown}{\boldsymbol{\tau}} = \mathfrak{a} : \boldsymbol{d}^{\mathrm{e}} \tag{3.70}$$

where the Zaremba–Jaumann derivative of $\boldsymbol{\tau}$ is defined as

$$\overset{\triangledown}{\boldsymbol{\tau}} \equiv \dot{\boldsymbol{\tau}} - \boldsymbol{w}^{\mathrm{e}}\boldsymbol{\tau} + \boldsymbol{\tau}\boldsymbol{w}^{\mathrm{e}}\,, \qquad \boldsymbol{w}^{\mathrm{e}} = \tfrac{1}{2}(\boldsymbol{l}^{\mathrm{e}} - \boldsymbol{l}^{\mathrm{eT}}) = \boldsymbol{l}^{\mathrm{e}} - \boldsymbol{d}^{\mathrm{e}}\,, \tag{3.71}$$

and the hypoelastic stiffness tensor $\mathfrak{a}$ has in the index notation the form

$$\mathfrak{a}_{ijkl} = \mathfrak{c}^{\mathrm{e}}_{ijkl} + \delta_{ik}\tau_{jl} + \delta_{jl}\tau_{ik}\,. \tag{3.72}$$

Efficient integration of this formulation typically requires that the stress-dependent terms in $\mathfrak{a}$ are neglected, i.e. $\mathfrak{a} \approx \mathfrak{c}^{\mathrm{e}}$. This is usually not a critical problem as stresses are frequently incomparably smaller than stiffness coefficients. However, this makes the formulation inconsistent with the purely elastic formulation, i.e. the time-integrated response of elasto-plastic formulation in which no plastic flow happened to occur will be different than the response of purely elastic, time-independent analysis.

Other difficulties arise when the deformation rates $\boldsymbol{d}$, $\boldsymbol{d}^{\mathrm{e}}$, $\tilde{\boldsymbol{d}}^{\mathrm{p}}$ are to be integrated in time. Generally, there is no closed-form strain tensor whose time derivative would equal $\boldsymbol{d}$. Assuming invariance of principal strain directions, this condition appears to be fulfilled by the Hencky logarithmic strain $\boldsymbol{\epsilon} = \ln \boldsymbol{U} = \ln(\sqrt{\boldsymbol{C}})$. Even in this simplified case, however, the use of a numerically expensive tensor spectral decomposition routine is necessary in the computational applications which must be considered as drawback, especially in view of the fact that, as it will be shown in the following section, its unique differentiation will also be necessary in order to obtain the tangent operator.

The difficulties may be overcome by considering another approach to combining large-deformation elasto-plastic kinematics with the associative constitutive model. An example is a formulation of Simo [110], presented below for the case of the Huber–Mises flow condition, in which the multiplicative split of the deformation gradient (3.66) does not imply additive approximation (3.68) of the geometrically linear kinematic equation (3.42). Instead, large-deformation reformulation of the von Mises' postulate of minimum plastic dissipation leads to the following equations.

Let us define the elastic and plastic Cauchy–Green deformation tensors, cf. Eqs. (3.31),

$$\boldsymbol{b}^{\mathrm{e}} = \boldsymbol{F}^{\mathrm{e}}\boldsymbol{F}^{\mathrm{e}\mathrm{T}}\,, \qquad \boldsymbol{C}^{\mathrm{p}} = \boldsymbol{F}^{\mathrm{p}\mathrm{T}}\boldsymbol{F}^{\mathrm{p}}\,. \tag{3.73}$$

The plastic flow is considered isochoric, i.e.

$$J^{\mathrm{p}} \equiv 1, \qquad J = J^{\mathrm{e}}. \tag{3.74}$$

Thus, cf. Eqs. (3.34)–(3.35), let us define

$$\bar{\boldsymbol{F}}^{\mathrm{e}} = J^{-\frac{1}{3}}\boldsymbol{F}^{\mathrm{e}}\,, \qquad \det \bar{\boldsymbol{F}}^{\mathrm{e}} = 1\,, \qquad \bar{\boldsymbol{b}}^{\mathrm{e}} = \bar{\boldsymbol{F}}^{\mathrm{e}}\bar{\boldsymbol{F}}^{\mathrm{e}\mathrm{T}} = J^{-\frac{2}{3}}\boldsymbol{b}^{\mathrm{e}}\,. \tag{3.75}$$

Let the material obey the neo-Hookean hyperelastic equation, cf. (3.38),

$$\boldsymbol{\tau} = \tfrac{1}{2}K(J^2{-}1)\,\boldsymbol{I} + G\,\mathrm{dev}\bar{\boldsymbol{b}}^{\mathrm{e}}\,. \tag{3.76}$$

so that, cf. Eqs. (3.19)–(3.40), (3.69),

$$\mathcal{L}_{\boldsymbol{v}^{\mathrm{e}}}\boldsymbol{\tau} = \mathfrak{c}^{\mathrm{e}} : \boldsymbol{d}^{\mathrm{e}}\,, \tag{3.77}$$

$$\begin{aligned}\mathfrak{c}^{\mathrm{e}} &= K\left[J^2\,\boldsymbol{I}\otimes\boldsymbol{I} - (J^2{-}1)\,\boldsymbol{\mathfrak{I}}\right]\\ &\quad + \tfrac{2}{3}G\left[(\mathrm{tr}\bar{\boldsymbol{b}}^{\mathrm{e}})\big(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\big) - \big(\mathrm{dev}(\bar{\boldsymbol{b}}^{\mathrm{e}})\otimes\boldsymbol{I} + \boldsymbol{I}\otimes\mathrm{dev}(\bar{\boldsymbol{b}}^{\mathrm{e}})\big)\right].\end{aligned} \tag{3.78}$$

with the strain energy referred to the intermediate stress-free configuration $\tilde{\mathcal{C}}^t$.

The associative plastic flow rule is expressed in $\tilde{\mathcal{C}}^t$ as

$$\tilde{\boldsymbol{d}}^{\rm p} = \sqrt{\frac{3}{2}}\,\dot{\bar{e}}^{\rm p}\,\psi\,\boldsymbol{F}^{{\rm e}-1}\boldsymbol{n}\boldsymbol{F}^{{\rm e}-{\rm T}}\,, \qquad \boldsymbol{n} = \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|} = \sqrt{\frac{3}{2}}\,\frac{\boldsymbol{s}}{\bar{\sigma}}\,, \tag{3.79}$$

cf. the small-deformation equation (3.47) for comparison, where $\boldsymbol{n}$ is a normal direction to the flow surface in the current configuration $\mathcal{C}^t$ (the operation $\boldsymbol{F}^{{\rm e}-1}\boldsymbol{n}\boldsymbol{F}^{{\rm e}-{\rm T}}$ simply transforms it into $\tilde{\mathcal{C}}^t$), the deviatoric relative stress $\boldsymbol{s}$ and the scalar equivalent stress $\bar{\sigma}$ are defined as (cf. Eq. (3.46))

$$\boldsymbol{s} = {\rm dev}(\boldsymbol{\tau} - \bar{\boldsymbol{\alpha}}), \qquad \bar{\boldsymbol{\alpha}} = J^{-\frac{2}{3}}\boldsymbol{\alpha}, \qquad \bar{\sigma} = \sqrt{\tfrac{3}{2}\,\boldsymbol{s} : \boldsymbol{s}} \tag{3.80}$$

and the coefficient

$$\psi = \tfrac{1}{3}\left({\rm tr}\,\boldsymbol{b}^{\rm e} - \tfrac{1}{G}\,{\rm tr}\,\boldsymbol{\alpha}\right) \tag{3.81}$$

converges to unity for small deformations. Note that the back-stress $\boldsymbol{\alpha}$ is a Kirchhoff-type tensor defined in the current configuration.

It will be more convenient to transform the flow equation into either the initial ($\mathcal{C}^0$) or the current ($\mathcal{C}^t$) configuration. To do so, let us first write down certain relations between an arbitrary pair of spatial and material tensors ($\boldsymbol{\psi}$, $\boldsymbol{\varPsi}$), their traces and deviators and time derivatives:

$$\begin{aligned}
&\boldsymbol{\psi} = \boldsymbol{F}\boldsymbol{\varPsi}\boldsymbol{F}^{\rm T}, && \boldsymbol{\varPsi} = \boldsymbol{F}^{-1}\boldsymbol{\psi}\boldsymbol{F}^{-{\rm T}},\\
&{\rm tr}\,\boldsymbol{\psi} = (\boldsymbol{F}^{\rm T}\boldsymbol{F}) : \boldsymbol{\varPsi} = \boldsymbol{C} : \boldsymbol{\varPsi}, &&\\
&{\rm dev}\,\boldsymbol{\psi} = \boldsymbol{F}\,{\rm Dev}\,\boldsymbol{\varPsi}\,\boldsymbol{F}^{\rm T}, && {\rm Dev}\,\boldsymbol{\varPsi} \equiv \boldsymbol{\varPsi} - \tfrac{1}{3}\boldsymbol{C}^{-1}(\boldsymbol{C} : \boldsymbol{\varPsi}),\\
&\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\psi} = \boldsymbol{F}\,\left[\tfrac{{\rm d}}{{\rm d}t}\left(\boldsymbol{F}^{-1}\boldsymbol{\psi}\boldsymbol{F}^{-{\rm T}}\right)\right]\boldsymbol{F}^{\rm T} = \boldsymbol{F}\dot{\boldsymbol{\varPsi}}\boldsymbol{F}^{\rm T}. &&
\end{aligned} \tag{3.82}$$

According to these rules, the Kirchhoff (spatial) stress tensors $\boldsymbol{\tau}$, $\boldsymbol{\alpha}$, $\bar{\boldsymbol{\alpha}}$ correspond to their 2nd Piola–Kirchhoff (material) counterparts $\boldsymbol{T}$, $\boldsymbol{A}$, $\bar{\boldsymbol{A}}$, respectively. Besides, transforming Eqs. (3.66) and (3.73), we can see that the spatial tensor $\boldsymbol{b}^{\rm e}$ corresponds in the same way to the material tensor $\boldsymbol{C}^{{\rm p}-1}$, e.g.

$$\boldsymbol{b}^{\rm e} = \boldsymbol{F}\boldsymbol{C}^{{\rm p}-1}\boldsymbol{F}^{\rm T}, \qquad \mathcal{L}_{\boldsymbol{v}}\boldsymbol{b}^{\rm e} = \boldsymbol{F}\,\frac{{\rm d}\boldsymbol{C}^{{\rm p}-1}}{{\rm d}t}\,\boldsymbol{F}^{\rm T}. \tag{3.83}$$

Realizing now that

$$\frac{{\rm d}\boldsymbol{F}^{{\rm p}-1}}{{\rm d}t} = -\boldsymbol{F}^{{\rm p}-1}\dot{\boldsymbol{F}}^{\rm p}\boldsymbol{F}^{{\rm p}-1} \quad\Rightarrow\quad \frac{{\rm d}\boldsymbol{C}^{{\rm p}-1}}{{\rm d}t} = -2\boldsymbol{F}^{{\rm p}-1}\tilde{\boldsymbol{d}}^{\rm p}\boldsymbol{F}^{{\rm p}-{\rm T}},$$

we can rewrite Eq. (3.79)–(3.80) in the spatial description as [113]

$$\mathcal{L}_{\boldsymbol{v}}\boldsymbol{b}^{\rm e} = -\frac{3\psi\dot{\bar{e}}^{\rm p}}{\bar{\sigma}}\,\boldsymbol{s}\,, \qquad \boldsymbol{s} = {\rm dev}(\boldsymbol{\tau} - \bar{\boldsymbol{\alpha}})\,, \tag{3.84}$$

and in the material description,

$$\frac{\mathrm{d}\boldsymbol{C}^{\mathrm{p}-1}}{\mathrm{d}t} = -\frac{3\psi\dot{\bar{e}}^{\mathrm{p}}}{\bar{\sigma}}\,\boldsymbol{S}\,, \qquad \boldsymbol{S} = \mathrm{Dev}(\boldsymbol{T}-\bar{\boldsymbol{A}}) = \boldsymbol{F}^{-1}\boldsymbol{s}\boldsymbol{F}^{-\mathrm{T}}, \tag{3.85}$$

where ψ, cf. Eq. (3.81), can be equivalently expressed in terms of $\mathcal{C}^0$-defined (material) tensors as $\psi = \frac{1}{3}\boldsymbol{C} : (\boldsymbol{C}^{\mathrm{p}-1} - \frac{1}{G}\boldsymbol{A})$.

Since $\boldsymbol{s}$ is deviatoric, so must be $\mathcal{L}_{\boldsymbol{v}}\boldsymbol{b}^{\mathrm{e}}$. Equations (3.84)–(3.85) can thus be rewritten as

$$\mathrm{dev}(\mathcal{L}_{\boldsymbol{v}}\boldsymbol{b}^{\mathrm{e}}) = -\frac{3\psi\dot{\bar{e}}^{\mathrm{p}}}{\bar{\sigma}}\,\boldsymbol{s}\,, \qquad \mathrm{tr}(\mathcal{L}_{\boldsymbol{v}}\boldsymbol{b}^{\mathrm{e}}) = 0\,, \tag{3.86}$$

$$\mathrm{Dev}\left(\frac{\mathrm{d}\boldsymbol{C}^{\mathrm{p}-1}}{\mathrm{d}t}\right) = -\frac{3\psi\dot{\bar{e}}^{\mathrm{p}}}{\bar{\sigma}}\,\boldsymbol{S}\,, \qquad \boldsymbol{C} : \frac{\mathrm{d}\boldsymbol{C}^{\mathrm{p}-1}}{\mathrm{d}t} = 0\,. \tag{3.87}$$

The evolution of the back stress $\boldsymbol{\alpha}$, similarly as in the small-deformation formulation, is associated with the direction of the plastic flow,

$$\mathrm{dev}(\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\alpha}) = H'\,\frac{\psi\dot{\bar{e}}^{\mathrm{p}}}{\bar{\sigma}}\,\boldsymbol{s} = -\tfrac{1}{3}H'\,\mathrm{dev}(\mathcal{L}_{\boldsymbol{v}}\boldsymbol{b}^{\mathrm{e}}), \qquad \mathrm{tr}(\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\alpha}) = 0, \tag{3.88}$$

$$\mathrm{Dev}(\dot{\boldsymbol{A}}) = H'\,\frac{\psi\dot{\bar{e}}^{\mathrm{p}}}{\bar{\sigma}}\,\boldsymbol{S} = -\tfrac{1}{3}H'\,\mathrm{Dev}\left(\frac{\mathrm{d}\boldsymbol{C}^{\mathrm{p}-1}}{\mathrm{d}t}\right), \qquad \boldsymbol{C} : \dot{\boldsymbol{A}} = 0, \tag{3.89}$$

where $H = H(\bar{e}^{\mathrm{p}})$ is a material function.

The above system of large-deformation rate-type constitutive equations (3.73)–(3.76) and (3.86)–(3.89) has to be completed with the evolution equation for the equivalent plastic strain $\bar{e}^{\mathrm{p}}$. The latter is assumed in exactly the same form as in the small-deformation formulation, i.e. Eq. (3.50) for the rate-independent model and Eq. (3.57) for the viscoplastic models. In the first case, the formulation may be again transformed to a rate-type stress strain relation, cf. Eq. (3.18) with $\boldsymbol{\eta} = \boldsymbol{0}$,

$$\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\tau} = \boldsymbol{\mathfrak{c}}^{\mathrm{tg}} : \boldsymbol{d} \tag{3.90}$$

where the elasto-plastic tangent stiffness tensor $\boldsymbol{\mathfrak{c}}^{\mathrm{tg}}$ has been derived in [110]. In the latter case, we can transform it to the form (3.18) as

$$\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\tau} = \boldsymbol{\mathfrak{c}}^{\mathrm{tg}} : \boldsymbol{d} + \boldsymbol{\eta}\,, \qquad \boldsymbol{\mathfrak{c}}^{\mathrm{tg}} = \boldsymbol{\mathfrak{c}}^{\mathrm{e}}. \tag{3.91}$$

Obviously, if the yield condition is not fulfilled (i.e. if $f < 0$, in material models to which it concerns), then the evolution equation reduces in all elasto-plastic models to $\dot{\bar{e}}^{\mathrm{p}} = 0$ and the constitutive model of neo-Hookean elasticity (3.39) is obtained, $\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\tau} = \boldsymbol{\mathfrak{c}}^{\mathrm{e}} : \boldsymbol{d}$.

The constitutive model presented above is a particular form of the general constitutive formulation (3.15), expressed in spatial stress and strain rate measures, with the set of internal state parameters

$$\mathbf{z} = \{\boldsymbol{b}^{\mathrm{e}}, \boldsymbol{\alpha}, \bar{e}^{\mathrm{p}}\} \qquad \text{or} \qquad \mathbf{z} = \{\bar{\boldsymbol{b}}^{\mathrm{e}}, \bar{\boldsymbol{\alpha}}, \bar{e}^{\mathrm{p}}\}\,. \tag{3.92}$$

It is clear that, writing the equations in material description, in the configuration $\mathcal{C}^r$, we may equivalently consider $\boldsymbol{C}^{\mathrm{p}}$ and $\boldsymbol{A}$ as internal state parameters, instead of $\boldsymbol{b}^{\mathrm{e}}$ and $\boldsymbol{\alpha}$. If $H \equiv 0$ (no kinematic hardening) we can set $\boldsymbol{\alpha} \equiv \mathbf{0}$ and remove it from $\mathbf{z}$.

3.3. Time integration of constitutive equations of elasto-plasticity

The subject of this section are the ways of solution of the following problem at an arbitrary time instant $t = t_c$

Given:

- history of deformation $\boldsymbol{F}(t)$ within $0 \le t \le t_c$,
- rate-type constitutive equations,

compute:

- stress $\tilde{\boldsymbol{\sigma}}$ and constitutive state parameters $\boldsymbol{z}_i$ at $t = t_c$,

for constitutive equations given in the general form (3.15), and for their particular examples discussed in Sections 3.2.2–3.2.4. The generic stress tensor $\tilde{\boldsymbol{\sigma}}$ may obviously be understood as $\boldsymbol{T}$, $\boldsymbol{\tau}$, or $\boldsymbol{\sigma}$ (they are in fact mutually related *via* known $\boldsymbol{F}$), in accordance with the geometric description and particular constitutive formulation used.

In the case of elasticity the issue is trivial — Eq. (3.13) or (3.14) allow to determine instantaneous stress as a function of instantaneous strain (with no historical data necessary) for both small- and large-deformation formulations and for any form of the strain energy function W.

In the case of elasto-plastic models, the rate-type constitutive formulations of Sections 3.2.3 and 3.2.4 need to be numerically time-integrated. Let us discretize the time axis $t \ge 0$ by introducing a monotonic series of time instants $t_0 = 0\,,\ t_1\,,\ t_2\,,\ \ldots$. The above formulated problem appears now to have the form

Given:

- deformation $\boldsymbol{F}$ and constitutive state parameters $\boldsymbol{z}_i$ at $t = t_n$,
- deformation $\boldsymbol{F}$ at $t = t_{n+1}$,
- rate-type constitutive equations, discretized in time in the interval $[t_n \, , \, t_{n+1}]$,

compute:

- stress $\tilde{\boldsymbol{\sigma}}$ and constitutive state parameters $\boldsymbol{z}_i$ at $t = t_{n+1}$,

(again with $\tilde{\boldsymbol{\sigma}}$ understood as $\boldsymbol{T}$, $\boldsymbol{\tau}$ or $\boldsymbol{\sigma}$, respectively). We can write it down shortly as

$$\tilde{\boldsymbol{\sigma}}_{n+1} = \tilde{\boldsymbol{\sigma}}_{n+1}(\boldsymbol{F}_{n+1}, \boldsymbol{F}_n, \mathbf{z}_n) \, , \qquad \boldsymbol{z}_{i\,n+1} = \boldsymbol{z}_{i\,n+1}(\boldsymbol{F}_{n+1}, \boldsymbol{F}_n, \mathbf{z}_n) \, , \tag{3.93}$$

which is an incremental counterpart of the general rate type constitutive equations (3.15).

A common approach to time-discretization of elasto-plastic formulations is frequently referred to as the *elastic predictor/plastic corrector* scheme. In it, the stress increment at t_{n+1} is first computed as if deformation were purely elastic (*trial elastic stress*) and, if it appears to violate the condition (3.44), it is corrected by mapping its trial value back onto the surface in the stress space that represents the updated consistency condition at t_{n+1}. For several simple cases (e.g. small-deformation isotropic Huber–Mises flow formulation) the mapping has a form of a multi-dimensional radial-return scheme, cf. e.g. [74, 115, 131], while otherwise we have to do with a general return-mapping problem, typically solved in iterations with the use of, e.g., the closest point projection scheme, cf. [112, 116, 130] and others.

In the remaining part of this section, the time-discretized (incremental) formulations of the rate-type constitutive models discussed in Sections 3.2.3 and 3.2.4 will be presented for a typical time interval $[t_n \, , \, t_{n+1}]$. For any quantity $(\cdot)$, the notations $(\cdot)_n$ and $(\cdot)_{n+1}$ are used to denote its values assumed at $t = t_n$ and $t = t_{n+1}$, respectively, while

$$\Delta(\cdot) \equiv (\cdot)_{n+1} - (\cdot)_n$$

(in particular, $\Delta t = t_{n+1} - t_n$). Time derivatives, assumed constant throughout the interval, are approximated as

$$\frac{\mathrm{d}(\cdot)}{\mathrm{d}t} \approx \frac{\Delta(\cdot)}{\Delta t} \, .$$

For reasons that will be explained in Section 3.4, an important quantity characterizing each of the presented algorithms is a 4-th rank tensor describing relation between infinitesimal change in the input variable $\boldsymbol{F}_{n+1}$ and the resulting change in stress $\tilde{\boldsymbol{\sigma}}_{n+1}$. The tensor, called *algorithmic constitutive tangent stiffness tensor* (consistent with the time integration scheme) is defined in different ways for different schemes, i.e.

- for small deformations

$$\mathrm{d}\boldsymbol{\sigma}_{n+1} = \boldsymbol{\mathcal{C}} : \mathrm{d}\boldsymbol{\varepsilon}_{n+1} \qquad \text{or} \qquad \mathrm{d}\boldsymbol{\sigma}_{n+1} = \boldsymbol{\mathcal{C}} : \mathrm{d}(\Delta\boldsymbol{\varepsilon}), \tag{3.94}$$

- for large deformation, material description

$$\mathrm{d}\boldsymbol{T}_{n+1} = \boldsymbol{\mathfrak{C}} : \mathrm{d}\boldsymbol{E}_{n+1} \tag{3.95}$$

- for large deformation, spatial description

$$\mathrm{d}^{\circ}\boldsymbol{\tau}_{n+1} = \boldsymbol{\mathfrak{c}} : \mathrm{d}\boldsymbol{\epsilon}_{n+1} \tag{3.96}$$

  where $\mathrm{d}^{\circ}\boldsymbol{\tau}$ is a differential counterpart of the Lie derivative $\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\tau}$, cf. Eq. (3.19),

$$\begin{aligned} \mathrm{d}^{\circ}\boldsymbol{\tau}_{n+1} &= \boldsymbol{F}_{n+1}\,\mathrm{d}\boldsymbol{T}_{n+1}\,\boldsymbol{F}_{n+1}^{\mathrm{T}} \\ &= \mathrm{d}\boldsymbol{\tau}_{n+1} - \mathrm{d}\boldsymbol{F}_{n+1}\,\boldsymbol{F}_{n+1}^{-1}\,\boldsymbol{\tau}_{n+1} - \boldsymbol{\tau}_{n+1}\left(\mathrm{d}\boldsymbol{F}_{n+1}\,\boldsymbol{F}_{n+1}^{-1}\right)^{\mathrm{T}} . \end{aligned} \tag{3.97}$$

The strain differentials $\mathrm{d}\boldsymbol{\varepsilon}_{n+1}$, $\mathrm{d}\boldsymbol{E}_{n+1}$ and $\mathrm{d}\boldsymbol{\epsilon}_{n+1}$ are defined in analogy to the variations[1] (3.10), i.e.

$$\mathrm{d}\boldsymbol{\varepsilon}_{n+1} = \tfrac{1}{2}\left(\mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{u}}_{n+1} + (\mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{u}}_{n+1})^{\mathrm{T}}\right), \qquad \mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{u}}_{n+1} = \boldsymbol{\nabla}\mathrm{d}\boldsymbol{u}_{n+1}\,, \tag{3.98a}$$

$$\mathrm{d}\boldsymbol{E}_{n+1} = \tfrac{1}{2}\left(\mathrm{d}\boldsymbol{E}^{\mathrm{u}}_{n+1} + (\mathrm{d}\boldsymbol{E}^{\mathrm{u}}_{n+1})^{\mathrm{T}}\right), \qquad \mathrm{d}\boldsymbol{E}^{\mathrm{u}}_{n+1} = \boldsymbol{F}^{\mathrm{T}}_{n+1}\,\mathrm{d}\boldsymbol{F}_{n+1}\,, \tag{3.98b}$$

$$\mathrm{d}\boldsymbol{\epsilon}_{n+1} = \tfrac{1}{2}\left(\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{u}}_{n+1} + (\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{u}}_{n+1})^{\mathrm{T}}\right), \qquad \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{u}}_{n+1} = \mathrm{d}\boldsymbol{F}_{n+1}\,\boldsymbol{F}^{-1}_{n+1}\,, \tag{3.98c}$$

and, in view of symmetry of the algorithmic tangent stiffness tensors, they can be replaced with $\mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{u}}_{n+1}$, $\mathrm{d}\boldsymbol{E}^{\mathrm{u}}_{n+1}$ and $\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{u}}_{n+1}$, respectively, in Eqs. (3.94)–(3.96). Note that $\boldsymbol{\mathcal{C}}$, $\boldsymbol{\mathfrak{C}}$ and $\boldsymbol{\mathfrak{c}}$ are usually *not* equal to their continuum counterparts $\boldsymbol{\mathcal{C}}^{\mathrm{tg}}$, $\boldsymbol{\mathfrak{C}}^{\mathrm{tg}}$, $\boldsymbol{\mathfrak{c}}^{\mathrm{tg}}$, respectively, appearing in the rate-type constitutive equations in Sections 3.2.3 and 3.2.4. Forms of the algorithmic tangent stiffness tensors will be given in the following part of this section for different time-integrated constitutive formulations.

[1] It is stressed once again that the differentials $\mathrm{d}\boldsymbol{\epsilon}$ and $\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{u}}$ do not correspond to any closed form tensors $\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon}^{\mathrm{u}}$, but such closed forms are not necessary in the presented formulation.

3.3.1. Small deformation formulation

Approximate time integration of constitutive equations presented in Sec. 3.2.3 leads to the following incremental formulation. The total strain increment is

$$\Delta\boldsymbol{\varepsilon} = \tfrac{1}{2}\left[\boldsymbol{\nabla}(\Delta\boldsymbol{u}) + (\boldsymbol{\nabla}(\Delta\boldsymbol{u}))^{\mathrm{T}}\right] = \Delta\boldsymbol{\varepsilon}^{\mathrm{e}} + \Delta\boldsymbol{\varepsilon}^{\mathrm{p}}\,, \tag{3.99}$$

while stress is approximated with the following incremental equations, cf. Eq. (3.43),

$$\begin{aligned} \Delta\boldsymbol{\sigma} &= \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \Delta\boldsymbol{\varepsilon}^{\mathrm{e}} = \boldsymbol{\mathcal{C}}^{\mathrm{e}} : (\Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}^{\mathrm{p}})\,, & (3.100)\\ \boldsymbol{\sigma}_{n+1} &= \boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma} = \boldsymbol{\sigma}_{\mathrm{trial}} - \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \Delta\boldsymbol{\varepsilon}^{\mathrm{p}}_{n+1}\,, & (3.101) \end{aligned}$$

where $\boldsymbol{\sigma}_{\mathrm{trial}} = \boldsymbol{\sigma}_n + \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \Delta\boldsymbol{\varepsilon}$ denotes the elastic trial stress. Convexity of the flow surface implies that

$$f_{n+1} \equiv f(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\alpha}_{n+1}, \bar{e}^{\mathrm{p}}_{n+1}) < 0 \quad \Longleftrightarrow \quad f_{\mathrm{trial}} \equiv f(\boldsymbol{\sigma}_{\mathrm{trial}}, \boldsymbol{\alpha}_n, \bar{e}^{\mathrm{p}}_n) < 0 \tag{3.102}$$

which enables checking the plastic yield condition after only computing the trial stress. If $f_{\mathrm{trial}} < 0$ then the deformation within the time increment is purely elastic and the solution is

$$\Delta\boldsymbol{\varepsilon}^{\mathrm{p}} = \boldsymbol{0}\,, \qquad \boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{\mathrm{trial}}\,, \qquad \boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n\,, \qquad \bar{e}^{\mathrm{p}}_{n+1} = \bar{e}^{\mathrm{p}}_n\,. \tag{3.103}$$

Otherwise, cf. Eq. (3.47),

$$\Delta\boldsymbol{\varepsilon}^{\mathrm{p}} = \sqrt{\frac{3}{2}}\,\Delta\bar{e}^{\mathrm{p}}\,\boldsymbol{n}\,, \qquad \boldsymbol{n} = \frac{\boldsymbol{s}_{n+\alpha}}{\|\boldsymbol{s}_{n+\alpha}\|} = \sqrt{\frac{3}{2}}\frac{\boldsymbol{s}_{n+\alpha}}{\bar{\sigma}_{n+\alpha}}\,, \tag{3.104}$$

where the index $n+\alpha$ $(0 \le \alpha \le 1)$ indicates an intermediate value taken at $t = (1-\alpha)t_n + \alpha t_{n+1}$[1]. Defining

$$\boldsymbol{s}_{\mathrm{trial}} = \operatorname{dev}(\boldsymbol{\sigma}_{\mathrm{trial}} - \boldsymbol{\alpha}_n) \tag{3.105}$$

and noting that, cf. Eq. (3.49),

$$\operatorname{dev}\Delta\boldsymbol{\alpha} = \tfrac{2}{3}H'_{n+\alpha}\Delta\boldsymbol{\varepsilon}^{\mathrm{p}} = \sqrt{\tfrac{2}{3}}H'_{n+\alpha}\Delta\bar{e}^{\mathrm{p}}\,\boldsymbol{n}\,, \tag{3.106}$$

we can write

$$\begin{aligned} \boldsymbol{s}_{n+1} &= \boldsymbol{s}_{\mathrm{trial}} - \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \Delta\boldsymbol{\varepsilon}^{\mathrm{p}} - \operatorname{dev}\Delta\boldsymbol{\alpha} \\ &= \boldsymbol{s}_{\mathrm{trial}} - \sqrt{\tfrac{2}{3}}\,\Delta\bar{e}^{\mathrm{p}}\left(\tfrac{3}{2}\boldsymbol{\mathcal{C}}^{\mathrm{e}} + H'_{n+\alpha}\,\boldsymbol{\mathfrak{I}}\right) : \boldsymbol{n} \end{aligned} \tag{3.107}$$

[1] In fact, only such values of α should be considered for which the generalized intermediate trial stress $\boldsymbol{\sigma}_{\mathrm{trial}\ n+\alpha} = \boldsymbol{\sigma}_n + \boldsymbol{\mathcal{C}}^{\mathrm{e}} : \alpha\Delta\boldsymbol{\varepsilon}$ violates the inequality $(3.102)_2$

which for isotropic elasticity (3.26) reduces to

$$\boldsymbol{s}_{n+1} = \boldsymbol{s}_{\text{trial}} - \sqrt{\tfrac{2}{3}}\,\Delta\bar{e}^{\text{p}}\,(3G + H'_{n+\alpha})\,\boldsymbol{n} \tag{3.108}$$

with $\boldsymbol{s}_{\text{trial}} = \operatorname{dev}(\boldsymbol{\sigma}_n - 2G\Delta\boldsymbol{\varepsilon} - \boldsymbol{\alpha}_n)$.

Consider first the fully implicit scheme with $\alpha = 1$ in Eqs. $(3.104)_2$, (3.106). Then, $\boldsymbol{s}_{n+1}$ is parallel to $\boldsymbol{n}$ and thus $\boldsymbol{s}_{\text{trial}}$ must be so, too, to fulfill Eq. (3.108). Employing

$$\boldsymbol{n} = \frac{\boldsymbol{s}_{n+1}}{\|\boldsymbol{s}_{n+1}\|} = \frac{\boldsymbol{s}_{\text{trial}}}{\|\boldsymbol{s}_{\text{trial}}\|}$$

we can consequently transform Eq. (3.108) to a scalar equation expressed in terms of tensorial norms,

$$\|\boldsymbol{s}_{n+1}\| = \|\boldsymbol{s}_{\text{trial}}\| - \sqrt{\tfrac{2}{3}}\,\Delta\bar{e}^{\text{p}}\,(3G + H'_{n+1}) \tag{3.109}$$

or

$$\bar{\sigma}_{n+1} = \bar{\sigma}_{\text{trial}} - \Delta\bar{e}^{\text{p}}\,(3G + H'_{n+1})\,. \tag{3.110}$$

This scalar equation with two unknowns $\bar{\sigma}_{n+1}$ and $\Delta\bar{e}^{\text{p}}$ must be completed with a time-integrated form of the consistency condition (3.50) or (3.57). It can be written in a general form

$$\tilde{f}(\bar{\sigma}_{n+1}, \bar{e}^{\text{p}}_{n+1}, \Delta\bar{e}^{\text{p}}) = 0 \tag{3.111}$$

where, for rate-independent elasto-plasticity

$$\tilde{f} = f_{n+1} = \bar{\sigma}_{n+1} - \kappa_{n+1}\,, \qquad \kappa_{n+1} \equiv \kappa(\bar{e}^{\text{p}}_{n+1}) \tag{3.112}$$

while for rate-dependent models

$$\tilde{f} = \Delta t\, g(\bar{\sigma}_{n+1}, \bar{e}^{\text{p}}_{n+1}) - \Delta\bar{e}^{\text{p}}\,. \tag{3.113}$$

Substitution of Eq. (3.110) into Eq. (3.111) leads to a scalar nonlinear equation that can be solved with respect to $\Delta\bar{e}^{\text{p}}$ with the use of the Newton iterative scheme

$$\begin{tikzcd} \delta(\Delta\bar{e}^{\text{p}}) = -\dfrac{\tilde{f}^{(i)}}{\tilde{f}^{(i)}_{,\Delta\bar{e}^{\text{p}}}} \arrow[rr] & & \Delta\bar{e}^{\text{p}(i+1)} = \Delta\bar{e}^{\text{p}(i)} + \delta(\Delta\bar{e}^{\text{p}}) \arrow[dl] \\ & i := i+1 \arrow[ul] & \end{tikzcd} \tag{3.114}$$

where i is the iteration counter, $\tilde{f}^{(i)} \equiv \tilde{f}(\Delta\bar{e}^{\mathrm{p}(i)})$, and the above loop is repeated until the corrector $\delta(\Delta\bar{e}^{\mathrm{p}})$ gets sufficiently small. The initial predictor is set as, e.g., $\Delta\bar{e}^{\mathrm{p}(0)} = 0$. The coefficient

$$\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}} = \frac{\partial\tilde{f}}{\partial\bar{\sigma}_{n+1}}\,\bar{\sigma}_{n+1,\Delta\bar{e}^{\mathrm{p}}} + \frac{\partial\tilde{f}}{\partial\bar{e}^{\mathrm{p}}_{n+1}} + \frac{\partial\tilde{f}}{\partial\Delta\bar{e}^{\mathrm{p}}}\,, \tag{3.115}$$

with, cf. Eq. (3.110),

$$\bar{\sigma}_{n+1,\Delta\bar{e}^{\mathrm{p}}} \equiv \frac{\mathrm{d}\bar{\sigma}_{n+1}}{\mathrm{d}\Delta\bar{e}^{\mathrm{p}}} = -(3G + H'_{n+1} + H''_{n+1}\Delta\bar{e}^{\mathrm{p}}), \tag{3.116}$$

will be useful in further derivations, too. In particular, for rate-independent case, it can be expressed as

$$\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}} = \bar{\sigma}_{n+1,\Delta\bar{e}^{\mathrm{p}}} - \kappa'_{n+1} = -\left(3G + \kappa'_{n+1} + H'_{n+1} + H''_{n+1}\Delta\bar{e}^{\mathrm{p}}\right) \tag{3.117}$$

while for the rate-dependent case as

$$\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}} = \Delta t\left[(g_{,\bar{\sigma}})_{n+1}\,\bar{\sigma}_{n+1,\Delta\bar{e}^{\mathrm{p}}} + (g_{,\bar{e}^{\mathrm{p}}})_{n+1}\right] - 1. \tag{3.118}$$

where, for overstress viscoplasticity, cf. Eq. (3.59),

$$\frac{\partial g}{\partial\bar{\sigma}} \equiv g_{,\bar{\sigma}} = \frac{gm}{\bar{\sigma} - \kappa}\,, \qquad \frac{\partial g}{\partial\bar{e}^{\mathrm{p}}} \equiv g_{,\bar{e}^{\mathrm{p}}} = -\frac{gm\bar{\sigma}\kappa'}{(\bar{\sigma} - \kappa)\kappa}\,, \tag{3.119}$$

and for the power-law strain and strain-rate hardening viscoplasticity, cf. Eq. (3.61),

$$\frac{\partial g}{\partial\bar{\sigma}} \equiv g_{,\bar{\sigma}} = \frac{g}{m\bar{\sigma}}\,, \qquad \frac{\partial g}{\partial\bar{e}^{\mathrm{p}}} \equiv g_{,\bar{e}^{\mathrm{p}}} = -\frac{g}{mn(\bar{e}^{\mathrm{p}} + \varepsilon_0)}\,. \tag{3.120}$$

For rate-independent elasto-plasticity with linear hardening ($\kappa' = \mathrm{const}$ and $H' = \mathrm{const}$) Eq. (3.112) becomes linear and has the solution

$$\Delta\bar{e}^{\mathrm{p}} = \frac{\bar{\sigma}_{\mathrm{trial}} - \kappa_n}{3G + \kappa' + H'}\,. \tag{3.121}$$

For $\alpha < 1$, Eq. (3.108) cannot be transformed into a scalar form, since $\boldsymbol{n}$ is not proportional to $\boldsymbol{s}_{n+1}$. However, expressing $\boldsymbol{n}$ in an explicit way as

$$\boldsymbol{n} = \frac{\boldsymbol{s}_{\mathrm{trial}\,n+\alpha}}{\|\boldsymbol{s}_{\mathrm{trial}\,n+\alpha}\|}\,, \qquad \boldsymbol{s}_{\mathrm{trial}\,n+\alpha} = \mathrm{dev}\,(\boldsymbol{\sigma}_n + 2G\,\alpha\Delta\boldsymbol{\varepsilon} - \boldsymbol{\alpha}_n)\,,$$

we can take the norm of the $\Delta\bar{e}^{\mathrm{p}}$-dependent r.h.s. of Eq. (3.108) and substitute it into Eq. (3.111), again obtaining a nonlinear scalar equation on $\Delta\bar{e}^{\mathrm{p}}$. It appears, however, that the existence of the solution of this equation is only guaranteed

Table 3.1. Solution algorithm for small-deformation elasto-plasticity (for the sake of legibility, subscripts $(n+1)$ have been omitted)

```
Given
    strain $\boldsymbol{\varepsilon}_n$ and state parameters[a] $\mathbf{z}_n = \{\, \operatorname{dev} \boldsymbol{\sigma}_n \,,\, \operatorname{dev} \boldsymbol{\alpha}_n \,,\, \bar{e}^{\mathrm{p}}_n \,\}$ at $t_n$ ,
    strain $\boldsymbol{\varepsilon}$ at $t_{n+1}$   $(\Delta\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_n)$
compute
    state parameters $\mathbf{z} = \{\, \operatorname{dev} \boldsymbol{\sigma} \,,\, \operatorname{dev} \boldsymbol{\alpha} \,,\, \bar{e}^{\mathrm{p}} \,\}$ and stress $\boldsymbol{\sigma}$ at $t_{n+1}$
using the following scheme:
1. $\operatorname{dev} \boldsymbol{\sigma}_{\mathrm{trial}} = \operatorname{dev} \boldsymbol{\sigma}_n + 2G \operatorname{dev} \Delta\boldsymbol{\varepsilon}$        – trial deviatoric stress
2. $\boldsymbol{s}_{\mathrm{trial}} = \operatorname{dev} \boldsymbol{\sigma}_{\mathrm{trial}} - \operatorname{dev} \boldsymbol{\alpha}_n$        – trial relative stress
3. $\bar{\sigma}_{\mathrm{trial}} = \sqrt{\tfrac{3}{2}\, \boldsymbol{s}_{\mathrm{trial}} : \boldsymbol{s}_{\mathrm{trial}}}$        – trial equivalent stress
if $\bar{\sigma}_{\mathrm{trial}} - \kappa(\bar{e}^{\mathrm{p}}_n) > 0$ (plastic flow) then[b]
    4. $\Delta\bar{e}^{\mathrm{p}} \longleftarrow$ from Table 3.2        – equiv. plastic strain increment
    5. $\boldsymbol{n} = \boldsymbol{s}_{\mathrm{trial}} / \|\boldsymbol{s}_{\mathrm{trial}}\|$        – flow direction
    6. $\Delta\boldsymbol{\varepsilon}^{\mathrm{p}} = \sqrt{\tfrac{3}{2}} \Delta\bar{e}^{\mathrm{p}}\, \boldsymbol{n}$        – plastic strain increment
    7. $\left.\begin{aligned} \operatorname{dev} \boldsymbol{\sigma} &= \operatorname{dev} \boldsymbol{\sigma}_{\mathrm{trial}} - 2G\, \Delta\boldsymbol{\varepsilon}^{\mathrm{p}} \\ \operatorname{dev} \boldsymbol{\alpha} &= \operatorname{dev} \boldsymbol{\alpha}_n + \tfrac{2}{3} H' \Delta\boldsymbol{\varepsilon}^{\mathrm{p}} \\ \bar{e}^{\mathrm{p}} &= \bar{e}^{\mathrm{p}}_n + \Delta\bar{e}^{\mathrm{p}} \end{aligned}\right\}$ update state parameters
else (purely elastic deformation)
    8. $\operatorname{dev} \boldsymbol{\sigma} = \operatorname{dev} \boldsymbol{\sigma}_{\mathrm{trial}}$ ,   $\operatorname{dev} \boldsymbol{\alpha} = \operatorname{dev} \boldsymbol{\alpha}_n$ ,   $\bar{e}^{\mathrm{p}} = \bar{e}^{\mathrm{p}}_n$
end if
9. $\boldsymbol{\sigma} = \operatorname{dev} \boldsymbol{\sigma} + K(\operatorname{tr}\boldsymbol{\varepsilon})\, \boldsymbol{I}$        – final stress
```

[a] if $H \equiv \text{const}$ (no kinematic hardening) then $\operatorname{dev} \boldsymbol{\alpha} \equiv \boldsymbol{0}$ is not a state parameter and all relevant computations are skipped

[b] for the power-law strain/strain-rate hardening viscoplasticity this condition is always fulfilled, except for $\bar{\sigma}_{\mathrm{trial}} = 0$

Table 3.2. Solution algorithm for the equivalent plastic strain increment $\Delta\bar{e}^{\mathrm{p}}$, cf. Eq. (3.114), small deformation formulation

```
1. $\Delta\bar{e}^{\mathrm{p}} = 0$
2. $\bar{e}^{\mathrm{p}} = \bar{e}^{\mathrm{p}}_n + \Delta\bar{e}^{\mathrm{p}}$
3. $\bar{\sigma} = \bar{\sigma}_{\mathrm{trial}} - [3G + H'(\bar{e}^{\mathrm{p}})]\,\Delta\bar{e}^{\mathrm{p}}$,        $\bar{\sigma}_{,\Delta\bar{e}^{\mathrm{p}}} = -\,[3G + H'(\bar{e}^{\mathrm{p}}) + H''(\bar{e}^{\mathrm{p}})\,\Delta\bar{e}^{\mathrm{p}}]$
4. $\tilde{f},\ \tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}} \quad \longleftarrow$    from Table 3.3
5. $\delta(\Delta\bar{e}^{\mathrm{p}}) = -\tilde{f}^{-1}_{,\Delta\bar{e}^{\mathrm{p}}}\,\tilde{f}$,       $\Delta\bar{e}^{\mathrm{p}} = \Delta\bar{e}^{\mathrm{p}} + \delta(\Delta\bar{e}^{\mathrm{p}})$
6. if  $\|\delta(\Delta\bar{e}^{\mathrm{p}})\| > \mathrm{tol}$  then go to step 2;  else exit
```

Table 3.3. Computation of $\tilde{f}$ and $\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}$ for different elasto-plastic and elasto-viscoplastic models

```
if  (rate-independent plasticity), cf. Eqs. (3.112), (3.117)
    1a. $\tilde{f} = \bar{\sigma} - \kappa(\bar{e}^{\mathrm{p}})$
    2a. $\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}} = \bar{\sigma}_{,\Delta\bar{e}^{\mathrm{p}}} - \kappa'(\bar{e}^{\mathrm{p}})$
else if (overstress viscoplasticity), cf. Eqs. (3.113), (3.118), (3.119)
    1b. $g = \frac{1}{\mu}\left(\frac{\bar{\sigma}-\kappa(\bar{e}^{\mathrm{p}})}{\kappa(\bar{e}^{\mathrm{p}})}\right)^m$,        $\tilde{f} = \Delta t\, g - \Delta\bar{e}^{\mathrm{p}}$
    2b. $g_{,\bar{\sigma}} = \frac{gm}{\bar{\sigma}-\kappa(\bar{e}^{\mathrm{p}})}$,        $g_{,\bar{e}^{\mathrm{p}}} = -g_{,\bar{\sigma}}\,\frac{\bar{\sigma}\kappa'(\bar{e}^{\mathrm{p}})}{\kappa(\bar{e}^{\mathrm{p}})}$
    3b. $\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}} = \Delta t\,(g_{,\bar{\sigma}}\,\bar{\sigma}_{,\Delta\bar{e}^{\mathrm{p}}} + g_{,\bar{e}^{\mathrm{p}}}) - 1$
else if (power-law hardening viscoplast.), cf. Eqs. (3.113), (3.118), (3.120)
    1c. $g_1 = E\varepsilon_0\left(1 + \frac{\bar{e}^{\mathrm{p}}}{\varepsilon_0}\right)^{\frac{1}{n}}$,        $g = \dot{\varepsilon}_0\left(\frac{\bar{\sigma}}{g_1}\right)^{\frac{1}{m}}$,        $\tilde{f} = \Delta t\, g - \Delta\bar{e}^{\mathrm{p}}$
    2c. $g_{,\bar{\sigma}} = \frac{g}{m\bar{\sigma}}$,        $g_{,\bar{e}^{\mathrm{p}}} = -\frac{g}{mn(\bar{e}^{\mathrm{p}}+\varepsilon_0)}$
    3c. $\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}} = \Delta t\,(g_{,\bar{\sigma}}\,\bar{\sigma}_{,\Delta\bar{e}^{\mathrm{p}}} + g_{,\bar{e}^{\mathrm{p}}}) - 1$
end if
```

if $\alpha = 1$ (the fully implicit backward Euler scheme), thus only this case will be considered in further discussion.

A compact step-by-step algorithm of incremental constitutive computations for small-deformation isotropic elasto-plasticity is displayed in Table 3.1. Additionally, Tables 3.2 and 3.3 display the iterative routine to compute, for different elasto-plastic and elasto-viscoplastic models, equivalent plastic strain increment $\Delta\bar{e}^{\mathrm{p}}$, needed in Step 4 in Table 3.1.

The algorithmic tangent stiffness tensor defined by Eq. (3.94) can be determined through appropriate differentiation of the equations displayed in Tables 3.1–3.3. Detailed derivations presented in Appendix A.1 lead finally to the following form,

$$\boldsymbol{\mathcal{C}} = K\,\boldsymbol{I}\otimes\boldsymbol{I} + 2G(1-\vartheta)\left(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\right) - 2G(\gamma-\vartheta)\,\boldsymbol{n}\otimes\boldsymbol{n}\,, \tag{3.122}$$

where

$$\vartheta = \frac{3G\Delta\bar{e}^{\mathrm{p}}}{\bar{\sigma}_{\mathrm{trial}}}$$

and

$$\gamma = \begin{cases} \dfrac{3G}{3G+H'+H''\Delta\bar{e}^{\mathrm{p}}+\kappa'} & \text{for rate-independent plasticity} \\[2ex] \dfrac{3G\Delta t\, g_{,\bar{\sigma}}}{\Delta t\left[(3G+H'+H''\Delta\bar{e}^{\mathrm{p}})\, g_{,\bar{\sigma}} - g_{,\bar{e}^{\mathrm{p}}}\right]+1} & \text{for viscoplasticity} \end{cases}$$

(with all quantities taken at t_{n+1}). For $\Delta t \to 0$ and for rate-independent formulation (which also implies $\Delta\bar{e}^{\mathrm{p}} \to 0$ and $\vartheta \to 0$), the tensor $\boldsymbol{\mathcal{C}}$ converges to its continuum value $\boldsymbol{\mathcal{C}}^{\mathrm{tg}}$ given by Eq. (3.56).

It is finally worth noting that the above constitutive formulation is fully compatible with purely elastic formulation (3.30) for the case of no plastic flow. Particularly, the tensor $\boldsymbol{\mathcal{C}}$ takes then (i.e. for $\vartheta = 0$ and $\gamma = 0$) the form (3.26).

3.3.2. Large deformation formulation

Similarly as in the rate-type formulations, we could easily extend the above derivations to the case of finite deformations by merely replacing linearized stresses $\boldsymbol{\sigma}$, $\boldsymbol{\alpha}$, $\boldsymbol{s}$ and strains $\boldsymbol{\varepsilon}$ with their finite-deformation counterparts $\boldsymbol{T}$, $\boldsymbol{A}$, $\boldsymbol{S}$ and $\boldsymbol{E}$, expressed in the initial reference configuration. The entire algorithm is exactly the same and schemes presented in Tables 3.1–3.3 can be directly reused after appropriate redefinition of variables. In particular, algorithmic tangent constitutive tensor $\boldsymbol{\mathfrak{C}}$, see Eq. (3.95), assumes the form (3.122), with only $\boldsymbol{n}$ replaced by $\boldsymbol{N} = \boldsymbol{S}_{\text{trial}}/\|\boldsymbol{S}_{\text{trial}}\|$.

Such a procedure can be applied for cases of moderate strains/rotations, see [18] for discussion on more precise requirements, as well as for the large deformation elastic problems. In a general case of large elasto-plastic deformations, however, we will have to discretize in time the formulation based on the multiplicative split of the deformation gradient (3.66), presented in Section 3.2.4.

Let us assume all quantities at $t = t_n$ to be known and the new configuration $\mathcal{C}^{t_{n+1}}$ to be given. Knowing deformation gradients $\boldsymbol{F}_n$ and $\boldsymbol{F}_{n+1}$ let us define an 'incremental' deformation gradient from $\mathcal{C}^{t_n}$ to $\mathcal{C}^{t_{n+1}}$ as

$$\boldsymbol{f}_{n+1} = \frac{\mathrm{d}\boldsymbol{x}_{n+1}}{\mathrm{d}\boldsymbol{x}_n} = \boldsymbol{F}_{n+1}\boldsymbol{F}_n^{-1}\,, \qquad j_{n+1} = \det\,\boldsymbol{f}_{n+1}\,. \tag{3.123}$$

Recalling relations (3.82) and applying the fully implicit time integration scheme ($\alpha = 1$), let us approximate time derivatives of an arbitrary pair of spatial and material tensors ($\boldsymbol{\psi}$, $\boldsymbol{\Psi}$) at t_{n+1} as

$$\dot{\boldsymbol{\Psi}}_{n+1} \approx \frac{1}{\Delta t}(\boldsymbol{\Psi}_{n+1} - \boldsymbol{\Psi}_n)\,, \tag{3.124}$$

$$(\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\psi})_{n+1} = \boldsymbol{F}_{n+1}\,\dot{\boldsymbol{\Psi}}_{n+1}\boldsymbol{F}_{n+1}^{\mathrm{T}} \approx \frac{1}{\Delta t}(\boldsymbol{\psi}_{n+1} - \boldsymbol{\psi}_{\text{trial}}), \tag{3.125}$$

where the trial value of $\boldsymbol{\psi}$, defined as

$$\boldsymbol{\psi}_{\text{trial}} = \boldsymbol{F}_{n+1}\,\boldsymbol{\Psi}_n\boldsymbol{F}_{n+1}^{\mathrm{T}} = \boldsymbol{f}_{n+1}\boldsymbol{\psi}_n\boldsymbol{f}_{n+1}^{\mathrm{T}}, \tag{3.126}$$

is explicitly known. We can thus approximate the final value of $\boldsymbol{\psi}_{n+1}$ as

$$\boldsymbol{\psi}_{n+1} \approx \boldsymbol{\psi}_{\text{trial}} + \Delta t\,(\mathcal{L}_{\boldsymbol{v}}\boldsymbol{\psi})_{n+1} \tag{3.127}$$

Defining additionally $\bar{\boldsymbol{\psi}}_{\text{trial}} = J_{n+1}^{-\frac{2}{3}}\boldsymbol{\psi}_{\text{trial}}$, $\bar{\boldsymbol{\psi}}_n = J_n^{-\frac{2}{3}}\boldsymbol{\psi}_n$, and $\bar{\boldsymbol{f}}_{n+1} = j_{n+1}^{-\frac{1}{3}}\,\boldsymbol{f}_{n+1}$, we may also write

$$\bar{\boldsymbol{\psi}}_{n+1} \approx \bar{\boldsymbol{\psi}}_{\text{trial}} + \Delta t\,(\mathcal{L}_{\boldsymbol{v}}\bar{\boldsymbol{\psi}})_{n+1}\,, \qquad \bar{\boldsymbol{\psi}}_{\text{trial}} = \bar{\boldsymbol{f}}_{n+1}\bar{\boldsymbol{\psi}}_n\bar{\boldsymbol{f}}_{n+1}^{\mathrm{T}}. \tag{3.128}$$

Making use of the above relations, we can write the constitutive equations presented in Section 3.2.4 in the incremental form. Applying Eq. (3.127) to particular tensors $\boldsymbol{b}^{\mathrm{e}}$ and $\boldsymbol{\alpha}$ and denoting (cf. Eq. (3.80))

$$\bar{\sigma}_{n+1} = \sqrt{\tfrac{3}{2}}\,\|\boldsymbol{s}_{n+1}\| = \sqrt{\tfrac{3}{2}\,\boldsymbol{s}_{n+1} : \boldsymbol{s}_{n+1}}\,, \tag{3.129a}$$

$$\boldsymbol{s}_{n+1} = \operatorname{dev}(\boldsymbol{\tau}_{n+1} - \bar{\boldsymbol{\alpha}}_{n+1}), \tag{3.129b}$$

we can approximate Eqs. (3.86) and (3.88) as

$$\operatorname{dev}(\boldsymbol{b}^{\mathrm{e}}_{n+1}) \;=\; \operatorname{dev}(\boldsymbol{b}^{\mathrm{e}}_{\mathrm{trial}}) - 3\psi_{n+1}\Delta\bar{e}^{\mathrm{p}}\,\frac{\boldsymbol{s}_{n+1}}{\bar{\sigma}_{n+1}}\,, \qquad \operatorname{tr}(\boldsymbol{b}^{\mathrm{e}}_{n+1}) = \operatorname{tr}(\boldsymbol{b}^{\mathrm{e}}_{\mathrm{trial}}), \tag{3.130}$$

$$\operatorname{dev}(\boldsymbol{\alpha}_{n+1}) \;=\; \operatorname{dev}(\boldsymbol{\alpha}_{\mathrm{trial}}) + H'_{n+1}\,\psi_{n+1}\Delta\bar{e}^{\mathrm{p}}\,\frac{\boldsymbol{s}_{n+1}}{\bar{\sigma}_{n+1}}\,, \qquad \operatorname{tr}(\boldsymbol{\alpha}_{n+1}) = \operatorname{tr}(\boldsymbol{\alpha}_{\mathrm{trial}}), \tag{3.131}$$

where $H'_{n+1} = H'(\bar{e}^{\mathrm{p}}_{n+1})$. To simplify further derivations, let us introduce the following notations,

$$\boldsymbol{\tau}^{\star} = G\boldsymbol{b}^{\mathrm{e}}, \qquad \bar{\boldsymbol{\tau}}^{\star} = G\bar{\boldsymbol{b}}^{\mathrm{e}}, \qquad \varphi = 3GJ^{-\frac{2}{3}}\psi\,, \tag{3.132}$$

so that, cf. Eq. (3.76), $\operatorname{dev}\bar{\boldsymbol{\tau}}^{\star} = \operatorname{dev}\boldsymbol{\tau}$ (but $\bar{\boldsymbol{\tau}}^{\star} \neq \boldsymbol{\tau}$!), and, cf. Eqs. $(3.80)_1$, (3.81),

$$\boldsymbol{s} = \operatorname{dev}(\bar{\boldsymbol{\tau}}^{\star} - \bar{\boldsymbol{\alpha}}), \qquad \varphi = \operatorname{tr}(\bar{\boldsymbol{\tau}}^{\star} - \bar{\boldsymbol{\alpha}}). \tag{3.133}$$

Scaling now Eqs. (3.130) and (3.131) by $GJ^{-\frac{2}{3}}$ and $J^{-\frac{2}{3}}$, respectively, and recalling additionally Eq. (3.76), we come at the following time-discrete system,

$$\operatorname{dev}(\bar{\boldsymbol{\tau}}^{\star}_{n+1}) \;=\; \operatorname{dev}(\bar{\boldsymbol{\tau}}^{\star}_{\mathrm{trial}}) - \varphi_{n+1}\Delta\bar{e}^{\mathrm{p}}\,\frac{\boldsymbol{s}_{n+1}}{\bar{\sigma}_{n+1}}\,, \qquad \operatorname{tr}(\bar{\boldsymbol{\tau}}^{\star}_{n+1}) = \operatorname{tr}(\bar{\boldsymbol{\tau}}^{\star}_{\mathrm{trial}}), \tag{3.134}$$

$$\operatorname{dev}(\bar{\boldsymbol{\alpha}}_{n+1}) \;=\; \operatorname{dev}(\bar{\boldsymbol{\alpha}}_{\mathrm{trial}}) + \frac{H'_{n+1}}{3G}\,\varphi_{n+1}\Delta\bar{e}^{\mathrm{p}}\,\frac{\boldsymbol{s}_{n+1}}{\bar{\sigma}_{n+1}}\,, \qquad \operatorname{tr}(\bar{\boldsymbol{\alpha}}_{n+1}) = \operatorname{tr}(\bar{\boldsymbol{\alpha}}_{\mathrm{trial}}), \tag{3.135}$$

$$\operatorname{dev}(\boldsymbol{\tau}_{n+1}) \;=\; \operatorname{dev}(\bar{\boldsymbol{\tau}}^{\star}_{n+1}), \qquad \operatorname{tr}(\boldsymbol{\tau}_{n+1}) = \tfrac{3}{2}K(J^2_{n+1} - 1), \tag{3.136}$$

where trial values $\bar{\boldsymbol{\tau}}^{\star}_{\mathrm{trial}}$ and $\bar{\boldsymbol{\alpha}}_{\mathrm{trial}}$ can be determined from Eq. $(3.128)_2$, and, in view of Eqs. $(3.133)_2$, $(3.134)_2$ and $(3.135)_2$,

$$\varphi_{n+1} = \operatorname{tr}\left(\bar{\boldsymbol{\tau}}^{\star}_{n+1} - \bar{\boldsymbol{\alpha}}_{n+1}\right) = \operatorname{tr}\left(\bar{\boldsymbol{\tau}}^{\star}_{\mathrm{trial}} - \bar{\boldsymbol{\alpha}}_{\mathrm{trial}}\right) = \varphi_{\mathrm{trial}}\,, \tag{3.137}$$

i.e. it is explicitly known.

The system of equations (3.134)–(3.136) together with notations (3.129), (3.133) includes the unknown end-of-increment values of stress $\boldsymbol{\tau}_{n+1}$ and state parameters $\bar{\boldsymbol{\tau}}^{\star}_{n+1}$ (or $\bar{\boldsymbol{b}}^{\mathrm{e}}_{n+1} = \frac{1}{G}\bar{\boldsymbol{\tau}}^{\star}_{n+1}$), $\bar{\boldsymbol{\alpha}}_{n+1}$ and $\bar{e}^{\mathrm{p}}_{n+1} = \bar{e}^{\mathrm{p}}_{n} + \Delta\bar{e}^{\mathrm{p}}$. To complete it, the evolution rule for $\Delta\bar{e}^{\mathrm{p}}$ has to be derived from the rate-type equations (3.50) (for rate-independent elasto-plasticity) or (3.57) (for viscoplasticity). It is assumed in exactly the same form as in the small-deformation formulation, cf. Eqs. (3.111)–(3.113). However, in the case of pure elastic deformation, $\Delta\bar{e}^{\mathrm{p}} = 0$ is set instead and Eqs. (3.134)–(3.136) yield the elastic Kirchhoff stress increment and unchanged tensorial constitutive state parameters push-forwarded to the current configuration.

To solve the system of incremental elasto-plastic flow equations we proceed as follows. Subtracting Eq. $(3.135)_1$ from Eq. $(3.136)_1$ and substituting Eqs. $(3.134)_1$ and (3.137), we come at

$$\boldsymbol{s}_{n+1}\left[1+\left(1+\frac{H'_{n+1}}{3G}\right)\varphi_{\mathrm{trial}}\Delta\bar{e}^{\mathrm{p}}\,\frac{1}{\bar{\sigma}_{n+1}}\right]=\boldsymbol{s}_{\mathrm{trial}} \tag{3.138}$$

where $\boldsymbol{s}_{\mathrm{trial}} = \mathrm{dev}(\bar{\boldsymbol{\tau}}^{\star}_{\mathrm{trial}} - \bar{\boldsymbol{\alpha}}_{\mathrm{trial}})$. This means that $\boldsymbol{s}_{\mathrm{trial}}$ determines the direction of plastic flow and back stress evolution and that the algorithm is of the 'radial-return' type. Taking a norm of both sides of Eq. (3.138) and defining

$$\bar{\sigma}_{\mathrm{trial}} = \sqrt{\tfrac{3}{2}\,\boldsymbol{s}_{\mathrm{trial}} : \boldsymbol{s}_{\mathrm{trial}}}$$

we get

$$\bar{\sigma}_{n+1} = \bar{\sigma}_{\mathrm{trial}} - \left(1+\frac{H'_{n+1}}{3G}\right)\varphi_{\mathrm{trial}}\Delta\bar{e}^{\mathrm{p}}, \qquad \frac{\boldsymbol{s}_{n+1}}{\bar{\sigma}_{n+1}} = \frac{\boldsymbol{s}_{\mathrm{trial}}}{\bar{\sigma}_{\mathrm{trial}}}\,. \tag{3.139}$$

Rewriting Eq. (3.111),

$$\tilde{f}(\bar{\sigma}_{n+1}, \bar{e}^{\mathrm{p}}_{n+1}, \Delta\bar{e}^{\mathrm{p}}) = 0,$$

we can see that, upon substitution of Eq. $(3.139)_1$, this is a nonlinear scalar equation for $\Delta\bar{e}^{\mathrm{p}}$ that can be solved with the Newton's iteration scheme, cf. (3.114)

$$\begin{tikzcd}
\delta(\Delta\bar{e}^{\mathrm{p}}) = -\dfrac{\tilde{f}^{(i)}}{\tilde{f}^{(i)}_{,\Delta\bar{e}^{\mathrm{p}}}} \arrow[rr] & & \Delta\bar{e}^{\mathrm{p}(i+1)} = \Delta\bar{e}^{\mathrm{p}(i)} + \delta(\Delta\bar{e}^{\mathrm{p}}) \arrow[dl] \\
& i := i+1 \arrow[ul] &
\end{tikzcd} \tag{3.140}$$

where the coefficient $\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}$ is expressed with Eq. (3.115), with, cf. Eq. $(3.139)_1$,

$$\bar{\sigma}_{n+1,\Delta\bar{e}^{\mathrm{p}}} \equiv \frac{\mathrm{d}\bar{\sigma}_{n+1}}{\mathrm{d}\Delta\bar{e}^{\mathrm{p}}} = -\left[1+\frac{H'+H''\Delta\bar{e}^{\mathrm{p}}}{3G}\right]\varphi_{\mathrm{trial}}\,. \tag{3.141}$$

The scheme appears to be nearly the same as for small deformations, particularly, $\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}$ can be computed from the equations given in Tables 3.2–3.3, with the only difference in $\bar{\sigma}_{n+1}$ and $\bar{\sigma}_{,\Delta\bar{e}^{\mathrm{p}}}$ that are given here by Eqs. $(3.139)_1$ and (3.141).

Having $\Delta\bar{e}^{\mathrm{p}}$, we can update stress and the remaining state parameters with the use of Eqs. (3.134)–(3.136). A compact step-by-step algorithm of incremental constitutive computations for small-deformation isotropic elasto-plasticity is displayed in Table 3.4. Additionally, Table 3.5 (nearly identical to Table 3.2) displays the iterative routine to compute equivalent plastic strain increment $\Delta\bar{e}^{\mathrm{p}}$, needed in Step 6 in Table 3.4.

The algorithmic tangent stiffness tensor $\boldsymbol{\mathfrak{c}}$ defined with Eqs. (3.96)–(3.97), consistent with the above discussed time integration scheme, can be derived as follows. In view of Eqs. $(3.134)_1$ and (3.136), the stress $\boldsymbol{\tau}_{n+1}$ can be expressed as

$$\boldsymbol{\tau}_{n+1} = \boldsymbol{\tau}_{\mathrm{trial}} - \boldsymbol{s}^{\mathrm{p}}_{n+1} \tag{3.142}$$

with

$$\boldsymbol{\tau}_{\mathrm{trial}} = \mathrm{dev}(\bar{\boldsymbol{\tau}}^{\star}_{\mathrm{trial}}) + \tfrac{1}{2}\boldsymbol{I}\,K(J^2_{n+1}-1)\,, \tag{3.143a}$$

$$\boldsymbol{s}^{\mathrm{p}}_{n+1} = \varphi_{n+1}\Delta\bar{e}^{\mathrm{p}}\,\frac{\boldsymbol{s}_{\mathrm{trial}}}{\bar{\sigma}_{\mathrm{trial}}} = \sqrt{\tfrac{2}{3}}\,\varphi_{n+1}\Delta\bar{e}^{\mathrm{p}}\,\boldsymbol{n}\,. \tag{3.143b}$$

Thus,

$$\boldsymbol{\mathfrak{c}} = \boldsymbol{\mathfrak{c}}^{\mathrm{e}} - \boldsymbol{\mathfrak{c}}^{\mathrm{p}}, \qquad \mathrm{d}^{\circ}\boldsymbol{\tau}_{\mathrm{trial}} = \boldsymbol{\mathfrak{c}}^{\mathrm{e}} : \mathrm{d}\boldsymbol{\epsilon}_{n+1}\,, \qquad \mathrm{d}^{\circ}\boldsymbol{s}^{\mathrm{p}}_{n+1} = \boldsymbol{\mathfrak{c}}^{\mathrm{p}} : \mathrm{d}\boldsymbol{\epsilon}_{n+1}\,. \tag{3.144}$$

The term $\boldsymbol{\mathfrak{c}}^{\mathrm{e}}$ is obviously the elastic tensor $\boldsymbol{\mathfrak{c}}^{\mathrm{e}}$ (3.78) in which $\bar{\boldsymbol{b}}^{\mathrm{e}}$ is replaced by $\bar{\boldsymbol{b}}^{\mathrm{e}}_{\mathrm{trial}}$,

$$\begin{aligned}
\boldsymbol{\mathfrak{c}}^{\mathrm{e}} &= K\left[J^2_{n+1}\,\boldsymbol{I}\otimes\boldsymbol{I} - (J^2_{n+1}-1)\,\boldsymbol{\mathfrak{I}}\right] \\
&\qquad + \tfrac{2}{3}G\left[(\mathrm{tr}\bar{\boldsymbol{b}}^{\mathrm{e}}_{\mathrm{trial}})\big(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\big) - \big(\mathrm{dev}(\bar{\boldsymbol{b}}^{\mathrm{e}}_{\mathrm{trial}})\otimes\boldsymbol{I} + \boldsymbol{I}\otimes\mathrm{dev}(\bar{\boldsymbol{b}}^{\mathrm{e}}_{\mathrm{trial}})\big)\right] \\
&= K\left[J^2_{n+1}\,\boldsymbol{I}\otimes\boldsymbol{I} - (J^2_{n+1}-1)\,\boldsymbol{\mathfrak{I}}\right] \\
&\qquad + \tfrac{2}{3}\left[(\mathrm{tr}\bar{\boldsymbol{\tau}}^{\star}_{\mathrm{trial}})\big(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\big) - (\mathrm{dev}(\bar{\boldsymbol{\tau}}^{\star}_{\mathrm{trial}})\otimes\boldsymbol{I} + \boldsymbol{I}\otimes\mathrm{dev}(\bar{\boldsymbol{\tau}}^{\star}_{\mathrm{trial}}))\right],
\end{aligned} \tag{3.145a}$$

while $\boldsymbol{\mathfrak{c}}^{\mathrm{p}}$ results from the appropriate differentiation of $\boldsymbol{s}^{\mathrm{p}}_{n+1}$. For the particular case of linear hardening rate-independent elasto-plasticity, the form of this tensor has been derived in [111]. Derivation of a more general form including nonlinear hardening and viscoplastic effects, presented in Appendix A.2, leads to the following result,

$$\boldsymbol{\mathfrak{c}}^{\mathrm{p}} = C_1\big(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\big) + C_2\,(\boldsymbol{n}\otimes\boldsymbol{I} + \boldsymbol{I}\otimes\boldsymbol{n}) + C_3\,\boldsymbol{n}\otimes\boldsymbol{n} + C_4\,\boldsymbol{n}\otimes\mathrm{dev}(\boldsymbol{n}^2) \tag{3.145b}$$

Table 3.4. Solution algorithm for large-deformation elasto-plasticity (for the sake of legibility, subscripts $(n{+}1)$ have been omitted)

```
Given
    deformation gradient $\boldsymbol{F}_n$ and state parameters[a] $\mathbf{z}_n = \{\, \bar{\boldsymbol{\tau}}^\star_n \,,\, \bar{\boldsymbol{\alpha}}_n \,,\, \bar{e}^{\mathrm{p}}_n \,\}$ at $t_n$ ,
    deformation gradient $\boldsymbol{F}$ at $t_{n+1}$ ,
compute
    state parameters $\mathbf{z} = \{\, \bar{\boldsymbol{\tau}}^\star \,,\, \bar{\boldsymbol{\alpha}} \,,\, \bar{e}^{\mathrm{p}} \,\}$ and stress $\boldsymbol{\tau}$ at $t_{n+1}$
using the following scheme:
1. $\boldsymbol{f} = \boldsymbol{F}\boldsymbol{F}_n^{-1}, \quad j = \det \boldsymbol{f}, \quad \bar{\boldsymbol{f}} = j^{-\frac{1}{3}}\boldsymbol{f}$ – 'incremental' deformation gradient
2. $\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}} = \bar{\boldsymbol{f}}\bar{\boldsymbol{\tau}}^\star_n\bar{\boldsymbol{f}}^{\mathrm{T}}, \quad \bar{\boldsymbol{\alpha}}_{\mathrm{trial}} = \bar{\boldsymbol{f}}\bar{\boldsymbol{\alpha}}_n\bar{\boldsymbol{f}}^{\mathrm{T}}$ – trial values of tensorial state param.
3. $\boldsymbol{s}_{\mathrm{trial}} = \operatorname{dev}(\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}} - \bar{\boldsymbol{\alpha}}_{\mathrm{trial}})$ – trial relative stress
4. $\bar{\sigma}_{\mathrm{trial}} = \sqrt{\frac{3}{2}\,\boldsymbol{s}_{\mathrm{trial}} : \boldsymbol{s}_{\mathrm{trial}}}$ – trial equivalent stress
if $\bar{\sigma}_{\mathrm{trial}} - \kappa(\bar{e}^{\mathrm{p}}_n) > 0$ (plastic flow) then[b]
    5. $\varphi = \operatorname{tr}(\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}} - \bar{\boldsymbol{\alpha}}_{\mathrm{trial}})$ – coefficient $\varphi$
    6. $\Delta\bar{e}^{\mathrm{p}} \longleftarrow$ from Table 3.5 – equiv. plastic strain increment
    7. $\boldsymbol{s}^{\mathrm{p}} = \varphi\Delta\bar{e}^{\mathrm{p}}\,\frac{\boldsymbol{s}_{\mathrm{trial}}}{\bar{\sigma}_{\mathrm{trial}}}$ – scaled flow direction
    8. $\left.\begin{aligned} \bar{\boldsymbol{\tau}}^\star &= \operatorname{dev}(\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}}) - \boldsymbol{s}^{\mathrm{p}} + \tfrac{1}{3}\boldsymbol{I}\operatorname{tr}\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}} \\ \bar{\boldsymbol{\alpha}} &= \operatorname{dev}(\bar{\boldsymbol{\alpha}}_{\mathrm{trial}}) + \tfrac{H'}{3G}\,\boldsymbol{s}^{\mathrm{p}} + \tfrac{1}{3}\boldsymbol{I}\operatorname{tr}\bar{\boldsymbol{\alpha}}_{\mathrm{trial}} \\ \bar{e}^{\mathrm{p}} &= \bar{e}^{\mathrm{p}}_n + \Delta\bar{e}^{\mathrm{p}} \end{aligned}\right\}$ update state parameters
else (purely elastic deformation)
    9. $\bar{\boldsymbol{\tau}}^\star = \bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}}\,, \quad \bar{\boldsymbol{\alpha}} = \bar{\boldsymbol{\alpha}}_{\mathrm{trial}}\,, \quad \bar{e}^{\mathrm{p}} = \bar{e}^{\mathrm{p}}_n$ – update state parameters
end if
10. $\boldsymbol{\tau} = \operatorname{dev}(\bar{\boldsymbol{\tau}}^\star) + \frac{1}{2}\boldsymbol{I}\,K[(\det\boldsymbol{F})^2 - 1]$ – final stress
```

[a] $\bar{\boldsymbol{b}}^{\mathrm{e}}$ in (3.92) has been replaced by $\bar{\boldsymbol{\tau}}^\star = G\bar{\boldsymbol{b}}^{\mathrm{e}}$; if $H \equiv \text{const}$ (no kinematic hardening) then $\bar{\boldsymbol{\alpha}} \equiv \mathbf{0}$ is not a state parameter and all relevant computations are skipped

[b] for the power-law strain/strain-rate hardening viscoplasticity this condition is always fulfilled, except for $\bar{\sigma}_{\mathrm{trial}} = 0$

Table 3.5. Solution algorithm for the equivalent plastic strain increment $\Delta\bar{e}^{\mathrm{p}}$ cf. Eq. (3.140), large deformation formulation

```
1. $\Delta\bar{e}^{\mathrm{p}} = 0$
2. $\bar{e}^{\mathrm{p}} = \bar{e}^{\mathrm{p}}_{n} + \Delta\bar{e}^{\mathrm{p}}$
3. $\bar{\sigma} = \bar{\sigma}_{\mathrm{trial}} - \left[1 + \frac{H'(\bar{e}^{\mathrm{p}})}{3G}\right] \varphi\Delta\bar{e}^{\mathrm{p}}$,    $\bar{\sigma}_{,\Delta\bar{e}^{\mathrm{p}}} = -\left[1 + \frac{H'(\bar{e}^{\mathrm{p}})+H''(\bar{e}^{\mathrm{p}})\,\Delta\bar{e}^{\mathrm{p}}}{3G}\right] \varphi$
4. $\tilde{f},\ \tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}$   $\longleftarrow$   from Table 3.3
5. $\delta(\Delta\bar{e}^{\mathrm{p}}) = -\tilde{f}^{-1}_{,\Delta\bar{e}^{\mathrm{p}}}\,\tilde{f}$,    $\Delta\bar{e}^{\mathrm{p}} = \Delta\bar{e}^{\mathrm{p}} + \delta(\Delta\bar{e}^{\mathrm{p}})$
6. **if** $\|\delta(\Delta\bar{e}^{\mathrm{p}})\| > \mathrm{tol}$ **then go to step 2; else exit**
```

(see Appendix A.2 for explanation of the coefficients C_i). It is important to note that, due to the last term with the coefficient C_4, this tensor is nonsymmetric.

It is finally worth noting that the above constitutive formulation is fully compatible with purely elastic formulation (3.38) for the case of no plastic flow. Particularly, $\boldsymbol{\mathfrak{c}}^{\mathrm{p}}$ vanishes and the tensor $\boldsymbol{\mathfrak{c}}$ takes then the form (3.40).

3.3.3. Reduced-dimension formulations

The constitutive models presented in the previous sections are suitable for solutions of 3-D continuum problems and they are deformation-driven, i.e. they assume that the full tensor of strain or deformation gradient is given while no kinematical restrictions are imposed on stress and state parameters. In practical engineering computations this is not always the case, especially in reduced-dimension (2-D, 1-D) continuum formulations. While the 3-D equations may be directly used to plane strain and axisymmetric problems in which kinematical limitations only concern deformation and thus they may be imposed on the given quantities, the computations of plane-stress problems, or even one-dimensional stretching, require modified constitutive algorithms in which some deformation state components are not explicitly known while some stress components are given.

Before the computational details are presented, it is necessary to introduce a notation, frequently referred to as the *finite element vector/matrix notation*, in which symmetric 2nd rank tensors are represented in a specified Cartesian coordinate system as 6×1 vectors,

$$\boldsymbol{a} = [a_{ij}] \quad\longrightarrow\quad \mathbf{a}_{6\times 1} = \Big\{ a_{11}\,,\ a_{22}\,,\ a_{33}\,,\ \sqrt{2}\,a_{23}\,,\ \sqrt{2}\,a_{31}\,,\ \sqrt{2}\,a_{12} \Big\}, \qquad (3.146\text{a})$$

while 4th rank tensors exhibiting symmetries typical for constitutive stiffness tensors, i.e. $\mathfrak{A}_{ijkl} = \mathfrak{A}_{jikl} = \mathfrak{A}_{ijlk} = \mathfrak{A}_{klij}$ — as symmetric 6×6 matrices,

$$\boldsymbol{\mathfrak{A}} = [\mathfrak{A}_{ijkl}] \longrightarrow \mathbf{A}_{6\times6} = \begin{bmatrix} \mathfrak{A}_{1111} & \mathfrak{A}_{1122} & \mathfrak{A}_{1133} & \sqrt{2}\,\mathfrak{A}_{1123} & \sqrt{2}\,\mathfrak{A}_{1131} & \sqrt{2}\,\mathfrak{A}_{1112} \\ & \mathfrak{A}_{2222} & \mathfrak{A}_{2233} & \sqrt{2}\,\mathfrak{A}_{2223} & \sqrt{2}\,\mathfrak{A}_{2231} & \sqrt{2}\,\mathfrak{A}_{2212} \\ & & \mathfrak{A}_{3333} & \sqrt{2}\,\mathfrak{A}_{3323} & \sqrt{2}\,\mathfrak{A}_{3331} & \sqrt{2}\,\mathfrak{A}_{3312} \\ & & & 2\,\mathfrak{A}_{2323} & 2\,\mathfrak{A}_{2331} & 2\,\mathfrak{A}_{2312} \\ & \text{sym.} & & & 2\,\mathfrak{A}_{3131} & 2\,\mathfrak{A}_{3112} \\ & & & & & 2\,\mathfrak{A}_{1212} \end{bmatrix}. \tag{3.146b}$$

In this notation, all scalar tensor products like $\boldsymbol{a} : \boldsymbol{b}$ or $\boldsymbol{a} : \boldsymbol{\mathfrak{A}} : \boldsymbol{b}$ (in the absolute notation) give the same results as the corresponding vector/matrix products $\mathbf{a}^{\mathrm{T}}\mathbf{b}$, $\mathbf{a}^{\mathrm{T}}\mathbf{A}\mathbf{b}$, respectively, while e.g. the tensor $\boldsymbol{c} = \boldsymbol{\mathfrak{A}} : \boldsymbol{b}$ (in the absolute notation) is represented by the vector $\mathbf{c} = \mathbf{A}\mathbf{b}$ (in the vector/matrix notation). In particular, we have

$$\begin{aligned} &\boldsymbol{I} && \longrightarrow && \mathbf{1}_{6\times1} = \{1\,,\,1\,,\,1\,,\,0\,,\,0\,,\,0\}, \\ &\boldsymbol{\mathfrak{I}} && \longrightarrow && \mathbf{I}_{6\times6} = \operatorname{diag}[1\,,\,1\,,\,1\,,\,1\,,\,1\,,\,1], \\ &\boldsymbol{\mathfrak{P}} = \boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I} && \longrightarrow && \mathbf{P}_{6\times6} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned}$$

so that $\operatorname{dev}\boldsymbol{a} = \boldsymbol{\mathfrak{P}} : \boldsymbol{a}$ in the absolute notation corresponds to $\operatorname{dev}\mathbf{a} = \mathbf{P}\mathbf{a}$ in the vector/matrix notation.

In reduced-dimension formulations, certain components can be simply removed from the arrays, thus reducing their size. E.g., in plane formulations,

$$\begin{aligned} &\boldsymbol{a} && \longrightarrow && \mathbf{a}_{3\times1} = \left\{ a_{11}\,,\, a_{22}\,,\, \sqrt{2}\,a_{12} \right\}, \\ &\boldsymbol{\mathfrak{A}} && \longrightarrow && \mathbf{A}_{3\times3} = \begin{bmatrix} \mathfrak{A}_{1111} & \mathfrak{A}_{1122} & \sqrt{2}\,\mathfrak{A}_{1112} \\ & \mathfrak{A}_{2222} & \sqrt{2}\,\mathfrak{A}_{2212} \\ \text{sym.} & & 2\,\mathfrak{A}_{1212} \end{bmatrix}. \end{aligned}$$

Two examples of reduced-dimension formulations will be presented: *one-dimensional bar* and *plane stress.* Both are limited to isotropic elastic behaviour and small deformation description, i.e. Eqs. (3.99)–(3.104) (for $\alpha = 1$) and (3.111)–(3.113) must be fulfilled.

3.3.3.1. One-dimensional bar

In this case, stress and strain are expressed as 6×1 vectors

$$\begin{aligned}
&\boldsymbol{\sigma} = \{\sigma_{\|}\,,0\,,0\,,0\,,0\,,0\}\,, && \mathrm{dev}\boldsymbol{\sigma} = \{\tfrac{2}{3}\sigma_{\|}\,,-\tfrac{1}{3}\sigma_{\|}\,,-\tfrac{1}{3}\sigma_{\|}\,,0\,,0\,,0\}\,,\\
&\boldsymbol{\alpha} = \{\alpha_{\|}\,,0\,,0\,,0\,,0\,,0\}\,, && \mathrm{dev}\boldsymbol{\alpha} = \{\tfrac{2}{3}\alpha_{\|}\,,-\tfrac{1}{3}\alpha_{\|}\,,-\tfrac{1}{3}\alpha_{\|}\,,0\,,0\,,0\}\,,\\
&\boldsymbol{\varepsilon} = \{\varepsilon_{\|}\,,\varepsilon_{\perp}\,,\varepsilon_{\perp}\,,0\,,0\,,0\}\,. && \qquad (3.147)
\end{aligned}$$

where the indices $\|$ and $\perp$ denote axial and transversal components.

Assume that all quantities, including $\bar{e}^{\mathrm{p}}$ are known at t_n and the axial strain increment $\Delta\varepsilon_{\|}$ is given at t_{n+1}, while $\sigma_{\|n+1}$ and $\Delta\varepsilon_{\perp}$ are unknown. Recalling the additive formula (3.99), elastic equation (3.100), and the assumption of isotropy, we have

$$\Delta\varepsilon^{\mathrm{e}}_{\perp} = -\nu\Delta\varepsilon^{\mathrm{e}}_{\|}\,, \qquad \Delta\sigma_{\|} = E\Delta\varepsilon^{\mathrm{e}}_{\|}\,.$$

Besides, Eq. (3.104) yields in this case

$$\begin{aligned}
\mathbf{n} \;&=\; \frac{\mathrm{dev}(\boldsymbol{\sigma}-\boldsymbol{\alpha})}{\|\mathrm{dev}(\boldsymbol{\sigma}-\boldsymbol{\alpha})\|} = \left\{\sqrt{\tfrac{2}{3}}\,,-\sqrt{\tfrac{1}{6}}\,,-\sqrt{\tfrac{1}{6}}\,,0\,,0\,,0\right\},\\
\Delta\varepsilon^{\mathrm{p}}_{\|} \;&=\; \Delta\bar{e}^{\mathrm{p}}\,, \qquad \Delta\varepsilon^{\mathrm{p}}_{\perp} = -\tfrac{1}{2}\Delta\varepsilon^{\mathrm{p}}_{\|}\,.
\end{aligned}$$

while from Eq. (3.106) we have

$$\Delta\alpha_{\|} = H'_{n+1}\Delta\bar{e}^{\mathrm{p}}\,.$$

Gathering the above results we can easily derive the formula for equivalent stress

$$\bar{\sigma}_{n+1} = \sigma_{\|n+1} - \alpha_{\|n+1} = \big(\sigma_{\|}-\alpha_{\|}\big)_n + E(\Delta\varepsilon_{\|}-\Delta\bar{e}^{\mathrm{p}}) - H'_{n+1}\Delta\bar{e}^{\mathrm{p}},$$

or

$$\bar{\sigma}_{n+1} = \bar{\sigma}_{\mathrm{trial}} - (E+H'_{n+1})\Delta\bar{e}^{\mathrm{p}}, \qquad \bar{\sigma}_{\mathrm{trial}} = \big(\sigma_{\|}-\alpha_{\|}\big)_n + E\Delta\varepsilon_{\|}\,, \qquad (3.148)$$

which, when substituted to the consistency condition (3.111)–(3.113), yields a scalar nonlinear equation to be solved with respect to $\Delta\bar{e}^{\mathrm{p}}$.

The compact step-by-step algorithm for this formulation is displayed in Tables 3.6–3.7.

Differentiation of equations in the tables leads to the following scalar tangent stiffness value consistent with the time integration scheme (cf. Eq. (3.94)),

$$\mathcal{C} = \frac{\mathrm{d}\sigma_{\|n+1}}{\mathrm{d}\varepsilon_{\|n+1}} = \frac{\mathrm{d}\sigma_{\|n+1}}{\mathrm{d}\Delta\varepsilon_{\|}} = E(1-\gamma), \qquad \gamma = -E\,\frac{\tilde{f}_{,\bar{\sigma}}}{\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}}\,. \qquad (3.149)$$

(for instance, for linear hardening rate-independent plasticity we have $\gamma = \frac{E}{E+\kappa'+H'}$).

Table 3.6. Solution algorithm for small-deformation 1-D bar formulation (for the sake of legibility, subscripts $(n{+}1)$ and $\|$ have been omitted)

```
Given
    state parameters[a] $\mathbf{z}_n = \{\,\sigma_n\,,\,\alpha_n\,,\,\bar{e}^{\mathrm{p}}_n\,\}$ at $t_n$ ,
    strain increment $\Delta\varepsilon$
compute
    state parameters (including stress) $\mathbf{z} = \{\,\sigma\,,\,\alpha\,,\,\bar{e}^{\mathrm{p}}\,\}$ at $t_{n+1}$ ,
    transverse strain increment $\Delta\varepsilon_{\perp}$
using the following scheme:

1. $\sigma_{\mathrm{trial}} = \sigma_n + E\Delta\varepsilon$                          – trial stress
2. $\bar{\sigma}_{\mathrm{trial}} = \sigma_{\mathrm{trial}} - \frac{2}{3}\alpha_n$                  – trial equivalent stress

if $\bar{\sigma}_{\mathrm{trial}} - \kappa(\bar{e}^{\mathrm{p}}_n) > 0$  (plastic flow) then[b]
    3. $\Delta\varepsilon^{\mathrm{p}} \equiv \Delta\bar{e}^{\mathrm{p}}$  $\longleftarrow$  from Table 3.7   – equiv. plastic strain increment
    4. $\Delta\varepsilon_{\perp} = -\nu(\Delta\varepsilon - \Delta\varepsilon^{\mathrm{p}}) - \frac{1}{2}\Delta\varepsilon^{\mathrm{p}}$   – transverse strain increment
    5. $\sigma = \sigma_{\mathrm{trial}} - E\Delta\varepsilon^{\mathrm{p}}$               $\Big\}$
       $\alpha = \alpha_n + H'\Delta\varepsilon^{\mathrm{p}}$                    $\Big\}$ update state parameters
       $\bar{e}^{\mathrm{p}} = \bar{e}^{\mathrm{p}}_n + \Delta\bar{e}^{\mathrm{p}}$                     $\Big\}$
else (purely elastic deformation)
    6. $\sigma = \sigma_{\mathrm{trial}}$ ,   $\alpha = \alpha_n$ ,   $\bar{e}^{\mathrm{p}} = \bar{e}^{\mathrm{p}}_n$   – update state parameters
end if
```

[a] if $H \equiv$ const (no kinematic hardening) then $\alpha \equiv 0$ is not a state parameter and all relevant computations are skipped

[b] for the power-law strain/strain-rate hardening viscoplasticity this condition is always fulfilled, except for $\bar{\sigma}_{\mathrm{trial}} = 0$

Table 3.7. Solution algorithm for the equivalent plastic strain increment $\Delta\bar{e}^{\mathrm{p}}$, 1-D bar formulation

```
1. $\Delta\bar{e}^{\mathrm{p}} = 0$
2. $\bar{e}^{\mathrm{p}} = \bar{e}^{\mathrm{p}}_{n} + \Delta\bar{e}^{\mathrm{p}}$
3. $\bar{\sigma} = \bar{\sigma}_{\mathrm{trial}} - [E + H'(\bar{e}^{\mathrm{p}})]\,\Delta\bar{e}^{\mathrm{p}}$,    $\bar{\sigma}_{,\Delta\bar{e}^{\mathrm{p}}} = -[E + H'(\bar{e}^{\mathrm{p}}) + H''(\bar{e}^{\mathrm{p}})\,\Delta\bar{e}^{\mathrm{p}}]$
4. $\tilde{f}$, $\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}$   $\longleftarrow$   from Table 3.3
5. $\delta(\Delta\bar{e}^{\mathrm{p}}) = -\tilde{f}^{-1}_{,\Delta\bar{e}^{\mathrm{p}}}\,\tilde{f}$,    $\Delta\bar{e}^{\mathrm{p}} = \Delta\bar{e}^{\mathrm{p}} + \delta(\Delta\bar{e}^{\mathrm{p}})$
6. if  $\|\delta(\Delta\bar{e}^{\mathrm{p}})\| > \mathrm{tol}$  then go to step 2;  else exit
```

3.3.3.2. Plane stress

The plane stress formulation presented below follows the idea of [116], written in slightly modified notation, and extends that formulation by including viscoplastic effects, cf. [68]. Assuming the coordinate x_3 to be perpendicular to the model plane, we can represent stresses and strains in the vector/matrix notation by reduced-dimension vectors

$$\begin{aligned} \boldsymbol{\sigma} &= \{\sigma_{11}\,,\ \sigma_{22}\,,\ \sqrt{2}\,\sigma_{12}\,\}\,, \\ \boldsymbol{\alpha} &= \{\alpha_{11}\,,\ \alpha_{22}\,,\ \sqrt{2}\,\alpha_{12}\,\}\,, \\ \boldsymbol{\varepsilon} &= \{\varepsilon_{11}\,,\ \varepsilon_{22}\,,\ \sqrt{2}\,\varepsilon_{12}\,\}\,, \end{aligned} \tag{3.150}$$

with all the other components equal to 0 except for the transverse strain, $\varepsilon_{33} = \varepsilon_{\perp}$, whose value is related to the other strain components through the constitutive equations. The reduced-dimension matrix form $\mathbf{P}$ of the deviator operator $\mathfrak{P} = \mathfrak{I} - \frac{1}{3}\boldsymbol{I} \otimes \boldsymbol{I}$,

$$\mathbf{P} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & \frac{2}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

can be, however, used only with respect to stresses, since it assumes the operand's transverse component to vanish. The operation $\mathbf{P}\boldsymbol{\sigma}$ allows to determine only in-plane components of $\mathrm{dev}\boldsymbol{\sigma}$; the non-zero transverse component can be determined from the condition $\mathrm{tr}(\mathrm{dev}\boldsymbol{\sigma}) = 0$. The norm of the stress deviator equals in this notation $\|\mathrm{dev}\boldsymbol{\sigma}\| = \sqrt{\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{P}\boldsymbol{\sigma}}\,$.

For isotropic elasticity, transformation of the constitutive tensor (3.26) according to Eqs. (3.27) and (3.150) leads to the matrix representation

$$\mathbf{C}^{\mathrm{e}} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{bmatrix} \tag{3.151}$$

and the elastic transverse strain increment is expressed as

$$\varepsilon^{\mathrm{e}}_{\perp} = -\frac{\nu}{1-\nu}\left(\varepsilon^{\mathrm{e}}_{11} + \varepsilon^{\mathrm{e}}_{22}\right) = -\frac{\nu}{E}\left(\sigma_{11} + \sigma_{22}\right).$$

The plastic transverse strain increment, due to incompressibility, must be

$$\varepsilon^{\mathrm{p}}_{\perp} = -(\varepsilon^{\mathrm{p}}_{11} + \varepsilon^{\mathrm{p}}_{22}).$$

Transforming Eqs. (3.99)–(3.104) for $\alpha = 1$ and denoting $\boldsymbol{\varsigma} = \boldsymbol{\sigma} - \boldsymbol{\alpha}$ (i.e. $\mathbf{s} \equiv \operatorname{dev}\boldsymbol{\varsigma} = \mathbf{P}\boldsymbol{\varsigma}$) and $\boldsymbol{\varsigma}_{\mathrm{trial}} = \boldsymbol{\varsigma}_n + \mathbf{C}^{\mathrm{e}}\Delta\boldsymbol{\varepsilon}$, we come at the following relations,

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \mathbf{C}^{\mathrm{e}}(\Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}^{\mathrm{p}}), \tag{3.152a}$$

$$\bar{\sigma}_{n+1} = \sqrt{\tfrac{3}{2}\boldsymbol{\varsigma}^{\mathrm{T}}_{n+1}\mathbf{P}\boldsymbol{\varsigma}_{n+1}}, \tag{3.152b}$$

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n + \frac{H'\Delta\bar{e}^{\mathrm{p}}}{\bar{\sigma}_{n+1}}\boldsymbol{\varsigma}_{n+1}, \tag{3.152c}$$

$$\Delta\boldsymbol{\varepsilon}^{\mathrm{p}} = \sqrt{\frac{3}{2}}\Delta\bar{e}^{\mathrm{p}}\,\frac{\mathbf{P}\boldsymbol{\varsigma}_{n+1}}{\|\mathbf{P}\boldsymbol{\varsigma}_{n+1}\|} = \lambda\,\mathbf{P}\boldsymbol{\varsigma}_{n+1}, \tag{3.152d}$$

where

$$\lambda = \frac{3\Delta\bar{e}^{\mathrm{p}}}{2\bar{\sigma}_{n+1}}. \tag{3.153}$$

Combined, they lead to the relation

$$\mathbf{Z}\,\boldsymbol{\varsigma}_{n+1} = \boldsymbol{\varsigma}_{\mathrm{trial}} \tag{3.154}$$

with

$$\mathbf{Z} = \vartheta\mathbf{I} + \lambda\mathbf{C}^{\mathrm{e}}\mathbf{P}, \qquad \vartheta = 1 + \tfrac{2}{3}\lambda H'_{n+1}. \tag{3.155}$$

Equation (3.154) indicates that in the reduced-dimension plane-stress space the relative stresses $\boldsymbol{\varsigma}_{n+1}$ and $\boldsymbol{\varsigma}_{\mathrm{trial}}$ are not proportional. Nor are their deviators, as it would be in the 3-D space, cf. Eq. (3.108). This means that the computation of the plastic corrector (i.e. mapping of $\boldsymbol{\varsigma}_{\mathrm{trial}}$ onto the plastic flow surface) does not conform to a radial-return scheme. However, Eq. (3.154) combined with the consistency conditions (3.111)–(3.113), form a set of nonlinear equations that can be iteratively solved for unknown ς_{n+1} , $\Delta\bar{e}^{\mathrm{p}}$ (or ς_{n+1} , λ).

It appears that, at least for linear kinematic hardening ($H' = \text{const}$), it is much more convenient to formulate the solution scheme in terms of the unknown λ, rather than $\Delta\bar{e}^{\mathrm{p}}$ (as it was done in the 3-D formulation). Differentiating Eq. (3.154),

$$\mathbf{Z}\,\frac{\mathrm{d}\varsigma_{n+1}}{\mathrm{d}\lambda} = -\left(\tfrac{2}{3}H'\mathbf{I} + \mathbf{C}^{\mathrm{e}}\mathbf{P}\right)\varsigma_{n+1}\,, \tag{3.156}$$

and realizing that, cf. Eqs. (3.152b), (3.153),

$$\frac{\mathrm{d}\Delta\bar{e}^{\mathrm{p}}}{\mathrm{d}\lambda} = \frac{2}{3}\left(\bar{\sigma}_{n+1} + \lambda\,\frac{\mathrm{d}\bar{\sigma}_{n+1}}{\mathrm{d}\lambda}\right), \qquad \frac{\mathrm{d}\bar{\sigma}_{n+1}}{\mathrm{d}\lambda} = \frac{3}{2\bar{\sigma}_{n+1}}\,\varsigma^{\mathrm{T}}_{n+1}\mathbf{P}\,\frac{\mathrm{d}\varsigma_{n+1}}{\mathrm{d}\lambda}\,, \tag{3.157}$$

we can formulate the following iterative scheme to find λ,

$$\begin{tikzcd} \arrow[r] & \delta\lambda = -\dfrac{\tilde{f}^{(i)}}{\tilde{f}^{(i)}_{,\lambda}} \arrow[r] & \lambda^{(i+1)} = \lambda^{(i)} + \delta\lambda \arrow[ld, bend left] \\ & i := i+1 \arrow[lu, bend left] & \end{tikzcd} \tag{3.158}$$

in which

$$\begin{aligned} \tilde{f}_{,\lambda} &= \frac{\partial\tilde{f}}{\partial\bar{\sigma}_{n+1}}\,\frac{\mathrm{d}\bar{\sigma}_{n+1}}{\mathrm{d}\lambda} + \left(\frac{\partial\tilde{f}}{\partial\bar{e}^{\mathrm{p}}_{n+1}} + \frac{\partial\tilde{f}}{\partial\Delta\bar{e}^{\mathrm{p}}}\right)\frac{\mathrm{d}\Delta\bar{e}^{\mathrm{p}}}{\mathrm{d}\lambda} = \vartheta_1\,\frac{\mathrm{d}\bar{\sigma}_{n+1}}{\mathrm{d}\lambda} + \vartheta_2\,, && (3.159)\\ \vartheta_1 &= \frac{\partial\tilde{f}}{\partial\bar{\sigma}_{n+1}} + \frac{2}{3}\,\lambda\left(\frac{\partial\tilde{f}}{\partial\bar{e}^{\mathrm{p}}_{n+1}} + \frac{\partial\tilde{f}}{\partial\Delta\bar{e}^{\mathrm{p}}}\right), && (3.160)\\ \vartheta_2 &= \frac{2}{3}\,\bar{\sigma}_{n+1}\left(\frac{\partial\tilde{f}}{\partial\bar{e}^{\mathrm{p}}_{n+1}} + \frac{\partial\tilde{f}}{\partial\Delta\bar{e}^{\mathrm{p}}}\right). && (3.161) \end{aligned}$$

A compact step-by-step algorithm of incremental constitutive computations for small-deformation isotropic plane-stress elasto-plasticity is displayed in Table 3.8. Additionally, Tables 3.9 and 3.10 display the iterative routine to compute, for different elasto-plastic and elasto-viscoplastic models, the consistency parameter λ and relative stress ς needed in Step 4 in Table 3.8.

To avoid frequent inverting of the matrix $\mathbf{Z}$, it is advisable, cf. [116], to perform the iteration loop in a stress-space coordinates rotated by $\pi/4$, i.e. to define

$$\tilde{\varsigma} = \mathbf{Q}^{\mathrm{T}}\varsigma, \quad \tilde{\varsigma}_{\mathrm{trial}} = \mathbf{Q}^{\mathrm{T}}\varsigma_{\mathrm{trial}}, \quad \tilde{\mathbf{P}} = \mathbf{Q}^{\mathrm{T}}\mathbf{P}\mathbf{Q}, \quad \tilde{\mathbf{C}}^{\mathrm{e}} = \mathbf{Q}^{\mathrm{T}}\mathbf{C}^{\mathrm{e}}\mathbf{Q}, \quad \tilde{\mathbf{I}} = \mathbf{I},$$

Table 3.8. Solution algorithm for small-deformation plane stress elasto-plasticity (for the sake of legibility, subscripts $(n{+}1)$ have been omitted)

```
Given
    state parameters[a] $\mathbf{z}_n = \{\boldsymbol{\sigma}_n\,, \boldsymbol{\alpha}_n\,, \bar{e}^{\mathrm{p}}_n\}$ at $t_n$ ,
    strain increment $\Delta\boldsymbol{\varepsilon}$
compute
    state parameters (including stress) $\mathbf{z} = \{\boldsymbol{\sigma}\,, \boldsymbol{\alpha}\,, \bar{e}^{\mathrm{p}}\}$ at $t_{n+1}$ ,
    transverse strain increment $\Delta\boldsymbol{\varepsilon}_{\perp}$
using the following scheme:
1. $\boldsymbol{\sigma}_{\mathrm{trial}} = \boldsymbol{\sigma}_n + \mathbf{C}^{\mathrm{e}}\Delta\boldsymbol{\varepsilon}$      – trial stress
2. $\boldsymbol{\varsigma}_{\mathrm{trial}} = \boldsymbol{\sigma}_{\mathrm{trial}} - \boldsymbol{\alpha}_n$      – trial relative stress
3. $\bar{\sigma}_{\mathrm{trial}} = \sqrt{\frac{3}{2}\,\boldsymbol{\varsigma}^{\mathrm{T}}_{\mathrm{trial}}\mathbf{P}\boldsymbol{\varsigma}_{\mathrm{trial}}}$      – trial equivalent stress
if $\bar{\sigma}_{\mathrm{trial}} - \kappa(\bar{e}^{\mathrm{p}}_n) > 0$ (plastic flow) then[b]
    4. $\lambda, \boldsymbol{\varsigma}, \bar{\sigma} \longleftarrow$ from Table 3.9
    5. $\Delta\boldsymbol{\varepsilon}^{\mathrm{p}} = \lambda\mathbf{P}\boldsymbol{\varsigma}$,   $\Delta\boldsymbol{\varepsilon}^{\mathrm{e}} = \Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}^{\mathrm{p}}$      – plastic & elastic strain increment
    6. $\Delta\varepsilon_{\perp} = -\nu(\Delta\varepsilon^{\mathrm{e}}_{11} + \Delta\varepsilon^{\mathrm{e}}_{22})$
              $- \frac{1}{2}(\Delta\varepsilon^{\mathrm{p}}_{11} + \Delta\varepsilon^{\mathrm{p}}_{22})$      – transverse strain increment
    7. $\boldsymbol{\alpha} = \boldsymbol{\alpha}_n + \frac{2}{3}\lambda H'\boldsymbol{\varsigma}$      } update state parameters
       $\boldsymbol{\sigma} = \boldsymbol{\varsigma} + \boldsymbol{\alpha}$
       $\bar{e}^{\mathrm{p}} = \bar{e}^{\mathrm{p}}_n + \frac{2}{3}\lambda\bar{\sigma}$
else (purely elastic deformation)
    8. $\boldsymbol{\sigma} = \boldsymbol{\sigma}_{\mathrm{trial}}$ ,   $\boldsymbol{\alpha} = \boldsymbol{\alpha}_n$ ,   $\bar{e}^{\mathrm{p}} = \bar{e}^{\mathrm{p}}_n$      – update state parameters
end if
```

[a] if $H \equiv$ const (no kinematic hardening) then $\boldsymbol{\alpha} \equiv \mathbf{0}$ is not a state parameter and all relevant computations are skipped

[b] for the power-law strain/strain-rate hardening viscoplasticity this condition is always fulfilled, except for $\bar{\sigma}_{\mathrm{trial}} = 0$

Table 3.9. Solution algorithm for the consistency parameter λ and relative stress $\boldsymbol{\varsigma}$ for the small-deformation plane-stress formulation, cf. Eq. (3.158)

```
1. $\lambda = 0$
2. $\vartheta = 1 + \frac{2}{3}\lambda H'_{n+1}\,, \qquad \mathbf{Z} = \vartheta\mathbf{I} + \lambda\mathbf{C}^{\mathrm{e}}\mathbf{P}$
3. $\boldsymbol{\varsigma} = \mathbf{Z}^{-1}\boldsymbol{\varsigma}_{\mathrm{trial}}\,, \qquad \bar{\sigma} = \sqrt{\frac{3}{2}\boldsymbol{\varsigma}^{\mathrm{T}}\mathbf{P}\boldsymbol{\varsigma}}$
4. $\Delta\bar{e}^{\mathrm{p}} = \frac{2}{3}\lambda\bar{\sigma}, \qquad \bar{e}^{\mathrm{p}} = \bar{e}^{\mathrm{p}}_{n} + \Delta\bar{e}^{\mathrm{p}}$
5. $\boldsymbol{\varsigma}_{,\lambda} = -\mathbf{Z}^{-1}\left(\frac{2}{3}H'\mathbf{I} + \mathbf{C}^{\mathrm{e}}\mathbf{P}\right)\boldsymbol{\varsigma}\,, \qquad \bar{\sigma}_{,\lambda} = \frac{3}{2\bar{\sigma}}\boldsymbol{\varsigma}^{\mathrm{T}}\mathbf{P}\,\boldsymbol{\varsigma}_{,\lambda}\,,$
6. $\tilde{f},\ \tilde{f}_{,\lambda} \quad \longleftarrow \quad$ from Table 3.10
7. $\delta\lambda = -\tilde{f}^{-1}_{,\lambda}\,\tilde{f}, \qquad \lambda = \lambda + \delta\lambda$
8. if $\|\delta\lambda\| > \mathrm{tol}$ then go to step 2; else exit
```

Table 3.10. Computation of $\tilde{f}$ and $\tilde{f}_{,\lambda}$ for different elasto-plastic and elasto-viscoplastic models

```
if (rate-independent plasticity) cf. Eq. (3.112),
    1a. $\tilde{f} = \bar{\sigma} - \kappa(\bar{e}^{\mathrm{p}})$
    2a. $\vartheta_1 = 1 - \frac{2}{3}\lambda\kappa', \qquad \vartheta_2 = -\frac{2}{3}\bar{\sigma}\kappa'$
else if (overstress viscoplasticity) cf. Eqs. (3.113), (3.119),
    1b. $g = \frac{1}{\mu}\left(\frac{\bar{\sigma}-\kappa(\bar{e}^{\mathrm{p}})}{\kappa(\bar{e}^{\mathrm{p}})}\right)^{m}, \qquad \tilde{f} = \Delta t\, g - \Delta\bar{e}^{\mathrm{p}}$
    2b. $g_{,\bar{\sigma}} = \frac{gm}{\bar{\sigma}-\kappa(\bar{e}^{\mathrm{p}})}\,, \qquad g_{,\bar{e}^{\mathrm{p}}} = -g_{,\bar{\sigma}}\,\frac{\bar{\sigma}\kappa'(\bar{e}^{\mathrm{p}})}{\kappa(\bar{e}^{\mathrm{p}})}$
    3b. $\vartheta_1 = \Delta t\, g_{,\bar{\sigma}} - \frac{2}{3}\lambda(1 - \Delta t\, g_{,\bar{e}^{\mathrm{p}}}), \qquad \vartheta_2 = -\frac{2}{3}\bar{\sigma}(1 - \Delta t\, g_{,\bar{e}^{\mathrm{p}}})$
else if (power-law hardening viscoplasticity) cf. Eqs. (3.113), (3.120),
    1c. $g_1 = E\varepsilon_0\left(1 + \frac{\bar{e}^{\mathrm{p}}}{\varepsilon_0}\right)^{\frac{1}{n}}, \qquad g = \dot{\varepsilon}_0\left(\frac{\bar{\sigma}}{g_1}\right)^{\frac{1}{m}}, \qquad \tilde{f} = \Delta t\, g - \Delta\bar{e}^{\mathrm{p}}$
    2c. $g_{,\bar{\sigma}} = \frac{g}{m\bar{\sigma}}\,, \qquad g_{,\bar{e}^{\mathrm{p}}} = -\frac{g}{mn(\bar{e}^{\mathrm{p}}+\varepsilon_0)}$
    3c. $\vartheta_1 = \Delta t\, g_{,\bar{\sigma}} - \frac{2}{3}\lambda(1 - \Delta t\, g_{,\bar{e}^{\mathrm{p}}}), \qquad \vartheta_2 = -\frac{2}{3}\bar{\sigma}(1 - \Delta t\, g_{,\bar{e}^{\mathrm{p}}})$
end if
4. $\tilde{f}_{,\lambda} = \vartheta_1\,\bar{\sigma}_{,\lambda} + \vartheta_2$
```

with

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The matrices $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{C}}^{\mathrm{e}}$ appear to be diagonal,

$$[\tilde{\mathrm{P}}_{ij}] = \operatorname{diag}\left[\, \tfrac{1}{3}\, ,\, 1\, ,\, 1 \right], \qquad [\tilde{\mathrm{C}}^{\mathrm{e}}_{ij}] = \operatorname{diag}\left[\, \frac{E}{1-\nu}\, ,\, \frac{E}{1+\nu}\, ,\, \frac{E}{1+\nu} \,\right],$$

which results in a diagonal, easy to invert, rotated matrix $\tilde{\mathbf{Z}}$. Performing the computations in Table 3.9 in these rotated coordinates, with $\tilde{\boldsymbol{\varsigma}}_{\mathrm{trial}} = \mathbf{Q}^{\mathrm{T}}\boldsymbol{\varsigma}_{\mathrm{trial}}$ as input and with the output transformed to $\boldsymbol{\varsigma} = \mathbf{Q}\tilde{\boldsymbol{\varsigma}}$, makes the algorithm more efficient.

The algorithmic tangent operator consistent with the time integration scheme, cf. Eq. (3.94), can be expressed in the following (3×3) matrix form

$$\mathbf{C} = \frac{\mathrm{d}\boldsymbol{\sigma}_{n+1}}{\mathrm{d}\boldsymbol{\varepsilon}_{n+1}} = \left[\mathbf{C}^{\mathrm{e}-1} + \tfrac{\lambda}{\vartheta}\left(\mathbf{P} + \tfrac{1}{\gamma_1}\left(\mathbf{P}\boldsymbol{\varsigma}_{n+1}\right)\left(\mathbf{P}\boldsymbol{\varsigma}_{n+1}\right)^{\mathrm{T}}\right)\right]^{-1} \tag{3.162}$$

(see Appendix A.3 for detailed derivation and explanation of the symbol γ_1).

3.4. Space- and time-discretized formulation of nonlinear statics

3.4.1. Spatial approximation of the displacement field

Solution of the variational problem (3.6) will be sought for in an approximate form with the use of the Ritz method. In particular, the finite element method (FEM) formalism will be utilized, see e.g. [8, 49, 141]. Consider a subspace of kinematically admissible fields of displacements

$$u_i(X_j,t) = \phi_{i\alpha}(X_j)\,\mathsf{q}_\alpha(t), \qquad i,j = 1,2,3, \quad \alpha = 1,\ldots,N_{np}\,, \tag{3.163}$$

where $\phi_{i\alpha}$ are prescribed shape functions defined for $X_j \in \Omega^r$, corresponding to N_{np} parameters q_α. The latter are associated with N_n material points called nodes ($N_{np} = N_n N_p$, N_p is the number of parameters per node), so that each index value $\alpha \in 1,\ldots,N_{np}$ can be expressed by a pair of index values (a, A), $a \in [1,\ldots,N_p]$, $A \in [1,\ldots,N_n]$,

$$\mathsf{q}_\alpha = \mathsf{q}_{aA}\,, \qquad \phi_{i\alpha} = \phi_{iaA}\,.$$

In the following derivations, the parameters at a particular node will be identified with the displacement components at this node (i.e. $N_p = 3$ in the general case, while $N_p < 3$ in reduced-dimension formulations) and a displacement field component u_i will be assumed to depend only on those nodal parameters $\mathsf{q}_\alpha = \mathsf{q}_{aA}$ for which $a = i$ (continuum element formulation). Assuming different displacement field components to be approximated by the same the shape functions, we can write

$$\phi_{i\alpha} = \phi_{iaA} = \Phi_A \, \delta_{ia}$$

where δ_{ia} is the Kronecker symbol and Φ_A are the nodal shape functions. Equation (3.163) thus becomes

$$u_i(X_j, t) = \Phi_A(X_j)\, \mathsf{q}_{iA}(t), \qquad i, j = 1, \ldots, N_p\,, \quad A = 1, \ldots, N_n\,. \tag{3.164}$$

Clearly, $\Phi_A(X_j)$ must take the value 1 at the node A and 0 at every other node.

Variation of the kinematically admissible approximate displacement field δu_i at a given time instant t is associated with variation of the nodal parameter array $\delta \mathsf{q}_\alpha$ at this time instant. Thus, Eq. (3.6) (with the modification (3.11)) can be rewritten in the form

$$\mathsf{f}_\alpha^{\text{int}}\, \delta \mathsf{q}_\alpha = \mathsf{f}_\alpha^{\text{ext}}\, \delta \mathsf{q}_\alpha \tag{3.165}$$

where

$$\mathsf{f}_\alpha^{\text{ext}} = \int_{\Omega^r} \hat{f}_i \phi_{i\alpha}\, \mathrm{d}\Omega + \int_{\partial\Omega^r} \hat{t}_i \phi_{i\alpha}\, \mathrm{d}(\partial\Omega), \tag{3.166}$$

$$\mathsf{f}_\alpha^{\text{int}} = \int_{\Omega^r} \tilde{\sigma}_{ij} \tilde{\mathsf{B}}_{ij\alpha}\, \mathrm{d}\Omega \tag{3.167}$$

are the arrays of external (load-resultant) and internal (stress-resultant) nodal forces, and the geometric operator $\tilde{\mathsf{B}}_{ij\alpha}$ is defined as

$$\delta \tilde{\varepsilon}_{ij}^{\text{u}} = \tilde{\mathsf{B}}_{ij\alpha}\, \delta \mathsf{q}_\alpha \qquad \text{or} \qquad \delta \tilde{\varepsilon}_{ij} = \operatorname{sym}(\tilde{\mathsf{B}}_{ij\alpha})\, \delta \mathsf{q}_\alpha \tag{3.168}$$

for each specific tensor strain variation measure. In view of symmetry of $\tilde{\sigma}_{ij}$, the operator $\tilde{\mathsf{B}}_{ij\alpha}$ can be used equivalently with $\operatorname{sym}(\tilde{\mathsf{B}}_{ij\alpha})$ in Eq. (3.167). Expressing the deformation gradient as[1] $F_{ij} = \delta_{ij} + \phi_{i\alpha,j} \mathsf{q}_\alpha$, we can write, cf. Eq. (3.10),

$$\tilde{\mathsf{B}}_{ij\alpha} = \begin{cases} \mathcal{B}_{ij\alpha} = \phi_{i\alpha,j} & \text{for } \tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma},\ \delta\tilde{\boldsymbol{\varepsilon}} = \delta\boldsymbol{\varepsilon}, \\ \mathsf{B}_{ij\alpha} = F_{ki} \phi_{k\alpha,j} & \text{for } \tilde{\boldsymbol{\sigma}} = \boldsymbol{T},\ \delta\tilde{\boldsymbol{\varepsilon}} = \delta\boldsymbol{E}, \\ \mathsf{b}_{ij\alpha} = \phi_{i\alpha,k} F_{kj}^{-1} & \text{for } \tilde{\boldsymbol{\sigma}} = \boldsymbol{\tau},\ \delta\tilde{\boldsymbol{\varepsilon}} = \delta\boldsymbol{\epsilon}, \end{cases} \tag{3.169}$$

[1] The notation $(\cdot)_{,j}$ stands for spatial differentiation of $(\cdot)$ in the reference configuration $\mathcal{C}^r$.

i.e.,

$$\mathsf{f}^{\mathrm{int}}_{\alpha} = \begin{cases} \displaystyle\int_{\Omega^r} \sigma_{ij}\mathcal{B}_{ij\alpha}\,\mathrm{d}\Omega & \text{for small deformations,} \\ \displaystyle\int_{\Omega^r} T_{ij}\mathsf{B}_{ij\alpha}\,\mathrm{d}\Omega = \int_{\Omega^r} \tau_{ij}\mathsf{b}_{ij\alpha}\,\mathrm{d}\Omega & \text{for large deformations.} \end{cases} \tag{3.170}$$

Clearly, both $\tilde{\sigma}_{ij}$ and $\tilde{\mathsf{B}}_{ij\alpha}$ depend on q_{β} (only for small deformations the dependence $\mathcal{B}_{ij\alpha}(\mathsf{q}_{\beta})$ can be neglected), thus $\mathsf{f}^{\mathrm{int}}_{\alpha} = \mathsf{f}^{\mathrm{int}}_{\alpha}(\mathsf{q}_{\beta})$. The external forces $\mathsf{f}^{\mathrm{ext}}_{\alpha}$ may in general also depend on q_{β} (e.g. follower forces) but this case will not be considered in our formulation.

Since Eq. (3.165) is supposed to be fulfilled for an arbitrary displacement variation, the solution of the nonlinear static problem consists in solving the nonlinear algebraic system

$$\mathsf{f}^{\mathrm{int}}_{\alpha}(\mathsf{q}_{\beta}) = \mathsf{f}^{\mathrm{ext}}_{\alpha} \qquad \text{or} \qquad \mathbf{f}^{\mathrm{int}}(\mathbf{q}) = \mathbf{f}^{\mathrm{ext}} \tag{3.171}$$

with respect to $\mathsf{q}_{\beta}(t)$ at each t. This obviously requires substitution of the appropriate constitutive equations and geometric relations to determine the dependence $\tilde{\sigma}_{ij}(\mathsf{q}_{\beta})$ and $\tilde{\mathsf{B}}_{ij\alpha}(\mathsf{q}_{\beta})$ in $\mathsf{f}^{\mathrm{int}}_{\alpha}$. Together with these equations, Eq. (3.171) forms a system of nonlinear differential equations with respect to the array of unknown time functions $\mathsf{q}_{\beta}(t)$ and constitutes the spatially discretized problem of continuum statics.

3.4.2. Approximate time integration

If the constitutive model is path-independent (elasticity), Eq. (3.171) may be solved independently for each time instant t. Otherwise, the solution consists rather in its time integration with application of the rate-type constitutive equations (3.15).

Recalling considerations of Section 3.3, let us discretize the time axis $t \geq 0$ by introducing a monotonic series of time instants $t_0 = 0, t_1, t_2, \ldots$. It has been demonstrated in Section 3.3, for different constitutive models and for different stress/strain measures, that appropriate time integration of a constitutive equation allows to express stress $\tilde{\boldsymbol{\sigma}}_{n+1}$ as a function of deformation at t_{n+1} and t_n and of certain state fields at t_n , cf. Eq, (3.93). The geometric operator $\tilde{\mathsf{B}}_{ij\alpha}$ at t_{n+1} is a function of deformation at t_{n+1}. Thus, $\mathbf{f}^{\mathrm{int}}_{n+1}$ can be expressed as an array of functionals

$$\mathbf{f}^{\mathrm{int}}_{n+1} = \mathbf{f}^{\mathrm{int}}_{n+1}[\boldsymbol{F}_{n+1}(\boldsymbol{X}), \boldsymbol{F}_{n}(\boldsymbol{X}), \mathbf{z}_{n}(\boldsymbol{X})] \tag{3.172}$$

whose first two argument fields $\boldsymbol{F}_{n+1}$ and $\boldsymbol{F}_n$ explicitly depend on $\mathbf{q}_{n+1}$ and $\mathbf{q}_n$, respectively. Realizing that spatial integration in Eqs. (3.166)–(3.167) is performed in an approximate manner, we can represent the state fields $\boldsymbol{z}$ at any time instant by a discretized array $\mathbf{z}$ of their values at a number of certain material points. This allows to formulate the system (3.171) at $t = t_{n+1}$ in the form

$$\mathbf{f}^{\mathrm{int}}_{n+1}(\mathbf{q}_{n+1}\,,\,\mathbf{q}_n\,,\,\mathbf{z}_n) = \mathbf{f}^{\mathrm{ext}}_{n+1} \tag{3.173}$$

from which $\mathbf{q}_{n+1}$ can be determined, provided $\mathbf{q}_n$ and $\mathbf{z}_n$ are known. Proceeding with the solution from $t = t_0$ through the subsequent time instants and appropriate updating $\mathbf{z}_n$ from step to step,

$$\mathbf{z}_{n+1} = \mathbf{z}_{n+1}(\mathbf{q}_{n+1}\,,\,\mathbf{q}_n\,,\,\mathbf{z}_n), \tag{3.174}$$

allows to solve the transient problem in the entire time domain of interest.

Equation (3.173) is a system of nonlinear algebraic equations with respect to the array of unknown nodal parameters $\mathbf{q}_{n+1}$ and it constitutes the space- and time-discretized problem of continuum statics. Defining the array of residual forces,

$$\mathbf{f}^{\mathrm{res}}_{n+1}(\mathbf{q}_{n+1}\,,\,\mathbf{q}_n\,,\,\mathbf{z}_n) = \mathbf{f}^{\mathrm{ext}}_{n+1} - \mathbf{f}^{\mathrm{int}}_{n+1}(\mathbf{q}_{n+1}\,,\,\mathbf{q}_n\,,\,\mathbf{z}_n), \tag{3.175}$$

we realize that Eq. (3.173) is a particular form of the general formulation (2.25) with $\mathbf{f}^{\mathrm{res}}_{n+1}$ identified with $\mathbf{F}_{n+1}$ (in its path-dependent form (2.32a)), and with the set of discrete variables $\{\mathbf{q}_n\,,\,\mathbf{z}_n\}$ identified with $\mathbf{s}_n$. The discretized input $\mathbf{p}$ present in the general formulation is involved in various ways in particular forms of $\mathbf{f}^{\mathrm{res}}_{n+1}$ and its arguments. For the special case of elasticity, dependence of $\mathbf{f}^{\mathrm{int}}_{n+1}$ (and consequently $\mathbf{f}^{\mathrm{res}}_{n+1}$) on $\mathbf{q}_n$ and $\mathbf{z}_n$ is neglected and the general path-independent form (2.26) is realized instead.

3.4.3. Solution method

The system (3.173) can be solved iteratively with the use of the following Newton–Raphson scheme, cf. Eq. (2.35),

$$\begin{tikzcd} \arrow[r] & \mathbf{K}^{(i)}_{n+1}\,\delta\mathbf{q} = \mathbf{f}^{\mathrm{res}\,(i)}_{n+1} \arrow[r] & \mathbf{q}^{(i+1)}_{n+1} = \mathbf{q}^{(i)}_{n+1} + \delta\mathbf{q} \arrow[d, dash] \\ & i := i+1 \arrow[lu, dash] & \arrow[l] \end{tikzcd} \tag{3.176}$$

where i is the iteration counter, $\mathbf{q}^{(i)}_{n+1}$ is the i-th approximate of the solution $\mathbf{q}_{n+1}$ (the initial predictor is set as, e.g., $\mathbf{q}^{(0)}_{n+1} = \mathbf{q}_n$), and the loop is repeated until the corrector $\delta\boldsymbol{\mathsf{q}}$ gets sufficiently small. The latter is determined in each iteration as the solution of the algebraic equation systems with the right hand side vector $\boldsymbol{\mathsf{f}}^{\text{res}}_{n+1}$ and with the operator $\boldsymbol{\mathsf{K}}$, defined as

$$\mathsf{K}_{\alpha\beta\;n+1} = -\frac{\mathrm{d}\mathsf{f}^{\text{res}}_{\alpha\;n+1}}{\mathrm{d}\mathsf{q}_{\beta\;n+1}} \qquad \text{or} \qquad \boldsymbol{\mathsf{K}}_{n+1} = -\frac{\mathrm{d}\boldsymbol{\mathsf{f}}^{\text{res}}_{n+1}}{\mathrm{d}\boldsymbol{\mathsf{q}}_{n+1}}\,, \tag{3.177}$$

called the algorithmic tangent stiffness matrix of the system. In the considered case of deformation-independent external loads, the matrix can be expressed as

$$\mathsf{K}_{\alpha\beta\;n+1} = \frac{\mathrm{d}\mathsf{f}^{\text{int}}_{\alpha\;n+1}}{\mathrm{d}\mathsf{q}_{\beta\;n+1}} \qquad \text{or} \qquad \boldsymbol{\mathsf{K}}_{n+1} = \frac{\mathrm{d}\boldsymbol{\mathsf{f}}^{\text{int}}_{n+1}}{\mathrm{d}\boldsymbol{\mathsf{q}}_{n+1}}\,. \tag{3.178}$$

Substituting Eqs. (3.169)–(3.170) to (3.178), we come at

$$\mathsf{K}_{\alpha\beta} = \int_{\Omega^r} \frac{\mathrm{d}\sigma_{ij}}{\mathrm{d}\mathsf{q}_\beta}\,\mathcal{B}_{ij\alpha}\,\mathrm{d}\Omega \tag{3.179a}$$

for small deformations, and

$$\begin{aligned} \mathsf{K}_{\alpha\beta} &= \int_{\Omega^r} \left(\frac{\mathrm{d}T_{ij}}{\mathrm{d}\mathsf{q}_\beta}\,\mathsf{B}_{ij\alpha} + T_{ij}\,\frac{\mathrm{d}\mathsf{B}_{ij\alpha}}{\mathrm{d}\mathsf{q}_\beta} \right) \mathrm{d}\Omega \\ &= \int_{\Omega^r} \left(\frac{\mathrm{d}\tau_{ij}}{\mathrm{d}\mathsf{q}_\beta}\,\mathsf{b}_{ij\alpha} + \tau_{ij}\,\frac{\mathrm{d}\mathsf{b}_{ij\alpha}}{\mathrm{d}\mathsf{q}_\beta} \right) \mathrm{d}\Omega \end{aligned} \tag{3.179b}$$

for large deformations. In view of

$$\frac{\mathrm{d}F_{ij}}{\mathrm{d}\mathsf{q}_\beta} = \phi_{i\beta,j}\,, \qquad\qquad \frac{\mathrm{d}F^{-1}_{ij}}{\mathrm{d}\mathsf{q}_\beta} = -F^{-1}_{ik}\,\phi_{k\beta,l}\,F^{-1}_{lj}\,,$$

we have

$$\frac{\mathrm{d}\mathsf{B}_{ij\alpha}}{\mathrm{d}\mathsf{q}_\beta} = \phi_{k\alpha,i}\phi_{k\alpha,j}\,, \qquad \frac{\mathrm{d}\mathsf{b}_{ij\alpha}}{\mathrm{d}\mathsf{q}_\beta} = -\mathsf{b}_{ik\alpha}\mathsf{b}_{kj\beta}\,. \tag{3.180}$$

Recalling definitions of algorithmic stiffness tensors (3.94)–(3.97) and employing their symmetry with respect to the last two indices, we get

$$\mathrm{d}\sigma_{ij} = \mathcal{C}_{ijkl}\,\frac{\mathrm{d}\varepsilon_{kl}}{\mathrm{d}\mathsf{q}_\beta}\,\mathrm{d}\mathsf{q}_\beta = \mathcal{C}_{ijkl}\,\mathcal{B}_{kl\beta}\,\mathrm{d}\mathsf{q}_\beta, \tag{3.181a}$$

$$\mathrm{d}T_{ij} = \mathfrak{C}_{ijkl}\,\frac{\mathrm{d}E_{kl}}{\mathrm{d}\mathsf{q}_\beta}\,\mathrm{d}\mathsf{q}_\beta = \mathfrak{C}_{ijkl}\,\mathsf{B}_{kl\beta}\,\mathrm{d}\mathsf{q}_\beta, \tag{3.181b}$$

$$\begin{aligned} \mathrm{d}\tau_{ij} &= \left(\mathfrak{c}_{ijkl}\,\frac{\mathrm{d}\epsilon_{kl}}{\mathrm{d}\mathsf{q}_\beta} + \tau_{lj}\,\frac{\mathrm{d}F_{ik}}{\mathrm{d}\mathsf{q}_\beta}\,F^{-1}_{kl} + \tau_{ik}\,\frac{\mathrm{d}F_{jl}}{\mathrm{d}\mathsf{q}_\beta}\,F^{-1}_{lk} \right) \mathrm{d}\mathsf{q}_\beta \\ &= \left(\mathfrak{c}_{ijkl}\,\mathsf{b}_{kl\beta} + \tau_{lj}\,\mathsf{b}_{il\beta} + \tau_{ik}\,\mathsf{b}_{jk\beta} \right) \mathrm{d}\mathsf{q}_\beta, \end{aligned} \tag{3.181c}$$

and we can finally express the algorithmic tangent stiffness matrix as

$$\mathsf{K}_{\alpha\beta} = \int_{\Omega^r} \mathcal{B}_{ij\alpha}\mathcal{C}_{ijkl}\mathcal{B}_{kl\beta}\,\mathrm{d}\Omega \tag{3.182a}$$

for small deformations, and

$$\mathsf{K}_{\alpha\beta} = \int_{\Omega^r} (\mathsf{B}_{ij\alpha}\mathfrak{C}_{ijkl}\mathsf{B}_{kl\beta} + T_{ij}\phi_{k\alpha,j}\phi_{k\beta,i})\,\mathrm{d}\Omega, \tag{3.182b}$$

or, equivalently,

$$\mathsf{K}_{\alpha\beta} = \int_{\Omega^r} (\mathsf{b}_{ij\alpha}\mathfrak{c}_{ijkl}\mathsf{b}_{kl\beta} + \tau_{ij}\mathsf{b}_{kj\alpha}\mathsf{b}_{ki\beta})\,\mathrm{d}\Omega, \tag{3.182c}$$

for large deformations.

The above solution procedure is efficient in the sense that it guarantees quadratic convergence of the iteration loop (3.176). However, its weakness lies in the necessity of decomposition of the usually very large matrix $\mathbf{K}^{(i)}_{n+1}$ in each iteration, which is a computationally expensive routine. A frequently encountered modification to the solution procedure, which may (but does not have to) improve its efficiency, consists in the use of an unchanged matrix (e.g. $\mathbf{K}^{(0)}_{n+1} = \mathbf{K}_n$) in each iteration run. This slows down convergence, increasing the necessary number of iterations, but this drawback is rewarded with low computational cost of each of them (except for the first one, only a back-substitution is required to solve the system of equations, after the new residual vector is created). However, it will be shown in Chapter 4 that the use of the most current, exactly derived algorithmic tangent stiffness matrix is essential from the point of view of the sensitivity analysis. Thus, only the solution algorithms that conform to the scheme (3.176) will be considered in this study.

Another modification to the procedure (3.176) is encountered in cases of a nonsymmetric stiffness matrix. Note that, in view of Eqs. (3.182), $\mathsf{K}_{\alpha\beta}$ is symmetric if (and only if)[1] the corresponding algorithmic constitutive stiffness tensor $\mathcal{C}_{ijkl}$, $\mathfrak{C}_{ijkl}$, or $\mathfrak{c}_{ijkl}$ exhibits symmetry with respect to the pairs of indices $\{ij\} \leftrightarrow \{kl\}$. It has been shown that in some cases, cf. e.g. Eqs. (3.144)–(3.145), this condition may not be fulfilled. Since equation systems' solvers assuming symmetric coefficient matrices are more efficient than others, and since the nonsymmetric terms in $\mathfrak{c}_{ijkl}$ are of low order of magnitude, it is reasonable to decompose only the symmetric part of $\mathsf{K}_{\alpha\beta}$ in (3.176). This may again slightly slow down convergence but does not affect the final solution. It will be shown in Chapter 4 how this issue is dealt with in the sensitivity analysis.

[1] The obvious symmetry of T_{ij} and τ_{ij} is not mentioned.

3.4.4. Enhanced formulations of FE strain approximation

The approximation of the displacement field (3.163) and the corresponding approximation of strain (3.168)–(3.169), associated with the simplest possible (and thus most frequently used) bi- or tri-linear isoparametric element shape functions $\phi_{i\alpha}$, cf. e.g. [8, 49, 141], proves to perform poorly in presence of incompressible or nearly incompressible deformations of modelled continuum. In particular, the isochoric plastic flow, when modelled with such spatial discretization, leads to solutions suffering from errors due to volumetric locking. Various modifications have been proposed to overcome the difficulties. Here, the enhanced assumed strain method [88], being a finite-deformation generalization of the selectively reduced integration (B-bar) approach [48], will be applied.

The method consists in splitting of the FE interpolated deformation gradient $F_{ij} = \delta_{ij} + \phi_{i\alpha,j}\,\mathsf{q}_\alpha$ into multiplicative superposition of the volumetric and isochoric components (3.33)–(3.34) and modify it by replacing the volumetric part $F^{\mathrm{v}}_{ij} = J^{\frac{1}{3}}\delta_{ij}$ with its equivalent counterpart $\tilde{F}^{\mathrm{v}}_{ij} = \tilde{J}^{\frac{1}{3}}\delta_{ij}$ averaged over the volume of the corresponding finite element. The latter can be determined from the relations

$$\tilde{J} = \det[\tilde{F}_{ij}]\,, \qquad \tilde{F}_{ij} = \delta_{ij} + \tilde{\phi}_{i\alpha,j}\,\mathsf{q}_\alpha\,,$$

where $\tilde{\phi}_{i\alpha,j}$ are averaged values of shape function gradients $\phi_{i\alpha,j}$ over the element volume. Thus, the modified deformation gradient, used in the kinematic relations, has in the 3D case the form[1]

$$\bar{F}_{ij} = \left(\frac{\tilde{J}}{J}\right)^{\frac{1}{3}} F_{ij}\,. \tag{3.183}$$

It is easy to verify that $\det[\bar{F}_{ij}] \equiv \tilde{J}$.

Such redefinition of the deformation gradient affects values of strain variations $\delta\tilde{\varepsilon}_{ij}$ and stresses $\tilde{\sigma}_{ij}$ present in the formula for internal forces $\mathsf{f}^{\mathrm{int}}_\alpha$ (3.167), which now also have to be replaced with their modified, ‘barred’ counterparts. Naturally, the forms of the tangent matrix $\mathsf{K}_{\alpha\beta}$ will also change.

Employing the differential formula

$$\mathrm{d}J = JF^{-1}_{ij}\,\mathrm{d}F_{ji}$$

[1] The notation $\bar{F}_{ij}$, traditionally used for the modified deformation gradient, should not be confused with a similar (and also traditional) notation used in Eqs. (3.33)–(3.34) for the isochoric part of F_{ij}; nb. the modification (3.183) affects only the volumetric part F^{v}_{ij} while the isochoric part remains unchanged.

(which holds as well for the ‘tilded’ measures), we come at

$$\begin{aligned}
\mathrm{d}\bar{F}_{ij} &= \left(\frac{\tilde{J}}{J}\right)^{\frac{1}{3}} \left[\mathrm{d}F_{ij} + \tfrac{1}{3}(\tilde{F}^{-1}_{kl}\,\mathrm{d}\tilde{F}_{lk} - F^{-1}_{kl}\,\mathrm{d}F_{lk})F_{ij}\right] \\
&= \left(\frac{\tilde{J}}{J}\right)^{\frac{1}{3}} (\phi_{i\alpha,j} + \mathsf{g}_\alpha F_{ij})\,\mathrm{d}\mathsf{q}_\alpha\,, \qquad (3.184)
\end{aligned}$$

where

$$\mathsf{g}_\alpha = \tfrac{1}{3}(\tilde{\mathsf{b}}_{ii\alpha} - \mathsf{b}_{ii\alpha})\,, \qquad \tilde{\mathsf{b}}_{ij\alpha} = \tilde{\phi}_{i\alpha,k}\tilde{F}^{-1}_{kj}\,, \qquad \mathsf{b}_{ij\alpha} = \phi_{i\alpha,k}F^{-1}_{kj} \qquad (3.185)$$

(cf. definition of $\mathsf{b}_{ij\alpha}$, (3.169)). In the general case of large deformations, the modified array of internal forces can be expressed as

$$\bar{\mathsf{f}}^{\mathrm{int}}_\alpha = \int_{\Omega^r} \bar{T}_{ij}\bar{\mathsf{B}}_{ij\alpha}\,\mathrm{d}\Omega = \int_{\Omega^r} \bar{\tau}_{ij}\bar{\mathsf{b}}_{ij\alpha}\,\mathrm{d}\Omega \qquad (3.186)$$

where $\bar{T}_{ij}$ and $\bar{\tau}_{ij}$ denote stresses computed from the constitutive equations for modified (‘barred’) deformation $\bar{F}_{ij}$, and the ‘barred’ geometric arrays are defined as, cf. Eqs. (3.168) and (3.10),

$$\begin{aligned}
\mathrm{d}\bar{E}^{\mathrm{u}}_{ij} &= \bar{F}_{ki}\,\mathrm{d}\bar{F}_{kj} = \bar{\mathsf{B}}_{ij\alpha}\,\mathrm{d}\mathsf{q}_\alpha\,, & \mathrm{d}\bar{E}_{ij} &= \mathrm{sym}(\mathrm{d}\bar{E}^{\mathrm{u}}_{ij})\,,\\
\mathrm{d}\bar{\epsilon}^{\mathrm{u}}_{ij} &= \mathrm{d}\bar{F}_{ik}\,\bar{F}^{-1}_{kj} = \bar{\mathsf{b}}_{ij\alpha}\,\mathrm{d}\mathsf{q}_\alpha\,, & \mathrm{d}\bar{\epsilon}_{ij} &= \mathrm{sym}(\mathrm{d}\bar{\epsilon}^{\mathrm{u}}_{ij})\,.
\end{aligned}$$

Denoting

$$\upsilon = \left(\frac{\tilde{J}}{J}\right)^{\frac{2}{3}}, \qquad (3.187)$$

we can express them as

$$\begin{aligned}
\bar{\mathsf{B}}_{ij\alpha} &= \upsilon F_{ki}\,(\phi_{k\alpha,j} + \mathsf{g}_\alpha F_{kj}) = \upsilon \mathsf{B}_{ij\alpha} + \mathsf{g}_\alpha \bar{F}_{ki}\bar{F}_{kj}\,, \qquad (3.188)\\
\bar{\mathsf{b}}_{ij\alpha} &= (\phi_{i\alpha,k} + \mathsf{g}_\alpha F_{ik})\,F^{-1}_{kj} = \mathsf{b}_{ij\alpha} + \mathsf{g}_\alpha\delta_{ij}\,. \qquad (3.189)
\end{aligned}$$

In the small deformation formulation, deformation gradients F_{ij} , $\tilde{F}_{ij}$ and $\bar{F}_{ij}$ are replaced with corresponding infinitesimal displacement gradients

$$u_{i,j} = \phi_{i\alpha,j}\,\mathsf{q}_\alpha\,, \qquad \tilde{u}_{i,j} = \tilde{\phi}_{i\alpha,j}\,\mathsf{q}_\alpha\,, \qquad \bar{u}_{i,j} = u_{i,j} + \tfrac{1}{3}(\tilde{u}_{k,k} - u_{k,k})\,\delta_{ij}\,,$$

and both the large-deformation formulations can be reduced to

$$\bar{\mathsf{f}}^{\mathrm{int}}_\alpha = \int_{\Omega^r} \bar{\sigma}_{ij}\bar{\mathcal{B}}_{ij\alpha}\,\mathrm{d}\Omega \qquad (3.190)$$

with

$$\mathrm{d}\bar{\varepsilon}^{\mathrm{u}}_{ij} = \mathrm{d}\bar{u}_{i,j} = \bar{\mathcal{B}}_{ij\alpha}\,\mathrm{d}\mathsf{q}_\alpha\,, \qquad \mathrm{d}\bar{\varepsilon}_{ij} = \mathrm{sym}(\mathrm{d}\bar{\varepsilon}^{\mathrm{u}}_{ij})\,, \tag{3.191}$$

and

$$\bar{\mathcal{B}}_{ij\alpha} = \mathcal{B}_{ij\alpha} + \mathsf{g}^{\ell}_{\alpha}\delta_{ij}\,, \qquad \mathsf{g}^{\ell}_{\alpha} = \tfrac{1}{3}(\tilde{\phi}_{i\alpha,i} - \phi_{i\alpha,i})\,. \tag{3.192}$$

To compute the modified algorithmic tangent matrix, let us first notice that

$$\mathrm{d}\upsilon = 2\upsilon\,\mathsf{g}_\alpha\,\mathrm{d}\mathsf{q}_\alpha\,, \qquad \mathrm{d}\mathsf{g}_\alpha = \tfrac{1}{3}\left(\tilde{\phi}_{i\alpha,j}\mathrm{d}\tilde{F}^{-1}_{ji} - \phi_{i\alpha,j}\mathrm{d}F^{-1}_{ji}\right) = -\mathsf{h}_{\alpha\beta}\,\mathrm{d}\mathsf{q}_\beta\,,$$

where

$$\mathsf{h}_{\alpha\beta} = \tfrac{1}{3}\left(\tilde{\mathsf{b}}_{ij\alpha}\tilde{\mathsf{b}}_{ji\beta} - \mathsf{b}_{ij\alpha}\mathsf{b}_{ji\beta}\right).$$

This allows to write

$$\begin{aligned}
\frac{\mathrm{d}\bar{\mathsf{B}}_{ij\alpha}}{\mathrm{d}\mathsf{q}_\beta} &= \upsilon\,\frac{\mathrm{d}\mathsf{B}_{ij\alpha}}{\mathrm{d}\mathsf{q}_\beta} + \frac{\mathrm{d}\upsilon}{\mathrm{d}\mathsf{q}_\beta}\,\mathsf{B}_{ij\alpha} + \mathsf{g}_\alpha\,2\bar{\mathsf{B}}_{ij\beta} + \frac{\mathrm{d}\mathsf{g}_\alpha}{\mathrm{d}\mathsf{q}_\beta}\,\bar{F}_{ki}\bar{F}_{kj} \\
&= \upsilon\phi_{k\alpha,j}\phi_{k\beta,i} + 2\mathsf{g}_\beta\bar{\mathsf{B}}_{ij\alpha} + 2\mathsf{g}_\alpha\bar{\mathsf{B}}_{ij\beta} - 2\mathsf{g}_\alpha\mathsf{g}_\beta\bar{F}_{ki}\bar{F}_{kj} - \mathsf{h}_{\alpha\beta}\bar{F}_{ki}\bar{F}_{kj}\,, \\
\frac{\mathrm{d}\bar{\mathsf{b}}_{ij\alpha}}{\mathrm{d}\mathsf{q}_\beta} &= \frac{\mathrm{d}\mathsf{b}_{ij\alpha}}{\mathrm{d}\mathsf{q}_\beta} + \frac{\mathrm{d}\mathsf{g}_\alpha}{\mathrm{d}\mathsf{q}_\beta}\,\delta_{ij} = -\mathsf{b}_{il\alpha}\mathsf{b}_{lj\beta} - \mathsf{h}_{\alpha\beta}\delta_{ij} \\
&= -\bar{\mathsf{b}}_{il\alpha}\bar{\mathsf{b}}_{lj\beta} + \mathsf{g}_\alpha\bar{\mathsf{b}}_{ij\beta} + \mathsf{g}_\beta\bar{\mathsf{b}}_{ij\alpha} - (\mathsf{g}_\alpha\mathsf{g}_\beta{+}\mathsf{h}_{\alpha\beta})\delta_{ij}\,.
\end{aligned}$$

In view of the fact that, cf. Eqs. (3.181),

$$\frac{\mathrm{d}\bar{T}_{ij}}{\mathrm{d}\mathsf{q}_\beta} = \bar{\mathfrak{C}}_{ijkl}\bar{\mathsf{B}}_{kl\beta}\,, \qquad \frac{\mathrm{d}\bar{\tau}_{ij}}{\mathrm{d}\mathsf{q}_\beta} = \bar{\mathfrak{c}}_{ijkl}\bar{\mathsf{b}}_{kl\beta} + \bar{\tau}_{kj}\bar{\mathsf{b}}_{ik\beta} + \bar{\tau}_{il}\bar{\mathsf{b}}_{jl\beta}$$

(where the tensors $\bar{\mathfrak{C}}_{ijkl}$ and $\bar{\mathfrak{c}}_{ijkl}$ are computed in the same way as $\mathfrak{C}_{ijkl}$ and $\mathfrak{c}_{ijkl}$, respectively, but for the 'barred' deformation state), we can finally write

$$\begin{aligned}
\bar{\mathsf{K}}_{\alpha\beta} &= \int_{\Omega^r}\left(\frac{\mathrm{d}\bar{T}_{ij}}{\mathrm{d}\mathsf{q}_\beta}\,\bar{\mathsf{B}}_{ij\alpha} + \bar{T}_{ij}\,\frac{\mathrm{d}\bar{\mathsf{B}}_{ij\alpha}}{\mathrm{d}\mathsf{q}_\beta}\right)\mathrm{d}\Omega \\
&= \int_{\Omega^r}\left(\bar{\mathsf{B}}_{ij\alpha}\bar{\mathfrak{C}}_{ijkl}\bar{\mathsf{B}}_{kl\beta} + \bar{T}_{ij}\bar{\mathsf{D}}_{ij\alpha\beta}\right)\mathrm{d}\Omega\,, && \text{(3.193a)} \\
\bar{\mathsf{D}}_{ij\alpha\beta} &= \upsilon\phi_{k\alpha,j}\phi_{k\beta,i} + 2\mathsf{g}_\beta\bar{\mathsf{B}}_{ij\alpha} + 2\mathsf{g}_\alpha\bar{\mathsf{B}}_{ij\beta} - (2\mathsf{g}_\alpha\mathsf{g}_\beta{+}\mathsf{h}_{\alpha\beta})\bar{F}_{ki}\bar{F}_{kj}\,, && \text{(3.193b)}
\end{aligned}$$

or

$$\begin{aligned}
\bar{\mathsf{K}}_{\alpha\beta} &= \int_{\Omega^r}\left(\frac{\mathrm{d}\bar{\tau}_{ij}}{\mathrm{d}\mathsf{q}_\beta}\,\bar{\mathsf{b}}_{ij\alpha} + \bar{\tau}_{ij}\,\frac{\mathrm{d}\bar{\mathsf{b}}_{ij\alpha}}{\mathrm{d}\mathsf{q}_\beta}\right)\mathrm{d}\Omega \\
&= \int_{\Omega^r}\left(\bar{\mathsf{b}}_{ij\alpha}\bar{\mathfrak{c}}_{ijkl}\bar{\mathsf{b}}_{kl\beta} + \bar{\tau}_{ij}\bar{\mathsf{d}}_{ij\alpha\beta}\right)\mathrm{d}\Omega\,, && \text{(3.194a)} \\
\bar{\mathsf{d}}_{ij\alpha\beta} &= \mathsf{b}_{kj\alpha}\mathsf{b}_{ki\beta} + 2\mathsf{g}_\beta\bar{\mathsf{b}}_{ij\alpha} + 2\mathsf{g}_\alpha\bar{\mathsf{b}}_{ij\beta} - (2\mathsf{g}_\alpha\mathsf{g}_\beta{+}\mathsf{h}_{\alpha\beta})\delta_{ij}\,. && \text{(3.194b)}
\end{aligned}$$

In the latter equations, symmetry o the stress tensors $\bar{T}_{ij}$ and $\bar{\tau}_{ij}$ has been utilized. For small deformations, the geometric array $\bar{\mathcal{B}}_{ij\alpha}$ does not depend on q_β and thus the above equations reduce to

$$\bar{\mathsf{K}}_{\alpha\beta} = \int_{\Omega^r} \frac{\mathrm{d}\bar{\sigma}_{ij}}{\mathrm{d}\mathsf{q}_\beta} \bar{\mathcal{B}}_{ij\alpha} \, \mathrm{d}\Omega = \int_{\Omega^r} \bar{\mathcal{B}}_{ij\alpha} \mathcal{C}_{ijkl} \bar{\mathcal{B}}_{kl\beta} \, \mathrm{d}\Omega \, . \tag{3.195}$$

Chapter 4

Sensitivity analysis of elasto-plastic static equilibrium problem

This chapter contains the main, original part of the thesis. In the beginning, the problem of sensitivity of the discrete primary problem presented in Section 3.4 is formulated, for both small- and large-deformation cases. Attention is focused on computational aspects of DSA, including, among others, the role of the consistent, algorithmic tangent operator. The problem of shape sensitivity is also addressed — it is demonstrated that there is actually no fundamental difference between the DSA formulation for shape and non-shape design parameters. Further, the incremental constitutive equations of elasticity and elasto-plasticity, cf. Section 3.3, are differentiated with respect to design-dependent input parameters in order to determine partial and total design derivatives of response state fields necessary in the global sensitivity formulation. Linearity of the sensitivity formulation a time step is underlined, even in presence of highly nonlinear plastic consistency equations in the constitutive model. Resulting formulations are presented in a form of closed-form algorithms, readily implementable in computer programs. A number of numerical examples illustrate the presented computational algorithms and inspire discussion.

4.1. General discrete formulation of nonlinear static response sensitivity

Let us start from recalling the space- and time-discretized formulation of nonlinear statics (3.173)–(3.175) at $t = t_{n+1}$,

$$\mathbf{f}^{\mathrm{res}}(\mathbf{q}\,,\,\mathbf{q}_n\,,\,\mathbf{z}_n) = \mathbf{0}\,, \qquad \mathbf{z} = \mathbf{z}(\mathbf{q}\,,\,\mathbf{q}_n\,,\,\mathbf{z}_n)\,, \tag{4.1}$$

in which, similarly as in all the following derivations in this section, indices $n+1$ are skipped for better legibility of the formulae.[1]

As already mentioned in Section 3.4.2, Eq. (4.1) is a particular form of the general discrete system (2.25) (in the path-dependent form (2.32a)) for $t = t_{n+1}$. The discretized input $\mathbf{p}$, present in the general formulation, is involved in various ways in particular forms of $\mathbf{f}^{\text{res}}_{n+1}$ and its arguments, and it is not explicitly displayed in equations of this chapter. Instead, the design parameters $\mathbf{h}$ is assumed to explicitly and/or implicitly affect all the above functions and their arguments, so that we can write

$$\mathbf{f}^{\text{res}}\left(\mathbf{q}(\mathbf{h})\,,\,\mathbf{q}_n(\mathbf{h})\,,\,\mathbf{z}_n(\mathbf{h})\,;\,\mathbf{h}\right) = \mathbf{0}\,, \qquad \mathbf{z} = \mathbf{z}\left(\mathbf{q}(\mathbf{h})\,,\,\mathbf{q}_n(\mathbf{h})\,,\,\mathbf{z}_n(\mathbf{h})\,;\,\mathbf{h}\right). \tag{4.2}$$

Recalling conclusions of Section 2.2.2, the discussion of the sensitivity formulation in nonlinear statics will be limited to DDM. General formulation in the path-dependent case, cf. Eq. (2.39), has, in the notation of Section 3.4, the form

$$\mathbf{K}\,\mathrm{d}_{\mathbf{h}}\mathbf{q} = \hat{\partial}_{\mathbf{h}}\mathbf{f}^{\text{res}} \tag{4.3}$$

where

$$\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\text{res}} = \mathrm{d}_{\mathbf{h}}\mathbf{f}^{\text{res}}|_{\text{frozen }\mathbf{q}} = \frac{\mathrm{d}\mathbf{f}^{\text{res}}}{\mathrm{d}\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \frac{\mathrm{d}\mathbf{f}^{\text{res}}}{\mathrm{d}\mathbf{z}_n}\,\partial_{\mathbf{h}}\mathbf{z}_n + \partial_{\mathbf{h}}\mathbf{f}^{\text{res}} \tag{4.4}$$

(since both the primary and sensitivity solutions at $t = t_n$ are assumed to be known and constitute an input to the problem written for $t = t_{n+1}$, the design derivatives of $\mathbf{q}_n$ and $\mathbf{z}_n$ will be considered 'explicit' and denoted by $\partial_{\mathbf{h}}\mathbf{q}_n$ and $\partial_{\mathbf{h}}\mathbf{z}_n$, respectively). Introducing a simplified notation $\mathrm{d}_{\mathbf{h}!b}f$ for a partial design derivative of a function $f(\ldots, b(\mathbf{h}), \ldots; \mathbf{h})$ taken at its argument b frozen,

$$\mathrm{d}_{\mathbf{h}!b}f = \mathrm{d}_{\mathbf{h}}f - \frac{\partial f}{\partial b}\,\mathrm{d}_{\mathbf{h}}b\,, \tag{4.5}$$

where f and b are quantities of any (scalar, vector, tensor, etc.) type, we can equivalently write

$$\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\text{res}} = \mathrm{d}_{\mathbf{h}!\mathbf{q}}\mathbf{f}^{\text{res}}\,. \tag{4.6}$$

The computation of the response sensitivity consists in determination of the discrete right-hand side vector $\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\text{res}}$ and solution of the system (4.3) with

[1] This omission may, however, cause some confusion as e.g. the notation $\mathbf{q}$, understood here as $\mathbf{q}_{n+1}$, has different meaning than it had in Chapter 2, where it denoted the entire time-domain array $\{\mathbf{q}_0\,,\,\mathbf{q}_1\,,\,\ldots\,,\,\mathbf{q}_k\,,\,\ldots\}$. Let us underline that the following derivations correspond to only a single time step, and the array $\mathbf{q}$ in its former meaning will never appear in this section's formulae.

respect to the unknown array $\mathrm{d}_{\mathbf{h}}\mathbf{q}$ with the use of the most recent stiffness matrix $\mathbf{K}^{(i)}$ decomposed in the equilibrium iterations (3.176). Then, design derivatives of the state parameters have to be updated and stored for the next time step computations,

$$\mathrm{d}_{\mathbf{h}}\mathbf{z} = \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}\mathbf{q}}\,\mathrm{d}_{\mathbf{h}}\mathbf{q} + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}\mathbf{z}_n}\,\partial_{\mathbf{h}}\mathbf{z}_n + \partial_{\mathbf{h}}\mathbf{z}\,, \tag{4.7}$$

Finally, the updated values of performance functionals related to $t = t_{n+1}$ can be determined from Eq. (2.36) or (2.46), depending on the particular form of $\mathbf{G}$. The latter issue will not be a subject of interest in this section — in view of the assumed explicit dependence of $\mathbf{G}$ on the response and design parameters, this is usually not a mathematically complicated task, while a wide variety of particular forms of $\mathbf{G}$ makes it difficult to present the issue in a compact and unified manner.

Table 4.1 presents the scheme of computations for both the primary equilibrium and the design sensitivity analysis at a typical time step $[t_n\,,\,t_{n+1}]$.

Since the state parameters $\mathbf{z}$ are constitutive state variables computed at particular material points, detailed update formulae for their sensitivities $\mathrm{d}_{\mathbf{h}}\mathbf{z}$ will be a subject of Section 4.2. In the following part of this section, attention will be focused on the methods of determination of the explicit derivatives of the residual forces, $\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{res}}$, i.e. the right-hand side array in the sensitivity system of equations (4.3). Since external loads $\hat{f}_i$ and $\hat{t}_i$ are assumed to be deformation-independent, it is easy to conclude that

$$\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{res}} = \partial_{\mathbf{h}}\mathbf{f}^{\mathrm{ext}} - \mathrm{d}_{\mathbf{h}!\mathbf{q}}\mathbf{f}^{\mathrm{int}}\,, \qquad \hat{\partial}_{\mathbf{h}}\mathsf{f}^{\mathrm{res}}_{\alpha} = \partial_{\mathbf{h}}\mathsf{f}^{\mathrm{ext}}_{\alpha} - \mathrm{d}_{\mathbf{h}!\mathbf{q}}\mathsf{f}^{\mathrm{int}}_{\alpha}\,. \tag{4.8}$$

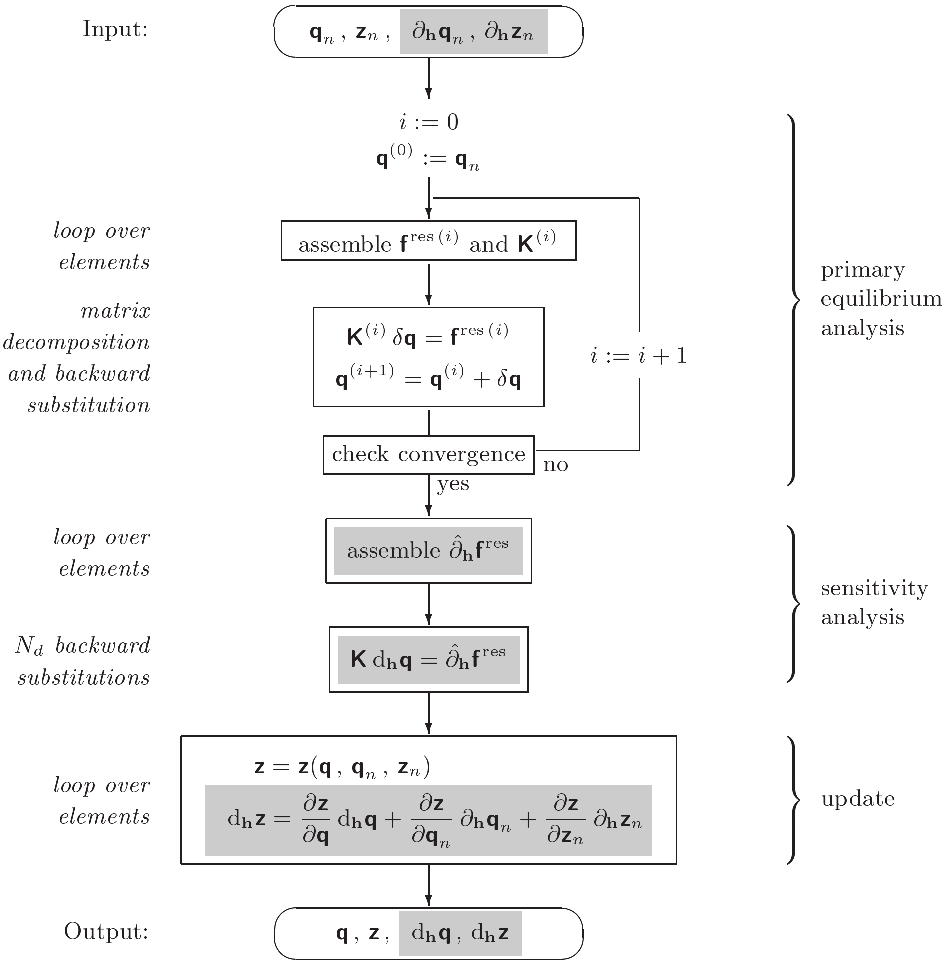
Two cases will be distinguished: (i) design-independent geometry and (ii) a more general case of design-dependent geometry (shape sensitivity). By geometry we mean the spatial positions $\boldsymbol{x}^0$ of material points in the initial, undeformed configuration of the body, identified here with the reference configuration on which all the state fields are defined, i.e. $\mathcal{C}^0 \equiv \mathcal{C}^r$ and $\boldsymbol{x}^0 \equiv \boldsymbol{X}$.

4.1.1. Design-independent geometry

If the set of design variables $\mathbf{h}$ includes only parameters that affect material constants, sizing variables (i.e. local thickness, cross-section areas, etc.), or external loads and boundary conditions, then the initial geometry of the model is design-independent ($\boldsymbol{X} \neq \boldsymbol{X}(\mathbf{h})$). In particular, the integration domain $\Omega^r \neq \Omega^r(\mathbf{h})$, and for all integrals over Ω^r we have

$$\mathrm{d}_{\mathbf{h}}\left[\int_{\Omega^r}(\cdot)\,\mathrm{d}\Omega\right] \equiv \int_{\Omega^r}\mathrm{d}_{\mathbf{h}}(\cdot)\,\mathrm{d}\Omega \tag{4.9}$$

Table 4.1. General scheme of computations for both the primary equilibrium and the design sensitivity analysis at a typical time step $[t_n\,,\,t_{n+1}]$ (DSA modifications to the pure primary algorithm are gray-shaded). Indices $n{+}1$ are skipped for better legibility


Input:
$\mathbf{q}_n\,,\ \mathbf{z}_n\,,\ \partial_{\mathbf{h}}\mathbf{q}_n\,,\ \partial_{\mathbf{h}}\mathbf{z}_n$
$i := 0$
$\mathbf{q}^{(0)} := \mathbf{q}_n$
loop over elements
assemble $\mathbf{f}^{\mathrm{res}\,(i)}$ and $\mathbf{K}^{(i)}$
matrix decomposition and backward substitution
$\mathbf{K}^{(i)}\,\delta\mathbf{q} = \mathbf{f}^{\mathrm{res}\,(i)}$
$\mathbf{q}^{(i+1)} = \mathbf{q}^{(i)} + \delta\mathbf{q}$
$i := i+1$
check convergence
no
yes
primary equilibrium analysis
loop over elements
assemble $\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{res}}$
N_d backward substitutions
$\mathbf{K}\,\mathrm{d}_{\mathbf{h}}\mathbf{q} = \hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{res}}$
sensitivity analysis
loop over elements
$\mathbf{z} = \mathbf{z}(\mathbf{q}\,,\ \mathbf{q}_n\,,\ \mathbf{z}_n)$
$\mathrm{d}_{\mathbf{h}}\mathbf{z} = \frac{\partial\mathbf{z}}{\partial\mathbf{q}}\,\mathrm{d}_{\mathbf{h}}\mathbf{q} + \frac{\partial\mathbf{z}}{\partial\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \frac{\partial\mathbf{z}}{\partial\mathbf{z}_n}\,\partial_{\mathbf{h}}\mathbf{z}_n$
update
Output:
$\mathbf{q}\,,\ \mathbf{z}\,,\ \mathrm{d}_{\mathbf{h}}\mathbf{q}\,,\ \mathrm{d}_{\mathbf{h}}\mathbf{z}$

(the same rule holds for boundary integrals over $\partial\Omega^r$). Besides, the shape functions and their gradients are design-independent, too, i.e. $\partial_{\mathbf{h}}\phi_{i\alpha} \equiv \mathbf{0}$, $\partial_{\mathbf{h}}\phi_{i\alpha,j} \equiv \mathbf{0}$.

The design derivative of the external force array is thus expressed as

$$\partial_{\mathbf{h}}\mathsf{f}^{\text{ext}}_{\alpha} = \int_{\Omega^r} \partial_{\mathbf{h}}\hat{f}_i\,\phi_{i\alpha}\,\mathrm{d}\Omega + \int_{\partial\Omega^r} \partial_{\mathbf{h}}\hat{t}_i\,\phi_{i\alpha}\,\mathrm{d}(\partial\Omega) \tag{4.10}$$

and can be explicitly determined at each time instance. The explicit design derivative of the internal force array is expressed by a general formula, cf. Eq. (3.167),

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{f}^{\text{int}}_{\alpha} = \int_{\Omega^r} \left(\mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\sigma}_{ij}\,\tilde{\mathsf{B}}_{ij\alpha} + \tilde{\sigma}_{ij}\,\mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\mathsf{B}}_{ij\alpha}\right)\mathrm{d}\Omega\,, \tag{4.11}$$

in which the explicit derivatives $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\sigma}_{ij}$ and $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\mathsf{B}}_{ij\alpha}$ have to be evaluated separately for different geometric descriptions of deformation and associated stress/strain measures. Generally, the derivation consists in writing down detailed formulae for the total design derivatives $\mathrm{d}_{\mathbf{h}}\tilde{\sigma}_{ij}$ and $\mathrm{d}_{\mathbf{h}}\tilde{\mathsf{B}}_{ij\alpha}$ and then removing from them all the terms that contain $\mathrm{d}_{\mathbf{h}}\mathsf{q}$ (i.e. retaining only those containing explicit design derivatives).

Let us first notice that the geometric operator $\tilde{\mathsf{B}}_{ij\alpha}$, cf. Eq. (3.169), is either design-independent (as in the small deformation case),

$$\mathrm{d}_{\mathbf{h}}\mathcal{B}_{ij\alpha} = \mathrm{d}_{\mathbf{h}}\phi_{i\alpha,j} = \mathbf{0},$$

or, in the large deformation case, it depends on design only implicitly, via q,

$$\begin{aligned}
\mathrm{d}_{\mathbf{h}}\mathsf{B}_{ij\alpha} &= \mathrm{d}_{\mathbf{h}}F_{ki}\,\phi_{k\alpha,j} = \phi_{k\alpha,j}\phi_{k\beta,i}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_{\beta}\,,\\
\mathrm{d}_{\mathbf{h}}\mathsf{b}_{ij\alpha} &= \phi_{i\alpha,k}\,\mathrm{d}_{\mathbf{h}}F^{-1}_{kj} = -\mathsf{b}_{ik\alpha}\mathsf{b}_{kj\beta}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_{\beta}\,.
\end{aligned}$$

Thus, in all the cases we have $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\mathsf{B}}_{ij\alpha} = \mathbf{0}$ and Eq. (4.11) can be reduced to

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{f}^{\text{int}}_{\alpha} = \int_{\Omega^r} \mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\sigma}_{ij}\,\tilde{\mathsf{B}}_{ij\alpha}\,\mathrm{d}\Omega\,. \tag{4.12}$$

For small deformations, Eq. (4.12) takes the form

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{f}^{\text{int}}_{\alpha} = \int_{\Omega^r} \mathrm{d}_{\mathbf{h}!\mathsf{q}}\sigma_{ij}\,\mathcal{B}_{ij\alpha}\,\mathrm{d}\Omega. \tag{4.13}$$

Noticing that the constitutive equation can be generally expressed as, cf. Section 3.3.1,

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}_n, \mathbf{z}_n; \mathbf{h}),$$

we can write

$$\mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma} = \boldsymbol{\mathcal{C}} : \mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon} + \frac{\partial\boldsymbol{\sigma}}{\partial\boldsymbol{\varepsilon}_n} : \partial_{\mathbf{h}}\boldsymbol{\varepsilon}_n + \frac{\partial\boldsymbol{\sigma}}{\partial\mathbf{z}_n} : \partial_{\mathbf{h}}\mathbf{z}_n + \partial_{\mathbf{h}}\boldsymbol{\sigma} \tag{4.14}$$

where the definition of the algorithmic constitutive tangent operator (3.94) has been utilized. Besides, upon the general definition (3.168),

$$\mathrm{d}_{\mathbf{h}}\varepsilon_{ij} = \mathrm{sym}(\mathcal{B}_{ij\alpha})\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_{\alpha}\,,$$

so the first term in the r.h.s. of Eq. (4.14) is the only one containing the implicit design derivative $\mathrm{d}_{\mathbf{h}}\boldsymbol{\mathsf{q}}$. Moreover, it depends solely on $\mathrm{d}_{\mathbf{h}}\boldsymbol{\mathsf{q}}$, i.e. $\mathrm{d}_{\mathbf{h}!\boldsymbol{\mathsf{q}}}\varepsilon_{ij} = \mathbf{0}$. Thus,

$$\mathrm{d}_{\mathbf{h}!\boldsymbol{\mathsf{q}}}\boldsymbol{\sigma} = \mathrm{d}_{\mathbf{h}!\boldsymbol{\varepsilon}}\boldsymbol{\sigma} = \frac{\partial\boldsymbol{\sigma}}{\partial\boldsymbol{\varepsilon}_n} : \partial_{\mathbf{h}}\boldsymbol{\varepsilon}_n + \frac{\partial\boldsymbol{\sigma}}{\partial\mathbf{z}_n} : \partial_{\mathbf{h}}\mathbf{z}_n + \partial_{\mathbf{h}}\boldsymbol{\sigma}. \tag{4.15}$$

Detailed computation scheme for $\mathrm{d}_{\mathbf{h}!\boldsymbol{\mathsf{q}}}\boldsymbol{\sigma}$ will be presented in Section 4.2.1.

For large deformations and material description, Eq. (4.12) takes the form

$$\mathrm{d}_{\mathbf{h}!\boldsymbol{\mathsf{q}}}\mathsf{f}^{\mathrm{int}}_{\alpha} = \int_{\Omega^r} \mathrm{d}_{\mathbf{h}!\boldsymbol{\mathsf{q}}}T_{ij}\,\mathsf{B}_{ij\alpha}\,\mathrm{d}\Omega\,. \tag{4.16}$$

The constitutive equation can be expressed as

$$\boldsymbol{T} = \boldsymbol{T}(\boldsymbol{E}, \boldsymbol{E}_n, \mathbf{z}_n; \mathbf{h}),$$

so that

$$\mathrm{d}_{\mathbf{h}}\boldsymbol{T} = \boldsymbol{\mathfrak{C}} : \mathrm{d}_{\mathbf{h}}\boldsymbol{E} + \frac{\partial\boldsymbol{T}}{\partial\boldsymbol{E}_n} : \partial_{\mathbf{h}}\boldsymbol{E}_n + \frac{\partial\boldsymbol{T}}{\partial\mathbf{z}_n} : \partial_{\mathbf{h}}\mathbf{z}_n + \partial_{\mathbf{h}}\boldsymbol{T}, \tag{4.17}$$

with $\boldsymbol{\mathfrak{C}}$ defined by Eq. (3.95) and

$$\mathrm{d}_{\mathbf{h}}E_{ij} = \mathrm{sym}(\mathsf{B}_{ij\alpha})\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_{\alpha}\,,$$

i.e. $\mathrm{d}_{\mathbf{h}!\boldsymbol{\mathsf{q}}}E_{ij} = \mathbf{0}$. Thus, the explicit design derivative of the second Piola–Kirchhoff stress contains all but the first term in the r.h.s. of Eq. (4.17),

$$\mathrm{d}_{\mathbf{h}!\boldsymbol{\mathsf{q}}}\boldsymbol{T} = \mathrm{d}_{\mathbf{h}!\boldsymbol{E}}\boldsymbol{T} = \frac{\partial\boldsymbol{T}}{\partial\boldsymbol{E}_n} : \partial_{\mathbf{h}}\boldsymbol{E}_n + \frac{\partial\boldsymbol{T}}{\partial\mathbf{z}_n} : \partial_{\mathbf{h}}\mathbf{z}_n + \partial_{\mathbf{h}}\boldsymbol{T}. \tag{4.18}$$

Detailed computation scheme for $\mathrm{d}_{\mathbf{h}!\boldsymbol{E}}\boldsymbol{T}$ is the same as for the small deformation case, upon only replacement of linear tensor measures with their large-deformation material counterparts.

For spatial geometric description of the large-deformation problem, Eq. (4.12) takes the form

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{f}_\alpha^{\mathrm{int}} = \int_{\Omega^r} \mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\tau}_{ij}\,\mathsf{b}_{ij\alpha}\mathrm{d}\Omega\,. \tag{4.19}$$

The constitutive equation can be expressed as, cf. Section 3.3.2,

$$\boldsymbol{\tau} = \boldsymbol{\tau}(\boldsymbol{F}, \boldsymbol{F}_n, \boldsymbol{\mathsf{z}}_n; \mathbf{h}),$$

so that

$$\mathrm{d}_{\mathbf{h}}\boldsymbol{\tau} = \frac{\partial\boldsymbol{\tau}}{\partial\boldsymbol{F}} : \mathrm{d}_{\mathbf{h}}\boldsymbol{F} + \frac{\partial\boldsymbol{\tau}}{\partial\boldsymbol{F}_n} : \partial_{\mathbf{h}}\boldsymbol{F}_n + \frac{\partial\boldsymbol{\tau}}{\partial\boldsymbol{\mathsf{z}}_n} : \partial_{\mathbf{h}}\boldsymbol{\mathsf{z}}_n + \partial_{\mathbf{h}}\boldsymbol{\tau}, \tag{4.20}$$

with

$$\mathrm{d}_{\mathbf{h}}F_{ij} = \phi_{i\alpha,j}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_\alpha\,,$$

i.e. $\mathrm{d}_{\mathbf{h}!\mathsf{q}}F_{ij} = \mathbf{0}$. Thus, the explicit design derivative of the Kirchhoff stress contains all but the first term in the r.h.s. of Eq. (4.20),

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\tau} = \mathrm{d}_{\mathbf{h}!\boldsymbol{F}}\boldsymbol{\tau} = \frac{\partial\boldsymbol{\tau}}{\partial\boldsymbol{F}_n} : \partial_{\mathbf{h}}\boldsymbol{F}_n + \frac{\partial\boldsymbol{\tau}}{\partial\boldsymbol{\mathsf{z}}_n} : \partial_{\mathbf{h}}\boldsymbol{\mathsf{z}}_n + \partial_{\mathbf{h}}\boldsymbol{\tau}. \tag{4.21}$$

Detailed computation scheme for $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\tau}$ will be presented in Section 4.2.2.

4.1.2. Design-dependent geometry (shape sensitivity)

Some design variables may affect initial geometry of the analysed body. In such a case $\boldsymbol{X} = \boldsymbol{X}(\mathbf{h})$ is considered a explicitly given field. Before formulation of sensitivity analysis for such a case can be derived, let us first focus on particular ways of explicit definition of geometry with the use of scalar design parameters $\mathbf{h}$.

There are two commonly recognized approaches to formulation of sensitivity equations at design-dependent geometry: the *material derivative approach* (MDA) and the *domain parametrization approach* (DPA). Discussion of their mathematical details, including proof of their formal equivalence and considerations on efficiency and applicability of each of them can be found e.g. in [65]. The approach outlined below follows the formalism of DPA, as the latter appears to naturally conform to the isoparametric finite element formulation employed in the numerical implementation of the presented DSA formulation.

In DPA, the design-dependent configuration $\mathcal{C}^r$ is assumed to be an image of a design-independent parent configuration $\mathcal{C}^p$ in a one-to-one mapping parameterized by $\mathbf{h}$,

$$\boldsymbol{X}(\mathbf{h}) = \hat{\chi}(\boldsymbol{\xi}, \mathbf{h}), \tag{4.22}$$

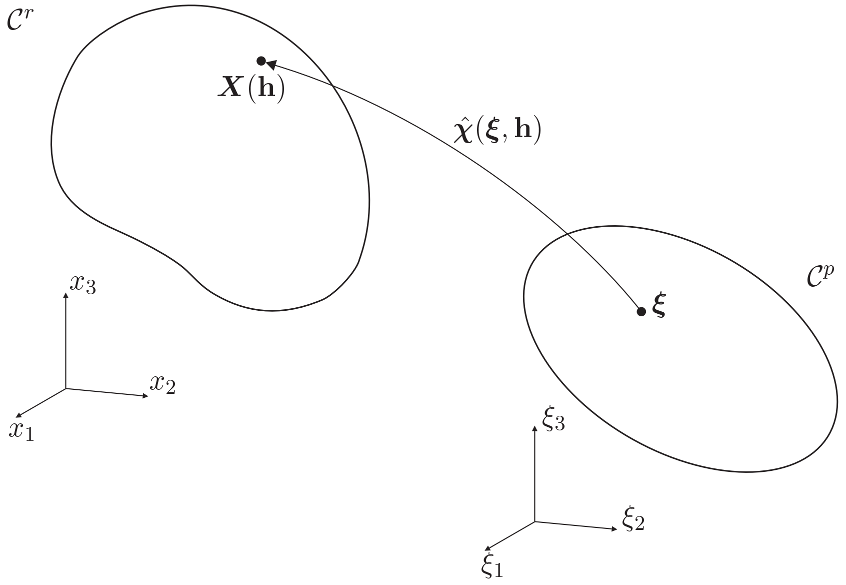


Figure 4.1. Domain parametrization approach

see Fig. 4.1. Denoting by $\hat{\boldsymbol{F}}$ the mapping's gradient in $\mathcal{C}^p$,

$$\hat{\boldsymbol{F}}(\mathbf{h}) = \frac{\mathrm{d}\hat{\boldsymbol{\chi}}}{\mathrm{d}\boldsymbol{\xi}}\,, \tag{4.23}$$

and by $\hat{J}$ its determinant, $\hat{J} = \det \hat{\boldsymbol{F}}$, we can express design derivatives of an arbitrary volume integral over Ω^r as

$$\begin{aligned} \mathrm{d}_{\mathbf{h}}\left[\int_{\Omega^r} (\cdot)\,\mathrm{d}\Omega\right] \;&=\; \mathrm{d}_{\mathbf{h}}\left[\int_{\Omega^p} (\cdot)\hat{J}\,\mathrm{d}\Omega\right] = \int_{\Omega^p}\left[\mathrm{d}_{\mathbf{h}}(\cdot)\,\hat{J} + (\cdot)\,\mathrm{d}_{\mathbf{h}}\hat{J}\right]\mathrm{d}\Omega \\ &=\; \int_{\Omega^r}\left[\mathrm{d}_{\mathbf{h}}(\cdot) + (\cdot)\,\mathrm{tr}(\partial_{\mathbf{h}}\hat{\boldsymbol{F}}\,\hat{\boldsymbol{F}}^{-1})\right]\mathrm{d}\Omega \end{aligned} \tag{4.24}$$

and of an arbitrary surface integral over $\partial\Omega^r$ as

$$\begin{aligned} \mathrm{d}_{\mathbf{h}}&\left[\int_{\partial\Omega^r} (\cdot)\,\boldsymbol{n}^r\,\mathrm{d}(\partial\Omega)\right] = \mathrm{d}_{\mathbf{h}}\left[\int_{\partial\Omega^p} (\cdot)\,\hat{\boldsymbol{F}}^{-\mathrm{T}}\boldsymbol{n}^p\hat{J}\,\mathrm{d}(\partial\Omega)\right] \\ &= \int_{\partial\Omega^p}\left[\mathrm{d}_{\mathbf{h}}(\cdot)\,\hat{\boldsymbol{F}}^{-\mathrm{T}}\boldsymbol{n}^p\hat{J} + (\cdot)\,\mathrm{d}_{\mathbf{h}}(\hat{\boldsymbol{F}}^{-\mathrm{T}}\hat{J})\,\boldsymbol{n}^p\right]\mathrm{d}(\partial\Omega) \\ &= \int_{\partial\Omega^r}\left[\mathrm{d}_{\mathbf{h}}(\cdot)\,\boldsymbol{n}^r + (\cdot)\,[\boldsymbol{I}\,\mathrm{tr}(\partial_{\mathbf{h}}\hat{\boldsymbol{F}}\,\hat{\boldsymbol{F}}^{-1}) - (\partial_{\mathbf{h}}\hat{\boldsymbol{F}}\,\hat{\boldsymbol{F}}^{-1})^{\mathrm{T}}]\,\boldsymbol{n}^r\right]\mathrm{d}(\partial\Omega) \end{aligned} \tag{4.25}$$

where $\boldsymbol{n}^r(\mathbf{h})$ and $\boldsymbol{n}^p$ denote normal surface unit vectors in $\mathcal{C}^r$ and $\mathcal{C}^p$, respectively.

The parent configuration $\mathcal{C}^p$ and the mapping $\hat{\boldsymbol{\chi}}$ can be defined in a variety of ways. From the numerical point of view, it appears to be very convenient to relate the mapping $\hat{\boldsymbol{\chi}}$ with the assumed finite-element discretization of the displacement field (3.163). Denoting by $\mathsf{\mathbf{X}} = \{\mathsf{X}_\alpha\}$ an array of nodal coordinates in $\mathcal{C}^r$ (obviously, $\mathsf{\mathbf{X}} = \mathsf{\mathbf{X}}(\mathbf{h})$) and appropriately defining in $\mathcal{C}^p$ an array of design-independent functions $\hat{\phi}_{i\alpha}(\boldsymbol{\xi})$, one can define the mapping $\hat{\boldsymbol{\chi}}$ as

$$\hat{\chi}_i(\boldsymbol{\xi}, \mathbf{h}) = \hat{\phi}_{i\alpha}(\boldsymbol{\xi})\, \mathsf{X}_\alpha(\mathbf{h}). \tag{4.26}$$

It is easy to see that design-dependent functions $\phi_{i\alpha}(\boldsymbol{X}(\mathbf{h}))$ defined as

$$\phi_{i\alpha}(\boldsymbol{X}) = \hat{\phi}_{i\alpha}\left(\hat{\boldsymbol{\chi}}^{-1}(\boldsymbol{X})\right) \tag{4.27}$$

satisfy requirements of shape functions in (3.163), and thus the spatial discretization (3.163) can be equivalently expressed in terms of design-independent shape functions $\hat{\phi}_{i\alpha}(\boldsymbol{\xi})$ defined in $\mathcal{C}^p$.

Defining in particular the configuration $\mathcal{C}^p$ as a sum of repeatable unit cubes, tetrahedra, and/or other simple polyhedra, and dividing the domain Ω^r into subdomains (elements) of simple geometry being diffeomorphic images of the

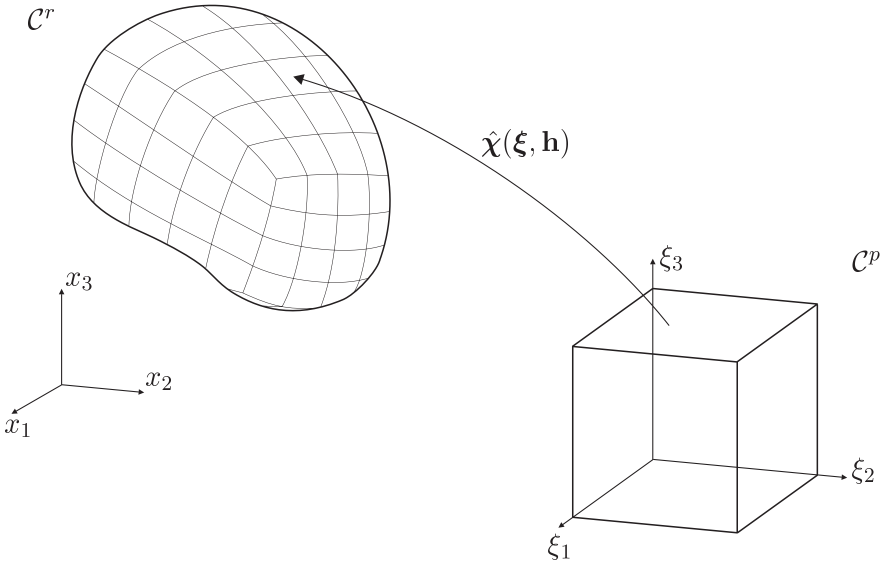


Figure 4.2. Domain parametrization approach in terms of the isoparametric element concept

above simple figures, we can generalize the above considerations and identify the design-independent configuration $\mathcal{C}^p$ with the parent configuration of isoparametric finite elements [49, 141], see Fig. 4.2. In such a case, functions $\hat{\phi}_{i\alpha}(\boldsymbol{\xi})$ are usually simple form polynomials defined for each parent polyhedron type in a way ensuring fulfillment of requirements of continuity and smoothness appropriate for the finite element method formulation. Division of Ω^r into elements is obviously design-dependent, i.e. all the nodal coordinates X_α are, as assumed in Eq. (4.26), explicit functions of $\mathbf{h}$.

Introducing the notation $(\cdot)_{;j} \equiv \partial(\cdot)/\partial\xi_j$ one can express the mapping's gradient and its design derivatives as

$$\hat{F}_{ij} = \hat{\phi}_{i\alpha;j}\,\mathsf{X}_\alpha\,, \qquad \partial_{\mathbf{h}}\hat{F}_{ij} = \hat{\phi}_{i\alpha;j}\,\partial_{\mathbf{h}}\mathsf{X}_\alpha\,, \qquad \partial_{\mathbf{h}}\hat{F}_{ik}\hat{F}^{-1}_{kj} = \phi_{i\alpha,j}\,\partial_{\mathbf{h}}\mathsf{X}_\alpha\,, \tag{4.28}$$

and rewrite the general formulae (4.24)–(4.25) as

$$\mathrm{d}_{\mathbf{h}}\left[\int_{\Omega^r}(\cdot)\,\mathrm{d}\Omega\right] = \int_{\Omega^r}\mathrm{d}_{\mathbf{h}}(\cdot)\,\mathrm{d}\Omega + \left[\int_{\Omega^r}(\cdot)\,\phi_{k\alpha,k}\,\mathrm{d}\Omega\right]\partial_{\mathbf{h}}\mathsf{X}_\alpha\,, \tag{4.29}$$

$$\begin{aligned}\mathrm{d}_{\mathbf{h}}\left[\int_{\partial\Omega^r}(\cdot)\,n^r_j\,\mathrm{d}(\partial\Omega)\right] &= \int_{\partial\Omega^r}\mathrm{d}_{\mathbf{h}}(\cdot)\,n^r_j\,\mathrm{d}(\partial\Omega) \\ &\quad + \left[\int_{\partial\Omega^r}(\cdot)\,[n^r_j\,\phi_{k\alpha,k} - n^r_k\,\phi_{k\alpha,j}]\,\mathrm{d}(\partial\Omega)\right]\partial_{\mathbf{h}}\mathsf{X}_\alpha\,.\end{aligned} \tag{4.30}$$

Appearance of the second terms (containing $\partial_{\mathbf{h}}\mathsf{X}_\alpha$) in the r.h.s. of the above formulae makes the fundamental difference between the shape and non-shape sensitivity formulation. However, as it will be demonstrated, several further differences may appear in the form of the first terms' integrand $\mathrm{d}_{\mathbf{h}}(\cdot)$, too.

Before we utilize Eqs. (4.29)–(4.30) to write down the expressions on $\partial_{\mathbf{h}}\boldsymbol{\mathsf{f}}^{\mathrm{ext}}$ and $\mathrm{d}_{\mathbf{h}!\boldsymbol{\mathsf{q}}}\boldsymbol{\mathsf{f}}^{\mathrm{int}}$ for the case of design-dependent geometry, let us make some observations about design-differentiation of displacement field and its gradient. Since the volume integration in Eq. (4.29) can be as well performed in the parent configuration $\mathcal{C}^p$, it seems reasonable to express the terms in the integrand as fields on this configuration. As mentioned above, the shape functions $\phi_{i\alpha}(\boldsymbol{X}(\mathbf{h}))$ are equal to design-independent functions $\hat{\phi}_{i\alpha}(\boldsymbol{\xi})$, $\boldsymbol{\xi} = \hat{\boldsymbol{\chi}}^{-1}(\boldsymbol{X})$, and their design derivatives computed in the parent configuration are null,

$$\partial_{\mathbf{h}}\phi_{i\alpha} = \mathbf{0}\,. \tag{4.31}$$

The gradients $\phi_{i\alpha,j}(\boldsymbol{X}(\mathbf{h}))$ can be mapped onto the $\mathcal{C}^p$ as

$$\phi_{i\alpha,j} = \hat{\phi}_{i\alpha,j}(\boldsymbol{\xi},\mathbf{h}) = \hat{\phi}_{i\alpha;l}(\boldsymbol{\xi})\,\hat{F}^{-1}_{lj}(\boldsymbol{\xi},\mathbf{h}) \tag{4.32}$$

which allows to express their design derivatives in this configuration as, cf. Eq. (4.28)$_3$,

$$\partial_{\mathbf{h}}\phi_{i\alpha,j} = \hat{\phi}_{i\alpha;l}\,\partial_{\mathbf{h}}\hat{F}^{-1}_{lj} = -\hat{\phi}_{i\alpha;l}\,\hat{F}^{-1}_{lk}\,\partial_{\mathbf{h}}\hat{F}_{km}\,\hat{F}^{-1}_{mj} = -\phi_{i\alpha,k}\,\phi_{k\beta,j}\,\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \tag{4.33}$$

The fields $\boldsymbol{u}$ and $\boldsymbol{F}$ can thus also be mapped onto $\mathcal{C}^p$,

$$u_i \;=\; u_i(\boldsymbol{\xi},\mathbf{h}) = \hat{\phi}_{i\alpha}(\boldsymbol{\xi})\,\mathsf{q}_{\alpha}(\mathbf{h})\,, \tag{4.34}$$

$$F_{ij} \;=\; F_{ij}(\boldsymbol{\xi},\mathbf{h}) = \delta_{ij} + u_{i,j}(\boldsymbol{\xi},\mathbf{h}) = \delta_{ij} + \hat{\phi}_{i\alpha,j}(\boldsymbol{\xi},\mathbf{h})\,\mathsf{q}_{\alpha}(\mathbf{h})\,, \tag{4.35}$$

so their design derivatives in this configuration are

$$\mathrm{d}_{\mathbf{h}}u_i \;=\; \hat{\phi}_{i\alpha}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_{\alpha}\,, \tag{4.36}$$

$$\mathrm{d}_{\mathbf{h}}F_{ij} \;=\; \phi_{i\alpha,j}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_{\alpha} + \partial_{\mathbf{h}}\phi_{i\alpha,j}\,\mathsf{q}_{\alpha} = \phi_{i\beta,j}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_{\beta} - u_{i,k}\,\phi_{k\beta,j}\,\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \tag{4.37}$$

From the latter two equations it can be concluded that

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}u_i = \mathbf{0}, \qquad\qquad \mathrm{d}_{\mathbf{h}!\mathsf{q}}F_{ij} = -u_{i,k}\,\phi_{k\beta,j}\,\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \tag{4.38}$$

We can now write down the expressions on $\partial_{\mathbf{h}}\boldsymbol{\mathsf{f}}^{\mathrm{ext}}$ and $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\mathsf{f}}^{\mathrm{int}}$ for the case of design-dependent geometry. Expressing the surface forces as $\hat{\boldsymbol{t}} = \hat{\boldsymbol{T}}\boldsymbol{n}^r$ where $\hat{\boldsymbol{T}}$ denotes the known nominal (first Piola–Kirchhoff) stress tensor, we have

$$\begin{aligned}
\partial_{\mathbf{h}}\mathsf{f}^{\mathrm{ext}}_{\alpha} \;=\; & \int_{\Omega^r}\partial_{\mathbf{h}}\hat{f}_i\,\phi_{i\alpha}\,\mathrm{d}\Omega + \int_{\partial\Omega^r}\partial_{\mathbf{h}}\hat{T}_{ij}n^r_j\,\phi_{i\alpha}\,\mathrm{d}(\partial\Omega) \\
& + \left[\int_{\Omega^r}\hat{f}_i\,\phi_{i\alpha}\,\phi_{k\beta,k}\,\mathrm{d}\Omega\right]\partial_{\mathbf{h}}\mathsf{X}_{\beta} \\
& + \left[\int_{\partial\Omega^r}\hat{T}_{ij}\,\phi_{i\alpha}\,[n^r_j\,\phi_{k\beta,k} - n^r_k\,\phi_{k\beta,j}]\,\mathrm{d}(\partial\Omega)\right]\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,,
\end{aligned} \tag{4.39}$$

and

$$\begin{aligned}
\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{f}^{\mathrm{int}}_{\alpha} \;=\; & \int_{\Omega^r}\Big(\mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\sigma}_{ij}\,\tilde{\mathsf{B}}_{ij\alpha} + \tilde{\sigma}_{ij}\,\mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\mathsf{B}}_{ij\alpha}\Big)\,\mathrm{d}\Omega \\
& + \left[\int_{\Omega^r}\tilde{\sigma}_{ij}\,\tilde{\mathsf{B}}_{ij\alpha}\,\phi_{k\beta,k}\,\mathrm{d}\Omega\right]\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,.
\end{aligned} \tag{4.40}$$

The first array, (4.39), contains terms that are explicitly known. In the other, the derivatives $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\sigma}_{ij}$ and $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\mathsf{B}}_{ij\alpha}$ have to be determined. Since both $\tilde{\sigma}_{ij}$ and $\tilde{\mathsf{B}}_{ij\alpha}$ depend on initial geometry, the forms of their design derivatives are different than those derived in Section 4.1.1.

For small deformations ($\tilde{\sigma}_{ij} = \sigma_{ij}$, $\tilde{\mathsf{B}}_{ij\alpha} = \mathcal{B}_{ij\alpha} = \phi_{i\alpha,j}$), the total design derivative of the geometric operator

$$\mathrm{d}_{\mathbf{h}}\mathcal{B}_{ij\alpha} = -\phi_{i\alpha,k}\,\phi_{k\beta,j}\,\partial_{\mathbf{h}}\mathsf{X}_{\beta} \tag{4.41}$$

is independent on $\mathrm{d}_{\mathbf{h}}\mathsf{q}_\alpha$ and thus equal to its explicit part $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathcal{B}_{ij\alpha}$. The total design derivative of stress has the form (4.14), however, the design derivative of strain, $\mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon} = \mathrm{sym}(\mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon}^{\mathrm{u}})$, $\mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon}^{\mathrm{u}} = \mathrm{d}_{\mathbf{h}}(\boldsymbol{\nabla}\boldsymbol{u})$, has now a nonzero explicit, geometry-dependent part,

$$\mathrm{d}_{\mathbf{h}}\varepsilon^{\mathrm{u}}_{ij} = \mathrm{d}_{\mathbf{h}}u_{i,j} = \phi_{i\beta,j}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta - u_{i,k}\,\phi_{k\beta,j}\,\partial_{\mathbf{h}}\mathsf{X}_\beta = \mathcal{B}_{ij\beta}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta + \mathrm{d}_{\mathbf{h}!\mathsf{q}}\varepsilon^{\mathrm{u}}_{ij}\,,$$

with

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\varepsilon^{\mathrm{u}}_{ij} = -u_{i,k}\,\mathcal{B}_{kj\beta}\,\partial_{\mathbf{h}}\mathsf{X}_\beta\,. \tag{4.42}$$

Thus, for the case of shape sensitivity, we have to complete the previous section's result (4.15) by

$$\begin{aligned}\mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\sigma} &= \mathrm{d}_{\mathbf{h}!\boldsymbol{\varepsilon}}\boldsymbol{\sigma} + \boldsymbol{\mathcal{C}} : \mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\varepsilon} \\ &= \boldsymbol{\mathcal{C}} : \mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\varepsilon} + \frac{\partial\boldsymbol{\sigma}}{\partial\boldsymbol{\varepsilon}_n} : \partial_{\mathbf{h}}\boldsymbol{\varepsilon}_n + \frac{\partial\boldsymbol{\sigma}}{\partial\mathbf{z}_n} : \partial_{\mathbf{h}}\mathbf{z}_n + \partial_{\mathbf{h}}\boldsymbol{\sigma}\end{aligned} \tag{4.43}$$

where, due to symmetry of $\boldsymbol{\mathcal{C}}$, the explicit strain design derivative $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\varepsilon}$ can be replaced with $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\varepsilon}^{\mathrm{u}}$ (4.42). Detailed computation scheme for $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\sigma}$ will be presented in Section 4.2.1.

For large deformations and material description ($\tilde{\sigma}_{ij} = \boldsymbol{T}_{ij}$, $\tilde{\mathsf{B}}_{ij\alpha} = \mathsf{B}_{ij\alpha} = F_{ki}\phi_{k\alpha,j}$), the total design derivative of the geometric operator is expressed as

$$\begin{aligned}\mathrm{d}_{\mathbf{h}}\mathsf{B}_{ij\alpha} &= \phi_{k\alpha,j}\phi_{k\beta,i}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta - u_{k,l}\,\phi_{k\alpha,j}\,\phi_{l\beta,i}\,\partial_{\mathbf{h}}\mathsf{X}_\beta - F_{ki}\,\phi_{k\alpha,l}\,\phi_{l\beta,j}\,\partial_{\mathbf{h}}\mathsf{X}_\beta \\ &= \phi_{k\alpha,j}\phi_{k\beta,i}(\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta + \partial_{\mathbf{h}}\mathsf{X}_\beta) - (\mathsf{B}_{lj\alpha}\,\phi_{l\beta,i} + \mathsf{B}_{il\alpha}\,\phi_{l\beta,j})\,\partial_{\mathbf{h}}\mathsf{X}_\beta\end{aligned}$$

and thus its explicit part has the form

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{B}_{ij\alpha} = (\phi_{k\alpha,j}\phi_{k\beta,i} - \mathsf{B}_{kj\alpha}\,\phi_{k\beta,i} - \mathsf{B}_{ik\alpha}\,\phi_{k\beta,j})\,\partial_{\mathbf{h}}\mathsf{X}_\beta\,. \tag{4.44}$$

The total design derivative of stress has the form (4.17) and again, the design derivative of strain $\mathrm{d}_{\mathbf{h}}\boldsymbol{E} = \mathrm{sym}(\mathrm{d}_{\mathbf{h}}\boldsymbol{E}^{\mathrm{u}})$, $\mathrm{d}_{\mathbf{h}}\boldsymbol{E}^{\mathrm{u}} = \boldsymbol{F}^{\mathrm{T}}\mathrm{d}_{\mathbf{h}}\boldsymbol{F}$ has now a nonzero explicit, geometry-dependent part,

$$\begin{aligned}\mathrm{d}_{\mathbf{h}}E^{\mathrm{u}}_{ij} &= F_{ki}\,\mathrm{d}_{\mathbf{h}}F_{kj} = F_{ki}\,(\phi_{k\beta,j}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta - u_{k,l}\,\phi_{l\beta,j}\,\partial_{\mathbf{h}}\mathsf{X}_\beta) \\ &= \mathsf{B}_{ij\beta}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta + \mathrm{d}_{\mathbf{h}!\mathsf{q}}E^{\mathrm{u}}_{ij}\,,\end{aligned} \tag{4.45}$$

with

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}E^{\mathrm{u}}_{ij} = -F_{ki}u_{k,l}\,\phi_{l\beta,j}\,\partial_{\mathbf{h}}\mathsf{X}_\beta = (\mathsf{B}_{ij\beta} - F_{ki}F_{kl}\,\phi_{l\beta,j})\,\partial_{\mathbf{h}}\mathsf{X}_\beta\,. \tag{4.46}$$

Thus, the previous section's result (4.18) has to be completed for the case of shape sensitivity by

$$\begin{aligned} \mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{T} &= \mathrm{d}_{\mathbf{h}!\boldsymbol{E}}\boldsymbol{T} + \boldsymbol{\mathfrak{C}} : \mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{E} \\ &= \boldsymbol{\mathfrak{C}} : \mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{E} + \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{E}_n} : \partial_{\mathbf{h}}\boldsymbol{E}_n + \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{\mathsf{z}}_n} : \partial_{\mathbf{h}}\boldsymbol{\mathsf{z}}_n + \partial_{\mathbf{h}}\boldsymbol{T}, \end{aligned} \tag{4.47}$$

where, due to symmetry of $\boldsymbol{\mathfrak{C}}$, the explicit strain design derivative $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{E}$ can be replaced with $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{E}^{\mathrm{u}}$ (4.46).

For spatial geometric description of the large-deformation problem ($\tilde{\sigma}_{ij} = \tau_{ij}$, $\tilde{\mathsf{B}}_{ij\alpha} = \mathsf{b}_{ij\alpha} = \phi_{i\alpha,k}F^{-1}_{kj}$), the total design derivative of the geometric operator is expressed as

$$\mathrm{d}_{\mathbf{h}}\mathsf{b}_{ij\alpha} \;=\; \mathrm{d}_{\mathbf{h}}\phi_{i\alpha,l}\,F^{-1}_{lj} - \phi_{i\alpha,l}\,F^{-1}_{lk}\,\mathrm{d}_{\mathbf{h}}F_{km}\,F^{-1}_{mj} = -\mathsf{b}_{ik\alpha}\mathsf{b}_{kj\alpha}(\mathrm{d}_{\mathbf{h}}\mathsf{q}_{\beta} + \partial_{\mathbf{h}}\mathsf{X}_{\beta})$$

and thus its explicit part has simply the form

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{b}_{ij\alpha} = -\mathsf{b}_{ik\alpha}\mathsf{b}_{kj\alpha}\,\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \tag{4.48}$$

The total design derivative of stress has the form (4.20) but the design derivative of the deformation gradient $\mathrm{d}_{\mathbf{h}}\boldsymbol{F}$ has now a nonzero explicit, geometry-dependent part, cf. Eq. $(4.38)_2$,

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}F_{ij} = -u_{i,k}\,\phi_{k\beta,j}\,\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \tag{4.49}$$

Utilizing Eqs. (3.96)–(3.97) we can also write

$$\mathrm{d}\tau_{ij} = \mathfrak{c}_{ijkl}\,\mathrm{d}F_{km}\,F^{-1}_{ml} + \tau_{lj}\,\mathrm{d}F_{ik}\,F^{-1}_{kl} + \tau_{ik}\,\mathrm{d}F_{jl}\,F^{-1}_{lk} \tag{4.50}$$

which allows to express the first term in the r.h.s. of Eq. (4.20) in the index notation as

$$\frac{\partial \tau_{ij}}{\partial F_{kl}}\,\mathrm{d}_{\mathbf{h}}F_{kl} = \mathfrak{c}_{ijkl}\,\mathrm{d}_{\mathbf{h}}F_{km}\,F^{-1}_{ml} + \tau_{lj}\,\mathrm{d}_{\mathbf{h}}F_{ik}\,F^{-1}_{kl} + \tau_{ik}\,\mathrm{d}_{\mathbf{h}}F_{jl}\,F^{-1}_{lk} \tag{4.51}$$

and, particularly, its explicit part as

$$\begin{aligned} \frac{\partial \tau_{ij}}{\partial F_{kl}}\,\mathrm{d}_{\mathbf{h}!\mathsf{q}}F_{kl} &= \mathfrak{c}_{ijkl}\,\mathrm{d}_{\mathbf{h}!\mathsf{q}}F_{km}\,F^{-1}_{ml} + \tau_{lj}\,\mathrm{d}_{\mathbf{h}!\mathsf{q}}F_{ik}\,F^{-1}_{kl} + \tau_{ik}\,\mathrm{d}_{\mathbf{h}!\mathsf{q}}F_{jl}\,F^{-1}_{lk} \\ &= -\left(\mathfrak{c}_{ijkl}\,u_{k,n} + \tau_{lj}\,u_{i,n} + \tau_{il}\,u_{j,n}\right)\mathsf{b}_{nl\beta}\,\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \end{aligned} \tag{4.52}$$

Thus, the previous section's result (4.21) has to be completed for the case of shape sensitivity by

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{\tau} = \mathrm{d}_{\mathbf{h}!\boldsymbol{F}}\boldsymbol{\tau} + \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{F}} : \mathrm{d}_{\mathbf{h}!\mathsf{q}}\boldsymbol{F} \tag{4.53a}$$

with

$$\mathrm{d}_{\mathbf{h}!\boldsymbol{F}}\boldsymbol{\tau} = \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{F}_n} : \partial_{\mathbf{h}} \boldsymbol{F}_n + \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{z}_n} : \partial_{\mathbf{h}} \mathbf{z}_n + \partial_{\mathbf{h}} \boldsymbol{\tau} \tag{4.53b}$$

and, cf. Eqs. (3.95), (3.97) and (3.98c),

$$\frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{F}} : \mathrm{d}_{\mathbf{h}!\mathbf{q}}\boldsymbol{F} = \mathfrak{c} : (\mathrm{d}_{\mathbf{h}!\mathbf{q}}\boldsymbol{F}\,\boldsymbol{F}^{-1}) + \mathrm{d}_{\mathbf{h}!\mathbf{q}}\boldsymbol{F}\,\boldsymbol{F}^{-1}\,\boldsymbol{\tau} + \boldsymbol{\tau}\,(\mathrm{d}_{\mathbf{h}!\mathbf{q}}\boldsymbol{F}\,\boldsymbol{F}^{-1})^{\mathrm{T}}, \tag{4.53c}$$

where $\mathrm{d}_{\mathbf{h}!\mathbf{q}}\boldsymbol{F}$ is given by Eq. (4.49). Detailed computation scheme for $\mathrm{d}_{\mathbf{h}!\mathbf{q}}\boldsymbol{\tau}$ will be presented in Section 4.2.2.

4.1.2.1. Remark on uniqueness of shape sensitivity solutions

In the cases of design-independent geometry (Section 4.1.1), the explicit design derivatives of the input fields can be usually defined in a strict, unique way. In the problems with design-dependent geometry, typically understood as design-dependent external shape, this advantage cannot be enjoyed. Explicit definition of the relationships $\mathbf{X}(\mathbf{h})$ and the resulting design derivatives $\partial_{\mathbf{h}}\mathbf{X}$ (also called *design velocities*) is nonunique and depends on the analysts choice to the same extent as definition of the nodal coordinates $\mathbf{X}$ themselves, i.e. the finite element discretization. Let us consider an example of a simple structure in Fig. 4.3. Assuming the rounding radius to be the design parameter, $h = R$, there is an

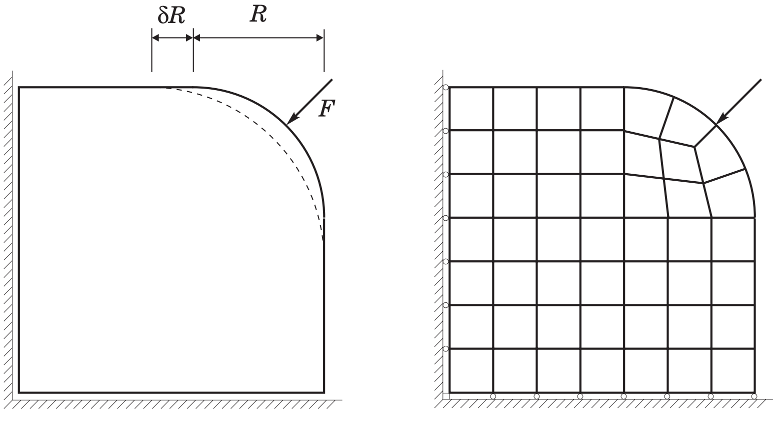


Figure 4.3. Rectangle with a rounded corner, the rounding radius R is the design parameter; left: geometry, load and boundary conditions (dotted line indicates possible geometry perturbation); right: primary finite element discretization

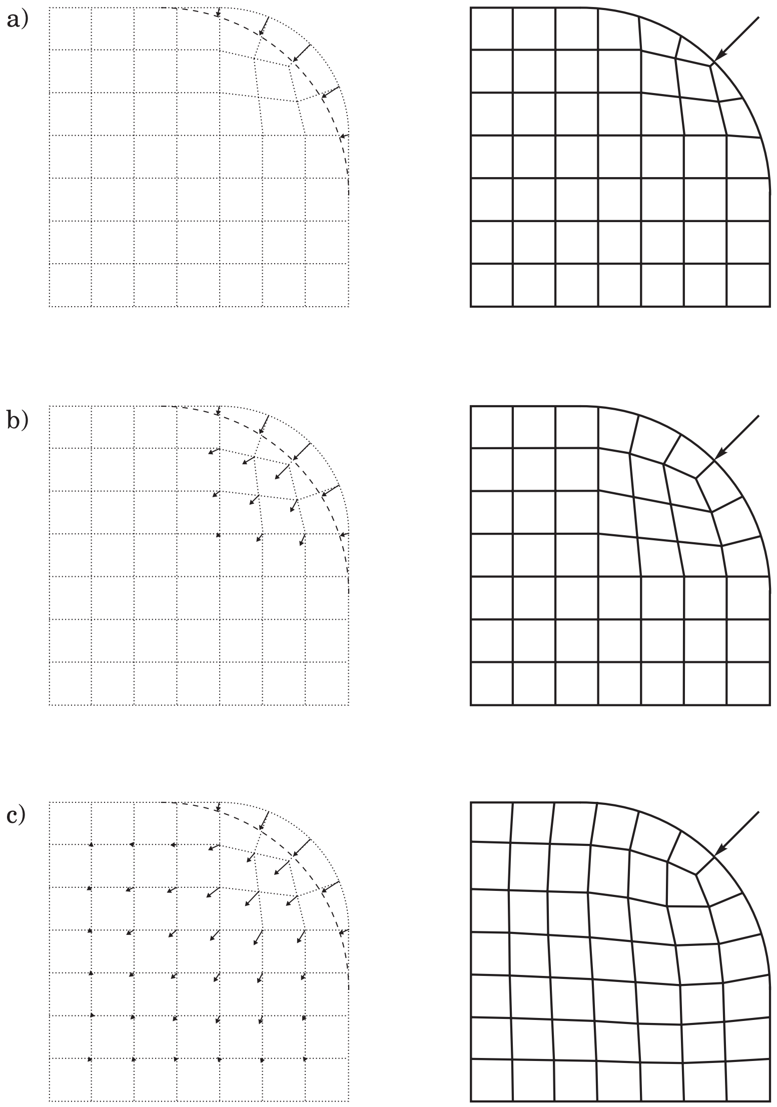


Figure 4.4. Consequences of different choices of the nodal coordinate design velocities. Design sensitivity results obtained for the values depicted on the left side (arrows denote the vectors $\partial_{\mathbf{h}}\mathbf{X}$ for particular nodes) allow to approximate the behaviour of a perturbed structure provided it is modelled with the mesh presented on the right side

infinite variety of possible design velocity definitions for the given finite element mesh. The left column in Fig. 4.4 presents a few characteristic examples in which (a) only surface nodes, (b) surface and neighbouring nodes, and finally (c) all nodes in the mesh are assigned non-zero design velocity (denoted by arrows). The sensitivity solution, which directly depends on these choices, is nonunique, too. It is necessary to make a comment on this issue and to explain physical meaning of this nonuniqueness of results.

Recalling the definition (2.11a) of a design derivative we can approximate, for small perturbations $\delta\mathbf{h}$, the perturbed response $\mathbf{q}(\mathbf{h}{+}\delta\mathbf{h})$ by

$$\mathbf{q}(\mathbf{h}{+}\delta\mathbf{h}) \approx \mathbf{q}(\mathbf{h}) + \mathrm{d}_{\mathbf{h}}\mathbf{q}(\mathbf{h})\,\delta\mathbf{h}\,. \tag{4.54}$$

If the solution $\mathrm{d}_{\mathbf{h}}\mathbf{q}(\mathbf{h})$ has been determined for an arbitrary $\partial_{\mathbf{h}}\mathbf{X}$, the above approximation can only be justified under the assumption that the perturbed input conforms to the following scheme,

$$\mathbf{X}(\mathbf{h}{+}\delta\mathbf{h}) \approx \mathbf{X}(\mathbf{h}) + \partial_{\mathbf{h}}\mathbf{X}(\mathbf{h})\,\delta\mathbf{h}\,, \tag{4.55}$$

i.e. the perturbed finite element mesh is characterized by (i) the same topology as the primary one and (ii) the nodal coordinates described with Eq. (4.55). Thus, different nodal design velocities shown in Fig. 4.4a,b,c (left column), when introduced as input to the shape DSA algorithm, will lead to the response sensitivity solutions that allow to approximate with Eq. (4.54) the responses of the corresponding perturbed structures shown in the right column of the same figure.

A question arises, which of the sensitivity solutions is ‘correct’, or at least ‘most accurate’. Bearing in mind that what we call response $\mathbf{q}$ is actually one of various possible discrete approximations of the continuum response fields, the first answer is ‘each of them’, by which we mean that an approximate $\mathbf{q}(\mathbf{h})$, corresponding to certain finite element discretization $\mathbf{X}(\mathbf{h})$, related to different possible approximates $\mathbf{q}(\mathbf{h}{+}\delta\mathbf{h})$, corresponding to different possible finite element discretizations $\mathbf{X}(\mathbf{h}{+}\delta\mathbf{h})$, may result in different sensitivity gradients, depending on how the perturbations $\delta\mathbf{h}$ affect the primary discretization. Thus, each of the sensitivity solutions is good for a particular way of understanding the term ‘perturbed structure’ or, more strictly, ‘perturbed discrete model of the structure’, expressed in the particular definition of design velocities $\partial_{\mathbf{h}}\mathbf{X}$.

This answer is, however, unsatisfactory since it is commonly known that accuracy of a finite element discretized problem solution strongly depends on the *mesh quality.* This term is not easily definable in terms of mathematical formulae but, based on results of error analysis in the theory of finite elements [141], one can point out features of finite element discretization that may be reasons of potential inaccuracies of the solution (e.g. distorted or extremely stretched elements, coincident nodes, etc.). Without coming into details (which are not the subject of this dissertation) we can observe that, in the computational practice, finite element meshes are constructed according to certain (precise or intuitive) criteria of regularity, believed to ensure the lowest approximation errors of the solution. It can be predicted that some design velocities $\partial_{\mathbf{h}}\mathsf{\mathbf{X}}$ of a good quality primary discretization may correspond to design perturbations that significantly worsen the mesh quality. On the other hand, the perturbed response one attempts to predict with the use of the sensitivity gradient, Eq. (4.54), would otherwise be sought for with a primary analysis featuring similar quality of the finite element mesh. Thus, the answer to the risen question is that $\partial_{\mathbf{h}}\mathsf{\mathbf{X}}$ should be chosen in such a way that the perturbed finite element mesh, with the same topology as the primary one and nodal coordinates defined with Eq. (4.55), fulfills the same criteria of regularity as the one used in the primary analysis. Coming back to Fig. 4.4, the solutions b) and especially c) define a much more regular mesh perturbation (and are thus expected to yield better sensitivity solution) than the solution a), featuring a few strongly distorted elements near the perturbed rounding surface. If the perturbed structure were to be meshed and analysed independently of the primary case, it is unlikely that the mesh a) would be chosen; verification of the sensitivity solution would rather be performed with one of the remaining two meshes in Fig. 4.4

Currently, the meshes are in most cases generated automatically with the use of specialized algorithms that optimize their layouts with respect to prescribed regularity criteria. If the criteria can be written down in the form of a general scalar equation, e.g.

$$f_{\mathrm{reg}}(\mathsf{\mathbf{X}}, \mathbf{h}) = 0$$

(where the argument $\mathbf{h}$ represents all design-induced bounds imposed on at least some nodal coordinates), then the desired velocity can be determined from differentiation of this formula with respect to $\mathbf{h}$,

$$\frac{\partial f_{\mathrm{reg}}}{\partial \mathsf{X}_\beta}\, \partial_{\mathbf{h}}\mathsf{X}_\beta + \partial_{\mathbf{h}} f_{\mathrm{reg}} = \mathbf{0}.$$

Unfortunately, most mesh generators will not provide us with such data like $\partial_{\mathbf{h}} f_{\mathrm{reg}}$ or $\partial f_{\mathrm{reg}}/\partial \mathsf{X}_\beta$, however, their acquisition is possible upon implementation of the above equation in a mesh generating/optimizing algorithm. This conclusion indicates a desired direction of research in this area.

4.1.3. Enhanced formulations of FE strain approximation

The formulae for both shape and non-shape sensitivity presented in Sections 4.1.1 and, especially, 4.1.2, extensively utilize specific properties of the finite-element discretization. It is obvious that they require appropriate modifications if the enhanced interpolation of volumetric/isochoric deformation presented in Section 3.4.4 is applied. The modifications are derived and presented below.

In the non-shape sensitivity analysis the changes are not significant. Since the shape function gradients $\phi_{i\alpha,j}$ are design independent, so are their averaged values, $\tilde{\phi}_{i\alpha,j}$. Thus, similarly as in the unmodified formulation, the geometric arrays $\bar{\mathcal{B}}_{ij\alpha}$, $\bar{\mathsf{B}}_{ij\alpha}$ and $\bar{\mathsf{b}}_{ij\alpha}$ depend on design only through q_{β} , which implies

$$\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\mathcal{B}}_{ij\alpha} = \mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\mathsf{B}}_{ij\alpha} = \mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\mathsf{b}}_{ij\alpha} = \mathbf{0}.$$

The design derivatives of modified internal forces at frozen $\mathbf{q}$ are expressed, for large deformations (cf. Eqs. (4.16), (4.19)), as

$$\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\mathsf{f}}^{\mathrm{int}}_{\alpha} = \int_{\Omega^r} \mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{T}_{ij}\bar{\mathsf{B}}_{ij\alpha}\,\mathrm{d}\Omega = \int_{\Omega^r} \mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\tau}_{ij}\bar{\mathsf{b}}_{ij\alpha}\,\mathrm{d}\Omega \tag{4.56}$$

and for small deformations (cf. Eq. (4.13)) as

$$\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\mathsf{f}}^{\mathrm{int}}_{\alpha} = \int_{\Omega^r} \mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\sigma}_{ij}\bar{\mathcal{B}}_{ij\alpha}\,\mathrm{d}\Omega\,. \tag{4.57}$$

The partial stress derivatives appearing in the above equations can be determined in the same way as in the unmodified formulation, however, with ‘barred’ deformation measures taken as input, i.e.

$$\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\boldsymbol{T}} = \mathrm{d}_{\mathbf{h}!\bar{\boldsymbol{E}}}\bar{\boldsymbol{T}}, \qquad \mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\boldsymbol{\tau}} = \mathrm{d}_{\mathbf{h}!\bar{\boldsymbol{F}}}\bar{\boldsymbol{\tau}}, \qquad \mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\boldsymbol{\sigma}} = \mathrm{d}_{\mathbf{h}!\bar{\boldsymbol{\varepsilon}}}\bar{\boldsymbol{\sigma}}.$$

In the shape sensitivity analysis, Eqs. (4.56)–(4.57) have to be extended to the form

$$\begin{aligned}\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\mathsf{f}}^{\mathrm{int}}_{\alpha} &= \int_{\Omega^r} (\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{T}_{ij}\,\bar{\mathsf{B}}_{ij\alpha} + \bar{T}_{ij}\,\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\mathsf{B}}_{ij\alpha})\,\mathrm{d}\Omega \\ &= \int_{\Omega^r} (\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\tau}_{ij}\,\bar{\mathsf{b}}_{ij\alpha} + \bar{\tau}_{ij}\,\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\mathsf{b}}_{ij\alpha})\,\mathrm{d}\Omega\end{aligned} \tag{4.58}$$

for large deformations and

$$\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\mathsf{f}}^{\mathrm{int}}_{\alpha} = \int_{\Omega^r} (\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\sigma}_{ij}\,\bar{\mathcal{B}}_{ij\alpha} + \bar{\sigma}_{ij}\,\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\mathcal{B}}_{ij\alpha})\,\mathrm{d}\Omega \tag{4.59}$$

for small deformations. To derive the design derivatives of stress and geometric arrays appearing in the above formulae, we proceed as follows.

The design derivatives of averaged shape function gradients $\tilde{\phi}_{i\alpha,j}$ can be expressed as, cf. Eq. (4.33),

$$\partial_{\mathbf{h}}\tilde{\phi}_{i\alpha,j} = \left\langle \hat{\phi}_{i\alpha;l}\, \partial_{\mathbf{h}}\hat{F}^{-1}_{lj} \right\rangle = -\left\langle \phi_{i\alpha,k}\, \phi_{k\beta,j} \right\rangle\, \partial_{\mathbf{h}}\mathsf{X}_{\beta}\,, \tag{4.60}$$

where the notation $\langle(\cdot)\rangle$ denotes averaged value of $(\cdot)$ on appropriate finite element volume. Design derivatives of averaged deformation gradient are thus, cf. Eq. (4.37),

$$\mathrm{d}_{\mathbf{h}}\tilde{F}_{ij} \;=\; \tilde{\phi}_{i\beta,j}\, \mathrm{d}_{\mathbf{h}}\mathsf{q}_{\beta} - \langle u_{i,k}\, \phi_{k\beta,j}\rangle\, \partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \tag{4.61}$$

Generalizing the formula (3.184), we can write down the total design derivative of the modified deformation gradient $\bar{F}_{ij}$ as

$$\begin{aligned} \mathrm{d}_{\mathbf{h}}\bar{F}_{ij} \;&=\; \left(\frac{\tilde{J}}{J}\right)^{\frac{1}{3}} \left[\mathrm{d}_{\mathbf{h}}F_{ij} + \tfrac{1}{3}(\tilde{F}^{-1}_{kl}\, \mathrm{d}_{\mathbf{h}}\tilde{F}_{lk} - F^{-1}_{kl}\, \mathrm{d}_{\mathbf{h}}F_{lk})F_{ij}\right] \\ &=\; \left(\frac{\tilde{J}}{J}\right)^{\frac{1}{3}} \left[(\phi_{i\beta,j} + \mathsf{g}_{\beta}F_{ij})\, \mathrm{d}_{\mathbf{h}}\mathsf{q}_{\beta} - (u_{i,k}\, \phi_{k\beta,j} - \breve{\mathsf{g}}_{\beta}F_{ij})\, \partial_{\mathbf{h}}\mathsf{X}_{\beta}\right], \end{aligned} \tag{4.62}$$

where g_{β} is given by Eq. (3.185) and (utilizing $u_{i,j} = F_{ij} - \delta_{ij}$)

$$\begin{aligned} \breve{\mathsf{g}}_{\beta} \;&=\; -\tfrac{1}{3}\left(\tilde{F}^{-1}_{kl}\langle u_{l,m}\, \phi_{m\beta,k}\rangle - F^{-1}_{kl}u_{l,m}\, \phi_{m\beta,k}\right) \\ &=\; -\tfrac{1}{3}\left(\tilde{F}^{-1}_{kl}\langle F_{lm}\, \phi_{m\beta,k}\rangle - \tilde{F}^{-1}_{kl}\, \tilde{\phi}_{l\beta,k} - F^{-1}_{kl}F_{lm}\, \phi_{m\beta,k} + F^{-1}_{kl}\, \phi_{l\beta,k}\right) \\ &=\; -\tfrac{1}{3}\Big(\underbrace{\tilde{F}^{-1}_{kl}\langle F_{lm}\, \phi_{m\beta,k}\rangle}_{\approx\, \tilde{\phi}_{k\beta,k}} - \tilde{\mathsf{b}}_{kk\beta} - \phi_{k\beta,k} + \mathsf{b}_{kk\beta}\Big) \approx \mathsf{g}_{\beta} - \mathsf{g}^{\ell}_{\beta}\,. \end{aligned}$$

From Eq. (4.62) we can easily conclude that

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{F}_{ij} = \left(\frac{\tilde{J}}{J}\right)^{\frac{1}{3}} (-u_{i,k}\, \phi_{k\beta,j} + \breve{\mathsf{g}}_{\beta}F_{ij})\, \partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \tag{4.63}$$

To derive total and partial (at frozen $\mathbf{q}$) design derivatives of modified geometric arrays $\bar{\mathsf{B}}_{ij\alpha}$, $\bar{\mathsf{b}}_{ij\alpha}$ and $\bar{\mathcal{B}}_{ij\alpha}$, let us first differentiate the expressions for $\tilde{\mathsf{b}}_{ij\alpha}$,

g_α and υ, cf. definitions (3.185) and (3.187),

$$\begin{aligned}
\mathrm{d}_{\mathbf{h}}\tilde{\mathsf{b}}_{ij\alpha} &= \partial_{\mathbf{h}}\tilde{\phi}_{i\alpha,k}\,\tilde{F}^{-1}_{kj} - \tilde{\phi}_{i\alpha,k}\tilde{F}^{-1}_{kl}\,\mathrm{d}_{\mathbf{h}}\tilde{F}_{lm}\,\tilde{F}^{-1}_{mj} \\
&= -\tilde{\mathsf{b}}_{kj\alpha}\tilde{\mathsf{b}}_{kj\beta}\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta - \Big[\langle\phi_{i\alpha,n}\phi_{n\beta,m}\rangle - \tilde{\phi}_{i\alpha,k}\tilde{F}^{-1}_{kl}\langle u_{l,n}\phi_{n\beta,m}\rangle\Big]\tilde{F}^{-1}_{mj}\,\partial_{\mathbf{h}}\mathsf{X}_\beta \\
&= -\tilde{\mathsf{b}}_{kj\alpha}\tilde{\mathsf{b}}_{kj\beta}\,(\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta + \partial_{\mathbf{h}}\mathsf{X}_\beta) \\
&\qquad - \underbrace{\Big[\langle\phi_{i\alpha,n}\phi_{n\beta,m}\rangle - \tilde{\phi}_{i\alpha,k}\tilde{F}^{-1}_{kl}\langle F_{ln}\phi_{n\beta,m}\rangle\Big]}_{\approx\,0}\,\tilde{F}^{-1}_{mj}\,\partial_{\mathbf{h}}\mathsf{X}_\beta\,, \\
\mathrm{d}_{\mathbf{h}}\mathsf{g}_\alpha &= -\mathsf{h}_{\alpha\beta}\,(\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta + \partial_{\mathbf{h}}\mathsf{X}_\beta) \\
&\qquad - \tfrac{1}{3}\Big[\langle\phi_{i\alpha,n}\phi_{n\beta,m}\rangle - \tilde{\phi}_{i\alpha,k}\tilde{F}^{-1}_{kl}\langle F_{ln}\phi_{n\beta,m}\rangle\Big]\,\tilde{F}^{-1}_{mj}\,\partial_{\mathbf{h}}\mathsf{X}_\beta\,, \\
\mathrm{d}_{\mathbf{h}}\upsilon &= \tfrac{2}{3}\upsilon\,\Big(\tilde{F}^{-1}_{ij}\,\mathrm{d}_{\mathbf{h}}\tilde{F}_{ji} - F^{-1}_{ij}\,\mathrm{d}_{\mathbf{h}}F_{ji}\Big) \\
&= \tfrac{2}{3}\upsilon\,\Big[(\tilde{\mathsf{b}}_{ii\beta} - \mathsf{b}_{ii\beta})\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta - \Big(\tilde{F}^{-1}_{ij}\langle u_{j,k}\phi_{k\beta,i}\rangle - F^{-1}_{ij}u_{j,k}\phi_{k\beta,i}\Big)\,\partial_{\mathbf{h}}\mathsf{X}_\beta\Big] \\
&= 2\upsilon\mathsf{g}_\beta\,\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta + 2\upsilon\breve{\mathsf{g}}_\beta\,\partial_{\mathbf{h}}\mathsf{X}_\beta\,, \\
\mathrm{d}_{\mathbf{h}}\mathsf{g}^{\ell}_\alpha &\equiv \partial_{\mathbf{h}}\mathsf{g}^{\ell}_\alpha = -\tfrac{1}{3}\,(\langle\phi_{i\alpha,k}\,\phi_{k\beta,j}\rangle - \phi_{i\alpha,k}\,\phi_{k\beta,j})\,\partial_{\mathbf{h}}\mathsf{X}_\beta\,.
\end{aligned}$$

The derivatives $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\tilde{\mathsf{b}}_{ij\alpha}$, $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{g}_\alpha$ and $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\upsilon$ can be obtained from the above formulae upon only setting $\mathrm{d}_{\mathbf{h}}\mathsf{q}_\beta = \mathbf{0}$.

Now, differentiation of Eqs. (3.188), (3.189) and (3.192) leads to the formulae

$$\begin{aligned}
\mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{\mathsf{B}}_{ij\alpha} &= \upsilon\,\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{B}_{ij\alpha} + \mathrm{d}_{\mathbf{h}!\mathsf{q}}\upsilon\,\mathsf{B}_{ij\alpha} \\
&\qquad + \mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{g}_\alpha\,\bar{F}_{ki}\bar{F}_{kj} + 2\mathsf{g}_\alpha\,\mathrm{sym}(\bar{F}_{ki}\,\mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{F}_{kj})\,, \qquad (4.64) \\
\mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{\mathsf{b}}_{ij\alpha} &= \mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{b}_{ij\alpha} + \mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{g}_\alpha\,\delta_{ij}\,, \qquad (4.65) \\
\mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{\mathcal{B}}_{ij\alpha} &= \mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathcal{B}_{ij\alpha} + \partial_{\mathbf{h}}\mathsf{g}^{\ell}_\alpha\,\delta_{ij}\,, \qquad (4.66)
\end{aligned}$$

in which the unmodified derivatives $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{B}_{ij\alpha}$, $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathsf{b}_{ij\alpha}$, and $\mathrm{d}_{\mathbf{h}!\mathsf{q}}\mathcal{B}_{ij\alpha}$ are taken from Eqs. (4.44), (4.48) and (4.41), respectively.

The derivatives of modified stress appearing in Eqs. (4.58)–(4.59) can be determined in exactly the same way as for unmodified formulation, i.e. from Eqs. (4.43), (4.47) and (4.53), with all strain and deformation tensors replaced with their modified counterparts, i.e.

$$\begin{aligned}
\mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{\boldsymbol{\sigma}} &= \mathrm{d}_{\mathbf{h}!\bar{\varepsilon}}\bar{\boldsymbol{\sigma}} + \boldsymbol{\mathcal{C}} : \mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{\boldsymbol{\varepsilon}}\,, \qquad (4.67) \\
\mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{\boldsymbol{T}} &= \mathrm{d}_{\mathbf{h}!\bar{\boldsymbol{E}}}\bar{\boldsymbol{T}} + \bar{\boldsymbol{\mathfrak{C}}} : \mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{\boldsymbol{E}}\,, \qquad (4.68) \\
\mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{\boldsymbol{\tau}} &= \mathrm{d}_{\mathbf{h}!\bar{\boldsymbol{F}}}\bar{\boldsymbol{\tau}} + \frac{\partial\bar{\boldsymbol{\tau}}}{\partial\bar{\boldsymbol{F}}} : \mathrm{d}_{\mathbf{h}!\mathsf{q}}\bar{\boldsymbol{F}}\,, \qquad (4.69)
\end{aligned}$$

where

$$\begin{aligned} \mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{\varepsilon}_{ij} &= \operatorname{sym}(\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{u}_{i,j})\,, \\ \mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{u}_{i,j} &= -\left[u_{i,k}\phi_{k\beta,j} + \tfrac{1}{3}\left(\langle u_{k,l}\phi_{l\beta,k}\rangle - u_{k,l}\phi_{l\beta,k}\right)\delta_{ij}\right]\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,, \\ \mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{E}_{ij} &= \operatorname{sym}(\bar{F}_{ki}\,\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{F}_{kj})\,, \end{aligned}$$

and $\mathrm{d}_{\mathbf{h}!\mathbf{q}}\bar{F}_{kj}$ is given by Eq. (4.63).

Summarizing the contents of Section 4.1, the sensitivity computations for the global equilibrium problem at a particular time instant of the assumed time integration scheme consist in assembling the sensitivity r.h.s. array $\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{res}}$ given explicit design derivatives of known input data, solving the system of equations with the coefficient matrix $\mathbf{K}$ against each column of this array (corresponding to each particular design parameter) to obtain the array $\mathrm{d}_{\mathbf{h}}\mathbf{q}$, and, finally, updating the sensitivities of the discrete state variable array $\mathrm{d}_{\mathbf{h}}\mathbf{z}$. All the computations have to be performed right after completion of the primary equilibrium analysis for this step, but before update of the state variable array $\mathbf{z}$ (which can be finally done simultaneously with the update of $\mathrm{d}_{\mathbf{h}}\mathbf{z}$). Detailed formulae for some quantities necessary for the assembly of $\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{res}}$, like $\mathrm{d}_{\mathbf{h}!\mathbf{q}}\tilde{\boldsymbol{\sigma}}$, as well as update schemes for sensitivities of constitutive state parameters $\mathrm{d}_{\mathbf{h}}\boldsymbol{z}_i$, can only be derived from design-differentiation of particular constitutive equations. The following section will be devoted to this subject.

Let us complete the considerations of this Section by one important remark. The entire derivation of the sensitivity formulation for the discrete problem of nonlinear statics is based on the assumption that the primary solution of the system (4.1) is known at a considered time instant. On the other hand, we know from Chapter 3 that this solution is sought for in an iterative scheme, cf. Eq. (3.176), and is only known up to certain accuracy, depending on the assumed convergence tolerance. This primary convergence error obviously affects the quality of the sensitivity solution. We will refer to this issue in Section 4.3 while discussing results of computational examples.

4.2. Design-differentiation of time-discrete constitutive equations

In this section, the constitutive algorithms presented in Section 3.3, will be analytically differentiated with respect to design $\mathbf{h}$ at a typical time step $[t_n\,,\,t_{n+1}]$. As a result, updated design derivatives of stress $\mathrm{d}_{\mathbf{h}}\tilde{\boldsymbol{\sigma}}_{n+1}$ and a set of appropriate constitutive state fields $\mathrm{d}_{\mathbf{h}}\mathbf{z}_{n+1}$ will be derived. The former ones, $\mathrm{d}_{\mathbf{h}}\tilde{\boldsymbol{\sigma}}_{n+1}$, do not explicitly appear in the formulation presented in the previous section, but they will be investigated for three reasons: (i) as it will be demonstrated, the forms of the explicit derivatives $\mathrm{d}_{\mathbf{h}!\mathbf{q}}\tilde{\boldsymbol{\sigma}}_{n+1}$ can be easily derived from the forms of the total derivatives $\mathrm{d}_{\mathbf{h}}\tilde{\boldsymbol{\sigma}}_{n+1}$, (ii) the system performances $\mathbf{G}$ frequently include (or depend on) stress components or invariants, in which case the stress sensitivities are finally required anyway, and (iii) anticipating discussion in Chapter 5, we mention here that total, rather than explicit design derivatives of stress, are necessary in the sensitivity analysis of explicit formulations of dynamics.

In other words, we are starting from the end, i.e. from the derivation of update equations on $\mathrm{d}_{\mathbf{h}}\mathbf{z}_{n+1} = \{\mathrm{d}_{\mathbf{h}}\boldsymbol{z}_{i\ n+1}\}$ and $\mathrm{d}_{\mathbf{h}}\tilde{\boldsymbol{\sigma}}_{n+1}$ for the already known $\mathrm{d}_{\mathbf{h}}\mathbf{q}_{n+1}$ (and thus also $\mathrm{d}_{\mathbf{h}}\boldsymbol{F}_{n+1}$). The subject of this section is thus to solve the following problem at a time instant t_{n+1} ,

Given:

- deformation $\boldsymbol{F}$ and constitutive state parameters $\boldsymbol{z}_i$ along with their sensitivities $\partial_{\mathbf{h}}\boldsymbol{F}$ and $\partial_{\mathbf{h}}\boldsymbol{z}_i$ at $t = t_n$,
- deformation $\boldsymbol{F}$ and its sensitivity $\mathrm{d}_{\mathbf{h}}\boldsymbol{F}$ at $t = t_{n+1}$,
- time-discretized constitutive equations in the interval $[t_n\, ,\, t_{n+1}]$, with coefficients explicitly dependent on design,

compute:

- sensitivity of stress $\mathrm{d}_{\mathbf{h}}\tilde{\boldsymbol{\sigma}}$ and constitutive state parameters $\mathrm{d}_{\mathbf{h}}\boldsymbol{z}_i$ at $t = t_{n+1}$,

with $\tilde{\boldsymbol{\sigma}}$ understood as $\boldsymbol{T}$, $\boldsymbol{\tau}$ or $\boldsymbol{\sigma}$, respectively, depending on the geometric formulation. Evaluation of explicit derivatives $\mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\tilde{\boldsymbol{\sigma}}$ computed at unknown $\mathrm{d}_{\mathbf{h}}\mathbf{q}_{n+1}$ (and aimed at assembly of the r.h.s. arrays for its evaluation, cf. Eq. (4.3)) will be discussed later in this section, for particular constitutive formulations.

The basis for the above general formulation is the primary problem presented in p. 57. They both can be solved independently, however, as it will be shown, several intermediate quantities computed while solving the primary one, can be useful in the sensitivity computations, too.

4.2.1. Small deformation formulation

Design differentiation of the incremental constitutive equations presented in Section 3.3.1, finally leading to determine $\mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma}_{n+1}$ and $\mathrm{d}_{\mathbf{h}}\mathbf{z}_{n+1}$ might be considered as a trivial task consisting in differentiation step by step of subsequent steps of the algorithm displayed in Tables 3.1–3.3. Indeed, such a procedure would lead to the required result, suffering, however, from numerical inefficiency.

The reason for this is Eq. (3.111), the nonlinear equation with respect to $\Delta\bar{e}^{\mathrm{p}}$. Differentiating it with respect to design $\mathbf{h}$ one obtains

$$\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}\,\mathrm{d}_{\mathbf{h}}(\Delta\bar{e}^{\mathrm{p}}) = -\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\tilde{f} \tag{4.70}$$

where $\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}$ (the full derivative of $\tilde{f}$, including dependence of all its arguments on $\Delta\bar{e}^{\mathrm{p}}$) is given by Eqs. (3.117) or (3.118), depending on particular plasticity formulation. Equation (4.70) is a linear equation for $\mathrm{d}_{\mathbf{h}}(\Delta\bar{e}^{\mathrm{p}})$ and, contrary to the primary algorithm, no iteration is required to solve it. Design differentiation of equations in Table 3.2 makes thus no sense. Instead, differentiation of Eqs. (3.112) or (3.113) at frozen $\Delta\bar{e}^{\mathrm{p}}$ is necessary to compute the term $\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\tilde{f}$ in the right-hand side of Eq. (4.70)

Before the derivation can be displayed, let us make a remark about material constants and constitutive functions appearing in the formulations of Chapter 3. It is assumed that material constants are explicitly design-dependent, i.e. their design derivatives, e.g. $\partial_{\mathbf{h}}E$, $\partial_{\mathbf{h}}\nu$, $\partial_{\mathbf{h}}m$, ..., are known data in the same sense as their primary values E, ν, $m, \ldots$ are. Design derivatives of dependent constants can be easily determined from the independent ones, e.g.

$$\partial_{\mathbf{h}}G = G\left[\frac{\partial_{\mathbf{h}}E}{E} - \frac{\partial_{\mathbf{h}}\nu}{1+\nu}\right], \qquad \partial_{\mathbf{h}}K = K\left[\frac{\partial_{\mathbf{h}}E}{E} + \frac{2\partial_{\mathbf{h}}\nu}{1-2\nu}\right]$$

(cf. Eqs. (3.27)). Constitutive functions, like hardening functions κ and H, are given functions of the state variable $\bar{e}^{\mathrm{p}}$, treated as material data, too. Particular forms of the functions have not been discussed, let us only mention that, apart of the argument $\bar{e}^{\mathrm{p}}$, their values depend on a number of material constants which are again explicitly design-dependent input data. Thus, we can write $\kappa = \kappa(\bar{e}^{\mathrm{p}};\mathbf{h})$, $H = H(\bar{e}^{\mathrm{p}};\mathbf{h})$ and express their derivatives as

$$\mathrm{d}_{\mathbf{h}}\kappa = \kappa'\mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} + \partial_{\mathbf{h}}\kappa\,, \qquad \mathrm{d}_{\mathbf{h}}H = H'\mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} + \partial_{\mathbf{h}}H\,,$$

where the terms $\partial_{\mathbf{h}}\kappa$, $\partial_{\mathbf{h}}H$ contain explicit design derivatives of the constants. For instance, if

$$\kappa(\bar{e}^{\mathrm{p}}) = \sigma_{y0} + \bar{\kappa}'\bar{e}^{\mathrm{p}} + (\sigma_{y\infty} - \sigma_{y0})\left(1 - e^{-a\bar{e}^{\mathrm{p}}}\right) \tag{4.71}$$

where σ_{y0} , $\sigma_{y\infty}$, $\bar{\kappa}'$, a, are material constants, then

$$\begin{aligned}\partial_{\mathbf{h}}\kappa \;=\; & \partial_{\mathbf{h}}\sigma_{y0} + \partial_{\mathbf{h}}\bar{\kappa}'\,\bar{e}^{\mathrm{p}} + (\mathrm{d}_{\mathbf{h}}\sigma_{y\infty} - \mathrm{d}_{\mathbf{h}}\sigma_{y0})\left(1 - e^{-a\bar{e}^{\mathrm{p}}}\right) \\ & + \partial_{\mathbf{h}}a\,\bar{e}^{\mathrm{p}}\,(\sigma_{y\infty} - \sigma_{y0})\,e^{-a\bar{e}^{\mathrm{p}}}\end{aligned}$$

while

$$\kappa' = \bar{\kappa}' + (\sigma_{y\infty} - \sigma_{y0})\,a\,e^{-a\bar{e}^{\mathrm{p}}}\,.$$

Now, since $\tilde{f}$ can be generally presented as a function $\tilde{f}(\bar{\sigma}_{n+1}, \bar{e}^{\mathrm{p}}_{n+1}, \Delta\bar{e}^{\mathrm{p}}; \mathbf{h})$, its derivative at frozen $\Delta\bar{e}^{\mathrm{p}}$ has the form

$$\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,\tilde{f} = \frac{\partial\tilde{f}}{\partial\bar{\sigma}_{n+1}}\,\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,\bar{\sigma}_{n+1} + \frac{\partial\tilde{f}}{\partial\bar{e}^{\mathrm{p}}_{n+1}}\,\partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} + \partial_{\mathbf{h}}\tilde{f}_{n+1} \tag{4.72}$$

where, cf. Eq. (3.110),

$$\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,\bar{\sigma}_{n+1} = \mathrm{d}_{\mathbf{h}}\bar{\sigma}_{\mathrm{trial}} - \Delta\bar{e}^{\mathrm{p}}\,\left[\partial_{\mathbf{h}}(3G) + \mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,H'_{n+1}\right] \tag{4.73}$$

and

$$\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,H'_{n+1} = H''_{n+1}\,\partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} + \partial_{\mathbf{h}}H'_{n+1}\,. \tag{4.74}$$

In particular, for the rate-independent constitutive model, cf. Eq. (3.112), we have

$$\frac{\partial\tilde{f}}{\partial\bar{\sigma}_{n+1}} = 1, \qquad \frac{\partial\tilde{f}}{\partial\bar{e}^{\mathrm{p}}_{n+1}} = -\kappa', \qquad \partial_{\mathbf{h}}\tilde{f} = \partial_{\mathbf{h}}\kappa\,, \tag{4.75}$$

and we finally obtain

$$\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,\tilde{f} = \mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,\bar{\sigma}_{n+1} - \kappa'_{n+1}\,\partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} - \partial_{\mathbf{h}}\kappa_{n+1}\,. \tag{4.76}$$

For rate-dependent models, cf. Eq. (3.113), we have

$$\begin{aligned} \mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,\tilde{f} &= \Delta t\,\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,g_{n+1} \\ &= \Delta t\,\left[(g_{,\bar{\sigma}})_{n+1}\,\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,\bar{\sigma}_{n+1} + (g_{,\bar{e}^{\mathrm{p}}})_{n+1}\,\partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} + \partial_{\mathbf{h}}g_{n+1}\right] \end{aligned} \tag{4.77}$$

where

- for the overstress plasticity, cf. Eq. (3.59),

$$g_{,\bar{\sigma}} = \frac{gm}{\bar{\sigma}-\kappa}\,, \qquad g_{,\bar{e}^{\mathrm{p}}} = -g_{,\bar{\sigma}}\,\frac{\bar{\sigma}\kappa'}{\kappa}\,, \tag{4.78a}$$

$$\partial_{\mathbf{h}}g = -g\left[\frac{m\,\bar{\sigma}}{\bar{\sigma}-\kappa}\,\frac{\partial_{\mathbf{h}}\kappa}{\kappa} + \frac{\partial_{\mathbf{h}}\mu}{\mu} - \ln\!\left(\frac{\bar{\sigma}-\kappa}{\kappa}\right)\partial_{\mathbf{h}}m\right], \tag{4.78b}$$

- for the power-law strain and strain-rate hardening viscoplasticity, cf. Eq. (3.61),

$$g_{,\bar{\sigma}} = \frac{g}{m\bar{\sigma}}\,, \qquad g_{,\bar{e}^{\mathrm{p}}} = -\frac{g}{mn(\bar{e}^{\mathrm{p}}+\varepsilon_0)}\,, \tag{4.79a}$$

$$\begin{aligned} \partial_{\mathbf{h}}g = g\,&\left[\frac{\partial_{\mathbf{h}}\dot{\varepsilon}_0}{\dot{\varepsilon}_0} - \ln\!\left(\frac{\bar{\sigma}}{g_1}\right)\frac{\partial_{\mathbf{h}}m}{m^2} - \frac{\partial_{\mathbf{h}}E}{mE}\right. \\ &\left. - \left(1 - \frac{\bar{e}^{\mathrm{p}}}{n(\bar{e}^{\mathrm{p}}+\varepsilon_0)}\right)\frac{\partial_{\mathbf{h}}\varepsilon_0}{m\varepsilon_0} + \ln\!\left(1+\frac{\bar{e}^{\mathrm{p}}}{\varepsilon_0}\right)\frac{\partial_{\mathbf{h}}n}{mn^2}\right]. \end{aligned} \tag{4.79b}$$

Table 4.2. Sensitivity algorithm for small-deformation elasto-plasticity (for the sake of legibility, subscripts $(n+1)$ have been omitted)

```
Given
    strain $\boldsymbol{\varepsilon}_n$ and state parameters $\mathbf{z}_n = \{\,\mathrm{dev}\,\boldsymbol{\sigma}_n\,,\ \mathrm{dev}\,\boldsymbol{\alpha}_n\,,\ \bar{e}^{\mathrm{p}}_n\,\}$ at $t_n$ and their
    design derivatives, $\partial_{\mathbf{h}}\boldsymbol{\varepsilon}_n\,,\ \partial_{\mathbf{h}}\mathbf{z}_n = \{\,\partial_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\sigma}_n)\,,\ \partial_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\alpha}_n)\,,\ \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n\,\}$,
    strain $\boldsymbol{\varepsilon}$ at $t_{n+1}$ and its design derivatives $\mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon}$ $(\mathrm{d}_{\mathbf{h}}\Delta\boldsymbol{\varepsilon} = \mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon} - \partial_{\mathbf{h}}\boldsymbol{\varepsilon}_n)$,
    material constants $(E,\ \nu,\ \ldots)$ and their design derivatives
    intermediate results of primary computations (Tables 3.1–3.3),
compute
    design derivatives of state parameters $\mathrm{d}_{\mathbf{h}}\mathbf{z} = \{\,\mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\sigma})\,,\ \mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\alpha})\,,\ \mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}}\,\}$
    and stress $\mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma}$ at $t_{n+1}$
using the following scheme:
1. $\mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\sigma}_{\mathrm{trial}}) = \partial_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\sigma}_n) + \partial_{\mathbf{h}}(2G)\,\mathrm{dev}\,\Delta\boldsymbol{\varepsilon} + 2G\,\mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\Delta\boldsymbol{\varepsilon})$
if (plastic flow) then
    2. $\mathrm{d}_{\mathbf{h}}\boldsymbol{s}_{\mathrm{trial}} = \mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\sigma}_{\mathrm{trial}}) - \partial_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\alpha}_n)$,   $\mathrm{d}_{\mathbf{h}}\bar{\sigma}_{\mathrm{trial}} = \frac{3}{2\bar{\sigma}_{\mathrm{trial}}}\,\boldsymbol{s}_{\mathrm{trial}} : \mathrm{d}_{\mathbf{h}}\boldsymbol{s}_{\mathrm{trial}}$
    3. $\mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}},\ \mathrm{d}_{\mathbf{h}}H' \quad \longleftarrow \quad$ from Table 4.3
    4. $\mathrm{d}_{\mathbf{h}}\Delta\boldsymbol{\varepsilon}^{\mathrm{p}} = \Delta\boldsymbol{\varepsilon}^{\mathrm{p}}\left(\frac{\mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}}}{\Delta\bar{e}^{\mathrm{p}}} - \frac{\mathrm{d}_{\mathbf{h}}\bar{\sigma}_{\mathrm{trial}}}{\bar{\sigma}_{\mathrm{trial}}}\right) + \frac{\vartheta}{2G}\,\mathrm{d}_{\mathbf{h}}\boldsymbol{s}_{\mathrm{trial}}$
    5. $\mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\sigma}) = \mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\sigma}_{\mathrm{trial}}) - \partial_{\mathbf{h}}(2G)\,\Delta\boldsymbol{\varepsilon}^{\mathrm{p}} - 2G\,\mathrm{d}_{\mathbf{h}}\Delta\boldsymbol{\varepsilon}^{\mathrm{p}}$
       $\mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\alpha}) = \partial_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\alpha}_n) + \frac{2}{3}\,\mathrm{d}_{\mathbf{h}}H'\,\Delta\boldsymbol{\varepsilon}^{\mathrm{p}} + \frac{2}{3}\,H'\,\mathrm{d}_{\mathbf{h}}\Delta\boldsymbol{\varepsilon}^{\mathrm{p}}$
       $\mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} = \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n + \mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}}$
else (purely elastic deformation)
    6. $\mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\sigma}) = \mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\sigma}_{\mathrm{trial}})\,,\quad \mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\alpha}) = \partial_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\alpha}_n)\,,\quad \mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} = \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n$
end if
7. $\mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma} = \mathrm{d}_{\mathbf{h}}(\mathrm{dev}\,\boldsymbol{\sigma}) + [\partial_{\mathbf{h}}K\,\mathrm{tr}\boldsymbol{\varepsilon} + K\,\mathrm{d}_{\mathbf{h}}(\mathrm{tr}\boldsymbol{\varepsilon})]\,\boldsymbol{I}$
```

Table 4.3. Computation of $\mathrm{d}_{\mathbf{h}}(\Delta\bar{e}^{\mathrm{p}})$ for different elasto-plastic and elasto-viscoplastic models

```
1.  $\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, H' = H''\, \partial_{\mathbf{h}} \bar{e}^{\mathrm{p}}_n + \partial_{\mathbf{h}} H'$
2.  $\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, \bar{\sigma} \quad \longleftarrow$   from Eq. (4.73) (small deformations)
                                   or Eq. (4.87) (large deformations)
                                   or Eq. (4.93) (1-d bar formulation)

if (rate-independent plasticity)

    3a. $\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, \tilde{f} = \mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, \bar{\sigma} - \kappa'\, \mathrm{d}_{\mathbf{h}} \bar{e}^{\mathrm{p}}_n - \partial_{\mathbf{h}} \kappa$

else if (overstress viscoplasticity)

    3b. $g_{,\bar{\sigma}}\,,\ g_{,\bar{e}^{\mathrm{p}}}\,,\ \partial_{\mathbf{h}} g \quad \longleftarrow$   from Eqs (4.78)
    4b. $\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, \tilde{f} = \Delta t\ (g_{,\bar{\sigma}}\, \mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, \bar{\sigma} + g_{,\bar{e}^{\mathrm{p}}}\, \partial_{\mathbf{h}} \bar{e}^{\mathrm{p}}_n + \partial_{\mathbf{h}} g)$

else if (power-law hardening viscoplasticity)

    3c. $g_{,\bar{\sigma}}\,,\ g_{,\bar{e}^{\mathrm{p}}}\,,\ \partial_{\mathbf{h}} g \quad \longleftarrow$   from Eqs (4.79)
    4c. $\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, \tilde{f} = \Delta t\ (g_{,\bar{\sigma}}\, \mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, \bar{\sigma} + g_{,\bar{e}^{\mathrm{p}}}\, \partial_{\mathbf{h}} \bar{e}^{\mathrm{p}}_n + \partial_{\mathbf{h}} g)$

end if

5.  $\mathrm{d}_{\mathbf{h}} \Delta\bar{e}^{\mathrm{p}} = -\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}} \tilde{f} / \tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}$
    $\mathrm{d}_{\mathbf{h}} H' = \mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, H' + H''\, \mathrm{d}_{\mathbf{h}} \Delta\bar{e}^{\mathrm{p}}$
```

The algorithm for constitutive sensitivity computations in small-deformation elasto-plasticity is displayed in Table 4.2. This is a linear algorithm in which design gradients of intermediate quantities appearing in the primary constitutive formulation are computed, cf. Tables 3.1–3.3. Values of the quantities themselves are frequently reused here, thus, in real sensitivity computations it is most efficient to execute both the algorithms, primary and sensitivity, simultaneously, so that the latter can on hand access all results of the former. At the stage of the update of results at the end of the step computations, this is not really a limiting restriction, as both $\mathbf{z}$ and $\mathrm{d}_{\mathbf{h}}\mathbf{z}$ are updated at the same time.

Looking at the algorithm, we can see that the finally computed stress sensitivity $\mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma}_{n+1}$ conforms to the general form (4.14) employed in in Section 4.1. Indeed, the analysis of equations in Tables 4.2–4.3 leads to the following expressions for the derivatives $\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \boldsymbol{\varepsilon}_n}$, $\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \boldsymbol{z}_{i\,n}}$, and $\partial_{\mathbf{h}}\boldsymbol{\sigma}_{n+1}$, appearing in the formulation (4.14),

$$\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \boldsymbol{\varepsilon}_n} = \boldsymbol{\mathcal{C}}\,, \tag{4.80}$$

$$\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial (\operatorname{dev}\boldsymbol{\sigma}_n)} = (1-\vartheta)\,\boldsymbol{\mathfrak{I}} - (\gamma-\vartheta)\,\boldsymbol{n}\otimes\boldsymbol{n}\,, \tag{4.81}$$

$$\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial (\operatorname{dev}\boldsymbol{\alpha}_n)} = \vartheta\,\boldsymbol{\mathfrak{I}} + (\gamma-\vartheta)\,\boldsymbol{n}\otimes\boldsymbol{n}\,, \tag{4.82}$$

$$\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \bar{e}^{\mathrm{p}}_n} = (\tilde{\kappa}'_{n+1} + H''_{n+1})\,\gamma\sqrt{\tfrac{2}{3}}\,\boldsymbol{n}\,, \tag{4.83}$$

$$\begin{aligned}\partial_{\mathbf{h}}\boldsymbol{\sigma}_{n+1} &= \partial_{\mathbf{h}}(2G)\left[(1-\vartheta)\operatorname{dev}(\Delta\boldsymbol{\varepsilon}) - (\gamma-\vartheta)(\boldsymbol{n}:\Delta\boldsymbol{\varepsilon})\,\boldsymbol{n} - (1-\gamma)\Delta\boldsymbol{\varepsilon}^{\mathrm{p}}\right] \\ &\quad + \partial_{\mathbf{h}}K\operatorname{tr}\boldsymbol{\varepsilon}_{n+1}\,\boldsymbol{I} + \left(\Delta\bar{e}^{\mathrm{p}}\,\partial_{\mathbf{h}}H'_{n+1} - \frac{\partial_{\mathbf{h}}\tilde{f}}{(\partial\tilde{f}/\partial\bar{\sigma}_{n+1})}\right)\sqrt{\tfrac{2}{3}}\,\gamma\,\boldsymbol{n}\,,\end{aligned} \tag{4.84}$$

where

$$\tilde{\kappa}'_{n+1} = \begin{cases} \kappa'_{n+1} & \text{for rate-independent plasticity,} \\ (g_{,\bar{e}^{\mathrm{p}}}/g_{,\bar{\sigma}})_{n+1} & \text{for rate-dependent plasticity,} \end{cases}$$

and, cf. Eqs.(4.75)$_3$, (4.78b), and (4.79b),

$$\frac{\partial_{\mathbf{h}}\tilde{f}}{(\partial\tilde{f}/\partial\bar{\sigma}_{n+1})} = \begin{cases} -\partial_{\mathbf{h}}\kappa_{n+1} & \text{(rate-independent),} \\[1ex] -\bar{\sigma}_{n+1}\,\dfrac{\partial_{\mathbf{h}}\kappa_{n+1}}{\kappa_{n+1}} - (\bar{\sigma}_{n+1}{-}\kappa_{n+1})\left[\dfrac{\partial_{\mathbf{h}}\mu}{m\mu}\right. & \\ \qquad\qquad \left. - \ln\left(\dfrac{\bar{\sigma}_{n+1}{-}\kappa_{n+1}}{\kappa_{n+1}}\right)\dfrac{\partial_{\mathbf{h}}m}{m}\right] & \text{(overstress),} \\[1ex] \bar{\sigma}_{n+1}\left[m\,\dfrac{\partial_{\mathbf{h}}\dot{\varepsilon}_0}{\dot{\varepsilon}_0} - \ln\!\left(\dfrac{\bar{\sigma}_{n+1}}{g_1(\bar{e}^{\mathrm{p}}_{n+1})}\right)\dfrac{\partial_{\mathbf{h}}m}{m} - \dfrac{\partial_{\mathbf{h}}E}{E}\right. & \\ \quad \left. - \left(1 - \dfrac{\bar{e}^{\mathrm{p}}_{n+1}}{n(\bar{e}^{\mathrm{p}}_{n+1}{+}\varepsilon_0)}\right)\dfrac{\partial_{\mathbf{h}}\varepsilon_0}{\varepsilon_0} + \ln\!\left(1{+}\dfrac{\bar{e}^{\mathrm{p}}_{n+1}}{\varepsilon_0}\right)\dfrac{\partial_{\mathbf{h}}n}{n^2}\right] & \\ & \text{(power-law).} \end{cases}$$

In a similar manner we could extract from the algorithm detailed expressions for analogous derivatives of state variables $\mathbf{z}$, i.e. $\frac{\partial \boldsymbol{z}_{i\,n+1}}{\partial \boldsymbol{\varepsilon}_n}$, $\frac{\partial \boldsymbol{z}_{i\,n+1}}{\partial \boldsymbol{z}_{i\,n}}$, $\partial_{\mathbf{h}}\boldsymbol{z}_{i\,n+1}$ for $\boldsymbol{z}_1 = \operatorname{dev}\boldsymbol{\sigma}$, $\boldsymbol{z}_2 = \operatorname{dev}\boldsymbol{\alpha}$, $\boldsymbol{z}_3 = \bar{e}^{\mathrm{p}}$. Note, however, that their explicit forms are not necessary in the presented formulation. Implementation of the formulae displayed in Tables 4.2–4.3 is sufficient to compute the final result without explicit evaluation of the derivatives (4.80)–(4.84).

Finally, let us focus our attention to computation of explicit stress derivatives $\mathrm{d}_{\mathbf{h}!\mathsf{q}_{n+1}}\boldsymbol{\sigma}_{n+1}$ or $\mathrm{d}_{\mathbf{h}!\boldsymbol{\varepsilon}_{n+1}}\boldsymbol{\sigma}_{n+1}$, cf. Eq. (4.15) or (4.43), necessary to evaluate the right-hand side array $\hat{\partial}_{\mathbf{h}}\mathsf{f}^{\mathrm{res}}_{n+1} = \mathrm{d}_{\mathbf{h}!\mathsf{q}_{n+1}}\mathsf{f}^{\mathrm{res}}_{n+1}$ in the DSA system (4.3). It is easy to see that, for design-independent geometry, the derivatives $\mathrm{d}_{\mathbf{h}!\mathsf{q}_{n+1}}\boldsymbol{\sigma}_{n+1} = \mathrm{d}_{\mathbf{h}!\boldsymbol{\varepsilon}_{n+1}}\boldsymbol{\sigma}_{n+1}$ can be evaluated with the use of the same algorithm, cf. Tables 4.2–4.3, upon only substitution of $\mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon}_{n+1} = \mathbf{0}$ in its input data[1]. This is a very advantageous feature from the point of view of implementation efficiency, as the constitutive DSA algorithm needs to be implemented only once and used at different stages of sensitivity analysis with only modified input data. In the case of shape sensitivity, we could proceed with the same way and then use Eq. (4.43), however, the much simpler way is to directly evaluate $\mathrm{d}_{\mathbf{h}!\mathsf{q}_{n+1}}\boldsymbol{\sigma}_{n+1}$ from Tables 4.2–4.3, upon substitution of $\mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon}_{n+1} = \mathrm{d}_{\mathbf{h}!\mathsf{q}_{n+1}}\boldsymbol{\varepsilon}_{n+1}$ where, cf. Eq. (4.42),

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}\varepsilon_{ij} = -\mathrm{sym}(u_{i,k}\,\mathcal{B}_{kj\beta})\,\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \tag{4.85}$$

4.2.2. Large deformation formulation

Similarly as in the primary analysis, we can formulate DSA equations for large deformation plasticity either as extension of the small-deformation formulation, cf. Section 4.2.1, expressed in terms of Lagrangian tensor measures, or as a result of design-differentiation of the primary equations based on the multiplicative split (3.66) whose example was given in Section 3.3.2.

In the first case, the algorithm in Tables 4.2–4.3 can be simply reused, with the quantities like $\boldsymbol{\sigma}$, $\boldsymbol{\alpha}$, $\boldsymbol{\varepsilon}$, replaced with their finite-deformation counterparts $\boldsymbol{T}$, $\boldsymbol{A}$, $\boldsymbol{E}$, respectively, expressed in the initial (reference) configuration. The only difference, resulting from the appearance of large deformations, is the expression for $\mathrm{d}_{\mathbf{h}!\mathsf{q}_{n+1}}\boldsymbol{E}_{n+1}$ needed as the input to the algorithm in computation of the explicit derivative $\mathrm{d}_{\mathbf{h}!\mathsf{q}_{n+1}}\boldsymbol{T}_{n+1}$. For design-independent geometry it equals zero, as in small deformations, while in the case of shape sensitivity, cf. Eq. (4.46),

$$\mathrm{d}_{\mathbf{h}!\mathsf{q}}E_{ij} = -\mathrm{sym}(F_{ki}u_{k,l}\,\phi_{l\beta,j})\,\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \tag{4.86}$$

In the other, general case, formulation of sensitivity analysis at the constitutive level requires design differentiation of the algorithm displayed in Table 3.4. Again, the iteration loop solving Eq. (3.111) (Table 3.5) does not need to be differentiated. Instead, Eq. (4.70) needs to be solved with the scheme displayed, as for small deformations, in Table 4.3. The only difference, compared to the

[1] Note: $\partial_{\mathbf{h}}\boldsymbol{\varepsilon}_n$ should remain unchanged, only the unknown $\mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon}$ at t_{n+1} should be set to zero!

small deformation formulation, is the explicit derivative $\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,\bar{\sigma}_{n+1}$, given now by, cf. Eq. $(3.139)_1$,

$$\begin{aligned}\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,\bar{\sigma}_{n+1} \;=\; \mathrm{d}_{\mathbf{h}}\bar{\sigma}_{\mathrm{trial}} - \Bigg\{&\left(1+\frac{H'_{n+1}}{3G}\right)\mathrm{d}_{\mathbf{h}}\varphi_{n+1}\\ &+\frac{1}{3G}\left[\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}H'_{n+1} - H'_{n+1}\,\frac{\partial_{\mathbf{h}}G}{G}\right]\varphi_{n+1}\Bigg\}\,\Delta\bar{e}^{\mathrm{p}}\,. \qquad (4.87)\end{aligned}$$

The algorithm for constitutive sensitivity computations in large-deformation elasto-plasticity is displayed in Table 4.4. This is a linear algorithm in which design gradients of intermediate quantities appearing in the primary constitutive formulation, cf. Table 3.4, are computed step-by-step. Values of the quantities themselves are frequently reused here, thus, in real sensitivity computations it is most efficient to execute both the algorithms, primary and sensitivity, simultaneously, so that the latter can on hand access all results of the former.

The only required comment to the sensitivity formulae in Table 4.4 concerns the differentiation of the trial values of spatial tensors. From design differentiation of Eq. (3.123) we obtain

$$\mathrm{d}_{\mathbf{h}}\boldsymbol{f}_{n+1} \;=\; \mathrm{d}_{\mathbf{h}}\boldsymbol{F}_{n+1}\,\boldsymbol{F}_n^{-1} + \boldsymbol{F}_{n+1}\,\mathrm{d}_{\mathbf{h}}\boldsymbol{F}_n^{-1} = (\mathrm{d}_{\mathbf{h}}\boldsymbol{F}_{n+1} - \boldsymbol{f}_{n+1}\,\mathrm{d}_{\mathbf{h}}\boldsymbol{F}_n)\boldsymbol{F}_n^{-1}, \qquad (4.88a)$$

$$\mathrm{d}_{\mathbf{h}}j_{n+1} \;=\; j_{n+1}\,\mathrm{tr}(\mathrm{d}_{\mathbf{h}}\boldsymbol{f}_{n+1}\,\boldsymbol{f}_{n+1}^{-1})\,. \qquad (4.88b)$$

Introducing the notation $\mathrm{d}^{\diamond}_{\mathbf{h}}(\cdot) = [\mathrm{d}_{\mathbf{h}}(\cdot)]\,(\cdot)^{-1}$, we can rewrite the above as

$$\mathrm{d}^{\diamond}_{\mathbf{h}}\boldsymbol{f}_{n+1} \;=\; (\mathrm{d}_{\mathbf{h}}\boldsymbol{f}_{n+1})\boldsymbol{f}_{n+1}^{-1} = (\mathrm{d}_{\mathbf{h}}\boldsymbol{F}_{n+1} - \boldsymbol{f}_{n+1}\,\mathrm{d}_{\mathbf{h}}\boldsymbol{F}_n)\boldsymbol{F}_{n+1}^{-1}\,, \qquad (4.89a)$$

$$\mathrm{d}^{\diamond}_{\mathbf{h}}j_{n+1} \;=\; \mathrm{tr}(\mathrm{d}^{\diamond}_{\mathbf{h}}\boldsymbol{f}_{n+1})\,. \qquad (4.89b)$$

Design differentiation of $\bar{\boldsymbol{f}}_{n+1} = j_{n+1}^{-\frac{1}{3}}\boldsymbol{f}_{n+1}$ yields after transformations

$$\begin{aligned}\mathrm{d}^{\diamond}_{\mathbf{h}}\bar{\boldsymbol{f}}_{n+1} = (\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{f}}_{n+1})\,\bar{\boldsymbol{f}}_{n+1}^{-1} \;&=\; \left(j_{n+1}^{-\frac{1}{3}}\,\mathrm{d}_{\mathbf{h}}\boldsymbol{f}_{n+1} - \tfrac{1}{3}\,j_{n+1}^{-\frac{4}{3}}\,\mathrm{d}_{\mathbf{h}}j_{n+1}\,\boldsymbol{f}_{n+1}\right)\bar{\boldsymbol{f}}_{n+1}^{-1}\\ &=\; \mathrm{d}^{\diamond}_{\mathbf{h}}\boldsymbol{f}_{n+1} - \tfrac{1}{3}\,\mathrm{d}^{\diamond}_{\mathbf{h}}j_{n+1}\,\boldsymbol{I} = \mathrm{dev}(\mathrm{d}^{\diamond}_{\mathbf{h}}\boldsymbol{f}_{n+1})\,. \qquad (4.90)\end{aligned}$$

Now, design differentiation of Eq. $(3.128)_2$ for an arbitrary spatial tensor $\boldsymbol{\psi}$ and the corresponding $\bar{\boldsymbol{\psi}} = J^{-\frac{2}{3}}\boldsymbol{\psi}$ yields

$$\begin{aligned}\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\psi}}_{\mathrm{trial}} \;&=\; \bar{\boldsymbol{f}}_{n+1}\,\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\psi}}_n\,\bar{\boldsymbol{f}}_{n+1}^{\mathrm{T}} + \mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{f}}_{n+1}\,\bar{\boldsymbol{\psi}}_n\bar{\boldsymbol{f}}_{n+1}^{\mathrm{T}} + \bar{\boldsymbol{f}}_{n+1}\bar{\boldsymbol{\psi}}_n\,\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{f}}_{n+1}^{\mathrm{T}}\\ &=\; \bar{\boldsymbol{f}}_{n+1}\,\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\psi}}_n\,\bar{\boldsymbol{f}}_{n+1}^{\mathrm{T}} + \mathrm{d}^{\diamond}_{\mathbf{h}}\bar{\boldsymbol{f}}_{n+1}\,\bar{\boldsymbol{\psi}}_{\mathrm{trial}} + \bar{\boldsymbol{\psi}}_{\mathrm{trial}}\,\mathrm{d}^{\diamond}_{\mathbf{h}}\bar{\boldsymbol{f}}_{n+1}^{\mathrm{T}} \qquad (4.91)\end{aligned}$$

where $\mathrm{d}^{\diamond}_{\mathbf{h}}\bar{\boldsymbol{f}}_{n+1}$ can be found from Eqs. (4.90) and (4.89a).

Table 4.4. Sensitivity algorithm for large-deformation elasto-plasticity (for the sake of legibility, subscripts $(n{+}1)$ have been omitted)

```
Given
    deformation gradient $\boldsymbol{F}_n$ and state parameters $\mathbf{z}_n = \{\, \bar{\boldsymbol{\tau}}^\star_n \,,\ \bar{\boldsymbol{\alpha}}_n \,,\ \bar{e}^{\mathrm{p}}_n \,\}$ at $t_n$
    and their design derivatives, $\partial_{\mathbf{h}}\boldsymbol{F}_n \,,\ \partial_{\mathbf{h}}\mathbf{z}_n = \{\, \partial_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star_n \,,\ \partial_{\mathbf{h}}\bar{\boldsymbol{\alpha}}_n \,,\ \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n \,\}$,

    deformation gradient $\boldsymbol{F}$ at $t_{n+1}$ and its design derivatives $\mathrm{d}_{\mathbf{h}}\boldsymbol{F}$,

    material constants $(E,\ \nu,\ \ldots)$ and their design derivatives

    intermediate results of primary computations (Tables 3.4–3.5),

compute
    design derivatives of state parameters $\mathrm{d}_{\mathbf{h}}\mathbf{z} = \{\, \mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star \,,\ \mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\alpha}} \,,\ \mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} \,\}$
    and stress $\mathrm{d}_{\mathbf{h}}\boldsymbol{\tau}$ at $t_{n+1}$

using the following scheme:

1.  $\mathrm{d}^{\diamond}_{\mathbf{h}}\bar{\boldsymbol{f}} = \operatorname{dev}[(\mathrm{d}_{\mathbf{h}}\boldsymbol{F} - \boldsymbol{f}\,\partial_{\mathbf{h}}\boldsymbol{F}_n)\boldsymbol{F}^{-1}]$        – cf. Eqs. (4.89a), (4.90)
2.  $\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}} = \bar{\boldsymbol{f}}\,\partial_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star_n\,\bar{\boldsymbol{f}}^{\mathrm{T}} + \mathrm{d}^{\diamond}_{\mathbf{h}}\bar{\boldsymbol{f}}\,\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}} + \bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}}\,\mathrm{d}^{\diamond}_{\mathbf{h}}\bar{\boldsymbol{f}}^{\mathrm{T}}$  }
    $\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\alpha}}_{\mathrm{trial}} = \bar{\boldsymbol{f}}\,\partial_{\mathbf{h}}\bar{\boldsymbol{\alpha}}_n\,\bar{\boldsymbol{f}}^{\mathrm{T}} + \mathrm{d}^{\diamond}_{\mathbf{h}}\bar{\boldsymbol{f}}\,\bar{\boldsymbol{\alpha}}_{\mathrm{trial}} + \bar{\boldsymbol{\alpha}}_{\mathrm{trial}}\,\mathrm{d}^{\diamond}_{\mathbf{h}}\bar{\boldsymbol{f}}^{\mathrm{T}}$  }  cf. Eq. (4.91)

if (plastic flow) then
    3.  $\mathrm{d}_{\mathbf{h}}\boldsymbol{s}_{\mathrm{trial}} = \operatorname{dev}(\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}} - \mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\alpha}}_{\mathrm{trial}}), \qquad \mathrm{d}_{\mathbf{h}}\bar{\sigma}_{\mathrm{trial}} = \frac{3}{2\bar{\sigma}_{\mathrm{trial}}}\,\boldsymbol{s}_{\mathrm{trial}} : \mathrm{d}_{\mathbf{h}}\boldsymbol{s}_{\mathrm{trial}}$
    4.  $\mathrm{d}_{\mathbf{h}}\varphi = \operatorname{tr}(\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}} - \mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\alpha}}_{\mathrm{trial}})$
    5.  $\mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}},\ \mathrm{d}_{\mathbf{h}}H' \quad \longleftarrow$    from Table 4.3
    6.  $\mathrm{d}_{\mathbf{h}}\boldsymbol{s}^{\mathrm{p}} = \frac{1}{\bar{\sigma}_{\mathrm{trial}}}\Big\{[\mathrm{d}_{\mathbf{h}}\varphi\,\Delta\bar{e}^{\mathrm{p}} + \varphi\,\mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}}]\,\boldsymbol{s}_{\mathrm{trial}} + \varphi\Delta\bar{e}^{\mathrm{p}}\,\mathrm{d}_{\mathbf{h}}\boldsymbol{s}_{\mathrm{trial}} - \boldsymbol{s}^{\mathrm{p}}\,\mathrm{d}_{\mathbf{h}}\bar{\sigma}_{\mathrm{trial}}\Big\}$
    7.  $\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star = \operatorname{dev}(\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}}) - \mathrm{d}_{\mathbf{h}}\boldsymbol{s}^{\mathrm{p}} + \tfrac{1}{3}\boldsymbol{I}\,\operatorname{tr}(\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}})$
        $\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\alpha}} = \operatorname{dev}(\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\alpha}}_{\mathrm{trial}}) + \frac{1}{3G}\Big[\Big(\mathrm{d}_{\mathbf{h}}H' + \frac{H'\partial_{\mathbf{h}}G}{G}\Big)\,\boldsymbol{s}^{\mathrm{p}} + H'\mathrm{d}_{\mathbf{h}}\boldsymbol{s}^{\mathrm{p}}\Big] + \tfrac{1}{3}\boldsymbol{I}\,\operatorname{tr}(\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\alpha}}_{\mathrm{trial}})$
        $\mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} = \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n + \mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}}$
else (purely elastic deformation)
    8.  $\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star = \mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star_{\mathrm{trial}}\,, \qquad \mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\alpha}} = \mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\alpha}}_{\mathrm{trial}}\,, \qquad \mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} = \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n$
end if

9.  $\mathrm{d}_{\mathbf{h}}\boldsymbol{\tau} = \operatorname{dev}(\mathrm{d}_{\mathbf{h}}\bar{\boldsymbol{\tau}}^\star) + \tfrac{1}{2}\boldsymbol{I}\,(\partial_{\mathbf{h}}K)[(\det\boldsymbol{F})^2 - 1] + \boldsymbol{I}\,K(\det\boldsymbol{F})^2\,\operatorname{tr}(\mathrm{d}_{\mathbf{h}}\boldsymbol{F}\,\boldsymbol{F}^{-1})$
```

The algorithm presented in Table 4.4 evaluates the stress sensitivity $\mathrm{d}_{\mathbf{h}}\boldsymbol{\tau}_{n+1}$, whose general form was discussed in Section 4.1, cf. Eq. (4.20). Recalling discussion on this subject accompanying the small deformation formulation, cf. Section 4.2.1, we will not present detailed equations on the derivatives $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{F}_n}$, $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{z}_{i\,n}}$, and $\partial_{\mathbf{h}}\boldsymbol{\tau}_{n+1}$, appearing in Eq. (4.20), as their evaluation in this form is not necessary.

Similarly as in the small deformation formulation, the presented algorithm can be also employed to compute the explicit design derivative $\mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\boldsymbol{\tau}_{n+1}$ required to determine the right-hand side array $\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{res}}_{n+1} = \mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\mathbf{f}^{\mathrm{res}}_{n+1}$ in the DSA system of equations (4.3). In the case of design-independent geometry we have, cf. Eq. (4.21), $\mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\boldsymbol{\tau}_{n+1} = \mathrm{d}_{\mathbf{h}!\boldsymbol{F}_{n+1}}\boldsymbol{\tau}_{n+1}$, and thus the only change necessary in the algorithm is just to set $\mathrm{d}_{\mathbf{h}}\boldsymbol{F}_{n+1} = \mathbf{0}$ in the input. In the case of shape sensitivity, we could proceed with the same way and then use Eq. (4.53), however, the much simpler way is to set $\mathrm{d}_{\mathbf{h}}\boldsymbol{F}_{n+1} = \mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\boldsymbol{F}_{n+1}$ in the input to Table 4.4, with $\mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\boldsymbol{F}_{n+1}$ given by, cf. Eq. (4.38),

$$\mathrm{d}_{\mathbf{h}!\mathbf{q}}F_{ij} = -u_{i,k}\,\phi_{k\beta,j}\,\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \tag{4.92}$$

4.2.3. Reduced-dimension formulations

The reduced dimension formulations for sensitivity analysis at the constitutive level can be derived in an analogous way to the 3-D formulations. The general idea is to differentiate step by step the algorithms for primary constitutive computations, with special treatment of the consistency equation which at the primary stage is nonlinear and solved in Newton iteration (3.114) or (3.158) while at the DSA stage it is a linear equation with respect to the design derivatives of the consistency parameter.

In the case of one-dimensional bar, we will not repeat details of discussion given in the two previous sections, and we limit ourselves to mere presentation of a tabularized step-by-step algorithm for sensitivity computations, based on the primary equations discussed in Section 3.3.3.1. Realizing that, cf. Eq. (3.148),

$$\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,\bar{\sigma}_{n+1} = \mathrm{d}_{\mathbf{h}}\bar{\sigma}_{\mathrm{trial}} - (\partial_{\mathbf{h}}E + \partial_{\mathbf{h}}H'_{n+1})\,\Delta\bar{e}^{\mathrm{p}}\,. \tag{4.93}$$

we can write down the design-differentiated equations of Table 3.6 in the form of a linear algorithm displayed in Table 4.5.

Computation of the explicit design derivatives $\mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\sigma_{\|n+1}$ can be performed using the same algorithm, upon substitution $\mathrm{d}_{\mathbf{h}}\varepsilon_{\|n+1} = \mathbf{0}$ for design-independent geometry, or, cf. Eq. (4.42),

$$\mathrm{d}_{\mathbf{h}}\varepsilon_{\|n+1} = \mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\varepsilon_{\|n+1} = -u'_{\|}\varphi'_{\beta}\,\partial_{\mathbf{h}}\mathsf{X}_{\beta} \tag{4.94}$$

Table 4.5. Sensitivity algorithm for small-deformation 1-D bar formulation (for the sake of legibility, subscripts $(n{+}1)$ and $\|$ have been omitted)

```
Given
    strain $\varepsilon_n$, $\varepsilon_{\perp n}$ and state parameters $\mathbf{z}_n = \{\, \sigma_n \,,\, \alpha_n \,,\, \bar{e}^{\mathrm{p}}_n \,\}$ at $t_n$
    and their design derivatives $\partial_{\mathbf{h}}\varepsilon_n$, $\partial_{\mathbf{h}}\varepsilon_{\perp n}$,
    $\partial_{\mathbf{h}}\mathbf{z}_n = \{\, \partial_{\mathbf{h}}\sigma_n \,,\, \partial_{\mathbf{h}}\alpha_n \,,\, \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n \,\}$,
    strain $\varepsilon$ at $t_{n+1}$ and its design derivatives $\mathrm{d}_{\mathbf{h}}\varepsilon$   $(\mathrm{d}_{\mathbf{h}}\Delta\varepsilon = \mathrm{d}_{\mathbf{h}}\varepsilon - \partial_{\mathbf{h}}\varepsilon_n)$,
    material constants $(E,\, \nu,\, \ldots)$ and their design derivatives
    intermediate results of primary computations (Tables 3.6–3.7),
compute
    design derivatives of state parameters (including stress)
    $\mathrm{d}_{\mathbf{h}}\mathbf{z} = \{\, \mathrm{d}_{\mathbf{h}}\sigma \,,\, \mathrm{d}_{\mathbf{h}}\alpha \,,\, \mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} \,\}$ at $t_{n+1}$
    design derivatives of transverse strain increment $\mathrm{d}_{\mathbf{h}}\Delta\varepsilon_{\perp}$
using the following scheme:
1.  $\mathrm{d}_{\mathbf{h}}\sigma_{\mathrm{trial}} = \partial_{\mathbf{h}}\sigma_n + \partial_{\mathbf{h}}E\,\Delta\varepsilon + E\,\mathrm{d}_{\mathbf{h}}\Delta\varepsilon$
if (plastic flow) then
    2.  $\mathrm{d}_{\mathbf{h}}\bar{\sigma}_{\mathrm{trial}} = \mathrm{d}_{\mathbf{h}}\sigma_{\mathrm{trial}} - \frac{2}{3}\,\partial_{\mathbf{h}}\alpha_n$
    3.  $\mathrm{d}_{\mathbf{h}}\Delta\varepsilon^{\mathrm{p}} \equiv \mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}}$, $\mathrm{d}_{\mathbf{h}}H'$   $\longleftarrow$   from Table 4.3
    4.  $\mathrm{d}_{\mathbf{h}}\Delta\varepsilon_{\perp} = -\partial_{\mathbf{h}}\nu(\Delta\varepsilon - \Delta\varepsilon^{\mathrm{p}}) - \nu(\mathrm{d}_{\mathbf{h}}\Delta\varepsilon - \mathrm{d}_{\mathbf{h}}\Delta\varepsilon^{\mathrm{p}}) - \frac{1}{2}\mathrm{d}_{\mathbf{h}}\Delta\varepsilon^{\mathrm{p}}$
    5.  $\mathrm{d}_{\mathbf{h}}\sigma = \mathrm{d}_{\mathbf{h}}\sigma_{\mathrm{trial}} - \partial_{\mathbf{h}}E\,\Delta\varepsilon^{\mathrm{p}} - E\,\mathrm{d}_{\mathbf{h}}\Delta\varepsilon^{\mathrm{p}}$
        $\mathrm{d}_{\mathbf{h}}\alpha = \partial_{\mathbf{h}}\alpha_n + \mathrm{d}_{\mathbf{h}}H'\,\Delta\varepsilon^{\mathrm{p}} + H'\,\mathrm{d}_{\mathbf{h}}\Delta\varepsilon^{\mathrm{p}}$
        $\mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} = \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n + \mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}}$
else (purely elastic deformation)
    6.  $\mathrm{d}_{\mathbf{h}}\sigma = \mathrm{d}_{\mathbf{h}}\sigma_{\mathrm{trial}}$,   $\mathrm{d}_{\mathbf{h}}\alpha = \partial_{\mathbf{h}}\alpha_n$,   $\mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} = \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n$
end if
```

for shape sensitivity (where $'$ denotes differentiation of the 1D displacement field or the shape functions with respect to x).

In the case of plane stress formulation, the consistency parameter sought as the unknown in the nonlinear consistency equation (3.111) is λ rather than $\Delta\bar{e}^{\mathrm{p}}$. Thus, the formulation slightly differs from the previous ones. Equation (4.70) is now replaced by

$$\tilde{f}_{,\lambda}\,\mathrm{d}_{\mathbf{h}}\lambda = -\mathrm{d}_{\mathbf{h}!\lambda}\tilde{f}\,, \tag{4.95}$$

a linear equation in which $\tilde{f}_{,\lambda}$ is given by Eqs. (3.159)–(3.161). The partial design derivative $\mathrm{d}_{\mathbf{h}!\lambda}\tilde{f}$ is determined as follows.

Since $\tilde{f}$ can be generally presented as a function $\tilde{f}(\bar{\sigma}_{n+1}, \bar{e}^{\mathrm{p}}_{n+1}, \Delta\bar{e}^{\mathrm{p}}; \mathbf{h})$, its derivative at frozen λ has the form

$$\begin{aligned}\mathrm{d}_{\mathbf{h}!\lambda}\tilde{f} \;=\; & \frac{\partial\tilde{f}}{\partial\bar{\sigma}_{n+1}}\,\mathrm{d}_{\mathbf{h}!\lambda}\bar{\sigma}_{n+1}\\ & + \frac{\partial\tilde{f}}{\partial\bar{e}^{\mathrm{p}}_{n+1}}\,(\partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} + \mathrm{d}_{\mathbf{h}!\lambda}\,\Delta\bar{e}^{\mathrm{p}}) + \frac{\partial\tilde{f}}{\partial\Delta\bar{e}^{\mathrm{p}}}\,\mathrm{d}_{\mathbf{h}!\lambda}\,\Delta\bar{e}^{\mathrm{p}} + \partial_{\mathbf{h}}\tilde{f}\,.\end{aligned} \tag{4.96}$$

In view of Eq. (3.153) (from which we have $\mathrm{d}_{\mathbf{h}!\lambda}\,\Delta\bar{e}^{\mathrm{p}} = \frac{2}{3}\lambda\,\mathrm{d}_{\mathbf{h}!\lambda}\bar{\sigma}_{n+1}$) we can rewrite Eq. (4.96) as

$$\mathrm{d}_{\mathbf{h}!\lambda}\tilde{f} = \vartheta_1\,\mathrm{d}_{\mathbf{h}!\lambda}\bar{\sigma}_{n+1} + \frac{\partial\tilde{f}}{\partial\bar{e}^{\mathrm{p}}_{n+1}}\,\partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} + \partial_{\mathbf{h}}\tilde{f}\,, \tag{4.97}$$

where ϑ_1 is given by Eq. (3.160) and

- for rate-independent models: $\dfrac{\partial\tilde{f}}{\partial\bar{e}^{\mathrm{p}}_{n+1}} = -\kappa'_{n+1}$ and $\partial_{\mathbf{h}}\tilde{f} = -\partial_{\mathbf{h}}\kappa$,
- for rate-dependent models: $\dfrac{\partial\tilde{f}}{\partial\bar{e}^{\mathrm{p}}_{n+1}} = \Delta t\,(g_{,\bar{e}^{\mathrm{p}}})_{n+1}$ and $\partial_{\mathbf{h}}\tilde{f}$, $g_{,\bar{e}^{\mathrm{p}}}$ are given by Eqs. (4.78) or (4.79), depending on a particular viscoplastic formulation.

To compute $\mathrm{d}_{\mathbf{h}!\lambda}\bar{\sigma}_{n+1}$, let us subsequently differentiate Eqs. (3.152b), (3.154)–(3.155), (3.153), and recall Eq. (4.74):

$$\begin{aligned}\mathrm{d}_{\mathbf{h}!\lambda}\bar{\sigma}_{n+1} &= \frac{3}{2\bar{\sigma}_{n+1}}\,\varsigma^{\mathrm{T}}_{n+1}\mathbf{P}\,\mathrm{d}_{\mathbf{h}!\lambda}\varsigma_{n+1}\,,\\ \mathrm{d}_{\mathbf{h}!\lambda}\varsigma_{n+1} &= \mathbf{Z}^{-1}\,(\mathrm{d}_{\mathbf{h}}\varsigma_{\mathrm{trial}} - \mathrm{d}_{\mathbf{h}!\lambda}\mathbf{Z}\,\varsigma_{n+1})\,,\\ \mathrm{d}_{\mathbf{h}!\lambda}\mathbf{Z} &= \tfrac{2}{3}\lambda\,\mathrm{d}_{\mathbf{h}!\lambda}H'_{n+1}\,\mathbf{I} + \lambda\,\partial_{\mathbf{h}}\mathbf{C}^{\mathrm{e}}\,\mathbf{P}\,,\\ \mathrm{d}_{\mathbf{h}!\lambda}H'_{n+1} &= \partial_{\mathbf{h}}H'_{n+1} + H''_{n+1}(\partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} + \mathrm{d}_{\mathbf{h}!\lambda}\Delta\bar{e}^{\mathrm{p}}) = \mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,H'_{n+1} + \mathrm{d}_{\mathbf{h}!\lambda}\Delta\bar{e}^{\mathrm{p}})\,,\\ \mathrm{d}_{\mathbf{h}!\lambda}\Delta\bar{e}^{\mathrm{p}} &= \tfrac{2}{3}\lambda\,\mathrm{d}_{\mathbf{h}!\lambda}\bar{\sigma}_{n+1}\,.\end{aligned}$$

Gathering the above equations in one, we can determine $\mathrm{d}_{\mathbf{h}!\lambda}\bar{\sigma}_{n+1}$ from the scalar equation

$$\begin{aligned}\tfrac{2}{3}\left(\bar{\sigma}_{n+1}+\tfrac{2}{3}\,\varsigma_{n+1}^{\mathrm{T}}\mathbf{P}\mathbf{Z}^{-1}\varsigma_{n+1}\,\lambda^{2}H''_{n+1}\right)\mathrm{d}_{\mathbf{h}!\lambda}\bar{\sigma}_{n+1} &= \\ =\varsigma_{n+1}^{\mathrm{T}}\mathbf{P}\mathbf{Z}^{-1}\left[\mathrm{d}_{\mathbf{h}}\varsigma_{\mathrm{trial}}-\lambda\mathbf{P}\varsigma_{n+1}\,\partial_{\mathbf{h}}\mathbf{C}^{\mathrm{e}}-\tfrac{2}{3}\varsigma_{n+1}\lambda\,\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\,H'_{n+1}\right]&, \qquad (4.98)\end{aligned}$$

substitute it to Eq. (4.97), and finally solve Eq. (4.95) with respect to $\mathrm{d}_{\mathbf{h}}\lambda$.

Differentiation of remaining equations in the algorithm of Table 3.8, combined with the above derivations, leads to the constitutive sensitivity algorithm presented in Tables 4.6, 4.7. The algorithm computes design derivatives of intermediate variables computed during primary analysis. Values of the variables themselves are assumed known and frequently reused in the sensitivity computations.

In full analogy to the general 3D formulations, the same algorithm can also be employed to compute explicit design derivatives $\mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\boldsymbol{\sigma}_{n+1}$. This can be done upon substitution $\mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon}_{n+1}=\mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\boldsymbol{\varepsilon}_{n+1}$ in the input data. Recalling considerations of Section 4.2.1, this quantity equals zero in the case of design-independent geometry and, cf. Eq. (4.42),

$$\mathrm{d}_{\mathbf{h}!\mathbf{q}}\varepsilon_{ij}=-\mathrm{sym}(u_{i,k}\,\mathcal{B}_{kj\beta})\,\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \qquad (4.99)$$

in the case of shape sensitivity analysis. For the presently discussed 2D formulation, indices i,j,k run over the values 1,2 only.

Summarizing results of this section, detailed algorithms for sensitivity computations at the level of constitutive equations for a variety of incremental elasto-plastic formulations discussed in Section 3.3 have been presented. The algorithms allow to determine the explicit derivatives of stress, necessary to build the right-hand side array of the global system of sensitivity equations (4.3) as well as total design derivatives of stress and constitutive state variables at the update stage, when the system (4.3) has already been solved. Two significant features of the algorithms have been underlined. First, the algorithms may serve in their unchanged form to both computation of explicit and total design derivatives of stress and/or constitutive state variables — the difference consists in introducing different values of input parameters (assumed sensitivity of deformation gradient or strain). This is important from the point of view of the implementation effort in a finite element code. Second, the nonlinear consistency condition, which in the primary analysis results in a necessity of Newton iteration, becomes linear when differentiated with respect to design parameters. This makes the constitutive sensitivity formulation efficient, compared to the primary formulation.

Table 4.6. Sensitivity algorithm for small-deformation plane stress formulation (for the sake of legibility, subscripts $(n{+}1)$ have been omitted)

```
Given

    strain $\boldsymbol{\varepsilon}_n\,,\ \varepsilon_{\perp n}$ and state parameters $\mathbf{z}_n = \{\,\boldsymbol{\sigma}_n\,,\ \boldsymbol{\alpha}_n\,,\ \bar{e}^{\mathrm{p}}_n\,\}$  at $t_n$
    and their design derivatives, $\partial_{\mathbf{h}}\boldsymbol{\varepsilon}_n\,,\ \partial_{\mathbf{h}}\varepsilon_{\perp n}\,,\ \partial_{\mathbf{h}}\mathbf{z}_n = \{\,\partial_{\mathbf{h}}\boldsymbol{\sigma}_n\,,\ \partial_{\mathbf{h}}\boldsymbol{\alpha}_n\,,\ \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n\,\}$,

    strain $\boldsymbol{\varepsilon}$  at $t_{n+1}$ and its design derivatives $\mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon}$   $(\mathrm{d}_{\mathbf{h}}\Delta\boldsymbol{\varepsilon} = \mathrm{d}_{\mathbf{h}}\boldsymbol{\varepsilon} - \partial_{\mathbf{h}}\boldsymbol{\varepsilon}_n)$,

    material constants $(E,\ \nu,\ \ldots)$ and their design derivatives

    intermediate results of primary computations (Tables 3.6–3.7),

compute

    design derivatives of state parameters (including stress)
    $\mathrm{d}_{\mathbf{h}}\mathbf{z} = \{\,\mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma}\,,\ \mathrm{d}_{\mathbf{h}}\boldsymbol{\alpha}\,,\ \mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}}\,\}$  at $t_{n+1}$

    design derivatives of transverse strain increment $\mathrm{d}_{\mathbf{h}}\Delta\varepsilon_{\perp}$

using the following scheme:

1.  $\mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma}_{\mathrm{trial}} = \partial_{\mathbf{h}}\boldsymbol{\sigma}_n + \partial_{\mathbf{h}}\mathbf{C}^{\mathrm{e}}\,\Delta\boldsymbol{\varepsilon} + \mathbf{C}^{\mathrm{e}}\,\mathrm{d}_{\mathbf{h}}\Delta\boldsymbol{\varepsilon}$

if (plastic flow) then

    2.  $\mathrm{d}_{\mathbf{h}}\boldsymbol{\varsigma}_{\mathrm{trial}} = \mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma}_{\mathrm{trial}} - \partial_{\mathbf{h}}\boldsymbol{\alpha}_n$,      $\mathrm{d}_{\mathbf{h}}\bar{\sigma}_{\mathrm{trial}} = \frac{3}{2\bar{\sigma}_{\mathrm{trial}}}\,\boldsymbol{\varsigma}_{\mathrm{trial}}\mathbf{P}\,\mathrm{d}_{\mathbf{h}}\boldsymbol{\varsigma}_{\mathrm{trial}}$
    3.  $\mathrm{d}_{\mathbf{h}}\lambda,\ \mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}},\ \mathrm{d}_{\mathbf{h}}H',\ \mathrm{d}_{\mathbf{h}}\boldsymbol{\varsigma}$   $\longleftarrow$   from Table 4.7
    4.  $\mathrm{d}_{\mathbf{h}}\Delta\boldsymbol{\varepsilon}^{\mathrm{p}} = \mathrm{d}_{\mathbf{h}}\lambda\,\mathbf{P}\boldsymbol{\varsigma} + \lambda\mathbf{P}\,\mathrm{d}_{\mathbf{h}}\boldsymbol{\varsigma}$,      $\mathrm{d}_{\mathbf{h}}\Delta\boldsymbol{\varepsilon}^{\mathrm{e}} = \mathrm{d}_{\mathbf{h}}\Delta\boldsymbol{\varepsilon} - \mathrm{d}_{\mathbf{h}}\Delta\boldsymbol{\varepsilon}^{\mathrm{p}}$
    5.  $\mathrm{d}_{\mathbf{h}}\Delta\varepsilon_{\perp} = -\partial_{\mathbf{h}}\nu\,(\Delta\varepsilon^{\mathrm{e}}_{11}{+}\Delta\varepsilon^{\mathrm{e}}_{22}) - \nu\,(\mathrm{d}_{\mathbf{h}}\Delta\varepsilon^{\mathrm{e}}_{11}{+}\mathrm{d}_{\mathbf{h}}\Delta\varepsilon^{\mathrm{e}}_{22})$
                    $-\frac{1}{2}(\mathrm{d}_{\mathbf{h}}\Delta\varepsilon^{\mathrm{p}}_{11}{+}\mathrm{d}_{\mathbf{h}}\Delta\varepsilon^{\mathrm{p}}_{22})$
    6.  $\mathrm{d}_{\mathbf{h}}\boldsymbol{\alpha} = \partial_{\mathbf{h}}\boldsymbol{\alpha}_n + \frac{2}{3}\,(\mathrm{d}_{\mathbf{h}}\lambda\,H'\boldsymbol{\varsigma} + \lambda\,\partial_{\mathbf{h}}H'\,\boldsymbol{\varsigma} + \lambda H'\,\mathrm{d}_{\mathbf{h}}\boldsymbol{\varsigma})$
        $\mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma} = \mathrm{d}_{\mathbf{h}}\boldsymbol{\varsigma} + \mathrm{d}_{\mathbf{h}}\boldsymbol{\alpha}$
        $\mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} = \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n + \mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}}$

else (purely elastic deformation)

    7.  $\mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma} = \mathrm{d}_{\mathbf{h}}\boldsymbol{\sigma}_{\mathrm{trial}}\,,$     $\mathrm{d}_{\mathbf{h}}\boldsymbol{\alpha} = \partial_{\mathbf{h}}\boldsymbol{\alpha}_n\,,$     $\mathrm{d}_{\mathbf{h}}\bar{e}^{\mathrm{p}} = \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_n$

end if
```

Table 4.7. Computation of $\mathrm{d}_{\mathbf{h}}\lambda$ and related design derivatives for different elasto-plastic and elasto-viscoplastic models

```
1. $\mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, H' = H''\, \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} + \partial_{\mathbf{h}} H'$
2. $\mathrm{d}_{\mathbf{h}!\lambda}\, \bar{\sigma} \quad \longleftarrow$ from Eq. (4.98)

if (rate-independent plasticity)

    3a. $\vartheta_1 = 1 - \frac{2}{3}\lambda\kappa'$
    4a. $\mathrm{d}_{\mathbf{h}!\lambda}\tilde{f} = \vartheta_1\, \mathrm{d}_{\mathbf{h}!\lambda}\, \bar{\sigma} - \kappa'\partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} - \partial_{\mathbf{h}}\kappa$

else if (overstress viscoplasticity)

    3b. $g_{,\bar{\sigma}}\, ,\, g_{,\bar{e}^{\mathrm{p}}}\, ,\, \partial_{\mathbf{h}} g \quad \longleftarrow$ from Eqs. (4.78)
    4b. $\vartheta_1 = \Delta t\, g_{,\bar{\sigma}} - \frac{2}{3}\lambda(1 - \Delta t\, g_{,\bar{e}^{\mathrm{p}}})$
    5b. $\mathrm{d}_{\mathbf{h}!\lambda}\tilde{f} = \vartheta_1\, \mathrm{d}_{\mathbf{h}!\lambda}\, \bar{\sigma} + \Delta t(g_{,\bar{e}^{\mathrm{p}}}\, \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} + \partial_{\mathbf{h}} g)$

else if (power-law hardening viscoplasticity)

    3c. $g_{,\bar{\sigma}}\, ,\, g_{,\bar{e}^{\mathrm{p}}}\, ,\, \partial_{\mathbf{h}} g \quad \longleftarrow$ from Eqs. (4.79)
    4c. $\vartheta_1 = \Delta t\, g_{,\bar{\sigma}} - \frac{2}{3}\lambda(1 - \Delta t\, g_{,\bar{e}^{\mathrm{p}}})$
    5c. $\mathrm{d}_{\mathbf{h}!\lambda}\tilde{f} = \vartheta_1\, \mathrm{d}_{\mathbf{h}!\lambda}\, \bar{\sigma} + \Delta t(g_{,\bar{e}^{\mathrm{p}}}\, \partial_{\mathbf{h}}\bar{e}^{\mathrm{p}}_{n} + \partial_{\mathbf{h}} g)$

end if

6. $\mathrm{d}_{\mathbf{h}}\lambda = -\mathrm{d}_{\mathbf{h}!\lambda}\tilde{f}/\tilde{f}_{,\lambda}$
7. $\mathrm{d}_{\mathbf{h}}\mathbf{Z} = \frac{\mathrm{d}_{\mathbf{h}}\lambda}{\lambda}\,(\mathbf{Z} - \mathbf{I}) + \lambda\left(\frac{2}{3}\,\mathrm{d}_{\mathbf{h}} H'\mathbf{I} + \partial_{\mathbf{h}}\mathbf{C}^{\mathrm{e}}\,\mathbf{P}\right)$
8. $\mathrm{d}_{\mathbf{h}}\boldsymbol{\varsigma} = \mathbf{Z}^{-1}\,(\mathrm{d}_{\mathbf{h}}\boldsymbol{\varsigma}_{\mathrm{trial}} - \mathrm{d}_{\mathbf{h}}\mathbf{Z}\,\boldsymbol{\varsigma}), \qquad \mathrm{d}_{\mathbf{h}}\bar{\sigma} = \frac{3}{2\bar{\sigma}}\,\boldsymbol{\varsigma}\mathbf{P}\,\mathrm{d}_{\mathbf{h}}\boldsymbol{\varsigma}$
9. $\mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}} = \frac{2}{3}(\mathrm{d}_{\mathbf{h}}\lambda\,\bar{\sigma} + \lambda\,\mathrm{d}_{\mathbf{h}}\bar{\sigma})$
   $\mathrm{d}_{\mathbf{h}} H' = \mathrm{d}_{\mathbf{h}!\Delta\bar{e}^{\mathrm{p}}}\, H' + H''\,\mathrm{d}_{\mathbf{h}}\Delta\bar{e}^{\mathrm{p}}$
```

4.3. Computational examples

In the following section, the above derivations of computational sensitivity analysis of nonlinear elasto-plastic static problem will be illustrated with computational examples. The primary and sensitivity formulations for both geometrically linear and large deformation cases have been implemented in the author's own finite element program written in Fortran 77. Short description of the program is given in Appendix B. Two typical structural problems will be analysed: plane stress stretching of a rectangular strip with a circular hole (in the small deformation range) and 3D bending of a carved beam (large deformations).

4.3.1. Stretching of a rectangular strip with a circular hole

This is a small deformation example formulated in terms of 2D plane stress analysis. Geometry and loads are shown in Fig. 4.5. The rectangle dimensions are 160×80 mm, the hole diameter $2r = 40$ mm, and the thickness $b = 1$ mm. The strip is loaded with prescribed displacement $\hat{u}(t)$ applied uniformly on the strip's shorter edges. Symmetry allows to only analyse a quarter of the structure; the finite element mesh is also shown in Fig. 4.5.

The strip is made of steel with the rate-independent, Huber–Mises elasto-plastic material model and with linear isotropic hardening. The following material constants are assumed: Young modulus $E = 210\,000$ MPa, Poisson ratio

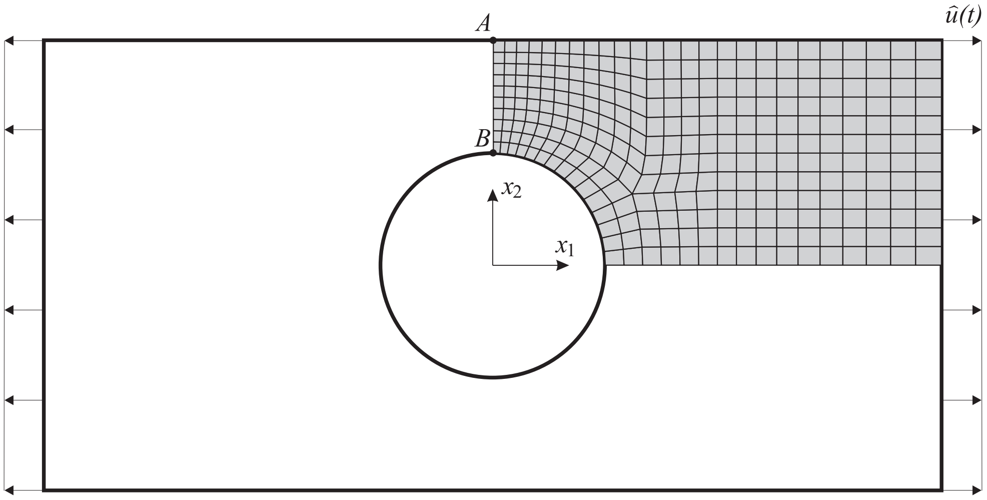


Figure 4.5. Stretching of a rectangular strip with a circular hole: geometry and finite element mesh

$\nu = 0.3$, yield stress $\sigma_{\mathrm{y}} = 400$ MPa, isotropic hardening modulus $\kappa' = 200$ MPa, no kinematic hardening ($H \equiv 0$).

The displacement $\hat{u}(t)$ increases monotonically in time from 0 to 0.2 mm (which corresponds to average strain $\varepsilon_{11} = 0.0025$) at $t = 0.4$ s. Equal time increments $\Delta t = 0.01$ s were applied in the analysis. Figure 4.6 presents selected snapshots of the strip deformation stages with evolution of plastic areas indicated. Up to $t = 0.1$ s ($\hat{u} = 0.05$ mm) the response is elastic. Later on, plastic deformation develop; at $t = 0.26$ s they already embrace the entire strip cross-section.

The following performances have been monitored during the analysis: vertical displacement at point A, $u_A \equiv -u_2(A)$, longitudinal stress at the highest concentration point B, $\sigma_B \equiv \sigma_{11}(B)$, and the total reaction force in the cross section $A{-}B$, F (due to symmetry, F is only a half of the total reaction force in the strip).

Figure 4.7 presents the primary analysis results for the above defined performances. Stress σ_B increases until about $t = 0.10$ s and then, upon development of plastic flow, it remains nearly constant at the level slightly exceeding (due to hardening) the yield limit σ_{y}. The reaction force F grows rapidly during the elastic loading, then slightly slower at plastic flow, and it gets virtually stabilized when the full strip cross-section becomes plastic (at $t > 0.26$ s).

Figures 4.8–4.10 present sensitivity results of the three performances with respect to six design parameters: hole radius r, strip thickness b, and the four material constants E, ν, σ_{y} and κ'. The first of them, $h_1 = r$, is the only shape design parameter (although thickness b is a geometric parameter, too, in 2D problems it is rather treated as a material constant since the integration over Ω^r is reduced to integration over a 2D area). To express the results in a form independent of the design parameter dimensions, all the sensitivities in the presented graphs have been multiplied by primary values of the corresponding design parameters (in other words, the so defined 'normalized sensitivities' express relations between changes of the primary response and *relative* changes of the design parameter values). Analytical results are compared to their finite-difference approximations computed with the central difference scheme (2.47c),

$$\tilde{\mathrm{d}}_{\mathbf{h}} G = \{\tilde{\mathrm{d}}_{h_i} G\}, \qquad \tilde{\mathrm{d}}_{h_i} G = \frac{G(h_i{+}\Delta h_i) - G(h_i{-}\Delta h_i)}{2\Delta h_i}\,, \tag{4.100}$$

with finite perturbations $\Delta h_i \approx 10^{-4} h_i$. The choice of these perturbation amounts was preceded by several trial-and-error tests in order to make sure that they belong to the appropriate range (see discussion in Section 2.2.3).

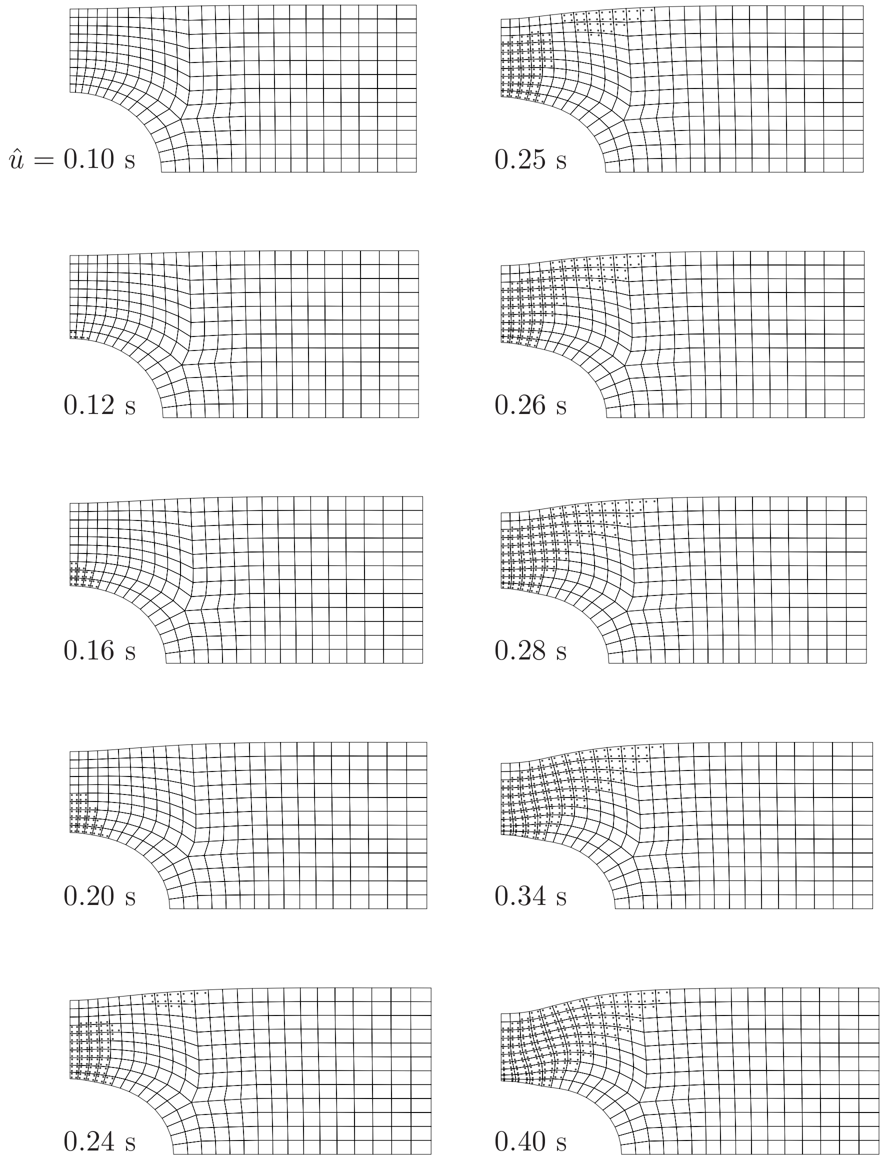


Figure 4.6. Deformation of the strip at different time instants. Displacements are magnified by the factor of 50. Dots denote the integration points at which plastic flow occurs

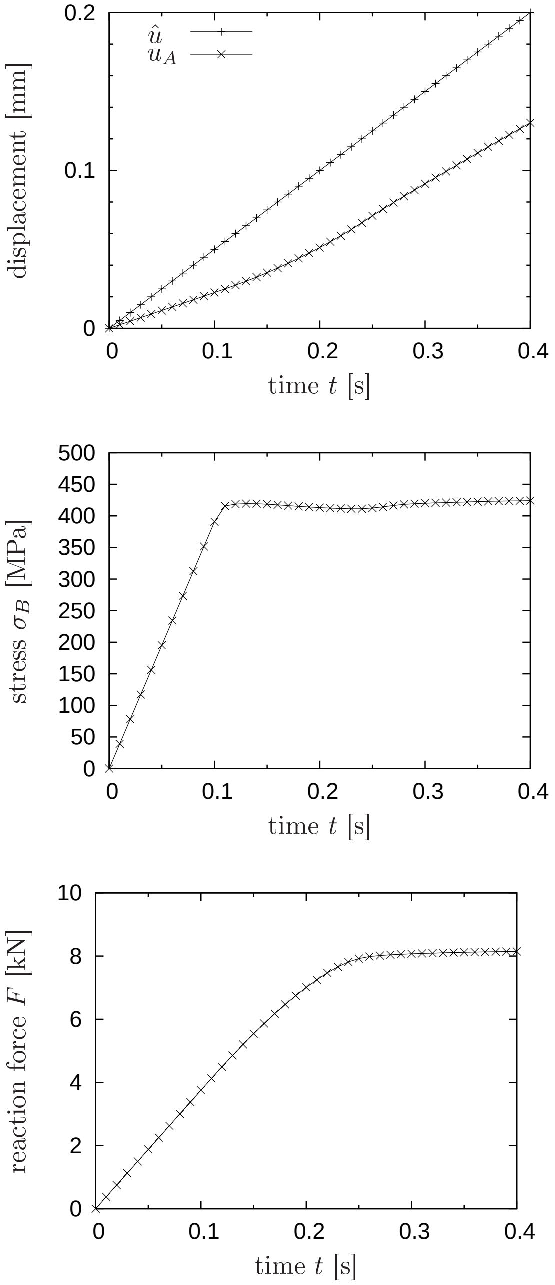


Figure 4.7. Transient results for the displacement u_A, stress σ_B and the reaction force F

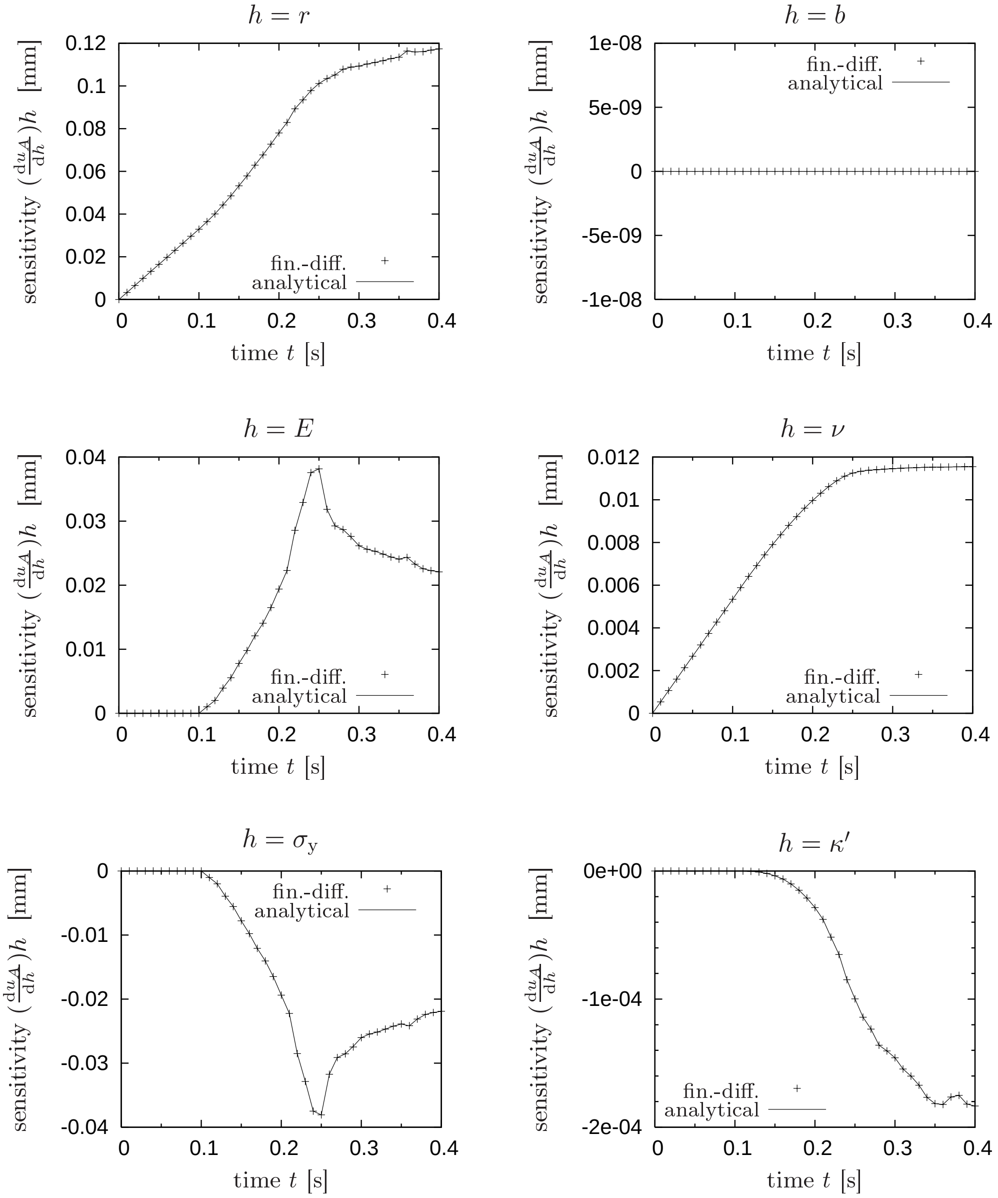


Figure 4.8. Transient normalized sensitivity of the displacement u_A with respect to six design parameters. Comparison of analytical design derivatives and their finite-difference approximations

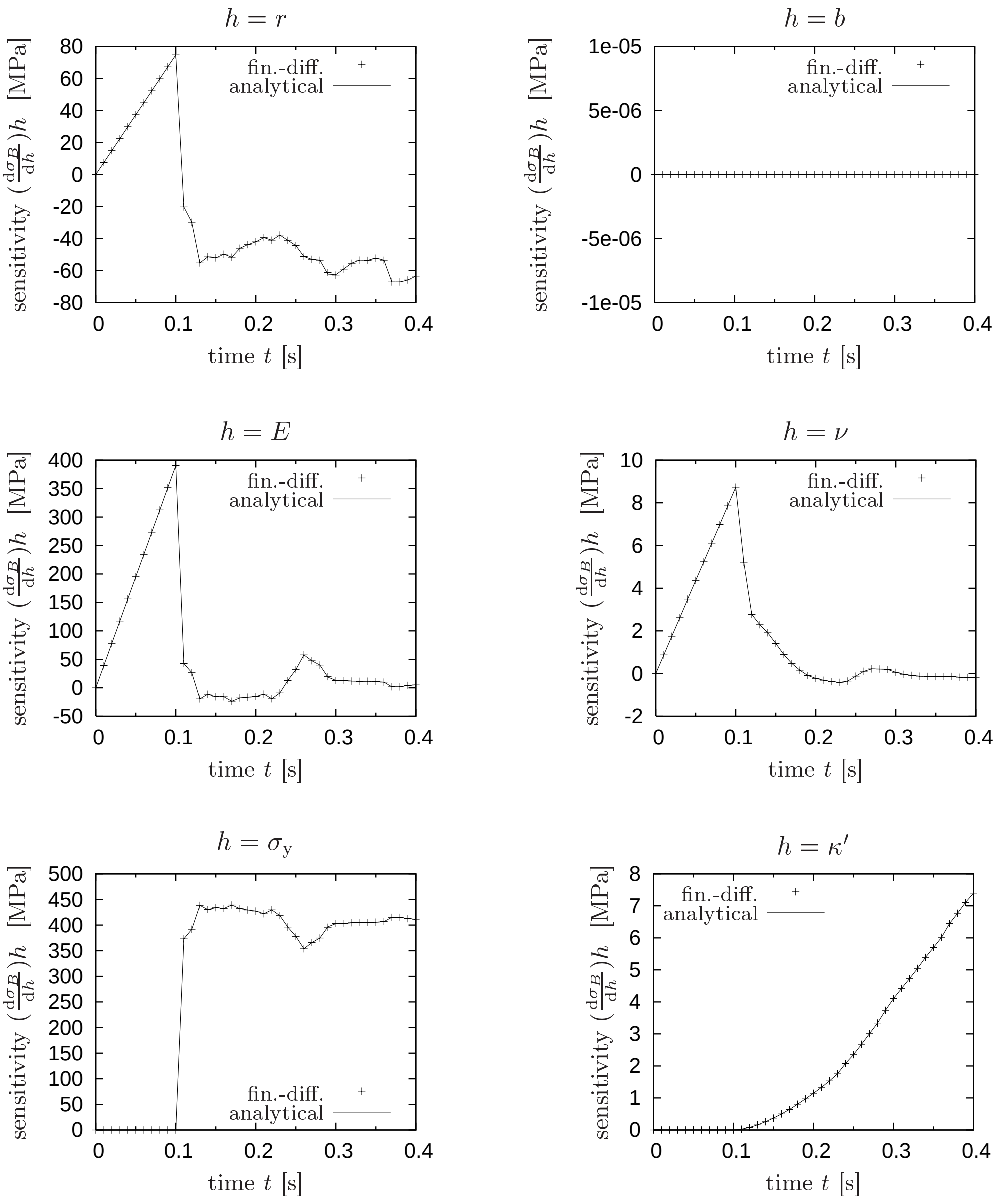


Figure 4.9. Transient normalized sensitivity of the stress σ_B with respect to six design parameters. Comparison of analytical design derivatives and their finite-difference approximations

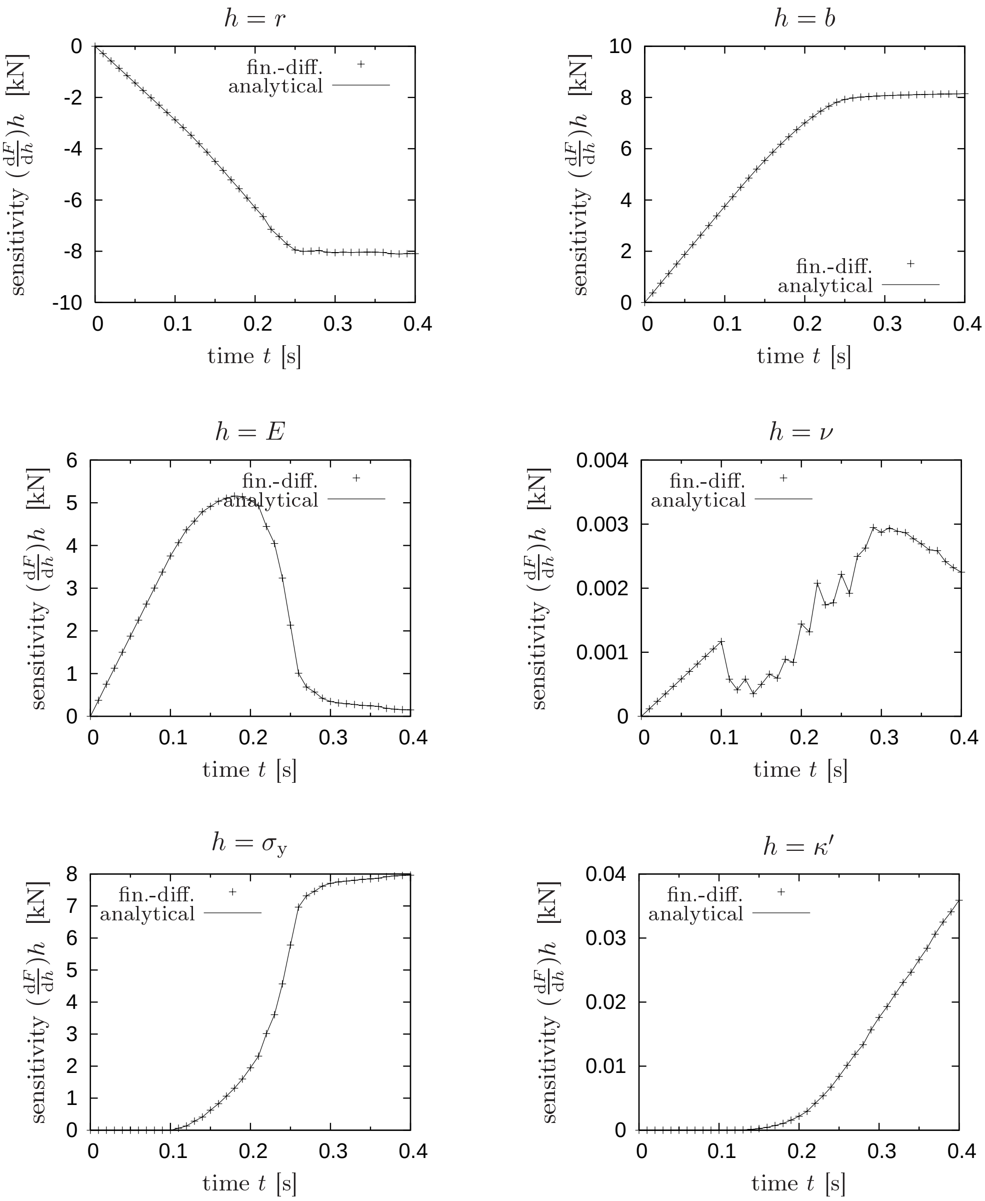


Figure 4.10. Transient normalized sensitivity of the reaction force F with respect to six design parameters. Comparison of analytical design derivatives and their finite-difference approximations

As it could be expected, the response sensitivity with respect to plastic constitutive parameters is initially zero until the first plastic flow occurs in the strip ($t \approx 0.10$ s). Remarkable, in some cases even abrupt changes in the other sensitivity graphs are visible at this time instant, too.

As it can be seen, coincidence of analytical and approximate sensitivity results is perfect. Possible errors limited to the range of line thickness have no importance from the point of view of engineering applications. Thus, the fundamental verification test of correctness of the presented computational algorithm is passed.

A visible feature of the sensitivity graphs is their poor smoothness. This refers particularly to sensitivities of stress σ_B (most design parameters) and reaction F (e.g. $h = \nu$). To deeper investigate this feature, an analysis with a much finer time discretization ($\Delta t = 0.0001$ s) has been run. Figure 4.11 presents transient sensitivity graphs of selected performance design derivatives, with magnified details on which the coarse time discretization solutions have been additionally depicted. Remarkable step-wise patterns correspond to discontinuous 'jumps' of the sensitivity solution at isolated time steps. They result from instantaneous changes of the structural stiffness that occur when subsequent Gauss integration points in finite elements of the model switch from elastic to inelastic behavior or oppositely. In the coarse time discretization, each time increment includes a large number of such discontinuous changes and thus the step-wise patterns are not easily noticeable in the results.

This observation illustrates the discussion on discontinuity of the response sensitivity presented in Section 2.2.4. As can be seen from Figs. 4.8–4.10, occurrences of such discontinuities in the computations do not affect the quality of sensitivity solutions. It must be noted, however, that application of the sensitivity results to prediction of response at finite perturbations of the design parameters may lead to erroneous results if a perturbation is large enough to drive the primary solution onto the opposite side of the neighbouring discontinuity (especially in cases of large discontinuous jumps, like those detected for stress sensitivity, Fig. 4.9, at transition of the point B from elastic to plastic range, occurring between $t = 0.10$ and 0.11).

Identical computational tests have been performed for the two viscoplastic models discussed in Section 3.2.3. For the overstress viscoplasticity, the same material constants have been assumed as in the rate-independent plasticity and, additionally, $m = 5$ and $\mu = 0.0005$ s have been taken. For the power-law strain and strain-rate hardening model, apart of the elastic constants taken as in the rate-independent case, the following constants have been assumed: $\varepsilon_0 = 0.001833$, $n = 10$, $\dot{\varepsilon}_0 = 0.1$ s^{-1} and $m = 0.1$. Primary analysis results of

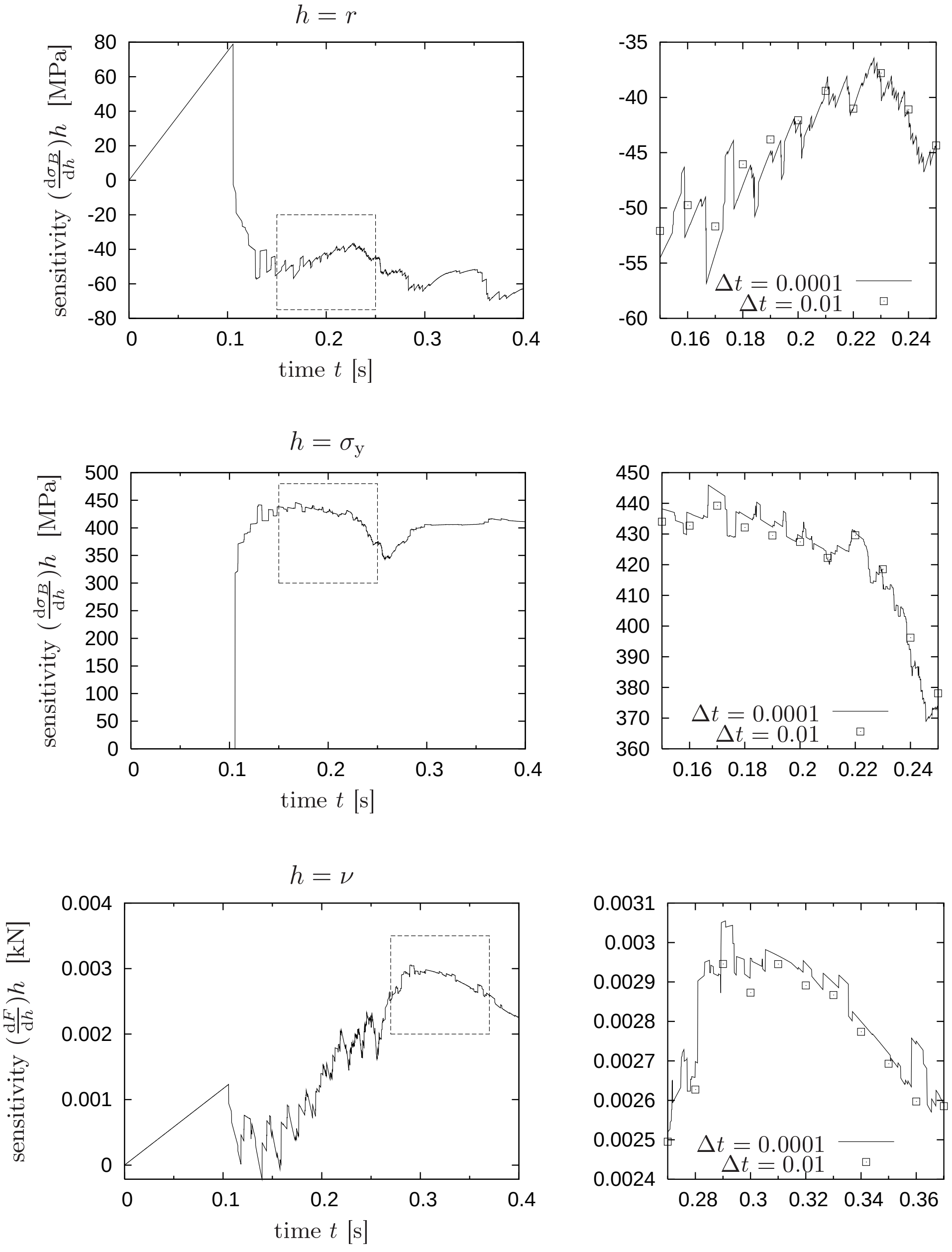


Figure 4.11. Selected design sensitivities computed at fine time discretization, $\Delta t = 0.0001$ s; left: transient results, right: magnified detail including comparison with coarse time discretization results

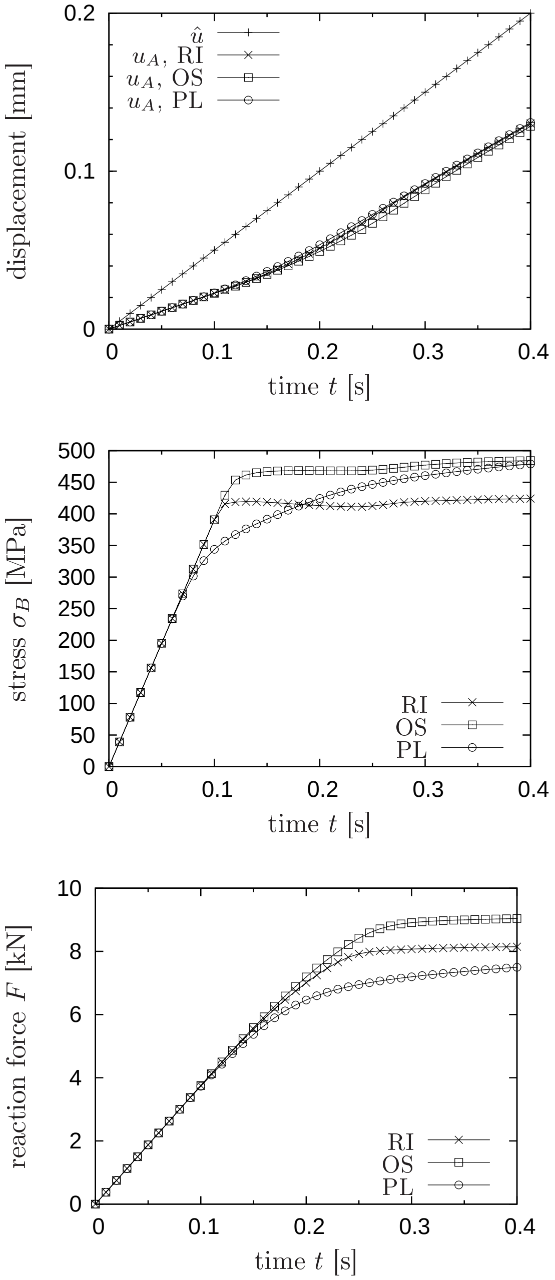


Figure 4.12. Transient results for the displacement u_A, stress σ_B and the reaction force F for various elasto-plastic models: rate-independent (RI), overstress (OS), and power-law hardening (PL)

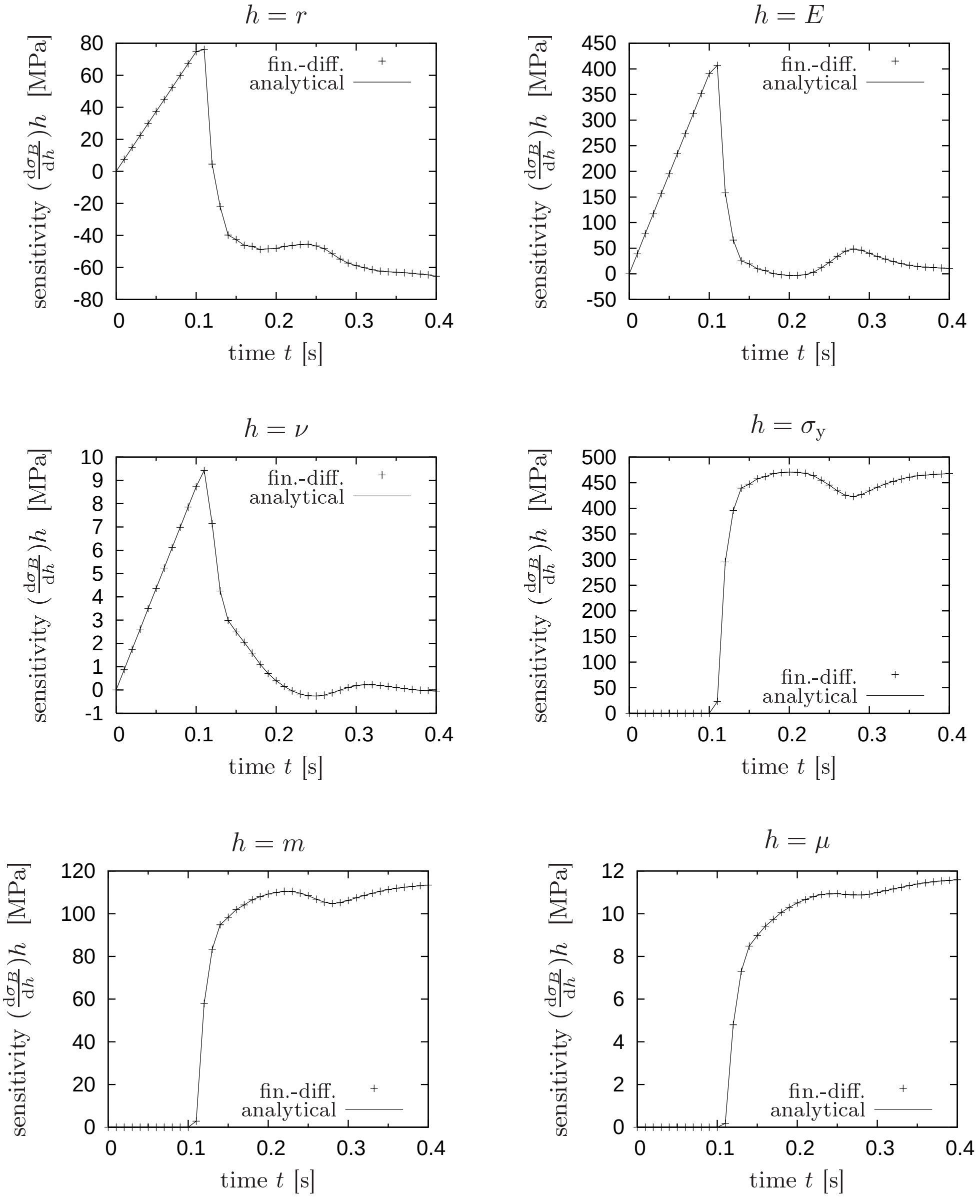


Figure 4.13. Transient normalized sensitivity of the stress σ_B with respect to six design parameters; overstress elasto-viscoplasticity. Comparison of analytical design derivatives and their finite-difference approximations

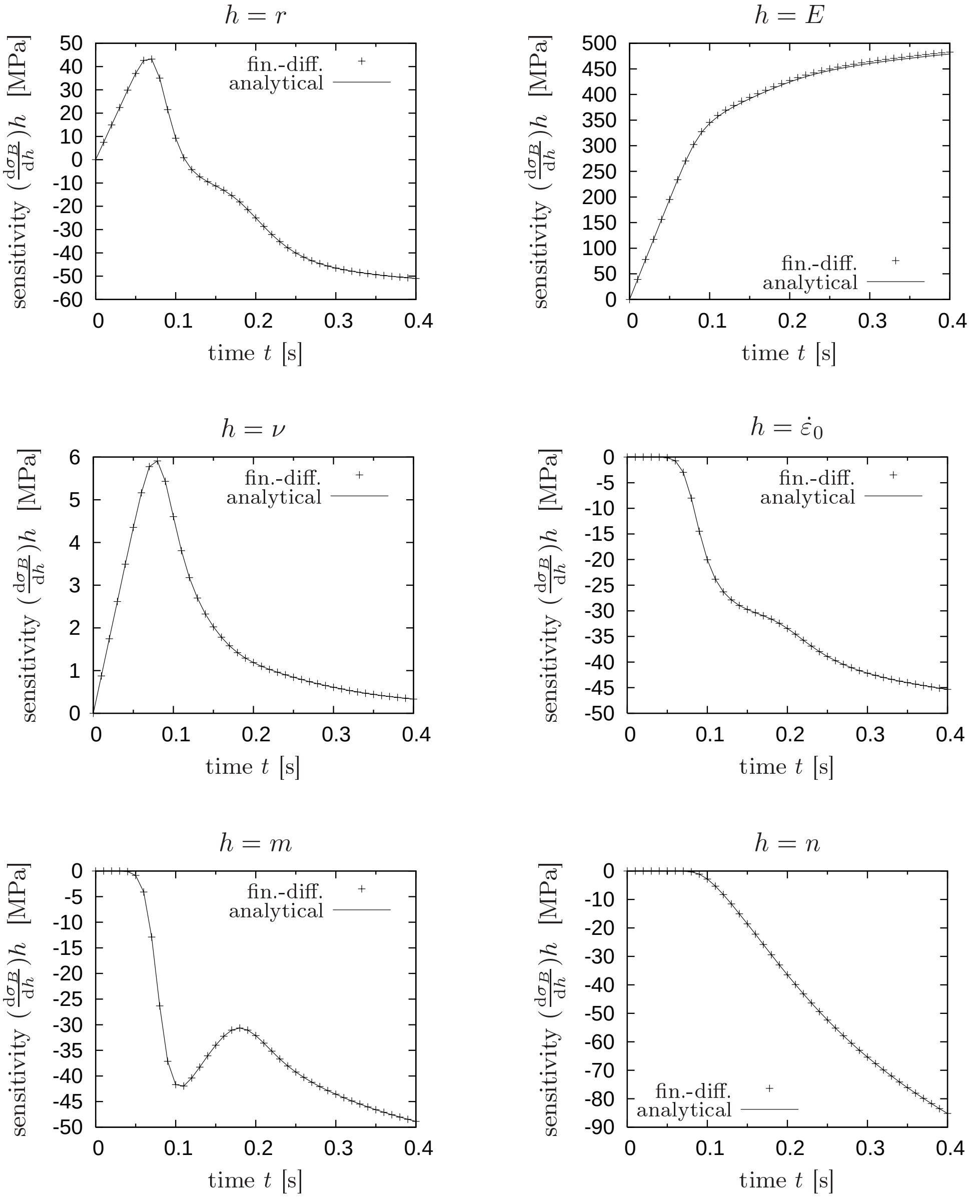


Figure 4.14. Transient normalized sensitivity of the stress σ_B with respect to six design parameters; power-law hardening elasto-viscoplasticity. Comparison of analytical design derivatives and their finite-difference approximations

both the tests compared to the rate-independent model results are presented in Fig. 4.12. Selected sensitivity results (for only σ_B) are presented in Figs. 4.13 and 4.14. Similarly as for the rate-independent elasto-plasticity, perfect coincidence of analytical and finite-difference sensitivity results proves correctness of the implemented DSA algorithm.

Let us finally make a reference to the remark formulated in the end of Section 4.1, on the influence of accuracy of the primary solution on quality of the sensitivity solution. There is a variety of convergence conditions upon which the equilibrium iteration (3.176) can be terminated in practical finite element analysis. Typically, the norm of the corrector array $\delta\mathbf{q}$, or of the residual force array $\mathbf{f}^{\text{res}}$, or their product, is checked and compared to a certain tolerance level $\epsilon > 0$. In our tests, iteration was assumed converged when $(\max_{\alpha=1\ldots N_p} \mathsf{q}_\alpha) < \epsilon = 10^{-5}$. This condition is rather strict — in typical equilibrium analysis, setting ϵ between 10^{-2} and 10^{-3} is usually considered sufficient. But let us look at Fig. 4.15 where both primary and sensitivity results of the example shown in Fig. 4.5, obtained with $\epsilon = 10^{-5}$ and $\epsilon = 5 \cdot 10^{-3}$, are compared. We limit ourselves to only the rate-independent material model and to tracking only the stress σ_B and its sensitivity with respect to the hole radius, $\mathrm{d}\sigma_B/\mathrm{d}r$. From Fig. 4.15a we can see that the primary results are the same (they obviously are not precisely the same, but the differences do not exceed the order of magnitude of the line width). However, we can see visible discrepancies between the graphs of the sensitivity solutions for the two primary accuracy levels. Moreover, since the finite-difference sensitivity approximates for the two accuracy levels are virtually the same, we can conclude that the sensitivity solution obtained with $\epsilon = 5 \cdot 10^{-3}$ is obviously inaccurate, at least at selected time instants.

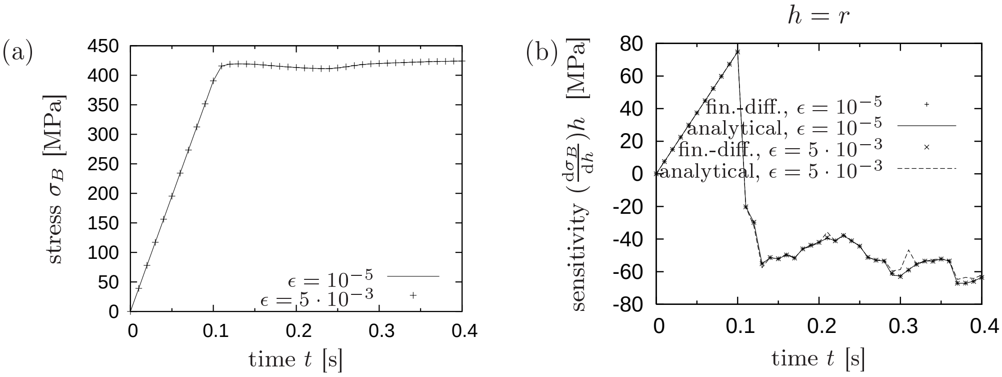


Figure 4.15. Stress σ_B (a) and its transient normalized sensitivity with respect to $h = r$ (b), for different values of equilibrium convergence tolerance

The conclusion suggested by the above example results is that the accuracy level of the primary equilibrium solution, i.e. the convercence tolerance of the Newton's equilibrium iterations, should be set at a much higher level in analyses that are supplemented by sensitivity calculations than in other cases. There is no strict arithmetical rule telling how much higher should this tolerance be, but experience of the author suggests that the decrease of the order of magnitude of ϵ by about 2 with respect to the value sufficient for the primary analysis accuracy should be desirable to achieve comparable accuracy of sensitivity analysis results.

4.3.2. Bending of a carved beam

This is a large-deformation example with the elasto-plastic constitutive model presented in Section 3.2.4 in its time-discrete scheme given in Section 3.3.2. Geometry and loads are shown in Fig. 4.16. An elasto-plastic beam with a half-cylindrical carve is supported at its free ends and bent by a stiff cylinder pressed into the beam's top surface at the centre of its length. No friction is assumed at the cylinder–beam contact area. The dimensions of the beam are: $w = b = 75$ mm, $l = 300$ mm, $r = 33$ mm, and the radius of the stiff cylinder is 25 mm. Deformation is driven by the cylinder's vertical displacement $u_3 = \hat{u}(t)$ monotonically increasing from 0 to 75 mm (Fig. 4.17).

The three constitutive models discussed in Section 3.2.3 are alternatively considered: the rate-independent elasto-plasticity (RI), the overstress (Perzyna–

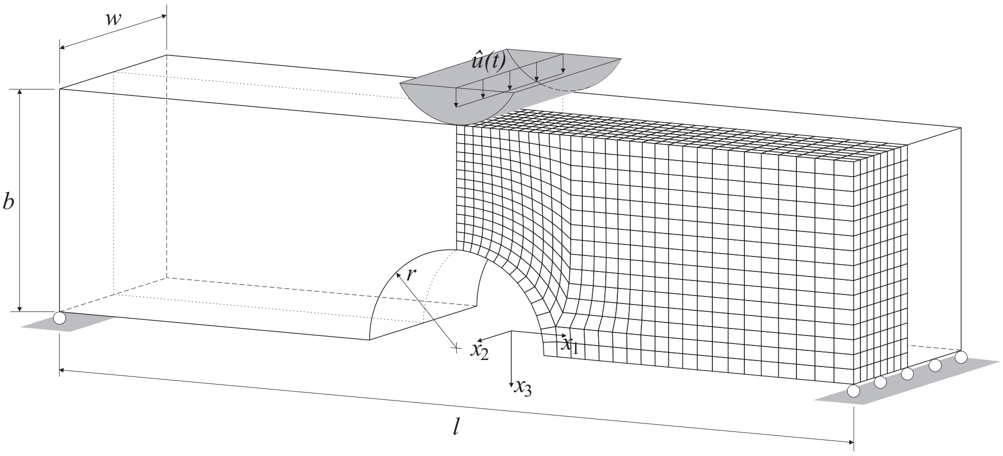


Figure 4.16. Bending of a carved beam: geometry, load and support conditions, and the finite element mesh

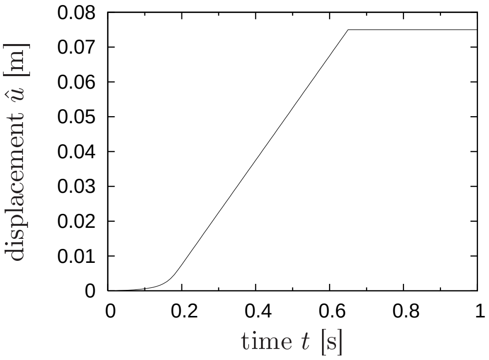


Figure 4.17. Carved beam: load history

Table 4.8. Bending of a carved beam: values of material constants

plasticity model	RI	OS	PL
elastic parameters	$E = 206\,900$ MPa, $\nu = 0.29$		
isotropic hardening parameters	$\sigma_y = \sigma_{y0} = 450$ MPa, $\sigma_{y\infty} = 715$ MPa, $\bar{\kappa}' = 129.24$ MPa, $a = 16.93$		$\varepsilon_0 = 0.002175$, $n = 10$
kinematic hardening parameter	$H' = 0$ MPa		
viscoplastic parameters		$\mu = 0.0001$ s, $m = 5$	$\dot{\varepsilon}_0 = 0.1$ s^{-1}, $m = 0.1$

Prager) elasto-viscoplasticity (OS), and the power-law strain and strain-rate hardening elasto-viscoplasticity (PL). In the first two cases, RI and OS, the nonlinear 'saturation' form of the isotropic hardening function (4.71) is assumed. In all the cases, kinematic hardening is assumed linear. Table 4.8 lists values of all material constants assumed in the analysis for all the three plasticity models.

In view of symmetry, only a quarter of the beam is analysed in the finite element model (Fig. 4.16). Tri-linear brick elements with a selectively reduced spatial integration (the enhanced assumed strain method [88], see Section 3.4.4) have been employed. The resultant support reaction force F (summed up for the entire beam) at various time instants of the deformation history is considered the structural performance. Its evolution, determined for various values of the time increment, is shown in Fig. 4.18. It appears from these graphs that the time increment $\Delta t = 0.01$ s corresponds to convergent results (i.e. differing very little from results at 5 times finer time discretization) for all the three models, and this value has been used in further computations.

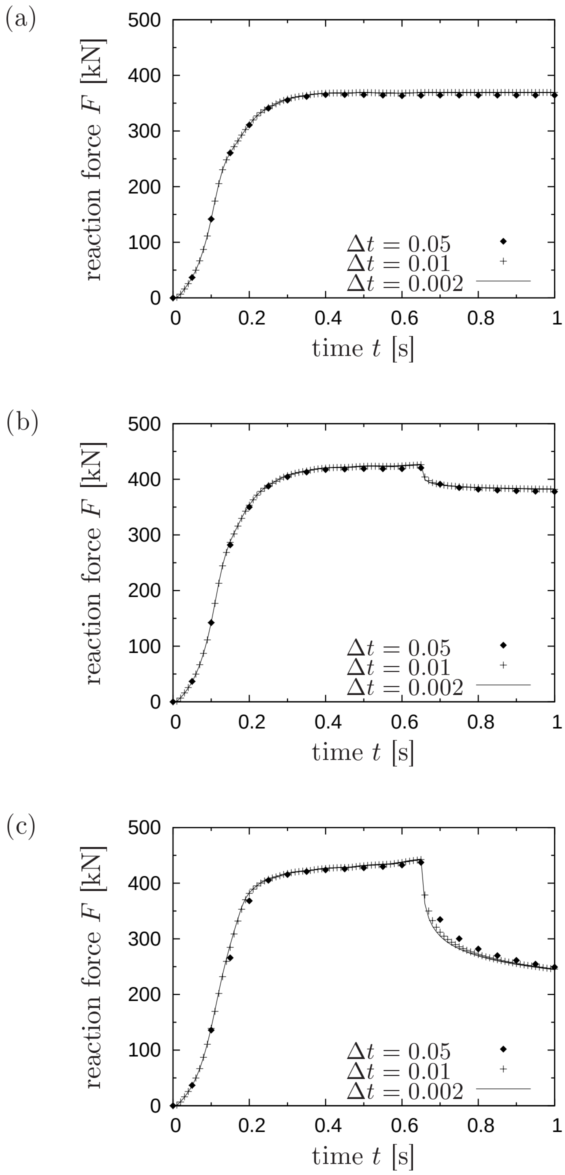


Figure 4.18. Bending of a carved beam: reaction forces for (a) RI model, (b) OS model, and (c) PL model

Figure 4.19 presents deformed configurations of the beam at selected load stages, for the example of the RI model. In the case of RI and OS material models, first plastic flow occurs at $t = 0.09$ s, while at $t > 0.3$ the entire beam's cross-section is plastic. In the case of PL, plastic flow occurs all the time.

Design sensitivity analysis of the solution has been performed along with the primary equilibrium analysis. For each material model, material constants listed in Table 4.8 were considered the design parameters. Besides, the carve radius r was considered a shape design parameter in each analysis.

Figures 4.20–4.22 present graphs of sensitivity (design derivatives) of the resultant reaction force at the beam supports vs. time, for selected design parameters for the three material models, RI, OS and PL. Lines denote analytical solution while symbols – finite difference approximations, obtained with the central difference scheme (4.100) and with the perturbation amount $\Delta h_i \approx 10^{-4} h_i$. The choice of the perturbation amounts was preceded by several trial-and-error tests in order to make sure that they belong to the appropriate range (see discussion in Section 2.2.3). Similarly as in the small deformation example, all the design derivatives have been normalized by multiplication by the primary value of the design parameter, i.e. the graphs display the quantity $(\mathrm{d}F/\mathrm{d}h)\,h$ for both the analytical and finite-difference approximated values. It is evident that analytical and approximate solutions are virtually equal to each other — indeed, the relative error never exceeded $0.005 \times$ the maximum absolute value of sensitivity for each design parameter.

The transient sensitivity graphs are again very nonsmooth. Similarly as in the small-deformation case, this is due to discontinuities of the structural stiffness at transition points between elastic and plastic deformation régimes. Here, an additional source of discontinuities appears, related to entering (or leaving) the active contact zone by subsequent mesh nodes of the beam top surface. The discontinuities have a form of step-wise patterns visible especially in the graphs made for a much finer time discretization. Figure 4.23 presents examples of such graphs, obtained for the rate-independent constitutive model for selected design parameters. As it can be seen, the sensitivity algorithm copes very well with the discontinuous jumps of the sensitivity solution and their presence does not affect the solution quality (understood as compatibility between analytical gradients and their finite-difference approximates).

It was concluded in Section 2.2.2 that one of fundamental advantages of the presented DSA formulation is its numerical efficiency. Linear formulation of sensitivity equations and the ability of reusing the once decomposed system coefficient matrix make the sensitivity computations a numerically cheap add-on to the primary analysis computations. Let us now verify this conclusion

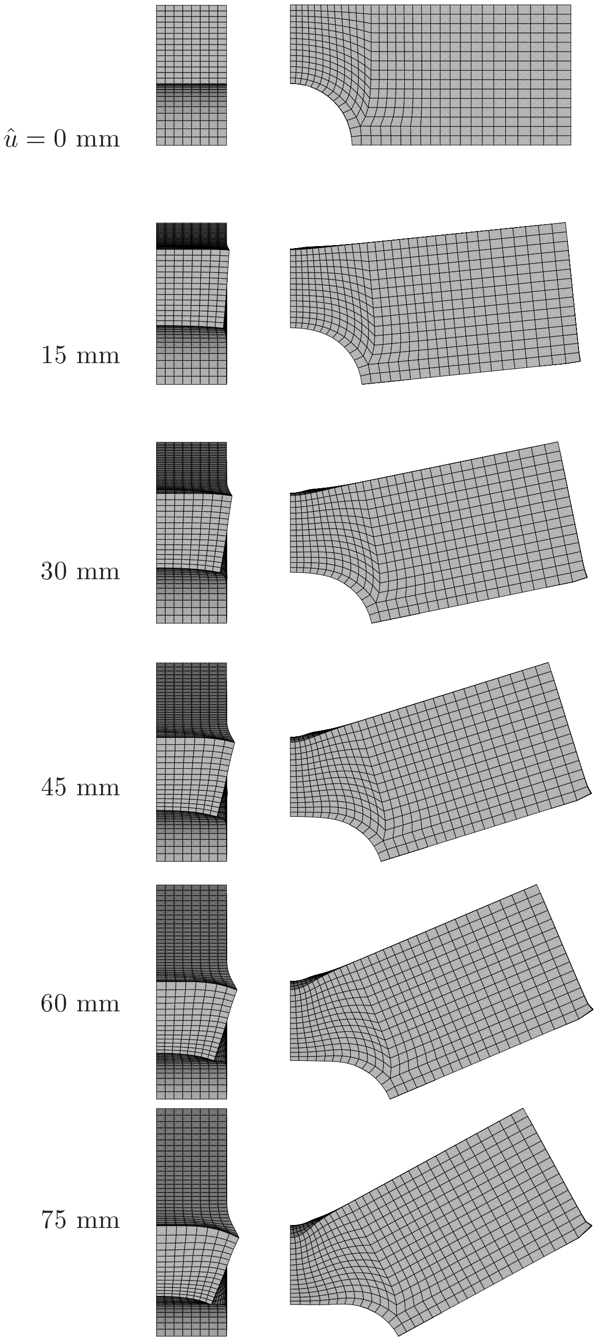


Figure 4.19. Bending of a carved beam: deformation stages

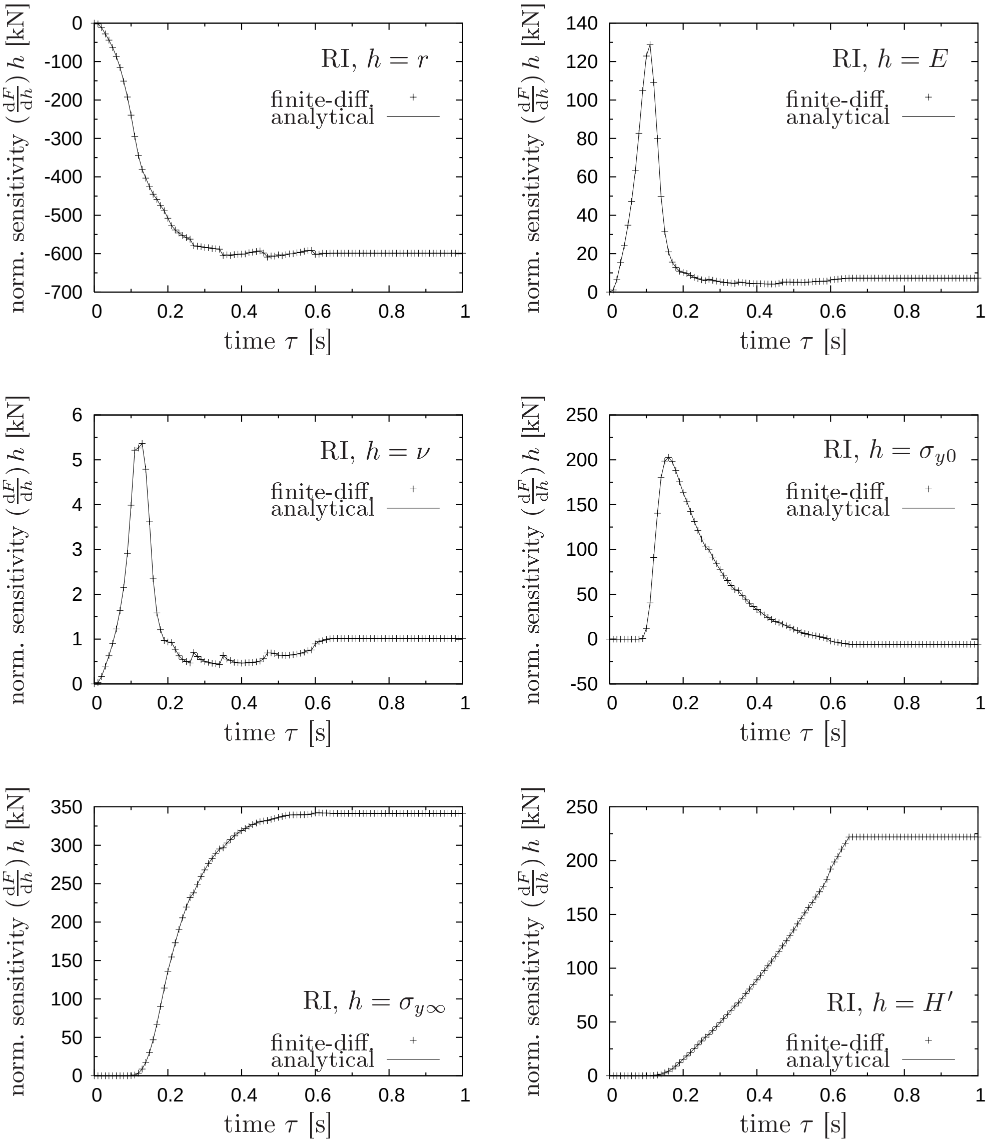


Figure 4.20. Bending of a carved beam: design sensitivities of reaction forces, rate-independent (RI) material model

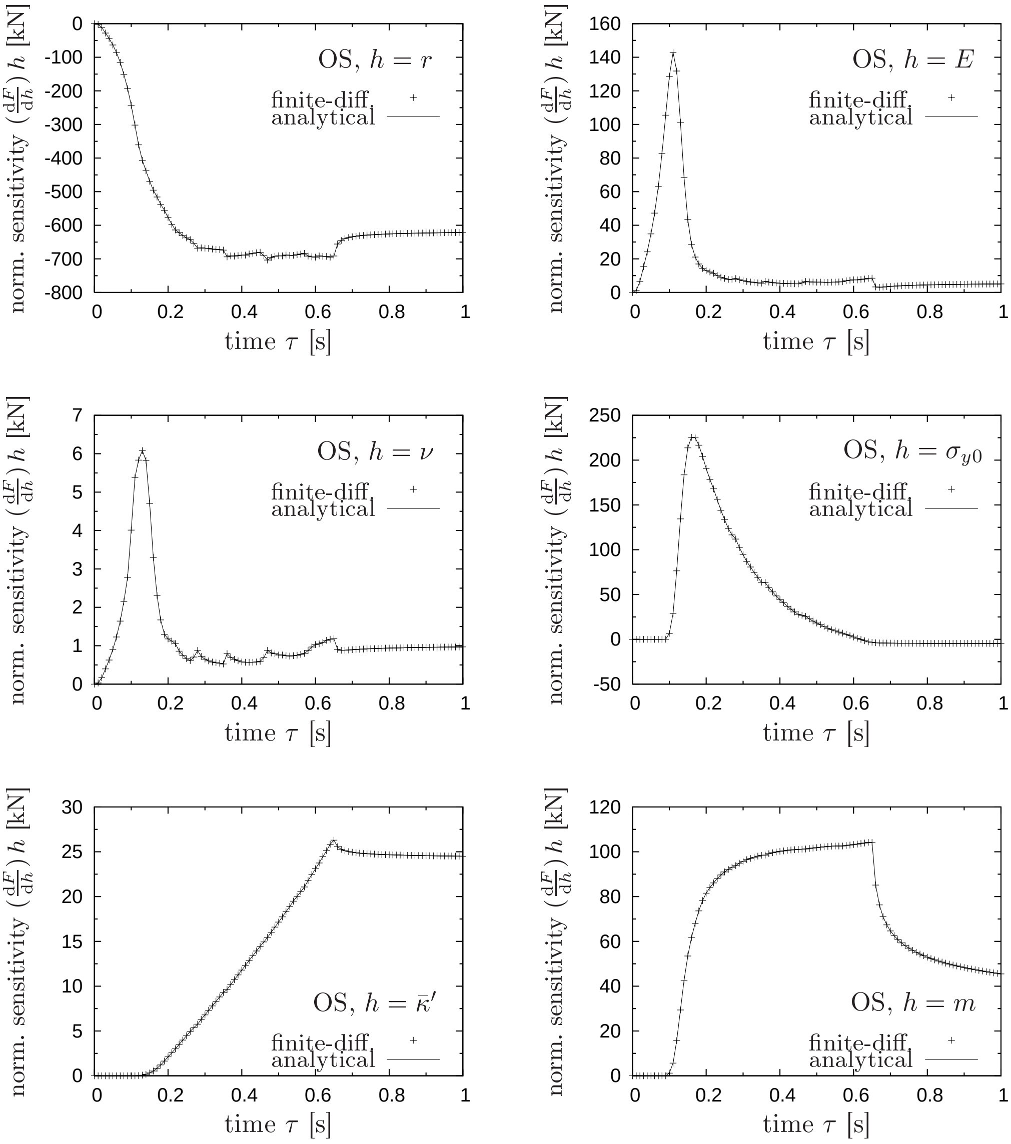


Figure 4.21. Bending of a carved beam: design sensitivities of reaction forces, overstress (OS) material model

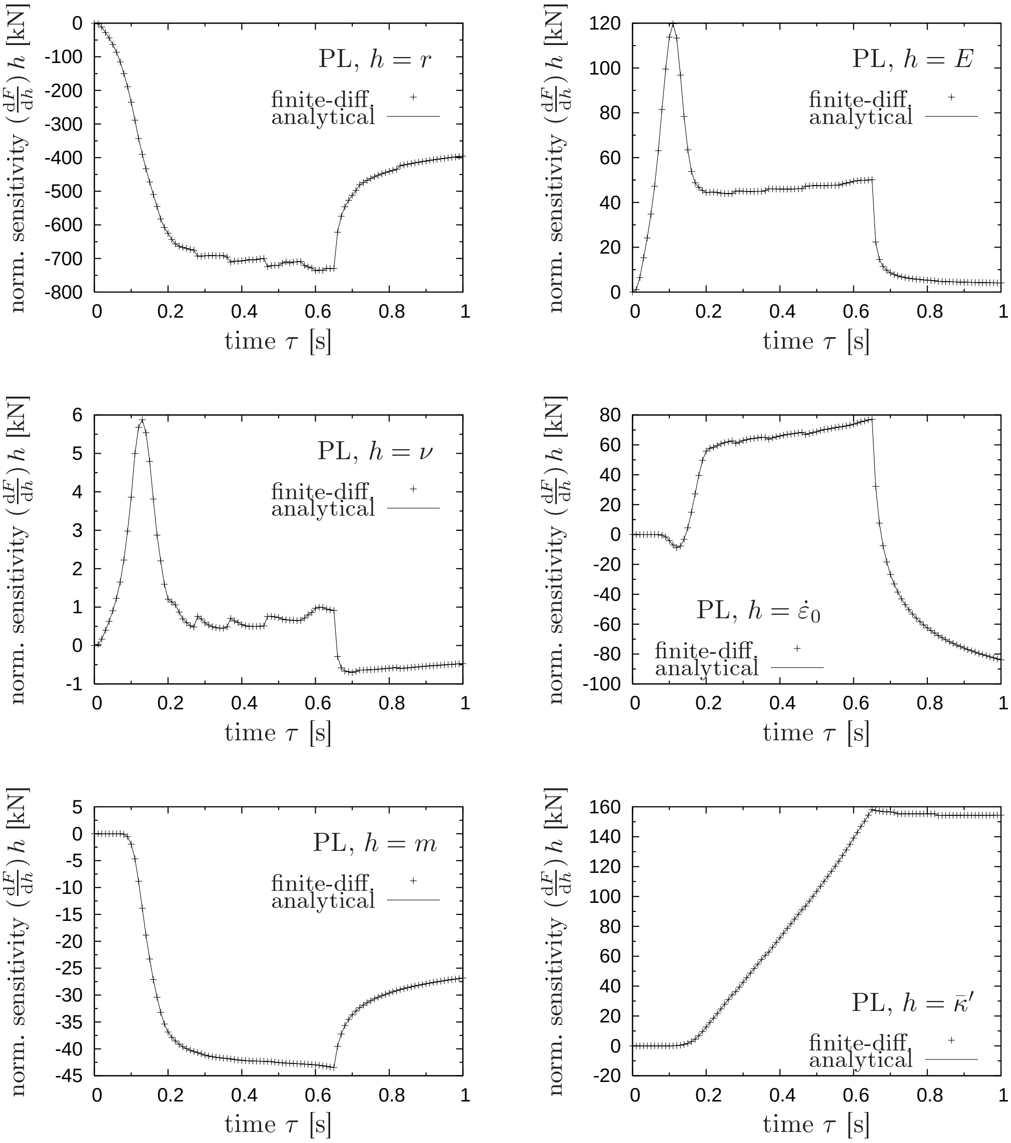


Figure 4.22. Bending of a carved beam: design sensitivities of reaction forces, power law hardening (PL) material model

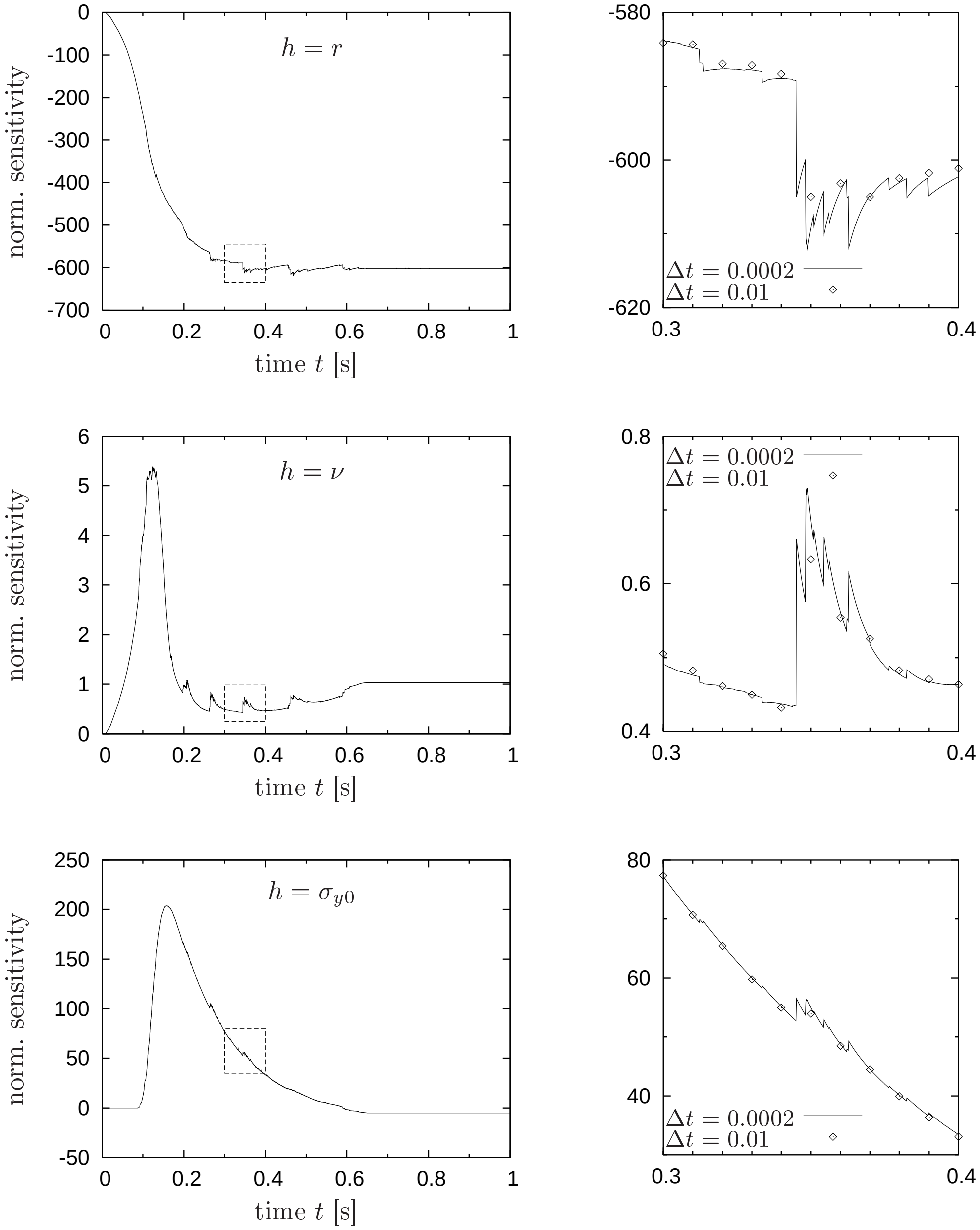


Figure 4.23. Bending of a carved beam: design sensitivities of reaction forces, RI material model, fine time discretization ($\Delta t = 0.0002$ s); left: transient results, right: magnified detail including comparison with coarse time discretization results

in practice by presenting the numerical costs of the sensitivity analysis in the real computational example. The problem presented in this section, Fig. 4.16, with the time range limited to 0.3 s has been taken as the primary problem. Primary and sensitivity analysis has been separately run for 0, 1, 2, 4, 6, and 9 design parameters. Since efficiency of sensitivity computations may depend on whether the design parameters affect geometry or only constitutive constants, the two types of design variables were distinguished and treated separately in the tests. Besides, the same analyses were additionally run for the small deformation formulation.

Let us define the following coefficient of numerical cost associated with sensitivity analysis,

$$\phi_{\mathrm{c}} = \frac{t_{\mathrm{Prim+DSA}} - t_{\mathrm{Prim}}}{t_{\mathrm{Prim}} \cdot N_d}\,, \tag{4.101}$$

where t_{Prim} is the computation time of only primary analysis, $t_{\mathrm{Prim+DSA}}$ is the time of primary and sensitivity analysis run together, and N_d is the number of design parameters. The coefficient ϕ_{c} measures the relative additional computational time necessary to perform sensitivity analysis, per one design variable. For example, if the sensitivity analysis were performed with the finite difference method, cf. Eqs. (2.47), the coefficient ϕ_{c} would take the unit value for the forward and backward difference schemes, and the value of 2 for the central difference scheme (in both the cases, changes of N_d would not affect the value of ϕ_{c} because the difference $t_{\mathrm{Prim+DSA}} - t_{\mathrm{Prim}}$ is proportional to N_d). In the analytical DSA computations, much lower values of the cost coefficient ϕ_{c} are expected.

Figure 4.24 presents the cost analysis results. As it can be seen, the cost coefficient is much less than unity in all cases, and the efficiency of DSA per one design variable increases with the number of design parameters analysed simultaneously in one run. Shape design sensitivity has slightly higher cost than the non-shape one, which is due to more complicated sensitivity formulae (cf. Section 4.1), but the differences are not significant. The efficiency of DSA is to a much more extent affected by the nonlinearity of kinematical description of the continuum motion — we can see that DSA for large deformation problems costs relatively more than for the geometrically linear formulation.

Obviously, the values of ϕ_{c} readable from the graphs correspond to only the particular numerical example analysed and cannot be directly extrapolated to other computational problems. However, the general rules (the higher N_d — the lower ϕ_{c}) and the order of magnitude of the cost coefficient will remain the same in other problems, too. It may generally be also expected that efficiency of

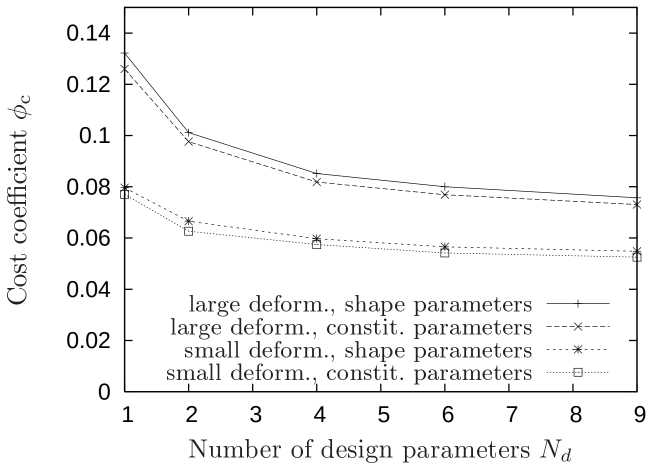


Figure 4.24. Numerical cost of sensitivity computations

DSA can be higher for larger tasks (i.e. larger sizes of the response array $\mathbf{q}$). The quality of implementation (e.g. the way of efficient reusing various intermediate primary results at the stage of DSA computations) may again significantly affect the overall computation time. Anyway, relatively low values of ϕ_c obtained in the tests prove efficiency of DSA compared to the primary analysis.

Chapter 5

Sensitivity analysis of elasto-plastic dynamic problem

In the previous two chapters, primary and sensitivity analysis of nonlinear static equilibrium problems have been investigated and discussed. In this chapter, our considerations are extended towards dynamic analysis of elasto-plastic structures. We are going to remain within the formalism of the finite element method. Our attention is mainly focused on explicit algorithms of dynamic analysis, first because this approach is most popular in practical engineering applications due to its simplicity and efficiency, and also because the sensitivity analysis in the case of explicit dynamic formulation follows a different strategy than in the case of statics or implicit dynamics. The primary solution algorithm is briefly introduced and the formulation of sensitivity analysis is consequently derived. Computational examples illustrate the presented algorithm, demonstrating its abilities and quality of the sensitivity solutions.

5.1. Primary formulation of dynamic analysis

In the dynamic analysis of deformable structures, inertia forces related to material acceleration have to be included in the momentum balance. This results in appearance of the additional term in the virtual work equation (3.6) which now takes the form of the Hamilton variational equation,

$$\int_{\Omega^r} \tilde{\boldsymbol{\sigma}} : \delta\tilde{\boldsymbol{\varepsilon}}\,\mathrm{d}\Omega = \int_{\Omega^r} \hat{\boldsymbol{f}}\,\delta\boldsymbol{u}\,\mathrm{d}\Omega + \int_{\partial\Omega^r} \hat{\boldsymbol{t}}\,\delta\boldsymbol{u}\,\mathrm{d}(\partial\Omega) + \int_{\Omega^r} \rho^r\,\ddot{\boldsymbol{u}}\,\delta\boldsymbol{u}\,\mathrm{d}\Omega\,, \tag{5.1}$$

where ρ^r is material density in the reference configuration $\mathcal{C}^r$.

Applying the finite element discretization (3.163), we approach at the following approximation of the variational principle (5.1),

$$\mathsf{M}_{\alpha\beta}\ddot{\mathsf{q}}_\beta\,\delta\mathsf{q}_\alpha + \mathsf{f}^{\mathrm{int}}_\alpha\,\delta\mathsf{q}_\alpha = \mathsf{f}^{\mathrm{ext}}_\alpha\,\delta\mathsf{q}_\alpha\,, \tag{5.2}$$

with $\mathsf{f}^{\rm int}_{\alpha}$ and $\mathsf{f}^{\rm ext}_{\alpha}$ given by Eqs. (3.166)–(3.167), and

$$\mathsf{M}_{\alpha\beta} = \int_{\Omega^r} \rho^r \phi_{i\alpha}\phi_{i\beta}\,\mathrm{d}\Omega\,. \tag{5.3}$$

In view of arbitrary choice of $\delta\mathsf{q}_{\alpha}$, Eq. (5.2) results in the space-discretized system of differential equations

$$\mathsf{M}_{\alpha\beta}\ddot{\mathsf{q}}_{\beta} + \mathsf{f}^{\rm int}_{\alpha}(\mathsf{q}_{\beta}) = \mathsf{f}^{\rm ext}_{\alpha} \tag{5.4}$$

for the unknown $\mathsf{q}_{\beta}(t)$. For mainly algorithmic reasons, this system is frequently completed with the damping term, yielding finally

$$\mathsf{M}_{\alpha\beta}\ddot{\mathsf{q}}_{\beta} + \mathsf{C}_{\alpha\beta}\dot{\mathsf{q}}_{\beta} + \mathsf{f}^{\rm int}_{\alpha}(\mathsf{q}_{\beta}) = \mathsf{f}^{\rm ext}_{\alpha}\,, \tag{5.5a}$$

or

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{f}^{\rm int}(\mathbf{q}) = \mathbf{f}^{\rm ext}\,, \tag{5.5b}$$

with the matrix $\mathsf{C}_{\alpha\beta}$ typically expressed as a linear function of $\mathsf{M}_{\alpha\beta}$ and/or $\mathsf{K}_{\alpha\beta}$. Let us assume in further derivation of this study that $\mathsf{C}_{\alpha\beta}$ is proportional to $\mathsf{M}_{\alpha\beta}$.

Similarly as in the static analysis, a time discretization is introduced as $t_0{=}0$, t_1, t_2, ... With the use of finite-difference formulae, one can introduce approximate relationships between $\mathbf{q}_k$, $\dot{\mathbf{q}}_k$ and $\ddot{\mathbf{q}}_k$ for different time instants t_k which allow to replace Eq. (5.5) with a series of recursive formulae in which unknown are arrays of $\mathbf{q}_k$ and/or their time derivatives at subsequent particular time instants. There is a variety of ways this can be done and the solution methods can be generally divided into two types: implicit and explicit.

5.1.1. Implicit methods

In the implicit methods, the first and second response time derivatives, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, at a typical time instant are extrapolated as linear functions of $\mathbf{q}$ at this time instant and values of $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ at previous time instants,

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_{n+1}(\mathbf{q}_{n+1}\,, \mathbf{q}_n\,, \dot{\mathbf{q}}_n\,, \ddot{\mathbf{q}}_n\,, \mathbf{q}_{n-1}\,, \dot{\mathbf{q}}_{n-1}\,, \ddot{\mathbf{q}}_{n-1}\,, \ldots), \tag{5.6a}$$

$$\ddot{\mathbf{q}}_{n+1} = \ddot{\mathbf{q}}_{n+1}(\mathbf{q}_{n+1}\,, \mathbf{q}_n\,, \dot{\mathbf{q}}_n\,, \ddot{\mathbf{q}}_n\,, \mathbf{q}_{n-1}\,, \dot{\mathbf{q}}_{n-1}\,, \ddot{\mathbf{q}}_{n-1}\,, \ldots). \tag{5.6b}$$

An example is the Newmark scheme in which

$$\dot{\mathbf{q}}_{n+1} = \frac{1}{\alpha(\Delta t)^2}\left[\mathbf{q}_{n+1} - \mathbf{q}_n - \Delta t\,\dot{\mathbf{q}}_n - (\Delta t)^2\left(\frac{1}{2}-\alpha\right)\ddot{\mathbf{q}}_n\right], \tag{5.7a}$$

$$\ddot{\mathbf{q}}_{n+1} = \frac{\delta}{\alpha\Delta t}\left(\mathbf{q}_{n+1} - \mathbf{q}_n\right) + \left(1-\frac{\delta}{\alpha}\right)\dot{\mathbf{q}}_n + \Delta t\left(1-\frac{\delta}{2\alpha}\right)\ddot{\mathbf{q}}_n\,, \tag{5.7b}$$

with parameters α and δ taken for instance as

$$\alpha = \tfrac{1}{4}\,, \qquad\qquad \delta = \tfrac{1}{2}\,. \tag{5.8}$$

Substituting Eqs. (5.6) to Eq. (5.5) and assuming that all arrays at $t \le t_n$ are known, one obtains a system of nonlinear equations on the unknown $\mathbf{q}_{n+1}$ which can be solved with the use of the Newton iteration scheme as, cf. Eq. (3.176),

$$\begin{tikzcd} \mathbf{K}^{\mathrm{dyn}\,(i)}_{n+1}\,\delta\mathbf{q} = \mathbf{f}^{\mathrm{dyn}\,(i)}_{n+1} \arrow[r] & \mathbf{q}^{(i+1)}_{n+1} = \mathbf{q}^{(i)}_{n+1} + \delta\mathbf{q} \arrow[l, bend left, "i := i+1"] \end{tikzcd} \tag{5.9}$$

where i is the iteration counter, and the array $\mathbf{f}^{\mathrm{dyn}}$ is the dynamic counterpart of the static residual force array $\mathbf{f}^{\mathrm{res}} = \mathbf{f}^{\mathrm{ext}} - \mathbf{f}^{\mathrm{int}}$,

$$\mathbf{f}^{\mathrm{dyn}}_{n+1} = \mathbf{f}^{\mathrm{ext}}_{n+1} - \mathbf{f}^{\mathrm{int}}(\mathbf{q}_{n+1}\,,\, \mathbf{q}_n\,,\, \mathbf{z}_n) - \mathbf{M}\ddot{\mathbf{q}}_{n+1} - \mathbf{C}\dot{\mathbf{q}}_{n+1}\,. \tag{5.10}$$

The ‘effective’ dynamic stiffness matrix $\mathbf{K}^{\mathrm{dyn}}$ is defined as

$$\mathbf{K}^{\mathrm{dyn}}_{n+1} = -\frac{\mathrm{d}\mathbf{f}^{\mathrm{dyn}}_{n+1}}{\mathrm{d}\mathbf{q}_{n+1}} = \mathbf{K}_{n+1} + \mathbf{M}\,\frac{\mathrm{d}\ddot{\mathbf{q}}_{n+1}}{\mathrm{d}\mathbf{q}_{n+1}} + \mathbf{C}\,\frac{\mathrm{d}\dot{\mathbf{q}}_{n+1}}{\mathrm{d}\mathbf{q}_{n+1}} \tag{5.11}$$

where the matrix $\mathbf{K}_{n+1}$ is identical to that of static analysis, i.e.

$$\mathbf{K}_{n+1} = -\frac{\mathrm{d}\mathbf{f}^{\mathrm{res}}_{n+1}}{\mathrm{d}\mathbf{q}_{n+1}} = \frac{\mathrm{d}\mathbf{f}^{\mathrm{int}}}{\mathrm{d}\mathbf{q}_{n+1}}\,.$$

In particular, for the Newmark algorithm, $\mathbf{K}^{\mathrm{dyn}}_{n+1}$ takes the form

$$\mathbf{K}^{\mathrm{dyn}}_{n+1} = \mathbf{K}_{n+1} + \frac{1}{\alpha(\Delta t)^2}\,\mathbf{M} + \frac{\delta}{\alpha\Delta t}\,\mathbf{C}\,. \tag{5.12}$$

The scheme allows to proceed with the solution over subsequent time instants of the assumed discretization. Stability of the solution depends on the particular scheme type and generally is guaranteed for only sufficiently small Δt; however, the Newmark scheme with the parameter values (5.8) is unconditionally stable.

Comparison of the implicit method of dynamics and statics reveals very close relations between each other. In fact, only redefinition of the residual force vector (5.10) and the stiffness matrix (5.11) is necessary to practically apply the static algorithm presented in Chapter 3 to the dynamic analysis. The implicit methods thus appear to be a simple extension of the static analysis methods which includes inertia forces as additional external load forces in the discretized equilibrium equation system.

5.1.2. Explicit methods

In the explicit methods, the response $\mathbf{q}$ and its first time derivative $\dot{\mathbf{q}}$ at a typical time instant are extrapolated as linear functions of $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ at previous time instants,

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_{n+1}(\mathbf{q}_n\,,\,\dot{\mathbf{q}}_n\,,\,\ddot{\mathbf{q}}_n\,,\,\mathbf{q}_{n-1}\,,\,\dot{\mathbf{q}}_{n-1}\,,\,\ddot{\mathbf{q}}_{n-1}\,,\,\ldots), \tag{5.13a}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_{n+1}(\mathbf{q}_n\,,\,\dot{\mathbf{q}}_n\,,\,\ddot{\mathbf{q}}_n\,,\,\mathbf{q}_{n-1}\,,\,\dot{\mathbf{q}}_{n-1}\,,\,\ddot{\mathbf{q}}_{n-1}\,,\,\ldots). \tag{5.13b}$$

Let us consider the following, most typical example

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + \Delta t\,\ddot{\mathbf{q}}_n\,, \qquad \mathbf{q}_{n+1} = \mathbf{q}_n + \Delta t\,\dot{\mathbf{q}}_{n+1}\,. \tag{5.14}$$

Substituting Eqs. (5.13) to Eq. (5.5) and assuming that all arrays at $t \leq t_n$ are known, one obtains a system of linear equations for the unknown $\ddot{\mathbf{q}}_{n+1}$, with the coefficient matrix $\mathbf{M}$,

$$\mathbf{M}\ddot{\mathbf{q}}_{n+1} = \mathbf{f}^{\text{ext}}_{n+1} - \mathbf{f}^{\text{int}}(\mathbf{q}_{n+1}\,,\,\mathbf{q}_n\,,\,\mathbf{z}_n) - \mathbf{C}\dot{\mathbf{q}}_{n+1}\,. \tag{5.15}$$

The scheme allows to proceed with the solution through subsequent time instants of the assumed discretization. It is only conditionally stable, for usually very small values of Δt. However, since it is linear, and since the matrix $\mathbf{M}$ can be replaced without much loss of accuracy with its diagonal (lumped) approximate

$$\mathsf{M}^l_{\alpha\alpha} = \sum_{\beta=1}^{N_{np}} \mathsf{M}_{\alpha\beta}\,, \tag{5.16}$$

the approach appears to be very efficient and is the base to most of the practical industrial applications of dynamic analysis of structures. Solution of a linear system of equations with a diagonal matrix is so cheap from the point of view of computational time that it is usually not a problem if this procedure must be repeated even thousands of times, for a large number of very small time increments.

5.2. Sensitivity formulation of dynamic analysis

To derive the analytical formulation of sensitivity analysis for dynamic problems, let us differentiate Eq. (5.5) with respect to design as

$$\partial_{\mathbf{h}}\mathbf{M}\,\ddot{\mathbf{q}} + \mathbf{M}\,\mathrm{d}_{\mathbf{h}}\ddot{\mathbf{q}} + \partial_{\mathbf{h}}\mathbf{C}\,\dot{\mathbf{q}} + \mathbf{C}\,\mathrm{d}_{\mathbf{h}}\dot{\mathbf{q}} + \mathrm{d}_{\mathbf{h}}\mathbf{f}^{\text{int}} = \partial_{\mathbf{h}}\mathbf{f}^{\text{ext}}\,. \tag{5.17}$$

The design derivatives that are explicitly known from the input data have been displayed as $\partial_{\mathbf{h}}(\cdot)$ in the above equation.

Sensitivity of the mass matrix $\partial_{\mathbf{h}}\mathbf{M}$ can be derived from Eq. (5.3). For design parameters that do not affect geometry, we have

$$\partial_{\mathbf{h}}\mathsf{M}_{\alpha\beta} = \int_{\Omega^r} \partial_{\mathbf{h}}\rho^r\, \phi_{i\alpha}\phi_{i\beta}\, \mathrm{d}\Omega\,. \tag{5.18}$$

In the case of shape sensitivity, for the interpolation functions defined according to the isoparametric element concept (4.26)–(4.27), we integration over Ω^r is replaced with integration in the design-independent parent configuration,

$$\partial_{\mathbf{h}}\mathsf{M}_{\alpha\beta} = \int_{\Omega^p} \partial_{\mathbf{h}}(\rho^r\phi_{i\alpha}\phi_{i\beta})\, \hat{J}\mathrm{d}\Omega\,, \tag{5.19}$$

which, in view of Eq. (4.31), assumes finally the same form as in the case of design-independent geometry (5.18). Sensitivity of the damping matrix $\partial_{\mathbf{h}}\mathbf{C}$ is, according to our previous assumption, a given function of $\partial_{\mathbf{h}}\mathbf{M}$.

Depending on whether implicit or explicit approach to the primary dynamic analysis has been utilized, the strategy of sensitivity analysis is different.

5.2.1. Implicit methods

In implicit formulations, the approach to DSA is nearly the same as in the case of static analysis, discussed in Chapter 4. Assuming the solution at $t = t_{n+1}$ is known and all design derivatives of the response $\mathbf{q}$ and its time derivatives $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ for $t \leq t_n$ are available from the previous computations, we can differentiate Eqs. (5.6) and substitute the results to Eq. (5.17) at $t = t_{n+1}$. We obtain the following linear equation, cf. e.g. [135],

$$\mathbf{K}^{\mathrm{dyn}}_{n+1}\, \mathrm{d}_{\mathbf{h}}\mathbf{q}_{n+1} = \hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{dyn}}_{n+1}\,, \tag{5.20}$$

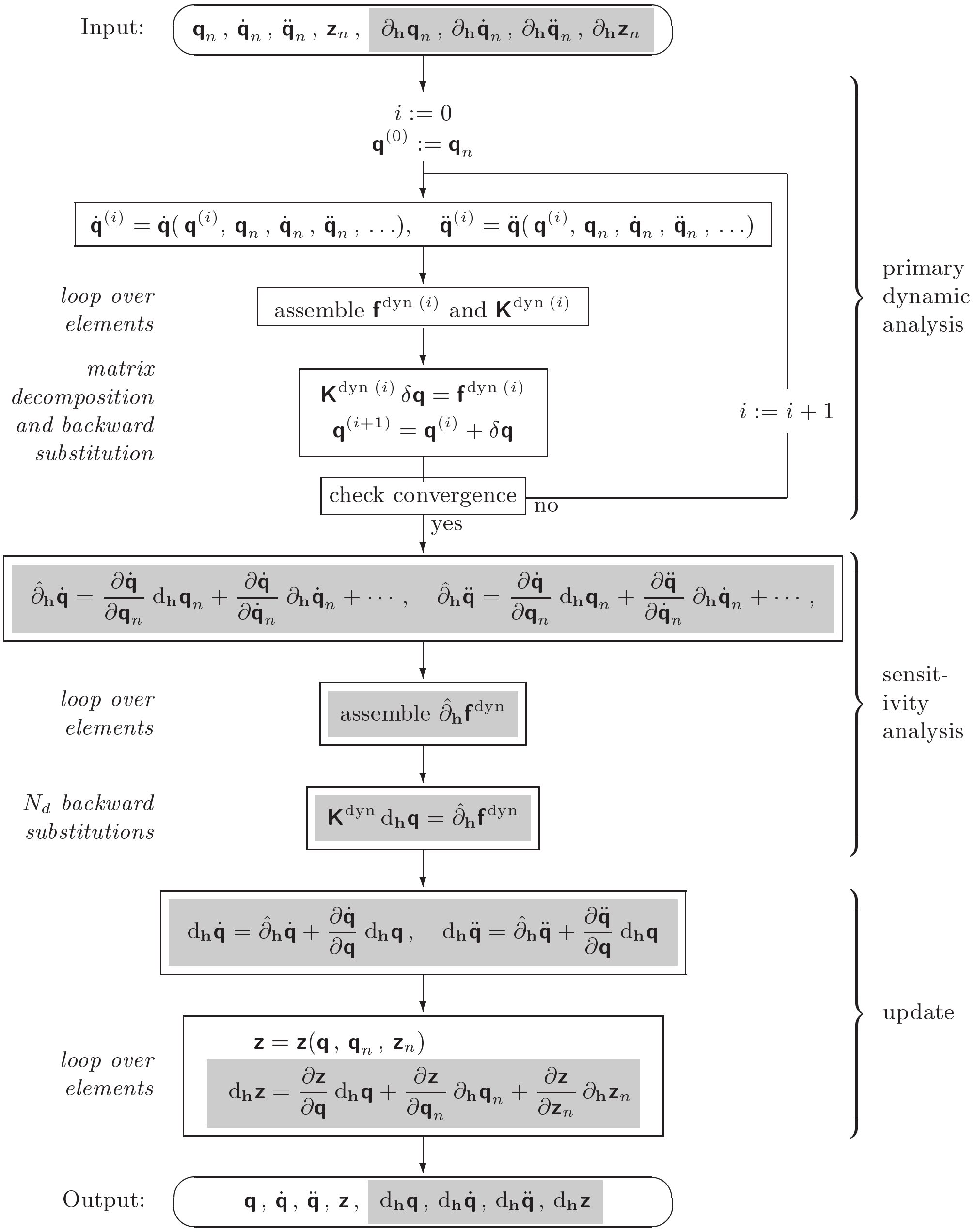
where

$$\begin{aligned}\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{dyn}}_{n+1} \;\equiv\; \mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\mathbf{f}^{\mathrm{dyn}}_{n+1} &= \partial_{\mathbf{h}}\mathbf{f}^{\mathrm{ext}}_{n+1} - \hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{int}}_{n+1} \\ &\quad - \partial_{\mathbf{h}}\mathbf{M}\,\ddot{\mathbf{q}}_{n+1} - \mathbf{M}\,\hat{\partial}_{\mathbf{h}}\ddot{\mathbf{q}}_{n+1} - \partial_{\mathbf{h}}\mathbf{C}\,\dot{\mathbf{q}}_{n+1} - \mathbf{C}\,\hat{\partial}_{\mathbf{h}}\dot{\mathbf{q}}_{n+1}\,,\end{aligned} \tag{5.21}$$

with $\partial_{\mathbf{h}}\mathbf{f}^{\mathrm{ext}}_{n+1}$ and $\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{int}}_{n+1}$ computed according to the derivations of Chapter 4, and, cf. Eqs. (5.6),

$$\hat{\partial}_{\mathbf{h}}\dot{\mathbf{q}}_{n+1} = \frac{\partial\dot{\mathbf{q}}_{n+1}}{\partial\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \frac{\partial\dot{\mathbf{q}}_{n+1}}{\partial\dot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n + \frac{\partial\dot{\mathbf{q}}_{n+1}}{\partial\ddot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\ddot{\mathbf{q}}_n + \frac{\partial\ddot{\mathbf{q}}_{n+1}}{\partial\mathbf{q}_{n-1}}\,\partial_{\mathbf{h}}\mathbf{q}_{n-1} + \cdots, \tag{5.22a}$$

Table 5.1. General scheme of computations for both the primary implicit dynamic and the design sensitivity analysis at a typical time step $[t_n\,,\,t_{n+1}]$ (DSA modifications to the pure primary algorithm are gray-shaded). Indices $n{+}1$ are skipped for better legibility


Input:
$\mathbf{q}_n\,,\,\dot{\mathbf{q}}_n\,,\,\ddot{\mathbf{q}}_n\,,\,\mathbf{z}_n\,,\;\partial_{\mathbf{h}}\mathbf{q}_n\,,\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n\,,\,\partial_{\mathbf{h}}\ddot{\mathbf{q}}_n\,,\,\partial_{\mathbf{h}}\mathbf{z}_n$
$i := 0$
$\mathbf{q}^{(0)} := \mathbf{q}_n$
$\dot{\mathbf{q}}^{(i)} = \dot{\mathbf{q}}(\,\mathbf{q}^{(i)},\,\mathbf{q}_n\,,\,\dot{\mathbf{q}}_n\,,\,\ddot{\mathbf{q}}_n\,,\ldots),\quad \ddot{\mathbf{q}}^{(i)} = \ddot{\mathbf{q}}(\,\mathbf{q}^{(i)},\,\mathbf{q}_n\,,\,\dot{\mathbf{q}}_n\,,\,\ddot{\mathbf{q}}_n\,,\ldots)$
loop over elements
assemble $\mathbf{f}^{\mathrm{dyn}\,(i)}$ and $\mathbf{K}^{\mathrm{dyn}\,(i)}$
matrix decomposition and backward substitution
$\mathbf{K}^{\mathrm{dyn}\,(i)}\,\delta\mathbf{q} = \mathbf{f}^{\mathrm{dyn}\,(i)}$
$\mathbf{q}^{(i+1)} = \mathbf{q}^{(i)} + \delta\mathbf{q}$
$i := i+1$
check convergence
no
yes
primary dynamic analysis
$\hat{\partial}_{\mathbf{h}}\dot{\mathbf{q}} = \dfrac{\partial\dot{\mathbf{q}}}{\partial\mathbf{q}_n}\,\mathrm{d}_{\mathbf{h}}\mathbf{q}_n + \dfrac{\partial\dot{\mathbf{q}}}{\partial\dot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n + \cdots,\quad \hat{\partial}_{\mathbf{h}}\ddot{\mathbf{q}} = \dfrac{\partial\dot{\mathbf{q}}}{\partial\mathbf{q}_n}\,\mathrm{d}_{\mathbf{h}}\mathbf{q}_n + \dfrac{\partial\ddot{\mathbf{q}}}{\partial\dot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n + \cdots,$
loop over elements
assemble $\hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{dyn}}$
N_d backward substitutions
$\mathbf{K}^{\mathrm{dyn}}\,\mathrm{d}_{\mathbf{h}}\mathbf{q} = \hat{\partial}_{\mathbf{h}}\mathbf{f}^{\mathrm{dyn}}$
sensitivity analysis
$\mathrm{d}_{\mathbf{h}}\dot{\mathbf{q}} = \hat{\partial}_{\mathbf{h}}\dot{\mathbf{q}} + \dfrac{\partial\dot{\mathbf{q}}}{\partial\mathbf{q}}\,\mathrm{d}_{\mathbf{h}}\mathbf{q}\,,\quad \mathrm{d}_{\mathbf{h}}\ddot{\mathbf{q}} = \hat{\partial}_{\mathbf{h}}\ddot{\mathbf{q}} + \dfrac{\partial\ddot{\mathbf{q}}}{\partial\mathbf{q}}\,\mathrm{d}_{\mathbf{h}}\mathbf{q}$
loop over elements
$\mathbf{z} = \mathbf{z}(\mathbf{q}\,,\,\mathbf{q}_n\,,\,\mathbf{z}_n)$
$\mathrm{d}_{\mathbf{h}}\mathbf{z} = \dfrac{\partial\mathbf{z}}{\partial\mathbf{q}}\,\mathrm{d}_{\mathbf{h}}\mathbf{q} + \dfrac{\partial\mathbf{z}}{\partial\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \dfrac{\partial\mathbf{z}}{\partial\mathbf{z}_n}\,\partial_{\mathbf{h}}\mathbf{z}_n$
update
Output:
$\mathbf{q}\,,\,\dot{\mathbf{q}}\,,\,\ddot{\mathbf{q}}\,,\,\mathbf{z}\,,\;\mathrm{d}_{\mathbf{h}}\mathbf{q}\,,\,\mathrm{d}_{\mathbf{h}}\dot{\mathbf{q}}\,,\,\mathrm{d}_{\mathbf{h}}\ddot{\mathbf{q}}\,,\,\mathrm{d}_{\mathbf{h}}\mathbf{z}$

$$\hat{\partial}_{\mathbf{h}}\ddot{\mathbf{q}}_{n+1} = \frac{\partial\ddot{\mathbf{q}}_{n+1}}{\partial\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \frac{\partial\ddot{\mathbf{q}}_{n+1}}{\partial\dot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n + \frac{\partial\ddot{\mathbf{q}}_{n+1}}{\partial\ddot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\ddot{\mathbf{q}}_n + \frac{\partial\dot{\mathbf{q}}_{n+1}}{\partial\mathbf{q}_{n-1}}\,\partial_{\mathbf{h}}\mathbf{q}_{n-1} + \cdots . \tag{5.22b}$$

In the particular case of the Newmark scheme (5.7),

$$\hat{\partial}_{\mathbf{h}}\dot{\mathbf{q}}_{n+1} = -\frac{1}{\alpha(\Delta t)^2}\left[\partial_{\mathbf{h}}\mathbf{q}_n + \Delta t\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n + (\Delta t)^2\left(\frac{1}{2}-\alpha\right)\partial_{\mathbf{h}}\ddot{\mathbf{q}}_n\right], \tag{5.23a}$$

$$\hat{\partial}_{\mathbf{h}}\ddot{\mathbf{q}}_{n+1} = -\frac{\delta}{\alpha\Delta t}\,\partial_{\mathbf{h}}\mathbf{q}_n + \left(1-\frac{\delta}{\alpha}\right)\partial_{\mathbf{h}}\dot{\mathbf{q}}_n + \Delta t\left(1-\frac{\delta}{2\alpha}\right)\partial_{\mathbf{h}}\ddot{\mathbf{q}}_n\,. \tag{5.23b}$$

As it can be seen from Eqs. (5.20)–(5.21), whose form is strikingly similar to the form of Eqs. (4.3) and (4.8), computations of sensitivity analysis follow the same strategy as in the case of static problems, i.e. they have to be performed step by step right after completion of the primary analysis (cf. Table 4.1). The only additional operations needed in the case of dynamics are computation of design derivatives of the mass matrix $\partial_{\mathbf{h}}\mathbf{M}$ (5.18) and, accordingly, of the damping matrix $\partial_{\mathbf{h}}\mathbf{C}$, and design-differentiation of the finite-difference time interpolation formulae. Table 5.1 presents the scheme of computations for both the primary dynamic and the design sensitivity analysis at a typical time step $[t_n\,,\,t_{n+1}]$. Differences compared to Table 4.1 appear to be minor — the 'statical' algorithm is only extended by a few additional operations.

We can also recall and repeat here all conclusions drawn for the static case about efficiency of the sensitivity algorithm compared to that of primary analysis. Again, the sensitivity equations system at each time step is linear (contrary to the primary analysis) and it reuses the same coefficient matrix as that has just been used in the last Newton iteration of primary analysis, which significantly decreases the necessary computation effort.

5.2.2. Explicit methods

In the explicit methods, differentiation of Eqs. (5.13) allows to explicitly determine design derivatives of $\dot{\mathbf{q}}_{n+1}$ and $\mathbf{q}_{n+1}$ as functions of already known derivatives from the previous computations,

$$\mathrm{d}_{\mathbf{h}}\dot{\mathbf{q}}_{n+1} = \frac{\partial\dot{\mathbf{q}}_{n+1}}{\partial\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \frac{\partial\dot{\mathbf{q}}_{n+1}}{\partial\dot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n + \frac{\partial\dot{\mathbf{q}}_{n+1}}{\partial\ddot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\ddot{\mathbf{q}}_n + \cdots\,, \tag{5.24a}$$

$$\mathrm{d}_{\mathbf{h}}\mathbf{q}_{n+1} = \frac{\partial\mathbf{q}_{n+1}}{\partial\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \frac{\partial\mathbf{q}_{n+1}}{\partial\dot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n + \frac{\partial\mathbf{q}_{n+1}}{\partial\ddot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\ddot{\mathbf{q}}_n + \cdots\,, \tag{5.24b}$$

and thus to denote them with the symbol $\partial_{\mathbf{h}}(\cdot)$, reserved for explicit design derivatives. Substitution to (5.17) leads to the following system of equations at t_{n+1} ,

$$\mathbf{M}\,\mathrm{d}_{\mathbf{h}}\ddot{\mathbf{q}}_{n+1} = \partial_{\mathbf{h}}\mathbf{f}^{\text{ext}}_{n+1} - \mathrm{d}_{\mathbf{h}}\mathbf{f}^{\text{int}}_{n+1} - \partial_{\mathbf{h}}\mathbf{M}\,\ddot{\mathbf{q}}_{n+1} - \partial_{\mathbf{h}}\mathbf{C}\,\dot{\mathbf{q}}_{n+1} - \mathbf{C}\,\mathrm{d}_{\mathbf{h}}\dot{\mathbf{q}}_{n+1}\,, \qquad (5.25)$$

in which the right-hand side is explicitly known after completion of the primary computations at $t = t_{n+1}$ and the coefficient matrix is the same as in the primary computations. If the diagonal, lumped mass matrix (5.16) has been used in the primary computations, then it replaces the 'full' one in Eq. (5.25) as well, which further implies that $\partial_{\mathbf{h}}\mathbf{M}$ has to be also replaced with its lumped counterpart $\partial_{\mathbf{h}}\mathbf{M}^l$. In such a case, Eq. (5.25) is a simple linear system of equations with a diagonal coefficient matrix.

It can be seen that, similarly as in statics and implicit dynamics, the sensitivity computations have to be performed step-by-step right after the primary computations. The matrices $\partial_{\mathbf{h}}\mathbf{M}$ and $\partial_{\mathbf{h}}\mathbf{C}$ have to be determined in the same way as in the implicit approach. The new array $\mathrm{d}_{\mathbf{h}}\mathbf{f}^{\text{int}}_{n+1}$, containing total design derivatives (computed at known $\mathrm{d}_{\mathbf{h}}\mathbf{q}_{n+1}$), has not appeared in the static and implicit dynamic formulations. It can be, however, explicitly determined, following the formulae that appeared in this study as auxiliary equations during derivation of the explicit partial derivatives $\mathrm{d}_{\mathbf{h}!\mathbf{q}_{n+1}}\mathbf{f}^{\text{int}}_{n+1}$ in Chapter 4. For design-independent geometry, we have, cf. Eqs. (3.167) and (4.9),

$$\mathrm{d}_{\mathbf{h}}\mathsf{f}^{\text{int}}_{\alpha} = \int_{\Omega^r} \left(\mathrm{d}_{\mathbf{h}}\tilde{\sigma}_{ij}\,\tilde{\mathsf{B}}_{ij\alpha} + \tilde{\sigma}_{ij}\,\mathrm{d}_{\mathbf{h}}\tilde{\mathsf{B}}_{ij\alpha}\right)\mathrm{d}\Omega\,, \qquad (5.26)$$

(with the subscripts $n+1$ skipped), where the total derivatives $\mathrm{d}_{\mathbf{h}}\tilde{\mathsf{B}}_{ij\alpha}$ have been given in Section 4.1.2 for different types of geometric description, while the total derivatives $\mathrm{d}_{\mathbf{h}}\tilde{\sigma}_{ij}$ have been given in Section 4.2 for different types of geometric description and for different constitutive models of elasto-plasticity. In the case of shape sensitivity, for the interpolation functions defined according to the isoparametric element concept (4.26)–(4.27), Eq. (5.26) has to be only completed with one additional term expressing design-dependence of the integration domain, cf. Eq. (4.29),

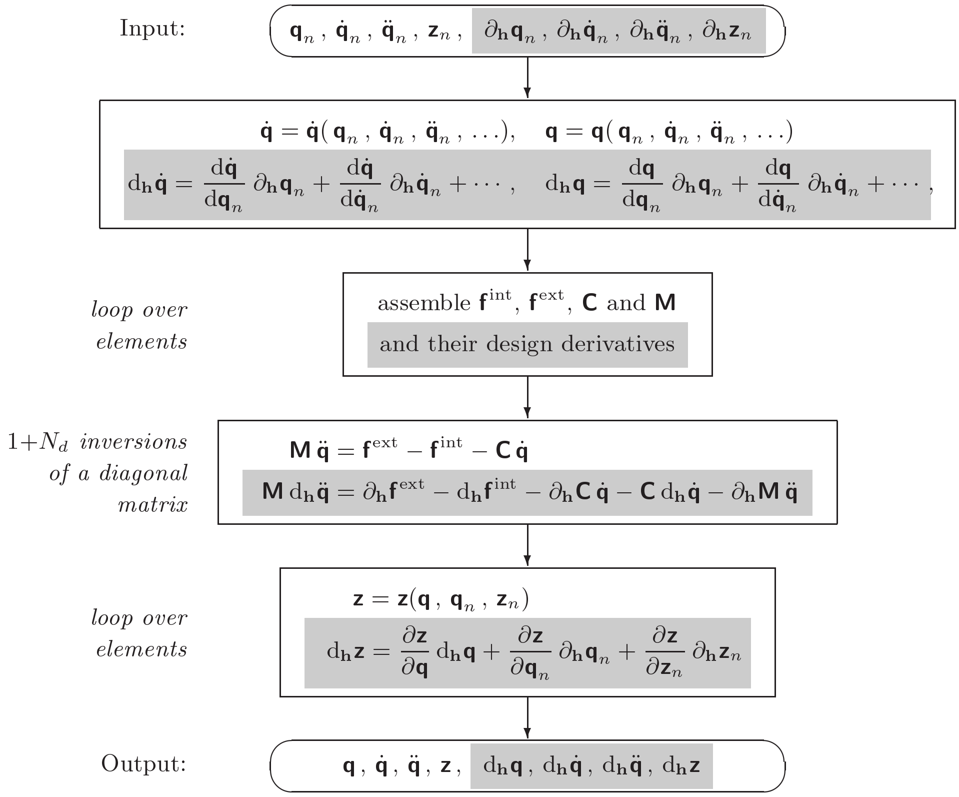
$$\mathrm{d}_{\mathbf{h}}\mathsf{f}^{\text{int}}_{\alpha} = \int_{\Omega^r} \left(\mathrm{d}_{\mathbf{h}}\tilde{\sigma}_{ij}\,\tilde{\mathsf{B}}_{ij\alpha} + \tilde{\sigma}_{ij}\,\mathrm{d}_{\mathbf{h}}\tilde{\mathsf{B}}_{ij\alpha}\right)\mathrm{d}\Omega + \left[\int_{\Omega^r} \tilde{\sigma}_{ij}\,\tilde{\mathsf{B}}_{ij\alpha}\,\phi_{k\beta,k}\,\mathrm{d}\Omega\right]\partial_{\mathbf{h}}\mathsf{X}_{\beta}\,. \qquad (5.27)$$

It must be admitted, however, that the above sensitivity formulation for explicit dynamics does not enjoy the advantage of efficiency compared to that of primary analysis, which was underlined in the case of static analysis and upheld for the implicit dynamics. The reason is that both in the primary and sensitivity analysis we have to do with the linear systems of equations which, at a typical time increment, has to be solved once during the primary analysis and N_d times during the sensitivity analysis. Moreover, since the system matrix is diagonal, we cannot take advantage of the fact that the sensitivity algorithm reuses the once decomposed matrix, saving the time needed for this operation — the time of solution is the same during primary and the sensitivity computations. Thus, the cost coefficient ϕ_c defined by Eq. (4.101) can never achieve so low values as in the case of static analysis. Its value can be now estimated as the time necessary to compute and assemble explicit design derivatives of the finite element arrays (per one design parameter), expressed as a fraction of time needed to compute the primary values of the arrays.

Since the formulae for design derivatives typically include more arithmetic operations than the formulae for the corresponding primary quantities (take e.g. the derivative of a product $c = ab$, $\mathrm{d}_h c = a\,\mathrm{d}_h b + \mathrm{d}_h a\, b$, for which the cost ratio is 3:1), the coefficient ϕ_c is expected to significantly exceed unity in explicit dynamics. This would mean that, in terms of computational efficiency, the analytical sensitivity approach in explicit dynamics may not be competitive against even the finite-difference approach (at least in its simplest, forward- or backward-difference version). Some hints to decrease the cost of DSA can be sought for in efficiency of implementation. Note for instance that, although the primary solution $\ddot{\mathbf{q}}_{n+1}$ is necessary to solve the system (5.25), it is not needed at the stage of computation of $\partial_\mathbf{h}\mathbf{f}^{\mathrm{int}}_{n+1}$ (which consists in a time consuming element-by-element assembly of local contributions). Thus, computation of both $\mathbf{f}^{\mathrm{int}}_{n+1}$ and its design derivatives $\partial_\mathbf{h}\mathbf{f}^{\mathrm{int}}_{n+1}$ can be performed in the same loop over elements which allows to avoid unnecessary repeated computation of numerous intermediate quantities needed for primary internal forces as well as for their sensitivities. Moreover, the local Newton iteration loops appearing in the constitutive algorithms, necessary to solve the nonlinear consistency equation and determine the final stress, are not repeated at the stage of sensitivity analysis when the stress design derivatives are to be determined. All these observations, when appropriately utilized during implementation, allow to decrease the computational cost of sensitivity analysis, however (as it will be shown in numerical examples in the next Section), never to the level typical of the static or implicit dynamic analysis.

Table 5.2 presents the general scheme of computations for both the primary explicit dynamic and the design sensitivity analysis at a typical time step $[t_n\,,\,t_{n+1}]$. Differences between the computation strategies for explicit and

Table 5.2. General scheme of computations for both the primary explicit dynamic and the design sensitivity analysis at a typical time step $[t_n\,,\,t_{n+1}]$ (DSA modifications to the pure primary algorithm are gray-shaded). Indices $n{+}1$ are skipped for better legibility



Input: $\mathbf{q}_n\,,\,\dot{\mathbf{q}}_n\,,\,\ddot{\mathbf{q}}_n\,,\,\mathbf{z}_n\,,\;\partial_{\mathbf{h}}\mathbf{q}_n\,,\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n\,,\,\partial_{\mathbf{h}}\ddot{\mathbf{q}}_n\,,\,\partial_{\mathbf{h}}\mathbf{z}_n$

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}(\,\mathbf{q}_n\,,\,\dot{\mathbf{q}}_n\,,\,\ddot{\mathbf{q}}_n\,,\ldots), \quad \mathbf{q} = \mathbf{q}(\,\mathbf{q}_n\,,\,\dot{\mathbf{q}}_n\,,\,\ddot{\mathbf{q}}_n\,,\ldots)$$

$$\mathrm{d}_{\mathbf{h}}\dot{\mathbf{q}} = \frac{\mathrm{d}\dot{\mathbf{q}}}{\mathrm{d}\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \frac{\mathrm{d}\dot{\mathbf{q}}}{\mathrm{d}\dot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n + \cdots\,, \quad \mathrm{d}_{\mathbf{h}}\mathbf{q} = \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}\dot{\mathbf{q}}_n}\,\partial_{\mathbf{h}}\dot{\mathbf{q}}_n + \cdots\,,$$

loop over elements

assemble $\mathbf{f}^{\mathrm{int}}$, $\mathbf{f}^{\mathrm{ext}}$, $\mathbf{C}$ and $\mathbf{M}$

and their design derivatives

$1{+}N_d$ inversions of a diagonal matrix

$$\mathbf{M}\,\ddot{\mathbf{q}} = \mathbf{f}^{\mathrm{ext}} - \mathbf{f}^{\mathrm{int}} - \mathbf{C}\,\dot{\mathbf{q}}$$

$$\mathbf{M}\,\mathrm{d}_{\mathbf{h}}\ddot{\mathbf{q}} = \partial_{\mathbf{h}}\mathbf{f}^{\mathrm{ext}} - \mathrm{d}_{\mathbf{h}}\mathbf{f}^{\mathrm{int}} - \partial_{\mathbf{h}}\mathbf{C}\,\dot{\mathbf{q}} - \mathbf{C}\,\mathrm{d}_{\mathbf{h}}\dot{\mathbf{q}} - \partial_{\mathbf{h}}\mathbf{M}\,\ddot{\mathbf{q}}$$

loop over elements

$$\mathbf{z} = \mathbf{z}(\mathbf{q}\,,\,\mathbf{q}_n\,,\,\mathbf{z}_n)$$

$$\mathrm{d}_{\mathbf{h}}\mathbf{z} = \frac{\partial\mathbf{z}}{\partial\mathbf{q}}\,\mathrm{d}_{\mathbf{h}}\mathbf{q} + \frac{\partial\mathbf{z}}{\partial\mathbf{q}_n}\,\partial_{\mathbf{h}}\mathbf{q}_n + \frac{\partial\mathbf{z}}{\partial\mathbf{z}_n}\,\partial_{\mathbf{h}}\mathbf{z}_n$$

Output: $\mathbf{q}\,,\,\dot{\mathbf{q}}\,,\,\ddot{\mathbf{q}}\,,\,\mathbf{z}\,,\;\mathrm{d}_{\mathbf{h}}\mathbf{q}\,,\,\mathrm{d}_{\mathbf{h}}\dot{\mathbf{q}}\,,\,\mathrm{d}_{\mathbf{h}}\ddot{\mathbf{q}}\,,\,\mathrm{d}_{\mathbf{h}}\mathbf{z}$

implicit approaches can be easily learned from comparison of Tables 5.1 and 5.2. In explicit dynamics, the sensitivity computations can be performed in parallel to the primary computations. This refers to all levels of analysis, i.e. both the global level at which the systems of equations are solved and the local level at which constitutive computations for material points are performed and global system arrays are assembled. Thus, for instance, the primary constitutive routines displayed in a tabularized form for different material models in Section 3.3 may be executed together with the corresponding DSA routines presented in Section 4.2 (note that the latter extensively reuse intermediate quantities computed in the former). This is not possible in the implicit algorithms where the element-by-element loop has to be called separately at different analysis stages, once for assembly of the primary system arrays and once for the sensitivity arrays.

In view of low relative efficiency of DSA in explicit dynamics, various mixed approaches, combining simplicity of the latter with efficiency of the implicit formulation, are proposed in the literature. An example is [121] where the sensitivity analysis of the explicit dynamic problem is performed only once per a number of time increments, and with the use of an implicit formulation, like that discussed in the previous Section. This can remarkably decrease the numerical cost, but of course makes the sensitivity solution only approximate, as it is not directly related to the primary computation scheme. Besides, application of the method requires assembly of the stiffness matrix K at the stage of sensitivity analysis, which usually is not available in standard codes devoted to explicit dynamic computations.

5.3. Computational examples

The sensitivity formulation of explicit dynamic analysis of elasto-plastic problem will be illustrated with two examples. Both include large-deformations and nonlinear hardening. In the first example, a thick carved beam is dynamically bent. In the second one, dynamic compression of a thin-walled tube is simulated.

5.3.1. Bending of a carved beam

An identical carved beam as that considered in Section 4.3.2 is modelled. The difference is the kinematical excitation: instead of prescribed displacement history $\hat{u}(t)$, the stiff excitator with cylindrical surface is now assigned an initial velocity $v(0) \equiv \dot{u}(0) = \hat{v}$, with $\hat{v} = 10$ m/s. Mechanical properties are given in Table 4.8 (only the rate-independent plasticity, RI, is considered). Additionally, the material density is assumed $\rho = 7800$ kg/m^3 and the stiff excitator mass is $m = 137.8$ kg. Gravity forces are neglected.

Explicit dynamic analysis has been performed on the time interval [0,0.01] s. The sub-critical, constant time step length $\Delta t = 0.4 \cdot 10^{-6}$ s has been chosen. The finite element mesh as in Fig. 4.16 has been applied. The structural performances tracked during the analysis are values of vertical displacement u_A, u_B, u_C of three characteristic points of the model surfaces, see Fig. 5.1, measured at different time instants.

Figure 5.2 presents results of primary analysis. Displacements u_A, u_B, u_C are displayed as functions of time. It can be read from the graphs that the excitator hits the beam, bends it, and finally springs back with about 5 times lower

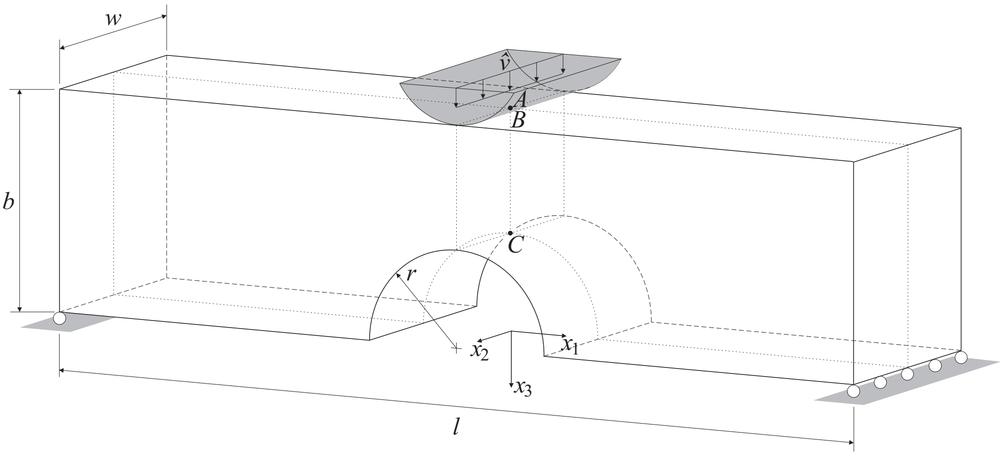


Figure 5.1. Bending of a carved beam: geometry, load and support conditions, and location of the points at which displacements are tracked; (A is the central point of the excitator's surface, B is the central point of the top beam's surface, initially coincident with A, and C is the central point of the carve surface)

velocity, leaving the plastically deformed beam freely vibrating with a small amplitude. Total system's energy remains the same which is a good verification of correctness of the explicit dynamic formulation.

Sensitivity analysis has been performed along with the primary dynamic analysis. Four design parameters were considered: the carve radius r, the excitator's mass m, and the two material constants of the beam: E and σ_{y} . In Figs. 5.3–5.6, sensitivities of the three displacements u_A, u_B, u_C with respect to the design parameters are plotted. Results are compared to the finite difference approximate sensitivities, cf. Eq. (4.100), obtained by running the primary dynamic analysis for each design parameter perturbed by $\Delta h_i = \pm 10^{-4} h_i$, respectively. Similarly as in the statical analysis tests, several other values of perturbations have been tested to make sure that the chosen values lie within the appropriate range (see discussion in Section 2.2.3). For the sake of legibility, the finite difference results are shown after only selected (every 500th) time increments.

The results presented suggest the following two conclusions. First, similarly as in the case of static analysis, the analytical results perfectly coincide with the finite difference approximates which proves correctness of the sensitivity algorithm and its implementation.

The second conclusion refers to the nonsmooth shape of the transient sensitivity solutions, visible especially in Figs. 5.4 and 5.5, where magnified details are additionally displayed. This feature is related to high-frequency vibrations

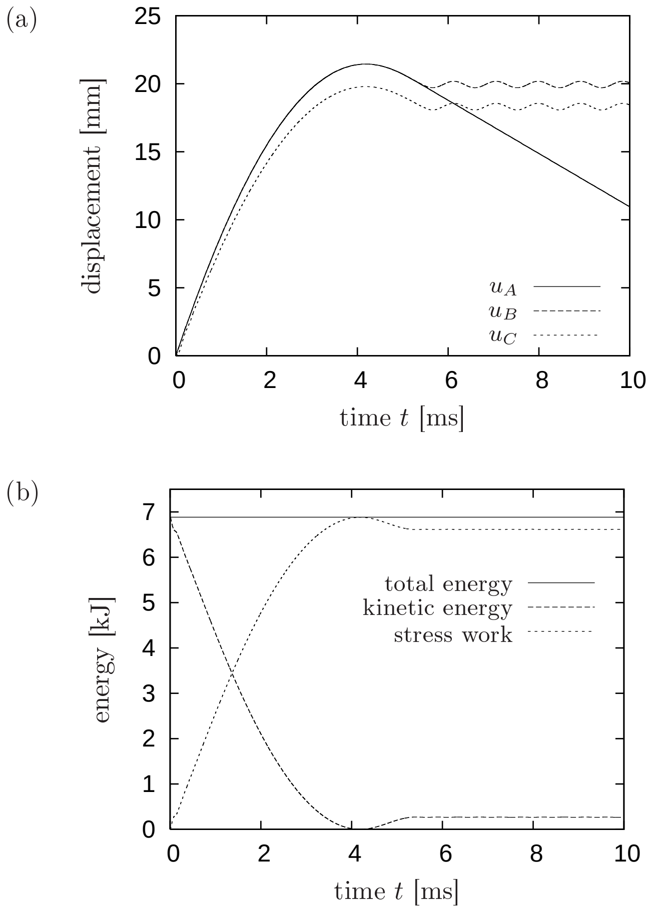


Figure 5.2. Bending of a carved beam: primary dynamic analysis results; (a) displacements of the characteristic points, (b) energy balance

being an inherent feature of the structural dynamics solutions. The vibrations have a very small amplitude which makes them practically invisible in the primary solution graphs (Fig. 5.2). However, the small amplitude appears to be strongly design-dependent which results in the locally rough shape of the sensitivity graphs in Figs. 5.3–5.6.

It was concluded in Section 5.2.2 that numerical cost of the sensitivity analysis per one design variable compared to that of the primary analysis is in case of explicit dynamics much higher than in the case of statics. The cost coefficient $\phi_{\rm c}$, defined by Eq. (4.101), is in this case expected to take values exceeding unity. A number of tests has been done, based on the example discussed in this section,

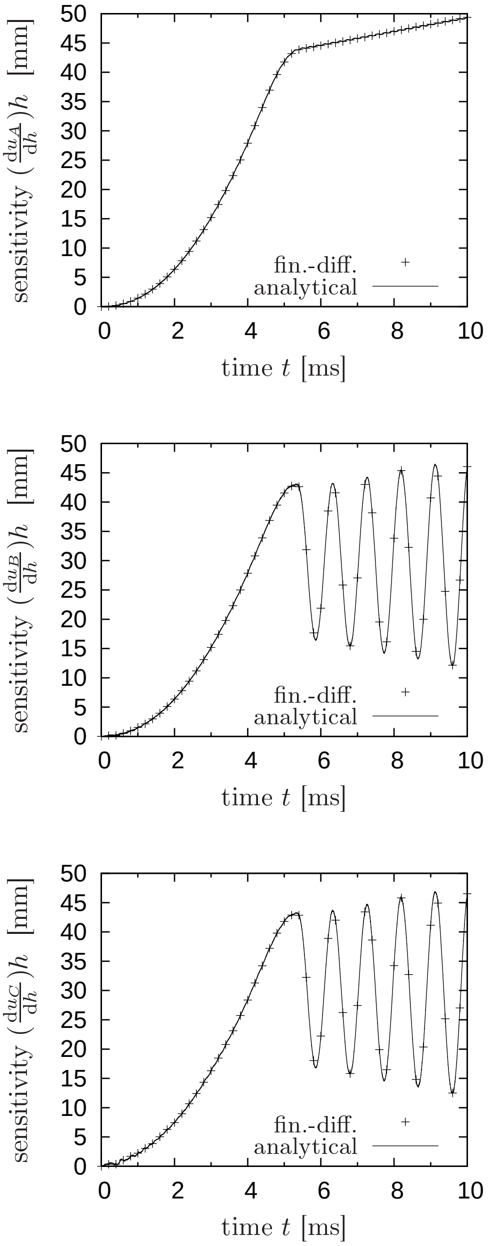


Figure 5.3. Transient normalized sensitivity of the displacements u_A, u_B, u_C with respect to $h = h_1 = r$. Comparison of analytical design derivatives and their finite-difference approximations

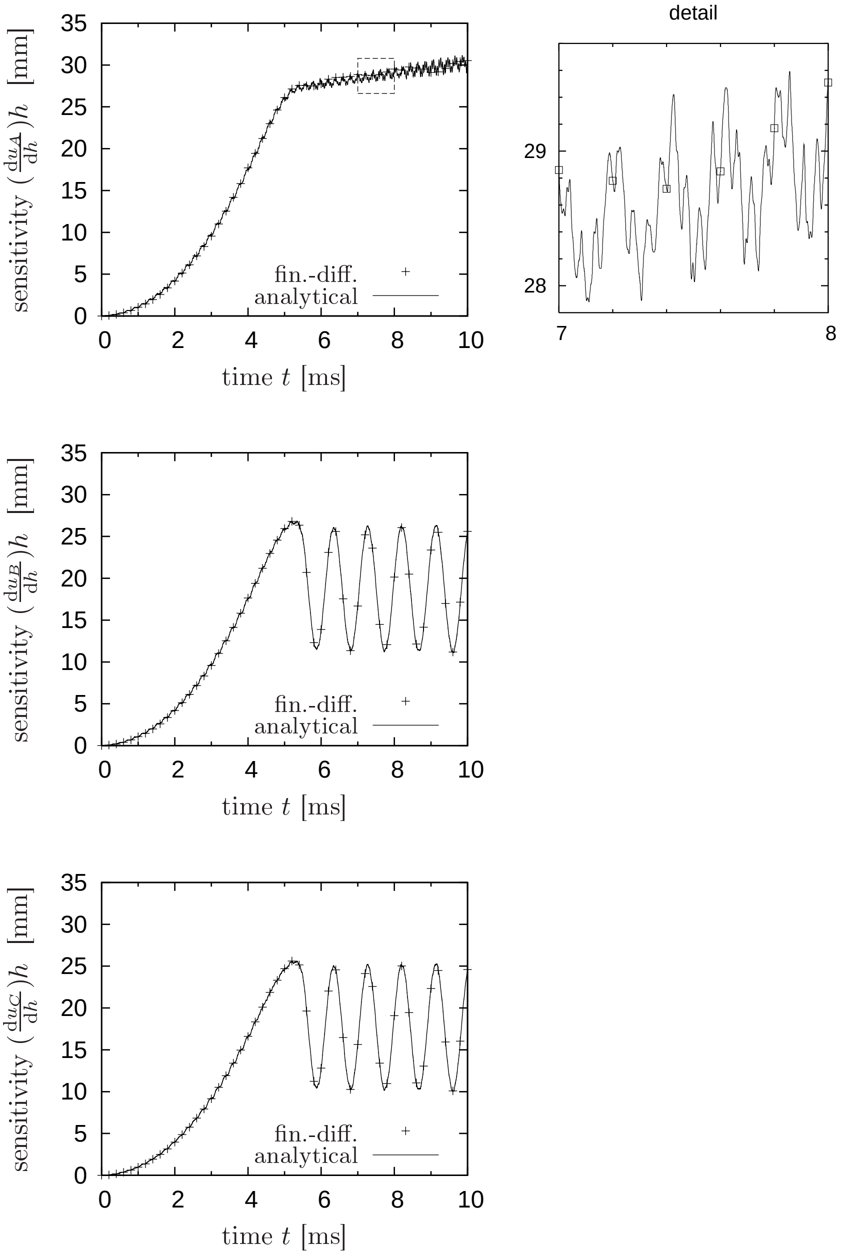


Figure 5.4. Transient normalized sensitivity of the displacements u_A, u_B, u_C with respect to $h = h_2 = m$. Comparison of analytical design derivatives and their finite-difference approximations

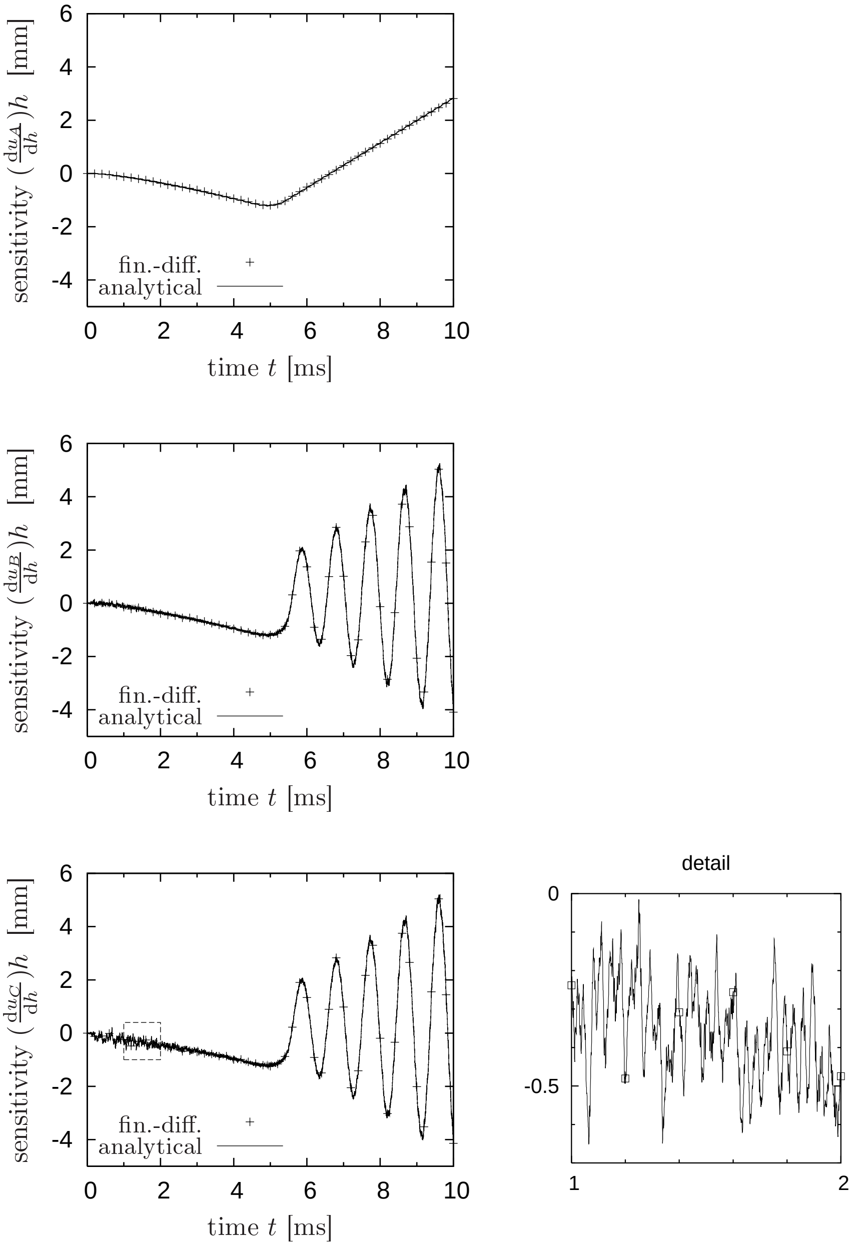


Figure 5.5. Transient normalized sensitivity of the displacements u_A, u_B, u_C with respect to $h = h_3 = E$. Comparison of analytical design derivatives and their finite-difference approximations

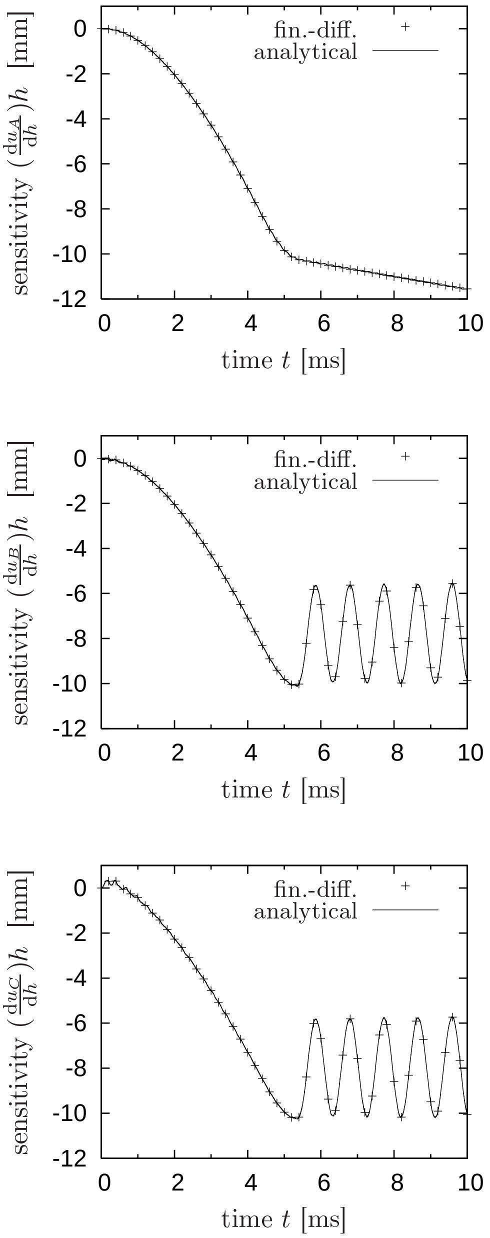


Figure 5.6. Transient normalized sensitivity of the displacements u_A, u_B, u_C with respect to $h = h_4 = \sigma_{\mathrm{y}}$. Comparison of analytical design derivatives and their finite-difference approximations

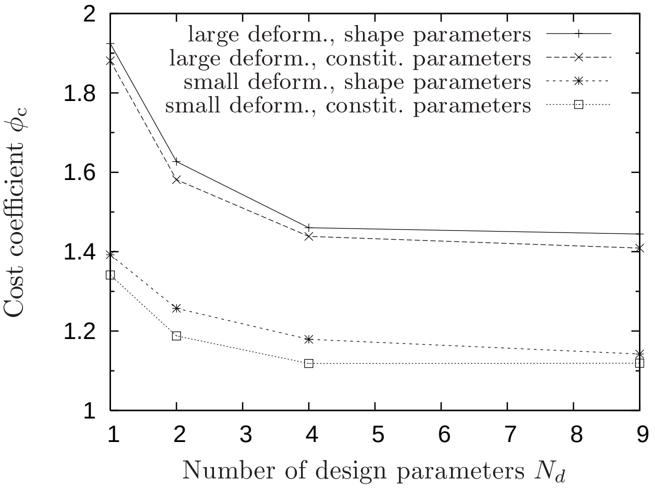


Figure 5.7. Numerical cost of sensitivity computations of explicit dynamic analysis

with different number of design parameters, and with the CPU time measured in each case. Shape and non-shape design parameters were treated separately. Besides, for comparison, the same problem was additionally computed with the use of the small-deformation geometric formulation. Figure 5.7 presents the cost analysis results. As it can be seen, the cost coefficient is higher than unity in all cases, but never exceeds 2. This means that the computational cost of the sensitivity analysis lies somewhere between the cost of forward- or backward-finite-difference analysis ($\phi_c = 1$) and the central-finite-difference analysis ($\phi_c = 2$). Comparing the results with those presented in Fig. 4.24, we can see that the relative computational cost of sensitivity analysis in explicit dynamics is about one order of magnitude higher than in the case of statics. This result can be considered discouraging. However, let us recall the discussion of Section 2.2.3, to stress that, even in this case, the analytical solution has still advantages over the finite-difference solution, as it is, e.g., independent on the perturbation size, and thus more reliable.

The values of the cost coefficient ϕ_c shown in Fig. 5.7 are obviously specific to the very problem analysed. It can be seen that, for instance, for problems with less complicated formulation, the coefficient takes lower values. Besides, since any iteration loops in the primary formulations are never repeated in the sensitivity computations (and replaced with corresponding linear equations), one can expect that formulations containing time-consuming iterations at the level of assembly of the finite element arrays may exhibit lower values of the coefficient ϕ_c. Stillman [120] argues that, in real complex dynamic analyses of e.g. vehicle crashworthiness, featuring nonlinear material properties, contact iterations, and frequent recomputations of critical time increment (only in primary analysis), the value of this coefficient may be dropped to even about 0.5. Nevertheless, it is still significantly higher than in statics or implicit dynamics.

5.3.2. Compression of a thin-walled tube

A subject of this example is an impact energy absorbing element in the form of a thin-walled tube with a notch near one of its ends. Figure 5.8 shows the finite element model of its symmetric quarter. The tube (external) dimensions are: $w = 40$ mm, $b = 60$ mm, $l = 370$ mm, and the wall thickness is 1.47 mm The

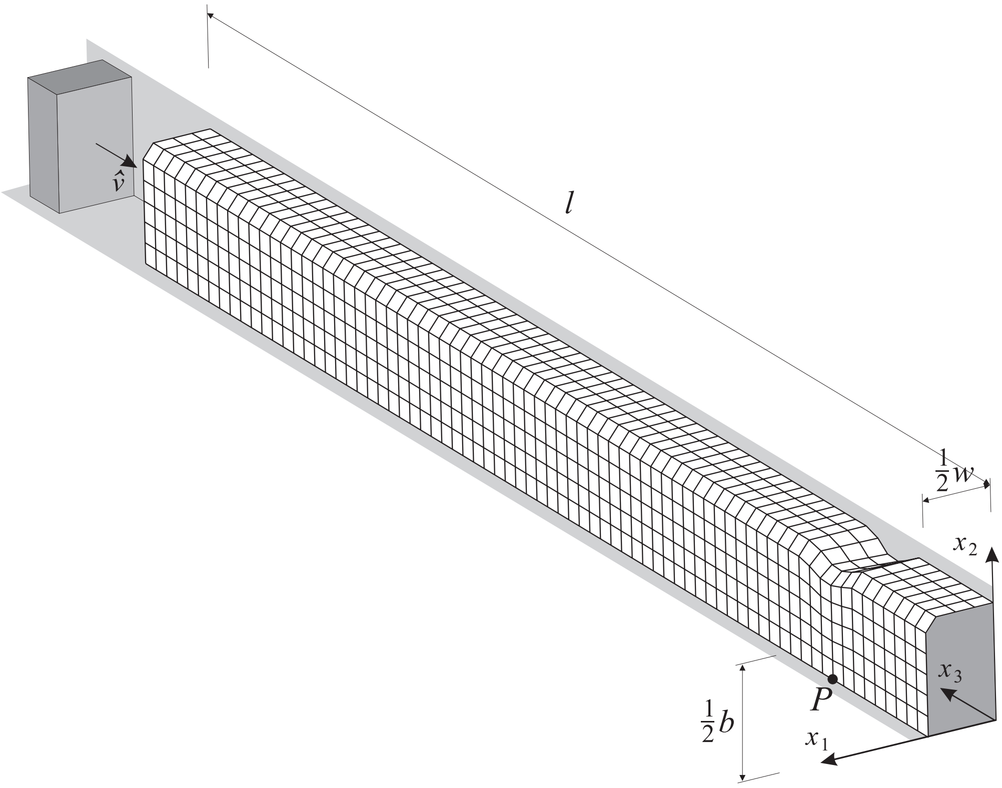


Figure 5.8. Compression of a thin-walled tube: geometry, load and support conditions, and finite element mesh (symmetric quarter)

tube is made of steel with the linear hardening elasto-plastic rate-independent material model, with the following material constants: $E = 2.10 \cdot 10^5$ MPa, $\nu = 0.3$, $\sigma_{\mathrm{y}} = 250$ MPa, $\kappa' = 2000$ MPa and $H' = 0$ MPa.

The tube is fixed at $x_3 = 0$ and is subject to dynamic compression absorbing kinetic energy of a mass $m = 270$ kg hitting the left end of the tube with the initial velocity $v_3(0) = -\hat{v} = -7.7778$ m/s.

Explicit dynamic analysis has been performed with a sub-critical time step, $\Delta t = 0.5 \cdot 10^{-6}$ s. A 4-node six-parameter shell finite element based on the Reissner kinematics and valid for finite rotations has been used. Details of the kinematical formulation (which has not been presented in this study) can be found in [133], along with the appropriate scheme for sensitivity analysis derived by the author for that formulation.

Deformed geometry at selected time instants is shown in Fig. 5.9. Since our computation scheme does not feature self-contact analysis, the computations

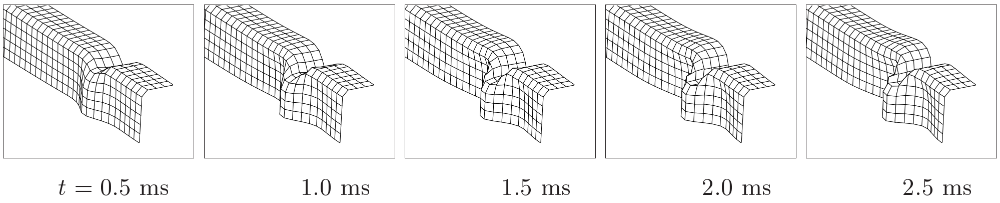

$t = 0.5$ ms 1.0 ms 1.5 ms 2.0 ms 2.5 ms

Figure 5.9. Compression of a thin-walled tube: deformation of notch vicinity

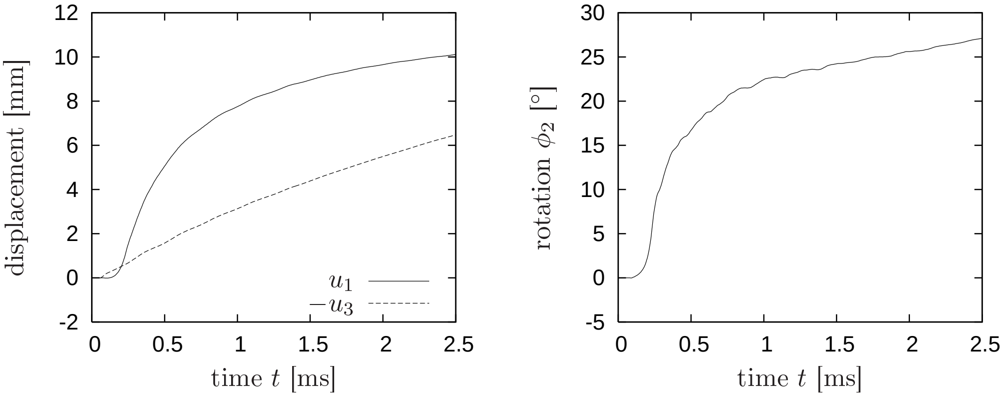


Figure 5.10. Compression of a thin-walled tube: displacement and rotation components of point P

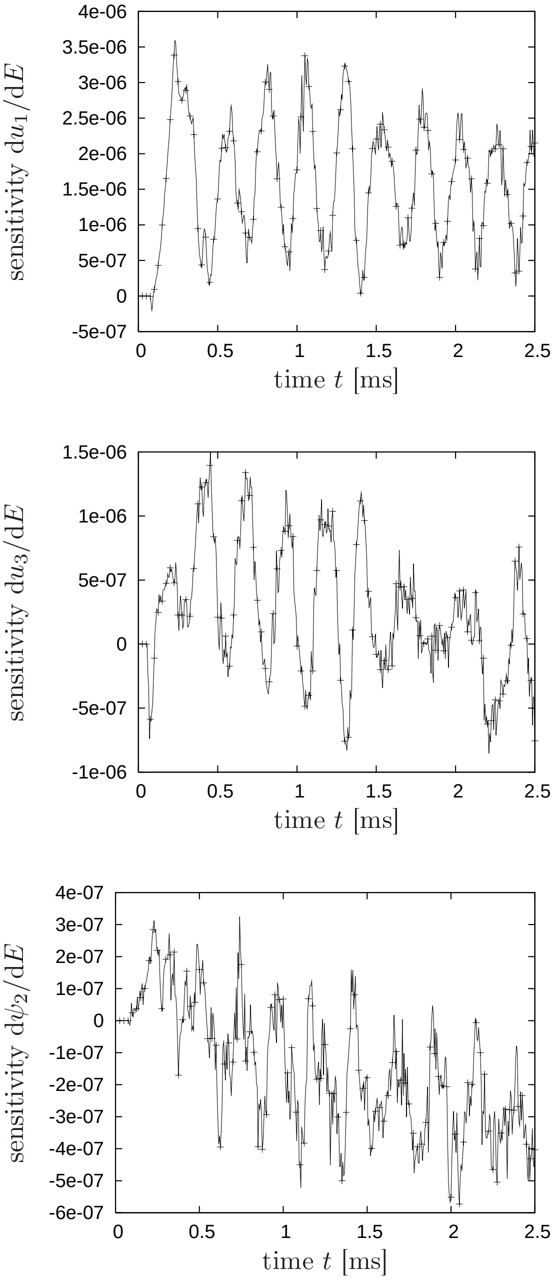


Figure 5.11. Compression of a thin-walled tube: Sensitivity of displacement and rotation components at point P with respect to the Young modulus, $h_1 = E$. Lines: analytical results, symbols: finite-difference results

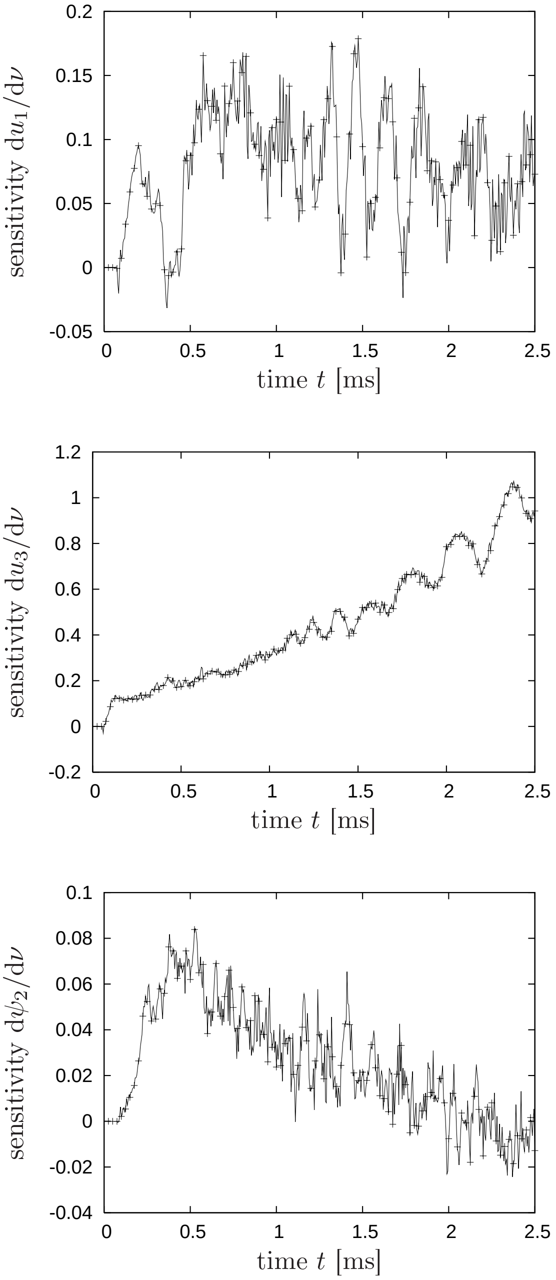


Figure 5.12. Compression of a thin-walled tube: Sensitivity of displacement and rotation components at point P with respect to the Poisson ratio, $h_2 = \nu$. Lines: analytical results, symbols: finite-difference results

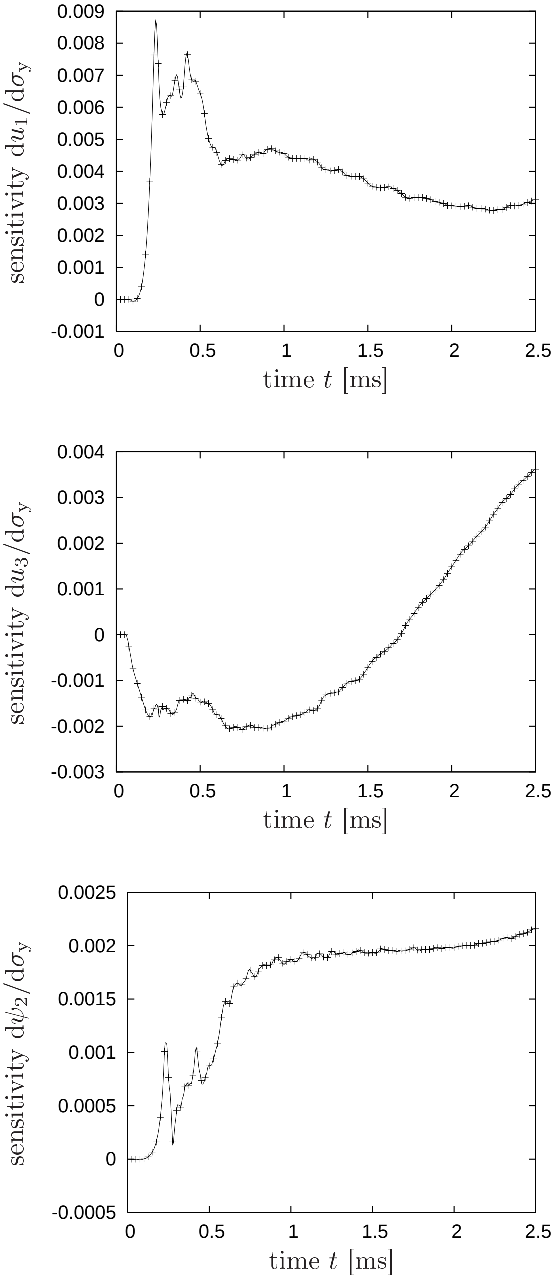


Figure 5.13. Compression of a thin-walled tube: Sensitivity of displacement and rotation components at point P with respect to the yield limit, $h_3 = \sigma_\mathrm{y}$. Lines: analytical results, symbols: finite-difference results

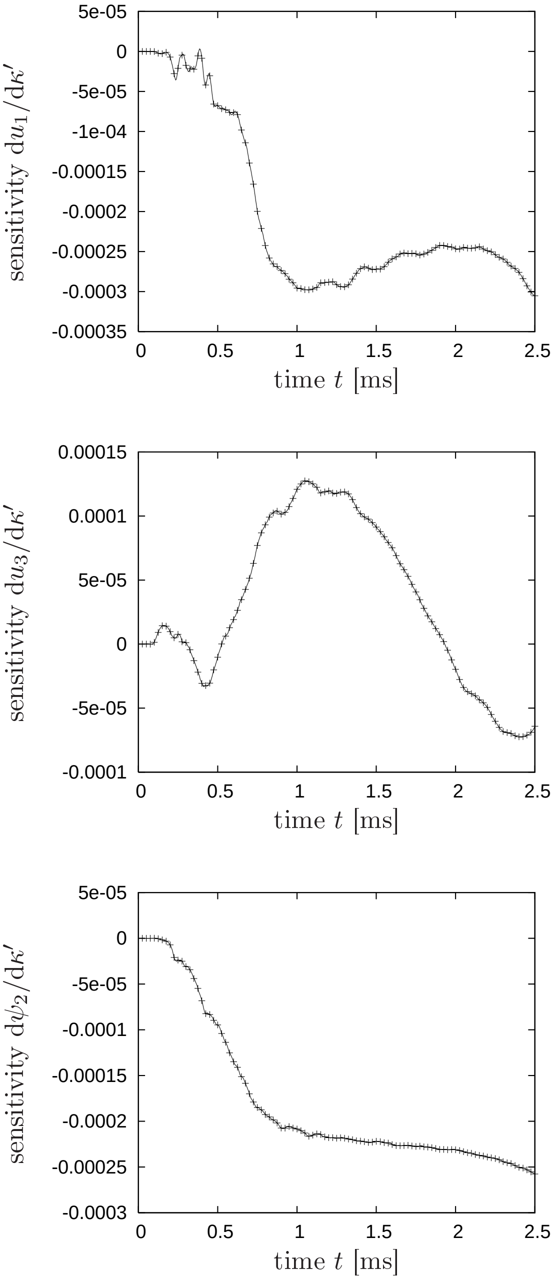


Figure 5.14. Compression of a thin-walled tube: Sensitivity of displacement and rotation components at point P with respect to the isotropic hardening modulus, $h_4 = \kappa'$. Lines: analytical results, symbols: finite-difference results

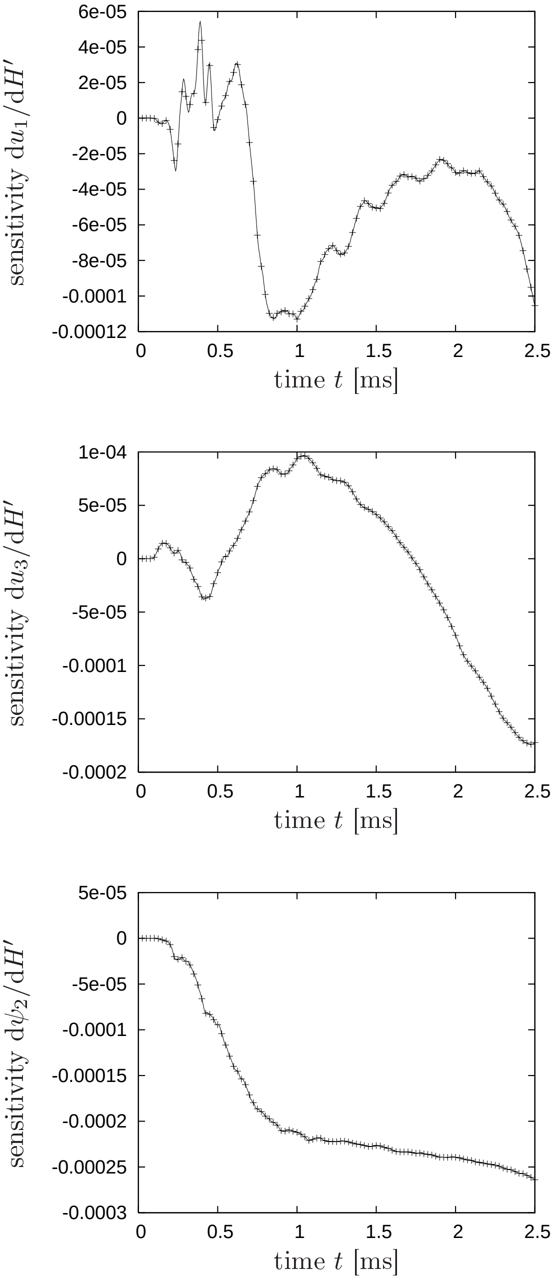


Figure 5.15. Compression of a thin-walled tube: Sensitivity of displacement and rotation components at point P with respect to the kinematic hardening modulus, $h_4 = H'$. Lines: analytical results, symbols: finite-difference results

could only proceed until the first surface-to-surface contact in the developing buckling fold occurred, i.e. up to $t = 2.5 \cdot 10^{-3}$ s.

Components of displacement (u_1 and u_3) and rotation (ψ_2) at the top of the buckling fold (point P in Fig. 5.8) were tracked as system performances. Figure 5.10 shows the transient primary results.

Graphs presenting transient sensitivity of the system performances with respect to selected design parametrs are presented in Figs. 5.11–5.15. Again, analytical results are compared to the finite difference approximates obtained with the use of the central difference scheme for perturbed parameter primary system solutions. There are no visible differences between the analytical and finite-difference results.

The results of the latter example demonstrate flexibility of the methodology presented in this study from the perspective of continuum formulations. It appears that, given a very specific kinematical formulation of a structural element (in this particular case – an advanced finite-rotation shell element with the control of in-plane rotations and transverse shear forces), one is able to utilize the general approach outlined in Chapters 2 and 4 to derive the appropriate analytical sensitivity formulation compatible with the particular scheme of the primary analysis, whether in the static or dynamic case. The quality of results appears to be as good as in the case of continuum formulation, discussed in detail in this study.

Chapter 6

Implementation of sensitivity algorithms in finite element codes

In the previous chapters, computational algorithms for sensitivity analysis in nonlinear finite element analysis, including elasto-plastic material behaviour, have been presented. It has been underlined that the sensitivity formulations are strictly related to the primary formulations to which they concern. Consequently, the algorithms for sensitivity analysis have many common elements with the primary analysis algorithms and it seems natural that their practical implementation in finite element codes is tightly bound with the implementation of the primary algorithms of nonlinear elasto-plastic analysis, whether in quasi-static or dynamic range of applications. Most of intermediate results determined during the primary analysis computations are then reused by the computational schemes of sensitivity analysis. It seems thus unreasonable that the sensitivity analysis could be practically implemented and performed 'outside' the finite element code written for simulation of the primary problem.

The obvious implication of this fact is the conclusion that, in order to implement the sensitivity analysis option in an existing numerical system of nonlinear mechanical analysis, one has to have access to its source code. This may sound discouraging, as in the case of commercial systems of finite element analysis the source is usually not available to regular users. The following discussion refers to circumstances in which the sensitivity analysis is to be implemented by either an author of the primary code or a user who controls the source code of the FE system.

All computational examples presented in Chapters 4 and 5 have been computed with the author's own finite element code written in Fortran 77 programming language. All sensitivity equations have been consequently implemented in the code, enabling it to perform sensitivity analysis along with the primary computations. Most of the implementation was done manually, while some part of it — with assistance of automatic techniques discussed further in this chapter. Let us now focus on several important practical aspects of the implementation of DSA algorithms in a numerical code.

As it has been mentioned in Section 2.2.3, in contrast to its numerical efficiency, implementation cost of analytical sensitivity analysis is high. The main part of the work concerns the routines that compute local contributions to the global array of residual forces $\mathbf{f}^{\mathrm{res}}$ (Eqs. (3.166)–(3.167)) and update local state variables, i.e. constitutive routines that determine end-of-the-increment stress $\tilde{\sigma}_{ij}$ and state fields $\boldsymbol{z}_k$, and geometric routines that determine the array $\tilde{\mathsf{B}}_{ij\alpha}$. The routines have to be supplemented with code that computes design derivatives of all the quantities, according to equations given in Chapter 4.

Table 6.1. Example of a Fortran 77 code in which primary computations are supplemented with a sensitivity analysis equations (the latter are grey-shaded). The code computes dev $\boldsymbol{\sigma}_{\mathrm{trial}}$, $\boldsymbol{s}_{\mathrm{trial}}$ and $\bar{\sigma}_{\mathrm{trial}}$ and their design derivatives. 2nd rank tensors are coded in 6-element arrays according to Eq. (3.146a)

```
c
c     PRIMARY ANALYSIS
c     --- compute deviatoric trial stress and relative stress
      str2 = 0.
      do ks=1,6
        devsigtr(k) = devsig(k) + 2.*G*devdeps(k)
        devstr(k) = devsigtr(k) - devalph(k)
        str2 = str2 + devstr(k)**2
      end do
c     --- compute equivalent Huber-Mises stress
      eqstr = sqrt(1.5*str2)
c
c     SENSITIVITY ANALYSIS
c     --- loop over design parameters
      if (dsa) then
       do kdes = 1,ndes
c       --- compute design derivatives of
c           deviatoric trial stress and relative stress
        dhstr2 = 0.
        do ks=1,6
          dhdevsigtr(k,kdes) = dhdevsig(k,kdes)
     &           + 2.*G*dhdevdeps(k,kdes) + 2.*dhG(kdes)*devdeps(k)
          dhdevstr(k,kdes) = dhdevsigtr(k,kdes) - dhdevalph(k,kdes)
          dhstr2 = dhstr2 + devstr(k)*dhdevstr(k,kdes)
        end do
c       --- compute design derivatives of equivalent Huber-Mises stress
        dheqstr(kdes) = 1.5*dhstr2/str2
       end do
      end if
c
```

Table 6.1 presents an example of such implementation in a piece of existing Fortran 77 code. The routine corresponds to steps 1–3 in Table 3.1 (computation of $\operatorname{dev}\boldsymbol{\sigma}_{\text{trial}}$, $\boldsymbol{s}_{\text{trial}}$ and $\bar{\sigma}_{\text{trial}}$). It is supplemented by operations corresponding to steps 1–2 in Table 4.2 where design derivatives of the primary quantities are determined.

Let us focus on two important features of the implementation that can be learned from the example.

First, the implementation is **code-intrusive**. Most arithmetic operations contained in the primary computation routines have to be supplemented with additional operations necessary to determine design derivatives of the quantities with respect to all design parameters considered in the analysis. This means that significant additional amount of code has to be added in virtually every single routine of the system. Few exceptions include routines responsible for local Newton iteration loops, like e.g. Eq. (3.114), which, as we have stressed, are not repeated when computing the design gradients, and e.g. routines that evaluate the flow condition or the critical time increment in dynamic analysis.

This feature, directly implying the high implementation effort and costs, is frequently risen as an argument against implementation of sensitivity analysis in its pure analytical form. An alternative is the semi-analytical approach, with its relative simplicity, but also with known drawbacks discussed in Section 2.2.3. The final decision has to be made in every case after considering actual weight of all advantages and disadvantages.

The disadvantage mentioned above may be much less painful in the case of codes written in higher-level object-oriented coding formalism than Fortran 77 utilized in the example. This programming approach allows for instance to define a new, more complex class of stress-type variables, consisting of not just the 6-element array containing the primary stress components, but also of an array of all design derivatives of the components. It is further possible to redefine the operations of summation, multiplication and square-rooting, in the way that accounts for the complex structure of the class objects and includes the well known rules of differentiation of such operation results. Having these definitions, the grey-shaded part of code in Table 6.1 is not necessary at all — the primary code lines, written in a more compact form as

```
devsigtr = devsig + 2.*G*devdeps
devstr = devsigt - devalph
eqstr = sqrt(1.5*devstr*devstr)
```

will yield desired results for both primary and sensitivity components. The implementation of DSA consists, instead, in appropriate modifications of subroutine and function headings, redefining all the concerned variable classes as well as mathematic operations on their representatives in the desired manner.

The second feature of the implementation is mostly **automated form** of the sensitivity formulae. In fact, all the design-differentiated equations in the formulation of the primary analysis include a limited number of different basic arithmetic operations and functions, whose differentiation is performed according to certain standard formulae. There is a number of publications in which the automatic code differentiation methods are advocated as efficient tools for the practical implementation of sensitivity analysis in existing codes. We can mention here professional systems like DAFOR [99], ADIFOR [12], ADOL-C [33] (see http://www.autodiff.org/Tools for more references), designed for fully automated differentiation of source codes written in Fortran or C++. The systems act as pre-compilers — given a source code that performs a sequence of certain operations, they prepare a new code that additionally performs operations evaluating derivatives of output with respect to input. They do it either by blindly differentiating the code line-by-line, as e.g. shown in Table 6.1, or, in the case of object-oriented languages like C++ or Fortran 90/95, by redefining variable classes and arithmetic operations on the variables (operator overloading). The systems were successfully utilized to differentiate linear analysis codes consisting of even several hundreds of thousand source lines [11].

Applicability of such techniques in nonlinear codes becomes, however, problematic. As it has been frequently underlined in this thesis, several nonlinear equations appearing in the primary analysis, both on the global and the local level, and solved with the Newton iteration routine, require special handling at the stage of sensitivity analysis. The iterations are not repeated, but replaced with a linear equation resulting from design differentiation of the primary nonlinear one. Application of automatic differentiation does not cover this special case. Instead, the so generated code will contain iteration loops differentiated line-by-line and supposed to be repeated at the stage of sensitivity analysis for each design parameter. Such iterative loops should eventually converge to the correct sensitivity solutions, but two important drawbacks depreciate the result. First, one cannot benefit from the advantage of efficiency of sensitivity analysis compared to the primary computation cost (which is due, among others, to the linear formulation that allows to avoid iterations in DSA). Second, the primary iteration is assumed converged when the primary solution fulfills a certain error tolerance condition. The same condition will then be checked during sensitivity computations, which poses a danger of inaccurate results because it cannot be guaranteed that the sensitivity solution needs the same, and not higher, number of iterations to converge within the same tolerance.

The efficiency of computations in the case of automatically generated code may be affected in another way, too. Let us consider for example a standard formula for the 3×3 matrix determinant, coded in Fortran 77 as

```
      det = x(1,1)*(x(2,2)*x(3,3) - x(2,3)*x(3,2))
     &    + x(1,2)*(x(2,3)*x(3,1) - x(2,1)*x(3,3))
     &    + x(1,3)*(x(2,1)*x(3,2) - x(2,2)*x(3,1))
```

Blind automatic differentiation leads to the following code instruction

```
      dhdet = dhx(1,1)*(x(2,2)*x(3,3) - x(2,3)*x(3,2))
     &      + x(1,1)*(dhx(2,2)*x(3,3) - dhx(2,3)*x(3,2)
     &              + x(2,2)*dhx(3,3) - x(2,3)*dhx(3,2))
     &      + dhx(1,2)*(x(2,3)*x(3,1) - x(2,1)*x(3,3))
     &      + x(1,2)*(dhx(2,3)*x(3,1) - dhx(2,1)*x(3,3)
     &              + x(2,3)*dhx(3,1) - x(2,1)*dhx(3,3))
     &      + dhx(1,3)*(x(2,1)*x(3,2) - x(2,2)*x(3,1))
     &      + x(1,3)*(dhx(2,1)*x(3,2) - dhx(2,2)*x(3,1)
     &      +       + x(2,1)*dhx(3,2) - x(2,2)*dhx(3,1))
```

which includes 36 floating point operations. Having saved the expressions in parentheses in the primary part of the code we can reduce this number to 27 operations. However, having also computed elsewhere the inverse matrix, frequently also needed in the primary analysis, we can make use of the standard formula $\mathrm{d}_h(\det \boldsymbol{X}) = (\det \boldsymbol{X})\,\mathrm{tr}(\mathrm{d}_h\boldsymbol{X}\,\boldsymbol{X}^{-1})$ coded as

```
      dhdet = dhx(1,1)*xi(1,1) + dhx(1,2)*xi(2,1) + dhx(1,3)*xi(3,1)
     &      + dhx(2,1)*xi(1,2) + dhx(2,2)*xi(2,2) + dhx(2,3)*xi(3,2)
     &      + dhx(3,1)*xi(1,3) + dhx(3,2)*xi(2,3) + dhx(3,3)*xi(3,3)
      dhdet = dhdet*det
```

which includes only 18 floating point operations. It is likely that an automatic differentiation system, even equipped with some optimization tools, will not allow this kind of time-saving simplifications, as well as many other, specific to particular types of arithmetic routines.

All the above mentioned drawbacks do not preclude application of the automatic differentiation techniques in implementation of the nonlinear sensitivity analysis, but they make them rather difficult and inconvenient in this area. Making practical use of the methods would at least require extensive manual interventions of the user in order to preserve desired efficiency and accuracy of the analytical approach. In particular, this includes manual separation of parts of the code that are not supposed to be automatically differentiated (e.g. iteration loops) from the other that can be treated with the numerical tools.

A promising approach to generate sensitivity codes, rapidly developing in the recent years, is automatic differentiation of mathematical formulations performed at the symbolic level. In other words, not the source code, but mathematical equations are differentiated by a specialized computer program and then coded in the form of e.g. Fortran or C subroutines. Applications of the approach are in fact much wider — it allows to generate the primary formulations of highly complicated mathematical models and to get them automatically coded in a computer programming language. Design-differentiation of such a symbolic formulation appears then to be just one of numerous abilities of the mathematical toolkit and the user can treat it as a useful add-on to the automated primary code generator.

Realization of this idea requires highly sophisticated computer programs. An example is AceGen [70], an overlay to the commercial package Mathematica, successfully utilized to implement advanced shell [133], contact [122], and plasticity models in FE codes. In particular, both the primary and sensitivity formulation utilized in the example of dynamic elasto-plastic shell analysis presented in Section 5.3.2 were formulated and implemented in large part with the use of the AceGen utility.

Since the differentiation is performed on the level of symbolic formulae, there is no problem with special treatment iteration loops (due to which the automatic code differentiation methods were criticized) — all the nonlinear equations can be symbolically differentiated with respect to design parameters to yield linear sensitivity equations. This does not mean, though, that the approach is free of possible errors and traps.

A source of errors may be e.g. if-statements frequently introduced to the algorithms in order to prevent e.g. possible division by zero (or by a number 'too close to zero' in terms of the machine precision) or taking the square root from a very small negative numbers. Consider an example of the standard routine to determine eigenvalues of a symmetric 3×3 matrix, Table 6.2. Execution of the steps 4–6 for $b \approx 0$ may result in badly determinate intermediate results and possibly even in unexpected termination of computations (note that b appears in the denominator of the expression for $n = \frac{3c}{mb}$), hence the conditional block is introduced and step 3 is executed instead in such a case, directly yielding the limit values of λ_i at $b \to 0$.

Without displaying explicit symbolic formulae on design derivatives of the eigenvalues λ_i, resulting from direct differentiation of the equations given in Table 6.2, we can notice that the symbolic algorithm for the derivatives must

Table 6.2. Algorithm for eigenvalues λ_i of a symmetric 3×3 matrix $[A_{ij}]$ [113] (ϵ is a small positive number depending on the machine precision)

```
1. $I_1 = A_{11} + A_{22} + A_{33}\,,$
   $I_2 = A_{11}A_{22} + A_{22}A_{33} + A_{11}A_{33} - A_{12}^2 - A_{13}^2 - A_{23}^2\,,$
   $I_3 = \det[A_{ij}]$
2. $b = I_2 - \frac{1}{3}I_1^2\,, \quad c = -\frac{2}{27}I_1^3 + \frac{1}{3}I_1 I_2 - I_3$

if $|b| < \epsilon$ then

    3. $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}I_1$

else

    4. $m = 2\sqrt{-\frac{b}{3}}, \quad n = \frac{3c}{mb}$
    5. $\varphi = \frac{1}{3}\arccos n$
    6. $\lambda_i = \frac{1}{3}I_1 + m\,\cos\left[\varphi + \frac{2}{3}(i-1)\pi\right], \quad i = 1,2,3$

end if
```

include a similar if-block whose branches are exclusively activated depending on the value of the intermediate variable b. It is easy to verify that, setting as input $[A_{ij}] = \left[\begin{smallmatrix} h & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right]$ (i.e. $I_1 = 2 + h$, $I_2 = 2h + 1$, $I_3 = h$, $b = -\frac{1}{3}(h-1)^2$, $c = -\frac{2}{27}(h-1)^3$, $m = \frac{1}{3}(h-1)$, and finally $\lambda_1 = h$, $\lambda_2 = \lambda_3 = 1$), and computing sensitivities $\mathrm{d}\lambda_i/\mathrm{d}h$ following strictly the differentiated formulae of Table 6.2, we obtain

- for $|b| \geq \epsilon$ (second branch of the if-block): the correct result $\{\mathrm{d}\lambda_i/\mathrm{d}h\} = \{1, 0, 0\}$, while
- for $|b| < \epsilon$ (i.e. $h \approx 1$, first branch of the if-block): an incorrect result $\{\mathrm{d}\lambda_i/\mathrm{d}h\} = \{\frac{1}{3}\,, \frac{1}{3}\,, \frac{1}{3}\}$

The error results from the fact that the first branch of the if-block, executed for $b \approx 0$, effectively sets to 0 certain variables that appear in the second branch. Namely, observing that $b \approx 0$ immediately implies $m \approx 0$, we can replace the formula in step 6 taken at $m = 0$ with the simple formula appearing in step 3 in the first branch. However, the undesired side effect of such a robust setting is that any dependence of m on design is nullified, too. The symbolic differentiation algorithm will thus assume $\mathrm{d}m/\mathrm{d}h \equiv 0$ in the first branch of the if-block which is obviously not true even at $m = 0$.

There are two ways of making the symbolic differentiation yield correct sensitivity equations in such circumstances. One is, again, manual intervention of the user whenever necessary, which is very inconvenient and not always allowed by the system. Another one is formulating the primary model equations in a way that does not contain such differentiation traps as the one described above. Note that errors of this type may appear not only in implementation of sensitivity analysis. Automatic symbolic derivation of the tangent stiffness matrix will also be prone to errors resulting from such traps in the algorithm formulation. In other words, a differentiation-oriented way of thinking is required in formulation of primary algorithms that are supposed to be automatically differentiated.

Chapter 7

Concluding remarks

The objective of this dissertation was to present the numerical methods of design sensitivity analysis of nonlinear mechanical systems with particular reference to elasto-plastic and elasto-viscoplastic structural response. The notions of design parameters and system performances have been introduced and the main problem of the thesis, i.e. determination of the performance gradients with respect to design, has been formulated. Following discussion on different combinations of continuum and discrete formulations, attention has been mainly focused on design-differentiation of discrete equations of nonlinear continuum mechanics under both statical and dynamical conditions. Direct differentiation method has been chosen as the only appropriate approach in the case of the path-dependent problem of mechanics.

The primary formulation of a nonlinear problem of continuum statics in terms of space- and time-discretized fields has been presented, including time-discrete constitutive equations of elasto-plasticity for both small and large deformations as well as for both rate-independent and rate-dependent plastic flow equations. The formulation, assuming the form of a series of nonlinear systems of algebraic equations associated with subsequent discrete time instants, has been differentiated with respect to design parameters, including both constitutive and shape parameters. Equations for design gradients of the solution have thus been formulated, in the form of linear algebraic equation systems. Several specific issues related to the formulation have been discussed, including the non-differentiability of the elasto-plastic response with respect to input parameters, the importance of consistent tangent operators in both the primary and sensitivity formulations, computational efficiency of sensitivity analysis, and relations between analytical, semi-analytical and finite difference approaches to sensitivity evaluation. Extensions towards dynamic analysis of nonlinear elasto-plastic systems have been then discussed, with appropriate derivations of sensitivity analysis formulations. The derived algorithms have been illustrated with computational examples demonstrating the wide area of applications and high quality of results.

The presented formulations have been successfully implemented in an author's own finite element code. Their closed and complete form allows to easily introduce them into any other finite element system whose source code is available. The conclusions drawn from the presented material can be summarized as follows.

1. The formulation of sensitivity analysis derived by analytical differentiation of discrete primary equations of statics appears to be a very efficient tool to determine design gradients of numerical equilibrium response. Even in highly nonlinear formulations, including elasto-plastic rate-independent and rate-dependent constitutive equations and the most general case of large deformations, the sensitivity problem is linear (at least at a single time step) and, which is particularly advantageous, its solution consists in reusing the once decomposed tangent stiffness coefficient matrix in back-substitution against new right-hand side vectors specific to sensitivity analysis. This makes the solution time of the sensitivity problem (per one design parameter) a small fraction of the primary solution time, and the sensitivity computation algorithm — a numerically cheap add-on to the finite element system of nonlinear analysis of structures. The necessary condition is, however, that the exact consistent tangent matrix is used in the iteration scheme of the primary analysis.

2. The above conclusion regarding efficiency does not refer to sensitivity formulations derived from the continuum differential or variational equations and then discretized independently of the primary problem. Other advantages of the latter approach, like possible gains in accuracy, have to be kept in mind and appreciated, but they do not seem to make the approach competitive to the one presented in detail in the thesis.

3. In the case of path-dependent problems, like those including elasto-plastic material behaviour, the only efficient method of sensitivity analysis is the direct differentiation method (DDM). Numerous advantages of the alternative adjoint system method (ASM), advocated in sensitivity analysis of linear or nonlinear conservative systems, cannot be enjoyed in this case.

4. Design parameters that affect the initial geometry of the analysed system make the sensitivity formulation somewhat more complicated but, upon introduction of the reference volume concept and association of the reference

volume with the parent configuration of the isoparametric finite element, they can be actually treated in the same manner as the other design parameters, affecting e.g. only material properties. This allows to formulate the problem of shape and non-shape sensitivity in a unified manner.

5. Possible non-differentiability of elasto-plastic response with respect to design parameters make the sensitivity solution discontinuous, and thus locally undefined in terms of a standard derivative. It has been demonstrated that this problem does not preclude practical application of the presented formulation and does not affect the quality of the sensitivity results.

6. Accuracy of the sensitivity solution depends on accuracy of the primary solution. It is essential that, in view of the following sensitivity computations, the error tolerance in primary equilibrium iterations is set at a more demanding level than that considered satisfactory from the point of view of required primary solution accuracy.

7. The general formalism of the space- and time-discrete sensitivity analysis can be easily extended towards dynamic analysis, both in the implicit and explicit approaches. In the latter case, however, the relative computational cost of sensitivity analysis compared to primary analysis is much higher than in the static or implicit dynamic analysis. The main reason is high efficiency of the primary formulation itself, namely its linearity and the diagonal form of the coefficient matrix in the equation system to be solved. This does not allow any remarkable time-saving simplifications in the sensitivity analysis formulation.

8. Practical implementation of the presented algorithms in a finite element program requires access to the source code and the implementation cost is high. In some cases, the implementation of semi-analytical or even finite-difference methods may be considered a reasonable alternative, even in view of numerous drawbacks of both the methods. Automatic code generation tools may also be utilized in sensitivity implementation, however, the issue of computational efficiency and reliability of the so generated codes must be taken into consideration.

The application area of the presented algorithms is wide. The main part of it are gradient optimization methods. Efficient gradient-based optimization requires information on both the system performance measures and their sensitivities with respect to design parameters [37]. Structural analysis codes capable of evaluating all of them are thus desirable elements of structural optimization procedures and numerical systems. The term ‘optimization’ includes in particular the rapidly gaining the interest of engineers issue of topology optimization which, despite of its inherently discontinuous character, can be though expressed and effectively solved with the methods of continuous gradient optimization [83, 84]. The reliability-based structural optimization [69] can also be mentioned as one more promising area in this branch. Another branch of applications are system identification methods. They are conceptually very similar to gradient optimization methods and the design sensitivity of the response may play the crucial role in their successful applications. Further, methods of error analysis, parametric ‘what-if’ studies, and stochastic analysis of the system response, can be also considered challenging areas of applications of the sensitivity analysis methods.

The shortly summarized above areas of application prove that the methods of sensitivity analysis still remain in the focus of interest of researchers dealing with computational mechanics. Nowadays, it is actually difficult to imagine that a modern, contemporary computer system of nonlinear analysis in mechanical engineering could not provide reliable tools to assess the response sensitivity with respect to wide variety of input parameters.

In the case of linear systems, the methods seem to be well documented in the literature, cf. the review paper [56]. In nonlinear mechanical problems there is still a lot to be done. Analytical derivations of sensitivity in advanced constitutive models of nonlinear, especially multiphase materials, as well as in highly complex formalisms of shell and beam kinematics and associated finite element formulations are still desired. Sensitivity formulations in structural vibration analysis requiring design differentiation of the generalized eigenproblem of a large system stiffness matrix, as well as the sensitivity analysis of the limit load factor in structural stability problem, have not been mentioned in this thesis and are also a subject of extensive research.

Application of shape sensitivity techniques in finite element analysis requires, as it was mentioned in Section 4.1.2, implementation of sensitivity algorithms in mesh generators, so that gradients of nodal coordinates with respect to geometric parameters of the meshed body are available as their output along with the coordinates themselves. This is another area of needed research in the field of computational mechanics.

Implementation of DSA algorithms in existing finite element codes is a tedious task, especially if a code is written in a low-level programming language like Fortran 77 and was not originally meant to provide, among others, results of sensitivity analysis. This is why the code developers are frequently discouraged with the amount of labour and choose less reliable semi-analytical or even finite-difference formulations. For codes written in object-oriented languages the task of sensitivity implementation is much easier.

The author believes that a progress in practical implementations of analytical algorithms of sensitivity can be made along with popularization of methods of symbolic programming mentioned in Chapter 6. Their advantages consist in straightforward formulation of mathematical models and in moving the burden of symbolic transformation of formulae and coding them in a programming language from the researcher's mind onto the computer. The idea of formulation of primary equations in terms of the symbolic formulae and then having the numerical code automatically generated allows to implement the sensitivity analysis in an easy, natural way, by simply extending the set of symbolic formulae with their automatically differentiated counterparts and then generating a code that includes both primary and sensitivity analysis. Bearing in mind all possible sources of errors in such an approach (cf. examples in Chapter 6), we can though foresee that an increasing number of numerical procedures for new geometric and constitutive formulations in structural mechanics will be generated with the use of such tools. One can thus hope that analytical sensitivity formulations, so natural and straightforward in this approach, will also accompany the so derived algorithms. Besides, along with development of the symbolic programming tools, we can expect increase of efficiency of the so generated codes, which is the issue of critical importance from the point of view of computational applications.

Appendix A

Algorithmic constitutive tangent operators

This appendix contains detailed derivation of algorithmic constitutive tangent operators for elasto-plastic models discussed in Chapter 3, consistent with the time integration schemes applied to corresponding rate-type constitutive equations.

A.1. Small deformation formulation

The algorithmic constitutive tangent operator $\boldsymbol{\mathcal{C}}$ for the incremental constitutive model presented in Section 3.3.1 can be defined as, cf. Eq. (3.94),

$$\boldsymbol{\mathcal{C}} = \frac{\mathrm{d}\boldsymbol{\sigma}_{n+1}}{\mathrm{d}\boldsymbol{\varepsilon}_{n+1}} = \frac{\mathrm{d}\boldsymbol{\sigma}_{n+1}}{\mathrm{d}\Delta\boldsymbol{\varepsilon}} \tag{A.1}$$

(for the sake of legibility the index $n{+}1$ will be skipped in further derivation in this section). Subsequent differentiation of equations displayed in Table 3.1, together with the consistency condition (3.111) at t_{n+1}, with respect to $\Delta\boldsymbol{\varepsilon}$, leads to the following formulae,

$$\begin{aligned}
\mathrm{d}\boldsymbol{s}_{\mathrm{trial}} &= \mathrm{d}(\operatorname{dev}\boldsymbol{\sigma}_{\mathrm{trial}}) = 2G\operatorname{dev}(\mathrm{d}\Delta\boldsymbol{\varepsilon}) = 2G\left(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\right)\mathrm{d}\Delta\boldsymbol{\varepsilon}\,, && \text{(A.2)}\\
\mathrm{d}\bar{\sigma}_{\mathrm{trial}} &= \frac{3}{2\bar{\sigma}_{\mathrm{trial}}}\,\boldsymbol{s}_{\mathrm{trial}} : \mathrm{d}\boldsymbol{s}_{\mathrm{trial}}\,, && \text{(A.3)}\\
\mathrm{d}\bar{\sigma} &= \mathrm{d}\bar{\sigma}_{\mathrm{trial}} + \bar{\sigma}_{,\Delta\bar{e}^{\mathrm{p}}}\,\mathrm{d}\Delta\bar{e}^{\mathrm{p}}\,, && \text{(A.4)}\\
0 &= \frac{\partial\tilde{f}}{\partial\bar{\sigma}}\,\mathrm{d}\bar{\sigma} + \frac{\partial\tilde{f}}{\partial\Delta\bar{e}^{\mathrm{p}}}\,\mathrm{d}\Delta\bar{e}^{\mathrm{p}}\,. && \text{(A.5)}
\end{aligned}$$

The last two of them can be easily replaced by

$$\tilde{f}_{,\bar{\sigma}}\,\mathrm{d}\bar{\sigma}_{\mathrm{trial}} + \tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}\,\mathrm{d}\Delta\bar{e}^{\mathrm{p}} = 0 \tag{A.6}$$

with $\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}} \equiv \frac{\partial\tilde{f}}{\partial\Delta\bar{e}^{\mathrm{p}}} + \frac{\partial\tilde{f}}{\partial\bar{\sigma}}\,\bar{\sigma}_{,\Delta\bar{e}^{\mathrm{p}}}$ given in Table 3.2 and

$$\tilde{f}_{,\bar{\sigma}} \equiv \frac{\partial\tilde{f}}{\partial\bar{\sigma}} = \begin{cases} 1 & \text{for rate-independent plasticity,} \\ \Delta t\, g_{,\bar{\sigma}} & \text{for viscoplasticity.} \end{cases}$$

Equation (A.6) can also be rewritten as

$$\mathrm{d}\Delta\bar{e}^{\mathrm{p}} = \frac{\gamma}{3G}\,\mathrm{d}\bar{\sigma}_{\mathrm{trial}}\,, \qquad \gamma = -\frac{3G\tilde{f}_{,\bar{\sigma}}}{\tilde{f}_{,\Delta\bar{e}^{\mathrm{p}}}}\,. \tag{A.7}$$

Differentiating the elasto-plastic stress deviator, expressed as

$$\operatorname{dev}\boldsymbol{\sigma} = \operatorname{dev}\boldsymbol{\sigma}_{\mathrm{trial}} - \vartheta\,\boldsymbol{s}_{\mathrm{trial}}, \qquad \vartheta = \frac{3G\Delta\bar{e}^{\mathrm{p}}}{\bar{\sigma}_{\mathrm{trial}}}\,,$$

we obtain

$$\mathrm{d}(\operatorname{dev}\boldsymbol{\sigma}) = \mathrm{d}(\operatorname{dev}\boldsymbol{\sigma}_{\mathrm{trial}}) - \frac{3G}{\bar{\sigma}_{\mathrm{trial}}}\left(\Delta\bar{e}^{\mathrm{p}}\,\mathrm{d}\boldsymbol{s}_{\mathrm{trial}} + \mathrm{d}\Delta\bar{e}^{\mathrm{p}}\,\boldsymbol{s}_{\mathrm{trial}} - \frac{\Delta\bar{e}^{\mathrm{p}}\,\mathrm{d}\bar{\sigma}_{\mathrm{trial}}}{\bar{\sigma}_{\mathrm{trial}}}\,\boldsymbol{s}_{\mathrm{trial}}\right)$$

which, upon subsequent substitution of Eqs. (A.7), (A.3) and (A.2), leads to

$$\begin{aligned} \mathrm{d}(\operatorname{dev}\boldsymbol{\sigma}) &= (1-\vartheta)\,\mathrm{d}\boldsymbol{s}_{\mathrm{trial}} - (\gamma-\vartheta)\,\frac{\mathrm{d}\bar{\sigma}_{\mathrm{trial}}}{\bar{\sigma}_{\mathrm{trial}}}\,\boldsymbol{s}_{\mathrm{trial}} \\ &= \left[(1-\vartheta)\,\boldsymbol{\mathfrak{I}} - (\gamma-\vartheta)\,\boldsymbol{n}\otimes\boldsymbol{n}\right] : \mathrm{d}\boldsymbol{s}_{\mathrm{trial}} \\ &= 2G\left[(1-\vartheta)\left(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\right) - (\gamma-\vartheta)\,\boldsymbol{n}\otimes\boldsymbol{n}\right] : \mathrm{d}\Delta\boldsymbol{\varepsilon}\,. \end{aligned}$$

Considering now also the elastic spherical part of stress, $\boldsymbol{\sigma} = \operatorname{dev}\boldsymbol{\sigma} + K(\mathrm{tr}\boldsymbol{\varepsilon})\,\boldsymbol{I}$, we finally come at

$$\mathrm{d}\boldsymbol{\sigma} = \boldsymbol{\mathcal{C}} : \mathrm{d}\Delta\boldsymbol{\varepsilon}$$

with

$$\boldsymbol{\mathcal{C}} = K\,\boldsymbol{I}\otimes\boldsymbol{I} + 2G(1-\vartheta)\left(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\right) - 2G(\gamma-\vartheta)\,\boldsymbol{n}\otimes\boldsymbol{n}\,. \tag{A.8}$$

A.2. Large deformation formulation

To derive the tensor $\mathfrak{c}$, and particularly the plastic corrector part $\mathfrak{c}^{\mathrm{p}}$ defined by Eq. (3.144), it is necessary to compute $\mathrm{d}^{\circ}\boldsymbol{s}^{\mathrm{p}}_{n+1}$ and to express it as a linear function of $\mathrm{d}\boldsymbol{\epsilon}_{n+1}$ (for the sake of legibility the index $n{+}1$ will be skipped in further derivation in this section). Differentiating Eq. (3.143b) we obtain

$$\mathrm{d}^{\circ}\boldsymbol{s}^{\mathrm{p}} = \mathrm{d}\vartheta\,\boldsymbol{s}_{\mathrm{trial}} + \vartheta\,\mathrm{d}^{\circ}\boldsymbol{s}_{\mathrm{trial}}\,, \tag{A.9}$$

where

$$\vartheta = \frac{\varphi_{\text{trial}}\Delta\bar{e}^{\text{p}}}{\bar{\sigma}_{\text{trial}}}\,, \qquad \mathrm{d}\vartheta = \vartheta\left(\frac{\mathrm{d}\varphi_{\text{trial}}}{\varphi_{\text{trial}}} + \frac{\mathrm{d}(\Delta\bar{e}^{\text{p}})}{\Delta\bar{e}^{\text{p}}} - \frac{\mathrm{d}\bar{\sigma}_{\text{trial}}}{\bar{\sigma}_{\text{trial}}}\right).$$

The scalar time derivatives $\mathrm{d}\varphi_{\text{trial}}$, $\mathrm{d}(\Delta\bar{e}^{\text{p}})$ and $\mathrm{d}\bar{\sigma}_{\text{trial}}$ appearing in the above formula are not independent. Time differentiation of the flow condition

$$\tilde{f}(\bar{\sigma}, \bar{e}^{\text{p}}, \Delta\bar{e}^{\text{p}}) = 0, \qquad \bar{\sigma} = \bar{\sigma}_{\text{trial}} - \left(1{+}\frac{H'}{3G}\right)\varphi_{\text{trial}}\Delta\bar{e}^{\text{p}}\,,$$

leads to the following relation,

$$\mathrm{d}(\Delta\bar{e}^{\text{p}}) = \frac{\gamma}{\varphi_{\text{trial}}}\left[\mathrm{d}\bar{\sigma}_{\text{trial}} - \left(1{+}\frac{H'}{3G}\right)\Delta\bar{e}^{\text{p}}\mathrm{d}\varphi_{\text{trial}}\right], \qquad \gamma = -\frac{\varphi_{\text{trial}}\,\tilde{f}_{,\bar{\sigma}}}{\tilde{f}_{,\Delta\bar{e}^{\text{p}}}}\,,$$

where $\tilde{f}_{,\Delta\bar{e}^{\text{p}}} \equiv \frac{\partial\tilde{f}}{\partial\Delta\bar{e}^{\text{p}}} + \frac{\partial\tilde{f}}{\partial\bar{\sigma}}\,\bar{\sigma}_{,\Delta\bar{e}^{\text{p}}}$ given in Table 3.5 and, similarly as for small deformations,

$$\tilde{f}_{,\bar{\sigma}} \equiv \frac{\partial\tilde{f}}{\partial\bar{\sigma}} = \begin{cases} 1 & \text{for rate-independent plasticity,} \\ \Delta t\, g_{,\bar{\sigma}} & \text{for viscoplasticity.} \end{cases}$$

Equation (A.9) takes thus the form

$$\mathrm{d}^{\circ}\boldsymbol{s}^{\text{p}} \;=\; \vartheta\left[\left(1-\gamma\left(1{+}\frac{H'}{3G}\right)\right)\frac{\mathrm{d}\varphi_{\text{trial}}}{\varphi_{\text{trial}}}\,\boldsymbol{s}_{\text{trial}} + \frac{\gamma-\vartheta}{\vartheta}\,\frac{\mathrm{d}\bar{\sigma}_{\text{trial}}}{\bar{\sigma}_{\text{trial}}}\,\boldsymbol{s}_{\text{trial}} + \mathrm{d}^{\circ}\boldsymbol{s}_{\text{trial}}\right], \tag{A.10}$$

in which $\mathrm{d}\varphi_{\text{trial}}$, $\mathrm{d}\bar{\sigma}_{\text{trial}}$ and $\mathrm{d}^{\circ}\boldsymbol{s}_{\text{trial}}$ have to be determined.

Let us introduce the notations

$$\boldsymbol{\varsigma} = G\boldsymbol{b}^{\text{e}} - \boldsymbol{\alpha} = \boldsymbol{\tau}^{\star} - \boldsymbol{\alpha}, \qquad \boldsymbol{S} = \boldsymbol{F}^{-1}\boldsymbol{\varsigma}\boldsymbol{F}^{-\text{T}} = G\boldsymbol{C}^{\text{p}-1} - \boldsymbol{A},$$

and, analogously,

$$\bar{\boldsymbol{\varsigma}} = \bar{\boldsymbol{\tau}}^{\star} - \bar{\boldsymbol{\alpha}} = J^{-\frac{2}{3}}\boldsymbol{\varsigma}, \qquad \bar{\boldsymbol{\varsigma}}_{\text{trial}} = \bar{\boldsymbol{\tau}}^{\star}_{\text{trial}} - \bar{\boldsymbol{\alpha}}_{\text{trial}} = J^{-\frac{2}{3}}\boldsymbol{F}\boldsymbol{S}_n\boldsymbol{F}^{\text{T}},$$

so that $\varphi_{\text{trial}} = \operatorname{tr}\bar{\boldsymbol{\varsigma}}_{\text{trial}}$ and $\boldsymbol{s}_{\text{trial}} = \operatorname{dev}\bar{\boldsymbol{\varsigma}}_{\text{trial}}$. We can now write

$$\mathrm{d}\varphi_{\text{trial}} = \operatorname{tr}\left(\mathrm{d}\bar{\boldsymbol{\varsigma}}_{\text{trial}}\right), \qquad \mathrm{d}\bar{\sigma}_{\text{trial}} = \frac{3}{2\bar{\sigma}_{\text{trial}}}\operatorname{dev}\bar{\boldsymbol{\varsigma}}_{\text{trial}} : \operatorname{dev}\left(\mathrm{d}\bar{\boldsymbol{\varsigma}}_{\text{trial}}\right).$$

Introducing the notation $\mathrm{d}\boldsymbol{L} = \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{u}} = \mathrm{d}\boldsymbol{F}\,\boldsymbol{F}^{-1}$ and recalling $\mathrm{d}\boldsymbol{\epsilon} = \frac{1}{2}(\mathrm{d}\boldsymbol{L} + \mathrm{d}\boldsymbol{L}^{\mathrm{T}})$, cf. Eqs. (3.98c), and employing the following relations (in which symmetry of $\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}$ has been utilized),

$$\begin{aligned}
\mathrm{d}\boldsymbol{\varsigma}_{\mathrm{trial}} &= \mathrm{d}\boldsymbol{F}\boldsymbol{S}_n\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{F}\boldsymbol{S}_n\mathrm{d}\boldsymbol{F}^{\mathrm{T}} = \mathrm{d}\boldsymbol{L}\,\boldsymbol{\varsigma}_{\mathrm{trial}} + \boldsymbol{\varsigma}_{\mathrm{trial}}\,\mathrm{d}\boldsymbol{L}^{\mathrm{T}}\,, \\
\mathrm{d}\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}} &= J^{-\frac{2}{3}}\mathrm{d}\boldsymbol{\varsigma}_{\mathrm{trial}} - \tfrac{2}{3}\,\mathrm{tr}(\mathrm{d}\boldsymbol{\epsilon})\,\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}} = \mathrm{dev}(\mathrm{d}\boldsymbol{L})\,\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}} + \bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}\,\mathrm{dev}(\mathrm{d}\boldsymbol{L}^{\mathrm{T}})\,, \\
\mathrm{d}\varphi_{\mathrm{trial}} &= \mathrm{tr}\,\mathrm{d}\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}} = 2\,\mathrm{tr}[\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}\,\mathrm{dev}(\mathrm{d}\boldsymbol{\epsilon})] = 2\,\mathrm{tr}[\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}})\,\mathrm{d}\boldsymbol{\epsilon}] \\
&= 2\,[\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) : \mathrm{d}\boldsymbol{\epsilon}] = 2\,(\boldsymbol{s}_{\mathrm{trial}} : \mathrm{d}\boldsymbol{\epsilon})\,, \\
\mathrm{d}\boldsymbol{s}_{\mathrm{trial}} &= \mathrm{dev}(\mathrm{d}\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) = \mathrm{d}\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}} - \tfrac{1}{3}\boldsymbol{I}\,(\mathrm{tr}\,\mathrm{d}\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) \\
&= \mathrm{dev}(\mathrm{d}\boldsymbol{L})\,\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}} + \bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}\,\mathrm{dev}(\mathrm{d}\boldsymbol{L}^{\mathrm{T}}) - \tfrac{2}{3}\boldsymbol{I}\,[\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) : \mathrm{d}\boldsymbol{\epsilon}]\,, \\
\mathrm{d}\bar{\sigma}_{\mathrm{trial}} &= \tfrac{3}{2\bar{\sigma}_{\mathrm{trial}}}\,(\boldsymbol{s}_{\mathrm{trial}} : \mathrm{d}\boldsymbol{s}_{\mathrm{trial}}) \\
&= \tfrac{3}{2\bar{\sigma}_{\mathrm{trial}}}\,\big\{\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) : [\mathrm{dev}(\mathrm{d}\boldsymbol{L})\,\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}} + \bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}\,\mathrm{dev}(\mathrm{d}\boldsymbol{L}^{\mathrm{T}})]\big\} \\
&= \tfrac{3}{2\bar{\sigma}_{\mathrm{trial}}}\,\big\{2\big[\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}})\,\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) - \tfrac{1}{3}\,\boldsymbol{I}\,\mathrm{tr}[\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}})\,\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}})]\big] : \mathrm{d}\boldsymbol{\epsilon} \\
&\qquad + \tfrac{2}{3}(\mathrm{tr}\,\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}})\,\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) : \mathrm{d}\boldsymbol{\epsilon}\big\} \\
&= \tfrac{1}{\bar{\sigma}_{\mathrm{trial}}}\,\big[3\,\mathrm{dev}(\boldsymbol{s}^2_{\mathrm{trial}}) + \varphi_{\mathrm{trial}}\,\boldsymbol{s}_{\mathrm{trial}}\big] : \mathrm{d}\boldsymbol{\epsilon}\,, \\
\mathrm{d}^{\circ}\boldsymbol{s}_{\mathrm{trial}} &= \mathrm{d}\boldsymbol{s}_{\mathrm{trial}} - \mathrm{d}\boldsymbol{L}\,\boldsymbol{s}_{\mathrm{trial}} - \boldsymbol{s}_{\mathrm{trial}}\,\mathrm{d}\boldsymbol{L}^{\mathrm{T}} \\
&= \mathrm{dev}(\mathrm{d}\boldsymbol{L})\,\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}} - \mathrm{d}\boldsymbol{L}\,\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) + \bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}\,\mathrm{dev}(\mathrm{d}\boldsymbol{L}^{\mathrm{T}}) - \mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}})\,\mathrm{d}\boldsymbol{L}^{\mathrm{T}} \\
&\qquad - \tfrac{2}{3}\boldsymbol{I}\,[\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) : \mathrm{d}\boldsymbol{\epsilon}] \\
&= \tfrac{2}{3}\,[(\mathrm{tr}\,\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}})\,\mathrm{dev}(\mathrm{d}\boldsymbol{\epsilon}) - \mathrm{tr}(\mathrm{d}\boldsymbol{\epsilon})\,\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) - \boldsymbol{I}\,(\mathrm{dev}(\bar{\boldsymbol{\varsigma}}_{\mathrm{trial}}) : \mathrm{d}\boldsymbol{\epsilon})] \\
&= \tfrac{2}{3}\,[\varphi_{\mathrm{trial}}\,\mathrm{dev}(\mathrm{d}\boldsymbol{\epsilon}) - \mathrm{tr}(\mathrm{d}\boldsymbol{\epsilon})\,\boldsymbol{s}_{\mathrm{trial}} - \boldsymbol{I}\,(\boldsymbol{s}_{\mathrm{trial}} : \mathrm{d}\boldsymbol{\epsilon})] \\
&= \tfrac{2}{3}\,\big[\varphi_{\mathrm{trial}}\,\big(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\big) - \boldsymbol{s}_{\mathrm{trial}}\otimes\boldsymbol{I} - \boldsymbol{I}\otimes\boldsymbol{s}_{\mathrm{trial}}\big] : \mathrm{d}\boldsymbol{\epsilon}\,, \\
\mathrm{d}\varphi_{\mathrm{trial}}\,\boldsymbol{s}_{\mathrm{trial}} &= 2\,(\boldsymbol{s}_{\mathrm{trial}} : \mathrm{d}\boldsymbol{\epsilon})\,\boldsymbol{s}_{\mathrm{trial}} = 2\,[\boldsymbol{s}_{\mathrm{trial}}\otimes\boldsymbol{s}_{\mathrm{trial}}] : \mathrm{d}\boldsymbol{\epsilon}\,, \\
\mathrm{d}\bar{\sigma}_{\mathrm{trial}}\,\boldsymbol{s}_{\mathrm{trial}} &= \tfrac{1}{\bar{\sigma}_{\mathrm{trial}}}\,\big\{\big[3\,\mathrm{dev}(\boldsymbol{s}^2_{\mathrm{trial}}) + \varphi_{\mathrm{trial}}\,\boldsymbol{s}_{\mathrm{trial}}\big] : \mathrm{d}\boldsymbol{\epsilon}\big\}\,\boldsymbol{s}_{\mathrm{trial}} \\
&= \tfrac{1}{\bar{\sigma}_{\mathrm{trial}}}\,\big[3\boldsymbol{s}_{\mathrm{trial}}\otimes\mathrm{dev}(\boldsymbol{s}^2_{\mathrm{trial}}) + \varphi_{\mathrm{trial}}\,\boldsymbol{s}_{\mathrm{trial}}\otimes\boldsymbol{s}_{\mathrm{trial}}\big] : \mathrm{d}\boldsymbol{\epsilon}\,,
\end{aligned}$$

we can rewrite Eq. (A.10) in the form $(3.144)_3$,

$$\mathrm{d}^{\circ}\boldsymbol{s}^{\mathrm{p}} = \boldsymbol{\mathfrak{c}}^{\mathrm{p}} : \mathrm{d}\boldsymbol{\epsilon},$$

where

$$\boldsymbol{\mathfrak{c}}^{\mathrm{p}} = C_1\,\big(\boldsymbol{\mathfrak{I}} - \tfrac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\big) + C_2\,(\boldsymbol{n}\otimes\boldsymbol{I} + \boldsymbol{I}\otimes\boldsymbol{n}) + C_3\,\boldsymbol{n}\otimes\boldsymbol{n} + C_4\,\boldsymbol{n}\otimes\mathrm{dev}(\boldsymbol{n}^2),$$

and

$$
\begin{aligned}
\boldsymbol{n} \;&=\; \frac{\boldsymbol{s}_{\text{trial}}}{\|\boldsymbol{s}_{\text{trial}}\|} = \sqrt{\frac{3}{2}}\,\frac{\boldsymbol{s}_{\text{trial}}}{\bar{\sigma}_{\text{trial}}}\,,\\
C_1 \;&=\; \tfrac{2}{3}\,\varphi_{\text{trial}}\,\vartheta\\
C_2 \;&=\; -\tfrac{2}{3}\sqrt{\tfrac{2}{3}}\,\varphi_{\text{trial}}\,\Delta\bar{e}^{\text{p}},\\
C_3 \;&=\; \tfrac{4}{3}\bar{\sigma}_{\text{trial}}\Delta\bar{e}^{\text{p}}\left[1-\gamma\left(1+\tfrac{H'}{3G}\right)\right]+\tfrac{2}{3}(\gamma-\vartheta)\,\varphi_{\text{trial}}\,,\\
C_4 \;&=\; 2(\gamma-\vartheta)\,\bar{\sigma}_{\text{trial}}\,.
\end{aligned}
$$

In the case of linear hardening rate-independent plasticity ($\kappa' = \text{const}$ and $H' = \text{const}$), we have

$$
\gamma = \frac{1}{1+\frac{H'}{3G}+\frac{\kappa'}{\varphi_{\text{trial}}}}\,,
$$

and the operator $\mathfrak{c}$ assumes the form derived in [111].

A.3. Plane stress formulation

For small-deformation plane stress elasto-plasticity, the algorithmic tangent operator consistent with the time integration scheme, is defined with the formula

$$
\mathbf{C} = \frac{\mathrm{d}\boldsymbol{\sigma}_{n+1}}{\mathrm{d}\boldsymbol{\varepsilon}_{n+1}} = \frac{\mathrm{d}\boldsymbol{\sigma}_{n+1}}{\mathrm{d}\Delta\boldsymbol{\varepsilon}} \tag{A.11}
$$

in which the particular quantities, unlike in Eq. (3.94), are understood as reduced-dimension arrays of size 3 in the vector-matrix notation introduced in Section 3.3.3. Differentiation of incremental equations presented in Section 3.3.3.2 leads to the following formulae (in which the index $n+1$ has been skipped for the sake of legibility),

$$
\begin{aligned}
\mathrm{d}\boldsymbol{\varsigma}_{\text{trial}} \;&=\; \mathbf{C}^{\text{e}}\,\mathrm{d}\Delta\boldsymbol{\varepsilon}, && \text{(A.12)}\\
\mathrm{d}\boldsymbol{\varsigma}_{\text{trial}} \;&=\; \mathrm{d}\mathbf{Z}\,\boldsymbol{\varsigma}+\mathbf{Z}\,\mathrm{d}\boldsymbol{\varsigma} = \frac{\mathrm{d}\lambda}{\lambda}\,(\mathbf{Z}-\mathbf{I})\boldsymbol{\varsigma}+\mathbf{Z}\,\mathrm{d}\boldsymbol{\varsigma}, && \text{(A.13)}\\
\mathrm{d}\bar{\sigma} \;&=\; \frac{3}{2\bar{\sigma}}\,\boldsymbol{\varsigma}^{\text{T}}\mathbf{P}\,\mathrm{d}\boldsymbol{\varsigma}, && \text{(A.14)}\\
0 \;&=\; \vartheta_1\,\mathrm{d}\bar{\sigma}+\vartheta_2\,\mathrm{d}\lambda, && \text{(A.15)}\\
\mathrm{d}\boldsymbol{\sigma} \;&=\; \mathrm{d}\boldsymbol{\varsigma}+\mathrm{d}\boldsymbol{\alpha} = \vartheta\,\mathrm{d}\boldsymbol{\varsigma}+(\vartheta-1)\frac{\mathrm{d}\lambda}{\lambda}\,\boldsymbol{\varsigma}, && \text{(A.16)}
\end{aligned}
$$

with $\mathbf{C}^{\mathrm{e}}$ defined with Eq. (3.151), $\mathbf{Z}$, ϑ defined with Eq. (3.155), and ϑ_1, ϑ_2 – with Eqs. (3.160)–(3.161). Equations (A.14) and (A.15) can easily be transformed to

$$\frac{\mathrm{d}\lambda}{\lambda} = -\gamma\, \boldsymbol{\varsigma}^{\mathrm{T}}\mathbf{P}\, \mathrm{d}\boldsymbol{\varsigma}, \qquad \gamma = \frac{3\vartheta_1}{2\bar{\sigma}\lambda\vartheta_2} = \frac{3}{2\bar{\sigma}^2}\left(\frac{3}{2\lambda}\,\frac{\frac{\partial\tilde{f}}{\partial\bar{\sigma}}}{\frac{\partial\tilde{f}}{\partial\Delta\bar{\varepsilon}^{\mathrm{p}}}} + 1\right), \tag{A.17}$$

which, when substituted to Eqs. (A.13) and (A.16), yields after transformations

$$\mathrm{d}\boldsymbol{\varsigma}_{\mathrm{trial}} = \mathbf{Z}_1\, \mathrm{d}\boldsymbol{\varsigma}\,, \qquad \mathrm{d}\boldsymbol{\sigma} = \mathbf{Z}_2\, \mathrm{d}\boldsymbol{\varsigma}\,, \tag{A.18}$$

with

$$\mathbf{Z}_1 = \vartheta\mathbf{I} + \lambda\mathbf{C}^{\mathrm{e}}\mathbf{P} - \gamma(\vartheta-1)\,\boldsymbol{\varsigma}\,(\mathbf{P}\boldsymbol{\varsigma})^{\mathrm{T}} - \gamma\lambda\,\mathbf{C}^{\mathrm{e}}\,(\mathbf{P}\boldsymbol{\varsigma})\,(\mathbf{P}\boldsymbol{\varsigma})^{\mathrm{T}}, \tag{A.19}$$

$$\mathbf{Z}_2 = \vartheta\mathbf{I} - \gamma(\vartheta-1)\,\boldsymbol{\varsigma}\,(\mathbf{P}\boldsymbol{\varsigma})^{\mathrm{T}}. \tag{A.20}$$

There are many ways of further transformation of the above formulae that lead to different equivalent forms of the algorithmic tangent operator $\mathbf{C}$. Let us continue with the observation that

$$\mathbf{Z}_1 = \mathbf{Z}_3\,\mathbf{Z}_2\,, \qquad \mathbf{Z}_3 = \mathbf{I} + \frac{\lambda}{\vartheta}\,\mathbf{C}^{\mathrm{e}}\mathbf{P} + \frac{\lambda}{\vartheta}\,\frac{1}{\gamma_1}\,\mathbf{C}^{\mathrm{e}}\,(\mathbf{P}\boldsymbol{\varsigma})\,(\mathbf{P}\boldsymbol{\varsigma})^{\mathrm{T}},$$

where

$$\gamma_1 = \frac{2}{3}\,\bar{\sigma}^2(\vartheta-1) - \frac{\vartheta}{\gamma}\,.$$

This allows to represent $\mathrm{d}\boldsymbol{\varsigma}_{\mathrm{trial}}$ as

$$\mathrm{d}\boldsymbol{\varsigma}_{\mathrm{trial}} = \mathbf{Z}_3\,\mathrm{d}\boldsymbol{\sigma}.$$

Consequently,

$$\begin{aligned}\mathbf{C} &= \frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}\Delta\boldsymbol{\varepsilon}} = \frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}\boldsymbol{\varsigma}_{\mathrm{trial}}}\,\frac{\mathrm{d}\boldsymbol{\varsigma}_{\mathrm{trial}}}{\mathrm{d}\Delta\boldsymbol{\varepsilon}} = \mathbf{Z}_3^{-1}\mathbf{C}^{\mathrm{e}} = (\mathbf{C}^{\mathrm{e}-1}\mathbf{Z}_3)^{-1} \\ &= \left[\mathbf{C}^{\mathrm{e}-1} + \frac{\lambda}{\vartheta}\left(\mathbf{P} + \frac{1}{\gamma_1}\,(\mathbf{P}\boldsymbol{\varsigma})\,(\mathbf{P}\boldsymbol{\varsigma})^{\mathrm{T}}\right)\right]^{-1}\end{aligned} \tag{A.21}$$

where

$$\mathbf{C}^{\mathrm{e}-1} = \frac{1}{E}\begin{bmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 1{+}\nu \end{bmatrix}.$$

In particular, for the case of rate-independent elasto-plasticity, we have

$$\gamma_1 = \frac{4\bar{\sigma}^2\lambda}{9\kappa'}(\kappa' + H') = \frac{2\bar{\sigma}^2}{3}\left(\frac{\vartheta}{\vartheta_1} - 1\right), \qquad \vartheta_1 = 1 - \frac{2}{3}\,\lambda\,\kappa'.$$

Upon this assumption, the form (A.21) appears to be equivalent to the form derived in [116],

$$\mathbf{C} = \mathbf{\Xi} - \frac{[\mathbf{\Xi}\mathbf{P}\varsigma]\,[\mathbf{\Xi}\mathbf{P}\varsigma]^{\mathrm{T}}}{\varsigma^{\mathrm{T}}\mathbf{P}\mathbf{\Xi}\mathbf{P}\varsigma + \gamma_1\,\frac{\vartheta}{\lambda}}\,, \qquad \mathbf{\Xi} = \left[\mathbf{C}^{\mathrm{e}-1} + \frac{\lambda}{\vartheta}\,\mathbf{P}\right]^{-1},$$

which can be easily verified after some transformations.

Appendix B

Short description of the numerical program

The numerical program used in the computational examples presented in the dissertation is a finite element analysis code named NS written in its most part by the author at the Department of Computational Science of the Institute of Fundamental Technological Research, PAS. The code is written in the Fortran 77 language.

The fundamental features of the program are listed below.

- Types of analysis: statics and explicit dynamics with geometrical and material nonlinearities, including optional sensitivity analysis with additional possibility of verification of some sensitivity results with the finite difference method.
- Material models:
  - isotropic and orthotropic elasticity (including hyperelasticity with a few forms of the strain energy function),
  - small- and large-deformation elasto-plasticity with Huber–Mises yield criterion and J_2 flow rule, with both kinematic and isotropic hardening
  - small- and large-deformation elasto-viscoplasticity with the overtress and the power-law strain/strain-rate hardening rules.
- Element types:
  - 2D tetragonal element with bi-linear and bi-quadratic shape functions and triangular element with linear and quadratic shape functions; for plane stress, plane strain and axisymmetric analysis
  - 3D brick element with tri-linear shape functions, optionally with reduced or selectively reduced (EAS) integration; 3D wedge and tetrahedral elements with linear shape functions; special elements like e.g. tetragon-based pyramids (the latter were not used in this study).

  - 2D and 3D no-friction large-slip contact elements
  - large rotation 4-node shell element with six parameters at a node

- Load types: precribed nodal forces or displacements (both given as time functions), in dynamical analysis also prescribed initial nodal velocities.

- Boundary conditions: standard blocked or prescribed (as a time function) displacement degrees of freedom; linear equation multi-point constraints.

- Equations system solver: Gauss elimination, sky-line storage of the symmetric system matrix.

Bibliography

[1] *ABAQUS v. 6.4, Theory Manual.* ABAQUS Inc., Pawtucket, RI, USA, 2003.

[2] L. Anand. Constitutive equations for hot-working of metals. *Int. J. Plasticity*, 1:213–231, 1985.

[3] H.J. Antúnez. *Bulk-Metal Forming Processes — From Computational Modelling via Sensitivity analysis to Tool Shape Optimization.* Prace IPPT PAN, Warszawa, 2001. habilitation thesis.

[4] H.J. Antúnez and M. Kleiber. Sensitivity of forming processes to shape parameters. *Comput. Methods Appl. Mech. Engng.*, 137:189–206, 1996.

[5] R.J. Asaro. Micromechanics of crystals and polycrystals. *Adv. Appl. Mech.*, 23:1–115, 1983.

[6] S. Badrinarayanan and N. Zabaras. A sensitivity analysis for the optimal design of metal-forming processes. *Comput. Methods Appl. Mech. Engng.*, 129:319–348, 1996.

[7] D. Balagangadhar and D.A. Tortorelli. Design of large deformation steady elastoplastic manufacturing processes. Part II: Sensitivity analysis and optimization. *Int. J. Numer. Methods Engng.*, 49:933–950, 2000.

[8] K.J. Bathe. *Finite Element Procedures in Engineering Analysis.* Prentice Hall, 1995.

[9] M.P. Bendsøe and J. Sokołowski. Sensitivity analysis and optimization of elastic-plastic structures. *Engng. Optim.*, 11:31–38, 1987.

[10] M.P. Bendsøe and J. Sokołowski. Design sensitivity analysis of elastic-plastic problems. *Struct. Mech.*, 16:81–102, 1988.

[11] C. Bischof, H.M. Bücker, B. Lang, A. Rasch, and J.W. Risch. Extending the functionality of the general-purpose finite element package SEPRAN by automatic differentiation. *Int. J. Numer. Methods Engng.*, 58:2225–2238, 2003.

[12] C. Bischof, A. Carle, P. Khademi, and A. Mauer. The ADIFOR 2.0 system for the automatic differentiation of Fortran 77 programs. *IEEE Comput. Sci. Eng.*, 3:18–32, 1996.

[13] M. Bonnet and S. Mukherjee. Implicit BEM formulations for usual and sensitivity analysis in elasto-plasticity using the consistent tangent operator concept. *Int. J. Solids Struct.*, 33:4461–4480, 1996.

[14] R.I. Borja. Cam-Clay plasticity, Part II: Implicit integration of constitutive equation based on a nonlinear elastic stress predictor. *Comput. Methods Appl. Mech. Engng.*, 88:225–240, 1991.

[15] R.I. Borja and S.R. Lee. Cam-Clay plasticity, Part I: Implicit integration of elasto-plastic constitutive relations. *Comput. Methods Appl. Mech. Engng.*, 78:49–72, 1990.

[16] R.I. Borja and C. Tamagnini. Cam-Clay plasticity, Part III: Extension of the infinitesimal model to include finite strains. *Comput. Methods Appl. Mech. Engng.*, 155:73–95, 1998.

[17] G. Bugeda and L. Gil. Shape sensitivity analysis for structural problems with non-linear material behaviour. *Int. J. Numer. Methods Engng.*, 46:1385–1404, 1999.

[18] J. Casey. Approximate kinematical relations in plasticity. *Int. J. Solids Struct.*, 21:671–682, 1985.

[19] J.L. Chaboche. Time independent constitutive theories for cyclic plasticity. *Int. J. Plasticity*, 2:149–188, 1986.

[20] G. Cheng and Y. Liu. A new computation scheme for sensitivity analysis. *Engng. Optim.*, 12:219–235, 1987.

[21] S. Cho and K.K. Choi. Design sensitivity analysis and optimization of non-linear transient dynamics. Part I: Sizing design. *Int. J. Numer. Methods Engng.*, 48:351–373, 2000.

[22] K.K. Choi and N.-H. Kim. *Structural Sensitivity Analysis and Optimization, Vol. 1 and 2.* Springer, 2005.

[23] K.K. Choi and N.H. Kim. Design optimization of springback in a deepdrawing process. *AIAA J.*, 40:147–153, 2002.

[24] C.C. Chu and A. Needleman. Void nucleation effects in biaxially stretched sheets. *J. Engng. Mater. Technol.*, 102:249–256, 1980.

[25] M.A. Crisfield. *Nonlinear Finite Element Analysis of Solids and Structures, Vol. 1: Essentials, Vol. 2: Advanced Topics.* J. Wiley & Sons, 1991.

[26] Y.F. Dafalias and E.P. Popov. Plastic internal variables formalism of cyclic plasticity. *Trans. ASME, J. Appl. Mech.*, 98:645–651, 1976.

[27] K. Dems and Z. Mróz. Variational approach to first- and second-order sensitivity analysis. *Int. J. Numer. Methods Engng.*, 21:637–646, 1985.

[28] V.S. Deshpande and N.A. Fleck. Isotropic constitutive model for metallic foams. *J. Mech. Phys. Solids*, 48:1253–1276, 2000.

[29] I. Doltsinis and T. Rodić. Process design and sensitivity analysis in metal forming. *Int. J. Numer. Methods Engng.*, 45:661–692, 1999.

[30] D.C. Drucker and W. Prager. Soil mechanics and plastic analysis or limit design. *Quart. Appl. Math.*, 10:157–165, 1952.

[31] W. Fellin and A. Ostermann. Parameter sensitivity in finite element analysis with constitutive models of rate type. *Int. J. Numer. Anal. Methods Geomech.*, 30:91–112, 2006.

[32] P.M. Frank. *Introduction to System Sensitivity Theory.* Academic Press, 1978.

[33] A. Griewank, D. Juedes, and J. Utke. ADOL-C, a package for the automatic differentiation of algorithms written in C/C++. *ACM Trans. on Math. Software*, 22:131–167, 1996.

[34] A.L. Gurson. Continuum theory of ductile rupture by void nucleation and growth: Part I — Yield criteria and flow rules for porous ductile media. *Trans. ASME, J. Engng. Mater. Technol.*, 99:2–15, 1977.

[35] M.A. Gutiérrez and R. de Borst. Studies in material parameter sensitivity of softening solids. *Comput. Methods Appl. Mech. Engng.*, 162:337–350, 1998.

[36] R.T. Haftka. Semi-analytical static nonlinear structural sensitivity analysis. *AIAA J.*, 31:1307–1312, 1993.

[37] R.T. Haftka and Z. Gürdal. *Elements of Structural Optimization.* Kluwer, 1992.

[38] R.T. Haftka and Z. Mróz. First- and second-order sensitivity analysis of linear and nonlinear structures. *AIAA J.*, 24:1187–1192, 1986.

[39] E.W. Hart. Constitutive relations for the nonelastic deformations of metals. *Trans. ASME, J. Engng. Mater. Technol.*, 98:193–202, 1976.

[40] E.J. Haug. Second-order design sensitivity analysis of structural systems. *AIAA J.*, 19:1087–1088, 1981.

[41] E.J. Haug, K.K. Choi, and V. Komkov. *Design Sensitivity Analysis of Structural Systems.* Series in Math. Sci. Engng. Academic Press, 1986.

[42] R. Hill. Continuum micromechanics of elastoplastic polycrystals. *J. Mech. Phys. Solids*, 13:89–101, 1965.

[43] R. Hill. Generalized constitutive relations for incremental deformation of metal crystals by multislip. *J. Mech. Phys. Solids*, 14:95–102, 1966.

[44] R. Hill. *The Mathematical Theory of Plasticity.* Oxford Univ. Press, 1998.

[45] R. Hill and J.R. Rice. Constitutive analysis of elastic/plastic crystals at arbitrary strain. *J. Mech. Phys. Solids*, 20:401–413, 1972.

[46] E. Hinton, editor. *Introduction to Nonlinear Finite Element Analysis.* NAFEMS, 1992.

[47] M.T. Huber. Właściwa praca odkształcenia jako miara wytężenia materyału. Przyczynek do podstaw teoryi wytrzymałości. *Czasopismo Techniczne*, XXII, 1904.

[48] T.J.R. Hughes. Generalization of selective integration procedures to anisotropic and nonlinear media. *Int. J. Numer. Methods Engng.*, 15:1413–1418, 1980.

[49] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Dover, 2000.

[50] J.W. Hutchinson. Elastic-plastic behavior of polycrystalline metals and composites. *Proc. Roy. Soc. London Ser. A*, 319:247–272, 1970.

[51] S.Y. Jao and J.S. Arora. Design sensitivity analysis of nonlinear structures using endochronic constitutive model. Part I: General development; Part II: Discretization and applications. *Comput. Mech.*, 10:39–57, 59–72, 1992.

[52] B. Jeremić, K. Runneson, and S. Sture. A model for elastic-plastic pressure sensitive material subjected to large deformations. *Int. J. Solids Struct.*, 36:4901–4918, 1999.

[53] P. Jetteur. Implicit integration algorithm for elastoplasticity in plane stress analysis. *Engng. Comput.*, 3:251–253, 1986.

[54] M.S. Joun and S.M. Hwang. Die shape optimal design in three-dimensional shape metal extrusion by the finite element method. *Int. J. Numer. Methods Engng.*, 41:311–335, 1998.

[55] F. van Keulen and H. de Boer. Rigorous improvement of semianalytical design sensitivities by exact differentiation of rigid body motions. *Int. J. Numer. Methods Engng.*, 42:71–91, 1997.

[56] F. van Keulen, R.T. Haftka, and N.H. Kim. Review of options for structural design sensitivity analysis. Part 1: Linear systems. *Comput. Meth. Appl. Mech. Engng.*, 194:3213–3243, 2005.

[57] A.S. Khan and S. Huang. *Continuum Theory of Plasticity.* J. Wiley & Sons, 1995.

[58] N.H. Kim and K.K. Choi. Design sensitivity analysis and optimization of nonlinear transient dynamics. *Mech. Struct. Mach.*, 29:351–371, 2001.

[59] N.H. Kim, K.K. Choi, and J.S. Chen. Shape design sensitivity analysis and optimization of elasto-plasticity with frictional contact. *AIAA J.*, 38:1742–1753, 2000.

[60] N.H. Kim, K.K. Choi, and J.S. Chen. Structural optimization of finite deformation elastoplasticity using continuum-based shape design sensitivity formulation. *Comput. Struct.*, 79:1959–1976, 2001.

[61] M. Kleiber. *Incremental Finite Element Modelling in Non-Linear Solid Mechanics.* Ellis Horwood – PWN, 1989.

[62] M. Kleiber. Shape and non-shape sensitivity analysis for problems with any material and kinematic nonlinearity. *Comput. Methods Appl. Mech. Engng.*, 108:73–97, 1993.

[63] M. Kleiber and T.D. Hien. Parameter sensitivity of inelastic buckling and post-buckling response. *Comput. Methods Appl. Mech. Engng.*, 145:239–262, 1998.

[64] M. Kleiber, T.D. Hien, H. Antúnez, and P. Kowalczyk. Parameter sensitivity of elasto-plastic response. *Engng. Comput.*, 12:263–280, 1995.

[65] M. Kleiber, T.D. Hien, H. Antúnez, and P. Kowalczyk. *Parameter Sensitivity in Nonlinear Mechanics.* J. Wiley & Sons, 1997.

[66] M. Kleiber and T. Hisada, editors. *Design Sensitivity Analysis.* Atlanta Tech. Publ., 1992.

[67] M. Kleiber and P. Kowalczyk. Constitutive parameter sensitivity in elasto-plasticity. *Comput. Mech.*, 17:36–48, 1995.

[68] M. Kleiber and P. Kowalczyk. Sensitivity analysis in plane stress elasto-plasticity and elasto-viscoplasticity. *Comput. Methods Appl. Mech. Engng.*, 137:395–409, 1996.

[69] M. Kleiber, A. Siemaszko, and R. Stocki. Interactive reliability-based design optimization of geometrically nonlinear structures. In *3rd World Congress of Structural and Multidisciplinary Optimization*, pages 538–540, Buffalo, NY, May 17-21, 1999.

[70] J. Korelc. Symbolic methods in numerical analysis. http://www.fgg.uni-lj.si/Symech/, 2000.

[71] P. Kowalczyk. Design sensitivity analysis in large deformation elasto-plastic and elasto-viscoplastic problems. *Int. J. Numer. Methods Engng.*, 66:1234–1270, 2006.

[72] P. Kowalczyk and M. Kleiber. Parameter sensitivity for large deformation inelastic problems. *Comput. Assisted Mech. Engng. Sci.*, 4:209–228, 1997.

[73] P. Kowalczyk and M. Kleiber. Shape sensitivity in elasto-plastic computations. *Comput. Methods Appl. Mech. Engng.*, 171:371–386, 1999.

[74] R.D. Krieg and S.W. Key. Implementation of a time-dependent plasticity theory into structural computer programs. In J.A. Stricklin and K.J. Saczalski, editors, *Constitutive Equations in Viscoplasticity: Computational and Engineering Aspects*, AMD-20. ASME, New York, 1976.

[75] E.H. Lee. Elastic-plastic deformations at finite strains. *Trans. ASME, J. Appl. Mech.*, 36:1–6, 1969.

[76] E.H. Lee and D.T. Liu. Finite-strain elatic-plastic theory with applications to plane-wave analysis. *J. Appl. Phys.*, 38:19–27, 1967.

[77] T.H. Lee and J.S. Arora. A computational method for design sensitivity analysis of elasto-plastic structures. *Comput. Methods Appl. Mech. Engng.*, 122:27–50, 1995.

[78] T.H. Lee, J.S. Arora, and V. Kumar. Shape design sensitivity analysis of viscoplastic structures. *Comput. Methods Appl. Mech. Engng.*, 108:237–259, 1993.

[79] J. Lemaitre and J.L. Chaboche. *Mechanics of Solid Materials.* Cambridge Univ. Press, 1990.

[80] L.-J. Leu and S. Mukherjee. Sensitivity analysis of hyperelastic-viscoplastic solids undergoing large deformations. *Comput. Mech.*, 15:101–116, 1994.

[81] J. Lubliner. *Plasticity Theory.* Macmillan, 1990.

[82] A.M. Maniatty and M.-F. Chen. Shape sensitivity analysis for steady metal-forming processes. *Int. J. Numer. Methods Engng.*, 39:1199–1217, 1996.

[83] K. Maute and E. Ramm. Adaptive topology optimization. *Struct. Optim.*, 10:100–112, 1995.

[84] H. Menrath, K. Maute, and E. Ramm. Topology optimization including elasto-plasticity. In D.R.J. Owen, E. Oñate, and E. Hinton, editors, *Computational Plasticity. Fundamentals and Applications*, pages 817–822. CIMNE, Barcelona, 1997.

[85] P. Michaleris, D.A. Tortorelli, and C.A. Vidal. Tangent operators and design sensitivity formulations for transient nonlinear coupled problems with applications to elastoplasticity. *Int. J. Numer. Methods Engng.*, 37:2471–2499, 1994.

[86] R. von Mises. Mechanik des festen Körpers im plastisch deformablen Zustand. *Nachr. Königl. Ges. Wiss. Göttingen, Math. Phys. Kl.*, pages 582–592, 1913.

[87] G.P. Mitchell and D.R.J. Owen. Numerical solutions for elasto-plastic problems. *Engng. Comput.*, 5:274–284, 1988.

[88] B. Moran, M. Ortiz, and C.F. Shih. Formulation of implicit finite element methods for multiplicative finite deformation plasticity. *Int. J. Numer. Methods Engng.*, 29:483–514, 1990.

[89] Z. Mróz. On the description of anisotropic hardening. *J. Mech. Phys. Solids*, 15:163–175, 1967.

[90] Z. Mróz and R.T. Haftka. Design sensitivity analysis of non-linear structures in regular and critical states. *Int. J. Solids Struct.*, 31:2071–2098, 1994.

[91] Z. Mróz and J. Piekarski. Sensitivity analysis and optimal design of non-linear structures. *Int. J. Numer. Methods Engng.*, 42:1231–1262, 1998.

[92] S. Mukherjee and A. Chandra. A boundary element formulation for design sensitivity in materially nonlinear problems. *Acta Mech.*, 78:243–253, 1989.

[93] S. Nemat-Nasser. On finite strain elasto-plasticity. *Int. J. Solids Struct.*, 18:857–872, 1982.

[94] J.T. Oden. *Finite Elements of Nonlinear Continua.* McGraw–Hill, 1972.

[95] M. Ohsaki. Sensitivity analysis of elasto-plastic structures by using explicit integration method. *Appl. Mech. Rev.*, 50:S156–S161, 1997.

[96] M. Ohsaki and J.S. Arora. Design sensitivity analysis of elasto-plastic structures. *Int. J. Numer. Methods Engng.*, 37:737–762, 1994.

[97] N. Olhoff and J. Rasmussen. Study of inaccuracy in semi-analytical sensitivity analysis — a model problem. *Struct. Optim.*, 3:203,213, 1991.

[98] N. Olhoff, J. Rasmussen, and E. Lund. A method of“exact” numerical differentiation for error elimination in finite-element-based semi-analytical shape sensitivity analyses. *Mech. Struct. Mach.*, 21:1–66, 1993.

[99] I. Ozaki, F. Kimura, and M. Berz. Higher-order sensitivity analysis of finite element method by automatic differentiation. *Comput. Mech.*, 16:223–234, 1995.

[100] E. Parente Jr. and L.E. Vaz. Improvement of semi-analytical design sensitivities of nonlinear structures using equilibrium relations. *Int. J. Numer. Methods Engng.*, 50:2127–2142, 2001.

[101] P. Pedersen, G. Cheng, and J. Rasmussen. On accuracy problems for semi-analytical sensitivity analysis. *Mech. Struct. Mach.*, 17:373–384, 1989.

[102] P. Perzyna. The constitutive equation for rate-sensitive plastic materials. *Quart. Appl. Math.*, 20:321–332, 1963.

[103] D. Pierce, C.F. Shih, and A. Needleman. A tangent modulus method for rate-dependent solids. *Comput. Struct.*, 18:875–887, 1984.

[104] E. Pramono and K. Willam. Implicit integration of composite yield surfaces with corners. *Engng. Comput.*, 6:186–197, 1989.

[105] J.N. Reddy. *An Introduction to the Finite Element Method.* McGraw–Hill, 2nd edition, 1993.

[106] E. Rohan and J.R. Whiteman. Shape optimization of elasto-plastic structures and continua. *Comput. Methods Appl. Mech. Engng.*, 187:261–288, 2000.

[107] Y.S. Ryu, M. Haririan, C.C. Wu, and J.S. Arora. Structural design sensitivity analysis of nonlinear response. *Comput. Struct.*, 21:245–255, 1985.

[108] A. Saltelli, K. Chan, and E.M. Scott, editors. *Sensitivity Analysis.* J. Wiley & Sons, 2000.

[109] A. Schofield and C.P. Wroth. *Critical State Soil Mechanics.* McGraw–Hill, 1968.

[110] J.C. Simo. A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition. Part I. Continuum formulation. *Comput. Methods Appl. Mech. Engng.*, 66:199–219, 1988.

[111] J.C. Simo. A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition. Part II. Computational aspects. *Comput. Methods Appl. Mech. Engng.*, 68:1–31, 1988.

[112] J.C. Simo and T.J.R. Hughes. General return mapping algorithms for rate independent plasticity. In C.S. Desai, editor, *Constitutive Equations for Engineering Materials*, pages 221–231. Elsevier, 1987.

[113] J.C. Simo and T.J.R. Hughes. *Computational Inelasticity.* Springer, 1998.

[114] J.C. Simo, J.G. Kennedy, and S. Govindjee. Non-smooth multisurface plasticity and viscoplasticity. Loading/unloading conditions and numerical algorithms. *Int. J. Numer. Methods Engng.*, 26:2161–2185, 1988.

[115] J.C. Simo and R.L. Taylor. Consistent tangent operators for rate-independent elastoplasticity. *Comput. Methods Appl. Mech. Engng.*, 48:101–118, 1985.

[116] J.C. Simo and R.L. Taylor. A return mapping algorithm for plane stress elastoplasticity. *Int. J. Numer. Methods Engng.*, 22:649–670, 1986.

[117] W. Sosnowski. *Finite Element Simulation of Industrial Sheet Metal Forming Processes.* Prace IPPT PAN, Warszawa, 1995. habilitation thesis.

[118] W. Sosnowski, I. Marczewska, and A. Marczewski. Sensitivity based optimization of sheet metal forming tools. *J. Mater. Proc. Technol.*, 124:319–328, 2002.

[119] A. Srikanth and N. Zabaras. An updated Lagrangian finite element sensitivity analysis of large deformations using quadrilateral elements. *Int. J. Numer. Methods Engng.*, 52:1131–1163, 2001.

[120] D.W. Stillman. Design sensitivity analysis for structures using explicit time integration. In *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000. Sept. 6–8, 2000, Long Beach, CA, *AIAA*-2000-4906.

[121] S. Stupkiewicz. Approximate response sensitivities for nonlinear problems in explicit dynamic formulation. *Struct. Multidisc. Optim.*, 21:283–291, 2001.

[122] S. Stupkiewicz, J. Korelc, M. Dutko, and T. Rodič. Shape sensitivity analysis of large deformation frictional contact problems. *Comput. Methods Appl. Mech. Engng.*, 191:3555–3581, 2002.

[123] H. Tresca. Sur l'écoulement des corps solides soumis á de fortes pressions. *Comptes Rendus Acad. Sci. Paris*, 59:754, 1864.

[124] J.J. Tsay and J.S. Arora. Nonlinear structural design sensitivity analysis for path dependent problems. Part 1: General theory. *Comput. Methods Appl. Mech. Engng.*, 81:183–208, 1990.

[125] J.J. Tsay, J.E.B. Cardoso, and J.S. Arora. Nonlinear structural design sensitivity analysis for path dependent problems. Part 2: Analytical examples. *Comput. Methods Appl. Mech. Engng.*, 81:209–228, 1990.

[126] V. Tvergaard. Material failure by void coalescence in localized shear bands. *Int. J. Solids Struct.*, 18:659–672, 1982.

[127] K.C. Valanis. A theory of viscoplasticity without a yield surface. *Arch. Mech.*, 23:517–533, 1971.

[128] C.A. Vidal and R.B. Haber. Design sensitivity analysis for rate-independent elasto-plasticity. *Comput. Methods Appl. Mech. Engng.*, 107:393–431, 1993.

[129] C.A. Vidal, H.S. Lee, and R.B. Haber. The consistent tangent operator for design sensitivity analysis of history-dependent response. *Computing Systems in Engng.*, 2:509–523, 1991.

[130] G. Weber and L. Anand. Finite deformation constitutive equations and a time integration procedure for isotropic hyperelastic-viscoplastic solids. *Comput. Methods Appl. Mech. Engng.*, 79:173–202, 1990.

[131] M.L. Wilkins. Calculation of elastic-plastic flow. In B. Adler, S. Fernbach, and M. Rottenberg, editors, *Methods in Computational Physics*. Academic Press, 1964.

[132] K. Wiśniewski, P. Kowalczyk, and E. Turska. On the computation of design derivatives for Huber-Mises plasticity with nonlinear hardening. *Int. J. Numer. Methods Engng.*, 57:271–300, 2003.

[133] K. Wiśniewski, P. Kowalczyk, and E. Turska. Analytical dsa for explicit dynamics of elastic-plastic shells. *Comput. Mech.*, 2006. (in print; on line at DOI 10.1007/s00466-006-0068-3).

[134] Q. Zhang, S. Mukherjee, and A. Chandra. Shape design sensitivity analysis for geometrically and materially nonlinear problems by the boundary element method. *Int. J. Solids Struct.*, 29:2503–2525, 1992.

[135] Y. Zhang and A. Der Kiureghian. Dynamic response sensitivity of inelastic structures. *Comput. Methods Appl. Mech. Engng.*, 108:23–36, 1993.

[136] G. Zhao, X. Ma, X. Zhao, and R.V. Grandhi. Studies on optimization of metal forming processes using sensitivity analysis methods. *J. Mater. Proc. Technol.*, 147:217–228, 2004.

[137] G. Zhao, E. Wright, and R.V. Grandhi. Preform die shape design using an optimization method. *Int. J. Numer. Methods Engng.*, 40:1213–1230, 1997.

[138] H. Ziegler. A modification of Prager's hardening rule. *Quart. Appl. Math.*, 17:55–61, 1959.

[139] O.C. Zienkiewicz, A.H.C. Chen, M. Pastor, B.A. Schrefler, and T. Shiomi. *Computational Geomechanics with Special Reference to Earthquake Engineering*. J. Wiley & Sons, 1999.

[140] O.C. Zienkiewicz, P. Jain, and E. Oñate. Flow of solids during forming and extrusion: some aspects of numerical solutions. *Int. J. Solids Struct.*, 14:15–38, 1978.

[141] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*. Butterworth–Heinemann, 5th edition, 2000.

Streszczenie i elementy oryginalne rozprawy

Niniejsza rozprawa jest poświęcona metodom analizy wrażliwości parametrycznej w obliczeniach numerycznych zagadnień sprężysto-plastycznych. Określenie gradientów wrażliwości dla rozwiązań obejmujących tak szeroki wachlarz zjawisk nieliniowych, zarówno geometrycznych jak materiałowych, stanowi wyzwanie dla badaczy, a jednocześnie odpowiedź na potrzeby przemysłu, gdzie zjawiska sprężysto-plastyczne w materiałach odgrywają istotna rolę z punktu widzenia bezpieczeństwa, kosztów i funkcjonalności produktu. W rozprawie przedstawione są przede wszystkim analityczne metody wrażliwości, w których gradienty wyznaczane są ściśle, w przeciwieństwie do metod opierających się na przybliżeniach metody różnic skończonych dla małych perturbacji parametrów projektowych.

Rozważania przedstawione w pracy dotyczą analizy wrażliwości pierwszego rzędu w geometrycznie liniowych oraz nieliniowych zagadnieniach obliczeniowych izotermicznej sprężysto-plastyczności metali, między innymi również z uwzględnieniem zjawisk lepkoplastycznych. Treść rozprawy jest w dużej mierze kompilacją wyników oryginalnych badań naukowych autora, prowadzonych w latach 1994–2006 w Instytucie Podstawowych Problemów Techniki PAN, przy współpracy z kolegami z Zakładu Metod Komputerowych, i opublikowanych w pracach [64, 65, 67, 68, 71–73, 132, 133]. Trzon rozprawy stanowią rozważania na temat układów statycznych, choć dyskusja obejmuje również rozszerzenia przedstawionych sformułowań w kierunku obliczeń dynamicznych. Problemy wrażliwości w zagadnieniach np. stateczności lub analizy drgań, w których niezbędne jest wyznaczanie wartości i wektorów własnych układu dyskretnego, nie są w pracy podejmowane.

Wszystkie zagadnienia brzegowo-początkowe rozważane w rozprawie są dyskutowane w kontekście przybliżonych sformułowań dyskretnych opartych na metodzie elementów skończonych (MES). Ograniczenie to nie umniejsza ogólnego charakteru rozważań i wniosków, przynajmniej dla przypadku wrażliwości na parametry materiałowe i wymiarowe. W zagadnieniach wrażliwości kształtu, kluczowe pojęcie niezależnej od parametrów projektowych geometrycznej konfiguracji bazowej jest w niniejszej pracy utożsamione z konfiguracją bazową izoparametrycznego elementu skończonego — stąd dyskusja tej grupy zagadnień ogranicza się do sformułowań dyskretnych w ujęciu MES.

Rozdział 1 zawiera wprowadzenie do tematu, z omówieniem historii i obecnego stanu badań w dziedzinie wrażliwości parametrycznej zagadnień sprężysto-plastycznych.

Rozdział 2 prezentuje ogólną ideę analizy wrażliwości nieliniowych układów mechanicznych. Zostało tu przedstawione ogólne sformułowanie problemu podstawowego, zarówno w ujęciu ciągłym jak też w dyskretnym, oraz wprowadzone zostały pojęcia parametrów projektowych i funkcjonałów stanu. Rozróżniono następnie dwa fundamentalne podejścia do analizy wrażliwości: ciągłe i dyskretne, z dyskusją na temat ich zalet i wad i wskazaniem pewnych metod pośrednich. W dalszej części rozprawy rozważano dokładniej jedynie podejście dyskretne, z dwiema alternatywnymi strategiami rozwiązania: metodą bezpośredniego różniczkowania (DDM) i metodą układu sprzężonego (ASM). Przedyskutowano w sposób krytyczny podejścia: analityczne, półanalityczne oraz różnic skończonych, do wyznaczania gradientów wrażliwościowych. Na koniec podniesiona została kwestia różniczkowalności rozwiązań sprężysto-plastycznych względem parametrów projektowych i możliwych w konsekwencji tego faktu nieciągłości rozwiązań wrażliwościowych.

Poprawne sformułowanie zagadnienia wrażliwości wymaga dogłębnego zrozumienia sformułowania zagadnienia podstawowego. Rozdział 3 przedstawia zatem szczegółowo zagadnienie statycznej deformacji ciała sprężysto-plastycznego, a w szczególności sformułowania konstytutywne dla małych i dużych deformacji w ujęciu prędkościowym i przyrostowym, oraz sformułowanie dyskretne globalnych równań równowagi w ujęciu MES. Wprawdzie materiał ten niemal nie zawiera oryginalnych wyników autora, jednak jego umieszczenie w pracy jest niezbędne, gdyż następujące w kolejnym rozdziale wyprowadzenie równań wrażliwości zawiera liczne odwołania do notacji i szczegółów równań zagadnienia podstawowego.

W Rozdziale 4 przedstawiono wyprowadzenie sformułowania wrażliwości dla tak zdefiniowanego problemu podstawowego. Jest to zasadnicza, oryginalna część rozprawy. Podniesione zostały w szczególności aspekty obliczeniowe analizy wrażliwości, między innymi kluczowa rola algorytmicznej, konsystentnej macierzy stycznej. Omówiono szczegółowo kwestię wrażliwości na parametry kształtu. Pokazano, że nie ma istotnych sprzeczności między sformułowaniami dla parametrów wpływających i niewpływających na początkową geometrię układu (podejście zunifikowane). Następnie, zróżniczkowano równania pojawiające się w procedurach konstytutywnych sprężysto-plastyczności względem zmiennych wejściowych (zależnych od parametrów projektowych) w celu otrzymania całkowitych i cząstkowych pochodnych projektowych zmiennych stanu, niezbędnych w globalnym sformułowaniu analizy wrażliwości. Podkreślono wniosek o liniowym (przynajmniej na kroku obliczeniowym) charakterze rozwiązania wrażliwościowego, nawet w przypadku silnych nieliniowości w równaniach konsystencji modelu konstytutywnego. Wyprowadzone sformułowania przedstawiono w postaci zamkniętych algorytmów obliczeniowych, gotowych do implementacji od ręki w programie komputerowym. Przykłady obliczeniowe, zamieszczone na końcu rozdziału, ilustrują zaprezentowane algorytmy i inspirują dyskusję na temat szczegółowych kwestii omówionych w rozdziale.

Rozdział 5 obejmuje rozszerzenie rozważań z poprzednich rozdziałów na przypadek analizy dynamicznej układów sprężysto-plastycznych. Przedstawiono równania analizy podstawowej oraz wyprowadzone z nich równania analizy wrażliwości w ujęciu metody elementów skończonych. Przedyskutowano dwie strategie obliczeń dynamicznych: niejawną i jawną, szczególną uwagę poświęcając tej drugiej, jako bardziej popularnej w praktycznych obliczeniach inżynierskich, a jednocześnie wymagającej nieco innego podejścia do analizy wrażliwości, niż dotychczas omówione. Wyprowadzenie równań matematycznych i podanie algorytmów obliczeniowych zostało zilustrowane kolejnymi przykładami obliczeniowymi.

Rozdział 6 zawiera uwagi na temat praktycznej implementacji omówionych algorytmów analizy wrażliwości w programie metody elementów skończonych. Wskazano zarówno na trudności (duża ilość niezbędnych dodatkowych instrukcji i procedur), jak też na zalety (np. zautomatyzowany charakter dużej części pracy do wykonania przez programistę). Przedyskutowano kwestię możliwości automatycznej generacji kodu na różnych poziomach sformułowań.

W Rozdziale 7 sformułowano wnioski i przewidywane kierunki dalszych badań.

Niniejsza praca jest rozprawą habilitacyjną autora. Poniżej wyszczególniono jej elementy stanowiące oryginalny wkład autora w uprawianej dziedzinie badań naukowych.

Rozdział 2:

- wnioski na temat zakresu stosowalności DDM i ASM w nieliniowych zagadnieniach zależnych i niezależnych od ścieżki obciążania,
- przykłady obliczeniowe i wnioski dotyczące nieciągłości rozwiązań wrażliwościowych w zagadnieniach sprężysto-plastycznych,

- przykład obliczeniowy i wnioski dotyczące wyboru wielkości perturbacji w analizie wrażliwości metodami pół-analityczną i różnic skończonych.

Rozdział 3:

- rozszerzenie modelu konstytutywnego Simo [110, 111] dla dużych deformacji sprężysto-plastycznych oraz [115, 116] dla małych deformacji o nieliniową postać funkcji wzmocnienia oraz lepkoplastyczne funkcje płynięcia.

Rozdział 4:

- zróżniczkowanie względem parametrów projektowych globalnych równań metody elementów skończonych, zapisanych zarówno w opisie materialnym jak też przestrzennym, dla parametrów projektowych wpływających na stałe konstytutywne oraz na początkową geometrię układu,
- zróżniczkowanie względem parametrów projektowych przyrostowych równań konstytutywnych sprężysto-plastyczności i wyprowadzenie algorytmów obliczeniowych służących do wyznaczania zarówno pełnych, jak też odpowiednich cząstkowych pochodnych projektowych naprężenia i parametrów stanu,
- przykłady obliczeniowe oraz wnioski z towarzyszącej im dyskusji na temat wpływu nieróżniczkowalności i niedokładności wyznaczenia rozwiązań zagadnienia podstawowego na jakość rozwiązań wrażliwościowych, oraz na temat efektywności numerycznej analizy wrażliwości.

Rozdział 5:

- rozszerzenie sformułowania wrażliwości dla statycznych zagadnień sprężysto-plastycznych na zagadnienia dynamiki,
- wnioski na temat efektywności numerycznej analizy wrażliwości w sformułowaniach jawnych i niejawnych dynamiki,
- przykłady obliczeniowe oraz wnioski z towarzyszącej im dyskusji.

Rozdział 6:

- przykłady i wnioski na temat ‘ręcznego’ i automatycznego różniczkowania kodów i sformułowań matematycznych oraz zakresu ich stosowalności w implementacji numerycznej analizy wrażliwości.