

# Numerical Modelling of Contact Elastic-Plastic Flows

N. M. Bessonov<sup>a1</sup>, S.F. Golovashchenko<sup>b</sup> and V.A. Volpert<sup>c</sup>

<sup>a</sup> Institute of the Problems of Mechanical Engineering, 199178 St.Petersburg, Russia

<sup>b</sup> Manufacturing Systems Department, Ford Research Laboratory, 481214 Dearborn, USA

<sup>c</sup> Institut Camille Jordan, UMR 5208 CNRS, University Lyon 1, 69622 Villeurbanne, France

**Abstract.** Wilkins' method has been successfully used since early 60s for numerical simulation of high velocity contact elastic-plastic flows. The present work proposes some effective modifications of this method including more sophisticated material model including the Baushinger effect; modification of the algorithm based on correction of the initial configuration of a solid; a contact algorithm based on the idea of a mild contact.

**Key words:** continuum mechanics, numerical simulation, Wilkins' method, contact algorithm, elastic-plastic flow

**AMS subject classification:** 74C15, 74M15, 65M06

## 1 Introduction

Wilkins' method was developed for simulations of high speed deformations of elastic-plastic solids for the 2D case in [65] (1964) and for the 2D and 3D cases in the monograph [67] (1998) and it has been widely used for a variety of problems. This method is implemented in the codes HEMP2D and HEMP3D [65, 67]. A number of results of specific elasto-plastic problems applying this method were published by Wilkins in [68, 69, 70] and Wilkins and Guinan [71, 72].

The description of the method [65, 67] successfully combines the general formulation of the problem of elasto-plastic flows with a detailed description of its numerical realization which was an important factor of its widespread employment. The most well known work by

---

<sup>1</sup>Corresponding author. E-mail: nickbessonov@yahoo.com

M. Wilkins is [65], published in 1964. The number of references only in the Russian technical literature on the translation of this work [66] exceeds several hundreds.

Currently, Wilkins' method is widely used both as a whole methodology and in some parts, in particular the algorithm of radial return-mapping plasticity. Among the recent references we can mention the works by Cherepanov [15], Cherepanov et al. [16], Makarov and Romanova [42], Popov et al. [51], Smolin et al. [59], Stefanov [62], Liu et al. [37], Makarov, et al. [43], Vaziri et al. [63], Bar-on et al. [7], Cocchetti and Perego, [19], Hoashi et al. [28], Jing [30], Mariotti et al. [44], Romanova et al. [54], Wünnemann and Ivanov [73], Balokhonov et al. [6], Bdzil et al. [8], Chen et al. [14], Jing and Hudson [31].

Mark Wilkins was among the first who tried a radial return-map two-step algorithm (elastic-predictor, inelastic-corrector) suggesting to split the total deformation between the elastic and plastic components, which is still being used with some modifications ([58]). "Initially suggested by Maenchen and Sack [40] (1964) and Wilkins [65] (1964), the return-map is an general a two-step procedure of the type elastic-predictor inelastic-corrector" Auricchio [2]. This method was used and discussed by many authors, Auricchio and Taylor [4], Brünig [13], Fish and Shek [23], Guo and Cox [26], Montans [46]. "Integration algorithm in plasticity have been introduced by Wilkins in the form of radial return algorithm with reference to the Mises criterion, and successively extended by Krieg and Key to the case of isotropic and kinematic hardening" Alfano et al. [1]. We refer also to the works by Auricchio [3], Balagansky et al. [5], Papadopoulos and Lu [49], Christensen [17], Düster and Rank [20], Howell and Ball [29], Mahnken [41], Parshikov and Medin [50], Rubin and Bodner [55], Le van et al. [35], Noels et al. [48], Balokhonov et al. [6], Cherepanov [15], Fellows and Barton [21], Luccioni et al. [39], Wallin and Ristinmaa [64], Kiselev and Kiselev [34], Ponthot [52], Bessette et al.[10] and others.

The references to the later publications by Wilkins using the same method [65, 66, 67, 68, 69, 70] and Wilkins and Guinan [71, 72] are listed in publications related to the analysis of high velocity impact, or penetration and perforation of elastic-plastic solids, where the method was employed as a whole or in its fragments: Benson and Stainier [9], Chen et al. [14], Feng et al. [22], Gupta and Ding [27], Johnson and Beissel [32], Lee and Yoo [36], Miller and Colella [45], Naboulsi and Palazotto [47], Roeder and Sun [53], Sherman [56], Simha et al. [57], Song et al. [60], Stevens and Batra [61], Zaera and Sanchez [75], Xiao [74], and others.

The authors of this paper use Wilkins' method to solve practical problems since the early 1980s. In this paper we will discuss some possible modifications of the method. The short description of the classical Wilkins' method is presented in Section 2.

Some possible modifications of Wilkins' method are given in Section 3 which is the main part of the paper. It is well known that vector-tensor notation is a compact and natural language for the mathematical formulation of continuum mechanics problems. So all the description will be given in direct vector-tensor form.

In Section 3.1 the formulation of elastic-plastic deformation of solids in more compact and general form will be discussed. However, it will not lead to any complications of the algorithm and, on contrary, will simplify some problems in Wilkins' formulation.

In Section 3.2 the direct vector technique of approximation of the initial equations will be described. This technique allows us to essentially shorten the presentation of the numerical scheme.

In Section 3.3 a variant of the algorithm of radial return-mapping plasticity is discussed. This modification is based on a transformation of the initial configuration of the medium. It allows in particular the description of workhardening and Bauschinger effects.

In Section 3.4 the contact algorithm based on the mild contact approach is described. This algorithm is simpler and more reliable, especially in the case of complex shape of the mesh.

## 2 Short description of Wilkins' method

The problem of elastics-plastics deformation is formulated by Wilkins for the 2D case in Lagrangian coordinates (Cartesian or cylindrical) with sliding interfaces between adjacent regions in [65] and for the 2D/3D cases in [67]. The work [65] published in 1964 is better known and almost all basic ideas of the method are presented here. We will analyze mostly the 3D case except just for a few cases where the 2D formulation can be used for simplicity without making it less general. At first, we describe schematically the classical Wilkins' method with the minimum level of detail needed below for the explanation of our modifications (see [65, 67] for further details). We will concentrate on these aspects of the method. More detailed discussion of Wilkins' method is out of the scope of this paper. In some cases, we have changed the original notation.

### 2.1 Governing equations

Wilkins' method is applied for the Lagrangian mesh which moves together with the medium. The governing equations (the momentum and energy conservations laws) are written in the Lagrangian form

$$\rho \frac{dv_i}{dt} = \frac{\partial \sigma_{ij}}{\partial x_j}, \quad i = 1, 2, 3, \quad (2.1)$$

$$\rho \frac{dE}{dt} = \frac{\partial v_i \sigma_{ij}}{\partial x_j}, \quad (2.2)$$

where  $t$  is time,  $x_j$  are the Cartesian coordinates,  $v_i$  are the components of the velocity vector,  $\rho$  is the density,  $\sigma_{ij}$  are components of the stress tensor,  $E$  is the internal energy. Throughout the paper we will use the summation convention over repeated subscripts. The mass conservation law is trivial when written in the Lagrangian coordinates. It means that the volume of any mesh cell is constant in time.

For any mesh node we write the equation

$$\frac{dx_i}{dt} = v_i, \quad i = 1, 2, 3. \quad (2.3)$$

The components of the stress tensor  $\sigma_{ij}$  are written as the sum of two parts:

$$\sigma_{ij} = \delta_{ij}p + S_{ij}, \quad i, j = 1, 2, 3, \quad (2.4)$$

where  $S_{ij}$  are the components of the deviator part of the stress tensor,  $p = \sigma_{ii}/3$  is the pressure.

A perfectly elastic material is characterized by a linear relationship between stress and strain in the form of Hooke's model:

$$S_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \delta_{ij} \frac{2}{3} \frac{\partial u_k}{\partial x_k} \right), \quad (2.5)$$

$$p = K \left( \frac{V}{V_0} - 1 \right), \quad \frac{V}{V_0} - 1 = \frac{\partial u_i}{\partial x_i}, \quad (2.6)$$

where  $\mu$  is the shear modulus,  $K$  is the bulk modulus,  $u_i$  are the components of the displacement vector,  $V$  is the actual volume,  $V_0$  is the initial volume ( $V_0 = V$  at  $t = 0$ ),  $\delta_{ij}$  is the Kronecker symbol (see also Appendix A). Relations (2.5) and (2.6) are differentiated with respect to time and are written respectively as

$$\frac{dS_{ij}}{dt} = \mu \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \delta_{ij} \frac{2}{3} \frac{\partial v_k}{\partial x_k} \right) + \gamma_{ij} \quad (2.7)$$

and

$$\frac{dp}{dt} = K \frac{\partial v_k}{\partial x_k}. \quad (2.8)$$

The term  $\gamma_{ij}$  in (2.7) corresponds to the correction resulting from the rigid body rotation. It is called Jaumann rate (see Appendix C).

The Von Mises yield criterion is conventionally used to describe the elastic limit:

$$J_2(\mathbf{S}) \leq \frac{\sigma_s^2}{3}, \quad (2.9)$$

where  $J_2(\mathbf{S}) = \frac{1}{2} [S_{11}^2 + S_{22}^2 + S_{33}^2 + 2(S_{12}^2 + S_{23}^2 + S_{31}^2)]$ ,  $\sigma_s$  is the plastic flow stress.

The basic system of equations for elastic-plastics problems (2.1)-(2.3), (2.7), (2.8), and (2.9) is written in the evolution form, which is more convenient for numerical solution.

## 2.2 Approximation

In the original version of Wilkins' method, the quadrangular regular grid was used. This approach was later on modified by a number of authors in order to get rid of the "water glass" problem, where the mesh element was not sensitive to modification of its shape from rectangular to trapezoidal without changing the length of its diagonals. In what follows, we will consider the modified approach.

The fragment of the simplest regular Lagrangian mesh, which allows one to avoid the "water glass" problem in the 2D case, is shown in Figure 1a. The computational domain is

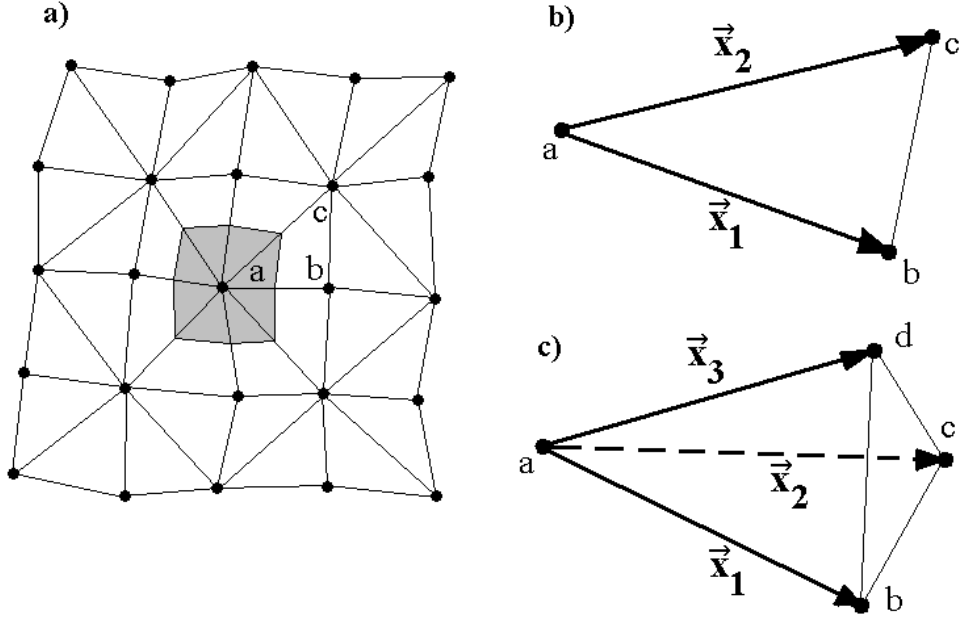


Figure 1: a) Fragment of 2D mesh ( $\bullet$  are nodes); b) 2D element; c) 3D element.

divided into cells. The 2D cell  $abcd$  (Figure 1a) consists of two triangles  $abc$  and  $acd$  (3D cell consists of 6 tetrahedra (Figure 1c)) called elements. The direction of diagonals can be the same in all mesh elements, or it can be different in different cells with an ordered or a random way to chose it. The center of the diagonal will be called the center of the cell.

We can see in the basic system of equations for elastic-plastics problems that some variables e.g.  $(\mathbf{S}, p)$  depend on the gradients of other variables e.g.  $(\vec{v}, \vec{x})$  and vice versa. So in the classical method, the first set of variables is approximated within elements (element variables) and the second set on the nodes of mesh (nodal variables).

The components of the velocity  $v_i$  are calculated at the nodes of the mesh (nodal variables), and the components of the stress tensor are calculated in the elements (element variables). The linear interpolation of the nodal variables is used. A control volume corresponds to every node of the mesh (gray in Figure 1a). The term volume is also employed for the 2D case.

Wilkins' method is based on the explicit approach. At the initial moment of time the distribution of nodal and element variables is known. This allows one to determine the right-hand side in equations (2.1), (2.2), (2.3), (2.7) and (2.8) at the previous time step  $n$ , and to determine the values of all variables at the new time step  $n + 1$ . For example, the new values of  $S_{ij}^{n+1}$  and  $p^{n+1}$  are found from the equations

$$\frac{S_{ij}^{n+1} - S_{ij}^n}{\Delta t} = \mu \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \delta_{ij} \frac{2}{3} \frac{\partial v_k}{\partial x_k} \right)^n + \gamma_{ij}^n, \quad (2.10)$$

$$\frac{p^{n+1} - p^n}{\Delta t} = K \left( \frac{\partial v_k}{\partial x_k} \right)^n, \quad (2.11)$$

where  $\Delta t$  is the time step. If it is necessary, the correction of  $S_{ij}^{n+1}$  is performed (see Section 2.3). Then the components of the stress tensor (2.4) are obtained at the new time step

$$\sigma_{ij}^{n+1} = \delta_{ij} p^{n+1} + S_{ij}^{n+1}.$$

The approximation of the right-hand side of equations (2.10) and (2.11) is carried out on the basis of the following relation

$$\frac{\partial \varphi}{\partial x_i} = \lim_{V \rightarrow 0} \frac{1}{V} \oint_S \varphi n_i dS, \quad (2.12)$$

where  $n_i$  are the components of the vector of external normal to the surface  $S$ ,  $\varphi$  is a nodal variable. For example, if (2.12) is used for  $\partial v_1 / \partial x_1$ , then the following approximation takes place in the tetrahedral element  $abcd$  (Figure 1c):

$$\begin{aligned} \frac{\partial v}{\partial x} = & \{ [(y_c - y_b)(z_d - z_c) - (z_c - z_b)(y_d - y_c)](v_b - v_a) + \\ & [(y_c - y_b)(z_d - z_c) - (z_c - z_b)(y_d - y_c)](v_c - v_b) + \\ & [(y_b - y_c)(z_d - z_b) - (z_b - z_a)(y_d - y_b)](v_d - v_c) \} / . \quad (2.13) \\ & \{ [(y_c - y_b)(z_d - z_c) - (z_c - z_b)(y_d - y_c)](x_b - x_a) + \\ & [(z_c - z_b)(x_d - x_c) - (x_c - x_b)(z_d - z_c)](y_c - y_b) + \\ & [(x_b - x_c)(y_d - y_b) - (y_b - y_a)(x_d - x_b)](z_d - z_c) \} \end{aligned}$$

In order to simplify the presentation, we use here the notation  $v = v_1$ ,  $x = x_1$ ,  $y = x_2$  and  $z = x_3$ . The approximation of other derivatives  $\partial v_i / \partial x_j$  in (2.10) and (2.11) is similar to this one.

We will now describe the approximation of the right-hand sides of (2.1) and (2.2). Let us consider for example (2.1) with  $i = 1$ . This equation is integrated over the control volume  $V$  corresponding to the node (gray in Figure 1a). For the component  $v_1$  of the velocity we have:

$$M \frac{v_1^{n+1} - v_1^n}{\Delta t} = \oint_S (\sigma_{11}^{n+1} n_1 + \sigma_{12}^{n+1} n_2 + \sigma_{13}^{n+1} n_3) dS, \quad (2.14)$$

where  $M = \sum \rho_i V_i$  is the mass corresponding to the given node; the summation is with respect to all nodes in the gray region in Figure 1, that is  $V = \sum V_i$ . After that we can find the new positions of the nodes from the equation

$$\frac{x_i^{n+1} - x_i^n}{\Delta t} = v_i^{n+1}.$$

Thus, the complete explicit discretization of the problem is obtained.

### 2.3 Plasticity algorithm

When the new values of the deviator components of the stress tensor  $S_{ij}^{n+1}$  are found, we should verify condition (2.9). If it is not satisfied, then the values  $S_{ij}^{n+1}$  are corrected according to the formula

$$S_{ij}^{n+1} := m S_{ij}^{n+1}, \quad (2.15)$$

where

$$m = \frac{\sigma_s}{\sqrt{-3J_2(\mathbf{S}^{n+1})}}. \quad (2.16)$$

This approach allows us to reduce the stress tensor components to the plastic yield circle. It is known as a return-mapping algorithm.

### 2.4 Contact algorithm

Wilkins' method includes a contact algorithm which allows one to simulate the dynamic deformation of several elasto-plastic solids taking into account their mutual contact interaction. The basic idea of the method is to analyze the mutual positions of the boundary mesh nodes. If one of the nodes of the first mesh appears inside the second mesh, a position of this node is corrected in such a way that it slides along the boundary. This correction is realized only on the basis of mutual position of the two meshes, i.e., only on the geometrical considerations. Some other questions, in particular destruction of solids or variable material parameters are also considered in [65, 67].

## 3 Modifications of the method

In this section, we will discuss some possible modifications of Wilkins' method in order to make it more effective for some problems. At the same time with some essential modifications of this method, we would like to attract the reader's attention to the direct vector notation which we use for the mathematical formulation, approximation and numerical scheme, and in the numerical code. This can give a considerable advantage at each stage. We are strongly concerned with the length and redundancy of the mathematical expressions used in numerical computations which can result in unnecessary complications and, in many cases, be a source of syntax errors. For example, the numerical scheme under consideration (see [65, 67]) takes several pages of complex formulas. Equations are written for each component of the velocity vector and for each component of the stress tensor. All these equations have to be transformed into the computer code in such an unfriendly form.

As we mentioned earlier, numerical modelling of continuous mechanics problems (in particular the problem of elastic-plastic deformation) includes three basic steps: mathematical formulation of the problem, approximation of the original equations, and finally the development of the computer code. All those three steps can be time consuming. It is well known that the vector form of mathematical formulations of continuum mechanics problems has

the most compact and clear representation invariant with respect to some changes of variables. However using the vector form only for governing equations does not provide any real benefit if they are transformed after that into the component form and used in this form in the computer code ([65] pages 241-260, [67] pages 83-100, 131-145 and others).

It will be shown below that the direct vector form can be used not only in the mathematical formulation but also for the numerical approximation and for the development of the computer code. As a result, the two last steps are essentially simplified, especially for complex 3D problems with various contact boundary conditions in irregular geometries where a nonorthogonal mesh is used.

*Remark.* We note that by the term vector we understand here the vector (tensor) variables in the sense of continuous mechanics, (the velocity  $\vec{v}$ , stress tensor  $\boldsymbol{\sigma}$ , etc.) contrary to the terminology employed in programming literature where any array of variables can be called a vector. The details of the vector notation are given in Appendix A).

### 3.1 Governing equations

The formulation of the elastic-plastics problem is given by Wilkins in the differential form (2.1) and (2.2), which follows from the conservation laws written in the general integral form. At the next step of Wilkins' method, these equations are integrated over a control volume in order to build their numerical approximation by the method of control volumes. Hence, we can use directly the formulation in the form of the general integral conservation laws instead of using its partial differential form (2.1) and (2.2):

$$\frac{d}{dt} \int_V \rho \vec{v} dV = \oint_S \boldsymbol{\sigma} \cdot d\vec{S}, \quad (3.1)$$

$$\frac{d}{dt} \int_V \rho E dV = \oint_S \vec{v} \cdot \boldsymbol{\sigma} \cdot d\vec{S}, \quad (3.2)$$

where  $V$  is an arbitrary volume, in particular, it can be a control volume corresponding to the given mesh node (Figure 1a),  $\vec{v}$  is the velocity,  $\boldsymbol{\sigma}$  is the stress tensor,  $S$  is the surface of  $V$ ,  $d\vec{S} = \vec{n} dS$ , and  $\vec{n}$  is an outward normal vector to  $S$ .

Equations (2.3) and (2.4) can also be rewritten in the vector form:

$$\frac{d\vec{x}}{dt} = \vec{v}, \quad (3.3)$$

where  $\vec{x}$  is a vector of actual position, in particular a position vector of the mesh node at time  $t$ , and

$$\boldsymbol{\sigma} = \mathbf{I}p + \mathbf{S}. \quad (3.4)$$

Instead of (2.7) and (2.8), we consider a more general model

$$\mathbf{S} = \mu \mathbf{B}_D, \quad (3.5)$$



$$p = K \left( \frac{V}{V_0} - 1 \right), \quad (3.6)$$

where

$$\mathbf{B}_D = \mathbf{B} - J_1(\mathbf{B})\mathbf{I}/3 \quad (3.7)$$

is the deviator part of the left Cauchy-Green tensor  $\mathbf{B}$ ,

$$\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^T, \quad (3.8)$$

$$\mathbf{F} = \frac{d\vec{\mathbf{x}}}{d\vec{\mathbf{X}}} = \vec{\mathbf{x}}\nabla_{\vec{\mathbf{x}}} = \frac{\partial x_i}{\partial X_m} \vec{\mathbf{e}}_i \vec{\mathbf{e}}_m = \begin{vmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{vmatrix} \quad (3.9)$$

is the deformation gradient,  $\vec{\mathbf{x}}$  is the actual position vector,  $\vec{\mathbf{X}}$  is the initial position vector ( $\vec{\mathbf{X}} = \vec{\mathbf{x}}$  at  $t = 0$ ),  $\vec{\mathbf{x}}\nabla_{\vec{\mathbf{x}}} \equiv (\nabla_{\vec{\mathbf{x}}}\vec{\mathbf{x}})^T$ .

It is easy to show that model (2.5) is a particular case of (3.5). Let  $\vec{\mathbf{x}} = \vec{\mathbf{X}} + \vec{\mathbf{u}}$ , then from (3.8) and (3.9) follows

$$\mathbf{B} = \vec{\mathbf{x}}\nabla_{\vec{\mathbf{x}}} \cdot \nabla_{\vec{\mathbf{x}}}\vec{\mathbf{x}} = (\vec{\mathbf{u}}\nabla_{\vec{\mathbf{x}}} + \mathbf{I}) \cdot (\nabla_{\vec{\mathbf{x}}}\vec{\mathbf{u}} + \mathbf{I}).$$

Next,  $\nabla_{\vec{\mathbf{x}}} \approx \nabla_{\vec{\mathbf{x}}}$  is assumed. Then the previous expression can be approximated by

$$\approx (\vec{\mathbf{u}}\nabla_{\vec{\mathbf{x}}} + \mathbf{I}) \cdot (\nabla_{\vec{\mathbf{x}}}\vec{\mathbf{u}} + \mathbf{I}) = \mathbf{I} + \vec{\mathbf{u}}\nabla_{\vec{\mathbf{x}}} + \nabla_{\vec{\mathbf{x}}}\vec{\mathbf{u}} + \vec{\mathbf{u}}\nabla_{\vec{\mathbf{x}}} \cdot \nabla_{\vec{\mathbf{x}}}\vec{\mathbf{u}}$$

Assuming that  $\vec{\mathbf{u}}$  is small and the second order terms can be neglected, we get

$$\approx \mathbf{I} + \vec{\mathbf{u}}\nabla_{\vec{\mathbf{x}}} + \nabla_{\vec{\mathbf{x}}}\vec{\mathbf{u}}. \quad (3.10)$$

Using the same assumptions, it is possible to obtain

$$J_1(\mathbf{B}) \approx 2\nabla_{\vec{\mathbf{x}}} \cdot \vec{\mathbf{u}} + 3. \quad (3.11)$$

Then using (3.7), (3.10) and (3.11), we obtain

$$\mu\mathbf{B}_D \approx \mu \left[ \vec{\mathbf{u}}\nabla_{\vec{\mathbf{x}}} + \nabla_{\vec{\mathbf{x}}}\vec{\mathbf{u}} - \frac{2}{3}\mathbf{I}(\nabla_{\vec{\mathbf{x}}} \cdot \vec{\mathbf{u}}) \right]. \quad (3.12)$$

The right-hand side of this expression is the same as in the equation (2.5).

The model (3.5) and (3.6) has some advantages compared to Hooke's model (2.7) and (2.8):

— There is no need to carry out correction of the element rotation, that is not necessary to compute the components of the tensor  $\gamma_{ij}$  in (2.7) at each time step. Even though the model (3.5) compared with (2.7) requires some additional calculations, they are less essential than

the computational expenses to calculate the correction of an element rotation in Hooke's model. So the total CPU time does not increase;

— Transition to the incremental stress form (2.7), (2.8) increases the order of the differential equation due to differentiation with respect to time. This can have some negative consequences;

— The modification of the rheological model allows the modification of the classical radial return-map algorithm applied by Wilkins (see Section (3.3)).

We restrict ourselves to the von Mises condition (2.9). Instead of (2.9), we write it in the equivalent form using (3.5):

$$J_2(\mathbf{B}_D) \leq \frac{\sigma_s^2}{3\mu^2}. \quad (3.13)$$

Thus, the modified system of equations includes governing equations (3.1) and (3.2), equation (3.3) describing the motion of the medium, the rheological model (3.4), (3.5) and (3.6), and the yield criterion (3.13).

## 3.2 Approximation

After formulating the governing equations in the previous section, we discuss their approximation. An explicit approximation is used in the classical method. We can see from [65], [67] that finite difference approximation of the differential equations can be related to some difficulties because of the complexity of approximating expressions. Especially this is true for the 3D case. The complete description of the difference scheme in Wilkins' method takes several pages.

In this section we will describe the technique which will allow us to reduce the length of approximating formulas at the first stage of approximation up to several times. As a result, the numerical code becomes less time consuming and easier to modify. In order to do this, we will use direct vector notation [11]. From the mathematical point of view, the vector technique of approximation is equivalent to the classical or component-wise presentation but from the user point of view it is several times more compact and, apart from the cleaner syntax, closer to the same visual appearance of initial equations. It is also a more efficient way to develop the computer code, which can prevent from syntax errors both at the stage of developing the numerical scheme and at the stage of writing the actual code.

Several types of numerical mesh and of finite difference schemes can be employed with different efficacy. For simplicity, we will describe the vector form of numerical approximation for the classical method with the mesh described in Section 2.2 and with an explicit scheme.

We will illustrate all the specifics of the direct vector technique for a finite difference approximation of the equation for the stress tensor (3.4) for which the component-wise approach is the most cumbersome. Stress tensor is an element variable.

Consider the tetrahedral element  $abcd$  (Figure 1c) or the triangle element  $abc$  (Figure 1b) for the 2D case. We denote the initial position of the tetrahedron vertices at the initial moment of time by the position-vectors  $\vec{\mathbf{X}}_a$ ,  $\vec{\mathbf{X}}_b$ ,  $\vec{\mathbf{X}}_c$  and  $\vec{\mathbf{X}}_d$ . Due to the deformation process, the vertices move from their initial positions. We denote the actual position of the vertices

of the element at time  $t$  by  $\vec{x}_a, \vec{x}_b, \vec{x}_c$  and  $\vec{x}_d$ . Let  $\vec{x}_i$  and  $\vec{X}_i$  ( $i = 1, 2, 3$ ) be the right-hand set of vectors directed along any three different ribs of the tetrahedron, for example:

$$\vec{x}_1 = \vec{x}_b - \vec{x}_a, \quad \vec{x}_2 = \vec{x}_c - \vec{x}_a, \quad \vec{x}_3 = \vec{x}_d - \vec{x}_a, \quad (3.14)$$

and

$$\vec{X}_1 = \vec{X}_b - \vec{X}_a, \quad \vec{X}_2 = \vec{X}_c - \vec{X}_a, \quad \vec{X}_3 = \vec{X}_d - \vec{X}_a. \quad (3.15)$$

For the 2D triangular element with 2D initial  $\vec{X}_a, \vec{X}_b, \vec{X}_c$  and actual  $\vec{x}_a, \vec{x}_b, \vec{x}_c$  position-vectors respectively, we can introduce two sets of vectors

$$\vec{x}_1 = \vec{x}_b - \vec{x}_a, \quad \vec{x}_2 = \vec{x}_c - \vec{x}_a, \quad (3.16)$$

and

$$\vec{X}_1 = \vec{X}_b - \vec{X}_a, \quad \vec{X}_2 = \vec{X}_c - \vec{X}_a. \quad (3.17)$$

To find the stress tensor  $\boldsymbol{\sigma}$  (see (3.4)) in the element at time  $t$  we need to find the values of  $\mathbf{S}$  and  $p$  from (3.5) and (3.6) respectively.

As a first step, we approximate the pressure  $p$  in the element. Since we introduce the initial and current positions of the vertices, then we can find the initial  $V_0$  and the actual  $V$  volumes of the element:

$$V = \frac{1}{6} \boldsymbol{\varepsilon} \cdots \vec{x}_3 \vec{x}_2 \vec{x}_1, \quad V_0 = \frac{1}{6} \boldsymbol{\varepsilon} \cdots \vec{X}_3 \vec{X}_2 \vec{X}_1 \quad (3.18)$$

for the 3D case, and

$$V = \frac{1}{2} \boldsymbol{\varepsilon} \cdots \vec{x}_2 \vec{x}_1, \quad V_0 = \frac{1}{2} \boldsymbol{\varepsilon} \cdots \vec{X}_2 \vec{X}_1 \quad (3.19)$$

for the 2D case, respectively. Here  $\boldsymbol{\varepsilon}$  is Levi-Civittá tensor whose rank corresponds to the space dimension - third for the 3D case, and second for the 2D case (for more details see Appendixes A. Having defined the values of  $V$  and  $V_0$ , we obtain an approximation for the actual value of  $p$  in the element according to (3.6) (or (2.6)).

Next, we will use the vector form of approximation for the terms in the right-hand side of equation (2.7),  $\partial v_i / \partial x_j$ ,  $\partial v_j / \partial x_i$ , and  $\partial v_k / \partial x_k$ , which we rewrite in the vector form as  $\vec{x} \nabla_{\vec{x}}$ ,  $\nabla_{\vec{x}} \vec{x}$ , and  $\nabla_{\vec{x}} \cdot \vec{x}$ .

In accordance with the classical method, the linear approximation of the velocity  $\vec{v}$  in tetrahedral element is used, i.e.

$$\vec{v} = \mathbf{A} \cdot \vec{x} + \vec{b}, \quad (3.20)$$

where  $\mathbf{A}$ , and  $\vec{b}$  are some parameters.

Using the equation (3.20) we can find

$$\nabla_{\vec{x}} \vec{x} = \frac{\partial (A_{ik} v_k + b_i)}{\partial x_m} \vec{e}_m \vec{e}_i = A_{ik} \delta_{km} \vec{e}_m \vec{e}_i = \mathbf{A}^T \quad (3.21)$$

$$\begin{aligned}
\nabla_{\vec{x}} \vec{v} &= \frac{\partial(A_{ik}x_k + b_i)}{\partial x_m} \vec{e}_m \vec{e}_i = A_{ik} \delta_{km} \vec{e}_m \vec{e}_i = \mathbf{A}^T \\
\vec{v} \nabla_{\vec{x}} &= \frac{\partial(A_{ik}x_k + b_i)}{\partial x_m} \vec{e}_i \vec{e}_m = A_{ik} \delta_{km} \vec{e}_i \vec{e}_m = \mathbf{A} \\
\nabla_{\vec{x}} \cdot \vec{x} &= \frac{\partial(A_{ik}x_k + b_i)}{\partial x_m} \vec{e}_i \cdot \vec{e}_m = A_{ik} \delta_{km} \delta_{im} = \mathbf{A} \cdot \mathbf{I}
\end{aligned} \tag{3.22}$$

As we can see from equation (3.20), in order to determine all approximations (3.22), it is necessary to find the tensor  $\mathbf{A}$ . Let  $\vec{v}_a$ ,  $\vec{v}_b$ ,  $\vec{v}_c$  and  $\vec{v}_d$  be the actual velocities in the vertices  $a$ ,  $b$ ,  $c$ , and  $d$  of the element, respectively. Then from equation (3.21) we can write

$$\begin{cases} \vec{v}_a = \mathbf{A} \cdot \vec{x}_a + \vec{b} \\ \vec{v}_b = \mathbf{A} \cdot \vec{x}_b + \vec{b} \\ \vec{v}_c = \mathbf{A} \cdot \vec{x}_c + \vec{b} \\ \vec{v}_d = \mathbf{A} \cdot \vec{x}_d + \vec{b} \end{cases} \tag{3.23}$$

or

$$\begin{cases} \vec{v}_1 = \mathbf{A} \cdot \vec{x}_1 \\ \vec{v}_2 = \mathbf{A} \cdot \vec{x}_2 \\ \vec{v}_3 = \mathbf{A} \cdot \vec{x}_3 \end{cases} \tag{3.24}$$

where

$$\vec{v}_1 = \vec{v}_b - \vec{v}_a, \quad \vec{v}_2 = \vec{v}_c - \vec{v}_a, \quad \vec{v}_3 = \vec{v}_d - \vec{v}_a, \tag{3.25}$$

(the same with relations (3.14)). Having solved the system (3.24), we can find the tensor  $\mathbf{A}$

$$\mathbf{A} = \vec{v}_i \vec{x}^i, \tag{3.26}$$

where

$$\vec{x}^1 = \frac{\boldsymbol{\varepsilon} \cdot \vec{x}_3 \vec{x}_2}{\boldsymbol{\varepsilon} \cdot \vec{x}_3 \vec{x}_2 \vec{x}_1}, \quad \vec{x}^2 = \frac{\boldsymbol{\varepsilon} \cdot \vec{x}_1 \vec{x}_3}{\boldsymbol{\varepsilon} \cdot \vec{x}_3 \vec{x}_2 \vec{x}_1}, \quad \vec{x}^3 = \frac{\boldsymbol{\varepsilon} \cdot \vec{x}_2 \vec{x}_1}{\boldsymbol{\varepsilon} \cdot \vec{x}_3 \vec{x}_2 \vec{x}_1}. \tag{3.27}$$

The sets of vectors  $\vec{x}_i$  and  $\vec{x}^i$  are called the reciprocal vectors (see more details in Appendix B).

For the 2D case instead of (3.27) we have

$$\vec{x}^1 = \frac{\boldsymbol{\varepsilon} \cdot \vec{x}_2}{\boldsymbol{\varepsilon} \cdot \vec{x}_2 \vec{x}_1}, \quad \vec{x}^2 = \frac{-\boldsymbol{\varepsilon} \cdot \vec{x}_1}{\boldsymbol{\varepsilon} \cdot \vec{x}_2 \vec{x}_1}. \tag{3.28}$$

To find  $\mathbf{S}$  in the element (see (3.5)), we need to approximate  $\mathbf{B}$ .

Following the classical method, we will use the linear transformation of the element. Then from its initial to actual position

$$\vec{x} = \mathbf{A} \cdot \vec{X} + \vec{b}, \tag{3.29}$$

where  $\mathbf{A}$ , and  $\vec{\mathbf{b}}$  are some parameters of the transformation. For transformation (3.29) the deformation gradient  $\mathbf{F}$  is

$$\mathbf{F} = \frac{d\vec{\mathbf{X}}}{d\mathbf{X}} = \frac{\partial x_i}{\partial X_j} \vec{\mathbf{e}}_i \vec{\mathbf{e}}_j = \frac{\partial(A_{ik}X_k + b_i)}{\partial X_j} \vec{\mathbf{e}}_i \vec{\mathbf{e}}_j = A_{ik} \delta_{kj} \vec{\mathbf{e}}_i \vec{\mathbf{e}}_j = \mathbf{A}. \quad (3.30)$$

Let us recall that the summation convention is used with the values of indices from 1 to 3 in the 3D case, and from 1 to 2 in the 2D case. In the 2D case, the vector and tensor objects have dimension 2.

We can write for the vertices of the element

$$\begin{cases} \vec{\mathbf{x}}_a = \mathbf{A} \cdot \vec{\mathbf{X}}_a + \vec{\mathbf{b}} \\ \vec{\mathbf{x}}_b = \mathbf{A} \cdot \vec{\mathbf{X}}_b + \vec{\mathbf{b}} \\ \vec{\mathbf{x}}_c = \mathbf{A} \cdot \vec{\mathbf{X}}_c + \vec{\mathbf{b}} \\ \vec{\mathbf{x}}_d = \mathbf{A} \cdot \vec{\mathbf{X}}_s + \vec{\mathbf{b}} \end{cases} \quad (3.31)$$

or

$$\begin{cases} \vec{\mathbf{x}}_1 = \mathbf{A} \cdot \vec{\mathbf{X}}_1 \\ \vec{\mathbf{x}}_2 = \mathbf{A} \cdot \vec{\mathbf{X}}_2 \\ \vec{\mathbf{x}}_3 = \mathbf{A} \cdot \vec{\mathbf{X}}_3 \end{cases} \quad (3.32)$$

Having solved system (3.32), we can find the tensor  $\mathbf{A}$

$$\mathbf{A} = \vec{\mathbf{x}}_i \vec{\mathbf{X}}^i, \quad (3.33)$$

where

$$\vec{\mathbf{X}}^1 = \frac{\boldsymbol{\varepsilon} \cdot \vec{\mathbf{X}}_3 \vec{\mathbf{X}}_2}{\boldsymbol{\varepsilon} \cdot \vec{\mathbf{X}}_3 \vec{\mathbf{X}}_2 \vec{\mathbf{X}}_1}, \quad \vec{\mathbf{X}}^2 = \frac{\boldsymbol{\varepsilon} \cdot \vec{\mathbf{X}}_1 \vec{\mathbf{X}}_3}{\boldsymbol{\varepsilon} \cdot \vec{\mathbf{X}}_3 \vec{\mathbf{X}}_2 \vec{\mathbf{X}}_1}, \quad \vec{\mathbf{X}}^3 = \frac{\boldsymbol{\varepsilon} \cdot \vec{\mathbf{X}}_2 \vec{\mathbf{X}}_1}{\boldsymbol{\varepsilon} \cdot \vec{\mathbf{X}}_3 \vec{\mathbf{X}}_2 \vec{\mathbf{X}}_1}. \quad (3.34)$$

The vectors  $\vec{\mathbf{X}}_i$  and  $\vec{\mathbf{X}}^i$  are called the reciprocal vectors (see more details in Appendix B).

For the 2D case,

$$\vec{\mathbf{X}}^1 = \frac{\boldsymbol{\varepsilon} \cdot \vec{\mathbf{X}}_2}{\boldsymbol{\varepsilon} \cdot \vec{\mathbf{X}}_2 \vec{\mathbf{X}}_1}, \quad \vec{\mathbf{X}}^2 = \frac{-\boldsymbol{\varepsilon} \cdot \vec{\mathbf{X}}_1}{\boldsymbol{\varepsilon} \cdot \vec{\mathbf{X}}_2 \vec{\mathbf{X}}_1}. \quad (3.35)$$

We can now introduce the approximation (3.8) for  $\mathbf{B}$  taking into account (3.30) and (3.33):

$$\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^T = \vec{\mathbf{x}}_s \vec{\mathbf{X}}^s \cdot \vec{\mathbf{X}}^k \vec{\mathbf{x}}_k = \vec{\mathbf{x}}_s \vec{\mathbf{e}}_s \cdot \vec{\mathbf{e}}_m \vec{\mathbf{X}}^m \cdot \vec{\mathbf{X}}^k \vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_j \vec{\mathbf{x}}_j = \mathbf{R}_{\vec{\mathbf{x}}}^T \cdot \mathbf{G}^{\vec{\mathbf{x}}} \cdot \mathbf{R}_{\vec{\mathbf{x}}}, \quad (3.36)$$

where  $\mathbf{G}^{\vec{\mathbf{x}}} = \mathbf{R}^{\vec{\mathbf{x}}} \cdot (\mathbf{R}^{\vec{\mathbf{x}}})^T$  is a symmetric tensor with the components  $\vec{\mathbf{X}}^m \cdot \vec{\mathbf{X}}^k$ ,  $\mathbf{R}_{\vec{\mathbf{x}}} = \vec{\mathbf{e}}_s \vec{\mathbf{x}}_s$ , and  $\mathbf{R}^{\vec{\mathbf{x}}} = \vec{\mathbf{X}}^s \vec{\mathbf{e}}_s$ . We will also use the tensors  $\mathbf{G}_{\vec{\mathbf{x}}}$ ,  $\mathbf{G}^{\vec{\mathbf{x}}}$  and  $\mathbf{G}_{\vec{\mathbf{x}}}$  with the components  $\vec{\mathbf{X}}_m \cdot \vec{\mathbf{X}}_k$ ,  $\vec{\mathbf{x}}^m \cdot \vec{\mathbf{x}}^k$  and  $\vec{\mathbf{x}}_m \cdot \vec{\mathbf{x}}_k$ , respectively.

Having defined  $\mathbf{B}$  in the element, we calculate  $\mathbf{B}_D$  and find the value of  $\mathbf{S}$  from equation (3.5). Finally, knowing  $p$  and  $\mathbf{S}$ , we calculate the stress tensor in the element from (3.4).

Let us assume that the initial and actual configurations of the element coincide up to translation and rotation, that is  $\mathbf{G}_{\bar{\mathbf{x}}} = \mathbf{G}_{\bar{\mathbf{x}}}$  and  $\mathbf{G}^{\bar{\mathbf{x}}} = \mathbf{G}^{\bar{\mathbf{x}}}$  (see (B.5)) and  $\bar{\mathbf{X}}^m \cdot \bar{\mathbf{X}}^k = \bar{\mathbf{x}}^m \cdot \bar{\mathbf{x}}^k$ . Obviously, in this case  $V = V_0$  and  $p = 0$  (see (2.6)). From (B.4) we have

$$\mathbf{B} = \mathbf{R}_{\bar{\mathbf{x}}}^T \cdot \mathbf{G}^{\bar{\mathbf{x}}} \cdot \mathbf{R}_{\bar{\mathbf{x}}} = \bar{\mathbf{x}}_s \bar{\mathbf{e}}_s \cdot \bar{\mathbf{e}}_m \bar{\mathbf{x}}^m \cdot \bar{\mathbf{x}}^k \bar{\mathbf{e}}_k \cdot \bar{\mathbf{e}}_j \bar{\mathbf{x}}_j = \mathbf{I} \quad (3.37)$$

As a result,  $\mathbf{B}_D = \mathbf{0}$ ,  $\mathbf{S} = \mathbf{0}$  and  $\boldsymbol{\sigma} = \mathbf{0}$ . It is important to note that for neo-Hooke's model based on the strain tensor  $\mathbf{B}$  as well as for all other models of higher order,  $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{B})$ , an arbitrary solid body rotation of the initial configuration does not change the value of the stress.

We can see that all the information about the initial configuration of an element is contained in  $\mathbf{G}^{\bar{\mathbf{x}}}$  and  $V_0$ . If we relieve the element from the external load,  $V$  will tend to  $V_0$  and  $\mathbf{G}^{\bar{\mathbf{x}}}$  will tend to  $\mathbf{G}^{\bar{\mathbf{x}}}$ . This important issue needs to be emphasized before we move to further discussion of the approximation technique.

Assume that the time moment  $t$  corresponds to the  $n$ -th time step, that is  $t = n\Delta t$ ,  $\bar{\mathbf{x}}^n = \bar{\mathbf{x}}$ ,  $V^n = V$ . On the basis of (2.5), (2.6) we can present the stress tensor in the element in the form

$$\boldsymbol{\sigma}^n = K \left( \frac{V^n}{V_0} - 1 \right) + \mu \left[ \mathbf{R}_{\bar{\mathbf{x}}^n}^T \cdot \mathbf{G}^{\bar{\mathbf{x}}} \cdot \mathbf{R}_{\bar{\mathbf{x}}^n} - \frac{1}{3} (\mathbf{G}_{\bar{\mathbf{x}}^n} \cdot \cdot \mathbf{G}^{\bar{\mathbf{x}}}) \mathbf{I} \right]. \quad (3.38)$$

Employment of the direct vector notation in equation (3.38) makes it possible to replace several pages of lengthy and redundant formulas (e.g. (2.13)) in [65, 67]. In addition, equation (3.38) is based on more general neo-Hooke's constitutive relations than that provided by Wilkins.

Therefore, the direct vector technique of approximation gives an equivalent formulation of the problem and provides a more efficient way to develop the computer code, which can allow one to reduce syntax errors both at the stage of developing the numerical scheme and at the stage of writing the actual code.

On the basis of (3.38), we can quickly design the explicit scheme. Let us consider a new position of the grid nodes at the  $n$ -th time step. Then for any element we can determine the set of vectors  $\bar{\mathbf{x}}_k^n$  and use formula (3.38) in order to find the stress tensor  $\boldsymbol{\sigma}^n$ .

The approximation for the nodal variables (see for example the approximation (2.14) for (3.1)) becomes:

$$M \frac{\bar{\mathbf{v}}^{n+1} - \bar{\mathbf{v}}^n}{\Delta t} = \oint_S \boldsymbol{\sigma}^n \cdot d\bar{\mathbf{s}}. \quad (3.39)$$

Having calculated the updated nodal velocities of the mesh from (3.39), we can find new positions of the nodes at the  $(n+1)$ -th time step.

We will finish this section with some remarks about programming of discretized equations taking (3.39) as example. In order to compute the right-hand side of this equation, it is not necessary to write the surface integral explicitly as a sum over all surfaces adjoining the node. First of all, we prepare the array corresponding to the nodes of the mesh with zeros in it. We compute the value of the integral in the right-hand side of (3.39) by parts. Consider,

for example, a part of the surface  $S$  in the element  $abcd$  (Figure 1). It consists of some parts of the control volumes for the nodes  $a$  and  $b$ . The force acting on the node  $a$  computed at the element  $abcd$  equals the force acting on the node  $b$  up to the sign. This allows us to reduce the number of operations. We organize a cycle with respect to all elements of the mesh. For each element, we compute the tensor  $\boldsymbol{\sigma}$ , find the area of the corresponding part of the element, and determine the vector normal to its surface. After that, we compute the vector  $\boldsymbol{\sigma} \cdot \vec{\mathbf{s}}$  and add it with  $+$  to the component of the array corresponding to the node  $a$ , and with  $-$  to the component  $b$ . Hence we do not need the explicit form of the approximation, we work with each element only once. To conclude, we can say that we work with mesh elements and not with mesh nodes.

Finally, we would like to stress it again that we have formulated the direct and compact vector form of approximation. Comparing it with the component-wise technique, we observe that contemporary achievements in programming tools should be used, and we should not write redundant and lengthy relations for each component (like (2.13), for example). From the mathematical point of view, the vector technique of approximation is equivalent to the traditional component-wise but from the user point of view the vector technique is several times more compact and, apart from the cleaner syntax, it is closer to the same visual appearance of initial equations.

### 3.3 Plasticity procedure

“All the possible stress states corresponding to yielding constitute a closed hypersurface in the stress space. The yield function can be written mathematically in the general form” [33]:

$$\begin{aligned} F(\mathbf{S}) &< K && \text{for elastic deformation domain,} \\ F(\mathbf{S}) &= K && \text{for plastic deformation domain.} \end{aligned} \tag{3.40}$$

Here  $K$  is a material constant. The Von Mises criterion (2.9) is a particular case of (3.40).

Taking into account equation (3.5) (or generally  $\mathbf{S} = \mathbf{S}(\mathbf{B})$ ) we can rewrite the condition (3.40) in the equivalent form which will serve as a basis for our future discussion;

$$\begin{aligned} f(\mathbf{B}) &< k && \text{for elastic deformation domain,} \\ f(\mathbf{B}) &= k && \text{for plastic deformation domain.} \end{aligned} \tag{3.41}$$

Here  $k$  is a material constant.

We start with a simple example. Let us consider an elastic-plastic spring with the initial length  $X$  and the actual length  $x$ . Assume that the force  $\sigma$  depends on  $x$  as

$$\sigma = E\left(\frac{x}{X} - 1\right), \tag{3.42}$$

and the plasticity condition is in the form of

$$|\sigma| \leq K, \tag{3.43}$$

where  $E$  and  $K$  are the material coefficients. From (3.42) it follows

$$\left| \frac{x}{X} - 1 \right| \leq k = \frac{K}{E}. \quad (3.44)$$

Let us solve the problem “numerically”, that is step by step with respect to time. Suppose that  $x$  increases by  $\Delta x$  at any time step  $\Delta t$ . At each time step we check condition (3.44). Assume that the new value of the actual length is  $x^{n+1}$ . We can call this stage of the algorithm predictor step. If condition (3.44) remains true, the deformation remains elastic. This means that once the force is removed, the spring returns back to its initial length  $x^{n+1} \rightarrow X$ .

Suppose that at some time step condition (3.44) is not satisfied any more. Therefore plastic deformation begins, and the deformation becomes irreversible, i.e., the spring will not return to its initial length when we remove the external force. Hence we need to change the initial length  $X$ , which should also depend on  $n$  and be denoted by  $X^n$  (we note that for elastic deformations  $X^{n+1} = X^n$ ). We can call this step a corrector step. It is clear that the new initial length  $X^{n+1}$  should be between  $X^n$  and  $x^{n+1}$ . In fact, if we consider that  $X^{n+1} = X^n$ , then condition (3.44) is violated but if we assume that  $X^{n+1} = x^{n+1}$ , then condition (3.44) is satisfied with the equality. Therefore, we will look for  $X^{n+1}$  within the interval  $X^n < X^{n+1} < x^{n+1}$  in the form

$$X^{n+1} = (a - 1)X^n + ax^{n+1}, \quad (3.45)$$

where  $a$  is a parameter,  $0 < a < 1$ . Let us substitute relation (3.45) in the condition (3.44) and require that the following equality is satisfied:

$$\frac{x^{n+1}}{(1 - a)X^n + ax^{n+1}} = 1 + k. \quad (3.46)$$

Then

$$a = \frac{x^{n+1} - (1 + k)X^n}{(1 + k)(x^{n+1} - X^n)}. \quad (3.47)$$

Therefore from (3.45)

$$X^{n+1} = \frac{x^{n+1}}{1 + k}. \quad (3.48)$$

The correction of the initial length is done and relation (3.44) is satisfied accurately. Hence, we can proceed to the next time step. Obviously, equation (3.48) follows from (3.46) directly. However, we have intentionally performed all the calculations in the way as it should be done for the 3D case.

Let us summarize the previous paragraph. At each time step where (3.44) is not satisfied we adjust the initial length. The old value of the initial length  $X^n$  is “forgotten” by the spring. At this point, the difference  $X^{n+1} - X^n$  is equal to the plastic (irreversible) deformation. If we remove the force, the spring returns to its new initial length  $X^{n+1}$ . We should emphasize that the meaning of the expression “initial length” has been changed here. It is not the



length at the initial moment of time any more, but the length to which the spring returns if at some moment of time we remove the external force. Hence, the initial length defined in this way depends on time.

Let us now return to the 3D elastic-plastic problem and consider the deformation of an element. As it was discussed in Section 3.2, the variables  $V_0$  and  $\mathbf{G}^{\bar{\mathbf{x}}}$  play the role of the memory of the element. Taking into account equation (3.36) we can conclude that  $\mathbf{G}^{\bar{\mathbf{x}}}$  can be determined from (3.41). In what follows we will pay all the attention only to this characteristic. Let us recall that the tensor  $\mathbf{G}^{\bar{\mathbf{x}}}$  contains the full information about the initial configuration of the element up to translation and rotation (see Appendix B).

We note that a more general case can be considered where the value  $V_0$  is also included in the yield function and depends on time,  $V_0^n$ , but for the current analysis we will restrict ourselves to the case (3.41). We will also consider only the Von Mises yield criterion (3.13) as a specific case of (3.41).

Condition (3.13) has the same physical meaning as the 1D condition (3.44) on the restriction of the geometrical difference between the initial and actual configurations of the element, which can differ one from another only up to a certain limit. If we keep changing the actual configuration, then we will inevitably have to change the initial configuration which is an equivalent of the plastics flow initiation. Let us rewrite the condition (3.13) in the form

$$-J_2(\mathbf{B}_D) \leq \frac{4\varepsilon_s^2(1+\nu)^2}{3}, \quad (3.49)$$

where  $\varepsilon_s = \sigma_s/E$  is the limit of the elastic strain,  $\nu$  is Poisson's ratio, and  $E = 2\mu(1+\nu)$  is Young's modulus.

In the elastic regime (3.49) holds and  $\mathbf{G}^{\bar{\mathbf{x}}}$  does not change. If the deformation of the element exceeds the critical value and condition (3.49) is not satisfied any more, the internal structure of the solid body is irreversibly changed. In other words, the initial configuration of the element, and, consequently, the tensor  $\mathbf{G}^{\bar{\mathbf{x}}}$  should be changed to satisfy the condition

$$-J_2(\mathbf{B}_D) = \frac{4\varepsilon_s^2(1+\nu)^2}{3}. \quad (3.50)$$

Let us go through the whole sequence of calculations at some time step when the actual configuration of the element is changed. Let us calculate the new set of vectors  $\bar{\mathbf{x}}_s^{n+1}$  (3.14) at the  $(n+1)$ -th time step. Then we fulfill the predictor step and define

$$\mathbf{B}' = \mathbf{R}_{\bar{\mathbf{x}}^{n+1}}^T \cdot \mathbf{G}^{\bar{\mathbf{x}}^n} \cdot \mathbf{R}_{\bar{\mathbf{x}}^{n+1}}, \quad (3.51)$$

where  $\mathbf{R}_{\bar{\mathbf{x}}^{n+1}} = \vec{\mathbf{e}}_s \bar{\mathbf{x}}^{n+1}$  and  $\mathbf{G}^{\bar{\mathbf{x}}^n}$  is known from the previous time increment. If the condition

$$-J_2(\mathbf{B}'_D) \leq \frac{4\varepsilon_s^2(1+\nu)^2}{3} \quad (3.52)$$

is satisfied, then the tensor  $\mathbf{G}^{\bar{\mathbf{x}}^n}$  does not change, i.e.  $\mathbf{G}^{\bar{\mathbf{x}}^{n+1}} = \mathbf{G}^{\bar{\mathbf{x}}^n}$  and we proceed to the next time step.

Suppose that (3.52) does not hold. Then we should find a new tensor  $\mathbf{G}^{\bar{\mathbf{x}}^{n+1}}$  for the element such that for  $\mathbf{B}^{n+1} = \mathbf{R}_{\bar{\mathbf{x}}^{n+1}}^T \cdot \mathbf{G}^{\bar{\mathbf{x}}^{n+1}} \cdot \mathbf{R}_{\bar{\mathbf{x}}^{n+1}}$  the following condition would be satisfied:

$$-J_2(\mathbf{B}_D^{n+1}) = \frac{4\varepsilon_s^2(1+\nu)^2}{3}. \quad (3.53)$$

Obviously, the searched value of  $\mathbf{G}^{\bar{\mathbf{x}}^{n+1}}$  should be in the range between  $\mathbf{G}^{\bar{\mathbf{x}}^n}$  and  $\mathbf{G}^{\bar{\mathbf{x}}^{n+1}}$ . Similar to the example above, we introduce the parameter  $a$  and put

$$\mathbf{G}^{\bar{\mathbf{x}}^{n+1}} = a\mathbf{G}^{\bar{\mathbf{x}}^n} + (1-a)\mathbf{G}^{\bar{\mathbf{x}}^{n+1}}. \quad (3.54)$$

Using (3.54), we can write

$$\mathbf{B}^{n+1} = \mathbf{R}_{\bar{\mathbf{x}}^{n+1}}^T \cdot \left[ a\mathbf{G}^{\bar{\mathbf{x}}^n} + (1-a)\mathbf{G}^{\bar{\mathbf{x}}^{n+1}} \right] \cdot \mathbf{R}_{\bar{\mathbf{x}}^{n+1}} = a\mathbf{B}' + (1-a)\mathbf{I} \quad (3.55)$$

and then

$$\mathbf{B}_D^{n+1} = a\mathbf{B}' + (1-a)\mathbf{I} - \frac{1}{3} [aJ_1(\mathbf{B}') + 3(1-a)] \mathbf{I} = a\mathbf{B}'_D. \quad (3.56)$$

Then from (3.56)

$$J_2(\mathbf{B}_D^{n+1}) = a^2 J_2(\mathbf{B}'_D). \quad (3.57)$$

We would like to satisfy the condition (3.53). Then substituting (3.57) into (3.53), we obtain

$$a = \frac{2\varepsilon_s(1+\nu)}{\sqrt{-3J_2(\mathbf{B}'_D)}}. \quad (3.58)$$

Substitution of  $a$  from (3.58) into (3.54) gives the new value of the tensor  $\mathbf{G}^{\bar{\mathbf{x}}^{n+1}}$  in the element. Then condition (3.53) is satisfied. Obviously, condition (2.9) also holds.

Plasticity theories assume that during plastic flow the rate of plastic strain is proportional at any instant to the stress deviator  $\mathbf{S}$ . The next assumption follows from (3.56) and (3.5):

$$d\mathbf{B}_p^{n+1} = \mathbf{B}_D^{n+1} - \mathbf{B}'_D = -(1-a)\mathbf{B}'_D = \frac{a-1}{a}\mathbf{B}_D^{n+1} = \frac{a-1}{\mu a}\mathbf{S}^{n+1}. \quad (3.59)$$

Thus, in the general case, at any time step, we deal with three configurations of the element: the actual one which is based on the set of vectors  $\bar{\mathbf{x}}_s^{n+1}$  which enable us to build the tensor  $\mathbf{G}^{\bar{\mathbf{x}}^{n+1}}$  (and tensor  $\mathbf{G}_{\bar{\mathbf{x}}^{n+1}} = (\mathbf{G}^{\bar{\mathbf{x}}^{n+1}})^{-1}$ ), “old” initial configuration based on  $\mathbf{G}^{\bar{\mathbf{x}}^n}$ , and “new” initial configuration based on  $\mathbf{G}^{\bar{\mathbf{x}}^{n+1}}$ . We need to know relative positions of these three configurations in order to determine both the translation and rotation.

The difference between  $\mathbf{G}^{\bar{\mathbf{x}}^{n+1}}$  and  $\mathbf{G}^{\bar{\mathbf{x}}^n}$  satisfies the critical condition (3.53). The difference between  $\mathbf{G}^{\bar{\mathbf{x}}^{n+1}}$  and  $\mathbf{G}^{\bar{\mathbf{x}}^n}$  configurations is the plastic (irreversible) deformation (Figure 2).

We will illustrate this discussion with a simple example. Let us consider a cylinder with material properties  $\nu = 0.45$  and  $E = 100$ , GPa subjected to a uniaxial cyclic loading. We

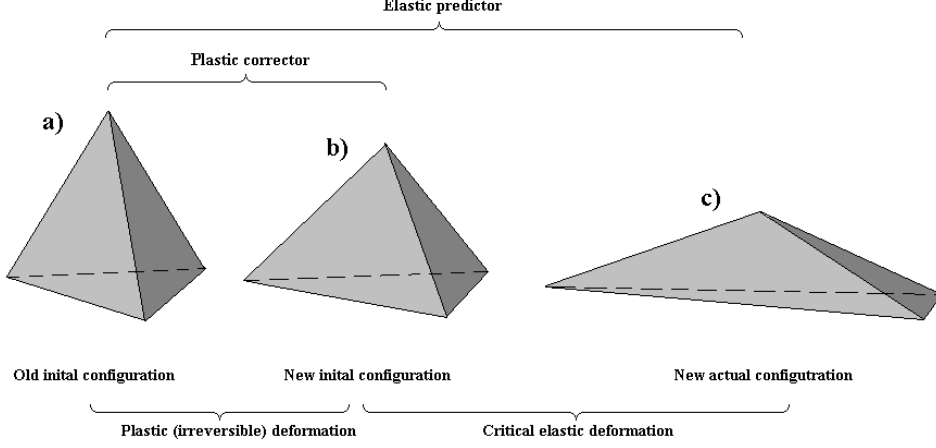


Figure 2: Diagram of elastic predictor and plastic corrector.

will assume that sliding is possible at the edges of the cylinder. Even though this problem reduces to 1D, we will use a 3D mesh for the analysis.

Cells of the mesh have tensile strain in the longitudinal direction and compression strain in the transverse direction.

One cycle for the perfectly elastic-plastic model with no hardening of the material is shown in Figure 3. Here  $\varepsilon_s = 0.01$  which corresponds to  $\sigma_s = 1$ , GPa;  $ab$  is the elastic part. When  $\varepsilon > \varepsilon_s = 0.01$ , plastic flow occurs along  $bc$ , where  $\sigma = \sigma_s$ . The unloading  $cd$  and loading  $ef$  parts are the same. This analysis coincides with the simulation which can be done employing Wilkins' method.

We can also take into account that the hardening of the material  $\varepsilon_s(d\varepsilon_p)$  depends on the plastic deformation  $d\varepsilon_p$ . Let the sets of vectors  $\vec{\mathbf{X}}_1^{n+1}, \vec{\mathbf{X}}_2^{n+1}, \vec{\mathbf{X}}_3^{n+1}$  and  $\vec{\mathbf{X}}_1^n, \vec{\mathbf{X}}_2^n, \vec{\mathbf{X}}_3^n$  correspond to the new and old initial configurations, respectively. In order to define the distortion of the initial configurations of the element, we can introduce the tensor

$$\mathbf{B}_p^{n+1} = (\mathbf{R}_{\vec{\mathbf{X}}}^T)^{n+1} \cdot (\mathbf{G}^{\vec{\mathbf{X}}})^n \cdot \mathbf{R}_{\vec{\mathbf{X}}}^{n+1} \quad (3.60)$$

and define the strain increment in the form

$$d\varepsilon_p = \frac{1}{\sqrt{3}} \sqrt{-J_2(\mathbf{P}_D^{n+1})}, \quad (3.61)$$

where  $\mathbf{P}^{n+1} = \mathbf{B}_p^{n+1} - \mathbf{I}$ . We can transform (3.61) to the following form

$$d\varepsilon_p = \frac{1}{\sqrt{18}} \sqrt{3 \left( \mathbf{G}_{\vec{\mathbf{X}}^{n+1}} \cdot \mathbf{G}_{\vec{\mathbf{X}}^n} \right) \cdot \left( \mathbf{G}_{\vec{\mathbf{X}}^{n+1}} \cdot \mathbf{G}_{\vec{\mathbf{X}}^n} \right) - \left( \mathbf{G}_{\vec{\mathbf{X}}^{n+1}} \cdot \mathbf{G}_{\vec{\mathbf{X}}^n} \right)^2} \quad (3.62)$$

Let us assume for simplicity that

$$\varepsilon_s = \varepsilon_s + \varepsilon_s' d\varepsilon_p \quad (3.63)$$

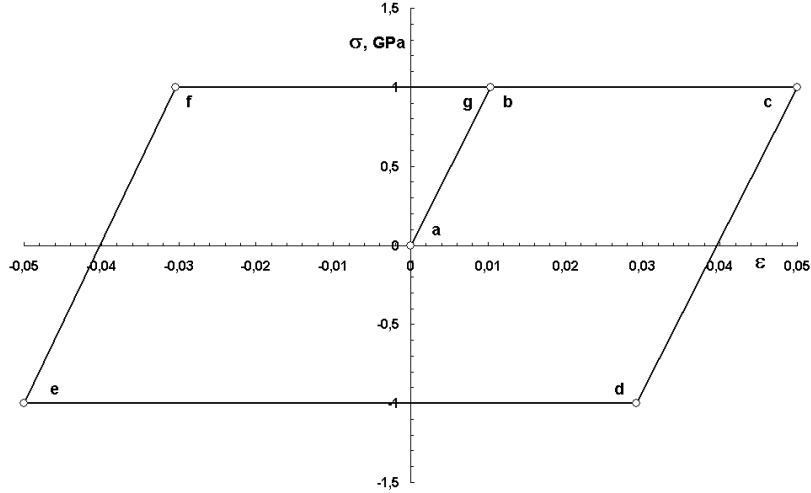


Figure 3: One cycle for the perfectly elastic-plastics model for  $\varepsilon_s = 0.01$ ,  $\nu = 0.45$  and  $E = 100$ , GPa.

One cycle for the elasto-plastic model with quasi-hardening is shown in Figure 4 for  $\varepsilon_s = 0.01$ ,  $\varepsilon'_s = 0.1, \text{m}^{-1}$ . The part  $bc$  of the curve corresponds to the plastic deformation with hardening. However, during the next cycle we will have unrestrained hardening of the material. This does not correspond to the physics of the problem and, therefore, we proceed to the multi-axial model.

Before doing this, let us first compare our method with Wilkins' method. In order to compare the results of modelling obtained with both methods, let us consider the following simple problem. We take a 2D Cartesian mesh, which consists of a single square cell with surface  $S_0$  and which includes four triangular elements (Figure 5a).

The material properties in this example are the same as in the previous example. The time step was the same in the both cases.

The stretching load is applied to the solid in the horizontal direction and then removed at the time moment of 0.01 sec (Figure 5b).

Obviously, the solid body starts free vibrations, and the actual area of the solid  $S$  will periodically change around its initial value  $S_0$ . A rapid change of compression and tension will take place correspondingly, and at the moment when  $S$  becomes equal to  $S_0$ , the value of  $p$  should be equal to zero. The shear stresses should be equal to zero at any time. The shape of the solid should stay rectangular. This is the expectation of how the solid body should behave in this example.

Now, let us review the results of the simulation. As it was expected, the calculated values of the pressure in the solid, after the load was removed, changed periodically around zero with the same amplitude: the values of pressure  $p$  obtained by two different methods are practically identical (Figure 6a). The value of  $\sigma_{xy}$  was equal to zero, as it was expected.

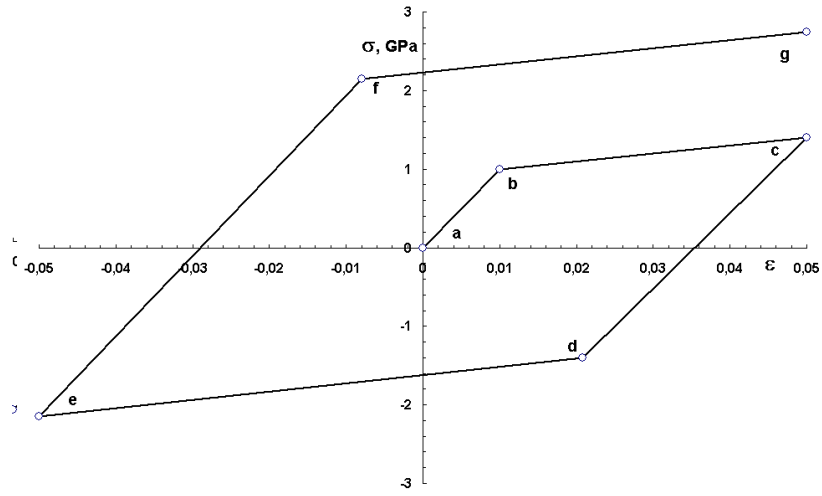


Figure 4: One cycle for the elastic-plastics model with quasi-hardening for  $\varepsilon_s = 0.01$ ,  $\varepsilon'_s = 0.1$ ,  $\nu = 0.45$  and  $E = 100$ , GPa.

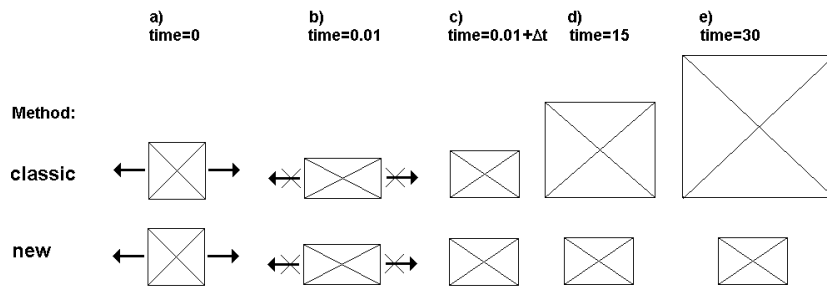


Figure 5: Shape of the solid vs time under tension parallel to the  $x$  axis.

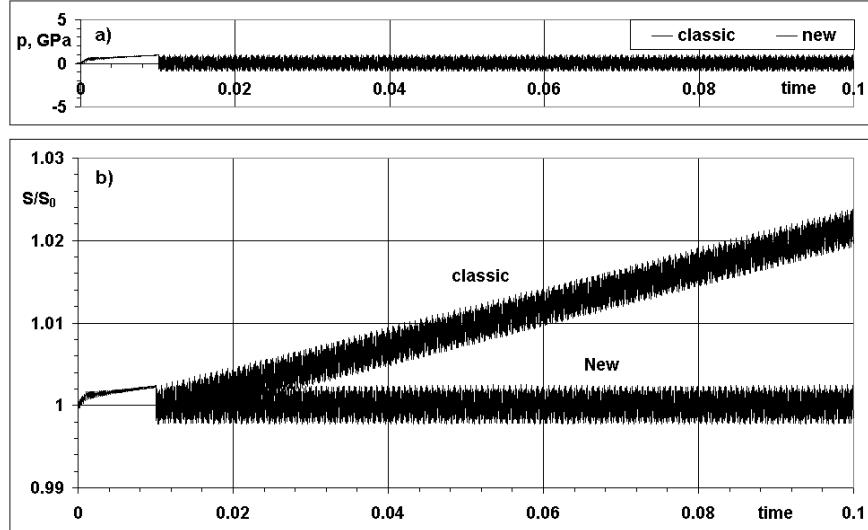


Figure 6: Simple tension. Variation of a) pressure  $p$  and b)  $S/S_0$

We now compare the ability of the two methods to predict the evolution of the shape of the solid during a substantial period of time. After relieving the external load, the solid took the shape of a rectangle slightly elongated in the horizontal direction. Then the shape stayed rectangular, and the solid rapidly vibrated around some average position.

During relatively short period of time after the load removal, the condition  $S = S_0$  is satisfied for both methods (Figure 5). However, for a longer period of time, the difference between the two methods becomes more and more visible. In the case of the new method, the value of  $S/S_0$  oscillates around 1, while in case of the classical method,  $S/S_0$  oscillates with the same amplitude but around the growing value (Figure 6b). The shapes of the solid obtained with both methods at the moments of 15 and 30 sec are shown in Figure 5d, e.

The results obtained with the classical method show that the average area of the solid is growing with a constant velocity, and the effect of “swelling” of the mesh was observed. This effect can be suppressed by reducing the time integration step but can not be eliminated completely. The process of relaxation of the solid after the external load is removed can be sufficiently long. Therefore, the decrease of the time step is strongly undesirable. The new method is stable with respect to the “swelling” of the mesh.

This effect is determined by the fundamental difference between the classical and the new methods: the calculation of the pressure at the new time step  $p^{n+1}$  in the classical method is conducted according to formula (2.11), while in the new method according to formula (3.6) where the initial volume  $V_0$  of the cell (or the initial area  $S_0$  in the 2D example) is included. As a result, according to formula (3.6), if the equation  $V = V_0$  (or  $S = S_0$ ) is violated, the counter pressure is generated. It tends to change the shape of the solid in such a way that the equation  $V = V_0$  (or  $S = S_0$ ) is restored.

When we calculate the pressure with the classical method according to formula (2.11),

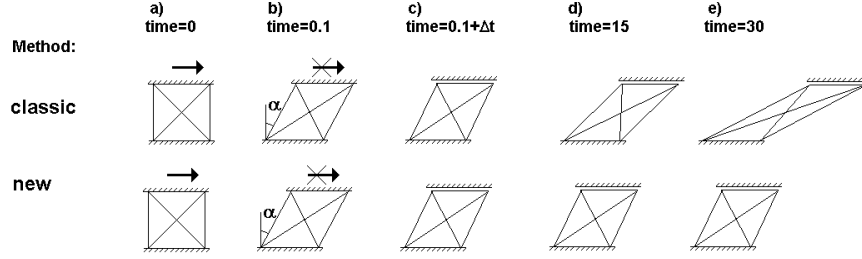


Figure 7: Shape of the solid vs time with simple shear.

$p^{n+1}$  is found as a result of adding the pressure increment  $\Delta t K \nabla_{\vec{x}} \cdot \vec{v}$  to its value from the previous integration step  $p^n$ . Therefore, after a large number of integration steps, due to the accumulation of integration error, the information about the initial volume (area) of the cell is gradually “forgotten” and, as a result, the observed non-physical and unlimited “swelling” of the mesh takes place.

Let us carry out a similar comparison for the same mesh and for the same properties and configuration in the case of a simple shear. The initial shape and loading conditions are shown in Figure 7a, where  $\alpha$  is the angle of the side deviation from the  $y$  axis.

We can expect that when the load is removed, the angle  $\alpha$  will gradually decrease, and its value will oscillate around some fixed average value  $\alpha_0$ . The pressure  $p$  should remain zero, and the stress tensor component  $\sigma_{xy}$  should vary around its average value equal to zero. The shape of the solid should be a parallelogram.

The results of the calculations of the value  $\sigma_{xy}$ , obtained with the help of both methods, are practically identical. The value of  $\sigma_{xy}$  varies periodically with consistent amplitude (Figure 8a). The value of pressure at any moment of time is equal to zero.

As in the previous example, let us follow the evolution of the shape of the solid after a substantial interval of time. The value of the angle of inclination of the parallelogram  $\alpha$ , obtained with the new method varies around its steady average value  $\alpha_0 \approx 27^\circ$ .

The value of  $\alpha_0$ , obtained with the help of the classical method is gradually increasing, as shown in Figure 8b. Additional calculations show that the growth of  $\alpha_0$  in the classical method can be decreased by decreasing the time step.

The difference between the two methods (for the same mesh and time step) is related to the fact that the calculation of the components of the tensor  $\mathbf{S}$  according to the new method employs equations (3.5), which take into account the tensor of the initial configuration  $\mathbf{G}^{\vec{x}}$ . In this example, the area  $S$  of the solid is constant according to the formulation of the problem, and behavior of the solid is determined by the shear strains. In this case, any deviation of the actual configuration of the cell from its initial configuration immediately generates opposite stresses which tend to return the actual shape of the cell to its original shape. The information about the initial configuration is kept in the tensor  $\mathbf{G}^{\vec{x}}$ .

In the case of the classical method, calculation of the tensor  $\mathbf{S}$  at the new time step

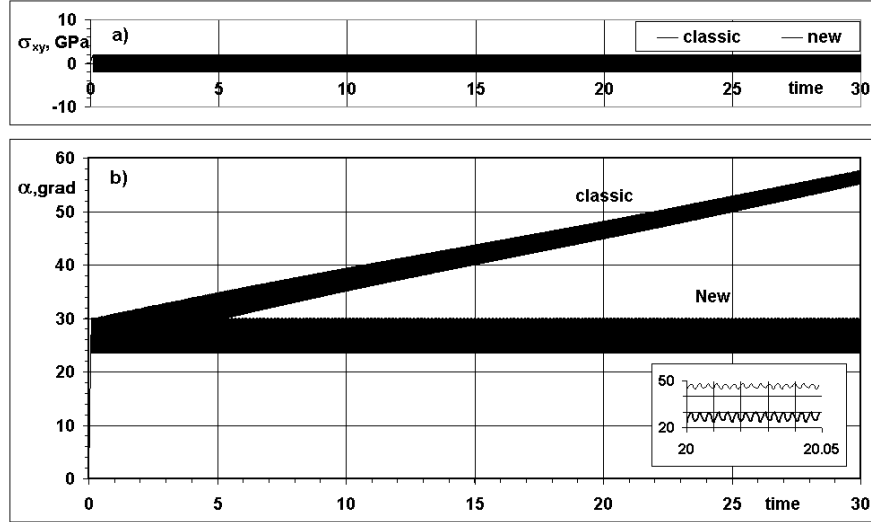


Figure 8: Simple shear. a) Variation of  $\sigma_{xy}$  and b) angle of inclination  $\alpha$

is conducted according to formula (2.10) which does not include directly the information about the initial and actual configurations, unlike the new method. This leads to gradual error accumulation for large number of integration steps. In this example, it results in a non-physical unbounded growth of the angle  $\alpha_0$  after removing the load.

Introduction of the artificial viscosity can suppress the high frequency oscillations in both methods but the problem of slow drift of  $S_0$  or  $\alpha_0$  when using the classical method can still appear.

Stability of the new method with respect to “swelling” of the mesh is a useful property. This method was used to solve a number of practical problems in [12], [24]-[25].

Now, after having proved that our method has some advantage in comparison with Wlknins’ method from the point of view of its stability, let us apply it for the multi-axial model of plasticity.

It is well known that the multi-axial generalization of the model of the stress space can be obtained using several multi-yield surfaces, i.e. instead of (3.40) we consider the following condition:

$$\begin{aligned} F_i(\mathbf{S} - \boldsymbol{\alpha}_i) &< K_i && \text{for elastic deformation domain,} \\ F(\mathbf{S} - \boldsymbol{\alpha}_i) &= K_i && \text{for plastic deformation domain,} \\ i &= 1, 2, \dots \end{aligned} \quad (3.64)$$

(see [38], [33] for more details). In (3.64) the restriction on the stress field is introduced by means of the multi-yield surface model.

In our case, where we can separate and if necessary change the initial configuration, we can assume that the medium is not homogeneous. For example we can consider a polycrystal structure with different sizes and properties of individual crystals. So we can assume that



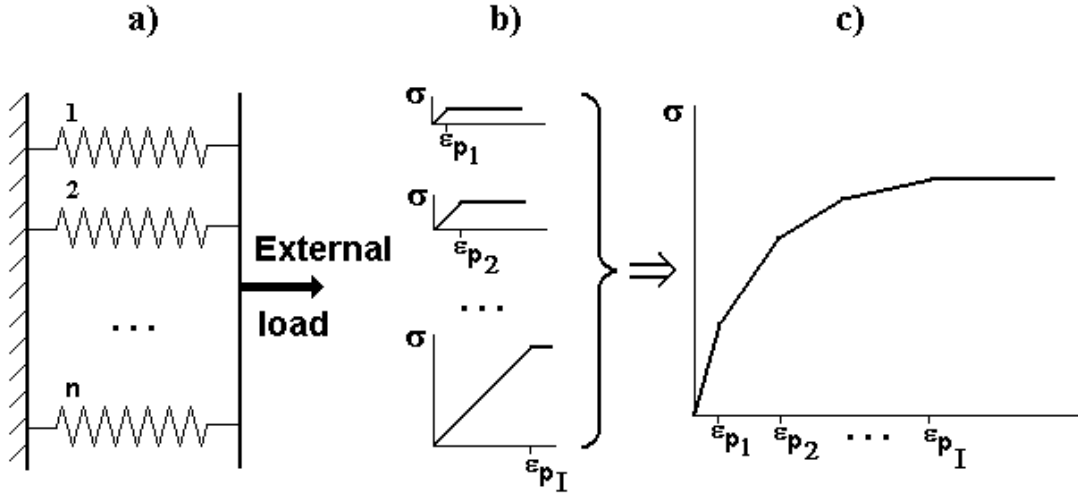


Figure 9: Scheme of multilevel elastic-plastics model.

plastic (irreversible) deformation for different parts of the structure starts at different strain levels.

Let an  $i$ -th component behave according to the perfectly elastic-plastic model due to the irreversible character of the deformation of this component. In the general case the corresponding value of  $\varepsilon_i$  can be different from other values.

Let us consider a model example. We consider a structure which consists of  $n$  perfectly elastic-plastic springs (Figure 9a) with

$$\varepsilon_{p1} < \varepsilon_{p2} < \dots < \varepsilon_{pn}. \quad (3.65)$$

We apply an external tensile load. If the total deformation of the system  $\varepsilon$  is less than  $\varepsilon_{p1}$ , then it behaves as a single spring. If we remove the external load, the system will return back to its initial configuration. If the load becomes sufficiently large, then the first spring turns to the elastic-plastic regime while the others are still in the elastic stage. The strain-stress dependence changes the slope. Then the next spring turns to the plastic regime (Figure 9b), and so on up to the last one, when finally all the springs will switch to the plastic regime and the external load will become constant. If we relieve the external load, the system will not return to its initial length. The residual stresses will stay inside the system with no external load applied. These residual stresses can be determined as a superposition of those for each spring.

Summarizing the above paragraph, we can replace (3.42) for this model problem by

$$\sigma = \sum_{i=1}^n \sigma_i = \sum_{i=1}^n E_i \left( \frac{x}{X_i} - 1 \right), \quad (3.66)$$

Here the total force  $\sigma$  is the sum of forces  $\sigma_i$  for each spring with its  $E_i$ . For all springs the

actual configuration  $x$  is the same but the initial configurations  $X_i$  in the general case can be different.

Instead of one condition (3.44), we have the system of conditions

$$\left| \frac{x}{X_i} - 1 \right| \leq k_i, \quad i = 1, 2, \dots, n. \quad (3.67)$$

Let us return to the continuous model. In the multi-level case we can write:

$$\mathbf{S} = \sum_{i=1}^n \mathbf{S}_i = \sum_{i=1}^n \mu_i \mathbf{B}_{iD} \quad (3.68)$$

$$\mathbf{B}_i = \mathbf{R}_{\bar{\mathbf{x}}}^T \cdot \mathbf{G}_i^{\bar{\mathbf{x}}} \cdot \mathbf{R}_{\bar{\mathbf{x}}}, \quad (3.69)$$

Instead of condition (3.49) we have now

$$-J_2(\mathbf{B}_{iD}) \leq \frac{4\varepsilon_{si}^2(1+\nu)^2}{3}, \quad i = 1, 2, \dots \quad (3.70)$$

A test simulation for a two-level model is shown in Figure 10. Here  $\varepsilon_{s1} = 0.01$ ,  $\varepsilon_{s2} = 0.02$ ,  $\mu_1 = \mu_2 = 1/2\mu$ ;  $ab$  is an elastic region;  $bc$  - first component is in a plastic regime, the second one is in a elastic regime;  $cd$  both components are under a plastic flow;  $de$  both components are elastic. We consider here both, work hardening and Bauschinger effect. The latter signifies that  $|\sigma_d| > |\sigma_e|$ .

It is clear that the two-level model is not sufficiently accurate and does not provide a smooth strain-stress dependence. For practical cases, the simulations can be performed with a more significant number of layers.

The results of the simulations for a 5-level model with  $\varepsilon_{s1} = 0.01$ ,  $\varepsilon_{s2} = 0.02$ ,  $\varepsilon_{s3} = 0.03$ ,  $\varepsilon_{s4} = 0.04$ ,  $\varepsilon_{s5} = 0.05$   $\mu_1 = \dots = \mu_5 = 1/5\mu$  are shown in Figure 11. The hardening part of the curve  $bcdef$  is more realistic than that one in Figure 10. The segment  $ab$  corresponds to the elastic regime for all components. The segments  $bc$ ,  $cd$ ,  $de$ ,  $ef$ ,  $fg$  correspond to the case where the components switch to the regime of plastic flow one after another in accordance with (3.70).

### 3.4 Contact algorithm

Numerical modeling of elastic-plastic problems becomes essentially more complex in the case of the interaction of two bodies or of two parts of the same body. The numerical meshes corresponding to these bodies are superimposed. To separate them, some formal geometrical rules are used though they are not uniquely defined and they are not justified physically. Moreover, they become extremely complicated technically for three-dimensional problems. The most widely used method of modeling contact interaction is based on the geometrical analysis of mutual position of boundary nodes of each mesh. At every integration step, we verify whether a boundary node has penetrated through the certain element of the surface

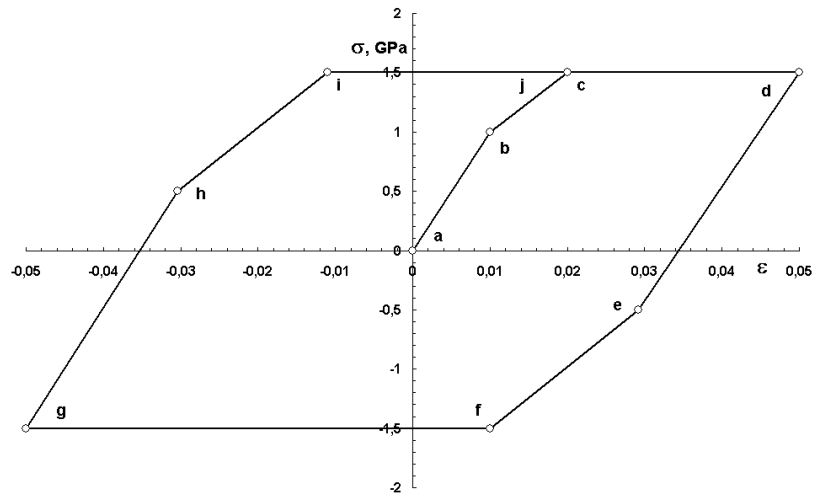


Figure 10: One cycle 2-level elastic-plastics model with  $\varepsilon_{s1} = 0.01$ ,  $\varepsilon_{s2} = 0.02$ ,  $E_1 = E_2 = E/2\text{GPa}$ ,  $E = 100\text{GPa}$ , and  $\nu = 0.45$ .  
<http://teramult.org.ua/>

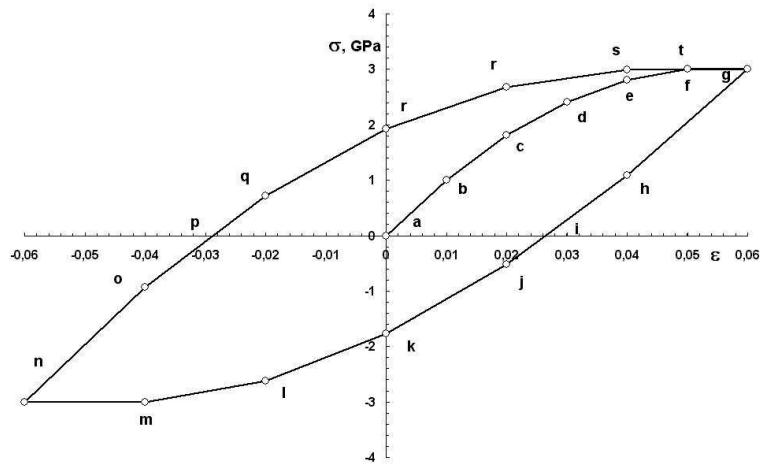


Figure 11: One cycle for 5-level elastic-plastics model with  $\varepsilon_{s1} = 0.01$ ,  $\varepsilon_{s2} = 0.02$ ,  $\varepsilon_{s3} = 0.03$ ,  $\varepsilon_{s4} = 0.04$ ,  $\varepsilon_{s5} = 0.05$   $\mu_1 = \dots = \mu_5 = 1/5\mu$ .

mesh of the obstacle. If it happens, we have to make certain corrections bringing the node back to the surface of the obstacle and letting the node slide on obstacle's surface instead of penetrating through this surface. A significant downside of this approach is in occasional penetration of the node through the surface. It usually happens due to the insufficient accuracy of the calculations. As soon as the node penetrates into the obstacle's surface, it is unable to return back.

Several modifications of the geometrical contact algorithm were developed, including various situations of mutual positions of the node and the surface element, with which given node is interacting. These modifications were able to resolve the issue for some practical problems. However, we could not be completely confident that the node would not penetrate through the surface in some other cases. The contact algorithm had to be corrected many times in order to prevent the penetration. In the case of simulations of a dynamic process employing an explicit integration procedure, we had to run millions of integration steps. Some nodes could come into contact, then move separately, and then contact again. In some cases the algorithm could give erroneous results: some node could come close to the obstacle but did not touch it but the algorithm considered it as contact interaction.

We will describe here another approach where the contact algorithm is based on the "mild" contact used in particular in molecular dynamics where the motion of particles is determined by the interaction potential. If they are too close to each other, then the repulsion between them pushes them away. This force is in the inverse proportion to the distance between molecules.

Let us discuss some important features of the mild contact and what is the mild contact in general. Mild contact is some variable boundary of an unknown shape where we have to satisfy the nonpenetration condition and equilibrium of forces between both surfaces. The classical contact algorithm is based on the geometrical analysis of mutual positions of the surfaces and it is aimed to prevent their penetration. The definition of the forces represents a separate issue to be resolved. Let us discuss the mild contact based on short distanced repulsion forces between two surfaces. Evidently, in this case we can provide both geometrical non-penetration and force balance. This idea is based on the introduction of acting-in-vicinity forces repelling the surfaces to be in contact. As a result, the surfaces do not come in contact but stay at some very small distance from each other. In addition, the algorithm and program code of mild contact simulation are much simpler than the geometrical algorithm.

From our point of view, this algorithm deserves much wider employment in the solid state mechanics, especially in the case of surfaces undergoing strong deformations. We suppose that two mesh elements interact by means of a repulsion force when they approach each other. The force is localized in a small neighborhood of the mesh elements, and it grows to infinity when the distance between them is approaching zero. In other words, when two bodies touch each other, the force between them is localized on their joint surface (Figure 12). Mathematically it can be considered as a measure localized at the surface. We approximate this measure by continuous functions with the support close to the surface.

A diagram defining contact variables is given in Figure 12 (2D case, for simplicity). Here

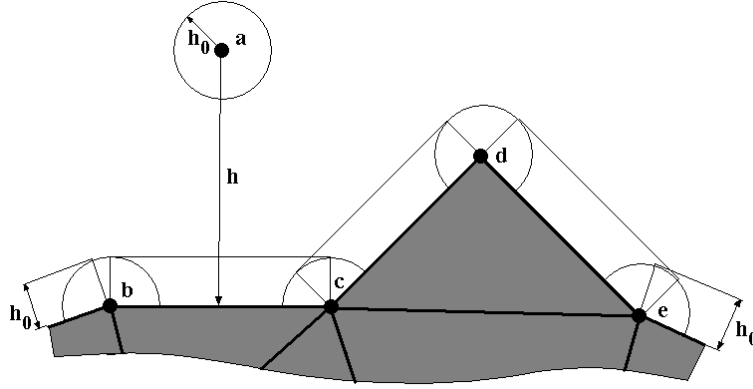


Figure 12: Scheme of interaction of a node with a boundary element of the mesh.

$b, c, d, e$  are the boundary nodes of the first mesh. The boundary of the second mesh is represented by the node  $a$ .

There are different possible realizations of the mild contact algorithm. All of them provide impenetration of the boundary mesh nodes and the equilibrium of forces during the contact.

We discuss one of possible realization. A node  $a$  can be repulsed from the nodes  $b, c, d, e$  of the other mesh, or from the sides  $b - c, c - d, d - e$  (in the 3D case from the nodes, sides and faces of the elements (Figure 13)). In the first case, we measure the distance  $h$  from  $a$  to the nearest node of the other mesh, in the second case, the distance from  $a$  to the nearest side of the other mesh. The absolute value of the force  $\vec{f}$  acting on the node  $a$  from another node or a side is given by

$$|\vec{f}| = \begin{cases} c \left( \frac{1}{h} - \frac{1}{h_0} \right), & \text{if } h < h_0 \\ 0, & \text{if } h \geq h_0 \end{cases}, \quad (3.71)$$

where  $h_0$  is the width of the zone where the force is different from zero. The force  $\vec{f}$  is directed along  $h$  (see Figure 13).

When the force is found from (3.71), we add it to the other forces acting on the node  $a$  and to those acting on the side  $b - c$  (with the opposite sign). The force acting on  $b - c$  is distributed proportionally between the nodes  $b$  and  $c$ . If the node  $a$  is in the region where several objects act on it (for example in the vicinity of the nodes  $c$  or  $e$ ), then we find the sum of forces acting on it.

The choice of function (3.71) is to some extent arbitrary if it provides a repulsing force that grows to infinity as  $h$  tends to zero. We can use for example a power dependence  $c(1/h^n - 1/h_0^n)$ , where  $n = 2, 3, \dots$ . The coefficient  $c$  should be found from model simulations. The width  $h_0$  should be taken several times less than the characteristic size of the mesh.

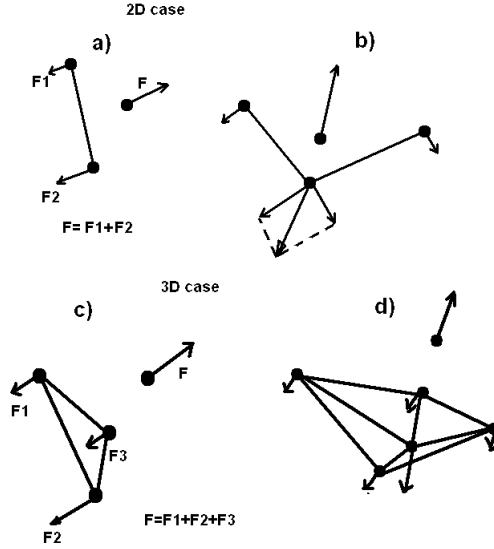


Figure 13: Forces between a node and a boundary element.

We note that this algorithm allows us to take into account the Coulomb friction. Indeed, when we compute the interaction of the node with an object of the other mesh, we know their mutual positions and velocities. It is sufficient to find the friction directed along the tangent to the surface and opposite to the tangential part of the relative velocity vector.

The formulas used to find the mutual positions of the node  $a$  and of the objects of the second mesh are well known from the analytical geometry. They have a compact and generic form in the vector variables. Therefore it is convenient to program the mild contact algorithm in the vector form.

The developed mild contact algorithm was tested for contact interaction of two elastic bodies with the following characteristics:  $\rho = 2800\text{kg/m}^3$ ,  $G = 26.5\text{GPa}$ ,  $K = 65.1\text{GPa}$ , the dimensions of the colliding bodies were  $2\text{m} \times 4\text{m}$ . The initial velocity of the first body (shown at the left in Figure 14) was set equal  $5\text{m/sec}$  (left in Figure 14), while the velocity of the second body (at the right in Figure 14) was set to be  $-5\text{m/sec}$ . For the one-dimensional case of bar impact, the contact stress can be calculated using the following formula (assuming that the materials of both bodies are identical):

$$\sigma_0 = \frac{\rho c v}{2}, \quad (3.72)$$

where  $c$  is speed of sound equal  $5000\text{m/sec}$ . For the selected material and speed of impact the contact stress is  $70\text{MPa}$ , which is below the yield stress of chosen material. The duration of the impact is equal to  $0.0016\text{sec}$ .

Since in the original formulation of the problem, we are modelling the impact of identical bodies, the impact plane also serves as a symmetry plane. From this perspective, it is possible to represent the problem as an impact of one body with the velocity of  $5\text{m/sec}$  into

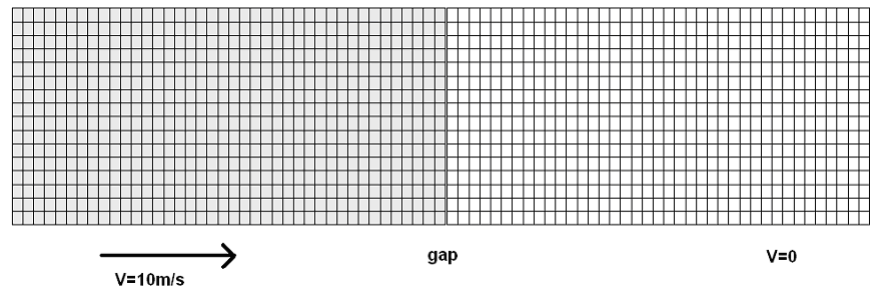


Figure 14: Initial position of the meshes.

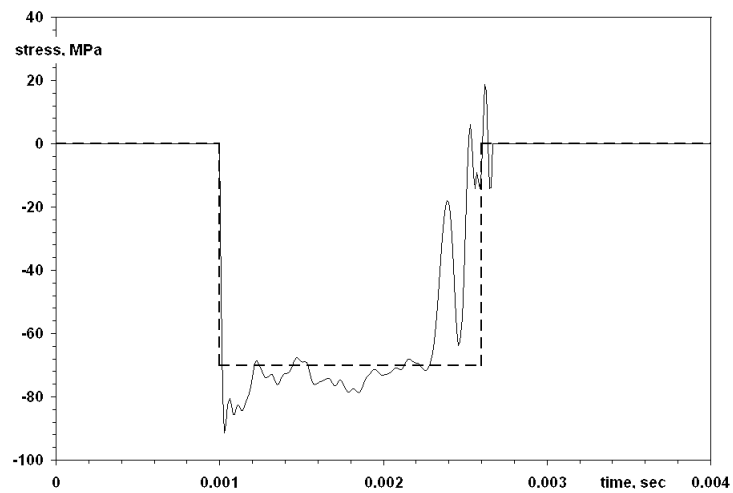


Figure 15: Comparison of the algorithm 1 (solid line) and the analytical results (dashed line) on the contact stress history zone.

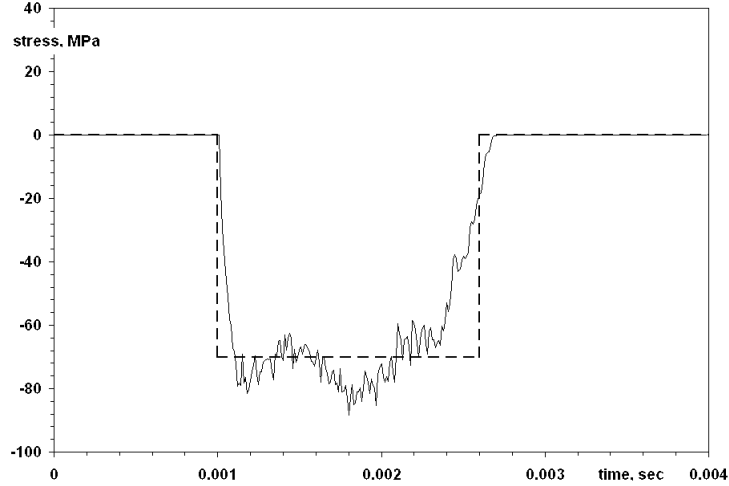


Figure 16: Comparison of the algorithm 2 (solid line) at  $c = 7 \cdot 10^7$  and  $h_0 = 0.01$  and the analytical results (dashed line) on the contact stress history zone.

the vertical rigid wall with coordinate  $x_0$ . For the rigid wall the boundary condition was formulated as follows: if  $x < x_0$  then  $x = x_0$  and  $v = 0$ . This is the simplest numerical algorithm providing non-penetration of the nodes through the rigid contact surface. We will call it algorithm 1.

In Figure 15, the stress history of the contact surface is shown with the solid line for the algorithm 1 and the result of the analytical solution is illustrated with the dashed line.

We will not analyze the difference between the numerical by algorithm 1 and the analytical solution. We are interested in the comparison of the algorithm 1 with the mild contact algorithm, which we call algorithm 2, with the same time and space steps.

In order to use the algorithm 2 we need to specify the parameters  $c$  and  $h_0$  in (3.71). Clearly, the width  $h_0$  of the layer where the force acts should be much less than the specific dimension of the bodies put into contact. For stationary or quasi-stationary problems we can use any value of  $c$  with the only restriction that it should not decrease the time step of the solution of the whole problem.

For nonstationary problems, like the impact of two solids in the considered example, we have the condition  $f/\Delta y = \sigma_0$ , where  $\Delta y$  is step of the mesh along the contact surface (Figure 14). Moreover, it is natural to assume that the elastic properties of the soft contact and of the solid must be similar, that is  $dh/h^2 = du/dx$ . Then taking into account (3.71), we obtain the following relation:

$$\frac{c}{\Delta y h_0} = E - \sigma_0, \quad (3.73)$$

where  $E = 9KG/(3K + G)$ ,  $\Delta y$  is the Young's modulus. For the case under consideration  $\Delta y = 0.1$ ,  $c/h_0 \approx 70\text{GPa}$ .



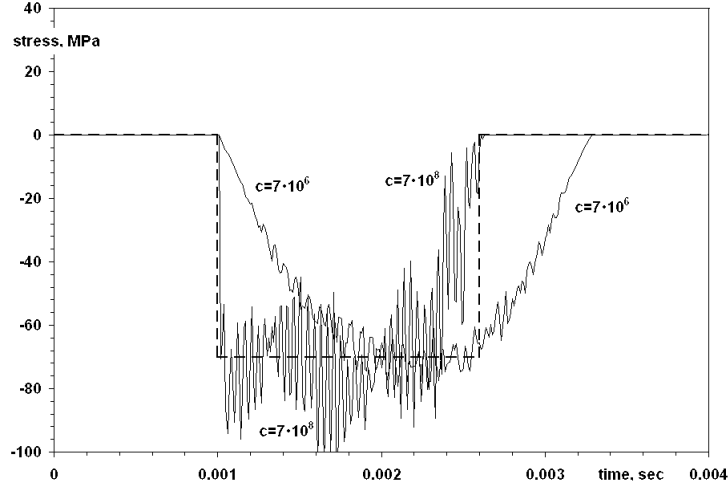


Figure 17: The influence of the parameter  $c$  on the contact stress history zone,  $h_0 = 0.01$ .

In Figure 16, the stress history of the contact surface is shown with the solid line for the algorithm 2 for  $c = 7 \cdot 10^7$  and  $h_0 = 0.01$ . The time step here is the same as in the algorithm 1. Like before, the result of the analytical solution is illustrated with the dashed line. We can see from Figures 15 and 16 that the results of the simulations with algorithms 1 and 2 are close to each other. In both case the contact between the two bodies starts at the same time,  $t = 0.001$ . We note that in the case of the algorithm 1 the stress changes from 0 to  $\sigma_0$  very rapidly and the curve is almost vertical (Figures 15), while in the algorithm 2 this transition is slower (Figures 16). The return to zero stress in the algorithm 2 is smoother than in the algorithm 1.

Figure 17 shows the results for different values of  $c$  and for the fixed value  $h_0 = 0.01$ . We decrease the value of  $c$  from  $7 \cdot 10^7$  to  $7 \cdot 10^6$ . Then the time of contact becomes longer but the amplitude of high frequency oscillations decreases. The time step in this case is the same as before. On the other hand, when we increased  $c$  up to  $7 \cdot 10^8$ , then the duration of the contact becomes closer to the theoretical one but the amplitude of the oscillations becomes much bigger. Moreover, the time step in this case should be decreased several times. The reason for this is that the choice of the parameters of the mild algorithm here is not optimal.

Figure 18 shows the influence of both parameters  $c$  and  $h_0$ . They are changed in such a way that their ratio  $c/h_0$  remains constant. The results weakly depend on them even when the values of the both parameters change their order of magnitude.

The results of the simulations of strain distribution after conventional  $180^\circ$  flanging of an aluminum sheet 6111-T4 with the 0.5 mm internal radius is shown in Figure 19. In Figure 19c we can see a thin gap between the two meshes, which corresponds to the mild contact.

Figure 20 shows the results of the simulations of the 3D hyperelastic solid similar to the natural rubber-incompressible media using the mild contact algorithm. We use the following

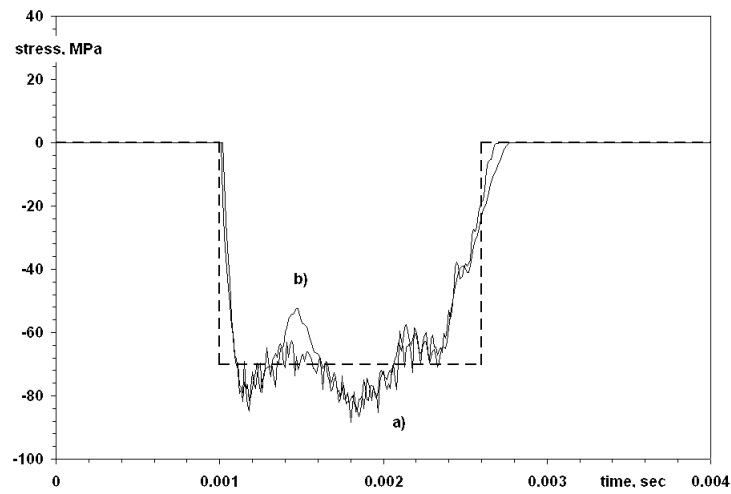


Figure 18: The influence of simultaneous change of parameters  $c$  and  $h_0$  on the contact stress history, when  $c/h_0$  remains constant (a)  $c = 7 \cdot 10^7$  and  $h_0 = 0.01$ ; b)  $c = 7 \cdot 10^6$  and  $h_0 = 0.001$ .

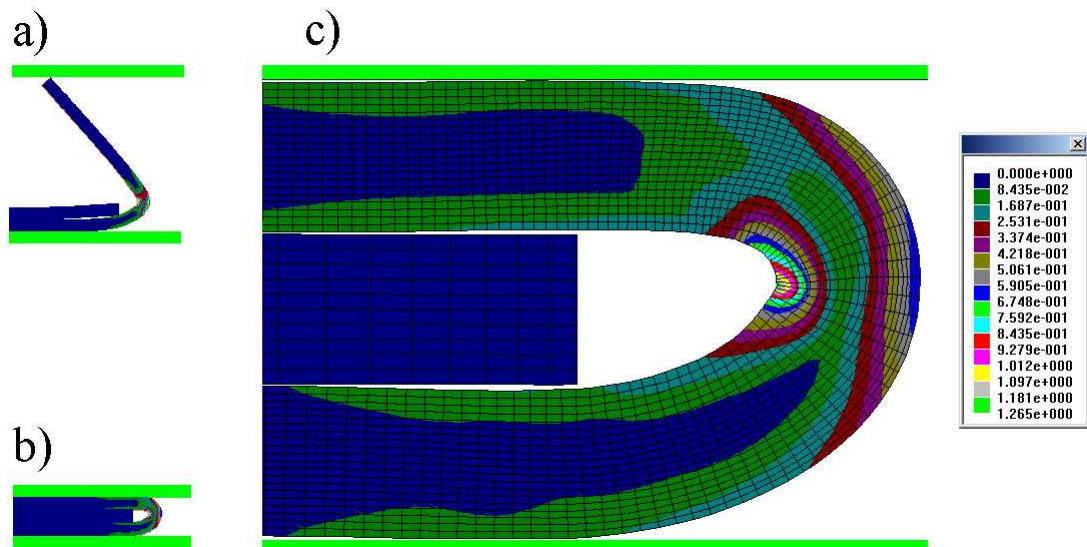


Figure 19: Distribution of plastic strains after flanging a sheet 0.93 mm thick with the internal radius of 0.5 mm.

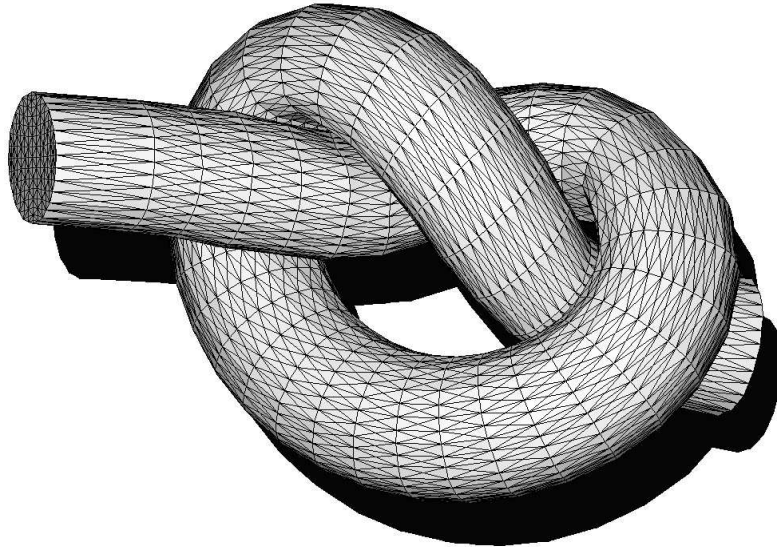


Figure 20: Deformation of a hyperelastic solid.

rheological model [18]:

$$\boldsymbol{\sigma} = -p\mathbf{I} + h_1\mathbf{B} + h_2\mathbf{B} \cdot \mathbf{B} \quad (3.74)$$

(see also [11]). Here we have a complex contact interaction in space. We can see in Figure 20 that there is a thin narrow void layer between the contacting surfaces.

In conclusion of this section we would like to emphasize that the mild contact algorithm is simple in realization, reliable, and it provides equality of forces acting on the surfaces in the contact zone. Moreover, it does not lead to the decrease of the time step in numerical simulations.

## 4 Concluding remarks

The modifications of Wilkins' method discussed in this work can give a number of advantages:

- We do not need to correct the stress rotation (like in equations (2.7)) any more. The necessity of these corrections is a serious disadvantage of the conventional approach which slows down the computations,
- This allows us to adopt a new point of view on plastic flow simulations (see Section 3.3),
- The mild contact algorithm (described in Section 3.4) is simple in realization, reliable, and it provides naturally the equality of forces acting on the surfaces in the contact zone.

## Acknowledgements

The authors would like to express their gratitude to Dr. Frode Maaseidvaag, the former Director of Ford Research Laboratory (USA), for his financial and friendly support of our research efforts. The authors would like to acknowledge prof. François Sidoroff from Ecole Centrale de Lyon (France) for his help and many useful discussions.

## A Vector/tensor notation

An orthogonal basis for the vector space (we use here Cartesian coordinates only) is a set of orthogonal unit vectors  $\vec{\mathbf{e}}_s$  (where  $s = 1, 2$  for 2D case,  $s = 1, 2, 3$  for 3D case, etc). The scalar product of any two of these vectors is

$$\vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_s = \delta_{ks} = \begin{cases} 1, & \text{if } k = s, \\ 0, & \text{if } k \neq s, \end{cases} \quad (\text{A.1})$$

where  $\delta_{ks}$  is the Kronecker delta symbol. A vector (first-order tensor)  $\vec{\mathbf{a}}$  can be decomposed in the introduced basis as

$$\vec{\mathbf{a}} = a_k \vec{\mathbf{e}}_k. \quad (\text{A.2})$$

The usual summation convention is over the repeated indices is assumed.

The dyadic (triadic, tetradic, etc) product of base vectors is the tensor  $\vec{\mathbf{e}}_j \vec{\mathbf{e}}_k$  ( $\vec{\mathbf{e}}_j \vec{\mathbf{e}}_k \vec{\mathbf{e}}_m$ ,  $\vec{\mathbf{e}}_j \vec{\mathbf{e}}_k \vec{\mathbf{e}}_m \vec{\mathbf{e}}_s$ , etc). These tensors serve as basis tensors for the representation of a second-order  $\mathbf{A} = A_{km} \vec{\mathbf{e}}_k \vec{\mathbf{e}}_m$ , third-order  $\mathbf{B} = B_{jkm} \vec{\mathbf{e}}_j \vec{\mathbf{e}}_k \vec{\mathbf{e}}_m$  tensors, and so on. For example, the unit (identity) tensor is  $\mathbf{I} = \delta_{ks} \vec{\mathbf{e}}_k \vec{\mathbf{e}}_s$ . The Levi-Civittá third-rank tensor is

$$\boldsymbol{\varepsilon} = \varepsilon_{abc} \vec{\mathbf{e}}_a \vec{\mathbf{e}}_b \vec{\mathbf{e}}_c. \quad (\text{A.3})$$

The Levi-Civittá  $n$ -rank tensor is

$$\boldsymbol{\varepsilon} = \varepsilon_{ab\dots s} \vec{\mathbf{e}}_a \vec{\mathbf{e}}_b \dots \vec{\mathbf{e}}_s, \quad (\text{A.4})$$

where

$$\varepsilon_{ab\dots s} = \begin{cases} 1, & \text{if } ab\dots s \text{ is an even permutation of } 12\dots n, \\ -1, & \text{if } ab\dots s \text{ is an odd permutation of } 12\dots n, \\ 0, & \text{if any of } ab\dots s \text{ are the same.} \end{cases} \quad (\text{A.5})$$

We denote the convolution of tensors of any rank by  $\mathbf{A} \cdot \mathbf{B}$ ,  $\mathbf{A} \cdot \cdot \mathbf{B}$ ,  $\mathbf{A} \cdot \cdot \cdot \mathbf{B}$  etc. Here we denote the double dot product by  $\mathbf{A} \cdot \cdot \mathbf{B}$  because this notation is easier to generalize for the case of multi-dot product in comparison with the notation  $\mathbf{A} : \mathbf{B}$ .

Consider two examples:

$$\begin{aligned} \vec{\mathbf{a}} \cdot \mathbf{A} \cdot \cdot \cdot \mathbf{B} &= a_a \vec{\mathbf{e}}_a \cdot A_{bcde} \vec{\mathbf{e}}_b \vec{\mathbf{e}}_c \vec{\mathbf{e}}_d \vec{\mathbf{e}}_e \cdot \cdot \cdot B_{fghij} \vec{\mathbf{e}}_f \vec{\mathbf{e}}_g \vec{\mathbf{e}}_h \vec{\mathbf{e}}_i \vec{\mathbf{e}}_j = \\ & a_a A_{bcde} B_{fghij} \vec{\mathbf{e}}_a \cdot \vec{\mathbf{e}}_b \vec{\mathbf{e}}_c \vec{\mathbf{e}}_d \vec{\mathbf{e}}_e \cdot \cdot \cdot \vec{\mathbf{e}}_f \vec{\mathbf{e}}_g \vec{\mathbf{e}}_h \vec{\mathbf{e}}_i \vec{\mathbf{e}}_j = \end{aligned}$$

$$a_a A_{bcde} B_{fghij} \delta_{ab} \delta_{ef} \delta_{dg} \delta_{ch} \vec{e}_i \vec{e}_j = a_a A_{acde} B_{edcij} \vec{e}_i \vec{e}_j.$$

$$\vec{a} \times \vec{b} = \varepsilon_{ijk} \vec{e}_i a_j b_k = \varepsilon_{ijk} \vec{e}_i \vec{e}_j \cdot \vec{e}_a a_a \vec{e}_k \cdot \vec{e}_b b_b = \varepsilon_{ijk} \vec{e}_i \vec{e}_j \vec{e}_k \cdot \vec{b} \vec{a} = \boldsymbol{\varepsilon} \cdot \vec{b} \vec{a}.$$

The principal invariants of a second-order tensor  $\mathbf{A}$  are

$$\begin{aligned} J_1(\mathbf{A}) &= \mathbf{I} \cdot \mathbf{A} = A_{11} + A_{22} + A_{33}, \\ J_2(\mathbf{A}) &= \begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix} + \begin{vmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{vmatrix} + \begin{vmatrix} A_{11} & A_{13} \\ A_{31} & A_{33} \end{vmatrix} = \\ &= \frac{1}{2} [J_1^2(\mathbf{A}) - J_1(\mathbf{A}^2)] = \frac{1}{2} [\mathbf{I} \cdot \mathbf{A} \mathbf{A} \cdot \mathbf{I} - \mathbf{A} \cdot \mathbf{A}], \\ J_3(\mathbf{A}) &= \det \mathbf{A} = \frac{1}{6} [J_1^3(\mathbf{A}) - 3J_1(\mathbf{A})J_1(\mathbf{A}^2) + 2J_1(\mathbf{A}^3)]. \end{aligned}$$

Let  $\mathbf{A}_D = \mathbf{A} - \frac{1}{3}J_1(\mathbf{A})\mathbf{I}$ . Then

$$J_1(\mathbf{A}_D) = 0, \quad J_2(\mathbf{A}_D) = \frac{1}{6}J_1^2(\mathbf{A}) - \frac{1}{2}J_1(\mathbf{A}^2). \quad (\text{A.6})$$

By definition,

$$\frac{d\vec{a}}{d\vec{b}} = \frac{\partial a_i}{\partial b_j} \vec{e}_i \vec{e}_j.$$

## B Reciprocal vectors

Consider a set of non-coplanar vectors  $\vec{x}_1, \vec{x}_2, \vec{x}_3$  in the 3D case. We can now define the set of reciprocal vectors  $\vec{x}^1, \vec{x}^2, \vec{x}^3$ :

$$\vec{x}^1 = \frac{\vec{x}_2 \times \vec{x}_3}{\vec{x}_1 \cdot (\vec{x}_2 \times \vec{x}_3)}, \quad \vec{x}^2 = \frac{\vec{x}_3 \times \vec{x}_1}{\vec{x}_2 \cdot (\vec{x}_3 \times \vec{x}_1)}, \quad \vec{x}^3 = \frac{\vec{x}_1 \times \vec{x}_2}{\vec{x}_3 \cdot (\vec{x}_1 \times \vec{x}_2)}. \quad (\text{B.1})$$

Relations (B.1) can be rewritten in the more universal form:

$$\vec{x}^1 = \frac{\boldsymbol{\varepsilon} \cdot \vec{x}_3 \vec{x}_2}{\boldsymbol{\varepsilon} \cdot \vec{x}_3 \vec{x}_2 \vec{x}_1}, \quad \vec{x}^2 = \frac{\boldsymbol{\varepsilon} \cdot \vec{x}_1 \vec{x}_3}{\boldsymbol{\varepsilon} \cdot \vec{x}_1 \vec{x}_3 \vec{x}_2}, \quad \vec{x}^3 = \frac{\boldsymbol{\varepsilon} \cdot \vec{x}_2 \vec{x}_1}{\boldsymbol{\varepsilon} \cdot \vec{x}_2 \vec{x}_1 \vec{x}_3}. \quad (\text{B.2})$$

Formulas (B.2) can be reversed, i.e., the following equations are valid:

$$\vec{x}_1 = \frac{\boldsymbol{\varepsilon} \cdot \vec{x}^3 \vec{x}^2}{\boldsymbol{\varepsilon} \cdot \vec{x}^3 \vec{x}^2 \vec{x}^1}, \quad \vec{x}_2 = \frac{\boldsymbol{\varepsilon} \cdot \vec{x}^1 \vec{x}^3}{\boldsymbol{\varepsilon} \cdot \vec{x}^1 \vec{x}^3 \vec{x}^2}, \quad \vec{x}_3 = \frac{\boldsymbol{\varepsilon} \cdot \vec{x}^2 \vec{x}^1}{\boldsymbol{\varepsilon} \cdot \vec{x}^2 \vec{x}^1 \vec{x}^3}. \quad (\text{B.3})$$

Thus, the sets of vectors  $\vec{x}_1, \vec{x}_2, \vec{x}_3$  and  $\vec{x}^1, \vec{x}^2, \vec{x}^3$  are mutually reciprocal vectors.

The following equalities are valid:

$$\vec{\mathbf{x}}_k \cdot \vec{\mathbf{x}}^m = \delta_{km}, \quad \vec{\mathbf{x}}_m \vec{\mathbf{x}}^m = \vec{\mathbf{x}}^m \vec{\mathbf{x}}_m = \mathbf{I}, \quad \varepsilon \cdots \vec{\mathbf{x}}_3 \vec{\mathbf{x}}_2 \vec{\mathbf{x}}_1 = \frac{1}{\varepsilon \cdots \vec{\mathbf{x}}^3 \vec{\mathbf{x}}^2 \vec{\mathbf{x}}^1}. \quad (\text{B.4})$$

Let us consider the set of vectors  $\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \vec{\mathbf{x}}_3$  directed along any three different edges of the tetrahedral element. We can define the symmetric tensors  $\mathbf{G}_{\vec{\mathbf{x}}}$  and  $\mathbf{G}^{\vec{\mathbf{x}}}$  with the components  $\vec{\mathbf{x}}_i \cdot \vec{\mathbf{x}}_j$  and  $\vec{\mathbf{x}}^i \cdot \vec{\mathbf{x}}^j$ , respectively. It can be shown that

$$\mathbf{G}^{\vec{\mathbf{x}}} \cdot \mathbf{G}_{\vec{\mathbf{x}}} = \mathbf{G}_{\vec{\mathbf{x}}} \cdot \mathbf{G}^{\vec{\mathbf{x}}} = \mathbf{I}, \quad \mathbf{G}^{\vec{\mathbf{x}}} \cdot \cdot \mathbf{G}_{\vec{\mathbf{x}}} = 3. \quad (\text{B.5})$$

If the tensor  $\mathbf{G}_{\vec{\mathbf{x}}}$  is known, then we can restore the geometry of the tetrahedral element up to translation and rotation. In fact, from the diagonal components of  $\mathbf{G}_{\vec{\mathbf{x}}}$  we can obtain the dimensions of the sides of the tetrahedron, and from the non-diagonal components of  $\mathbf{G}_{\vec{\mathbf{x}}}$  we can determine the angles between these sides.

We can obtain similar relations in the 2D case. For example, the set of vectors reciprocal to 2D vectors  $\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2$  are

$$\vec{\mathbf{x}}^1 = \frac{\varepsilon \cdot \vec{\mathbf{x}}_2}{\varepsilon \cdot \cdot \vec{\mathbf{x}}_2 \vec{\mathbf{x}}_1}, \quad \vec{\mathbf{x}}^2 = \frac{-\varepsilon \cdot \vec{\mathbf{x}}_1}{\varepsilon \cdot \cdot \vec{\mathbf{x}}_2 \vec{\mathbf{x}}_1}. \quad (\text{B.6})$$

## C Jaumann rate

Let the tensor  $\mathbf{A}$  be given in the basis  $\vec{\mathbf{e}}'_1, \vec{\mathbf{e}}'_2, \vec{\mathbf{e}}'_3$  rotating with respect to the basis  $\vec{\mathbf{e}}_1, \vec{\mathbf{e}}_2, \vec{\mathbf{e}}_3$  with the angular velocity  $\boldsymbol{\omega}$ . Let the bases  $\vec{\mathbf{e}}'_1, \vec{\mathbf{e}}'_2, \vec{\mathbf{e}}'_3$  and  $\vec{\mathbf{e}}_1, \vec{\mathbf{e}}_2, \vec{\mathbf{e}}_3$  coincide at time  $t$ . Then

$$\frac{dA_{ks} \vec{\mathbf{e}}'_k \vec{\mathbf{e}}'_s}{dt} = \frac{dA_{ks}}{dt} \vec{\mathbf{e}}_k \vec{\mathbf{e}}_s + A_{ks} \frac{d\vec{\mathbf{e}}'_k}{dt} \vec{\mathbf{e}}_s + A_{ks} \vec{\mathbf{e}}_k \frac{d\vec{\mathbf{e}}'_s}{dt}, \quad (\text{C.1})$$

or, taking into account that  $d\vec{\mathbf{e}}'_k/dt = \boldsymbol{\omega} \times \vec{\mathbf{e}}_k$ ,

$$\overset{\nabla}{\mathbf{A}} = \dot{\mathbf{A}} + \boldsymbol{\gamma}, \quad (\text{C.2})$$

$\boldsymbol{\gamma} = \mathbf{A} \cdot \boldsymbol{\Omega} - \boldsymbol{\Omega} \cdot \mathbf{A}$ , where  $\overset{\nabla}{\mathbf{A}} = d\mathbf{A}/dt$ ,  $\dot{\mathbf{A}} = dA_{ks}/dt \vec{\mathbf{e}}_k \vec{\mathbf{e}}_s$ ,  $\boldsymbol{\Omega} = \varepsilon \cdot \boldsymbol{\omega}$ . The components of the second order tensor  $-\boldsymbol{\Omega} \cdot \mathbf{A} + \mathbf{A} \cdot \boldsymbol{\Omega}$  in the right-hand side of (C.2) are the same as  $\gamma_{nm}$  in (2.7).

## References

- [1] G. Alfano, F. De Angelis, L. Rosati. *General solution procedures in elasto/viscoplasticity*. Computer Methods in Applied Mechanics and Engineering, 190 (2001), 5123-5147.

- [2] F. Auricchio. *A viscoplastic constitutive equation bounded between two generalized plasticity models*. International Journal of Plasticity, 13 (1997), 697-721.
- [3] F. Auricchio. *A robust integration-algorithm for a finite-strain shape-memory-alloy superelastic model*. International Journal of Plasticity, 17 (2001), 971-990.
- [4] F. Auricchio, R.L. Taylor. *Shape-memory alloys: modelling and numerical simulations of the finite-strain superelastic behavior*. Computer Methods in Applied Mechanics and Engineering, 143 (1997), 175-194.
- [5] I.A. Balagansky, Yu.A. Karanik, V.A. Agureikin, A.V. Vinogradov, A.I. Balagansky. *Fracture behavior of explosively loaded spherical molded steel shells*. Theoretical and applied fracture mechanics, 36 (2001), 165-173.
- [6] R.R. Balokhonov, P.V. Makarov, V.A. Romanova, I.Yu. Smolin. *Simulation of crystal plasticity under dynamic loading*. Computational Materials Science, 16 (1999), 355-361.
- [7] E. Baron, M.B. Rubin, D.Z. Yankelevsky. *Thermomechanical constitutive equations for the dynamic response of ceramics*. International Journal of Solids and Structures, 40 (2003), 4519-4548.
- [8] J.B. Bdzil, D. S. Stewart, T. L. Jackson. *Program burn algorithms based on detonation shock dynamics: discrete approximations of detonation flows with discontinuous front models*. Journal of Computational Physics, 174 (2001), No. 2, 870-902.
- [9] D.J. Benson, L. Stainier. *An Eulerian shell formulation for fluid-structure interaction*. Computer Methods in Applied Mechanics and Engineering, 187 (2000), 571-590.
- [10] G. C. Bessette, E. B. Becker, L. M. Taylor, D. L. Littlefield. *Modeling of impact problems using an h-adaptive, explicit Lagrangian finite element method in three dimensions*. Computer Methods in Applied Mechanics and Engineering, 192 (2003), No. 13-14, 1649-1679.
- [11] N.M. Bessonov, D.J. Song. *Application of vector calculus to numerical simulation of continuum mechanics problems*. Journal of Computational Physics, 167 (2001), No. 1, 22-38.
- [12] N. Bessonov, S. Golovashchenko. *Numerical simulation of pulsed electromagnetic stamping processes*. Proceedings of 1st International Conference on High Speed Forming, Dortmund, Germany, 2004, 83-91.
- [13] M. Brünig. *Nonlinear finite element analysis based on a large strain deformation theory of plasticity*. Computers & Structures, 69 (1998), 117-128.
- [14] D. Chen, S.T.S. Al-Hassani, Z. Yin, Y. Yu. *Modelling shock loading behavior of concrete*. International Journal of Solids and Structures, 38 (2001), 8787-8803.



- [15] O.I. Cherepanov. *Localized viscoelastoplastic strain in mesovolume of heterogeneous medium under different loading types*. Theoretical and applied fracture mechanics, 31 (1999), 189-202.
- [16] O.I. Cherepanov, I.Yu. Smolin, Yu.P. Stefanov, P.V. Makarov. *Integration of influence of internal structure of heterogeneous materials on plastic flow and fracture*. Computational Materials Science, 16 (1999), 25-31.
- [17] P.W. Christensen. *A nonsmooth Newton method for elastoplastic problems*. Computer Methods in Applied Mechanics and Engineering, 191 (2002), 1189-1219.
- [18] P.G. Ciarlet. *Mathematical Elasticity, 1, Three Dimensional Elasticity*. North-Holland, Amsterdam, 1993.
- [19] G. Cocchetti, U. Perego. *A rigorous bound on error in backward-difference elastoplastic time-integration*. Computer Methods in Applied Mechanics and Engineering, 192 (2003), 4909-4927.
- [20] A. Düster, E. Rank. *A p-version finite element approach for two- and three-dimensional problems of the  $J_2$  flow theory with non-linear isotropic hardening*. International Journal for Numerical Methods in Engineering, 53 (2002), 49-63.
- [21] N.A. Fellows, P.C. Barton. *Development of impact model for ceramic-faced semi-infinite armor*. International Journal of Impact Engineering, 22 (1999), 793-811.
- [22] Z.Q. Feng, Z. Feng, M. Domaszewski. *Some computational aspects for analysis of low- and high- velocity impact of deformable bodies*. International Journal of Non-Linear Mechanics, 37 (2002), 1029-1036.
- [23] J. Fish, K. Shek. *Computational aspects of incrementally objective algorithms for large deformation plasticity*. International Journal for numerical methods in engineering, 44 (1999), 839-851.
- [24] S. Golovashchenko N. Bessonov. *Development of Sharp Flanging Technology for Aluminum Panels*. Proceedings of the 6th International Conference on Numerical Simulation of 3D Sheet Forming Processes, NUMISHEET 2005, Detroit, MI, 687-690.
- [25] S. Golovashchenko, N. Bessonov, R. Davies. *Analysis of Blank-Die Contact Interaction in Pulsed Forming Processes*. 3rd International Conference on High Speed Forming, Germany, 2008.
- [26] J. Guo, J.V. Cox. *Implementation of a plasticity bond model for reinforced concrete*. Computers & Structures, 77 (2000), 65-82.
- [27] Y.M. Gupta, J.L. Ding. *Impact load spreading in layered materials and structures: concept and quantitative measure*. International Journal of Impact Engineering, 27 (2002), 277-291.



- [28] E. Hoashi, T. Yokomine, A. Shimizu, T. Kunugi. *Numerical analysis of wave-type heat transfer propagating in a thin foil irradiated by short-pulsed laser*. International Journal of Heat and Mass Transfer, 46 (2003), 4083-4095.
- [29] B. P. Howell, G. J. Ball. *A Free-Lagrange augmented Godunov method for the simulation of elasticplastic solids*. Journal of Computational Physics, 175 (2002), No. 1, 128-167.
- [30] L. Jing. *A review of techniques, advances and outstanding issues in numerical modelling for rock mechanics and rock engineering*. International Journal of Rock Mechanics and Mining Sciences, 40 (2003), No. 3, 283-353.
- [31] L. Jing, J. A. Hudson. *Numerical methods in rock mechanics*. International Journal of Rock Mechanics and Mining Sciences, 39 (2002), No. 4, 409-427.
- [32] G.R. Johnson, S.R. Beissel. *Damping algorithms and effects for explicit dynamics computations*. International Journal of Impact Engineering, 25 (2001), 911-925.
- [33] A.S. Khan, S. Huang, Continuum theory of plasticity. John Wiley & Sons, New-York, 1995.
- [34] S.P. Kiselev, V.P. Kiselev. *Superdeep penetration of particles into a metal target*. International Journal of Impact Engineering, 27 (2002), 135-152.
- [35] A. Le van, G. de Saxcaae, P. Le Grogneec. *General formulation for local integration in standard elastoplasticity with an arbitrary hardening model*. Computers and Structures, 81 (2003), 2099-2109.
- [36] M. Lee, Y.H. Yoo. *Analysis of ceramic/metal armour systems*. International Journal of Impact Engineering, 25 (2001), 819-829.
- [37] Z.Y. Liu, S. Kubota, S. Itoh. *Numerical study on hypervelocity acceleration of flyer plates by overdriven detonation of high explosive*. International Journal of Impact Engineering, 26 (2001), 443-452.
- [38] V. A. Lubarda. Elastoplasticity theory, CRP Press, New York, 2000.
- [39] L.X. Luccioni, J.M. Pestana, A. Rodriguez-Marek. *An implicit integration algorithm for the finite element implementation of a nonlinear anisotropic material model including hysteretic nonlinearity*. Computer Methods in Applied Mechanics and Engineering, 190 (2000), 1827-1844.
- [40] M. Maenchen, S. Sack. *The tensor code*. In: B. Alder (Ed.), Methods in Computational Physics, Vol. 3, Academic Press, New-York, 1964, 181-210.

- [41] R. Mahnken. *Anisotropic creep modelling based on elastic projection operators with applications to CMSX-4 superalloy*. Computer Methods in Applied Mechanics and Engineering, 191 (2002), 1611-1637.
- [42] P.V. Makarov, V.A. Romanova. *Mesoscale plastic flow generation and development for polycrystals*. Theoretical and Applied Fracture Mechanics, 33 (2000), 1-7.
- [43] P.V. Makaraov, S. Schmauder, O.I. Cherepanov, I.Yu. Smolin, V.A. Romanova, R.R. Balokhonov, D.Yu. Saraev, E. Soppa, P. Kizler, G. Fischer, S. Hu, M. Ludwig. *Simulation of elastic-plastic deformation and fracture of materials at micro-, meso- and macrolevels*. Theoretical and Applied Fracture Mechanics, 37 (2001), 183-244.
- [44] Mariotti, J. P. Perlat, J. M. Guérin. *A numerical approach for partially saturated geomaterials under shock*. International Journal of Impact Engineering, 28 (2003), No. 7, 717-741.
- [45] G. H. Miller, P. Colella. *A High-order Eulerian Godunov method for elasticplastic flow in solids*. Journal of Computational Physics, 167 (2001), No. 1, 131-176.
- [46] F.J. Montáns. *Implicit algorithms for multilayer  $J_2$  plasticity*. Computer Methods in Applied Mechanics and Engineering, 189 (2000), 673-700.
- [47] S. K. Naboulsi, A. N. Palazotto. *Damage model for metalmatrix composite under high intensity loading*. International Journal of Plasticity, 19 (2003), No. 4, 435-468.
- [48] L. Noels, L. Stainier, J.-P. Ponthot. *On the use of large time steps with an energy momentum conserving algorithm for non-linear hypoelastic constitutive models*. International Journal of Solids and Structures, 41 (2004), 663693.
- [49] P. Papadopoulos, J. Lu. *On the formulation and numerical solution of problems in anisotropic finite plasticity*. Computer Methods in Applied Mechanics and Engineering, 190 (2001), 4889-4910.
- [50] A.N. Parshikov, S.A. Medin. *Smoothed particle hydrodynamics using interparticle contact algorithms*. Journal of Computational Physics, 180 (2002), No. 1, 358-382.
- [51] V.L. Popov, A. Gervé, B. Kehrwald, I. Yu. Smolin. *Simulation of wear in combustion engines*. Computational Materials Science, 19 (2000), 285-291.
- [52] J.P. Ponthot. *Unified stress update algorithms for the numerical simulation of large deformation elasto-plastic and elasto-viscoplastic processes*. International Journal of Plasticity, 18 (2002), 91-126.
- [53] B.A. Roeder, C.T. Sun. *Dynamic penetration of alumina/aluminum laminates: experiments and modelling*. International Journal of Impact Engineering, 25 (2001), 169-185.

- [54] V. Romanova, R. Balokhonov, P. Makarov, S. Schmauder, E. Soppa. *Simulation of elasto-plastic behaviour of an artificial 3D-structure under dynamic loading*. Computational Materials Science, 28 (2003), 5185-28.
- [55] M. B. Rubin, S. R. Bodner. *A three-dimensional nonlinear model for dissipative response of soft tissue*. International Journal of Solids and Structures, 39 (2002), No. 19, 5081-5099.
- [56] D. Sherman. *Impact failure mechanisms in alumina tiles on finite thickness support and the effect of confinement*. International Journal of Impact Engineering, 24 (2000), 313-328.
- [57] C.H.M. Simha, S.J. Bless, A. Bedford. *Computational modelling of the penetration response of a high-purity ceramic*. International Journal of Impact Engineering, 27 (2002), 65-86.
- [58] J. C. Simo, T. J. R. Hughes. Computational inelasticity. In: Interdisciplinary Applied Mathematics, Vol.7, 1998.
- [59] I.Y. Smolin, P.V. Makarov, D.V. Shmick, I.V. Savlevich. *A micropolar model of plastic deformation of polycrystals at the mesolevels*. Computational Materials Science, 19 (2000), 133-142.
- [60] S.C. Song, Z.P. Duan, D.W. Tan. *The application of B-P constitutive equations in finite element analysis of high velocity impact*. International Journal of Solids and Structures, 38 (2001), 5215-5222.
- [61] J.B. Stevens, R.C. Batra. *Adiabatic shear bands in the Taylor impact test for a WHA rod*. International Journal of Plasticity, 14 (1998), No. 9, 841-854.
- [62] Y.P. Stefanov. *Wave dynamics of cracks and multiple contact surface interaction*. Theoretical and Applied Fracture Mechanics, 34 (2000), 101-108.
- [63] H.H. Vaziri, J.S. Jalali, R. Islam. *An analytical model for stability analysis of rock layers over a circular opening*. International Journal of Solids and Structures, 38 (2001), 3735-3757.
- [64] M. Wallin, A. Ristinmaa. *Accurate stress updating algorithm based on constant strain rate assumption*. Computer Methods in Applied Mechanics and Engineering, 190 (2001), 5583-5601.
- [65] M.L. Wilkins. Calculation of elastic-plastic flow. In: B. Alder (Ed.), Methods in Computational Physics, Vol. 3, Academic Press, New-York, 1964, 211-263.
- [66] M.L. Wilkins. Calculation of elastic-plastic flow. In: B. Alder (Ed.), Numerical Methods in Hydrodynamics, translated from English, Mir, Moscow, 1967.

- [67] M.L. Wilkins. Computer simulation of dynamic phenomena. Scientific Computation, Springer, 1998.
- [68] M.L. Wilkins. *Mechanics of penetration and perforation*. International Journal of Engineering Sciences, 16 (1978), 793-807.
- [69] M.L. Wilkins. *Use of artificial viscosity in multidimensional fluid dynamics calculations*. Journal of Computational Physics, 36 (1980), 291-303.
- [70] M.L. Wilkins. *Modelling the behavior of materials*. In: International Conference on Structural Impact and Crachworthiness, 1984, 243-277.
- [71] M.L. Wilkins, M.W. Guinan. *Impact of cylinders on a rigid boundary*. Journal of Applied Physics, 44 (1973), 1200-1206.
- [72] M.L. Wilkins, M.W. Guinan. *Plane stress calculations with a two dimensional elastic-plastic computer program*. UCRL-77251. University of California, Lawrence Livermore Laboratory, (1976).
- [73] K. Wünnemann, B.A. Ivanov. *Numerical modelling of the impact crater depthdiameter dependence in an acoustically fluidized target*. Planetary and Space Science, 51 (2003), 831-845.
- [74] F. Xiao. *Unified formulation for compressible and incompressible flows by using multi-integrated moments I: one-dimensional inviscid compressible flow*. Journal of Computational Physics, 195 (2004), 629-654.
- [75] R. Zaera, S. Sánchez-Sáez, J.L. Pérez-Castellanos, C. Navarro. *Modelling of the adhesive layer in mixed ceramic/metal armours subjected to impact*. Composites part A: Applied Science and Manufacturing, 31 (2000), 823-833.