Parameter optimization for a multiaxial stress-strain correction scheme

Holger Lang[1], Klaus Dressler[2], Rene Pinnau[3]

[1, 2] Fraunhofer Institut für Techno- und Wirtschaftsmathematik,
Fraunhofer Platz 1, 67663 Kaiserslautern, Germany
holger.lang@itwm.fraunhofer.de, *klaus.dressler@itwm.fraunhofer.de*

[3] Technische Universität Kaiserslautern,
Erwin Schrödinger Strasse, Geb. 48, 67663 Kaiserslautern, Germany
pinnau@mathematik.uni-kaiserslautern.de

Abstract

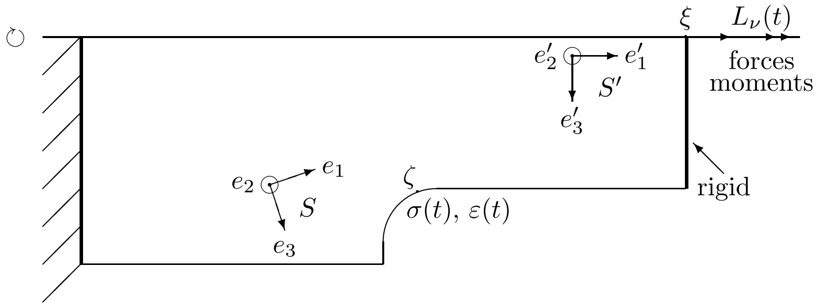
A gradient based algorithm for parameter identification (least-squares) is applied to a multiaxial correction method for elastic stresses and strains at notches. The correction scheme, which is numerically cheap, is based on Jiang's model of elastoplasticity. Both mathematical stress-strain analyses (nonlinear finite element method with Jiang's model of elastoplasticity) and physical strain measurements have been approximized. The gradient evaluation with respect to the parameters, which is very large in scale, is realized by the automatic forward differentiation technique.

Keywords. Jiang's model of elastoplasticity, multiaxial high cycle fatigue analysis, durability, parameter identification, automatic differentiation, AD, sensitivities

MSC classification: 74C15, 74D10, 65Z05

1 Introduction

We consider a specimen in form of an axle, which exhibits rotational symmetry around the global $1'$-direction (i.e. in coordinate system S'). It is made of S460N steel, which shows nonlinear hysteresis phenomena as ratchetting. An appropriate contitutive material law is therefore given by the model of Jiang/Sehitoglu [9, 10, 11, 12].



(1)

Loads (normal force/torsional moment, scalar values $L_\nu(t)$) are applied at the point ξ, along the global $1'$-direction. All stresses and strains at ζ, appearing in the sequel, are measured in the local ζ-coordinate-system S, whose 3-direction is normal to the tangential plane at ζ. Consequently, all stress tensors are plane, more precisely: The 13-, 23- and 33-components of the stresses at ζ are equal to zero.

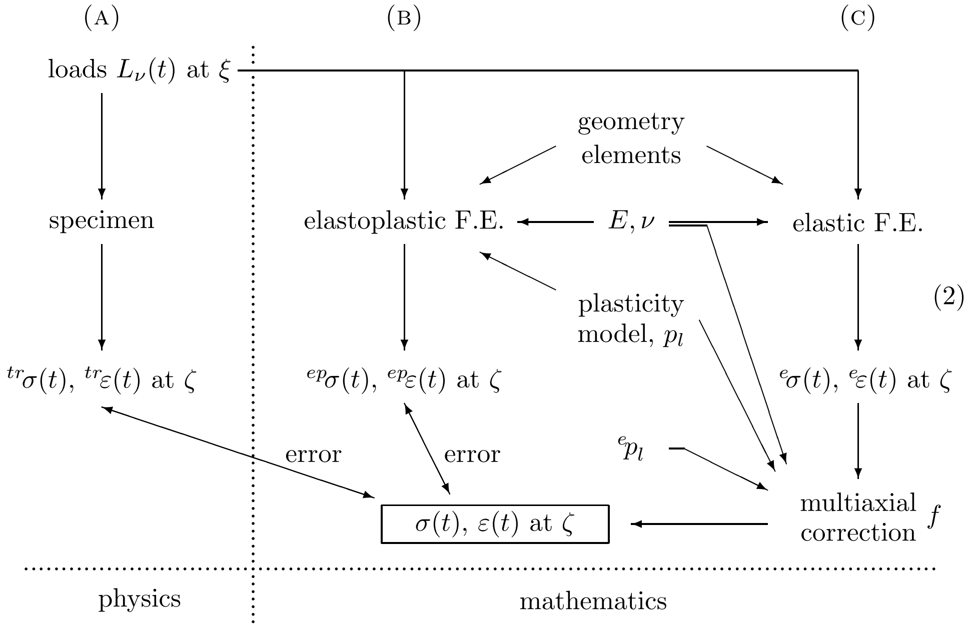
We consider three ways to get the stresses and strains at the notch ζ, namely

(A) *Measurements.* You get the true stresses ${}^{tr}\sigma(t)$ and strains ${}^{tr}\varepsilon(t)$ or at least some components (normal/shear) of them. Let us assume throughout this paper, that there are no errors in measurements.

(B) *Nonlinear quasi-static elastoplastic finite element analysis.* You get the elastplastic stresses ${}^{ep}\sigma(t)$ and strains ${}^{ep}\varepsilon(t)$ – full tensors – under the assumption of the existence of an appropriate parameter set p_l for Jiang's model. ($\rightsquigarrow$ numerically expensive)

(C) *Linear quasi-static elastic finite element analysis + subsequent correction* In a first step, you get elastic stresses ${}^{e}\sigma(t)$ and strains ${}^{e}\varepsilon(t)$ – full tensors – which have to be fed into a correction scheme f to get better stresses $\sigma(t)$ and strains $\varepsilon(t)$. In addition to the p_l, a set of notch parameters ${}^{e}p_l$ for ζ has to be determined. ($\rightsquigarrow$ numerically cheap)

We want to focus here on the third way (C) to approximate the targets (A) resp. (B). The diagram (2), giving input/output relations, illustrates these three ways.


(A)
(B)
(C)
loads $L_\nu(t)$ at ξ
geometry elements
specimen
elastoplastic F.E.
E, ν
elastic F.E.
plasticity model, p_l
(2)
${}^{tr}\sigma(t)$, ${}^{tr}\varepsilon(t)$ at ζ
${}^{ep}\sigma(t)$, ${}^{ep}\varepsilon(t)$ at ζ
${}^{e}\sigma(t)$, ${}^{e}\varepsilon(t)$ at ζ
error
error
${}^{e}p_l$
multiaxial correction f
$\sigma(t)$, $\varepsilon(t)$ at ζ
physics
mathematics

In this paper, we want to focus on the correction scheme f from LANG, DRESSLER, PINNAU [14] (where the plane stress situation at ζ is referred to as the 'notch' case) and HERTEL [6]. For detailed mathematical formulation and an appropriate algorithm for f see [14].

Summary. The correction idea is based on a simultaneity assumption. It states that the movements in the 'elastic' (fictive, 'pseudo') stress space, controlled by ${}^{e}\sigma(t)$, follow the movements in the 'real' stress in order to define $\sigma(t)$. The difference tensors

$$ {}^{e}\sigma(t) - (\text{center } {}^{e}\alpha(t) \text{ of the 'elastic' yield surface}) \tag{3} $$

and

$$ \sigma(t) - (\text{center } \alpha(t) \text{ of the 'real' yield surface}) \tag{4} $$

are defined parallel at each point in time. Through a rescaling by the ratio of the yield surfaces radii, it is garuanteed that ${}^{e}\sigma(t)$ and $\sigma(t)$ get into contact/lose contact with their respective yield surfaces at exactly the same point of time. (As a consequence, the normals to the corresponding yield surfaces $n(t)$ and ${}^{e}n(t)$ are always equal and active plastic yielding starts/stops at exactly the same time.)
It makes sence (but is not necessary from the simultaneity assumption), that in f the same constitutive model as in (B) is used, i.e. Jiang's model with $m = 5$ backstresses in our case here. So, each constitutive parameters p_l, determinig the

movements in the ‘real’ stress space, has exactly one ‘elastic’ counterpart ${}^e\!p_l$ in order to determine the movements in the ‘elastic’ stress space.

$$\begin{tikzcd}
 & {}^e\!p_l \arrow[d] & & p_l \arrow[d] & \\
{}^e\sigma(t) \arrow[r] & \text{‘elastic’ stress space} \arrow[d] \arrow[rr, rightsquigarrow] & & \text{‘real’ stress space} \arrow[r] & \sigma(t) \arrow[d] \\
 & \varepsilon^{pl}(t) \arrow[dr] & & & \varepsilon^{el}(t) = C^{-1}\,\sigma(t) \arrow[dll] \\
 & & \varepsilon(t) = \varepsilon^{pl}(t) + \varepsilon^{el}(t) & &
\end{tikzcd} \tag{5}$$

Let us combine the parameters in two vectors

$$p = (p_1, \dots, p_L), \qquad {}^e\!p = ({}^e\!p_1, \dots, {}^e\!p_L)$$

where

${}^e\!p_1, \dots, {}^e\!p_6$	$\leftrightsquigarrow$	${}^e c_R, {}^e\rho_0, {}^e a_\rho, {}^e b_\rho, {}^e a_\chi, {}^e b_\chi$	$p_1, \dots p_6$	$\leftrightsquigarrow$	$c_R, \rho_0, a_\rho, b_\rho, a_\chi, b_\chi$
${}^e\!p_7, \dots, {}^e\!p_{11}$	$\leftrightsquigarrow$	${}^e c_1^\infty, \dots, {}^e c_5^\infty$	$p_7, \dots, p_{11}$	$\leftrightsquigarrow$	$c_1^\infty, \dots, c_5^\infty$
${}^e\!p_{12}, \dots, {}^e\!p_{16}$	$\leftrightsquigarrow$	${}^e r_1, \dots, {}^e r_5$	$p_{12}, \dots, p_{16}$	$\leftrightsquigarrow$	$r_1, \dots, r_5$
${}^e\!p_{17}, \dots, {}^e\!p_{21}$	$\leftrightsquigarrow$	${}^e Q_1, \dots, {}^e Q_5$	$p_{17}, \dots, p_{21}$	$\leftrightsquigarrow$	$Q_1, \dots, Q_5$
${}^e\!p_{22}, \dots, {}^e\!p_{26}$	$\leftrightsquigarrow$	${}^e a_1^{(1)}, \dots, {}^e a_5^{(1)}$	$p_{22}, \dots, p_{26}$	$\leftrightsquigarrow$	$a_1^{(1)}, \dots, a_5^{(1)}$
${}^e\!p_{27}, \dots, {}^e\!p_{31}$	$\leftrightsquigarrow$	${}^e a_1^{(2)}, \dots, {}^e a_5^{(2)}$	$p_{27}, \dots, p_{31}$	$\leftrightsquigarrow$	$a_1^{(2)}, \dots, a_5^{(2)}$
${}^e\!p_{33}, \dots, {}^e\!p_{36}$	$\leftrightsquigarrow$	${}^e b_1^{(1)}, \dots, {}^e b_5^{(1)}$	$p_{33}, \dots, p_{36}$	$\leftrightsquigarrow$	$b_1^{(1)}, \dots, b_5^{(1)}$
${}^e\!p_{37}, \dots, {}^e\!p_{41}$	$\leftrightsquigarrow$	${}^e b_1^{(2)}, \dots, {}^e b_5^{(2)}$	$p_{37}, \dots, p_{41}$	$\leftrightsquigarrow$	$b_1^{(2)}, \dots, b_5^{(2)}$

$L = 7m + 6$ is the number of Jiang parameters, here $L = 41$ as $m = 5$. Young's modulus E and Poisson' ratio ν are considered as given and remain unchanged.

The elastic stresses ${}^e\sigma(t)$ and strains ${}^e\varepsilon(t)$ are linear superposition of finitely many static linear results ${}^e\sigma(t) = \sum_\nu L_\nu(t)\,{}^e\sigma_\nu$, ${}^e\varepsilon(t) = \sum_\nu L_\nu(t)\,{}^e\varepsilon_\nu$ (typically ‘unit load cases’). They are linearly coupled via Hook's law ${}^e\sigma(t) = C\,{}^e\varepsilon(t)$. Consequently, it is sufficient to have just ${}^e\sigma(t)$ as input for f. Let us denote the values of the tensors at the time points $t_0, \dots, t_N$ of consideration by

$${}^e\sigma_n = {}^e\sigma(t_n), \qquad \sigma_n = \sigma(t_n), \qquad \varepsilon_n = \varepsilon(t_n), \qquad n = 0, \dots, N$$

and

$${}^e\sigma = ({}^e\sigma_0, \dots, {}^e\sigma_N), \qquad \sigma = (\sigma_0, \dots, \sigma_N), \qquad \varepsilon = (\varepsilon_0, \dots, \varepsilon_N).$$

The correction function f can be written as

$$(\sigma_0, \dots, \sigma_N,\, \varepsilon_0, \dots, \varepsilon_N) = f({}^e\sigma_0, \dots, {}^e\sigma_N,\, p_1, \dots, p_L,\, {}^e\!p_1, \dots, {}^e\!p_L) \tag{6}$$

or briefly

$$(\sigma,\, \varepsilon) = f({}^e\sigma,\, p,\, {}^e\!p). \tag{7}$$

For parameter identification we have in principal two possibilites to be the targets

$$\tilde\sigma_n = \tilde\sigma(t_n), \qquad \tilde\varepsilon_n = \tilde\varepsilon(t_n), \qquad n = 0, \dots, N$$

depending on what we want to approximize.

(A) We target at the physical results

$$\tilde\sigma(t) = {}^{tr}\sigma(t), \qquad \tilde\varepsilon(t) = {}^{tr}\varepsilon(t) \tag{8}$$

Here you solely have those components (e.g. normal/shear) that are measurable in practice.

(B) We target at the mathematical results

$$\tilde{\sigma}(t) = {}^{ep}\sigma(t), \qquad \tilde{\varepsilon}(t) = {}^{ep}\varepsilon(t) \tag{9}$$

Here you usually have the full 3 × 3-tensor information for the targets at disposal.

For each ζ a full appropriate set of parameters has to be determined. We distinguish two possibilities

(i) We iterate the ${}^{e}p_l$ and consider the p_l as physically true and unchangeable. That's the way the correction model originally was designed for.

(ii) We iterate *both* the ${}^{e}p_l$ *and* the p_l. Additional (double) freedom in the movements in the both stress spaces [13, 14] has better chances to reflect reality. The additional costs is low (as we will see below).

Former work.

(a) The correction idea, introducing a set of 'pseudo' parameters ${}^{e}p_l$ at the notch ζ, was born by KÖTTGEN et al. [13]. They used Mroz's model of elastoplasticity. Our correction method is called there the '${}^{e}\sigma$-approach'.

(b) The *material.* For a set of Jiang material parameters p_l of S460N with 5 backstresses, we refer to HOFFMEYER et al. [7].

(c) The *axle.* The geometry of the axle (1) and the correspondig FE model is descripted in detail in HERTEL [6], sect. 4.1, 4.2. A set of notch parameters ${}^{e}p_l$ at ζ has been determined as well in sect. 4.4. ■

The total parameter identification procedure consists of two steps

(1) determining good initial values for the parameters (see [6], sect. 4.4, derivation from load-notch strain curves)

(2) mathematical optimization ('fine tuning') of parameters (especially the doubtful ones)

In this paper, we are solely concerned with step (2). Without good initial values out of step (1), a mathematical minimization method alone would find any local minimum, which is far too large compared to the global minimum. As we have a huge amount of parameters, there will be very many local minima. ■

The big advantage of our kind of correction approach compared to multiaxial differential Neuber approaches like [1, 4, 15] lies in the fact, that the latter do not comprise 'elastic' parameters ${}^{e}p_l$ which may serve as additional tuning devices.

2 The parameter optimization procedure

Let $\tilde{\sigma} = (\tilde{\sigma}_0, \ldots, \tilde{\sigma}_N)$ and $\tilde{\varepsilon} = (\tilde{\varepsilon}_0, \ldots, \tilde{\varepsilon}_N)$ denote the *targets*, which we want to approximate optimally, i.e. (8) or (9). The actuals corresponding to the actual parameters set (p_l, ${}^{e}p_l$) are given by the evaluation of the correction scheme with the constant input ${}^{e}\sigma$ (i.e. not depending on p_l and ${}^{e}p_l$) and the parameters

$$\big(\sigma(p, {}^{e}p),\, \varepsilon(p, {}^{e}p)\big) = f\big({}^{e}\sigma,\, p,\, {}^{e}p\big) \tag{10}$$

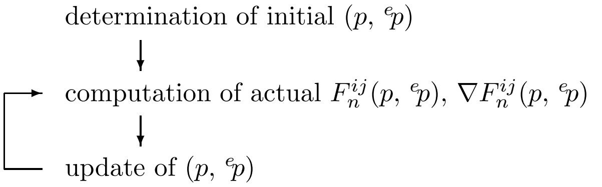
For the objective function F, giving the error between actuals and targets, and which has to be minimized, we choose the weighted least squares residue

$$F(p, {}^{e}p) = \sum_{n=0,\ldots,N} \; \sum_{i,j=1,2,3} F_n^{ij}(p, {}^{e}p) \tag{11}$$

where the summands are given by

$$F_n^{ij}(p, {}^{e}\!p) = \frac{w_\sigma^{ij}}{2}\Big(\sigma_n^{ij}(p, {}^{e}\!p) - \tilde{\sigma}_n^{ij}\Big)^2 + \frac{w_\varepsilon^{ij}}{2}\Big(\varepsilon_n^{ij}(p, {}^{e}\!p) - \tilde{\varepsilon}_n^{ij}\Big)^2 \tag{12}$$

and positive weights w_σ^{ij}, $w_\varepsilon^{ij} > 0$, $i, j = 1, 2, 3$, not depending on p_l and ${}^{e}\!p_l$. For the minimization of F, we have used an iterative gradient based optimization method:



determination of initial $(p,\ {}^{e}\!p)$

computation of actual $F_n^{ij}(p,\ {}^{e}\!p)$, $\nabla F_n^{ij}(p,\ {}^{e}\!p)$ (13)

update of $(p,\ {}^{e}\!p)$

For the initial step in (13), we refer to [6], sect. 4.4. For the choice of the update in (13), i.e. choice of descent direction/length, MATLAB matlab provides

- Gauss-Newton method
- Levenberg-Marquardt method
- Coleman-Li trust region methods
- ...

which are all gradient based methods and where the gradients may be provided by the user. We chose a Coleman-Li trust-region algorithm for the iteration step. It is especially designed for large numbers of parameters. The gradient

$$\nabla = \nabla_{p,{}^{e}\!p} = \Big(\frac{\partial}{\partial p_1}, \ldots, \frac{\partial}{\partial p_L}, \frac{\partial}{\partial {}^{e}\!p_1}, \ldots, \frac{\partial}{\partial {}^{e}\!p_L}\Big) \quad \text{or} \quad \nabla = \nabla_{{}^{e}\!p} = \Big(\frac{\partial}{\partial {}^{e}\!p_1}, \ldots, \frac{\partial}{\partial {}^{e}\!p_L}\Big)$$

with respect to the parameters ${}^{e}\!p_l$ (case (i)) resp. $(p_l,\ {}^{e}\!p_l)$ (case (ii)) of each summand F_n^{ij} in (11) is computated from the gradients of σ_n^{ij} and ε_n^{ij} via the chain rule

$$\nabla F_n^{ij}(p, {}^{e}\!p) = w_\sigma^{ij}\Big(\sigma_n^{ij}(p, {}^{e}\!p) - \tilde{\sigma}_n^{ij}\Big)\nabla\sigma_n^{ij}(p, {}^{e}\!p) + w_\varepsilon^{ij}\Big(\varepsilon_n^{ij}(p, {}^{e}\!p) - \tilde{\varepsilon}_n^{ij}\Big)\nabla\varepsilon_n^{ij}(p, {}^{e}\!p)$$

Then obviously $\nabla F = \sum_n \sum_{ij} \nabla F_n^{ij}$. For the evaluation of the gradients

$$(\nabla\sigma,\ \nabla\varepsilon) = \nabla f$$

(each time, the correction model (10) is evaluated), we have in principal two elemental possibilities

- FD, **finite differences** (based on the definition of the partial derivatives)

$$\frac{\partial f}{\partial p_l}({}^{e}\!\sigma,\ p,\ {}^{e}\!p) \ = \ \frac{1}{\delta p_l}\Big(f({}^{e}\!\sigma,\ p + \delta p_l\, e_l,\ {}^{e}\!p) - f({}^{e}\!\sigma,\ p,\ {}^{e}\!p)\Big) + \mathcal{O}\big(\delta p_l^2\big) \tag{14}$$

$$\frac{\partial f}{\partial {}^{e}\!p_l}({}^{e}\!\sigma,\ p,\ {}^{e}\!p) \ = \ \frac{1}{\delta {}^{e}\!p_l}\Big(f({}^{e}\!\sigma,\ p,\ {}^{e}\!p + \delta {}^{e}\!p_l\, e_l) - f({}^{e}\!\sigma,\ p,\ {}^{e}\!p)\Big) + \mathcal{O}\big(\delta {}^{e}\!p_l^2\big) \tag{15}$$

  (e_l is the lth unit vector of length L and $|\delta {}^{e}\!p_l|, |\delta p_l| \ll 1$)

- AD, **automatic differentiation** (based on the multidimensional chain rule)

As f includes the integration of discontinuous nonlinear differential equations, it is very komplex: Simple computation of the derivatives ('by hand') or adjoining the differential equations is very elaborate and prone to errors.

The weights. We have the freedom to choose the weights in (12) in order to switch on/off the components that are needed. If both ε and σ have to be targeted at once, the w_ε^{ij} have to be much larger than the w_σ^{ij}. To this end, we recommend $w_\varepsilon^{ij} \approx E^2 w_\sigma^{ij}$. Note, that E does belong neither to the p_l nor to the ${}^{e}\!p_l$. ■

3 Automatic differentiation

The reader finds an appropriate algorithm for the evaluation of f – i.e. scheme (6), (7) – in [14], sect. 3. Independent of which algorithm you consider, its code internally will be looking like

$$\begin{array}{rcll}
x_i & := & {}^{\mathrm{e}}\sigma_i & (i = 0, \ldots, N) \qquad \leftarrow \text{IN} \\
x_i & := & {}^{\mathrm{e}}p_{i-N} & (i = N+1, \ldots, N+L) \qquad \leftarrow \text{IN} \\
x_i & := & p_{i-N-L} & (i = N+L+1, \ldots, N+2L) \;\leftarrow \text{IN} \\
x_i & := & \varphi_i\big((x_j)_{j \preceq i}\big) & (i = N+2L+1, \ldots, M) \\
\text{OUT} \leftarrow \; (\sigma_0, \ldots, \sigma_N) & := & (x_{j_0}, \ldots, x_{j_N}) & (0 \leq j_0, \ldots, j_N \leq M) \\
\text{OUT} \leftarrow \; (\varepsilon_0, \ldots, \varepsilon_N) & := & (x_{i_0}, \ldots, x_{i_N}) & (0 \leq i_0, \ldots, i_N \leq M)
\end{array} \tag{16}$$

where all variables appearing in the code are numbered by $x_0, \ldots, x_M$, corresponding to ascending occurrence, altogether $M+1$ ones. They are allowed to be any *single/double*-array of size $n_{i,1} \times \ldots \times n_{i,r_i}$, that is

- scalars: $r_i = 1$, $n_{i,1} = 1$ (e.g. the p_l and ${}^{\mathrm{e}}p_l$)
- vectors: $r_i = 1$, $n_{i,1} \geq 2$ (e.g. the σ_n, ε_n, ${}^{\mathrm{e}}\sigma_n$ in vector-notation)
- matrices: $r_i = 2$, $n_{i,1}, n_{i,2} \geq 1$ (e.g. the σ_n, ε_n, ${}^{\mathrm{e}}\sigma_n$ in matrix-notation)

or

- tensors of any higher order.

The first three lines in (16) is just the initialisation of the input, the last two lines the selection of the output. All the functions φ_i in the fourth line are elemental operations, i.e.

$$+,\ -,\ \cdot,\ /,\ \text{inv},\ \exp,\ \ln,\ \log,\ \sin,\ \cos,\ \tan,\ \text{abs},\ \max,\ \min,\ \sinh,\ \text{Artanh},\ \ldots$$

whose derivatives are known. The symbol $j \preceq i$ indicated all the variables x_j, on which the variable x_i depends (and which have consequently appeared in the code already before). Of course, this may include the variable x_i itself (e.g. $x_i := x_i + 1$, or $x_i := 2x_i$).

Choice of AD mode. The number of output variables $2N$ (usually several thousands) is much larger than the number of parameters $L = 41$ in case (ii) resp. $2L = 82$ in case (i). That's why the AD *forward* mode is to be preferred to the *backward* mode for the sakes of speed and memory. For details/theory and further explanation, the reader is referred to the book of Griewank [5]. ■

Automatic forward differentiation now embeds this code for f ('original' code) into a code for f *and* ∇f ('extended' code). It additionally contains the lines for the derivatives according to the chain rule (CR)

$$\begin{array}{rcll}
\dfrac{\partial x_i}{\partial\, {}^{\mathrm{e}}p_l} & := & \dfrac{\partial\, {}^{\mathrm{e}}\sigma_i}{\partial\, {}^{\mathrm{e}}p_l} = 0 & (i = \ldots) \\
\dfrac{\partial x_i}{\partial\, {}^{\mathrm{e}}p_l} & := & \dfrac{\partial\, {}^{\mathrm{e}}p_{i-N}}{\partial\, {}^{\mathrm{e}}p_l} = \delta_{l,(i-N)} & (i = \ldots) \\
\dfrac{\partial x_i}{\partial\, {}^{\mathrm{e}}p_l} & := & \dfrac{\partial p_{i-N-L}}{\partial\, {}^{\mathrm{e}}p_l} = 0 & (i = \ldots) \\
\dfrac{\partial x_i}{\partial\, {}^{\mathrm{e}}p_l} & \overset{CR}{:=} & \sum_{j \preceq i} \dfrac{\partial \varphi_i}{\partial x_j}\big((x_j)_{j \preceq i}\big)\dfrac{\partial x_j}{\partial\, {}^{\mathrm{e}}p_l} & (i = \ldots) \\
\Big(\dfrac{\partial \sigma_0}{\partial\, {}^{\mathrm{e}}p_l}, \ldots, \dfrac{\partial \sigma_N}{\partial\, {}^{\mathrm{e}}p_l}\Big) & := & \Big(\dfrac{\partial x_{j_0}}{\partial\, {}^{\mathrm{e}}p_l}, \ldots, \dfrac{\partial x_{j_N}}{\partial\, {}^{\mathrm{e}}p_l}\Big) & (0 \leq \ldots) \\
\Big(\dfrac{\partial \varepsilon_0}{\partial\, {}^{\mathrm{e}}p_l}, \ldots, \dfrac{\partial \varepsilon_N}{\partial\, {}^{\mathrm{e}}p_l}\Big) & := & \Big(\dfrac{\partial x_{i_0}}{\partial\, {}^{\mathrm{e}}p_l}, \ldots, \dfrac{\partial x_{i_N}}{\partial\, {}^{\mathrm{e}}p_l}\Big) & (0 \leq \ldots)
\end{array} \tag{17}$$

(each code line for $l = 1, \ldots, L$) and

$$\begin{array}{rcll}
\dfrac{\partial x_i}{\partial p_l} & := & \dfrac{\partial\, {}^e\sigma_i}{\partial p_l} = 0 & (i = \ldots) \\
\dfrac{\partial x_i}{\partial p_l} & := & \dfrac{\partial\, {}^e p_{i-N}}{\partial p_l} = 0 & (i = \ldots) \\
\dfrac{\partial x_i}{\partial p_l} & := & \dfrac{\partial p_{i-N-L}}{\partial p_l} = \delta_{l,(i-N-L)} & (i = \ldots) \\
\dfrac{\partial x_i}{\partial p_l} & \overset{CR}{:=} & \sum_{j \preceq i} \dfrac{\partial \varphi_i}{\partial x_j}\big((x_j)_{j \preceq i}\big) \dfrac{\partial x_j}{\partial p_l} & (i = \ldots) \\
\Big(\dfrac{\partial \sigma_0}{\partial p_l}, \ldots, \dfrac{\partial \sigma_N}{\partial p_l}\Big) & := & \Big(\dfrac{\partial x_{j_0}}{\partial p_l}, \ldots, \dfrac{\partial x_{j_N}}{\partial p_l}\Big) & (0 \leq \ldots) \\
\Big(\dfrac{\partial \varepsilon_0}{\partial p_l}, \ldots, \dfrac{\partial \varepsilon_N}{\partial p_l}\Big) & := & \Big(\dfrac{\partial x_{i_0}}{\partial p_l}, \ldots, \dfrac{\partial x_{i_N}}{\partial p_l}\Big) & (0 \leq \ldots)
\end{array} \tag{18}$$

(each code line for $l = 1, \ldots, L$). The original code lines remain without exception. The symbol δ denotes Kronecker's delta

$$\delta_{xy} = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

The first equations in (17) resp. (18) express the fact, that the input ${}^e\sigma$ is independent of all p_l resp. ${}^e p_l$. The second/third equations in (17) and (18) express the fact, that each parameter is independent of all others (surely, except for itself, where the derivative is trivially equal to unity).

Examples. (a) Consider the general exponential function $x_3 = \varphi_3(x_1, x_2) = x_1^{x_2}$, where $x_1 > 0$, $x_2 \in \mathbb{R}$. Here we have

$$\frac{\partial x_3}{\partial p_l} \overset{CR}{=} \Big(\frac{\partial}{\partial x_1}\, x_1^{x_2}\Big) \frac{\partial x_1}{\partial p_l} + \Big(\frac{\partial}{\partial x_2}\, x_1^{x_2}\Big) \frac{\partial x_2}{\partial p_l} = x_2 x_1^{x_2 - 1} \frac{\partial x_1}{\partial p_l} + \ln(x_1)\, x_1^{x_2} \frac{\partial x_2}{\partial p_l} \tag{19}$$

for $l = 1, \ldots, L$. Analogeously for the ${}^e p_l$. □

(b) Consider matrix multiplication $x_3 = \varphi_3(x_1, x_2) = x_1 x_2$, where $x_1 \in \mathbb{R}^{n_{1,1} \times n_{1,2}}$, $x_2 \in \mathbb{R}^{n_{2,1} \times n_{2,2}}$ and $n_{1,2} = n_{2,1}$. Here

$$\frac{\partial x_3}{\partial p_l} \overset{CR}{=} \frac{\partial x_1}{\partial p_l} \frac{\partial}{\partial x_1}(x_1 x_2) + \frac{\partial}{\partial x_2}(x_1 x_2) \frac{\partial x_2}{\partial p_l} = \frac{\partial x_1}{\partial p_l}\, x_2 + x_1 \frac{\partial x_2}{\partial p_l} \tag{20}$$

for $l = 1, \ldots, L$. Analogeously same for ${}^e p_l$. □

The variables x_1, x_2, $\partial x_1/\partial p_l$, $\partial x_2/\partial p_l$ are known, as they have already been calculated in the code before, and thus x_3, $\partial x_3/\partial p_l$. ■

In modern object orientated languages as MATLAB every array x_i of type *single/double* is overloaded by an object $(x_i.V, x_i.D)$ consisting of

> $x_i.V \rightsquigarrow$ array of size $n_{i,1} \times \ldots \times n_{i,r_i}$, type *single/double*, exactly identical to the original x_i ($V =$ 'VALUE')
>
> $x_i.D \rightsquigarrow$ array of size $n_{i,1} \times \ldots \times n_{i,r_i} \times 2L$, type *single/double*, containing the $2L$ partial derivatives $\partial x_i/\partial p_l$, $\partial x_i/\partial\, {}^e p_l$ ($D =$ 'DERIVATIVE')

All elemental operations φ_i are overloaded as in the example above for this new kind of objects. This way, the additional lines (17), (18) need not be really inserted between the original code lines. (In fact, this legitimates the terminology 'automatic'.)

An AD tool called ADMAT, developped by VERMA et al., for the language MATLAB is free available under

$$\textit{http://www.cs.cornell.edu/home/verma/AD} \tag{21}$$

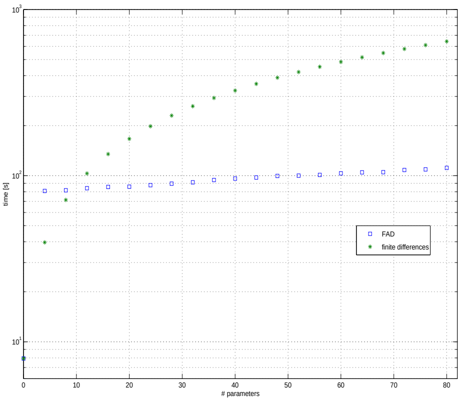
but still not free of bugs/sophisticated errors! At the moment, a new site is arising with many links to AD tools for other languages

http://www.autodiff.org/

Performance of MATLAB-code: If the number of parameters is very large, as in our case, you should realize the code for the derivatives *not* with for-loops, as it is (easily) formulated in (17), ..., (20), [5] and realized in (21), but use the typically Matlab-kind operations '.∗', './', 'repmat', 'vertcat', etc. In our case, we have designed our own AD code optimized especially for a large number of parameters, such that we have gained a speed factor of about 80 compared to (21). For more information about this, you should simply contact us.

AD advantages vs. FD

(a) *Speed.* Compared to evaluating the derivatives by FD, we find with our own designed AD code that e.g. in the proportional case, i.e. loading path A, see sect. 4



It is clear that the running times for FD process almost linear in the total numbers of parameters: For the forward/backward differences (14), (15), you need $2L$ evaluations of f – at the disturbed operating points $({}^e\sigma,\, p + \delta p_l\, e_l, {}^e p)$ resp. $({}^e\sigma,\, p, {}^e p + \delta {}^e p_l\, e_l)$ – plus 1 evaluation of f – at the operating point $({}^e\sigma,\, p,\, {}^e p)$.

Performing AD, we have gained a speed factor of more than 5.8 in the case (ii) of $2L = 82$ parameters $(p_l,\, {}^e p_l)$ and a factor of more than 3.3 in the case (i) of $L = 41$ parameters ${}^e p_l$, compared to FD.

In addition, the additional costs for the derivatives with respect to the p_l are less than 20%, and not $L/(L+1) \approx 98\%$ as it is when applying FD.

(b) *Accuracy.* When applying AD, the only errors that occur are roundoff errors – due to the relative machine accuracy $\epsilon \approx 2.22\text{E}^{-16}$ –, there are *no* truncation errors – due to the neglecting of higher order terms in (14), (15). Furthermore, there is no dependence on the choice of the perturbations $\delta {}^e p_l$ and δp_l.

AD disadvantages vs. FD

Memory. For each variable, that has been used in code for f, you additionally need $2L$ variables in the extended code for $(f, \nabla f)$.

The original code has $\sum_{i=0}^{M} \prod_{j=1}^{r_i} n_{i,j}$ *single/double*-variables, so the extended code has $(2L+1) \sum_{i=0}^{M} \prod_{j=1}^{r_i} n_{i,j}$ ones.

4 Results

Linear static analysis. The elastic stress is linear superposition

$$^{e}\sigma(t) = L_1(t)\, {}^{e}\sigma_1 + L_2(t)\, {}^{e}\sigma_2 \tag{22}$$

of the unit load case results

$$^{e}\sigma_1 = \left(\begin{array}{cc|c} 2.44 & 0 & 0 \\ 0 & 0.56 & 0 \\ \hline 0 & 0 & 0 \end{array}\right) = \begin{array}{l}\text{linear static FE response to static } \textit{unit} \\ \text{tension force along the rotational axis} \\ \text{(positively directed, i.e. } 1'\text{-direction)}\end{array}$$

and

$$^{e}\sigma_2 = \left(\begin{array}{cc|c} 0 & 1.56 & 0 \\ 1.56 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right) = \begin{array}{l}\text{linear static FE response to static } \textit{unit} \\ \text{torsional moment along the rotational axis} \\ \text{(positively directed, i.e. } 1'\text{-direction)}\end{array}$$

(details and corresponding notch factors in [6] sect. 4.3, independently verified by the author). ■

The targets and weights. Altogether 6 iterations have been performed, distinguishing the following 3 cases and cases (i), (ii).

(A) Measurements have been carried out (see [6], sect. 7) for the normal and shear strain ${}^{tr}\varepsilon_{11}(t)$ resp. ${}^{tr}\varepsilon_{12}(t)$ at the notch ζ, but not for the stresses. Here

$$w_\varepsilon^{11} = w_\varepsilon^{12} = w_\varepsilon^{21} = 1, \qquad w_\varepsilon^{22} = w_\varepsilon^{33} = w_\varepsilon^{23} = w_\varepsilon^{32} = w_\varepsilon^{13} = w_\varepsilon^{31} = 0$$

and all $w_\sigma^{ij} = 0$.

(B) Nonlinear quasi-static elastoplastic finite element computations (with Jiang's model and the experimental p_l) have been performed in [6] and independently verified by the author, yielding the results ${}^{ep}\sigma(t)$, ${}^{ep}\varepsilon(t)$. • To target at the stresses

$$w_\sigma^{11} = w_\sigma^{12} = w_\sigma^{21} = 1, \qquad w_\sigma^{22} = w_\sigma^{33} = w_\sigma^{23} = w_\sigma^{32} = w_\sigma^{13} = w_\sigma^{31} = 0$$

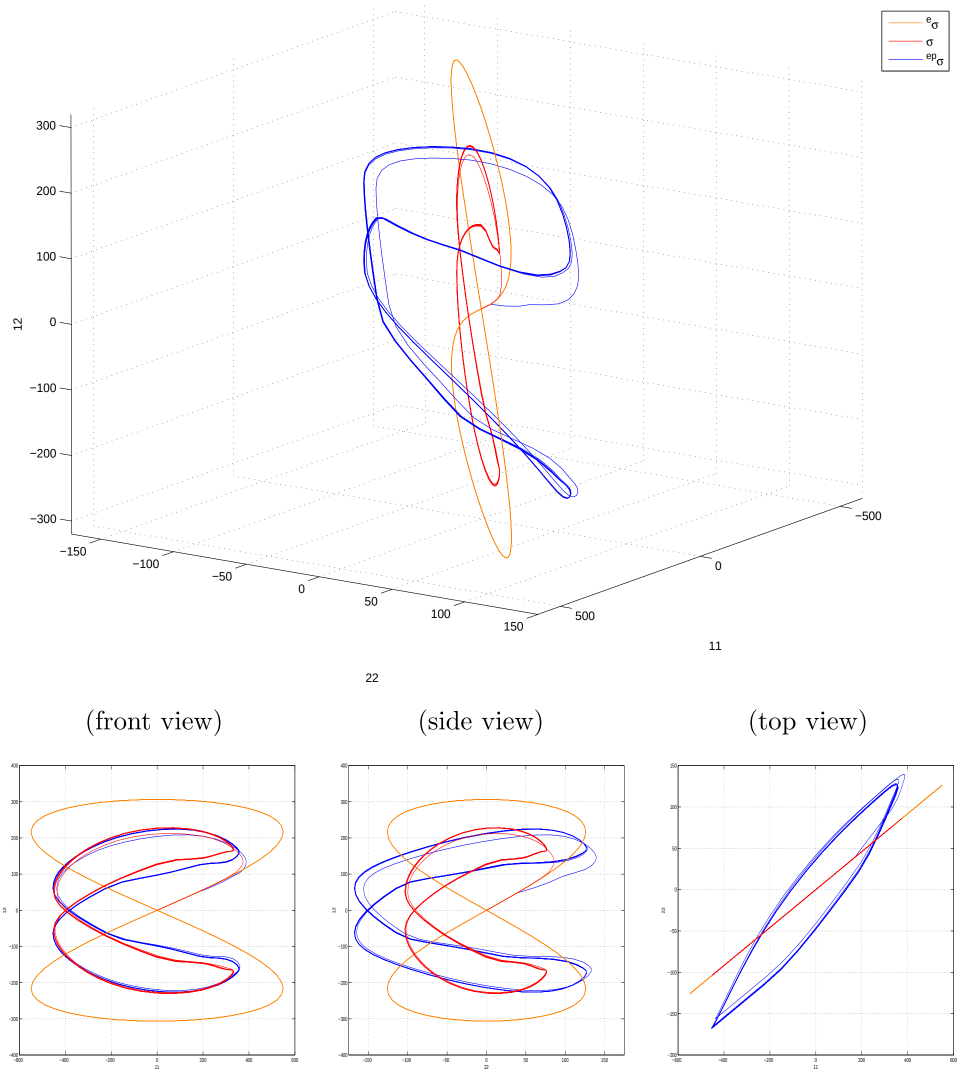
and all $w_\varepsilon^{ij} = 0$. • To target at the strains

$$w_\varepsilon^{11} = w_\varepsilon^{12} = w_\varepsilon^{21} = 1, \qquad w_\varepsilon^{22} = w_\varepsilon^{33} = w_\varepsilon^{23} = w_\varepsilon^{32} = w_\varepsilon^{13} = w_\varepsilon^{31} = 0$$

and all $w_\sigma^{ij} = 0$. ■

As a consequence of the simultaneity assumption in the model, we have insight in the the following facts:

Capability of the model. The whole notch stress correction model at ζ considers plane stresses, thus the spaces of ${}^{e}\sigma(t) = ({}^{e}\sigma_{11}(t), {}^{e}\sigma_{22}(t), {}^{e}\sigma_{12}(t))$ and $\sigma(t) = (\sigma_{11}(t), \sigma_{22}(t), \sigma_{12}(t))$ are both of dimension three. ${}^{e}\sigma(t)$, which is linear superposition (22), is completely contained in the plane (a two-dimensional linear subspace) through the origin spanned by the linear independent tensors ${}^{e}\sigma_1$ and ${}^{e}\sigma_2$. Thus, simultaneity in the model formulation implies parallelity of (3) and (4). Consequently, as the initial conditions for σ and ${}^{e}\sigma$ are the same, the corrected $\sigma(t)$ is trapped exactly in the same plane as ${}^{e}\sigma(t)$ is. For illustration see butterfly path D in the following figure.



(front view) (side view) (top view)

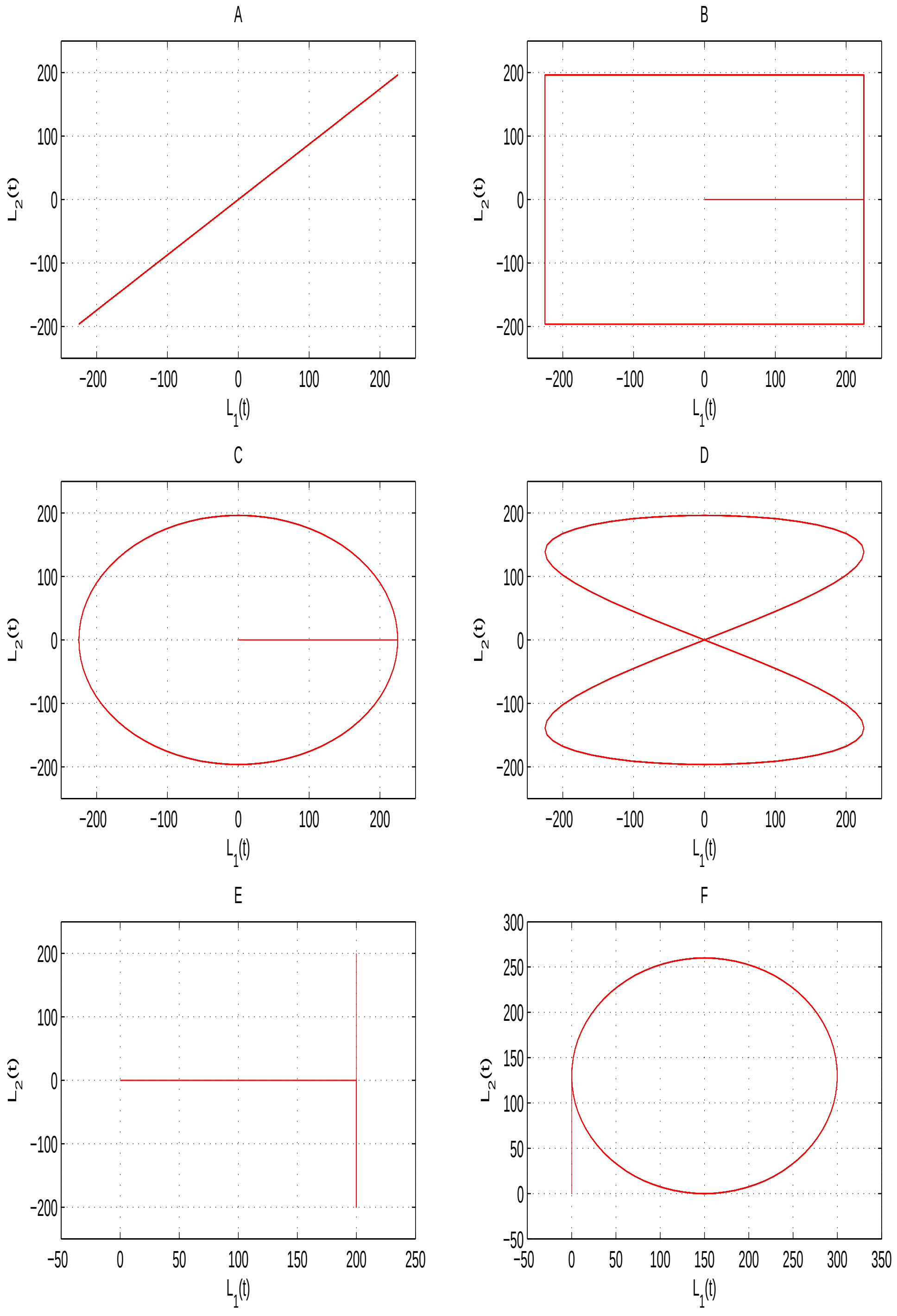
Note, that from the start until the first plastic yielding, all three paths $\sigma(t)$, ${}^{ep}\sigma(t)$ and ${}^{e}\sigma(t)$ are identical. (The path for ${}^{tr}\sigma(t)$ – which is not at our disposal – will looks somewhat like ${}^{ep}\sigma(t)$.)

(In the proportional case path A, where $L_1(t) = \alpha L_2(t)$ with a constant $\alpha > 0$, we even have ${}^{e}\sigma(t)$ and $\sigma(t)$ constrained in a one-dimensional linear subspace, i.e. a straight line.)

The nonlinear FE result ${}^{ep}\sigma(t) = {}^{ep}\sigma(t, \zeta)$ meanders somewhere around this plane, which is obviously due to the deformation of the nodes/elements surrounding ζ. Of course, the correction scheme f does *not* see any neighbourhood.

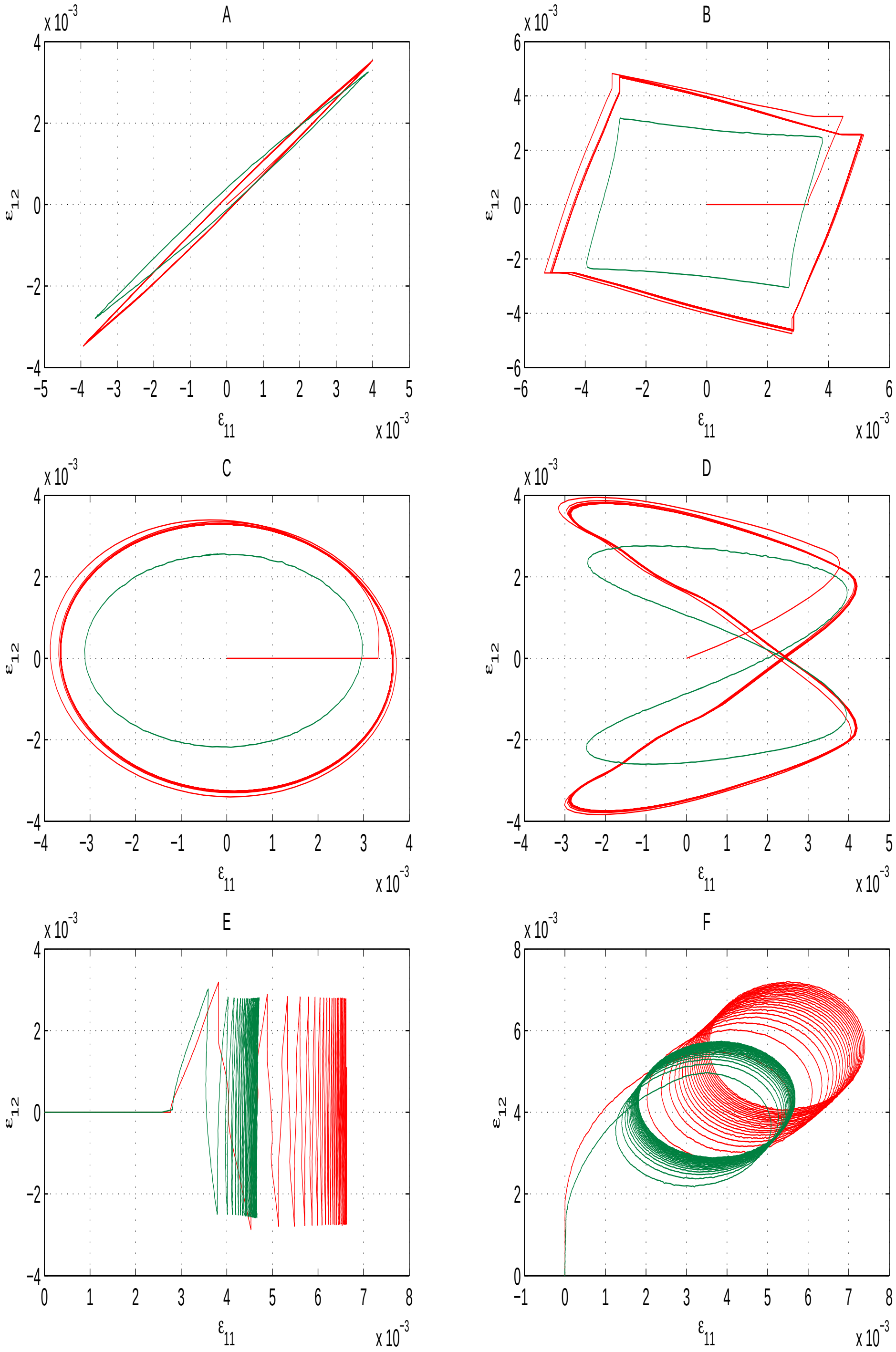
For this reason, there is *no* chance at all to fit all components of $\sigma(t)$ or $\varepsilon(t)$ sufficiently exact. But there seem to be and there in fact *are* – see results below – very good chances for the 11 and 12 components, but none for the 22-component. Anyway, this drawback cannot be simply removed. ■

The loading paths. We consider six different cyclic loading paths ($L_1(t)$, $L_2(t)$),



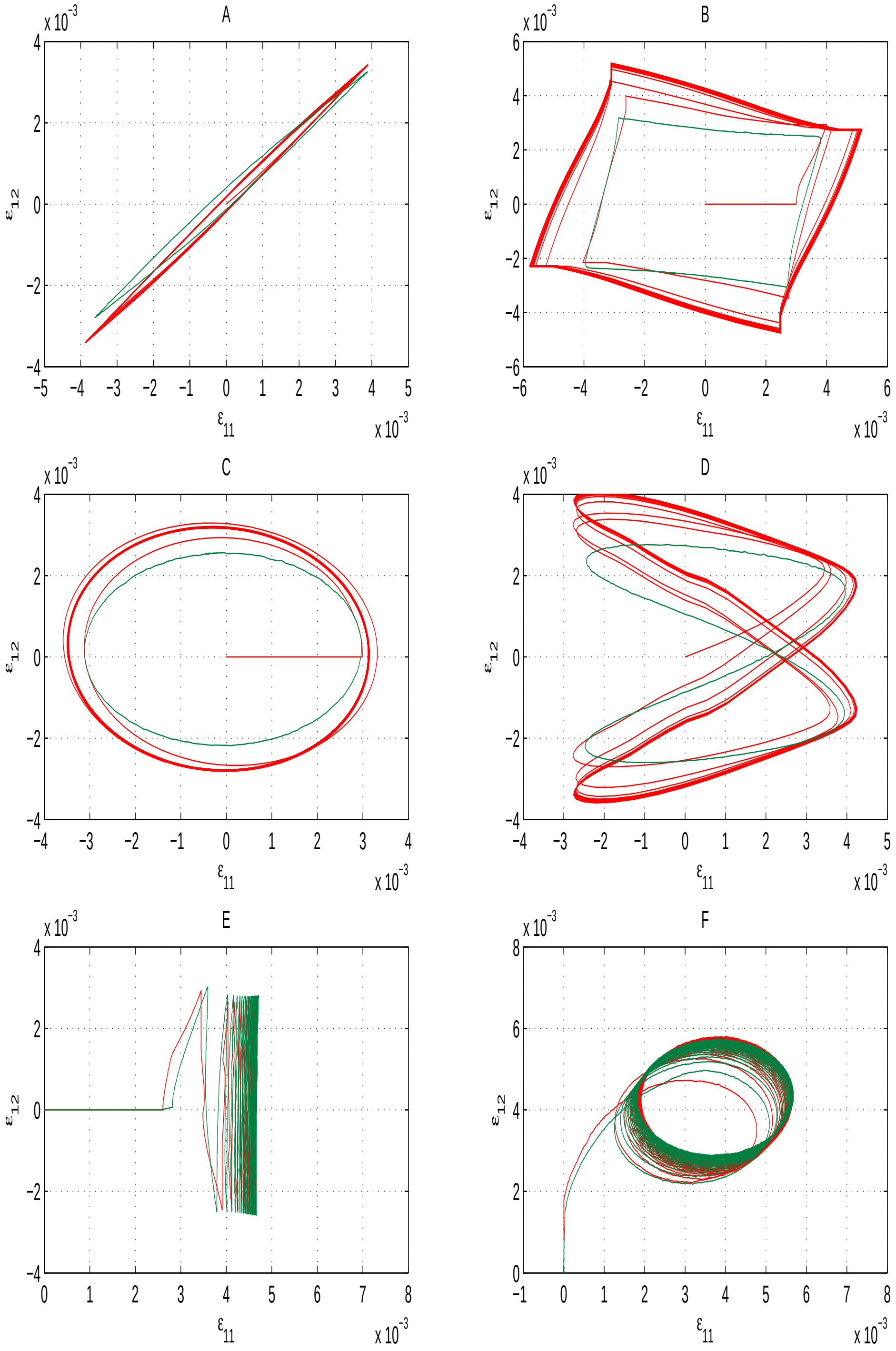
numbered by A, ..., F. Loads E, F will produce the ratchetting effect, as the 'elastic' mean stress (see load paths E, F) and thus the real mean stress is not equal to zero. For each path, we consider virgin material, thus the initial conditions for all of them are the same. For details about initial conditions see [14], sect. 1. ■

(A) target: measurements (strains), start of iteration (step 0)



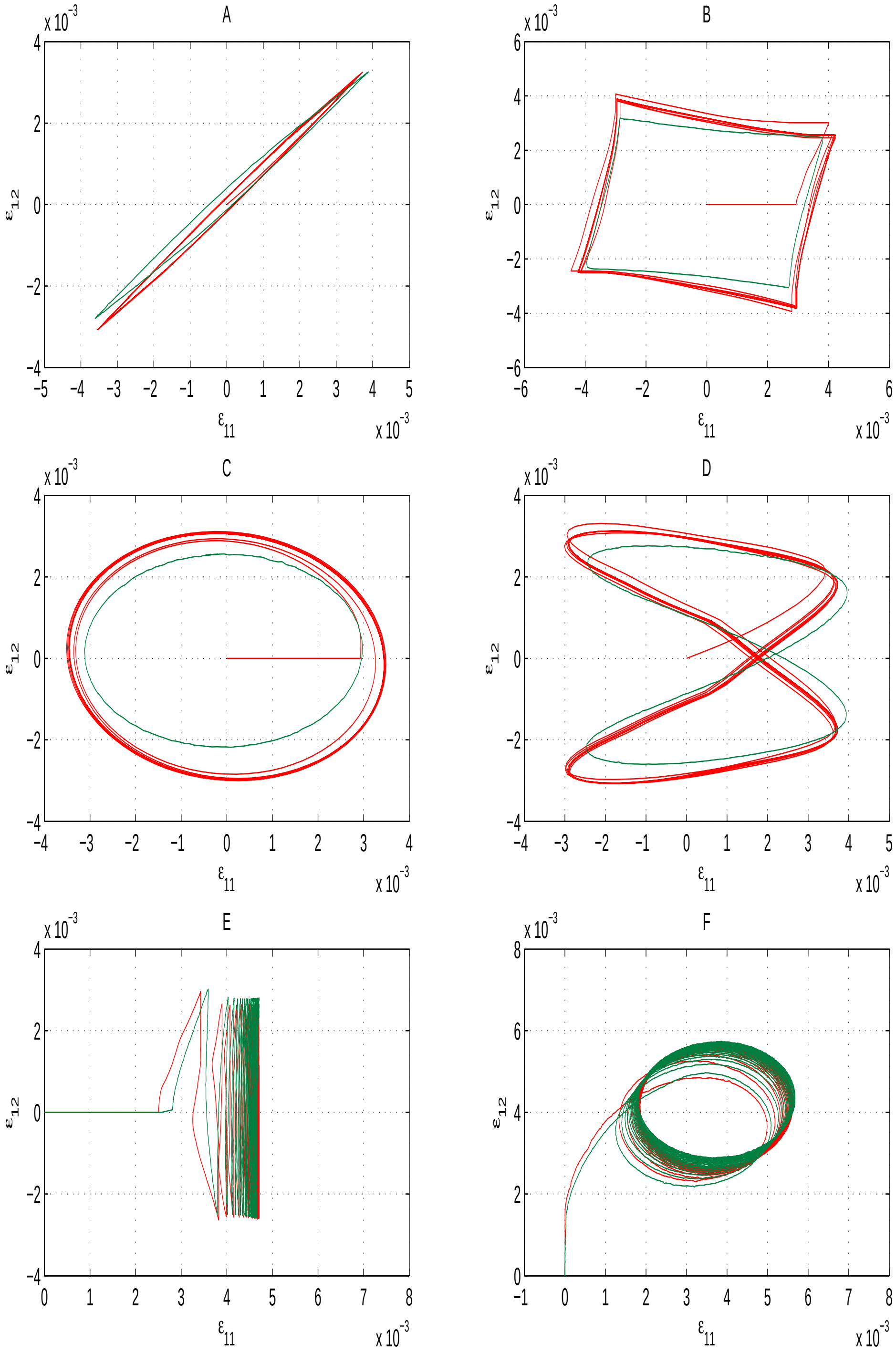
red – correction scheme results
blue – nonlinear finite element results
green – measurements

(A) target: measurements (strains), end of iteration (step 30), case (i)



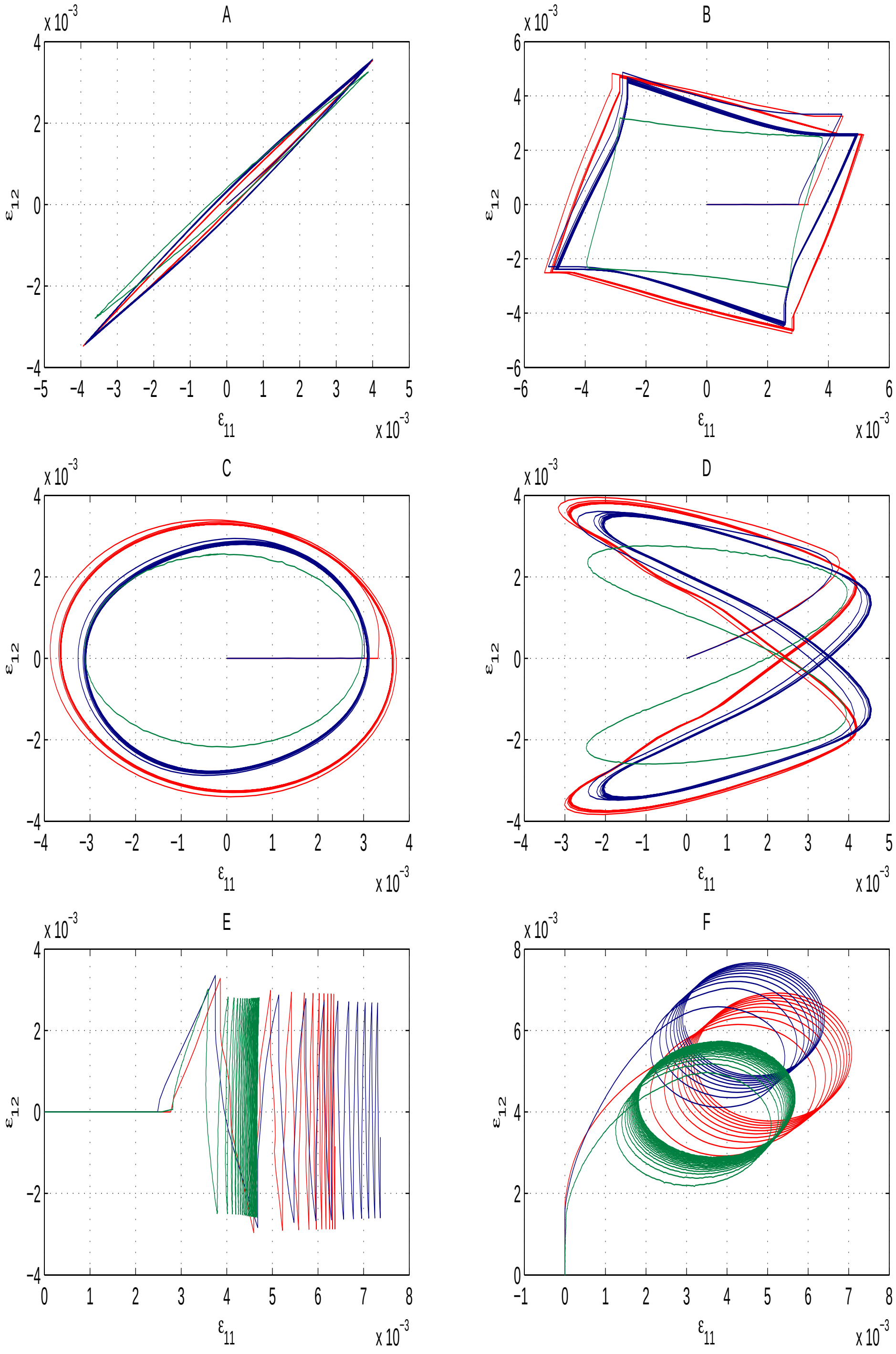
red – correction scheme results
blue – nonlinear finite element results
green – measurements

(A) target: measurements (strains), end of iteration (step 30), case (ii)



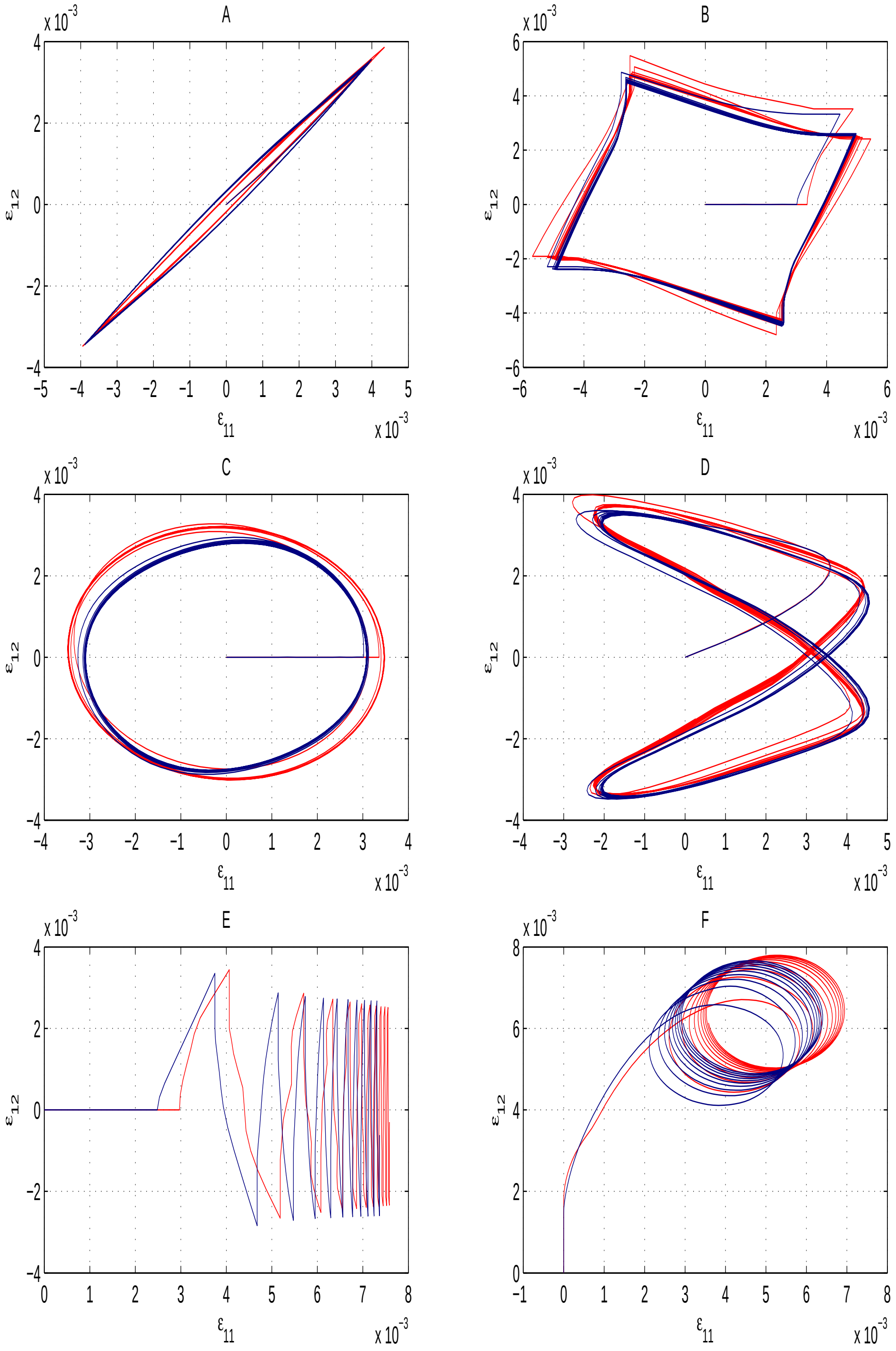
red – correction scheme results
blue – nonlinear finite element results
green – measurements

(B) target: nonlinear FE results (strains), start of iteration (step 0)



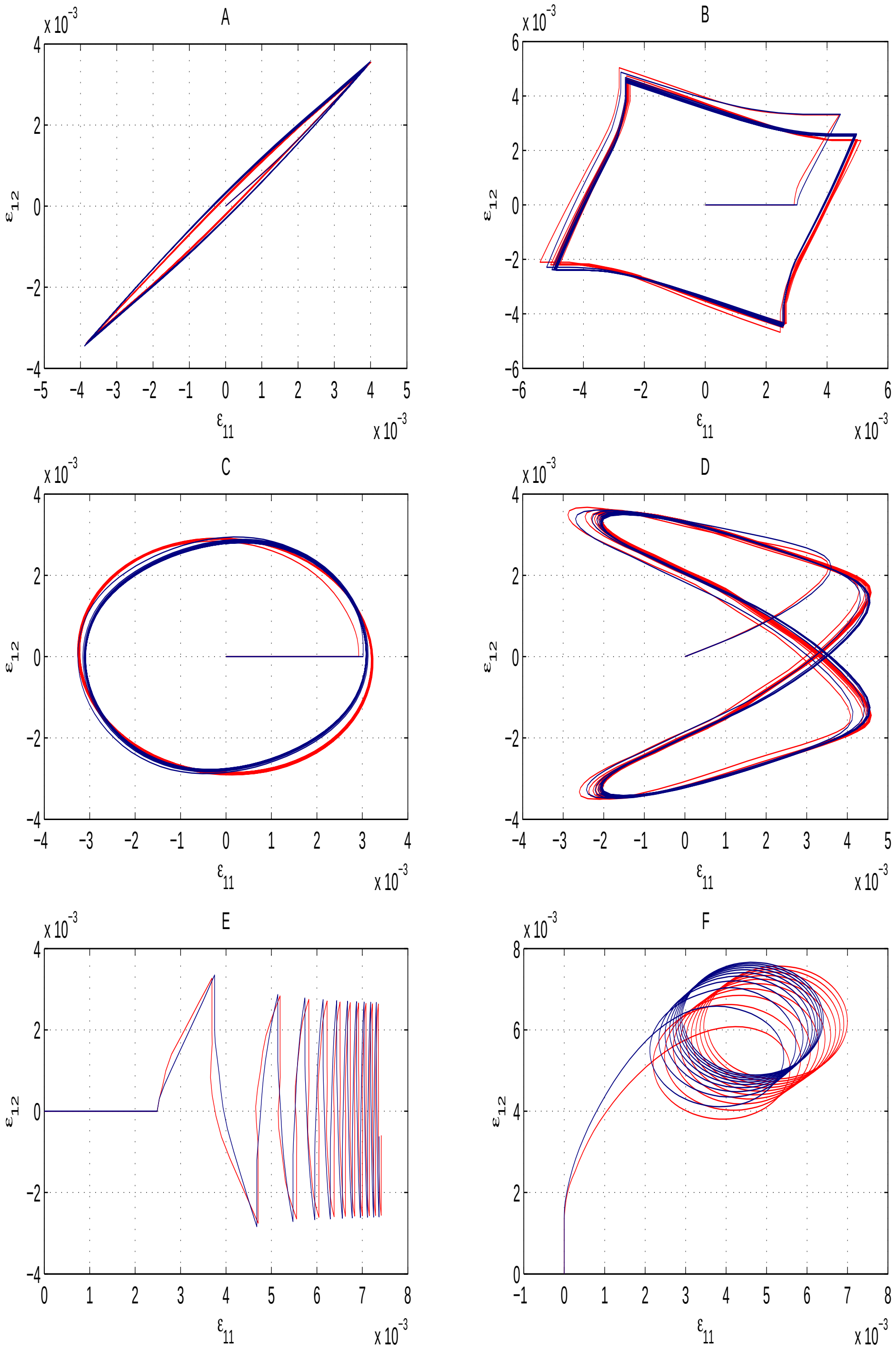
red – correction scheme results
blue – nonlinear finite element results
green – measurements

(B) target: nonlinear FE results (strains), end of iteration (step 30), case (i)



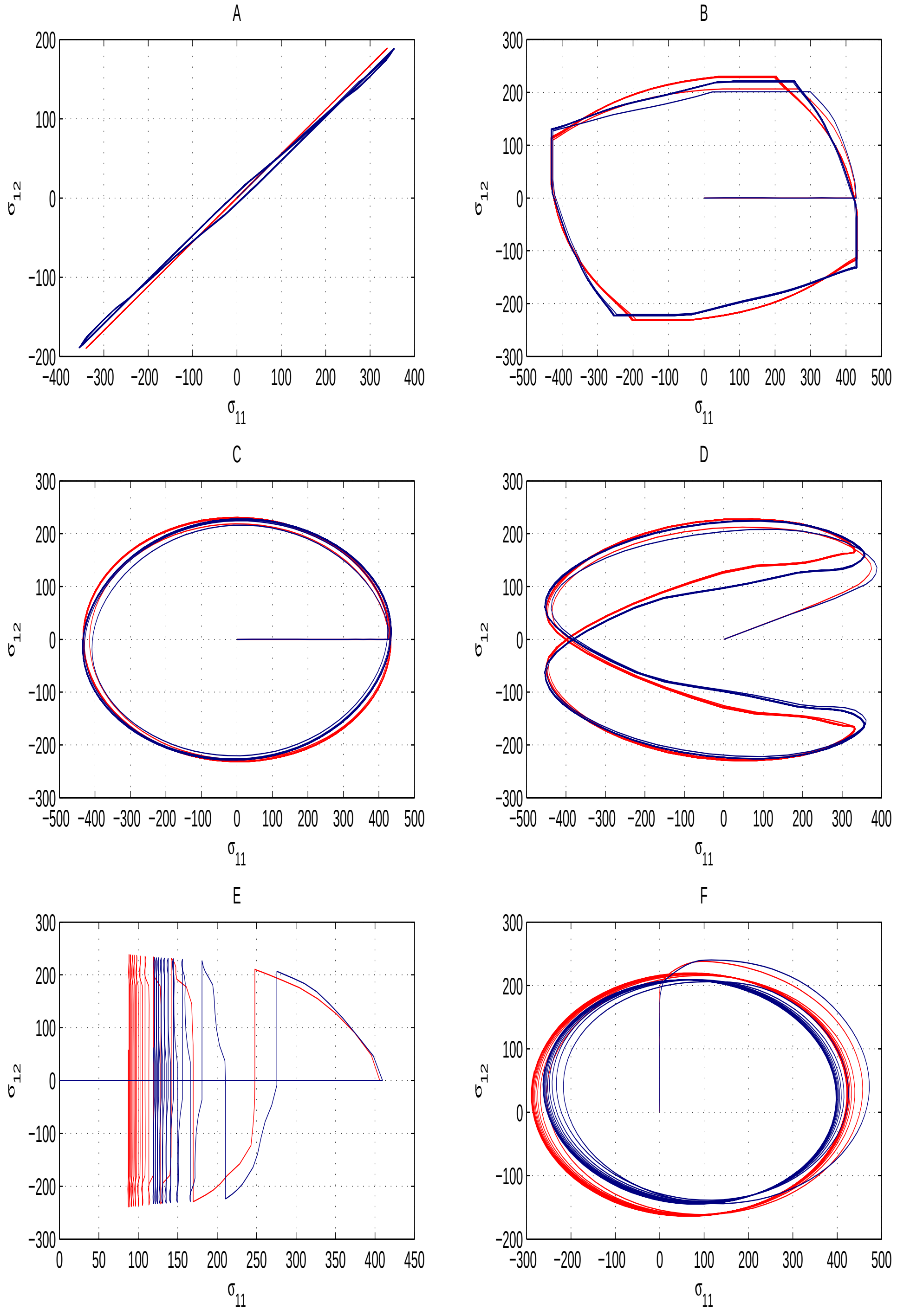
red – correction scheme results
blue – nonlinear finite element results
green – measurements

(B) target: nonlinear FE results (strains), end of iteration (step 30), case (ii)



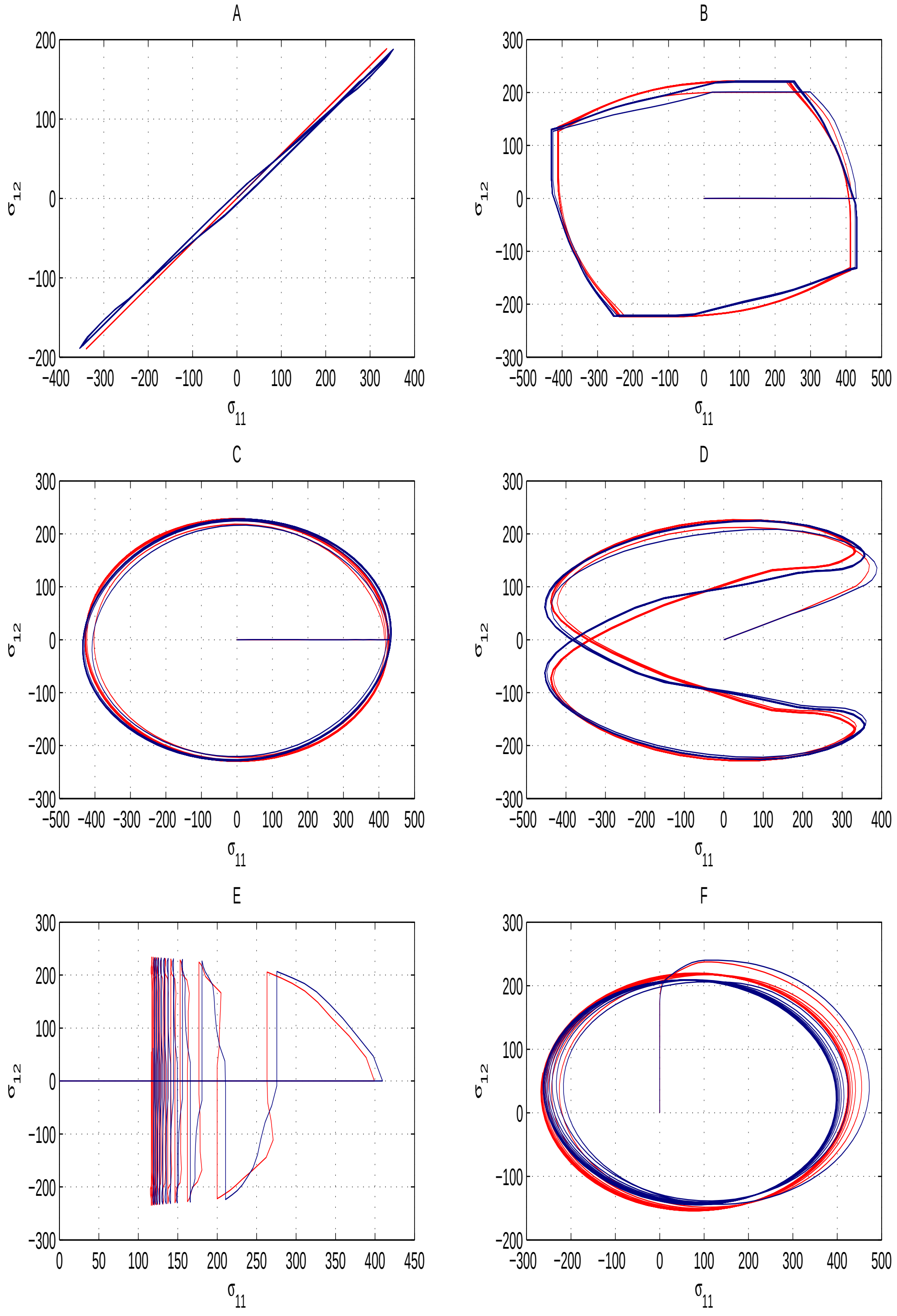
red – correction scheme results
blue – nonlinear finite element results
green – measurements

(B) target: nonlinear FE results (stresses), start of iteration (step 0)



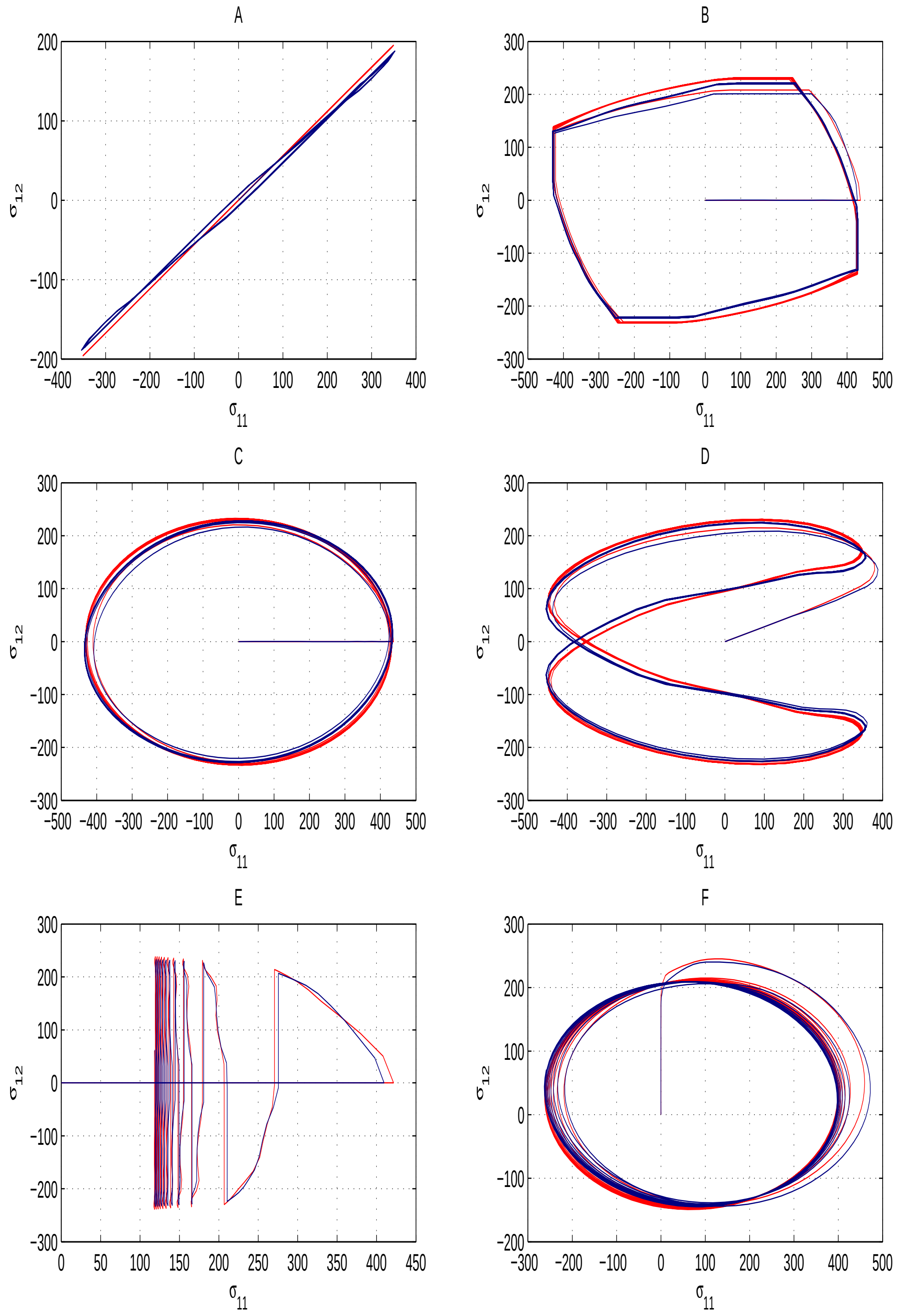
red – correction scheme results
blue – nonlinear finite element results
green – measurements

(B) target: nonlinear FE results (stresses), end of iteration (step 30), case (i)



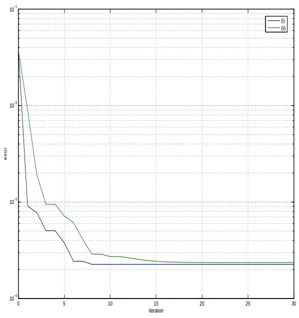
red – correction scheme results
blue – nonlinear finite element results
green – measurements

(B) target: nonlinear FE results (stresses), end of iteration (step 30), case (ii)

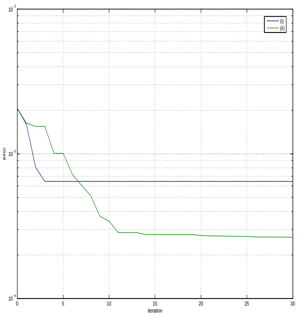


Discussion. In any case, the approach (ii) yields better results as approach (i). From a mathematical point of view this is too surprising, as we are minimizing over a larger parameter domain. But in case (A) the results are really *much* better.

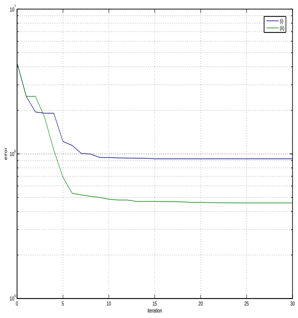
The development of the error $F(p, {}^{e}p)$ in (11) as a function of the iteration step number is depicted here:



(A) error strains



(B) error strains



(B) error stresses

For each of our six computations, good results are already obtained after approx. 20 steps or even earlier. This is quite satisfying, as it indicates that the Coleman-Li method is appropriate for our problem.

The error curves for (B) (ii) lie much below the ones for (B)(i). Even if the idea to consider both ${}^{e}p_l$ and p_l as notch parameters seems physically strange, its application may be justified by the good quality of results. It is seen in particular, that the maxima/minima (turning points) in the stress/strain phase diagrams (B) (ii) are hit very accurately.

In B, after iteration strains (i) 31.2%, (ii) 12.8%, stresses (i) 21.7%, (ii) 10.7% error of the initial error.

The reason is – see diagram (5) – that $\varepsilon^{pl}(t)$ is affected directly by ${}^{e}p_l$, but $\varepsilon^{el}(t)$ affected by both ${}^{e}p_l$ and p_l. Note that the p_l have a direct influence on $\varepsilon^{el}(t)$, whereas the influence of ${}^{e}p_l$ on $\varepsilon^{el}(t)$ is indirect. This is a kind of partial decoupling: the elastic stress and space mainly determines $\varepsilon^{pl}(t)$, the real stress space mainly $\varepsilon^{el}(t)$ (or equivalently $\sigma(t)$).

For the (A) (i/ii) iterations we had cycles only from the paths E and F at our disposal. Thus the error curves only comprise the error in E, F. They converge to almost the same minimum value.

(i) But it is seen that iterating the ${}^{e}p_l$ alone unfortunately leads do a detoriation of paths A, ..., D.

(ii) The good thing is, that identification just with E and F improves paths A, ..., D. This indicates that the parameters found are good and gives trust for more arbitrary loading. This really looks surprisingly fine. The results obtained for all paths except C are even significantly better than the FE results, which is remarkable. You can see, that the computed strains are still slightly overestimating the measured strains, which is usually desired when performing a subsequent fatigue analysis. ■

Future work.

(a) Fast AD techniques and parameter optimization for transient nonlinear finite element computations in low cycle fatigue analysis.

(b) The '${}^{e}\varepsilon$-approach' in [13], which is as well based on the introduction of additional elastic parameters ${}^{e}p_l$, does not exhibit parallelity of (3) and (4) as the '${}^{e}\sigma$-approach' investigated here does. There may be chances to have good parameter optimization for the 22-component too.

(c) Application of these techniques to improved versions of Jiang's model (e.g. [3]). Due to Hertel [6], sect. 7, it is impossible to fit the parameters both for proportional and nonproportional case, and that Jiang's model exhibits bad properties in out-of-phase loadings. ■

Acknowledgements. Thanks to the DFG and the GKMP Kaiserslautern for the financial support and to Olaf Hertel (TU Darmstadt, dept. of material science) for his kind provision of data. ■

References

[1] Bucynski A., Glinka G.: *An analysis of elasto-plastic strains and stresses in notched bodies subjected to cyclic non-proportional loading paths.* Biaxial/multiaxial fatigue and fracture, pp. 265-283, 2000.

[2] Coleman T. F., Li Y.: *An interior trust region approach for nonlinear minimization subject to bounds.* SIAM journal on optimization, Vol. 6, pp. 418-445, 1996.

[3] Döring R., Hoffmeyer J., Seeger T., Vormwald M.: *A plasticity model for calculating stress-strain sequences under multiaxial nonproportional cyclic loading.* Computational materials science, Vol. 28, pp. 587-596, 2003.

[4] Glinka G., Bucynski A., Ruggeri A.: *Elastic-plastic stress-strain analysis of notches under non-proportinal loading paths.* Archives of mechanics, Vol. 52, No. 4-5, pp. 589-607, 2000.

[5] Griewank A.: *Evaluating derivatives. Principles and techniques of automatic differentiation.* Society for industrial and applied mathematics. Frontiers in applied mathematics 19, 2000.

[6] Hertel O.: *Weiterentwicklung und Verifizierung eines Näherungsverfahrens zur Berechnung von Kerbbeanspruchungen bei mehrachsiger nichtproportionaler Schwingbelastung.* Diplomarbeit, Bauhaus-Universität Weimar, 2003.

[7] Hoffmeyer J., Döring R., Schliebner R., Vormwald M., Seeger T.: *Lebensdauervorhersage für mehrachsig nichtproportional schwingbeanspruchte Werkstoffe mit Hilfe des Kurzrissfortschrittkonzeptes.* FD-2/2000, FG Werkstoffmechanik, Technische Universität Darmstadt, 2000.

[8] Jiang Y.: *Cyclic plasticity with an emphasis on ratchetting.* PhD thesis, University of Illinois, Urbana-Champaign, 1993.

[9] Jiang Y., Sehitoglu H.: *Cyclic ratchetting of 1070 steel under multiaxial stress states.* International journal of plasticity, Vol. 10, No. 5, pp. 579-608, 1994.

[10] Jiang Y., Sehitoglu H.: *Multiaxial cyclic ratchetting under multiple step loading.* International journal of plasticity, Vol. 10, No. 8, pp. 849-870, 1994.

[11] Jiang Y., Sehitoglu H.: *Modeling of cyclic ratchetting plasticity, part I: Development of constitutive relations.* Transactions of the ASME, Vol. 63, pp. 720-725, 1996.

[12] Jiang Y., Sehitoglu H.: *Modeling of cyclic ratchetting plasticity, part II: Comparison of model simulations with experiments.* Transactions of the ASME, Vol. 63, pp. 726-733, 1996.

[13] Köttgen V. B., Barkey M. E., Socie D. F.: *Pseudo stress and pseudo strain based approaches to multiaxial notch analysis.* Fatigue and fracture of engineering materials and structures, Vol. 18, No. 9, pp. 981-1006, 1995.

[14] Lang H., Dressler K., Pinnau R.: *A multiaxial stress-strain correction scheme.* AGTM report, TU Kaiserslautern, No. 263, 2005.

[15] Moftakhar A., Bucynski A., Glinka G.: *Calculation of elasto-plastic strains and stresses in notches under multiaxial loading.* International journal of fracture, Vol. 70, pp. 357-373, 1995.