Testing of Johnson-Cook material model VPJC in EUROPLEXUS

Folco Casadei
Vegard Aune
Georgios Valsamos
Martin Larcher
2015

CUBEC02
TIME: 2.00000E+00   STEP: 100000
Testing of Johnson-Cook material model VPJC in EUROPLEXUS
Testing of the Johnson-Cook material model VPJC in EUROPLEXUS

F. Casadei, Vegard Aune, Georgios Valsamos, Martin Larcher

December 15, 2015

Contents

1 Introduction 2

2 The modified Johnson-Cook material model 2
   2.1 Constitutive law .................................................. 2
   2.2 Temperature softening .......................................... 3
   2.3 Gauss point failure and element erosion ...................... 4

3 Numerical tests 4
   3.1 Uniaxial tests at constant velocity .......................... 4
      3.1.1 Analytical static solution ................................. 5
      3.1.2 Calculations with continuum elements in 3D ............ 6
      3.1.3 Calculations with continuum elements in 2D ............ 8
      3.1.4 Calculations with 3D shells ............................... 9
      3.1.5 Calculations with beam/bar elements .................... 11
   3.2 Uniaxial tests at constant strain rate ...................... 12
      3.2.1 Stress ........................................................... 12
      3.2.2 Strain ........................................................... 12
      3.2.3 Strain rate ..................................................... 13
      3.2.4 Equivalent stress ............................................ 13
      3.2.5 Equivalent plastic strain .................................. 13
      3.2.6 Relative displacement ...................................... 14
      3.2.7 Choice of the test parameters ............................. 16
      3.2.8 Analytical solution .......................................... 17
      3.2.9 Calculations with 3D continuum elements ................ 18

4 Conclusions and future work 21

References 22

Appendix A — Coordinate transformations 24

Appendix B — Input files 31

List of input files 55

List of Figures

1 Uniaxial stress problem. ............................................. 4
2 Some results of test case CUBE02. ................................. 7
3 Some results of test case CUBE05. ................................. 21
1 Introduction

This document presents some numerical tests for the verification of the Johnson-Cook material model (VPJC) recently introduced in the EUROPLEXUS code (EPX).

EUROPLEXUS [1] is a computer code jointly developed by the French Commissariat à l’Energie Atomique (CEA DMT Saclay) and by EC-JRC. The code application domain is the numerical simulation of fast transient phenomena such as explosions, crashes and impacts in complex three-dimensional fluid-structure systems.

The Cast3m [2] software from CEA is used as a pre-processor to EPX when it is necessary to generate complex meshes.

2 The modified Johnson-Cook material model

We briefly recall the characteristics of the modified Johnson-Cook material model (VPJC material in EPX) adopted in the present report. This is a Von Mises elasto-thermo-viscoplastic material with non-linear isotropic hardening governed by a Voce saturation behaviour and a Cockcroft-Latham failure criterion, elastic predictor and return mapping algorithm. The material model was developed at NTNU (Trondheim, Norway), see [3].

The original Johnson-Cook model was first introduced in reference [4]. The so-called “modified” Johnson Cook material law, in which the strain-rate sensitivity term is adjusted so as to avoid non-physical softening, was introduced in reference [5] (see also [6]). The Voce saturation type of hardening was proposed in [7].

2.1 Constitutive law

The expression of the constitutive law is the following:

$$
\sigma_Y = \left[ A + Q_1 \left( 1 - e^{-C_1 p} \right) + Q_2 \left( 1 - e^{-C_2 p} \right) \right] \left( 1 + \dot{p}^* \right) C (1 - T^{*m})
$$

and is the product of three factors (from left to right): a strain hardening term (in square brackets), a strain-rate hardening term and a temperature softening term. The symbols indicate the following:

- $\sigma_Y$ is the current yield stress of the material.
- $A$ is the initial yield stress of the material, sometimes also indicated as $\sigma_0$.
- $p$ is the equivalent (or cumulated) plastic strain, i.e. the energy-conjugated variable to the equivalent stress.
- $\dot{p}$ is the equivalent plastic strain rate.
\( \dot{p}^* \) is the dimensionless plastic strain rate \( \dot{p}^* = \dot{p}/\dot{p}_0 \), with \( \dot{p}_0 \) as the user-defined reference strain rate.

\( T^* \) is the dimensionless temperature \( T^* = (T - T_r) / (T_m - T_r) \), with \( T \) the absolute temperature, \( T_r \) the absolute room temperature and \( T_m \) the absolute melting temperature.

\( Q_1, C_1, Q_2 \) and \( C_2 \) are material constants used in the first factor on the right-hand side of the material law (strain-hardening term).

\( C \) is a material constant, the exponent appearing in the second factor, which represents the strain-rate hardening.

\( m \) is a material constant, the exponent appearing in the third factor, which represents the temperature softening.

### 2.2 Temperature softening

The last term of the constitutive law (1) accounts for the thermal softening of the yield stress at elevated temperatures. However, the evolution of the temperature remains to be established. The heat transfer is modelled by assuming adiabatic conditions. This implies that there is no heat transfer into or out of the system (typically each individual element in the numerical model) during plastic straining.

The plastic energy dissipation \( D_p \) per unit volume in the form of heat \((W/m^3)\) is given by:

\[
D_p = \chi \sigma_{eq} \dot{p} = \rho C_T \dot{T}
\]

where:

- \( \chi \) is the Taylor-Quinney coefficient, i.e. the fraction of plastic power that is converted to heat. The remaining fraction \((1 - \chi)\) is assumed to remain in the material due to structural rearrangements (i.e. elastic “fields” around dislocations). A value of 0.9 is commonly found in the literature for metals, however, a conservative choice would be 1.0 which implies maximum thermal softening. In reality the \( \chi \)-value will not be constant, however, advanced experimental techniques are necessary to determine the evolution of this parameter. Moreover, one should always keep in mind that, while strain hardening (i.e. work hardening due to creation of dislocations) implies an increase of strength locally in the material and distributes the plasticity, softening results in localization of plasticity.

- \( \sigma_{eq} \) is the equivalent stress.

- \( \dot{p} \) is the equivalent plastic strain rate.

- \( \rho \) is the material density.

- \( C_T \) is the material heat capacity.

- \( \dot{T} \) is the temperature rate due to adiabatic heating.

From the above expression, the temperature rate \( \dot{T} \) is obtained:

\[
\dot{T} = \frac{D_p}{\rho C_T} = \frac{\chi \sigma_{eq} \dot{p}}{\rho C_T}
\]

and then this value is integrated in time at each Gauss point to obtain the current temperature at the point. The initial temperature is set to the room temperature \( T_r \) at each Gauss point. If during the calculation the temperature at a Gauss point reaches the melting temperature \( T_m \), the Gauss point is assumed to fail.
2.3 Gauss point failure and element erosion

The Cockcroft-Latham fracture criterion based on plastic work per unit volume is assumed, see [8]. Material failure takes place at a Gauss point when a damage parameter $D$ reaches the (user-defined) damage threshold $D_C$ (with $0 < D_c \leq 1$). The value of $D_c$ should be set to 1 when not considering damage softening. The damage is computed according to the following expression:

$$D = \frac{W}{W_c} = \frac{1}{W_c} \int_0^p \langle \sigma_1 \rangle dp$$

(4)

where:

- $\sigma_1$ is the maximum principal stress at the Gauss point.
- The expression $\langle \sigma_1 \rangle$ is equivalent to the function $\max(0, \sigma_1)$, which implies that only positive values of the maximum principal stress $\sigma_1$ (i.e. tensile stress) contribute to the damage evolution.
- $W_c$ is the failure material parameter, which can be found by integrating the major principal stress in a uniaxial tension test during the entire equivalent plastic strain path until the plastic strain at failure $p_f$. In this case (uniaxial traction) the major principal stress is just the (longitudinal) stress.

An element’s Gauss point is considered as failed if $D \geq D_c$, i.e. if the damage reaches the chosen threshold. If the “erosion” algorithm is activated (see [1], page A.30, Section 5.4, keyword EROS), then an element is eroded as soon as a chosen fraction (see ldam parameter of the EROS keyword) of its Gauss points reach failure.

3 Numerical tests

3.1 Uniaxial tests at constant velocity

The simplest possible test is a uniaxial test for a single element. A unit cube centered in the origin and aligned with the global axes is subjected to an imposed displacement (linear in time) of the faces normal to the $z$-axis (blue axis in Figure 1 (a)). The degrees of freedom along the other axes are left free. The “lower” face of the element (of nodes 1, 2, 3 and 4) is displaced along the negative $z$-axis while the “upper” face (of nodes 5, 6, 7 and 8) is displaced along the positive $z$-axis. The initial and final shape of the cube are shown in Figure 1 (a) and (b), respectively.

![Uniaxial stress problem](image)

Figure 1: Uniaxial stress problem.

A first series of tests is done at constant velocity. The imposed (relative) velocity is 1 m/s (i.e. $-0.5$ m/s on the lower face and $+0.5$ m/s on the upper face). The final time of the calculation is 2 s, when the cube reaches a length of 3 m along the $z$-axis. This corresponds to an engineering strain of 200 % ($\epsilon = 2.000$) and to a natural strain of 109.9 % ($e = 1.099$).
This test is similar to the one proposed in reference [9] where, however, the elongation occurs along the \(x\)-axis instead of the \(z\)-axis, the calculation is stopped after an elongation of 1 m instead of 2 m, and the material is von Mises rate-independent elasto-plastic with isotropic hardening (VM23) instead of VPJC.

Although the imposed velocity is constant in this test, the strain rate is not constant due to the non-linear nature of large deformations. In order to simplify the test and to obtain an approximated analytical solution to be used as a reference, we assume a VPJC material insensitive to strain rate by setting \(C = 0\). Also the temperature softening is neglected (\(m = 0\)).

### 3.1.1 Analytical static solution

Under these assumptions, the analytical solution of the equivalent static problem (i.e., by neglecting any dynamic or inertia effects, which is plausible since the imposed velocity is rather low) is obtained as follows. The longitudinal (natural) strain at the final elongation is:

\[
\varepsilon_z = \ln \left( \frac{L}{L_0} \right) = \ln \frac{3}{1} = 1.0986 \tag{5}
\]

At such large strains the elastic deformations can be neglected compared with plastic deformations (which occur at constant volume), so that the volume remains approximately constant. Therefore, the lateral strains are:

\[
\varepsilon_x = \varepsilon_y \approx -0.5 \varepsilon_z = -0.5 \times 1.0986 = -0.54931 \tag{6}
\]

In a uniaxial stress case the equivalent plastic strain \(p\) is simply equal to the longitudinal plastic strain:

\[
p = \varepsilon_z^{pl} \tag{7}
\]

If we again neglect the elastic part of the deformation, we have:

\[
p = \varepsilon_z^{pl} \approx \varepsilon_z = 1.0986 \tag{8}
\]

Since the stress is uniaxial and monotonically increasing in this test, the longitudinal stress \(\sigma_z\) coincides with the (current) yield stress \(\sigma_Y\) given by the constitutive law (1) which in this case \((C = 0, m = 0)\) reduces to:

\[
\sigma_z = \sigma_Y = A + Q_1 \left( 1 - e^{-C_1p} \right) + Q_2 \left( 1 - e^{-C_2p} \right) \tag{9}
\]

With the assumed material parameters:

\[
A = 3.7 \times 10^8 \\
Q_1 = 2.364 \times 10^8 \\
C_1 = 39.3 \\
Q_2 = 4.081 \times 10^8 \\
C_2 = 4.5 \tag{10}
\]

and with the value of \(p\) given by (8), we obtain from (9):

\[
\sigma_z \approx 10.116 \times 10^8 \tag{11}
\]

and all other stress components are zero.

The longitudinal force at the final elongation is:

\[
F_z = \sigma_z A_z \tag{12}
\]
where $A_z$ is the cross-section of the deformed specimen. Since the volume has been assumed to be approximately constant, and the final length is three times the initial one, the final cross-section must be:

$$A_z \approx \frac{1}{3} A_{x0} = 0.3333$$  \hspace{1cm} (13)

so that from (12):

$$F_z \approx 10.116 \times 10^8 \cdot 0.3333 = 3.3720 \times 10^8$$  \hspace{1cm} (14)

### 3.1.2 Calculations with continuum elements in 3D

The uniaxial calculations performed by EPX with constant velocity and no material sensitivity to strain rate and temperature are summarized in Table 1, for 3D continuum cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>Mesh</th>
<th>$\epsilon_x$</th>
<th>$\epsilon_y$</th>
<th>$\epsilon_z$</th>
<th>$\sigma_x$</th>
<th>$F_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUBE02</td>
<td>CUBE</td>
<td>-0.54831</td>
<td>-0.54831</td>
<td>1.0986</td>
<td>10.116 x 10^8</td>
<td>3.3720 x 10^8</td>
</tr>
<tr>
<td>CUB82</td>
<td>CUB8</td>
<td>-0.54830</td>
<td>-0.54830</td>
<td>1.0986</td>
<td>10.081 x 10^8</td>
<td>3.3658 x 10^8</td>
</tr>
<tr>
<td>C81L02</td>
<td>C81L</td>
<td>-0.54850</td>
<td>-0.54850</td>
<td>1.0986</td>
<td>10.065 x 10^8</td>
<td>3.3606 x 10^8</td>
</tr>
<tr>
<td>C82L02</td>
<td>C82L</td>
<td>-0.54850</td>
<td>-0.54850</td>
<td>1.0986</td>
<td>10.065 x 10^8</td>
<td>3.3606 x 10^8</td>
</tr>
<tr>
<td>C272C2</td>
<td>C272</td>
<td>-0.54848</td>
<td>-0.54848</td>
<td>1.0986</td>
<td>10.150 x 10^8</td>
<td>3.3891 x 10^8</td>
</tr>
<tr>
<td>C27SG2</td>
<td>C273</td>
<td>-0.54848</td>
<td>-0.54848</td>
<td>1.0986</td>
<td>10.150 x 10^8</td>
<td>3.3891 x 10^8</td>
</tr>
<tr>
<td>PRIS02</td>
<td>PRIS</td>
<td>-0.54848</td>
<td>-0.54848</td>
<td>1.0986</td>
<td>10.116 x 10^8</td>
<td>3.3777 x 10^8</td>
</tr>
<tr>
<td>PR6002</td>
<td>PR6</td>
<td>-0.54848</td>
<td>-0.54848</td>
<td>1.0986</td>
<td>10.116 x 10^8</td>
<td>3.3777 x 10^8</td>
</tr>
<tr>
<td>TETR02</td>
<td>TETR</td>
<td>-0.54848</td>
<td>-0.54848</td>
<td>1.0986</td>
<td>10.117 x 10^8</td>
<td>3.3778 x 10^8</td>
</tr>
</tbody>
</table>

Table 1: Calculations with 3D continuum elements.

**CUBE02**

The first case is CUBE02, which uses just one CUBE element. The input is:

```
cube02.epx
```

Note that the JAUM option is activated. This is used in CEA continuum elements to treat large strains.

Some results are shown in Figure 2: the total strains, the stresses, the nodal forces and the stress vs. strain curve. The agreement with the analytical solution at the final time is very good, as shown...
also in Table 1. Some small oscillations in the numerical solution are due to dynamic effects in the lateral directions, which are left free.

Figure 2: Some results of test case CUBE02.

**CUB802**
This test is similar to CUBE02, but uses the fully-integrated CUB8 element. Results are similar to those for the underintegrated CUBE element and in very good agreement with the reference.

**C81L02**
This test is similar to CUBE02, but uses JRC’s underintegrated hexahedron element C81L. Results are similar to those for the underintegrated CUBE element and in very good agreement with the reference.

**C82L02**
This test is similar to CUBE02, but uses JRC’s fully integrated hexahedron element C82L. Results are similar to those for the CUBE element and in very good agreement with the reference.

**C272G2**
This test is similar to CUBE02, but uses JRC’s 27-node under-integrated parabolic hexahedron element C272. Like for the parabolic 2D continuum elements Q92 and Q93 mentioned below (see tests Q92G01 and Q93G01 in the next Section), this calculation suffers from mechanism-like motions which, however, can be suppressed by suitable `RELA EQUAL` additional constraints, so as to achieve the correct results. The `LINK` directive for this test case is:

```
LINK COUP 7
```
C273G2
This test is similar to CUBE02, but uses JRC’s 27-node fully-integrated parabolic hexahedron element C273. Like for the parabolic 2D continuum elements Q92 and Q93 mentioned below (see tests Q92G01 and Q93G01 in the next Section), this calculation suffers from mechanism-like motions which, however, can be suppressed by suitable RELA EGAL additional constraints, so as to achieve the correct results.

PRIS02
This test is similar to CUBE02, but uses two 6-node PRIS elements obtained by cutting the cube by a vertical diagonal plane. Results are similar to those for the CUBE element and in very good agreement with the reference.

PR6002
This test is similar to PRIS02, but uses two 6-node PR6 elements, the fully integrated version of the prism element. Results are similar to those for the CUBE element.

TETR02
This test is similar to PRIS02, but uses twelve 4-node TETR elements, obtained by splitting the cube into 6 pyramids (one for each face of the cube) by adding a central node, and then by splitting each pyramid into two tetrahedra. Results are similar to those for the CUBE element and in very good agreement with the reference.

3.1.3 Calculations with continuum elements in 2D
Similar calculations are performed also in 2D continuum, plane stress conditions. In that case, traction is exerted along the y direction instead of the z direction. Results are summarized in Table 2. Recall that for continuum elements in EPX all quantities (both elemental ones like stresses and nodal ones like forces) are expressed along the global axes.

<table>
<thead>
<tr>
<th>Case</th>
<th>Mesh</th>
<th>$\varepsilon_x$</th>
<th>$\varepsilon_y$</th>
<th>$\varepsilon_z$</th>
<th>$\sigma_y$</th>
<th>$F_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical</td>
<td>—</td>
<td>$-0.54931$</td>
<td>$1.0986$</td>
<td>$-0.54931$</td>
<td>$10.116 \times 10^6$</td>
<td>$3.3720 \times 10^8$</td>
</tr>
<tr>
<td>CAR101</td>
<td>CAR1</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54893$</td>
<td>$10.116 \times 10^6$</td>
<td>$3.3777 \times 10^8$</td>
</tr>
<tr>
<td>CAR401</td>
<td>CAR4</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54893$</td>
<td>$10.116 \times 10^6$</td>
<td>$3.3777 \times 10^8$</td>
</tr>
<tr>
<td>Q41L01</td>
<td>Q41L</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54893$</td>
<td>$10.116 \times 10^6$</td>
<td>$3.3776 \times 10^8$</td>
</tr>
<tr>
<td>Q42L01</td>
<td>Q42L</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54893$</td>
<td>$10.116 \times 10^6$</td>
<td>$3.3776 \times 10^8$</td>
</tr>
<tr>
<td>TRIA01</td>
<td>TRIA</td>
<td>$-0.54893$</td>
<td>$1.0986$</td>
<td>$-0.54892$</td>
<td>$10.116 \times 10^6$</td>
<td>$3.3776 \times 10^8$</td>
</tr>
<tr>
<td>Q92G01</td>
<td>Q92</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54893$</td>
<td>$10.116 \times 10^6$</td>
<td>$3.3776 \times 10^8$</td>
</tr>
<tr>
<td>Q93G01</td>
<td>Q93</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54893$</td>
<td>$10.116 \times 10^6$</td>
<td>$3.3776 \times 10^8$</td>
</tr>
</tbody>
</table>

Table 2: Calculations with 2D continuum elements.

CAR101
This test is the 2D plane stress equivalent of CUBE02, and used CEA’s underintegrated quadrilateral element CAR1. Results are similar to those for the CUBE element and in very good agreement with the reference.

CAR401
This test is similar to CAR101, but uses the fully integrated quadrilateral element CAR4. Results are similar to those for the CAR1 element and in very good agreement with the reference.
Q41L01
This test is similar to CAR101, but uses JRC’s underintegrated quadrilateral element Q41L. Results are similar to those for the CAR1 element and in very good agreement with the reference.

Q42L01
This test is similar to Q41L01, but uses the fully integrated quadrilateral element Q42L. Results are similar to those for the CAR1 element and in very good agreement with the reference.

TRIA01
This test is similar to CAR101, but uses CEA’s triangular element TRIA. Results are similar to those for the CAR1 element and in very good agreement with the reference.

Q92G01
This test is similar to CAR101, but uses JRC’s 9-node parabolic under-integrated quadrilateral element Q92. When run with the same boundary conditions as CAR101 (i.e. imposed vertical velocities in nodes 1, 2, 3, 5, 6 and 7), this calculation shows a sort of mechanism-like (hourglass) pattern as soon as the longitudinal strain exceeds about 30%. However, this is unlikely to be a real hourglassing problem (although this element is under-integrated) because also the Q93 element (fully integrated) exhibits a similar behaviour (see below).

Some additional tests revealed that the problem shows up also with other materials (e.g. VM23) and therefore it is a problem of the element, and not of the VPJC material. The test was even run with PLEXIS-3C (one of EPX ancestors, 1999 version) and showed the same behaviour.

In order to eliminate the mechanism-like motions, additional constraints have been imposed (the “G” in the test name stays for “guided”) by means of the \texttt{RELA EGAL} directive: one relation imposes that the $x$-displacement of nodes 1, 7 and 8 be equal, and another equality relation is imposed for the $x$-displacement of nodes 3, 4 and 5.

With these additional constraints the result is in very good agreement with the analytical solution.

Q93G01
This test is similar to Q92G01, but uses JRC’s 9-node parabolic fully-integrated quadrilateral element Q93. The same additional constraints already discussed for the previous case are necessary also for this element (Q93) despite the fact that the element is fully integrated.

With the additional constraints the result is in very good agreement with the analytical solution.

3.1.4 Calculations with 3D shells

We consider now solutions with 3D shells, see Table 3. Since the element is set to lie in the $xy$-plane (so that $z$ is the “thickness” direction), the traction is directed along the (global) $y$ axis, which coincides with the (local) $y$ axis of the shell. Recall that for non-continuum elements (shells, plates, bars) in EPX the elemental quantities (like stresses) are expressed along the local (or convected) axes, while the nodal quantities (such as forces) are expressed along the global axes.

CQD4S3
This test is similar to CUBE02 (in 3D), but uses JRC’s quadrilateral degenerated shell element CQD4. Results are similar to those for the CUBE element and in very good agreement with the reference.

CQD3S3
This test is similar to CQD4S3, but uses JRC’s triangular degenerated shell element CQD3. Note that for this element not all results are directly comparable with those of the quadrilateral, due to the fact that the strains and stresses are expressend along a local reference frame attached to the element (and which in general rotates as the element deforms). However, those results which can be directly
noted in reference [9]. For this reason, the driving force $F_y$ suffers from the fact that the element thickness is not updated when large membrane strains occur, as this test is similar to CQD4S3, but uses CEA’s quadrilateral shell element Q4GS. This element still implementation of VPJC with the COQI element.

A. discussion on the transformation of results from the local to the global reference frame see Appendix A below. The conclusion is that the calculation shows an excessive “shrinkage” of the element as the deformation proceeds. The problem has been investigated and details are given in Appendix A. The conclusion is that the strains and stresses are expressed along a local reference frame attached to the element (and which in general rotates as the element deforms). However, those results which can be directly compared, i.e. $\epsilon_z$ and $F_y$, are in very good agreement with the reference. For a detailed discussion on the transformation of results from the local to the global reference frame see Appendix A.

CQD9G3
This test is similar to CQD4S3, but uses JRC’s 9-node parabolic degenerated quadrilateral shell CQD9. Like for the parabolic continuum elements Q92 and Q93 mentioned above, this calculation suffers from mechanism-like motions which, however, can be suppressed by suitable RELA EGAL additional constraints, so as to achieve the correct results.

CQD6S3
This test is similar to CQD9G3, but uses JRC’s triangular parabolic degenerated shell element CQD6. Note that for this element not all results are directly comparable with those of the quadrilateral, due to the fact that the strains and stresses are expressed along a local reference frame which, once in general rotates as the element deforms. However, those results which can be directly compared, i.e. $\epsilon_z$ and $F_y$, are in very good agreement with the reference. For a detailed discussion on the transformation of results from the local to the global reference frame see Appendix A.

COQIS3
This test is similar to CQD3S3, but uses JRC’s 3-node triangular plate/shell element COQI. This calculation shows an excessive “shrinkage” of the element as the deformation proceeds. The problem has been investigated and details are given in Appendix A below. The conclusion is that the element is limited to small strains, in particular small membrane strains, therefore it is not suited for this particular test problem. A different (small-strain) test case should be designed to check the implementation of VPJC with the COQI element.

Q4GSS3
This test is similar to CQD4S3, but uses CEA’s quadrilateral shell element Q4GS. This element still suffers from the fact that the element thickness is not updated when large membrane strains occur, as noted in reference [9]. For this reason, the driving force $F_y$ is largely overestimated in this example. All other quantities are in good agreement with the reference.

T3GSS3
This test is similar to Q4GSS3, but uses CEA’s triangular shell element T3GS. This element still suffers from the fact that the element thickness is not updated when large membrane strains occur, as noted in reference [9]. For this reason, the driving force $F_y$ is largely overestimated in this example. All other quantities are in good agreement with the reference.

Q4GRS3

<table>
<thead>
<tr>
<th>Case</th>
<th>Mesh</th>
<th>$\epsilon_x$</th>
<th>$\epsilon_y$</th>
<th>$\epsilon_z$</th>
<th>$\sigma_y$</th>
<th>$F_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical</td>
<td>—</td>
<td>$-0.54931$</td>
<td>$1.0986$</td>
<td>$-0.54931$</td>
<td>$10.116 \times 10^{9}$</td>
<td>$3.3720 \times 10^{6}$</td>
</tr>
<tr>
<td>CQD4S3</td>
<td>CQD4</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54893$</td>
<td>$10.116 \times 10^{9}$</td>
<td>$3.3776 \times 10^{6}$</td>
</tr>
<tr>
<td>CQD3S3</td>
<td>CQD3</td>
<td>$1.3541$</td>
<td>$-0.80421$</td>
<td>$-0.54819$</td>
<td>$3.6181 \times 10^{7}$</td>
<td>$3.3773 \times 10^{6}$</td>
</tr>
<tr>
<td>CQD9G3</td>
<td>CQD9</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54803$</td>
<td>$10.116 \times 10^{8}$</td>
<td>$3.3776 \times 10^{6}$</td>
</tr>
<tr>
<td>CQD6S3</td>
<td>CQD6</td>
<td>$1.3540$</td>
<td>$-0.80422$</td>
<td>$-0.54819$</td>
<td>$3.5638 \times 10^{7}$</td>
<td>$3.3767 \times 10^{6}$</td>
</tr>
<tr>
<td>COQIS3</td>
<td>COQI</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Q4GSS3</td>
<td>Q4GS</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54025$</td>
<td>$10.116 \times 10^{8}$</td>
<td>$5.8428 \times 10^{6}$</td>
</tr>
<tr>
<td>T3GSS3</td>
<td>T3GS</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54025$</td>
<td>$10.116 \times 10^{8}$</td>
<td>$5.8428 \times 10^{6}$</td>
</tr>
<tr>
<td>Q4GRS3</td>
<td>Q4GR</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54025$</td>
<td>$10.116 \times 10^{8}$</td>
<td>$5.8428 \times 10^{6}$</td>
</tr>
<tr>
<td>QPPSS3</td>
<td>QPPS</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54025$</td>
<td>$10.116 \times 10^{8}$</td>
<td>$5.8428 \times 10^{6}$</td>
</tr>
<tr>
<td>DKT3S3</td>
<td>DKT3</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54025$</td>
<td>$10.116 \times 10^{8}$</td>
<td>$5.8428 \times 10^{6}$</td>
</tr>
<tr>
<td>DST3S3</td>
<td>DST3</td>
<td>$-0.54894$</td>
<td>$1.0986$</td>
<td>$-0.54025$</td>
<td>$10.116 \times 10^{8}$</td>
<td>$5.8428 \times 10^{6}$</td>
</tr>
</tbody>
</table>

Table 3: Calculations with 3D shell elements.
This test is similar to CQD4S3, but uses CEA’s reduced-integrated quadrilateral shell element Q4GR. This element still suffers from the fact that the element thickness is not updated when large membrane strains occur, as noted in reference [9]. For this reason, the driving force $F_y$ is largely overestimated in this example. All other quantities are in good agreement with the reference.

**QPPSS3**

This test is similar to Q4GSS3, but uses CEA’s quadrilateral shell element QPPS. This element still suffers from the fact that the element thickness is not updated when large membrane strains occur, as noted in reference [9]. For this reason, the driving force $F_y$ is largely overestimated in this example. All other quantities are in good agreement with the reference.

**DKT3S3**

This test is similar to T3GSS3, but uses CEA’s discrete-Kirchhoff triangular shell element DKT3. This element still suffers from the fact that the element thickness is not updated when large membrane strains occur, as noted in reference [9]. For this reason, the driving force $F_y$ is largely overestimated in this example. All other quantities are in good agreement with the reference.

**DST3S3**

This test is similar to T3GSS3, but uses CEA’s triangular shell element DST3. This element still suffers from the fact that the element thickness is not updated when large membrane strains occur, as noted in reference [9]. For this reason, the driving force $F_y$ is largely overestimated in this example. All other quantities are in good agreement with the reference.

### 3.1.5 Calculations with beam/bar elements

Solutions with beam/bar elements are also obtained, according to Table 4. In this case we choose to align the element along the $y$ global axis. Therefore, as it concerns nodal forces and other nodal quantities, the component to be considered is along $y$ so that one should take $F_y$. However, for the rule stated above, the “longitudinal” stress to be considered is now $\sigma_x$.

<table>
<thead>
<tr>
<th>Case</th>
<th>Mesh</th>
<th>$\epsilon_x$</th>
<th>$\epsilon_y$</th>
<th>$\epsilon_z$</th>
<th>$\sigma_x$</th>
<th>$F_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical</td>
<td>—</td>
<td>1.0986</td>
<td>-0.54931</td>
<td>-0.54931</td>
<td>10.116 \times 10^7</td>
<td>3.3720 \times 10^8</td>
</tr>
<tr>
<td>ED0101</td>
<td>ED01</td>
<td>1.0986</td>
<td>-0.54849</td>
<td>-0.54849</td>
<td>10.116 \times 10^7</td>
<td>3.3776 \times 10^8</td>
</tr>
<tr>
<td>FUVP01</td>
<td>FUN3</td>
<td>1.0986</td>
<td>-0.54849</td>
<td>-0.54849</td>
<td>10.116 \times 10^6</td>
<td>3.3776 \times 10^8</td>
</tr>
<tr>
<td>FUVP02</td>
<td>FUN2</td>
<td>1.0986</td>
<td>-0.54849</td>
<td>-0.54849</td>
<td>10.116 \times 10^6</td>
<td>3.3776 \times 10^8</td>
</tr>
</tbody>
</table>

Table 4: Calculations with beam/bar elements.

**ED0101**

This test is in 2D plane stress conditions (CPLA) and uses JRC’s beam element ED01. Results are in very good agreement with the reference.

**FUVP01**

This test is in 3D and uses JRC’s bar element FUN3. Results are in very good agreement with the reference.

**FUVP02**

This test is in 2D and uses JRC’s bar element FUN2. Results are in very good agreement with the reference.
3.2 Uniaxial tests at constant strain rate

Next, a similar series of tests is performed by including the effect of strain rate on the material. For simplicity, the tests are conducted at constant strain rate, a situation in which it is still easy to find an analytical solution to the problem, against which the numerical results can then be compared. To find the reference solution we proceed as follows.

3.2.1 Stress

We assume a static (or quasi-static) uniaxial test along $z$ so that the lateral stresses in the “specimen” (initially a regular unit cube, see Figure 1) along $x$ and $y$ can be neglected. Since no shear stresses are present, the only non-zero stress is the longitudinal stress $\sigma_z$.

3.2.2 Strain

As concerns the strains, again the shear components are all null. The longitudinal strain $e_z$ is the one imposed by the driven longitudinal displacement (or velocity) while the lateral strains, in the two normal directions, are equal ($e_x = e_y$) and can be computed as follows.

Since the test involves large strains we use the (natural, or logarithmic) strain, denoted by the symbol $e$, as opposed to the engineering strain, which is denoted $\epsilon$. The Cauchy (or true) stress is indicated by $\sigma$. We have, for the longitudinal strain:

$$e_z = \ln \frac{L_z}{L_{z0}}$$

where $L_z$ is the current length and $L_{z0}$ is the initial length of the specimen along $z$. The strain can be separated into an elastic and a plastic part:

$$e_z = e_z^{el} + e_z^{pl}$$

If $\sigma_z$ denotes the longitudinal stress, and $\sigma_x$, $\sigma_y$ are the lateral stresses, which are null by assumption, then the longitudinal elastic strain is given by:

$$e_z^{el} = \sigma_z/E$$

where $E$ is Young’s modulus of the material, while the lateral elastic strains are:

$$e_x^{el} = e_y^{el} = -\frac{\nu}{E}\sigma_z = -\nu e_z^{el}$$

where $\nu$ is Poisson’s coefficient of the material. The lateral plastic strains can be obtained from the assumption, which is usually adopted in plasticity, that the plastic deformations are incompressible, i.e. occur without change of volume:

$$e_x^{pl} + e_y^{pl} + e_z^{pl} = 0$$

Since by symmetry it must be $e_x^{pl} = e_y^{pl}$ we obtain from (19):

$$e_x^{pl} = e_y^{pl} = -\frac{1}{2}e_z^{pl}$$

Finally, from (18) and (20) the total lateral strains result in:

$$e_x = e_y = -\frac{\nu}{E}\sigma_z - \frac{1}{2}\left(e_z - \frac{\sigma_z}{E}\right)$$
3.2.3 Strain rate

If the test is conducted at constant (natural) equivalent plastic strain rate, then the longitudinal plastic strain varies in time \( t \) according to the expression:

\[
\varepsilon_{pl}^z(t) = kt
\]

where

\[
k = \frac{d\varepsilon_{pl}^z}{dt} = \frac{\varepsilon_{pl}^z}{t}
\]

is the (longitudinal) plastic strain rate, usually expressed in s\(^{-1}\).

3.2.4 Equivalent stress

The equivalent stress \( \sigma_{eq} \) according to von Mises (which is used e.g. in the temperature softening term as well as in the yield function) is defined as:

\[
\sigma_{eq} = \sqrt{3J_2} = \sqrt{\frac{3}{2} s_{ij} s_{ij}}
\]

where

\[
J_2 = \frac{1}{2} s_{ij} s_{ij}
\]

is the second invariant of the deviatoric stress tensor \( s_{ij} \)

\[
s_{ij} = \sigma_{ij} - \frac{1}{3} \sigma_{kk} \delta_{ij}
\]

with \( \delta_{ij} \) Kronecker’s delta and with Einstein’s summation convention on repeated indices. By expanding (24) in terms of the stress components one obtains:

\[
\sigma_{eq} = \sqrt{\frac{1}{2} \left[ (\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 \right] + 3\tau_{xy}^2 + 3\tau_{yz}^2 + 3\tau_{zx}^2}
\]

In the uniaxial stress test considered here the only non-zero stress component is the longitudinal stress \( \sigma_z \) so that (27) reduces to:

\[
\sigma_{eq} = \sigma_z
\]

as it is obvious.

3.2.5 Equivalent plastic strain

The equivalent (or cumulated) plastic strain, indicated by \( \varepsilon_{pl}^{\text{eq}} \) or \( p \), is the energy-conjugated variable to the equivalent stress and is defined (incrementally) as:

\[
d\varepsilon_{pl}^{\text{eq}} = dp = \sqrt{\frac{2}{3} d\varepsilon_{ij}^{\text{pl}} d\varepsilon_{ij}^{\text{pl}}}
\]

From (20) we have, in a uniaxial stress test:

\[
d\varepsilon_{x}^{\text{pl}} = d\varepsilon_{y}^{\text{pl}} = -\frac{1}{2} d\varepsilon_{z}^{\text{pl}}
\]

\[
d\varepsilon_{xy}^{\text{pl}} = d\varepsilon_{yz}^{\text{pl}} = d\varepsilon_{xz}^{\text{pl}} = 0
\]

so that from (29) we get:
\[ dp = \sqrt{\frac{2}{3} \left[ \left( \varepsilon_{xz}^{pl} \right)^2 + \left( \varepsilon_{yz}^{pl} \right)^2 + \left( \varepsilon_{zz}^{pl} \right)^2 \right]} \]
\[ = \sqrt{\frac{2}{3} \left[ \frac{2}{4} \left( \varepsilon_{xz}^{pl} \right)^2 + \frac{2}{4} \left( \varepsilon_{yz}^{pl} \right)^2 + \frac{2}{4} \left( \varepsilon_{zz}^{pl} \right)^2 \right]} \]
\[ = \sqrt{\frac{2}{3} 4 \left( \varepsilon_{zz}^{pl} \right)^2} \]
\[ = \varepsilon_{zz}^{pl} \]  

By integrating this in time we obtain:
\[ p = \varepsilon_{zz}^{pl} \]  

i.e., the equivalent plastic strain coincides with the longitudinal plastic strain, as it seems obvious in a uniaxial stress test.

### 3.2.6 Relative displacement

The numerical test will be displacement-driven. Therefore, we need to determine the expression in time of the (relative) displacement \( d(t) \) of the two extremities of the specimen which should be used to generate the desired plastic strain rate, i.e. a constant value \( k \) in the present example:

\[ p = kt \]  

or, equivalently

\[ \dot{p} = k \]

If one would neglect the elastic part of the longitudinal strain \( \varepsilon_{zz}^{el} \) compared with the plastic part \( \varepsilon_{zz}^{pl} \), which for large strains is much larger \( (\varepsilon_{zz}^{pl} \gg \varepsilon_{zz}^{el}) \), one would obtain simply

\[ \varepsilon_{zz} = \ln \frac{L}{L_0} = \varepsilon_{zz}^{el} + \varepsilon_{zz}^{pl} \approx \varepsilon_{zz}^{pl} = p = kt \]

and from this

\[ L = L_0 \exp(kt) \]

so that the displacement would result in

\[ d(t) = L(t) - L_0 = L_0 [\exp(kt) - 1] \]

By including also the elastic contribution, we obtain instead:

\[ \varepsilon_{zz} = \ln \frac{L}{L_0} = \varepsilon_{zz}^{el} + \varepsilon_{zz}^{pl} = \frac{\sigma_{zz}}{E} + kt \]

where \( \sigma_{zz} \) is the longitudinal stress, \( E \) is Young’s modulus and \( \sigma_{zz}/E \) is the elastic longitudinal strain in the assumed uniaxial stress test. Then, instead of (36)

\[ L = L_0 \exp\left(\frac{\sigma_{zz}}{E} + kt\right) \]

and in place of (37)

\[ d(t) = L(t) - L_0 = L_0 \left[ \exp \left( \frac{\sigma_{zz}}{E} + kt \right) - 1 \right] \]
One sees that the traction test should be subdivided in two phases. The first phase is purely
elastic while the second phase is elasto-plastic. During the elastic phase the length of the specimen
passes from the initial value \(L_0\) (unloaded specimen) to the value \(L_{el}\) at which the longitudinal strain
(which is still purely elastic) reaches the elastic limit
\[e_{el} = \frac{A}{E}\] (41)
and the longitudinal stress is
\[\sigma_z = A\] (42)
so that
\[L_{el} = L_0 \exp\left(\frac{A}{E}\right)\] (43)

During this phase, the displacement is governed by an expression similar to (40) but without the
term \(kt\), i.e.
\[d_{el}(t) = L_{el}(t) - L_0 = L_0 \left\{ \exp \left( \frac{\sigma_z}{E} \right) - 1 \right\}\] (44)

This is because the model (elasto-viscoplastic) includes viscosity only in the plastic part of the
deformation, as it can be seen from (1) where the plastic strain rate \(\dot{\rho}\) (not the total strain rate)
appears in the strain-rate hardening term \((1 + \dot{\rho}^*)^C\).

From this instant on, as the longitudinal strain is increased some plasticity starts to occur. Therefore
one sees that, in order to use the expression (40), the time origin must be set so that \(t = 0\) when
the elastic limit is reached. Then, for convenience, the numerical test will be started at a suitable
negative time.

With this convention, the two expressions (40) for the elastic phase and (44) for the elastoplastic
phase may be combined into a single expression
\[d(t) = L(t) - L_0 = L_0 \left\{ \exp \left[ \frac{\sigma_z}{E} + \max(kt, 0) \right] - 1 \right\}\] (45)
which is valid through the entire uniaxial traction test.

In order to use (45) to pilot the displacement in the numerical test, the quantity \(\sigma_z\) must be
explicit in terms of other (known) quantities. To this end, one may note that during the elastoplastic
phase of the uniaxial traction test considered here (which is done at constant equivalent plastic
strain rate and without temperature softening effects), the longitudinal stress \(\sigma_z\) is constantly equal
to the current yield stress
\[\sigma_z = \sigma_Y\] (46)

By using (1), in the present case \((m = 0)\) this reduces to
\[\sigma_z = \sigma_Y = \left[ A + Q_1 \left( 1 - e^{-C_1\rho} \right) + Q_2 \left( 1 - e^{-C_2\rho} \right) \right] \left( 1 + \dot{\rho}^* \right)^C\] (47)
or, by using (33) and (34) and by recalling that \(\dot{\rho}^* = \dot{\rho}/p_0\)
\[\sigma_z(t) = \sigma_Y(t) = \left[ A + Q_1 \left( 1 - e^{-C_1kt} \right) + Q_2 \left( 1 - e^{-C_2kt} \right) \right] \left( 1 + \frac{k}{p_0} \right)^C\] (48)
3.2.7 Choice of the test parameters

We want to choose the test parameters so that a longitudinal natural plastic strain of 100% is reached in 2 s, at a constant rate. From (23) and (32) the equivalent plastic strain rate $k$ results in:

$$k = \frac{e_{pl}}{t} = \frac{p}{t} = \frac{1}{2} \tag{49}$$

From this we obtain for the equivalent plastic strain:

$$p = \frac{t}{2} \tag{50}$$

and for the equivalent plastic strain rate:

$$\dot{p} = \frac{1}{2} = \text{const.} \tag{51}$$

By combining (32), (16) and (17) we obtain:

$$e_z = e_{el} + e_{pl} = \sigma_z \frac{E}{t} + \frac{t}{2} \tag{52}$$

For simplicity, we do not want to consider the temperature softening effect in this test. To this end, we set the exponent $m$ to the special value $m = 0$. In this case by convention the code assumes $T^m = 0$ and therefore the temperature term becomes $(1 - T^m) = (1 - 0) = 1$, thus in practice disabling temperature softening. Note also that in this case the code does not update the temperature, so that it remains at the initial (room) value $T_r$.

With these settings, the expression (48) of the longitudinal stress (valid for $t > 0$) becomes

$$\sigma_z(t) = \left\{ A + Q_1 \left[ 1 - \exp \left( \frac{-C_1}{2} t \right) \right] + Q_2 \left[ 1 - \exp \left( \frac{-C_2}{2} t \right) \right] \right\} \left( 1 + \frac{1}{2\dot{p}_0} \right)^C \tag{53}$$

One last parameter to be chosen is the duration of the elastic phase $t_{el}$. In fact, the initial time of the numerical test will be set to $t_0 = -t_{el}$ and the final time will be set to $t_1 = 2$ s. In this way, as already mentioned, the elastic limit will be reached at $t = 0$. The choice of $t_{el}$, therefore, determines the rate at which loading occurs during the elastic phase.

In principle the behaviour of the material during the elastic phase is independent of the rate of loading. However, this is valid only in an ideal static case. If loading would be too rapid, then non-negligible lateral stresses would build up due to lateral inertia in the specimen, producing oscillations and a multi-axial stress state instead of the purely uniaxial stress state assumed in the test specifications.

In order to try avoid such phenomena, the loading rate during the first (elastic) phase is chosen such that the (elastic) longitudinal strain rate $\dot{e}_{el}^{el}$ is constant and equals the chosen equivalent plastic strain rate $k$ during the second (elasto-plastic) phase

$$\dot{e}_{el}^{el} = k \tag{54}$$

Since the longitudinal strain at the elastic limit $e_{el}^{el}$ is given by (41), we obtain for $t_{el}$:

$$t_{el} = \frac{e_{el}^{el}}{\dot{e}_{el}^{el}} = \frac{e_{el}^{el}}{k} = \frac{A}{Ek} \tag{55}$$

i.e., in this case

$$t_{el} = \frac{2A}{E} \tag{56}$$

The displacement during the elastic phase is obtained from (54) as follows:
\[ d_{el}(t) = L_{el} - L_0 \]
\[ = L_0 \exp(e_z^{el}) - L_0 \]
\[ = L_0 \left[ \exp(e_z^{el}) - 1 \right] \]
\[ = L_0 \left\{ \exp \left[ e_z^{el}(t - t_0) \right] - 1 \right\} \]
\[ = L_0 \left\{ \exp \left[ k(t - t_0) \right] - 1 \right\} \] (57)

### 3.2.8 Analytical solution

The analytical value of the longitudinal stress \( \sigma_z \) at the final time \( t = 2 \) s, at which a longitudinal plastic strain of 100 \% is reached, can be obtained from (48) by using the values of the material constants given in (10), the value of \( k \) given by (49) and the following values of \( \dot{p}_0 \) and \( C \):

\[
\dot{p}_0 = 5.0 \times 10^{-4}
\]
\[ C = 0.01 \] (58)

resulting in:

\[ \sigma_z(2.0) = \sigma_Y(2.0) = 1.08221 \times 10^9 \] (59)

By assuming for the elastic constants of the material the following values:

\[ E = 2.1 \times 10^{11} \]
\[ \nu = 0.33 \] (60)

the final longitudinal elastic strain can be computed from (17)

\[ e_z^{el}(2.0) = \sigma_z(2.0)/E = 1.08221 \times 10^9 / 2.1 \times 10^{11} = 5.15338 \times 10^{-3} \] (61)

while the final longitudinal plastic strain is given by (22) and results in:

\[ e_z^{pl}(2.0) = 2.0k = 1.0 \] (62)

so that the final longitudinal total strain is:

\[ e_z(2.0) = e_z^{el}(2.0) + e_z^{pl}(2.0) = 5.15338 \times 10^{-3} + 1.0 = 1.00515 \] (63)

The final lateral strains can be computed from (21)

\[ e_x(2.0) = e_y(2.0) = -\frac{\nu}{E} \sigma_z - \frac{1}{2} \left( e_z - \frac{\sigma_z}{E} \right) = -0.501701 \] (64)

The final cross section of the specimen is

\[ S(2.0) = L_y L_z = 1.0 \exp(e_x) \cdot 1.0 \exp(e_y) = 0.605500 \cdot 0.605500 = 0.366630 \] (65)

and the final longitudinal force is

\[ F_z(2.0) = S \sigma_z = 0.366630 \cdot 1.08221 \times 10^9 = 3.96771 \times 10^8 \] (66)
3.2.9 Calculations with 3D continuum elements

The calculations performed by EPX with constant strain rate are summarized in Table 5.

**CUBE05**

The test CUBE05 uses a unit hexahedron (CUBE element) with the JAUM option. This option is necessary with some CEA’s continuum elements (e.g. CUBE) in order to activate large strain computation with Jaumann stress rate (the option is not needed with JRC’s continuum elements).

The input file is listed and commented below:

```plaintext
CUBE05
CUB9
CUBE1
LAGR TRID
DIME TTHI 50101 MNTI 701 TERM

Some dimensioning is needed due to the large number of time points in tables (TTHI) and of prescribed output time steps (MNTI), see [1].

GEOM LIBR POIN 8 CUBE 1 TERM
-0.5 -0.5 -0.5 0.5 -0.5 -0.5 0.5 -0.5 0.5 -0.5 0.5 -0.5 0.5 0.5 0.5

GEOM LIBR POIN 8 CUBE 1 TERM
-0.5 -0.5 -0.5 0.5 -0.5 -0.5 0.5 -0.5 0.5 -0.5 0.5 -0.5 0.5 0.5 0.5
1 2 3 4 5 6 7 8

The material is VPJC with the material parameters given by (10) and (58).

MATE VPJC RO 7850.0 ! Docol 600 steel parameters
YOUN 2.1E11
NU 0.33
ELAS 3.70E8
QR1 2.364E8
CR1 39.3
QR2 4.081E8
CR2 4.5
PDOT 5.E-4
C 1.0E-2 ! This is 1.E-3 in Table 1 of material parameters (?)
TQ 0.9 ! We assume adiabatic conditions
CP 452.0
TM 1800.0
M 0.0 ! M = 0 : no temperature-induced softening
DC 1.0
WC 15.95E8 ! Large WC : no failure (realistic value is 4.73E8)
LECT 1 TERM

Note that, as concerns the material parameter \( C \) (strain-rate hardening parameter \( C \)) used in the input data set, values \( C = 0.01 \), \( C = 0.005 \) and \( C = 0.001 \) are found in the literature. It is reasonable to use \( C = 0.01 \) in this report since the scope here is to show that the routine gives acceptable results, rather than to obtain a perfect match to some experimental data.

The relative displacements history \( d(t) \) is read from an external file (see below) for convenience, since it contains 50,101 time points:

```plaintext
INCLUDE 'fonc_disp.txt'
```

The displacement is imposed along the vertical direction \( z \), half of it to the lower face and half of it to the upper face of the cube:

```plaintext
LINK COUP
DEPL 3 -0.5 FONC 1 LECT 1 2 3 4 TERM
DEPL 3 0.5 FONC 1 LECT 5 6 7 8 TERM
```

The results are printed at every step initially (up to step 201) and then every 100 steps:

```plaintext
ECRI DEPL VITE ACCE FINT FEXT FLIA FDEC CONT ECRO EPST
NUPA LECT 0 PAS 1 201 PAS 100 50101 TERM
FICH ALIC
NUPA LECT 0 PAS 1 201 PAS 100 50101 TERM
```

---

### Table 5: Calculations with 3D continuum elements.

<table>
<thead>
<tr>
<th>Case</th>
<th>Mesh</th>
<th>( \epsilon_x )</th>
<th>( \epsilon_y )</th>
<th>( \epsilon_z )</th>
<th>( \sigma_z )</th>
<th>( F_z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical</td>
<td>—</td>
<td>-0.50170</td>
<td>-0.50170</td>
<td>1.0052</td>
<td>10.822 \times 10^8</td>
<td>3.9677 \times 10^8</td>
</tr>
<tr>
<td>CUBE05</td>
<td>CUBE</td>
<td>-0.50169</td>
<td>-0.50169</td>
<td>1.0051</td>
<td>10.821 \times 10^8</td>
<td>3.9675 \times 10^8</td>
</tr>
</tbody>
</table>
The user-defined step is chosen (PAS UTIL) and the JAUM option is set for CUBE:

```fortran
OPTI NOTE PAS UTIL LOG 100
JAUM
```

The initial (negative) and final times are set, and the time step is entirely piloted by prescribing all the time instants at which the solution will be computed (HIST), in this case 50,100 time instants, which are conveniently specified by the PROG directive. Note that the initial time instant must not be included (it is commented out).

```fortran
REAL(8) :: e, a, q1, c1, q2, c2, pdot0, c, k, m
IMPLICIT NONE
PROGRAM displ
  
  t0 = -t0l  
t = t0  
d = 0.0  
write (6,1) t, d  
1 FORMAT (6I1, T, D)  
DO i = 1, nel  
d = 0.D0  
  ELSE  
  IF (i < nel) THEN  
t = t + dtel  
  ELSE  
  t = t + dt  
ENDIF  
d = 10*(exp(a*(t-t0)) ) - 1.0D0  
write (6,1) t, d  
ENDDO  
  x = 2.1011  
a = 3.7028  
q1 = 2.3648  
c1 = 39.300  
q2 = 4.0181  
c2 = 4.500  
pdot0 = 5.0  
c = 1.0  
k = 0.50  
m = 0.0  
t0 = -qt  
d0 = 1.0  
dt = 4.0  
t2 = 2.0  
```

The imposed displacement is computed by a small ad-hoc Fortran program `displ.f`, which implements the formulas presented above.

```fortran
```

The formulas presented above are implemented in the `displ.f` program. The user-defined step is chosen using the `PAS UTIL` option and the `JAUM` option is set for CUBE. The initial (negative) and final times are set, and the time step is entirely piloted by prescribing all the time instants at which the solution will be computed using the `HIST` option. Note that the initial time instant must not be included (it is commented out).
The output of the program is piped onto file `fonc_disp.txt`. At the beginning of this file we insert the FONC directive, declaring one “table” function with 50,101 points:

```
FONC 1 TABL 50101
```

Next come 50,101 lines (generated by the Fortran program) containing each one a couple of values $t$, $d(t)$. At the end of the file we insert the line:

```
RETURN
```

This instructs EPX to continue reading from the main EPX input file.

The stress along $z$ is computed by a small ad-hoc Fortran program `sigma.f`, which implements the formulas presented above.

```
PROGRAM sigma
IMPLICIT NONE
REAL(8) :: e, a, q1, c1, q2, c2, pdot0, c, k, m
REAL(8) :: l0, dt, t2, l, t, t0, tel, dtel, d
REAL(8) :: pdotstar, b, f1, f2, ez, sz
INTEGER :: nel, n, i, ntot
*
e = 2.1D11
a = 3.70D8
q1 = 2.364D8
c1 = 39.3D0
q2 = 4.081D8
c2 = 4.5D0
pdot0 = 5.D-4
c = 1.D-2
k = 0.5D0
m = 0.D0
l0 = 1.D0
dt = 4.D-5
t2 = 2.D0
nel = 100
*
tel = 2.0D0 / e
dtel = tel / nel
*
t0 = -tel
t = t0
sz = 0.D0
n = NINT (t2 / dt)
ntot = nel + n + 1
write (6,2) ntot
2 FORMAT ('VALUES',I6,' COMPONENTS 1',/,'* Time [s] STRESS [PA]',/,,'* T sz')
write (6,1) t, sz
1 FORMAT (1P,2E17.6)
do i = 1, nel
  if (i < nel) then
    t = t + dtel
  else
    t = 0.D0
  endif
  d = l0*(exp(k*(t-t0)) - 1.D0)
ez = LOG ((l0+d) / l0)
sz = e*ez
write (6,1) t, sz
end do
*
  pdotstar = k / pdot0
  b = (1.D0 + pdotstar)**c
  do i = 1, n
    t = dt*i
    f1 = q1*(1.D0 - exp(-c1*k*t))
    f2 = q2*(1.D0 - exp(-c2*k*t))
    sz = (a + f1 + f2)*b
    d = l0*(exp(sz/e + k*t) - 1.D0)
    write (6,1) t, sz
  end do
*
end program sigma
```

The output of the program is piped onto file `sigma.pun`, which is written in EPX’s .PUN format. This format is the one produced by EPX by means of the LIST command, and can be used to enter in EPX a set of external data (in tabular form) to be plotted for comparison with EPX’s own solution.

Some results are shown in Figure 3: the displacements, the cross-section area, the total strains, the stresses, the nodal forces and the stress vs. strain curve. The agreement with the analytical solution at the final time is very good, as shown also in Table 5. Some small oscillations in the numerical solution are due to dynamic effects in the lateral directions, which are left free.

In Figure 3 (d) the computed stress (in black) is compared with the analytical result (in red) from the `sigma.pun` file obtained as explained above, showing excellent overall agreement.

When looking at results in some more detail, one sees that at the threshold between elastic and elasto-plastic behaviour (which in this test occurs at time $t = 0$ by convention, as explained in Section 3.2.6), a sudden “jump” occurs in the stresses.

Figure 4 shows the stress along $z$ in a small time window around $t = 0$. The black curve is the numerical result, which exhibits some oscillations. The red curve is the analytical result, i.e. the plot of the contents of file `sigma.pun` described above. Note that not only the numerical, but also the analytical solution exhibits a sudden jump at $t = 0$. The numerical model is unable to represent such a jump exactly and this explains the oscillations observed in the numerical solution.

The (analytical) discontinuity in stress is due to the fact that the model is elastic-viscoplastic. In other words, as long as the material behaves elastically, no viscous effects are included in the material law. However, as soon as some plasticity occurs, some viscosity starts to develop.

In the material law (1) viscosity is represented by the strain-rate hardening multiplicative term $\eta = (1 + \dot{p}^*)^{C}$, which is related to the dimensionless plastic strain rate $\dot{p}^*$. With the values assumed above ($\dot{p} = 0.5$, $\dot{p_0} = 5.0 \times 10^{-4}$, $C = 0.01$), one finds $\dot{p}^* = 1000$ and then $\eta = 1.0715$. The value of $\eta$ passes suddenly from 1.0 in the elastic phase to 1.0715 in the elasto-plastic phase. This corresponds precisely to the sudden increase by 7.15% of the stress in the red curve shown in Figure 4.
4 Conclusions and future work

The VPJC material has been tested with a large variety of finite elements in the EPX code. Overall, the models behave as expected, but there remain a few malfunctionings:

- Some shell elements of CEA (Q4GS, T3GS, Q4GR, QPPS, DKT3 and DST3) do not give the exact answer expected because the element thickness is not updated as a result of large membrane strains. However, this is a problem of the element, as already noted in [9], and not of the VPJC material. The problem has been reported to CEA and might be corrected in the near future.
The JRC parabolic elements of the quadrilateral and cube families (C272, C273, CQD9, Q92 and Q93) suffer from mechanism-like motions if the standard boundary conditions of the large-strain test (only vertical velocities) are imposed. The mechanism can be suppressed by suitable additional constraints on the equality of some relevant displacements.

The JRC triangular shell elements CQD3 and CQD6 give the correct nodal forces and the correct stresses (once the latter are rotated in the global system for comparison with the analytical solution). The total strains, however, are wrong even after rotation. This is not an issue in practical applications since the total strains are only a post-processing result and do not intervene in the numerical solution.

The COQI element is not designed to undergo large strains, in particular large membrane strains, such as those encountered in the test cases considered above. Therefore, a specific (small strain) test should be designed to check this element in conjunction with the VPJC material.

The tests performed so far verify only pure (uniaxial) traction loading situations, without and with strain rate (viscous) effects. The temperature effect ($m \neq 0$) has not been investigated. Furthermore, element failure and/or melting due to temperature effect has not been tested yet. Single element testing in rate-independent simple shear, uniaxial tension and simple shear at elevated strain rates, temperature softening and element failure are treated in [3].

References


to Determine the Structural Response in Fast Transient Dynamics. NTNU/JRC Report (2015), in publication.


Appendix A — Coordinate transformations

This Section recalls the coordinate transformations needed to interpret the results from triangular elements. Although the elements considered here (COQI, CQD3, CQD6) are 3D, the numerical examples considered in the previous Sections are such that the elements lie in the XY plane of the global reference, and the loading is such that the deformation occurs in the element’s plane.

Therefore, for simplicity, the Z direction can be ignored and the transformations are carried out in 2D space.

Transformation of a vector

Let \( \mathbf{v} = [v_x, v_y]^T \) be a vector in 2D space, referred to the (global) coordinate system \( X, Y \). Consider a second (rotated) reference frame \( x, y \), obtained from \( X, Y \) upon rotation by an angle \( \beta \) in the anti-clockwise direction, as shown in Figure 5.

\[ \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \mathbf{R} \begin{bmatrix} v_X \\ v_Y \end{bmatrix} \]  

(67)

where \( \mathbf{R} \) is the transformation (rotation) matrix:

\[ \mathbf{R} = \begin{bmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{bmatrix} \]  

(68)

Since it represents a rigid rotation, the matrix \( \mathbf{R} \) is orthogonal:

\[ \mathbf{R}^{-1} = \mathbf{R}^T \]  

(69)

and therefore the inverse transformation, from the rotated to the global reference, is:

\[ \begin{bmatrix} v_X \\ v_Y \end{bmatrix} = \mathbf{R}^{-1} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \mathbf{R}^T \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \]  

(70)

Transformation of a tensor

A tensor such as the stress \( \sigma \) (or the strain \( \epsilon \)) transforms according to:

\[ \begin{bmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{bmatrix} = \mathbf{R} \begin{bmatrix} \sigma_X & \tau_{XY} \\ \tau_{XY} & \sigma_Y \end{bmatrix} \mathbf{R}^T \]  

(71)

where the notation \( \sigma_X \) has been used instead of \( \sigma_{XX} \) for simplicity and, as concerns the shear components, \( \tau_{XY} = \tau_{YX} \) for reciprocity.

By developing (71) one obtains:

\[ \begin{align*}
\sigma_x &= \cos^2 \beta \sigma_X + 2 \sin \beta \cos \beta \tau_{XY} + \sin^2 \beta \sigma_Y \\
\tau_{xy} &= -\sin \beta \cos \beta \sigma_X + (\cos^2 \beta - \sin^2 \beta) \tau_{XY} + \sin \beta \cos \beta \sigma_Y \\
\sigma_y &= \sin^2 \beta \sigma_X - 2 \sin \beta \cos \beta \tau_{XY} + \cos^2 \beta \sigma_Y
\end{align*} \]  

(72)
The inverse transformation of stress, from the rotated to the global reference, is:

\[
\begin{bmatrix}
\sigma_X & \tau_{XY} \\
\tau_{XY} & \sigma_Y
\end{bmatrix} = R^T \begin{bmatrix}
\sigma_x & \tau_{xy} \\
\tau_{xy} & \sigma_y
\end{bmatrix} R
\]  
(73)

which, upon development, gives:

\[
\begin{align*}
\sigma_X &= \cos^2 \beta \sigma_x - 2 \sin \beta \cos \beta \tau_{xy} + \sin^2 \beta \sigma_y \\
\tau_{XY} &= \sin \beta \cos \beta \sigma_x + (\cos^2 \beta - \sin^2 \beta) \tau_{xy} - \sin \beta \cos \beta \sigma_y \\
\sigma_Y &= \sin^2 \beta \sigma_x + 2 \sin \beta \cos \beta \tau_{xy} + \cos^2 \beta \sigma_y
\end{align*}
\]  
(74)

Note that, of course, (74) can also be obtained directly from (72) by replacing the angle \(\beta\) with \(-\beta\) and by noting that:

\[
\begin{align*}
\sin(-\beta) &= -\sin \beta \\
\cos(-\beta) &= \cos \beta \\
\sin^2(-\beta) &= \sin^2 \beta \\
\cos^2(-\beta) &= \cos^2 \beta \\
\sin(-\beta) \cos(-\beta) &= -\sin \beta \cos \beta
\end{align*}
\]  
(75)

The above equations (71–74) can also be used to transform the strain tensor \(\epsilon\). However, if one uses the engineering shear strain \(\gamma_{xy}\) instead of the tensor shear strain \(\epsilon_{xy}\) (as it is the case in EPX), one must take into account the fact that \(\gamma_{xy} = 2\epsilon_{xy}\), so that the transformation equations become, respectively:

\[
\begin{bmatrix}
\epsilon_x & \gamma_{xy}/2 \\
\gamma_{xy}/2 & \epsilon_y
\end{bmatrix} = R \begin{bmatrix}
\epsilon_X & \gamma_{XY}/2 \\
\gamma_{XY}/2 & \epsilon_Y
\end{bmatrix} R^T
\]  
(76)

\[
\begin{align*}
\epsilon_x &= \cos^2 \beta \epsilon_x + \sin \beta \cos \beta \gamma_{XY} + \sin^2 \beta \epsilon_y \\
\gamma_{xy} &= -2 \sin \beta \cos \beta \epsilon_x + (\cos^2 \beta - \sin^2 \beta) \gamma_{XY} + 2 \sin \beta \cos \beta \epsilon_y \\
\epsilon_y &= \sin^2 \beta \epsilon_x - \sin \beta \cos \beta \gamma_{XY} + \cos^2 \beta \epsilon_y
\end{align*}
\]  
(77)

\[
\begin{bmatrix}
\epsilon_X & \gamma_{XY}/2 \\
\gamma_{XY}/2 & \epsilon_Y
\end{bmatrix} = R^T \begin{bmatrix}
\epsilon_x & \gamma_{xy}/2 \\
\gamma_{xy}/2 & \epsilon_y
\end{bmatrix} R
\]  
(78)

\[
\begin{align*}
\epsilon_X &= \cos^2 \beta \epsilon_x - \sin \beta \cos \beta \gamma_{xy} + \sin^2 \beta \epsilon_y \\
\gamma_{XY} &= 2 \sin \beta \cos \beta \epsilon_x + (\cos^2 \beta - \sin^2 \beta) \gamma_{xy} - 2 \sin \beta \cos \beta \epsilon_y \\
\epsilon_Y &= \sin^2 \beta \epsilon_x + \sin \beta \cos \beta \gamma_{xy} + \cos^2 \beta \epsilon_y
\end{align*}
\]  
(79)

Local reference frames

The COQI triangular plate element [12, 13] uses a local reference frame as shown in Figure 6 (a). The element’s stresses and strains (at Gauss points) are expressed in this reference frame, which is the same for all Gauss points of each element, but varies of course from element to element. The reference frame axes \(x\) and \(y\) lie in the plane of the element. The \(x\) axis bisectates the angle at the first vertex of the element. The \(z\) axis is normal to the element plane and directed along the “positive” normal to the element according to right-hand rule. The \(y\) axis is normal to the first two according to the right-hand rule.

The CQD3 triangular shell element [10, 11] uses a local “lamina” reference frame which in general varies from Gauss point to Gauss point in the lamina, but is the same for all Gauss points along
the fiber (i.e., through the element thickness) at any given lamina position. Stresses and strains are expressed in the lamina system.

The construction of the lamina basis is detailed in Section 2.1.4 of reference [10]. However, upon inspection, it was found out that the algorithm actually implemented in the code is slightly different from the one in the report.

The implemented algorithm is as follows. Let \( L_1, L_2, L_3 \) be the local (or parent) coordinates of the triangular element, as shown in Figure 7 (a) for the linear element CQD3 and in Figure 7 (b) for the parabolic element CQD6. These are also called the area coordinates of the triangle. Only two of the area coordinates are independent, since \( L_1 + L_2 + L_3 = 1 \).

The local \( x \) axis is taken along the \( L_1 \) direction, i.e. along the line that joins node 3 to node 1 of the CQD3 triangle, in local numbering. The local \( z \) axis is normal to the triangle (according to the right-hand rule) and the local \( y \) axis is normal to the other two, see Figure 6 (b). This construction is different (in the element plane) from the one reported in reference [10], where the local \( x,y \) reference would be “averaged” around the local axes \( L_1,L_2 \) (so that \( x \) would not coincide with \( L_1 \) in general, unless the triangle has a right angle at vertex 3).

By following reference [10], we see that at each Gauss point a lamina (orthogonal) matrix \( q \) is built up:

\[
q = [q_{ij}] = [\hat{\ell}_1 \hat{\ell}_2 \hat{\ell}_3]^T : \text{global} \rightarrow \text{lamina} \quad (80)
\]

where the \( \hat{\ell}_I \) are the unit lamina frame vectors which coincide with the \( x,y,z \) directions built as explained above. A vector \( v \) is transformed from the global basis to the lamina basis according to:

\[
v_{\text{lamina}} = qv_{\text{global}} \quad (81)
\]

By comparing equations (67) and (81) one sees that the lamina matrix \( q \) coincides with the rotation matrix \( R \) that has been considered above:
\[ q = R \] \hspace{2cm} (82)

Therefore, in order to compute the stresses or strains in the global frame starting from those in the lamina frame, one can use two alternative (equivalent) methods:

- Compute the lamina basis, i.e., the \( q \) matrix, which coincides with \( R \) as noted, and then apply equation (73) or (78), or

- Compute the angle \( \beta \) and then apply equation (74) or (79) directly.

In order to use the first method above, one can either compute the \( q \) matrix by hand according to the definition (80), which is probably a bit laborious, or extract the values directly from the code. In the EPX code the \( q \) matrix values are stored in the array \texttt{VARELM} for the CQDx family of elements.

The second method is simpler in the present case (2D), since the angle \( \beta \) can be evaluated directly by looking at the mesh and by considering the local element numbering: it is the angle which brings the global \( X \) axis onto the local \( x \) axis by rotating in the anti-clockwise direction.

With the conventions explained above, the local reference frames for the COQI elements used in test case COQIS3 and those for the CQD3 elements used in case CQD3S3 are as shown in Figure 8 (a) and (b), respectively. The \( X \) and \( Y \) axes are the global axes, while \( x_1, y_1 \) are the local axes for element \( I \). Note that, as already mentioned, the COQI has a single local reference for all Gauss points, while for CQD3 if one uses the reduced integration rule, which is the default and the one used in case CQD3S3, there is just one Gauss point in the lamina, so that there is just one lamina reference per element. The thin arrows indicate the numbering of the elements: for example, element 1 is numbered 1, 2, 3.

![Figure 8: Local reference frames in test cases COQIS3 and CQD3S3.](image)

It is readily seen that the \( \beta \) angles for the elements 1 and 2 are \( \beta_1 = 22.5^\circ, \beta_2 = 202.5^\circ \) in the COQIS3 case (Figure 8 (a)), and \( \beta_1 = 225^\circ, \beta_2 = 45^\circ \) in the CQD3S3 case (Figure 8 (b)).

The lamina reference frames shown in Figure 8 are those in the initial configuration of the tests. However, in order to compute the global stresses or strains at the end of the test (for comparison with the analytical solution), one must consider the lamina references in the deformed (final) configuration, as detailed below.

**Analysis of COQI results**

For the COQI, according to reference [12], the coordinates of the three nodes of an element in the \( xyz \) reference frame are supposed to be constant. In other words, the element is formulated in such a way that large displacements and large rotations of the element as a whole (rigid body) are taken into account (by means of a co-rotational approach), but large strains, and in particular large membrane
strains, are not taken into account. This limitation has been now made much clearer in the EPX Users’ manual.

This point had been initially overlooked, since the element even takes into account the thickness variation due to membrane strains. However, the fact that strains must be small (of the order of a few percent at most) explains why the behaviour of the element is so weird in the present test, where engineering strains of 300 % occur in the membrane at the end of the test.

Therefore, it is useless to go on with the analysis of the COQI solution for this test (case COQIS3). Of course, this does not mean that COQI cannot be used with VPJC, but simply that the element must be used within the (membrane) strain range for which it is designed. A different (small-strain) test case should be set up for COQI to check the VPJC implementation for this element.

Analysis of CQD3 results

For the CQD3 the final shape of element 1 in test case CQD3S3 is shown in Figure 9 (b). At this moment the “height” \( L_y \) of the element is 3 units, as imposed in the test. The element “width” \( L_x \) can be computed from the analytical solution. According to Table 3 the “horizontal” strain should be \( \epsilon_x = -0.54931 \). Since this is a natural (logarithmic) strain, we get:

\[
L_x = L_{x0} \exp(\epsilon_x) = 1.0 \exp(-0.54931) = 0.577
\]

\[\text{(83)}\]

![Figure 9: Element deformation in test CQD3S3.](image)
This value of the width is exactly confirmed by reading out the results in the code listing. The angle $\alpha$ at the element’s top (see Figure 9 (b)) is:

$$\alpha = \arctan\left(\frac{L_x}{L_y}\right) = \arctan\left(\frac{0.577}{3.000}\right) = 10.9^\circ \quad (84)$$

so that the angle $\beta$ is:

$$\beta = 270.0 - \alpha = 259.1^\circ \quad (85)$$

and:

$$\sin \beta = -0.982$$
$$\cos \beta = -0.189$$
$$\sin^2 \beta = 0.964$$
$$\cos^2 \beta = 0.036$$
$$\sin \beta \cos \beta = 0.186$$
$$2 \sin \beta \cos \beta = 0.371 \quad (86)$$

By dumping out from the code the contents of the $q$ matrix at the final time step we find, for element 1:

$$q_1 = \begin{bmatrix} -0.189 & -0.982 & 0.000 \\ 0.982 & -0.189 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix} = R_1 = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (87)$$

which is in excellent agreement with the expression (68) of $R$. For element 2, incidentally, we have $\beta_2 = 90 - 10.9 = 79.1^\circ$, $\sin \beta_2 = 0.982$, $\cos \beta_2 = 0.189$ and, from the code:

$$q_2 = \begin{bmatrix} 0.189 & 0.982 & 0.000 \\ -0.982 & 0.189 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix} = R_2 = \begin{bmatrix} \cos \beta_2 & \sin \beta_2 & 0 \\ -\sin \beta_2 & \cos \beta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (88)$$

which is also correct. From the code listing at the final time we find the following results in terms of stress and strain components expressed in the local (lamina) basis for element 1 (see also Table 3):

$$\sigma_x = 9.7537 \times 10^8 \quad \sigma_y = 0.3618 \times 10^8 \quad \tau_{xy} = 1.8781 \times 10^8 \quad \sigma_z = 0.0 \quad (89)$$

From equation (74) we obtain the stresses expressed in the global system:

$$\sigma_X = \cos^2 \beta \sigma_x - 2 \sin \beta \cos \beta \tau_{xy} + \sin^2 \beta \sigma_y$$
$$= 0.036 \cdot 9.7537 \times 10^8 - 0.371 \cdot 1.8781 \times 10^8 + 0.964 \cdot 0.3618 \times 10^8$$
$$= 0.003 \times 10^8 \quad (90)$$

$$\tau_{XY} = \sin \beta \cos \beta \sigma_x + (\cos^2 \beta - \sin^2 \beta) \tau_{xy} - \sin \beta \cos \beta \sigma_y$$
$$= 0.186 \cdot 9.7537 \times 10^8 + (0.036 - 0.964) \cdot 1.8781 \times 10^8 - 0.186 \cdot 0.3618 \times 10^8$$
$$= 0.004 \times 10^8 \quad (91)$$

$$\sigma_Y = \sin^2 \beta \sigma_x + 2 \sin \beta \cos \beta \tau_{xy} + \cos^2 \beta \sigma_y$$
$$= 0.964 \cdot 9.7537 \times 10^8 + 0.371 \cdot 1.8781 \times 10^8 + 0.036 \cdot 0.3618 \times 10^8$$
$$= 10.112 \times 10^8 \quad (92)$$
These results are in very good agreement with the analytical solution: \( \sigma_X = 0 \), \( \tau_{XY} = 0 \) and \( \sigma_Y = 10.116 \times 10^8 \) (see also Table 3).

From equation (79) we obtain the strains expressed in the global system:

\[
\begin{align*}
\epsilon_X &= \cos^2 \beta \epsilon_x - \sin \beta \cos \beta \gamma_{xy} + \sin^2 \beta \epsilon_y \\
&= 0.036 \cdot 1.3540 - 0.186 \cdot 0.77012 + 0.964 \cdot (-0.80421) \\
&= -0.869 \\

g_{XY} &= 2 \sin \beta \cos \beta \epsilon_x + (\cos^2 \beta - \sin^2 \beta) \gamma_{xy} - 2 \sin \beta \cos \beta \epsilon_y \\
&= 0.371 \cdot 1.3540 + (0.036 - 0.964) \cdot 0.77012 - 0.371 \cdot (-0.80421) \\
&= 0.086 \\
\epsilon_Y &= \sin^2 \beta \epsilon_x + \sin \beta \cos \beta \gamma_{xy} + \cos^2 \beta \epsilon_y \\
&= 0.964 \cdot 1.3540 + 0.186 \cdot 0.77012 + 0.036 \cdot (-0.80421) \\
&= 1.420
\end{align*}
\]

The results for the strains are very different from the analytical solution: \( \epsilon_X = -0.54931 \), \( g_{XY} = 0 \) and \( \epsilon_Y = 1.0986 \) (see also Table 3). These results have been checked independently by using the expression (73) involving the rotation matrix \( R \) instead of equation (79) (see small Fortran programs \( \text{chkeps3.f} \) or \( \text{chkeps2.f} \) and the corresponding outputs \( \text{chkeps3.txt} \) or \( \text{chkeps2.txt} \) in Appendix B) and are confirmed.

This indicates that the total strains are wrongly computed by the code in example CQD3S3. Still, the calculation is correct in the sense that the stresses and the resultant forces are in excellent agreement with the reference solution (which of course is the most important thing).

While it would be nice to understand why the total strains are incorrectly estimated, it is recalled that the total strain tensor is just a post-processing quantity in the EPX code. In other words, total strains are computed and printed out for the user’s convenience, but do not intervene in any way in the actual calculations, due to the incremental nature of the constitutive law implementations (perhaps with the exception of some failure criteria which may sometimes be based upon total strain).

A final general observation may be raised on the presentation of results from the EPX code. The calculations presented in this Section are very laborious and error-prone. They are needed just to express the stress or strain in a different reference frame (the global frame in this case) from the standard one in which results are normally presented (the element- or Gauss-point based frame, for shell elements).

One might argue that such operations would be best done by the code itself rather than by cumbersome hand calculations. Indeed, the code contains all the information (transformation matrices etc.) needed to transform the results in any reference system that would be convenient for the user. So, one might think of adding an option or a similar feature that would instruct the code to produce the results (also) in a different reference frame from the standard one.

However, it is also true that for shells and similar elements, expressing the stresses or strains in the global reference frame would very rarely make sense. It is only useful in the present case, due to the specific set-up of the numerical test (for ease of comparison with an analytical solution).

To summarize the conclusions of this Section: i) the COQI element is inadequate for tests involving large membrane strains and ii) the CQD3 triangular shell element (and also the parabolic element CQD6 which behaves similarly) gives wrong total strain results, although the main engineering results (stresses and forces) are correct.
All the input files used in the previous Sections are listed below.

**Appendix B — Input files**

1. **c272g2.epx**
2. **c272g2a.epx**
3. **c272g2p.epx**

**c272g2.epx**

```plaintext
VITE -5.0 FDNC 1 LECT 1 2 3 4 9 15 16 17 18 TERM ! Relative velocity
VITE 3.0 0.5 FDNC 1 LECT 5 6 7 8 13 14 24 25 26 TERM ! a/s
NELA 4.0 EQL 1 LECT 1 4 5 6 8 13 15 16 19 22 26 27 TERM
EQL 1 LECT 2 3 6 7 11 16 20 21 24 TERM
EQL 2 LECT 1 5 6 10 15 20 23 TERM
EQL 2 LECT 3 4 7 8 17 21 22 25 TERM
EQL 3 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 TERM

21 22 23 24 25 26

**c272g2a.epx**

```plaintext
ALL the input files used in the previous Sections are listed below.

**c272g2p.epx**

```plaintext
VITE 3.0 FDNC 1 LECT 1 2 3 4 9 15 16 17 18 TERM ! Relative velocity
VITE 3.0 0.5 FDNC 1 LECT 5 6 7 8 13 14 24 25 26 TERM ! a/s
NELA 4.0 EQL 1 LECT 1 4 5 6 8 13 15 16 19 22 26 27 TERM
EQL 1 LECT 2 3 6 7 11 16 20 21 24 TERM
EQL 2 LECT 1 5 6 10 15 20 23 TERM
EQL 2 LECT 3 4 7 8 17 21 22 25 TERM
```
SUBROUTINE MULMA3(A,B,C)
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(3,3),B(3,3)
REAL(8), INTENT(OUT) :: C(3,3)
INTEGER :: I,J,K
* compute c = a . b
DO 50 I=1,3
DO 40 J=1,3
C(I,J)=0.
DO 30 K=1,3
C(I,J)=C(I,J)+A(I,K)*B(K,J)
30 CONTINUE
40 CONTINUE
50 CONTINUE
END SUBROUTINE MULMA3

SUBROUTINE MULMA2(A,B,C)
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(2,2),B(2,2)
REAL(8), INTENT(OUT) :: C(2,2)
INTEGER :: I,J,K
* compute c = a . b
DO 50 I=1,2
DO 40 J=1,2
C(I,J)=0.
DO 30 K=1,2
C(I,J)=C(I,J)+A(I,K)*B(K,J)
30 CONTINUE
40 CONTINUE
50 CONTINUE
END SUBROUTINE MULMA2

SUBROUTINE MATRA3(A,B)
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(3,3)
REAL(8), INTENT(OUT) :: B(3,3)
* compute b = a ^ T
B(1,1) = A(1,1)
B(1,2) = A(2,1)
B(1,3) = A(3,1)
B(2,1) = A(1,2)
B(2,2) = A(2,2)
B(2,3) = A(3,2)
B(3,1) = A(1,3)
B(3,2) = A(2,3)
B(3,3) = A(3,3)
END SUBROUTINE MATRA3

SUBROUTINE MATRA2(A,B)
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(2,2)
REAL(8), INTENT(OUT) :: B(2,2)
* compute b = a ^ T
B(1,1) = A(1,1)
B(1,2) = A(2,1)
B(2,1) = A(1,2)
B(2,2) = A(2,2)
END SUBROUTINE MATRA2

SUBROUTINE PRMAT3(LAB,A)
IMPLICIT NONE
CHARACTER*3, INTENT(IN) :: LAB
REAL(8), INTENT(IN) :: A(3,3)
WRITE(6,1) LAB
1 FORMAT (A3,':')
WRITE(6,2) A(1,1), A(1,2), A(1,3)
WRITE(6,2) A(2,1), A(2,2), A(2,3)
WRITE(6,2) A(3,1), A(3,2), A(3,3)
2 FORMAT (1P3E12.5)
END SUBROUTINE PRMAT3

SUBROUTINE PRMAT2(LAB,A)
IMPLICIT NONE
CHARACTER*3, INTENT(IN) :: LAB
REAL(8), INTENT(IN) :: A(2,2)
WRITE(6,1) LAB
1 FORMAT (A3,':')
WRITE(6,2) A(1,1), A(1,2)
WRITE(6,2) A(2,1), A(2,2)
2 FORMAT (1P2E12.5)
END SUBROUTINE PRMAT2

SUBROUTINE MULMA3(A,B,C)
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(3,3),B(3,3)
REAL(8), INTENT(OUT) :: C(3,3)
INTEGER :: I,J,K
* compute c = a . b
DO 50 I=1,3
DO 40 J=1,3
C(I,J)=0.
DO 30 K=1,3
C(I,J)=C(I,J)+A(I,K)*B(K,J)
30 CONTINUE
40 CONTINUE
50 CONTINUE
END SUBROUTINE MULMA3

SUBROUTINE MULMA2(A,B,C)
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(2,2),B(2,2)
REAL(8), INTENT(OUT) :: C(2,2)
INTEGER :: I,J,K
* compute c = a . b
DO 50 I=1,2
DO 40 J=1,2
C(I,J)=0.
DO 30 K=1,2
C(I,J)=C(I,J)+A(I,K)*B(K,J)
30 CONTINUE
40 CONTINUE
50 CONTINUE
END SUBROUTINE MULMA2

SUBROUTINE MATRA3(A,B)
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(3,3)
REAL(8), INTENT(OUT) :: B(3,3)
* compute b = a ^ T
B(1,1) = A(1,1)
B(1,2) = A(2,1)
B(1,3) = A(3,1)
B(2,1) = A(1,2)
B(2,2) = A(2,2)
B(2,3) = A(3,2)
B(3,1) = A(1,3)
B(3,2) = A(2,3)
B(3,3) = A(3,3)
END SUBROUTINE MATRA3

SUBROUTINE MATRA2(A,B)
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(2,2)
REAL(8), INTENT(OUT) :: B(2,2)
* compute b = a ^ T
B(1,1) = A(1,1)
B(1,2) = A(2,1)
B(2,1) = A(1,2)
B(2,2) = A(2,2)
END SUBROUTINE MATRA2

chkeps2.txt
bet= 4.522
s =-0.982
c =-0.189
s2 = 0.964
c2 = 0.036
sc = 0.186
sc2= 0.371

R :
-1.88976E-01-9.81982E-01
9.81982E-01-1.88976E-01

EL :
1.35400E+00 3.85060E-01
3.85060E-01-8.04210E-01

RT :
-1.88976E-01 9.81982E-01
-9.81982E-01-1.88976E-01

TMP:
1.22249E-01-1.40237E+00
-8.62487E-01-2.26146E-01

EG :
-8.70048E-01 4.29426E-02
4.29426E-02 1.41984E+00

chkeps3.f
PROGRAM chkeps3
IMPLICIT NONE
REAL(8) :: el(3,3), eg(3,3), r(3,3), rt(3,3), tmp(3,3)
REAL(8) :: pgr, bet, s, c, s2, c2, sc, sc2
DATA pgr /3.141592654D0/
bet = 259.107E0*pgr/180.D0
s = SIN (bet)
c = COS (bet)
s2 = s*s
sc = s*c
sc2 = 2.D0*s*c
WRITE (6,1) bet, s, c, s2, c2, sc, sc2
1 FORMAT(' bet=',F10.3,/,>
' s =',F6.3,/,>
' c =',F6.3,/,>
' s2 =',F6.3,/,>
' c2 =',F6.3,/,>
' sc =',F6.3,/,>
' sc2=',F6.3)
r(1,1) = c
r(1,2) = s
r(1,3) = 0.D0
r(2,1) = -s
r(2,2) = c
r(2,3) = 0.D0
r(3,1) = 0.D0
r(3,2) = 0.D0
r(3,3) = 1.D0
CALL PRMAT3 ('R ',r)
el(1,1) = 1.3540D0
el(1,2) = 0.77012D0 / 2.D0
el(1,3) = 0.00
el(2,1) = 0.77012D0 / 2.00
el(2,2) = -0.80410D0
el(2,3) = 0.00
el(3,1) = 0.00
el(3,2) = 0.00
el(3,3) = -0.54500D0
CALL PRMAT3 ('EL ',el)
CALL MULMA3 (el, r, tmp)
CALL PRMAT3 ('TMP',tmp)
CALL MULMA3 (rt, tmp, eg)
CALL PRMAT3 ('EG ',eg)
END PROGRAM chkeps3

SUBROUTINE PRMAT3(LAB,A)
IMPLICIT NONE
CHARACTER*3, INTENT(IN) :: LAB
REAL(8), INTENT(IN) :: A(3,3)
WRITE(6,1) LAB
1 FORMAT (A3,':')
WRITE(6,2) A(1,1), A(1,2), A(1,3)
WRITE(6,2) A(2,1), A(2,2), A(2,3)
WRITE(6,2) A(3,1), A(3,2), A(3,3)
2 FORMAT (1P3E12.5)
END SUBROUTINE PRMAT3

SUBROUTINE PRMAT2(LAB,A)
IMPLICIT NONE
CHARACTER*3, INTENT(IN) :: LAB
REAL(8), INTENT(IN) :: A(2,2)
WRITE(6,1) LAB
1 FORMAT (A3,':')
WRITE(6,2) A(1,1), A(1,2)
WRITE(6,2) A(2,1), A(2,2)
2 FORMAT (1P2E12.5)
END SUBROUTINE PRMAT2
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(3,3)
REAL(8), INTENT(OUT) :: B(3,3)

* compute b = a ^ T
B(1,1) = A(1,1)
B(1,2) = A(2,1)
B(1,3) = A(3,1)
B(2,1) = A(1,2)
B(2,2) = A(2,2)
B(2,3) = A(3,2)
B(3,1) = A(1,3)
B(3,2) = A(2,3)
B(3,3) = A(3,3)

END SUBROUTINE MATRA3

*-----------------------------------------------------------------------
SUBROUTINE MATRA2(A,B)
* computes transpose of a 2x2 matrix
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(2,2)
REAL(8), INTENT(OUT) :: B(2,2)

* compute b = a ^ T
B(1,1) = A(1,1)
B(1,2) = A(2,1)
B(2,1) = A(1,2)
B(2,2) = A(2,2)

END SUBROUTINE MATRA2

chkeps3.txt

bet= 4.522
s =-0.982
\(c = -0.189\)
s2 = 0.964
c2 = 0.036
sc = 0.186
sc2= 0.371

R :
-1.88976E-01 9.81982E-01 0.00000E+00
9.81982E-01-1.88976E-01 0.00000E+00
0.00000E+00 0.00000E+00 1.00000E+00

EL :
1.35400E+00 3.85060E-01 0.00000E+00
3.85060E-01-8.04210E-01 0.00000E+00
0.00000E+00 0.00000E+00 -5.48200E-01

RT :
-1.88976E-01 9.81982E-01 0.00000E+00
-9.81982E-01-1.88976E-01 0.00000E+00
0.00000E+00 0.00000E+00 1.00000E+00

TMP:
1.22249E-01-1.40237E+00 0.00000E+00
-8.62487E-01-2.26146E-01 0.00000E+00
0.00000E+00 0.00000E+00 -5.48200E-01

EG :
-8.70048E-01 4.29426E-02 0.00000E+00
4.29426E-02 1.41984E+00 0.00000E+00
0.00000E+00 0.00000E+00 -5.48200E-01

chksig2.f

PROGRAM chksig2
IMPLICIT NONE
REAL(8) :: sl(2,2), sg(2,2), r(2,2), rt(2,2), tmp(2,2)
REAL(8) :: pgr, bet, s, c, s2, c2, sc, sc2
DATA pgr /3.141592654D0/

* 
 bet = 259.107E0*pgr/180.D0
 s = SIN (bet)
c = COS (bet)
s2 = s*s
c2 = c*c
sc = s*c
sc2 = 2.D0*s*c

WRITE (6,1) bet, s, c, s2, c2, sc, sc2
1 FORMAT(' bet=',F10.3,/
> ' s =',F6.3,/
> ' c =',F6.3,/
> ' s2 =',F6.3,/
> ' c2 =',F6.3,/
> ' sc =',F6.3,/
> ' sc2=',F6.3)
* 
 r(1,1) = c
 r(1,2) = s
 r(2,1) = -s
 r(2,2) = c
 CALL PRMAT2 ('R ',r)

 sl(1,1) = 9.7537D8
 sl(1,2) = 1.8781D8
 sl(2,1) = 1.8781D8
 sl(2,2) = 0.3618D8
 CALL PRMAT2 ('SL ',sl)

 CALL MATRA2 (r, rt)
 CALL PRMAT2 ('RT ',rt)
 CALL MULMA2 (sl, r, tmp)
 CALL PRMAT2 ('TMP',tmp)
 CALL MULMA2 (rt, tmp, sg)
 CALL PRMAT2 ('SG ',sg)

* END PROGRAM chksig2

*-----------------------------------------------------------------------
SUBROUTINE PRMAT3(LAB,A)
* print a 3x3 matrix
IMPLICIT NONE
CHARACTER*3, INTENT(IN) :: LAB
REAL(8), INTENT(IN) :: A(3,3)
WRITE(6,1) LAB
1 FORMAT (A3,':')
WRITE(6,2) A(1,1), A(1,2), A(1,3)
WRITE(6,2) A(2,1), A(2,2), A(2,3)
WRITE(6,2) A(3,1), A(3,2), A(3,3)
2 FORMAT (1P3E12.5)

END SUBROUTINE PRMAT3

*-----------------------------------------------------------------------
SUBROUTINE PRMAT2(LAB,A)
* print a 2x2 matrix
IMPLICIT NONE
CHARACTER*3, INTENT(IN) :: LAB
REAL(8), INTENT(IN) :: A(2,2)
WRITE(6,1) LAB
1 FORMAT (A3,':')
WRITE(6,2) A(1,1), A(1,2)
WRITE(6,2) A(2,1), A(2,2)
2 FORMAT (1P2E12.5)

END SUBROUTINE PRMAT2

*-----------------------------------------------------------------------
SUBROUTINE MULMA3(A,B,C)
* multiplies two 3x3 matrices
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(3,3),B(3,3)
REAL(8), INTENT(OUT) :: C(3,3)
INTEGER :: I,J,K
* compute c = a . b
DO 50 I=1,3
Do 40 J=1,3
C(I,J)=0.
Do 30 K=1,3
C(I,J)=C(I,J)+A(I,K)*B(K,J)
30 CONTINUE
40 CONTINUE
50 CONTINUE

END SUBROUTINE MULMA3

*-----------------------------------------------------------------------
SUBROUTINE MULMA2(A,B,C)
* multiplies two 2x2 matrices
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(2,2),B(2,2)
REAL(8), INTENT(OUT) :: C(2,2)
INTEGER :: I,J,K
* compute c = a . b
Do 50 I=1,2
Do 40 J=1,2
C(I,J)=0.
Do 30 K=1,2
C(I,J)=C(I,J)+A(I,K)*B(K,J)
30 CONTINUE
40 CONTINUE
50 CONTINUE

END SUBROUTINE MULMA2

*-----------------------------------------------------------------------
SUBROUTINE MATRA3(A,B)
* computes transpose of a 3x3 matrix
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(3,3)
REAL(8), INTENT(OUT) :: B(3,3)

* compute b = a ^ T
B(1,1) = A(1,1)
B(1,2) = A(2,1)
B(1,3) = A(3,1)
B(2,1) = A(1,2)
B(2,2) = A(2,2)
B(2,3) = A(3,2)
B(3,1) = A(1,3)
B(3,2) = A(2,3)
B(3,3) = A(3,3)

END SUBROUTINE MATRA3

*-----------------------------------------------------------------------
SUBROUTINE MATRA2(A,B)
* computes transpose of a 2x2 matrix
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(2,2)
REAL(8), INTENT(OUT) :: B(2,2)

* compute b = a ^ T
B(1,1) = A(1,1)
B(1,2) = A(2,1)
B(2,1) = A(1,2)
B(2,2) = A(2,2)

END SUBROUTINE MATRA2
**chksig3.f**

```fortran
! PROGRAM chksig3
IMPLICIT NONE
REAL(8) :: x(3,3), y(3,3), r(3,3), t(3,3), p(3,3)
DATA pgr /3.141592654D0/
! compute b = a^T
! computes transpose of a 2x2 matrix
*-----------------------------------------------------------------------
* compute c = a . b
* multiplies two 2x2 matrices
IMPLICIT NONE
REAL(8), INTENT(IN) :: A(2,2), B(2,2)
REAL(8), INTENT(INOUT) :: C(2,2)
INTEGER :: i, j, k
! compute a = b . c
DO 50 I = 1,2
  DO 40 J = 1,2
    C(I,J) = A(I,J) * B(J,I)
 40 CONTINUE
50 CONTINUE
END SUBROUTINE MATRA2
```

```fortran
! print a 3x3 matrix
IMPLICIT NONE
CHARACTER(3), INTENT(IN) :: LAB
REAL(8), INTENT(IN) :: A(3,3)
WRITE(6,1) LAB
1 FORMAT(' bet=',F10.3,/, ' sc=',F6.3,/, ' s=',F6.3,/, ' sc2=',F6.3,/, ' c2=',F6.3)
*-----------------------------------------------------------------------
* print a 2x2 matrix
IMPLICIT NONE
CHARACTER(3), INTENT(IN) :: LAB
REAL(8), INTENT(IN) :: A(2,2)
WRITE(6,1) LAB
1 FORMAT(' bet=',F10.3,/, ' sc=',F6.3,/, ' s=',F6.3,/, ' sc2=',F6.3,/, ' c2=',F6.3)
*-----------------------------------------------------------------------
SUBROUTINE MATRA3(LAB,C)
IMPLICIT NONE
CHARACTER(3), INTENT(IN) :: LAB
REAL(8), INTENT(OUT) :: C(2,2)
END SUBROUTINE MATRA3
```

**chksig3.txt**

```plaintext
chksig3.txt
```

```plaintext
37
```

**coqis3.epx**

```plaintext
# CGQ13
# CONV WIN
# LAGR TRID
# GEOM LIBR POIN 4 COQ1 2 TERM
# 0.5 -0.5 0.0 0.5 -0.5 0.0 0.5 -0.5 0.0 -0.5 -0.5 0.0
# 1 2 3
# 3 4 1
```

---

**coqis3.epx**

```plaintext
# CGQ13
# CONV WIN
# LAGR TRID
# GEOM LIBR POIN 4 COQ1 2 TERM
-0.5 -0.5 0.0 0.5 -0.5 0.0 0.5 -0.5 0.0 -0.5 -0.5 0.0
1 2 3
3 4 1
COPP EPI 1.0 LECT 1 2 TERM
MATE VPCS RS 7850.0
YDDN 2.1E11
NU 0.53
ELAS 3.70E8
```
cqd4s3.epx

CQD4S3
ECNO
"CQD4S3.epx"
REFE FRAM
INIT WIRE
VECT SCIO FIEX SCAL USER PRG 0.0 0.0 0.49 TERM
TEXT VSCA
COLD PAPE
SLEF CAM 1 NPRA 1
FREQ 10
TRAC OFFS FICH AVI NOCL NFTO 101 FPS 15 KFRE 10 COMP -1 DEFO REND
ENDPLAY
TRAC OFFS FICH AVI CONT DEFO REND
DEPLOY
FIN

CQD9G3.epx

CQD9G3
ECNO
"CQD9G3.epx"
REFE FRAM
INIT WIRE
VECT SCIO FIEX SCAL USER PRG 0.0 0.0 0.49 TERM
TEXT VSCA
COLD PAPE
SLEF CAM 1 NPRA 1
FREQ 10
TRAC OFFS FICH AVI NOCL NFTO 101 FPS 15 KFRE 10 COMP -1 DEFO REND
ENDPLAY
TRAC OFFS FICH AVI CONT DEFO REND
DEPLOY
FIN
m = 0.00
10 = 1.00
dt = 0.05
it2 = 2.00

tel = 2.00 x / a
dtel = tel / nel

t0 = -tel
t = 0.00
write (6,1) t, d

1 FORMAT (1P,2E23.15)

END DO
DO i = 1, n
b = (1.D0 + pdotstar)**c
pdotstar = k / pdot0

n = NINT (t2 / dt)

END DO
write (6,1) t, d

d = 0.00

t = t0

t0 = -tel

tel = 2.D0 / a
	nel = 100

t2 = 2.00

dt = 4.D-5

l0 = 1.D0

m = 0.D0

COUR 1 'dy_1' DEPL COMP 2 NOEU LECT 1 TERM

COUR 2 'dy_2' DEPL COMP 2 NOEU LECT 2 TERM

COUR 3 'dy_3' DEPL COMP 2 NOEU LECT 3 TERM

COUR 4 'dy_4' DEPL COMP 2 NOEU LECT 4 TERM

AXTE 1.0 'Time [s]'

RESU ALIC GARD PSCR

SUIT

CALC TINI 0 TFIN 2.0 PASF 2.E-5 NMAX 100000

OPTI NOTE PAS UTIL LOG 1000

ECRI COOR DEPL VITE ACCE FINT FEXT FLIA FDEC CONT EPST ECRO FREQ 10000

INIT VITE 2 -0.5 LECT 1 2 TERM ! Relative velocity 1 m/s

CALC TINO 0 TFIN 2.0 PASP 2.5-8 NMAX 100000

SUIT

Post-treatment

END

SDKT3S

END
FIN
ENDPLAY
GO
GOTR LOOP 99 OFFS FICH AVI CONT NOCL DEFO REND
TRAC OFFS FICH AVI NOCL NFTO 101 FPS 15 KFRE 10 COMP -1 DEFO REND
FREQ 10
SCEN GEOM NAVI FREE
!NEAR : 6.16644E+00
!ASPECT : 1.00000E+00
!RADIUS : 7.90569E+00
!RSPHERE: 1.58114E+00
!CENTER : 0.00000E+00 0.00000E+00 0.00000E+00
!NAVIGATION MODE: ROTATING CAMERA
MATE VPJC RO 7850.0
COMP EPAI 1.0 LECT 1 TERM
GEOM LIBR POIN 2 ED01 1 TERM
!CONV WIN
ECHO
! Q 1.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
PLAY
SORT VISU NSTO 1
OPTI PRIN
ECHO
DST3S3A
QUAL EPST COMP 1 LECT 1 TERM REFE 0.10986E+01 TOLE 5.E-3 ! EPS-ZZ
TRAC 50 AXES 1.0 'TOT. FORCE [N]' YZER
TRAC 32 AXES 1.0 'EQ. STRESS [PA]' XAXE 33 1.0 'EQ PL STRAIN [-]' YZER
TRAC 11 AXES 1.0 'STRESS [PA]' XAXE 21 1.0 'STRAIN [-]' YZER
TRAC 21 22 23 24 AXES 1.0 'STRAIN [-]' YZER
TRAC 11 12 13 14 AXES 1.0 'STRESS [PA]' YZER
COUR 50 REFE 0.33720E+09 TOLE 5.E-3 ! FY-SUM
FIN
dst3s3a.epx

POST-treatment
ECOD
RESU ALIC GARD PSCR
OPTI NOTE PAS UTIL LOG 1000
ECRI COOR DEPL VITE ACCE FINT FEXT FLIA FDEC CONT EPST ECRO FREQ 10000
FICH ALIC FREQ 100
FIN

fuvp01.epx

Post-treatment
ECOD
RESU ALIC GARD PSCR
OPTI NOTE PAS UTIL LOG 1000
ECRI COOR DEPL VITE ACCE FINT FEXT FLIA FDEC CONT EPST ECRO FREQ 10000
FICH ALIC FREQ 100
FIN

ed0101.epx
COUR 21 \{+1\} EPST COMP 1 ELEM LECT 1 TERM
COUR 22 \{+2\} EPST COMP 2 ELEM LECT 1 TERM
COUR 23 \{+3\} EPST COMP 3 ELEM LECT 1 TERM
COUR 24 \{+4\} EPST COMP 4 ELEM LECT 1 TERM
COUR 31 \{-1\} ECHO COMP 3 ELEM LECT 1 TERM
COUR 32 \{-2\} ECHO COMP 2 ELEM LECT 1 TERM
COUR 33 \{-3\} ECHO COMP 3 ELEM LECT 1 TERM
COUR 50 \{fy\}_sum FILLA COMP 2 ZONE LECT 2 TERM
TRAC 1 2 AXES 1.0 DISPL. [M] YZER
TRAC 5 6 AXES 1.0 FORCE [N] YZER
TRAC 11 12 13 14 AXES 1.0 STRAIN [\%] YZER
TRAC 21 22 23 24 AXES 1.0 STRAIN [\%] YZER
TRAC 11 AXES 1.0 STRAIN [\%] YZER
TRAC 21 22 23 24 AXES 1.0 STRAIN [\%] YZER
COUR 50 \{fy\}_sum FILLA COMP 2 ZONE LECT 2 TERM
TRAC 1 2 AXES 1.0 DISPL. [M] YZER
TRAC 5 6 AXES 1.0 FORCE [N] YZER
ECHO PR6002
MATE VPJC RO 7850.0
COMP EPAI 1.0 LECT 1 TERM
GEOM LIBR POIN 2 FUN2 1 TERM
LAGR CPLA
TRAC 50 AXES 1.0 TOT. FORCE [N] YZER
QUAL EPST COMP 1 LECT 1 TERM REFE 0.10986E+01 TOLE 1.0E-3 EPS-XX
EPST COMP 2 LECT 1 TERM REFE -0.54916E+00 TOLE 1.0E-3 EPS-YY
EPST COMP 4 LECT 1 TERM REFE -0.54916E+00 TOLE 1.0E-3 EPS-ZZ
CONT LECT 1 TERM REFE 0.10116E+10 TOLE 1.0E-3 SIG-XX
COUR 50 REFE 0.33720E+09 TOLE 1.0E-3 FY-SUM
FIN

FUVPO2.epx

FUVPO2 ECHO
!CONV WIN
ECHO
PR6002
MATE VPJC RO 7850.0
COMP EPAI 1.0 LECT 1 TERM
GEOM LIBR POIN 2 FUN2 1 TERM
LAGR CPLA
TRAC 50 AXES 1.0 TOT. FORCE [N] YZER
QUAL EPST COMP 1 LECT 1 TERM REFE 0.10986E+01 TOLE 1.0E-3 EPS-XX
EPST COMP 2 LECT 1 TERM REFE -0.54916E+00 TOLE 1.0E-3 EPS-YY
EPST COMP 4 LECT 1 TERM REFE -0.54916E+00 TOLE 1.0E-3 EPS-ZZ
CONT LECT 1 TERM REFE 0.10116E+10 TOLE 1.0E-3 SIG-XX
COUR 50 REFE 0.33720E+09 TOLE 1.0E-3 FY-SUM
FIN

fuvp02.epx
**q4grs3.epx**

```
CONV WIN
SUIT
CALC TINI 0 TFIN 2.0 PASF 2.E-5 NMAX 100000
ECRI COOR DEPL VITE ACCE FINT FEXT FLIA FONC 1 LECT 1 2 TERM
INIT VITE 2 -0.5 LECT 1 2 TERM
```

**q4grs3a.epx**

```
CONV WIN
```

**q4grs3a.epx**

```
CONV WIN
```

**q4grs3.epx**

```
CONV WIN
LAG TRID
GEM LISM FIGH 4 Q4GR 1 TERM
```

```
VITE 2 0.5 FONC 1 LECT 1 2 TERM
VITE 2 0.5 FONC 1 LECT 3 4 TERM
INIT VITE 2 0.5 LECT 1 2 TERM
ECNI CODR DEPL VITE ACCE FINT FEXT FLIA FONC 1 LECT 3 4 TERM
FICH ACLR FREQ 100
OPTI NOTE PAS UTIL LOG 1000
```

```
CALC TINI 0 TFIN 2.0 PASF 2.E-5 HMAX 100000
SUIT
Post-treatment
ECNO
```

```
CONV WIN
```

```
LINC TRID
```

```
LINC TRID
```

```
CONV WIN
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```

```
LINC TRID
```
TRAC 5 6 7 8 AXES 1.0 'FORCE [N]' YZER
TRAC 11 12 13 14 15 16 AXES 1.0 'STRESS [PA]' YZER
TRAC 21 22 23 24 25 26 AXES 1.0 'STRAIN [-]' YZER
TRAC 14 AXES 1.0 'STRESS [PA]' AXE 24 1.0 'STRAIN [-]' YZER
TRAC 52 AXES 1.0 'EQ. STRESS [PA]' AXE 33 1.0 'EQ PL STRAIN [-]' YZER
TRAC 50 AXES 1.0 'TOT. FORCE [N]' YZER
QUAL EPST 1 LECT 1 TERM REFE -0.54931*00 TOLE 1.0*E-2 EPS-XX
EPST COMP 2 LECT 1 TERM REFE 0.10986*01 TOLE 1.0*EPS-YY
EPST COMP 4 LECT 1 TERM REFE -0.54931*00 TOLE 1.0*EPS-ZZ
CONT COMP 2 LECT 1 TERM REFE 0.10116*10 TOLE 1.0* EPS-DY
CONT COMP 2 LECT 1 TERM REFE 0.53700*09 TOLE 1.0* EPS-SY
FIN

* END PROGRAM sigma

TRAC 52 AXES 1.0 'TOT. FORCE [N]' YZER
TRAC 11 12 13 14 15 16 AXES 1.0 'STRESS [PA]' YZER
TRAC 21 22 23 24 25 26 AXES 1.0 'STRAIN [-]' YZER
TRAC 14 AXES 1.0 'STRESS [PA]' AXE 24 1.0 'STRAIN [-]' YZER
TRAC 52 AXES 1.0 'EQ. STRESS [PA]' AXE 33 1.0 'EQ PL STRAIN [-]' YZER
TRAC 50 AXES 1.0 'TOT. FORCE [N]' YZER
QUAL EPST 1 LECT 1 TERM REFE -0.54931*00 TOLE 1.0*EPS-XX
EPST COMP 2 LECT 1 TERM REFE 0.10986*01 TOLE 1.0*EPS-YY
EPST COMP 4 LECT 1 TERM REFE -0.54931*00 TOLE 1.0*EPS-ZZ
CONT COMP 2 LECT 1 TERM REFE 0.10116*10 TOLE 1.0* EPS-DY
CONT COMP 2 LECT 1 TERM REFE 0.53700*09 TOLE 1.0* EPS-SY
FIN

* END PROGRAM sigma

TRAC 52 AXES 1.0 'TOT. FORCE [N]' YZER
TRAC 11 12 13 14 15 16 AXES 1.0 'STRESS [PA]' YZER
TRAC 21 22 23 24 25 26 AXES 1.0 'STRAIN [-]' YZER
TRAC 14 AXES 1.0 'STRESS [PA]' AXE 24 1.0 'STRAIN [-]' YZER
TRAC 52 AXES 1.0 'EQ. STRESS [PA]' AXE 33 1.0 'EQ PL STRAIN [-]' YZER
TRAC 50 AXES 1.0 'TOT. FORCE [N]' YZER
QUAL EPST 1 LECT 1 TERM REFE -0.54931*00 TOLE 1.0*EPS-XX
EPST COMP 2 LECT 1 TERM REFE 0.10986*01 TOLE 1.0*EPS-YY
EPST COMP 4 LECT 1 TERM REFE -0.54931*00 TOLE 1.0*EPS-ZZ
CONT COMP 2 LECT 1 TERM REFE 0.10116*10 TOLE 1.0* EPS-DY
CONT COMP 2 LECT 1 TERM REFE 0.53700*09 TOLE 1.0* EPS-SY
FIN

* END PROGRAM sigma
List of input files

<table>
<thead>
<tr>
<th>C</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>c272g2.epx</td>
<td>pr6002.epx</td>
</tr>
<tr>
<td>c272g2a.epx</td>
<td>pr6002a.epx</td>
</tr>
<tr>
<td>c272g2p.epx</td>
<td>pr6002p.epx</td>
</tr>
<tr>
<td>c273g2.epx</td>
<td>pris02.epx</td>
</tr>
<tr>
<td>c273g2a.epx</td>
<td>pris02a.epx</td>
</tr>
<tr>
<td>c273g2p.epx</td>
<td>pris02p.epx</td>
</tr>
<tr>
<td>c81102.epx</td>
<td></td>
</tr>
<tr>
<td>c82102.epx</td>
<td></td>
</tr>
<tr>
<td>car101.epx</td>
<td></td>
</tr>
<tr>
<td>car401.epx</td>
<td></td>
</tr>
<tr>
<td>chkeps2.f</td>
<td></td>
</tr>
<tr>
<td>chkeps2.txt</td>
<td></td>
</tr>
<tr>
<td>chkeps3.f</td>
<td></td>
</tr>
<tr>
<td>chkeps3.txt</td>
<td></td>
</tr>
<tr>
<td>chksig2.f</td>
<td></td>
</tr>
<tr>
<td>chksig2.txt</td>
<td></td>
</tr>
<tr>
<td>chksig3.f</td>
<td></td>
</tr>
<tr>
<td>chksig3.txt</td>
<td></td>
</tr>
<tr>
<td>coqis3.epx</td>
<td></td>
</tr>
<tr>
<td>coqis3a.epx</td>
<td></td>
</tr>
<tr>
<td>cqd3s3.epx</td>
<td></td>
</tr>
<tr>
<td>cqd3s3a.epx</td>
<td></td>
</tr>
<tr>
<td>cqd4s3.epx</td>
<td></td>
</tr>
<tr>
<td>cqd4s3a.epx</td>
<td></td>
</tr>
<tr>
<td>cqd9g3.epx</td>
<td></td>
</tr>
<tr>
<td>cqd9g3a.epx</td>
<td></td>
</tr>
<tr>
<td>cqd9g3a.epx</td>
<td></td>
</tr>
<tr>
<td>cqd9g3a.epx</td>
<td></td>
</tr>
<tr>
<td>cub802.epx</td>
<td></td>
</tr>
<tr>
<td>cube02.epx</td>
<td></td>
</tr>
<tr>
<td>cube02a.epx</td>
<td></td>
</tr>
<tr>
<td>cube05.epx</td>
<td></td>
</tr>
<tr>
<td>cube05a.epx</td>
<td></td>
</tr>
<tr>
<td>cube05a.epx</td>
<td></td>
</tr>
<tr>
<td>cube05p.epx</td>
<td></td>
</tr>
<tr>
<td>cube05q.epx</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>displ.f</td>
<td>sigma.f</td>
</tr>
<tr>
<td>dst3s3.epx</td>
<td></td>
</tr>
<tr>
<td>dst3s3a.epx</td>
<td></td>
</tr>
<tr>
<td>dst3s3.epx</td>
<td></td>
</tr>
<tr>
<td>dst3s3a.epx</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>ed0101.epx</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>F</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>fuvp01.epx</td>
<td></td>
</tr>
<tr>
<td>fuvp02.epx</td>
<td></td>
</tr>
</tbody>
</table>
How to obtain EU publications

Our publications are available from EU Bookshop (http://bookshop.europa.eu), where you can place an order with the sales agent of your choice.

The Publications Office has a worldwide network of sales agents. You can obtain their contact details by sending a fax to (352) 29 29-42758.
JRC Mission

As the Commission’s in-house science service, the Joint Research Centre’s mission is to provide EU policies with independent, evidence-based scientific and technical support throughout the whole policy cycle.

Working in close cooperation with policy Directorates-General, the JRC addresses key societal challenges while stimulating innovation through developing new methods, tools and standards, and sharing its know-how with the Member States, the scientific community and international partners.

Serving society
Stimulating innovation
Supporting legislation