2 SENSITIVITY ANALYSIS IN FINITE ELEMENT SIMULATIONS

2.1 Finite Element Simulations

The aim of numerical simulations is to predict the behaviour of a system under consideration. In the finite element approach this is performed by solving a set of algebraic equations, which can be expressed in the residual form

$$\mathbf{R}(\mathbf{u}) = \mathbf{0}\,. \tag{2.1}$$

The above equations represent the discretised form of the governing equations including balance laws, constitutive equations, and initial and boundary conditions, which arise in mechanical, thermal, or electromagnetic problems. Unknowns $\mathbf{u}$ define approximate solution and are considered as the primary system response. System (2.1) represents a wide variety of problems and description of finite element techniques to solve particular problems are beyond the scope of this work. A large amount of literature covers this topic, e.g.. references [1] - [7]. This section is focused on basic aspects of sensitivity analysis[10]-[14] for nonlinear problems, which is crucial for efficient optimisation procedures.

The system (2.1) can be solved by the Newton-Raphson method, in which the following iteration is performed (chapter 3, [1]-[7]):

$$\frac{d\mathbf{R}}{d\mathbf{u}}\left(\mathbf{u}^{(i)}\right)\delta\,\mathbf{u} = -\mathbf{R}\left(\mathbf{u}^{(i)}\right), \tag{2.2}$$

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \delta\,\mathbf{u}\,. \tag{2.3}$$

The term $\mathbf{R}\left(\mathbf{u}^{(i)}\right)$ is referred to as the residual (or load) vector and the term $\frac{d\mathbf{R}}{d\mathbf{u}}\left(\mathbf{u}^{(i)}\right)$ is referred to as the tangent operator (or tangential stiffness matrix).

For time dependent problems the iteration scheme given by (2.2) and (2.3) is not sufficient since the state of the system at different times must be determined. Time is usually treated differently to the spatial independent variables. The time domain is discretised according to the finite difference scheme in which approximate states are evaluated for discrete times ${}^{(1)}t, {}^{(2)}t, \ldots, {}^{(M)}t$. Solution for intermediate times is usually linearly interpolated within the intervals $\left[{}^{(n)}t, {}^{(n+1)}t\right]$ and time derivatives of the time dependent quantities are approximated by finite difference expressions.

The approximate solution for the *n*-th time step is obtained by solution of the residual equations

$$ {}^{(n)}\mathbf{R}\left({}^{(n)}\mathbf{u}, {}^{(n-1)}\mathbf{u}\right) = \mathbf{0}\,, \tag{2.4} $$

which are solved for each time step (or increment) for ${}^{(n)}\mathbf{u}$ while ${}^{(n-1)}\mathbf{u}$ is known from the previous time step. Dependence on earlier increments (${}^{(n-2)}\mathbf{u}$, etc.) is possible when higher order time derivatives are present in the continuum equations (e.g. [8]). The system (24) can again be solved by the Newton-Raphson method in which the following iteration is performed[1]:

$$ \frac{d\,{}^{(n)}\mathbf{R}}{d\,{}^{(n)}\mathbf{u}}\left({}^{(n)}\mathbf{u}^{(i)}\right)\delta\,\mathbf{u} = -{}^{(n)}\mathbf{R}\left({}^{(n)}\mathbf{u}^{(i)}\right), \tag{2.5} $$

$$ {}^{(n)}\mathbf{u}^{(i+1)} = {}^{(n)}\mathbf{u}^{(i)} + \delta\,\mathbf{u}\,. \tag{2.6} $$

The incremental scheme is not used only for transient but also for path dependent problems such as plasticity[22] where constitutive laws depend on evolution of state variables, which inherently calls for an incremental approach[9],[10]. Material response is not necessarily time dependent and the time can be replaced by some other parameter, referred to as pseudo time. Treatment of path dependent material behaviour requires introduction of additional internal state variables, which serve for description of the history effect.

The state of a continuum system is often defined by two distinct fields, e.g. the temperature and displacement fields. Two sets of governing equations define the solution for both types of variables. When neither of these variables can be

1 The Euler backward integration scheme is considered here, but other schemes such as variable midpoint algorithms can also be incorporated.

eliminated by using one set of equations, both sets must be solved simultaneously and the system is said to be coupled. The approximate solution is obtained by solving two sets of residual equations in each time step:

$$^{(n)}\mathbf{R}\left({}^{(n)}\mathbf{u}, {}^{(n-1)}\mathbf{u}, {}^{(n)}\mathbf{v}, {}^{(n-1)}\mathbf{v}\right) = \mathbf{0} \tag{2.7}$$

and

$$^{(n)}\mathbf{H}\left({}^{(n)}\mathbf{u}, {}^{(n-1)}\mathbf{u}, {}^{(n)}\mathbf{v}, {}^{(n-1)}\mathbf{v}\right) = \mathbf{0}\,. \tag{2.8}$$

Different solution schemes[9],[16] include either solution of both systems simultaneously in an iteration system, or solution of the systems separately for one set of variables while keeping the other set fixed; the converged sets of variables are in this case exchanged between the two systems.

In the present work the developed optimisation methodology was applied to metal forming processes. Simulations of these processes must take into account complex path dependent and coupled material behaviour. A survey of modelling approaches fot this behaviour can be found in [9].

2.2 Sensitivity Analysis

For the purpose of optimisation the notion of parametrisation is introduced. We want to change the setup of the considered system either in terms of geometry, constitutive parameters, initial or boundary conditions, or a combination of these. A set of design parameters $\Phi = \left[\phi_1, \phi_2, ..., , \phi_n\right]$ is used to describe the properties of the system which can be varied. The equations which govern the system and therefore the numerical solution depend on the design parmeters.

To define optimisation problems certain quantities of interest such as the objective and constraint functions must be defined. For many optimisation algorithms the derivatives of these quantities with respect to the design parameters (i.e. sensitivities) are important. Evaluation of these derivatives is the subject of sensitivity analysis[10]-[13], which is introduced in this section in terms of basic formalism. For this purpose, let us consider a general function, which is dependent on the design parameters which define the system of interest:

$$F(\Phi) = G(\mathbf{u}(\Phi), \Phi) \tag{2.9}$$

F is referred to as the response functional and appears as a term in the objective or constraint functions. F will be typically defined through a system response $\mathbf{u}$, but may in addition include explicit dependence on the design parameters, as is indicated by the right hand side of (2.9). One way of evaluating derivatives $dF/d\phi_i$ is numerical evaluation by the finite difference formula

$$\frac{dF}{d\phi_k}(\phi_1,\phi_2,...,\phi_n) \approx \frac{F(\phi_1,...,\phi_{k-1},\phi_k+\Delta\phi_k,\phi_{k+1}...,\phi_n)-F(\phi_1,...,\phi_{k-1},\phi_k,\phi_{k+1}...,\phi_n)}{\Delta\phi_k}. \quad (2.10)$$

Evaluation of each derivative requires additional evaluation of F at a perturbed set of design parameters, which includes numerical evaluation of the system response $\mathbf{u}$ at the perturbed parameters. More effective schemes, which are incorporated in a solution procedure for evaluation of the system response, are described below.

Derivation of (2.9) with respect to a specific design parameter $\phi = \phi_k$ [1] gives

$$\frac{dF}{d\phi} = \frac{\partial G}{\partial \mathbf{u}}\frac{d\mathbf{u}}{d\phi} + \frac{\partial G}{\partial \phi}. \quad (2.11)$$

Derivatives $\partial G/\partial \mathbf{u}$ and $\partial G/\partial \phi$ are determined explicitly by definition of the functional F. The main task of the sensitivity analysis is therefore evaluation of the term $d\mathbf{u}/d\phi$, which is an implicit quantity because the system response $\mathbf{u}$ depends on the design parameters implicitly through numerical solution of the governing equations.

Let us first consider *steady state problems* where the approximate system response can be obtained by solution of a single set of non-linear equations (2.1). Since the system is parametrised, these equations depend on the design parameters and can be restated as

$$\mathbf{R}(\mathbf{u}(\Phi),\Phi) = \mathbf{0}. \quad (2.12)$$

This equation defines implicit dependence of the system response on the design parameters and will be used for derivation of formulae for implicit sensitivity terms.

In the *direct differentiation method* the term $d\mathbf{u}/d\phi$ is obtained directly by derivation of (2.12) with respect to a specific parameter ϕ, which yields

[1] Index k is suppressed in order to simplify the derived expressions.

$$\frac{\partial \mathbf{R}}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\phi} = -\frac{\partial \mathbf{R}}{\partial \phi}. \tag{2.13}$$

This set of linear equations must be solved for each design parameter in order to obtain the appropriate implicit term $d\mathbf{u}/d\phi$. This term is then substituted into (2.11) in order to obtain the derivative of the functional F with respect to that parameter. The equation resembles (2.2), which is solved iteratively to obtain the approximate system response. According to this analogy (2.13) is often referred to as a pseudoproblem for evaluation of the implicit sensitivity terms, and the right-hand side $-\partial \mathbf{R}/\partial \phi$ is referred to as the pseudoload. As opposed to (2.2), (2.13) is solved only once at the end of the iterative scheme, because the tangent operator $\partial \mathbf{R}/\partial \mathbf{u}$ evaluated for the converged solution $\mathbf{u}$ (where equations (2.12) are satisfied) must be taken into account for evaluation of sensitivities. If the system of equations (2.2) is solved by decomposition of the stiffness matrix, then the decomposed tangent stiffness matrix from the last iteration can be used for solution of (2.13), which means that the additional the computational cost includes only back substitution. Evaluation of derivatives with respect to each design parameter therefore contributes only a small portion of computational cost required for solution of (2.12) as opposed to the finite difference scheme, where evaluation of the derivative with respect to each parameter requires a complete solution of (2.12) for the corresponding perturbed design. An additional complication is evaluation of the load vector $-\partial \mathbf{R}/\partial \phi$. It requires explicit derivation of the finite element formulation (more precisely the formulae for evaluation of element contributions to the stiffness matrix) with respect to design parameters, which must be incorporated in the numerical simulation.

An alternative method for evaluation of sensitivities is the *adjoint method*. In this method the implicit term $d\mathbf{u}/d\phi$ is eliminated from (2.11). An augmented functional

$$\tilde{F}(\Phi) = G(\mathbf{u}(\Phi), \Phi) - \lambda^T \mathbf{R}(\mathbf{u}(\Phi), \Phi) \tag{2.14}$$

is defined, where λ is the vector[1] of Lagrange multipliers, which will be used for elimination of implicit sensitivity terms. $\tilde{F} = F$ because $\mathbf{R} = 0$. Differentiation of (2.14) with respect to a specific design parameter ϕ yields

$$\frac{d\tilde{F}}{d\phi} = \frac{\partial G}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\phi} + \frac{\partial G}{\partial \phi} - \left(\frac{d\lambda}{d\phi} \right)^T \mathbf{R} - \lambda^T \left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\phi} + \frac{\partial \mathbf{R}}{d\phi} \right). \tag{2.15}$$

[1] Vectors denoted by Greek letters are not typed in bold, but it should be clear from the context when some quantity is a vector and when scalar.

Since $\mathbf{R}=0$ by (2.12) and $\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\frac{d\mathbf{u}}{d\phi}+\frac{\partial \mathbf{R}}{d\phi}=0$ by (2.13),

$$\frac{dF}{d\phi}=\frac{d\tilde{F}}{d\phi}. \tag{2.16}$$

The terms in (2.14) which include implicit derivatives are

$$\frac{\partial G}{\partial \mathbf{u}}\frac{d\mathbf{u}}{d\phi}-\lambda^T\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\frac{d\mathbf{u}}{d\phi}=\left(\frac{\partial G}{\partial \mathbf{u}}-\lambda^T\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)\frac{d\mathbf{u}}{d\phi} \tag{2.17}$$

These terms are eliminated from (2.15) by defining λ so that the term in round brackets in (2.17) is zero. This is achieved if λ solves the system

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)^T\lambda=\left(\frac{\partial G}{\partial \mathbf{u}}\right)^T. \tag{2.18}$$

System (2.18) is referred to as the adjoint problem for the adjoint response λ with the adjoint load $\left(\partial G/\partial \mathbf{u}\right)^T$. Once multipliers λ are evaluated, the derivative of F with respect to a specific parameter ϕ is obtained as

$$\frac{dF}{d\phi}=\frac{d\tilde{F}}{d\phi}=\frac{\partial G}{\partial\phi}-\lambda^T\frac{\partial \mathbf{R}}{\partial\phi}. \tag{2.19}$$

The adjoint method requires the solution of the adjoint problem (2.18) for each response functional F. It is efficient when the number of response functionals is small compared to the number of design parameters.

A similar approach can be adopted for *transient problems* where sensitivities are evaluated within the incremental solution scheme. As for steady state problems the dependence on the design parameter is taken into account in the discretised governing equations (2.4):

$${}^{(n)}\mathbf{R}\left({}^{(n)}\mathbf{u}(\phi),{}^{(n-1)}\mathbf{u}(\phi),\phi\right)=\mathbf{0}. \tag{2.20}$$

It will be assumed that the response functional is defined through the response for the final time ${}^{(M)}t$, although it can be easily defined as a function of response for intermediate times[10],[12]:

$$F(\Phi)=G\left({}^{(M)}\mathbf{u}(\Phi),\Phi\right). \tag{2.21}$$

Derivation with respect to the parameter ϕ yields

$$\frac{dF}{d\phi} = \frac{dG}{d^{(M)}\mathbf{u}} \frac{D^{(M)}\mathbf{u}}{d\phi} + \frac{\partial G}{\partial \phi}. \tag{2.22}$$

In the *direct differentiation method* the implicit derivative is obtained directly by derivation of (2.20), which yields (after setting the increment index to *M*)

$$\frac{\partial^{(M)}\mathbf{R}}{\partial^{(M)}\mathbf{u}} \frac{d^{(M)}\mathbf{u}}{d\phi} = -\left(\frac{\partial^{(M)}\mathbf{R}}{\partial^{(M-1)}\mathbf{u}} \frac{d^{(M-1)}\mathbf{u}}{d\phi} + \frac{\partial^{(M)}\mathbf{R}}{\partial \phi_i} \right) \tag{2.23}$$

The pseudoload on the above equation contains the sensitivity of the response evaluated in the previous step. By applying the direct differentiation procedure back in time we see that the system

$$\frac{\partial^{(n)}\mathbf{R}}{\partial^{(n)}\mathbf{u}} \frac{d^{(n)}\mathbf{u}}{d\phi} = -\left(\frac{\partial^{(n)}\mathbf{R}}{\partial^{(n-1)}\mathbf{u}} \frac{d^{(n-1)}\mathbf{u}}{d\phi} + \frac{\partial^{(n)}\mathbf{R}}{\partial \phi_i} \right) \tag{2.24}$$

must be solved for $d^{(i)}\mathbf{u} / d\phi$ after each time step (i.e. for i=1, 2, …, M) after convergence of the iteration (2.5) and (2.6), while the derivative of the initial condition $d^{(0)}\mathbf{u} / d\phi$ needed after the first increment is assumed to be known.

In the *adjoint method* the implicit terms are again eliminated by the appropriate definition of the Lagrange multipliers. The augmented functional is defined by combination of (2.21) and (2.20) for all increments:

$$F(\Phi) = G\left({}^{(M)}\mathbf{u}(\Phi), \Phi\right) - \sum_{n=1}^{M} {}^{(n)}\lambda(\Phi)^{T\,(n)}\mathbf{R}\left({}^{(n)}\mathbf{u}(\Phi), {}^{(n-1)}\mathbf{u}(\Phi), \Phi\right) \tag{2.25}$$

Again $F = \tilde{F}$ follows from (2.20) and $\frac{dF}{d\phi} = \frac{d\tilde{F}}{d\phi}$ follows from (2.20) and (2.24). Derivation of (2.25) yields after rearrangement and some manipulation

$$\begin{aligned}
&\frac{dF}{d\phi}=\frac{d\tilde{F}}{d\phi}=\frac{\partial G}{\partial \phi}-\sum_{n=1}^{M}{}^{(n)}\lambda^T\frac{\partial^{(n)}\mathbf{R}}{\partial \phi}-{}^{(1)}\lambda^T\frac{d^{(1)}\mathbf{R}}{d^{(0)}\mathbf{u}}\frac{d^{(0)}\mathbf{u}}{d\phi}-\\
&-\sum_{n=1}^{M-1}{}^{(n)}\lambda^T\frac{\partial^{(n)}\mathbf{R}}{\partial^{(n)}\mathbf{u}}\frac{d^{(n)}\mathbf{u}}{d\phi}+{}^{(n+1)}\lambda^T\frac{\partial^{(n+1)}\mathbf{R}}{\partial^{(n)}\mathbf{u}}\frac{d^{(n)}\mathbf{u}}{d\phi}-\\
&-{}^{(M)}\lambda^T\frac{\partial^{(M)}\mathbf{R}}{\partial^{(M)}\mathbf{u}}\frac{d^{(M)}\mathbf{u}}{d\phi}+\left(\frac{\partial G}{\partial^{(M)}\mathbf{u}}\right)^T\frac{d^{(M)}\mathbf{u}}{d\phi}
\end{aligned}\quad , \tag{2.26}$$

where the first line contains explicit terms and the other two lines contain implicit terms which must be eliminated.

Elimination of implicit terms from (2.26) is achieved by solution of the following set of adjoint problems for the Lagrange multiplier vectors:

$$\begin{aligned}
&\left(\frac{\partial^{(M)}\mathbf{R}}{\partial^{(M)}\mathbf{u}}\right)^T {}^{(M)}\lambda=\frac{\partial G}{\partial^{(M)}\mathbf{u}},\\
&\left(\frac{\partial^{(n)}\mathbf{R}}{\partial^{(n)}\mathbf{u}}\right)^T {}^{(n)}\lambda=-\left(\frac{\partial^{(n+1)}\mathbf{R}}{\partial^{(n)}\mathbf{u}}\right)^T {}^{(n+1)}\lambda,\ n=M-1,M-2,\ldots,M-1
\end{aligned}\quad . \tag{2.27}$$

Once this is done, the functional derivative is obtained from

$$\frac{dF}{d\phi}=\frac{d\tilde{F}}{d\phi}=\frac{\partial G}{\partial \phi}-\sum_{n=1}^{M}{}^{(n)}\lambda^T\frac{\partial^{(n)}\mathbf{R}}{\partial \phi}-{}^{(1)}\lambda^T\frac{d^{(1)}\mathbf{R}}{d^{(0)}\mathbf{u}}\frac{d^{(0)}\mathbf{u}}{d\phi} \tag{2.28}$$

Since the equations (2.27) are evaluated in the reverse order to the tangent operators, the complete problem must be solved before the sensitivity analysis can begin. This requires storage of converged (and possibly decomposed) tangent operators from all increments. The adjoint analysis may still be preferred when the number of the design parameters is significantly larger than the number of response functionals.

A similar derivation can be performed for coupled systems (i.e. equations (2.5) and (2. 6)). The procedure is outlined e.g. in [12], and [16]. In the direct method sensitivity of one field is expressed in terms of the sensitivity of another, which gives the dependent and the independent pseudoproblem. In the adjoint methods, two sets of Lagrange multipliers must be introduced, one for each corresponding equation. Two adjoint problems are solved for each set of multipliers for each increment, otherwise the procedure is the same as for non-coupled problems. Sensitivity analysis for various finite element formulations in metal forming is reviewed in [15] and [16] and discussed in detail in [10].

Sensitivity analysis significantly increases the complexity of the simulation code. One complication comes at the global level where the assembled problem is solved in the incremental/iterative scheme. Solution of the adjoint or pseudoproblems must be included in the scheme, which includes assembling of pseudoloads from element terms. This is followed by appropriate substitutions in order to evaluate the complete sensitivities. An additional complication in the adjoint method is that the converged tangent operators must be stored for increments, since solution of the adjoint problems is reversed in time. In this level the additional complexity can be relatively easily kept under control if the programme structure is sufficiently flexible. The number of necessary updates in the code which is primarily aimed for solution of the direct problem is small and the additional complexity in the programme flow chart is comparable to the complexity of the original flow chart.

A more serious problem is the complexity which arises on the element level, where element terms of the pseudoloads are evaluated, i.e. derivatives of the residual with respect to design parameters. The code should be able to evaluate the pseudoload for any parametrisation that might be used, which can include shape, material, load parameters, etc. Implementation of a general solution code which could provide response sensitivities for any possible set of parameters turns out to be a difficult task. It must be taken into account that such a code must include different complex material models and finite element formulations and that derivation of the process of evaluation of element residual terms with respect to any of the possible parameters can be itself a tedious task. Another complication which should not be overlooked is the evaluation of the terms $\partial G / \partial \mathbf{u}$. Although these are regarded as explicit terms, for complex functionals their evaluation is closely related to the numerical model and can include spatial and time integration and derivation of quantities dependent upon history parameters, with respect to the response **u**.

The reasons outlined above make use of symbolic systems for automatic generation of element level code[17]-[20] (Figure 1.1) highly desirable. In the case of sensitivity analysis use of such systems enables implementation of new finite element formulations and physical models in times drastically shorter then would be needed for manual development. Additionally, use of these systems enables definition of functionals which are used in optimisation and the necessary sensitivity terms on abstract mathematical level where the basic formulation of the numerical model is defined. These definitions can be readily adjusted to new types of problems, because the necessary derivations are performed by the symbolic systems and the appropriate computer code is generated automatically. The system for automatic code generation is connected with a flexible solution environment framework (referred to as the finite element driver[21]) into which the generated code can be readily incorporated. The complexity of inherently combinatorial nature, which would arise in a static simulation code applicable for sensitivity analysis in general problems, can be avoided in this way.

References:

[1] O. C. Zienkiewicz, R. Taylor, *The Finite Element Method*, Vol. 1, 2 (fourth edition), McGraw-Hill, London, 1991.

[2] K.J. Bathe, *Finite Element Procedures,* p.p. 697-745, Prentice Hall, New Jersey, 1996.

[3] J. Bonet, R.D Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis*, Cambridge University Press, Cambridge, 1997.

[4] M. A. Crisfield, *Non-Linear Finite Element Analysis of Solids and Structures,* Vol. 1, John Wiley & Sons, Chichester, 1991.

[5] M. A. Crisfield, *Non-Linear Finite Element Analysis of Solids and Structures,* Vol. 2, John Wiley & Sons, Chichester, 1997.

[6] D. R. J. Owen, E. Hinton, *An Introduction to Finite Element Computations*, Pineridge Press, Swansea, 1979.

[7] D. R. J. Owen, E. Hinton, *Finite Elements in Plasticity*, Pineridge Press, Swansea, 1980.

[8] N. Petrinić, *Aspects of Discrete Element Modelling Involving Facet-to-Facet Contact Detection and Interaction*, Ph. D. thesis, University of Wales, Swansea, 1996.

[9] T. Rodič, *Numerical Analysis of Thermomechanical Processes During Deformation of Metals at High Temperatures*, Ph. D. thesis, University of Wales, Swansea, 1989.

[10] M. Kleiber, H. Antunez, T. D. Hien, P. Kowalczyk, *Parameter Sensitivity in Nonlinear Mechanics – Theory and Finite Element Computations*, John Wiley & Sons, Chichester, 1997.

[11] K. Dems, Z. Mroz, *Variational Approach to First and Second Order Sensitivity Analysis of Elastic Structures,* International Journal for Numerical Methods in Engineering, Vol. 21, p.p. 637-661, John Wiley & Sons, 1984.

[12] P. Michaleris, D. A. Tortorelli, C. A Vidal, *Tangent Operators and Design Sensitivity Formulations for Transient Non-Linear Coupled Problems with Applications to Elastoplasticity,* Int. Jour. For Numerical Methods in Engineering, vol. 37, pp. 2471-2499, John Wiley & Sons, 1994.

[13] D. A. Tortorelli, *Non-linear and Time Dependent Structural Systems: Sensitivity Analysis and Optimisation*, Material for the DCAMM

course (held in Lingby, Denmark), Technical University of Denmark, 1997.

[14] D. A. Tortorelli, P. Michaleris, *Design Sensitivity Analysis: Overview and Review*, Inverse Problems in Engineering, vol. 1, pp. 71-105, Overseas Publishers Association, 1994.

[15] S. Badrinarayanan, N. Zabaras, *A Sensitivity Analysis for the Optimal Design of Metal Forming Processes*, Comput. Methods Appl. Mech. Engrg. 129, pp. 319-348, Elsevier, 1996.

[16] I. Doltsinis, T. Rodič, *Process Design and Sensitivity Analysis in Metal Forming*, Int. J. Numer. Meth. Engng., vol. 45, p.p. 661-692, John Wiley & Sons, 1999.

[17] J. Korelc, *Automatic generation of finite-element code by simultaneous optimalization of expressions*, Theoretical Computer Science, 187, p.p. 231-248, 1997.

[18] J.Korelc, P. Wriggers, *Symbolic approach in computational mechanics*, in Computational Plasticity Fundamentals and Applications (ed. D.R.J. Owen, E. Onate and E. Hinton), pp. 286-304, CIMNE, Barcelona, 1997.

[19] J. Korelc, *Symbolic Approach in Computational Mechanics and its Application to the Enhanced Strain Method*, Ph. D. thesis, Technische Hochschule Darmstadt, 1996.

[20] T. Rodič, I. Grešovnik, D. Jelovšek, J. Korelc, *Optimisation of Prestressing of a Cold Forging Die by Using Symbolic Templates*, in ECCM '99 : European Conference on Computational Mechanics (held in Munich, Germany), pp. 362-372, Technische Universität München, 1999.

[21] T. Šuštar, *FEM Driver*, electronic document at http://www.c3m.si/driver/ , Ljubljana, 2000.

[22] J. Lubliner, *Plasticity Theory*, Macmillan Publishing Company, New York, 1990.