CalculiX CrunchiX USER'S MANUAL
development version

Guido Dhondt

February 15, 2005

Contents

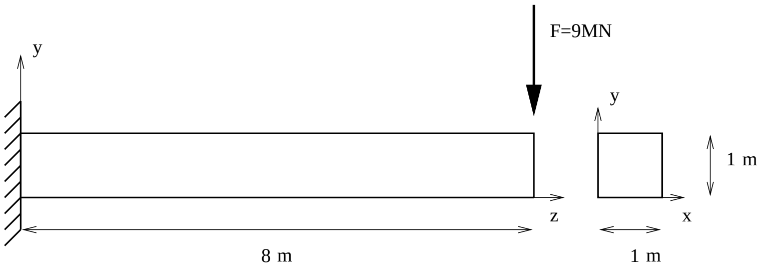

Figure 1: Geometry and boundary conditions of the beam problem

1 Introduction.

This is a description of CalculiX CrunchiX. If you have any problems using the program, this document should solve them. If not, send us an E-mail (dhondt@t-online.de). The next section contains a simple example problem to wet your appetite. Section three is a theoretical section giving some background on the analysis types, elements, materials etc. Then, an overview is given of all the available keywords in alphabetical order, followed by detailed instructions on the format of the input deck. If CalculiX does not run because your input deck has problems, this is the section to look at. The following section contains a description of the verification examples you should have obtained along with the code of the program. If you try to solve a new kind of problem you haven't dealt with in the past, check this section for examples. You can also use this section to check whether you installed CalculiX correctly. Finally, the User's Manual ends with some references used while writing the code.
This manual is not a textbook on finite elements. Indeed, a working knowledge of the Finite Element Method is assumed. For people not familiar with the Finite Element Method, I recommend the book by Zienkiewicz and Taylor [27] for engineering oriented students and the publication by Hughes [10] for mathematically minded readers.

2 Simple example problem

In this section, a cantilever beam loaded by point forces at its free end is analyzed. The geometry, loading and boundary conditions are shown in Figure 1. The size of the beam is 1x1x8 m^3, the loading consists of a point force of $9x10^6$ N and the beam is completely fixed (in all directions) on the left end. Let us take 1 m and 1 MN as units of length and force, respectively. Assume that

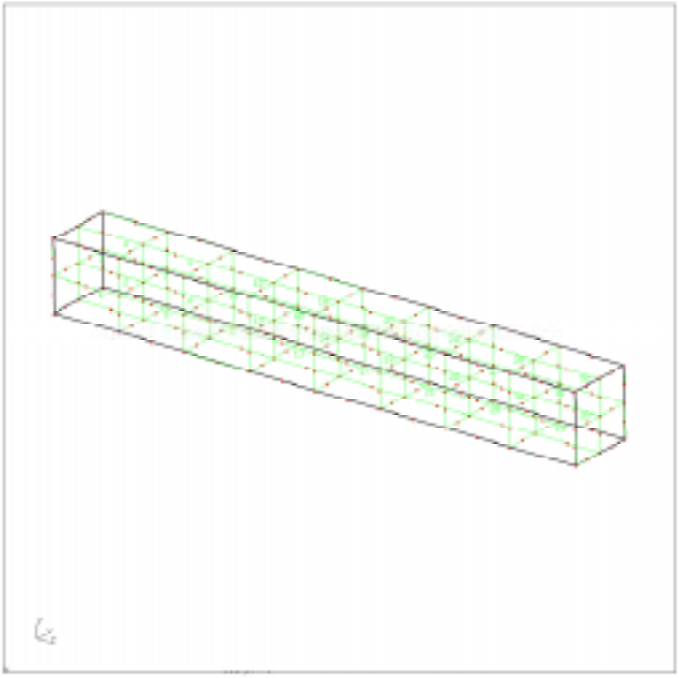

Figure 2: Mesh for the beam

the beam geometry was generated and meshed with CalculiX GraphiX (cgx) resulting in the mesh in Figure 2. For reasons of clarity, only element labels are displayed.

A CalculiX input deck basically consists of a model definition section describing the geometry and boundary conditions of the problem and one or more steps (Figure 3) defining the loads.

The model definition section starts at the beginning of the file and ends at the occurrence of the first *STEP card. All input is preceded by keyword cards, which all start with an asterisk (*), indicating the kind of data which follows. *STEP is such a keyword card. Most keyword cards are either model definition cards (i.e. they can only occur before the first *STEP card) or step cards (i.e. they can only occur between *STEP and *END STEP cards). A few can be both.

In our example (Figure 4), the first keyword card is *HEADING, followed by a short description of the problem. This has no effect on the output and only serves for identification. Then, the coordinates are given as triplets preceded by the *NODE keyword. Notice that data on the same line are separated by comma's and must not exceed a record length of 132 columns. A keyword card can be repeated as often as needed. For instance, each node could have been preceded by its own *NODE keyword card.

Next, the topology is defined by use of the keyword card *ELEMENT. Defining the topology means listing for each element its type, which nodes belong to the element and in what order. The element type is a parameter on the keyword card. In the beam case 20-node brick elements with reduced integration have been used, abbreviated as C3D20R. In addition, by adding ELSET=Eall, all elements following the *ELEMENT card are stored in set Eall. This set will be later referred to in the material definition. Now, each element is listed followed by the 20 node numbers defining it. With *NODE and *ELEMENT, the core of the geometry description is finished. Remaining model definition items are geomtric boundary conditions and the material description.

The only geometric boundary condition in the beam problem is the fixation at z=0. To this end, the nodes at z=0 are collected and stored in node set FIX defined by the keyword card *NSET. The nodes belonging to the set follow on the lines underneath the keyword card. By means of the card *BOUNDARY, the nodes belonging to set FIX are subsequently fixed in 1, 2 and 3-direction, corresponding to x,y and z. The three *BOUNDARY statements in Figure 4 can actually be grouped yielding:

```
*BOUNDARY
FIX,1
FIX,2
FIX,3
```

or even shorter:

```
*BOUNDARY
FIX,1,3
```

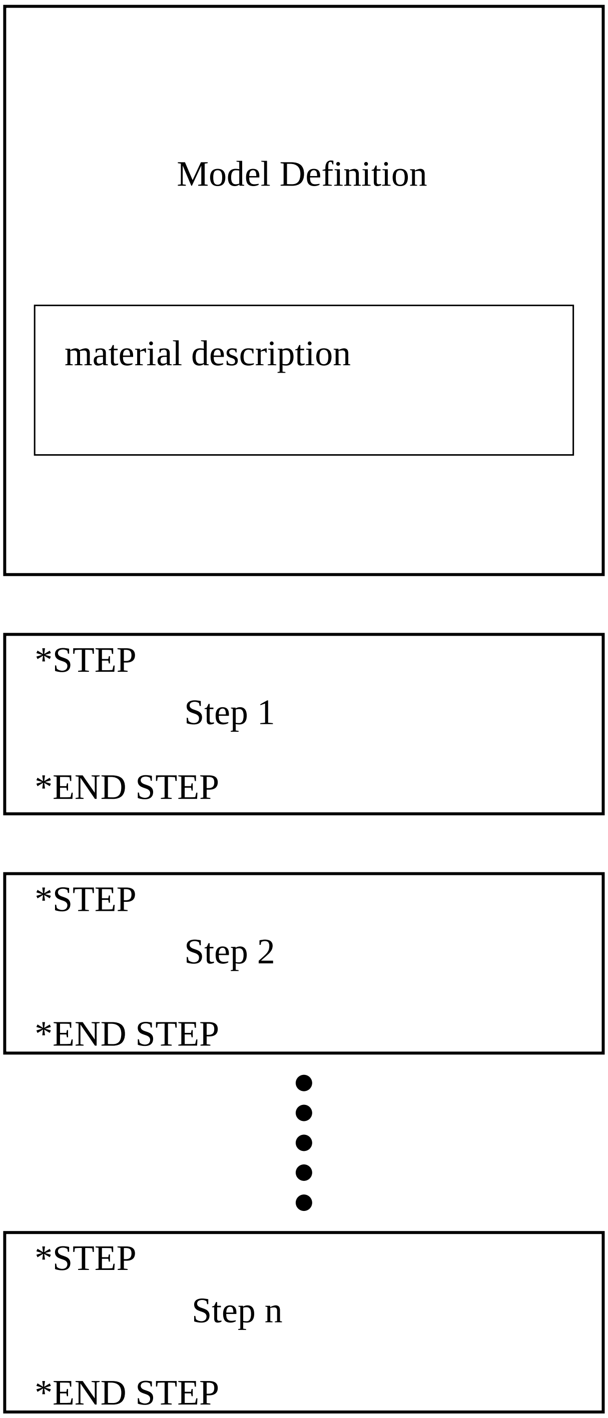

Figure 3: Structure of a CalculiX input deck

```
*HEADING
Model: beam     Date: 10-Mar-1998
*NODE
     1,      0.000000,      0.000000,      0.000000
     2,      1.000000,      0.000000,      0.000000
     3,      1.000000,      1.000000,      0.000000
     .
     .
     .
   260,      0.500000,      0.750000,      7.000000
   261,      0.500000,      0.500000,      7.500000
*ELEMENT, TYPE=C3D20R   , ELSET=Eall
     1,     1,    10,    95,    19,    61,   105,   222,   192,     9,    93,
           94,    20,   104,   220,   221,   193,    62,   103,   219,   190
     2,    10,     2,    13,    95,   105,    34,   134,   222,    11,    12,
           96,    93,   106,   133,   223,   220,   103,    33,   132,   219
     .
     .
     .
     .
    32,   258,   158,    76,   187,   100,    25,     7,    28,   259,   159,
          186,   260,   101,    26,    27,   102,   261,   160,    77,   189
*NSET, NSET=FIX
    97,    96,    95,    94,    93,    20,    19,    18,    17,    16,    15,
    14,    13,    12,    11,    10,     9,     4,     3,     2,     1
*BOUNDARY
FIX, 1
*BOUNDARY
FIX, 2
*BOUNDARY
FIX, 3
*NSET,NSET=Nall,GENERATE
1,261
*MATERIAL,NAME=EL
*ELASTIC
  210000.0,       .3
*SOLID SECTION,ELSET=Eall,MATERIAL=EL
*NSET,NSET=LOAD
5,6,7,8,22,25,28,31,100
**
*STEP
*STATIC
*CLOAD
LOAD,2,1.
*NODE PRINT,NSET=Nall
U
*EL PRINT,ELSET=Eall
S
*NODE FILE
U
*EL FILE
S
*END STEP
```

Figure 4: Beam input deck

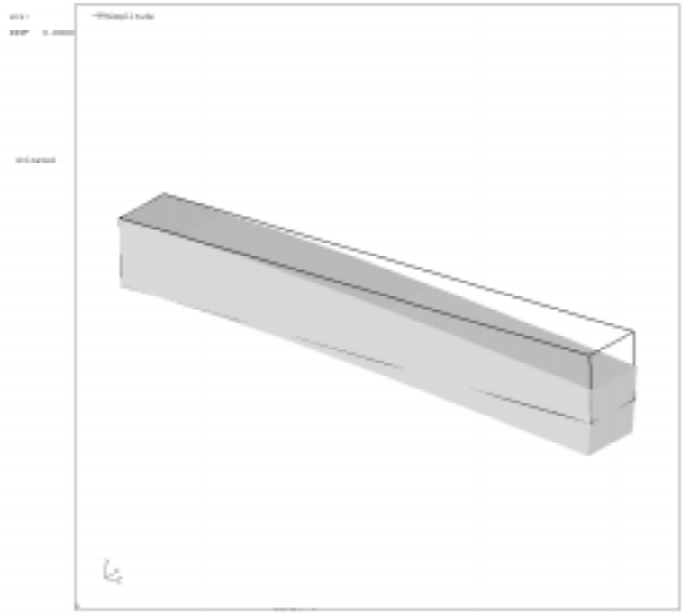

Figure 5: Deformation of the beam

meaning that degrees of freedom 1 through 3 are to be fixed (i.e. set to zero). The geometric boundary conditions here allows us to explain one extremely important rule which it is impossible to violate with impunity, the so-called precedence rule. This rule states that an entity can NEVER be used before its definition. Thus, the set FIX can only be used in the *BOUNDARY definition because it has been defined before in the input deck.

The next section in the input deck is the material description. This section is special since the cards describing one and the same material must be grouped together, although the section itself can occur anywhere before the first *STEP card, provided the precedence rule is not violated. A material section is always started by a *MATERIAL card defining the name of the material by means of the parameter NAME. Depending on the kind of material several keyword cards can follow. Here, the material is linear elastic, characterized by a Young's modulus of 210,000.0 MN/m^2 and a Poisson coefficient of 0.3 (steel). These properties are stored beneath the *ELASTIC keyword card, which here concludes the material definition. Next, the material is assigned to the element set Eall by means of the keyword card *SOLID SECTION.

Finally, the last card in the model definition section defines a node set LOAD which will be needed to define the load. The card starting with two asterisks in between the model definition section and the first step section is a comment line. A comment line can be introduced at any place. It is completely ignored by CalculiX and serves for input deck clarity only.

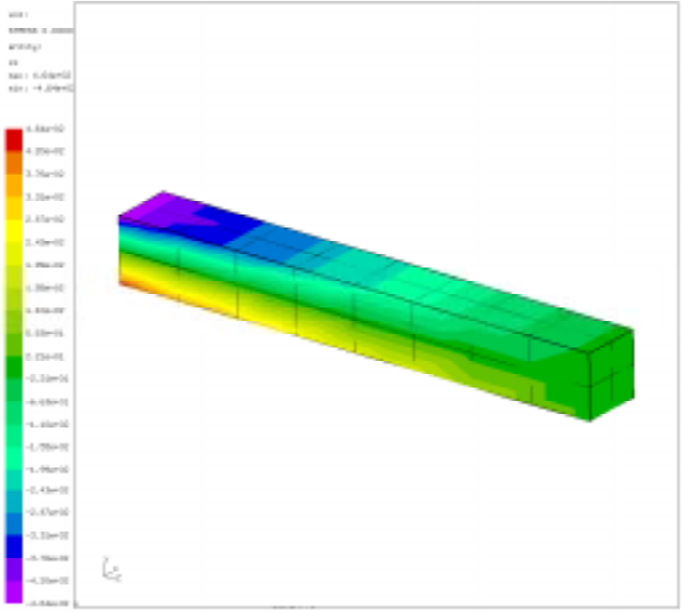

Figure 6: Axial normal stresses in the beam

In the present problem, only one step is needed. A step always starts with a *STEP card and concludes with a *END STEP card. The keyword card *STATIC defines the procedure. The *STATIC card indicates that the load is applied in a quasi-static way, i.e. so slow that mass inertia does not play a role. Other procedures are *FREQUENCY, *BUCKLE, *MODAL DYNAMIC and *DYNAMIC. Next, the concentrated load is applied (keyword *CLOAD) to node set LOAD. The forces act in y-direction and their magnitude is 1, yielding a total load of 9.

Finally, the printing and file storage cards allow for user-directed output generation. The print cards (*NODE PRINT and *EL PRINT) lead to an ASCII file with extension .dat. If they are not selected, no .dat file is generated. The *NODE PRINT and *EL PRINT cards must be followed by the node and element sets for which output is required, respectively. Element information is stored at the integration points.

The *NODE FILE and *EL FILE cards, on the other hand, govern the output written to an ASCII file with extension .frd. The results in this file can be viewed with CalculiX GraphiX (cgx). If no *NODE FILE nor *EL FILE cards are included, the displacements and stresses are stored in the .frd file by default. Else, the explicitly requested fields are stored. Quantities selected by the *NODE FILE and *EL FILE cards are always stored for the complete model. Element quantities are extrapolated to the nodes, and all contributions in the same node are averaged. Selection of fields for the *NODE PRINT, *EL

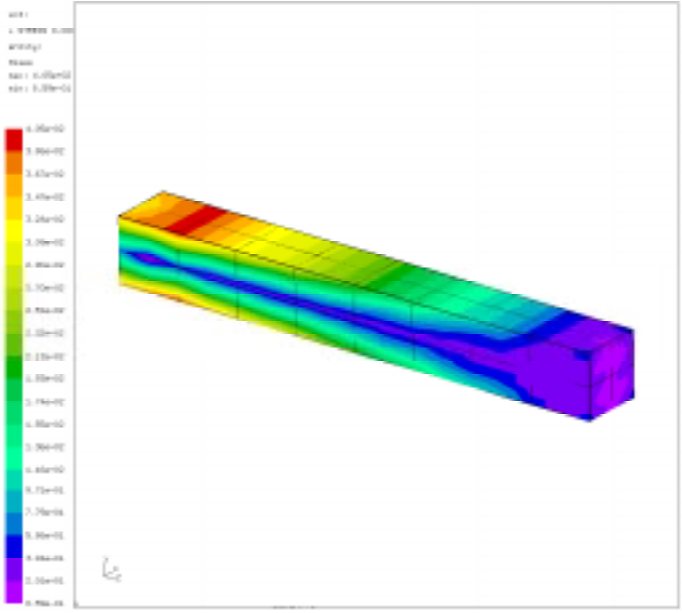

Figure 7: Von Mises stresses in the beam

PRINT, *NODE FILE and *EL FILE cards is made by character codes: for instance, U are the displacements and S are the (Cauchy) stresses.
The input deck is concluded with an *END STEP card.
The output files for the beam problem consist of file beam.dat and beam.frd. The beam.dat file contains the displacements for set Nall and the stresses in the integration points for set Eall. The file beam.frd contains the displacements and extrapolated stresses in all nodes. It is the input for the visualisation program CalculiX GraphiX (cgx). An impression of the capabilities of cgx can be obtained by looking at Figures 5, 6 and 7.
Figure 5 shows the deformation of the beam under the prevailing loads. As expected, the beam bends due to the lateral force at its end. Figure 6 shows the normal stress in axial direction. Due to the bending moment one obtains a nearly linear distribution across the height of the beam. Finally, Figure 7 shows the Von Mises stress in the beam.

3 Theory

The finite element method is basically concerned with the determination of field variables. The most important ones are the stress and strain fields. As basic measure of strain in CalculiX the Lagrangian strain tensor E is used for elastic media, the Eulerian strain tensor e is used for deformation plasticity and the deviatoric elastic left Cauchy-Green tensor is used for incremental plasticity.

The Lagrangian strain satisfies ([8]):

$$E_{KL} = (U_{K,L} + U_{L,K} + U_{M,K}U_{M,L})/2, \quad K, L, M = 1, 2, 3 \tag{1}$$

where U_K are the displacement components in the material frame of reference and repeated indices imply summation over the appropriate range. In a linear analysis, this reduces to the familiar form:

$$E_{KL} = (U_{K,L} + U_{L,K})/2, \quad K, L = 1, 2, 3. \tag{2}$$

The Eulerian strain satisfies ([8]):

$$e_{kl} = (u_{k,l} + u_{l,k} - u_{m,k}u_{m,l})/2, \quad k, l, m = 1, 2, 3 \tag{3}$$

where u_k are the displacements components in the spatial frame of reference. Finally, the deviatoric elastic left Cauchy-Green tensor is defined by ([22]):

$$\bar{b}^e_{kl} = J^{e-2/3} x^e_{k,K} x^e_{l,K} \tag{4}$$

where J^e is the elastic Jacobian and $x^e_{k,K}$ is the elastic deformation gradient. The above formulas apply for Carthesian coordinate systems.

The stress measure consistent with the Lagrangian strain is the second Piola-Kirchhoff stress S. This stress, which is internally used in CalculiX for all applications (the so-called total Lagrangian approach, see [6]), can be transformed into the first Piola-Kirchhoff stress P (the so-called engineering stress, a non-symmetric tensor) and into the Cauchy stress t (true stress). All CalculiX input (e.g. distributed loading) and output is in terms of true stress. In a tensile test on a specimen with length L the three stress measures are related by:

$$t = P/(1 - \epsilon) = S/(1 - \epsilon)^2 \tag{5}$$

where ϵ is the engineering strain defined by

$$\epsilon = dL/L. \tag{6}$$

3.1 Element Types

3.1.1 Eight-node brick element (C3D8)

This is a general purpose linear brick element, fully integrated (2x2x2 integration points). The shape functions can be found in [11]. The node numbering follows the convention of Figure 8 and the integration points are numbered according to Figure 9. This latter information is important since element variables printed with the *EL PRINT keyword are given in the integration points.

Although the structure of the element is straightforward, it should not be used in the following situations:

- due to the full integration, the element will behave badly for isochoric material behavior, i.e. for high values of Poisson's coefficient or plastic behavior.

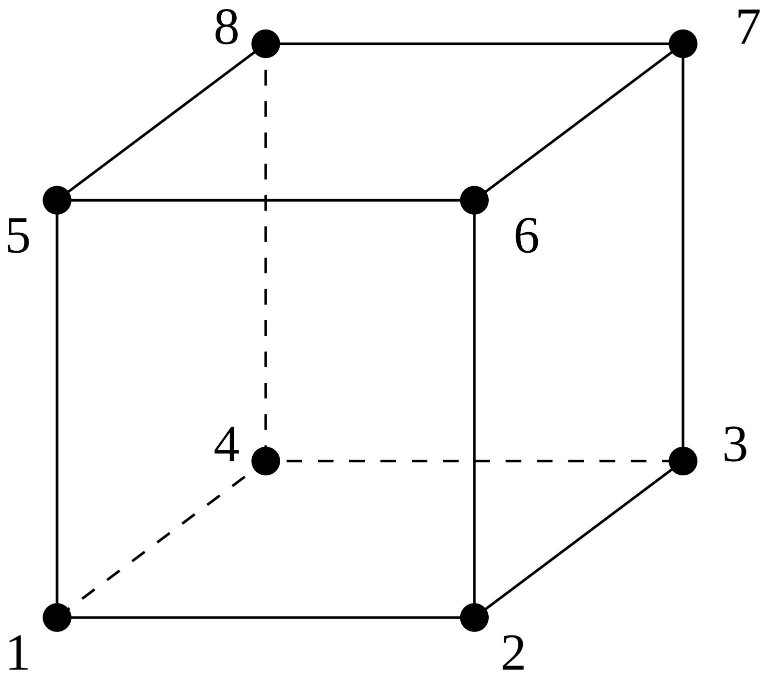

Figure 8: 8-node brick element

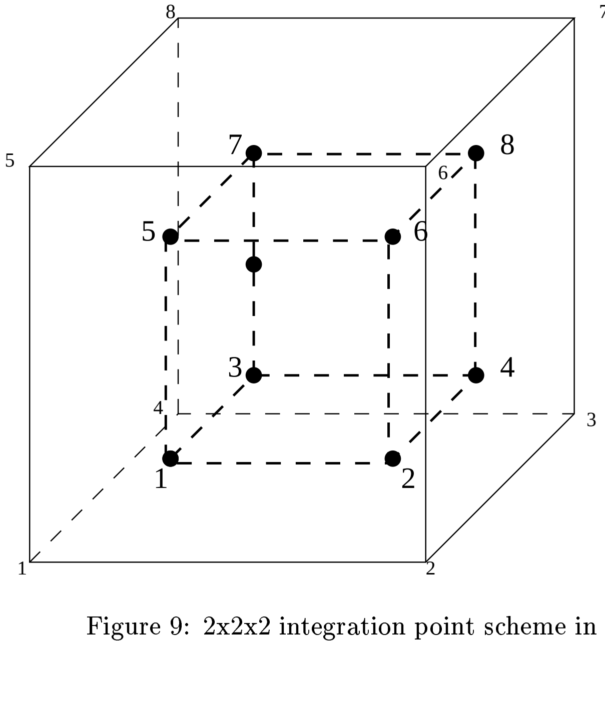

Figure 9: 2x2x2 integration point scheme in hexahedral elements

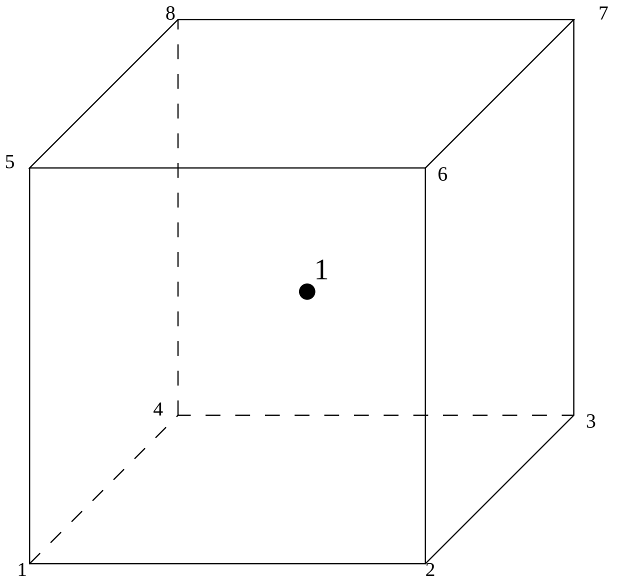

Figure 10: 1x1x1 integration point scheme in hexahedral elements

- the element tends to be too stiff in bending, e.g. for slender beams or thin plates under bending. [27].

3.1.2 Eight-node brick element with reduced integration (C3D8R)

This is a general purpose linear brick element, with reduced integration (1 integration point). The shape functions are the same as for the C3D8 element and can be found in [11]. The node numbering follows the convention of Figure 8 and the integration point is shown in Fig 10.
Due to the reduced integation, the locking phenomena observed in the C3D8 element do not show. However, the element exhibits other shortcomings:

- The element tends to be not stiff enough in bending.

- Stresses, strains.. are most accurate in the integation points. The integration point of the C3D8R element is located in the middle of the element. Thus, small elements are required to capture a stress concentration at the boundary of a structure.

- There are 12 spurious zero energy modes leading to massive hourglassing: this means that the correct solution is superposed by arbitrarily large displacements corresponding to the zero energy modes. Thus, the displacements are completely wrong. Since the zero energy modes do no lead to any stresses, the stress field is still correct. In practice, the C3D8R element is not very useful without hourglass control. So far, no hourglass control has been implemented in CalculiX.

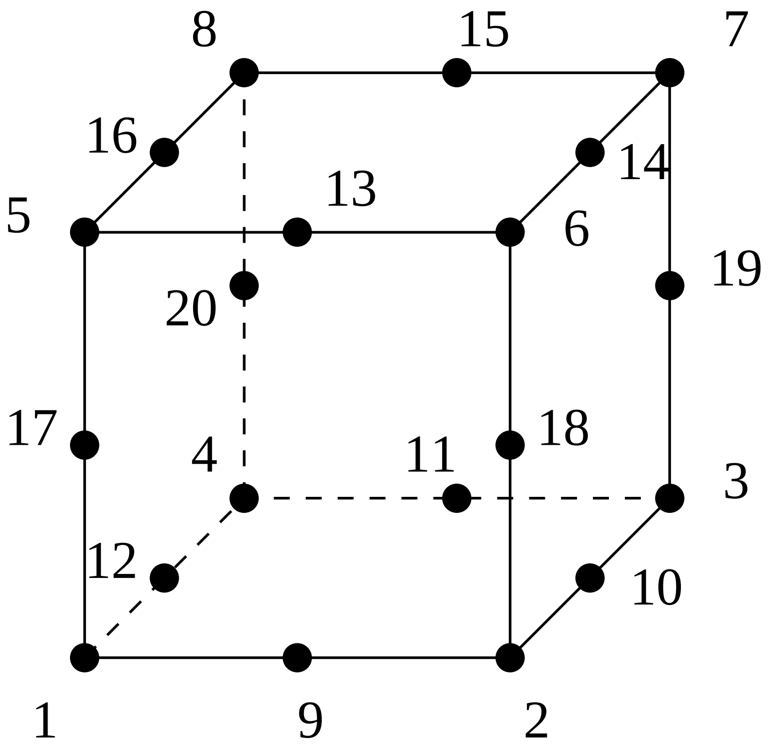

Figure 11: 20-node brick element

3.1.3 Twenty-node brick element (C3D20)

This is a general purpose quadratic brick element (3x3x3 integration points). The shape functions can be found in [11]. The node numbering follows the convention of Figure 11 and the integration scheme is given in Figure 12.
This is an excellent element for linear elastic calculations. Due to the location of the integration points, stress concentrations at the surface of a structure are well captured. However, for nonlinear calculations the elements exhibits the same disadvantages as the C3D8 element, albeit to a much lesser extent:

- due to the full integration, the element will behave badly for isochoric material behavior, i.e. for high values of Poisson's coefficient or plastic behavior.

- the element tends to be too stiff in bending, e.g. for slender beams or thin plates under bending. [27].

3.1.4 Twenty-node brick element with reduced integration (C3D20R)

This is a general purpose quadratic brick element, with reduced integration (2x2x2 integration points). The shape functions can be found in [11]. The node

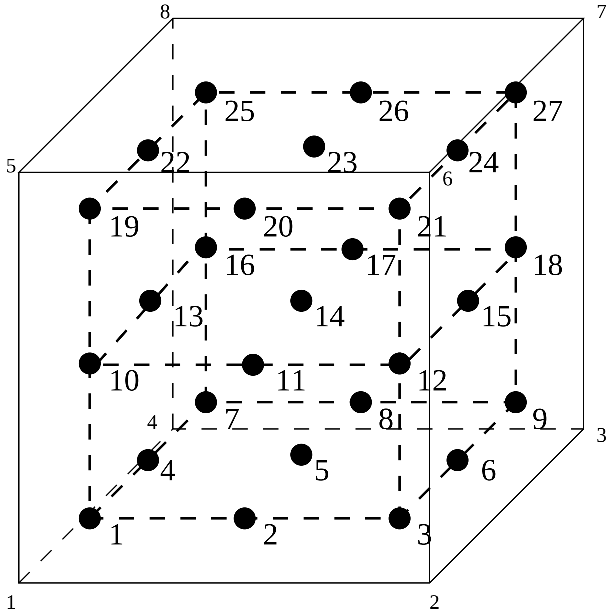

Figure 12: 3x3x3 integration point scheme in hexahedral elements

numbering follows the convention of Figure 11 and the integration scheme is shown in Figure 9.

The element behaves very well and is an excellent general purpose element (if you are setting off for a long journey and you are allowed to take only one element type with you, that's the one to take). It also performs well for isochoric material behavior and in bending and rarely exhibits hourglassing despite the reduced integration (hourglassing generally occurs when not enough integration points are used for numerical integration and spurious modes pop up resulting in crazy displacement fields but correct stress fields). The reduced integration points are so-called superconvergent points of the element [4]. Just two caveats:

- the integration points are about one quarter of the typical element size away from the boundary of the element, and the extrapolation of integration point values to the nodes is trilinear. Thus, high stress concentrations at the surface of a structure might not be captured if the mesh is to coarse.
- all quadratic elements cause problems in contact calculations, because the nodal forces in the vertex nodes equivalent to constant pressure on an element side (section 3.5.2) are zero or have the opposite sign of those in the midside nodes . So far, no contact capabilities were implemented in CalculiX.

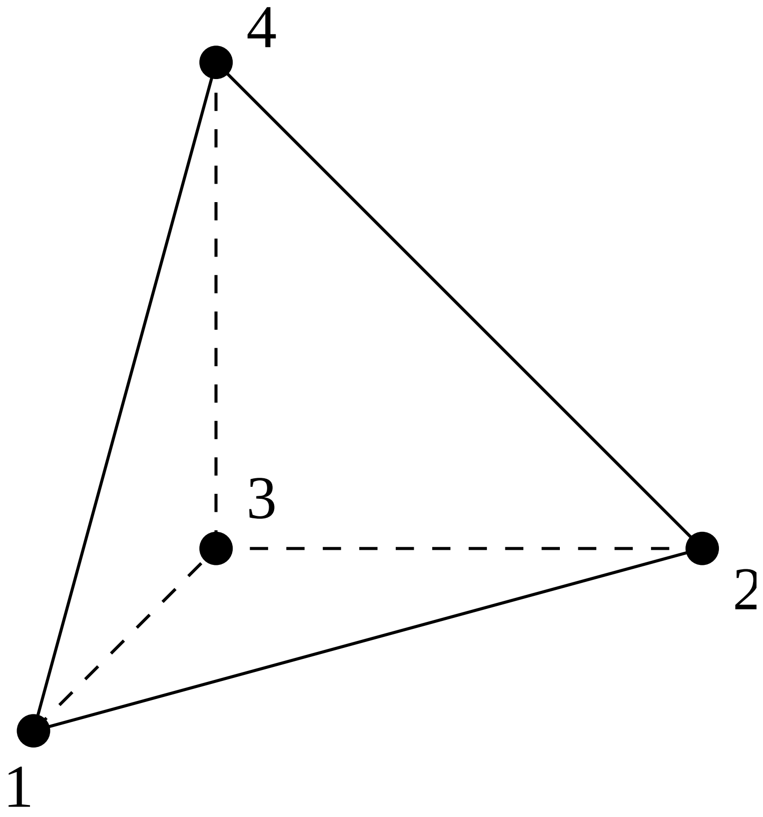

Figure 13: 4-node tetrahedral element

3.1.5 Four-node tetrahedral element (C3D4)

This is a general purpose tetrahedral element (1 integration point). The shape functions can be found in [27]. The node numbering follows the convention of Figure 13.
This element is included for completeness, however, it is not suited for structural calculations unless a lot of them are used (the element is too stiff). Please use the 10-node tetrahedral element instead.

3.1.6 Ten-node tetrahedral element (C3D10)

This is a general purpose tetrahedral element (4 integration points). The shape functions can be found in [27]. The node numbering follows the convention of Figure 14.
The element behaves very well and is a good general purpose element, although the C3D20R element yields still better results for the same number of degrees of freedom. The C3D10 element is especially attractive because of the existence

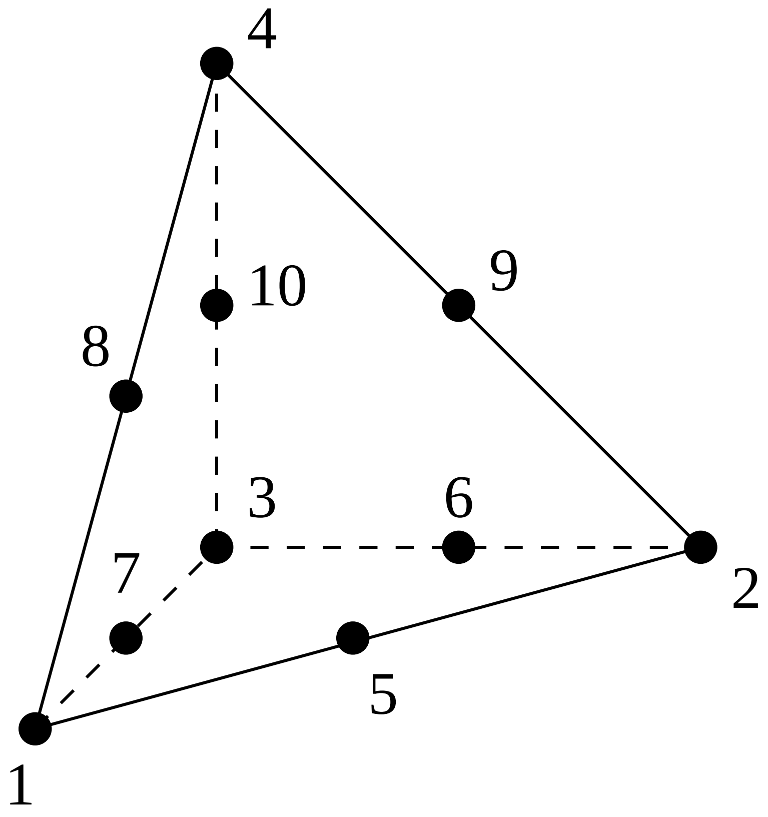

Figure 14: 10-node tetrahedral element

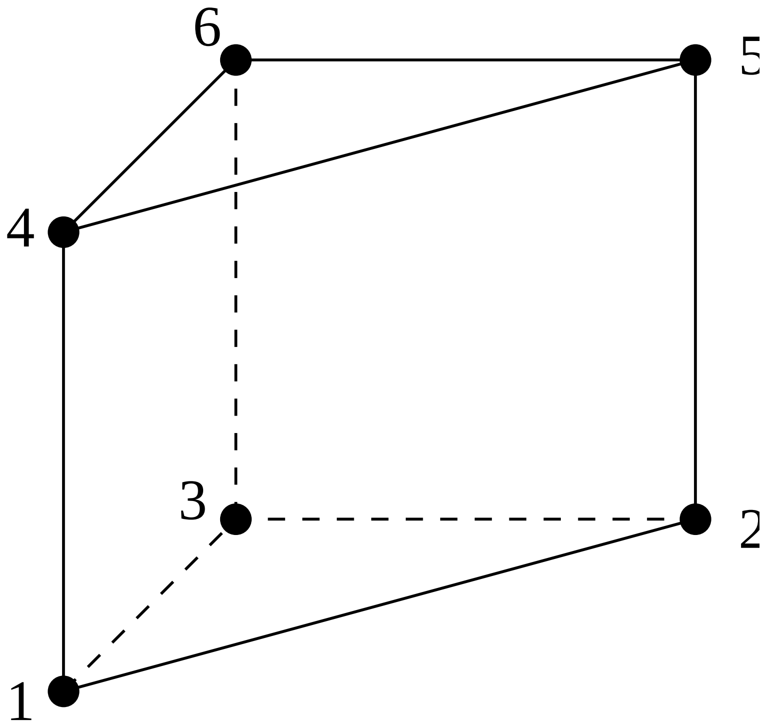

Figure 15: 6-node wedge element

of fully automatic tetrahedral meshers.

3.1.7 Six-node wedge element (C3D6)

This is a general purpose wedge element (2 integration points). The shape functions can be found in [1]. The node numbering follows the convention of Figure 15.
This element is included for completeness, however, it is probably not very well suited for structural calculations unless a lot of them are used. Please use the 15-node wedge element instead.

3.1.8 Fifteen-node wedge element (C3D15)

This is a general purpose wedge element (9 integration points). The shape functions can be found in [1]. The node numbering follows the convention of Figure 16.
The element behaves very well and is a good general purpose element, although the C3D20R element yields still better results for the same number of degrees of freedom. The wedge element is often used as fill element in “automatic”

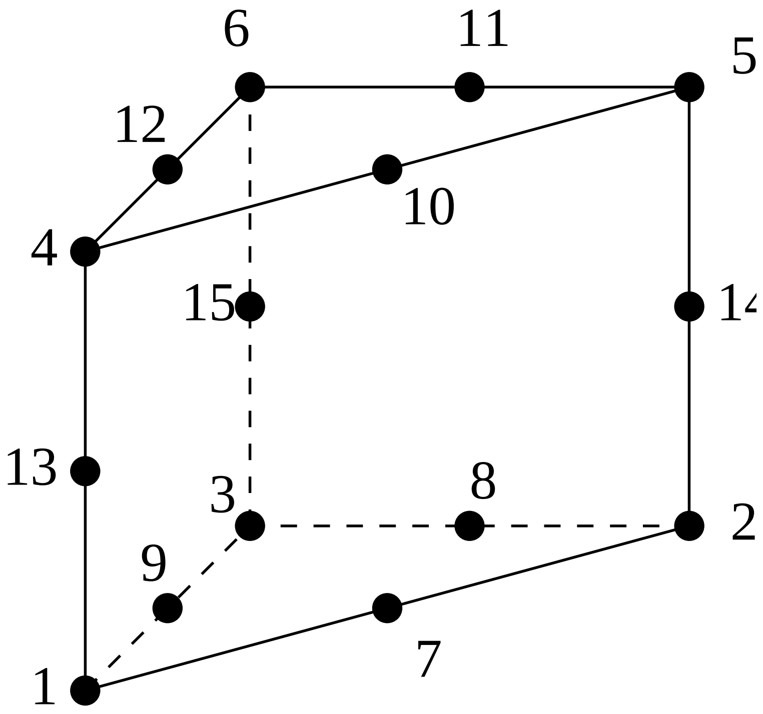

Figure 16: 15-node wedge element

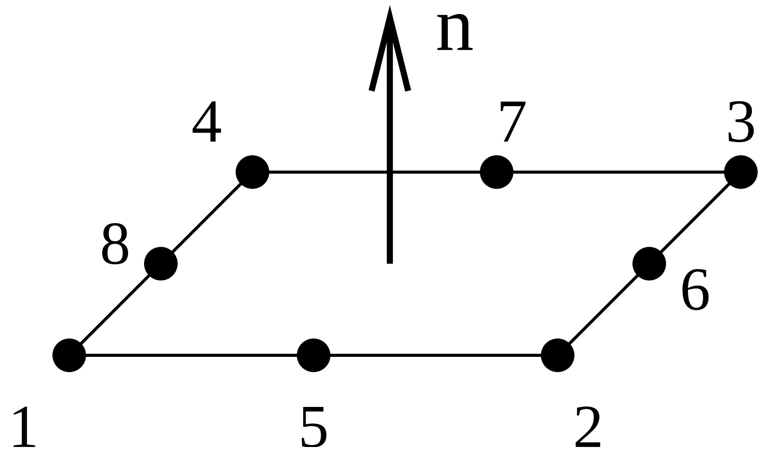

Figure 17: 8-node quadratic element

hexahedral meshers.

3.1.9 Eight node shell element (S8 and S8R)

This element is a general purpose quadratic shell element. The node numbering and the direction of the normal to the surface is shown in Figure 17.
In CalculiX, shell elements are automatically expanded into 20-node brick elements. The way this is done is illustrated in Figure 18. For each shell node three new nodes are generated according to the scheme on the right of Figure 18. With these nodes a new 20-node brick element is generated: for a S8 element a C3D20 element, for a S8R element a C3D20R element.
Since a shell element can be curved, the normal to the shell surface is defined in each node separately. For this purpose the *NORMAL keyword card can be used. If no normal is defined by the user, it will be calculated automatically by CalculiX based on the local geometry.
If a node belongs to more than one shell element, all, some or none of the normals on these elements in the node at stake might have been defined by the user (by means of *NORMAL). The failing normals are determined based on the local geometry. The number of normals is subsequently reduced using the following procedure. First, the element with the lowest element number with an explicitly defined normal in this set, if any, is taken and used as reference. Its normal is defined as reference normal and the element is stored in a new subset. All other elements of the same type in the set for which the normal has an angle smaller than 0.5° with the reference normal and which have the same local thickness and offset are also included in this subset. The elements in the subset are considered to have the same normal, which is defined as the normed mean of all normals in the subset. This procedure is repeated for the elements in the set minus the subset until no elements are left with an explicitly defined

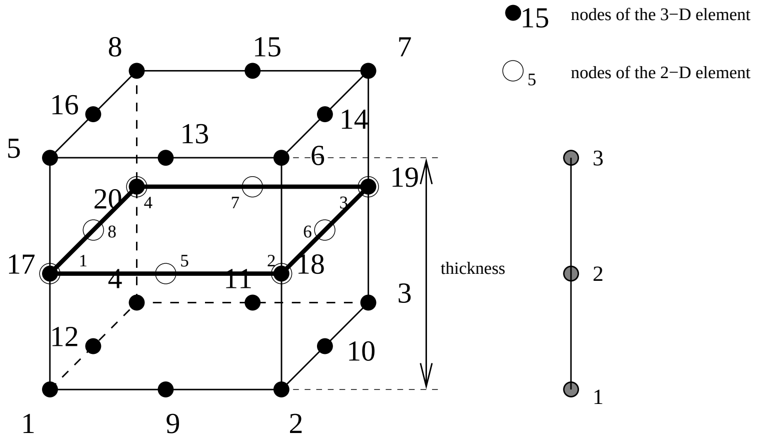

Figure 18: Expansion of a 2-D 8-node element into a 3-D element

normal. Now, the element with the lowest element number of all elements left in the set is used as reference. Its normal is defined as reference normal and the element is stored in a new subset. All other elements left in the set for which the normal has an angle smaller than 20° with the reference normal and which have the same local thickness and offset are also included in this subset. The normed mean of all normals in the subset is assigned as new normal to all elements in the subset. This procedure is repeated for the elements left until a normal has been defined in each element.
This procedure leads to one or more normals in one and the same node. If only one normal is defined, this node is expanded once into a set of three new nodes and the resulting three-dimensional expansion is continuous in the node. If more than one normal is defined, the node is expanded as many times as there are normals in the node. To assure that the resulting 3-D elements are connected, the newly generated nodes are considered as a rigid body knot. Graphically, the shell elements partially overlap (Figure 19).
Consequently, a node leads to a knot if

- the direction of the local normals in the elements participating in the node differ beyond a given amount. Notice that this also applies to neighboring elements having the inverse normal. Care should be taken that the elements in plates and similar structures are oriented in a consistent way to avoid the generation of knots and the induced nonlinearity.
- several types of elements participate (e.g. shells and beams).
- the thickness is not the same in all participating elements.
- the offset is not the same in all participating elements.

In addition, a knot is also generated if

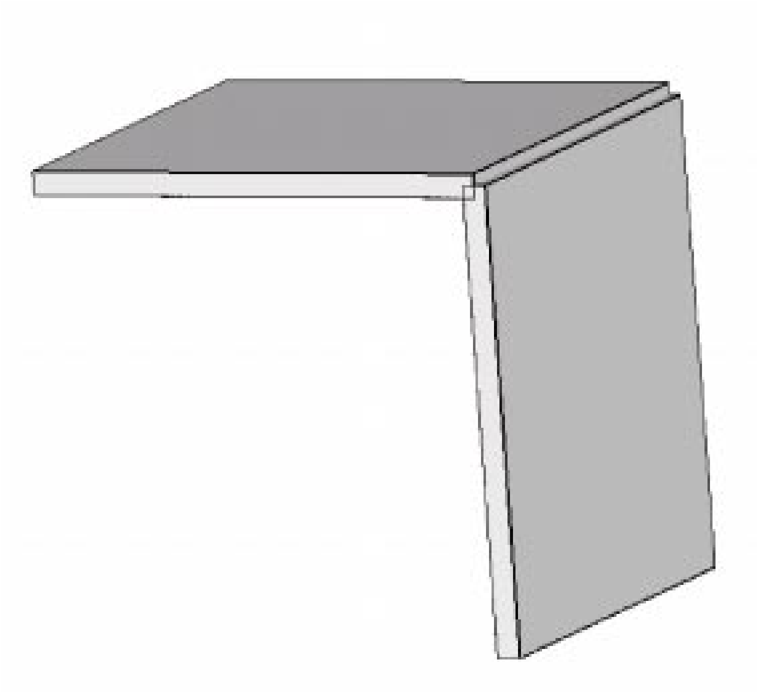

Figure 19: Overlapping shell elements at a rigid body knot

- a rotational degree of freedom in the node is constrained through a SPC or MPC. In this case, the rigid body knot introduces the necessary rotational degrees of freedom.
- a bending moment or torque is defined in the nodes. Here too, the rigid body introduces the necessary rotational degrees of freedom.

Beam and shell elements are always connected in a stiff way if they share common nodes. This, however, does not apply to plane stress, plane strain and axisymmetric elements. Although any mixture of 1-D and 2-D elements generates a knot, the knot is modeled as a hinge for any plane stress, plane strain or axisymmetric elements involved in the knot. This is necessary to account for the special nature of these elements (the displacement normal to the symmetry plane and normal to the radial planes is zero for plane elements and axisymmetric elements, respectively).

The translational node of the rigid body knot (cfr REF NODE in the *RIGID BODY keyword card) is the knot generating node, the rotational node is extra generated.

The thickness of the shell element can be defined on the *SHELL SECTION keyword card. It applies to the complete element. Alternatively, a nodal thickness in each node separately can be defined using *NODAL THICKNESS. In that way, a shell with variable thickness can be modeled. Thicknesses defined by a *NODAL THICKNESS card take precedence over thicknesses defined by a *SHELL SECTION card. The thickness always applies in normal direction. The *SHELL SECTION card is also used to assign a material to the shell elements and is therefore indispensible.

The offset of a shell element can be set on the *SHELL SECTION card. Default is zero. The unit of the offset is the local shell thickness. An offset of 0.5 means that the user-defined shell reference surface is in reality the top surface of the expanded element. The offset can take any real value. Consequently, it can be used to define composite materials. Defining three different shell elements using exactly the same nodes but with offsets -1,0 and 1 (assuming the thickness is the same) leads to a three-layer composite.

The treatment of the boundary conditions for shell elements is straightforward. The user can independently fix any translational degree of freedom (DOF 1 through 3) or any rotational DOF (DOF 4 through 6). Here, DOF 4 is the rotation about the global x-axis, DOF 5 about the global y-axis and DOF 6 about the global z-axis. No local coordinate system should be defined in nodes with constrained rotational degrees of freedom. A hinge is defined by fixing the translational degrees of freedom only.

For an internal hinge between 1-D or 2-D elements the nodes must be doubled and connected with MPC's. The connection between 3-D elements and all other elements (1-D or 2-D) is always hinged.

Point forces defined in a shell node are modified if a knot is generated (the reference node of the rigid body is the shell node). If no knot is generated, the point load is divided among the expanded nodes according to a 1/2-1/2 ratio

for a shell midnode and a 1/6-2/3-1/6 ratio for a shell endnode. Concentrated bending moments or torques are defined as point loads (*CLOAD) acting on degree four to six in the node. Their use generates a knot in the node.
Distributed loading can be defined by the label P in the *DLOAD card. A positive value corresponds to a pressure load in normal direction.
In addition to a temperature for the reference surface of the shell, a temperature gradient in normal direction can be specified on the *TEMPERATURE card. Default is zero.
Finally, in thin structures a word of caution is due with respect to reduced integration. Due to the small thickness hourglassing can readily occur, especially if point loads are applied. In that case, full integration might be necessary.

3.1.10 Eight node plane stress element (CPS8 and CPS8R)

The eight node plane stress element is a general purpose plane stress element. It is actually a special case of shell element: the structure is assumed to have a symmetry plane parallel to the x-y plane and the loading only acts in-plane. In general, the z-coordinates are zero. Just like in the case of the shell element, the plane stress element is expanded into a C3D20 or C3D20R element. Figures 17 and 18 apply. From the above premises the following conclusions can be drawn:

- The displacement in z-direction of the midplane is zero. This condition is introduced in the form of SPC's. MPC's must not be defined in z-direction!
- The displacements perpendicular to the z-direction of nodes not in the midplane is identical to the displacements of the corresponding nodes in the midplane.
- The normal is by default (0,0,1)
- The thickness can vary. It can be defined in the same way as for the shell element, except that the *SOLID SECTION card is used instead of the *SHELL SECTION card.
- Different offsets do not make sense.
- Point loads are treated in a similar way as for shells.

The use of plane stress elements can also lead to rigid body knots, namely, if the thickness varies in a discontinuous way, or if plane stress elements are combined with other 1D or 2D elements such as axisymmetric elements. The connection with the plane stress elements, however, is modeled as a hinge.
Distributed loading in plane stress elements is different from shell distributed loading: for the plane stress element it is in-plane, for the shell element it is out-of-plane. Distributed loading in plane stress elements is defined on the *DLOAD card with the labels P1 up to P4. The number indicates the face as defined in Figure 20.

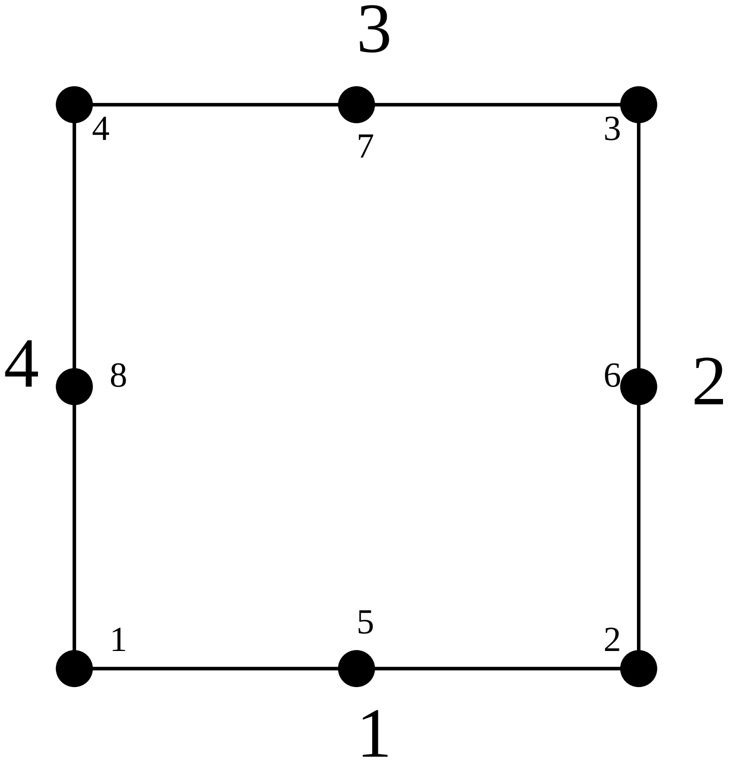

Figure 20: Face numbering for quadrilateral elements

3.1.11 Eight node plane strain element (CPE8 and CPE8R)

The eight node plane strain element is a general purpose plane strain element. It is actually a special case of plane stress element: the treatise of the previous section also applies here. In addition we have:

- The displacement in z-direction of all nodes (not only the midnodes) is zero. This condition is introduced in the form of MPC's, expressing that the displacement in z-direction of nodes not in the midplane is identical to the displacement of the corresponding nodes in the midplane.

- Different thicknesses do not make sense: one thickness applicable to all plane strain elements suffices.

Plane strain elements are used to model a slice of a very long structure, e.g. of a dam.

3.1.12 Eight node axisymmetric element (CAX8 and CAX8R)

This is a general purpose quadratic axisymmetric element. Just as the shell, plane stress and plane strain element it is internally expanded into a C3D20 or C3D20R element according to Figure 18 and the node numbering of Figure 17 applies.

For axisymmetric elements the coordinates of the nodes correspond to the radial direction (first coordinate) and the axial direction (second or y-coordinate). The axisymmetric structure is expanded by rotation about the second coordinate axis, half clockwise and half counterclockwise. The radial direction corresponds to the x-axis in the 3-D expansion, the axial direction with the y-axis. The x-y plane cuts the expanded structure in half. The z-axis is perpendicular to the x-y plane such that a right-hand-side axis system is obtained.

The same rules apply as for the plane strain elements, except that in-plane conditions in a plane strain construction now correspond to radial plane conditions in the axisymmetric structure. Expressed in another way, the z-direction in plane strain now corresponds to the circumferential direction in a cylindrical coordinate system with the y-axis as defining axis.

Compared to plane strain elements, the following conditions apply:

- The expansion angle can be defined on the *SOLID SECTION card. Only one angle for the complete structure makes sense.

- The displacements in cylindrical coordinates of all nodes not in the defining plane are identical to the displacements of the corresponding nodes in the defining plane. This is formulated using MPC's.

- Forces act in radial planes.

- For distributed loading Figure 20 applies.

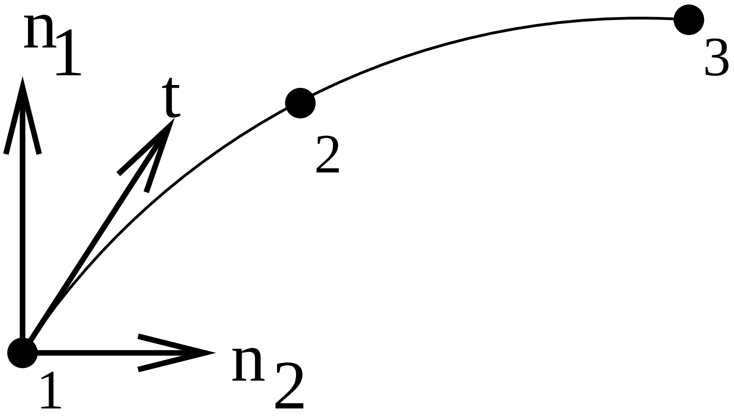

Figure 21: 3-node quadratic beam element

3.1.13 Three node beam element (B32 and B32R)

In CalculiX this is the general purpose beam element. The node numbering is shown in Figure 21.

In each node a local Carthesian system $\mathbf{t}-\mathbf{n_1}-\mathbf{n_2}$ is defined. $\mathbf{t}$ is the normalized local tangential vector, $\mathbf{n_1}$ is a normalized vector in the local 1-direction and $\mathbf{n_2}$ is a normalized vector in the local 2-direction, also called the normal. The local directions 1 and 2 are used to expand the beam element into a C3D20 or C3D20R element according to Figure 22.

For each node of the beam element 8 new nodes are generated according to the scheme on the right of Figure 22. These new nodes are used in the definition of the brick element, and their position is defined by the local directions together with the thickness and offset in these directions.

The tangential direction follows from the geometry of the beam element. The normal direction (2-direction) can be defined in two ways:

- either by defining the normal explicitly by using the *NORMAL keyword card.
- if the normal is not defined by the *NORMAL card, it is defined implicitly by $\mathbf{n_2} = \mathbf{t} \times \mathbf{n_1}$

In the latter case, $\mathbf{n_1}$ can be defined either

- explicitly on the *BEAM SECTION card.
- implicitly through the default of (0,0,-1).

If a node belongs to more than one beam element, the tangent and the normal is first calculated for all elements to which the node belongs. Then, the element with the lowest element number in this set for which the normal was defined

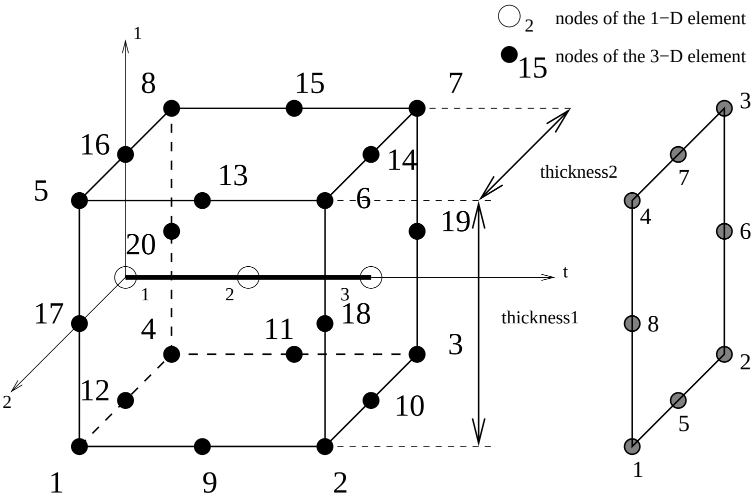

Figure 22: Expansion of a beam element

explicitly using a *NORMAL card is used as reference. Its normal and tangent are defined as reference normal and reference tangent and the element is stored in a new subset. All other elements of the same type in the set for which the normal and tangent have an angle smaller than 0.5° with the reference normal and tangent and which have the same local thicknesses, offsets and sections are also included in this subset. All elements in the subset are considered to have the same normal and tangent. The normal is defined as the normed mean of all normals in the subset, the same applies to the tangent. Finally, the normal is slightly modified within the tangent-normal plane such that it is normal to the tangent. This procedure is repeated until no elements are left with an explicitly defined normal. Then, the element with the lowest element number left in the set is used as reference. Its normal and tangent are defined as reference normal and reference tangent and the element is stored in a new subset. All other elements of the same type in the set for which the normal and tangent have an angle smaller than 20° with the reference normal and tangent and which have the same local thicknesses, offsets and sections are also included in this subset. All elements in the subset are considered to have the same normal and tangent. This normal is defined as the normed mean of all normals in the subset, the same applies to the tangent. Finally, the normal is slightly modified within the tangent-normal plane such that it is normal to the tangent. This procedure is repeated until a normal and tangent have been defined in each element. Finally, the 1-direction is defined by $\mathbf{n_1} = \mathbf{n_2} \times \mathbf{t}$.

If this procedure leads to more than one local coordinate system in one and the same node, all expanded nodes are considered to behave as a rigid body knot with the generating node as reference node. Graphically, the beam elements partially overlap (Figure 23).

Consequently, a node leads to a knot if

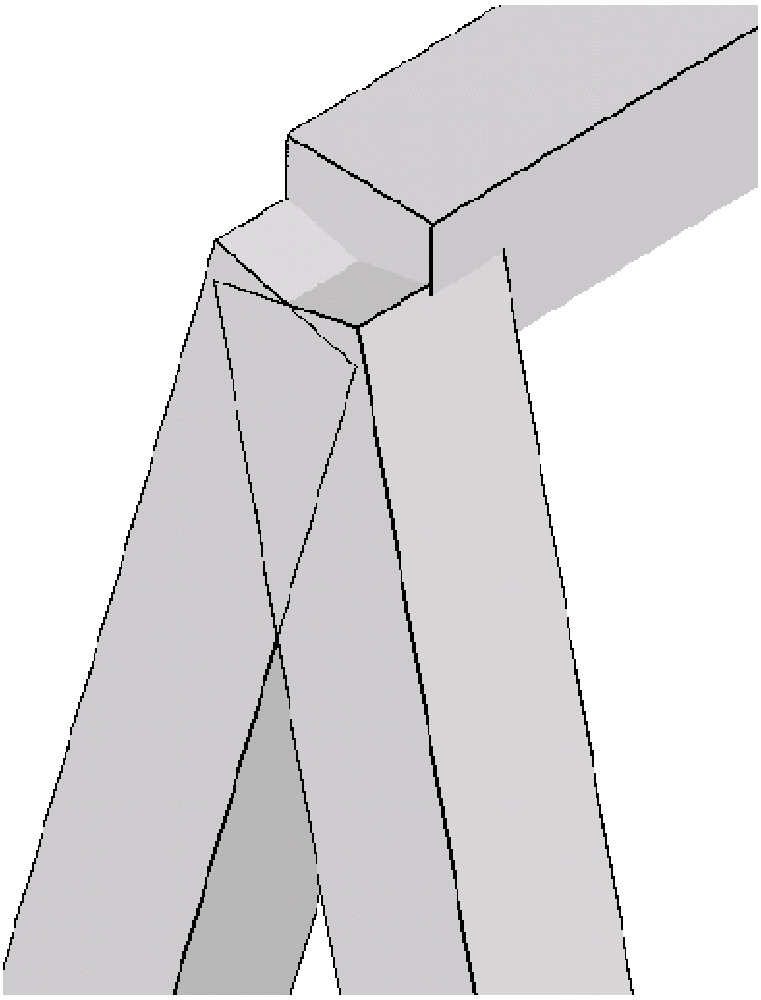

Figure 23: Overlapping beam elements at a rigid body knot

- the direction of the local normals in the elements participating in the node differ beyond a given amount. Notice that this also applies to neighboring elements having the inverse normal. Care should be taken that the elements in beams are oriented in a consistent way to avoid the generation of knots.
- several types of elements participate (e.g. shells and beams).
- the thickness is not the same in all participating elements.
- the offset is not the same in all participating elements.
- the section is not the same in all participating elements.

In addition, a knot is also generated if

- a rotational degree of freedom in the node is constrained through a SPC or MPC. In this case, the rigid body knot introduces the necessary rotational degrees of freedom.
- a bending moment or torque is defined in the nodes. Here too, the rigid body introduces the necessary rotational degrees of freedom.

Beam and shell elements are always connected in a stiff way if they share common nodes. This, however, does not apply to plane stress, plane strain and axisymmetric elements. Although any mixture of 1-D and 2-D elements generates a knot, the knot is modeled as a hinge for any plane stress, plane strain or axisymmetric elements involved in the knot. This is necessary to account for the special nature of these elements (the displacement normal to the symmetry plane and normal to the radial planes is zero for plane elements and axisymmetric elements, respectively).

The section of the beam must be specified on the *BEAM SECTION keyword card. It can be rectangular (SECTION=RECT) or elliptical (SECTION=CIRC). A circular cross section is a special case of elliptical section. For a rectangular cross section the local axes must be defined parallel to the sides of the section, for an elliptical section they are parallel to the minor and major axes of the section. The thickness of a section is the distance between the free surfaces, i.e. for a circular section it is the diameter.

The thicknesses of the beam element (in 1- and 2-direction) can be defined on the *BEAM SECTION keyword card. It applies to the complete element. Alternatively, the nodal thicknesses can be defined in each node separately using *NODAL THICKNESS. That way, a beam with variable thickness can be modeled. Thicknesses defined by a *NODAL THICKNESS card take precedence over thicknesses defined by a *BEAM SECTION card.

The offsets of a beam element (in 1- and 2-direction) can be set on the *BEAM SECTION card. Default is zero. The unit of the offset is the beam thickness in the appropriate direction. An offset of 0.5 means that the user-defined beam reference line lies in reality on the positive surface of the expanded beam (i.e.

the surface with an external normal in direction of the local axis). The offset can take any real value. Consequently, it can be used to define composite structures, such as a plate supported by a beam, or a I cross section built up of rectangular cross sections.
The treatment of the boundary conditions for beam elements is straightforward. The user can independently fix any translational degree of freedom (DOF 1 through 3) or any rotational DOF (DOF 4 through 6). Here, DOF 4 is the rotation about the global x-axis, DOF 5 about the global y-axis and DOF 6 about the global z-axis. No local coordinate system should be defined in nodes with constrained rotational degrees of freedom. A hinge is defined by fixing the translational degrees of freedom only.
For an internal hinge between 1-D or 2-D elements the nodes must be doubled and connected with MPC's. The connection between 3-D elements and all other elements (1-D or 2-D) is always hinged.
Point forces defined in a beam node are not modified if a knot is generated (the reference node is the beam node). If no knot is generated, the point load is divided among the expanded nodes according to a 1/4-1/4-1/4-1/4 ratio for a beam midnode and a (-1/12)-(1/3)-(-1/12)-(1/3)-(-1/12)-(1/3)-(-1/12)-(1/3) ratio for a beam endnode. Concentrated bending moments or torques are defined as point loads (*CLOAD) acting on degree four to six in the node. Their use generates a knot in the node.
Distributed loading can be defined by the labels P1 and P2 in the *DLOAD card. A positive value corresponds to a pressure load in direction 1 and 2, respectively.
In addition to a temperature for the reference surface of the beam, a temperature gradient in 1-direction and in 2-direction can be specified on the *TEMPERATURE. Default is zero.

3.2 Boundary conditions

3.2.1 Single point constraints (SPC)

In a single point constraint one or more degrees of freedom are fixed for a given node. The prescribed value can be zero or nonzero. Nonzero SPC's cannot be defined outside a step. Zero SPC's can be defined inside or outside a step. SPC's are defined with the keyword *BOUNDARY.

3.2.2 Multiple point constraints (MPC)

Multiple point constraints establish a relationship between degrees of freedom in one or more nodes. In CalculiX only homogeneous linear relationships are considered. MPC's must be defined with the keyword *EQUATION before the first step. An inhomogeneous linear relationship can be defined by assigning the inhomogeneous term to one of the degrees of freedom (DOF) of a dummy node (using a SPC) and including this DOF in the MPC, thus homogenizing it.

3.3 Materials

A material definition starts with a *MATERIAL key card followed by material specific cards such as *ELASTIC, *EXPANSION, *DENSITY, *HYPERELASTIC, *HYPERFOAM, *DEFORMATION PLASTICITY, *PLASTIC, *CREEP or *USER MATERIAL. To assign a material to an element, the *SOLID SECTION card is used. An element can consist of one material only. Each element in the structure must have a material assigned. Some types of loading require specific material properties: gravity loading requires the density of the material, temperature loading requires the thermal expansion coefficient. A material property can also be required by the type of analysis: a frequency analysis requires the material's density.
Some of the material cards are mutually exclusive, while others are interdependent. Exactly one of the following is required: *ELASTIC, *HYPERELASTIC, *HYPERFOAM, *DEFORMATION PLASTICITY and *USER MATERIAL. The keyword *PLASTIC must be preceded by *ELASTIC(,TYPE=ISO). The same applies to the *CREEP card. A *PLASTIC card in between the *ELASTIC and *CREEP card defines a viscoplastic material. The other keywords can be used according to your needs.
If any of the materials defined in the input deck is not a linear elastic material, geometric nonlinearities are automatically taken into account (i.e. NLGEOM is activated).

3.3.1 Linear elastic materials

Linear elastic materials are characterized by an elastic potential of which only the quadratic terms in the strain are kept. It can be defined in a isotropic, orthotropic or fully anisotropic version. Isotropic linear elastic materials are characterized by their Young's modulus and Poisson's coefficient. Common steels are usually isotropic. Orthotropic materials, such as wood or cubic single crystals are characterized by 9 nonzero constants and fully anisotropic materials by 21 constants. For elastic materials the keyword *ELASTIC is used.

3.3.2 Hyperelastic and hyperfoam materials

Hyperelastic materials are materials for which a potential function exists such that the second Piola-Kirchhoff stress tensor can be written as the derivative of this potential with respect to the Lagrangian strain tensor. This definition includes linear elastic materials, although the term hyperelastic material is usually reserved for proper nonlinear elastic materials. One important class constitutes the isotropic hyperelastic materials, for which the potential function is a function of the strain invariants only. All rubber material models presently included in CalculiX are of that type (Arruda-Boyce, Mooney-Rivlin, Neo Hooke, Ogden, Polynomial, Reduced Polynomial and Yeoh). They are selected by the keyword *HYPERELASTIC. Rubber materials are virtually incompressible (virtually no dependence on the third Lagrangian strain invariant which takes values close to 1). The dependence on the third invariant (the compressibility) is separated

from the dependence on the first two invariants and is governed by so called compressibility coefficients, taking the value 0 for perfectly incompressible materials. Perfectly incompressible materials require the use of hybrid finite elements, in which the pressure is taken as an additional independent variable (in addition to the displacements). CalculiX does not provide such elements. Consequently, a slight amount of compressibility is required for CalculiX to work. If the user inserts zero compressibility coefficients, CalculiX uses a default value corresponding to an initial value of the Poisson coefficient of 0.475.
Another example of isotropic hyperelastic materials are the hyperfoam materials, which are also implemented in CalculiX (activated by the keyword *HYPERFOAM). Hyperfoam materials are very compressible.
Other materials frequently simulated by a hyperelastic model are human tissue (lung tissue, heart tissue..). To simulate these classes of materials anisotropic hyperelastic models are used, in which the potential function depends on the Lagrangian strain tensor components. No such models are implemented in CalculiX, although their inclusion is not difficult to manage. For further information the reader is referred to [5]. A very nice treatment of the large deformation theory for hyperelastic materials is given in [23].

3.3.3 Deformation plasticity

Deformation plasticity is characterized by a one-to-one (bijective) relationship between the strain and the stress. This relationship is a three-dimensional generalisation of the one-dimensional Ramberg-Osgood law frequently used for metallic materials (e.g. in the simple tension test) yielding a monotonic increasing function of the stress as a function of the strain. Therefore, deformation plasticity is very well suited to model the relation between the Cauchy (true) stress and the strain. Because tensile and compressive test results coincide well when plotting the Cauchy stress versus the logarithmic strain (soon to be defined), these quantities are generally used in the deformation plasticity law. The implementation in CalculiX (keyword card *DEFORMATION PLASTICITY), however, uses the relationship to model the dependence of the Cauchy (true) stress on the Eulerian strain. For all practical purposes, the Eulerian strain coincides with the logarithmic strain. For a tensile test specimen, with initial length L, initial cross section A_0, final length $L + \Delta L$ and final cross section A, loaded by a force F, the Cauchy stress σ, the logarithmic strain ϵ_{log} and the Eulerian strain ϵ_{Euler} satisfy:

$$\sigma = F/A = \frac{F(L+\Delta L)}{A_0 L} \tag{7}$$

$$\epsilon_{log} = ln\left[1 + \frac{\Delta L}{L}\right] \tag{8}$$

$$\epsilon_{Euler} = \frac{\Delta L}{L}\left[1 - \frac{\Delta L}{2L}\right] \tag{9}$$

The difference between the logarithmic strain and the Eulerian strain is about 1.3 % for an Engineering strain $\Delta L/L = 20\%$. The user should give the Ramberg-Osgood material constants directly (by plotting a Cauchy stress versus Eulerian strain curve and performing a fit).

3.3.4 Incremental (visco)plasticity

The implementation of incremental plasticity in CalculiX follows the algorithms in [24] and [25] and is based on the notion of an intermediate stress-free configuration. The deformation is viewed as a plastic flow due to dislocation motion followed by elastic stretching and rotation of the crystal lattice. This is synthesized by a local multiplicative decomposition of the deformation gradient $\mathbf{F} = \mathbf{F}^e\mathbf{F}^p$ where $F_{kK} = x_{k,K}$ in Carthesian coordinates.

In the present implementation, the elastic response is isotropic and is deduced from a stored-energy function (hyperelastic response). Furthermore, the plastic flow is isochoric (the volume is conserved) and the classical von Mises-Huber yield condition applies. This condition can be visualized as a sphere in principal deviatoric stress space.

The hardening can consist of isotropic hardening, resulting in an expansion or contraction of the yield surface, of kinematic hardening, resulting in a translation of the yield surface, or of a combination of both. The hardening curve should yield the von Mises true stress versus the equivalent plastic logarithmic strain (cf. deformation plasticity for its definition).

Incremental plasticity is defined by the *PLASTIC card, followed by the isotropic hardening curve for isotropic hardening or the kinematic hardening curve for kinematic and combined hardening. For combined hardening, the isotropic hardening curve is defined by the *CYCLIC HARDENING card. The *PLASTIC card should be preceded within the same material definition by an *ELASTIC card, defining the isotropic elastic properties of the material.

By allowing the stress to leave the yield surface temporarily in order to regain it with time, creep effects can be modeled [22]. The viscous part of the viscoplastic law is defined by the *CREEP card. Default is a Norton type law. However, the user can also define his own law in user subroutine ucreep.f. If the *CREEP card is not preceded by a *PLASTIC card, a zero yield surface without any hardening effects is assumed. The *CREEP card must be preceded by an *ELASTIC card defining the isotropic elastic properties of the material.

For this model, there are 13 internal state variables:

- the accumulated equivalent plastic strain $\bar{e^p}$ (1)
- the plastic right Cauchy-Green tensor $\mathbf{C^p}$ (6)
- the back stress $\mathbf{\Gamma}$ (6)

These variables are accessible through the *EL PRINT (.dat file) and *EL FILE (.frd file) keywords in exactly this order (label SDV).

3.3.5 User materials

Other material laws can be defined by the user by means of the *USER MATERIAL keyword card. More information and examples can be found in section 5.3.

3.4 Types of analysis

An analysis type applies to a complete step, which starts with a *STEP card and ends with an *END STEP card. The analysis type, the loading and field output requests must be defined in between.

3.4.1 Static analysis

In a static analysis the time dimension is not involved. The loading is assumed to be applied in a quasi-static way, i.e. so slow that inertia effects can be neglected. A static analysis is defined by the key word *STATIC. A static step can be geometrically linear or nonlinear. In both cases a Lagrangian point of view is taken and all variables are specified in the material frame of reference [8]. Thus, the stress used internally in CalculiX is the second Piola-Kirchhoff tensor acting on the undeformed surfaces.
For geometrically linear calculations the infinitesimal strains are taken (linearized version of the Lagrangian strain tensor), and the loads do not interfere with each other. Thus, the deformation due to two different loads is the sum of the deformation due to each of them. For linear calculations the difference between the Cauchy and Piola-Kirchhoff stresses is neglected.
For geometrically nonlinear calculations, which are triggered by the parameter NLGEOM on the *STEP card, the full Lagrangian strain tensor is used. A geometrically nonlinear calculation is also automatically triggered by nonlinear meterial behavior (e.g. *HYPERELASTIC, *PLASTIC..) no matter whether the parameter NLGEOM is used or not. The step is usually divided into increments, and the user is supposed to provide an initial increment length and the total step length on the *STATIC card. The increment length can be fixed (parameter DIRECT on the *STATIC card) or automatic. In case of automatic incrementation, the increment length is automatically adjusted according to the convergence characteristics of the problem. In each increment, the program iterates till convergence is reached, or a new attempt is made with a smaller increment size. In each iteration the geometrically linear stiffness matrix is augmented with an initial displacement stiffness due to the deformation in the last iteration and with an initial stress stiffness due to the last iteration's stresses [27]. For the output on file the second Piola-Kirchhoff stress is converted into the Cauchy or true stress, since this is the stress which is really acting on the structure.
Special provisions are made for cyclic symmetric structures. A cyclic symmetric structure is characterized by N identical sectors, see Figure 24 and the discussion in next section. Static calculations for such structures under cyclic

symmetric loading lead to cyclic symmetric displacements. Such calculations can be reduced to the consideration of just one sector, the so-called datum sector, subject to cyclic symmetry conditions, i.e. the right boundary of the sector exhibits the same displacements as the left boundary, in cylindrical coordinates (NOT in rectangular coordinates!). The application of these boundary conditions is greatly simplified by the use of the keyword cards *SURFACE, *TIE and *CYCLIC SYMMETRY MODEL, defining the nodes on left and right boundary and the sector size. Then, the appropriate multiple point constraints are generated automatically. This can also be used for a static preload step prior to a perturbative frequency analysis.

3.4.2 Frequency analysis

In a frequency analysis the lowest eigenfrequencies and eigenmodes of the structure are calculated. In CalculiX, the mass matrix is not lumped, and thus a generalized eigenvalue problem has to be solved. The theory can be found in any textbook on vibrations or on finite elements, e.g. [27]. A crucial point in the present implementation is that, instead of looking for the smallest eigenfrequencies of the generalized eigenvalue problem, the largest eigenvalues of the inverse problem are determined. For large problems this results in execution times cut by about a factor of 100 (!). The inversion is performed by calling the linear equation solver SPOOLES. A frequency step is triggered by the key word *FREQUENCY and can be perturbative or not.

If the perturbation parameter is not activated on the *STEP card, the frequency analysis is performed on the unloaded structure, constrained by the homogeneous SPC's and MPC's. Any steps preceding the frequency step do not have any influence on the results.

If the perturbation parameter is activated, the stiffness matrix is augmented by contributions resulting from the displacements and stresses at the end of the last nonperturbative static step, if any, and the material parameters are based on the temperature at the end of that step. Thus, the effect of the centrifugal force on the frequencies in a turbine blade can be analyzed by first performing a static calculation with these loads, and selecting the perturbation parameter on the *STEP card in the subsequent frequency step. The loading at the end of a perturbation step is reset to zero.

If the input deck is stored in the file "problem.inp", where "problem" stands for any name, the eigenfrequencies are stored in the "problem.dat" file. Furthermore, the eigenfrequencies, eigenmodes and mass matrix are stored in binary form for further use (e.g. in a linear dynamic step) in a "problem.eig" file.

All output of the eigenmodes is normalized by means of the mass matrix, i.e. the generalized mass is one.

A special kind of frequency calculations is a cyclic symmetry calculation for which the keyword cards *SURFACE, *TIE, *CYCLIC SYMMETRY MODEL and *SELECT CYCLIC SYMMETRY MODES are available. This kind of calculation applies to structures consisting of identical sectors ordered in a cyclic way such as in Figure 24.

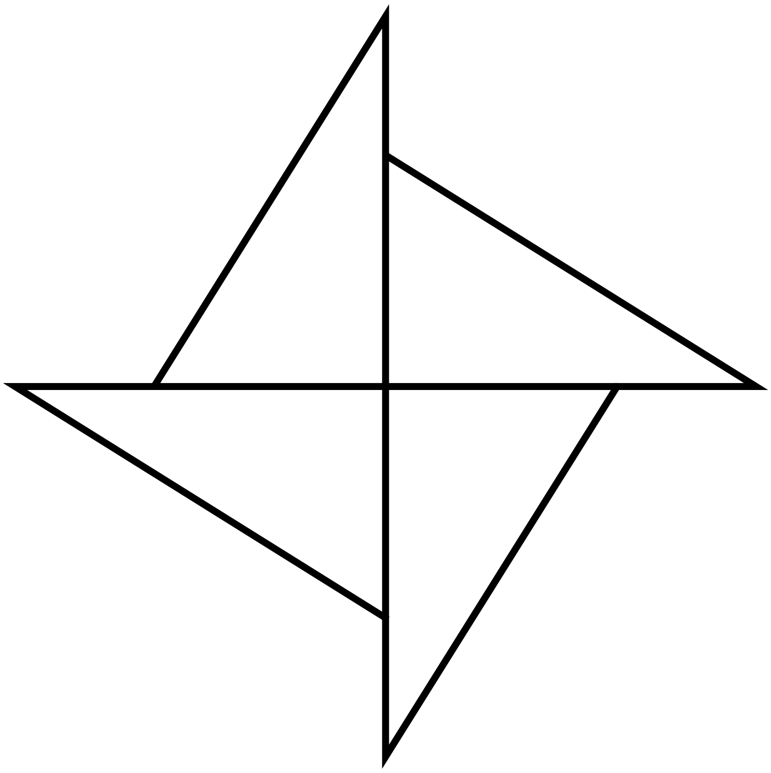

Figure 24: Cyclic symmetry structure consisting of four identical sectors

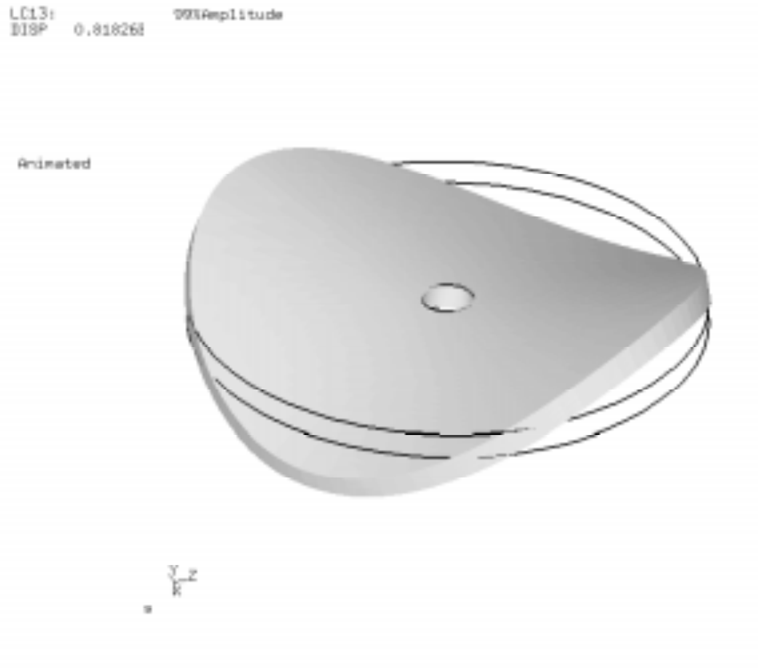

Figure 25: Eigenmode for a full disk with a nodal diameter of two

For such structures it is sufficient to model just one sector (also called datum sector) to obtain the eigenfrequencies and eigenmodes of the whole structure. The displacement values on the left and right boundary (or surfaces) of the datum sector are phase shifted. The shift depends on how many waves are looked for along the circumference of the structure. Figure 25 shows an eigenmode for a full disk exhibiting two complete waves along the circumference. This corresponds to four zero-crossings of the waves and a nodal diameter of two. This nodal diameter (also called cyclic symmetry mode number) can be considered as the number of waves, or also as the number of diameters in the structure along which the displacements are zero.

The lowest nodal diameter is zero and corresponds to a solution which is identical on the left and right boundary of the datum sector. For a structure consisting of N sectors, the highest feasible nodal diameter is N/2 for N even and (N+1)/2 for N odd. The nodal diameter is selected by the user on the *SELECT CYCLIC SMMETRY MODES card. On the same card, the user also indicates the number of sectors for which the solution is to be stored in the .frd file. In this way, the solution can be plotted for the whole structure, although the calculation was done for only one sector.

Mathematically the left and right boundary of the datum sector are coupled by MPC's with complex coefficients. This leads to a complex generalized eigenvalue problem with a Hermitian stiffness matrix, which can be reduced to a real

eigenvalue problem the matrices of which are twice the size as those in the original problem.
For models containing the axis of cyclic symmetry (e.g. a full disk), the nodes on the symmetry axis are treated differently depending on whether the nodal diameter is 0, 1 or exceeds 1. For nodal diameter 0, these nodes are fixed in a plane perperdicular to the cyclic symmetry axis, for nodal diameter 1 they cannot move along the cyclic symmetry axis and for higher nodal diameters they cannot move at all. For these kind of structures calculations for nodal diameters 0 or 1 must be performed in separate steps.

3.4.3 Buckling analysis

In a linear buckling analysis the initial stiffness matrix is augmented by the initial stress matrix corresponding to the load specified in the *BUCKLE step, multiplied with a factor. This so-called buckling factor is determined such that the resulting matrix has zero as its lowest eigenfrequency. Ultimately, the buckling load is the buckling factor multiplied with the step load. The buckling factor(s) are always stored in the .dat file. The load specified in a *BUCKLE step should not contain prescribed displacements.
If the perturbation parameter is not activated on the *STEP card, the initial stiffness matrix corresponds to the stiffness matrix of the unloaded structure.
If the perturbation parameter is activated, the initial stiffness matrix includes the deformation and stress stiffness matrix corresponding to the deformation and stress at the end of the last static or dynamic step performed previous to the buckling step, if any, and the material parameters are based on the temperature at the end of that step. In this way, the effect of previous loadings can be included in the buckling analysis.
In a buckling step, all loading previous to the step is removed and replaced by the buckling step's loading, which is reset to zero at the end of the buckling step. Thus, to continue a static step interrupted by a buckling step, the load has to be reapplied after the buckling step. Due to the intrinsic nonlinearity of temperature loading (the material properties usually change with temperature), this type of loading is not allowed in a linear buckling step. If temperature loading is an issue, a nonlinear static or dynamic calculation should be performed instead.

3.4.4 Modal dynamic analysis

In a modal dynamic analysis, triggered by the *MODAL DYNAMIC key word, the response of the structure to dynamic loading is assumed to be a linear combination of the lowest eigenmodes. These eigenmodes are recovered from a file "problem.eig", where "problem" stands for the name of the structure. These eigenmodes must have been determined in a previous step, either in the same input deck, or in an input deck run previously. The dynamic loading can be defined as a piecewise linear function by means of the *AMPLITUDE key word. The displacement boundary conditions (only zero displacement boundary

conditions are allowed) in a modal dynamic analysis should be the same as those used in the determination of the eigenmodes. Nonzero displacement boundary conditions, temperature loading or residual stresses are not allowed. If such loading arises, the direct integration dynamics procedure should be used.
Damping can be included by means of the *MODAL DAMPING key card. The damping model provided in CalculiX is the Rayleigh damping, which assumes the damping matrix to be a linear combination of the problem's stiffness matrix and mass matrix. This splits the problem according to its eigenmodes, and leads to ordinary differential equations. The results are exact for piecewise linear loading, apart from the inaccuracy due to the finite number of eigenmodes.

3.4.5 Direct integration dynamic analysis

In a direct integration dynamic analysis, activated by the *DYNAMIC key word, the equation of motion is integrated in time using the α-method developed by Hilber, Hughes and Taylor [19]. The parameter α lies in the interval [-1/3,0] and controls the high frequency dissipation: α=0 corresponds to the classical Newmark method inducing no dissipation at all, while α=-1/3 corresponds to maximum dissipation. The user can choose between an implicit and explicit version of the algorithm. The implicit version (default) is unconditionally stable. In the explicit version, triggered by the parameter EXPLICIT in the *DYNAMIC keyword card, the mass matrix is lumped, and a forward integration scheme is used so that the solution can be calculated without solving a system of equations. Each iteration is much faster than with the implicit scheme. However, the explicit scheme is only conditionally stable: the maximum time step size is proportional to the time a mechanical wave needs to cross the smallest element in the mesh. For linear elements the proportionality factor is 1., for quadratic elements it is $1/\sqrt{6}$. For elastic materials, the wave speed in a rod is $\sqrt{E/\rho}$, where E is Young's modulus and ρ is the material density.
The direct integration dynamics implementation in CalculiX performs a geometrically nonlinear analysis, and thus always iterates. In addition, depending on the material law, material nonlinearities can be present.

3.5 Loading

All loading, except residual stresses, must be specified within a step. Its magnitude can be modified by a time dependent amplitude history using the *AMPLITUDE keyword. This makes sense for nonlinear static, nonlinear dynamic and modal dynamic procedures only. Default loading history is a ramp function for *STATIC procedures and step loading for *DYNAMIC and *MODAL DYNAMIC procedures.

3.5.1 Point loads

Point loads are applied to the nodes of the mesh by means of the *CLOAD key word. Applying a point load at a node in a direction for which a point load was

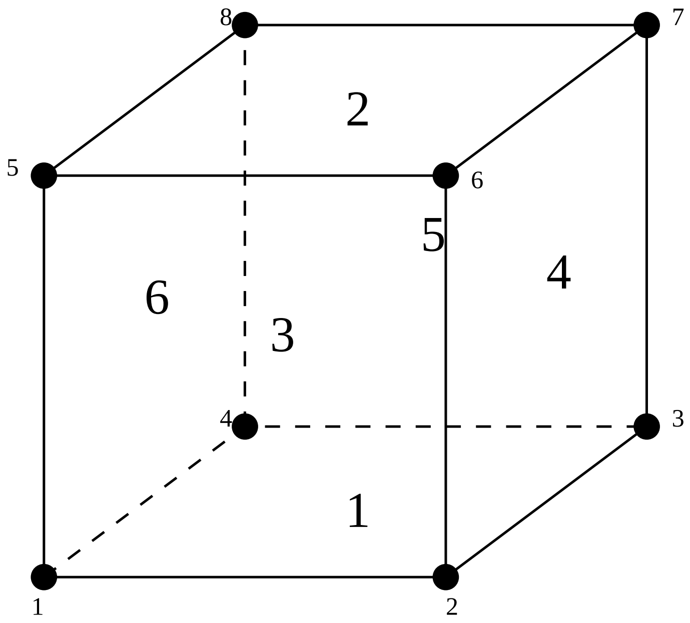

Figure 26: Face numbering for hexahedral elements

specified in a previous step replaces this point load, otherwise it is added. The parameter OP=NEW on the *CLOAD card removes all previous point loads. It takes only effect for the first *CLOAD card in a step. A buckling step always removes all previous loads.

3.5.2 Facial distributed loading

Distributed loading is triggered by the *DLOAD card. Facial distributed loads are entered as pressure loads on the element faces, which are for that purpose numbered according to Figures 26, 27 and 28.
Thus, for hexahedral elements the faces are numbered as follows:

- Face 1: 1-2-3-4
- Face 2: 5-8-7-6
- Face 3: 1-5-6-2
- Face 4: 2-6-7-3
- Face 5: 3-7-8-4
- Face 6: 4-8-5-1

and for tetrahedral elements:

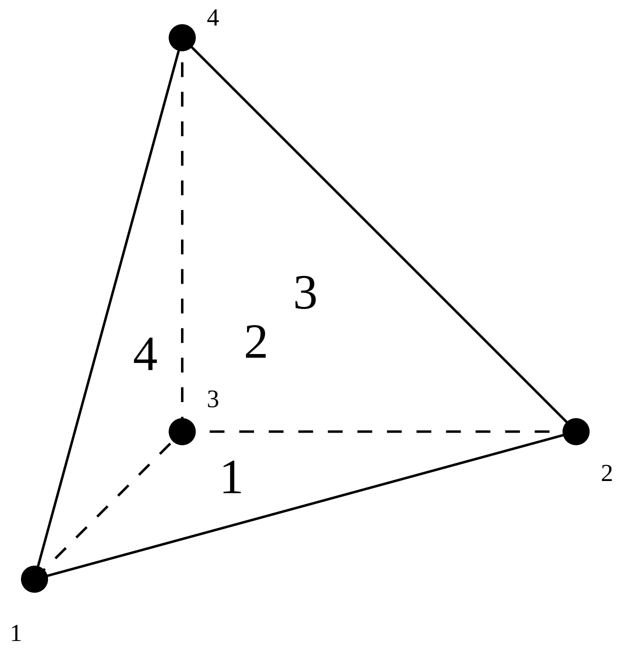

Figure 27: Face numbering for tetrahedral elements

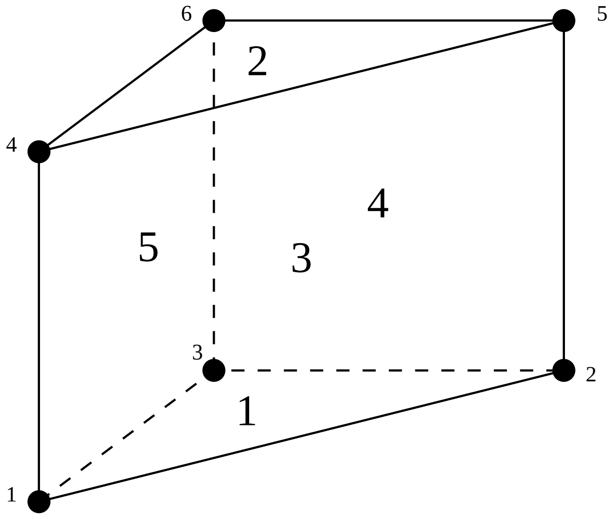

Figure 28: Face numbering for wedge elements

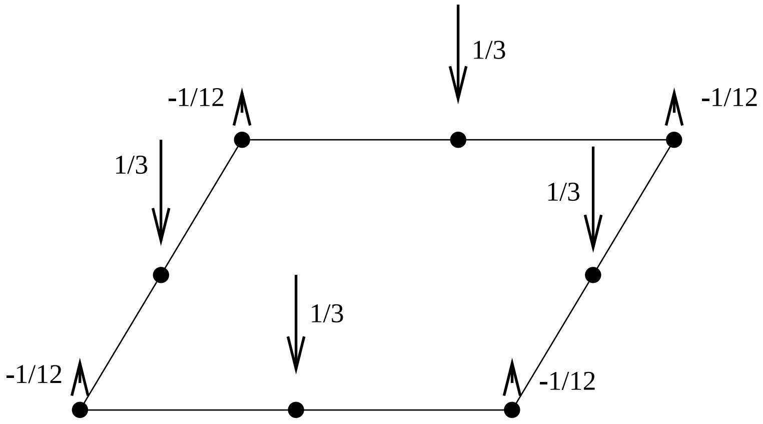

Figure 29: Equivalent nodal forces for a face of a C3D20(R) element

- Face 1: 1-2-3
- Face 2: 1-4-2
- Face 3: 2-4-3
- Face 4: 3-4-1

Applying a pressure to a face for which a pressure was specified in a previous step replaces this pressure. The parameter OP=NEW on the *DLOAD card removes all previous distributed loads. It only takes effect for the first *DLOAD card in a step. A buckling step always removes all previous loads.
In a large deformation analysis the pressure is applied to the deformed face of the element. Thus, if you pull a rod with a constant pressure, the total force will decrease due to the decrease of the cross-sectional area of the rod. This effect may or may not be intended. If not, the pressure can be replaced by nodal forces. Figures 29 and 30 show the equivalent forces for a unit pressure applied to a face of a C3D20(R) and C3D10 element. Notice that the force is zero (C3D10) or has the opposite sign (C3D20(R)) for quadractic elements. For the linear C3D8(R) elements, the force takes the value 1/4 in each node of the face.

3.5.3 Centrifugal distributed loading

Centrifugal loading is selected by the *DLOAD card, together with the CENTRIF label. Centrifugal loading is characterized by its magnitude (defined as the rotational speed square ω^2) and two points on the rotation axes. Once selected, the rotational speed applies to ALL elements of the structure. Thus, if centrifugal loading is specified for one element of the structure, it is applied to all elements in the structure by multiplying its value with the density of the material. Consequently, the material density is required. The parameter

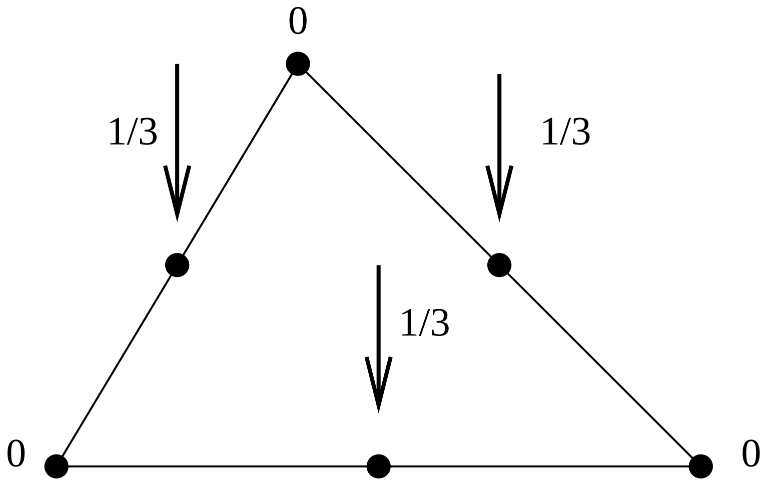

Figure 30: Equivalent nodal forces for a face of a C3D10 element

OP=NEW on the *DLOAD card removes all previous distributed loads. It only takes effect for the first *DLOAD card in a step. A buckling step always removes all previous loads.

3.5.4 Gravity distributed loading

Gravity loading is selected by the *DLOAD card, together with the GRAV label. It is characterized by a vector representing the acceleration. It applies to ALL elements in the structure. The material density is required. The parameter OP=NEW on the *DLOAD card removes all previous distributed loads. It only takes effect for the first *DLOAD card in a step. A buckling step always removes all previous loads.

3.5.5 Temperature loading

Temperature loading is triggered by the keyword *TEMPERATURE. Specification of initial temperatures (*INITIAL CONDITIONS, TYPE=TEMPERATURE) and expansion coefficients (*EXPANSION) is required. The temperature is specified at the nodes. Redefined temperatures replace existing ones.

3.5.6 Initial(residual) stresses

For each element one residual stress tensor can be specified by the keyword *INITIAL CONDITIONS, TYPE=STRESS. The residual stress should be defined before the first *STEP card.

4 Input deck format

This section describes the input of CalculiX.
The jobname is defined by the argument after the -i flag on the command line. When starting CalculiX, it will look for an input file with the name jobname.inp. Thus, if you called the executable “CalculiX” and the input deck is “beam.inp” then the program call looks like

```
CalculiX -i beam
```

The -i flag can be dropped provided the jobname follows immediately after the CalculiX call.
CalculiX will generate an output file with the name jobname.dat and another output file the name of which can be selected with the -o flag. If the -o flag is followed by frd a jobname.frd file is generated. It can be viewed with cgx. If the -o option is followed by onf several .onf files are generated. These are needed if an optimization is performed with the proprietary software TOSCA. Default is frd. Hence, the following two commands have the same effect:

```
CalculiX -i beam

CalculiX -i beam -o frd
```

If the step is a *FREQUENCY step, CalculiX will generate a binary file containing the eigenfrequencies, the eigenmodes and the mass matrix with the name jobname.eig. If the step is a *MODAL DYNAMIC step, CalculiX will look for a file with that name. If any of the files it needs does not exist, an error message is generated and CalculiX will stop.
The input deck basically consists of a set of keywords, followed by data required by the keyword on lines underneath the keyword. The keywords can be accompanied by parameters on the same line, separated by a comma. If the parameters require a value, an equality sign must connect parameter and value. Blanks in the input have no significance and can be inserted as you like. The keywords and any other alphanumeric information can be written in upper case, lower case, or any mixture. The input deck is case insensitive: internally, all alphanumeric characters are changed into upper case. The data do not follow a fixed format and are to be separated by a comma. A line can only contain as many data as dictated by the keyword definition. The maximum length for user defined names, e.g. for materials or sets, is 20 characters, unless specified otherwise.The structure of an input deck consists of geometric, topological and material data before the first step definition, and loading data (mechanical, thermal, or prescribed displacements) in one or more subsequent steps. The

user must make sure that all data are given in consistent units (the units do not appear in the calculation).

A keyword can be of type step or model definition. Model Definition cards must be used before the first *STEP card. Step keywords can only be used within a step. Among the model definition keywords, the material ones occupy a special place: they define the properties of a material and should be grouped together following a *MATERIAL card. Apart from this, there is one additional major requirement for all keywords: if any parameter following a keyword points to another entity, such as a set name or a material name, this entity must have been defined BEFORE. In other words, an entity must have been defined before one can refer to it.

An additional particularity concerns transformations. They only apply to SPC's, MPC's and concentrated loads defined further down the input deck. For example, if a SPC is defined for a node for which a transformation is defined further down, the SPC is applied in the global Carthesian coordinate system. Any SPC defined for the same node after the transformation definition applies in the transformed coordinates. It is strongly recommended not to resort to such a mixture and to define the transformations before the definition of any SPC's and MPC's. For concentrated loads there is no problem since transformations have to be defined BEFORE the first step, and concentrated loads at the earliest IN the first step.

Node and element sets can share the same name. Internally, the names are switched to upper case and a 'N' is appended after the name of a node set and a 'E' after the name of an element set. Therefore, set names printed in error or warning messages will be discovered to be written in upper case and to have a 'N' or 'E' appended.

Keyword cards in alphabetical order:

4.1 *AMPLITUDE

Keyword type: model definition

This option may be used to specify an amplitude history versus time. The amplitude history should be given in pairs, each pair consisting of a value of the reference time and the corresponding value of the amplitude. If the parameter TIME=TOTAL TIME is used the reference time is the total time since the start of the calculation, else it is the local step time. Use as many pairs as needed, maximum four per line. The parameter NAME, specifying a name for the amplitude so that it can be used in loading definitions (*BOUNDARY, *CLOAD, *DLOAD and *TEMPERATURE) is required.This option makes sense for nonlinear static and dynamic (nonlinear and modal dynamic) calculations only. In all other procedures, this card is ignored.

In each step, the local step time starts at zero. Its upper limit is given by the time period of the step. This time period is specified on the *STATIC, *DYNAMIC or *MODAL DYNAMIC keyword card. The default step time period is 1.

The total time is the time accumulated until the beginning of the actual step augmented by the local step time.
The loading values specified in the loading definitions (*BOUNDARY, *CLOAD, *DLOAD and *TEMPERATURE) are reference values. If an amplitude is selected in a loading definition, the actual load value is obtained by multiplying the reference value with the amplitude for the actual (local step) time. If no amplitude is specified, the actual load value depends on the procedure: for a *STATIC procedure, ramp loading is assumed connecting the load value at the end of the previous step (0 if there was none) to the reference value at the end of the present step in a linear way. For *DYNAMIC and *MODAL DYNAMIC procedures, step loading is assumed, i.e. the actual load equals the reference load for all time instances within the step. Reference loads which are not changed in a new step remain active, their amplitude description, however, becomes void. Beware that at the end of a step, all reference values for which an amplitude was specified are replaced by their actual values at that time.

First line:

- *AMPLITUDE
- Enter the required parameter.

Following line, using as many entries as needed:

- Time.
- Amplitude.
- Time.
- Amplitude.
- Time.
- Amplitude.
- Time.
- Amplitude.

Repeat this line if more than eight entries (four data points) are needed.

```
Example:

*AMPLITUDE,NAME=A1
0.,0.,10.,1.
```

defines an amplitude function with name A1 taking the value 0. at t=0. and the value 1. at t=10. The time used is the local step time.

Example files: beamdy1, beamnldy.

4.2 *BEAM SECTION

Keyword type: model definition
This option is used to assign material properties to beam element sets. The parameters ELSET, MATERIAL and SECTION are required, the parameters ORIENTATION, OFFSET1 and OFFSET2 are optional. The parameter ELSET defines the shell element set to which the material specified by the parameter MATERIAL applies. The parameter ORIENTATION allows to assign local axes to the element set. If activated, the material properties are applied to the local axis. This is only relevant for non isotropic material behavior.
The parameter SECTION defines the cross section of the beam and can take the value RECT for a rectangular cross section and CIRC for an elliptical cross section. A rectangular cross section is defined by its thickness in two perpendicular directions, an elliptical cross section is defined by the length of its principal axes. These directions are defined by specifying direction 1 on the third line of the present keyword card
The OFFSET1 and OFFSET2 parameters indicate where the axis of the beam is in relation to the reference line defined by the line representation given by the user. The index 1 and 2 refer to the local axes of the beam which are perpendicular to the local tangent. To use the offset parameters direction the local directions must be defined. This is done by defining local direction 1 on the third line of the present keyword card. The unit of the offset is the thickness of the beam in the direction of the offset. Thus, OFFSET1=0 means that in 1-direction the reference line is the axis of the shell, OFFSET2=0.5 means that in 2-direction the reference line is the top surface of the beam. The offset can take any real value and allows to construct beam of nearly arbitrary cross section and the definition of composite beams.

First line:

- *BEAM SECTION
- Enter any needed parameters.

Second line:

- thickness in 1-direction
- thickness in 2-direction

Third line:

- global x-coordinate of a unit vector in 1-direction (default:0)
- global y-coordinate of a unit vector in 1-direction (default:0)
- global z-coordinate of a unit vector in 1-direction (default:-1)

```
Example:

*BEAM SECTION,MATERIAL=EL,ELSET=Eall,OFFSET1=-0.5,SECTION=RECT
3.,1.
1.,0.,0.
```

assigns material EL to all elements in (element) set Eall. The reference line is in 1-direction on the back surface, in 2-direction on the central surface. The thickness in 1-direction is 3 unit lengths, in 2-direction 1 unit length. The 1-direction is the global x-axis.

Example files: beamcom, beammix, shellbeam, swing.

4.3 *BOUNDARY

Keyword type: step or model definition
This option is used to prescribe geometrical boundary conditions applied to the structure. Optional parameters are OP and AMPLITUDE. OP can take the value NEW or MOD. OP=MOD is default and implies that previously prescribed displacements remain active in subsequent steps. Specifying a displacement in the same node and direction for which a displacement was defined in a previous step replaces this value. OP=NEW implies that previously prescribed displacements are removed. If multiple *BOUNDARY cards are present in a step this parameter takes effect for the first *BOUNDARY card only.
The AMPLITUDE parameter allows for the specification of an amplitude by which the boundary values are scaled (mainly used for nonlinear static and dynamic calculations). This only makes sense for nonzero boundary values. Thus, in that case the values entered on the *BOUNDARY card are interpreted as reference values to be multiplied with the (time dependent) amplitude value to obtain the actual value. At the end of the step the reference value is replaced by the actual value at that time, for use in subsequent steps.
A distinction is made whether the conditions are homogeneous (fixed conditions) or inhomogeneous (prescribed displacements).

4.3.1 Homogeneous Conditions

Homogeneous conditions should be placed before the first *STEP keyword card.

First line:

- *BOUNDARY
- Enter any needed parameters and their value.

Following line:

- Node number or node set label

- First degree of freedom constrained
- Last degree of freedom constrained. This field may be left blank if only one degree of freedom is constrained.

Repeat this line if needed.

```
Example:

*BOUNDARY
73,1,3
```

fixes the degrees of freedom one through three (global if no transformation was defined for node 73, else local) of node 73.

Example files: achteld.

4.3.2 Inhomogeneous Conditions

Inhomogeneous conditions can be defined between a *STEP card and an *END STEP card only.

First line:

- *BOUNDARY
- Enter any needed parameters and their value.

Following line:

- Node number or node set label
- First degree of freedom constrained
- Last degree of freedom constrained. This field may be left blank if only one degree of freedom is constrained.
- Actual magnitude of the prescribed displacement

Repeat this line if needed.

```
Example:

*BOUNDARY
Nall,2,2,.1
```

assigns to degree of freedom two of all nodes belonging to node set Nall the value 0.1.

Example files: achteld.

4.4 *BUCKLE

Keyword type: step
This procedure is used to determine the buckling load of a structure. The load active in the last nonperturbative *STATIC step, if any, will be taken as preload if the perturbation parameter is specified on the *STEP card. All loads previous to a perturbation step are removed at the start of the step; only the load specified within the buckling step is scaled till buckling occurs. Right now, only the stress stiffness due to the buckling load is taken into account and not the large deformation stiffness it may cause.
Buckling leads to an eigenvalue problem whose lowest eigenvalue is the scalar the load in the buckling step has to be multiplied with to get the buckling load. Thus, generally only the lowest eigenvalue is needed. This value is also called the buckling factor and it is always stored in the .dat file.

First line:

- *BUCKLE

Second line:

- Number of buckling factors desired (usually 1).
- Accuracy desired (default: 0.01).
- # Lanczos vectors calculated in each iteration (default: 4 * #eigenvalues).
- Maximum # of iterations (default: 1000).

It is rarely needed to change the defaults.
The eigenvalues are automatically stored in file jobname.dat.

```
Example:

*BUCKLE
2
```

calculates the lowest two buckling modes and the corresponding buckling factors. For the accuracy, the number of Lanczos vectors and the number of iterations the defaults are taken.

Example files: beam8b,beamb.

4.5 *CLOAD

Keyword type: step
This option allows concentrated forces to be applied to any node in the model which is not fixed by a single or multiple point constraint. Optional parameters are OP and AMPLITUDE. OP can take the value NEW or MOD. OP=MOD

is default and implies that the concentrated loads applied to different nodes are kept over all steps starting from the last perturbation step. Specifying a force in a node for which a force was defined in a previous step replaces this value. OP=NEW implies that all previously applied concentrated loads are removed. If multiple *CLOAD cards are present in a step this parameter takes effect for the first *CLOAD card only.
The AMPLITUDE parameter allows for the specification of an amplitude by which the force values are scaled (mainly used for nonlinear static and dynamic calculations). Thus, in that case the values entered on the *CLOAD card are interpreted as reference values to be multiplied with the (time dependent) amplitude value to obtain the actual value. At the end of the step the reference value is replaced by the actual value at that time, for use in subsequent steps.
The AMPLITUDE parameter applies to all loads specified by the same *CLOAD card. This means that, by using several *CLOAD cards, different amplitudes can be applied to the forces in different coordinate directions in one and the same node. An important exception to this rule are nodes in which a transformation applies (by using the *TRANSFORM card): an amplitude defined for such a node applies to ALL coordinate directions. If several are defined, the last one applies.

First line:

- *CLOAD
- Enter any needed parameters and their value.

Following line:

- Node number or node set label.
- Degree of freedom.
- Magnitude of the load

Repeat this line if needed.

```
Example:

*CLOAD,OP=NEW,AMPLITUDE=A1
1000,3,10.3
```

removes all previous point load forces and applies a force with magnitude 10.3 and amplitude A1 for degree of freedom three (global if no transformation was defined for node 1000, else local) of node 1000.

Example files: achtelp.

4.6 *CONTROLS

Keyword type: step
This option is used to change the iteration control parameters. It should only be used by those users who know what they are doing and are expert in the field. There are two, mutually exclusive parameter: PARAMETERS and RESET. The RESET parameter resets the control parameters to their defaults. With the parameter PARAMETERS is used to change the defaults. It can take the value TIME INCREMENTATION or FIELD. These values are mutually exclusive. If the TIME INCREMENTATION value is selected, the number of iterations before certain actions are taken (e.g. the number of divergent iterations before the increment is reattempted) can be changed and effect of these actions (e.g. the increment size is divided by two). The FIELD parameter can be used to change the convergence criteria themselves.
First line:

- *CONTROLS
- Enter the PARAMETERS parameter and its value, or the RESET parameter.

There are no subsequent lines if the parameter RESET is selected.
Following lines if PARAMETER=TIME INCREMENTATION is selected:
Second line:

- I_0 iteration after which a check is made whether the residuals increase in two consecutive iterations (default: 4). If so, the increment is reattempted with D_f times its size.
- I_R iteration after which a logarithmic convergence check is performed in each iteration (default: 8). If more than I_C iterations are needed, the increment is reattempted with D_C its size.
- I_P iteration after which the residual tolerance R_p^α is used instead of R_n^α (default: 9).
- I_C maximum number of iterations allowed (default: 16).
- I_L number of iterations after which the size of the subsequent increment will be reduced (default: 10).
- I_G maximum number of iterations allowed in two consecutive increments for the size of the next increment to be increased (default: 4).
- I_S Currently not used.
- I_A Maximum number of cutbacks per increment (default: 5). A cutback is a reattempted increment.
- I_J Currently not used.

- I_T Currently not used.

Third line:

- D_f Cutback factor if the solution seems to diverge(default: 0.25).
- D_C Cutback factor if the logarithmic extrapolation predicts too many iterations (default: 0.5).
- D_B Cutback factor for the next increment if more than I_L iterations were needed in the current increment (default: 0.75).
- D_A Currently not used.
- D_S Currently not used.
- D_H Currently not used.
- D_D Factor by which the next increment will be increased if less than I_G iterations are needed in two consecutive increments.
- W_G Currently not used.

Following lines if PARAMETER=FIELD is selected:
Second line:

- R_n^α Convergence criterion for the ratio of the largest residual to the average force (default: 0.005). The average force is defined as the average over all increments in the present step of the instantaneous force. The instantaneous force in an increment is defined as the mean of the absolute value of the nodal force components within all elements.
- C_n^α Convergence criterion for the ratio of the largest solution correction to the largest incremental solution value (default: 0.01).
- q_0^α Initial value at the start of a new step of the time average force (default: the time average force from the previous steps or 0.01 for the first step).
- q_u^α User defined average force. If defined, the calculation of the average force is replaced by this value.
- R_p^α Alternative residual convergence criterion to be used after I_P iterations instead of R_n^α (default: 0.02).
- ϵ^α Criterion for zero flux relative to q^α (default: 10^{-5}).
- C_ϵ^α Convergence criterion for the ratio of the largest solution correction to the largest incremental solution value in case of zero flux (default: 10^{-3}).
- R_l^α Convergence criterion for the ratio of the largest residual to the average force for convergence in a single iteration (default: 10^{-8}).

```
Example:

*CONTROLS,PARAMETERS=FIELD
1.e30,1.e30,0.01,,0.02,1.e-5,1.e-3,1.e-8
```

leads to convergence in just one iteration since nearly any residuals are accepted for convergence ($R_n^\alpha = 10^{30}$ and $C_n^\alpha = 10^{30}$.

Example files: beammrco.

4.7 *CREEP

Keyword type: model definition, material
This option is used to define the creep properties of a viscoplastic material. There is one optional parameter LAW. Default is LAW=NORTON, the only other value is LAW=USER for a user defined creep law. The Norton law satisfies:

$$\dot{\epsilon} = A\sigma^n \tag{10}$$

where ϵ is the equivalent creep strain and σ is the true Von Mises stress. For LAW=USER the creep law must be defined in user subroutine ucreep.f.
All constants may be temperature dependent. The card should be preceded by a *ELASTIC card within the same material definition, defining the isotropic elastic properties of the material. For LAW=NORTON the temperature data points under the *CREEP card must be the same as those under the *ELASTIC CARD. If a *PLASTIC card is defined within the same material definition, it should be placed after the *ELASTIC and before the *CREEP card. If no *PLASTIC card is found, a zero yield surface without any hardening is assumed.

First line:

- *CREEP
- Enter the USER parameter and its value, if needed

Following lines are only needed for USER=NORTON (default): First line:

- A.
- n.
- Temperature.

Use as many lines as needed to define the complete temperature dependence.

Example:

```
*CREEP
1.E-10,5.,100.
2.E-10,5.,200.
```

defines a creep law with A=10^{-10} and n=5 for T(temperature)=100. and A=$2 \cdot 10^{-10}$ and n=5 for T(temperature)=200. The card must be preceded by an *ELASTIC card defining the elastic properties for exactly the same temperatures.

Example files: beamcr.

4.8 *CYCLIC HARDENING

Keyword type: model definition,material
This option is used to define the isotropic hardening curves of an incrementally plastic material with combined hardening. All constants may be temperature dependent. The card should be preceded by an *ELASTIC card within the same material definition, defining the isotropic elastic properties of the material.

First line:

- *CYCLIC HARDENING

Following sets of lines defines the isotropic hardening curve: First line in the first set:

- Von Mises stress.
- Equivalent plastic strain.
- Temperature.

Use as many lines in the first set as needed to define the complete hardening curve for this temperature.
Use as many sets as needed to define complete temperature dependence.

Example:

```
*CYCLIC HARDENING
800.,0.,100.
1000.,.1,100.
900.,0.,500.
1050.,.11,500.
```

defines two (stress,plastic strain) data points at T=100. and two data points at T=500. Notice that the temperature must be listed in ascending order. The same is true for the plastic strain within a temperature block.

Example files: beampik.

4.9 *CYCLIC SYMMETRY MODEL

Keyword type: model definition
This keyword is used to define the number of sectors and the axis of symmetry in a cylic symmetric structure for use in a cyclic symmetry calculation. It must be preceded by two *SURFACE cards defining the nodes belonging to the left and right boundary of the sector, respectively, and a *TIE card linking those surfaces, in that order. The axis of symmetry is defined by two points a and b, defined in global Carthesian coordinates.
There are two parameters, N and NGRAPH. The parameter N, specifying the number of sectors, is required.
The parameter NGRAPH is optional and indicates for how many sectors the solutions should be stored in .frd format. Setting NGRAPH=N for N sectors stores the solution for the complete structure for subsequent plotting purposes. Default is NGRAPH=1.

First line:

- *CYCLIC SYMMETRY MODEL
- Enter the required parameter N, and its value.

Second line:

- X-coordinate of point a.
- Y-coordinate of point a.
- Z-coordinate of point a.
- X-coordinate of point b.
- Y-coordinate of point b.
- Z-coordinate of point b.

```
Example:

*CYCLIC SYMMETRY MODEL, N=12, NGRAPH=3
0.,0.,0.,1.,0.,0.
```

defines a cylic symmetric structure consisting of 30 sectors and axis of symmetry through the points (0.,0.,0.) and (1.,0.,0.). The solution will be stored for three connected sectors (120).

Example files: segment, fullseg.

4.10 *DEFORMATION PLASTICITY

Keyword type: model definition, material
This option defines the elasto-plastic behavior of a material by means of the generalized Ramberg-Osgood law. The one-dimensional model takes the form:

$$E\epsilon = \sigma + \alpha \left(\frac{|\sigma|}{\sigma_0} \right)^{n-1} \sigma \tag{11}$$

where ϵ is the logarithmic strain and σ the Cauchy stress. In the present implementation, the Eulerian strain is used, which is very similar to the logarithmic strain (about 1.3 % difference dat 20 % engineering strain). All coefficients may be temperature dependent.

First line:

- *DEFORMATION PLASTICITY

Following line:

- Young's modulus (E).
- Poisson's ratio (ν).
- Yield stress (σ_0)
- Exponent (n).
- Yield offset (α).
- Temperature.

Repeat this line if needed to define complete temperature dependence.

```
Example:

*DEFORMATION PLASTICITY
210000.,.3,800.,12.,0.4
```

defines a Ramberg-Osgood law. No temperature dependence is introduced.

Example files: beampl.

4.11 *DENSITY

Keyword type: model definition, material
With this option the mass density of a material can be defined. The mass density is required for a frequency analysis (*FREQUENCY), for a dynamic analysis (*DYNAMIC) and for a static analysis with gravity loads (GRAV) or centrifugal loads (CENTRIF). The density can be temperature dependent.

First line:

- *DENSITY

Following line:

- Mass density.
- Temperature.

Repeat this line if needed to define complete temperature dependence.

```
Example:

*DENSITY
7.8E-9
```

defines a density with value 7.8^{-9} for all temperatures.

Example files: achtelc, segment1, segment2, beamf.

4.12 *DEPVAR

Keyword type: model definition, material
This keyword is used to define the number of internal state variables for a user defined material. They are initialized to zero at the start of the calculation and can be used within a material user subroutine. There are no parameters. This card must be preceded by a *USER MATERIAL card.

First line:

- *DEPVAR

Second line:

- Number of internal state variables.

```
Example:

*DEPVAR
12
```

defines 12 internal state variables for the user defined material at stake.

Example files: .

4.13 *DLOAD

Keyword type: step
This option allows the specification of distributed loads. These include constant pressure loading on element faces and volume loading either by gravity forces or by centrifugal forces.
For surface loading the faces of the elements are numbered as follows (for the node numbering of the elements see Figure 1 and 2, section 2.1): C3D20R and C3D8:

- face 1: 1-2-3-4
- face 2: 5-8-7-6
- face 3: 1-5-6-2
- face 4: 2-6-7-3
- face 5: 3-7-8-4
- face 6: 4-8-5-1

The surface loading is entered as a uniform pressure with distributed load type label Px where x is the number of the face. Thus, for pressure loading the magnitude of the load is positive, for tension loading it is negative. Optional parameters are OP and AMPLITUDE. OP is used for surface loading only, taking the value NEW or MOD. OP=MOD is default and implies that the surface loads on different faces are kept over all steps starting from the last perturbation step. Specifying a distributed load on a face for which such a load was defined in a previous step replaces this value. OP=NEW implies that all previous surface loading is removed. If multiple *DLOAD cards are present in a step this parameter takes effect for the first *DLOAD card only.
For centrifugal loading the rotational speed square (ω^2) and two points on the rotation axis are required, for gravity loading the size and direction of the gravity vector are to be given. Although both formats call for an element number or element set the load applies to, in the present implementation centrifugal and gravity loading apply to ALL elements. Respecifying either of them replaces the old value. The parameter OP has no effect.
The AMPLITUDE parameter allows for the specification of an amplitude by which the force values are scaled (mainly used for dynamic calculations). Thus, in that case the values entered on the *DLOAD card are interpreted as reference values to be multiplied with the (time dependent) amplitude value to obtain the actual value. At the end of the step the reference value is replaced by the actual value at that time, for use in subsequent steps.

First line:

- *DLOAD

- Enter any needed parameters and their value

Following line for surface loading:

- Element number or element set label.
- Distributed load type label.
- Actual magnitude of the load.

Repeat this line if needed.

```
Example:

*DLOAD,AMPLITUDE=A1
Se1,P3,10.
```

assigns a pressure loading with magnitude 10. times the amplitude curve of amplitude A1 to face number three of all elements belonging to set Se1.

Example files: beamd.
Following line for centrifugal loading:

- Element number or element set label.
- CENTRIF
- rotational speed square (ω^2)
- Coordinate 1 of a point on the rotation axis
- Coordinate 2 of a point on the rotation axis
- Coordinate 3 of a point on the rotation axis
- Component 1 of the normalized direction of the rotation axis
- Component 2 of the normalized direction of the rotation axis
- Component 3 of the normalized direction of the rotation axis

Repeat this line if needed. In the present version, centrifugal loading always applies to the whole structure and the element number or element set label is not used.

```
Example:

*DLOAD
Eall,CENTRIF,100000.,0.,0.,0.,1.,0.,0.
```

Example files: achtelc.
assigns a centrifugal loading with $\omega^2 = 100000.$ about an axis through the points (0.,0.,0.) and (1.,0.,0.) to all elements.
Following line for gravity loading:

- Element number or element set label.
- GRAV
- Actual magnitude of the gravity vector.
- Coordinate 1 of the normalized gravity vector
- Coordinate 2 of the normalized gravity vector
- Coordinate 3 of the normalized gravity vector

Repeat this line if needed. Here "gravity" really stands for any acceleration vector. In the present version gravity loading always applies to the whole structure and the element number or element label is not used.

```
Example:

*DLOAD
Eall,GRAV,9810.,0.,0.,-1.
```

assigns gravity loading in the negative z-direction with magnitude 9810. to all elements.

Example files: achtelg.

4.14 *DYNAMIC

Keyword type: step
This procedure is used to calculate the response of a structure subject to dynamic loading using a direct integration procedure of the equations of motion. Geometrically nonlinear response is assumed. There are four optional parameters: DIRECT, ALPHA, EXPLICIT and SOLVER. The parameter DIRECT specifies that the user defined initial time increment should not be changed. In case of no convergence with this increment size, the calculation stops with an error message. If this parameter is not set, the program will adapt the increment size depending on the rate of convergence. The parameter ALPHA takes an argument between -1/3 and 0. It controls the dissipation of the high frequency responce: lower numbers lead to increased numerical damping ([19]). The default value is -0.05. The parameter EXPLICIT takes no argument and specifies that an explicit integration procedure should be activated. Default is implicit integration.
The last parameter SOLVER determines the package used to solve the ensuing system of equations. This only applies to implicit integration, since explicit

integration does not require the solution of a system of equations. If selected in an explicit integration computation (triggered by the parameter EXPLICIT), the selection of a solver has no effect whatsoever. The default for SOLVER is SPOOLES, denoting the SPOOLES solver [2, 3]. Alternatively, the iterative solver by Rank and Ruecker [20] can be chosen, which is based on the algorithms by Schwarz [21]. If SOLVER=ITERATIVE SCALING is selected, the preconditioning is limited to a scaling of the diagonal terms, SOLVER=ITERATIVE CHOLESKY triggers Incomplete Cholesky preconditioning. Cholesky preconditioning leads to a better convergence and maybe to shorter execution times, however, it requires additional storage roughly corresponding to the nonzeros in the matrix. If you are short of memory, diagonal scaling might be your last resort. The iterative methods perform well for truely three-dimensional structures. For instance, calculations for a hemisphere were about nine times faster with the ITERATIVE SCALING solver, and three times faster with the ITERATIVE CHOLESKY solver than with SPOOLES. For two-dimensional structures such as plates or shells, the performance might break down drastically and convergence often requires the use of Cholesky preconditioning. SPOOLES performs well in most situations with emphasis on slender structures but requires much more storage than the iterative solver.

First line:

- *DYNAMIC
- enter any parameters and their values, if needed.

Second line:

- Initial time increment. This value will be modfied due to automatic incrementation, unless the parameter DIRECT was specified.
- Time period of the step.
- Minimum time increment allowed. Only active if DIRECT is not specified. Default is the initial time increment or 1.e-5 times the time period of the step, whichever is smaller.
- Maximum time increment allowed. Only active if DIRECT is not specified. Default is 1.e+30.

```
Examples:

*DYNAMIC,DIRECT,EXPLICIT
1.E-7,1.E-5
```

defines an explicit dynamic procedure with fixed time increment 10^{-7} for a step of length 10^{-5}.

```
*DYNAMIC,ALPHA=-0.3,SOLVER=ITERATIVE CHOLESKY
1.E-7,1.E-5,1.E-9,1.E-6
```

defines an implicit dynamic procedure with variable increment size. The numerical damping was increased ($\alpha = -0.3$ instead of the default $\alpha = -0.05$, and the iterative solver with Cholesky preconditioning was selected. The starting increment has a size 10^{-7}, the subsequent increments should not have a size smaller than 10^{-9} or bigger than 10^{-6}. The step size is 10^{-5}.

Example files: beamnldy, beamnldye, beamnldyp, beamnldype.

4.15 *ELASTIC

Keyword type: model definition, material
This option is used to define the elastic properties of a material. There is one optional parameter TYPE. Default is TYPE=ISO, other values are TYPE=ORTHO for orthotropic materials and TYPE=ANISO for anisotropic materials. All constants may be temperature dependent. For orthotropic and fully anisotropic materials, the coefficients D_{IJKL} satisfy the equation:

$$S_{IJ} = D_{IJKL} E_{KL}, \tag{12}$$

where S_{IJ} is the second Piola-Kirchhoff stress and E_{KL} is the Lagrange deformation tensor. For linear calculations, these reduce to the generic stress and strain tensors.

First line:

- *ELASTIC
- Enter the TYPE parameter and its value, if needed

Following line for TYPE=ISO:

- Young's modulus.
- Poisson's ratio.
- Temperature.

Repeat this line if needed to define complete temperature dependence.
Following lines, in a pair, for TYPE=ORTHO: First line of pair:

- D_{1111}.
- D_{1122}.
- D_{2222}.
- D_{1133}.
- D_{2233}.
- D_{3333}.

- D_{1212}.
- D_{1313}.

Second line of pair:

- D_{2323}.
- Temperature.

Repeat this pair if needed to define complete temperature dependence. Following lines, in sets of 3, for TYPE=ANISO: First line of set:

- D_{1111}.
- D_{1122}.
- D_{2222}.
- D_{1133}.
- D_{2233}.
- D_{3333}.
- D_{1112}.
- D_{2212}.

Second line of set:

- D_{3312}.
- D_{1212}.
- D_{1113}.
- D_{2213}.
- D_{3313}.
- D_{1213}.
- D_{1313}.
- D_{1123}.

Third line of set:

- D_{2223}.
- D_{3323}.
- D_{1223}.
- D_{1323}.

- D_{2323}.
- Temperature.

Repeat this set if needed to define complete temperature dependence.

`Example:`

```
*ELASTIC,TYPE=ORTHO
500000.,157200.,400000.,157200.,157200.,300000.,126200.,126200.,
126200.,294.
```

defines an orthotropic material for temperature T=294. Since the definition includes values for only one temperature, they are valid for all temperatures.

Example files: aniso, beampo1.

4.16 *ELEMENT

Keyword type: model definition
With this option elements are defined. There is one required parameter, TYPE and one optional parameter, ELSET. The parameter TYPE defines the kind of element which is being defined. The following types can be selected:

- C3D8 (3-D 8-node linear isoparametric element)
- C3D8R (the C3D8 element with reduced integration)
- C3D10 (10-node quadratic tetrahedral element)
- C3D20 (3-D 20-node quadratic isoparametric element)
- C3D20R (the C3D20 element with reduced integration)
- S8 (8-node quadratic shell element)
- S8R (the S8 element with reduced integration)
- CPS8 (8-node plane stress element)
- CPS8R (the CPS8 element with reduced integration)
- CPE8 (8-node plane strain element)
- CPE8R (the CPS8 element with reduced integration)
- CAX8 (8-node axisymmetric element)
- CAX8R (the CAX8 element with reduced integration)
- B32 (3-node beam element)

- B32R (the B32 element with reduced integration)

Notice that the S8, S8R, CPS8, CPS8R, CPE8, CPE8R, CAX8, CAX8R, B32 and B32R element are internally expanded into 20-node brick elements. Please have a look at 3.1 for details and decision criteria which element to take. The element choice determines to a large extent the quality of the results. Do not take element choice lightheartedly! The parameter ELSET is used to assign the elements to an element set. If the set already exists, the elements are ADDED to the set.

First line:

- *ELEMENT
- Enter any needed parameters and their values.

Following line:

- Element number.
- Node numbers forming the element. The order of nodes around the element is given in section 2.1. Use continuation lines for elements having more than 15 nodes (maximum 16 entries per line).

Repeat this line if needed.

```
Example:

*ELEMENT,ELSET=Eall,TYPE=C3D20R
1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
16,17,18,19,20
```

defines one 20-node element with reduced integration and stores it in set Eall.

Example files: beam8p, beam10p, beam20p.

4.17 *EL FILE

Keyword type: step
This option is used to save selected element variables averaged at the nodal points in a frd file (extension .frd) for subsequent viewing by CalculiX GraphiX. The following element variables can be selected:

- true (Cauchy) stress (key=S)
- strain (key=E). This is the Lagrangian strain for (hyper)elastic materials, the Eulerian strain for deformation plasticity and the deviatoric elastic left Cauchy-Green tensor for incremental plasticity.
- equivalent plastic strain (key=PE)

- the energy density (key=ENER)
- the internal state variables (key=SDV)

The selected variables are stored for the complete model. Due to the averaging process jumps at material interfaces are smeared out unless you model the materials on both sides of the interface independently and connect the coinciding nodes with MPC's.
The first occurrence of an *EL FILE keyword card within a step wipes out all previous element variable selections for file output. If no *EL FILE card is used within a step the selections of the previous step apply. If there is no previous step, and no element variable selection nor nodal variable selection is made for file output, the stresses (S) are stored by default.
The only optional parameter is FREQUENCY, and applies to nonlinear calculations where a step can consist of several increments. Default is FREQUENCY=1, which indicates that the results of all increments will be stored. FREQUENCY=N with N an integer indicates that the results of every Nth increment will be stored. The final results of a step are always stored. If you only want the final results, choose N very big. The value of N applies to *EL FILE, *ELPRINT, *NODE FILE and *NODE PRINT. If the FREQUENCY parameter is used for more than one of these keywords with conflicting values of N, the biggest value applies to all.

First line:

- *EL FILE
- Enter any needed parameters and their values.

Second line:

- Identifying keys for the variables to be printed, separated by kommas.

```
Example:

*EL FILE
S,PE
```

requests that the (Cauchy) stresses and the equivalent plastic strain is stored in .frd format for subsequent viewing with CalculiX GraphiX.

Example files: beamt, fullseg, segment1.

4.18 *EL PRINT

Keyword type: step
This option is used to print selected element variables in an ASCII file with the name jobname.dat. The following element variables can be selected:

- true (Cauchy) stress (key=S)
- strain (key=E). This is the Lagrangian strain for (hyper)elastic materials, the Eulerian strain for deformation plasticity and the deviatoric elastic left Cauchy-Green tensor for incremental plasticity.
- equivalent plastic strain (key=PE)
- the energy density (key=ENER)
- the internal state variables (key=SDV)

There are two parameter, ELSET and FREQUENCY. The parameter ELSET is required, defining the set of elements for which these stresses should be printed. If this card is omitted, no values are printed. Several *EL PRINT cards can be used within one and the same step. In that case, the variables selected in any of these *EL PRINT cards will be printed for all sets.
The FREQUENCY parameter is optional and applies to nonlinear calculations where a step can consist of several increments. Default is FREQUENCY=1, which indicates that the results of all increments will be stored. FREQUENCY=N with N an integer indicates that the results of every Nth increment will be stored. The final results of a step are always stored. If you only want the final results, choose N very big. The value of N applies to *EL FILE, *ELPRINT, *NODE FILE and *NODE PRINT. If the FREQUENCY parameter is used for more than one of these keywords with conflicting values of N, the biggest value applies to all.
The first occurrence of an *EL FILE keyword card within a step wipes out all previous element variable selections for print output. If no *EL FILE card is used within a step the selections of the previous step apply, if any.

First line:

- *EL PRINT
- Enter the parameter ELSET and its value.

Second line:

- Identifying keys for the variables to be printed, separated by kommas.

```
Example:

*EL PRINT,ELSET=Copper
E
```

requests to store the strains at the integration points in the elements of set Copper in the .dat file.

Example files: beampt, beamrb, beamt4.

4.19 *ELSET

Keyword type: model definition
This option is used to assign elements to an element set. The parameter ELSET containing the name of the set is required, whereas the parameter GENERATE (without value) is optional. If present, element ranges can be expressed by their initial value, their final value, and an increment. If a set with the same name already exists, it is reopened and complemented. The name of a set is case insensitive. Internally, it is modified into upper case and a 'E' is appended to denote it as element set.

First line:

- *ELSET
- Enter any needed parameters and their values.

Following line if the GENERATE parameter is omitted:

- List of elements and/or sets of elements previously defined to be assigned to this element set (maximum 16 entries per line).

Repeat this line if needed.
Following line if the GENERATE parameter is included:

- First element in set.
- Last element in set.
- Increment in element numbers between elements in the set. Default is 1.

Repeat this line if needed.

```
Example:

*ELSET,ELSET=E1,GENERATE
20,25
*ELSET,ELSET=E2
E1,50,51
```

assigns the elements with numbers 20, 21, 22, 23, 24 and 25 to element set E1 and the elements with numbers 20, 21, 22, 23, 24, 25 (= set E1), 50 and 51 to element set E2.

Example files: segment, beampo1, beampset.

4.20 *END STEP

Keyword type: step
This option concludes the definition of a step.

First and only line:

- *END STEP

```
Example:

*END STEP
```

concludes a step. Each *STEP card must at some point be followed by an *END STEP card.

Example files: beamstraight, beamt.

4.21 *EQUATION

Keyword type: model definition
With this option, a linear equation constraint between arbitrary displacement components at any nodes where these components are active can be imposed. The equation is assumed to be homogeneous, and all variables are to be written on the left hand side of the equation. The first variable is considered to be the dependent one, and is subsequently eliminated from the equation, i.e. the corresponding degree of freedom does not show up in the stiffness matrix. This reduces the size of the matrix. A node can only be used once as the dependent node in an equation or in a SPC.

First line:

- *EQUATION

Following lines, in a set: First line of set:

- Number of terms in the equation.

Following lines of set (maximum 12 entries per line):

- Node number of the first variable.
- Degree of freedom at above node for the first variable.
- Value of the coefficient of the first variable.
- Node number of the second variable.
- Degree of freedom at above node for the second variable.
- Value of the coefficient of the second variable.

- Etc..

Continue, giving node number, degree of freedom, value of the coefficient, etc. Repeat the above line as often as needed if there are more than four terms in the *EQUATION. Specify exactly four terms per line for each constraint, except for the last line which may have less than four terms.

```
Example:

*EQUATION
3
3,2,2.3,28,1,4.05,17,1,-8.22
```

defines an equation of the form $2.3v_3 + 4.05u_{28} - 8.22u_{17} = 0$, where u, v and w are the displacement for degree of freedom one, two and three, respectively.

Example files: achtel2, achtel29, achtel9, achtelcas, beamnlmpc.

4.22 *EXPANSION

Keyword type: model definition, material

This option is used to define the thermal expansion coefficients of a material. They are interpreted as total expansion coefficients with respect to a reference temperature T_{ref}, i.e. the thermal strain ϵ_{th} of a material at a final temperature T and with initial temperature T_0 is determined by

$$\epsilon_{th} = \alpha(T)(T - T_{ref}) - \alpha(T_o)(T_o - T_{ref}), \tag{13}$$

where $\alpha(T)$ is the thermal coefficient at a temperature T. There are two optional parameters TYPE and ZERO. Default for TYPE is TYPE=ISO, other values are TYPE=ORTHO for orthotropic materials and TYPE=ANISO for anisotropic materials. All constants may be temperature dependent. The parameter ZERO is used to determine the reference temperature, default is 0.

First line:

- *EXPANSION
- Enter the TYPE and ZERO parameters and their values, if needed

Following line for TYPE=ISO:

- α.
- Temperature.

Repeat this line if needed to define complete temperature dependence.
Following line for TYPE=ORTHO:

- α_{11}.
- α_{22}.
- α_{33}.
- Temperature.

Repeat this line if needed to define complete temperature dependence.
Following line for TYPE=ANISO:

- α_{11}.
- α_{22}.
- α_{33}.
- α_{12}.
- α_{13}.
- α_{23}.
- Temperature.

Repeat this line if needed to define complete temperature dependence.

```
Example:

*EXPANSION,ZERO=273.
12.E-6,373.
20.E-6,573.
```

tells you that the thermal strain in a body made of this material is $12.^{-6}$ if heated from T=273 to T=373, and 20^{-6} if heated from T=273 to T=573.

Example files: beamt, beamt2.

4.23 *FREQUENCY

Keyword type: step
This procedure is used to determine eigenfrequencies and the corresponding eigenmodes of a structure. If the PERTURBATION parameter is used in the *STEP card, the load active in the last *STATIC step, if any, will be taken as preload. Otherwise, no preload will be active.
There are no parameters. Default and only solver is the ARPACK solver [12]. The eigenfrequencies are always stored in file jobname.dat.

First line:

- *FREQUENCY

Second line:

- Number of eigenfrequencies desired.
- Accuracy desired (default: 0.01).
- # Lanczos vectors calculated in each iteration (default: 4 * #eigenfrequencies).
- Maximum # of iterations (default: 1000).

It is rarely needed to change the defaults.

```
Example:

*FREQUENCY
10
```

requests the calculation of the 10 lowest eigenfrequencies and corresponding eigenmodes. For the accuracy, the number of Lanczos vectors and the number of iterations, the defaults are taken.

Example files: beam8f, beamf.

4.24 *HEADING

Keyword type: model definition
The heading block allows for a short problem description for identification and retrieval purposes. This description is reproduced at the top of the output file.

First line:

- *HEADING

Following line:

- Description of the problem.

```
Example:

*HEADING
Cantilever beam under tension and bending.
```

gives a title to the problem.

Example files: beampt, segment1.

4.25 *HYPERELASTIC

Keyword type: model definition, material
This option is used to define the hyperelastic properties of a material. There are two optional parameters. The first one defines the model and can take one of the following strings: ARRUDA-BOYCE, MOONEY-RIVLIN, NEO HOOKE, OGDEN, POLYNOMIAL, REDUCED POLYNOMIAL or YEOH. The second parameter N makes sense for the OGDEN, POLYNOMIAL and REDUCED POLYMIAL model only, and determines the order of the strain energy potential. Default is the POLYNOMIAL model with N=1. All constants may be temperature dependent.
Let $\bar{I}_1$,$\bar{I}_2$ and J be defined by:

$$\begin{aligned}\bar{I}_1 &= III_C^{-1/3} I_C && (14)\\ \bar{I}_2 &= III_C^{-1/3} II_C && (15)\\ J &= III_C^{1/2} && (16)\end{aligned}$$

where I_C, II_C and III_C are the invariants of the right Cauchy-Green deformation tensor $C_{KL} = x_{k,K} x_{k,L}$. The tensor C_{KL} is linked to the Lagrange strain tensor E_{KL} by:

$$2E_{KL} = C_{KL} - \delta_{KL} \qquad (17)$$

where δ is the Kronecker symbol.
The Arruda-Boyce strain energy potential takes the form:

$$\begin{aligned}U &= \mu \left\{ \frac{1}{2}(\bar{I}_1 - 3) + \frac{1}{20\lambda_m^2}(\bar{I}_1^2 - 9) + \frac{11}{1050\lambda_m^4}(\bar{I}_1^3 - 27) \right.\\ &+ \left. \frac{19}{7000\lambda_m^6}(\bar{I}_1^4 - 81) + \frac{519}{673750\lambda_m^8}(\bar{I}_1^5 - 243) \right\} && (18)\\ &+ \frac{1}{D}\left(\frac{J^2 - 1}{2} - \ln J \right)\end{aligned}$$

The Mooney-Rivlin strain energy potential takes the form:

$$U = C_{10}(\bar{I}_1 - 3) + C_{01}(\bar{I}_2 - 3) + \frac{1}{D_1}(J - 1)^2 \qquad (19)$$

The Mooney-Rivlin strain energy potential is identical to the polynomial strain energy potential for $N = 1$.
The Neo-Hooke strain energy potential takes the form:

$$U = C_{10}(\bar{I}_1 - 3) + \frac{1}{D_1}(J - 1)^2 \qquad (20)$$

The Neo-Hooke strain energy potential is identical to the reduced polynomial strain energy potential for $N = 1$.

The polynomial strain energy potential takes the form:

$$U = \sum_{i+j=1}^{N} C_{ij}(\bar{I}_1 - 3)^i(\bar{I}_2 - 3) + \sum_{i=1}^{N} \frac{1}{D_i}(J-1)^{2i} \tag{21}$$

In CalculiX $N \leq 3$.
The reduced polynomial strain energy potential takes the form:

$$U = \sum_{i=1}^{N} C_{i0}(\bar{I}_1 - 3)^i + \sum_{i=1}^{N} \frac{1}{D_i}(J-1)^{2i} \tag{22}$$

In CalculiX $N \leq 3$. The reduced polynomial strain energy potential can be viewed as a special case of the polynomial strain energy potential
The Yeoh strain energy potential is nothing else but the reduced polynomial strain energy potential for $N = 3$.
Denoting the principal stretches by λ_1, λ_2 and λ_3 (λ_1^2, λ_2^2 and λ_3^2 are the eigenvalues of the right Cauchy-Green deformation tensor) and the deviatoric stretches by $\bar{\lambda}_1$, $\bar{\lambda}_2$ and $\bar{\lambda}_3$, where $\bar{\lambda}_i = III_C^{-1/6}\lambda_i$, the Ogden strain energy potential takes the form:

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2}(\bar{\lambda}_1^{\alpha_i} + \bar{\lambda}_2^{\alpha_i} + \bar{\lambda}_3^{\alpha_i} - 3) + \sum_{i=1}^{N} \frac{1}{D_i}(J-1)^{2i}. \tag{23}$$

The input deck for a hyperelastic material looks as follows:

First line:

- *HYPERELASTIC
- Enter parameters and their values, if needed

Following line for the ARRUDA-BOYCE model:

- μ.
- λ_m.
- D.
- Temperature

Repeat this line if needed to define complete temperature dependence.
Following line for the MOONEY-RIVLIN model:

- C_{10}.
- C_{01}.
- D_1.

- Temperature

Repeat this line if needed to define complete temperature dependence. Following line for the NEO HOOKE model:

- C_{10}.
- D_1.
- Temperature.

Repeat this line if needed to define complete temperature dependence. Following line for the OGDEN model with N=1:

- μ_1.
- α_1.
- D_1.
- Temperature.

Repeat this line if needed to define complete temperature dependence. Following line for the OGDEN model with N=2:

- μ_1.
- α_1.
- μ_2.
- α_2.
- D_1.
- D_2.
- Temperature.

Repeat this line if needed to define complete temperature dependence. Following lines, in a pair, for the OGDEN model with N=3: First line of pair:

- μ_1.
- α_1.
- μ_2.
- α_2.
- μ_3.
- α_3.
- D_1.

- D_2.

Second line of pair:

- D_3.
- Temperature.

Repeat this pair if needed to define complete temperature dependence.
Following line for the POLYNOMIAL model with N=1:

- C_{10}.
- C_{01}.
- D_1.
- Temperature.

Repeat this line if needed to define complete temperature dependence.
Following line for the POLYNOMIAL model with N=2:

- C_{10}.
- C_{01}.
- C_{20}.
- C_{11}.
- C_{02}.
- D_1.
- D_2.
- Temperature.

Repeat this line if needed to define complete temperature dependence.
Following lines, in a pair, for the POLYNOMIAL model with N=3: First line of pair:

- C_{10}.
- C_{01}.
- C_{20}.
- C_{11}.
- C_{02}.
- C_{30}.
- C_{21}.

- C_{12}.

Second line of pair:

- C_{03}.
- D_1.
- D_2.
- D_3.
- Temperature.

Repeat this pair if needed to define complete temperature dependence. Following line for the REDUCED POLYNOMIAL model with N=1:

- C_{10}.
- D_1.
- Temperature.

Repeat this line if needed to define complete temperature dependence. Following line for the REDUCED POLYNOMIAL model with N=2:

- C_{10}.
- C_{20}.
- D_1.
- D_2.
- Temperature.

Repeat this line if needed to define complete temperature dependence. Following line for the REDUCED POLYNOMIAL model with N=3:

- C_{10}.
- C_{20}.
- C_{30}.
- D_1.
- D_2.
- D_3.
- Temperature.

Repeat this line if needed to define complete temperature dependence. Following line for the YEOH model:

- C_{10}.
- C_{20}.
- C_{30}.
- D_1.
- D_2.
- D_3.
- Temperature.

Repeat this line if needed to define complete temperature dependence.

```
Example:

*HYPERELASTIC,OGDEN,N=1
3.488,2.163,0.
```

defines an ogden material with one term: $\mu_1 = 3.488$, $\alpha_1 = 2.163$, D_1=0. Since the compressibility coefficient was chosen to be zero, it will be replaced by CalculiX by a small value to ensure some compressibility to garantuee convergence (cfr. page 34).

Example files: beamnh, beamog.

4.26 *HYPERFOAM

Keyword type: model definition, material
This option is used to define a hyperfoam material. There is one optional parameters, N. N determines the order of the strain energy potential. Default is N=1. All constants may be temperature dependent.
The hyperfoam strain energy potential takes the form

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2} \left[\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3 + \frac{1}{\beta}(J^{-\alpha_i \beta_i} - 1) \right] \tag{24}$$

where λ_1, λ_2 and λ_3 are the principal stretches.

First line:

- *HYPERFOAM
- Enter parameters and their values, if needed

Following line for N=1:

- μ_1.

- α_1.
- ν_1.
- Temperature.

Repeat this line if needed to define complete temperature dependence. Following line for N=2:

- μ_1.
- α_1.
- μ_2.
- α_2.
- ν_1.
- ν_2.
- Temperature.

Repeat this line if needed to define complete temperature dependence. Following lines, in a pair, for N=3: First line of pair:

- μ_1.
- α_1.
- μ_2.
- α_2.
- μ_3.
- α_3.
- ν_1.
- ν_2.

Second line of pair:

- ν_3.
- Temperature.

Repeat this pair if needed to define complete temperature dependence.

```
Example:

*HYPERFOAM,N=2
0.164861,8.88413,2.302e-5,-4.81798,0.,0.
```

defines a hyperfoam material with two terms in the series.

Example files: beamhf.

4.27 *INCLUDE

Keyword type: step or model definition
The include statement allows to store part of the input deck in another file. There is only one required parameter, INPUT, taking the name of the file in or without double quotes ("). The double quotes are needed if the file name contains one or more blanks.

First line:

- *INCLUDE
- Enter the parameter and its value.

```
Example:

*INCLUDE,INPUT=/home/guido/test/beam.spc
```

is at execution time replaced by the contents of file /home/guido/test/beam.spc.

Example files: .

4.28 *INITIAL CONDITIONS

Keyword type: model definition
This option is used to define initial temperatures and initial stresses. There is one parameter required: TYPE. TYPE=STRESS is used to define initial stresses, TYPE=TEMPERATURE is used to define initial temperatures and TYPE=VELOCITY defines initial velocities (for dynamic calculations).

First line:

- *INITIAL CONDITIONS
- Enter any needed parameters and their values.

Following line for TYPE=STRESS:

- Element number or element set label.
- Value of first stress component.
- Value of second stress component.
- Etc.

Repeat this line if needed. The stress components should be given in the form of second Piola-Kirchhoff stresses.
Following line for TYPE=TEMPERATURE:

- Node number or node set label.

- Initial temperature value at the node.

Repeat this line if needed.
Following line for TYPE=VELOCITY:

- Node number or node set label.
- Degree of freedom in the GLOBAL coordinate system.
- Magnitude of the velocity.

```
Examples:

*INITIAL CONDITIONS,TYPE=TEMPERATURE
Nall,273.
```

assigns the initial temperature T=273. to all nodes in (node) file Nall.

```
*INITIAL CONDITIONS,TYPE=VELOCITY
18,2,3.15
```

assigns the initial velocity 3.15 to degree of freedom 2 of node 18.

Example files: beam20t, beamnlt, beamt3.

4.29 *MATERIAL

Keyword type: model definition
This option is used to indicate the start of a material definition. A material data block is defined by the options between a *MATERIAL line and either another *MATERIAL line or a keyword line that does not define material properties. All material options within a data block will be assumed to define the same material. If a property is defined more than once for a material, the last definition is used. There is one required parameter, NAME, defining the name of the material with which it can be referenced in element property options (e.g. *SOLID SECTION). The name can contain up to 8 characters.
Material data requests outside the defined ranges are extrapolated in a constant way and a warning is generated. Be aware that this occasionally occurs due to rounding errors.

First line:

- *MATERIAL
- Enter the NAME parameter and its value.

```
Example:

*MATERIAL,NAME=EL
```

starts a material block with name EL.

Example files: fullseg, beamnldype, beamog.

4.30 *MODAL DAMPING

Keyword type: step
This card is used within a step in which the *MODAL DYNAMIC procedure has been selected. It allows for Raleigh damping in a global way, i.e. the damping matrix is taken to be a linear combination of the stiffness matrix and the mass matrix. The coefficients apply to all modes.

First line:

- *MODAL DAMPING

Second line:

- Coefficient of the mass matrix.
- Coefficient of the stiffness matrix.

```
Example:

*MODAL DAMPING
0.,2.e-4
```

indicates that the damping matrix is obtained by multiplying the stiffness matrix with $2 \cdot 10^{-4}$

Example files: beamdy3, beamdy4, beamdy5, beamdy6.

4.31 *MODAL DYNAMIC

Keyword type: step
This procedure is used to calculate the response of a structure subject to dynamic loading. Although the deformation up to the onset of the dynamic calculation can be nonlinear, this procedure is basically linear and assumes that the response can be written as a linear combination of the lowest modes of the structure. To this end, these modes must have been calculated in a previous *FREQUENCY step (not necessarily in the same calculation). In the *MODAL DYNAMIC step the eigenfrequencies, modes and mass matrix are recovered from the file jobname.eig. The time period of the loading is characterized by its total length and the length of an increment. Within each increment the loading is assumed to be linear, in which case the solution is exact apart from modelling inaccuracies and the fact that not all eigenmodes are used. The number of eigenmodes used is taken from the previous *FREQUENCY step. The dynamic loading is the static loading at the start of the step multiplied by the amplitude history for each load as specified by the AMPLITUDE parameter on the loading card, if any. Loading histories extending beyond the amplitude time scale are extrapolated in a constant way. The absence of the AMPLITUDE parameter on a loading card leads to a constant load.

First line:

- *MODAL DYNAMIC

Second line:

- Time increment.
- Total time period.

```
Example:

*MODAL DYNAMIC
1.E-5,1.E-4
```

defines a modal dynamic procedure with time increment 10^{-5} and time period 10^{-4}.

Example files: beamdy1, beamdy2, beamdy3, beamdy4, beamdy5, beamdy6.

4.32 *MPC

Keyword type: model definition
With this keyword card a multiple point constraint is defined, usually a nonlinear one. Right now, three different MPC's can be selected.

- A plane MPC (name PLANE). This MPC specifies that all nodes listed within this MPC card should stay in a plane. The first three nodes are the defining nodes and should not lie on a line. For all subsequent nodes a nonlinear MPC is generated expressing that they stay within the plane. Notice that the plane can move during deformation, depending on the motion of the defining nodes.
- A straight line MPC (name STRAIGHT). This MPC expresses that all nodes listed within this MPC card should stay on a straight line. The first two nodes are the defining nodes and should not coincide. For all subsequent nodes two nonlinear MPC's are generated expressing that they stay on the straight line. Notice that the straight line can move during deformation, depending on the motion of its defining nodes.
- A user MPC (name to be defined by the user). With this option the user can define new nonlinear MPC's.

If NLGEOM is not specified on the *STEP card, the MPC is linearized. Note that depending on the degree of nonlinearity of the MPC linearization can lead to significant errors in the results. It is generally advisable to start with a linear calculation to check the correctness of the structure, loading and boundary conditions. If the linear calculation runs satisfactorily, a geometrically nonlinear calculation should be performed.
There are no parameters for this keyword card.

First line:

- *MPC

Second line:

- MPC type
- list of nodes participating in the MPC: maximum 15 entries. Zero entries are discarded.

Following lines (as many as needed):

- list of nodes participating in the MPC: maximum 16 entries. Zero entries are discarded.

```
Example:

*MPC
PLANE,3,8,15,39,14
```

specifies that nodes 3, 8, 15, 39 and 14 should stay in a plane. The plane is defined by nodes 3, 8 and 15. They should not be colinear.

Example files: beammr, beamplane, beamstraight.

4.33 *NO ANALYSIS

Keyword type: step
This procedure is used for input deck and geometry checking only. No calculation is performed. There are no parameters.

First and only line:

- *NO ANALYSIS

```
Example:

*NO ANALYSIS
```

requests the no analysis procedure, in which the set of equations is built but not solved (the Jacobian determinant is checked).

Example files: beamnoan.

4.34 *NODAL THICKNESS

Keyword type: model definition
This option is used to assign a thickness to a node or to a node set. There are no parameters. This keyword only makes sense for nodes belonging to plane stress elements, axisymmetric elements, shell elements and beam elements. For all of these except for the beam elements one thickness value should be given. For plane stress and shell elements this is the thickness in normal direction. The normal direction can be defined by using the *NORMAL keyword card. If none is defined, the normal is calculated based on the geometrical data. For axisymmetric elements the thickness is defined as the angle of the sector to be modeled. Finally, for beam elements two thicknesses can be defined: one in 1-direction and one in 2-direction. The 1-direction can be defined on the *BEAM SECTION card, the 2-direction by the *NORMAL card.
The *NODAL THICKNESS card takes precedence over any thickness definitions on the *BEAM SECTION or *SHELL SECTION card.

First line:

- *NODAL THICKNESS

Following line:

- Node or set of nodes previously defined
- Thickness 1
- Thickness 2

`Example:`

```
*NODAL THICKNESS
22,0.05,0.08
```

assigns to node 22 the thickness 0.05 and 0.08. Any plane stress or shell element containing node 22 will have a local thickness of 0.05 unit lengths at node 22. Any axisymmetric element containing node 22 will have a local angle of 0.05 radians. Any beam element containing node 22 will have a thickness of 0.05 unit length in local 1-direction and a thickness of 0.08 unit length in local 2-direction.

Example files: shell1.

4.35 *NODE

Keyword type: model definition
This option allows nodes and their coordinates to be defined. The parameter NSET is optional and is used to assign the nodes to a node set. If the set already exists, the nodes are ADDED to the set.

First line:

- *NODE
- Enter the optional parameter, if desired.

Following line:

- node number.
- Value of first coordinate.
- Value of second coordinate.
- Value of third coordinate.

Repeat this line if needed.

```
Example:

*NODE,NSET=Nall
1,0.,0.,0.
2,1.,0.,0.
3,0.,1.,0.
```

defines three nodes with node number one, two and three and rectangular coordinates (0.,0.,0.), (1.,0.,0.) and (0.,1.,0.) respectively.

Example files: beam8t, beamb, beamdy1.

4.36 *NODE FILE

Keyword type: step
This option is used to print selected nodal variables in file jobname.frd for subsequent viewing by CalculiX GraphiX. Displacements (key=U),temperatures (key=NT) or reaction forces (key=RF)can be selected. The selected variables are stored for the complete model.
The first occurrence of an *NODE FILE keyword card within a step wipes out all previous nodal variable selections for file output. If no *NODE FILE card is used within a step the selections of the previous step apply. If there is no previous step, and no element variable selection nor nodal variable selection is made for file output, the displacements (U) are stored by default.
Notice that only values in nodes belonging to elements are stored. Values in nodes not belonging to any element (e.g. the rotational node in a *RIGID BODY option) can only be obtained using *NODE PRINT.
The only optional parameter is FREQUENCY, and applies to nonlinear calculations where a step can consist of several increments. Default is FREQUENCY=1, which indicates that the results of all increments will be stored. FREQUENCY=N with N an integer indicates that the results of every Nth increment will be stored. The final results of a step are always stored. If you only want the final results, choose N very big. The value of N applies to *EL

FILE, *ELPRINT, *NODE FILE and *NODE PRINT. If the FREQUENCY parameter is used for more than one of these keywords with conflicting values of N, the biggest value applies to all.

First line:

- *NODE FILE
- Enter any needed parameters and their values.

Second line:

- Identifying keys for the variables to be printed, separated by kommas.

```
Example:

*NODE FILE,FREQUENCY=2
RF,NT
```

requests the storage of reaction forces and temperatures in the .frd file every second increment.

Example files: beampt, beampo1.

4.37 *NODE PRINT

Keyword type: step
This option is used to print selected nodal variables in file jobname.dat. Displacements (key=U), temperatures (key=NT) or reaction forces (key=RF) can be selected . There are two parameters, FREQUENCY and NSET. The parameter NSET is required, defining the set of nodes for which the displacements should be printed. If this card is omitted, no values are printed. Several *NODE PRINT cards can be used within one and the same step. In that case, the variables selected in any of these *NODE PRINT cards will be printed for all sets. The parameter FREQUENCY is optional, and applies to nonlinear calculations where a step can consist of several increments. Default is FREQUENCY=1, which indicates that the results of all increments will be stored. FREQUENCY=N with N an integer indicates that the results of every Nth increment will be stored. The final results of a step are always stored. If you only want the final results, choose N very big. The value of N applies to *EL FILE, *ELPRINT, *NODE FILE and *NODE PRINT. If the FREQUENCY parameter is used for more than one of these keywords with conflicting values of N, the biggest value applies to all.
The first occurrence of an *NODE FILE keyword card within a step wipes out all previous nodal variable selections for print output. If no *NODE FILE card is used within a step the selections of the previous step apply, if any.

First line:

- *NODE PRINT
- Enter the parameter NSET and its value.

Second line:

- Identifying keys for the variables to be printed, separated by kommas.

```
Example:

*NODE PRINT,NSET=N1
RF
```

requests the storage of the reaction forces in the nodes belonging to (node) set N1 in the .dat file.

Example files: beampkin, beamrb.

4.38 *NORMAL

Keyword type: model definition
With this option a normal can be defined for a (node,element) pair. This only makes sense for shell elements and beam elements. For beam elements the normal direction is the local 2-direction. If no normal is specified in a node it is calculated on basis of the local geometry. If the normal defined by the user has not unit length, it will be normalized. There are no parameters for this keyword card.

First line:

- *NORMAL
- Element number
- Node number
- Global x-coordinate of the normal
- Global y-coordinate of the normal
- Global z-coordinate of the normal

```
Example:

*NORMAL
5,18,0.707,0.,0.707
```

Defines a normal with components (0.707,0.,0.707) in node 18 of element 5.

Example files: shellnor.

4.39 *NSET

Keyword type: model definition
This option is used to assign nodes to a node set. The parameter NSET containing the name of the set is required, whereas the parameter GENERATE (without value) is optional. If present, nodal ranges can be expressed by their initial value, their final value, and an increment. If a set with the same name already exists, it is reopened and complemented. The name of a set is case insensitive. Internally, it is modified into upper case and a 'N' is appended to denote it as node set.

First line:

- *NSET
- Enter any needed parameters and their values.

Following line if the GENERATE parameter is omitted:

- List of nodes and/or sets of nodes previously defined to be assigned to this node set (maximum 16 entries per line).

Repeat this line if needed.
Following line if the GENERATE parameter is included:

- First node in set.
- Last node in set.
- Increment in nodal numbers between nodes in the set. Default is 1.

Repeat this line if needed.

```
Example:

*NSET,NSET=N1
1,8,831,208
*NSET,NSET=N2
100,N1
```

assigns the nodes with number 1, 8, 831 and 208 to (node) set N1 and the nodes with numbers 1, 8, 831, 208 (= set N1) and 100 to set N2.

Example files: segmentm, shell2.

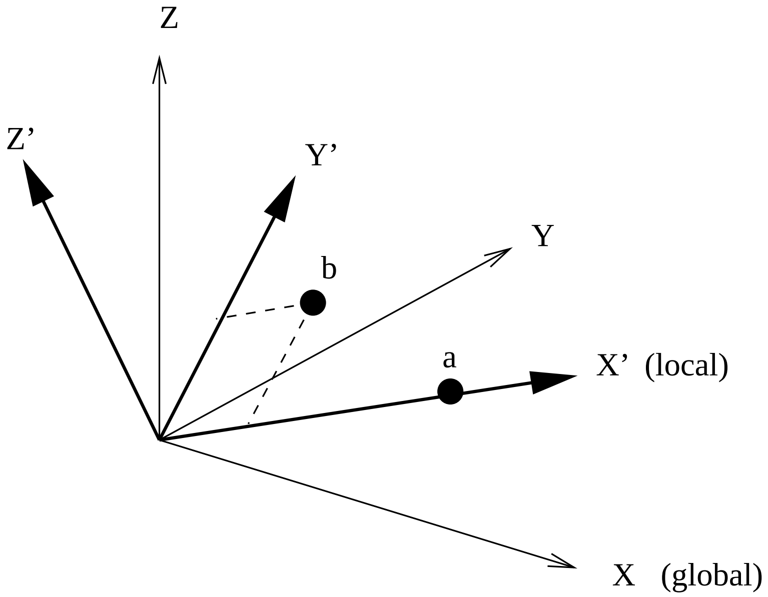

Figure 31: Definition of a rectangular coordinate system

4.40 *ORIENTATION

Keyword type: model definition
This option may be used to specify a local axis system X'-Y'-Z' to be used for defining material properties. For now, rectangular and cylindrical systems can be defined, triggered by the parameter SYSTEM=RECTANGULAR (default) and SYSTEM=CYLINDRICAL.
A rectangular system is defined by specifying a point a on the local X' axis and a point b belonging to the X'-Y' plane but not on the X' axis. A right hand system is assumed (Figure 31).
When using a cylindrical system two points a and b on the axis must be given. The X' axis is in radial direction, the Z' axis in axial direction from point a to point b, and Y' is in tangential direction such that X'-Y'-Z' is a right hand system (Figure 32).
The parameter NAME, specifying a name for the orientation so that it can be used in an element property definition (e.g. *SOLID SECTION) is required.

First line:

- *ORIENTATION
- Enter the required parameter NAME, and the optional parameter SYSTEM if needed.

Second line:

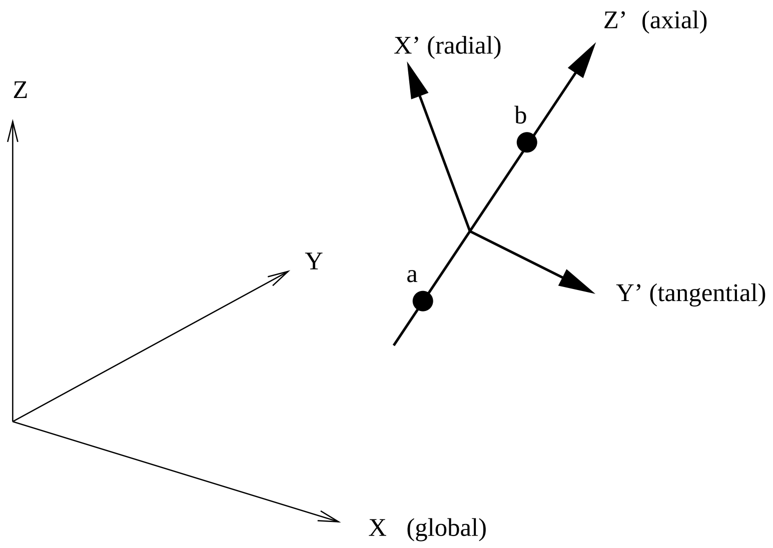

Figure 32: Definition of a cylindrical coordinate system

- X-coordinate of point a.
- Y-coordinate of point a.
- Z-coordinate of point a.
- X-coordinate of point b.
- Y-coordinate of point b.
- Z-coordinate of point b.

Example:

```
*ORIENTATION,NAME=OR1,SYSTEM=CYLINDRICAL
0.,0.,0.,1.,0.,0.
```

defines a cylindrical coordinate system with name OR1 and axis through the points (0.,0.,0.) and (1.,0.,0.). Thus, the x-axis in the global coordinate system is the axial direction in the cylindrical system.

Example files: beampo2.

4.41 *PLASTIC

Keyword type: model definition, material

This option is used to define the plastic properties of an incrementally plastic material. There is one optional parameter HARDENING. Default is HARDENING=ISOTROPIC, other values are HARDENING=KINEMATIC for kinematic hardening, HARDENING=COMBINED for combined isotropic and kinematic hardening and HARDENING=USER for user defined hardening curves. All constants may be temperature dependent. The card should be preceded by a *ELASTIC card within the same material definition, defining the isotropic elastic properties of the material. User defined hardening curves should be defined in the user subroutine uhardening.f

First line:

- *PLASTIC
- Enter the HARDENING parameter and its value, if needed

Following sets of lines define the isotropic hardening curve for HARDENING=ISOTROPIC and the kinematic hardening curve for HARDENING=KINEMATIC or HARDENING=COMBINED: First line in the first set:

- Von Mises stress.
- Equivalent plastic strain.
- Temperature.

Use as many lines in the first set as needed to define the complete hardening curve for this temperature.
Use as many sets as needed to define complete temperature dependence.
For the definition of the isotropic hardening curve for HARDENING=COMBINED the keyword *CYCLIC HARDENING is used.

```
Example:

*PLASTIC
800.,0.,273.
900.,0.05,273.
1000.,0.15,273.
700.,0.,873.
750.,0.04,873.
800.,0.13,873.
```

defines two stress-strain curves: one for temperature T=273. and one for T=873. The curve at T=273 connects the points (800.,0.), (900.,0.05) and (1000.,0.15), the curve at T=873 connects (700.,0.), (750.,0.04) and (800.,0.13). Notice that the first point on the curves represents first yielding and must give the Von Mises stress for a zero equivalent plastic strain.

Example files: beampd, beampiso, beampkin, beampt.

4.42 *RESTART

Keyword type: step
Sometimes you wish to continue a previous run without having to redo the complete calculation. This is where the *RESTART keyword comes in. It can be used to store results for a later restart, or to continue a previous calculation. There is one required parameter specifying whether you want to read previous results (READ) or store the results of the present calculation for future restarts (WRITE). This parameter must follow immediately after the *RESTART keyword card.
If you specify READ, you can indicate with the parameter STEP which step of the previous run is to be read. Default is one. The results will be read from the binary file “jobname.rstrt” which should have been generated in the previous run. A restart file can contain any number of steps and anything which is allowed within a step. For instance, one can define new loads based on sets generated in previous runs.
If you specify WRITE, you can specify the frequency (parameter FREQUENCY) at which results are stored. A frequency of two means that the results of every second step will be stored. Default is one. The results will be stored in binary format in file “jobname.rstrt”.

First and only line:

- *RESTART
- Enter any needed parameters and their values

```
Example:

*RESTART,READ,STEP=2
```

will read the results of step two in the previous calculation.

```
Example:

*RESTART,WRITE,FREQUENCY=3
```

will write the results every third step.

Example files: .

4.43 *RIGID BODY

Keyword type: model definition
With this card a rigid body can be defined consisting of nodes or elements. Optional parameters are REF NODE and ROT NODE.
One of the parameters NSET or ELSET is required. Use NSET to define a rigid body consisting of the nodes belonging to a node set and ELSET for a

rigid body consisting of the elements belonging to an element set. In the latter case, the rigid body really consists of the nodes belonging to the elements. The parameters NSET and ELSET are mutually exclusive. The rigid body definition ensures that the distance between any pair of nodes belonging to the body does not change during deformation. This means that the degrees of freedom are reduced to six: three translational and three rotational degrees of freedom. Thus, the motion is reduced to a translation of a reference node and a rotation about that node.
The reference node can be specified by the parameter REF NODE and should have been assigned coordinates using the *NODE card. The reference node can belong to the rigid body, but does not necessarily have to. For the rotational degrees of freedom a dummy rotational node is used whose translational degrees of freedom are interpreted as the rotations about the reference node. Thus, the first degree of freedom is used as the rotation about the x-axis of the rigid body, the second as the the rotation about the y-axis and the third as the rotation about the z-axix. The rotational node can be defined explicitly using the parameter ROT NODE. In that case, this node must be been assigned coordinates and should not belong to any element of the structure. If a rotational node is specified, a translational node has to be specified as well having a node number not exceeding the rotational node number. Indeed, the specification of a rotational node without specifying a translational node does not make physically sense.
In the absence of any of the parameters REF NODE or ROT NODE, extra nodes are generated internally assuming their tasks. However, defining the nodes explicitly can be useful if a rotation about a specific point is to be defined (using *BOUNDARY or *CLOAD), or if rigid body values (displacements or forces) are to be printed using *NODE PRINT. Notice that a force defined in a rotational node has the meaning of a moment.
Internally, a rigid body is enforced by using nonlinear multiple point constraints (MPC). If NLGEOM is not specified on the *STEP card, the MPC is linearized. Note that depending on the degree of nonlinearity of the MPC linearization can lead to significant errors in the results. It is generally advisable to start with a linear calculation to check the correctness of the structure, loading and boundary conditions. If the linear calculation runs satisfactorily, a geometrically nonlinear calculation should be performed.
If the participating nodes in a rigid body definition lie on a straight line, the rigid body rotation about the line is not defined and an error will occur. To remove the rotational degree of freedom, specify that the rotation about the axis is zero. If $\mathbf{a}$ is a unit normal on the axis and $\mathbf{u}_R$ is the displacement of the ROT NODE, this results in a linear MPC of the form $\mathbf{a}.\mathbf{u}_R = 0$ to be specified by the user by means of a *EQUATION card.

First and only line:

- *RIGID BODY
- Enter any needed parameters and their values

```
Example:

*RIGID BODY,NSET=rigid1,REF NODE=100,ROT NODE=101
```

defines a rigid body consisting of the nodes belonging to node set rigid1 with reference node 100 and rotational node 101.
Using

```
*CLOAD
101,3,0.1
```

in the same input deck (see *CLOAD) defines a moment about the z-axis of 0.1 acting on the rigid body.

Example files: beamrb.

4.44 *SELECT CYCLIC SYMMETRY MODES

Keyword type: step
This option is used to trigger an eigenmode analysis for cyclic symmetric structures. It must be preceded by a *FREQUENCY card. There are two optional parameters NMIN, NMAX. NMIN is the lowest cyclic symmetry mode number (also called nodal diameter)to be considered (default 0), NMAX is the highest cyclic symmetry mode number (default N/2 for N even and (N+1)/2 for N odd, where N is the number of sectors on the *CYCLIC SYMMETRY MODEL card. For models containing the axis of cyclic symmetry (e.g. a full disk), the nodes on the symmetry axis are treated differently depending on whether the cyclic symmetry mode number is 0, 1 or exceeds 1. Therefore, for such structures calculations for cyclic symmetry mode numbers 0 or 1 must be performed in separate steps with NMIN=0,NMAX=0 and NMIN=1,NMAX=1, respectively.

First and only line:

- *SELECT CYCLIC SYMMETRY MODES
- Enter the parameters NMIN and NMAX and their values, if appropriate.

```
Example:

*SELECT CYCLIC SYMMETRY MODES, NMIN=2, NMAX=4
```

triggers a cyclic symmetry calculation for mode numbers 2 up to and including 4.

Example files: segment, fullseg.

4.45 *SHELL SECTION

Keyword type: model definition
This option is used to assign material properties to shell element sets. The parameters ELSET and MATERIAL are required, the parameters ORIENTATION, NODAL THICKNESS and OFFSET is optional. The parameter ELSET defines the shell element set to which the material specified by the parameter MATERIAL applies. The parameter ORIENTATION allows to assign local axes to the element set. If activated, the material properties are applied to the local axis. This is only relevant for non isotropic material behavior. The parameter NODAL THICKNESS indicates that the thickness for nodes in the element set are defined with an extra *NODAL THICKNESS card and that any thicknesses defined on the *SHELL SECTION card are irrelevant. Finally, the OFFSET parameter indicates where the midsurface of the shell should be in relation to the reference surface defined by the surface representation given by the user. The unit of the offset is the thickness of the shell. Thus, OFFSET=0 means that the reference surface is the midsurface of the shell, OFFSET=0.5 means that the reference surface is the top surface of the shell. The offset can take any real value.

First line:

- *SHELL SECTION
- Enter any needed parameters.

Second line (only read if the first line does not contain NODAL THICKNESS):

- thickness

```
Example:

*SHELL SECTION,MATERIAL=EL,ELSET=Eall,ORIENTATION=OR1,OFFSET=-0.5
3.
```

assigns material EL with orientation OR1 to all elements in (element) set Eall. The reference surface is the bottom surface of the shell and the shell thickness is 3 length units.

Example files: shell1, shell2, shellbeam.

4.46 *SOLID SECTION

Keyword type: model definition
This option is used to assign material properties to 3-D, plane stress, plane strain and axisymmetric element sets. The parameters ELSET and MATERIAL are required, the parameter ORIENTATION is optional. The parameter ELSET defines the element set to which the material specified by the parameter MATERIAL applies. The parameter ORIENTATION allows to assign local axes

to the element set. If activated, the material properties are applied to the local axis. This is only relevant for non isotropic material behavior. For plane stress and plane strain elements the thickness can be specified on the second line, for axisymmetric elements the expansion angle can be given. Default is 1.

First line:

- *SOLID SECTION
- Enter any needed parameters.

Second line (only relevant for plane stress, plane strain and axisymmetric elements; can be omitted for 3-D elements):

- thickness (plane stress and plane strain elements) or expansion angle (axisymmetric elements)

```
Example:

*SOLID SECTION,MATERIAL=EL,ELSET=Eall,ORIENTATION=OR1
```

assigns material EL with orientation OR1 to all elements in (element) set Eall.

Example files: beampo2, planestress.

4.47 *STATIC

Keyword type: step
This procedure is used to perform a static analysis. If no perturbation parameter was specified in the *STEP card the load consists of the sum of the load of the last *STATIC step and the load specified in the present step with replacement of redefined loads. If the perturbation parameter was specified (makes sense for *FREQUENCY and *BUCKLE steps only), all loads previous to the step are removed and the only active loads consist of those specified in the present step (for *BUCKLE steps only; for *FREQUENCY steps no loads are taken into account). However, the displacements and stresses at the start of the step are taken into account in the stiffness matrix.
There are two optional parameters: SOLVER and DIRECT. SOLVER determines the package used to solve the ensuing system of equations. Default is SPOOLES, denoting the SPOOLES solver [2, 3]. Alternatively, the PROFILE solver listed in [27] can be selected (SOLVER=PROFILE), or the iterative solver by Rank and Ruecker [20], which is based on the algorithms by Schwarz [21]. If SOLVER=ITERATIVE SCALING is selected, the preconditioning is limited to a scaling of the diagonal terms, SOLVER=ITERATIVE CHOLESKY triggers Incomplete Cholesky preconditioning. Cholesky preconditioning leads to a better convergence and maybe to shorter execution times, however, it requires additional storage roughly corresponding to the nonzeros in the matrix. If you are short of memory, diagonal scaling might be your last resort. The iterative

methods perform well for truely three-dimensional structures. For instance, calculations for a hemisphere were about nine times faster with the ITERATIVE SCALING solver, and three times faster with the ITERATIVE CHOLESKY solver than with SPOOLES. For two-dimensional structures such as plates or shells, the performance might break down drastically and convergence often requires the use of Cholesky preconditioning. SPOOLES performs well in most situations with emphasis on slender structures but requires much more storage than the iterative solver.
The parameter DIRECT is relevant for nonlinear calculations only, and indicates that automatic incrementation should be switched off.

First line:

- *STATIC
- Enter any needed parameters and their values.

Second line (only relevant for nonlinear analyses)

- Initial time increment. This value will be modfied due to automatic incrementation, unless the parameter DIRECT was specified (default 1.).
- Time period of the step (default 1.).
- Minimum time increment allowed. Only active if DIRECT is not specified. Default is the initial time increment or 1.e-5 times the time period of the step, whichever is smaller.
- Maximum time increment allowed. Only active if DIRECT is not specified. Default is 1.e+30.

```
Example:

*STATIC,DIRECT
.1,1.
```

defines a static step and selects the SPOOLES solver as linear equation solver in the step (default). If the step is a linear one, the other parameters are of no importance. If the step is nonlinear, the second line indicates that the initial time increment is .1 and the total step time is 1. Furthermore, the parameter DIRECT leads to a fixed time increment. Thus, if successful, the calculation consists of 10 increments of length 0.1.

Example files: beampic, beampis.

4.48 *STEP

Keyword type: step
This card describes the start of a new STEP. PERTURBATION, NLGEOM and INC are the only optional parameters.
The parameter PERTURBATION is allowed for *FREQUENCY and *BUCKLE steps only. If it is specified, the last *STATIC step is taken as reference state and used to calculate the stiffness matrix. This means the inclusion of previous deformations (large deformation stiffness) and the inclusion of previous loads as preloads (stress stiffness), taking the temperatures into account to determine the material properties. The loads active (mechanical and thermal) are those specified in the perturbation step. The displacements and stresses are those corresponding to the eigenmodes. At the end of the step the perturbation load is reset to zero.
The loading active in a nonperturbative step is the accumulation of the loading in all previous steps since but not including the last perturbation step (or, if none has occurred, since the start of the calculation), unless OP=NEW has been specified since.
If NLGEOM is specified, the calculation takes geometrically nonlinear effects into account. To this end a nonlinear strain tensor is used (Lagrangian strain for hyperelastic materials, Eulerian strain for deformation plasticity and the deviatoric elastic left Cauchy-Green tensor for incremental plasticity), the step is divided into increments and a Newton iteration is performed within each increment. Although the internally used stresses are the Piola stresses of the second kind, they are transformed into Cauchy (true) stresses before being printed. In the present version of the program geometrically nonlinear calculations only apply to static calculations, and consequently the *STATIC or *DYNAMIC keyword card should be used within the step. The latter card also allows for the specification of the step size and increment size. The maximum number of increments in the step (for automatic incrementation) can be specified by using the parameter INC (default is 10). Once the NLGEOM parameter has been selected, it remains active in all subsequent static calculations. Analyses involving nonlinear materials automatically trigger the NLGEOM option. Thus, for these types of analysis nonlinear geometric effects are always taken into account.

First and only line:

- *STEP
- Enter any needed parameters and their values

`Example:`

```
*STEP,INC=100
```

starts a step and increases the maximum number of increments to complete the step to 100.

Example files: beamnlp.

4.49 *SURFACE

Keyword type: model definition
This option is used to assign nodes to a surface. The parameter NAME containing the name of the surface is required. At present, surfaces are used to establish cyclic symmetry conditions. Internally, surfaces are treated as node sets.

First line:

- *SURFACE
- Enter the parameter NAME and its value.

Following line:

- Node or node set to be assigned to this surface (maximum 1 entry per line).

Repeat this line if needed.

```
Example:

*SURFACE,NAME=left
part,
1,
8
```

assigns the nodes with number 1, and 8 and the nodes belonging to node set part to a surface with name left.

Example files: segment, fullseg.

4.50 *TEMPERATURE

Keyword type: step
This option is used to define temperatures and, for shell and beam elements, temperature gradients within a *STEP definition. Optional parameter are OP and AMPLITUDE. OP can take the value NEW or MOD. OP=MOD is default and implies that thermal load in different nodes is accumulated over all steps starting from the last perturbation step. Specifying the temperature for a node for which a temperature was defined in a previous step replaces this last value. OP=NEW implies that the temperatures are reinitialised to the initial values. If multiple *TEMPERATURE cards are present in a step this parameter takes effect for the first *TEMPERATURE card only.
For shell elements a temperature gradient can be defined in addition to a temperature. The temperature applies to nodes in the reference surface, the gradient acts in normal direction. For beam elements two gradients can be defined: one in 1-direction and one in 2-direction. Default for the gradients is zero.

The AMPLITUDE parameter allows for the specification of an amplitude by which the difference between the actual and initial temperature is scaled (mainly used for dynamic calculations). Thus, in that case the values entered on the *TEMPERATURE card are interpreted as reference values to be multiplied with the (time dependent) amplitude value to obtain the actual value. At the end of the step the reference value is replaced by the actual value at that time, for use in subsequent steps.
Temperature gradients are not influenced by the AMPLITUDE parameter.

First line:

- *TEMPERATURE

Following line:

- Node number or node set label.
- Temperature value at the node.
- Temperature gradient in normal direction (shells) or in 2-direction (beams).
- Temperature gradient in 1-direction (beams).

Repeat this line if needed.

```
Example:

*TEMPERATURE
N1,293.
300,473.
301,473.
302,473.
```

assigns a temperature T=293 to all nodes in (node) set N1, and T=473 to nodes 300, 301 and 302.

Example files: beam8t, beam20t, beamnlt, beamt4.

4.51 *TIE

Keyword type: model definition
This option is used to tie the two surfaces bounding the datum sector in a cyclic symmetry calculation. There is one optional parameter POSITION TOLERANCE. The dependent surface is called the slave surface, the independent surface is the master surface. The user can freely decide which surface he takes as slave and which as master. The surfaces are defined using the *SURFACE as sets of nodes. Nodes belonging to the dependent surface cannot be used as dependent nodes in other SPC's or MPC's. Only nodes on the axis of cyclic symmetry can belong both to the slave as well as to the master surface. For

each slave node, a master node is determined which matches the slave node within a tolerance specified by the parameter POSITION TOLERANCE after rotation about the cyclic symmetry axis . If this parameter is not used, the closest master node after rotation is taken. Thus, it is important that master and slave node match as well as possible.

First line:

- *TIE

Following line:

- Name of the slave surface.
- Name of the master surface.

```
Example:

*TIE,POSITION TOLERANCE=0.01
left,right
```

defines a datum sector with slave surface left and master surface right, and defines a position tolerance of 0.01 length units.

Example files: segment, fullseg.

4.52 *TRANSFORM

Keyword type: model definition
This option may be used to specify a local axis system X'-Y'-Z' to be used for defining SPC's, MPC's and nodal forces. For now, rectangular and cylindrical systems can be defined, triggered by the parameter TYPE=R (default) and TYPE=C.
A rectangular system is defined by specifying a point a on the local X' axis and a point b belonging to the X'-Y' plane but not on the X' axis. A right hand system is assumed (Figure 33).
When using a cylindrical system two points a and b on the axis must be given. The X' axis is in radial direction, the Z' axis in axial direction from point a to point b, and Y' is in tangential direction such that X'-Y'-Z' is a right hand system (Figure 34).
The parameter NSET, specifying the node set for which the transformation applies, is required.
It is important to note that SPC's and MPC's defined BEFORE a *TRANSFORM card are NOT affected by it. Thus all SPC's and MPC's defined before the first occurence of a *TRANSFORM card are defined in the global Carthesian system. The effect of a *TRANSFORM card extends up to the occurrence of another *TRANSFORM card, at least for the nodes in common.

First line:

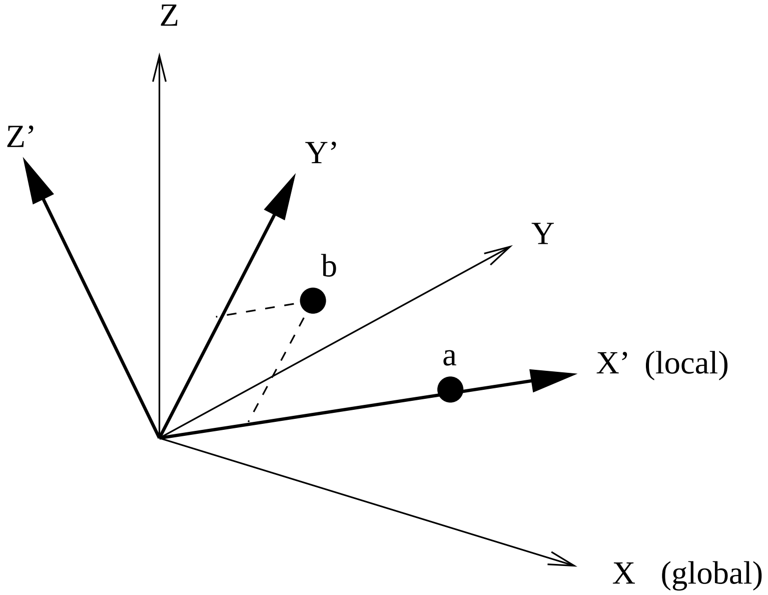

Figure 33: Definition of a rectangular coordinate system

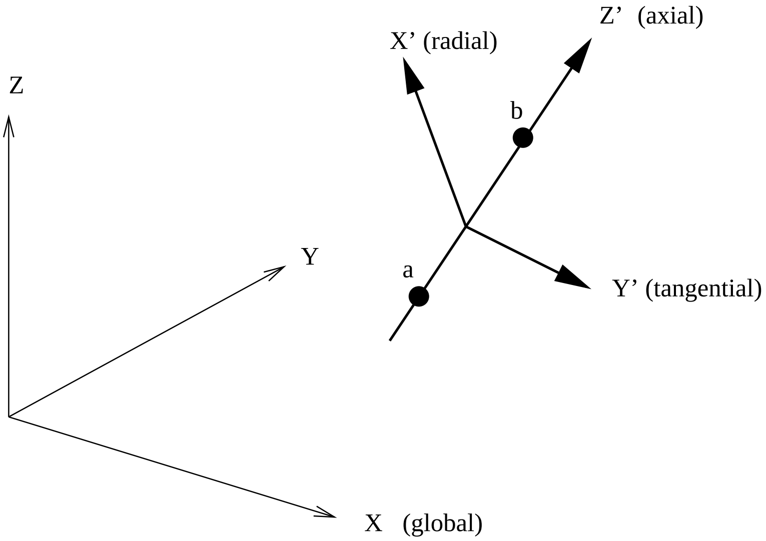

Figure 34: Definition of a cylindrical coordinate system

- *TRANSFORM
- Enter the required parameter NSET, and the optional parameter TYPE if needed.

Second line:

- X-coordinate of point a.
- Y-coordinate of point a.
- Z-coordinate of point a.
- X-coordinate of point b.
- Y-coordinate of point b.
- Z-coordinate of point b.

```
Example:

*TRANSFORM,NSET=No1,TYPE=R
0.,1.,0.,0.,0.,1.
```

assigns a new rectangular coordinate system to the nodes belonging to (node) set No1. The x- and the y-axes in the local system are the y- and z-axes in the global system.

Example files: segment1, segment2, segmentf, segmentm.

4.53 *USER MATERIAL

Keyword type: model definition, material
This option is used to define the properties of a user defined material. There is one required parameter CONSTANTS. The value of this parameter indicates how many material constants are to be defined for this type of material. Right now, there is an upper limit of 21 constants. If you need more, incorporate them in your user subroutine, change the source code, or contact the author to do so. The material is identified by means of the NAME parameter on the *MATERIAL card.

First line:

- *USER MATERIAL
- Enter the CONSTANTS parameter and its value

Give on the following int(CONSTANTS/8)+1 lines the constants followed by the temperature value for which they are valid, 8 values per line. The value of the temperature can be left blank, however, if CONSTANTS is a multiple of 8 a blank line must be provided if the temperature is left blank. Repeat the set of constants if values for more than one temperature are given.

```
Example:

*USER MATERIAL,CONSTANTS=8
500000.,157200.,400000.,157200.,157200.,300000.,126200.,126200.,
294.
300000.,57200.,300000.,57200.,57200.,200000.,26200.,26200.,
394.
```

defines a user defined material with eight constants for two different temperatures, 294 and 394.

Example files: beamu.

5 User subroutines.

Although the present software is protected by the GNU General Public License, and the user should always get the source code, it is sometimes more practical to get a nicely described user interface to plug in your own routines, instead of having to analyze the whole program. Therefore, for specific tasks well-defined interfaces are put at the disposal of the user. These interfaces are basically FORTRAN subroutines containing a subroutine header, a description of the input and output variables and declaration statements for these variables. The body of the routine has to be written by the user.

5.1 creep.

The user subroutine "ucreep.f" is made available to allow the user to incorporate his own creep law by selecting the keyword sequence *CREEP,LAW=USER in the input deck. The creep law gives the Von Mises stress as a function of the equivalent creep strain rate. The header and a description of the input and output variables is as follows:

```
      subroutine ucreep(amat,iel,iint,t1l,epini,ep,dtime,svm,dsvm)

!
!     INPUT:
!
!     amat:  material name
!     iel:   element number
!     iint:  integration point number
!     t1l:   temperature
!     epini: equivalent creep strain at the start
!            of the increment
!     ep:    present equivalent creep strain
!     dtime: time increment
```

```
!
!     OUTPUT:
!
!     svm:   present Von Mises stress
!     dsvm:  derivative of the Von Mises true stress with respect
!            to the present equivalent creep strain.
!            Numerically: change the present equivalent
!            strain with a small amount, calculate the amount
!            of change this causes in the present Von Mises
!            true stress, and divide the latter amount through the
!            former amount.
```

5.2 hardening.

In subroutine “uhardening.f”, the user can insert his own isotropic and/or kinematic hardening laws for (visco)plastic behavior governed by the keyword sequence *PLASTIC,HARDENING=USER. The header and variable description is as follows:

```
      subroutine uhardening(amat,iel,iint,t1l,epini,ep,dtime,fiso,dfiso,
     &                     fkin,dfkin)
!
!     INPUT:
!
!     amat:    material name (maximum 20 characters)
!     iel:     element number
!     iint:    integration point number
!     t1l:     temperature at the end of the increment
!     epini:   equivalent irreversible strain at the start
!              of the increment
!     ep:      present equivalent irreversible strain
!     dtime:   time increment
!
!     OUTPUT:
!
!     fiso:    present isotropic hardening Von Mises stress
!     dfiso:   present isotropic hardening tangent (derivative
!              of the Von Mises stress with respect to the
!              equivalent irreversible strain)
!     fkin:    present kinematic hardening Von Mises stress
!     dfkin:   present kinematic hardening tangent (derivative
!              of the Von Mises stress with respect to the
!              equivalent irreversible strain)
!
```

5.3 user-defined material laws.

This is an extremely important and powerful interface, allowing the user to define his/her own material. The subroutine "umat.f" is a driver subroutine, calling user defined routines similar to "umat_user.f", depending on the kind of material present in the model. To create a new material law, a "umat_user.f" routine must be written and an appropriate call must be inserted in routine "umat.f". In "umat.f" the name of the user material is to be defined. This is a character string the NAME parameter following the keyword card *MATERIAL has to start with. For instance, if you define a new material with the name FUNNY_MATERIAL, then in order to use this material, the material name has to start with this string. This is the main difference in usage between predefined and user defined materials in CalculiX: if you use predefined materials you are completely free to choose a name for your material, if you use a user defined material, its name has to start with a predefined string. Since a material name can be up to 20 characters long, there is generally enough freedom to define several versions of this material, e.g. FUNNY_MATERIAL1, FUNNY_MATERIAL2 etc.
The header and input/output variables of the umat_user routine are as follows:

```
      subroutine umat_user(amat,iel,iint,kode,elconloc,eloc,eth,
     &        beta,xokl,voj,xkl,vj,ithermal,t1l,dtime,icmd,ielas,mint_,
     &        nstate_,xstateini,xstate,stre,stiff,iorien,pgauss,orab)
!
!     calculates stiffness and stresses for a user defined material
!     law
!
!     icmd=3: calcutates stress at mechanical strain
!     else: calculates stress at mechanical strain and the stiffness
!           matrix
!
!     INPUT:
!
!     amat               material name
!     iel                element number
!     iint               integration point number
!
!     kode               material type (-100-#of constants entered
!                        under *USER MATERIAL): can be used for materials
!                        with varying number of constants
!
!     elconloc(21)       user defined constants defined by the keyword
!                        card *USER MATERIAL (max. 21, actual # =
!                        -kode-100), interpolated for the
!                        actual temperature t1l
!
```

```
!     eloc(6)            Lagrange strain tensor (component order:
!                        11,22,33,12,13,23)
!     eth(6)             thermal strain tensor
!     beta(6)            residual stress tensor (the stress entered under
!                        the keyword *INITIAL CONDITIONS,TYPE=STRESS)
!
!     xokl(3,3)          deformation gradient at the start of the increment
!     voj                Jacobian at the start of the increment
!     xkl(3,3)           deformation gradient at the end of the increment
!     vj                 Jacobian at the end of the increment
!
!     ithermal           0: no thermal effects are taken into account
!                        1: thermal effects are taken into account (triggered
!                        by the keyword *INITIAL CONDITIONS,TYPE=TEMPERATURE)
!     t1l                temperature at the end of the increment
!     dtime              time length of the increment
!
!     icmd               not equal to 3: calculate stress and stiffness
!                                        at mechanical strain
!                        3: calculate only stress at mechanical strain
!     ielas              0: no elastic iteration: irreversible effects
!                        are allowed
!                        1: elastic iteration, i.e. no irreversible
!                           deformation allowed
!
!     mint_              max. # of integration points per element in the
!                        model
!     nstate_            max. # of state variables in the model
!
!     xstateini(nstate_,mint_,# of elements)
!                        state variables at the start of the increment
!     xstate(nstate_,mint_,# of elements)
!                        state variables at the end of the increment
!
!     stre(6)            Piola-Kirchhoff stress of the second kind
!                        at the start of the increment
!
!     iorien             number of the local coordinate axis system
!                        in the integration point at stake (takes the value
!                        0 if no local system applies)
!     pgauss(3)          global coordinates of the integration point
!     orab(7,*)          description of all local coordinate systems.
!                        If a local coordinate system applies the global
!                        tensors can be obtained by premultiplying the local
!                        tensors with skl(3,3). skl is  determined by calling
!                        the subroutine transformatrix:
```

```
!                        call transformatrix(orab(1,iorien),pgauss,skl)
!
!     OUTPUT:
!
!     xstate(nstate_,mint_,# of elements)
!                        updated state variables at the end of the increment
!     stre(6)            Piola-Kirchhoff stress of the second kind at the
!                        end of the increment
!     stiff(21):         consistent tangent stiffness matrix in the material
!                        frame of reference at the end of the increment. In
!                        other words: the derivative of the PK2 stress with
!                        respect to the Lagrangian strain tensor. The matrix
!                        is supposed to be symmetric, only the upper half is
!                        to be given in the same order as for a fully
!                        anisotropic elastic material (*ELASTIC,TYPE=ANISO).
!
```

The parameter ielas indicates whether irreversible effects should be taken into account. Forced displacements can lead to huge strains in the first iteration. Therefore, convergence in quasistatic calculations is often enhanced if the first iteration is completely linear, i.e. material and geometric nonlinearities are turned off. The parameter ielas is the appropriate flag.
Two extra routines are at the user's disposal for conversion purposes. "str2mat.f" can be used to convert Lagrangian strain into Eulerian strain, Cauchy stress into PK2 stress, or Kirchhoff stress into PK2 stress. The header and a short description are as follows:

```
      subroutine str2mat(str,ckl,vj,cauchy)
!
!     converts the stress in spatial coordinates into material coordinates
!     or the strain in material coordinates into spatial coordinates.
!
!     INPUT:
!
!     str(6):     Cauchy stress, Kirchhoff stress or Lagrange strain
!                 component order: 11,22,33,12,13,23
!     ckl(3,3):   the inverse deformation gradient
!     vj:         Jakobian determinant
!     cauchy:     logical variable
!                 if true: str contains the Cauchy stress
!                 if false: str contains the Kirchhoff stress or
!                           Lagrange strain
!
!     OUTPUT:
```

```
!
!     str(6):     Piola-Kirchhoff stress of the second kind (PK2) or
!                 Euler strain
!
```

The second routine, “stiff2mat.f” converts the tangent stiffness matrix from spatial coordinates into material coordinates.

```
      subroutine stiff2mat(elas,ckl,vj,cauchy)
!
!     converts an element stiffness matrix in spatial coordinates into
!     an element stiffness matrix in material coordinates.
!
!     INPUT:
!
!     elas(21):   stiffness constants in the spatial description, i.e.
!                 the derivative of the Cauchy stress or the Kirchhoff
!                 stress with respect to the Eulerian strain
!     ckl(3,3):   inverse deformation gradient
!     vj:         Jacobian determinant
!     cauchy:     logical variable
!                 if true: elas is written in terms of Cauchy stress
!                 if false: elas is written in terms of Kirchhoff stress
!
!     OUTPUT:
!
!     elas(21):   stiffness constants in the material description,i.e.
!                 the derivative of the second Piola-Kirchhoff stress (PK2)
!                 with respect to the Lagrangian strain
!
```

Now, some examples are given.

5.3.1 Fiber reinforced materials.

This is a model which was conceived by G. Holzapfel et al. [9] to model arterial walls. It is an anisotropic hyperelastic model, consisting of an isotropic neo-Hooke potential for the base material, complemented by exponential strenghtening terms in fiber direction. The mathematical form of the potential satisfies:

$$U = C_{10}(\bar{I}_1 - 3) + \frac{1}{D_1}(J-1)^2 + \sum_{i=1}^{n} \frac{k_{1i}}{2k_{2i}} \left[e^{k_{2i}\langle \bar{J}_{4i} - 1 \rangle^2} - 1 \right] \tag{25}$$

where $\langle x \rangle = 0$ for $x < 0$ and $\langle x \rangle = x$ for $x \geq 0$. Thus, the fibers do not take up any force under compression. Although the material was originally defined

for arteries, it is expected to work well for other fiber reinforced materials too, such as reinforced nylon. The material model implemented thus far can cope with up to 4 different fibers. The material definition consists of a *MATERIAL card defining the name of the material. This name HAS TO START WITH "ELASTIC_FIBER" but can be up to 20 characters long. Thus, the last 7 characters can be freely chosen by the user. Within the material definition a *USER MATERIAL card has to be used satisfying:
First line:

- *USER MATERIAL
- Enter the CONSTANTS parameter and its value. The value of this parameter is 2+4n, where n is the number of fiber directions.

Following line if one fiber direction is selected:

- C_{10}.
- D_1.
- n_{x1}: x-direction cosine of fiber direction.
- n_{y1}: y-direction cosine of fiber direction.
- k_{11}.
- k_{21}.
- Temperature.

Repeat this line if needed to define complete temperature dependence. The z-direction cosine of the fiber direction is determined from the x- and y-direction cosine since the direction norm is one. If a local axis system is defined for an element consisting of this material (with *ORIENTATION)the direction cosines are defined in the local system.
If more than one fiber direction is selected (up to a maximum of four), the four entries characterizing fiber direction 1 are repeated for the subsequent directions. Per line no more than eight entries are allowed. If more are needed, continue on the next line.

```
Example:

*MATERIAL,NAME=ELASTIC_FIBER
*USER MATERIAL,CONSTANTS=18
1.92505,0.026,0.,0.7071,2.3632,0.8393,0,-0.7071,
2.3632,0.8393,0.7071,0.,2.3632,0.8393,-0.7071,0.,
2.3632,0.8393
```

defines an elastic fiber materials with four different fiber directions (0,0.7071,0.7071), (0,-0.7071,0.7071), (0.7071,0.,0.7071) and (-0.7071,0.,0.7071). The constants are $C_{10} = 1.92505$, $D_1 = 0.026$ and $k_{1i} = 2.3632$, $k_{2i} = 0.8393 \; \forall \; i \; \in \; \{1, 2, 3, 4\}$.

5.3.2 The Cailletaud single crystal model.

The single crystal model of Georges Cailletaud and co-workers [17][18] describes infinitesimal viscoplasticity in metallic components consisting of one single crystal. The orientations of the slip planes and slip directions in these planes is generally known and described by the normal vectors $\mathbf{n}^\beta$ and direction vectors $\mathbf{l}^\beta$, respectively, where β denotes one of slip plane/slip direction combinations. The slip planes and slip directions are reformulated in the form of a slip orientation tensor $\mathbf{m}^\beta$ satisfying:

$$\mathbf{m}^\beta = (\mathbf{n}^\beta \otimes \mathbf{l}^\beta + \mathbf{l}^\beta \otimes \mathbf{n}^\beta)/2. \tag{26}$$

The total strain is supposed to be the sum of the elastic strain and the plastic strain:

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon^e} + \boldsymbol{\epsilon^p}. \tag{27}$$

In each slip plane an isotropic hardening variable q_1 and a kinematic hardening variable q_2 are introduced representing the isotropic and kinematic change of the yield surface, respectively. The yield surface for orientation β takes the form:

$$h^\beta := \left|\boldsymbol{\sigma} : \mathbf{m}^\beta + q_2^\beta\right| - r_0^\beta + \sum_{\alpha=1}^{n^\beta} H_{\beta\alpha} q_1^\alpha = 0 \tag{28}$$

where n^β is the number of slip orientations for the material at stake, $\boldsymbol{\sigma}$ is the stress tensor, r_0^β is the size of the elastic range at zero yield and $H_{\beta\alpha}$ is a matrix of interaction coefficients. The constitutive equations for the hardening variables satisfy:

$$q_1^\beta = -b^\beta Q^\beta \alpha_1^\beta \tag{29}$$

and

$$q_2^\beta = -c^\beta \alpha_2^\beta \tag{30}$$

where α_1^β and α_2^β are the hardening variables in strain space. The constitutive equation for the stress is Hooke's law:

$$\boldsymbol{\sigma} = \boldsymbol{C} : \boldsymbol{\epsilon^e}. \tag{31}$$

The evolution equations for the plastic strain and the hardening variables in strain space are given by:

$$\dot{\boldsymbol{\epsilon}}^{\boldsymbol{p}} = \sum_{\beta=1}^{n^\beta} \dot{\gamma}^\beta \mathbf{m}^\beta sgn(\boldsymbol{\sigma} : \mathbf{m}^\beta + q_2^\beta), \tag{32}$$

$$\dot{\alpha}_1^\beta = 1 + \frac{q_1^\beta}{Q^\beta} \tag{33}$$

and

$$\dot{\alpha}_2^\beta = \varphi^\beta sgn(\boldsymbol{\sigma} : \mathbf{m}^\beta) + \frac{d^\beta q_2^\beta}{c^\beta}. \tag{34}$$

The variable $\dot{\gamma}^\beta$ is the consistency coefficient known from the Kuhn-Tucker conditions in optimization theory [14]. It can be proven to satisfy:

$$\dot{\gamma}^\beta = \left|\dot{\epsilon}^{p^\beta}\right|, \tag{35}$$

where $\dot{\epsilon}^{p^\beta}$ is the flow rate along orientation β. The plastic strain rate is linked to the flow rate along the different orientations by

$$\dot{\boldsymbol{\epsilon}}^{\boldsymbol{p}} = \sum_{\beta=1}^{n^\beta} \dot{\epsilon}^{p^\beta} \mathbf{m}^\beta. \tag{36}$$

The parameter φ^β in equation (34) is a function of the accumulated shear flow in absolute value through:

$$\varphi^\beta = \phi^\beta + (1 - \phi^\beta) e^{-\delta^\beta \int_0^t \dot{\gamma}^\beta dt} \tag{37}$$

Finally, in the Cailletaud model the creep rate is a power law function of the yield exceedance:

$$\dot{\gamma}^\beta = \left\langle \frac{h^\beta}{K^\beta} \right\rangle^{n^\beta}. \tag{38}$$

The brackets $\langle\rangle$ reduce negative function values to zero while leaving positive values unchanged, i.e. $\langle x \rangle = 0$ if $x < 0$ and $\langle x \rangle = x$ if $x \geq 0$.
In the present umat routine, the Cailletaud model is implemented for a Nickel base single crystal. It has two slip systems, a octaeder slip system with three slip directions $< 011 >$ in four slip planes $\{111\}$, and a cubic slip system with two slip directions $< 011 >$ in three slip planes $\{001\}$. The constants for all octaeder slip orientations are assumed to be identical, the same applies for the cubic slip orientations. Furthermore, there are three elastic constants for this material. Consequently, for each temperature 21 constants need to be defined: the elastic constants C_{1111}, C_{1122} and C_{1212}, and a set $\{K^\beta, n^\beta, c^\beta, d^\beta, \phi^\beta, \delta^\beta, r_0^\beta, Q^\beta, b^\beta\}$ per slip system. Apart from these constants 18^2 interaction coefficients need to be defined. These are taken from the references [17][18] and assumed to be constant. Their values are included in the routine and cannot be influence by the user through the input deck.
The material definition consists of a *MATERIAL card defining the name of the material. This name HAS TO START WITH "SINGLE_CRYSTAL" but can be up to 20 characters long. Thus, the last 6 characters can be freely chosen by the user. Within the material definition a *USER MATERIAL card has to be used satisfying:
First line:

- *USER MATERIAL
- Enter the CONSTANTS parameter and its value, i.e. 21.

Following lines, in sets of 3:

First line of set:

- C_{1111}.
- C_{1122}.
- C_{1212}.
- K^β (octaeder slip system).
- n^β (octaeder slip system).
- c^β (octaeder slip system).
- d^β (octaeder slip system).
- ϕ^β (octaeder slip system).

Second line of set:

- δ^β (octaeder slip system).
- r_0^β (octaeder slip system).
- Q^β (octaeder slip system).
- b^β (octaeder slip system).
- K^β (cubic slip system).
- n^β (cubic slip system).
- c^β (cubic slip system).
- d^β (cubic slip system).

Third line of set:

- ϕ^β (cubic slip system).
- δ^β (cubic slip system).
- r_0^β (cubic slip system).
- Q^β (cubic slip system).
- b^β (cubic slip system).
- Temperature.

Repeat this set if needed to define complete temperature dependence.
The crystal principal axes are assumed to coincide with the global coordinate system. If this is not the case, use an *ORIENTATION card to define a local system.
For this model, there are 60 internal state variables:

- the plastic strain tensor $\boldsymbol{\epsilon^p}$ (6)
- the isotropic hardening variables q_1^β (18)
- the kinematic hardening variables q_2^β (18)
- the accumulated absolute value of the slip rate $\int_0^t \dot{\gamma^\beta} dt$ (18)

These variables are accessible through the *EL PRINT (.dat file) and *EL FILE (.frd file) keywords in exactly this order (label SDV). The *DEPVAR card must be included in the material definion with a value of 60.

```
Example:

*MATERIAL,NAME=SINGLE_CRYSTAL
*USER MATERIAL,CONSTANTS=21
135468.,68655.,201207.,1550.,3.89,18.E4,1500.,1.5,
100.,80.,-80.,500.,980.,3.89,9.E4,1500.,
2.,100.,70.,-50.,400.
*DEPVAR
60
```

defines a single crystal with elastic constants $\{135468., 68655., 201207.\}$, octaeder parameters $\{1550., 3.89, 18.E4, 1500., 1.5, 100., 80., -80., 500.\}$ and cubic parameters $\{980., 3.89, 9.E4, 1500., 2., 100., 70., -50., 400.\}$.

5.3.3 Elastically anisotropic material with isotropic viscoplastic behavior.

This model describes small deformations for elastically anisotropic materials with a von Mises type yield surface. Often, this model is used as a compromise for anisotropic materials with lack of data or detailed knowledge about the anisotropic behavior in the viscoplastic range.
The total strain is supposed to be the sum of the elastic strain and the plastic strain:

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon^e} + \boldsymbol{\epsilon^p}. \tag{39}$$

An isotropic hardening variable q_1 and a kinematic hardening tensor $\mathbf{q_2}$ are introduced representing the isotropic and kinematic change of the yield surface, respectively. The yield surface takes the form:

$$f := \|\boldsymbol{dev(\sigma)} + \mathbf{q_2}\| + \sqrt{\frac{2}{3}}(q_1 - r_0) = 0 \tag{40}$$

where $\boldsymbol{dev(\sigma)}$ is the deviatoric stress tensor, and r_0 is the size of the elastic range at zero yield. The constitutive equations for the hardening variables satisfy:

$$q_1 = -d_1\alpha_1 \tag{41}$$

and

$$\mathbf{q_2} = -\frac{2}{3}d_2\boldsymbol{\alpha_2} \tag{42}$$

where α_1 and $\boldsymbol{\alpha_2}$ are the hardening variables in strain space. It can be shown that

$$\alpha_1 = \epsilon^{peq}, \tag{43}$$

$$\alpha_2{}^{eq} = \epsilon^{peq}, \tag{44}$$

where ϵ^{peq} is the equivalent plastic strain defined by

$$\epsilon^{peq} = \sqrt{\frac{2}{3}}\,\|\boldsymbol{\epsilon^p}\|\,. \tag{45}$$

and $\alpha_2{}^{eq}$ is the equivalent value of the tensor $\boldsymbol{\alpha_2}$ defined in a similar way. Thus, the constitutive equations amount to

$$q_1 = -d_1\epsilon^{peq} \tag{46}$$

and

$$q_2{}^{eq} = d_2\epsilon^{peq}, \tag{47}$$

where

$$q_2{}^{eq} = \sqrt{\frac{3}{2}}\,\|\mathbf{q_2}\| \tag{48}$$

has the meaning of an equivalent stress value or von Mises value. The same applies to q_1. Consequently, the constitutive equations assume a linear relationship between the hardening stress and the equivalent plastic strain.
The constitutive equation for the stress is Hooke's law:

$$\boldsymbol{\sigma = C : \epsilon^e.} \tag{49}$$

The evolution equations for the plastic strain and the hardening variables in strain space are given by:

$$\dot{\boldsymbol{\epsilon}}^{\boldsymbol{p}} = \dot{\gamma}\mathbf{n}, \tag{50}$$

$$\dot{\alpha}_1 = \sqrt{\frac{2}{3}}\dot{\gamma}, \tag{51}$$

and

$$\dot{\boldsymbol{\alpha}}_{\mathbf{2}} = \dot{\gamma}\mathbf{n}, \tag{52}$$

where

$$\mathbf{n} = \frac{\boldsymbol{dev(\sigma)} + \mathbf{q_2}}{||\boldsymbol{dev(\sigma)} + \mathbf{q_2}||}. \tag{53}$$

The variable $\dot{\gamma}$ is the consistency coefficient known from the Kuhn-Tucker conditions in optimization theory [14]. It can be proven to satisfy:

$$\dot{\gamma} = \sqrt{\frac{3}{2}}\dot{\epsilon}^{peq}, \tag{54}$$

Finally, the creep rate is modeled as a power law function of the yield exceedance:

$$\epsilon^{\dot{peq}} = A\left\langle\sqrt{\frac{3}{2}}f\right\rangle^n. \tag{55}$$

The brackets $\langle\rangle$ reduce negative function values to zero while leaving positive values unchanged, i.e. $\langle x\rangle = 0$ if $x < 0$ and $\langle x\rangle = x$ if $x \geq 0$.
In the present implementation orthotropic elastic behavior is assumed. Consequently, for each temperature 14 constants need to be defined: the elastic constants C_{1111}, C_{1122}, C_{2222},C_{1133}, C_{2233}, C_{3333},C_{1212}, C_{1313}, C_{2323}, and the viscoplastic constants r_0, d_1, d_2, A, n.
The material definition consists of a *MATERIAL card defining the name of the material. This name HAS TO START WITH "ANISO_PLAS" but can be up to 20 characters long. Thus, the last 10 characters can be freely chosen by the user. Within the material definition a *USER MATERIAL card has to be used satisfying:
First line:

- *USER MATERIAL
- Enter the CONSTANTS parameter and its value, i.e. 14.

Following lines, in sets of 2:

First line of set:

- C_{1111}.

- C_{1122}.
- C_{2222}.
- C_{1133}.
- C_{2233}.
- C_{3333}.
- C_{1212}.
- C_{1313}.

Second line of set:

- C_{2323}.
- r_0.
- d_1.
- d_2.
- A.
- n.
- Temperature.

Repeat this set if needed to define complete temperature dependence.
The principal axes of the material are assumed to coincide with the global coordinate system. If this is not the case, use an *ORIENTATION card to define a local system.
For this model, there are 14 internal state variables:

- the equivalent plastic strain ϵ^{peq} (1)
- the plastic strain tensor $\boldsymbol{\epsilon^p}$ (6)
- the isotropic hardening variable α_1 (1)
- the kinematic hardening tensor $\boldsymbol{\alpha_2}$ (6)

These variables are accessible through the *EL PRINT (.dat file) and *EL FILE (.frd file) keywords in exactly this order (label SDV). The *DEPVAR card must be included in the material definition with a value of 14.

```
Example:

*MATERIAL,NAME=ANISO_PLAS
*USER MATERIAL,CONSTANTS=14
500000.,157200.,500000.,157200.,157200.,500000.,126200.,126200.,
126200.,0.,0.,0.,1.E-10,5
*DEPVAR
14
```

defines a single crystal with elastic constants 500000., 157200., 500000., 157200., 157200., 500000., 126200., 126200., 126200., and viscoplastic parameters $r_0 = 0.$, $d_1 = 0.$, $d_2 = 0.$, $A = 10^{-10}$ and $n = 5$. Thus, the yield surface has a zero radius and there is no hardening. Only creep is activated.

5.4 user-defined nonlinear equations

This user subroutine allows the user to insert his/her own nonlinear equations (also called Multiple Point Constraints or MPC's). The driver routine is "nonlinmpc.f". For each new type of equation the user can define a name, e.g. FUN (maximum length 20 characters). To be consistent, the user subroutine should be called umpc_fun and stored in "umpc_fun.f". In file "nonlinmpc.f" the lines

```
            elseif(labmpc(ii)(1:4).eq.'USER') then
               call umpc_user(aux,aux(3*maxlenmpc+1),const,
     &            aux(6*maxlenmpc+1),iaux,n)
```

should be duplicated and user (USER) replaced by fun (FUN).
It is assumed that the nonlinear equation is a function of the displacements only. Then it can generally be written as

$$f(u_1, u_2, u_3, ..., u_n) = 0 \tag{56}$$

where u_i represents the displacement in node n_i in direction l_i. Nonlinear equations are solved by approximating them linearly and using an iterative procedure. It is the linearization which must be provided by the user in the subroutine. Assume we arrived at an itermediate solution $u_1^0, u_2^0, ...u_n^0$. Then the above equation can be linearly approximated by:

$$f(u_1^0, u_2^0, ..., u_n^0) + \sum_{i=1}^{i=n} \left.\frac{df}{du_i}\right|_0 (u_i - u_i^0) \tag{57}$$

To use a user-defined equation its name must be specified on the line beneath the keyword *MPC, followed by a list of all the nodes involved in the MPC. This list of nodes is transferred to the user routine, as specified by the following header and input/output variables of the umpc_user routine:

```
      subroutine umpc_user(x,u,f,a,jdof,n)
!
!     updates the coefficients in a user mpc
!
!     INPUT:
!
!     x(3,n)             Carthesian coordinates of the nodes in the
!                               user mpc.
!     u(3,n)                    Actual displacements of the nodes in the
!                               user mpc.
!     n                         number of terms in the user mpc
!
!     OUTPUT:
!
!     f                         Actual value of the mpc. If the mpc is
!                               exactly satisfied, this value is zero
!     a(n)                      coefficients of the linearized mpc
!     jdof(n)                   degrees of freedom of the mpc terms
!
```

The subroutine returns the value of f ($f(u_1^0, u_2^0, ..., u_n^0)$), the coefficients of the linearization ($\left.\frac{df}{du_i}\right|_0$) and the degrees of freedom involved. An example is given next.

5.4.1 Mean rotation MPC.

This MPC is used to apply a rotation to a set of nodes. The rotation is characterized by its size (angle in radians) and its axis (normal vector). All nodes participating in the rotation should be listed three times (once for each DOF). The user must define an extra node at the end in order to define the size and axis of rotation: the coordinates of the extra node are the components of a vector on the rotation axis, the first DOF of the node is interpreted as the size of the rotation. This size can be defined using a boundary card. Applying a mean rotation implies that the mean of the rotation of all participating nodes amounts to a given value, but not the individual rotations per se.

```
Example:

*NODE
162,0.,1.,0.
*MPC
MEANROT,3,3,3,2,2,2,14,14,14,39,39,39,42,42,42,
50,50,50,48,48,48,162
..
*STEP
*STATIC
*BOUNDARY
162,1,1,.9
..
*END STEP
```

specifies a mean rotation MPC. Its size is 0.9 radians = 51.56° and the global y-axis is the rotation axis. The participating nodes are 3,2,14,39,42,50 and 48.

6 Verification examples.

The verification examples are simple examples suitable to test distinct features. They can be used to check whether the installation of CalculiX is correct, or to find examples when using a new feature. Here, they are listed alphabetically with a short description of what is being tested. For the input files, append ".inp", for the result file, append ".dat.ref". All files are contained in the distribution.

6.1 achtel2

```
Structure: cube.
Test objective: equations with 2 terms.
```

6.2 achtel29

```
Structure: cube.
Test objective: mixture of equations with 2 and 9 terms.
```

6.3 achtel9

```
Structure: cube.
Test objective: equations with 9 terms.
```

6.4 achtelc

```
Structure: cube.
Test objective: centrifugal forces.
```

6.5 achtelcas

```
Structure: cube.
Test objective: cascaded equations.
```

6.6 achteld

```
Structure: cube.
Test objective: prescribed displacements.
```

6.7 achtelg

```
Structure: cube.
Test objective: gravity load.
```

6.8 achtelp

```
Structure: cube.
Test objective: point loads.
```

6.9 aniso

```
Structure: cantilever beam.
Test objective: fully anisotropic material.
```

6.10 beam

```
Structure: cantilever beam subject to point loads;
Test objective: only 8 elements; leads to hourglassing!
```

6.11 beam10p

```
Structure: cantilever beam under pressure.
Test objective: C3D10 elements.
```

6.12 beam20p

```
Structure: cantilever beam under shear forces.
Test objective: element type C3D20.
```

6.13 beam20t

```
Structure: heated beam fixed in between two walls.
Test objective: element type C3D20.
```

6.14 beam8b

```
Structure: beam fixed at one end and compressed on
           the other end.
Test objective: *BUCKLING option with C3D8 elements;
                beamb uses C3D20 elements.
```

6.15 beam8f

```
Structure: cantilever beam.
Test objective: Calculation of eigenfrequencies and
                eigenmodes with C3D8 elements.
```

6.16 beam8p

```
Structure: cantilever beam under shear forces.
Test objective: C3D8 elements.
```

6.17 beam8rp

```
Structure: cantilever beam under self weight.
Test objective: C3D8R elements; HOURGLASSING occurs!
```

6.18 beam8t

```
Structure: heated cantilever beam consisting of 2
           different materials
Test objective: C3D8 elements.
```

6.19 beamb

```
Structure: beam fixed at one end and compressed
           on the other end.
Test objective:  *BUCKLING option; comparable with beamf2.
```

6.20 beamcr

```
Structure: Cantilever beam under tensile forces
Test objective: Material card *CREEP
```

6.21 beamd

```
Structure: cantilever beam under tension.
Test objective: distributed loads.
```

6.22 beamdy1

```
Structure: cantilever beam.
Test objective: dynamic response to a constant impact;
                no damping.
```

6.23 beamdy2

```
Structure: cantilever beam.
Test objective: dynamic response to highly transient loading;
                no damping.
```

6.24 beamdy3

```
Structure: cantilever beam.
Test objective:  dynamic response to a constant impact;
                 Rayleigh damping is active: alpha=5000.,
                 beta=0.: subkritical for all modes.
```

6.25 beamdy4

```
Structure: cantilever beam.
Test objective: dynamic response to a constant impact;
                Rayleigh damping is active: alpha=0.,
                beta=2.e-4: supercritical for all modes.
```

6.26 beamdy5

```
Structure: cantilever beam.
Test objective: dynamic response to highly transient loading;
                Rayleigh damping is active: alpha=5000.,
                beta=0.: subkritical for all modes.
```

6.27 beamdy6

```
Structure: cantilever beam.
Test objective: dynamic response to highly transient loading;
                Rayleigh damping is active: alpha=0.,
                beta=2.e-4: supercritical for all modes.
```

6.28 beamf

```
Structure: cantilever beam.
Test objective: eigenfrequencies and eigenmodes.
```

6.29 beamf2

```
Structure: beam under compressive forces.
Test objective: Frequency analysis; the forces are that
                high that the lowest frequency is nearly
                zero, i.e. the buckling load is reached.
```

6.30 beamft

```
Structure: cantilever beam.
```

```
Test objective: eigenfrequencies and eigenmodes calculated in a
                perturbation step following a static step with
                temperature loading only. Due to the zero
                expansion coefficient the loading creates
                displacements nor stresses, and the change of
                frequencies compared to a nonperturbative step
                is only due to the lower Young's modulus at high
                temperature.
```

6.31 beamhf

```
Structure: cantilever beam.
Test objective: hyperfoam material (N=2) under tension.
```

6.32 beamnh

```
Structure: cantilever beam.
Test objective: Neo-Hooke material under tension.
```

6.33 beamnld

```
Structure: cantilever beam.
Test objective: axial nonzero displacements;
nonlinear geometric calculation.
```

6.34 beamnldy

```
Structure: cantilever beam.
Test objective: nonlinear dynamic response to a constant
                impact; implicit procedure.
```

6.35 beamnldye

```
Structure: cantilever beam.
Test objective: nonlinear dynamic response to a constant
                impact; no damping; explicit procedure.
```

6.36 beamnldyp

```
Structure: Cantilever beam under tensile forces
Test Objective: Nonlinear dynamic response of a plastic
                material; implicit procedure
```

6.37 beamnldype

```
Structure: Cantilever beam under tensile forces
Test Objective: Nonlinear dynamic response of a plastic material;
                material; explicit procedure
```

6.38 beamnlmpc

```
Structure: cantilever beam under shear forces.
Test objective: MPC's in geometrically nonlinear calculations.
```

6.39 beamnlp

```
Structure: cantilever beam under shear forces.
Test objective: geometrically nonlinear calculation.
```

6.40 beamnlt

```
Structure: cantilever beam under temperature loading.
Test objective: geometrically nonlinear calculation.
```

6.41 beamog

```
Structure: cantilever beam.
Test objective: Ogden material (N=1) under tension
```

6.42 beamp

Structure: cantilever beam.
Test objective: shear forces.

6.43 beampd

Structure: cantilever beam under forced displacements.
Test objective: plasticity with kinematic hardening.

6.44 beampic

Structure: cantilever beam under shear forces.
Test objective: check of the iterative solver with Cholesky preconditioning

6.45 beampis

Structure: cantilever beam under shear forces.
Test objective: check of the iterative solver with diagonal scaling.

6.46 beampiso

Structure: cantilever beam loaded by tensile forces.
Test objective: plasticity with isotropic hardening.

6.47 beampkin

Structure: cantilever beam loaded by tensile forces.
Test objective: plasticity with kinematic hardening.

6.48 beampl

```
Structure: cantilever beam under tension.
Test objective: deformation plasticity.
```

6.49 beampo1

```
Structure: cantilever beam.
Test objective: orthotropic material.
```

6.50 beampo2

```
Structure: cantilever beam.
Test objective: *ORIENTATION card.
```

6.51 beampt

```
Structure: cantilever beam loaded by temperature.
Test objective: plasticity with isotropic hardening.
```

6.52 beamt

```
 Structure: heated beam between two fixed walls.
 Test objective: *EXPANSION.
```

6.53 beamt2

```
Structure: heated cantilever beam.
Test objective: 2 different materials.
```

6.54 beamt3

```
Structure: cantilever beam.
Test objective: arbitrary temperature field;
```

material properties are not temperature dependent.

6.55 beamt4

Structure: cantilever beam.
Test objective: varying temperature field.
Material properties are temperature dependent.

6.56 beamu

Structure: hinged beam.
Test objective: umat routine.

6.57 fullseg

Structure: disk segment
Test objective: cyclic symmetry for a structure containing the
axis of cyclic symmetry for nodal diameter 1

6.58 segment

Structure: disk segment
Test objective: cyclic symmetry
output of two sectors

6.59 segment1

Structure: disk segment
Test objective: SPC's in cylindrical coordinates;
no cascading.

6.60 segment2

Structure: disk segment

```
Test objective: SPC's in cylindrical coordinates;
                leads to cascaded MPC's
```

6.61 segmentf

```
Structure: disk segment
Test objective: Force application in cylindrical coordinates;
```

6.62 segmentm

```
Structure: disk segment
Test objective: MPC's in cylindrical coordinates;
                leads to cascaded MPC's
```

References

[1] ABAQUS Theory Manual. Hibbitt, Karlsson & Sorensen, Inc., 1080 Main Street, Pawtucket, RI 02860-4847, U.S.A. (1997).

[2] Ashcraft, C., Grimes, R.G., Pierce, D.J., and Wah, D.K., The User Manual for SPOOLES, Release 2.0: An object oriented software library for solving sparse linear systems of equations. Boeing Shared Services Group, P.O. Box 24346, Mail Stop 7L-22, Seattle, Washington 98124 U.S.A. (1998).

[3] Ashcraft, C. and Wah, D.K., The Reference Manual for SPOOLES, Release 2.0: An object oriented software library for solving sparse linear systems of equations. Boeing Shared Services Group, P.O. Box 24346, Mail Stop 7L-22, Seattle, Washington 98124 U.S.A. (1998).

[4] Barlow, J., Optimal stress locations in finite element models. *Int. J. Num. Meth. Engng.* **10** , 243-251 (1976).

[5] Beatty, M.F., Topics in finite elasticity: hyperelasticity of rubber, elastomers, and biological tissues - with examples. *Appl. Mech. Rev.* **40(12)** , 1699-1734 (1987).

[6] Belytschko, T., Liu, W.K. and Moran, B., *Nonlinear Finite Elements for Continua and Structures.* John Wiley & Sons, New York (2001).

[7] Ciarlet, P.G., *Mathematical Elasticity, Volume I: Three-dimensional Elasticty.* North Holland, New York (1988).

[8] Eringen, A.C., *Mechanics of Continua.* Robert E. Krieger Publishing Company, Huntington, New York (1980).

[9] Holzapfel, G.A., Gasser, T.C. and Ogden, R.W., A New Constitutive Framework for Arterial Wall Mechanics and a Comparative Study of Material Models. *J. Elasticity* **61**, 1-48 (2000).

[10] Hughes, T.J.R., *The Finite Element Method.* Dover Publications Inc., Mineola, New York (2000).

[11] Lapidus, L. and Pinder, G.F., *Numerical solution of partial differential equations in science and engineering.* John Wiley & Sons, New York (1982).

[12] Lehoucq, R.B., Sorensen, D.C. and Yang, C., *ARPACK Users' Guide, Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* (1998).

[13] Liew, K.M. and Lim, C.W., A higher-order theory for vibration of doubly curved shallow shells. *Journal of Applied Mechanics* **63** , 587-593 (1996).

[14] Luenberger, D.G., *Linear and nonlinear programming.* Addison-Wesley Publishing Company, Reading, Massachusetts (1984).

[15] Marsden, J.E. and Hughes, T.J.R., *Mathematical foundations of elasticity.* Dover Publications Inc, New York (1993).

[16] Meirovitch, L., *Analytical Methods in Vibrations.* The MacMillan Company, Collier MacMillan Limited, London (1967).

[17] Méric, L., Poubanne, P. and Cailletaud, G., Single Crystal Modeling for Structural Calculations: Part 1 - Model Presentation. *Journal of Engineering Materials and Technology* **113** , 162-170 (1991).

[18] Méric and Cailletaud, G., Single Crystal Modeling for Structural Calculations: Part 2 - Finite Element Implementation. *Journal of Engineering Materials and Technology* **113** , 171-182 (1991).

[19] Miranda, I., Ferencz, R.M. and Hughes, T.J.R., An improved implicit-explicit time integration method for structural dynamics. *Earthquake Engineering and Structural Dynamics* **18** , 643-653 (1989).

[20] Rank, E., Ruecker, M. *Private Communication* . TU Munich (2000).

[21] Schwarz, H.R., *FORTRAN-Programme zur Methode der finiten Elemente* . Teubner (1981).

[22] Simo, J.C. and Hughes, T.J.R., *Computational Inelasticity* . Springer, New York (1997).

[23] Simo, J.C. and Taylor, R.L., Quasi-incompressible finite elasticity in principal stretches. Continuum basis and numerical algorithms. *Computer Methods in Applied Mechanics and Engineering.* **85** , 273-310 (1991).

[24] Simo, J.C., A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: Part I. Continuum formulation. *Computer Methods in Applied Mechanics and Engineering.* **66** , 199-219 (1988).

[25] Simo, J.C., A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: Part II: computational aspects. *Computer Methods in Applied Mechanics and Engineering.* **68** , 1-31 (1988).

[26] Washizu, K., Some considerations on a naturally curved and twisted slender beam. *Journal of Mathematics and Physics* **43** , 111-116.

[27] Zienkiewicz, O.C. and Taylor, R.L., *The finite element method.*McGraw-Hill Book Company (1989).