Onderzoekersopleiding Fundamentele Werktuigkunde

**A EULERIAN APPROACH TO THE
FINITE ELEMENT MODELLING OF
NEO-HOOKEAN RUBBER MATERIAL**

Peter A.A. van Hoogstraten

W&W 90.040

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Hoogstraten, Peter A.A. van

A Eulerian approach to the finite element modelling of neo-Hookean rubber material /
Peter A.A. van Hoogstraten. - Eindhoven : Instituut Vervolgopleidingen, TUE. -111.

Met lit.opg.

ISBN 90-5282-055-4

SISO 650.2 UDC 621.3:678.4

Trefw.: eindige elementen methode / rubber.

Abstract

A Eulerian approach is applied to the finite element modelling of neo-Hookean rubber material. Two major problems are encountered. The first problem is the construction of an algorithm to calculate stresses in the rubber material from velocities instead of displacements. This problem is solved with an algorithm based on the definition of the velocity gradient. The second problem is the convection of stresses through the finite element mesh. This problem is solved by adapting the so-called Taylor-Galerkin technique. Solutions for both problems are implemented in a finite element program and their validity is shown by test problems. Results of these implementations are compared with results obtained by a standard Lagrangian approach finite element package and good agreement has been found.

Table of Contents

	page
0. General Introduction	1
1. Introduction: Statement of the problem	5
2. Balance laws and the neo-Hookean constitutive equation	9
3. The Eulerian approach: Derivation of the velocity-stress algorithm	11
3.1 Incremental objectivity of the algorithm	14
3.1.1 Rigid body rotation	14
3.1.2 Uniform isovolumetric elongation	16
4. The Eulerian approach: Convection of stresses	19
5. Implementation aspects	23
6. Test problem	25
7. Conclusions	31
References	33
Appendix A: Incremental objectivity of the stress algorithm	35
Appendix B: PC-MATLAB files	39
B.1: Rigid body rotation	39
B.2: Isovolumetric elongation	42

List of Figures

figure 1: Layout of VHS video recorder

figure 2: Capstan Drive

figure 3: Results of calculations on rigid body rotation: component τ_{11} or τ_{22}

figure 4: Results of calculations on isovolumetric elongation: component τ_{11}

figure 5: Results of calculations on isovolumetric elongation: component τ_{22}

figure 6: Geometry and kinematic boundary conditions of the test problem

figure 7: Cauchy normal stress in $\vec{e}_1$ -direction

figure 8: Cauchy normal stress in $\vec{e}_2$ -direction

figure 9: Cauchy shear stresses

Nomenclature and Symbols

$\vec{a}$	vector
$\mathbf{A}$	tensor
$\mathbf{A}^c$	conjugate tensor of $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse tensor of $\mathbf{A}$
$\dot{\mathbf{A}}$	time derivative of tensor $\mathbf{A}$
$\mathbf{A}_t$	tensor $\mathbf{A}$ at time t_t
$tr(\mathbf{A})$	trace of tensor $\mathbf{A}$
$\vec{a}\vec{b}$	dyadic product of vectors $\vec{a}$ and $\vec{b}$
$\frac{\partial}{\partial t}$	spatial time derivative
$\frac{D}{Dt}$	material time derivative
$[\cdot, \cdot, \cdot]$	scalar inner product on the domain
$\langle \cdot, \cdot, \cdot \rangle$	scalar inner product on the boundary
$(\cdot; \cdot)$	tensor inner product on the domain
$\mathbf{B}$	left Cauchy Green strain tensor
C	neo-Hookean material parameter
C_{10}, C_{01}	Mooney-Rivlin material parameters
$\mathbf{D}$	rate of strain tensor

$\vec{e}_i$	basis vector
$\mathbf{F}$	deformation gradient
$\mathbf{I}$	unity tensor
I_1, I_2	first and second invariant of $\mathbf{B}$
$\mathbf{L}$	velocity gradient
$\vec{n}$	unit outward normal vector
p	hydrostatic pressure
p^*	pressure in rearranged neo-Hookean equation
q	weighting function
t	time
$T(\vec{x}_i; t_i)$	value of scalar T at position $\vec{x}_i$ and time t_i
$\vec{u}$	velocity vector
W	strain energy function
$\vec{w}$	weighting vector
$\vec{x}$	position vector
α	interpolation parameter
Δt	time step
$\vec{\nabla}$	gradient with respect to the current configuration
$\vec{\nabla}_0$	gradient with respect to the reference configuration
σ	Cauchy stress tensor
τ	extra stress tensor in rearranged neo-Hookean equation
Ω	rate of rotation tensor

0. General Introduction

Magnetic tapes and discs are very important data storage media; Magnetic recording is used for a broad range of applications, such as the storage of music, TV-images, pictures and computer data. The main advantages of magnetic media are the high density of information, which can be stored on tape or disc, their low costs, and their excellent reusability properties.

Philips has been, and still is, spending much effort on the research on magnetic recording. Video recording is one of the most important research topics on magnetic recording. In this general introduction, the basic principles behind video recording are briefly explained. Extra attention is paid to the capstan drive.

Fig.1. Layout of VHS video recorder

A schematic layout of a regular VHS video recorder is depicted in figure 1. As can be seen in this picture, the tape path in a videorecorder is much more complicated than the path in an audio cassette recorder. The tape is taken out of the cassette and guided by pens and rollers along several units. The guidance elements lift the tape out of its cassette plane, which results in a true three dimensional tape path. The main unit in the tape path is the scanner. This is a complex high precision rotating drum, containing the magnetic heads, which read and write the video signal on the tape. In general, the tape is driven by two reel motors, above which the tape cassette is positioned, and by the so-called capstan drive. An enlarged picture of this capstan drive is presented in figure 2.

Fig.2. Capstan Drive

The capstan drive consists of a rigid metal shaft (capstan), driven by a DC-motor, and a pinch roller, which is pressed against the capstan by an elastic spring. This pinch roller is a rigid metal cylinder, covered by a rubber coating. In the metal cylinder one or two roller bearings or sliding bearings are installed. The tape is placed in the nip between capstan and pinch roller. Due to frictional forces in the contact areas, the tape is transported.

Important trends in modern video recording are, amongst others, a strong increase of the information density on tapes and the introduction of new features. A higher density of information is obtained by the application of smaller track widths on the tape and by a reduction of the tape thickness. Smaller track widths require a more precise tape traction. A thinner tape requires a reduction of tape forces; tape strains must be restricted to about 0.5%, to avoid damage of the

magnetic tape coating. New features can, for example, introduce a dynamic component in the tape forces.

To be able to predict, for example, tape forces and the traction accuracy, a numerical model of the tape path in a video recorder is necessary. At Philips' Research Laboratories Eindhoven, a number of research projects are aimed to obtain such a numerical model. One of these projects is the the modelling of the capstan drive. This project is carried out in cooperation with the section of Engineering Fundamentals of the department of Mechanical Engineering, Eindhoven University of Technology. Its ultimate goal is a three dimensional Finite Element model of the capstan drive, verified by experiments.

1. Introduction: Statement of the problem

In this section, some characteristics of the capstan drive are discussed. Based on these characteristics, a strategy is presented for the setup of a computationally efficient FEM model.

Experiments have shown that three dimensional phenomena can play an important role in the performance of the capstan drive. A good example of such a phenomena is the (assumed) self-adjusting property of the pinch roller, if it is constructed with one central roller bearing. To be able to investigate these 3D phenomena, the model of the drive will have to be a three dimensional model.

Important phenomena occur in a restricted part of the capstan drive: Only in small surrounding area of the contact zone, stresses and deformations will exist in the rubber coating. The major part of the pinch roller is without stresses and deformations.

In a model of the capstan drive, at least two components, the rubber coating and the tape, will have to be modelled as elastic bodies. In a regular VHS-recorder, the indentation of the rubber coating is about 10%, so this deformation has to be taken into account and even has to be modelled as a geometrically nonlinear deformation to obtain reliable results. The tape strain, though restricted to about 0.5%, plays an important role in the finally obtained accuracy of the capstan drive.

A Finite Element model of the capstan drive will be highly nonlinear, due to three nonlinearities. The geometrically nonlinear deformation of the rubber coating is already discussed. Previous investigations have shown that the rubber coating material can very well be modelled by the neo-Hookean constitutive equation, a simple, though nonlinear, constitutive equation for incompressible, isotropic, elastic material behaviour. Finally, the rolling contact between tape and capstan, tape and pinch roller, and (for wide pinch rollers) capstan and pinch roller is inherent highly nonlinear.

The characteristics, mentioned above, indicate that a computationally efficient FEM model can be obtained if it is possible to model only the small, interesting part of the drive near the contact zones. This will lead to a strong decrease of the amount of degrees of freedom in the model, especially if mesh refinement in the contact zone can be applied.

A standard Lagrangian (material) approach can not be used for such a model: In a Lagrangian approach all finite elements at the coating surface will ultimately pass the contact zone. To obtain accurate results, a fine, tangentially equidistant finite element mesh, covering the total coating geometry, has to be applied. Particularly in 3D calculations, this will lead to extremely high computation times.

An alternative approach, suitable for a model as mentioned above, is the so-called Updated Eulerian approach. According to this approach, in the tangential direction of the coating a Eulerian (spatial) approach is followed. Under the assumption that the solid neo-Hookean material "flows" in and out of the finite element mesh in a state of zero stress and deformation, obeying the global balance of mass, this enables us to model only the interesting deforming part of the elastic roller. In the radial direction an algorithm has to be formulated for the updating of the finite element mesh, in order to trace the deforming surface and to adapt the finite element mesh in a correct manner.

As part of the research program on the modelling of the capstan drive, in this report the first step to obtain an Updated Eulerian formulation for the neo-Hookean rubber coating is treated.

First, the neo-Hookean constitutive equation is introduced. Then, a Eulerian approach to this constitutive equation is presented. Doing this, two major problems are encountered. First of all, in a Eulerian approach velocities rather than displacements are state variables. Therefore, an algorithm has to be proposed for the calculation of stresses using velocities instead of displacements. Such an algorithm is presented. It is based on the formal definition of the velocity gradient. Its objectivity and accuracy aspects are discussed, and the importance of property of incremental objectivity is shown by some numerical experiments. The second problem encountered is the convection of stresses, which has to be taken into account. The Taylor-Galerkin method is used to solve this problem. The solutions for both problems are implemented in a finite element program and some implementation aspects are discussed. Finally, a test problem is presented, which

compares the results of our implementations with the results of a standard Lagrangian oriented calculation. Good agreement has been found.

2. Balance laws and the neo-Hookean constitutive equation

Each particle of an incompressible body, subjected to an isothermal deformation process, has to satisfy the following three local balance laws, assuming no external volume forces or inertia effects are present:

$$\text{Balance of mass:} \quad \vec{\nabla} \cdot \vec{u} = 0 \quad (1)$$

$$\text{Balance of momentum:} \quad \vec{\nabla} \cdot \sigma^c = \vec{0} \quad (2)$$

$$\text{Balance of angular momentum:} \quad \sigma = \sigma^c \quad (3)$$

The symbol $\vec{\nabla}$ denotes the gradient operator, with respect to the current configuration, vector $\vec{u}$ is the velocity vector and tensor σ is the Cauchy stress tensor. A conjugate tensor is denoted by a superindex c.

Furthermore, a constitutive equation has to be satisfied. Since rubberlike materials mostly behave highly elastic and (nearly) incompressible, for the modelling of such a material often perfectly elastic isotropic isothermal incompressible behaviour is assumed. Then, the existence of a strain-energy function (or stored energy function), which fully characterizes the mechanical properties of the rubberlike material, can be postulated. The most common form of strain-energy functions for incompressible materials is a series in terms of the first and second invariant I_1 and I_2 of the right (or left) Cauchy-Green strain tensor [1,2,3]:

$$W = \sum_{i=0}^p \sum_{j=0}^q C_{ij} (I_1 - 3)^i (I_2 - 3)^j \quad ; \quad C_{00} = 0 \quad (4)$$

Two well-known members of this class of strain-energy functions are the Mooney-Rivlin strain energy function (only material parameters C_{10} and C_{01}

unequal to zero) and the neo-Hookean strain energy function (only C_{10} unequal to zero). A remarkable aspect of the latter energy function is the fact that it can also be derived applying Gaussian statistics to a simple model of a network of long chain molecules [4]. For moderate deformations, as occurring in rubber rollers, good fits to experimental data can be obtained using the neo-Hookean energy function. In that case the Mooney-Rivlin form only gives marginally better results [5]. That is why the more simple neo-Hookean function is frequently preferred above the Mooney-Rivlin function.

The constitutive equation given by the neo-Hookean strain energy function can be denoted as

$$\sigma = -p\mathbf{I} + 2C\{\mathbf{B} - \frac{1}{3} \text{tr}(\mathbf{B}) \mathbf{I}\} \quad (5)$$

In this equation the Cauchy stress tensor σ is decomposed into a hydrostatic part, given by the hydrostatic pressure p , and a deviatoric part, depending on the left Cauchy Green strain tensor $\mathbf{B}$. This strain tensor depends on the position vector field $\vec{x}$ as follows:

$$\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^c \quad \text{with } \mathbf{F} = (\bar{\nabla}_0 \vec{x})^c \quad (6)$$

Tensor $\mathbf{F}$ is the deformation tensor, equal to the conjugate gradient of the position vector field, with respect to the current configuration.

Due to the incompressibility of the material, an arbitrary constant pressure can be applied to the body under consideration, contributing to the Cauchy stress tensor without causing any deformation. So, pressure p is determined up to a constant term. This term follows from the boundary conditions applied to the body.

For the sake of convenience Equation (5) is rearranged into

$$\sigma = -\{p + 2C(\frac{1}{3} \text{tr}(\mathbf{B}) - 1)\} \mathbf{I} + 2C\{\mathbf{B} - \mathbf{I}\} = -p^*\mathbf{I} + \tau \quad (7)$$

Pressure p^* now depends not only on the boundary conditions, but also on the left Cauchy Green strain tensor. The evaluation of the extra stress tensor $\tau = 2C\{\mathbf{F} \cdot \mathbf{F}^c - \mathbf{I}\}$ for a known velocity history is the subject of the next section.

3. The Eulerian approach: Derivation of the velocity-stress algorithm

Both the original and the rearranged form of the neo-Hookean constitutive equation express the Cauchy stress tensor in terms of pressure and position. In a Eulerian formulation we must relate the Cauchy tensor to pressure and velocity. In this section an algorithm for the calculation of tensor τ using the velocity vector field is constructed, based on the definition of velocity gradient $\mathbf{L}$. This gradient is related to the deformation tensor and its time derivative:

$$\mathbf{L} = (\vec{\nabla} \vec{u})^c = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1} \Leftrightarrow \mathbf{L} \cdot \mathbf{F} = \dot{\mathbf{F}} \quad (8)$$

Furthermore, time discretization is applied. Three distinct configurations are used. We will distinguish between the reference configuration at time t_0 and the intermediate configuration at time t_n . We assume that stresses and deformations in these configurations are fully known. At time t_{n+1} the configuration is called the current configuration. At this time the stress tensor, just like the deformation tensor, is assumed to be unknown, but an estimate of the velocity field is assumed to be solved previously.

Symbol $(\mathbf{F}^{tot})_n$ denotes the total deformation, starting from the reference configuration t_0 to the intermediate configuration t_n . The total deformation from t_0 to the current configuration t_{n+1} is denoted by $(\mathbf{F}^{tot})_{n+1}$. Tensor $\mathbf{F}_{n+1}$ represents the deformation from t_n to t_{n+1} and is called the *incremental* deformation tensor. These deformation tensors are related to each other:

$$(\mathbf{F}^{tot})_{n+1} = \mathbf{F}_{n+1} \cdot (\mathbf{F}^{tot})_n = (\vec{\nabla}_0 \vec{x}_{n+1})^c \quad (9)$$

A subindex $(n+\alpha)$ with $(0 \leq \alpha \leq 1)$ is used to indicate a configuration at $t_{n+\alpha}$, $t_n \leq t_{n+\alpha} \leq t_{n+1}$.

For small time steps $\Delta t (= t_{n+1} - t_n)$ it is allowed to interpolate the position vector of a particle of the body linearly:

$$\vec{x}_{n+\alpha} \approx \alpha \vec{x}_{n+1} + (1 - \alpha) \vec{x}_n \quad (0 \leq \alpha \leq 1) \quad (10)$$

Based on this linear interpolation the following estimation for the total deformation tensor at $t_{n+\alpha}$ can be derived using Eq.(9):

$$\begin{aligned} (\mathbf{F}^{tot})_{n+\alpha} &= (\bar{\nabla}_0 \vec{x}_{n+\alpha})^C \approx \alpha (\bar{\nabla}_0 \vec{x}_{n+1})^C + (1 - \alpha) (\bar{\nabla}_0 \vec{x}_n)^C \\ &= \alpha (\mathbf{F}^{tot})_{n+1} + (1 - \alpha) (\mathbf{F}^{tot})_n = [\alpha \mathbf{F}_{n+1} + (1 - \alpha) \mathbf{I}] \cdot (\mathbf{F}^{tot})_n \end{aligned} \quad (11)$$

Integration of Equation (8) from time t_n to t_{n+1} using Eq.(9) leads to the following relation:

$$\int_{t_n}^{t_{n+1}} \mathbf{L} \cdot \mathbf{F} dt = \int_{t_n}^{t_{n+1}} \dot{\mathbf{F}} dt = (\mathbf{F}^{tot})_{n+1} - (\mathbf{F}^{tot})_n = [\mathbf{F}_{n+1} - \mathbf{I}] \cdot (\mathbf{F}^{tot})_n \quad (12)$$

The left hand side of Eq.(12) is approximated by

$$\int_{t_n}^{t_{n+1}} \mathbf{L} \cdot \mathbf{F} dt \approx \int_{t_n}^{t_{n+1}} \mathbf{L} dt \cdot (\mathbf{F}^{tot})_{n+\alpha} \quad (0 \leq \alpha \leq 1) \quad (13)$$

where subindex $(n + \alpha)$ denotes the tensor at $t_{n+\alpha}$. According to the Mean Value Theorem there even must be a value of α for which this approximation is exact. The integral in the right hand side of Eq.(13) is approximated by $(\mathbf{L}_{n+1} \Delta t)$, which is allowed for small time steps Δt . After substitution of Eq.(11) into (13), the right hand sides of Eqs.(12) and (13) are equated:

$$\mathbf{L}_{n+1} \Delta t \cdot [\alpha \mathbf{F}_{n+1} + (1 - \alpha) \mathbf{I}] \cdot (\mathbf{F}^{tot})_n = (\mathbf{F}_{n+1} - \mathbf{I}) \cdot (\mathbf{F}^{tot})_n \quad (14)$$

By rearranging this equation the incremental deformation tensor is found to be

$$\mathbf{F}_{n+1} = (\mathbf{I} - \alpha \mathbf{L}_{n+1} \Delta t)^{-1} \cdot (\mathbf{I} + (1 - \alpha) \mathbf{L}_{n+1} \Delta t) \quad (15)$$

According to constitutive Equation (7), stress tensor τ at time t_n and at time t_{n+1} is equal to respectively:

$$\tau_n = 2C [(\mathbf{F}^{tot})_n \cdot (\mathbf{F}^{tot})_n^C - \mathbf{I}] \quad (16)$$

$$\tau_{n+1} = 2C [(\mathbf{F}^{tot})_{n+1} \cdot (\mathbf{F}^{tot})_{n+1}^C - \mathbf{I}] \quad (17)$$

Tensor τ_n is known from previous calculations while τ_{n+1} has to be calculated. Using Eq.(9), an expression for tensor τ_{n+1} can be deduced, which depends on τ_n and $\mathbf{F}_{n+1}$:

$$\tau_{n+1} = \mathbf{F}_{n+1} \cdot \tau_n \cdot \mathbf{F}_{n+1}^C + 2C \{ \mathbf{F}_{n+1} \cdot \mathbf{F}_{n+1}^C - \mathbf{I} \} \quad (18)$$

Now, with Equations (15) and (18), stress tensor τ_{n+1} can be estimated if the velocity gradient $\mathbf{L}_{n+1}$ is known. Equation (15) approximates the incremental deformation tensor $\mathbf{F}_{n+1}$ and Equation (18) uses this approximation for the calculation of stress tensor τ_{n+1} . Parameter α can be regarded as an interpolation parameter. The proper choice for the value of this parameter is the subject of the next section.

3.1 Incremental objectivity of the algorithm

Pinsky, Ortiz, and Pister [6] investigated the objectivity properties of algorithms showing some similarity with the algorithm presented in the previous section. An algorithm is called *incrementally objective* if the following conditions hold:

condition A:

If the incremental deformation tensor $\mathbf{F}_{n+1}$ represents a symmetric stretch tensor, then the rate of rotation tensor $\mathbf{\Omega}$ at time t_{n+1} has to equal the null tensor and vice versa.

condition B:

If the incremental deformation tensor $\mathbf{F}_{n+1}$ represents an orthogonal rotation tensor, then the rate of strain tensor $\mathbf{D}$ at time t_{n+1} has to equal the null tensor and vice versa.

The rate of strain and rate of rotation tensor are defined as the symmetric and skew-symmetric part of the velocity gradient respectively:

$$\mathbf{D}_{n+1} = \frac{1}{2} \{ \mathbf{L}_{n+1} + \mathbf{L}_{n+1}^c \} \quad (19a)$$

$$\mathbf{\Omega}_{n+1} = \frac{1}{2} \{ \mathbf{L}_{n+1} - \mathbf{L}_{n+1}^c \} \quad (19b)$$

In Appendix A it is proven that the proposed velocity-stress algorithm is incrementally objective if and only if parameter α is chosen equal to a half. Some numerical experiments, using the computer program PC-MATLAB [7], have been performed to illustrate the property of incremental objectivity. These experiments are presented in the next two subsections.

3.1.1 Rigid body rotation

An initially undeformed body, made of neo-Hookean material with material parameter $C = 1 \text{ N/mm}^2$, is subjected to a homogeneous rigid body rotation over 360 degrees. This rotation lasts for a total time of 1 s. In the reference

configuration a zero stress condition holds. Using a Cartesian reference system $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$ the axis of rotation is the $\vec{e}_3$ -axis.

The deformation tensor, describing such a rotation with respect to this reference system for all material points, can be denoted as a sum of dyadic products:

$$\mathbf{F}(t) = \cos(2\pi t) \vec{e}_1 \vec{e}_1 - \sin(2\pi t) \vec{e}_1 \vec{e}_2 + \sin(2\pi t) \vec{e}_2 \vec{e}_1 + \cos(2\pi t) \vec{e}_2 \vec{e}_2 + \vec{e}_3 \vec{e}_3 \quad (20)$$

According to Eq.(8), for the velocity gradient holds

$$\mathbf{L}(t) = 2\pi \{ -\vec{e}_1 \vec{e}_2 + \vec{e}_2 \vec{e}_1 \} \quad (21)$$

This velocity gradient is assumed to be known.

Fig.3. Results of calculations on rigid body rotation: component τ_{11} or τ_{22}

Stress tensor τ at $t = 1$ s. is calculated incrementally using the derived algorithm. This is done for several values of the interpolation parameter ($\alpha = 0, 0.25, 0.5, 0.75, 1$). The number of increments N is also varied ($N = 4, 8, 16, 32, 64, 128, 256, 512, 1024$).

For this deformation process, the algorithm, given by Eqs.(15) and (18), is implemented in a PC-MATLAB file. This file is imbedded in Appendix B.1.

In Figure 3 results for stress component τ_{11} are presented. Component τ_{22} turns out to be identical to this component while both τ_{12} and τ_{21} remain equal to zero for all calculations. Theoretically, no stresses may occur due to a rigid body rotation. The computations show that this is only true for α equal to 0.5, the value for which the algorithm is incrementally objective. For values of α smaller than 0.5 extremely high stresses have been found, particularly for a low number of increments.

3.1.2 Uniform isovolumetric elongation

Under the same conditions as used for the rigid body rotation now the body is subjected to a homogeneous isovolumetric elongation. After 1 s. all material particles have reached an elongation of 100% in the $\vec{e}_1$ -direction. In the $\vec{e}_3$ -direction the body does not deform, so, due to the property of incompressibility, the elongation factor in the $\vec{e}_2$ -direction is equal to 0.5 .

A deformation tensor which realizes the desired end configuration is

$$\mathbf{F}(t) = 2^t \vec{e}_1 \vec{e}_1 + 2^{-t} \vec{e}_2 \vec{e}_2 + \vec{e}_3 \vec{e}_3 \quad (22)$$

This deformation tensor results in a constant velocity gradient:

$$\mathbf{L}(t) = (\ln 2) \{ \vec{e}_1 \vec{e}_1 - \vec{e}_2 \vec{e}_2 \} \quad (23)$$

Theoretically only the τ_{11} and τ_{22} components of the extra stress tensor τ are unequal to zero. In the end configuration component τ_{11} is equal to 6.0 N/mm² and component τ_{22} is equal to -1.5 N/mm². The PC-MATLAB file for this problem is given in appendix B.2.

Some results of the simulations are shown in Figures 4 and 5. As can be expected, only the components τ_{11} and τ_{22} are unequal to zero. Again better results have been found for these components if α is chosen equal to a half. This is a strong indication that for α equal to a half the algorithm is not only incrementally objective, but also more accurate even if rigid body rotation is completely absent.

Fig.4. Results of calculations on isovolumetric elongation: component τ_{11}

Fig.5. Results of calculations on isovolumetric elongation: component τ_{22}

4. The Eulerian approach: Convection of stresses

The velocity-stress algorithm, given by Eqs.(15) and (18), requires the knowledge of the velocity gradient $\mathbf{L}_{n+1}$ at time t_{n+1} and of stress tensor τ_n at time t_n and position $\bar{x}_n$. The current stress tensor τ_{n+1} however, is evaluated at time t_{n+1} and position $\bar{x}_{n+1}$. There are several ways to calculate tensor τ_n .

A straightforward method to determine tensor τ_n is by tracing it along the trajectory of the particle that passes through $\bar{x}_{n+1}$ at time t_{n+1} . This method is rather cumbersome to implement in a Finite Element-code and is not without numerical difficulties [8] .

A particularly attractive way to reconstruct tensor τ_n is by a method proposed by Löhner, Morgan, and Zienkiewicz [9] . Essentially their technique, known as the Taylor-Galerkin method, approximates τ_n at particle position $\bar{x}_n$ and time t_n by quantities known at position $\bar{x}_{n+1}$. If we denote a component, with respect to a Cartesian reference system, of stress tensor τ at position $\bar{x}_i$ and time t_j by $T(\bar{x}_i : t_j)$, the value $T(\bar{x}_i : t_j)$ is transported using the transport equation

$$\frac{DT}{Dt} = \frac{\partial T}{\partial t} + \bar{u} \cdot \bar{\nabla} T = 0 \quad (24)$$

According to this equation the following relation holds:

$$T(\bar{x}_n : t_n) = T(\bar{x}_{n+1} : t_{n+1}) \quad (25)$$

For an approximation of the right-hand side of Eq.(25) a Taylor series is used:

$$\begin{aligned} T(\bar{x}_n : t_n) &= T(\bar{x}_{n+1} : t_{n+1}) \\ &= T(\bar{x}_{n+1} : t_n) + \Delta t \left. \frac{\partial T}{\partial t} \right|_{\bar{x}_{n+1} : t_n} + \frac{1}{2} (\Delta t)^2 \left. \frac{\partial^2 T}{\partial t^2} \right|_{\bar{x}_{n+1} : t_n} + O(\Delta t)^3 \end{aligned} \quad (26)$$

For the spatial derivative of T holds, according to the transport equation,

$$\frac{\partial T}{\partial t} = -\vec{u} \cdot \vec{\nabla} T \quad (27)$$

so

$$\begin{aligned} \frac{\partial^2 T}{\partial t^2} &= \frac{\partial}{\partial t} (-\vec{u} \cdot \vec{\nabla} T) = -\frac{\partial \vec{u}}{\partial t} \cdot \vec{\nabla} T - \vec{u} \cdot \vec{\nabla} \left[\frac{\partial T}{\partial t} \right] \\ &= -\frac{\partial \vec{u}}{\partial t} \cdot \vec{\nabla} T + \vec{u} \cdot \vec{\nabla} (\vec{u} \cdot \vec{\nabla} T) \end{aligned} \quad (28)$$

Now, we assume that $(\partial \vec{u} / \partial t)$ equals to the null vector during the transport step. Using the equality

$$\vec{u} \cdot \vec{\nabla} (\vec{u} \cdot \vec{\nabla} T) = \vec{\nabla} \cdot (\vec{u} \vec{u} \cdot \vec{\nabla} T) - \vec{u} (\vec{\nabla} \cdot \vec{u}) \cdot \vec{\nabla} T \quad (29)$$

for $T(\vec{x}_n : t_n)$ can be found

$$\begin{aligned} T(\vec{x}_n : t_n) &= T(\vec{x}_{n+1} : t_n) - \Delta t \vec{u} \cdot \vec{\nabla} T \Big|_{\vec{x}_{n+1} : t_n} + \\ &+ \frac{1}{2} (\Delta t)^2 [\vec{\nabla} \cdot (\vec{u} \vec{u} \cdot \vec{\nabla} T) - \vec{u} (\vec{\nabla} \cdot \vec{u}) \cdot \vec{\nabla} T] \Big|_{\vec{x}_{n+1} : t_n} + \dots \end{aligned} \quad (30)$$

Due to the incompressibility of the neo-Hookean material, the divergence of the velocity equals to zero, so the last term in brackets disappears.

The attractiveness of the Taylor-Galerkin method is displayed nicely if Eq.(30) is posed in a weak form. Using partial integration and the Divergence Theorem, in this weak form only first order derivatives of T occur:

$$\begin{aligned} [w, T(\vec{x}_n : t_n)] &= [w, T(\vec{x}_{n+1} : t_n)] - \Delta t [w, \vec{u} \cdot \vec{\nabla} T] \Big|_{\vec{x}_{n+1} : t_n} + \\ &+ \frac{1}{2} (\Delta t)^2 [\vec{\nabla} w, (\vec{u} \vec{u} \cdot \vec{\nabla} T)] \Big|_{\vec{x}_{n+1} : t_n} + \\ &+ \frac{1}{2} (\Delta t)^2 \langle w, \vec{n} \cdot (\vec{u} \vec{u} \cdot \vec{\nabla} T) \Big|_{\vec{x}_{n+1} : t_n} \rangle \quad \forall w \end{aligned} \quad (31)$$

where $[\cdot, \cdot]$ and $\langle \cdot, \cdot \rangle$ denote the scalar inner product on respectively the domain and the boundary and $\vec{n}$ denotes the unit outward normal vector of the boundary.

The use of Eq.(31) avoids tedious tracing of particles along the characteristics; only values at the current particle positions are required.

5. Implementation aspects

Both the solution for the calculation of stresses using velocities given by Eqs.(15) and (18) and the solution for the convection problem according to Eq.(31) are implemented in the finite element package SEPRAN [10] for the plane strain case. For each time increment, typically the following three steps are taken:

Step 1. Averaging of stresses

As indicated by the weak form (31), the Taylor-Galerkin technique only works for continuous fields. Yet, the extra stress tensor τ_n is evaluated at the element integration points. From this, a piecewise continuous field may be constructed, with discontinuities at the element interfaces. By averaging at the nodal points, continuous approximations of $\tau(\bar{x}_{n+1} : t_n)$ and $\tau(\bar{x}_n : t_n)$ are obtained.

Step 2. Solution of the convection problem

The continuous fields $\tau(\bar{x}_{n+1} : t_n)$ and $\tau(\bar{x}_n : t_n)$ are used to apply the derived weak form, given by Eq.(31). Linear triangular elements are implemented.

Step 3. Solution of $\bar{u}$ and p

After obtaining an approximation for τ_n , the velocities and pressure are solved. This is done using the well-known triangular 7-node Crouzeix-Raviart element, which satisfies the Brezzi-Babuska condition [11]. It has three nodes at the vertices of the triangle, three nodes at the mid-sides, and one node in the centre. All nodes have two velocity degrees of freedom, while at the centroid the pressure and two pressure derivatives are also unknown. An important advantage of this element is the possibility to eliminate the velocity unknowns at the centroid on element level.

In combination with the Crouzeix-Raviart element a penalty method is used. Penalty methods are known to be computationally efficient. The penalty method perturbs the continuity equation by a small term consisting of a penalty parameter ε times the pressure p:

$$\varepsilon p + \vec{\nabla} \cdot \vec{u} = 0 \quad (\text{typically } 10^{-6} < \varepsilon p < 10^{-8}) \quad (32)$$

Doing this, the pressure can be eliminated from the momentum equations and the velocities can be solved separately from the pressure unknowns. The pressure is simply calculated using the perturbed continuity equation.

Both the elimination of the velocity unknowns at the element centroids and the use of a penalty method reduce the computational burden considerably compared to mixed or integrated methods.

A Picard iteration scheme has been implemented. The resulting system of equations to be solved iteratively is given by Eqs. (33) and (34).

$$[q, \varepsilon p_{n+1}^{*i} + \vec{\nabla}_{n+1} \cdot \vec{u}_{n+1}^{*i}] = 0 \quad \forall q \quad (33)$$

$$(\vec{\nabla}_{n+1} \vec{w}; -p_{n+1}^{*i} \mathbf{I} + 4C\Delta t \mathbf{D}_{n+1}^{*i}) = -(\vec{\nabla}_{n+1} \vec{w}; \mathbf{H}_{n+1}^i) + \langle \vec{w}, \vec{t} \rangle \quad \forall \vec{w} \quad (34)$$

$$\text{with } \mathbf{H}_{n+1}^i = \mathbf{F}_{n+1}^i \cdot \tau_n^i (\mathbf{F}^C)_{n+1}^i - 4C\Delta t \mathbf{D}_{n+1}^i + 2C(\mathbf{F}_{n+1}^i \cdot (\mathbf{F}^C)_{n+1}^i - \mathbf{I})$$

In this system of equations a subindex denotes the time at which the tensor, vector, or scalar has to be evaluated. A superindex denotes the iteration number. A semicolon in between parentheses (. ; .) represents the tensor inner product on the domain. Scalar q and vector $\vec{w}$ are respectively a weighting function and a weighting vector. Vector $\vec{t}$ is the surface traction vector.

The rate of strain tensor $\mathbf{D}$, multiplied by $4C\Delta t$, is chosen as a linearisator in the Picard iteration scheme. The Neo-Hookean constitutive equation may be written as $\overset{\vee}{\tau} = 4C\mathbf{D}$, where $(\overset{\vee}{\cdot})$ denotes the Truesdell rate. An implicit approximation for this relation is: $\tau_{n+1} = \mathbf{F}_{n+1} \cdot \tau_n \cdot \mathbf{F}_{n+1}^C + 4C\Delta t \mathbf{D}_{n+1}$. It can be shown that for small Δt , the term $4C\Delta t \mathbf{D}_{n+1} \approx 2C(\mathbf{F}_{n+1} \cdot \mathbf{F}_{n+1}^C - \mathbf{I})$. This motivates the use of $4C\Delta t \mathbf{D}_{n+1}$ as a linearisator.

6. Test problem

In Figure 6 the geometry and the kinematic boundary conditions of a test problem are presented. The symbols u and v represent the velocity components in $\vec{e}_1$ -direction and $\vec{e}_2$ -direction. The neo-Hookean material parameter is chosen equal to 1 N/mm^2 . Incremental objectivity of the velocity-stress algorithm is obtained by choosing the interpolation parameter equal to a half. The maximal total prescribed displacement at the outlet side is taken equal to 0.4 mm .

We have compared our computations to simulations done using the MARC finite element package [12]. In this package, a Lagrangian approach is applied to the modelling of neo-Hookean material. For the MARC calculations linear quadrilateral 5 node elements are used, with two velocity unknowns at the vertices and a pressure unknown at the center node.

Fig.6. Geometry and kinematic boundary conditions of the test problem

The results are presented in Figures 7, 8 and 9. To compare the MARC Lagrangian results with our results the MARC Cauchy stresses are presented on the deformed geometry. Then the stresses in an arbitrary spatial point should be comparable for both the MARC computations and our computations.

Figure 7 contains the resulting Cauchy normal stresses in $\vec{e}_1$ -direction, Figure 8 contains the resulting Cauchy normal stresses in $\vec{e}_2$ -direction, and in Figure 9 the Cauchy shear stresses are shown. Stresses are given in MPa. Good agreement is found.

Fig.7. Cauchy normal stresses in $\vec{e}_1$ -direction

Fig.8. Cauchy normal stresses in $\vec{e}_2$ -direction

Fig.9. Cauchy shear stresses

7. Conclusions

The neo-Hookean constitutive equation for incompressible elastic material behaviour can be used in a Eulerian approach.

The presented algorithm to obtain stresses from the velocity field is sensitive to the choice of the value for the interpolation parameter α . If this value is chosen equal to a half the algorithm is incrementally objective and gives the best results, especially for deformations with large rotations. Even for deformations, where rigid-body rotation is completely absent, better results are found.

To obtain a complete Eulerian formulation the convection problem is solved by a fairly simple algorithm based on a Taylor series for the tensor to be calculated. This algorithm is efficient and reliable.

The results of calculations on a test problem, performed by the FEM-implementation of both algorithms, show good agreement with the results obtained using the MARC finite element package, which applies a Lagrangian approach.

To expand the formulation to an Updated Eulerian approach, an algorithm will have to be proposed for the tracing of moving mesh boundaries.

Acknowledgements

The author wishes to thank especially: dr.ir. Frank Baaijens (Philips Nat.Lab.), for his pleasant and thorough guidance of this work, ir. Paul Slaats (E.U.T./W.F.W.), for his contribution to this work and the very pleasant way he cooperated in the research project, doing his final MSc. thesis, dr.ir. Willem Potze (Philips Nat.Lab.) and dr.ir. Piet Schreurs (E.U.T./W.F.W.), for their support and helpful suggestions and, finally, all members of the group Sastra (Philips Nat.Lab.) for their generous adoption of an 'ambtenaar' in their 'industrial world'.

References

- [1] M. Mooney, A theory of large elastic deformation, J. of Appl. Phys. 11 (1940) 582-592
- [2] R.S. Rivlin, Large elastic deformations of isotropic materials - I. Fundamental concepts, Phil. Trans. R. Soc. A 240 (1948) 459-490
- [3] R.S. Rivlin, Large elastic deformations of isotropic materials - IV. Further developments of the general theory, Phil. Trans. R. Soc. A 241 (1948) 379-397
- [4] L.R.G. Treloar, The mechanics of rubber elasticity, Proc. R. Soc. Lond. A 351 (1976) 301-330
- [5] R.W. Ogden, Large deformation isotropic elasticity - On the correlation of theory and experiment for incompressible rubberlike solids, Proc. R. Soc. Lond. A 326 (1972) 565-584
- [6] P.M. Pinsky, M. Ortiz and K.S. Pister, Numerical integration of rate constitutive equations in finite deformation analysis, Comp. Meth. in Appl. Mech. and Eng. 40 (1983) 137-158
- [7] PC-MATLAB User's Guide, The Math Works, Inc.
- [8] K.W. Morton, A. Priestly and E. Suli, Stability analysis of the Lagrange-Galerkin method with non-exact integration, Oxford University Computing Laboratory, Report 86/14
- [9] R. Löhner, K. Morgan and O.C. Zienkiewicz, The solution of non-linear hyperbolic equation systems by the finite element method, Int. J. Num. Meth. Fluids 4 (1984) 1043-1063
- [10] SEPRAN user manual and SEPRAN programmers guide, ingenieursbureau SEPRAN, Leidschendam, the Netherlands
- [11] C. Cuvelier, A. Segal and A.A. van Steenhoven, *Finite element methods and Navier-Stokes equations* (D. Reidel Publishing Company Dordrecht 1986)
- [12] MARC Manuals A, B, C, and D, Revision K.3 (July 1988), MARC Analysis Research Corporation

Appendix A: Incremental objectivity of the stress algorithm

Definition

The proposed stress algorithm is incrementally objective if the following conditions are satisfied:

Condition A.1

If the incremental deformation tensor $\mathbf{F}_{n+1}$ is symmetric, then the rate of rotation tensor $\mathbf{\Omega}_{n+1} = (\mathbf{L}_{n+1} - \mathbf{L}_{n+1}^C)/2$ must be equal to the null tensor.

Condition A.2

If the rate of rotation tensor $\mathbf{\Omega}_{n+1}$ equals to the null tensor, then the incremental deformation tensor $\mathbf{F}_{n+1}$ must be a symmetric stretch tensor.

Condition B.1

If the incremental deformation tensor $\mathbf{F}_{n+1}$ is orthogonal, then the rate of strain tensor $\mathbf{D}_{n+1} = (\mathbf{L}_{n+1} + \mathbf{L}_{n+1}^C)/2$ must be equal to the null tensor.

Condition B.2

If tensor $\mathbf{D}_{n+1}$ is equal to the null tensor, then tensor $\mathbf{F}_{n+1}$ must be an orthogonal rigid body rotation tensor.

Proposition

For all tensors $\mathbf{A}$ and $\mathbf{B}$ for which (i) $(\mathbf{I} + \mathbf{B})$ is regular
(ii) $\mathbf{A}^{-1} \cdot \mathbf{B} = \mathbf{B} \cdot \mathbf{A}^{-1}$

holds

(A) $\mathbf{M} = (\mathbf{I} + \mathbf{A}) \cdot (\mathbf{I} + \mathbf{B})^{-1} = (\mathbf{I} + \mathbf{B})^{-1} \cdot (\mathbf{I} + \mathbf{A})$

(B) $\mathbf{M}$ is orthogonal ($\mathbf{M}^{-1} = \mathbf{M}^C$) if and only if

$$\mathbf{A} + \mathbf{A}^C + \mathbf{A}^C \cdot \mathbf{A} = \mathbf{B} + \mathbf{B}^C + \mathbf{B}^C \cdot \mathbf{B}$$

Proof: (A) $(\mathbf{I} + \mathbf{A}).(\mathbf{I} + \mathbf{B})^{-1} = (\mathbf{I} + \mathbf{B})^{-1} + ((\mathbf{I} + \mathbf{B}).\mathbf{A}^{-1})^{-1}$
 $= (\mathbf{I} + \mathbf{B})^{-1} + (\mathbf{A}^{-1} + \mathbf{B}.\mathbf{A}^{-1})^{-1} = (\mathbf{I} + \mathbf{B})^{-1} + (\mathbf{A}^{-1} + \mathbf{A}^{-1}.\mathbf{B})^{-1}$
 $= (\mathbf{I} + \mathbf{B})^{-1} + (\mathbf{I} + \mathbf{B})^{-1}.\mathbf{A} = (\mathbf{I} + \mathbf{B})^{-1}.\mathbf{A}$

(B) $\mathbf{M}^{-1} = (\mathbf{I} + \mathbf{B}).(\mathbf{I} + \mathbf{A})^{-1} ; \quad \mathbf{M}^C = (\mathbf{I} + \mathbf{B})^C.(\mathbf{I} + \mathbf{A})^C$
 $\mathbf{M}^{-1} = \mathbf{M}^C \Leftrightarrow (\mathbf{I} + \mathbf{B})^C.(\mathbf{I} + \mathbf{B}) = (\mathbf{I} + \mathbf{A})^C.(\mathbf{I} + \mathbf{A})$
 $\Leftrightarrow \mathbf{B} + \mathbf{B}^C + \mathbf{B}^C.\mathbf{B} = \mathbf{A} + \mathbf{A}^C + \mathbf{A}^C.\mathbf{A} \quad \blacksquare$

Theorem

The stress algorithm is incrementally objective if and only if the interpolation parameter α is chosen equal to 1/2.

Proof:

The theorem is proven using the proposition above and the following equations for $\mathbf{L}_{n+1}$ and $\mathbf{F}_{n+1}$, based on Eq.(15):

$$\begin{aligned} \mathbf{L}_{n+1} &= \frac{1}{\Delta t} (\mathbf{F}_{n+1} - \mathbf{I}).((1 - \alpha)\mathbf{I} + \alpha\mathbf{F}_{n+1})^{-1} \\ &= -\frac{1}{\Delta t} \frac{1}{1 - \alpha} (\mathbf{I} - \mathbf{F}_{n+1}).\left(\mathbf{I} + \frac{\alpha}{(1 - \alpha)} \mathbf{F}_{n+1}\right)^{-1} \end{aligned} \quad (\text{A.1})$$

$$\mathbf{F}_{n+1} = (\mathbf{I} - \alpha\mathbf{L}_{n+1}\Delta t)^{-1}.(\mathbf{I} + (1 - \alpha)\mathbf{L}_{n+1}\Delta t) \quad (\text{A.2})$$

Condition A.1

If $\mathbf{F}_{n+1}$ is symmetric then also $\mathbf{L}_{n+1}$ is symmetric:

$$\begin{aligned} \mathbf{L}_{n+1}^C &= -\frac{1}{(1 - \alpha)\Delta t} \left(\mathbf{I} + \frac{\alpha}{1 - \alpha} \mathbf{F}_{n+1}^C\right)^{-1}.(\mathbf{I} - \mathbf{F}_{n+1}^C) \\ &= -\frac{1}{(1 - \alpha)\Delta t} \left(\mathbf{I} + \frac{\alpha}{1 - \alpha} \mathbf{F}_{n+1}\right)^{-1}.(\mathbf{I} - \mathbf{F}_{n+1}) \\ &= -\frac{1}{(1 - \alpha)\Delta t} (\mathbf{I} - \mathbf{F}_{n+1}).\left(\mathbf{I} + \frac{\alpha}{1 - \alpha} \mathbf{F}_{n+1}\right)^{-1} = \mathbf{L}_{n+1} \quad (\text{proposition A}) \end{aligned}$$

So, Ω_{n+1} vanishes and condition A.1 holds for all values of α .

Condition A.2

If $\mathbf{L}_{n+1}$ is symmetric then

$$\begin{aligned}\mathbf{F}_{n+1}^C &= (\mathbf{I} + (1 - \alpha)\mathbf{L}_{n+1}^C \Delta t) \cdot (\mathbf{I} - \alpha\mathbf{L}_{n+1}^C \Delta t)^{-1} \\ &= (\mathbf{I} + (1 - \alpha)\mathbf{L}_{n+1} \Delta t) \cdot (\mathbf{I} - \alpha\mathbf{L}_{n+1} \Delta t)^{-1} \\ &= (\mathbf{I} - \alpha\mathbf{L}_{n+1} \Delta t)^{-1} \cdot (\mathbf{I} + (1 - \alpha)\mathbf{L}_{n+1} \Delta t) = \mathbf{F}_{n+1} \quad (\text{proposition A})\end{aligned}$$

So, $\mathbf{F}_{n+1}$ is also symmetric. Condition A.2 holds for all values of α .

Condition B.1

If $\mathbf{F}_{n+1}$ is orthogonal then $\mathbf{F}_{n+1}^{-1} = \mathbf{F}_{n+1}^C$. Tensor $\mathbf{F}_{n+1}$ equals

$$\mathbf{F}_{n+1} = (\mathbf{I} - \alpha\mathbf{L}_{n+1} \Delta t)^{-1} \cdot (\mathbf{I} + (1 - \alpha)\mathbf{L}_{n+1} \Delta t)$$

Using $\mathbf{A} = (1 - \alpha)\mathbf{L}_{n+1} \Delta t$ and $\mathbf{B} = -\alpha\mathbf{L}_{n+1} \Delta t$, according to proposition (B) for $\mathbf{F}_{n+1}$ must hold

$$\begin{aligned}(1 - \alpha)\Delta t (\mathbf{L}_{n+1} + \mathbf{L}_{n+1}^C) + (1 - \alpha)^2 \Delta t^2 (\mathbf{L}_{n+1}^C \cdot \mathbf{L}_{n+1}) &= \\ = -\alpha \Delta t (\mathbf{L}_{n+1} + \mathbf{L}_{n+1}^C) + \alpha^2 \Delta t^2 (\mathbf{L}_{n+1}^C \cdot \mathbf{L}_{n+1}) \\ \Leftrightarrow (\mathbf{L}_{n+1} + \mathbf{L}_{n+1}^C) &= (\alpha^2 - (1 - \alpha)^2) \Delta t (\mathbf{L}_{n+1}^C \cdot \mathbf{L}_{n+1})\end{aligned}$$

Tensor $\mathbf{D}_{n+1} = (\mathbf{L}_{n+1} + \mathbf{L}_{n+1}^C)/2$ only vanishes if $\alpha^2 - (1 - \alpha)^2 = 0$

So, to satisfy condition B.1, α has to be chosen equal to 1/2

Condition B.2

If $\mathbf{D}_{n+1}$ is equal to the null tensor, $\mathbf{L}_{n+1}$ is equal to $-\mathbf{L}_{n+1}^C$. For $\mathbf{F}_{n+1}^{-1}$ and $\mathbf{F}_{n+1}^C$ holds

$$\begin{aligned}\mathbf{F}_{n+1}^{-1} &= (\mathbf{I} + (1 - \alpha)\mathbf{L}_{n+1}\Delta t)^{-1}(\mathbf{I} - \alpha\mathbf{L}_{n+1}\Delta t) \\ \mathbf{F}_{n+1}^C &= (\mathbf{I} + (1 - \alpha)\mathbf{L}_{n+1}^C\Delta t)(\mathbf{I} - \alpha\mathbf{L}_{n+1}^C\Delta t)^{-1} \\ &= (\mathbf{I} + (\alpha - 1)\mathbf{L}_{n+1}\Delta t)(\mathbf{I} + \alpha\mathbf{L}_{n+1}\Delta t)^{-1} \\ &= (\mathbf{I} + \alpha\mathbf{L}_{n+1}\Delta t)^{-1}(\mathbf{I} + (\alpha - 1)\mathbf{L}_{n+1}\Delta t) \quad (\text{proposition A})\end{aligned}$$

Tensor $\mathbf{F}_{n+1}$ is orthogonal if $\mathbf{F}_{n+1}^{-1}$ equals $\mathbf{F}_{n+1}^C$ and this is only true if α equals 1/2.

■

Appendix B: PC-MATLAB files

B.1 Rigid body rotation

```
%  
% Example 1: numerical integration using the stress-velocity  
% algorithm, applied to a rigid body rotation  
% of 360 degrees  
-----
```

```
constitutive equation: Neo-Hookean equation
```

$$\sigma = -p I + \tau$$

```
%  
with: p      = (pseudo) hydrostatic pressure,  
sigma = Cauchy stress tensor  
I      = Unity tensor  
tau    = 2C*(B-I)  
        C = material parameter  
        B = left Cauchy-Green strain tensor
```

```
calculation of tau(n+1), with tau(n) and L(n+1) given  
-----
```

```
varied parameters:
```

- interpolation parameter alpha
 (alpha = 0.5 -> incrementally objective)
- number of increments

```
-> N      resp.: 4, 8, 16, 32, 64, 128, 256, 512  
-> alpha  resp.: 0.00, 0.25, 0.50, 0.75, 1.00
```

```
deformation:  in N increments a rigid body rotation over  
               360 degrees is realized
```

parameters:

PHI = total rigid body rotation angle
T = total time in which the end
configuration is reached
dt = time step = T/N
N = number of increments
alpha = interpolation parameter
C = material parameter
T0 = initial tau matrix

PHI = 2*pi ;
T = 1.0 ;
C = 1.0 ;

T0 = [0 0
0 0];

I = eye(2);

aa = [0.00 0.25 0.50 0.75 1.00];
NN = [2 4 8 16 32 64 128 256 512 1024];

fprintf('\nPC Matlab calculation of tensor Tau ')

fprintf('\n-----')

fprintf('\n')

fprintf('\ntotal elapsed time : %5.2f s. ',T);

fprintf('\ntotal angle of rotation : %5.2f rad. ',PHI);

fprintf('\nmaterial parameter : %5.2f N/mm2 ',C);

fprintf('\n\n\n\n')

for iii=1:5;
alpha=aa(iii);

for ii=1:10;
N = NN(ii);

Input of incremental L matrix

L = [0 -PHI/T
PHI/T 0];

Approximation of the incremental deformation tensor

dt = T/N;

% % % % % % % % % %

Category	Percentage
Category 1	10%
Category 2	20%
Category 3	30%
Category 4	40%
Category 5	50%
Category 6	60%
Category 7	70%
Category 8	80%
Category 9	90%
Category 10	100%

Category	Percentage
1. General	100%
2. Specific	100%
3. Detailed	100%
4. Comprehensive	100%
5. In-depth	100%
6. Thorough	100%
7. Extensive	100%
8. Exhaustive	100%
9. Comprehensive	100%
10. Detailed	100%
11. Specific	100%
12. General	100%
13. Comprehensive	100%
14. Detailed	100%
15. Specific	100%
16. General	100%
17. Comprehensive	100%
18. Detailed	100%
19. Specific	100%
20. General	100%
21. Comprehensive	100%
22. Detailed	100%
23. Specific	100%
24. General	100%
25. Comprehensive	100%
26. Detailed	100%
27. Specific	100%
28. General	100%
29. Comprehensive	100%
30. Detailed	100%
31. Specific	100%
32. General	100%
33. Comprehensive	100%
34. Detailed	100%
35. Specific	100%
36. General	100%
37. Comprehensive	100%
38. Detailed	100%
39. Specific	100%
40. General	100%
41. Comprehensive	100%
42. Detailed	100%
43. Specific	100%
44. General	100%
45. Comprehensive	100%
46. Detailed	100%
47. Specific	100%
48. General	100%
49. Comprehensive	100%
50. Detailed	100%
51. Specific	100%
52. General	100%
53. Comprehensive	100%
54. Detailed	100%
55. Specific	100%
56. General	100%
57. Comprehensive	100%
58. Detailed	100%
59. Specific	100%
60. General	100%
61. Comprehensive	100%
62. Detailed	100%
63. Specific	100%
64. General	100%
65. Comprehensive	100%
66. Detailed	100%
67. Specific	100%
68. General	100%
69. Comprehensive	100%
70. Detailed	100%
71. Specific	100%
72. General	100%
73. Comprehensive	100%
74. Detailed	100%
75. Specific	100%
76. General	100%
77. Comprehensive	100%
78. Detailed	100%
79. Specific	100%
80. General	100%
81. Comprehensive	100%
82. Detailed	100%
83. Specific	100%
84. General	100%
85. Comprehensive	100%
86. Detailed	100%
87. Specific	100%
88. General	100%
89. Comprehensive	100%
90. Detailed	100%
91. Specific	100%
92. General	100%
93. Comprehensive	100%
94. Detailed	100%
95. Specific	100%
96. General	100%
97. Comprehensive	100%
98. Detailed	100%
99. Specific	100%
100. General	100%

Category	Percentage
1	100%
2	100%
3	100%
4	100%
5	100%
6	100%
7	100%
8	100%
9	100%
10	100%
11	100%
12	100%
13	100%
14	100%
15	100%
16	100%
17	100%
18	100%
19	100%
20	100%
21	100%
22	100%
23	100%
24	100%
25	100%
26	100%
27	100%
28	100%
29	100%
30	100%
31	100%
32	100%
33	100%
34	100%
35	100%
36	100%
37	100%
38	100%
39	100%
40	100%
41	100%
42	100%
43	100%
44	100%
45	100%
46	100%
47	100%
48	100%
49	100%
50	100%

Category	Percentage
1. General population	100%
2. Urban population	100%
3. Rural population	100%
4. Male population	100%
5. Female population	100%
6. Age group 15-64	100%
7. Age group 65+	100%
8. Education level	100%
9. Income level	100%
10. Employment status	100%
11. Health status	100%
12. Social status	100%
13. Cultural status	100%
14. Political status	100%
15. Religious status	100%
16. Ethnic status	100%
17. Nationality	100%
18. Citizenship	100%
19. Residence	100%
20. Migration	100%
21. Urbanization	100%
22. Ruralization	100%
23. Industrialization	100%
24. Modernization	100%
25. Globalization	100%
26. Internationalization	100%
27. Transnationalization	100%
28. Multinationalization	100%
29. Supranationalization	100%
30. Transnationalization	100%
31. Multinationalization	100%
32. Supranationalization	100%
33. Transnationalization	100%
34. Multinationalization	100%
35. Supranationalization	100%
36. Transnationalization	100%
37. Multinationalization	100%
38. Supranationalization	100%
39. Transnationalization	100%
40. Multinationalization	100%
41. Supranationalization	100%
42. Transnationalization	100%
43. Multinationalization	100%
44. Supranationalization	100%
45. Transnationalization	100%
46. Multinationalization	100%
47. Supranationalization	100%
48. Transnationalization	100%
49. Multinationalization	100%
50. Supranationalization	100%
51. Transnationalization	100%
52. Multinationalization	100%
53. Supranationalization	100%
54. Transnationalization	100%
55. Multinationalization	100%
56. Supranationalization	100%
57. Transnationalization	100%
58. Multinationalization	100%
59. Supranationalization	100%
60. Transnationalization	100%
61. Multinationalization	100%
62. Supranationalization	100%
63. Transnationalization	100%
64. Multinationalization	100%
65. Supranationalization	100%
66. Transnationalization	100%
67. Multinationalization	100%
68. Supranationalization	100%
69. Transnationalization	100%
70. Multinationalization	100%
71. Supranationalization	100%
72. Transnationalization	100%
73. Multinationalization	100%
74. Supranationalization	100%
75. Transnationalization	100%
76. Multinationalization	100%
77. Supranationalization	100%
78. Transnationalization	100%
79. Multinationalization	100%
80. Supranationalization	100%
81. Transnationalization	100%
82. Multinationalization	100%
83. Supranationalization	100%
84. Transnationalization	100%
85. Multinationalization	100%
86. Supranationalization	100%
87. Transnationalization	100%
88. Multinationalization	100%
89. Supranationalization	100%
90. Transnationalization	100%
91. Multinationalization	100%
92. Supranationalization	100%
93. Transnationalization	100%
94. Multinationalization	100%
95. Supranationalization	100%
96. Transnationalization	100%
97. Multinationalization	100%
98. Supranationalization	100%
99. Transnationalization	100%
100. Multinationalization	100%

Category	Percentage
Category 1	100%
Category 2	100%
Category 3	100%
Category 4	100%
Category 5	100%
Category 6	100%
Category 7	100%
Category 8	100%
Category 9	100%
Category 10	100%
Category 11	100%
Category 12	100%
Category 13	100%
Category 14	100%
Category 15	100%
Category 16	100%
Category 17	100%
Category 18	100%
Category 19	100%
Category 20	100%
Category 21	100%
Category 22	100%
Category 23	100%
Category 24	100%
Category 25	100%
Category 26	100%
Category 27	100%
Category 28	100%
Category 29	100%
Category 30	100%
Category 31	100%
Category 32	100%
Category 33	100%
Category 34	100%
Category 35	100%
Category 36	100%
Category 37	100%
Category 38	100%
Category 39	100%
Category 40	100%
Category 41	100%
Category 42	100%
Category 43	100%
Category 44	100%
Category 45	100%
Category 46	100%
Category 47	100%
Category 48	100%
Category 49	100%
Category 50	100%
Category 51	100%
Category 52	100%
Category 53	100%
Category 54	100%
Category 55	100%
Category 56	100%
Category 57	100%
Category 58	100%
Category 59	100%
Category 60	100%
Category 61	100%
Category 62	100%
Category 63	100%
Category 64	100%
Category 65	100%
Category 66	100%
Category 67	100%
Category 68	100%
Category 69	100%
Category 70	100%
Category 71	100%
Category 72	100%
Category 73	100%
Category 74	100%
Category 75	100%
Category 76	100%
Category 77	100%
Category 78	100%
Category 79	100%
Category 80	100%
Category 81	100%
Category 82	100%
Category 83	100%
Category 84	100%
Category 85	100%
Category 86	100%
Category 87	100%
Category 88	100%
Category 89	100%
Category 90	100%
Category 91	100%
Category 92	100%
Category 93	100%
Category 94	100%
Category 95	100%
Category 96	100%
Category 97	100%
Category 98	100%
Category 99	100%
Category 100	100%

% % % % %

[illegible]

	%	%	%	%
1	100	100	100	100
2	100	100	100	100
3	100	100	100	100
4	100	100	100	100
5	100	100	100	100
6	100	100	100	100
7	100	100	100	100
8	100	100	100	100
9	100	100	100	100
10	100	100	100	100
11	100	100	100	100
12	100	100	100	100
13	100	100	100	100
14	100	100	100	100
15	100	100	100	100
16	100	100	100	100
17	100	100	100	100
18	100	100	100	100
19	100	100	100	100
20	100	100	100	100
21	100	100	100	100
22	100	100	100	100
23	100	100	100	100
24	100	100	100	100
25	100	100	100	100
26	100	100	100	100
27	100	100	100	100
28	100	100	100	100
29	100	100	100	100
30	100	100	100	100
31	100	100	100	100
32	100	100	100	100
33	100	100	100	100
34	100	100	100	100
35	100	100	100	100
36	100	100	100	100
37	100	100	100	100
38	100	100	100	100
39	100	100	100	100
40	100	100	100	100
41	100	100	100	100
42	100	100	100	100
43	100	100	100	100
44	100	100	100	100
45	100	100	100	100
46	100	100	100	100
47	100	100	100	100
48	100	100	100	100
49	100	100	100	100
50	100	100	100	100
51	100	100	100	100
52	100	100	100	100
53	100	100	100	100
54	100	100	100	100
55	100	100	100	100
56	100	100	100	100
57	100	100	100	100
58	100	100	100	100
59	100	100	100	100
60	100	100	100	100
61	100	100	100	100
62	100	100	100	100
63	100	100	100	100
64	100	100	100	100
65	100	100	100	100
66	100	100	100	100
67	100	100	100	100
68	100	100	100	100
69	100	100	100	100
70	100	100	100	100
71	100	100	100	100
72	100	100	100	100
73	100	100	100	100
74	100	100	100	100
75	100	100	100	100
76	100	100	100	100
77	100	100	100	100
78	100	100	100	100
79	100	100	100	100
80	100	10		

%%
%%
%%
%%
%%

%%

%

end.

B.2 Isovolumetric Elongation

Example 2: numerical integration using the stress-velocity algorithm, applied to an isovolumetric elongation of 100%

constitutive equation: Neo-Hookean equation

$$\sigma = -p I + \tau$$

with: p = (pseudo) hydrostatic pressure,
 σ = Cauchy stress tensor
 I = Unity tensor
 $\tau = 2C*(B-I)$
 C = material parameter
 B = left Cauchy-Green strain tensor

calculation of $\tau(n+1)$, with $\tau(n)$ and $L(n+1)$ given

varied parameters:

- interpolation parameter α
 ($\alpha = 0.5 \rightarrow$ incrementally objective)
- number of increments

$\rightarrow N$ resp.: 4, 8, 16, 32, 64, 128, 256, 512, 1024
 $\rightarrow \alpha$ resp.: 0.00, 0.25, 0.50, 0.75, 1.00

deformation: in N increments an elongation of 100% is realized

parameters:

Λ = total elongation factor
 T = total time in which the end configuration is reached
 dt = time step = T/N
 N = number of increments
 α = interpolation parameter
 C = material parameter
 T_0 = initial τ matrix

$\Lambda = 2.0$;
 $T = 1.0$;

```

C      = 1.0      ;

T0     = [ 0      0
           0      0 ];

I      = eye(2);

aa     = [ 0.00 0.25 0.50 0.75 1.00 ];
NN     = [ 2 4 8 16 32 64 128 256 512 1024 ];

fprintf('\nPC Matlab calculation of tensor Tau ')
fprintf('\n-----')
fprintf('\n')
fprintf('\ntotal elapsed time      : %5.2f s.      ',T);
fprintf('\ntotal elongation factor : %5.2f [-]      ',LABDA);
fprintf('\nmaterial parameter          : %5.2f N/mm2 ',C);
fprintf('\n\n\n\n')

for iii=1:5;
    alpha=aa(iii);

    for ii=1:10;
        N  = NN(ii);

        Input of incremental L matrix
        -----

        L = log(LABDA)* [ 1  0
                          0 -1 ];

        Approximation of the incremental deformation tensor
        -----

        dt = T/N;

        Fn1 = (inv(I - alpha*dt*L))*(I+(1-alpha)*dt*L);

        Tn  = T0;

        Calculation of tau(n+1)
        -----

        for i=1:N;

            Tn1=Fn1*Tn*Fn1'+ 2*C*(Fn1*Fn1'-I);

            Tn = Tn1;
    
```

```
end;

%
    fprintf('-----')
    fprintf('\nIsovolumetric Elongation  ')
    fprintf('\n-----')
    fprintf('\nalpha          = %5.2f  ',alpha);
    fprintf('\nnumber of increments = %3.0f \n',i)    ;
    fprintf('-----\n')

    print tau
    -----

    Tn

end;
fprintf('\n')
end;

end.
```