
# NOSA-ITACA 1.0 documentation[1]

V. Binante, M. Girardi, C. Padovani, G. Pasquinelli, D. Pellegrini, M. Porcelli

*Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo"*

*Consiglio Nazionale delle Ricerche, Pisa, Italy*

September 2014




[1] The NOSA-ITACA code has been developed within the framework of the project "Tools for modelling and assessing the structural behaviour of ancient constructions" (2011-2013), funded by the Region of Tuscany (PAR FAS) and conducted in collaboration with the Department of Civil and Environmental Engineering of the University of Florence.

# Contents

# Chapter 1

# The NOSA-ITACA code

## 1.1 Introduction

**NOSA-ITACA** is the result of the integration of the finite element code **NOSA** into the open-source **SALOME** platform (Figure 1.1).

The finite element code NOSA (Non-Linear Structural Analysis) [1] has been developed by the Mechanics of Materials and Structures Laboratory of the ISTI-CNR with the aim of testing new constitutive models for materials. It has moreover been applied to checking the algorithms used for integrating the equations of motion, as well as other numerical techniques for solving structural engineering problems. The development of NOSA began in 1980 and has continued over the ensuing years along the research lines of the Lab.

The first version of the code included plane, three-dimensional and axisymmetric isoparametric elements [2] and allowed for elastic and elastic-plastic analyses in the presence of infinitesimal strains with the work-hardening models described in [3]. The code has subsequently been extended to include cases of finite strains and contact problems, based on studies performed on both the constitutive equations [4 - 7] and the methods for numerical integration of the equations of motion, in the presence of follower forces [8, 9, 10].

Over recent decades, constitutive models and calculation techniques have become available that enable realistic description of the static behaviour of masonry structures. Several studies [11 - 16] have led to a better understanding of the constitutive equation of materials not withstanding tension, known in the literature as masonry-like or no-tension materials. Within this framework, masonry is modelled as a nonlinear elastic material, with zero tensile strength and infinite or bounded compressive strength.

In order to study real problems, the equilibrium problem of masonry structures can be solved via the finite element method. To this end, suitable numerical techniques have been developed [13]-[16] based on the Newton-Raphson method for solving the nonlinear system obtained by discretising the structure into finite elements. Their application requires that the derivative of the stress with respect to the strain be explicitly known, as this is needed in order to calculate the tangent stiffness matrix. The numerical method studied has therefore been implemented into the NOSA code to enable determination of the stress state and the presence of any cracking. It can moreover be applied to modelling needed restoration and reinforcement operations on constructions of particular architectural interest [17].

The code has been further enhanced to be able to perform nonlinear heat-conduction analysis on solids even in the non-stationary case, with boundary conditions concerning temperature and thermal fluxes. Today, the code provides for thermo-mechanical analysis of no-tension solids whose mechanical characteristics depend on temperature in the presence of thermal loads [18], [19], [16]. Finally, numerical solution of dynamic problems requires direct integration of the equations of motion [20]. In fact, due to the nonlin-

earity of the adopted constitutive equation, the mode-superposition method is meaningless. With an aim to solving such problems, we have instead implemented the Newmark [21] method in NOSA to perform the integration with respect to time of the system of ordinary differential equations obtained by discretising the structure into finite elements. Moreover, the Newton-Raphson scheme, needed to solve the nonlinear algebraic system obtained at each time step, has been adapted to the dynamic case. In the framework of this formulation, the uniqueness of the solution of the dynamic problem in not guaranteed, even in terms of stress, which on the contrary holds for the static case [16]. In order to overcome this drawback, a viscous stress depending linearly on the strain rate has been introduced, thanks to which the uniqueness of the displacement, strain and stress fields is ensured [22].

The code has been successfully applied to the analysis of arches and vaults [23], as well as some buildings of historical and architectural interest, amongst which the chimney of the Vecchi Macelli [24], the Medici Arsenal [25, 16] and the San Pietro in Vinculis Church [26] in Pisa, the San Nicolo' Motherhouse in Noto [27], the Goldoni Theatre in Livorno [28], the Baptistery of the Volterra Cathedral, the bell tower of Buti [29], the church of Santa Maria Maddalena in Morano Calabro [16], the church of San Ponziano in Lucca [30], the church of Santa Maria della Roccella in Roccelletta di Borgia [31] and the Rognosa tower in San Gimignano [32, 33].

Development of the code has been made possible through the funding of the C.N.R. (progetto finalizzato Informatica, progetto finalizzato Materiali Speciali per Tecnologie Avanzate, progetto finalizzato Beni Culturali, progetto COMES - Network for Computational Solid Mechanics), the Italian Ministry of Universities and Research (MIUR - Fondo Speciale per la Ricerca di Interesse Strategico "Diagnostica e salvaguardia di manufatti architettonici con particolare riferimento agli effetti derivanti da eventi sismici e altre calamita' naturali") and the region of Tuscany (structural funds, Regional High Tech Network and, last but not least, the NOSA-ITACA project).

Over the past twenty years many engineering students at the University of Pisa have collaborated on the development of the code as part of their degree or doctorate thesis preparation.

COMES-NOSA [34], a version of NOSA for static analysis of masonry constructions, can be freely downloaded at http://www.isti.cnr.it/research/unit.php?unit=MMS&section=software .

Within the framework of the NOSA-ITACA project, the NOSA code has been substantially modified and deeply improved, in the light of the FORTRAN 90 specifications, and equipped with new finite elements, thus enhancing its application capabilities. The subroutine OUTSYM devoted to the solution to linear systems via the modified LU factorization has been optimized. The performances of the code are now improved, in particular, a speed-up of about 7.3 with respect to the original version has been obtained.

An efficient implementation of numerical methods for constrained eigenvalue problems, specialized for the modal analysis of structures taking into account both the sparsity of the matrices and the features of master-slave constraints (tying or multipoint constraints) has been moreover analysed. The implementation is based on open-source packages embedded in NOSA: SPARSEKIT [35] to manage matrices in sparse format (allocation, matrix-vector products) and ARPACK[1] [36], which implements a method based on the Lanczos factorization combined with spectral techniques that improve the convergence. In particular, for using ARPACK a linear system solver is necessary, with the coefficient matrix given by the stiffness or mass matrix. The package ICFS is used, which provides an advanced implementation of the conjugate gradient method, accelerated with a preconditioner based on the incomplete Cholesky factorization [37].

The implementation of the NOSA-ITACA code for the structural analysis of historical masonry constructions has been then completed, by integrating the finite element code NOSA with the open source graphic interactive code SALOME [38].

In particular, the integration implements the NOSA code within the SALOME architecture (developed mostly in the C/C++ and Python languages) as an additional module on a par with those already existing (MESH, GEOM, POST-PRO).

[1] Copyright (c) 1996-2008 Rice University. Developed by D.C. Sorensen, R.B. Lehoucq, C. Yang, and K. Maschhoff. All rights reserved.

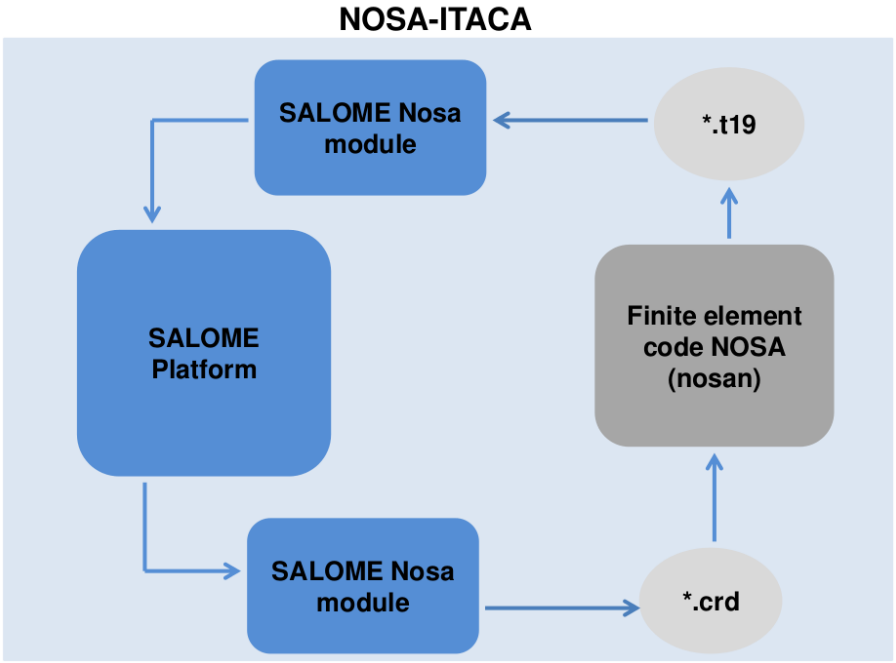



Figure 1.1: The NOSA-ITACA structure

The SALOME Nosa module allows the user to define physical quantities to be associated to the mesh (materials, elements' thickness, boundary conditions, loads, type of analysis, etc.), display the load applied to the structure, generate the input file for the running and monitoring of the finite element analysis, etc. The module includes the executable file "nosan" and several CORBA interfaces (with extension ".idl") for data exchange between the SALOME Nosa module and the MESH and/or POST-PRO modules. The executable nosan carries out the numerical analysis using as input the card ".crd" created via the SALOME Nosa module. Moreover, the SALOME Nosa module allows the user to monitor the analysis. Finally, NOSA transmits the results of the numerical study to the POST-PRO module by means of the resulting ".med" output file.

Applications of the NOSA-ITACA code are described in [39]-[42]. In particular, in [40] dynamical analyses of masonry towers conducted via the NOSA-ITACA and MADY [43] codes are compared and in [41] and [42] NOSA-ITACA is used to study the "Voltone", a large vaulted masonry structure located under Piazza della Repubblica in Livorno, Italy.

For a complete and detailed description of the theories and algorithms used in the NOSA-ITACA, the interested reader is referred to the following bibliography.

## 1.2 NOSA-ITACA Documentation

The documentation is organized as follows:

- SALOME Nosa User's Guide, which describes how to use SALOME Nosa module to define physical properties (e.g., materials, boundary conditions, loads, etc.), and assign them to mesh items.
- NOSA Theory Manual, which describes basic theory of the NOSA solver.
- NOSA Keywords Reference Guide, describing all keywords used to generate the input file crd, and processed by Nosa solver.

- User Subroutines Reference Guide, showing fortran utility routines required by Nosa solver to perform numerical analyses.
- Miscellanea, dealing with miscellaneous arguments, such as COMMON blocks used by the NOSA fem code, error codes returning by a fem analysis, output files, and so on.

## 1.3 REFERENCES

[1] DEGL'INNOCENTI S., LUCCHESI M., PADOVANI C., PAGNI A., PASQUINELLI G., ZANI N. "The finite element code NOSA Version 2.0 - User's Manual". Internal note ISTI-006/2007.

[2] HINTON E., OWEN D. R. J., Finite Element Programming, Academic Press, 1977.

[3] GUIDOTTI P, LUCCHESI M, PAGNI A., PASQUINELLI G., "Elastic-Plastic Behaviour with Work Hardening: an Appropriate Model for Structural Software". Meccanica 19, 1984.

[4] LUCCHESI M., PODIO GUIDUGLI P., "Materials with Elastic Range: a Theory with a view toward Applications. Part I". Arch. Rat. Mech. Anal., 102, pp. 23-43, 1988.

[5] LUCCHESI M., PODIO GUIDUGLI P., "Materials with Elastic Range: a Theory with a view toward Applications. Part II". Arch. Rat. Mech. Anal., 110, pp. 9-42, 1990.

[6] LUCCHESI M., OWEN D. R., PODIO GUIDUGLI P., "Materials with Elastic Range: a Theory with a view toward Applications. Part III. Approximate Constitutive Relations". Arch. Rat. Mech. Anal., 117, pp. 53-96, 1992.

[7] LUCCHESI M., PODIO GUIDUGLI P., "Materials with Elastic Range and the Possibility of Stress Oscillations in Pure Shear". Proc. Int. Conf. on Comp. Plasticity, Model, Software and Applications, Barcelona, 6-10 April 1987.

[8] GUIDOTTI P., LUCCHESI M., "A Numerical Method for Solving Boundary-Value problems in Finite Plasticity". Meccanica, 23, pp. 43-54, 1988.

[9] DEGL'INNOCENTI S., PADOVANI C., PASQUINELLI G., "An improved numerical method to integrate the equation of motion in finite elastoplasticity problems". Complas II, Second International Conference on Computational Plasticity, Barcelona, September 1989.

[10] PASQUINELLI G., "Simulation of Metal-Forming Processes by the Finite Element Method". Int. J. Plasticity, 11, pp. 623-651, 1995.

[11] DI PASQUALE S., "New trends in the analysis of masonry structures". Meccanica, 27, pp. 173-184, 1992.

[12] DEL PIERO G., "Constitutive equation and compatibility of the external loads for linearly-elastic masonry-like materials". Meccanica, 24 pp.150-162, 1989.

[13] LUCCHESI M., PADOVANI C., PAGNI A., "A numerical method for solving equilibrium problems of masonry-like solids". Meccanica, 24, pp. 175-193, 1994.

[14] LUCCHESI M., PADOVANI C. and PASQUINELLI G., "On the numerical solution of equilibrium problems of elastic solids with bounded tensile strength". Comput. Methods Appl. Mech. Engrg. 127, pp. 37-56, 1995.

[15] LUCCHESI M., PADOVANI C. and ZANI N., "Masonry-like materials with bounded compressive strength". Int. J. Solids Structures 33, pp. 1961-1994, 1996.

[16] LUCCHESI M., PADOVANI C., PASQUINELLI G., ZANI N. , "Masonry constructions: mechanical models and numerical applications", Series: Lecture Notes in Applied and Computational Mechanics, Vol. 39, Berlin Heidelberg, Springer-Verlag, 2008.

[17] PASQUINELLI G., "On the modeling of the reinforcement rings in masonry buildings: an example".

Proceedings of the Third International Conference on Contact Mechanics, Contact Mechanics III, Madrid 1997.

[18] LUCCHESI M., PADOVANI C., PASQUINELLI G., "Thermodynamics of no-tension materials". Int. J. Solids and Structures 37, pp. 6581-6604, 2000.

[19] PADOVANI C., PASQUINELLI G., ZANI N., "A numerical method for solving equilibrium problems of no-tension solids subjected to thermal loads". Comput. Methods Appl. Mech. Engrg., 190, pp. 55-73, 2000.

[20] DEGL'INNOCENTI S., PADOVANI C. & PASQUINELLI G., "Numerical methods for the dynamic analysis of masonry structures". Structural Engineering and Mechanics, 22, pp.107-130, 2006.

[21] BATHE, JK. J., WILSON, WE. L., "Numerical methods in finite element analysis", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[22] PADOVANI C., PASQUINELLI G., ŠILHAVÝ M., "Processes in masonry bodies and the dynamical significance of collapse". Mathematics and Mechanics of Solids, 13, pp.573-610, 2008.

[23] LUCCHESI M., PADOVANI C., PASQUINELLI G., ZANI N., "Static analysis of masonry vaults, constitutive model and numerical analysis". Journal of Mechanics of Materials and Structures, 2, pp. 221-244, 2007.

[24] LUCCHESI M., ZANI N., "Analisi sismica di tipo statico della ciminiera dei Vecchi Macelli, Via Nicola Pisano (Pisa)". Relazione Tecnica Consorzio Pisa Ricerche, 29/04/1997.

[25] LUCCHESI M., ZANI N., "Analisi sismica di tipo statico dell'arsenale mediceo situato in Pisa, lungarno Simonelli". Relazione Tecnica Consorzio Pisa Ricerche, 12/11/1996.

[26] BERNARDESCHI K., PADOVANI C., PASQUINELLI G., "Studio del comportamento statico del muro della navata destra della chiesa di San Pietro in Vinculis a Pisa". Bollettino Ingegneri, 8/9, pp. 9-17, 2003.

[27] LUCCHESI M., PADOVANI C., ZANI N., "Studio del comportamento statico di alcuni elementi strutturali della Chiesa Madre S. Nicolo' di Noto". Relazione Tecnica Consorzio Pisa Ricerche, 09/03/1998.

[28] LUCCHESI M., DE FALCO A., ZANI N., "Studio del comportamento statico dell'arco scenico del Teatro Goldoni di Livorno". Relazione Tecnica Consorzio Pisa Ricerche, 20/05/1998.

[29] BERNARDESCHI K., PADOVANI C., PASQUINELLI G., "Numerical modelling of the structural behaviour of Buti's bell tower". International Journal of Cultural Heritage, 5(4), pp. 371-378, 2004.

[30] GIRARDI M., PADOVANI C., PAGNI A., PASQUINELLI G., "Static analysis of masonry vaults and domes". International workshop IN SITU MONITORING OF MONUMENTAL SURFACES (Firenze, 27-29 October 2008). Proceedings, pp. 335 - 340. P. Tiano and C. Pardini (eds.). Edifir - Edizioni Firenze, 2008.

[31] GIRARDI M., LUCCHESI M, PASQUINELLI G., ZANI N., "Analisi strutturale della chiesa di Santa Maria di Roccella", Technical Report n° cnr.isti/2009-TR-051, 2009.

[32] GIRARDI M., PADOVANI C., PAGNI A., PASQUINELLI G., "Numerical modeling of masonry towers: the case study of the Rognosa Tower in San Gimignano", Advances and Trends in Structural Engineering, Mechanics and Computation, A. Zingoni (Editor), CRC Press/Balkema, AK Leiden, The Netherlands, 2010.

[33] CALLIERI M., CORSINI M., GIRARDI M., PADOVANI C., PAGNI A., PASQUINELLI G., Scopigno R., "The Rognosa Tower in San Gimignano: digital acquisition and structural analysis", CST 2010 – Proceedings of The Tenth International Conference on Computational Structures Technology, art. n 138. Topping B.H.V. , Adam J. M., Pallarés F.J., Bru R., Romero M.L. (Editors), Civil Comp Press, 2010.

[34] DEGL'INNOCENTI S., LUCCHESI M., PADOVANI C., PAGNI A., PASQUINELLI G., ZANI N. "The finite element code COMES-NOSA for the structural analysis of ancient masonry constructions -

Version 2.0 - User's Manual". Technical Report ISTI-02/2008.

[35] SAAD Y., "SPARSKIT: A basic tool kit for sparse matrix computations". Technical Report, Computer Science Department, University of Minnesota, June 1994.

[36] LEHOUCQ R. B., SORENSEN D. C., YANG C., "ARPACK Users Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods". SIAM, 1998.

[37] LIN C.-J., MORE' J. J., "Incomplete Cholesky Factorizations with Limited Memory". SIAM J. Sci. Comput., 21(1), 24-45, 1999.

[38] http://www.salome-platform.org/

[39] BINANTE V., GIRARDI M., PADOVANI C., PAGNI A., PASQUINELLI G., "The NOSA-ITACA code for the modelling of the structural behaviour of historic masonry constructions", 5th International Congress on Science and Technology for the Safeguard of Cultural Heritage in the Mediterranean Basin (Istanbul, Turkey, 22-25 November 2011). Proceedings, vol. II (1st part) pp. 40 - 48. Valmar, 2012.

[40] GIRARDI M., LUCCHESI M., PADOVANI C., PINTUCCHI B., PASQUINELLI G., ZANI N., "Numerical methods for slender masonry structures: a comparative study", CST 2012 - The Eleventh International Conference on Computational Structures Technology (Dubrovnik, Croatia, 4-7 September 2012). Proceedings, article n. 118. B.H.V. Topping (ed.). Civil-Comp Press, 2012.

[41] BINANTE V., BRICCOLI BATI S., GIRARDI M., LUCCHESI M., PADOVANI C., PELLEGRINI D., "A Case Study for the NOSA-ITACA Project: the "Voltone" in Livorno", CC2013 - 14th International Conference on Civil, Structural and Environmental Engineering Computing (Cagliari, Italy, 3-6 September 2013).

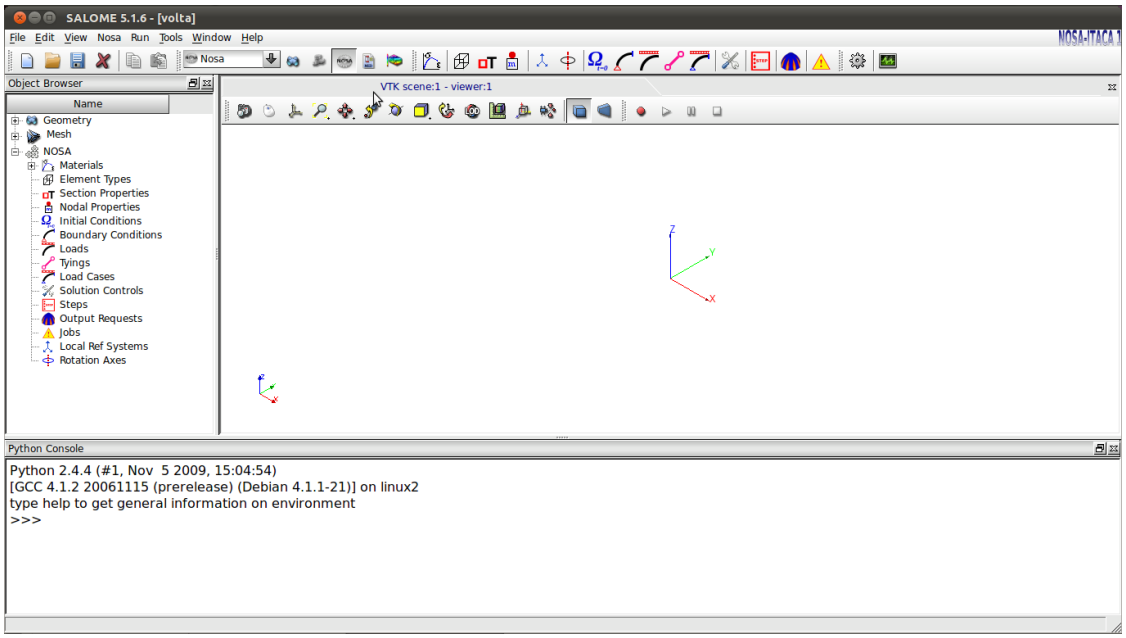
[42] PORCELLI M., BINANTE V., GIRARDI M., PADOVANI C., PASQUINELLI G., "A solution procedure for constrained eigenvalue problems and its application within the structural finite-element code NOSA-ITACA ", Calcolo, Springer, [Online First 29 March 2014].

[43] LUCCHESI M., PINTUCCHI B., "A numerical model for non-linear dynamic analysis of slender masonry structures", European Journal of Mechanics A/Solids, 26, 88-105, 2007.

# Chapter 2

# SALOME Nosa User's Guide

This section describes how to use the **Nosa** module to define physical properties and assign them to the mesh items previously created through the **Mesh** module.




In particular, the guide allows us to:

- Create Materials
- Specify the Type of Element
- Define Element Section Thickness
- Define Concentrated Masses to Nodes
- Define local reference systems
- Define Rotation Axes
- Set Initial Conditions
- Set Boundary Conditions
- Define Loads
- Define Multi-Point Constraints
- Set Solution Controls
- Define Load-Steps
- Specify Output Requests
- Define Jobs
- Submit a Job
- Monitor a Job

All Nosa module functionalities are accessible via Nosa module Python interface.

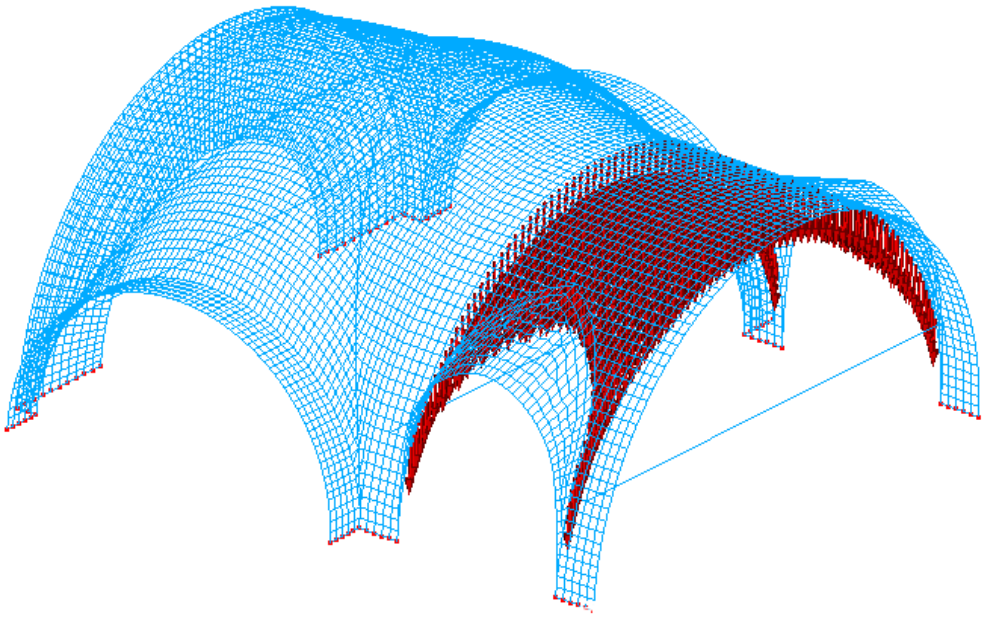


Figure 2.1: Example of a masonry construction constrained at the base and partially subjected to its own weight.

## 2.1 Create Materials

### 2.1.1 About Materials

The user can describe the behaviour of four kinds of material; they are:

- **linear-elastic material**;
- **elastic-plastic material**, with different hardening rules and isotropic and/or anisotropic plasticity;
- **masonry-like material**, with different options about compressive and tensile strength;
- **heat-transfer analysis** .

With reference to elastic-plastic materials, NOSA-ITACA software supplies a wide range of material behaviours; thus, ideal-plasticity, isotropic-hardening, kinematic-hardening as well as mixed-hardening can be described. Both infinitesimal and finite strains can be modelled. Anisotropic behaviour is also supported according to the Hill's theory.

For masonry-like material, the user can choose a finite or unbounded compressive strength as well as null and/or bounded tensile strength; constituive modelling of a masonry material is made under the assumption of infinitesimal deformations.

For each material, a non-linear behaviour of mechanical properties can be described; in particular, the user can model the dependence of a material property on the temperature field (temperature dependent property).

**Elastic-plastic and heat transfer analyses are not available in the present version of NOSA-ITACA.**

Further details can be found at NOSA Theory Manual

### 2.1.2 Defining a material behaviour

Definition of a material behaviour consists of:

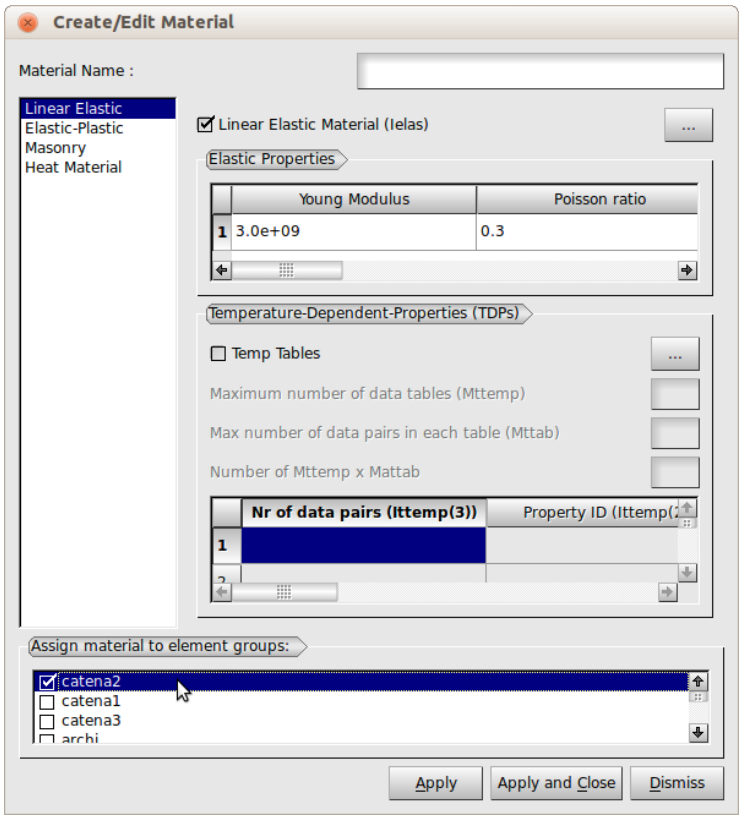
- selecting a material type among those above mentioned;
- defining the mechanical properties (e.g. Young modulus, Poisson ratio, density, etc...);
- defining temperature dependence of material parameters, if this is the case;
- assigning the material behaviour to groups of mesh elements.

*To define a material behaviour:*

1. in the **Nosa** menu select **Create Material** or click *"Create Material"* button in the toolbar.

Figure 2.2: Create Material button

the following dialog box will appear:



2. type the name of the material to be defined into the *"Material Name"* box; if no name is specified, a default name *"Mat-1"* is assigned. Then select a material type, from the list view on the left; the corresponding page on the right is update, where mechanical properties of the selected material must be defined.

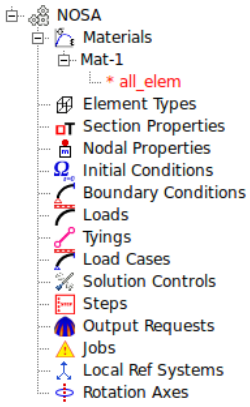
3. enable the push button to define the material parameters (e.g. in the above figure enable the *"Linear Elastic Material (Ielas)"* push button); in the single row table insert values of the mechanical properties. If a cell (i.e. material property) is left blank a null value is associated to this property.

4. to define temperature dependence of a material parameter, enable the *"Temp Tables"* push button; then specify the number of tables and for each table the maximum number of rows; for a material you can specify one or more temperature tables, depending whether one or more material properties depend on temperature field. In the below table, with (number of tables x maximum number of rows) rows and 4 columns, specify:

   - in the first column, the exact number, N, of rows used to define the temperature dependence of a material property; this number must be repeated for the following N rows;
   - in the second column, the identification number of the material property, *propID*, whose values are shown in Table 2.1; *propID* must be repeated for the following N rows;
   - in the third column, specify the J-th value of the temperature ( with J = 1,2,...,N)
   - in the fourth column, specify the value of the material property specified by *propID* and associated with the J-th temperature value.

5. from the list box select one or more element groups which the material so defined must be applied to.

6. click on the *"Apply"* button to create the material and define a new one, or *"Apply and Close"* to create the material and quit from the material dialog box. The material object so created will be stored into the **Object browser**;




from here, it is possible to edit the material object, by right clicking on it and selecting *"Edit Material"*

**Remarks:**

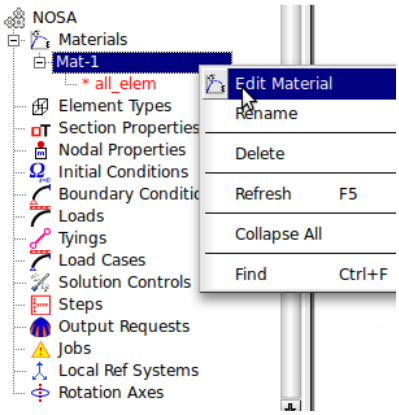
- For groups of shell elements with **COMPOSITE** option, **material cannot be assigned** to **every layer** of **shell**; that is, all shell layers will be made of the same material, as applied through the create/edit material dialog box. Shell element with multiple material behaviours can be defined only by means of the input file usage (see also COMPOSITE and Nosa Keywords Reference Guide: COMPOSITION).

| *Analysis type* | *Value of propID* |
|---|---|
| Stress analysis | propID = 1 : Young's modulus, $E$<br>propID = 2 : Poisson's ratio, $\nu$<br>propID = 3 : coefficient of the linear thermal expansion, $\alpha^{th}$ |
| heat transfer analysis | propID = 1 : thermal conductivity, $\kappa$<br>propID = 2 : specific heat per unit mass, $c$<br>propID = 3 : emissivity, $\epsilon$ |

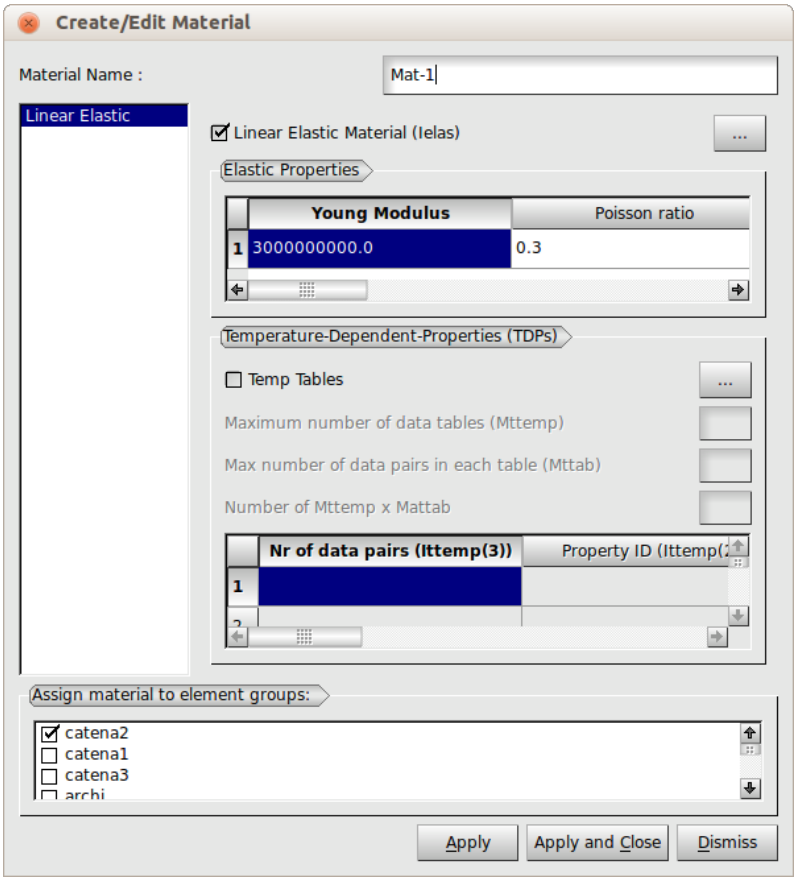
Table 2.1: Id number of material property depending on the temperature field

### 2.1.3 Editing materials

To edit a material, right click on the material object from the Object browser, and select *"Edit Material"*;




the create/edit material dialog box will appear:




from here, you can edit the material name, material properties, disable/enable temperature dependence and changing the element sets to which material applies. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

## 2.2 Specify the Type of Element

### 2.2.1 About NOSA Element Library

Mesh elements can be classified by means of their space dimension (e.g., 1-D, 2-D, 3-D elements), shape (e.g., 2-D elements can be grouped into triangle, quadrangle subsets) and/or connectivity (e.g., quadrangle elements can be grouped into quad4 or quad8 classes, depending whether the number of cell nodes is 4 or 8). In addition to this topological classification (used by the Mesh module of NOSA-ITACA software) there is another one, based on mathematical and physical aspects, which are on the basis of the finite element programming (e.g., degrees of freedom of an element, output variables available for a particular element, isoparametric space,...etc). **NOSA Element Library** belongs to the last kind of element classification, and the user must bear clearly in mind its difference from the Mesh Element Library. According to NOSA Element Library, elements can be classified by means of:

- analysis type (i.e. static, dynamic, modal and heat transfer analysis);
- geometric order (i.e, linear or quadratic isoparametric elements);
- problem type (e.g., axisymmetric, plane strain, plane stress, etc ...);

Currently, only seventeen element types are implemented into NOSA-ITACA software; the next release will be update to thirty-six element types.

Further details can be found at NOSA Theory Manual: Element Library

### 2.2.2 Specifying an element type

The choice of an element type consists of:

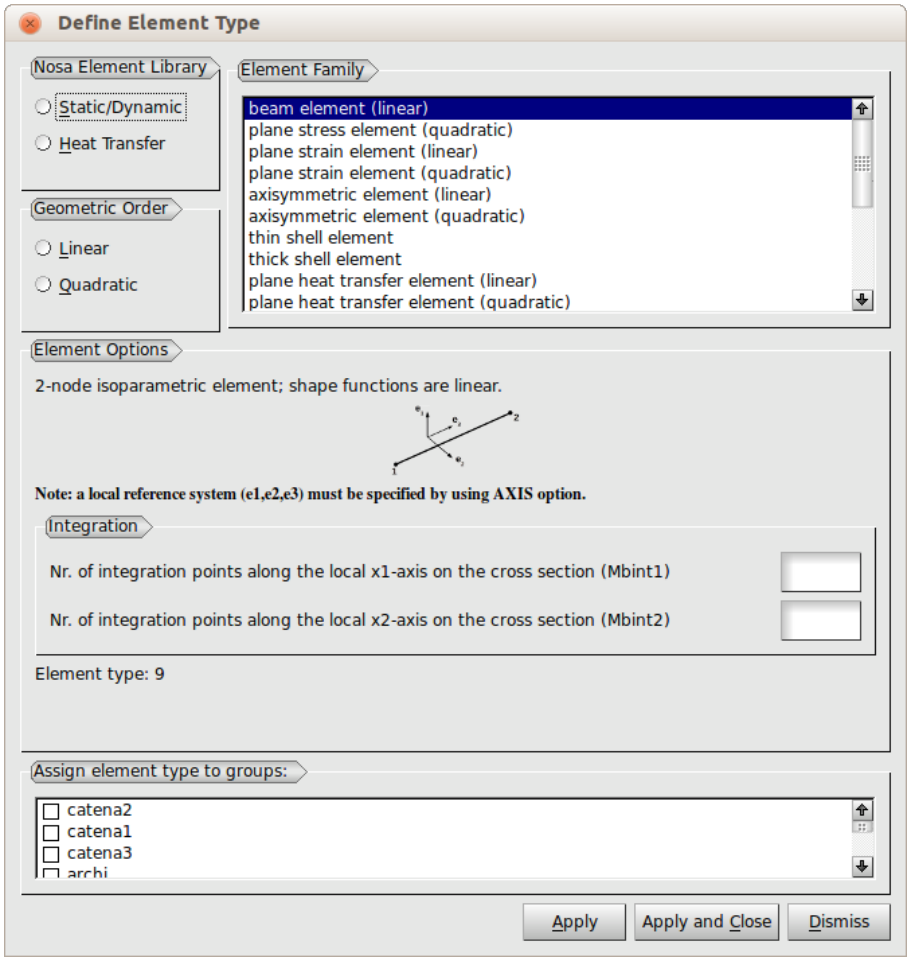
- selecting an element type among those available into NOSA Element Library;
- defining some properties, when it is required;
- assigning the element type to groups of mesh elements.

*To choose an element type of the NOSA Element Library:*

1. in the **Nosa** menu select **Nosa Element Library** or click on *"Nosa Element Library"* button in the toolbar.

Figure 2.3: Nosa Element Library button

the following dialog box will appear:




2. from the *"Nosa Element Library"* group box select the analysis type to be performed (by clicking on one between the radio buttons available).

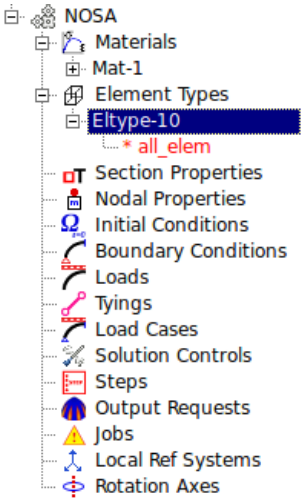
3. from the *"Geometric Order"* group box specify the type of isoparametric element (by clicking on one between the radio buttons available).

4. now, the list view of the *"Element Family"* group box will contain only those elements with the above specified attributes; from this list, select the element type required.

5. the *"Element Options"* group box will show a picture of the selected element type together with a brief description of this element, some notes and element attributes to be defined (not all element types require further specifications).

6. from the list box select one or more element groups which the element type so defined must be applied to.

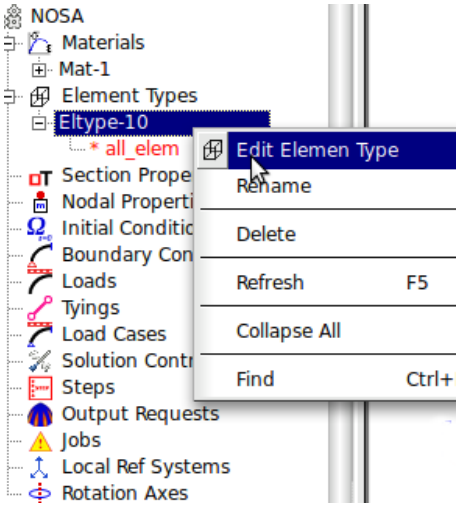
7. click on the *"Apply"* button to specify the element type and define a new one, or *"Apply and Close"* to specify the element type and quit from the dialog box. The element type object so created will be stored into the **Object browser**;



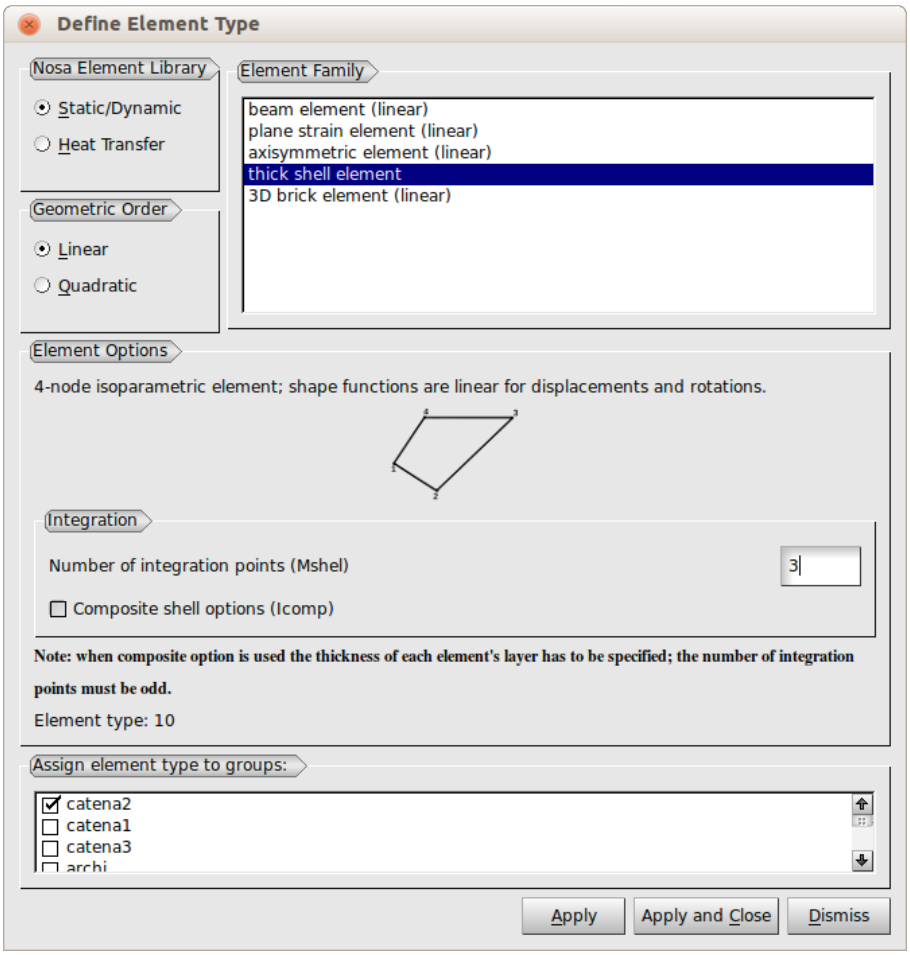

from here, it is possible to edit the object, by right clicking on it and selecting *"Edit Element Type"*

### 2.2.3 Editing element types

To edit an element type, right click on the element type object from the Object browser, and select *"Edit Element Type"*;




the define element type dialog box will appear:




from here, you can edit the element type, element attributes, and changing the element set to which element type applies. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

## 2.3 Define Element Section Thickness

### 2.3.1 About Element Cross-Section Properties

Once a master element of NOSA Element Library has been defined and associated with some element groups, the next step could require further element attributes to be defined; that is, the cross-section of the selected master element has to be specified. This is done by assigning the thickness(es) of the cross-section for:

- homogeneous plane strain/stress elements;
- homogeneous shell elements;
- composite shell elements;
- homogeneous beam elements;
- composite beam elements.

For other element types no further attribute must be specified, and thus the user can ignore this step.

### 2.3.2 Defining element cross-section properties

The definition of the cross-section of a master element consists of:

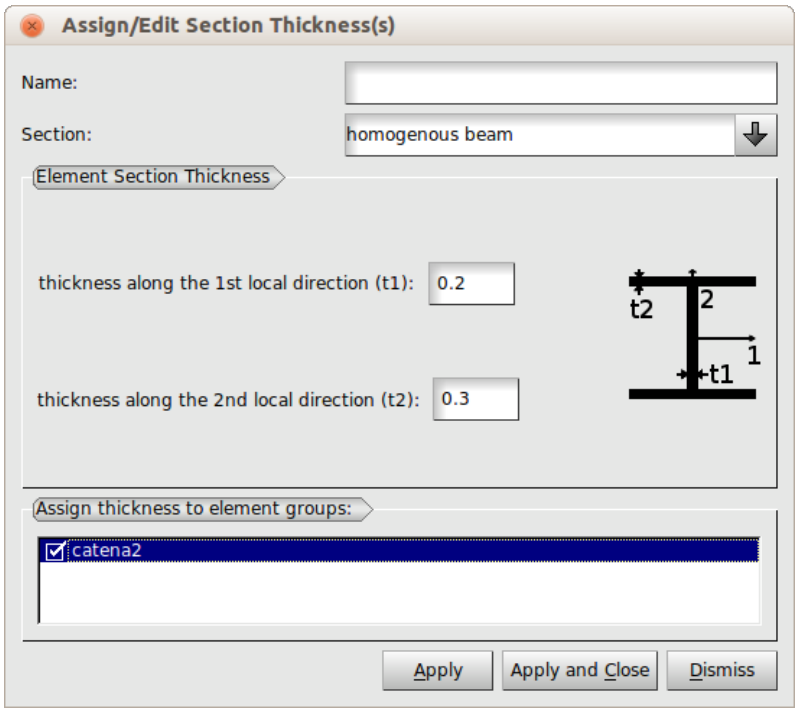
- selecting a cross-section type among those above mentioned;
- specifying the thickness(es);
- assigning these attributes to groups of mesh elements.

*To define the cross-section of a master element:*

1. in the **Nosa** menu select **Element Section Thickness** or click on *"Element Section Thickness"* button in the toolbar.

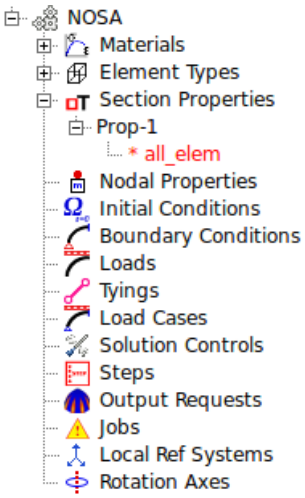
Figure 2.4: Element Section Thickness button

the following dialog box will appear:




2. into the *"Name"* box type the name of the section to be defined; if no name is specified, a default name *"Prop-1"* is assigned.
3. from the *"Section"* drop-down list box select the section type to be defined.
4. the *"Element Section Thickness"* group box will show a picture of the selected cross-section type together with the thickness(es) to be specified.
5. from the list box select one or more element groups which these attributes applied to.

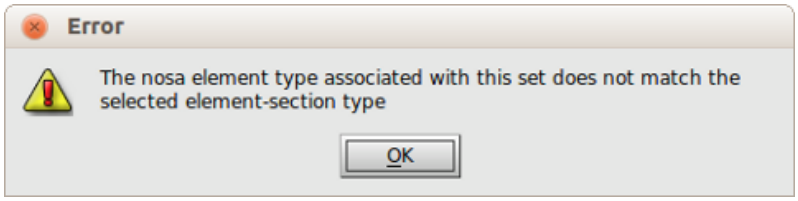
6. click on the *"Apply"* button to create the element cross-section and define a new one, or *"Apply and Close"* to create the element cross-section and quit from the dialog box. The element cross-section object so created will be stored into the **Object browser**;




from here, it is possible to edit the object, by right clicking on it and selecting *"Edit Element Section"*

**Remarks:**

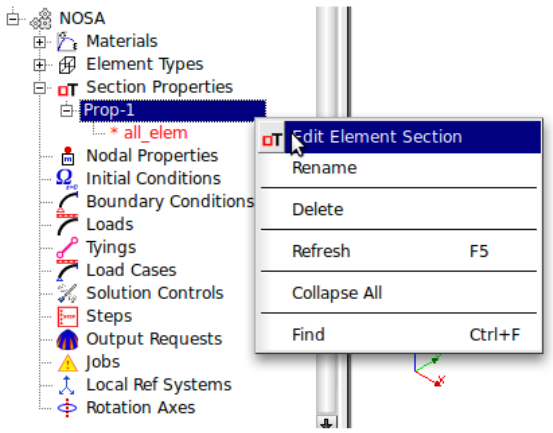
- When a wrong cross-section is defined for a given element set, an error dialog box will appear, and the creation of the cross-section will be aborted:



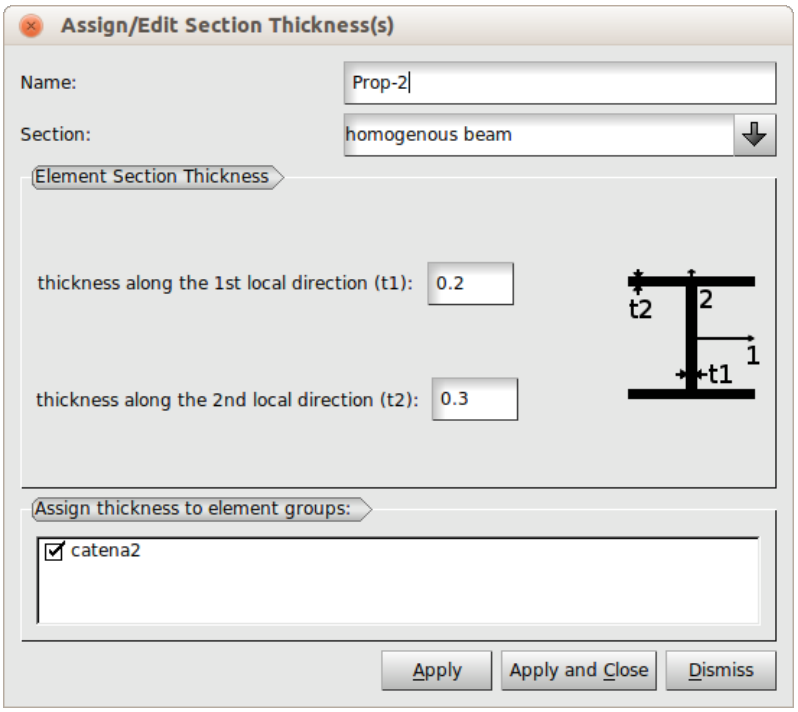

for example, this occurs when homogeneous plane stress/strain cross-section is defined for the element set *"catena2"*, which is made up of beam elements.

### 2.3.3 Editing element cross-section

To edit an element cross-section, right click on the element section object from the Object browser, and select *"Edit Element Section"*;




the following dialog box will appear:




from here, you can edit the element cross-section type, thickness(es), and change the element set to which element cross-section applies. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

## 2.4 Define Concentrated Masses to Nodes

### 2.4.1 About Concentrated Masses

When it is required for a particular structural analysis, generalized concentrated mass(es) can be applied to one or more mesh nodes; it is also possible to associate masses to node sets.

### 2.4.2 Defining concentrated masses to nodes

The definition of a concentrated mass consists of:

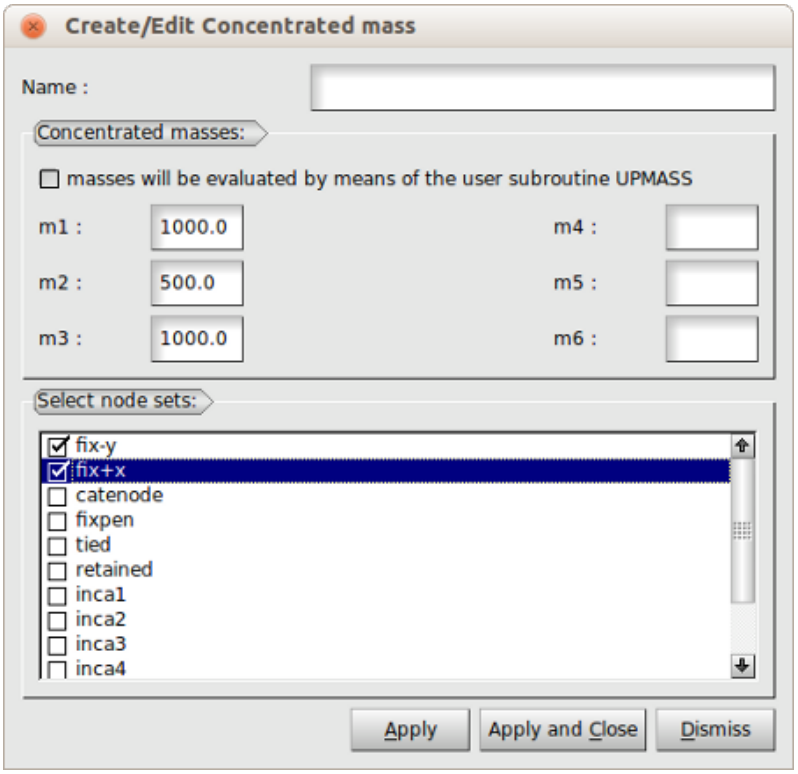
- specifying the value of the concentrated mass, directly or by means of the user subroutine **UPMASS** (see User Subroutines Reference Guide: UPMASS);
- applying the concentrated mass(es) to groups of mesh nodes.

*To apply a concentrated mass to node sets:*

1. in the **Nosa** menu select **Nodal Property** or click on *"Nodal Property"* button in the toolbar.

Figure 2.5: Nodal Property button

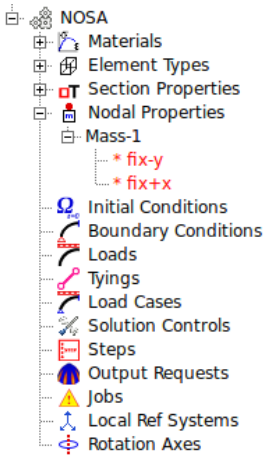
the following dialog box will appear:




2. in the *"Name"* box type the name of the nodal property to be defined; if no name is specified, a default name *"Mass-1"* is assigned.
3. in the *"Concentrated masses"* group box specify the value(s) of the mass(es); enable the push button for specifying that the values will be defined through the user routine UPMASS (in such a case every value typed into the mass boxes will be overwritten by the user routine, thus you can leave them blank), or define mass values directly by typing them into the mass boxes.

4. in the list view of the *"Select node sets"* group box select one or more nodal groups to which concentrated masses apply.

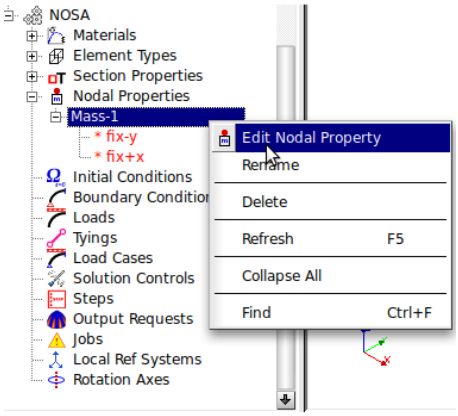
5. click on the *"Apply"* button to assign concentrated masses and define a new one, or *"Apply and Close"* to assign concentrated masses and quit from the dialog box. The concentrated masses object so created will be stored into the **Object browser**;



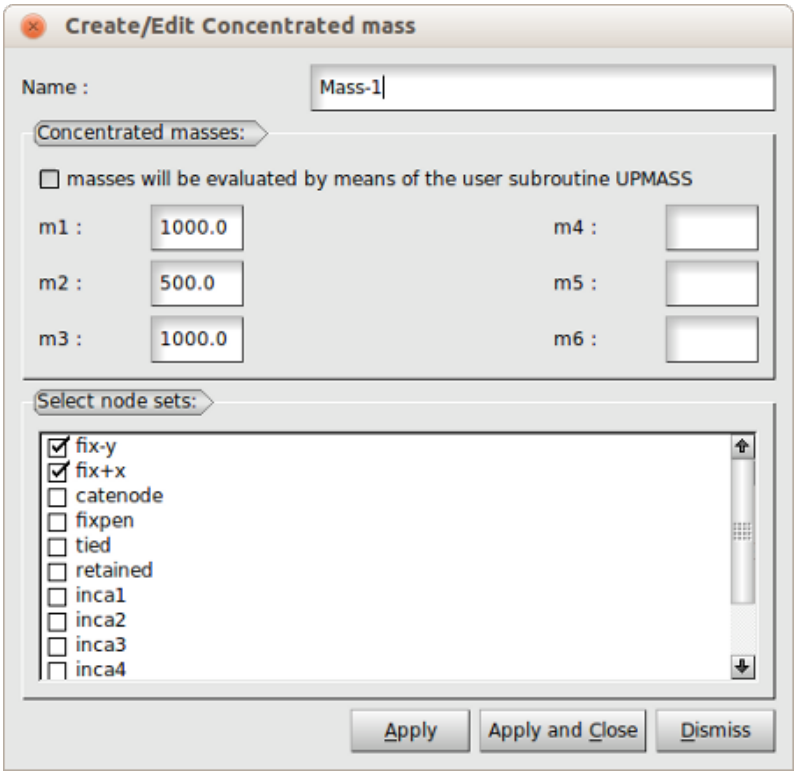

from here, it is possible to edit the object, by right clicking on it and selecting *"Edit Nodal Property"*

### 2.4.3 Editing nodal properties

To edit a nodal property, right click on the mass object from the Object browser, and select *"Edit Nodal Property"*;




the mass dialog box will appear:




from here, you can edit the mass values and changing the node sets to which masses apply. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

## 2.5 Define local reference systems

### 2.5.1 About Local Reference Systems

Local reference systems must be defined only when beam elements are involved in the mesh; for other element types this is not required. The user must bear in mind that the definition of a local reference system used to write output results along user-defined local directions can be done by means of the user routine **PLOTV** (see User Subroutines Reference Guide: PLOTV) and not with this procedure.

### 2.5.2 Defining local reference systems

The definition of a local reference system for beam elements consist of:

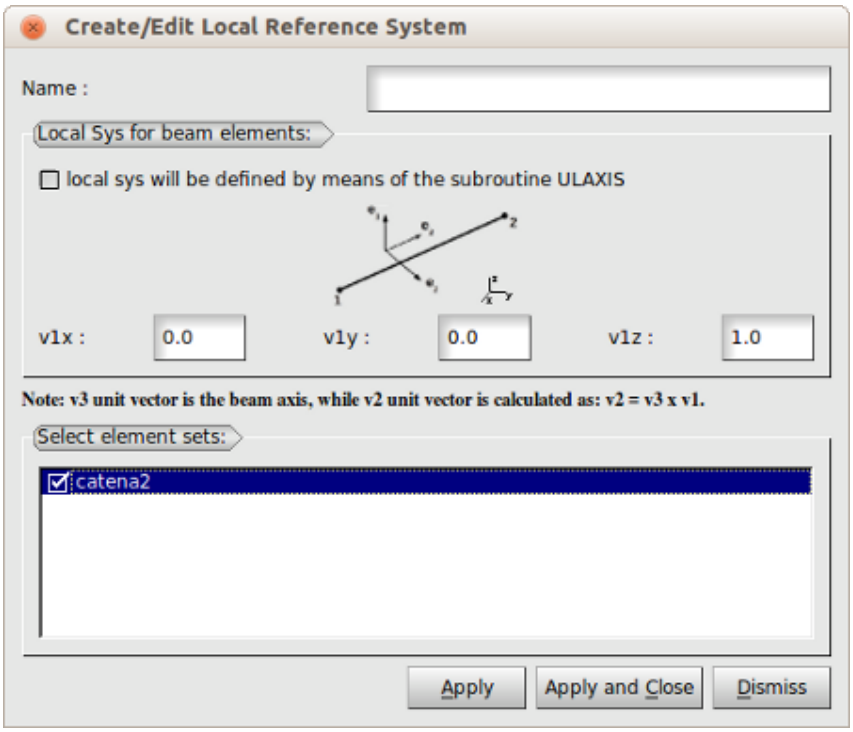
- specifying the values of the components of the unit vector **v1**, defining the first local direction perpendicular to beam axis; this can be done directly or by means of the user subroutine **ULAXIS** (see User Subroutines Reference Guide: ULAXIS);
- specifying groups of mesh elements with this reference system.

*To define a local reference system:*

1. in the **Nosa** menu select **Local Reference System** or click on *"Local Reference System"* button in the toolbar.

Figure 2.6: Local Reference System button

the following dialog box will appear:

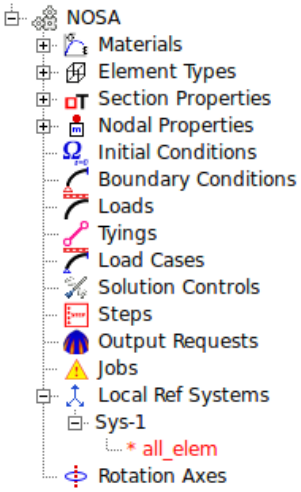



2. in the *"Name"* box type the name of the local system to be defined; if no name is specified, a default name *"Sys-1"* is assigned.

3. in the *"Local sys for beam elements"* group box specify the values of the components of the unit vector **v1**; enable the push button for specifying that the components will be defined through the user routine ULAXIS (in such a case the values typed into the component boxes will be overwritten by the user routine, thus you can leave them blank), or define the components directly by typing them into the component boxes.

4. in the list view of the *"Select element sets"* group box select one or more element groups for which local system has been defined.

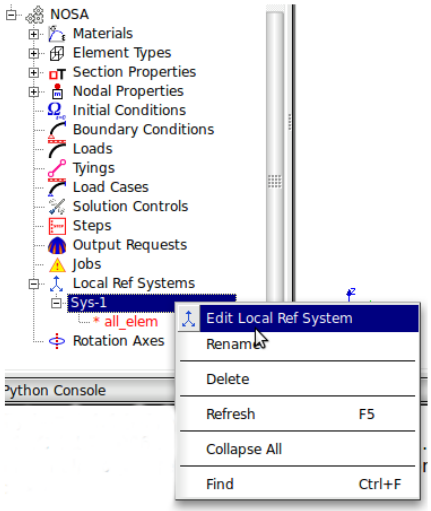
5. click on the *"Apply"* button to create the local system and define a new one, or *"Apply and Close"* to create the local system and quit from the dialog box. The local system object so created will be stored into the **Object browser**;



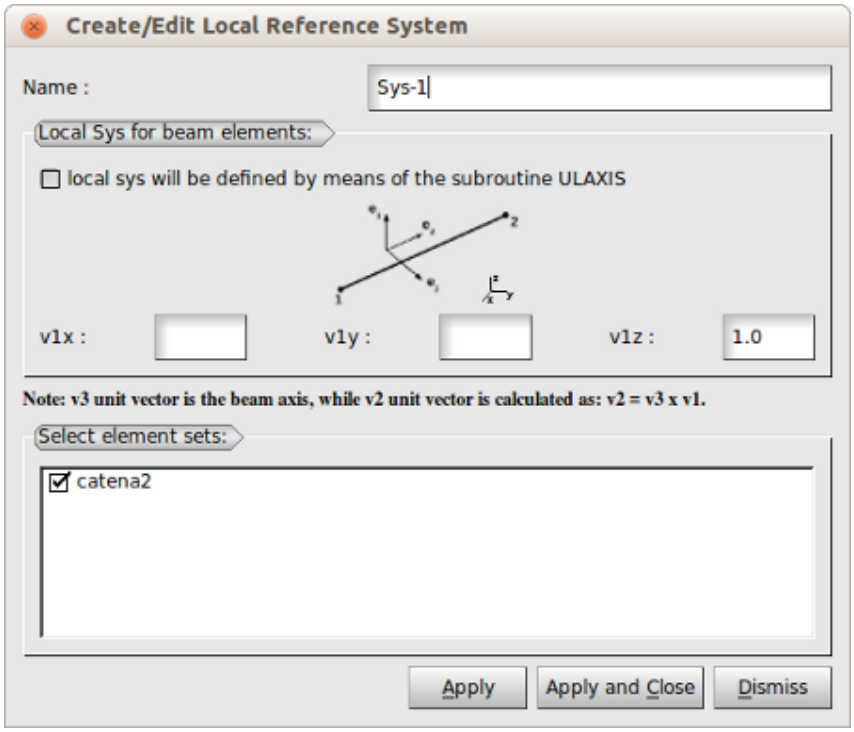

from here, it is possible to edit the object, by right clicking on it and selecting *"Edit Local Ref System"*

### 2.5.3 Editing local reference systems

To edit a local reference system, right click on the system object from the Object browser, and select *"Edit Local Ref System"*;




the system dialog box will appear:




from here, you can edit the local direction and changing the element sets to which local system applies. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

## 2.6 Define Rotation Axes

### 2.6.1 About Rotation Axis

The definition of rotation axes is required when centrifugal forces must be applied on a structure; in problems where such forces are not involved the user can ignore this procedure.

### 2.6.2 Defining a rotation axis

The definition of a rotation axis consist of:

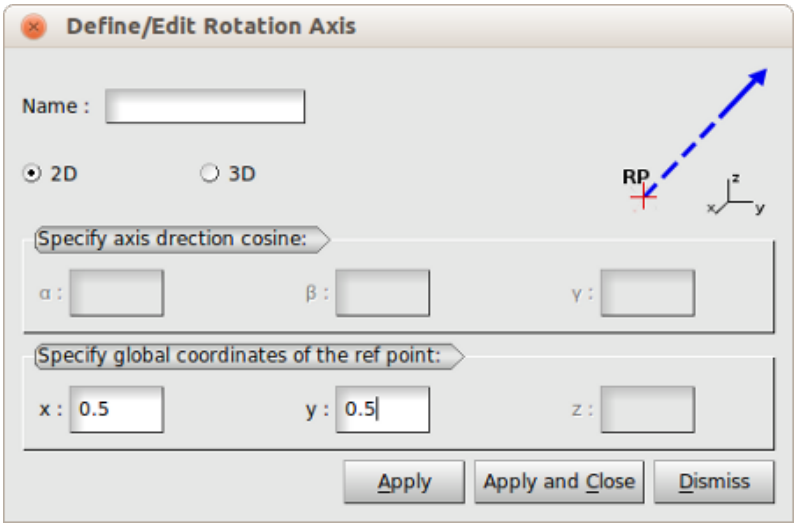
- specifying the space dimension of the problem (i.e., mesh build up in 2D or 3D space);
- assigning the coordinates of a reference point which the rotation axis passes through;
- specifying the components of the unit vector defining the rotation axis (only for 3D problems).

*To define a rotation axis:*

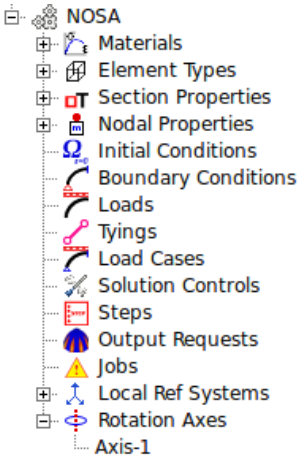
1. in the **Nosa** menu select **Rotation Axis** or click on *"Rotation Axis"* button in the toolbar.

Figure 2.7: Rotation Axis button

the following dialog box will appear:




2. in the *"Name"* box type the name of the axis to be defined; if no name is specified, a default name *"Axis-1"* is assigned.
3. enable the *"2D"* button to specify a two-dimensional problem or the *"3D"* button else.
4. If it is enabled, in the *"Specify axis direction cosine"* group box define the components of the normalized rotation axis.
5. specify the coordinates of a reference point which the rotation axis passes through.
6. click on the *"Apply"* button to create the rotation axis and define a new one, or *"Apply and Close"* to create the rotation axis and quit from the dialog box. The axis object so created will be stored into the **Object browser**;



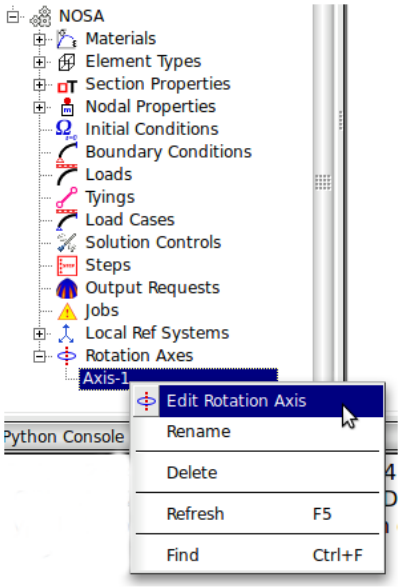

from here, it is possible to edit the object, by right clicking on it and selecting *"Edit Rotation Axis"*.

**Remarks:**

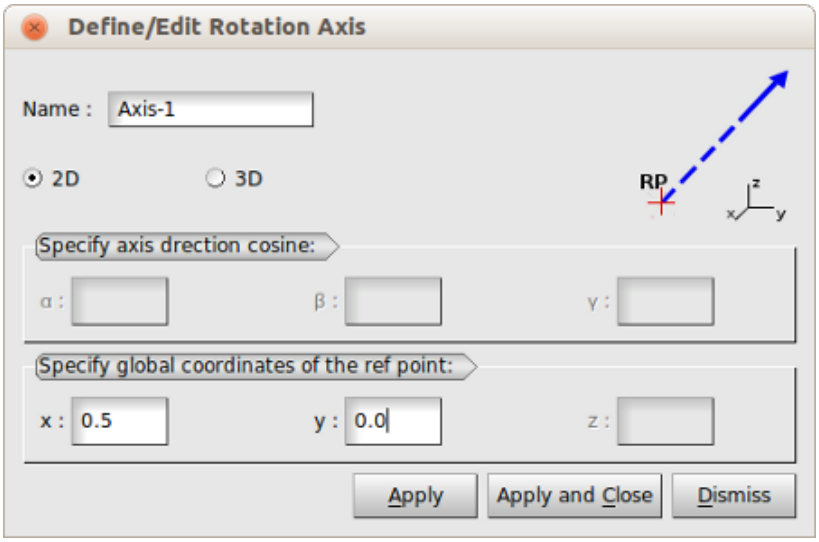
For 2D problems the rotation axis is perpendicular to the plane of the structure, and, thus, only the coordinates x and y of the reference point are required. For axisymmetric problems, the rotation axis coincides with the axis of simmetry; in such a case the user must disregard this procedure.

### 2.6.3 Editing rotation axes

To edit a rotation axis, right click on the axis object from the Object browser, and select *"Edit Rotation Axis"*;




the following dialog box will appear:




from here, you can edit the components of the rotation axis and the coordinates of the reference point. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

## 2.7 Set Initial Conditions

### 2.7.1 About Initial Conditions

Usually, the reference configuration of a body - i.e. a configuration which a structure lies in before applying a deformation process - is undeformed and characterized by a null state of stresses and strains; when the reference configuration is different from the undeformed one, the user must define the conditions describing the initial configuration of the structure. Initial conditions regard the following fields:

- displacement;
- velocity;
- temperature;
- stress.

Each of these conditions applies to a particular analyis to be solved and one or more sets of mesh nodes.

### 2.7.2 Specifying an initial condition

The definition of an initial condition consists of:

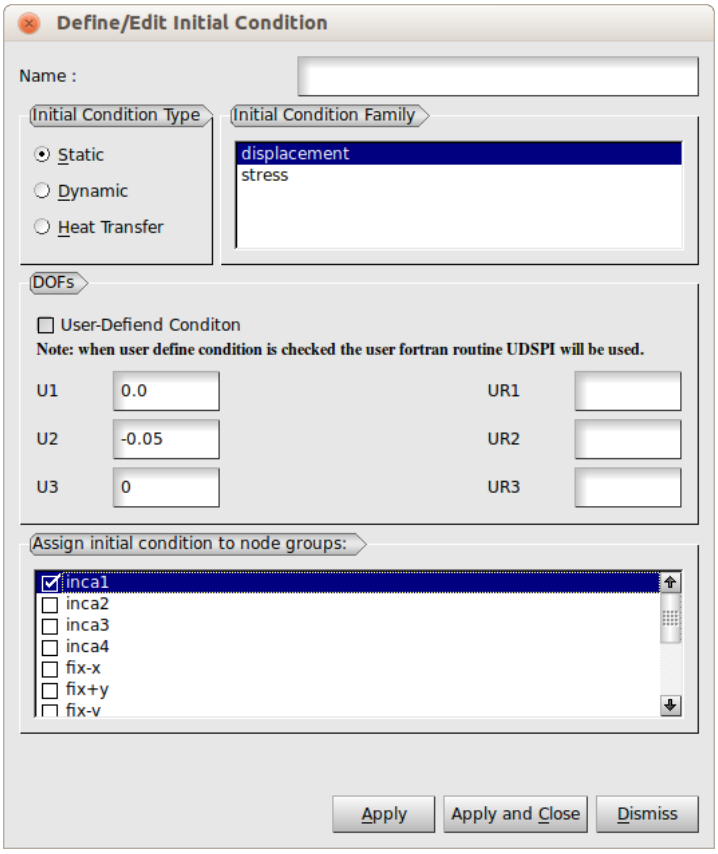
- selecting the initial condition type according to the analysis to be performed;
- specifying the values of the degrees of freedom to be constrained, directly or by means of user routines;
- applying these conditions to groups of mesh nodes.

*To define an initial condition:*

1. in the **Nosa** menu select **Initial Conditions** or click on *"Initial Conditions"* button in the toolbar.

Figure 2.8: Initial Conditions button

the following dialog box will appear:




2. into the *"Name"* box type the name of the initial condition to be defined; if no name is specified, a default name *"IC-1"* is assigned.

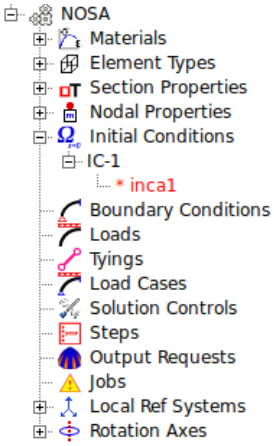
3. in the *"Initial Condition Type"* group box select the analysis type to be performed (by clicking on one between the radio buttons available).

4. the list view of the *"Initial Condition Family"* group box will contain only the initial conditions which can be imposed with the specified analysis type; from this list, select the initial condition to be defined.

5. in the *"DOFs"* group box you can specify the values of degrees of freedom to be constrained, or enable the push button to specify that initial condition will be defined by means of the user routines.

6. from the list view select one or more node groups which the initial condition refers to.

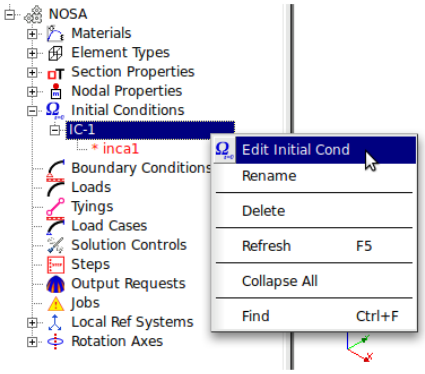
7. click on the *"Apply"* button to create the initial condition and define a new one, or *"Apply and Close"* to create the initial condition and quit from the dialog box. The initial condition object so created will be stored into the **Object browser**;



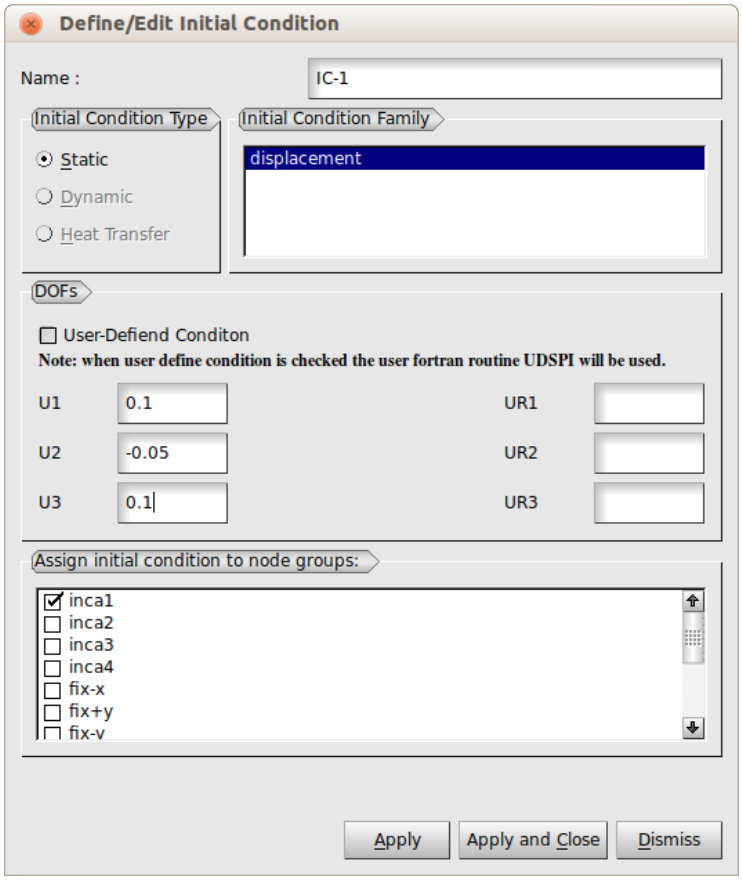

from here, it is possible to edit the object, by right clicking on it and selecting *"Edit Initial Cond"*

### 2.7.3 Editing initial condition

To edit an initial condition, right click on the object from the Object browser, and select *"Edit Initial Cond"*;




the initial condition dialog box will appear:




from here, you can edit the values of the degrees of freedom, and changing the node set to which initial condition applies. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

**See also:**

- User Subroutines Reference Guide: UDSPI
- User Subroutines Reference Guide: UVELI
- User Subroutines Reference Guide: UTEMPI
- User Subroutines Reference Guide: INSTRE

## 2.8 Set Boundary Conditions

### 2.8.1 About Boundary Conditions

In NOSA-ITACA software it is possible to define boundary conditions in terms of:

- displacement;
- acceleration;
- temperature;
- film coefficient;

- damping.

Each of these conditions applies to a particular analyis to be solved and one or more sets of mesh items. As an example, a displacement boundary condition is used in static analyses to constrain one or more degrees of freedom of mesh nodes belonging to a mesh group. In such a case, the user can impose a null or finite value for one or more components of the displacement field; the imposed values will be retained during the whole analysis. If a load increment requires that a degree of freedom can change with evolving the analysis, the user must redefine (i.e. create a new one) the boundary condition for this load increment.

### 2.8.2 Specifying a boundary condition

The definition of a boundary condition consists of:

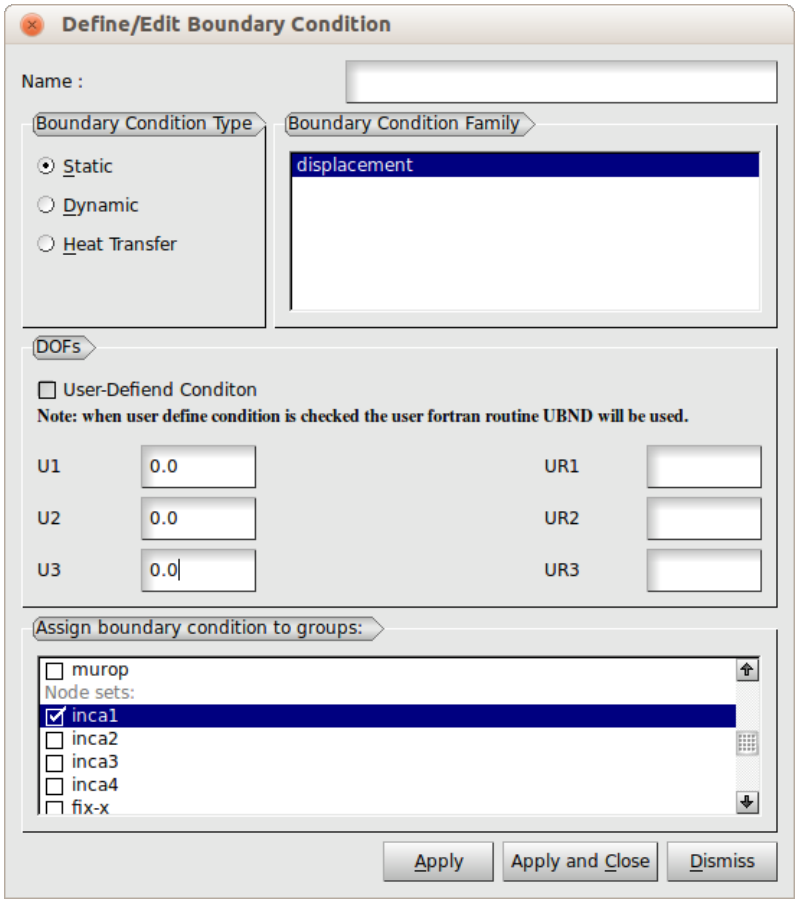
- selecting the boundary condition type according to the analysis to be performed;
- specifying the values of the degrees of freedom to be constrained;
- applying these conditions to groups of mesh elements/nodes.

*To define a boundary condition:*

1. in the **Nosa** menu select **Boundary Conditions** or click on *"Boundary Conditions"* button in the toolbar.

Figure 2.9: Boundary Conditions button

the following dialog box will appear:




2. into the *"Name"* box type the name of the boundary condition to be defined; if no name is specified, a default name *"BC-1"* is assigned.

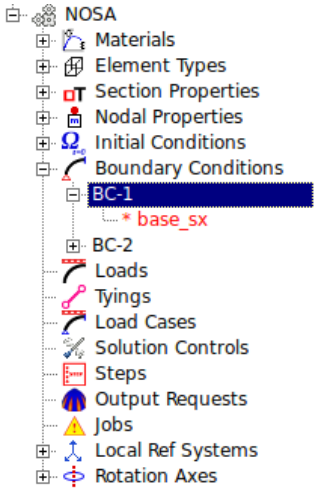
3. in the *"Boundary Condition Type"* group box select the analysis type to be performed (by clicking on one between the radio buttons available).

4. the list view of the *"Boundary Condition Family"* group box will contain only the boundary conditions associated with the specified analysis type; from this list, select the boundary condition to be applied.

5. in the *"DOFs"* group box specify the values of degrees of freedom to be constrained.

6. from the list view select one or more element/node groups which the boundary condition refers to.

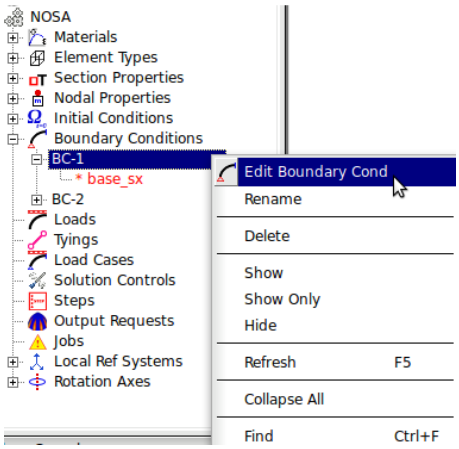
7. click on the *"Apply"* button to create the boundary condition and define a new one, or *"Apply and Close"* to create the boundary condition and quit from the dialog box. The boundary condition object so created will be stored into the **Object browser**;



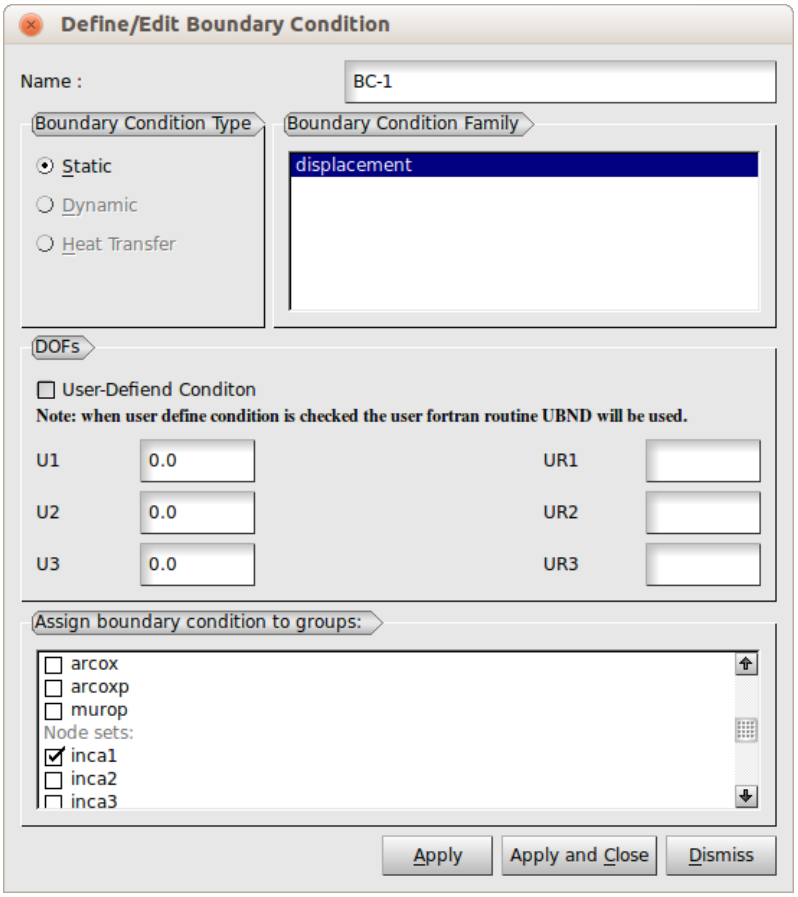

from here, it is possible to edit the object, as well as show it into the *"VTK-view"* window

### 2.8.3 Editing boundary condition

To edit a boundary condition, right click on the object from the Object browser, and select *"Edit Boundary Cond"*;



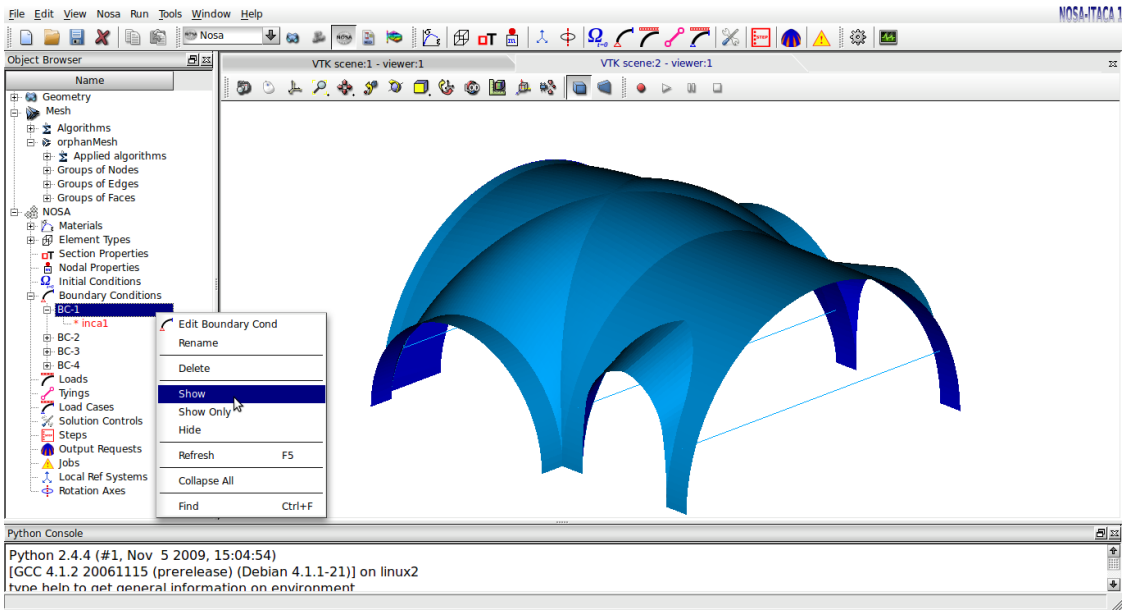

the boundary condition dialog box will appear:



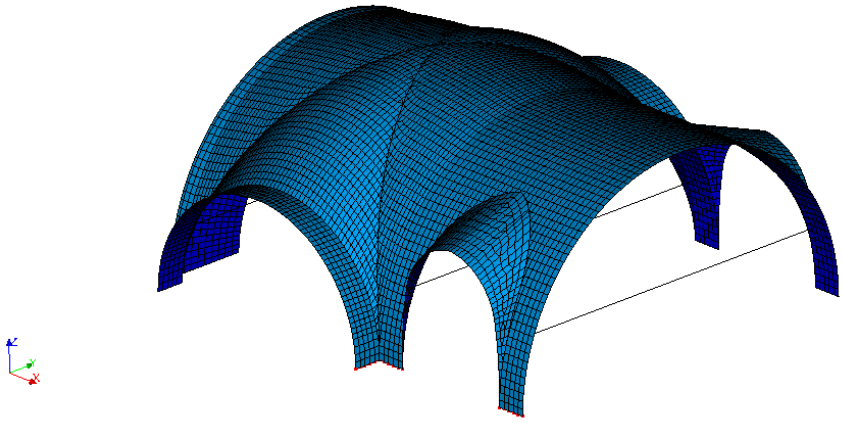

from here, you can edit the values of the degrees of freedom, and changing the element/node set to which boundary condition applies. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

### 2.8.4 Showing/Hiding boundary condition

To show a boundary condition in a VTK-view window, right click on the object from the Object browser, and select *"Show"*;




in the current VTK-view window the boundary condition is shown together with all other objects displayed:



the set of nodes or elements will be shown with **red dot-markers** and a text will be displayed specifying the values of the degrees of freedom constrained.

By selecting the commands *"Show Only"* or *"Hide"* from the popup-menu it is possible to view only the boundary condition object in the current window or hide it, respectively.

## 2.9 Define Loads

### 2.9.1 About NOSA Load Library

In **NOSA Load Library** loads are grouped into the following categories:

- analysis type (i.e. static/dynamic or heat transfer analysis);

- load type (i.e, concentrated or distributed loads);
- element type (e.g., surface load for shell elements, distributed fluxes for heat transfer elements, etc ...);

Furthermore, a load can be defined in the global reference system or in the local system of the element, and its components can be defined by means of the user routine FORCEM (see User Subroutines Reference Guide: FORCEM) or specifying the components value directly.

An identifier key is associated with each load; more precisely, for each load type there are several identifier codes depending on the above classification and the element constituents (e.g, edges or faces of an element) where load must be applied.

Further details can be found at NOSA Theory Manual: Load Library

### 2.9.2 Specifying a load type

The definition of a load consists of:

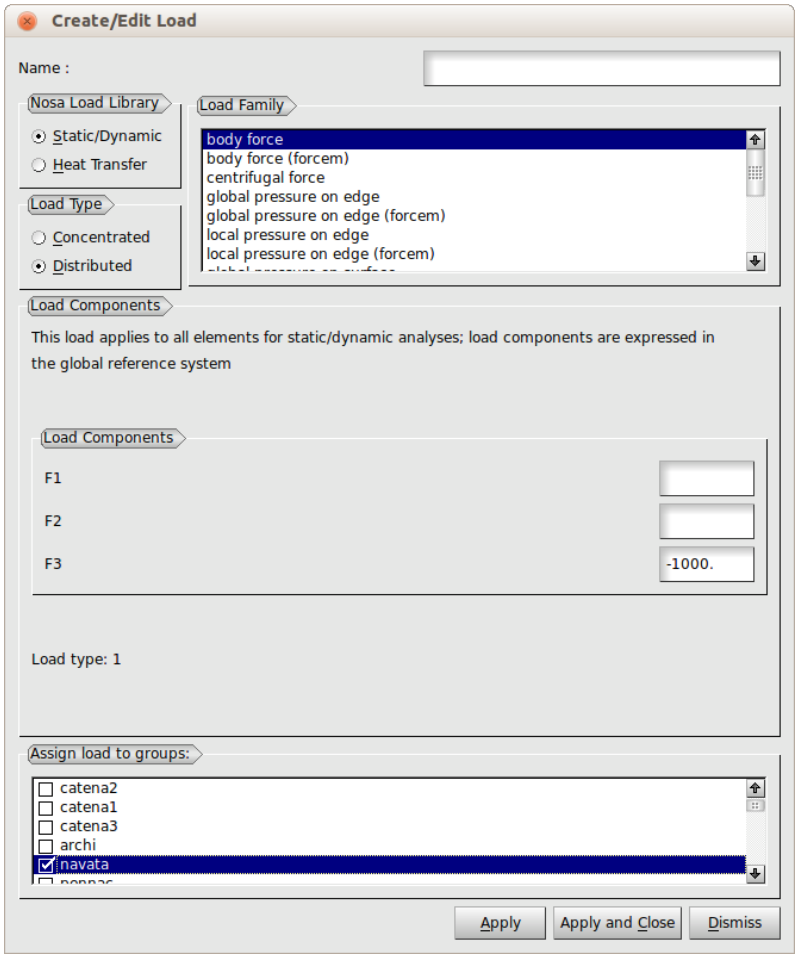
- selecting an analysis type;
- specifying the load type among those available into NOSA Load Library;
- defining the values of load components, directly or by means of the user routine FORCEM;
- specifying the mesh constituents where load is to be applied (e.g. element edges/faces/surface, nodes);
- assigning the load to groups of mesh elements or nodes.

*To define a load of the NOSA Load Library:*

1. in the **Nosa** menu select **Loads** or click on *"Loads"* button in the toolbar.

Figure 2.10: Loads button

the following dialog box will appear:




2. into the *"Name"* box type the name of the load to be defined; if no name is specified, a default name *"Load-1"* is assigned.

3. from the *"Nosa Load Library"* group box select the analysis type to be performed (by clicking on one between the radio buttons available).

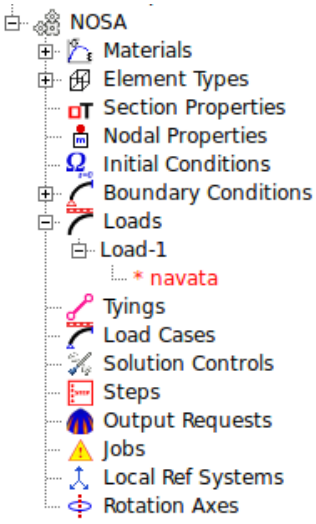
4. from the *"Load Type"* group box specify the kind of load to be applied (by clicking on one between the radio buttons available).

5. now, the list view of the *"Load Family"* group box will contain only those load type with the above specified attributes; from this list, select the load type required.

6. in the *"Load Components"* group box specify the element constituents (i.e. edges or faces) where the load is to be applied, and the components value, when they are available for this kind of load; here, a brief description of the load is supplied together with some notes and the load identifier key (according to the NOSA Load Library).

7. from the list box select one or more element/node groups which the load so defined must be applied to.

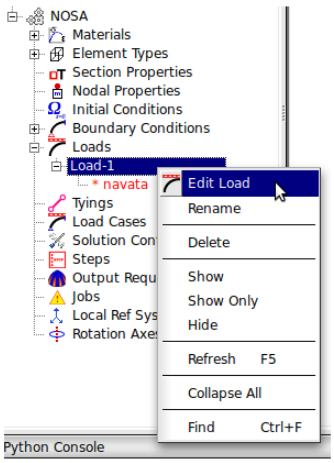
8. click on the *"Apply"* button to create the load and define a new one, or *"Apply and Close"* to create the load and quit from the dialog box. The load object so created will be stored into the **Object browser**;



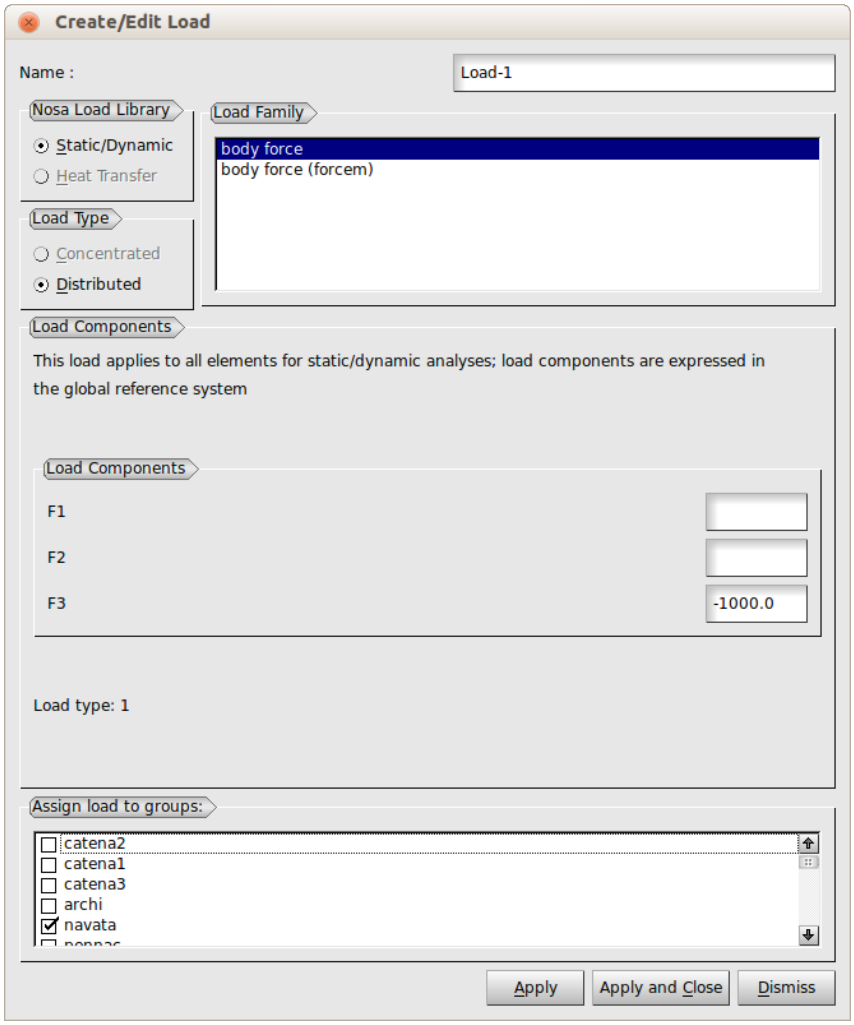

from here, it is possible to edit the object, as well as show it into the *"VTK-view"* window

### 2.9.3 Editing loads

To edit a load, right click on the load object from the Object browser, and select *"Edit Load"*;



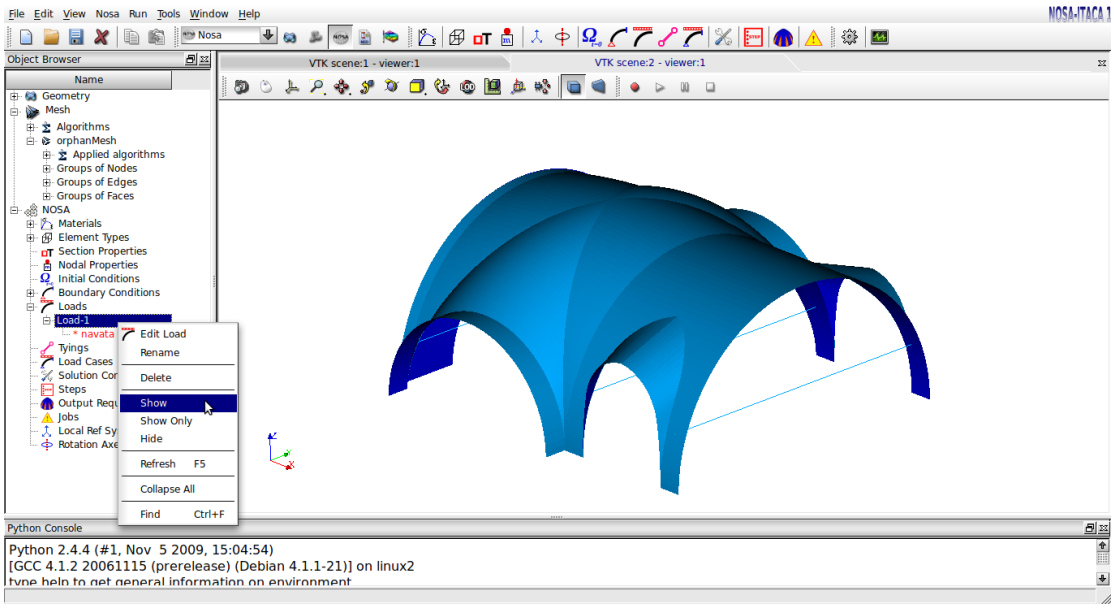

the load dialog box will appear:



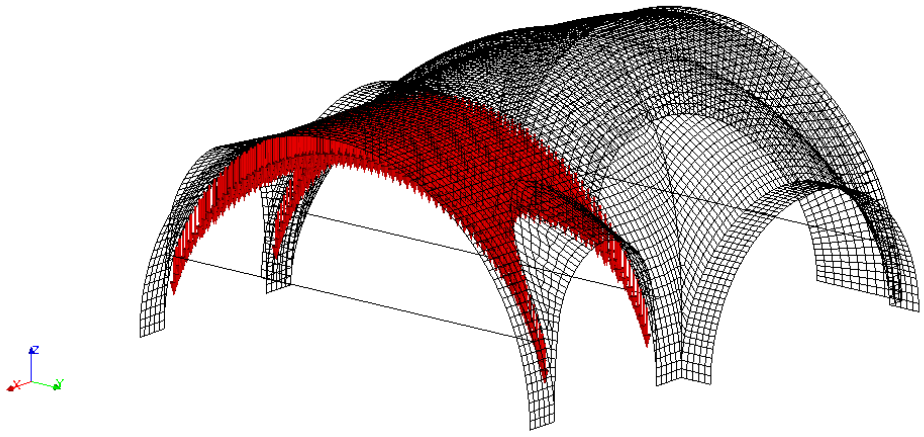

from here, you can edit the values of load components, and changing the element/node set which load applies on. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

### 2.9.4 Showing/Hiding load

To show an applied load in a VTK-view window, right click on the object from the Object browser, and select *"Show"*;




in the current VTK-view window the load is shown together with all other objects displayed:



**red-coloured arrow-glyphs** show the load direction and the mesh region where load is applied; furthermore a text will be displayed on the bottom of the window, specifying the magnitude of this load.

By selecting the commands *"Show Only"* or *"Hide"* from the popup-menu it is possible to view only the load object in the current window or hide it, respectively.

## 2.10 Define Multipoint Constraints

### 2.10.1 About Multipoint Constraints

In the field of structural finite element analysis, constraints enforce relationships between degrees of freedom (DOFs). A simple example of a constraints is the imposition Dirichlet boundary conditions which

usually consists in setting certain DOF to some known value (most commonly zero). This kind of constraints is referred to **single-point or fixed constraint** and is of the form

$$U_i = 0, \tag{2.1}$$

where $U_i$ is the displacement of a single DOF. With NOSA-ITACA software this kind of constraints can be imposed by means of the boundary conditions (see Applying boundary conditions) A next step in complexity involves **multipoint constraints** (MPCs) which relate more than one DOF including interface elements surface and contact conditions. Among linear multipoint constraints, a special role is played by the so called *master-slave constraints* (MSCs). A MSC is a condition imposed such that the displacement of a node (called the slave) depends linearly on the displacement of another node (called the master). Master-slave constraints may relate the displacements of a slave node in different directions to different master nodes. MSCs can be expressed as follows: there exist subsets

$$I_S \subset \{1, ..., n\} \quad and \quad I_{M_S} \subset \{1, ..., n\} \setminus I_S \tag{2.2}$$

such that

$$U_S = \sum_{m \in I_{M_S}} c_{sm} U_M, \qquad s \in I_S, m \in I_{M_S}. \tag{2.3}$$

The $U_s$ DOF is the slave (or tied) DOF whereas $U_m$ are the master (or guide) DOF. These constraints are also known as mutual constraints, or tying relations, and are crucial, e.g., in modeling the contact interaction between masonry and reinforcement.

MPCs can be defined in two different ways:

1 tabular form;

2 specifying the name of the sets of slave nodes and master nodes.

By using the first mode, the user must fill a table $n$ x 2, where $n$ is the number of slave nodes; for each slave node the user must specify a sequence of master nodes involved into the master-slave relationship. At this scope the user must also furnish the maximum number of master nodes involved into a relationship. In a MPC relationship only a slave node is involved; multiple slave nodes are not allowed. For a given row, in the first column of this table the node label (or identification number, id) of a slave node must be supplied; in the second column one or more master node ids (comma separated) must be supplied. If the second way of defining the MPCs is used, the user must take care that slave and master sets are ordered in accordance with the multipoint relationship; that means i-th item of the slave set is in relationship with the i-th item of the master set. It is clear that if a DOF of the slave node depends on DOFs of several master nodes, this way of defining MPCs is not suitable, and, thus, the user must use the tabular form.

### 2.10.2 Specifying a multipoint constraint

The definition of a multipoint constraint consists of:

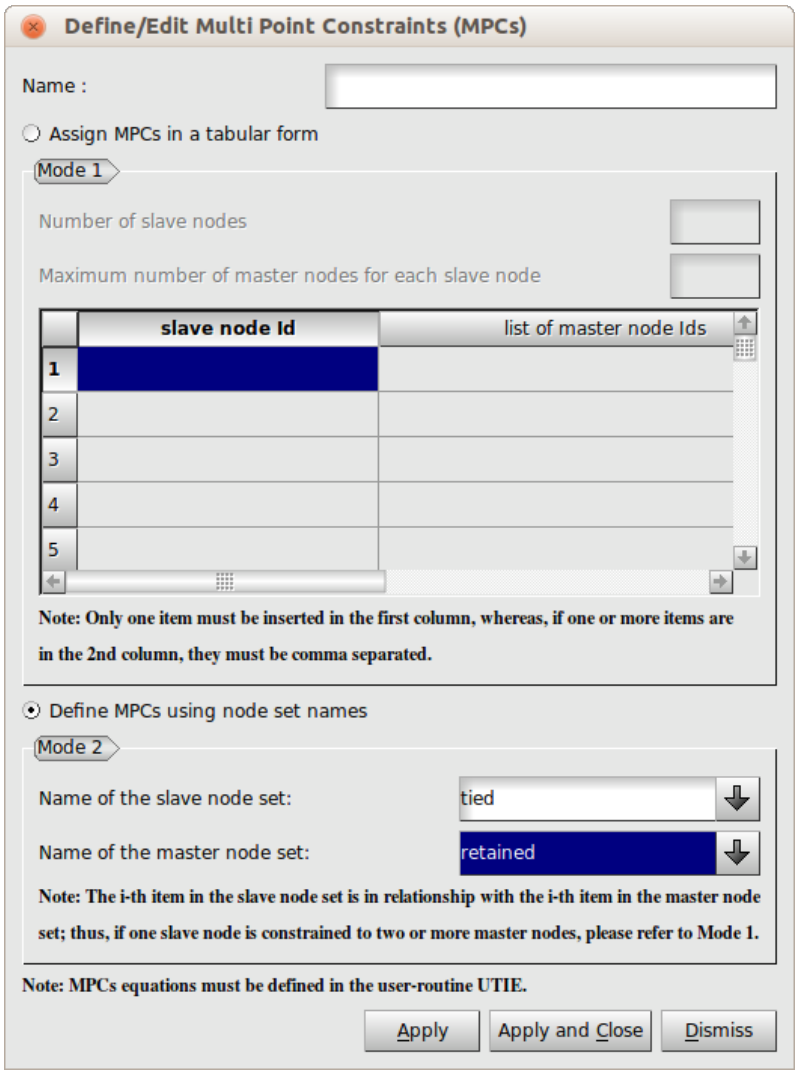
- specifying master and slave nodes according one of the way of defining the MPC (above discussed);
- defining the MPC relationship among master and slave nodes by means of the user routine UTIE (see User Subroutines Reference Guide: UTIE).

*To define a multipoint constraint:*

1. in the **Nosa** menu select **MPCs** or click on *"MPCs"* button in the toolbar.

Figure 2.11: MPCs button

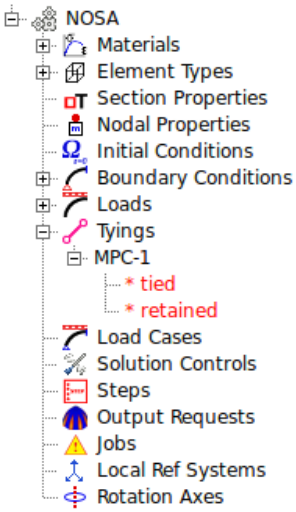
the following dialog box will appear:




2. into the *"Name"* box type the name of the MPC to be defined; if no name is specified, a default name *"MPC-1"* is assigned.

3. select the way of specifying the master and slave nodes involved in the MPC relationship (by clicking on one between two radio buttons available).

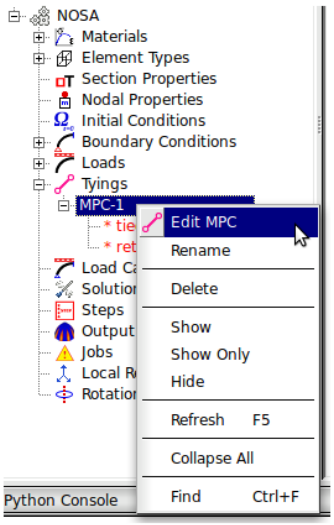
4. click on the *"Apply"* button to create the MPC and define a new one, or *"Apply and Close"* to create the MPC and quit from the dialog box. The MPC object so created will be stored into the **Object browser**;



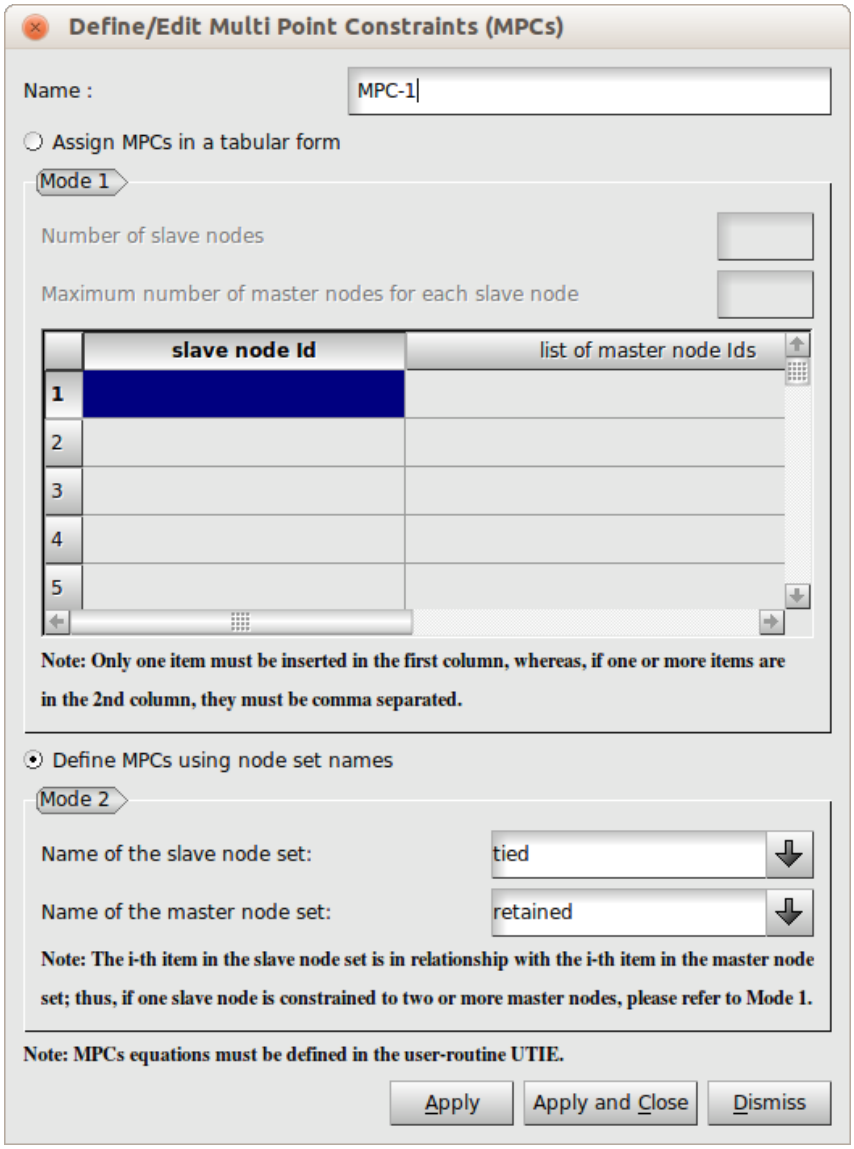

from here, it is possible to edit the object, as well as show it into the *"VTK-view"* window

### 2.10.3 Editing multipoint constraints

To edit a multipoint constraint, right click on the mpc object from the Object browser, and select *"Edit MPC"*;



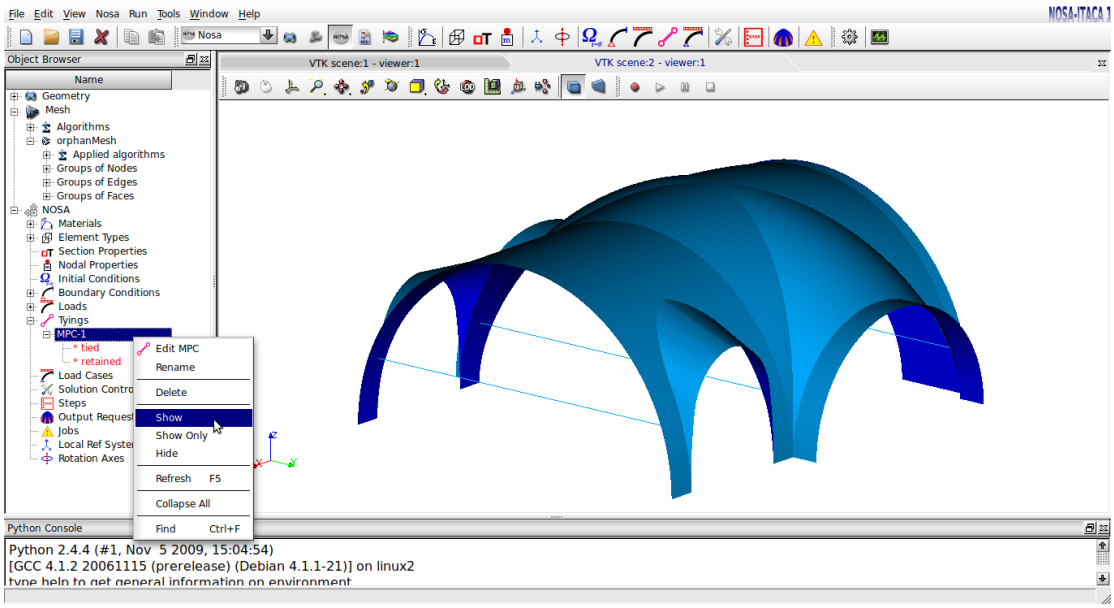

the MPC dialog box will appear:




from here, you can edit the name of the MPC, the master/slave sets and the way of specifying them. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

### 2.10.4 Showing/Hiding multipoint constraint

To show in a VTK-view window master and slave nodes involved in a multipoint constraint, right click on the object from the Object browser, and select *"Show"*;




in the current VTK-view window the MPC is shown together with all other objects displayed.

By selecting the commands *"Show Only"* or *"Hide"* from the popup-menu it is possible to view only the MPC object in the current window or hide it, respectively.

## 2.11 Set Solution Controls

### 2.11.1 About Solution Controls

Through this procedure the user can set some parameters governing the resolution of a numerical analysis; in most cases, default values of these parameters are able to achieve the solution convergence, however, there will be problems where solution parameters must be changed. For example, for non-linear analysis involving contact problems or multipoint constraints default values are no longer suitable to achieve numerical convergence.

### 2.11.2 Setting solution controls

The setting of solution parameters consists of:

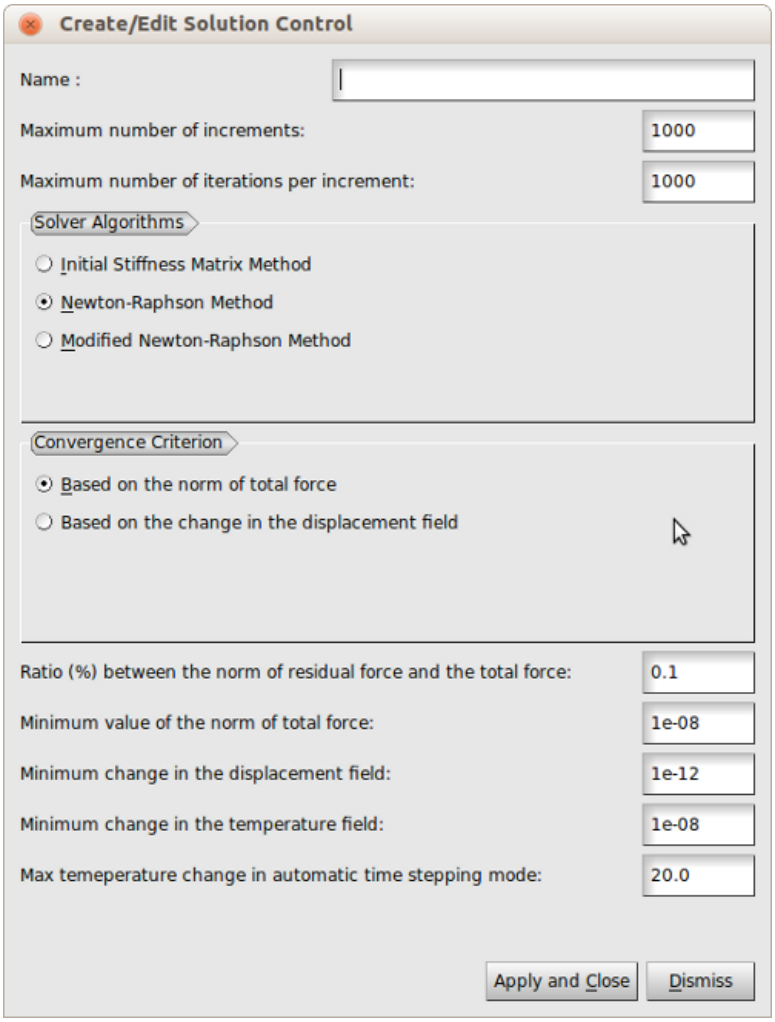
- specifying the maximum number of load increments to be applied after which analysis will be aborted;
- specifying the maximum number of iterations used to achieve numerical convergence;
- choosing the numerical algorithm to be used to perform a numerical analysis;
- specifying the convergence criteria and other parameters.

*To define solution controls* :

1. in the **Nosa** menu select **Solution Controls** or click on *"Solution Controls"* button in the toolbar.

Figure 2.12: Solution Controls button

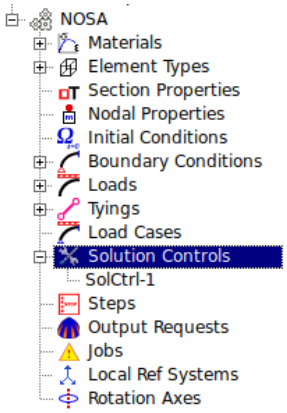
the following dialog box will appear:




2. in the *"Name"* box type the name of the solution controls to be defined; if no name is specified, a default name *"SolCtrl-1"* is assigned.

3. specify all parameters in the dialog box if their values must be different from the default ones.

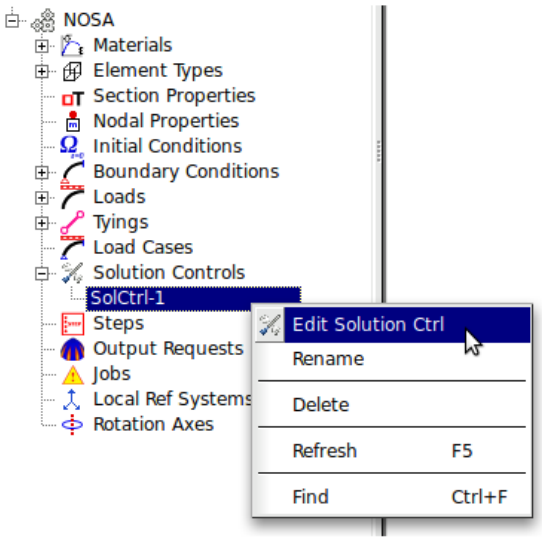
4. click on the *"Apply"* button to create the set of solution controls and define a new one, or *"Apply and Close"* to create the set of solution controls and quit from the dialog box. The solution controls object so created will be stored into the **Object browser**;



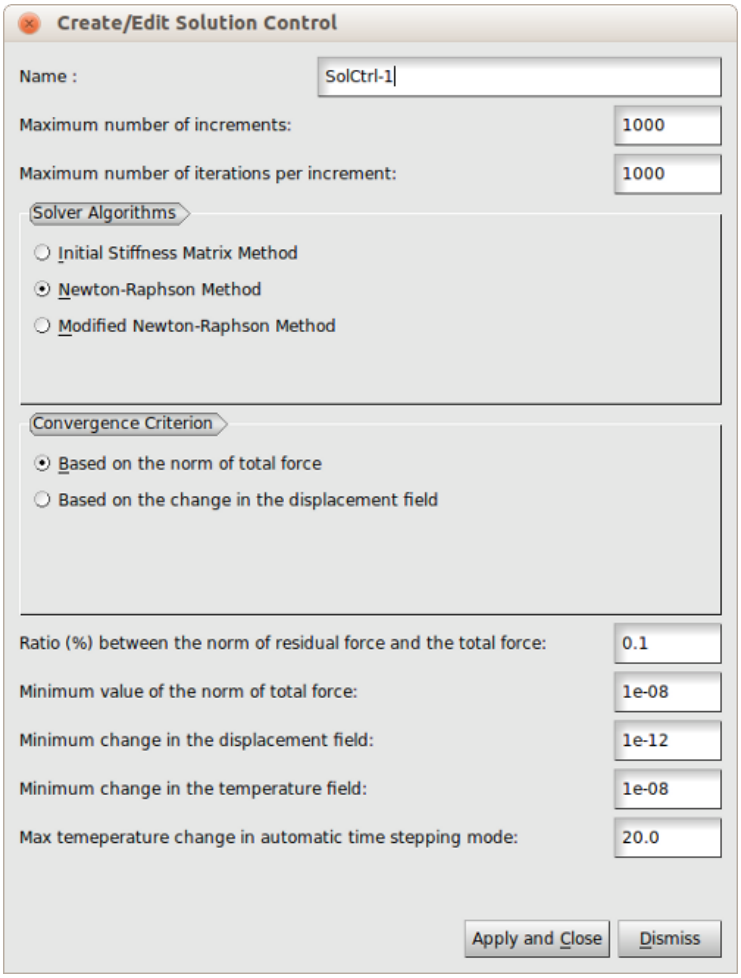

from here, it is possible to edit the object, by right clicking on it and selecting *"Edit Solution Ctrl"*

### 2.11.3 Editing solution controls

To edit a set of solution controls, right click on the object from the Object browser, and select *"Edit Solution Ctrl"*;




the following dialog box will appear:




from here, you can edit the values of solution parameters. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

## 2.12 Define Load-Steps

to be continued

## 2.13 Specify Output Requests

### 2.13.1 Overview

Through this procedure the user can select the output variables required for post-processing the results of a numerical analysis. The accessibility of a particular output variable depends on:

- the analysis type (e.g, static, dynamic, modal or heat transfer analysis);
- material type (e.g, plastic strain, crashing or fracture strain, thermal, and so on);
- mesh element type (e.g, beam elements, shells, heat transfer elements or nodes).

Furthermore, it is possible to require output variables which are not directly available from numerical calculation of the equilibrium solution; this is the case when derived field outputs are required by manipulating default variables (e.g, requiring the principal stress/strain from the output stress/strain tensor, or evaluating the maximum eccentricity in a masonry structure induced by the combined compressive and bending stress). Derived fields can be available by requiring *solution dependent variables* (**SDVs**); for them the user must provide a name for each SDV, and supply the equations which they are obtained from. These equations must be defined by means of the user routine PLOTV (see User Subroutines Reference Guide: PLOTV)

Table 2.2 shows the output variables available for a given analysis type, material and element types.

| *Field* | *Analysis type* | *Material* | *Element Type* |
|---|---|---|---|
| Stress tensor | static, dynamic | elastic-plastic, masonry | 1-4, 5*, 6-8, 9*, 10* |
| Total strain tensor | static, dynamic | elastic-plastic, masonry | 1-4, 5*, 6-8, 9*, 10* |
| Plastic strain tensor | static, dynamic | elastic-plastic | 1-4, 6-8 |
| Fracture strain tensor | static, dynamic | masonry | 1-4, 5*, 6-8, 9*, 10* |
| Crushing strain tensor | static, dynamic | masonry | 1-4, 5*, 6-8, 9*, 10* |
| Displacement | static, dynamic, modal | elastic-plastic, masonry | mesh nodes |
| Velocity | dynamic | elastic-plastic, masonry | mesh nodes |
| Acceleration | dynamic | elastic-plastic, masonry | mesh nodes |
| Reaction forces | static, dynamic, modal | elastic-plastic, masonry | mesh nodes |
| Energy | dynamic | elastic-plastic, masonry | 1-10 |
| Thermal | heat transfer | heat transfer | 11-17 |
| Stress resultants | static, dynamic | elastic, masonry | 5, 10 |
| Strain resultants | static | elastic, masonry | 5, 10 |
| SDVs | all | all | all |

Table 2.2: Overview of the output variables available in the NOSA-ITACA

The user can decide of requiring only some components of a tensor field; as an example, one can require only normal stress along a particular direction, and, thus, the other components can be disregarded. All fields defined on mesh nodes are given in the global reference system; unlike nodal fields, element variables (i.e. those defined at the integration points of the element) can be given or in the global reference system or in the local one (i.e. the othonormal reference system of the element). Thus, for shell and beam elements output variables are given in the local reference system of the element. Anyway, an element field can be expressed in a system different from the reference one, by considering linear transformations; this can be done by means of the routine PLOTV. In such a case, the element field (even though it is a default field) must be defined as derived one and, thus, the user must require a SDV field. For instance, let us consider a finite element mesh made up of only shell elements and for them we want to require the normal stress $\sigma_{11}$ along the global reference system. For this we require a SDV, named as "s11-global"; if the default stress tensor or $\sigma_{11}$ has been also required, this is given in the local reference system of the shell, while the variable "s11-global" is given in the global reference system.

For beam elements, stress and total strain tensor represent the stress and strain resultants rather than the fields calculated for other element types. Further details about output fields can be found in NOSA Theory Manual: Element Library

Usually, output fields are stored in the post file at the end of each load increment; however, the user can specify a different frequency of writing.For instance, output variables can be written every M iterations at N load increments.

Finally, for shell elements, the user can require output variables at one or more layers rather than the middle surface; this can be done specifying a sequence of layers for which output fields must be written to the post

file.

### 2.13.2 Specifying output variables

Requiring an output field consists of:

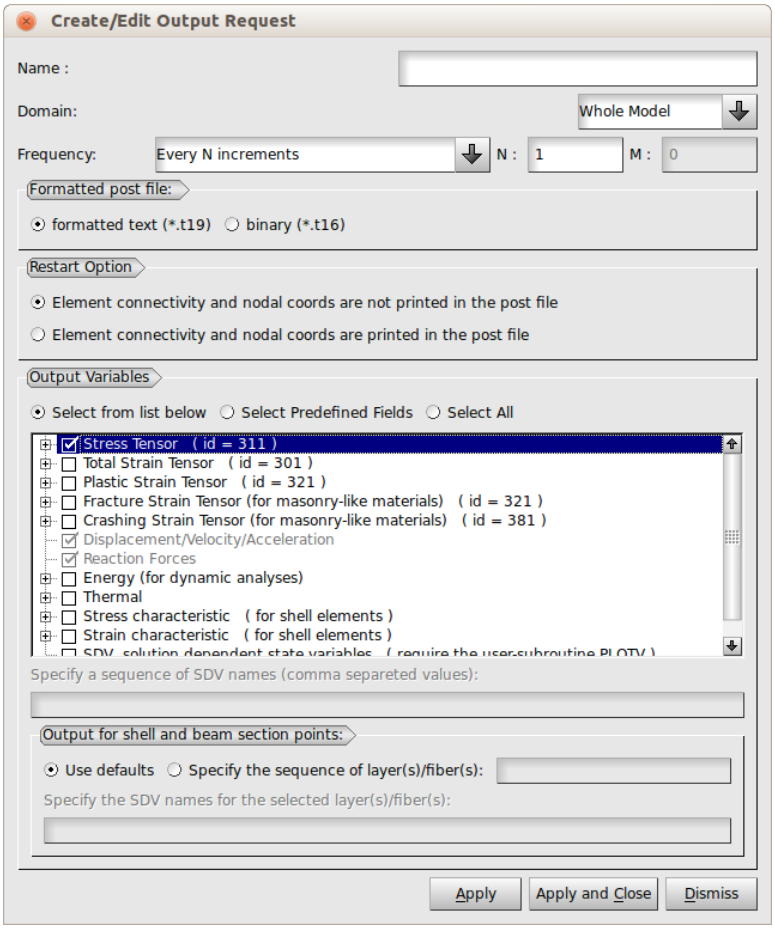
- selecting the output variables among those available for a given analysis type, material and element type;
- specifying the frequency of writing to the post files;
- selecting the format of the post file;
- specifying the layers which the fields refer to (only for shell elements);

*To require a field output:*

1. in the **Nosa** menu select **Output Requests** or click on *"Output Requests"* button in the toolbar.

Figure 2.13: Output Requests button

the following dialog box will appear:




2. in the *"Name"* box type the name of the output request to be defined; if no name is specified, a default name *"OutRequest-1"* is assigned.

3. from the drop-down list box select the frequency of writing, among the options available; the user can decide to store output variables at the end of every N load increments or at every M iterations of every N load increments.

4. specify the format of the post file (i.e., formatted text file or binary file); default value is formatted ascii file.

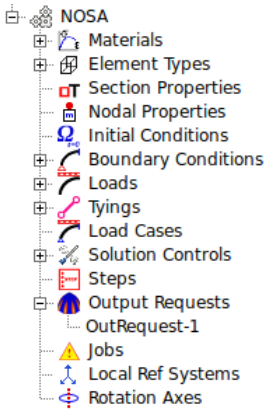
5. specify the options for a restart analysis; when a restart analysis is not required, the user can disregard these options. On the other hand, for a restart analysis it is important that node coordinates and element connectivity are stored in a post file; in this case the user must select the non-default option.

6. from the list view select the output variables required for post-processing numerical results of the analysis.

7. if a SDV variable is required, specify its name.

8. for shell elements select the layers for which the variables are to be written to the post file; by default, output fields refer to the middle surface of the shell.

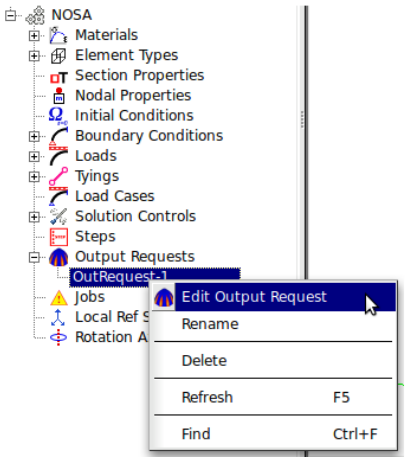
9. click on the *"Apply"* button to create the output request and define a new one, or *"Apply and Close"* to create the output request and quit from the dialog box. The output request object so created will be stored into the **Object browser**;



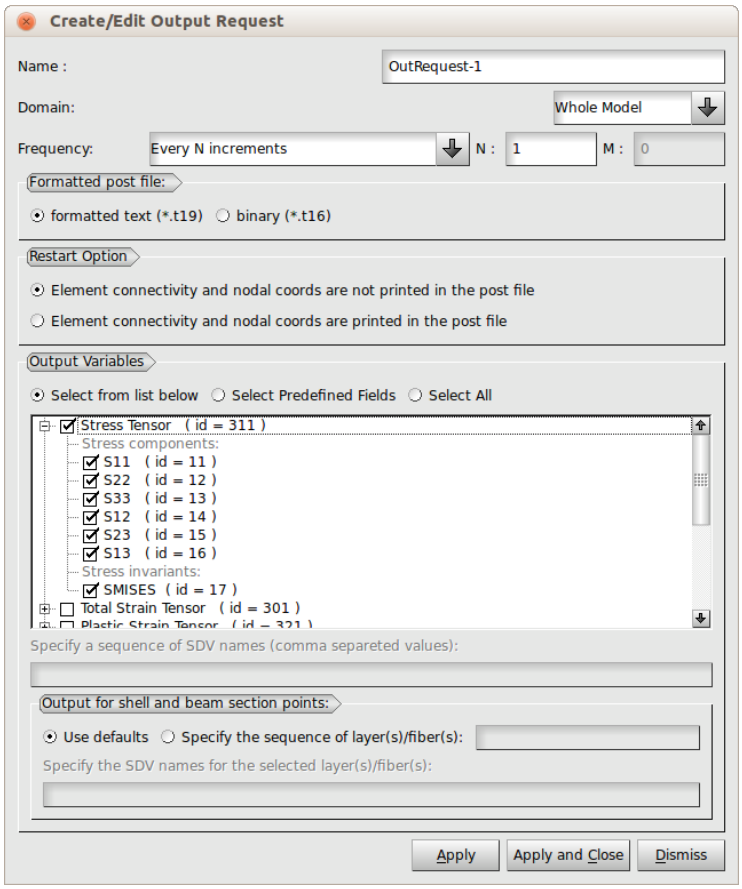

from here, it is possible to edit the object, by right clicking on it and selecting *"Edit Output Request"*

### 2.13.3 Editing output requests

To edit an output request, right click on the object from the Object browser, and select *"Edit Output Request"*;




the following dialog box will appear:




from here, you can edit the output request. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

## 2.14 Define Jobs

### 2.14.1 Overview

Through this procedure the user defines the job for the current analysis to be performed. It represents the input of the NOSA solver to carry out numerical calculation and solve equilibrium equations. More precisely, the job of the current analysis consists of:

- selecting the analysis type to be run;
- for a given analysis, setting the analysis parameters;
- specifying the memory allocation for performing the analysis; this option is obsolete;
- specifying options for printing numerical results in the print file (i.e., the file with extension "*.prt");
- specifying further options.

With NOSA-ITACA code, the user can perform the following analysis types:

- static analyses;

- modal analyses;
- dynamic analyses;
- heat transfer analyses.

For a modal analysis, the user must supply the number of eigenvalues (that means the number of modes) to be evaluated; if no number is specified, NOSA solver will extract the first 10 frequencies.

For a dynamic analysis, the user must specify the algorithm to be used to perform the numerical calculation; one can choose the Newmark algorithm or the Hilber-Hughes-Taylor (HHT) method. For both of them the user must specify the parameters of the algorithm.

Other options can be specified; for instance, the user can enable:

- finite strain formulations (this option is relevant to only elastic-plastic materials);
- follower forces;
- restart analysis;
- scale option;
- stop option;
- use of lumped mass matrix for carrying out dynamic analyses;
- reduced selective integration, as far as it is recommended in finite strain analyses.

### 2.14.2 Creating jobs

Defining a job consists of:

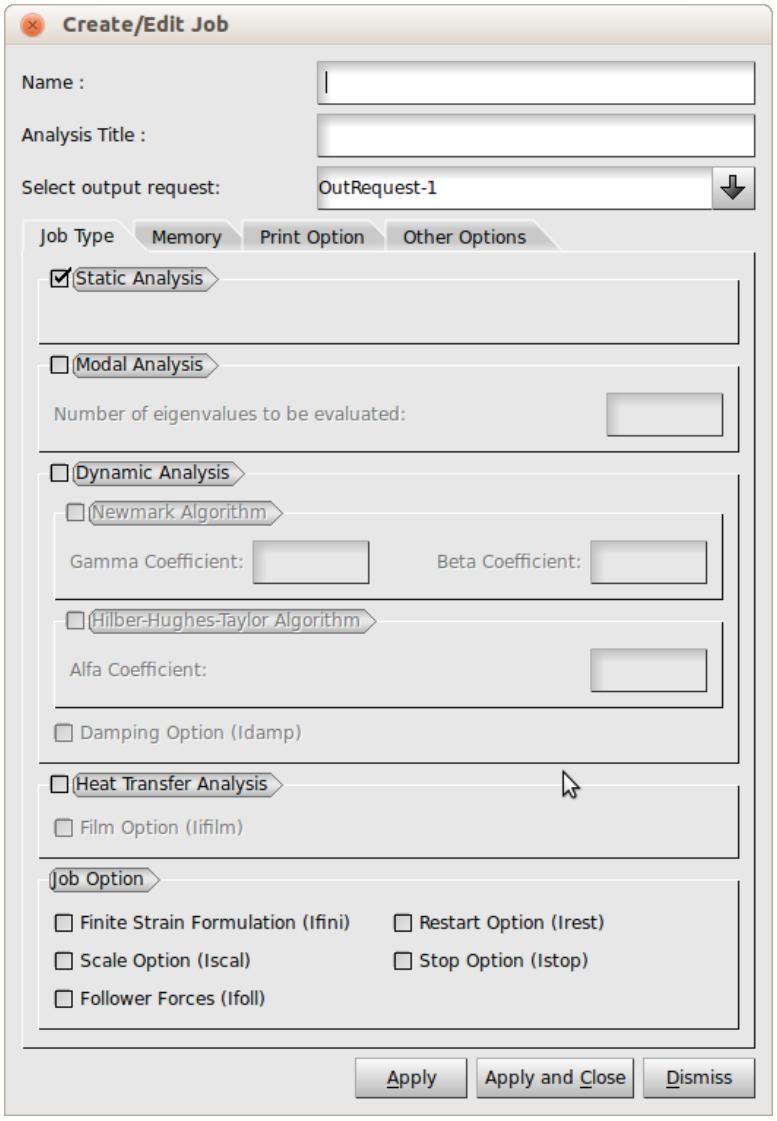
- selecting the analysis type to be performed;
- specifying the parameters of the current analysis;
- setting the desired options among those available;

*To create a job:*

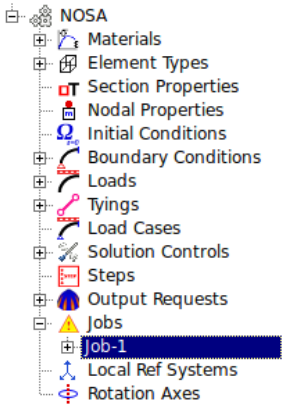
1. in the **Nosa** menu select **Jobs** or click on *"Jobs"* button in the toolbar.

Figure 2.14: Jobs button

the following dialog box will appear:



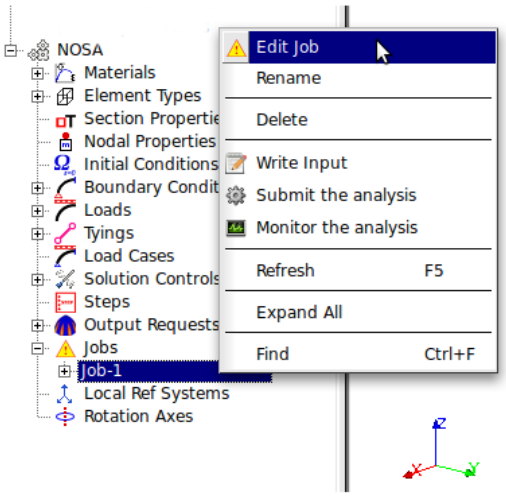

2. in the *"Name"* box type the name of the job to be run; if no name is specified, a default name *"Job-1"* is assigned.
3. in the *"Analysis Title"* box type the name of the analysis; if no name is specified, a default name *"Untitled"* is assigned.
4. from the drop-down list box select the output requests, previously defined;
5. from the *Job Type* tab select the analysis type to be run; also specify the parameters of the analysis.
6. from the *Memory* tab specify the amount of memory to be allocated to perform numerical calculation (this command is obsolete)
7. from the *Print Option* tab specify the results to be written to the prt file;
8. from the *Other Options* tab enable "lumped mass matrix" or "selective integration" options, if required.
9. click on the *"Apply"* button to create the output request and define a new one, or *"Apply and Close"* to create the output request and quit from the dialog box. The output request object so created will be stored into the **Object browser**;



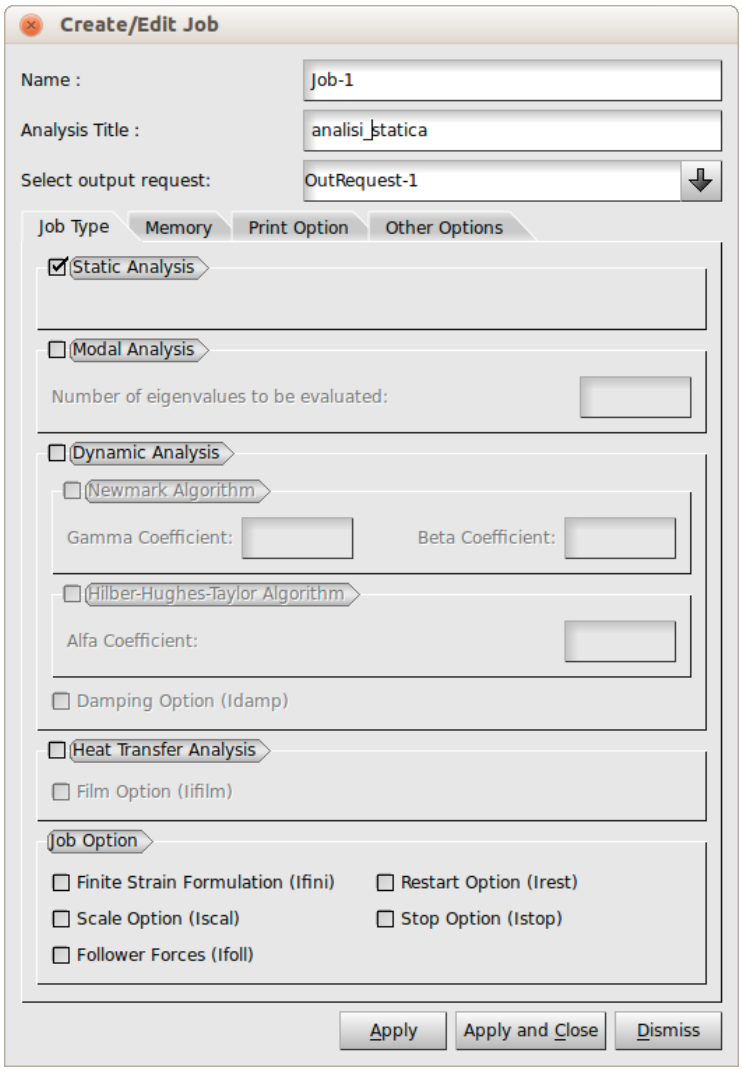

from here, it is possible to edit the object, write the input file ("*.crd"), submit the job and monitor the analysis.

### 2.14.3 Editing jobs

To edit a job, right click on the object from the Object browser, and select *"Edit Job"*;



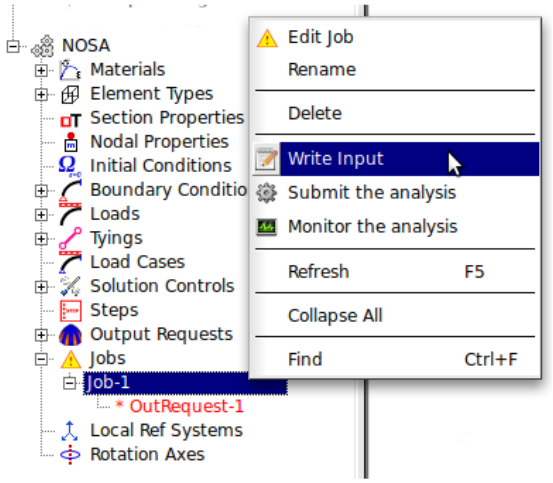

the following dialog box will appear:




from here, you can edit the job parameters. Click on *"Apply"* or *"Apply and Close"* button to store the changes.

### 2.14.4 Writing input file

Once the job has been created, the user can decide to write the crd file as input for the NOSA solver, and submit the job later; in so doing, the user can read this file and verify that all data have been written right or specify further analysis options not available in NOSA-ITACA/GUI. To write a crd file, right click on the object from the Object browser, and select *"Write Input"*;




the input file will be generated; the user can find this file at the current working directory.

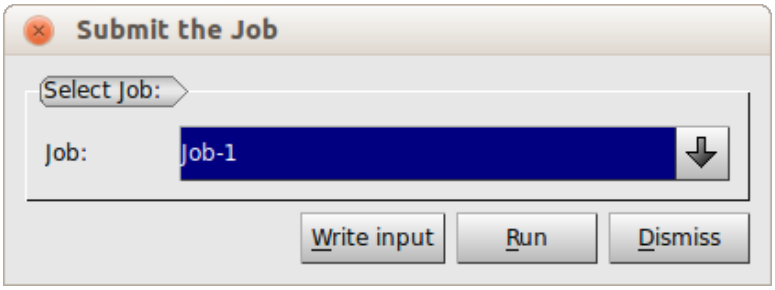
## 2.15 Submit a Job

*To submit a job:*

1. in the **Run** menu select **Submit the analysis** or click on *"Submit the analysis"* button in the toolbar.

Figure 2.15: Submit the analysis button

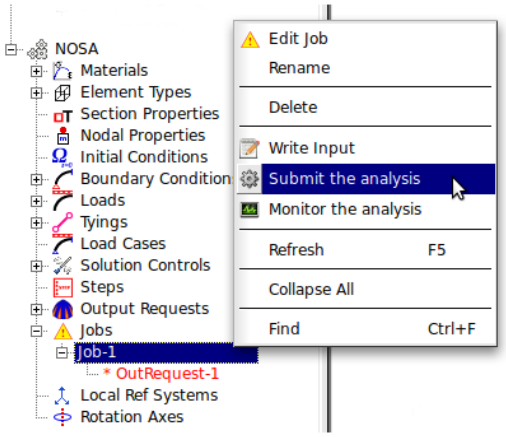
the following dialog box will appear:




2. from the drop-down list box select a job among those available in the Object Browser.

3. click on the *"Write input"* button to create the input file, or *"Run"* to submit the job.

   Alternatively, the user can select the job directly from the Object Browser and by right clicking select *"Submit the analysis"*




## 2.16 Monitor a Job

Once the job has been submitted, the user can verify the status of the current analysis; in particular, it is possible to check:

- for a given load increment, the convergence status of each iteration;
- for a given load increment, the residual at each iteration;
- warnings and/or errors arisen during the numerical calculation;

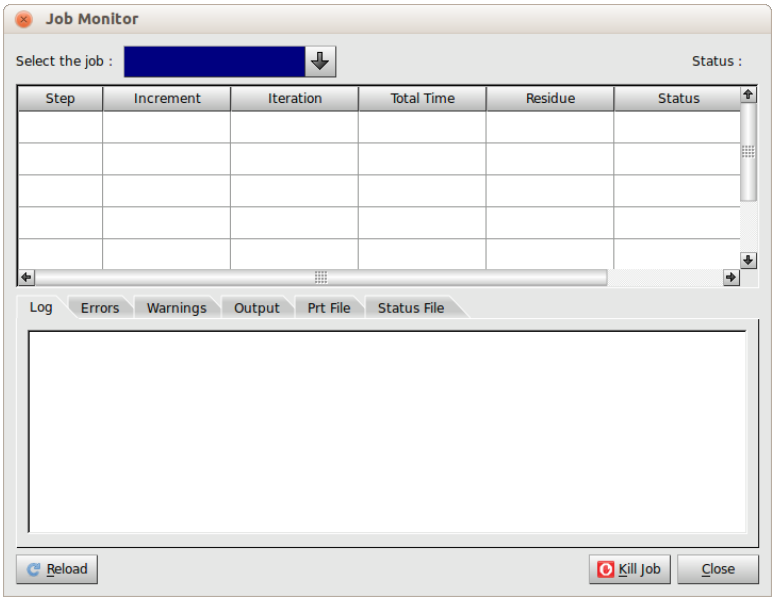
Furthermore, when the numerical calculation shows a slowly convergence or the numerical problems arise the user can abort the job.

*To monitor a job:*

1. in the **Run** menu select **Job Monitor** or click on *"Job Monitor"* button in the toolbar.

Figure 2.16: Job Monitor button

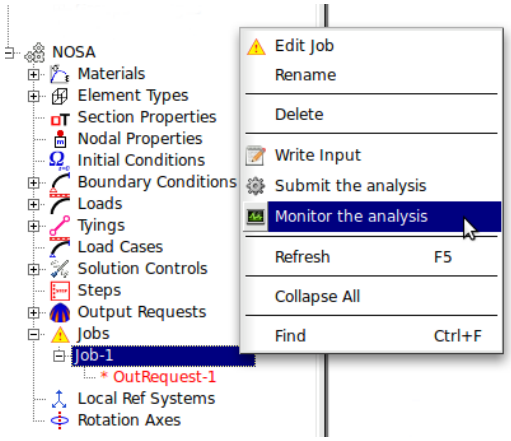
the following dialog box will appear:




2. from the drop-down list box select a job among those available in the Object Browser; the dialog box will display several information relevant to this job

3. click on the *"Reload"* button to update all information about analysis progress, or *"Kill Job"* to abort the job, when numerical problems arise.

   Alternatively, the user can select the job directly from the Object Browser and by right clicking select *"Monitor the analysis"*



# Chapter 3

# NOSA Theory Manual

## 3.1 Historical overview of the NOSA code

The finite element code **NOSA** (*Non-linear Structural Analysis*) [1] has been developed by the team of Mechanics of Materials and Structures Laboratory of the Institute of Computer Science and Technologies "A. Faedo" (ISTI-CNR). The work's primary goal has been the development of a suitable set of tools for experimentation on engineering problems involving material constitutive equations and algorithms for integrating the equation of motion as well as other numerical techniques of research interest to the group.

A first version of the code, written in the early 1980's, included plane, three-dimensional and axisymmetric isoparametric elements [2], and it was able to perform elastic-plastic analyses, under the assumption of small scale yielding (SSY), with several work-hardening models as described in [3]. The code has subsequently been extended to finite strain conditions, based on studies carried out on both constitutive equations [4 - 7] and methods of numerical integration of the equation of motion [8 - 10] in the presence of follower forces, and contact problems. At the same time, the element library of the code was enhanced of shell elements (thin and thick shells) [11].

Since 1990, nonlinear elastic materials, with null or bounded tensile strength and infinite or bounded resistance to compression, have been added to the material library of the code, in order to describe static behaviour of masonry structures; the numerical modelling of such materials (known in the literature as *masonry-like* or *no-tension* materials) has been refined over recent decades, once constitutive models and newer numerical techniques for their implementation into a finite element code have become available, and several studies [12 - 18] have led to a better understanding of the constitutive equations governing the behaviour of masonry-like materials.

Since 2000, the code has been enhanced to be able to perform nonlinear heat-conduction analyses, even in non-stationary cases, with boundary conditions concerning temperature and thermal fluxes; nowadays, the code provides for thermo-mechanical analyses of no-tension solids, whose mechanical characteristics depend on temperature in the presence of thermal loads [17, 19 - 20].

At the same time, the numerical analyses available within the code have been increased to account for dynamic problems in structures; dynamic analyses can be carried out by using Hilber-Hughes-Taylor algorithm or Newmark one [21 - 22], and the uniqueness of the solution can be restored by introducing a viscous stress depending linearly on the strain rate [23].

Within the framework of the NOSA-ITACA project, the NOSA code has been substantially modified and deeply improved; in particular, the code has been rewritten according to the FORTRAN 90-95 specifications, and equipped with new finite elements, thus enhancing its application capabilities. The element library has been completed with the triangular and tetrahedral elements, and, today, contains 35 element types. Furthermore, the performances of the code are now improved, in particular, a speed-up of about 7.3 with respect to the original version has been obtained.

In addition, the code has been enabled to perform modal analyses of structures, by means of an efficient implementation of numerical methods for constrained eigenvalue problems; the numerical implementation is based on open-source packages embedded in NOSA, such as *SPARSEKIT* [24], *ARPACK* [25] and *ICFS*. In particular, ARPACK implements a method based on the Lanczos factorization combined with spectral techniques that improve the convergence; the package ICFS provides an advanced implementation of the conjugate gradient method, accelerated with a preconditioner based on the incomplete Cholesky factorization [26].

Over these years, the development of the NOSA code has also been possible thanks to the work of many engineering students of the University of Pisa, as part of their thesis research.

## 3.2 Theoretical background of the NOSA code

This section describes the theoretical basis of the fem code; in particular, it is organized as:

- Materials, which deals with material behaviours modelled into the fem code;
- NOSA Element Library, which deals with the element types used to perform numerical analyses;
- NOSA Load Library, which describes loads to be applied on a structure;
- Analyses and Procedures, describing the numerical analyses and procedures which can be performed with the Nosa code.

## 3.3 REFERENCES

[1] DEGL'INNOCENTI S., LUCCHESI M., PADOVANI C., PAGNI A., PASQUINELLI G., ZANI N. "The finite element code NOSA Version 2.0 - User's Manual". Internal note ISTI-006/2007.

[2] HINTON E., OWEN D. R. J., Finite Element Programming, Academic Press, 1977.

[3] GUIDOTTI P, LUCCHESI M, PAGNI A., PASQUINELLI G., "Elastic-Plastic Behaviour with Work Hardening: an Appropriate Model for Structural Software". Meccanica 19, 1984.

[4] LUCCHESI M., PODIO GUIDUGLI P., "Materials with Elastic Range: a Theory with a view toward Applications. Part I". Arch. Rat. Mech. Anal., 102, pp. 23-43, 1988.

[5] LUCCHESI M., PODIO GUIDUGLI P., "Materials with Elastic Range: a Theory with a view toward Applications. Part II". Arch. Rat. Mech. Anal., 110, pp. 9-42, 1990.

[6] LUCCHESI M., OWEN D. R., PODIO GUIDUGLI P., "Materials with Elastic Range: a Theory with a view toward Applications. Part III. Approximate Constitutive Relations". Arch. Rat. Mech. Anal., 117, pp. 53-96, 1992.

[7] LUCCHESI M., PODIO GUIDUGLI P., "Materials with Elastic Range and the Possibility of Stress Oscillations in Pure Shear". Proc. Int. Conf. on Comp. Plasticity, Model, Software and Applications, Barcelona, 6-10 April 1987.

[8] GUIDOTTI P., LUCCHESI M., "A Numerical Method for Solving Boundary-Value problems in Finite Plasticity". Meccanica, 23, pp. 43-54, 1988.

[9] DEGL'INNOCENTI S., PADOVANI C., PASQUINELLI G., "An improved numerical method to integrate the equation of motion in finite elastoplasticity problems". Complas II, Second International Conference on Computational Plasticity, Barcelona, September 1989.

[10] PASQUINELLI G., "Simulation of Metal-Forming Processes by the Finite Element Method". Int. J. Plasticity, 11, pp. 623-651, 1995.

[11] GUIDOTTI P., LUCCHESI M., PAGNI A., PASQUINELLI G., "Application of Shell Theory to Structural Problem Using the Finite Element Method". Quaderni de "La Ricerca Scientifica", 115, 1986.

[12] DI PASQUALE S., "New trends in the analysis of masonry structures". Meccanica, 27, pp. 173-184, 1992.

[13] DEL PIERO G., "Constitutive equation and compatibility of the external loads for linearly-elastic masonry-like materials". Meccanica, 24 pp.150-162, 1989.

[14] LUCCHESI M., PADOVANI C., PAGNI A., "A numerical method for solving equilibrium problems of masonry-like solids". Meccanica, 24, pp. 175-193, 1994.

[15] LUCCHESI M., PADOVANI C. and PASQUINELLI G., "On the numerical solution of equilibrium problems of elastic solids with bounded tensile strength". Comput. Methods Appl. Mech. Engrg. 127, pp. 37-56, 1995.

[16] LUCCHESI M., PADOVANI C. and ZANI N., "Masonry-like materials with bounded compressive strength". Int. J. Solids Structures 33, pp. 1961-1994, 1996.

[17] LUCCHESI M., PADOVANI C., PASQUINELLI G., ZANI N. , "Masonry constructions: mechanical models and numerical applications", Series: Lecture Notes in Applied and Computational Mechanics, Vol. 39, Berlin Heidelberg, Springer-Verlag, 2008.

[18] PASQUINELLI G., "On the modeling of the reinforcement rings in masonry buildings: an example". Proceedings of the Third International Conference on Contact Mechanics, Contact Mechanics III, Madrid 1997.

[19] LUCCHESI M., PADOVANI C., PASQUINELLI G., "Thermodynamics of no-tension materials". Int. J. Solids and Structures 37, pp. 6581-6604, 2000.

[20] PADOVANI C., PASQUINELLI G., ZANI N., "A numerical method for solving equilibrium problems of no-tension solids subjected to thermal loads". Comput. Methods Appl. Mech. Engrg., 190, pp. 55-73, 2000.

[21] DEGL'INNOCENTI S., PADOVANI C. & PASQUINELLI G., "Numerical methods for the dynamic analysis of masonry structures". Structural Engineering and Mechanics, 22, pp.107-130, 2006.

[22] BATHE, JK. J., WILSON, WE. L., "Numerical methods in finite element analysis", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[23] PADOVANI C., PASQUINELLI G., ŠILHAVÝ M., "Processes in masonry bodies and the dynamical significance of collapse". Mathematics and Mechanics of Solids, 13, pp.573-610, 2008.

[24] SAAD Y., "SPARSKIT: A basic tool kit for sparse matrix computations". Technical Report, Computer Science Department, University of Minnesota, June 1994.

[25] LEHOUCQ R. B., SORENSEN D. C., YANG C., "ARPACK Users Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods". SIAM, 1998.

[26] LIN C.-J., MORE' J. J., "Incomplete Cholesky Factorizations with Limited Memory". SIAM J. Sci. Comput., 21(1), 24-45, 1999.

[27] PORCELLI M., BINANTE V., GIRARDI M., PADOVANI C., PASQUINELLI G., PELLEGRINI D., "On the numerical solution of constrained eigenvalue problems in structural engineering", Recent Advances on Optimization (Toulouse, France, 24-26 July 2013), poster session.

## 3.4 Materials

### 3.4.1 Overview

In the NOSA fem code the following material behaviours have been modelled:

- elastic materials;
- elastic-plastic materials (not available in the present version of NOSA-ITACA);
- masonry-like materials;
- heat transfer analysis (not available in the present version of NOSA-ITACA).

## 3.5 NOSA Element Library

### 3.5.1 Overview

The elements available in NOSA fem code are listed in Table 3.1

Table 3.1: Overview of the element types available in the NOSA Element Library

| *Structural type* | *Element* | Shape Functions Order | Element Topology |
|---|---|---|---|
| three-dimensional | 1 | quadratic | 20-node hexahedron |
| plane stress | 2 | quadratic | 8-node quadrilateral |
| plane strain | 3 | quadratic | 8-node quadrilateral |
| axisymmetric | 4 | quadratic | 8-node quadrilateral |
| thin shell | 5 | linear for displacements<br>quadratic for rotations | 8-node quadrilateral |
| plane strain | 6 | linear | 4-node quadrilateral |
| axisymmetric | 7 | linear | 4-node quadrilateral |
| three-dimensional | 8 | linear | 8-node hexahedron |
| straight beam | 9 | linear | 2-node segment |
| thick shell | 10 | linear | 4-node quadrilateral |
| plane heat transfer | 11 | quadratic | 8-node quadrilateral |
| plane heat transfer | 12 | linear | 4-node quadrilateral |
| axisymmetric heat transfer | 13 | quadratic | 8-node quadrilateral |
| axisymmetric heat transfer | 14 | linear | 4-node quadrilateral |
| 3D heat transfer | 15 | linear | 8-node hexahedron |
| 3D heat transfer | 16 | quadratic | 20-node hexahedron |
| heat transfer shell | 17 | linear | 4-node quadrilateral |
| plane stress | 18 | linear | 4-node quadrilateral |
| plane stress | 19 | linear | 3-node triangle |
| plane stress | 20 | quadratic | 6-node triangle |
| plane strain | 21 | linear | 3-node triangle |
| plane strain | 22 | quadratic | 6-node triangle |
| axisymmetric | 23 | linear | 3-node triangle |
| axisymmetric | 24 | quadratic | 6-node triangle |
| three-dimensional | 25 | linear | 4-node tetrahedron |
| three-dimensional | 26 | quadratic | 10-node tetrahedron |
| thick shell | 27 | linear | 3-node triangle |
| plane heat transfer | 28 | linear | 3-node triangle |
| plane heat transfer | 29 | quadratic | 6-node triangle |
| axisymmetric heat transfer | 30 | linear | 3-node triangle |
| axisymmetric heat transfer | 31 | quadratic | 6-node triangle |
| 3D heat transfer | 32 | linear | 4-node tetrahedron |
| 3D heat transfer | 33 | quadratic | 10-node tetrahedron |
| heat transfer shell | 34 | linear | 3-node triangle |
| elastic truss | 35 | linear | 2-node segment |

*Table 3.1:completed*

**Remarks:**

Currently, **only the first 17 element types** of the NOSA Element Library have been implemented into the NOSA-ITACA software; future releases will make all element types available to any applications.

### 3.5.2 Element 1: 3D 20-node hexahedron

This element is a 20-node isoparametric brick whose shape functions are quadratic; its main characteristic is that to ensure a quickly convergence for three-dimensional analyses. For thick-shell situations, one element through the thickness will usually provide an acceptable solution for both displacement and stress. The length to thickness ratio should not exceed a value of 20.

#### 3.5.2.1 Nodal Connectivity

The element has 20 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.1.

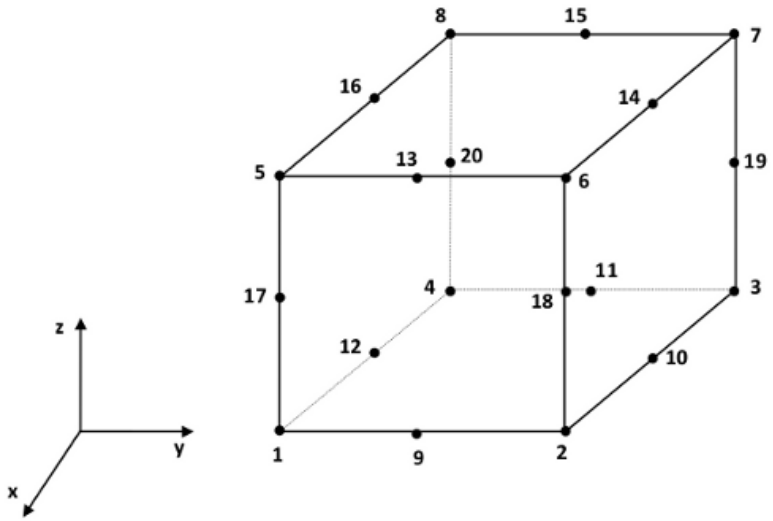



Figure 3.1: Nodal connectivity of the 20-node isoparametric hexahedric element

#### 3.5.2.2 Integration Points

The element is integrated numerically using twenty-seven points (Gaussian quadrature); the plane of the first series of 9 integration points is shown in Figure 3.2.

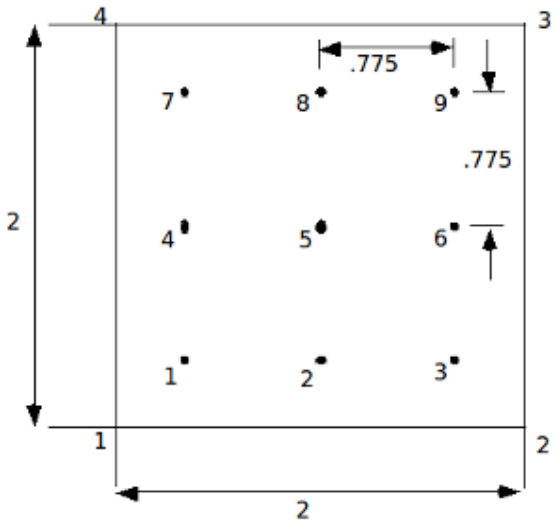



Figure 3.2: Gauss points of the 20-node isoparametric hexahedric element

#### 3.5.2.3 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.2

Table 3.2: Distributed loads relevant to the 20-node hexahedric element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the three components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the three global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis must be defined, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2-3-4 face**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2-3-4 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 1-2 direction; the third component is orthogonal to the face and is positive if directed towards the interior of the element, the second component has the direction given by the vector product of the first and third directions. |
| 13 | **Pressure on the 1-2-3-4 face** in the global reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2-3-4 face** in the local reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 5-8-7-6 face**; the force per unit area is defined in the global reference system. |

*Table 3.2:continue in the next page*

*Table 3.2:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 22 | **Pressure on the 5-8-7-6 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 5-8 direction; the third component is orthogonal to the face and is positive if directed towards the interior of the element. The second component has the direction given by the vector product of the first and third directions. |
| 23 | **Pressure on the 5-8-7-6 face** in the global reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 5-8-7-6 face** in the local reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 1-5-6-2 face**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 1-5-6-2 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 1-5 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 33 | **Pressure on the 1-5-6-2 face** in the global reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 1-5-6-2 face** in the local reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 2-6-7-3 face**; the force per unit area is defined in the global reference system. |
| 42 | **Pressure on the 2-6-7-3 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 2-6 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 43 | **Pressure on the 2-6-7-3 face** in the global reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 44 | **Pressure on the 2-6-7-3 face** in the local reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 51 | **Pressure on the 3-7-8-4 face**; the force per unit area is defined in the global reference system. |
| 52 | **Pressure on the 3-7-8-4 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 3-7 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 53 | **Pressure on the 3-7-8-4 face** in the global reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 54 | **Pressure on the 3-7-8-4 face** in the local reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |

*Table 3.2:continue in the next page*

*Table 3.2:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 61 | **Pressure on the 4-8-5-1 face**; the force per unit area is defined in the global reference system. |
| 62 | **Pressure on the 4-8-5-1 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 4-8 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 63 | **Pressure on the 4-8-5-1 face** in the global reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 64 | **Pressure on the 4-8-5-1 face** in the local reference system; the three components of the force per unit area on the eight nodes of the face are to be specified in the user subroutine **FORCEM**. |

*Table 3.2:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.2.4 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangular prism or a tetrahedron.

#### 3.5.2.5 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.2.6 Degrees of freedom

Three degrees of freedom u, v, w (the displacements along the global coordinate directions).

#### 3.5.2.7 Field Output

1 = xx, 2 = yy, 3 = zz, 4 = xy, 5 = yz, 6 = xz.

#### 3.5.2.8 Analysis types

- Infinitesimal elasto-plasticity.
- Masonry-like materials.

### 3.5.3 Element 2: Plane stress 8-node quadrilateral

This element is a 8-node isoparametric quadrilateral whose shape functions are quadratic.

#### 3.5.3.1 Nodal Connectivity

The element has 8 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.3.

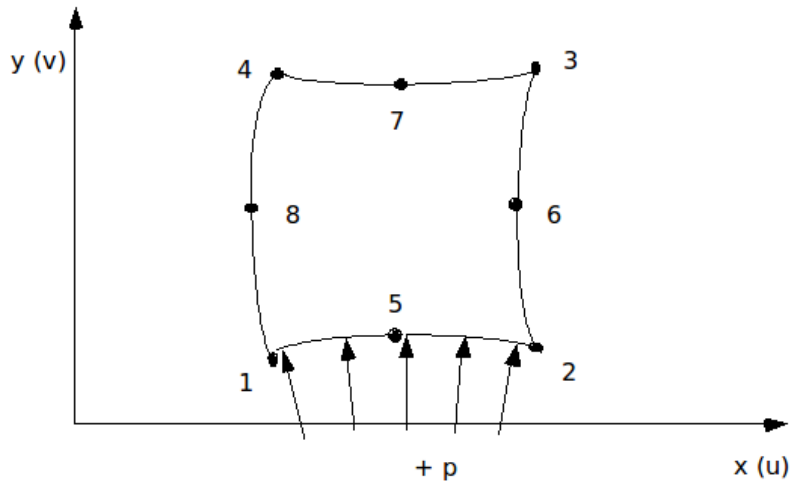



Figure 3.3: Nodal connectivity of the 8-node isoparametric quadrilateral element

#### 3.5.3.2 Integration Points

The element is integrated numerically using nine points (Gaussian quadrature); the integration points are shown in Figure 3.4.

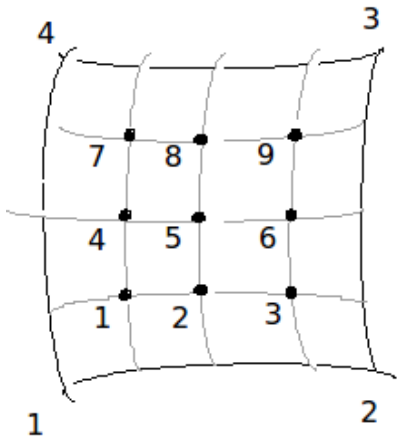



Figure 3.4: Gauss points of the 8-node isoparametric quadrilateral element

#### 3.5.3.3 Geometrical Attributes

For this element the user must specify the thickness of the element cross-section; by default, a thickness equal to 1 is assigned.

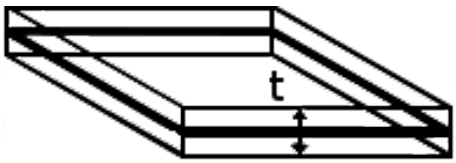



Figure 3.5: Element thickness

#### 3.5.3.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.3

Table 3.3: Distributed loads relevant to the 8-node plane stress element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis is orthogonal to the plane of the element, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2-5 edge**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2-5 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |
| 13 | **Pressure on the 1-2-5 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2-5 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3-6 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3-6 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3-6 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3-6 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-4-7 edge**; the force per unit area is defined in the global reference system. |

*Table 3.3:continue in the next page*

*Table 3.3:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 32 | **Pressure on the 3-4-7 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-4 direction. |
| 33 | **Pressure on the 3-4-7 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-4-7 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 4-1-8 edge**; the force per unit area is defined in the global reference system. |
| 42 | **Pressure on the 4-1-8 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 4-1 direction. |
| 43 | **Pressure on the 4-1-8 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 44 | **Pressure on the 4-1-8 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.3:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.3.5 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.3.6 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.3.7 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.3.8 Field Output

1 = xx, 2 = yy, 3 = xy.

#### 3.5.3.9 Analysis types

- Infinitesimal elasto-plasticity.

- Masonry-like materials.

### 3.5.4 Element 3: Plane strain 8-node quadrilateral

This element is a 8-node isoparametric quadrilateral whose shape functions are quadratic.

#### 3.5.4.1 Nodal Connectivity

The element has 8 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.6.

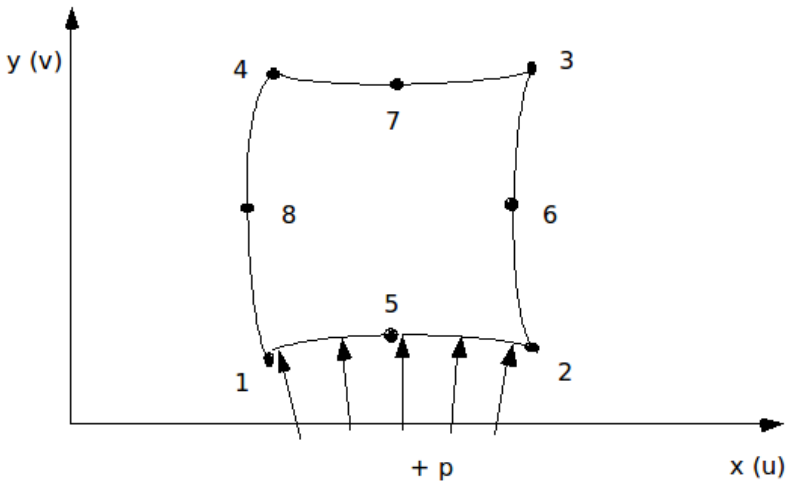



Figure 3.6: Nodal connectivity of the 8-node isoparametric quadrilateral element

#### 3.5.4.2 Integration Points

The element is integrated numerically using nine points (Gaussian quadrature); the integration points are shown in Figure 3.7.

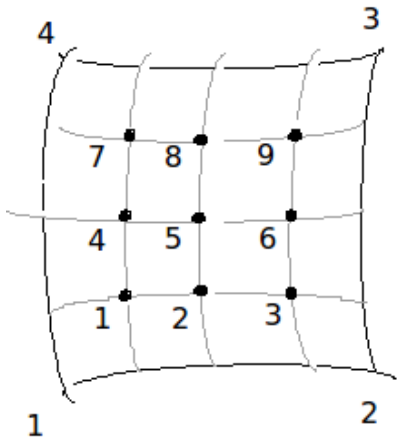



Figure 3.7: Gauss points of the 8-node isoparametric quadrilateral element

#### 3.5.4.3 Geometrical Attributes

For this element the user must specify the thickness of the element; by default, a thickness equal to 1 is assigned.

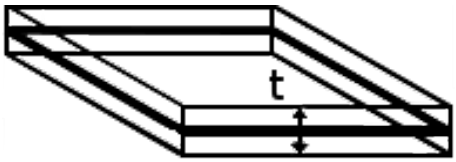



Figure 3.8: Element thickness

#### 3.5.4.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.4.

Table 3.4: Distributed loads relevant to the 8-node plane strain element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis is orthogonal to the plane of the element, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2-5 edge**; the force per unit area is defined in the global reference system. |

*Table 3.4:continue in the next page*

*Table 3.4:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 12 | **Pressure on the 1-2-5 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |
| 13 | **Pressure on the 1-2-5 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2-5 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3-6 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3-6 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3-6 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3-6 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-4-7 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-4-7 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-4 direction. |
| 33 | **Pressure on the 3-4-7 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-4-7 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 4-1-8 edge**; the force per unit area is defined in the global reference system. |
| 42 | **Pressure on the 4-1-8 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 4-1 direction. |
| 43 | **Pressure on the 4-1-8 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 44 | **Pressure on the 4-1-8 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.4:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.4.5 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.4.6 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.4.7 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.4.8 Field Output

1 = xx, 2 = yy, 3 = xy.

#### 3.5.4.9 Analysis types

- Infinitesimal elasto-plasticity.
- Masonry-like materials.

### 3.5.5 Element 4: Axisymmetric 8-node quadrilateral

This element is a 8-node isoparametric quadrilateral whose shape functions are quadratic.

#### 3.5.5.1 Nodal Connectivity

The element has 8 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.9.

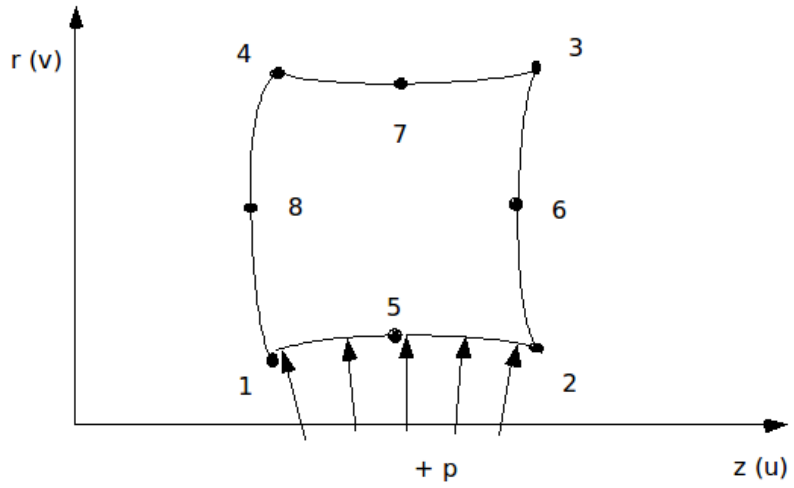



Figure 3.9: Nodal connectivity of the 8-node isoparametric quadrilateral element

#### 3.5.5.2 Integration Points

The element is integrated numerically using nine points (Gaussian quadrature); the integration points are shown in Figure 3.10.

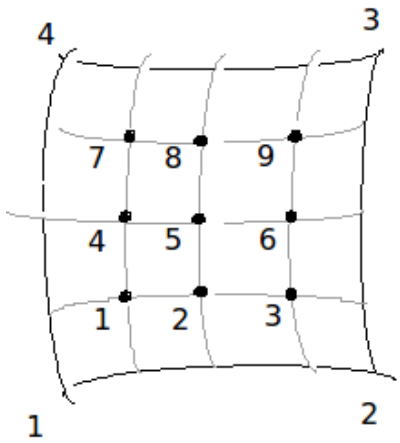



Figure 3.10: Gauss points of the 8-node isoparametric quadrilateral element

#### 3.5.5.3 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.5.

Table 3.5: Distributed loads relevant to the 8-node axisymmetric element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; The rotation axis coincides with the axis of symmetry, the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2-5 edge**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2-5 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |
| 13 | **Pressure on the 1-2-5 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2-5 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3-6 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3-6 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3-6 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3-6 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-4-7 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-4-7 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-4 direction. |
| 33 | **Pressure on the 3-4-7 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-4-7 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 4-1-8 edge**; the force per unit area is defined in the global reference system. |

*Table 3.5:continue in the next page*

*Table 3.5:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 42 | **Pressure on the 4-1-8 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 4-1 direction. |
| 43 | **Pressure on the 4-1-8 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 44 | **Pressure on the 4-1-8 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.5:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.5.4 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.5.5 Nodal Coordinates

Two global coordinates in the z and r directions.

#### 3.5.5.6 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.5.7 Field Output

1 = zz, 2 = rr, 3 = $\theta\theta$, 4 = zr.

#### 3.5.5.8 Analysis types

- Infinitesimal elasto-plasticity.
- Masonry-like materials.

### 3.5.6 Element 5: Thin shell 8-node quadrilateral

This element is a 8-node isoparametric quadrilateral, whose shape functions are linear for displacement and quadratic for rotations.

#### 3.5.6.1 Nodal Connectivity

The element has 8 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.11.

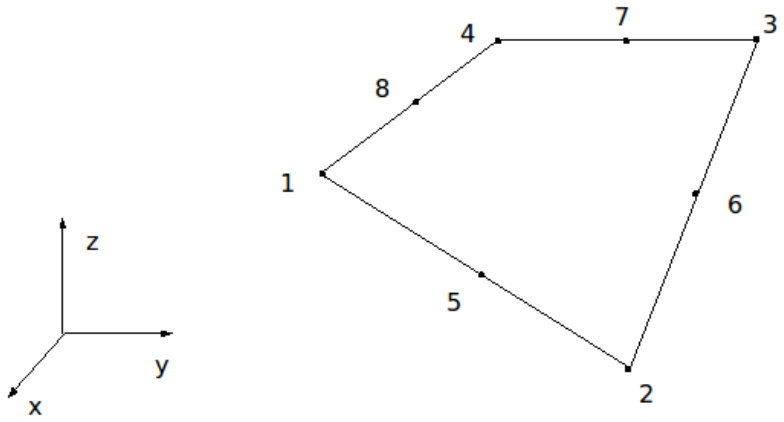



Figure 3.11: Nodal connectivity of the 8-node isoparametric quadrilateral element

#### 3.5.6.2 Integration Points

Numerical integration is performed both on the middle surface of the element and within the thickness; for the former integration four-point Gaussian quadrature is used, whose integration points are shown in Figure 3.12. The latter integration is performed using the Simpson method, with the number of integration points depending on the number of layers of the shell (see SHELL SECT; the default value is 3, the maximum number of layers is 99).

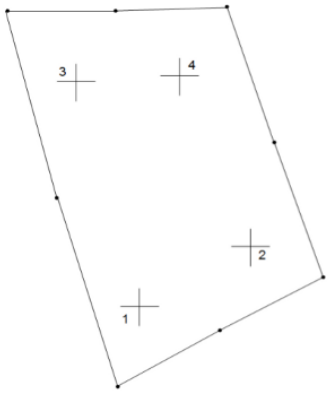



Figure 3.12: Gauss points of the 8-node isoparametric quadrilateral element

#### 3.5.6.3 Geometrical Attributes

For this element the user must specify the thickness(es) of the element cross-section; for homogeneous shells it is enough to define the thickness of the whole cross-section. For non-homogeneous shells, the thickness of each layer must be provided (COMPOSITE option).

The thickness of the whole shell or each layer can be specified at each corner node by using the user routine **UGEOM**.

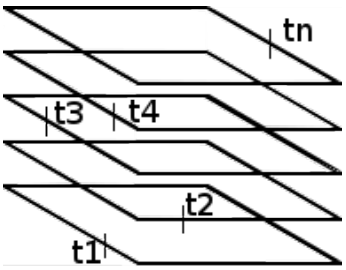



Figure 3.13: Layers thickness for composite shells

#### 3.5.6.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.6.

Table 3.6: Distributed loads relevant to the 8-node thin shell element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the three components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the three global components of the force per unit volume for each shell layer are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis must be defined (see ROTATION AXIS), and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit length is defined in the global reference system. |
| 13 | **Pressure on the 1-2 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit length is defined in the global reference system. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-4 edge**; the force per unit length is defined in the global reference system. |
| 33 | **Pressure on the 3-4 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 4-1 edge**; the force per unit length is defined in the global reference system. |
| 43 | **Pressure on the 4-1 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 51 | **Pressure on the surface** of the element; the force per unit area is defined in the global reference system. |
| 52 | **Pressure on the surface** of the element in the local reference system, as defined below. |

*Table 3.6:continue in the next page*

*Table 3.6:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 53 | **Pressure on the surface** of the element in the global reference system; the three components of the force per unit area on the four nodes of the element are to be specified in the user subroutine **FORCEM**. |
| 54 | **Pressure on the surface** of the element in the local reference system; the three components of the force per unit area on the four nodes of the element are to be specified in the user subroutine **FORCEM**. |

*Table 3.6:completed*

Concentrate loads are point loads applied on corner nodes; moreover, the element may be subjected to concentrated moments on mid-side nodes and thermal dilatation loads.

#### 3.5.6.5 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle; in such a case, the collapsed edge has zero rotational stiffness.

#### 3.5.6.6 Nodal Coordinates

Three global coordinates in the x, y and z directions; the coordinates of the midside nodes are recalculated inside the program in such a way to put them halfway between the corresponding corner nodes.

#### 3.5.6.7 Degrees of freedom

The corner nodes have three degrees of freedom u, v and w (the displacements along the global coordinate directions); the degree of freedom of the mid-side nodes is the rotation q about the corresponding edge (positive if counterclockwise as seen by the corner node with the higher (external) node number).

#### 3.5.6.8 Field Output

Strains are printed for the four Gauss points of the mean surface of the shell. The components of the strain are given in the local orthonormal reference system s, t, n in the following order:

1 = ss, 2 = tt, 3 = st.

The three values of stress are printed for the four integration points for each layer of the element. The stress components are given in the local orthonormal reference system s, t, n in the following order:

1 = ss, 2 = tt, 3 = st.

#### 3.5.6.9 Analysis types

- Linear elasticity.
- Masonry-like materials.

#### 3.5.6.10 Local reference frame

The element is defined geometrically by the (x, y, z) coordinates of the four corner nodes.

Due to the bilinear interpolation, the surface forms a ruled surface which is allowed to degenerate to a plate.

The stress-strain output is given in local orthogonal surface directions $v_1$, $v_2$, and $v_3$ which, for the centroid, are defined in the following way:

First, the vectors tangent to the curves with constant isoparametric coordinates $\xi_1$ and $\xi_2$ are normalized

$t_1 = \frac{\frac{\partial r}{\partial \xi_1}}{\|\frac{\partial r}{\partial \xi_1}\|}, t_2 = \frac{\frac{\partial r}{\partial \xi_2}}{\|\frac{\partial r}{\partial \xi_2}\|},$

Now a new basis is being defined as:

$s = t_1 + t_2, d = t_1 - t_2,$

After normalizing these vectors by:

$\hat{s} = \frac{s}{\sqrt{2\|s\|}}, \hat{d} = \frac{d}{\sqrt{2\|d\|}}.$

The local orthogonal directions are then obtained as:

$v_1 = \hat{s} + \hat{d}, v_2 = \hat{s} - \hat{d}, v_3 = v_1 \times v_2.$

In this way, the vectors $t_1$, $t_2$ and $v_1$, $v_2$ have the same bisecting plane.

The local directions for the Gaussian integrations points are found by projection of the centroid directions. Hence, if the element is flat, the directions at the Gauss points are identical to those at the centroid.

### 3.5.7 Element 6: Plane strain 4-node quadrilateral

This element is a 4-node isoparametric quadrilateral, whose shape functions are bilinear.

#### 3.5.7.1 Nodal Connectivity

The element has 4 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.14.

#### 3.5.7.2 Integration Points

The element is integrated numerically using four points (Gaussian quadrature); the integration points are shown in Figure 3.14.

If selective reduced integration is required, the hydrostatic part of the strain is calculated using one integration point, namely the centroid of the element.

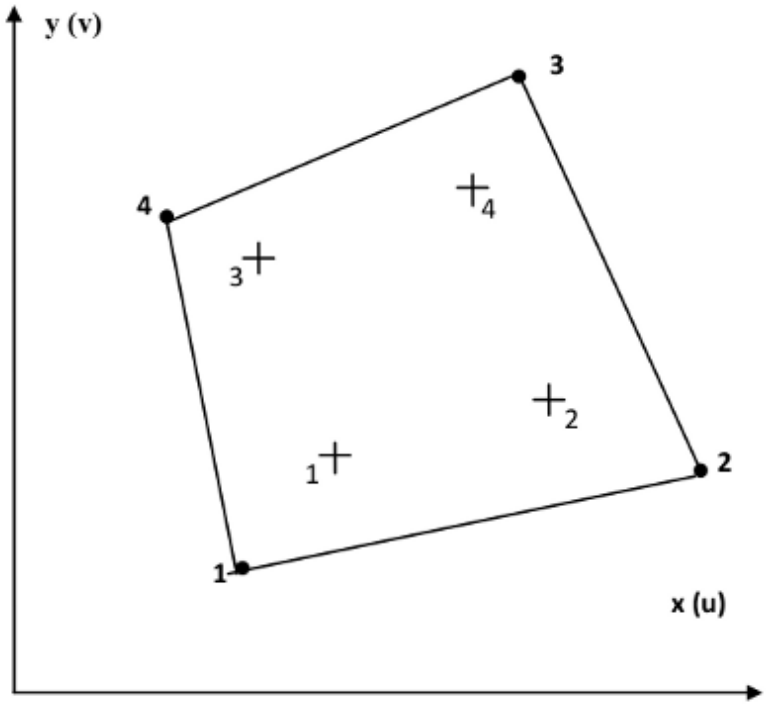



Figure 3.14: Nodal connectivity and integration points of the 4-node isoparametric quadrilateral element

#### 3.5.7.3 Geometrical Attributes

For this element the user must specify the thickness of the element; by default, a thickness equal to 1 is assigned.

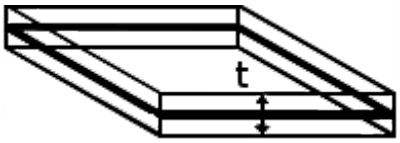



Figure 3.15: Element thickness

#### 3.5.7.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.7.

Table 3.7: Distributed loads relevant to the 4-node plane strain element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |

*Table 3.7:continue in the next page*

*Table 3.7:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis is orthogonal to the plane of the element, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |
| 13 | **Pressure on the 1-2 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-4 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-4 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-4 direction. |
| 33 | **Pressure on the 3-4 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-4 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 4-1 edge**; the force per unit area is defined in the global reference system. |
| 42 | **Pressure on the 4-1 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 4-1 direction. |
| 43 | **Pressure on the 4-1 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.7:continue in the next page*

*Table 3.7:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 44 | **Pressure on the 4-1 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.7:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.7.5 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.7.6 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.7.7 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.7.8 Field Output

1 = xx, 2 = yy, 3 = xy.

#### 3.5.7.9 Analysis types

- Infinitesimal elasto-plasticity.
- Finite elasto-plasticity.
- Masonry-like materials.

### 3.5.8 Element 7: Axisymmetric 4-node quadrilateral

This element is a 4-node isoparametric quadrilateral whose shape functions are bilinear.

#### 3.5.8.1 Nodal Connectivity

The element has 8 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.16.

#### 3.5.8.2 Integration Points

The element is integrated numerically using four points (Gaussian quadrature); the integration points are shown in Figure 3.16.

If selective reduced integration is required, the hydrostatic part of the strain is calculated using one integration point, namely the centroid of the element.

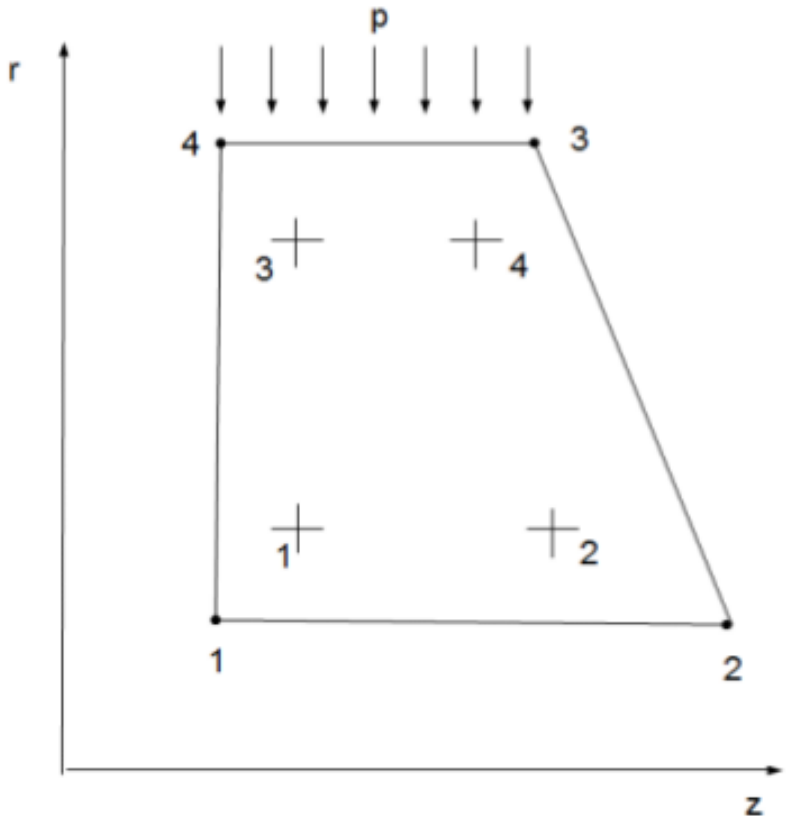



Figure 3.16: Nodal connectivity and integration points of the 4-node isoparametric quadrilateral element

#### 3.5.8.3 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.8.

Table 3.8: Distributed loads relevant to the 4-node axisymmetric element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |

*Table 3.8:continue in the next page*

*Table 3.8:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 3 | **Centrifugal force**; The rotation axis coincides with the axis of symmetry, the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |
| 13 | **Pressure on the 1-2 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-4 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-4 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-4 direction. |
| 33 | **Pressure on the 3-4 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-4 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 4-1 edge**; the force per unit area is defined in the global reference system. |
| 42 | **Pressure on the 4-1 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 4-1 direction. |
| 43 | **Pressure on the 4-1 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.8:continue in the next page*

*Table 3.8:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 44 | **Pressure on the 4-1 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.8:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.8.4 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.8.5 Nodal Coordinates

Two global coordinates in the z and r directions.

#### 3.5.8.6 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.8.7 Field Output

1 = zz, 2 = rr, 3 = $\theta\theta$, 4 = zr.

#### 3.5.8.8 Analysis types

- Infinitesimal elasto-plasticity.
- Finite elasto-plasticity.
- Masonry-like materials.

### 3.5.9 Element 8: 3D 8-node hexahedron

This element is a 8-node isoparametric brick whose shape functions are bilinear; it is suitable for three-dimensional analysis with finite strains.

#### 3.5.9.1 Nodal Connectivity

The element has 8 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.17.

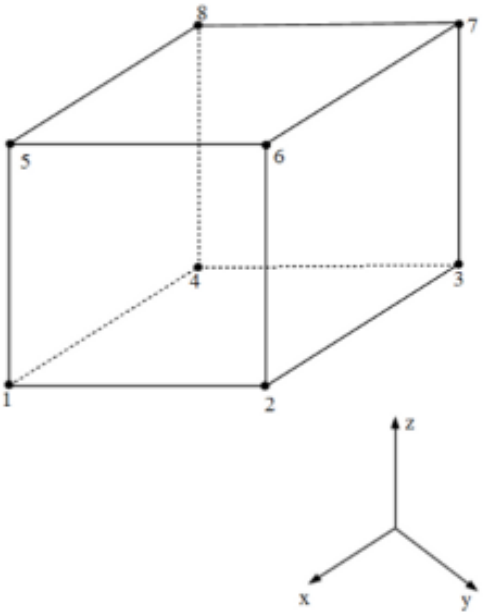



Figure 3.17: Nodal connectivity of the 8-node isoparametric hexahedric element

#### 3.5.9.2 Integration Points

The element is integrated numerically using eight points (Gaussian quadrature); the plane of the first series of 4 integration points is shown in Figure 3.18.

If selective reduced integration is required, the hydrostatic part of the strain is calculated using one integration point, namely the centroid of the element.

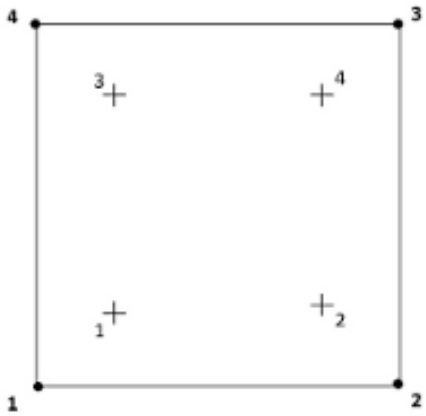



Figure 3.18: Gauss points of the 8-node isoparametric hexahedric element

#### 3.5.9.3 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.9.

Table 3.9: Distributed loads relevant to the 8-node hexahedric element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the three components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the three global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis must be defined, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2-3-4 face**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2-3-4 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 1-2 direction; the third component is orthogonal to the face and is positive if directed towards the interior of the element, the second component has the direction given by the vector product of the first and third directions. |
| 13 | **Pressure on the 1-2-3-4 face** in the global reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2-3-4 face** in the local reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 5-8-7-6 face**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 5-8-7-6 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 5-8 direction; the third component is orthogonal to the face and is positive if directed towards the interior of the element. The second component has the direction given by the vector product of the first and third directions. |
| 23 | **Pressure on the 5-8-7-6 face** in the global reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 5-8-7-6 face** in the local reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 1-5-6-2 face**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 1-5-6-2 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 1-5 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 33 | **Pressure on the 1-5-6-2 face** in the global reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 1-5-6-2 face** in the local reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 2-6-7-3 face**; the force per unit area is defined in the global reference system. |

*Table 3.9:continue in the next page*

*Table 3.9:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 42 | **Pressure on the 2-6-7-3 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 2-6 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 43 | **Pressure on the 2-6-7-3 face** in the global reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 44 | **Pressure on the 2-6-7-3 face** in the local reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 51 | **Pressure on the 3-7-8-4 face**; the force per unit area is defined in the global reference system. |
| 52 | **Pressure on the 3-7-8-4 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 3-7 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 53 | **Pressure on the 3-7-8-4 face** in the global reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 54 | **Pressure on the 3-7-8-4 face** in the local reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 61 | **Pressure on the 4-8-5-1 face**; the force per unit area is defined in the global reference system. |
| 62 | **Pressure on the 4-8-5-1 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 4-8 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 63 | **Pressure on the 4-8-5-1 face** in the global reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 64 | **Pressure on the 4-8-5-1 face** in the local reference system; the three components of the force per unit area on the four nodes of the face are to be specified in the user subroutine **FORCEM**. |

*Table 3.9:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.9.4 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a tetrahedron.

#### 3.5.9.5 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.9.6 Degrees of freedom

Three degrees of freedom u, v, w (the displacements along the global coordinate directions).

#### 3.5.9.7 Field Output

1 = xx, 2 = yy, 3 = zz, 4 = xy, 5 = yz, 6 = xz.

#### 3.5.9.8 Analysis types

- Infinitesimal elasto-plasticity.
- Finite elasto-plasticity.
- Masonry-like materials.

### 3.5.10 Element 9: Beam 2-node segment

This element is a 2-node isoparametric line segment, whose shape functions are linear for both displacements and rotations.

#### 3.5.10.1 Nodal Connectivity

The element has 2 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.19. Moreover, a local reference system ($e_1$, $e_2$, $e_3$) has to be specified, by using the AXIS option; $e_1$ and $e_2$ define the principal directions on the cross-section and $e_3$ is the axial direction, from node 1 to node 2.

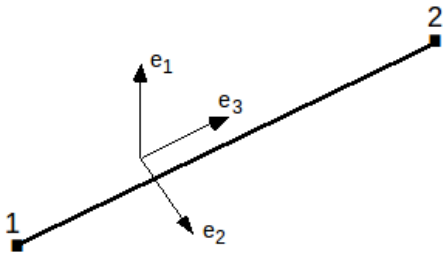



Figure 3.19: Nodal connectivity and local reference frame of the 2-node isoparametric element

#### 3.5.10.2 Integration Points

Numerical integration is performed both along the element axis and on the element cross-section; for the former integration one-point Gaussian quadrature (the centroid) is used. The latter integration is performed

using the Simpson method, with the number of integration points depending on the number of fibers of the beam (see BEAM SECT; the default value is 3x3, the maximum number of fibers is 21x21).

#### 3.5.10.3 Geometrical Attributes

For this element the user must specify the thickness of the beam cross-section, which, by default, is assumed to have a rectangular shape; for homogeneous beams it is enough to define the thickness of the whole cross-section along $e_1$ and $e_2$ directions. For non-homogeneous beams, the thickness of each fiber must be provided along the two principal directions of the cross-section (COMPOSITE option), as shown in Figure 3.20.

The thicknesses of the whole beam cross-section or each fiber can be given on a nodal basis by using the user routine **UGEOM**.

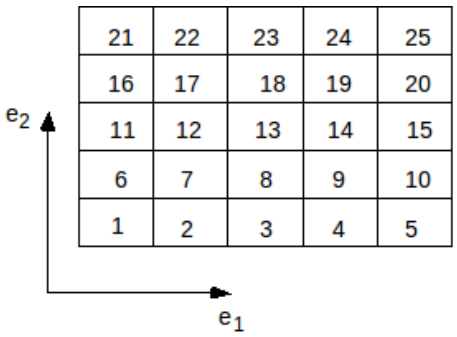



Figure 3.20: Numbering scheme for the fibers of the beam on a 5x5 example

Shapes of the beam cross-section different from the rectangular one can also be defined; in such cases, the shape is derived from the rectangular one. The user must enable the COMPOSITE option and, for each fiber of a rectangular cross-section, define mechanical properties of a given material; the final shape of the beam cross-section is obtained by giving null material properties to those fibers of the rectangular cross-section which don't make up the current cross-section.

#### 3.5.10.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.10.

Table 3.10: Distributed loads relevant to the 2-node beam element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force and/or moment**; the three components of the force and/or moment per unit volume in the global reference system must be assigned. |
| 2 | **Body force and/or moment**; the three global components of the force and/or moment per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis must be provided (see ROTATION AXIS option), and the angular velocity must be specified in the field reserved for the force magnitudes. |

*Table 3.10:continue in the next page*

*Table 3.10:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 11 | **Pressure on beam axis**; the force per unit length is defined in the global reference system. |
| 12 | **Pressure on beam axis** in the local reference system ($e_1$, $e_2$, $e_3$). |
| 13 | **Pressure on beam axis** in the global reference system; the components of the force per unit length at the two nodes of the beam are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on beam axis** in the local reference system; the components of the force per unit length at the two nodes of the beam are to be specified in the user subroutine **FORCEM**. |

*Table 3.10:completed*

Moreover, the element may be subjected to point loads and/or concentrated moments on nodes.

#### 3.5.10.5 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.10.6 Degrees of freedom

Six degrees of freedom u, v, w (the displacements along the global coordinate directions) and qx, qy, qz (the rotations, counterclockwise positive, around the global coordinate directions).

#### 3.5.10.7 Field Output

Strains and stress resultants are printed for the unique Gauss point; the components of the strain resultants are given in the local reference system ($e_1$, $e_2$, $e_3$) in the following order:

1. $\epsilon$ axial stretch,
2. $\kappa_1$ curvature change around $e_1$ axis,
3. $\kappa_2$ curvature change around $e_2$ axis,
4. $\psi$ twist around $e_3$ axis,
5. $\gamma_1$ shear deformation on the $e_1$-$e_3$ plane,
6. $\gamma_2$ shear deformation on the $e_2$-$e_3$ plane.

Similarly, for the stress resultants we have:

1. N axial force,
2. $M_1$ moment around $e_1$ axis,
3. $M_2$ moment around $e_2$ axis,
4. $M_3$ torque around $e_3$ axis,
5. $Q_1$ shear force on the $e_1$-$e_3$ plane,
6. $Q_2$ shear force on the $e_2$-$e_3$ plane.

#### 3.5.10.8 Analysis types

- Linear elasticity.

- Masonry-like materials.

### 3.5.11 Element 10: Thick shell 4-node quadrilateral

This element is a 4-node isoparametric quadrilateral, whose shape functions are linear for both displacements and rotations.

#### 3.5.11.1 Nodal Connectivity

The element has 4 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.21.

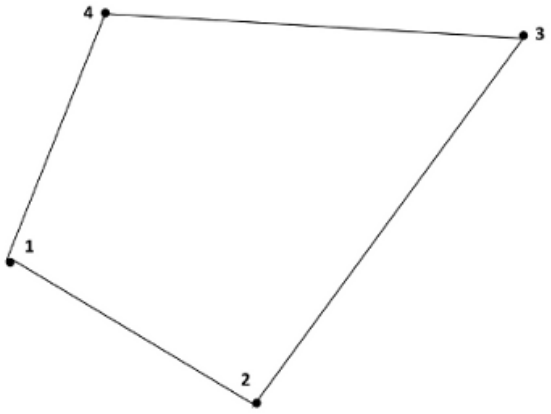



Figure 3.21: Nodal connectivity of the 4-node isoparametric quadrilateral element

#### 3.5.11.2 Integration Points

Numerical integration is performed both on the middle surface of the element and within the cross-section thickness; for the former integration four-point Gaussian quadrature is used, whose integration points are shown in Figure 3.22. The latter integration is performed using the Simpson method, with the number of integration points depending on the number of layers of the shell (see SHELL SECT; the default value is 3, the maximum number of layers is 99).

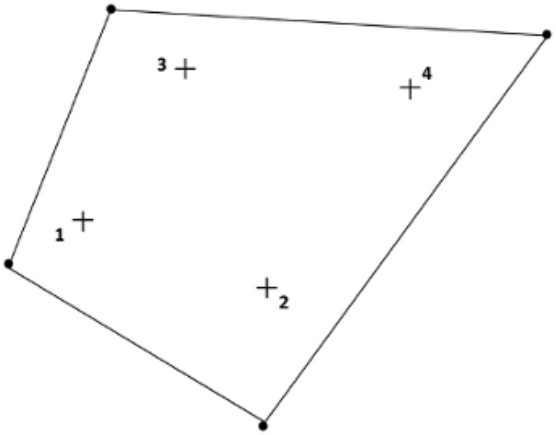



Figure 3.22: Gauss points of the 4-node isoparametric quadrilateral element

#### 3.5.11.3 Geometrical Attributes

For this element the user must specify the thickness(es) of the element cross-section; for homogeneous shells it is enough to define the thickness of the whole cross-section. For non-homogeneous shells, the thickness of each layer must be provided (COMPOSITE option).

The thickness of the whole shell or each layer can be specified at each corner node by using the user routine **UGEOM**.

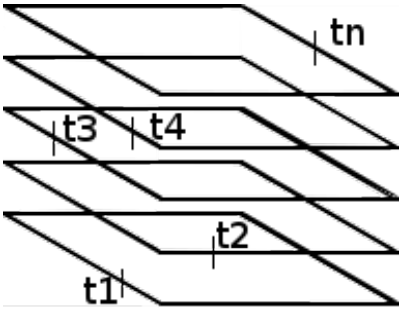



Figure 3.23: Layers thickness for composite shells

#### 3.5.11.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.11.

Table 3.11: Distributed loads relevant to the 4-node thick shell element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the three components of the force per unit volume in the global reference system must be assigned. |

*Table 3.11:continue in the next page*

*Table 3.11:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 2 | **Body force**; the three global components of the force per unit volume for each shell layer are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis must be defined (see ROTATION AXIS), and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit length is defined in the global reference system. |
| 13 | **Pressure on the 1-2 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit length is defined in the global reference system. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-4 edge**; the force per unit length is defined in the global reference system. |
| 33 | **Pressure on the 3-4 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 4-1 edge**; the force per unit length is defined in the global reference system. |
| 43 | **Pressure on the 4-1 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 51 | **Pressure on the surface** of the element; the force per unit area is defined in the global reference system. |
| 52 | **Pressure on the surface** of the element in the local reference system, as defined below. |
| 53 | **Pressure on the surface** of the element in the global reference system; the three components of the force per unit area on the four nodes of the element are to be specified in the user subroutine **FORCEM**. |
| 54 | **Pressure on the surface** of the element in the local reference system; the three components of the force per unit area on the four nodes of the element are to be specified in the user subroutine **FORCEM**. |

*Table 3.11:completed*

Moreover, the element may be subjected to point loads and/or concentrated moments on nodes, and thermal dilatation loads.

#### 3.5.11.5 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.11.6 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.11.7 Degrees of freedom

Six degrees of freedom u, v, w (the displacements along the global coordinate directions) and $q_x$, $q_y$, $q_z$ (the rotations, counterclockwise positive, around the global coordinate directions).

#### 3.5.11.8 Field Output

Strains are printed for the four Gauss points of the mean surface of the shell; the components of the strain are given in the local orthonormal reference system s, t, n in the following order :

1 = ss, 2 = tt, 3 = st, 4 = tn, 5 = sn.

The values of stress are printed for the four integration points for each layer of the element; the stress components are given in the local orthonormal reference system s, t, n in the following order :

1 = ss, 2 = tt, 3 = st, 4 = tn, 5 = sn.

#### 3.5.11.9 Analysis types

- Linear elasticity.
- Masonry-like materials.

#### 3.5.11.10 Local reference frame

The element is defined geometrically by the (x, y, z) coordinates of the four corner nodes.

Due to the bilinear interpolation, the surface forms a ruled surface which is allowed to degenerate to a plate.

The stress-strain output is given in local orthogonal surface directions ($v_1$, $v_2$, and $v_3$) which, for the centroid, are defined in the following way:

First, the vectors tangent to the curves with constant isoparametric coordinates $\xi_1$ and $\xi_2$ are normalized

$t_1 = \frac{\frac{\partial r}{\partial \xi_1}}{\|\frac{\partial r}{\partial \xi_1}\|}, t_2 = \frac{\frac{\partial r}{\partial \xi_2}}{\|\frac{\partial r}{\partial \xi_2}\|},$

Now a new basis is being defined as:

$s = t_1 + t_2, d = t_1 - t_2,$

After normalizing these vectors by:

$\hat{s} = \frac{s}{\sqrt{2\|s\|}}, \hat{d} = \frac{d}{\sqrt{2\|d\|}}.$

The local orthogonal directions are then obtained as:

$v_1 = \hat{s} + \hat{d}, v_2 = \hat{s} - \hat{d}, v_3 = v_1 \times v_2.$

In this way, the vectors $t_1$, $t_2$ and $v_1$, $v_2$ have the same bisecting plane.

The local directions for the Gaussian integrations points are found by projection of the centroid directions. Hence, if the element is flat, the directions at the Gauss points are identical to those at the centroid.

### 3.5.12 Element 11: Plane heat transfer 8-node quadrilateral

This element is a 8-node isoparametric quadrilateral whose shape functions are quadratic.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.12.1 Nodal Connectivity

The element has 8 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.24.

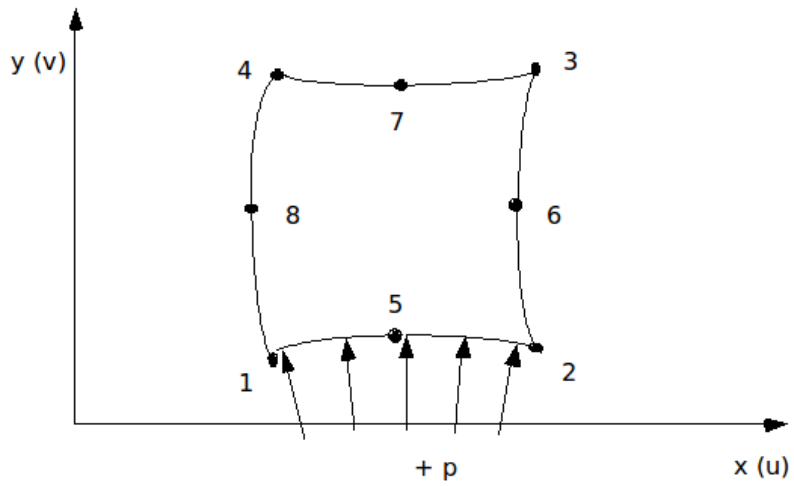



Figure 3.24: Nodal connectivity of the 8-node isoparametric quadrilateral element

#### 3.5.12.2 Integration Points

The element is integrated numerically using nine points (Gaussian quadrature); the integration points are shown in Figure 3.25.

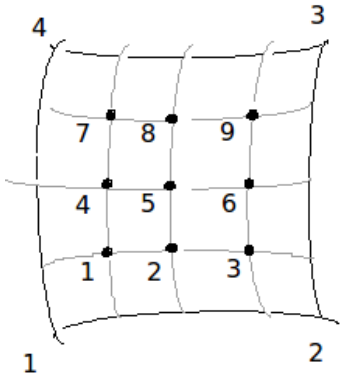



Figure 3.25: Gauss points of the 8-node isoparametric quadrilateral element

#### 3.5.12.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.12.

Table 3.12: Distributed heat fluxes relevant to the 8-node plane heat transfer element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Surface flux**; the heat quantity per unit area and unit time must be assigned. |
| 2 | **Surface flux**; the heat quantity per unit area and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Edge flux on the 1-2-5 edge** (heat per unit length and unit time). |
| 13 | **Edge flux on the 1-2-5 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 21 | **Edge flux on the 2-3-6 edge** (heat per unit length and unit time). |
| 23 | **Edge flux on the 2-3-6 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 31 | **Edge flux on the 3-4-7 edge** (heat per unit length and unit time). |
| 33 | **Edge flux on the 3-4-7 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 41 | **Edge flux on the 4-1-8 edge** (heat per unit length and unit time). |
| 43 | **Edge flux on the 4-1-8 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |

*Table 3.12:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.12.4 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.12.5 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.12.6 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.12.7 Field Output

Temperature gradient defined on Gauss points of the element.

1 = x, 2 = y.

#### 3.5.12.8 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.13 Element 12: Plane heat transfer 4-node quadrilateral

This element is a 4-node isoparametric quadrilateral whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.13.1 Nodal Connectivity

The element has 4 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.26.

#### 3.5.13.2 Integration Points

The element is integrated numerically using four points (Gaussian quadrature); the integration points are shown in Figure 3.26.

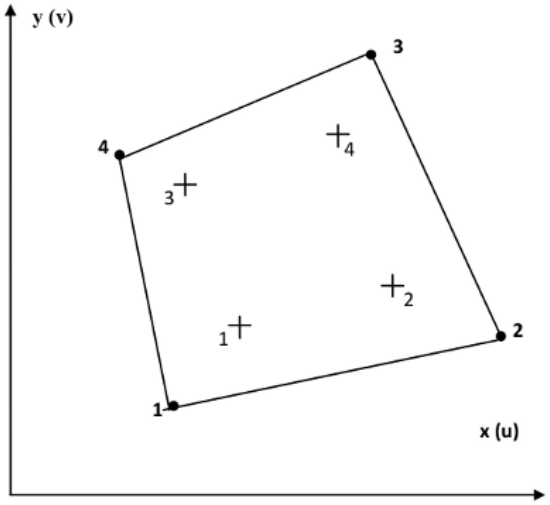



Figure 3.26: Nodal connectivity and integration points of the 4-node isoparametric quadrilateral element

#### 3.5.13.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.13.

Table 3.13: Distributed heat fluxes relevant to the 4-node plane heat transfer element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Surface flux**; the heat quantity per unit area and unit time must be assigned. |
| 2 | **Surface flux**; the heat quantity per unit area and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Edge flux on the 1-2 edge** (heat per unit length and unit time). |
| 13 | **Edge flux on the 1-2 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |

*Table 3.13:continue in the next page*

*Table 3.13:continue from the previous page*

| *Flux Identifier Number* | *Description* |
|---|---|
| 21 | **Edge flux on the 2-3 edge** (heat per unit length and unit time). |
| 23 | **Edge flux on the 2-3 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 31 | **Edge flux on the 3-4 edge** (heat per unit length and unit time). |
| 33 | **Edge flux on the 3-4 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 41 | **Edge flux on the 4-1 edge** (heat per unit length and unit time). |
| 43 | **Edge flux on the 4-1 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |

*Table 3.13:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.13.4 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.13.5 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.13.6 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.13.7 Field Output

Temperature gradient defined on Gauss points of the element.

1 = x, 2 = y.

#### 3.5.13.8 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.14 Element 13: Axisymmetric heat transfer 8-node quadrilateral

This element is a 8-node isoparametric quadrilateral whose shape functions are quadratic.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.14.1 Nodal Connectivity

The element has 8 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.27.

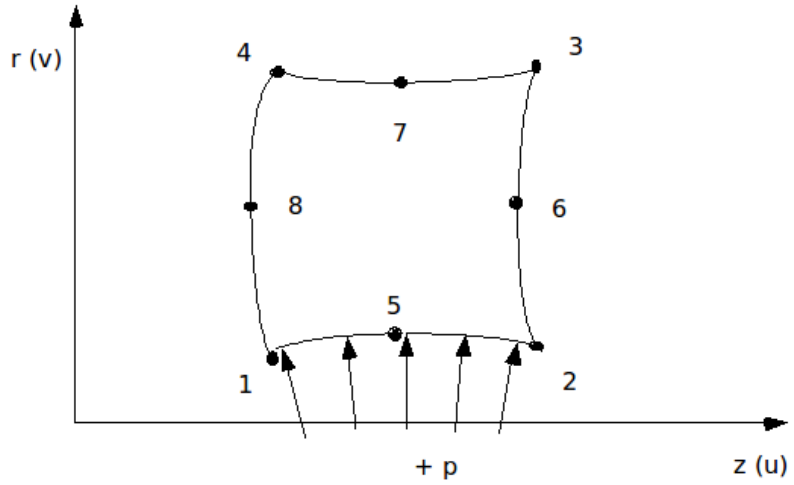



Figure 3.27: Nodal connectivity of the 8-node isoparametric quadrilateral element

#### 3.5.14.2 Integration Points

The element is integrated numerically using nine points (Gaussian quadrature); the integration points are shown in Figure 3.28.

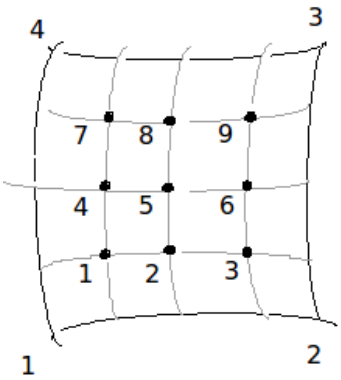



Figure 3.28: Gauss points of the 8-node isoparametric quadrilateral element

#### 3.5.14.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.14.

Table 3.14: Distributed heat fluxes relevant to the 8-node axisymmetric heat transfer element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Surface flux**; the heat quantity per unit area and unit time must be assigned. |
| 2 | **Surface flux**; the heat quantity per unit area and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Edge flux on the 1-2-5 edge** (heat per unit length and unit time). |
| 13 | **Edge flux on the 1-2-5 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 21 | **Edge flux on the 2-3-6 edge** (heat per unit length and unit time). |
| 23 | **Edge flux on the 2-3-6 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 31 | **Edge flux on the 3-4-7 edge** (heat per unit length and unit time). |
| 33 | **Edge flux on the 3-4-7 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 41 | **Edge flux on the 4-1-8 edge** (heat per unit length and unit time). |
| 43 | **Edge flux on the 4-1-8 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |

*Table 3.14:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.14.4 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.14.5 Nodal Coordinates

Two global coordinates in the z and r directions.

#### 3.5.14.6 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.14.7 Field Output

Temperature gradient defined on Gauss points of the element.

1 = z, 2 = r.

#### 3.5.14.8 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.15 Element 14: Axisymmetric heat transfer 4-node quadrilateral

This element is a 4-node isoparametric quadrilateral whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.15.1 Nodal Connectivity

The element has 4 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.29.

#### 3.5.15.2 Integration Points

The element is integrated numerically using four points (Gaussian quadrature); the integration points are shown in Figure 3.29.

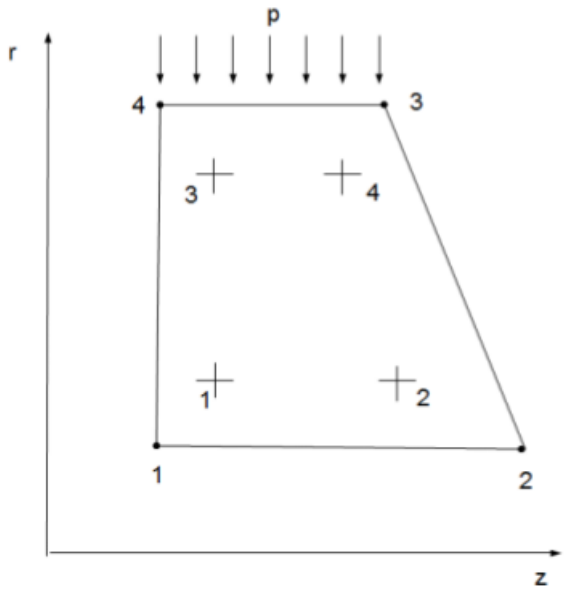



Figure 3.29: Nodal connectivity and integration points of the 4-node isoparametric quadrilateral element

#### 3.5.15.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.15.

Table 3.15: Distributed heat fluxes relevant to the 4-node axisymmetric heat transfer element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Surface flux**; the heat quantity per unit area and unit time must be assigned. |
| 2 | **Surface flux**; the heat quantity per unit area and unit time is calculated in the user subroutine **FORCEM**. |

*Table 3.15:continue in the next page*

*Table 3.15:continue from the previous page*

| *Flux Identifier Number* | *Description* |
|---|---|
| 11 | **Edge flux on the 1-2 edge** (heat per unit length and unit time). |
| 13 | **Edge flux on the 1-2 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 21 | **Edge flux on the 2-3 edge** (heat per unit length and unit time). |
| 23 | **Edge flux on the 2-3 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 31 | **Edge flux on the 3-4 edge** (heat per unit length and unit time). |
| 33 | **Edge flux on the 3-4 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 41 | **Edge flux on the 4-1 edge** (heat per unit length and unit time). |
| 43 | **Edge flux on the 4-1 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |

*Table 3.15:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.15.4 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.15.5 Nodal Coordinates

Two global coordinates in the z and r directions.

#### 3.5.15.6 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.15.7 Field Output

Temperature gradient defined on Gauss points of the element.

1 = z, 2 = r.

#### 3.5.15.8 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.16 Element 15: 3D heat transfer 8-node hexahedron

This element is a 8-node isoparametric brick whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.16.1 Nodal Connectivity

The element has 8 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.30.

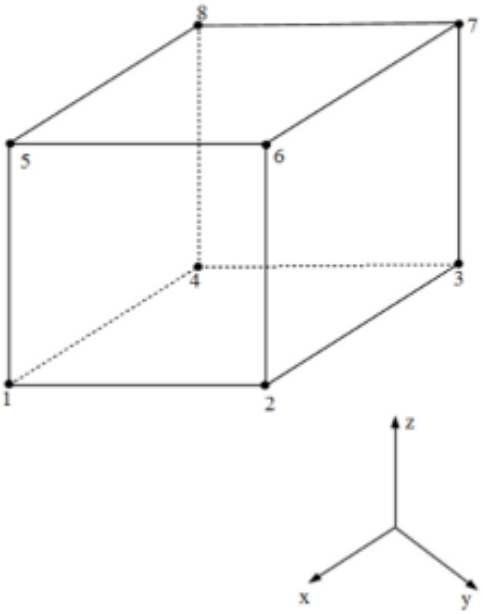



Figure 3.30: Nodal connectivity of the 8-node isoparametric hexahedric element

#### 3.5.16.2 Integration Points

The element is integrated numerically using eight points (Gaussian quadrature); the plane of the first series of 4 integration points is shown in Figure 3.31.

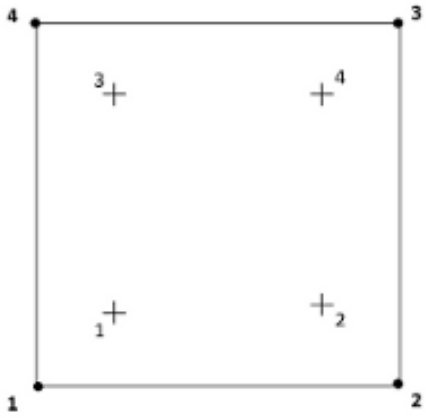



Figure 3.31: Gauss points of the 8-node isoparametric hexahedric element

#### 3.5.16.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.16.

Table 3.16: Distributed heat fluxes relevant to the 8-node hexahedric element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Volumetric flux**; the heat quantity per unit volume and unit time must be assigned. |
| 2 | **Volumetric flux**; the heat quantity per unit volume and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Surface flux on the 1-2-3-4 face** (heat per unit area and unit time). |
| 13 | **Surface flux on the 1-2-3-4 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 21 | **Surface flux on the 5-8-7-6 face** (heat per unit area and unit time). |
| 23 | **Surface flux on the 5-8-7-6 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 31 | **Surface flux on the 1-5-6-2 face** (heat per unit area and unit time). |
| 33 | **Surface flux on the 1-5-6-2 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 41 | **Surface flux on the 2-6-7-3 face** (heat per unit area and unit time). |
| 43 | **Surface flux on the 2-6-7-3 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 51 | **Surface flux on the 3-7-8-4 face** (heat per unit area and unit time). |
| 53 | **Surface flux on the 3-7-8-4 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 61 | **Surface flux on the 4-8-5-1 face** (heat per unit area and unit time). |
| 63 | **Surface flux on the 4-8-5-1 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |

*Table 3.16:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.16.4 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangular prism or to a tetrahedron.

#### 3.5.16.5 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.16.6 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.16.7 Field Output

Temperature gradient defined on Gauss points of the element.

1 = x, 2 = y, 3=z.

#### 3.5.16.8 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.17 Element 16: 3D heat transfer 20-node hexahedron

This element is a 20-node isoparametric brick whose shape functions are quadratic.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.17.1 Nodal Connectivity

The element has 20 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.32.

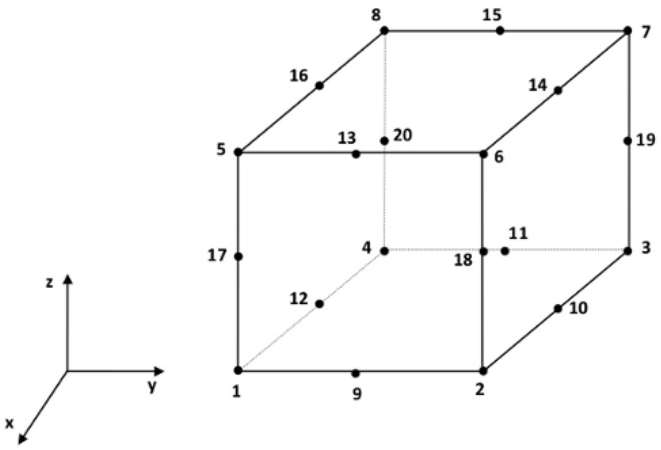



Figure 3.32: Nodal connectivity of the 20-node isoparametric hexahedric element

#### 3.5.17.2 Integration Points

The element is integrated numerically using twenty-seven points (Gaussian quadrature); the plane of the first series of 9 integration points is shown in Figure 3.33.

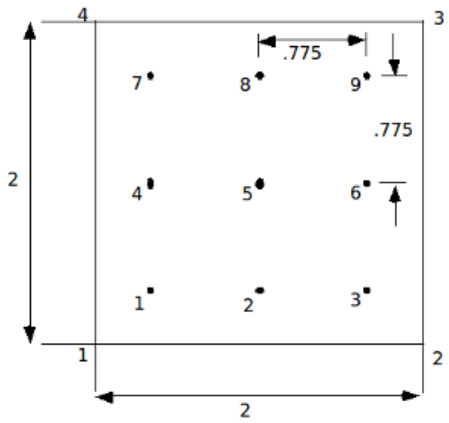



Figure 3.33: Gauss points of the 20-node isoparametric hexahedric element

#### 3.5.17.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.17.

Table 3.17: Distributed heat fluxes relevant to the 20-node hexahedric element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Volumetric flux**; the heat quantity per unit volume and unit time must be assigned. |
| 2 | **Volumetric flux**; the heat quantity per unit volume and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Surface flux on the 1-2-3-4 face** (heat per unit area and unit time). |
| 13 | **Surface flux on the 1-2-3-4 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 21 | **Surface flux on the 5-8-7-6 face** (heat per unit area and unit time). |
| 23 | **Surface flux on the 5-8-7-6 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 31 | **Surface flux on the 1-5-6-2 face** (heat per unit area and unit time). |
| 33 | **Surface flux on the 1-5-6-2 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 41 | **Surface flux on the 2-6-7-3 face** (heat per unit area and unit time). |
| 43 | **Surface flux on the 2-6-7-3 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 51 | **Surface flux on the 3-7-8-4 face** (heat per unit area and unit time). |
| 53 | **Surface flux on the 3-7-8-4 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 61 | **Surface flux on the 4-8-5-1 face** (heat per unit area and unit time). |
| 63 | **Surface flux on the 4-8-5-1 face**; the heat per unit area and unit time on the four nodes of the face is to be specified in the user subroutine **FORCEM**. |

*Table 3.17:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.17.4 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangular prism or a tetrahedron.

#### 3.5.17.5 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.17.6 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.17.7 Field Output

Temperature gradient defined on Gauss points of the element.

1 = x, 2 = y, 3=z.

#### 3.5.17.8 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.18 Element 17: Shell heat transfer 4-node quadrilateral

This element is a 4-node isoparametric quadrilateral, whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.18.1 Nodal Connectivity

The element has 4 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.34.

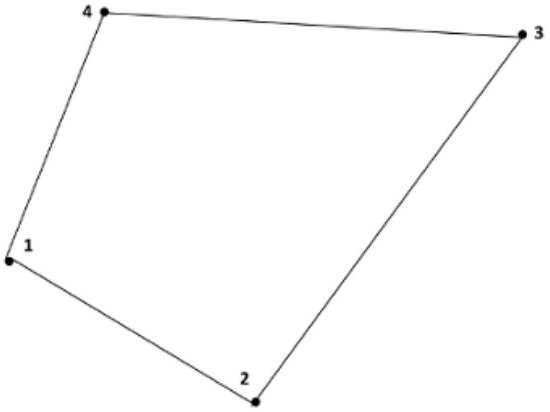



Figure 3.34: Nodal connectivity of the 4-node isoparametric quadrilateral element

#### 3.5.18.2 Integration Points

Numerical integration is performed both on the middle surface of the element and within the thickness; for the former integration four-point Gaussian quadrature is used, whose integration points are shown in Figure 3.35. The latter integration is performed using the Simpson method, with the number of integration points depending on the number of layers of the shell (see SHELL SECT; the default value is 3, the maximum number of layers is 99).

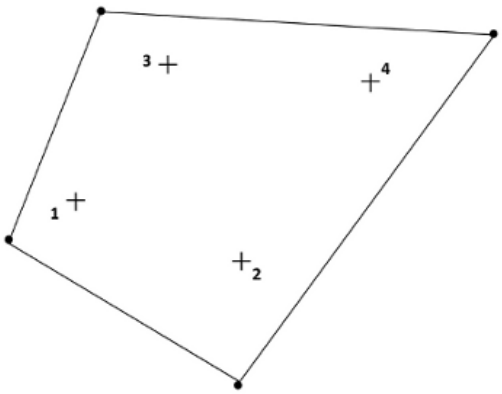



Figure 3.35: Gauss points of the 4-node isoparametric quadrilateral element

#### 3.5.18.3 Geometrical Attributes

For this element the user must specify the thickness(es) of the element; for homogeneous shells it is enough to define the thickness of the whole cross-section. For non-homogeneous shells, the thickness of each layer must be provided (COMPOSITE option).

The thickness of the whole shell or each layer can be specified at each corner node by using the user routine **UGEOM**.

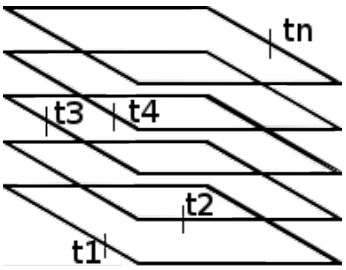



Figure 3.36: Layers thickness for composite shells

#### 3.5.18.4 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.18.

Table 3.18: Distributed heat fluxes relevant to the 4-node plane heat transfer element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Volumetric flux**; the heat quantity per unit volume and unit time must be assigned. |
| 2 | **Volumetric flux**; the heat quantity per unit volume and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Edge flux on the 1-2 edge** (heat per unit length and unit time). |
| 13 | **Edge flux on the 1-2 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 21 | **Edge flux on the 2-3 edge** (heat per unit length and unit time). |
| 23 | **Edge flux on the 2-3 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 31 | **Edge flux on the 3-4 edge** (heat per unit length and unit time). |
| 33 | **Edge flux on the 3-4 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 41 | **Edge flux on the 4-1 edge** (heat per unit length and unit time). |
| 43 | **Edge flux on the 4-1 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 51 | **Surface flux** (heat per unit area and unit time). |
| 53 | **Surface flux**; the heat per unit area and unit time on the four nodes is to be specified in the user subroutine **FORCEM**. |

*Table 3.18:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.18.5 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.18.6 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.18.7 Degrees of freedom

ITYRD degrees of freedom (temperatures) per node (the value of the ITYRD parameter is specified into the HEAT TRANSFER option)

ITYRD=2 (linear distribution trough the thickness)

1. Top surface temperature
2. Bottom surface temperature

ITYRD=3 (parabolic distribution trough the thickness)

1. Top surface temperature
2. Bottom surface temperature
3. Mid surface temperature.

#### 3.5.18.8 Field Output

The components of the temperature gradient are given in the local reference system of the shell. Namely, the first two components are along the local tangent directions, as defined for shell element type 10, whereas the third component is along the shell normal unit vector.

#### 3.5.18.9 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.19 Element 18: Plane stress 4-node quadrilateral

This element is a 4-node isoparametric quadrilateral, whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.19.1 Nodal Connectivity

The element has 4 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.37.

#### 3.5.19.2 Integration Points

The element is integrated numerically using four points (Gaussian quadrature); the integration points are shown in Figure 3.37.

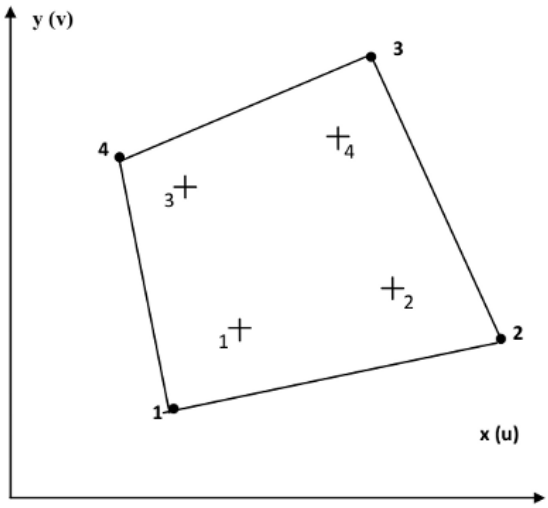



Figure 3.37: Nodal connectivity and integration points of the 4-node isoparametric quadrilateral element

#### 3.5.19.3 Geometrical Attributes

For this element the user must specify the thickness of the element; by default, a thickness equal to 1 is assigned.

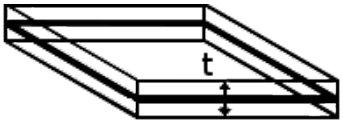



Figure 3.38: Element thickness

#### 3.5.19.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.19.

Table 3.19: Distributed loads relevant to the 4-node plane stress element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis is orthogonal to the plane of the element, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit area is defined in the global reference system. |

*Table 3.19:continue in the next page*

*Table 3.19:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 12 | **Pressure on the 1-2 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |
| 13 | **Pressure on the 1-2 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-4 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-4 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-4 direction. |
| 33 | **Pressure on the 3-4 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-4 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 4-1 edge**; the force per unit area is defined in the global reference system. |
| 42 | **Pressure on the 4-1 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 4-1 direction. |
| 43 | **Pressure on the 4-1 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 44 | **Pressure on the 4-1 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.19:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.19.5 Collapsed forms of the element

By suitably repeating node numbers in the connectivity, the element may be reduced as far as a triangle.

#### 3.5.19.6 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.19.7 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.19.8 Field Output

1 = xx, 2 = yy, 3 = xy.

#### 3.5.19.9 Analysis types

- Infinitesimal elasto-plasticity.
- Masonry-like materials.

### 3.5.20 Element 19: Plane stress 3-node triangle

This element is a 3-node isoparametric triangle, whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.20.1 Nodal Connectivity

The element has 3 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.39.

#### 3.5.20.2 Integration Points

The stiffness and the mass matrix of this element are evaluated using four points of integration (Gaussian quadrature); the integration points are shown in Figure 3.39 (cross points), where the first Gauss point is located at the centroid of the element.

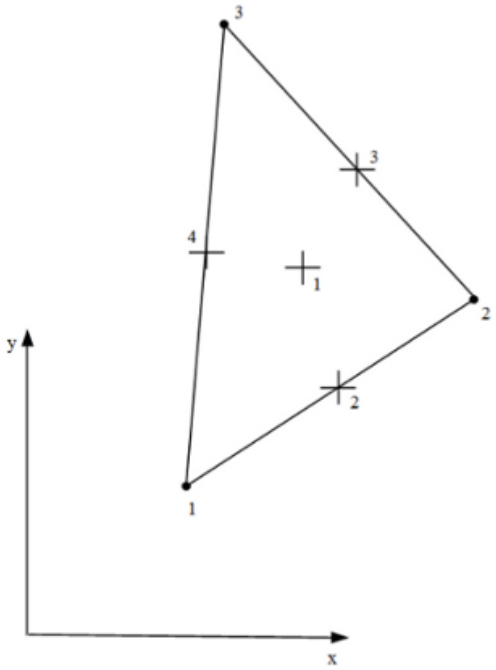



Figure 3.39: Nodal connectivity and integration points of the 3-node isoparametric triangular element

#### 3.5.20.3 Geometrical Attributes

For this element the user must specify the thickness of the element cross-section; by default, a thickness equal to 1 is assigned.

#### 3.5.20.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.20.

Table 3.20: Distributed loads relevant to the 3-node plane stress element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis is orthogonal to the plane of the element, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |

*Table 3.20:continue in the next page*

*Table 3.20:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 13 | **Pressure on the 1-2 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-1 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-1 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-1 direction. |
| 33 | **Pressure on the 3-1 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-1 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.20:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.20.5 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.20.6 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.20.7 Field Output

1 = xx, 2 = yy, 3 = xy.

#### 3.5.20.8 Analysis types

- Infinitesimal elasto-plasticity.
- Masonry-like materials.

### 3.5.21 Element 20: Plane stress 6-node triangle

This element is a 6-node isoparametric triangle, whose shape functions are quadratic.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.21.1 Nodal Connectivity

The element has 6 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.40.

#### 3.5.21.2 Integration Points

Numerical integration of this element is carried out by using three-points Gaussian quadrature; the integration points are shown in Figure 3.40 (cross points).

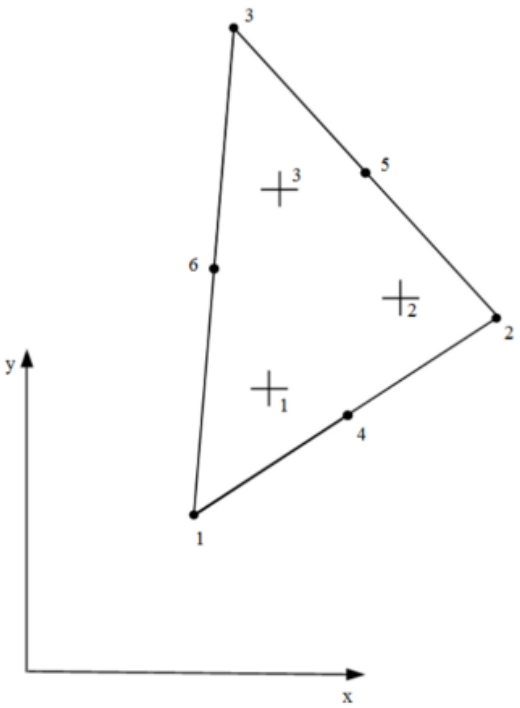



Figure 3.40: Nodal connectivity and integration points of the 6-node isoparametric triangular element

#### 3.5.21.3 Geometrical Attributes

For this element the user must specify the thickness of the element; by default, a thickness equal to 1 is assigned.

#### 3.5.21.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.21.

Table 3.21: Distributed loads relevant to the 6-node plane stress element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis is orthogonal to the plane of the element, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |
| 13 | **Pressure on the 1-2 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-1 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-1 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-1 direction. |
| 33 | **Pressure on the 3-1 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-1 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.21:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.21.5 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.21.6 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.21.7 Field Output

1 = xx, 2 = yy, 3 = xy.

#### 3.5.21.8 Analysis types

- Infinitesimal elasto-plasticity.
- Masonry-like materials.

### 3.5.22 Element 21: Plane strain 3-node triangle

This element is a 3-node isoparametric triangle, whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.22.1 Nodal Connectivity

The element has 3 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.41.

#### 3.5.22.2 Integration Points

The stiffness and the mass matrix of this element are evaluated using four points of integration (Gaussian quadrature); the integration points are shown in Figure 3.41 (cross points), where the first Gauss point is located at the centroid of the element.

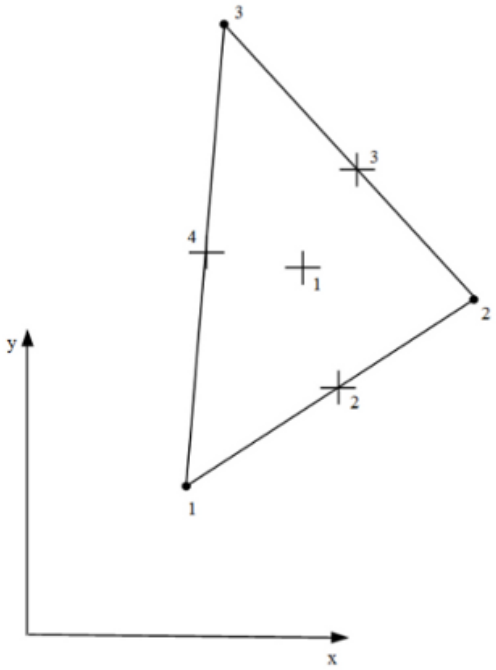



Figure 3.41: Nodal connectivity and integration points of the 3-node isoparametric triangular element

#### 3.5.22.3 Geometrical Attributes

For this element the user must specify the thickness of the element; by default, a thickness equal to 1 is assigned.

#### 3.5.22.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.22.

Table 3.22: Distributed loads relevant to the 3-node plane strain element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis is orthogonal to the plane of the element, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |

*Table 3.22:continue in the next page*

*Table 3.22:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 13 | **Pressure on the 1-2 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-1 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-1 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-1 direction. |
| 33 | **Pressure on the 3-1 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-1 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.22:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.22.5 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.22.6 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.22.7 Field Output

1 = xx, 2 = yy, 3 = xy.

#### 3.5.22.8 Analysis types

- Infinitesimal elasto-plasticity.
- Finite elasto-plasticity.
- Masonry-like materials.

### 3.5.23 Element 22: Plane strain 6-node triangle

This element is a 6-node isoparametric triangle, whose shape functions are quadratic.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.23.1 Nodal Connectivity

The element has 6 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.42.

#### 3.5.23.2 Integration Points

Numerical integration of this element is carried out by using three-points Gaussian quadrature; the integration points are shown in Figure 3.42 (cross points).

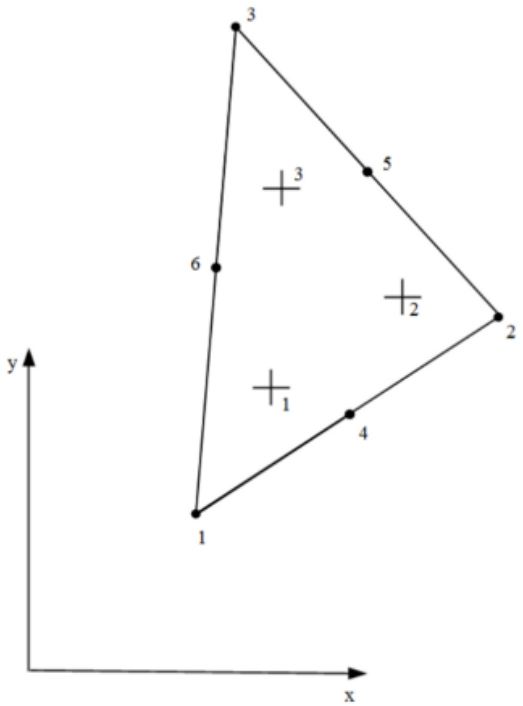



Figure 3.42: Nodal connectivity and integration points of the 6-node isoparametric triangular element

#### 3.5.23.3 Geometrical Attributes

For this element the user must specify the thickness of the element; by default, a thickness equal to 1 is assigned.

#### 3.5.23.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.23.

Table 3.23: Distributed loads relevant to the 6-node plane strain element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis is orthogonal to the plane of the element, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |
| 13 | **Pressure on the 1-2 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-1 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-1 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-1 direction. |

*Table 3.23:continue in the next page*

*Table 3.23:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 33 | **Pressure on the 3-1 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-1 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.23:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.23.5 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.23.6 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.23.7 Field Output

1 = xx, 2 = yy, 3 = xy.

#### 3.5.23.8 Analysis types

- Infinitesimal elasto-plasticity.
- Masonry-like materials.

### 3.5.24 Element 23: Axisymmetric 3-node triangle

This element is a 3-node isoparametric triangle, whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.24.1 Nodal Connectivity

The element has 3 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.43.

#### 3.5.24.2 Integration Points

Numerical integration is performed using four-points Gaussian quadrature; the integration points are shown in Figure 3.43 (cross positions).

If selective reduced integration is required, the hydrostatic part of the strain is calculated using one integration point, namely the centroid of the element.

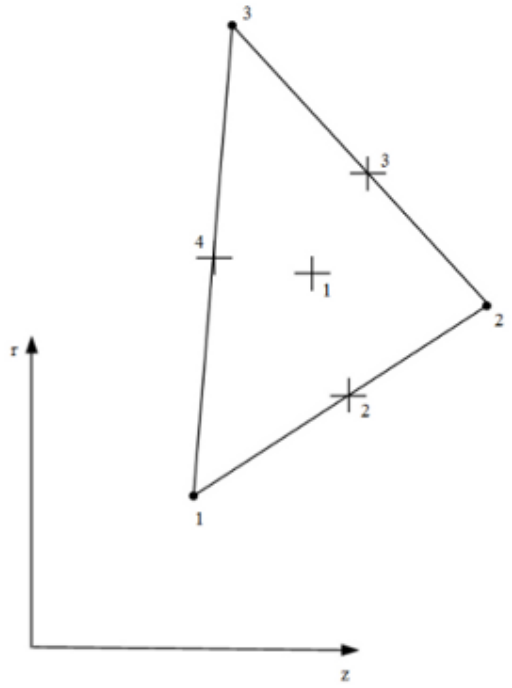



Figure 3.43: Nodal connectivity and integration points of the 3-node isoparametric triangular element

#### 3.5.24.3 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.24.

Table 3.24: Distributed loads relevant to the 3-node axisymmetric element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis is the axis of symmetry, the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |

*Table 3.24:continue in the next page*

*Table 3.24:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 13 | **Pressure on the 1-2 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-1 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-1 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-1 direction. |
| 33 | **Pressure on the 3-1 edge** in the global reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-1 edge** in the local reference system; the two components of the force per unit area on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.24:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.24.4 Nodal Coordinates

Two global coordinates in the z and r directions.

#### 3.5.24.5 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.24.6 Field Output

1 = zz, 2 = rr, 3 =$\theta\theta$, 4 = zr.

#### 3.5.24.7 Analysis types

- Infinitesimal elasto-plasticity.
- Masonry-like materials.

### 3.5.25 Element 24: Axisymmetric 6-node triangle

This element is a 6-node isoparametric triangle, whose shape functions are quadratic.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.25.1 Nodal Connectivity

The element has 6 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.44.

#### 3.5.25.2 Integration Points

Numerical integration is performed using three-points Gaussian quadrature; the integration points are shown in Figure 3.44 (cross positions).

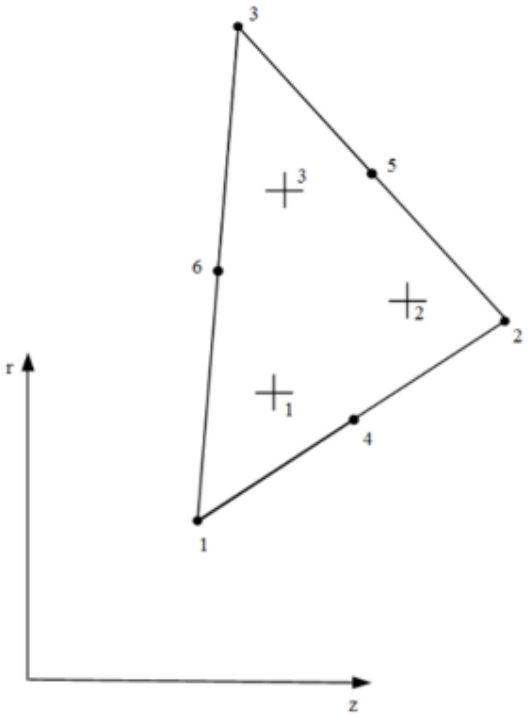



Figure 3.44: Nodal connectivity and integration points of the 6-node isoparametric triangular element

#### 3.5.25.3 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.25.

Table 3.25: Distributed loads relevant to the 6-node axisymmetric element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the two components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the two global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis is the axis of symmetry, the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 1-2 direction. |
| 13 | **Pressure on the 1-2 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 2-3 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 2-3 direction. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 2-3 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-1 edge**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 3-1 edge** in the local reference system; the first component of the force per unit area is orthogonal to the edge and is positive if directed towards the element's interior; the second component is tangential to the edge in the 3-1 direction. |
| 33 | **Pressure on the 3-1 edge** in the global reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 3-1 edge** in the local reference system; the two components of the force per unit area on the three nodes of the edge are to be specified in the user subroutine **FORCEM**. |

*Table 3.25:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.25.4 Nodal Coordinates

Two global coordinates in the z and r directions.

#### 3.5.25.5 Degrees of freedom

Two degrees of freedom u, v (the displacements along the global coordinate directions).

#### 3.5.25.6 Field Output

1 = zz, 2 = rr, 3 =$\theta\theta$, 4 = zr.

#### 3.5.25.7 Analysis types

- Infinitesimal elasto-plasticity.

- Masonry-like materials.

### 3.5.26 Element 25: 3D 4-node tetrahedron

This element is a 4-node isoparametric tetrahedron whose shape functions are linear; it is suitable for three-dimensional analysis with finite strains.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.26.1 Nodal Connectivity

The element has 4 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.45.

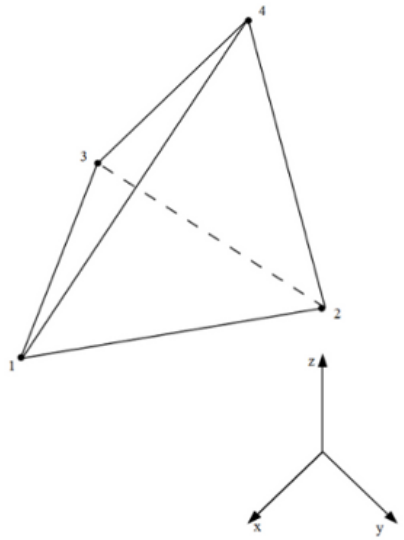



Figure 3.45: Nodal connectivity of the 4-node isoparametric tetrahedric element

#### 3.5.26.2 Integration Points

The element is integrated numerically using only one point (Gaussian quadrature), namely the centroid of the element.

#### 3.5.26.3 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.26.

Table 3.26: Distributed loads relevant to the 4-node tetrahedric element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the three components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the three global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis must be defined, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2-3 face**; the force per unit area is defined in the global reference system. |
| 12 | **Pressure on the 1-2-3 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 1-2 direction; the third component is orthogonal to the face and is positive if directed towards the interior of the element, the second component has the direction given by the vector product of the first and third directions. |
| 13 | **Pressure on the 1-2-3 face** in the global reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |

*Table 3.26:continue in the next page*

*Table 3.26:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 14 | **Pressure on the 1-2-3 face** in the local reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 1-2-4 face**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 1-2-4 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 1-2 direction; the third component is orthogonal to the face and is positive if directed towards the interior of the element. The second component has the direction given by the vector product of the first and third directions. |
| 23 | **Pressure on the 1-2-4 face** in the global reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 1-2-4 face** in the local reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 2-3-4 face**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 2-3-4 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 2-3 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 33 | **Pressure on the 2-3-4 face** in the global reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 2-3-4 face** in the local reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 3-1-4 face**; the force per unit area is defined in the global reference system. |
| 42 | **Pressure on the 3-1-4 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 3-1 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 43 | **Pressure on the 3-1-4 face** in the global reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 44 | **Pressure on the 3-1-4 face** in the local reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |

*Table 3.26:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.26.4 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.26.5 Degrees of freedom

Three degrees of freedom u, v, w (the displacements along the global coordinate directions).

#### 3.5.26.6 Field Output

1 = xx, 2 = yy, 3 = zz, 4 = xy, 5 = yz, 6 = xz.

#### 3.5.26.7 Analysis types

- Infinitesimal elasto-plasticity.
- Finite elasto-plasticity.
- Masonry-like materials.

### 3.5.27 Element 26: 3D 10-node tetrahedron

This element is a 10-node isoparametric tetrahedron whose shape functions are quadratic.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.27.1 Nodal Connectivity

The element has 10 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.46; in particular, nodes 1, 2, 3 are the corners of the first face (given in counterclockwise order when viewed from inside the element), and node 4 is on the opposing vertex. Nodes 5, 6, 7 are on the first face between nodes 1 and 2, 2 and 3, 3 and 1, respectively. Nodes 8, 9, 10 are along the edges between the first face and node 4, between nodes 1 and 4, 2 and 4, 3 and 4, respectively.

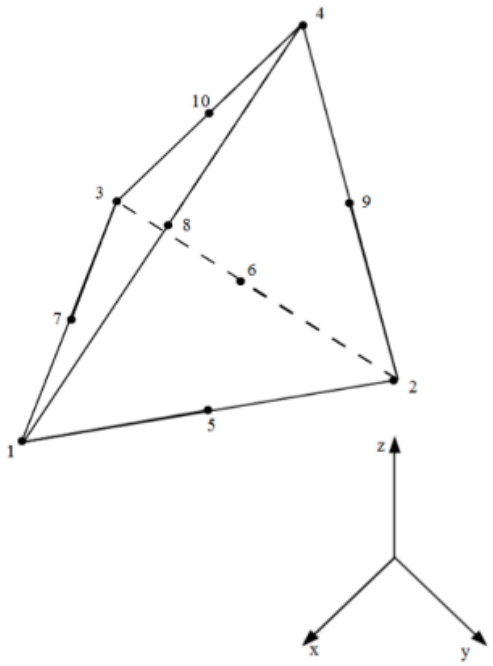



Figure 3.46: Nodal connectivity of the 10-node isoparametric tetrahedric element

#### 3.5.27.2 Integration Points

The element is integrated numerically using five points (Gaussian quadrature); the first integration point is located at the centroid of the element.

#### 3.5.27.3 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.27.

Table 3.27: Distributed loads relevant to the 10-node tetrahedric element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the three components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the three global components of the force per unit volume are calculated in the user subroutine **FORCEM**. |
| 3 | **Centrifugal force**; the rotation axis must be defined, and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2-3 face**; the force per unit area is defined in the global reference system. |

*Table 3.27:continue in the next page*

*Table 3.27:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 12 | **Pressure on the 1-2-3 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 1-2 direction; the third component is orthogonal to the face and is positive if directed towards the interior of the element, the second component has the direction given by the vector product of the first and third directions. |
| 13 | **Pressure on the 1-2-3 face** in the global reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 14 | **Pressure on the 1-2-3 face** in the local reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 1-2-4 face**; the force per unit area is defined in the global reference system. |
| 22 | **Pressure on the 1-2-4 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 1-2 direction; the third component is orthogonal to the face and is positive if directed towards the interior of the element. The second component has the direction given by the vector product of the first and third directions. |
| 23 | **Pressure on the 1-2-4 face** in the global reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 24 | **Pressure on the 1-2-4 face** in the local reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 2-3-4 face**; the force per unit area is defined in the global reference system. |
| 32 | **Pressure on the 2-3-4 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 2-3 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 33 | **Pressure on the 2-3-4 face** in the global reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 34 | **Pressure on the 2-3-4 face** in the local reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the 3-1-4 face**; the force per unit area is defined in the global reference system. |
| 42 | **Pressure on the 3-1-4 face** in the local reference system; the first component of the force per unit area is tangential to the face in the 3-1 direction, the third component is orthogonal to the face and is positive if directed towards the element's interior; the second component has the direction given by the vector product of the first and third directions. |
| 43 | **Pressure on the 3-1-4 face** in the global reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |

*Table 3.27:continue in the next page*

*Table 3.27:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 44 | **Pressure on the 3-1-4 face** in the local reference system; the three components of the force per unit area on the three nodes of the face are to be specified in the user subroutine **FORCEM**. |

*Table 3.27:completed*

Concentrate loads are point loads applied on nodes; moreover, the element may be subjected to thermal dilatation loads.

#### 3.5.27.4 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.27.5 Degrees of freedom

Three degrees of freedom u, v, w (the displacements along the global coordinate directions).

#### 3.5.27.6 Field Output

1 = xx, 2 = yy, 3 = zz, 4 = xy, 5 = yz, 6 = xz.

#### 3.5.27.7 Analysis types

- Infinitesimal elasto-plasticity.
- Finite elasto-plasticity.
- Masonry-like materials.

### 3.5.28 Element 27: Thick shell 3-node triangle

This element is a 3-node isoparametric triangle, whose shape functions are linear for both displacements and rotations.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.28.1 Nodal Connectivity

The element has 3 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.47.

#### 3.5.28.2 Integration Points

Numerical integration is performed both on the middle surface of the element and within the thickness; for the former integration three-point Gaussian quadrature is used, whose integration points are shown in Figure 3.47 (cross points) and located in the middle of the element edges. The latter integration is

performed using the Simpson method, with the number of integration points depending on the number of layers of the shell (see SHELL SECT; the default value is 3, the maximum number of layers is 99).

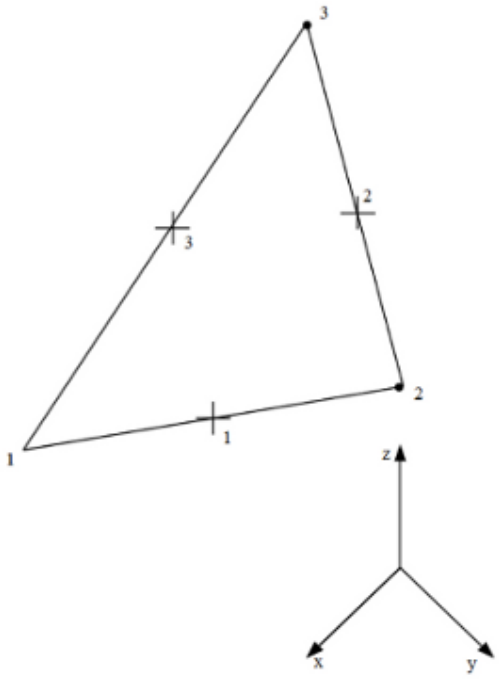



Figure 3.47: Nodal connectivity and Gauss points of the 3-node thick shell

#### 3.5.28.3 Geometrical Attributes

For this element the user must specify the thickness(es) of the element; for homogeneous shells it is enough to define the thickness of the whole cross-section. For non-homogeneous shells, the thickness of each layer must be provided (COMPOSITE option).

The thickness of the whole shell or each layer can be specified at each corner node by using the user routine **UGEOM**.

#### 3.5.28.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.28.

Table 3.28: Distributed loads relevant to the 3-node thick shell element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force**; the three components of the force per unit volume in the global reference system must be assigned. |
| 2 | **Body force**; the three global components of the force per unit volume for each shell layer are calculated in the user subroutine **FORCEM**. |

*Table 3.28:continue in the next page*

*Table 3.28:continue from the previous page*

| *Load Identifier Number* | *Description* |
|---|---|
| 3 | **Centrifugal force**; the rotation axis must be defined (see ROTATION AXIS), and the angular velocity (in revolutions per unit time) must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on the 1-2 edge**; the force per unit length is defined in the global reference system. |
| 13 | **Pressure on the 1-2 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 21 | **Pressure on the 2-3 edge**; the force per unit length is defined in the global reference system. |
| 23 | **Pressure on the 2-3 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 31 | **Pressure on the 3-1 edge**; the force per unit length is defined in the global reference system. |
| 33 | **Pressure on the 3-1 edge** in the global reference system; the three components of the force per unit length on the two nodes of the edge are to be specified in the user subroutine **FORCEM**. |
| 41 | **Pressure on the surface** of the element; the force per unit area is defined in the global reference system. |
| 42 | **Pressure on the surface** of the element in the local reference system, as defined below. |
| 43 | **Pressure on the surface** of the element in the global reference system; the three components of the force per unit area on the three nodes of the element are to be specified in the user subroutine **FORCEM**. |
| 44 | **Pressure on the surface** of the element in the local reference system; the three components of the force per unit area on the three nodes of the element are to be specified in the user subroutine **FORCEM**. |

*Table 3.28:completed*

Moreover, the element may be subjected to point loads and/or concentrated moments on nodes, and thermal dilatation loads.

#### 3.5.28.5 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.28.6 Degrees of freedom

Six degrees of freedom u, v, w (the displacements along the global coordinate directions) and $q_x$, $q_y$, $q_z$ (the rotations, counterclockwise positive, around the global coordinate directions).

#### 3.5.28.7 Field Output

Strains are printed for the three Gauss points of the mean surface of the shell; the components of the strain are given in the local orthonormal reference system s, t, n in the following order :

1 = ss, 2 = tt, 3 = st, 4 = tn, 5 = sn.

The values of stress are printed for the three integration points for each layer of the element; the stress components are given in the local orthonormal reference system s, t, n in the following order :

1 = ss, 2 = tt, 3 = st, 4 = tn, 5 = sn.

#### 3.5.28.8 Analysis types

- Linear elasticity.
- Masonry-like materials.

#### 3.5.28.9 Local reference frame

The element is defined geometrically by the (x, y, z) coordinates of the three corner nodes.

The stress-strain output is given in local orthogonal surface directions ($v_1$, $v_2$, and $v_3$) which are defined in the following way:

The first unit vector $v_1$ is going from node 1 to node 2

$v_1 = \frac{r_2 - r_1}{\|r_2 - r_1\|}$,

the normal unit vector is calculated as follows

$v_3 = \frac{v_1 \times (r_3 - r_1)}{\|v_1 \times (r_3 - r_1)\|}$,

and the unit vector $v_2$ is calculated in such a way to form a right handed reference frame

$v_2 = v_3 \times v_1$.

### 3.5.29 Element 28: Plane heat transfer 3-node triangle

This element is a 3-node isoparametric triangle, whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.29.1 Nodal Connectivity

The element has 3 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.48.

#### 3.5.29.2 Integration Points

Numerical integration is performed using four-points Gaussian quadrature; the integration points are shown in Figure 3.48 (cross points), where the first Gauss point is located at the centroid of the element.

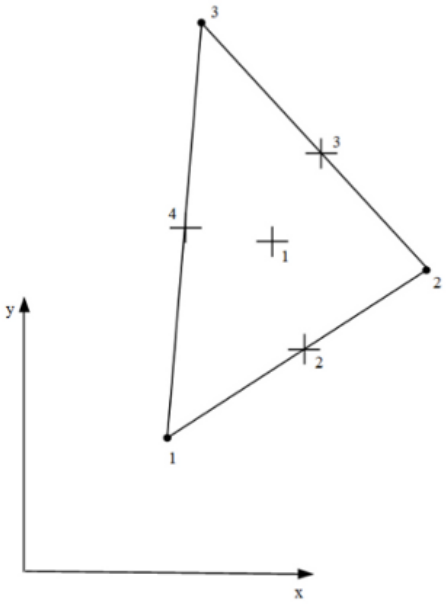



Figure 3.48: Nodal connectivity and integration points of the 3-node isoparametric triangular element

#### 3.5.29.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.29.

Table 3.29: Distributed heat fluxes relevant to the 3-node plane heat transfer element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Surface flux**; the heat quantity per unit area and unit time must be assigned. |
| 2 | **Surface flux**; the heat quantity per unit area and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Edge flux on the 1-2 edge** (heat per unit length and unit time). |
| 13 | **Edge flux on the 1-2 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 21 | **Edge flux on the 2-3 edge** (heat per unit length and unit time). |
| 23 | **Edge flux on the 2-3 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 31 | **Edge flux on the 3-1 edge** (heat per unit length and unit time). |
| 33 | **Edge flux on the 3-1 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |

*Table 3.29:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.29.4 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.29.5 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.29.6 Field Output

Temperature gradient defined on Gauss points of the element.
1 = x, 2 = y.

#### 3.5.29.7 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.30 Element 29: Plane heat transfer 6-node triangle

This element is a 6-node isoparametric triangle, whose shape functions are quadratic.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.30.1 Nodal Connectivity

The element has 6 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.49.

#### 3.5.30.2 Integration Points

Numerical integration of this element is carried out by using three-points Gaussian quadrature; the integration points are shown in Figure 3.49 (cross points).

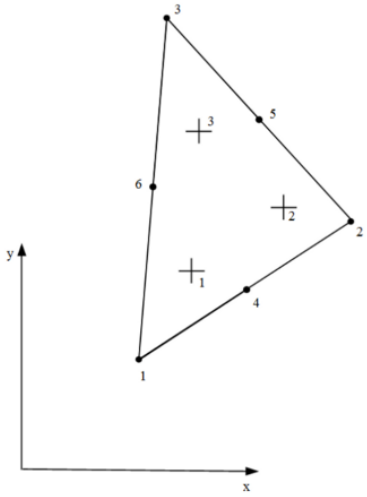



Figure 3.49: Nodal connectivity and integration points of the 6-node isoparametric triangular element

#### 3.5.30.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.30.

Table 3.30: Distributed heat fluxes relevant to the 6-node plane heat transfer element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Surface flux**; the heat quantity per unit area and unit time must be assigned. |
| 2 | **Surface flux**; the heat quantity per unit area and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Edge flux on the 1-2 edge** (heat per unit length and unit time). |
| 13 | **Edge flux on the 1-2 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 21 | **Edge flux on the 2-3 edge** (heat per unit length and unit time). |
| 23 | **Edge flux on the 2-3 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 31 | **Edge flux on the 3-1 edge** (heat per unit length and unit time). |
| 33 | **Edge flux on the 3-1 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |

*Table 3.30:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.30.4 Nodal Coordinates

Two global coordinates in the x and y directions.

#### 3.5.30.5 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.30.6 Field Output

Temperature gradient defined on Gauss points of the element.

1 = x, 2 = y.

#### 3.5.30.7 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.31 Element 30: Axisymmetric heat transfer 3-node triangle

This element is a 3-node isoparametric triangle, whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.31.1 Nodal Connectivity

The element has 3 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.50.

#### 3.5.31.2 Integration Points

Numerical integration is performed using four-points Gaussian quadrature; the integration points are shown in Figure 3.50 (cross positions).

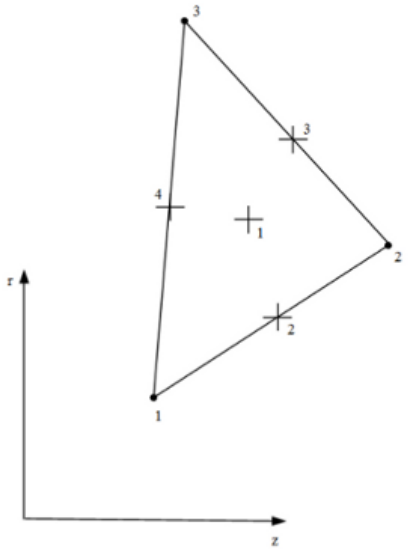



Figure 3.50: Nodal connectivity and integration points of the 3-node isoparametric triangular element

#### 3.5.31.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.31.

Table 3.31: Distributed heat fluxes relevant to the 3-node axisymmetric heat transfer element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Surface flux**; the heat quantity per unit area and unit time must be assigned. |
| 2 | **Surface flux**; the heat quantity per unit area and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Edge flux on the 1-2 edge** (heat per unit length and unit time). |
| 13 | **Edge flux on the 1-2 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 21 | **Edge flux on the 2-3 edge** (heat per unit length and unit time). |
| 23 | **Edge flux on the 2-3 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 31 | **Edge flux on the 3-1 edge** (heat per unit length and unit time). |
| 33 | **Edge flux on the 3-1 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |

*Table 3.31:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.31.4 Nodal Coordinates

Two global coordinates in the z and r directions.

#### 3.5.31.5 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.31.6 Field Output

Temperature gradient defined on Gauss points of the element. 1 = z, 2 = r.

#### 3.5.31.7 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.32 Element 31: Axisymmetric heat transfer 6-node triangle

This element is a 6-node isoparametric triangle, whose shape functions are quadratic.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.32.1 Nodal Connectivity

The element has 6 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.51.

#### 3.5.32.2 Integration Points

Numerical integration is performed using three-points Gaussian quadrature; the integration points are shown in Figure 3.51 (cross positions).

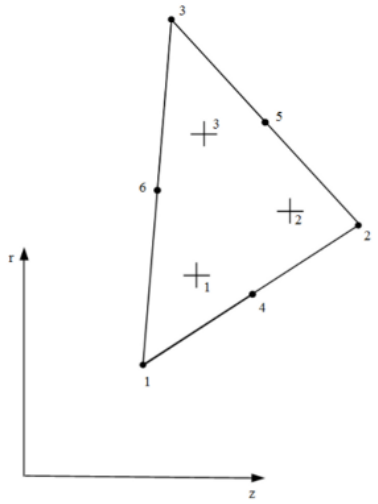



Figure 3.51: Nodal connectivity and integration points of the 6-node isoparametric triangular element

#### 3.5.32.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.32.

Table 3.32: Distributed heat fluxes relevant to the 6-node axisymmetric heat transfer element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Surface flux**; the heat quantity per unit area and unit time must be assigned. |
| 2 | **Surface flux**; the heat quantity per unit area and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Edge flux on the 1-2 edge** (heat per unit length and unit time). |
| 13 | **Edge flux on the 1-2 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 21 | **Edge flux on the 2-3 edge** (heat per unit length and unit time). |
| 23 | **Edge flux on the 2-3 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 31 | **Edge flux on the 3-1 edge** (heat per unit length and unit time). |
| 33 | **Edge flux on the 3-1 edge**; the heat per unit length and unit time on the three nodes of the edge is to be specified in the user subroutine **FORCEM**. |

*Table 3.32:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.32.4 Nodal Coordinates

Two global coordinates in the z and r directions.

#### 3.5.32.5 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.32.6 Field Output

Temperature gradient defined on Gauss points of the element.
1 = z, 2 = r.

#### 3.5.32.7 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.33 Element 32: 3D heat transfer 4-node tetrahedron

This element is a 4-node isoparametric tetrahedron whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.33.1 Nodal Connectivity

The element has 4 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.52.

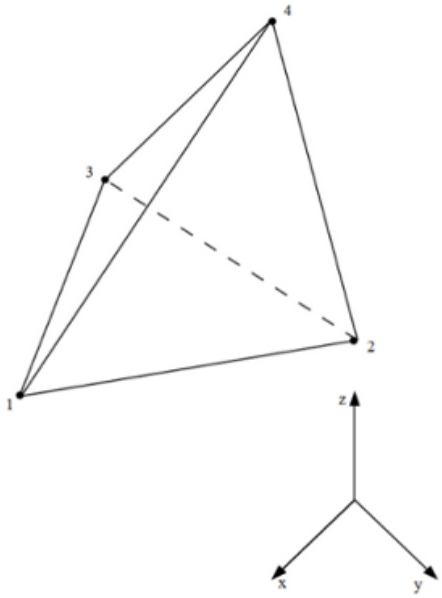



Figure 3.52: Nodal connectivity of the 4-node isoparametric tetrahedric element

#### 3.5.33.2 Integration Points

The element is integrated numerically using only one point (Gaussian quadrature), located at the centroid of the element.

#### 3.5.33.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.33.

Table 3.33: Distributed heat fluxes relevant to the 4-node tetrahedric element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Volumetric flux**; the heat quantity per unit volume and unit time must be assigned. |
| 2 | **Volumetric flux**; the heat quantity per unit volume and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Surface flux on the 1-2-3 face** (heat per unit area and unit time). |
| 13 | **Surface flux on the 1-2-3 face**; the heat per unit area and unit time on the three nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 21 | **Surface flux on the 1-2-4 face** (heat per unit area and unit time). |
| 23 | **Surface flux on the 1-2-4 face**; the heat per unit area and unit time on the three nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 31 | **Surface flux on the 2-3-4 face** (heat per unit area and unit time). |
| 33 | **Surface flux on the 2-3-4 face**; the heat per unit area and unit time on the three nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 41 | **Surface flux on the 3-1-4 face** (heat per unit area and unit time). |

*Table 3.33:continue in the next page*

*Table 3.33:continue from the previous page*

| *Flux Identifier Number* | *Description* |
|---|---|
| 43 | **Surface flux on the 3-1-4 face**; the heat per unit area and unit time on the three nodes of the face is to be specified in the user subroutine **FORCEM**. |

*Table 3.33:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.33.4 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.33.5 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.33.6 Field Output

Temperature gradient defined on Gauss points of the element.

1 = x, 2 = y, 3=z.

#### 3.5.33.7 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.34 Element 33: 3D heat transfer 10-node tetrahedron

This element is a 10-node isoparametric tetrahedron whose shape functions are quadratic.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.34.1 Nodal Connectivity

The element has 10 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.53; in particular, nodes 1, 2, 3 are the corners of the first face (given in counterclockwise order when viewed from inside the element), and node 4 is on the opposing vertex. Nodes 5, 6, 7 are on the first face between nodes 1 and 2, 2 and 3, 3 and 1, respectively. Nodes 8, 9, 10 are along the edges between the first face and node 4, between nodes 1 and 4, 2 and 4, 3 and 4, respectively.

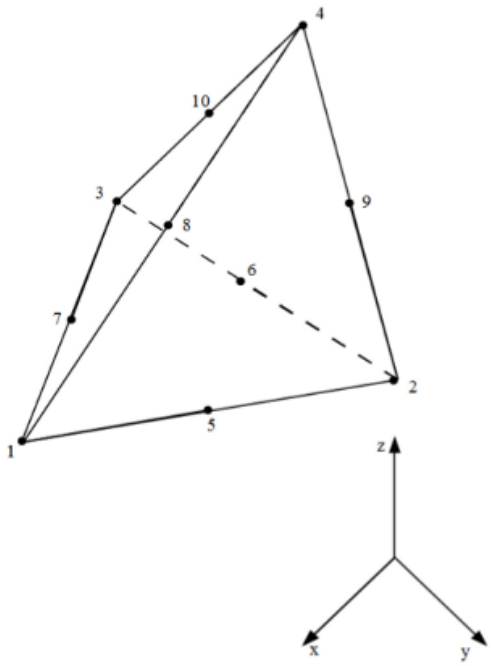



Figure 3.53: Nodal connectivity of the 10-node isoparametric tetrahedric element

#### 3.5.34.2 Integration Points

The element is integrated numerically using five points (Gaussian quadrature); the first integration point is located at the centroid of the element.

#### 3.5.34.3 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.34.

Table 3.34: Distributed heat fluxes relevant to the 10-node tetrahedric element

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Volumetric flux**; the heat quantity per unit volume and unit time must be assigned. |
| 2 | **Volumetric flux**; the heat quantity per unit volume and unit time is calculated in the user subroutine **FORCEM**. |
| 11 | **Surface flux on the 1-2-3 face** (heat per unit area and unit time). |
| 13 | **Surface flux on the 1-2-3 face**; the heat per unit area and unit time on the six nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 21 | **Surface flux on the 1-2-4 face** (heat per unit area and unit time). |
| 23 | **Surface flux on the 1-2-4 face**; the heat per unit area and unit time on the six nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 31 | **Surface flux on the 2-3-4 face** (heat per unit area and unit time). |

*Table 3.34:continue in the next page*

*Table 3.34:continue from the previous page*

| *Flux Identifier Number* | *Description* |
|---|---|
| 33 | **Surface flux on the 2-3-4 face**; the heat per unit area and unit time on the six nodes of the face is to be specified in the user subroutine **FORCEM**. |
| 41 | **Surface flux on the 3-1-4 face** (heat per unit area and unit time). |
| 43 | **Surface flux on the 3-1-4 face**; the heat per unit area and unit time on the six nodes of the face is to be specified in the user subroutine **FORCEM**. |

*Table 3.34:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.34.4 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.34.5 Degrees of freedom

One degree of freedom T (the nodal temperature).

#### 3.5.34.6 Field Output

Temperature gradient defined on Gauss points of the element.

1 = x, 2 = y, 3=z.

#### 3.5.34.7 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.35 Element 34: Shell heat transfer 3-node triangle

This element is a 3-node isoparametric triangle, whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.35.1 Nodal Connectivity

The element has 3 nodes, and its nodal connectivity is counterclockwise as shown in Figure 3.54.

#### 3.5.35.2 Integration Points

Numerical integration is performed both on the middle surface of the element and within the cross-section thickness; for the former integration three-point Gaussian quadrature is used, whose integration points are shown in Figure 3.54 (cross points) and located in the middle of the element edges. The latter integration

is performed using the Simpson method, with the number of integration points depending on the number of layers of the shell (see SHELL SECT; the default value is 3, the maximum number of layers is 99).

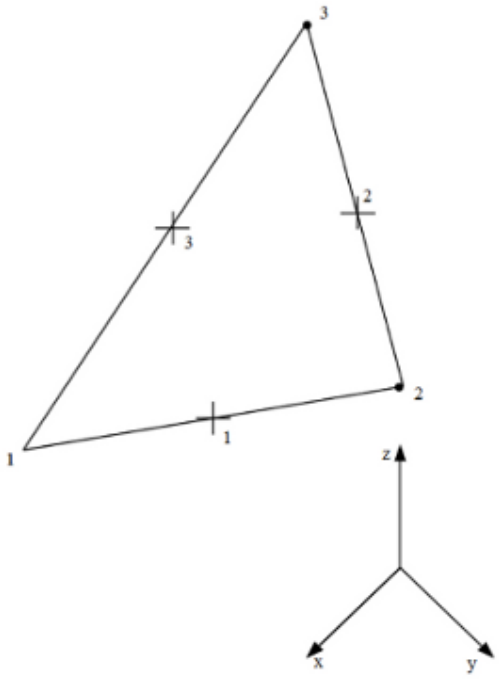



Figure 3.54: Nodal connectivity and Gauss points of the 3-node thick shell

#### 3.5.35.3 Geometrical Attributes

For this element the user must specify the thickness(es) of the element cross-section; for homogeneous shells it is enough to define the thickness of the whole cross-section. For non-homogeneous shells, the thickness of each layer must be provided (COMPOSITE option).

The thickness of the whole shell or each layer can be specified at each corner node by using the user routine **UGEOM**.

#### 3.5.35.4 Distributed fluxes

Heat fluxes applied on this element can be both concentrated and distributed fluxes; the latter ones are shown in Table 3.35.

Table 3.35: Distributed heat fluxes relevant to the 3-node heat transfer thick shell

| *Flux Identifier Number* | *Description* |
|---|---|
| 1 | **Volumetric flux**; the heat quantity per unit volume and unit time must be assigned. |
| 2 | **Volumetric flux**; the heat quantity per unit volume and unit time is calculated in the user subroutine **FORCEM**. |

*Table 3.35:continue in the next page*

*Table 3.35:continue from the previous page*

| *Flux Identifier Number* | *Description* |
|---|---|
| 11 | **Edge flux on the 1-2 edge** (heat per unit length and unit time). |
| 13 | **Edge flux on the 1-2 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 21 | **Edge flux on the 2-3 edge** (heat per unit length and unit time). |
| 23 | **Edge flux on the 2-3 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 31 | **Edge flux on the 3-1 edge** (heat per unit length and unit time). |
| 33 | **Edge flux on the 3-1 edge**; the heat per unit length and unit time on the two nodes of the edge is to be specified in the user subroutine **FORCEM**. |
| 41 | **Surface flux** (heat per unit area and unit time). |
| 43 | **Surface flux**; the heat per unit area and unit time on the three nodes is to be specified in the user subroutine **FORCEM**. |

*Table 3.35:completed*

Moreover, the element may be subjected to point fluxes on nodes and convective fluxes by specifying the film coefficients and sink temperatures for each edge.

#### 3.5.35.5 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.35.6 Degrees of freedom

ITYRD degrees of freedom (temperatures) per node (the value of the ITYRD parameter is specified into the HEAT TRANSFER option)

ITYRD=2 (linear distribution trough the thickness)

- 1st dof: Top surface temperature
- 2nd dof: Bottom surface temperature

ITYRD=3 (parabolic distribution trough the thickness)

- 1st dof: Top surface temperature
- 2nd dof: Bottom surface temperature
- 3rd dof: Mid surface temperature.

#### 3.5.35.7 Field Output

The components of the temperature gradient are given in the local reference system of the shell. Namely, the first two components are along the local tangent directions, as defined for shell element type 10, whereas the third component is along the shell normal unit vector.

#### 3.5.35.8 Analysis types

- Linear and non-linear transient heat transfer analysis.

### 3.5.36 Element 35: Three-dimensional 2-node truss

This is a 2-node isoparametric truss element, whose shape functions are linear.

**This element is not available in the present version of NOSA-ITACA.**

#### 3.5.36.1 Nodal Connectivity

The element has 2 nodes, and the truss axis is in the direction from the first node to the second one.

#### 3.5.36.2 Integration Points

Numerical integration is performed along the element axis using one-point Gaussian quadrature, with the integration point located at the centroid.

#### 3.5.36.3 Geometrical Attributes

The truss cross-section area is to be specified with the GEOMETRY option; it is also possible to define the area on a nodal basis by using the user routine UGEOM.

#### 3.5.36.4 Distributed loads

Loads applied on this element can be both concentrated and distributed loads; the latter ones are shown in Table 3.36.

Table 3.36: Distributed loads relevant to the 2-node truss element

| *Load Identifier Number* | *Description* |
|---|---|
| 1 | **Body force and/or moment**; the three components of the force and/or moment per unit volume in the global reference system must be assigned. |
| 3 | **Centrifugal force**; the rotation axis must be provided (see ROTATION AXIS option), and the angular velocity must be specified in the field reserved for the force magnitudes. |
| 11 | **Pressure on truss axis**; the force per unit length is defined in the global reference system. |

*Table 3.36:completed*

Moreover, the element may be subjected to point loads and/or concentrated moments on nodes.

Thermal loads are not yet available.

#### 3.5.36.5 Nodal Coordinates

Three global coordinates in the x, y and z directions.

#### 3.5.36.6 Degrees of freedom

Three degrees of freedom u, v, w (the displacements along the global coordinate directions).

#### 3.5.36.7 Field Output

Strains and stress resultants are printed for the unique Gauss point; the strain characteristic is given as:

1. $\epsilon$ axial stretch.

Similarly, the stress characteristic is:

1. N axial force.

#### 3.5.36.8 Analysis types

- Linear elasticity.

# Chapter 4

# NOSA Keywords Reference Guide

## 4.1 Introduction

This guide shows all keywords processed by **NOSA** solver to perform numerical analyses, by means of which it is possible to define a mesh model (e.g., nodal coordinates, elements connectivity, groups of elements and/or nodes), and assign physical properties to it (e.g., materials, element section properties, boundary conditions, load histories, etc...). Furthermore, it is possible to define the kind of the analysis to be performed, and set up solution controls (e.g, choose a particular numerical algorithm to solve equilibrium equations, define convergence criteria for solution, etc...).

The keywords make up the NOSA input file (i.e., a file with extension ".crd", and hereinafter referred as **crd file**), which will be described in the next section.

## 4.2 Description of the crd file

The structure of the NOSA input file can be logically grouped into three sections:

- Control Section : This is the first part of a crd file, necessary to control the program flow, allocate the memory required for carrying out the analysis, and define general features.
- Model Section : This part follows the control section, and describes the mesh model, assigns physical properties, and provides information concerning the algorithm used and the accuracy of analysis.
- Load Section : This is the last part of a crd file, necessary to define the load increments, and allows re-definition of the boundary conditions.

Each part is characterized by a subset of keywords; thus in an input file, the keywords of each section follow one another in the order as above described. Within each section, the keywords can be placed in any order.

The keywords have a free format and fields are separated by blanks and/or commas. Integer or real fields in a card can be constituted by a number with a maximum length of ten characters (including sign, point and exponent). For some keywords (e.g, element connectivity, element and/or node sets definition, etc...) it is required one or more lines of definition; in such cases, each line, except the last one, ends with the character '=', which represents the symbol of continuation of the previous line. The maximum number of continuations is 999.

## 4.3 Control Section

Below a list of keywords ordered as occurring into the input file.

- BEAM SECT
- COMMENT
- COMPOSITE
- CONTACT
- DAMPING
- DEFORMABLE
- DIST LOADS

- DYNAMIC
- ELASTIC
- ELEMENTS
- END
- ENERGY
- FILM
- FINITE STRAIN
- FOLLOWER FORCES
- FRICTION
- HARDENING
- HEAT TRANSFER
- HHT
- ITRESS
- LUMPED MASS
- MASONRY
- MATERIALS
- MODAL
- POTENTIAL
- PRINT LEVEL
- RESTART
- SCALE
- SETNAME
- SHELL SECT
- SIZING
- STOP
- TEMP TABLES
- THERMAL LOADS
- TIE
- TITLE

### 4.3.1 BEAM SECT

This command defines the number of integration points on the cross section of the beam elements.

**Command syntax** :

*beam sect n1 n2*

where

- n1 is the number of integration points along the first local direction on the cross section of the beam. This number must be an odd number between 3 and 21 (including the extreme values);
- n2 is the number of integration points along the second local direction on the cross section of the beam. This number must be an odd number between 3 and 21 (including the extrem values).

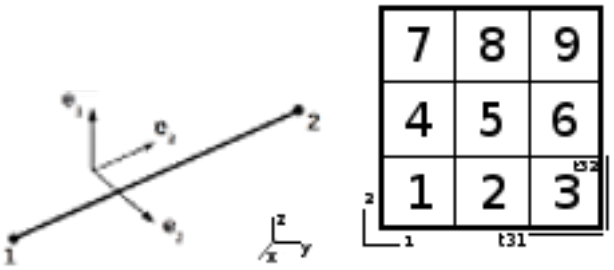



Figure 4.1: Example of a 3 x 3 beam cross section; two integration points along both e1 and e2 directions are used.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

***beam sect 6 5***

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

In the above example, six integration points along the first local direction and five integration points along the second local direction on the cross section of the beam will be used to apply the *Cavalieri-Simpson* quadrature technique rule.

Default value of both n1 and n2 is 3, for which the keyword is optional.

### 4.3.2 COMMENT

This command is used to insert a comment, anywhere.

**Command syntax** :

*comment "the \em text \em of \em the \em comment"*

or:

*$ "the \em text \em of \em the \em comment"*

**Input file usage**

*title Example 1*

***comment This is a first comment***

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***$ This is another comment***

*shell section 11*

*tie 914 1*

*end*

.....

### 4.3.3 COMPOSITE

This command indicates that shell elements with sections made of different materials and/or with different thickness will be used.

**Command syntax** :

*composite*

**Input file usage** :

*title Example 1*

*sizing 43228 45379*

***composite***

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

If composite option is set each layer of a shell element can have a different thickness and a different material.

### 4.3.4 CONTACT

The keyword is used in contact analyses involving rigid surfaces. Currently, contact analyses apply only to plane strain and axisymmetric problems.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*contact msurf mpart*

where

- msurf is the maximum number of rigid surfaces involved into contact problems;
- mpart is the maximum number of basic shapes (e.g., segments, circumference, arcs), of which each rigid surface is made up.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

***contact 2 5***

*friction 1*

*beam sect 6 5*

*dist load 19*

*shell* section 11

*tie 914 1*

*end*

.....

In this example there are two rigid surfaces made up of five basic shapes.

### 4.3.5 DAMPING

The keyword has to be set when viscous and/or numerical damping occurs in dynamic analysis.

**Command syntax** :

*damping*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

***damping***

*dynamic*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

See also DYNAMIC

### 4.3.6 DEFORMABLE

The keyword is used when deformable bodies are involved in contact problems. Currently, only one deformable body can be considered.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*deformable mdefo*

where

- mdefo is the maximum number of deformable bodies (i.e., 1).

**Input file usage**.

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias v*

***deformable 1***

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

### 4.3.7 DIST LOADS

The keyword is used to define the maximum number of distributed loads required by the analysis to be carried out.

**Command syntax** :

*dist loads mdist*

where

- mdist is the maximum number of distributed loads.

Default value for mdist is 3.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

***dist loads 19***

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

In this example a maximum of 19 distributed loads are applied.

### 4.3.8 DYNAMIC

This command indicates that dynamic analysis using the Newmark or Hylber-Hughes-Taylor methods of integration will be performed.

**Command syntax** :

*dynamic*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 9 9 45379* 1

*alias 75 10 98 9*

***dynamic***

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

See also HHT

### 4.3.9 ELASTIC

This keyword is used when a linear elastic analysis with multiple loads must be performed.

**Command syntax** :

*elastic*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

***elastic***

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

**Remarks:**

This keyword has to be considered as a kind of analyses (in the same manner as static, dynamic and so on) and not a material behaviour. When the keyword is set, all defined loads will be applied immediately; if not used each load will be applied according to a given loading sequence (i.e., as loads are defined into the input file), and each load after the first one will be considered as a load increment to be added to the previous loads.

### 4.3.10 ELEMENTS

The keyword defines the element types according to the **NOSA Element Library**, which will be used in the current analysis.

**Command syntax** :

*elements ntype1 ntype2 ... ntypeN*

where

- ntype1, ntype2...,ntypeN represent the element types and N is the number of different element types used in the current analysis. The maximum number of element types is 30.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

***elements 10 9***

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

In this example two types of element will be used: thick shells (i.e., element type 10) and beams (i.e., element type 9).

### 4.3.11 END

This command indicates the end of the control cards.

**Command syntax** :

*end*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

***end***

.....

**Remarks:**

This keyword is mandatory.

### 4.3.12 ENERGY

When this keyword is set the calculation of the mechanical energy (kinetic and potential), as well as the work done by the applied external forces, is required for both dynamic and static analyses.

**Command syntax** :

*energy*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***energy***

*shell section 11*

*tie 914 1*

*end*

*.....*

### 4.3.13 FILM

This command indicates that film coefficients of convective heating/cooling will be read in.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*film*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***film***

*shell section 11*

*tie 914 1*

*end*

*.....*

### 4.3.14 FINITE STRAIN

This command indicates that finite deformations will be considered in the analysis.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*finite strain*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***finite strain***

*shell section 11*

*tie 914 1*

*end*

.....

### 4.3.15 FOLLOWER FORCES

When the keyword is set forces for the current analysis will be applied to the current configuration. This keyword is relevant to only element types 6, 7 and 8 (i.e., plane strain, axisymmetric and brick linear elements).

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*follower forces*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***follower forces***

*shell section 11*

*tie 914 1*

*end*

.....

See also DIST LOADS

### 4.3.16 FRICTION

The keyword is used to define whether or not friction forces will be taken into account, in contact problems.

**This keyword is not available in the presence version of the NOSA-ITACA.**

**Command syntax** :

*friction ifric*

where

- ifric is equal to 0 for frictionless contact and equal to 1 for contact with friction.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***friction 1***

*shell section 11*

*tie 914 1*

*end*

.....

In the above example a problem contact with friction has been considered.

### 4.3.17 HARDENING

The keyword is used when elastic-plastic materials with work-hardening behaviour will be considered for the current analysis.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*hardening mhard*

where

- mhard is the maximun number of data pairs defining every hardening curve (in a tabular form); the default value is 0.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***hardening 20***

*shell section 11*

*tie 914 1*

*end*

.....

In this example every hardening curve is defined by means of 20 data pairs.

### 4.3.18 HEAT TRANSFER

This keyword is used to carry out a transient heat transfer analysis.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*heat transfer ityrd*

where

- ityrd defines the number of nodal temperatures and the order of interpolation trough the thickness for the heat transfer shell element type 17. Ityrd can assume the following values:
  - 1, valid for all element types except type 17. This is the default value;
  - 2, for linear interpolation through the thickness; each node will have two degrees of freedom, i.e. the temperature at the top and bottom surface of the shell.
  - 3, for parabolic interpolation through the thickness; each node will have three degrees of freedom, i.e. the temperature at the top, bottom and mid surface of the shell.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***heat transfer 2***

*shell section 11*

*tie 914 1*

*end*

.....

In this example we consider a transient heat transfer analysis with linear interpolation.

### 4.3.19 HHT

This command indicates that the Hiber-Hughes-Taylor method will be used to integrate the equation of motion in a dynamic analysis.

**Command syntax** :

*HHT*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***HHT***

*shell section 11*

*tie 914 1*

*end*

.....

See also DYNAMIC

### 4.3.20 ITRESS

This command indicates that the initial stress field will be read from a suitable file (FORTRAN unit 22).

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*itress*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***itress***

*shell section 11*

*tie 914 1*

*end*

.....

### 4.3.21 LUMPED MASS

This command indicates that a lumped mass matrix will be used in carrying out dynamic analysis.

**Command syntax** :

*lumped*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***lumped***

*shell section 11*

*tie 914 1*

*end*

.....

See also DYNAMIC

### 4.3.22 MASONRY

The keyword is used to indicate that masonry-like materials are used in the analysis.

**Command syntax** :

*masonry*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

***masonry***

*shell section 11*

*tie 914 1*

*end*

.....

### 4.3.23 MATERIALS

This keyword sets the maximum number of different materials used in the analysis.

**Command syntax** :

*materials nmats*

where

- nmats is the maximum number of different materials.

By default *nmats* is equal to 1.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

***materials 4***

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

In this example four different materials are considered.

### 4.3.24 MODAL

This keyword is used to perform a modal analysis.

**Command syntax** :

*modal nfreq*

where

- nfreq is the number of frequencies required.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*material 4*

*setname 19 9 45379*

*dist load 19*

*shell section 11*

*dynamic*

***modal 15***

*end*

.....

*end option*

.....

*stepsize*

*0 1 0.1 0.1 0*

In this example fifteen frequencies are required.

**Remarks:**

In order to perform a modal analysis the keywords DYNAMIC and STEPSIZE are necessary.

### 4.3.25 POTENTIAL

The keyword sets the maximum number of different elastic - plastic materials, whose anisotropic yield is described by means of the Hill's potential theory.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*potential nmaph*

where

- nmaph is the maximum number of different materials with plastic anisotropy.

By default, *nmaph* is equal to *nmats*. Anyway, *nmaph* is not greater than *nmats*.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*hardening*

*materials 4*

***potential 2***

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

*title Example 2*

*sizing 43228 45379*

*element 10 9*

*hardening*

*materials 4*

***potential***

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

In Example 1 two materials with plastic anisotropy have been considered, while in Example 2 all materials (*nmaph* = *nmats*) have anisotropic plasticity.

**See also:**

HARDENING and MATERIALS.

### 4.3.26 PRINT LEVEL

The keyword is used to set the print output level.

**Command syntax** :

*print lvprt ifreqp nelpr nodpr*

where

- *lvprt* is a value greater or equal to 1; when *lvprt* is equal to 1 displacement, reaction, stress and strain fields at the end of each load increment will be written to the **prt file** (i.e., NOSA output file with extension .prt). For *lvprt* $>$ 1 displacement, reaction, stress and strain fields and incremental nodal forces for each element, at the end of each load increment will be written to the prt file. By default, *lvprt* is equal to 1.

- *ifreqp* sets the frequency for which data will be written to prt file; *ifreqp* equal to *n* (with n>=1) means that data will be written every n load increments. By default *ifreqp* is equal to 1.

- *nelpr* sets the group of mesh elements which data into prt file refer to; when *nelpr* is equal to 0 data for all elements will be written. For *nelpr* = 1 data of a selected subset of elements will be written to the prt file.

- *nodpr* sets the group of mesh nodes which data into prt file refer to; when *nodpr* is equal to 0 data of all nodes will be written. For *nodpr* = 1 data of a selected subset of nodes will be written to the prt file.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias* 75 10 98 9

*beam sect 6 5*

*dist load 19*

*masonry*

***print 3 2 1 0***

*shell section 11*

*tie 914 1*

*end*

.....

In this example displacement, reaction, stress and strain fields and incremental nodal forces will be written every two load increments for each element of a selected subset of elements and for all nodes.

### 4.3.27 RESTART

The keyword is used when the analysis to be submitted is a continuation of a previous one. All information about the problem must be read from a suitable fortran file (unit n.99).

**Command syntax** :

*restart*

**Input file usage**

*title Example 1*

***restart***

*end*

.....

**Remarks:**

For restarting an analysis as a continuation of a previous one, details relevant to only incremental loads and the frequency to which data will be written to the **post file** (i.e., formattated output file $*$.t19 or med file) must be supplied. All other necessary information will be read from the **restart file** (i.e., file with extension .rst).

**See also:**

SAVE INCREMENT

### 4.3.28 SCALE

This command is used for scaling the results of the problem at the first incremental load in such a way that the points nearest the first yield are brought to that limit.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*scale*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

***scale***

*element 10 9*

*materials 4*

*setname 19 9 45379*

*malias 75 10 98 9*

*dist load 19*

*end*

.....

**Remarks:**

This option works only in the case of infinitesimal elastic-plastic behaviour.

### 4.3.29 SETNAME

The keyword is used to define how many sets of elements and nodes should be created, as well as how many items should be contained in a set.

**Command syntax** :

*setname neset nnset nitems ncomit*

where

- *neset* is the maximum number of element sets;
- *nnset* is the maximum number of node sets;
- *nitems* is the maximum number of items in each set;
- *ncomit* is the maximum number of items in each compound set.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

***setname 19 9 45379***

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

In the above example nineteen element sets and nine node sets will be defined. Each set can include a maximum number of 45379 items. Because no compound set will be created, *ncomit* can be omitted.

### 4.3.30 SHELL SECT

This keyword defines the number of integration points within the thickness for shell elements.

**Command syntax** :

*shell sect mshel*

where

- *mshel* is the number of integration points; by default *mshel* is equal to 3. Maximum value of *mshel* is equal to 99.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

***shell section 11***

*tie 914 1*

*end*

.....

In the example eleven points will be used for numerical integration within the thickness of shell elements.

**Remarks:**

The number of integration points must be odd. The *Cavalieri-Simpson* integration points are located on the mean surface of each section for composite shells, whereas for homogeneus shells they are located on the boundary of layers.

### 4.3.31 SIZING

This keyword defines the analysis dimension.

**Command syntax** :

*sizing nelem npoint eltype*

where

- *nelem* is the total number of elements ;
- *npoint* is the total numer of points ;

- *eltype* is the element type according to **NOSA Element Library**; this item can be omitted.

**Input file usage**

*title Example 1*

***sizing 43228 45379***

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

**Remarks:**

This keyword is mandatory.

### 4.3.32 STOP

The keyword is used for estimating the amount of memory required to carried out the analysis. No numerical calculation will be performed.

**Command syntax** :

*stop*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

***stop***

*end*

.....

### 4.3.33 TEMP TABLES

This keyword is used when mechanical properties of materials (e.g., Young modulus, Poisson ratio and coefficient of thermal expansion) depend on the temperature field. In such a case the dependence on temperature must be supplied in a tabular form.

**Command syntax** :

*temp tables mttemp mttab*

where

- *mttemp* is the maximum number of data tables;
- *mttab* is the maximum number of data pairs in each table.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

***temp tables 2 10***

*tie 914 1*

*end*

.....

In this example materials have two mechanical properties depending on temperature field; for each mechanical property, the dependence on the temperature field is described by a table containing ten data pairs.

### 4.3.34 THERMAL LOADS

Thermal loads due to temperature changes will be taken into account in the analysis; the keyword defines the way of reading the temperature field.

**Command syntax** :

*thermal loads ityrd*

where *ityrd* is defined as follows:

- *ityrd* = 1 means that nodal temperature will be read; this case applies to all element types, but shells.
- *ityrd* = 2 means that temperature field along the thickness will be calculated by linear interpolation of the temperature values of the lower and upper layers; this case applies to only shell elements.
- *ityrd* = 3 means that temperature field along the thickness will be calculated by parabolic interpolation of the temperature values of the lower, middle and upper layers; this case applies to only shell elements.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

***thermal loads 2***

*tie 914 1*

*end*

.....

In this example temperature field through the thickness of shell elements will be calculated by linear interpolation of the temperature values of the lower and upper layers.

### 4.3.35 TIE

The keyword is used when multipoint constraints must be defined.

**Command syntax** :

*tie Nbslave Nbmaster*

where

- *Nbslave* is the maximum number of nodes with dependent degree of freedom (slave nodes);
- *Nbmaster* is the maximum number of nodes with independent DOFs (master nodes) for each slave node.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

***tie 914 1***

*end*

.....

In Example 1 there are 914 nodes whose degree of freedom are described as dependent linear combination of the dofs of two master nodes.

### 4.3.36 TITLE

The keyword is used to assign a title to the current analysis.

**Command syntax** :

*title Title* of analysis

**Input file usage**

***title Example 1***

*sizing 43228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

.....

**Remarks:**

This keyword is mandatory. The title can be a string of maximum 70 characters (including blanks), and must be left-aligned.

## 4.4 Model Section

### 4.4.1 Introduction

For some keywords it can be necessary to define lists of mesh nodes and/or elements; a list can be specified as follows:

- a sequence of integers (e.g., 1 10 33.... and so on).
- in the way as *n TO m BY k*; if k is not specified, it is assumed equal to 1.
- a sequence of names of sets connected by logic operators (e.g., set1 and set2 and set3)

The logical operators are:

- AND : all items belonging to the set defined on the right side of this operator will be included into the current list of nodes and/or elements;
- EXCEPT : items belonging to the set declared on the right side of this operator will not be included into the current list;
- INTERSECT : all items belonging to the intersection of the sets will be considered.

A series of logical operations are always interpreted from left to right; thus, each operator always acts upon the list resulting from operations performed up to that point (i.e., the result of all operations on its left) and the set declared on its immediate right.

Below a list of keywords ordered as occurring into the input file.

- BODY
- BOUNDARY CONDITIONS
- COMMENT
- COMPOSITION
- CONNECTIVITY
- CONTROL
- COORDINATES
- DAMPING
- DEFINE
- DYNAMIC
- END OPTION
- FILM COEFFICIENT
- FIXED ACCELERATION
- FIXED DISPLACEMENT
- FIXED TEMPERATURE

- FRICTION
- GEOMETRY
- HARDENING
- INITIAL DISPLACEMENT
- INITIAL TEMPERATURE
- INITIAL VELOCITY
- LOCAL AXIS
- LUMPED MASS
- MASONRY
- MASSES
- POST
- POTENTIAL HILL
- PROPERTY
- ROTATION AXIS
- SELECTIVE INTEGRATION
- SURFACE
- TEMP TABLE
- TYING

### 4.4.2 BODY

This command defines deformable bodies present in the structure.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*body*

$N$

$L_1$

$L_2$

...

$L_N$

where

- $N$ is the number of deformable bodies;
- $L_n$ is the list of elements belonging to the n-th deformable body.

**Input file usage**

*title Example 1*

.....

*deformable 1*

.....

*end*

***body***

***1***

***35 36 37 38 39 40 41 42 43 =***

***44 45 46***

.....

*end option*

.....

In this example we define one body with 12 elements. You can also write the elements list in the following ways :

.....

***body***

***1***

***35 to 46***

..... if the elements numbering is sequential, or

.....

***body***

***1***

***setName***

..... if the elements belong to a set named *setName*.

### 4.4.3 BOUNDARY CONDITIONS

This command defines displacement boundary conditions.

**Command syntax** :

*boundary conditions*

*N*

$presc_1$

$ifpre_1$

$nset_1$

$presc_2$

$ifpre_2$

$nset_2$

.....

$presc_N$

$ifpre_N$

$nset_N$

where

- $N$ is the number of boundary conditions to be defined;
- $presc_n$ is a list of values of the degrees of freedom to be constrained relavant to the n-th boundary condition;
- $ifpre_n$ is a list of the degrees of freedom to be constrained relevant to the n-th boundary condition;
- $nset_n$ is a list of constrained nodes relevant to the n-th boundary condition.

**Input file usage**

*title Example 1*

.....

*end*

***boundary conditions***

***2***

***0 0 0 0***

***1 3 5 6***

***base***

***0 0***

***2 4***

***17 28 73 45 90 100 1 4 98 =***

***34 33***

.....

*end option*

.....

In this example two boundary conditions have been assigned. For the first boundary condition, null displacement along x and z axes and null rotation around the y and z axes are applied to all nodes belonging to the set *base*; in the second case null displacement along y axis and null rotation around the x axis are applied to the list of nodes.

**Remarks:**

this keyword is obsolete and the use of FIXED DISPLACEMENT option should be preferred.

### 4.4.4 COMMENT

This keyword is used to insert a comment; the user can insert a comment anywhere.

**Command syntax** :

*comment "comment text"*

or

*$ "comment text"*

**Input file usage**

*title Example 1*

*.....*

*end*

*connectivity*

*10*

*1 10 67 92 37 45*

*2 10 29 31 47 89*

***comment this example explain the use of NOSA card***

*.....*

*end option*

*.....*

### 4.4.5 COMPOSITION

This keyword is used to assign a material (previously defined) to a set of mesh elements.

**Command syntax** :

*composition*

*N*

*mat-1*

*set-1*

*mat-2*

*set-2*

*....*

*mat-N*

*set-N*

where

- *N* is the number of sets of elements to which material behavior has to be assigned;
- *mat-n* is the identification number of the n-th material (previously defined);
- *set-n* is the list of elements which the n-th material refers to.

**Input file usage**

*title Example 1*

*sizing 543228 45379*

.....

*end*

.....

***composition***

***2***

***1***

***arco1***

***2***

***arco2***

*property*

*2*

*1 1.2300e+08 2.2000e-01 2.0350e+02 0.0 1000.0 1000000000000000000.0 0.0 0.0 0.0 0.0*

*2 2.8000e+08 2.2000e-01 2.3000e+02 0.0 1000.0 1000000000000000000.0 0.0 0.0 0.0 0.0*

.....

*end option*

.....

In this example, the first set of elements "arco1" is made of material 1, whose Young Modulus is equal to 1.23e+08 Pa; the other set "arco2" is made of material 2, whose Young modulus is equal to 2.8e+08 Pa.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Create Material*

**Remarks:**

1. If the COMPOSITE option has been set, *matN* is a **list** of identification number of the materials associated with each layer of shell elements belonging to n-th set; the number of materials into this list is less or equal to the number of layers of shells (i.e., MSHEL). If it is less than MSHEL, the last material in the list applies to all other layers of shell elements.
2. COMPOSITE option can be used together with only shell elements.
3. Currently, COMPOSITE option **cannot be defined through NOSA-ITACA/GUI**.

**See also:**

- PROPERTY,
- SHELL SECT,
- SALOME Nosa User's Guide: Create Materials

### 4.4.6 CONNECTIVITY

This keyword is used to define nodal connectivity of mesh elements.

**Command syntax** :

*connectivity*

*Nelem*

*elemID elemType nodeI nodeJ nodeK ...*

where

- *Nelem* is the number of elements for which nodal connectivity has to be defined. *Nelem* is less or equal to the maximum number of elements of the mesh model;
- *elemID* is the identifier number of an element;
- *elemType* is the type of element, according to **NOSA Element Library**;
- *nodeI*, *nodeJ*, *nodeK ...* are the identifier numbers of nodes of the element, representing the connectivity; with reference to the theory of **Finite Element Modelling**, the sequence of node Ids into the connectivity of an element is *counterclockwise*, and such that firstly corner nodes are specified and then the mid-side nodes;
- the last line of the command syntax has to be repeated (Nelem -1) times, with each line referring to the nodal connectivity of one element; these line can contain a maximum number of nine items, thus for elements having a number of nodes greater or equal to 8, the connectivity must be continuated to the next line ( by means of the symbol "=").

**Input file usage**

*title Example 1*

*sizing 100 150*

.....

*end*

.....

***connectivity***

***100***

***1 10 25 34 72 65***

***2 10 23 27 45 92***

***3 9 138 27***

.....

***100 1 600 601 602 603 604 605 606 =***

***607 608 609 610 611 612 613 614 615 =***

***616 617 618 619***

.....

*end option*

.....

In this example the mesh model is made up of 100 elements; the first element (elemID = 1) is a 4-node thick shell (element type = 10) with nodal connectivity expressed by the node sequence 25, 34, 72, 65; the third element (elemID = 3) is a 2-node beam element (element type = 9) with nodal connectivity given by node Ids 138 and 27. The last element (elemID = 100) is a 20-node hexahedric element (element type = 1) with nodal connectivity given by a sequence of twenty nodes; the last one is defined on three lines.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Nosa Element Library*

**See also:**

- SIZING,
- NOSA Theory Manual: Element Library,
- SALOME Nosa User's Guide: Specify the Type of Element

### 4.4.7 CONTROL

The keyword is used to set up the parameters for controlling the accuracy of numerical analyses, to choose algorithms to be used to solve equilibrium equations and setting other solution controls.

**Command syntax** :

*control*

*maxinc miter algoID irsdis*

*toler fuztol fuzdis fuztmp tmpfin*

where

- *maxinc* is the maximum number of load increments; by default, *maxinc* is equal to 1000;
- *miter* is the maximum number of iterations for each load increment; the default value is 1000;
- *algoID* is the identifier of the algorithm to be used to solve equilibrium equations; possible values are:
  - *algoID = 1* : The **initial stiffness matrix method** is used;
  - *algoID = 2* : The **modified Newton-Raphson method** is used;
  - *algoID = 3* : The **Newton-Raphson method** is used; by default, this method is applied to solve equilibrium equations.
- *irsdis* is a flag to set up convergence criteria based on displacement field; when *irsdis* is not null, the displacement based convergence check is turned on in dynamic analyses;
- *toler* is the ratio in % between the norm of residual force and the norm of total force (including reactions), representing the tolerance used for the convergence of a load increment. For shell and beam elements, the moments are divided by the corresponding thickness to be dimensionally homogeneous to forces. If the convergence check is to be done on displacement variation, it is the ratio between the iteration displacement change and the incremental displacement variation; in such a case, at least two iterations are required in order that an increment can achieve the convergence;
- *fuztol* is the minimum value of the norm of total force; if this norm is less than *fuztol*, the convergence control is skipped (default 1.0D-08) in static analyses, or the displacement based convergence check is automatically turned on in dynamic runs;
- *fuzdis* is the minimum displacements change for dynamic analyses; if the norm of displacement is less than *fuzdis*, the convergence control is skipped (default 1.0D-12);
- *fuztmp* is the minimum temperature change for heat transfer analyses; if the maximum temperature change is less than *fuztmp*, the convergence control is skipped (default 1.0D-08);

- *tmpfin* is the maximum temperature change allowed in automatic time stepping mode (default 20.D0), for heat transfer analyses; if the maximum temperature change is greater than *tmpfin*, the increment is repeated with reduced time step in order that the maximum temperature change is equal to *tmpfin* (this can be obtained with precision in linear cases).

**Input file usage**

*title Example 1*

*sizing 43228 45379*

.....

*end*

.....

***control***

***1000 1000 3***

***0.10 1.00e-08***

.....

*end option*

.....

In the above example the Newton-Raphson method will be used to solve equilibrium equations; a maximum number of 1000 load increments are allowed, for each of them a maximum number of 1000 iterations will be performed to achieve the numerical convergence; the latter is achieved within a residual equal to 0.1% of the total force. All other parameters have been set to default values.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Solution Controls*

**See also:**

- NOSA Theory Manual: Solution Controls,
- SALOME Nosa User's Guide: Set Solution Controls

### 4.4.8 COORDINATES

This command defines the coordinates of nodes.

**Command syntax** :

*coordinates*

*dimSpace nNodes*

*nodeID xcoord ycoord ...*

where

- *dimSpace* is the maximum number of coordinates per node;
- *nNodes* is the total number of nodes;
- *nodeID* is the identifier number of a node;
- *xcoord*, *ycoord*, ... are the coordinates of the nodeID along the x-axis, y-axis ..., respectively.

- The last line of the command syntax has to be repeated (nNodes -1) times with each line representing the coordinates of a single node.

**Input file usage**

*title Example 1*

*sizing 100 150*

.....

*end*

.....

***coordinates***

***3 150***

***1 0.0 0.0 0.0***

***2 10.0 10.0 10.0***

.....

***150 23.0 45.0 -2.0***

.....

*end option*

.....

### 4.4.9 DAMPING

This command defines the damping coefficients in dynamic analyses, according to the Rayleigh assumption.

**Command syntax** :

*damping*

*N*

$coeff1_1$ $coeff2_1$ $coeff3_1$

$set_1$

$coeff1_2$ $coeff2_2$ $coeff3_2$

$set_2$

...

$coeff1_N$ $coeff2_N$ $coeff3_N$

$set_N$

where

- *N* is the number of element sets which damping coefficients have to be assigned to;
- $coeff1_n$ is the value of the multiplier of the mass matrix for viscous damping, relevant to n-th set of elements; by default $coeff1_n = 0.0$;
- $coeff2_n$ is the value of the multiplier of the stiffness matrix for viscous damping, relevant to n-th set of elements; by default $coeff2_n = 0.0$;

- $coeff3_n$ is the value of the multiplier of the stiffness matrix for numerical damping, relevant to n-th set of elements; by default $coeff3_n = 0.0$;

- $set_n$ is the group of elements for which damping coefficients are defined.

**Input file usage**

*title Example 1*

*sizing 100 150*

*element 10 9*

*materials 1*

*setname 2 2 150*

*damping*

*dynamic*

*shell section 3*

*end*

.....

*coordinates*

.....

*connectivity*

.....

***damping***

***2***

***1.0 1.0 1.0***

***23 45 19 22 47 93 102 100 87 =***

***1 3***

***2.0 5.0 0.0***

***33 27 91***

.....

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Boundary Conditions: Boundary Condition Type:* ***Dynamic*** *, Boundary Boundary Condition Family:* ***Damping***

**See also:**

- DYNAMIC,
- DAMPING,
- SALOME Nosa User's Guide: Set Boundary Conditions

### 4.4.10 DEFINE

This command defines sets of element and/or nodes.

**Command syntax** :

*define element/node* (or *element/node set*) *setName*

*itemList*

where

- *setName* is the name of the set (maximum 8 characters);
- *itemList* is a list of elements or nodes constituting the set.

**Input file usage**

*title Example 1*

*sizing 100 150*

*element 10*

*materials 1*

*setname 1 1 150*

*shell section 3*

*end*

.....

***define element set GRUPPO1***

***23 45 19 22 47 93 102 100 87 =***

***1 3***

***define node set GRUPPO2***

***33 27 91***

.....

*end option*

.....

**Remarks:**

- The names of other sets **cannot** be included into the list of set definition.
- A set **cannot** be referenced into the input data before being defined.

### 4.4.11 DYNAMIC

This keyword defines the integration coeffients to be used for dynamic analyses.

**Command syntax** :

*dynamic*

$\gamma_{NEW}$ $\beta_{NEW}$ $\alpha_{HHT}$

where

- $\gamma_{NEW}$ is the value of the $\gamma$-coefficient of the Newmark method; by default, $\gamma_{NEW} = 0.5$.
- $\beta_{NEW}$ is the value of the $\beta$-coefficient of the Newmark method; by default, $\beta_{NEW} = 0.25$.
- $\alpha_{HHT}$ is the value of the $\alpha$-coefficient of the Hilber-Hughes-Taylor method; by default, $\alpha_{HHT} = 0.0$. If the Hilber-Hughes-Taylor method has been chosen, $\gamma_{NEW}$ and $\beta_{NEW}$ coefficients will be reset to $0.5 + \alpha_{HHT}$ and $0.25 \cdot (1.0 + \alpha_{HHT})^2$, respectively.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 19 9 45379*

*dynamic*

*shell section 3*

*end*

*coordinates*

.....

*connectivity*

.....

***dynamic***

***0.6 0.3***

.....

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Jobs: Job Type tab:* ***Dynamic Analysis*** *-> Newmark Algorithm or Hilber-Hughes-Taylor Algorithm*

**See also:**

- HHT
- DYNAMIC

### 4.4.12 END OPTION

This command declares the end of the model section of a crd file.

**Command syntax** :

*end option*

**Input file usage**

*title Example 1*

*sizing 543228 45379*

*element 10 9*

*materials 4*

*setname 19 9 45379*

*alias 75 10 98 9*

*beam sect 6 5*

*dist load 19*

*masonry*

*shell section 11*

*tie 914 1*

*end*

*coordinates*

.....

*connectivity*

.....

.....

***end option***

.....

**Remarks:**

This command is mandatory.

### 4.4.13 FILM COEFFICIENT

This command defines film coefficients to be used in heat transfer analyses.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*film coefficient*

*N*

*faceID film sink indfil indsin*

*set-1*

where

- *N* is the number of element sets which film coefficients have to be assigned to;
- *faceID* is the identifier number of a face of elements belonging to the *set-1*;
- *film* is the value of the film coefficient to be applied to the elements belonging to the *set-1*; by default, *film = 0.0*.

- *sink* is the value of the sink temperature to be assigned to the elements of the *set-1*; by default, *sink = 0.0*.

- *indfil* is a flag such that the film coefficient will be evaluated by means of the user routine **UFILM**, when *indfil* is not null;

- *indsin* is a flag such that the sink temperature will be calculated by means of the user routine **UFILM**, when *indsin* is not null;

- *set-1* is a group of elements which these coefficients refer to.

- the third and the last line have to be repeated (N-1) times, because they are relevant to only one set of elements.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 15*

*materials 2*

*setname 19 9 45379*

*film*

*end*

.....

***film coefficient***

***2***

***1 0.5 15 0 0***

***34 45 97 88 23 12 1 4 5 =***

***101 107 109***

***4 0.5 15 0 0***

***GRUPPO1***

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Boundary Conditions: Boundary Condition Type:* ***Heat Transfer*** *, Boundary Condition Family:* ***Film Coefficient***

**See also:**

- FILM
- SALOME Nosa User's Guide: Set Boundary Conditions
- User Subroutines Reference Guide: UFILM,

### 4.4.14 FIXED ACCELERATION

This command defines acceleration boundary conditions for dynamic analyses.

**Command syntax** :

*fixed acceleration*

*N*

*prsca-1*

*iprea-1*

*set-1*

*prsca-2*

*iprea-2*

*set-2*

*.....*

*prsca-N*

*iprea-N*

*set-N*

where

- *N* is the number of boundary conditions to be defined;
- *prsca-n* is a list of values of the degrees of freedom to be constrained, relevant to the n-th boundary condition;
- *iprea-n* is a list of the degrees of freedom to be constrained, relevant to the n-th boundary condition;
- *set-n* is a list of constrained nodes relevant to the n-th boundary condition.

**Remarks:**

When the list of degrees of freedom contains negative values, the user routine **UBND** will be called to constrain the DOF with negative value of all nodes belonging to the given set.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*dynamic*

*shell section 3*

*end*

*....*

*dynamic*

*0.6* 0.3

***fixed acceleration***

***2***

***0.3 0.1***

***1 2***

***27 28 33 45 120***

***0.1 0.1***

***1 -3***

***GRUPPO1***

.....

*end option*

.....

In the above example two boundary conditions have been assigned. For the first boundary condition finite values have been imposed to the translational accelerations along x and y axes to nodes into the given list; in the second case finite value to the translational acceleration along x-axis have been imposed to all nodes of the set *GRUPPO1*. With reference to the translational acceleration along the z-axis, its value have been defined by means of the user routine UBND.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Boundary Conditions: Boundary Condition Type:* ***Dynamic****, Boundary Condition Family:* ***Acceleration***

**See also:**

- DYNAMIC
- SALOME Nosa User's Guide: Boundary Coditions
- User Subroutines Reference Guide: UBND

### 4.4.15 FIXED DISPLACEMENT

This command defines displacement boundary conditions.

**Command syntax** :

*fixed displacement*

*N*

*presc-1*

*ifpre-1*

*set-1*

*presc-2*

*ifpre-2*

*set-2*

.....

*presc-N*

*ifpre-N*

*set-N*

where

- *N* is the number of boundary conditions to be defined;
- *presc-n* is a list of values of the degrees of freedom to be constrained, relavant to the n-th boundary condition;
- *ifpre-n* is a list of the degrees of freedom to be constrained, relevant to the n-th boundary condition;
- *set-n* is a list of constrained nodes relevant to the n-th boundary condition.

**Remarks:**

- When the list of degrees of freedom contains negative values, the user routine **UBND** will be called to constrain the DOF with negative value of all nodes belonging to the given set.
- For static analyses displacement boundary conditions have to be defined in order to avoid rigid body motions.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*shell section 3*

*end*

.....

***fixed displacement***

***2***

***0.0 0.0***

***1 2***

***27 28 33 45 120***

***0.0 0.0***

***3 -4***

***GRUPPO1***

.....

*end option*

.....

In the above example two boundary conditions have been assigned. For the first boundary condition null values have been imposed to the translation along x and y axes for nodes into the given list; in the second

case null value to the translation along z-axis have been imposed to all nodes of the set *GRUPPO1*. With reference to the rotation around the x-axis, the constraint have been defined by means of the user routine UBND.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Boundary Conditions: Boundary Condition Type:* ***Static****, Boundary Condition Family:* ***Displacement***

**See also:**

- SALOME Nosa User's Guide: Set Boundary Conditions
- User Subroutines Reference Guide: UBND

### 4.4.16 FIXED TEMPERATURE

This command defines temperature boundary conditions for heat transfer analyses.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*fixed temperature*

*N*

*prsct-1*

*ipret-1*

*set-1*

*prsct-2*

*ipret-2*

*set-2*

.....

*prsct-N*

*ipret-N*

*set-N*

where

- *N* is the number of boundary conditions to be defined;
- *prsct-n* is a list of values of the degrees of freedom to be constrained, relavant to the n-th boundary condition;
- *ipret-n* is a list of the degrees of freedom to be constrained, relevant to the n-th boundary condition;
- *set-n* is a list of constrained nodes relevant to the n-th boundary condition.

**Remarks:**

When the list of degrees of freedom contains negative values, the user routine **UBND** will be called to constrain the DOF with negative value of all nodes belonging to the given set.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*shell section 3*

*end*

.....

***fixed temperature***

***2***

***0.0 0.0***

***1 2***

***27 28 33 45 120***

***1 -0.2***

***1 -3***

***GRUPPO1***

.....

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Boundary Conditions: Boundary Condition Type:* ***Heat Transfer****, Boundary Condition Family:* ***Temperature***

**See also:**

- HEAT TRANSFER
- SALOME Nosa User's Guide: Set Boundary Conditions
- User Subroutines Reference Guide: UBND

### 4.4.17 FRICTION

The keyword is used to define friction coefficients for contact problems.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*friction*

*N*

*Value-1*

*D-1*

*S-1*

*P-1*

*Value-2*

*D-2*

*S-2*

*P-2*

...

*Value-N*

*D-N*

*S-N*

*P-N*

where

- *N* is the number of friction coefficients to be defined;
- *Value-n* is the value of the n-th friction coefficient;
- *D-n* is a list of deformable bodies which the n-th friction coefficient refers to;
- *S-n* is a list of rigid surfaces which the n-th friction coefficient refers to;
- *P-n* is a list of basic shapes of the rigid surfaces into the list S-n.

**Remarks:**

- The values of the friction coefficient will be stored into the 3-D array *CFRICT*, such that *CFRICT(I,J,K)* represents the friction coefficient of the interface between the K-th deformable body and the I-th part of the J-th rigid surface;
- If *Value-n* is negative, the friction coefficient will be defined by means of the user routine **UFRI**;
- Currently only one deformable body can be defined in the release NOSA-ITACA 1.0.
- This keyword **is not implemented** into NOSA-ITACA/GUI.

**See also:**

- DEFORMABLE BODY
- RIGID SURFACE
- User Subroutines Reference Guide: UFRI

### 4.4.18 GEOMETRY

This keyword is used to define the thickness for plane stress elements (element type 2), plane strain elements (element types 3 and 6) and the thickness of layers/fibers for shell/beam elements (element type 5, 9 and 10). For other element types this keyword is unnecessary.

**Command syntax** :

*geometry*

*N*

*thick-1*

*set-1*

*thick-2*

*set-2*

.....

*thick-N*

*set-N*

where

- *N* is the number of element sets for which cross-section property has to be defined;
- *thick-n* is the thickness of elements or layers/fibers; by default *thick-n = 1.0*.
- *set-n* is a list of elements which cross-section property refers to.

**Input file usage**

*title Example*

*sizing 100 150*

*element 10 9*

*materials 1*

*setname 4 3150*

*end*

.....

*define element set plate*

*1 2 3 4 5 6 7 8 9 =*

*10 11*

*define element set cordolo*

*12 13 14*

***geometry***

***2***

***0.43***

***plate***

***0.6 0.6***

***cordolo***

.....

*end option*

.....

In this example, the homogeneous thick shell elements of the first set "plate" have a thickness equal to 0.43 m; the homogeneous beam elements of the set "cordolo" have a thickness equal to 0.6 m along the two local direction of beam cross-section.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Element Section Thickness*

**Remarks:**

1. If the COMPOSITE option has been set, *thick-n* is a **list** of thicknesses, where each item into the list represents the thickness of each layer of shell elements belonging to n-th set; the number of thicknesses into this list is less or equal to the number of layers of shells (i.e., MSHEL). If it is less than MSHEL, the last thickness into the list applies to all other layers of shell elements.
2. COMPOSITE option can be also used together with beam elements; in such a case, for each fiber of beam elements two thicknesses have to be defined, each of them referring to a local direction of beam cross-section. Thus, for composite beam elements *thick-n* is a list with size equals to $2*Nfibers$ and such that the m-th pair into the list represents the thickness along the two local direction of the beam cross-section relevant to the m-th fiber.
3. Currently, COMPOSITE option **cannot be defined through NOSA-ITACA/GUI**.
4. Only for shell elements, thick-n can be a negative value (or can contain negative values if composite option has been used); in such a case, the thickness(es) will be evaluated by means of the user routine **UGEOM**.

**See also:**

- Control Section: BEAM SECT
- Control Section: SHELL SECT
- Control Section: COMPOSITE
- SALOME Nosa User's Guide: Element Section Thickness
- User Subroutines Reference Guide: UGEOM

### 4.4.19 HARDENING

This keyword is used to define work-hardening behaviour of elastic-plastic materials; hardening curves will be defined in a tabular form.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*hardening*

*N*

```
(the following lines must be repeated N times)
```

*matID M haiso*

*h(1,1) h(1,2)*

*h(2,1) h(2,2)*

....

*h(M,1) h(M,2)*

where:

- *N* is the number of hardening curves to be defined;
- *matID* is the identification number of the material (as defined in the PROPERTY keyword), which the hardening curve refers to;

- *M* is the number of rows of the hardening table, *h*, defining the hardening curve;
- *haiso* is a value defining the amount of isotropic behaviour of a combined work-hardening rule; thus, for *haiso = 0.0* the hardening behaviour is purely kinematic. For *haiso = 1.0* the hardening rule is purely isotropic;
- *h* is the hardening table, with size *M∗2*, and such that:
  - *h(M,1)* is the M-th value of the equivalent accumulated plastic strain; in this table *h(1,1)* is equal to 0.0;
  - *h(M,2)* is the M-th value of the von-Mises equivalent stress corresponding to the M-th plastic strain.

**Input file usage**

*title 2-D plane strain analysis of CT specimen under SSY condition*

*sizing 863 2716*

*elements 3*

*materials 1*

*setname 1 4 2716*

*energy*

*hardening 5*

*end*

*coordinates*

*3 2716*

*1 2.55000000e+01 0.00000000e+00 0.00000000e+00*

*2 3.00000000e+01 0.00000000e+00 0.00000000e+00*

*...*

*2716 1.05640411e+00 2.44548570e+01 0.00000000e+00*

*connectivity*

*863*

*1 3 2 18 212 17 933 1146 1145 932*

*2 3 18 3 22 212 934 939 1147 1146*

*...*

*863 3 927 172 13 211 2700 1103 1144 2716*

*property*

*1*

*1 7.1000e+04 3.0000e-01 0.0000e+00 0.0000e+00 6.5000e+02*

*.....*

***hardening***

***1***

***1 5 1.00***

***0.0000 650.0000***

***0.0001 660.0000***

***0.0010 680.0000***

***0.0100 700.0000***

***0.4000 900.0000***

*end option*

...

In this example an elastic-plastic material with isotropic hardening behaviour has been considered; hardening rule has been described by means of a table with 5 rows: (0.0;650.0),(1.0e-04;660.0),(1.0e-03;680.0),(1.0e-02;700.0) and (4.0e-01;900.0).

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Create Material* : ***Elastic-Plastic*** *->* ***Hardening Rules*** *tab*

**See also:**

- Control Section: HARDENING
- PROPERTY
- SALOME Nosa User's Guide: Create Materials

### 4.4.20 INITIAL DISPLACEMENT

This command defines the initial nodal displacement in dynamic analyses.

**Command syntax** :

*initial displacement*

*N*

*dispi-1*

*idofs-1*

*set-1*

*dispi-2*

*idofs-2*

*set-2*

.....

*dispi-N*

*idofs-N*

*set-N*

where

- *N* si the number of initial boundary conditions to be defined;
- *dispi-n* is a list of initial values of the components of the displacement field, relavant to the n-th boundary condition;

- *idofs-n* is a list of the degrees of freedom associated with *dispi-n* ;
- *set-n* is a list of nodes relevant to the n-th boundary condition, which initial displacement field applies to.

**Remarks:**

- When the list of degrees of freedom contains negative values, the user routine **UDSPI** will be called to define the initial displacement field relevant to the DOF with negative value of all nodes belonging to the given set.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*dynamic*

*shell section 3*

*end*

.....

*dynamic*

*0.6 0.3*

***initial displacement***

***2***

***0.3 0.1***

***1 2***

***27 28 33 45 120***

***0.1 0.1***

***1 -3***

***GRUPPO1***

.....

*end*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Initial Conditions: Initial Condition Type:* ***Static****, Initial Condition Family:* ***Displacement***

**See also:**

- Control Section: DYNAMIC
- DYNAMIC
- SALOME Nosa User's Guide: Set Initial Conditions
- User Subroutines Reference Guide: UDSPI

### 4.4.21 INITIAL TEMPERATURE

This command defines the initial or reference nodal temperature.

**Command syntax** :

*initial temperature*

*N*

*tempi-1*

*iuser-1*

*set-1*

*tempi-2*

*iuser-2*

*set-2*

...

*tempi-N*

*iuser-N*

*set-N*

where

- *N* is the number of the initial conditions to be defined;
- *tempi-n* is a list of values of the initial nodal temperature; by default the initial nodal temperature is equal to 0.0.
- *iuser-n* is a sequence of integers 1 and/or -1 (i.e. index of the temperature degree of freedom); if negative values are specified into this list, the user routine **UTEMPI** will be called to define the initial temperature for the d.o.f with negative value;
- *set-n* is a list of nodes which the initial temperature condition applies to.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*shell section 3*

*end*

.....

***initial temperature***

***2***

***25.0***

***1***

***GRUPPO1***

***17.0***

***-1***

***27 28 33 45 120***

.....

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Initial Conditions: Initial Condition Type:* ***Heat Transfer****, Initial Condition Family:* ***Temperature***

**See also:**

- Control Section: HEAT TRANSFER
- SALOME Nosa User's Guide: Set Initial Conditions
- User Subroutines Reference Guide: UTEMPI

### 4.4.22 INITIAL VELOCITY

This command defines the initial nodal velocity in dynamic analyses.

**Command syntax** :

*initial velocity*

*N*

*veloi-1*

*idofs-1*

*set-1*

*veloi-2*

*idofs-2*

*set-2*

.....

*veloi-N*

*idofs-N*

*set-N*

where

- *N* si the number of initial boundary conditions to be defined;
- *veloi-n* is a list of initial values of the components of the velocity field, relavant to the n-th boundary condition;
- *idofs-n* is a list of the degrees of freedom associated with *veloi-n* ;
- *set-n* is a list of nodes relevant to the n-th boundary condition, which initial velocity field applies to.

**Remarks:**

- When the list of degrees of freedom contains negative values, the user routine **UVELI** will be called to define the initial velocity field relevant to the DOF with negative value of all nodes belonging to the given set.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*dynamic*

*shell section 3*

*end*

.....

*dynamic*

*0.6* 0.3

***initial velocity***

***2***

***5.0 6.0***

***1 2***

***27 28 33 45 120***

***0.5 0.7***

***1 -3***

***GRUPPO1***

.....

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Initial Conditions: Initial Condition Type:* ***Dynamic****, Initial Condition Family:* ***Velocity***

**See also:**

- Control Section: DYNAMIC
- DYNAMIC
- SALOME User' Guide: Set Initial Conditions
- User Subroutines Reference Guide: UVELI

### 4.4.23 LOCAL AXIS

The keyword is used to define local reference systems for beam elements.

**Command syntax** :

*local axis*

*N*

```
(the following lines must be repeated N times)
```

*iuaxi axis*

*set*

where

- *N* is the number of local systems to be defined;
- *iuaxi* is an integer, if different from zero indicates that the local system will be defined by means of the user routine **ULAXIS**; by default, *iuaxi = 0*
- *axis* is the unit vector *v1* along the first local direction of the beam cross section. The third local direction *v3* is the beam axis, and the second local direction *v2* is defined by the cross product *v2 = v3 x v1*.
- *set* is the group of beam elements which the local reference system refers to.

**Input file usage**

*title Example*

*sizing 100 150*

*element 10 9*

*materials 1*

*setname 4 3150* 4 3150

*end*

.....

*define element set plate*

*1 2 3 4 5 6 7 8 9 =*

*10 11*

*define element set cordolo*

*12 13 14*

*geometry*

*2*

*0.43*

*plate*

*0.6 , 0.6*

*cordolo*

***local axis***

***1***

***0 0.0 ,0.0 , 1.0***

***cordolo***

.....

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Local Reference System*

**See also:**

- SALOME Nosa User's Guide: Define local reference systems
- User Subroutines Reference Guide: ULAXIS

### 4.4.24 LUMPED MASS

This command declares the use of lumping in the calculation of the mass matrix.

**Command syntax** :

*lumped mass*

*N*

*lump-1*

*set-1*

*lump-2*

*set-2*

....

*lump-N*

*set-N*

where

- *N* is the number of element sets, for which lumping option is active;
- *lump-n* is an integer, if different from zero indicates that the mass matrix will be lumped by summing a row into the diagonal element; by default, *lump-n = 0*
- *set-n* is a list of elements with lumping option.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 2*

*setname 19 9 45379*

*lumped*

*dynamic*

*shell section 11*

*end*

.....

*dynamic*

*0.6 0.3*

***lumped mass***

***2***

***0***

***GRUPPO1***

***1***

***GRUPPO2***

.....

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Jobs:* ***Other Options*** *tab -> Lumped Mass Matrix*

**See also:**

- SALOME Nosa User's Guide: Job Options

### 4.4.25 MASONRY

This command defines masonry-like materials.

**Command syntax** :

*masonry*

*N*

*maso-1*

*set-1*

*maso-2*

*set-2*

...

*maso-N*

*set-N*

where

- *N* is the number of element groups made of masonry-like material;
- *maso-n* is an integer defining the kind of masonry-like material; possible values are:

  - *maso-n = 0* for elastic-plastic material;
  - *maso-n = 1* for masonry material with low tensile strength and infinite compressive strength;
  - *maso-n = 2* for masonry material with low tensile strength and bounded compressive strength;

- *set-n* is the list of elements with these material properties.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 3*

*setname*

*shell 3 2 45379* section 3

*masonry*

*end*

.....

***masonry***

***3***

***1***

***GRUPPO1***

***2***

***GRUPPO2***

***0***

***GRUPPO3***

.....

*composition*

*3*

*3*

*GRUPPO1*

*1*

*GRUPPO2*

*2*

*GRUPPO3*

*property*

*3*

*1 3.0e+04 2.0e-01 1.83e+02 0.0000e+00 1.0000e-01 1.0000e+05*

*2 2.0e+04 1.0e-01 1.83e+02 0.0000e+00 1.0000e+20 1.0000e+20*

*3 1.0e+04 1.5e-01 1.83e+02 0.0000e+00 1.0000e-01 1.0000e+20*

*end option*

.....

In this example three sets of elements, GRUPPO1, GRUPPO2 and GRUPPO3 have been defined; GRUPPO1 is made of a masonry-like material with low tensile strength and unbounded compressive strength. Its mechanical properties are given by the last line of the "PROPERTY" keyword (i.e, from the "COMPOSITION" keyword the material associated with GRUPPO1 has an identification number equal to 3). GRUPPO2 is made of masonry-like material with low tensile strength and bounded compressive strength (see the properties of material id 1). Finally, GRUPPO3 is made of an elasti-plastic material with unbounded tensile and compressive strength (i.e., an elastic material whose properties are referred to material id 2).

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Create Material -> Masonry*

**See also:**

- COMPOSITION
- PROPERTY
- SALOME Nosa User's Guide: Create Materials

### 4.4.26 MASSES

The keyword is used to define concentrated generalized masses for dynamic analyses.

**Command syntax** :

*masses*

*N*

```
(the following lines must be repeated N times)
```

*iuser pmass*

*set*

where

- *N* is the number of node sets to which concentrated masses have to be applied;
- *iuser* is an integer, if different from zero, the user routine **UPMASS** will be called to define the values of the concentrated masses for each DOF; by default, *iuser = 0*
- *pmass* is a list of values of the concentrated masses;
- *set* is a list of nodes with concentrated mass.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 2*

*setname 19 9 45379*

*dynamic*

*shell section 11*

*end*

.....

*dynamic*

*0.6 0.3*

***masses***

***2***

***0 40***

***GRUPPO1***

***1 1***

***GRUPPO2***

.....

*end option*

.....

In this example two nodal sets, GRUPPO1 and GRUPPO2, have been defined; a concentrated mass of 40 Kg along x-axis has been applied to nodes belonging to GRUPPO1, while the concentrated mass applied to nodes of GRUPPO2 has been defined by means of the user routine UPMASS.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Nodal Property*

**See also:**

- SALOME Nosa User's Guide: Define Concentrated Masses to Nodes
- User Subroutines Reference Guide: UPMASS

### 4.4.27 POST

The keyword defines the output requests and the format of the output file.

**Command syntax** :

*post*

*NumVar freq irev format iterFreq icont wrtim*

```
(The following line must be repeated NumVar times)
```

*IdVar nameVar*

where

- *NumVar* is the number of variables to be written to the output file; the maximum number of output variables is 200.
- *freq* is the frequency with which output variables will be written to the post-processing file at the end of a load increment. If *freq* is not null, the output will be written at the end of every *freq-th* load increments, after the last written. If *freq = 0* no data will be written to the output file;

- *irev*, has to be set to its default value (i.e., equal to 0);
- *format* defines the type of the output file:
  - *format = 0* the output file will be a formatted text file with extenstion **".t19"**;
  - *format = 1* the output file will be a binary file with extension **".t16"**;
- *iterFreq* is the frequency with which output variables will be written to the output file at several iterations of each load increment; by default, *iterFreq = 0*, which means no iteration result will be written;
- *icont* is an integer number useful if a restart analysis is required; in such a case:
  - *icont = 0* means that the initial block of the output file relevant to element connectivity and nodal coordinates will not be printed;
  - *icont = 1* means that the initial block of the output file will be printed;
- *wrtim* is a deprecated option to be set to its default value (i.e., equal to 0.0);
- *IdVar* is the code number of the output variable (see Table 4.1);
- *nameVar* is the name of the output variable (see Table 4.1); the name is a string of characters not longer than 24.

| *IdVar* | *Output Variables* |
|---|---|
| 1, 2, ..., 6 | components of the total strain tensor (i.e., E11, E22,..., E13) |
| 9 | total temperature |
| 11, 12, ..., 16 | components of the Cauchy stress tensor (i.e., S11, S22,...,S13) |
| 17 | Von Mises equivalent stress (i.e. SMISES) |
| 21, 22, ..., 26 | components of the plastic strain (i.e., EPL11, EPL22,...,EPL13)<br>or fracture strain for masonry materials (i.e., EF11, EF22,...,EF13) |
| 27 | equivalent plastic strain (EPLEQV) for elastic-plastic materials<br>or norm of the tensile inelastic strain (EFEQV) for masonry materials |
| 31, 32, ..., 36 | components of the compression inelastic strain<br>(i.e., EC11, EC22,...,EC13) for masonry materials |
| 37 | norm of the compressive inelastic strain (ECEQV) for masonry materials |
| 41, 42, ..., 48 | components of strain resultants (i.e., EC11, EC22,...,EC13) for shell elements |
| 51, 52, ..., 58 | components of stress resultants (i.e., N1, N2,...,Q23) for shell elements |
| 71 | total enthalpy (kinetic energy plus strain energy<br>minus the work done by the external forces) |
| 72 | total kinetic energy |
| 73 | total strain energy |
| 74 | total work done by the external forces (including the reactions) |
| 75 | total mechanical energy ( kinetic plus strain energy) |
| 81 | density of the enthalpy |
| 82 | density of the kinetic energy |
| 83 | density of the strain energy |
| 84 | density of the external work |
| 85 | density of the mechanical energy |
| 181, 182, 183 | components of temperature gradient in heat transfer analysis |

Table 4.1: Identification Code of output variables

**Remarks:**

A negative number of *IdVar* means that the variable will be defined through the user routine **PLOTV**. If an output variable relevant to a particular layer of shell elements must be written to the post file, the variable code must be in the form *1000∗I+J*, where *I* is the layer and *J* is the variable code as previously defined (or -(1000∗I+J) when the user routine PLOTV is required).

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*shell section 3*

*end*

*.....*

***post***

***17 1 0 0 0 0***

***51 N11***

***52 N22***

***53 N12***

***54 Q23***

***55 Q13***

***56 M11***

***57 M22***

***58 N12***

***-1011 s11-layer1***

***-1012 s22-layer1***

***-1014 s12-layer1***

***-4011 s11-layer4***

***-4012 s22-layer4***

***-4014 s12-layer4***

***-6011 s11-layer6***

***-6012 s22-layer6***

***-6014 s12-layer6***

*end option*

*.....*

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Output Requests*

**See also:**

- SALOME Nosa User's Guide: Output Requests
- User Subroutines Reference Guide: PLOTV

### 4.4.28 POTENTIAL HILL

This keyword is used to define plastic anisotropy according to the Hill's theory; furthermore, the initial rotation matrix, which defines the principal axes of anisotropy, is defined.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*potential hill*

*kmaph*

```
(the following lines must be repeated kmaph times)
```

*matID*, *rx*, *ry*, *rz*, *rxy*, *ryz*, *rxz*

*rsp0(1,1)*, *rsp0(1,2)*, *rsp0(1,3)*, *rsp0(2,1)*, ..., *rsp0(2,3)*, ..., *rsp0(3,3)*

where

- *kmaph* is the number of materials with anisotropic plasticity;
- *matID* is the identifier number of the material, which the data refer to;
- *rx*, *ry*, ..., *rxz* are the ratios between the yield stress values along the different directions and the reference value;
- *rsp0(1,1)*, ..., *rsp0(3,3)* are the items of the initial rotation matrix, evaluated between the global reference system and the anisotropy principal system.

**Input file usage**

*title Example*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 19 9 45379*

*potential 1*

*shell Section 11*

*end*

.....

*property*

*1*

*1 21.000e+10 3.0000e-01 7.8000e+03 0.0000e+00 6.5000e+08 1.0000e+20*

***potential hill***

***1***

***1 1.5 0.0 0.0 2.0 2.0 2.0***

***1.0 0.0 0.0 0.0 -1.0 0.0 0.0 0.0 -1.0***

.....

*end option*

.....

In this example an elastic- plastic material with anisotropic plasticity has been defined; according to the Hill's theory, plastic behaviour has been modelled by considering six ratios between yield stresses along the principal directions of plastic anisotropy and the reference value (equal to 650 MPa). Principal directions of plastic anisotropy have been evaluated with respect to the global reference system, by means of the rotation matrix.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Create Material:* ***Elastic-Plastic*** *->* ***Anisotropic Plasticity*** *tab*

**See also:**

- Control Section: POTENTIAL
- SALOME Nosa User's Guide: Create Materials

### 4.4.29 PROPERTY

This command defines mechanical properties of the material constituting the mesh elements; these properties depend on the type of analysis (static and/or dynamic and heat transfer analyses).

**Command syntax** :

`Static and/or dynamic analyses:`

*property*

*nMats*

`(the following lines must be repeated nMats times)`

*matID*, $E$, $\nu$, $\rho$, $\alpha_{th}$, $\sigma$, $\sigma_c$, $\beta$, $\alpha$, $E_1$, $E_2$

`heat transfer analyses:`

*property*

*nMats*

`(the following lines must be repeated nMats times)`

*matID*, $\kappa$, $c$, $\rho$, $\epsilon$

where:

- *nMats* is the number of materials, whose mechanical properties have to be defined;
- *matID* is the material identification number;
- $E$ is the *Young's modulus*;
- $\nu$ is the *Poisson's ratio*;
- $\rho$ is the mass density;

- $\alpha_{th}$ is the coefficient of thermal expansion;
- $\sigma$ is the uniaxial tensile yield stress for elastic-plastic materials, or the maximum tensile strength for masonry-like materials; by default, it's equal to 1.0E+20.
- $\sigma_c$ is the maximum compressive strength for masonry-like materials; default value is 1.0E+20;
- $\beta$ is the slope of the linear isotropic hardening curve, under the assumption of infinitesimal plasticity;
- $\alpha$ is the characteristic parameter of a kinematic hardening rule, defining the center of the elastic range, under the assumption of infinitesimal plasticity (similar to the *back stress* in finite strain condition);
- $E_1,\ E_2$ are other parameters of a kinematic hardening rule, under the assumption of infinitesimal plasticity;
- $\kappa$ is the *heat conductivity* (or thermal conductivity) of the material;
- *c* is the specific heat per unit mass;
- $\epsilon$ is the *emissivity* of the material.

**Remarks:**

- Usually isotropic, kinematic and mixed hardening rules are described by means of the user routines **CPHI** and **CPSI** and the keyword "HARDENING".
- This keyword is mandatory.

**Input file usage**

*title Example*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 19 9 45379*

*masonry*

*shell section 11*

*end*

.....

***property***

***1***

***1 3.0e+04 2.0e-01 1.83e+02 0.0000e+00 0.00000 1.0000e+20***

.....

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Create Material*

**See also:**

- Control Section: HARDENING
- HARDENING
- COMPOSITION
- SALOME Nosa User's Guide: Create Materials
- User Subroutines Reference Guide: CPHI
- User Subroutines Reference Guide: CPSI

### 4.4.30 ROTATION AXIS

This keyword is used to define the components of the normalized rotation axis and the coordinates of a point belonging to this axis, when centrifugal loads must be applied on the structure.

**Command syntax** :

*rotation axis*

*rotax-1 rotax-2 rotax-3*

*x1 x2 x3*

where

- *rotax-1*, *rotax-2* and *rotax-3* are the components of the unit vector defining the rotation axis;
- *x1*, *x2* and *x3* are the coordinates of the reference point belonging to the rotation axis, with respect to the global reference system;

**Remarks:**

For 2D problems the rotation axis is perpendicular to the plane of the structure, and, thus, only the coordinates *x1* and *x2* of the reference point are required. For axisymmetric problems, the rotation axis coincides with the axis of symmetry; in such a case the keyword can be omitted.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*shell section 3*

*end*

.....

***rotation axis***

***0.866 0.5 0.00663325***

***10.0 5.7 12.8***

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Rotation Axis*

**See also:**

- SALOME Nosa User's Guide: Define Rotation Axes

### 4.4.31 SELECTIVE INTEGRATION

The keyword is used to enable the reduced selective integration option; this feature can be used only for plane strain (element type 6), axisymmetric (element type 7) and brick elements (element type 8). If the option is enabled, reduced integration (i.e., one integration point) of the volumetric part of the deformation will be used.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*selective integration*

*N*

*sele-1*

*set-1*

*sele-2*

*set-2*

...

*sele-N*

*set-N*

where

- *N* is the numer of element set for which reduced integration is required;
- *sele-N* is a flag used to enable reduced integration; in particular, it assume the following values:
  - *sele-N = 0* full integration is required;
  - *sele-N = 1* reduced integration is required;
- *set-N* is the group of elements which reduced/full integration applies to.

**Remarks:**

The use of this keyword is recommended in finite strain analyses.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element*

*materials 1*

*setname 2 1 45379*

*end*

.....

*define* element set ALL_ELEM

*1* TO 43228

***selective integration***

***1***

***1***

***ALL_ELEM***

.....

*end option*

.....

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Jobs:* ***Other Options*** *tab -> Selective Integration*

**See also:**

- SALOME Nosa User's Guide: Job Options

### 4.4.32 SURFACE

This command is used to define rigid surfaces for 2D contact problems.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*surface*

*Nsurf*

`(the following lines must be repeated Nsurf times)`

*Npart*, $x^{RP}$, $y^{RP}$, $\dot{x}^{RP}$, $\dot{y}^{RP}$, $\dot{\Omega}^{RP}$

`(the following line must be repeated Npart times)`

*typeId*, *geometry*

where:

- *Nsurf* is the number of rigid surfaces to be defined;
- *Npart* is the number of basic shapes constituting a rigid surface;
- $x^{RP}$, $y^{RP}$ are global coordinates of the reference point of the rigid surface;
- $\dot{x}^{RP}$, $\dot{y}^{RP}$ are the components of velocity of the reference point (measured with respect to the global reference system);
- $\dot{\Omega}^{RP}$ is the angular velocity of the reference point (expressed in rad/sec, and assumed positive if counterclockwise);
- *typeId* is an integer defining the basic shape type; a basic shape of a rigid surface can be:

  - *typeId = 1* the basic shape is a segment;
  - *typeId = 2* the basic shape is a circumference arc (1st approach as shown in Table 4.2);
  - *typeId = 3* the basic shape is a circumference arc (2nd approach as shown in Table 4.2);

- *geometry* is a sequence of data required to describe the basic shape; depending on the basic shape type, data as shown in Table 4.2;

| *Basic shape* | *Data Request* |
|---|---|
| Segment | coordinate x of the first point<br>coordinate y of the first point<br>coordinate x of the end point<br>coordinate y of the end point |
| Circumference arc (1st approach) | coordinate x of the first point<br>coordinate y of the first point<br>coordinate x of the center<br>coordinate y of the center<br>angle in degree at center (positive if counterclockwise) |
| Circumference arc (2nd approach) | coordinate x of the first point<br>coordinate y of the first point<br>coordinate x of the end point<br>coordinate y of the end point<br>radius (positive if the center is within the rigid surface, negative otherwise) |

Table 4.2: Geometric data required to describe a basic shape of a rigid surface

**Remarks:**

- A rigid surface must be bounded by a 2D closed curve. The parts constituting the curve must be defined in such a way that the curve is counterclockwise oriented.
- This keyword **is not yet implemented** into NOSA-ITACA/GUI

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 3*

*materials 1*

*setname 2 1 45379*

*contact 2 2*

*end*

.....

***surface***

***2***

***2 5.0 7.0 1.0 0.5 0.01***

***1 -3.0 -2.*** **0.5 0 7.0**

***2 3.0 2.0 0.0 0.0 60***

***2 -3.0 -2.*** **0.5 0 7.0 40**

***3 3.0 2.0 0.0 0.0 5***

***2 1.0 0.0 3.0 0.7 0.02***

***1 -5.0 -4.*** **0.3 0 1.0**

***2 3.0 2.0 0.0 0.0 60***

***1 -3.0 -2.*** **0.5 0 7.0**

***3 3.0 2.0 0.0 0.0 -4***

.....

*end option*

.....

**See also:**

- Control Section: CONTACT
- Control Section: DEFORMABLE
- BODY

### 4.4.33 TEMP TABLE

This keyword is used to define temperature-dependent variables of a material; the variation of material properties with respect to temperature change will be defined in a tabular form;

**Command syntax** :

*temp table*

*N*, *matID*, *propID*

*t1* , *p1*

*t2* , *p2*

...

*tN* , *pN*

where

- *N* is the number of rows of the table used to define the temperature dependence;
- *matID* is the identifier number of the material, which the table refers to;
- *propID* is an identifier number of the material property, which depends on the temperature field; the meaning of *propID* depends whether stress analysis or heat transfer analysis will be carried out, as shown in Table 4.3.
- *tn* is the n-th temperature value;
- *pn* is the n-th value of *propID* associated with *tN*;

| *Analysis type* | *Value of propID* |
|---|---|
| Stress analysis | propID = 1 : Young's modulus, $E$<br>propID = 2 : Poisson's ratio, $\nu$<br>propID = 3 : coefficient of the linear thermal expansion, $\alpha^{th}$ |
| heat transfer analysis | propID = 1 : thermal conductivity, $\kappa$<br>propID = 2 : specific heat per unit mass, $c$<br>propID = 3 : emissivity, $\epsilon$ |

Table 4.3: Id number of material property depending on the temperature field

**Remarks:**

- The keyword defines only one material property as temperature dependent variable; if a material has more properties which depend on temperature field, the keyword must be used newly.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*temp* tables 1 3 *setname* 3 2 45379

*shell* section 3

*end*

.....

*property*

*1*

*1 2.1e+05 2.0e-01 1.83e+02 0.0000e+00 1.0000e+20 1.0000e+20*

***temp table***

***3 1 1***

***0 2.1e+05***

***25 2.1e+05***

***100 2.1e+04***

*end option*

.....

In this example the curve defining the dependence of the Young's modulus, *E*, on the temperature has been described through a table with three rows: (0;2.1e+05),(25;2.1e+05),(100;2.1e+04).

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Create Material*

**See also:**

- Control Section: TEMP TABLES
- SALOME Nosa User's Guide: Create Materials

### 4.4.34 TYING

The keyword is used to define multipoint constraints (MPCs) between nodal degrees of freedom; this option can be used for mesh refinements, as well as for defining tying relations between the degrees of freedom of different element types and setting the kinematic constraints along directions not aligned with the Cartesian reference axes.

**Command syntax** :

*tying*

*N*

```
(the following line must be repeated N times)
```

*mpcId*, *slaveId*, *masterIds*

where

- *N* is the number of multipoint constraints;
- *mpcId* is the identifier number of the tying relation;
- *slaveId* is the identifier number of the tied node (slave) of the tying relation or the name of the node set containing the tied nodes of the tying relation;
- *mastrIds* is a list of identifier numbers of the retained nodes (master) of the tying relation or a list of the names of the node sets containing the retained nodes of the tying relation.

**Remarks:**

- Multipoint constraints will be defined by means of the user routine **UTIE**.
- When slave and master nodes are specified by sets, these sets must be in one-to-one correspondence, that is, the i-th items of all master sets will be associated with the i-th item of the slave set into the multipoint constraint relationship. Furthermore, the third line of the command syntax is also the last one, irrespective of the number of MPCs to be defined; this is true because the number of nodes of each set coincides with the number of MPCs.
- A slave node can be in relationship with one or more master nodes (i.e., one or more degrees of freedom of a slave node can be constrained to one or more degrees of freedom of one or more master nodes); viceversa, multipoint constraints which link a master node to several slave nodes is not allowed.
- A node which is master in a multipoint constraint relationship cannot become a slave node into another MPC;

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 2*

*setname 3 4 45379*

*shell section 3*

*tie 420 2*

*end*

*coordinates*

.....

*define node set SLAVE*

*1 to 420*

*define node set MASTER1*

*421 to 840*

*define node set MASTER2*

*841 to 1260*

***tying***

***420***

***1 SLAVE MASTER1 MASTER2***

*end option*

.....

In the above example, multipoint constraints have been defined by specifying slave and master nodes via sets; the number of items in each node set is equal to the number of MPCs defined (i.e., 420). One or more degrees of freedom of a slave node is constrained to one or more degrees of freedom of two master nodes; the constrained degrees of freedom are specified into the user routine UTIE.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> MPCs*

**See also:**

- Control Section: TIE
- SALOME Nosa User's Guide: Define Multi Point Constraints
- User Subroutines Reference Guide: UTIE

## 4.5 Load Section

Below a list of keywords ordered as occurring into the input file.

- AUTO LOAD
- BOUNDARY CHANGE
- COMMENT
- CONTROL
- DISTRIBUTED FLUXES
- DISTRIBUTED LOADS
- ELPRINT
- END INCREMENT
- FIXED ACCELERATION

- FIXED DISPLACEMENT
- FIXED TEMPERATURE
- NODPRINT
- POINT FLUXES
- POINT LOADS
- POST
- PRINT CHOICE
- PROPORTIONAL INCREMENT
- SAVE INCREMENT
- STEP SIZE
- SURFACE CHANGE
- THERMAL LOADS
- TIME STEP
- TYING CHANGE

### 4.5.1 AUTO LOAD

The keyword is used to define the number of automatic load or heat flux increments.

**Command syntax** :

*auto load*

*N* , *mod*

where

- *N* is the number of times the load increment has to be applied;
- *mod* is an integer specifying how subsequent load increments have to be applied; possible values are:
  - *mod = 0* means that subsequent load increments are equal to the first load increment of the loading sequence;
  - *mod = 1* means that each load increment of the loading sequence will be defined by means of the user routines **FORCEM** and **UPLOAD**.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 2*

*setname 3 4 45379*

*shell section 3*

*end*

.....

*distributed loads*

*1*

*1* 1

*0.0 0.0 -100.0*

*GRUPPO1*

*point loads*

*1*

*0.0000 0.0000 -1200.0 0.0000 0.0000 0.0000*

*GRUPPO2*

***auto load***

***10 0***

*end increment*

In this example the distributed body force ( i.e. load type 1 according to **Nosa Load Library** ) will be applied ten times.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Steps* : ***Load incrementation*** *tab ->Automatic load/heat flux* increment

**See also:**

- SALOME Nosa User's Guide: Define Load-Steps
- User Subroutines Reference Guide: FORCEM
- User Subroutines Reference Guide: UPLOAD

### 4.5.2 BOUNDARY CHANGE

This command is used to redefine boundary conditions.

**Command syntax** :

*boundary change*

*N*

```
(The following lines must be repeated N times)
```

*presc*

*ifpre*

*nset*

where

- *N* is the number of boundary conditions to be redefined;
- *presc* is a list of values of the degrees of freedom to be constrained relevant to a given boundary condition;

- *ifpre* is a list of the degrees of freedom to be constrained relevant to a given boundary condition;
- *nset* is list of constrained nodes relevant to a given boundary condition.

**Remarks:**

1. the usage of this keyword implies a completely redefinition of a boundary condition; other boundary conditions which do not change with respect to the previous load increment must be repeated as well.
2. the keyword applies to boundary conditions on displacement or temperature fields.
3. if a unconstrained degree of freedom is now constrained or viceversa, the analysis requires a new linear-equations system for the equilibrium solution; thus, the CONTROL keyword must be used, and the Newton-Raphson algorithm must be selected.

**Input file usage**

*title Example 1*

*.....*

*end*

*.....*

*boundary conditions*

*2*

*0 0 0 0*

*1 3 5 6*

*base1*

*0 0*

*2 4*

*base2*

*.....*

*control*

*1000 1000 3*

*0.10 1.00e-08*

*.....*

*end option*

*distributed loads*

*1*

*1 1*

*0.0 0.0 -100.0*

*GRUPPO1*

*end increment*

*distributed loads*

*1*

*1 1*

*0.0 0.0 0.0*

*GRUPPO1*

*point loads*

*1*

*0.0000 0.0000 -1200.0 0.0000 0.0000 0.0000*

*GRUPPO2*

***boundary change***

***2***

***0 0 0 0***

***1 3 5 6***

***base1***

***0 0***

***1 3***

***base2***

*end increment*

In this example two boundary conditions on displacement field have been defined; the first one applies to all nodes belonging to the set "base1", and assigns null values to displacements along the x and z-axes and rotations around the y and z axes. The other boundary conditions is relevant to nodes of the set "base2", for which null values to displacement along the y axis and rotation around the x axis have been imposed. These boundary conditions are relative to the first load increment; in the next load increment only the boundary condition relevant to the set "base2" has been changed. However, the boundary condition applied to the set base1 must be repeated.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Boundary Conditions: Boundary Condition Type:* ***Static****, Boundary Condition Family:* ***Displacement*** *or* ***Temperature***

**See also:**

- Model Section: CONTROL
- CONTROL
- SALOME Nosa User's Guide: Boundary Coditions

### 4.5.3 COMMENT

This keyword is used to insert a comment; the user can insert a comment anywhere.

**Command syntax** :

*comment "comment text"*

or

*$ "comment text"*

**Input file usage**

*title Example 1*

.....

*end*

*connectivity*

*10*

*1 10 67 92 37 45*

*2 10 29 31 47 89*

***comment this example explains the use of NOSA card***

.....

*end option*

.....

### 4.5.4 CONTROL

The keyword is used to set the parameters for controlling the accuracy of numerical analyses, to choose algorithms to be used to solve equilibrium equations and setting other solution controls.

**Command syntax** :

*control*

*maxinc miter algoID irsdis*

*toler fuztol fuzdis fuztmp tmpfin*

where

- *maxinc* is the maximum number of load increments; by default, *maxinc* is equal to 1000;
- *miter* is the maximum number of iterations for each load increment; the default value is 1000;
- *algoID* is the identifier of the algorithm to be used to solve equilibrium equations; possible values are:
  - *algoID = 1* : The **initial stiffness matrix method** is used;
  - *algoID = 2* : The **modified Newton-Raphson method** is used;
  - *algoID = 3* : The **Newton-Raphson method** is used; by default, this method is applied to solve equilibrium equations.
- *irsdis* is a flag to set convergence criteria based on displacement field; when *irsdis* is not null, the displacement based convergence check is turned on in dynamic analyses;
- *toler* is a ratio in % between the norm of residual force and the norm of total force (including reactions), representing the tolerance used for the convergence of a load increment. For shell and beam elements, the moments are divided by the corresponding thickness to be dimensionally homogeneous to forces. If the convergence check is to be done on displacement variation, it is the ratio between the iteration displacement change and the incremental displacement variation; in such a case, at least two iterations are required in order that an increment can achieve the convergence;
- *fuztol* is the minimum value of the norm of total force; if this norm is less than *fuztol*, the convergence control is skipped (default 1.0D-08) in static analyses, or the displacement based convergence check is automatically turned on in dynamic runs;

- *fuzdis* is the minimum displacements change for dynamic analyses; if the norm of displacement is less than *fuzdis*, the convergence control is skipped (default 1.0D-12);
- *fuztmp* is the minimum temperature change for heat transfer analyses; if the maximum temperature change is less than *fuztmp*, the convergence control is skipped (default 1.0D-08);
- *tmpfin* is the maximum temperature change allowed in automatic time stepping mode (default 20.D0), for heat transfer analyses; if the maximum temperature change is greater than *tmpfin*, the increment is repeated with reduced time step in order that the maximum temperature change is equal to *tmpfin* (this can be obtained with precision in linear cases).

**Input file usage**

*title Example 1*

*sizing 43228 45379*

.....

*end*

.....

***control***

***1000 1000 3***

***0.10 1.00e-08***

.....

*end option*

.....

In the above example the Newton-Raphson method will be used to solve equilibrium equations; a maximum number of 1000 load increments are allowed, for each of them a maximum number of 1000 iterations will be evaluated to achieve the numerical convergence; the latter is achieved within a residual of nodal force equal to 0.1% of the total force. All other parameters have been set to default values.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Solution Controls*

**See also:**

- NOSA Theory Manual: Solution Controls,
- SALOME Nosa User's Guide: Setting Solution Controls

### 4.5.5 DISTRIBUTED FLUXES

The command is used to define distributed fluxes on faces or edge of mesh elements.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*distributed fluxes*

*N*

```
(the following lines must be repeated N times)
```

*loadID*, *sequenceID*

*hf1, hf2,...,hfn*

*eset*

where

- *N* is the number of heat fluxes to be defined;
- *loadID* is the identifier code of the flux type (see the library of elements);
- *sequenceID* is the order number of the flux, as specified into the sequence of fluxes to be applied;
- *hf1*, *hf2*, ..., *hfn* are the values of the heat flux applied on the element (see the library of elements);
- *eset* is a list of element which the flux is applied to.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 11*

*materials 2*

*setname 3 4 45379*

*heat transfer 2*

*end*

.....

***distributed fluxes***

***2***

***1 1***

***2.0***

***GRUPPO1***

***1 2***

***3.0***

***GRUPPO2***

*end increment*

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Loads: Nosa Load Library:* ***Heat Transfer*** *, Load Type:* ***Distributed*** *, Load Family:* ***heat flux on edge/surface***

**See also:**

- NOSA Theory Manual: Load Library
- SALOME Nosa User's Guide: Define Loads

### 4.5.6 DISTRIBUTED LOADS

This command defines distributed loads on faces or edge of elements in the global or local reference system.

**Command syntax** :

*distributed loads*

*N*

```
(the following lines must be repeated N times)
```

*loadID*, *sequenceID*

*F1*, *F2*, ..., *F6*

*eset*

where

- *N* is the number of distribute loads to be defined;
- *loadID* is the identifier code of the load type, according to the **Nosa Load Library**;
- *sequenceID* is the order number of the load, as specified into the sequence of loads to be applied;
- *F1*, *F2*, ... are values of the load components, expressed in a local or the global reference system (see the Nosa Load Library);
- *eset* is the group of elements which the load is applied to.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 2*

*setname 3 4 45379*

*dist loads 3*

*end*

.....

***distributed loads***

***3***

***2 1***

***0.0 0.0 1.0***

***GRUPPO1***

***51 2***

***0.0 -20.0 0.0***

***GRUPPO2***

***23 3***

***0.0 -20.0 0.0***

***GRUPPO3***

.....

*end increment*

In this example three loads have been assigned: the first one is a distributed body force, whose components are determined by means of the user routine **FORCEM**, and it is applied to the set "GRUPPO1"; the second applied load (as specified by its order number) is a pressure on the surface of elements belonging to the set "GRUPPO2". The last load is a pressure on the 2-3 edge of elements constituting the set "GRUPPO3".

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Loads: Nosa Load Library:* ***Static/Dynamic*** *, Load Type:* ***Distributed*** *, Load Family: select from the list of load types*

**See also:**

- NOSA Theory Manual: Load Library
- SALOME Nosa User's Guide: Define Loads
- User Subroutines Reference Guide: FORCEM

### 4.5.7 ELPRINT

This command is used to specify element sets for which field ouput (e.g. stress, strain fields and so on) must be also written to the prt file (i.e. the output file with extension ".prt").

**Command syntax** :

*elprint*

*N*

*eset-1*

*eset-2*

.....

*eset-N*

where

- *N* is the number of element groups, whose output data are to be written to the prt file;
- *eset-n* is the n-th element set.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 2*

*setname 3 4 45379*

*dist loads 6*

*end*

.....

***elprint***

***1***

***ALL_ELEM***

.....

*end increment*

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Steps* : ***Print Frequency*** *tab*

**See also:**

- PRINT CHOICE
- SALOME Nosa User's Guide: Define Load-Steps

### 4.5.8 END INCREMENT

This command ends an incremental load.

**Command syntax** :

*end increment*

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 2*

*setname 3 4 45379*

*dist loads 2*

*end*

.....

*distributed loads*

*1*

*1 1*

*0.0 0.0 100.0*

*gruppo1*

***end increment***

*distributed* loads

*1*

*1 1*

*0.0 0.0 100.0*

*gruppo2*

***end increment***

**Remarks:**

- This keyword **is mandatory** and must be used at the end of the definition of each load increment.

### 4.5.9 FIXED ACCELERATION

This command defines acceleration boundary conditions for dynamic analyses.

**Command syntax** :

*fixed acceleration*

*N*

*prsca-1*

*iprea-1*

*set-1*

*prsca-2*

*iprea-2*

*set-2*

.....

*prsca-N*

*iprea-N*

*set-N*

where

- *N* is the number of boundary conditions to be defined;
- *prsca-n* is a list of values of the degrees of freedom to be constrained, relavant to the n-th boundary condition;
- *iprea-n* is a list of the degrees of freedom to be constrained, relevant to the n-th boundary condition;
- *set-n* is a list of constrained nodes relevant to the n-th boundary condition.

**Remarks:**

1. when the list of degrees of freedom contains negative values, the user routine **UBND** will be called to constrain the DOF with negative value of all nodes belonging to the given set.
2. the usage of this keyword implies a completely redefinition of a boundary condition; other boundary conditions which do not change with respect to the previous load increment must be repeated as well.
3. the values assigned to the degrees of freedom through this keyword are total accelerations; therefore, if a unconstrained degree of freedom is now constrained or viceversa, the analysis requires a new linear-equations system for the equilibrium solution. Thus, the CONTROL keyword must be used, and the Newton-Raphson algorithm must be selected.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*dynamic*

*shell section 3*

*end*

*....*

*dynamic*

*0.6 0.3*

*.....*

*end option*

*.....*

***fixed acceleration***

***2***

***0.3 0.1***

***1 2***

***27 28 33 45 120***

***0.9 0.7***

***1 -3***

***GRUPPO1***

*.....*

*end increment*

In the above example two boundary conditions have been assigned. For the first boundary condition finite values have been imposed to the translational accelerations along x and y axes to nodes into the given list; in the second case finite value to the translational acceleration along x-axis have been imposed to all nodes of the set *GRUPPO1*. With reference to the translational acceleration along the z-axis, its value has been defined by means of the user routine UBND. These boundary conditions are relative to the first load increment; in the next load increment only the boundary condition relevant to the set "GRUPPO1" has been changed. However, the other boundary condition which is not changed must be repeated.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Boundary Conditions: Boundary Condition Type:* ***Dynamic****, Boundary Condition Family:* ***Acceleration***

**See also:**

- Model Section: FIXED ACCELERATION
- Model Section: CONTROL
- CONTROL
- DYNAMIC
- SALOME Nosa User's Guide: Boundary Coditions
- User Subroutines Reference Guide: UBND

### 4.5.10 FIXED DISPLACEMENT

This command defines displacement boundary conditions.

**Command syntax** :

*fixed displacement*

*N*

*presc-1*

*ifpre-1*

*set-1*

*presc-2*

*ifpre-2*

*set-2*

.....

*presc-N*

*ifpre-N*

*set-N*

where

- *N* is the number of boundary conditions to be defined;
- *presc-n* is a list of values of the degrees of freedom to be constrained, relavant to the n-th boundary condition;
- *ifpre-n* is a list of the degrees of freedom to be constrained, relevant to the n-th boundary condition;
- *set-n* is a list of constrained nodes relevant to the n-th boundary condition.

**Remarks:**

1. when the list of degrees of freedom contains negative values, the user routine **UBND** will be called to constrain the DOF with negative value of all nodes belonging to the given set.
2. for static analyses displacement boundary conditions have to be defined in order to avoid rigid body motions.
3. the usage of this keyword implies a completely redefinition of a boundary condition; other boundary conditions which do not change with respect to the previous load increment must be repeated as well.
4. the values assigned to the degrees of freedom through this keyword are displacements increments; therefore, if a unconstrained degree of freedom is now constrained or viceversa, the analysis requires a new linear-equations system for the equilibrium solution. Thus, the CONTROL keyword must be used, and the Newton-Raphson algorithm must be selected.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*shell section 3*

*end*

.....

*end option*

.....

***fixed displacement***

***2***

***0.0 0.0***

***1 2***

***27 28 33 45 120***

***0.0 0.0***

***1 2***

***GRUPPO1***

.....

*end increment*

In the above example two boundary conditions have been assigned. For the first boundary condition null values have been imposed to the translation along x and y axes for nodes into the given list; in the second case null value to the translation along z-axis has been imposed to all nodes of the set *GRUPPO1*. With reference to the rotation around the x-axis, the constraint has been defined by means of the user routine UBND. These boundary conditions are relative to the first load increment; in the next load increment only the boundary condition relevant to the set "GRUPPO1" has been changed. However, the other boundary condition which is not changed must be repeated.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Boundary Conditions: Boundary Condition Type:* ***Static****, Boundary Condition Family:* ***Displacement***

**See also:**

- Model Section: FIXED DISPLACEMENT
- Model Section: CONTROL
- CONTROL
- SALOME Nosa User's Guide: Boundary Coditions
- User Subroutines Reference Guide: UBND

### 4.5.11 FIXED TEMPERATURE

This command defines temperature boundary conditions for heat transfer analyses.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*fixed temperature*

*N*

*prsct-1*

*ipret-1*

*set-1*

*prsct-2*

*ipret-2*

*set-2*

.....

*prsct-N*

*ipret-N*

*set-N*

where

- *N* is the number of boundary conditions to be defined;
- *prsct-n* is a list of values of the degrees of freedom to be constrained, relavant to the n-th boundary condition;
- *ipret-n* is a list of the degrees of freedom to be constrained, relevant to the n-th boundary condition;
- *set-n* is a list of constrained nodes relevant to the n-th boundary condition.

**Remarks:**

1. When the list of degrees of freedom contains negative values, the user routine **UBND** will be called to constrain the DOF with negative value of all nodes belonging to the given set;
2. the usage of this keyword implies a completely redefinition of a boundary condition; other boundary conditions which do not change with respect to the previous load increment must be repeated as well.
3. the values assigned to the degrees of freedom through this keyword are temperature increments; therefore, if a unconstrained degree of freedom is now constrained or viceversa, the analysis requires a new linear-equations system for the equilibrium solution. Thus, the CONTROL keyword must be used, and the Newton-Raphson algorithm must be selected.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*shell section 3*

*end*

.....

*end option*

.....

***fixed temperature***

***2***

***10.0 20.0***

***1 2***

***27 28 33 45 120***

***11.0 -9.0***

***1 2***

***GRUPPO1***

*end increment*

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Boundary Conditions: Boundary Condition Type:* ***Heat Transfer****, Boundary Condition Family:* ***Temperature***

**See also:**

- Model Section: FIXED TEMPERATURE
- Model Section: CONTROL
- CONTROL
- HEAT TRANSFER
- SALOME Nosa User's Guide: Boundary Coditions
- User Subroutines Reference Guide: UBND

### 4.5.12 NODPRINT

This command is used to specify node sets for which field ouput (e.g. displacement field, nodal reactions and so on) must be also written to the prt file (i.e. the output file with extension ".prt").

**Command syntax** :

*nodprint*

*N*

*nset-1*

*nset-2*

.....

*nset-N*

where

- *N* is the number of node groups, whose output data are to be written to the prt file;
- *nset-n* is the n-th node set.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 2*

*setname 3 4 45379*

*dist loads 6*

*end*

.....

***nodprint***

***1***

***ALL_NODE***

.....

*end increment*

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Steps* : ***Print Frequency*** *tab*

**See also:**

- PRINT CHOICE
- SALOME Nosa User's Guide: Define Load-Steps

### 4.5.13 POINT FLUXES

This command defines concentrated heat fluxes applied on the nodes.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*point fluxes*

*N*

*flux-1*

*nset-1*

*flux-2*

*nset-2*

...

*flux-N*

*nset-N*

where

- *N* is the number of point fluxes to be defined;
- *flux-n* is a list of values of the n-th concentrated heat flux;
- *nset-n* is the list of nodes to which the n-th flux is applied.

**Remarks:**

- In any case the user routine **UPLOAD** will be called at the end of the load definition section; thus, it is recommended to check that this routine does not introduce any spurious point flux.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 11*

*materials 2*

*setname 3 4 45379*

*heat transfer 2*

*end*

*.....*

***point fluxes***

***2***

***2.0***

***GRUPPO1***

***3.0***

***GRUPPO2***

*end increment*

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Loads: Nosa Load Library:* ***Heat Transfer*** *, Load Type:* ***Concentrated*** *, Load Family:* ***concentrated heat flux***

**See also:**

- NOSA Theory Manual: Load Library
- SALOME Nosa User's Guide: Define Loads
- User Subroutines Reference Guide: UPLOAD

### 4.5.14 POINT LOADS

This command defines concentrated loads applied on the nodes of the structure in the global reference system.

**Command syntax** :

*point loads*

*N*

*F1*

*nset-1*

*F2*

*nset-2*

*...*

*FN*

*nset-N*

where

- *N* is the number of point loads to be defined;
- *Fn* is a list of values of the n-th concentrated point load;
- *nset-n* is the list of nodes to which the n-th load is applied.

**Remarks:**

- In any case the user routine **UPLOAD** will be called at the end of the load definition section; thus, it is recommended to check that this routine does not introduce any spurious point load.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 11*

*materials 2*

*setname 3 4 45379*

*end*

.....

***point loads***

***2***

***2.0 1.0 3.0 0.0 0.0 0.0***

***GRUPPO1***

***6.0 0.0 0.0 0.0 3.0 0.0***

***GRUPPO2***

*end increment*

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Loads: Nosa Load Library:* ***Static/Dynamic*** *, Load Type:* ***Concentrated*** *, Load Family:* ***concentrated force/moment***

**See also:**

- NOSA Theory Manual: Load Library
- SALOME Nosa User's Guide: Define Loads
- User Subroutines Reference Guide: UPLOAD

### 4.5.15 POST

This command redefines the frequency with which output variables will be written to the post-processing file.

**Command syntax** :

*post*

*freq*, *iterFreq*, *icont*, *wrtim*

where

- *freq* is the frequency with which output variables will be written to the post-processing file at the end of a load increment. If *freq* is not null, the output will be written at the end of every *freq-th* load increment, after the last written. If *freq = 0* no data will be written to the output file;
- *iterFreq* is the frequency with which output variables will be written to the output file at several iterations of each load increment; by default, *iterFreq = 0*, which means no iteration result will be written;
- *icont* is an integer number useful if a restart analysis is required; in such a case:
  - *icont = 0* means that the initial block of the output file relevant to element connectivity and nodal coordinates will not be printed;
  - *icont = 1* means that the initial block of the output file will be printed;
- *wrtim* has to be set to its default value (i.e., equal to 0.0);

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*shell section 3*

*dist loads 10*

*end*

.....

*post*

*8 1 0 0 0 0*

*51 N11*

*52 N22*

*53 N12*

*54 Q23*

*55 Q13*

*56 M11*

*57 M22*

*58 N12*

*end option*

*distributed loads*

*1*

*2 1*

*0.0 0.0 1.0*

*GRUPPO1*

*auto load*

*11 0*

***post 5 0 0 0***

.....

*end increment*

In this example a distributed load will be applied to the elements of the set "GRUPPO1" eleven times; the frequency with which the output variables will be written on post file is no longer 1 but 5. This means that output data will not be written for each load increment, but only those relevant to the first, 6-th and 11-th load increments will be stored into the post file.

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Steps* : ***Output Frequency*** *tab*

**See also:**

- Model Section: POST
- SALOME Nosa User's Guide: Define Load-Steps

### 4.5.16 PRINT CHOICE

This keyword is used to define the frequency with which output data will be written to the prt file.

**Command syntax** :

*print choice*

*prtFreq*

where

- *prtFreq* is an integer specifying the output frequency; possible values are:
  - *prtFreq* = *0* means no data will be written to the prt file;
  - *prtFreq* = *1* means that output data will be written once convergence has been achieved (this is the default value);
  - *prtFreq* = *2* means that output data will be written for every iteration.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 2*

*setname 3 4 45379*

*dist loads 1*

*end*

.....

*end option*

***print choice***

***0***

*distributed loads*

*1*

*1* 1

*0.0 0.0 1.0*

*end increment*

**NOSA-ITACA/GUI usage**

*NOSA-ITACA/GUI -> Nosa module -> Nosa menu -> Steps* : ***Print Frequency*** *tab*

**See also:**

- ELPRINT
- NODPRINT
- SALOME Nosa User's Guide: Define Load-Steps

### 4.5.17 PROPORTIONAL INCREMENT

This command defines the proportionality factor between the next load and the initial load for analyses containing more than one load increment.

**Command syntax** :

*proportional increment*

*facto*

where

- *facto* is a real multiplicative factor of the last defined increment load.

**Remarks:**

- a defined load is a load determined by the following keywords:
  - DISTRIBUTED LOADS;
  - POINT LOADS;
  - FIXED ACCELERATION;
  - FIXED DISPLACEMENT;
  - BOUNDARY CONDITIONS;
  - BOUNDARY CHANGE;
  - SURFACE;
  - SURFACE CHANGE.

### 4.5.18 SAVE INCREMENT

This command is used to save the incremental results on a suitable file (unit FORTRAN n.99).

**Command syntax** :

*save increment*

*ISAVE*

where

- *ISAVE* is a parameter greater or equal to 0; when it is equal to 0 the incremental results are unsaved. If positive, the results will be saved every ISAVE increments.

**Remarks:**

Since the fil is overwritten, it will contain only data from the last increment saved.

**Input file usage**

*title Example 1*

*sizing 543228 45379*

*element 10*

*materials 2*

*setname 3 4 45379*

*dist loads 6*

*end*

*coordinates*

.....

*connectivity*

.....

.....

*end option*

*print choice*

*0*

*distributed loads*

*6*

*1 1*

*0.0 0.0 1.0*

.....

***save increment***

***10***

*end increment*

**See also:**

RESTART

### 4.5.19 STEPSIZE

This command is used to define magnitude and number of steps in heat transfer or dynamic analysis.

**Command syntax** :

*stepsize*

*AUTTIM NUMSTP DELTII PERIOD IACCI*

where

- *AUTTIM*, if different from zero sets the automatic calculation of step size in heat transfer analysis;
- *NUMSTP*, maximum number of the steps in this series;
- *DELTII*, initial time step;
- *PERIOD*, time interval to be covered by this series of steps;
- *IACCI*, if different from zero requires the calculation of the initial acceleration in dynamic analysis.

**Remarks:**

In case of heat transfer analysis with automatic step size calculation, denoting by DTMAX the absolute value of the maximum temperature change we have to compare it with the maximum temperature change allowed TMPFIN. If DTMAX>TMPFIN, the step will be repeated with a smaller step size DELTIM=DELTIM∗TMPFIN/DTMAX, whereas if DTMAX<TMPFIN the step size for the next step will be increased to DELTIM=0.9∗TMPFIN/DTMAX.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 1*

*setname 3 2 45379*

*dynamic*

*shell section 3*

*end*

*coordinates*

*.....*

*connectivity*

*.....*

*define node set GRUPPO1*

*.....*

*dynamic*

*0.6 0.3*

*fixed acceleration*

*2*

*0.3 0.1*

*1 2*

*27 28 33 45 120*

*0.1 0.1*

*1 3*

*GRUPPO1*

*.....*

*end option*

***stepsize***

***0 100 10 1***

*.....*

*distributed loads*

*.....*

*end increment*

### 4.5.20 SURFACE CHANGE

This command defines the velocity of rigid surfaces used in 2D contact problems.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*surface change*

*NSURF*

`(the following lines must be repeated NSURF times)`

*INDSUR SURF(3,1) SURF(4,1) SURF(5,1)*

where

- *NSURF*, number of rigid surfaces whose velocity must be redefined;
- *INDSUR*, ordinal number of the rigid surface;
- *SURF(3,1)*, first velocity component of the rigid surface's guide node velocity;
- *SURF(4,1)*, second velocity component of the rigid surface's guide node velocity;
- *SURF(5,1)*, Angular velocity of the rigid surface's guide node (rad/sec, positive if counterclockwise).

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 3*

*materials 1*

*setname 2 1 45379*

*contact 2 2*

*end*

.....

*surface*

*2*

*2 5.0 7.0 1.0 0.5 0.01*

*1 -3.0 -2.0 5.0 7.0*

*2 3.0 2.0 0.0 0.0 60*

*2 1.0 0.0 3.0 0.7 0.02*

*1 -5.0 -4.0 3.0 1.0*

*2 3.0 2.0 0.0 0.0 60*

*1 -3.0 -2.0 5.0 7.0*

.....

*end option*

.....

***surface change***

***1***

***2 3.0 2.0 0.07***

.....

*end increment*

### 4.5.21 THERMAL LOADS

This command defines the nodal increments of temperature for calculating loads due to thermal dilatation.

**Command syntax** :

*thermal loads*

*IRDTMP*

```
(the following lines must be used only when IRDTMP is equal to 1)
```

*NDIST*

```
(the following lines must be repeated NDIST times)
```

*TEMPE*

*NSET*

```
(the following lines must be used only when IRDTMP>1)
```

*INCTMP NAUTO*

where

- *IRDTMP:* if equal to 1, nodal temperature values are to be read from cards; if equal to 2, nodal temperature values are to be read from binary post file (Fortran unit 26); if equal to 3 nodal temperature value are to be read from formatted post file; if equal to 4 nodal temperature values are to be

calculated by the user routine **URDTEM**;

- *NDIST* is the number of node groups to which the temperature increments are applied;
- *TEMPE* are the values of the temperature increments applied to the node group;
- *NSET* is the group of nodes subjected to the temperature increment;
- *INCTMP* is the number increment of the post file from which temperature data are to be read;
- *NAUTO* is number of thermal load steps necessary to achieve the thermal state read from increment INCTMP.

**Remarks:**

For shell elements, temperature data are to be indicated at the bottom and top layer when ITYRD=2 and at the bottom, middle and top layer when ITYRD=3.

**Input file usage**

*title Example 1*

*sizing 43228 45379*

*element 10*

*materials 2*

*setname 3 4 45379*

*dist loads 6*

*end*

.....

***thermal loads***

***1***

***1***

***25.0 10.0 30.0***

***GRUPPO1***

.....

*end increment*

**See also:**

- User Subroutines Reference Guide: URDTEM

### 4.5.22 TIME STEP

This command defines time increments for calculating the motion of rigid surfaces in two-dimensional contact problems.

**This keyword is not available in the present version of NOSA-ITACA.**

**Command syntax** :

*time step*

*iusurf deltii*

where

- *iusurf* is a parameter which if different from zero the user routine **UMOTIO** is used for defining the rigid surface's velocity;
- *deltii* is the time increment.

**Input file usage**

*title Example 1*

.....

***time step***

***0 2***

.....

*end increment*

### 4.5.23 TYING CHANGE

This keyword is used to re-define multipoint constraints (MPCs).

**Command syntax** :

*tying change*

*ngtie*

The following commands must be repeated NGTIE times. *ITIE(1) ITIE(2) ITIE(3.....N)*

where

- *ITIE(1)*, identifier of the tying relation;
- *ITIE(2)*, identifier number of the tied node of the tying relation or the name of the node set containing the tied nodes of the tying relation;
- *ITIE(3.....N)*, identifier numbers of the retained nodes of the tying relation or the names of the node sets containing the retained nodes of the tying relation.

**Remarks:**

The management of the tying relations is carried out by the user routine UTIE.
If the tied and retained nodes are specified by sets, then these sets must be in one to one correspondence in the sense that the i-th items of the sets containing the retained nodes must be the identifiers of the retained nodes pertaining to the tied node identified by the i-th item of the set of tied nodes.
If the card tying change is present, the table of tying relations must be completely redefined, even for those parts which did not change with respect to the previous situation.

**See also:**

UTIE, ref tying_page "TYING"

# Chapter 5

# User Subroutines Reference Guide

## 5.1 Introduction

This guide describes all fortran routines which the user can use to define some analysis attributes, when they are not available in NOSA-ITACA/GUI or their definition is not suitable to be described through the input keywords. In particular, these routines allow the user to describe:

- work-hardening behaviour of elastic-plastic materials (isotropic, kinematic and mixed hardening rules);
- non-uniform loads;
- user-defined output variables (such as derived element outputs);
- user-defined reference system with respect to which output variables are written into the post file;
- variable boundary conditions;
- variable initial conditions;
- variable element cross-section thickness;
- multipoint constraints;

Once these routines have been written, the user has to compile and link them to **Nosa library** in order to perform the numerical analysis.

For each utility routine, this guide shows also one or more examples of usage.

## 5.2 Compiling and linking utility routines

By default, before running an analysis NOSA-ITACA software checks if utility routines have been modified since last run; if it is true, they are compiled and linked before running a job analysis.

## 5.3 Utility routines

All utility routines are listed below.

- CPHI
- CPSI, CPS1
- FORCEM
- PLOTV
- UBND
- UDSPI
- UFILM
- UFRI
- UGEOM
- ULAXIS

- UMOTIO
- UPLOAD
- UPMAS
- URDTEM
- UTEMPI
- UTIE
- UVELI

## 5.4 CPHI

The subroutine CPHI allows one to define the parameters which control the isotropic hardening for elastic-plastic materials; more precisely the radius $\rho$ of the elastic range and its derivative $\dot{\rho}$ with respect to the Odqvist parameter $\zeta$ are assigned (the Odqvist parameter is defined as the time integral of the norm of the plastic deformation rate). In general, the dependence on the Odqvist parameter is specified through a table containing points belonging to an experimental hardening curve. This routine is called by the subroutines for integration of the constitutive law.

**This subroutine is not available in the presence version of NOSA-ITACA.**

```
SUBROUTINE CPHI (PHI, PHI1, ZETA, NHARD, HAISO, HARD)
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'LOCAL'
C
DIMENSION HARD (2, MHARD)
...
user code
...
RETURN
END
```

Arguments of the call:

- *PHI:* Radius of the elastic range $\rho(\zeta)$ (output).
- *PHI1:* Derivative of the radius $\rho(\zeta)$ with respect to $\zeta$ (output).
- *ZETA:* Odqvist parameter $\zeta$ (input).
- *NHARD:* Number of data pairs in the table HARD, which defines the hardening curve (input).
- *HAISO:* Percent of isotropic hardening present in the hardening curve (input).
- *HARD:* Table which defines the hardening curve (input), where:
  - HARD(1, I) is the accumulated equivalent plastic strain;
  - HARD(2, I) is the Von Mises equivalent stress.
- *MHARD:* Max number of data pairs in a hardening table (input from COMMON/CNTR/).

*Example of usage*

Let us suppose that the function $\sigma(\zeta)$ is given by a monotonic increasing piecewise linear curve with *NHARD* points $(\zeta_i, \sigma(\zeta_i))$, with $\zeta_1 = 0$ and $\sigma(\zeta_1) = \sigma_0$ the initial yield stress. *HAISO* is the percent of the isotropic hardening. In this case we have the following routine:

```
      subroutine cphi(phi,phi1,zeta,nhard,haiso,hard)
c**********************************************************************
c
c     compute function phi,phi1 for isotropic hardening
c     piecewise linear load curve
c
c**********************************************************************
c
c****
c     version 23/10/2012
c****
c
      implicit real*8 (a-h,o-z)
      include 'cntr'
      include 'local'
      dimension hard(2,mhard)
      sqrt2=1.41421356d0
      phi=sqrt2
      phi1=0.d0
      if(nhard.le.1)return
      c32=dsqrt(1.5d0)
      do i=1,nhard-1
       zet1=hard(1,i)*c32
       zet2=hard(1,i+1)*c32
c
c**** beta(1) is in common/local/c
       beta(1)=haiso*(hard(2,i+1)-hard(2,i))/(zet2-zet1)
       ii=i
       if(zeta.ge.zet1.and.zeta.lt.zet2)goto 11
      enddo
c
c**** if zeta is greater than the last value in the table the values
c**** PHI and PHI1 are hels constant to the last value calculated.
c      ii=nhard-1
      zet1=hard(1,ii)*c32
 11   continue
      phi=sqrt2*(hard(2,ii)+beta(1)*(zeta-zet1))/hard(2,1)
      phi1=sqrt2*beta(1)/hard(2,1)
      return
      end
```

## 5.5 CPSI, CPS1

The CPSI and CPS1 routines allows definition of the tensor **M** which controls kinematic hardening for elastic-plastic materials; more precisely tensor **M** may depend on the Odqvist parameter $\zeta$, the plastic strain, the tensor orthogonal to the elastic range and, finally the center of the elastic range. Usually the dependence on the Odqvist parameter is specified by a table containing points belonging to an experimental hardening curve. This routine is called by those routines, that integrate the constitutive equation.CPS1 differs from CPSI because the tensors inside the routine are stored in the vectorized form.

**This subroutine is not available in the presence version of NOSA-ITACA.**

```
SUBROUTINE CPSI (ZETA, TC, TEP, TN, TM, NHARD, HAKIN, HARD)
C
```

```
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'LOCAL'
C
DIMENSION TC(3, 3), TEP(3, 3), TN(3, 3), TM(3, 3).
DIMENSION HARD (2, MHARD)
...
user code
...
RETURN
END

and

SUBROUTINE CPS1 (ZETA, TC, TEP, TN, TM, NHARD, HAKIN, HARD)
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'LOCAL'
C
DIMENSION TC(LSTR1), TEP(LSTR1), TN(LSTR1), TM(LSTR1).
DIMENSION HARD (2, MHARD)
...
user code
...
RETURN
END
```

Arguments of the call:

- *ZETA:* Odqvist parameter $\zeta$ (input).
- *TC:* Center of the elastic range (input).
- *TEP:* Plastic strain tensor (input).
- *TN:* Outward unit normal to the elastic range (input).
- *TM:* Tensor that controls the kinematic hardening (output).
- *NHARD:* Number of data pairs in the table HARD, which defines the hardening curve (input).
- *HAKIN:* Percent of kinematic hardening present in the hardening curve (input).
- *HARD:* Table defining the hardening curve (input), where:
  - HARD(1, I) is the accumulated equivalent plastic strain;
  - HARD(2, I) is the Von Mises equivalent stress.
- *MHARD:* Max number of data pairs in a hardening table (input from COMMON/CNTR/).
- *LSTR1:* Number od components of the tensors TC, TEP, TN and TM in vectorized form.

*Example of usage*

Let us suppose that the function $\sigma(\zeta)$ be given by a monotonic increasing piecewise linear curve with *NHARD* points $(\zeta_i, \sigma(\zeta_i))$, with $\zeta_1 = 0$ and $\sigma(\zeta_1) = \sigma_0$ the initial yield stress. *HAKIN* is the percent of the kinematic hardening. In this case we have the following routine:

```
      subroutine cpsi(zeta,tc,tep,tn,tm,nhard,hakin,hard)
c**********************************************************************
c
c     compute kinematic hardening tensor
c     piecewise linear load curve
c
c**********************************************************************
c
c****
c     version 23/10/2012
c****
c
      implicit real*8 (a-h,o-z)
      include 'cntr'
      include 'local'
      dimension tc(3,3),tep(3,3),tn(3,3),tm(3,3)
      dimension hard(2,mhard)
      sqrt2=1.41421356d0
      sqrt3=dsqrt(3.d0)
      const=yeld(1)/(2.d0*shear(1))
      c32=sqrt3/sqrt2
      tm(1:3,1:3)=0.d0
      if(nhard.le.1) return
      do i=1,nhard-1
       zet1=hard(1,i)*c32
       zet2=hard(1,i+1)*c32
       eta=hakin*(hard(2,i+1)-hard(2,i))/(sqrt3*(zet2-zet1))
       ii=i
       if(zeta.ge.zet1.and.zeta.lt.zet2)goto 11
      enddo
      ii=nhard-1
      zet1=hard(1,ii)*c32
      zet2=hard(1,ii+1)*c32
      eta=hakin*(hard(2,ii+1)-hard(2,ii))/(sqrt3*(zet2-zet1))
 11   continue
      psi=sqrt2*eta/(2.d0*shear(1))
      tm(1:3,1:3)=tn(1:3,1:3)*psi
      return
      end
```

## 5.6 FORCEM

The routine FORCEM allows definition of non-uniform distributed loads or fluxes. The library of elements contains the list of the identifier numbers to be declared in the cards DIST LOADS or DIST FLUXES, relative to the definition of loads through this routine. If required, the routine FORCEM is called at the beginning of each increment by the subroutine LOADIN that manages the load increments.

```
SUBROUTINE FORCEM ( LNODS, COORD, MATNO, PROPS, GEOM,
      * DELTAH, ZETSH, TDISP, IELEM, LOACOD, LOANUM, BFORCE,
      * NNODE, NINTEG )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'CONVRG'
INCLUDE 'FILES'
INCLUDE 'LOCAL'
INCLUDE 'MOTION'
C
DIMENSION
      * BFORCE (LDOFN, NNODE, NINTEG),
```

```
      * COORD (MCORD, NPOIN),
      * DELTAH (LDELTA, LGASP),
      * GEOM (MDT, MGEOM),
      * LNODS (MNODE),
      * MATNO (MTHICK),
      * PROPS (MPROP, NMATS),
      * TDISP (MDOFN,NPOIN),
      * ZETSH(LINT,LGASP)
...
user code
...
RETURN
END
```

Arguments of the call:

- *LNODS:* Numbers of nodes constituting the element (input).
- *COORD:* Coordinates of all points of the mesh (input).
- *MATNO:* Array containing the identifier numbers of the materials constituting the element (input). The material constituting the element is always unique, except for shell and beam elements where it may vary from layer to layer or from fiber to fiber.
- *PROPS:* Array of the properties of the materials present in the mesh (input).
- *GEOM:* Element thickness. For homogeneous shells and beams GEOM(1,I) is the thickness of the first layer or fiber at the I-th node and the other layers or fibers have the same thickness. For non-homogeneous shell GEOM(J, I) is the thickness of the J-th layer or fiber at the I-th node. (input).
- *DELTAH:* Shell or beam thickness. DELTAH(J, I) is the thickness of the J-th layer or fiber at the I-Gauss integration point (input).
- *ZETSH:* ZETSH(J, I) is the distance from the shell mean surface, or from the beam axis, of the J-th Simpson integration point at the I-th Gauss integration point.
- *TDISP:* Array of the total displacements or temperatures of all degrees of freedom in the mesh (input).
- *IELEM:* Ordinal number of the element (input).
- *LOACOD:* Code of the load or flux to be defined, as declared in the cards DIST LOADS or DIST FLUXES (input).
- *LOANUM:* Ordinal number of the load or flux to be defined, as declared in the cards DIST LOADS or DIST FLUXES (input).
- *BFORCE:* Magnitude of the load or flux to be defined (output). In the case of body forces or volumetric fluxes, BFORCE must be defined for all global degrees of freedom of all element nodes. In the case of pressures or surface fluxes, BFORCE must be defined for all degrees of freedom (global or local) of the nodes belonging to the face or the edge indicated by the load code in LOACOD.
- *NNODE:* Number of nodes for which BFORCE must be defined (input). In the case of body forces or volumetric fluxes, NNODE is the number of element nodes. In the case of pressures or surface fluxes, NNODE is the number of nodes of the loaded face or edge.
- *NINTEG:* Its value is 1 except for shell or beam elements, in which case it indicates the number of integration points along the thickness or across the section (input).
- *LDOFN:* Number of degrees of freedom of the element nodes (input from COMMON/LOCAL/).

- *LGASP:* Number of Gauss integration points of the element (input from COMMON/LOCAL/).
- *MCORD:* Maximum number of coordinates per node (input from COMMON/CNTR/).
- *NPOIN:* Total number of nodes in the mesh (input from COMMON/CNTR/).
- *MDT:* For shell or beam elements, it is the number of layers or fibers of the element; for the other types of elements, its value is 1 (input from COMMON/CNTR/).
- *LDELTA:* Same as MDT (input from COMMON/CNTR/).
- *MTHICK:* For shell or beam elements, it is the number of integration points on the cross section; for the other types of elements, its value is 1 (input from COMMON/CNTR/).
- *LINT:* Same as MTHICK (input from COMMON/CNTR/).
- *MNODE:* Maximum number of element nodes (input from COMMON/CNTR/).
- *MPROP:* Number of mechanical properties of a material (input from COMMON/CNTR/).
- *NMATS:* Number of different materials contained in the mesh (input from COMMON/CNTR/).

*Example of usage*

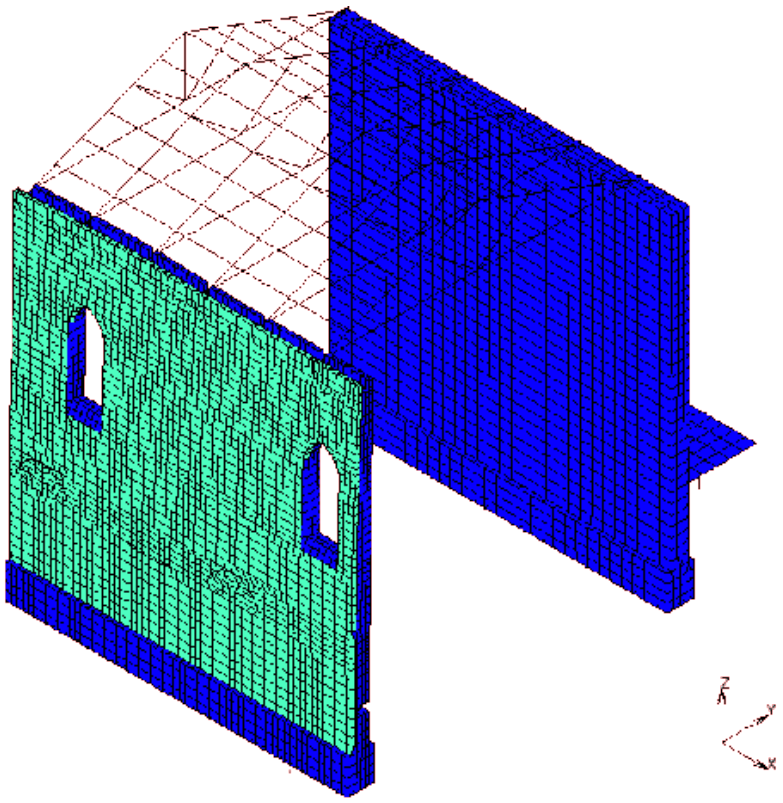


Figure 5.1: FEM model of a part of an ancient church.

In Figure 5.1 the FEM model of a part of an ancient church is shown. We want to study the effect of the wind acting in $y$ direction on the highlighted wall. The pressure of the wind is a function of the height $z$ following the formula:

$$p_y = \begin{cases} 673.0 & for\ z \leq 12.0 \\ 455. * 23^2 * log(\frac{z}{0.7}) * (7. + log(\frac{z}{0.7})) & for\ z > 12. \end{cases}$$

Bearing in mind that the face subjected to the wind is formed by the nodes 1, 5, 6, 2 of the elements, we have the following routine:

```
      subroutine forcem(lnods,coord,matno,props,geom,deltah,zetsh,tdisp,
     *ielem,loacod,loanum,bforce,nnode,nint)
       implicit real*8 (a-h,o-z)
      include 'cntr'
      include 'convrg'
      include 'files'
      include 'local'
      include 'motion'
      dimension
     *bforce(ldofn,nnode,nint),
     *coord(mcord,npoin),
     *deltah(ldelta,lgasp),
     *geom(mdt,mgeom),
     *lnods(mnode),
     *matno(mthick)
     *props(mprop,nmats),
     *tdisp(mdofn,npoin),
     *zetsh(lint,lgasp)
c
      dimension nodel(4)
      data nodel/1,5,6,2/
c
      do inode=1,nnod
       bforce(1:ldofn,inode,1)=0.d0
       ipoin=lnods(nodel(inode))
       z=coord(3,ipoin)
       if(z.le.12.d0)then
        bforce(2,inode,1)=673.d0
       else
        bforce(2,inode,1)=455.d0*.23d0**2*dlog(z/0.7d0)*
     *  (7.d0+dlog(z/0.7d0))
       endif
      enddo
      return
      end
```

## 5.7 PLOTV

The routine PLOTV allows the user to define quantities, on the basis of elemental data, to be written on the post-processing file. The routine PLOTV is called for each layer or fiber (shell and beam elements only) of each integration point of each element.

```
SUBROUTINE PLOTV ( VAR, T, E, EAN, EANC, TEMP, GPCOD,
      * DISGP, VELGP, ACCGP, IELEM, IGASP, ILAY, INDVA, LNODS,
      * COORD, TDISP,GRADT, IFBE, SECT, GEI, DELTAH, ZETSH,
      * GPBASE, CARSTR, CARSTN)
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'LOCAL'
```

```
C
DIMENSION
      * ACCGP(LCORD),
      * CARSTN(8),
      * CARSTR(8),
      * COORD(MCORD,NPOIN),
      * DELTAH(MBTHCK),
      * DISGP(LCORD)
      * E(6),
      * EAN(6),
      * EANC(6),
      * GEI(2, LGASP),
      * GPBASE(3,3),
      * GPCOD(LCORD),
      * GRADT(LCORD, MCASI, LGASP),
      * IFBE(MBTHCK),
      * LNODS(MNODE,NELEM),
      * SECT(LGASP),
      * TDISP(MDOFN,NPOIN),
      * T(6),
      * VELGP(LCORD),
      * ZETSH(MINT)

C
REAL VAR
C
...
user code
...
RETURN
END
```

Arguments of the call:

- *VAR:* value of the variable to be defined (output).
- *T:* array of the stress components in the order x-x, y-y, z-z, x-y, y-z, x-z (input).
- *E:* array of the total strain components (input).
- *EAN:* array of the total plastic strain or, for masonry-like materials, total crack strain(input).
- *EANC:* array of the total crushing strain for masonry-like materials with bounded compressive strength (input).
- *TEMP:* Value of the total temperature (input).
- *GPCOD:* Global coordinates of the integration point (input).
- *DISGP:* Displacement at the integration points (input).
- *VELGP:* Velocity at the integration points (input).
- *ACCGP:* Acceleration at the integration points (input).
- *IELEM:* Number of the element (input).
- *IGASP:* Number of the integration point (input).
- *ILAY:* Number of the layer or fiber(input).
- *INDVA:* Identifier number of the post processing variable to be defined (input).
- *LNODS:* Array of the connectivities of all elements (input).

- *COORD:* Array of the coordinates of all nodes (input).
- *TDISP:* Array of the total displacement for all degrees of freedom (input).
- *GRADT:* Values of the temperature gradient (input).
- *IFBE:* If IFBE(I) is zero it indicates that the I-th fiber of the current beam element is dummy (input).
- *SECT:* For beam elements is the area of the cross section at the current Gauss point (input).
- *GEI:* For beam elements is the array of the moment of inertia, around the first two local axes, of the cross section at the current Gauss point (input).
- *DELTAH:* Shell or beam thickness. DELTAH is the thickness array of the layers or fibers at the current Gauss integration point (input).
- *ZETSH:* ZETSH(I) is the distance from the shell mean surface, or from the beam axis, of the I-th Simpson integration point at the current Gauss integration point.
- *GPBASE:* The local base on the midsurface of the shell or on the beam axis at the current Gauss integration point.
- *CARSTR:* Stress resultants at the current Gauss integration point.
- *CARSTN:* Strain resultants at the current Gauss integration point.
- *LCORD:* Number of direction coordinates of the integration point (input from COMMON/LOCAL/).
- *LGASP:* Number of Gauss integration points for the current element (input from COMMON/LOCAL/).
- *MCORD:* Maximum number of direction coordinates (input from COMMON/CNTR/).
- *NPOIN:* Maximum number of nodes in the mesh (input from COMMON/CNTR/).
- *MCASI:* Number of cross section integration points for shell or beam elements (input from COMMON/CNTR/).
- *MINT:* Same as MCASI (input from COMMON/CNTR/).
- *MBTHCK:* Same as MCASI, for beam elements only (input from COMMON/CNTR/).
- *MNODE:* Maximum number of nodes in an element (input from COMMON/CNTR/).
- *NELEM:* Maximum number of elements in the mesh (input from COMMON/CNTR/).

*Example of usage*

In studying vaulted structures, we must consider that for shell and beam elements the elemental fields, such as stresses and strains, are expressed in the local reference system of the element, so their visualization can lead to "misleading" results due to inconsistency of different reference systems in different parts of the model. To avoid such a kind of problems, very often it is necessary to express the elemental quantities in a reference frame chosing by the user, so it is necessary using the subroutine PLOTV. Given a tensor $\mathbf{T}$, and $\{\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}\}$ and $\{\mathbf{f_1}, \mathbf{f_2}, \mathbf{f_3}\}$ two orthonormal bases such that

$$\mathbf{f_i} = \mathbf{Q}\mathbf{e_i}$$

with $\mathbf{Q}$ orthogonal tensor, by indicating with $[\mathbf{T}]$, $[\mathbf{T}]'$ the matrices of the components of $\mathbf{T}$ with respect to the basis $\{\mathbf{f_1}, \mathbf{f_2}, \mathbf{f_3}\}$ and $\{\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}\}$, respectively, it holds that

$$[\mathbf{T}]' = [\mathbf{Q}]^T[\mathbf{T}][\mathbf{Q}]$$

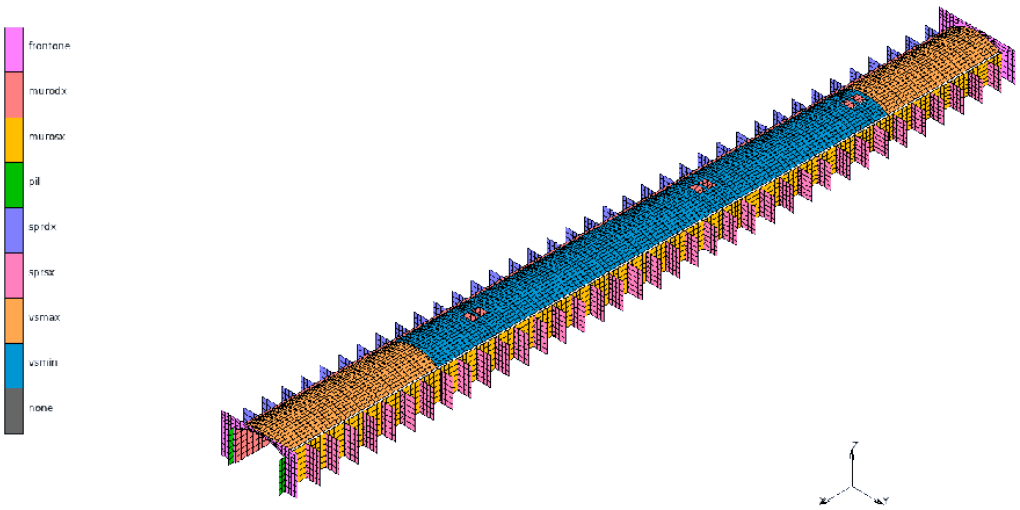



Figure 5.2: FEM model of the "Voltone" in Livorno, Italy

Figure 5.2 shows the mesh model of a structure made up of a vault which, by means of a series of tying constraints, is linked to two side walls, each of them reinforced by a set of transversal buttresses. The mesh model is built up of shell elements. We want to define a user reference system such that the third coordinate axis is parallel to the element normal unit vector; the first coordinate axis must be parallel to the global x axis for the vault and the side walls, and global y axis for the spurs. The second axis will be calculated by means of vector product.

In so doing, in the input file we will write the following instructions:

```
...
post
 36    1    0    0    0    0
-51  N11
-52  N22
-53  N12
-54  Q23
-55  Q13
-56  M11
-57  M22
-58  M12
-61  ECC11
-62  ECC22
-1011  s11 layer 1
-1012  s22 layer 1
-1014  s12 layer 1
-4011  s11 layer 4
-4012  s22 layer 4
-4014  s12 layer 4
-6011  s11 layer 6
-6012  s22 layer 6
-6014  s12 layer 6
-15  s23
-16  s13
-8011  s11 layer 8
-8012  s22 layer 8
-8014  s12 layer 8
-11011  s11 layer 11
-11012  s22 layer 11
-11014  s12 layer 11
-1021 ef11 layer 1
-1022 ef22 layer 1
```

```
-1024 ef12 layer 1
-6021 ef11 layer 6
-6022 ef22 layer 6
-6024 ef12 layer 6
-11021 ef11 layer 11
-11022 ef22 layer 11
-11024 ef12 layer 11
...
```

and the plotv subroutine will be:

```
      subroutine plotv(var,t,e,ean,eanc,ttemp,gpcod,disgp,
     *velgp,accgp,ielem,igasp,ilay,indva,lnods,coord,tdisp,gradt,
     *ifbe,sect,gei,deltah,zetsh,gpbase,carstr,carstn)
      IMPLICIT REAL*8 (A-H,O-Z)
      include 'cntr'
      include 'files'
      include 'local'
      dimension
     * accgp(lcord),
     * carstr(8),
     * carstn(8),
     * coord(mcord,npoin),
     * deltah(mbthck),
     * disgp(lcord),
     * e(6),
     * ean(6),
     * eanc(6),
     * gei(2,lgasp),
     * gpbase(3,3),
     * gpcod(lcord),
     * gradt(lcord,mcasi,lgasp),
     * ifbe(mbthck),
     * lnods(mnode,nelem),
     * sect(lgasp),
     * tdisp(mdofn,npoin),
     * t(6),
     * velgp(lcord),
     * zetsh(mint)
c
      real*4 var
      real*8 xmat(3,3),ubase(3,3),tens(3,3),sigma(3,3)
c
c***  xmat transformation matrix
c
c*** ubase user defined base
c
c*** tens tensor in local base
c
c*** sigma tensor in user base
c
c***  principal dimensions of the structure (m)
c
      fuz=0.01d0
      ic=0
      yv=6.765d0
      ym=7.365d0
      var=0.0
      end1=-220.d0
      end2=0.d0
      fuzend=0.01d0
      if(ktype.eq.10)then
c
c*** vault
       if(gpcod(2).le.yv.or.gpcod(2).ge.-yv)ic=1
c
```

```
c** +y side wall
       if(dabs(gpcod(2)-ym).le.fuz)ic=2
c
c*** -y side wall
       if(dabs(gpcod(2)+ym).le.fuz)ic=3
c
c*** spurs
       if(gpcod(2).le.-ym-fuz.or.gpcod(2).ge.ym+fuz)ic=4
c
c*** +x end wall
       if(dabs(gpcod(1)-end1).le.fuzend)ic=5
c
c*** x=0 end wall
       if(dabs(gpcod(1)-end2).le.fuzend)ic=6
c
       ind=iabs(indva)
       var=0.0
c
       defel: select case (ic)
c
c*** user base on the vault
c
        case (1) defel
         ubase(1:3,3)=gpbase(1:3,3)
         inode1=lnods(2,ielem)
         inode2=lnods(3,ielem)
         do i=1,3
          ubase(i,1)=coord(i,inode2)-coord(i,inode1)
         enddo
         unorm=dsqrt(ubase(1,1)**2+ubase(2,1)**2+ubase(3,1)**2)
         ubase(1:3,1)=ubase(1:3,1)/unorm
         call vprod(ubase(1,3),ubase(1,1),ubase(1,2))
c
c*** +y side wall
c
        case (2) defel
         ubase(1:3,1:3)=0.d0
         ubase(1,1)=1.d0
         ubase(3,2)=-1.d0
         ubase(2,3)=1.d0
c
c*** -y side wall
c
        case (3) defel
         ubase(1:3,1:3)=0.d0
         ubase(1,1)=1.d0
         ubase(3,2)=1.d0
         ubase(2,3)=-1.d0
c
c*** spursl
c
        case (4) defel
         ubase(1:3,1:3)=0.d0
         ubase(2,1)=1.d0
         ubase(3,2)=1.d0
         ubase(1,3)=1.d0
c
c*** +x end wall
c
        case (5) defel
         ubase(1:3,1:3)=0.d0
         ubase(2,1)=1.d0
         ubase(3,2)=-1.d0
         ubase(1,3)=-1.d0
c
c*** x=0 end wall
c
```

```
         case (6) defel
          ubase(1:3,1:3)=0.d0
          ubase(2,1)=1.d0
          ubase(3,2)=1.d0
          ubase(1,3)=1.d0
        end select defel
c
c*** setting up the transformation matrix
c
        do i=1,3
         do j=1,3
          xmat(j,i)=sprod(ubase(1,i),gpbase(1,j),3,3)
         enddo
        enddo
c
c*** stresses put into local tensor
        if(ind.le.16)then
         tens(1,1)=t(1)
         tens(1,2)=t(4)
         tens(1,3)=t(6)
         tens(2,1)=tens(1,2)
         tens(2,2)=t(2)
         tens(2,3)=t(5)
         tens(3,1)=tens(1,3)
         tens(3,2)=tens(2,3)
         tens(3,3)=0.d0
c
c*** tranformed stresses
c
         sigma(1:3,1:3)=0.d0
         do i=1,3
          do j=1,3
           do k=1,3
            do l=1,3
             sigma(j,i)=sigma(j,i)+xmat(j,k)*tens(k,l)*xmat(i,l)
            enddo
           enddo
          enddo
         enddo
         if(ind.eq.11)var=sngl(sigma(1,1))
         if(ind.eq.12)var=sngl(sigma(2,2))
         if(ind.eq.14)var=sngl(sigma(1,2))
         if(ind.eq.15)var=sngl(sigma(2,3))
         if(ind.eq.16)var=sngl(sigma(1,3))
        endif
c
c*** anelastic strains put into local tensor
        if(ind.ge.21.and.ind.le.26)then
         tens(1,1)=ean(1)
         tens(1,2)=ean(4)/2.d0
         tens(1,3)=0.d0
         tens(2,1)=tens(1,2)
         tens(2,2)=ean(2)
         tens(2,3)=0.d0
         tens(3,1)=tens(1,3)
         tens(3,2)=tens(2,3)
         tens(3,3)=0.d0
c
c*** transformed anelastic strains
         sigma(1:3,1:3)=0.d0
         do i=1,3
          do j=1,3
           do k=1,3
            do l=1,3
             sigma(j,i)=sigma(j,i)+xmat(j,k)*tens(k,l)*xmat(i,l)
            enddo
           enddo
```

```
         enddo
        enddo
        if(ind.eq.21)var=sngl(sigma(1,1))
        if(ind.eq.22)var=sngl(sigma(2,2))
        if(ind.eq.24)var=sngl(2.d0*sigma(1,2))
       endif
c
c*** in plane stress resultants  put into local tensor
       if(ind.ge.51.and.ind.le.55)then
        tens(1,1)=carstr(1)
        tens(1,2)=carstr(3)
        tens(1,3)=carstr(5)
        tens(2,1)=tens(1,2)
        tens(2,2)=carstr(2)
        tens(2,3)=carstr(4)
        tens(3,1)=tens(1,3)
        tens(3,2)=tens(2,3)
        tens(3,3)=0.d0
c
c*** transformed in plane stress resultants
        sigma(1:3,1:3)=0.d0
        do i=1,3
         do j=1,3
          do k=1,3
           do l=1,3
            sigma(j,i)=sigma(j,i)+xmat(j,k)*tens(k,l)*xmat(i,l)
enddo
          enddo
         enddo
        enddo
        if(ind.eq.51)var=sngl(sigma(1,1))
        if(ind.eq.52)var=sngl(sigma(2,2))
        if(ind.eq.53)var=sngl(sigma(1,2))
        if(ind.eq.54)var=sngl(sigma(2,3))
        if(ind.eq.55)var=sngl(sigma(1,3))
       endif
c
c*** shear stress resultants
       if(ind.eq.56)
     * var=sngl(carstr(6)*xmat(1,1)**2+
     * 2.d0*carstr(8)*xmat(1,1)*xmat(1,2)+
     * carstr(7)*xmat(1,2)**2)
       if(ind.eq.57)
     * var=sngl(carstr(6)*xmat(2,1)**2+
     * 2.d0*carstr(8)*xmat(2,1)*xmat(2,2)+
     * carstr(7)*xmat(2,2)**2)
       if(ind.eq.58)
     * var=sngl(carstr(6)*xmat(2,1)*xmat(1,1)+
     * carstr(8)*(xmat(1,2)*xmat(2,1)+xmat(1,1)*xmat(2,2))+
     * carstr(7)*xmat(2,2)*xmat(1,2))
c
c***recalculation of in plane stress resultants
       if(ind.eq.61.or.ind.eq.62)then
        tens(1,1)=carstr(1)
        tens(1,2)=carstr(3)
        tens(1,3)=carstr(5)
        tens(2,1)=tens(1,2)
        tens(2,2)=carstr(2)
        tens(2,3)=carstr(4)
        tens(3,1)=tens(1,3)
        tens(3,2)=tens(2,3)
        tens(3,3)=0.d0
        sigma(1:3,1:3)=0.d0
        do i=1,3
         do j=1,3
          do k=1,3
           do l=1,3
```

```
            sigma(j,i)=sigma(j,i)+xmat(j,k)*tens(k,l)*xmat(i,l)
           enddo
          enddo
         enddo
        enddo
c
c*** 1-1 eccentricity
        if(ind.eq.61)then
         enne=sigma(1,1)
         emme=carstr(6)*xmat(2,1)**2+
     *   2.d0*carstr(8)*xmat(2,1)*xmat(2,2)+
     *   carstr(7)*xmat(2,2)**2
         if(enne.ne.0.d0)var=sngl(-emme/enne)
        endif
c
c*** 2-2 eccentricity
        if(ind.eq.62)then
         enne=sigma(2,2)
         emme=carstr(6)*xmat(2,1)**2+
     *   2.d0*carstr(8)*xmat(2,1)*xmat(2,2)+
     *   carstr(7)*xmat(2,2)**2
         if(enne.ne.0.d0)var=sngl(emme/enne)
        endif
       endif
      else
c
c***  case of no shell element
       if(ind.le.16)then
        if(ind.eq.11)var=sngl(t(1))
        if(ind.eq.12)var=sngl(t(2))
        if(ind.eq.14)var=sngl(t(4))
        if(ind.eq.15)var=sngl(t(5))
        if(ind.eq.16)var=sngl(t(6))
       endif
      endif
      return
      end
```

**Remarks:**

output variables with key code "-61" and "-62" are known as eccentricities and are calculated as the minus of the ratio between a bending moment and the corresponding normal force per unit length of the elemental edge.

## 5.8 UBND

The routine UBND allows definition of kinematic constraints (displacements, temperatures, accelerations), which change with varying position and time; the routine is called as needed at the beginning of each iteration.

```
SUBROUTINE UBND ( IFFIX, KFIX, IDOFN, COORD, TDISP,
      * TREAC, FIXED, ICODE)
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'FILES'
INCLUDE 'MOTION'
C
DIMENSION
      * COORD (MCORD),
```

```
      * TDISP (MDOFN),
      * TREAC (MDOFN)
...
user code
...
RETURN
END
```

Arguments of the call:

- *IFFIX:* Fixity code; it is different from zero if the degree of freedom considered has been constrained by means of the keywords BOUNDARY CONDITIONS, BOUNDARY CHANGE, FIXED DISPLACEMENT, FIXED ACCELERATION or FIXED TEMPERATURE (input/output). If the user wants to release the degree of freedom considered, IFFIX must be set to zero (0).
- *KFIX:* Number of the node (input).
- *IDOFN:* Ordinal number of the degree of freedom of the node KFIX whose constraint condition is managed by the routine (input).
- *COORD:* Array of the KFIX node coordinates (input).
- *TDISP:* Array of total current displacements of the node KFIX (input).
- *TREAC:* Array of the total current reactions at the node KFIX (input).
- *FIXED:* If IFFIX is non-zero, it represents the displacement or temperature value prescribed for the degree of freedom IDOFN of the node KFIX (input/output).
- *ICODE:* If set to zero (0), indicates that a displacement or temperature change is to be calculated. If set to one (1), indicates that a total acceleration is to be calculated.
- *MCORD:* Maximum number of coordinate directions (input from COMMON/CNTR/).
- *MDOFN:* Maximum number of degrees of freedom per node (input from COMMON/CNTR/).

*Example of usage*

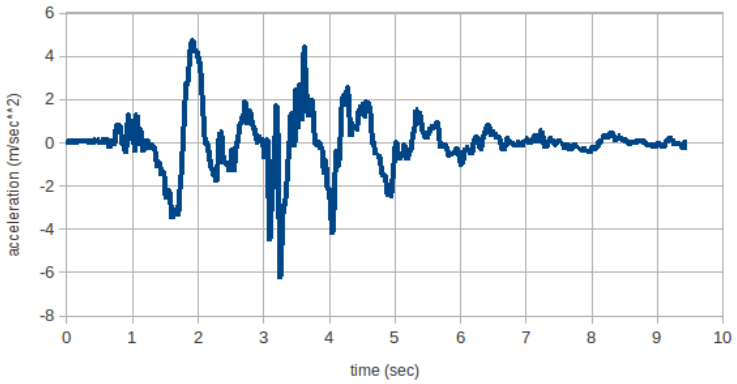



Figure 5.3: Acceleration time history

In carrying out dynamic analyses, we may need to apply a given acceleration time history to the foundations. In Figure 5.3 is plotted a such kind of time history, and we want to apply it, along the *y* direction, to the foundations of a structure. The UBND routine, which is used to apply the acceleration is as follows:

```
      subroutine ubnd (iffix,kfix,idofn,coord,tdisp,treac,fixed,
     *icode)
      implicit real*8 (a-h,o-z)
      include 'cntr'
      include 'files'
      include 'motion'
      dimension
     *coord(mdofn),
     *tdisp(mdofn),
     *treac(mdofn)
      dimension acc(2,971)
c
c*** acceleration time history  like fig. E.3
c*** (not completely shown)
      data ((acc(i,j),i=1,2),j=1,500)/
     *0.000000E+00, 0.000000E+00,
     *0.976000E-02, 0.260000E-02,
     *0.195200E-01, 0.207000E-01,
     *0.292800E-01, 0.312000E-01,
     *0.390400E-01, 0.338000E-01,
     *0.488000E-01, 0.301000E-01,
     *0.585600E-01, 0.238000E-01,
     *0.683200E-01, 0.201000E-01,
     *0.780800E-01, 0.245000E-01,
     *0.878400E-01, 0.369000E-01,
     *0.976000E-01, 0.456000E-01,
     *0.107360E+00, 0.459000E-01,

      ...

     *0.934032E+01, -.738000E-01,
     *0.935008E+01, -.851000E-01,
     *0.935984E+01, -.968000E-01,
     *0.936960E+01, -.112400E+00,
     *0.937936E+01, -.136500E+00,
     *0.938912E+01, -.167100E+00,
     *0.939888E+01, -.196000E+00,
     *0.940864E+01, -.215600E+00,
     *0.941840E+01, -.220200E+00,
     *0.942816E+01, -.160000E-02,
     *0.943792E+01, -.800000E-03,
     *0.944768E+01, -.400000E-03,
     *0.945744E+01, -.200000E-03,

     *0.946720E+01, -.100000E-03/
c***
c
      fuzz=1.d-05
      fixed=0.d0
      tend=0.946720D+01
      if(iincs.gt.1)then
       tt=tottim
       if(tt.gt.tend)then
        fixed=0.d0
       else
        itim=iincs
        if(dabs(acc(1,itim)-tt).gt.fuzz)then
         write(jout,*)' wrong time ',tt,itim,acc(1,itim)
         stop
        endif
        fixed=acc(2,itim)
       endif
```

```
      endif
      return
      end
```

## 5.9 UDSPI

The routine UDSPI allows definition of initial displacement values; it is called once at the beginning of the analysis.

```
SUBROUTINE UDSPI ( DISPI, COORD, IUSER )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
C
DIMENSION
      * COORD (MCORD)
...
user code
...
RETURN
END
```

Arguments of the call:

- *DISPI:* Value of the component of initial displacement (output).
- *COORD:* Coordinates of the node considered (input).
- *IUSER:* Its absolute value indicates which degree of freedom must be initialized (input).
- *MCORD:* Maximum number of coordinates of a node (input, from COMMON/CNTR/).

*Example of usage*

A straight beam is constrained to have a triangle-like initial longitudinal displacement, with the maximum amplitude at the center point. The following routine is used:

```
      SUBROUTINE udspi(DISPI,COORD,iuser)
C*****************************************
C
C     variable initial displacements
C     beam with triangle like displacement
C*****************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      INCLUDE 'cntr'
      DIMENSION
     *COORD(MCORD)
      amp=1.d-04
      xlen=1.d0
      XM=0.5D0*xlen
      dispi=0.d0
      x=coord(1)
      IF(x.le.xm)then
       DISPI=amp*x/xm
      else
```

```
       dispi=amp*(xlen-x)/xm
      endif
      RETURN
      END
```

## 5.10 UFILM

The routine UFILM allows definition of the film coefficients in heat transfer analysis; it is called once at the beginning of each time step.

**This routine is not available in the presence version of NOSA-ITACA.**

```
SUBROUTINE UFILM ( IFAC, IFILM, LNODS, COORD, TTEMP,
      * FILM, SINK, FILDOT )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'LOCAL'
C
DIMENSION
      * COORD (MDIME,NPOIN),
      * FILDOT(MSIDE),
      * FILM(MSIDE),
      * IFILM(2),
      * LNODS(MNODE),
      * SINK(MSIDE),
      * TTEMP(MDOFN, MNODE)
...
user code
...
RETURN
END
```

Arguments of the call:

- *COORD:* Array of the nodal coordinates (input).
- *FILDOT:* Derivative of film coefficient with respect to the temperature at the nodes of the element face (output).
- *FILM:* Film coefficient at the nodes of the element face (output).
- *IFILM:* If IFILM(1) is different from zero, the film coefficient is to be calculated; if IFILM(2) is different from zero, the sink temperature is to be calculated (input).
- *LNODS:* Connectivity of the element (input).
- *SINK:* Sink temperature at the nodes of the element face (output).
- *TTEMP:* Array of the total nodal temperatures.
- *MDIME:* Maximum number of coordinates for each node (input, from COMMON/CNTR/)
- *NPOIN:* Total number of nodes in the mesh (input, form COMMON/CNTR/)
- *MNODE:* Maximum number of nodes for each element (input, from COMMON/CNTR/)

- *MSIDE:* Maximum number of nodes on an element face or edge (input, from COMMON/CNTR/)
- *MDOFN* =1 for all element types, except for the shell element where it is set to ITYRD (input, from COMMON/CNTR/)

## 5.11 UFRI

The routine UFRI allows definition of the friction coefficient as a function of position, time and conditions at the interface between the deformable body and a rigid surface. This routine is activated by inserting the character "-" before the value of the coefficient declared in the card FRICTION. The routine UFRI is called as needed at the beginning of each iteration.

**This routine is not available in the presence version of NOSA-ITACA.**

```
SUBROUTINE UFRI ( THRS, REACN, REACT, KFRICT, CFRICT,
      * COORD, VERSN, VERST )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'MOTION'
C
DIMENSION
      * COORD (MCORD),
      * VERSN (MDIME),
      * VERST (MDIME)
...
user code
...
RETURN
END
```

Arguments of the call:

- *THRS:* Limit value of the friction force (input/output).
- *REACN:* Value of the contact force along the vector orthogonal to the interface between the deformable body and rigid surface (input).
- *REACT:* Value of the force along the vector tangential to the interface between the deformable body and rigid surface (input).
- *KFRICT:* Code indicating the current condition of the friction (input). Possible values are:
  - *KFRICT* = 1, no friction;
  - *KFRICT* = 2, blockage condition;
  - *KFRICT* = 3, slipping condition at constant force.
- *CFRICT:* Friction coefficient (input/output).
- *COORD:* Array of coordinates of the node considered (input).
- *VERSN:* Unit vector orthogonal to the interface oriented towards the interior of the deformable body (input).
- *VERST:* Unit vector tangential to the interface (VERST can be obtained by rotating VERSN 90° in the counter-clockwise direction) (input).

- *MCORD:* Maximum number of coordinates per node (input, from COMMON/CNTR/).
- *MDIME:* Dimensions of the structure (2 in this version of NOSA) (input, from COMMON/CNTR/).

## 5.12 UGEOM

The routine UGEOM allows definition of thickness of the layers of a shell element, or the thickness of the fibers of a beam element, on a nodal basis.

```
SUBROUTINE UGEOM ( COORD, LNODS, GEOM, KTYPE, LCORN )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'ELEM'
INCLUDE 'FILES'
C
DIMENSION
      * COORD(MCORD, NPOIN),
      * GEOM(MDT, LCORN),
      * LNODS(MNODE)
...
user code
...
RETURN
END
```

Arguments of the call:

- *COORD:* Array of coordinates of the node considered (input).
- *LNODS:* Connectivity of the element (input).
- *GEOM:* Array of the nodal thickness (output)
- *KTYPE:* Identifier of the element type (input).
- *LCORN:* Number of the corner nodes of the shell element (input).
- *MCORD:* Maximum number of coordinates for a node (input from COMMON/CNTR/ ).
- *NPOIN:* Maximum number of nodes in the mesh (input from COMMON/CNTR/ ).
- *MDT:* Number of layers/fibers of a shell/beam element (input from COMMON/CNTR/ ).
- *MNODE:* Maximum number of nodes in an element (input from COMMON/CNTR/ ).

*Example of usage*

In studying vaults or domes, we have to model shell structures with variable thickness. In the following example, UGEOM is used to model such a kind of thickness variable dome.

```
      subroutine ugeom(coord,lnods,geom,ktype,lcorn)
      implicit real*8 (a-h,o-z)
      include 'cntr'
      include 'elem'
      include 'files'
```

```
      dimension
     *coord(mcord,npoin),
     *geom(mthick,lcorn),
     *lnods(mnode)
c
c*** bottom level
      bot=10.d0
c
c*** top level
      top=15.0d0
c*** thickness at the bottom level
      thickb=1.d0
c*** thickness at the top level
      thickt=0.5d0
c
      do  inode=1,lcorn
       ipoin=lnods(inode)
       quota=coord(3,ipoin)
c
c*** linearly varying thickness with the z level
       thick=thickb-thickt*(quota-bot)/(top-bot)
       thick=thick/dfloat(mshel)
       geom(1:mshel,inode)=thick
      enddo
      return
      end
```

## 5.13 ULAXIS

The routine ULAXIS allows definition of the local reference system on the cross section of the beam elements.

```
SUBROUTINE ULAXIS ( IELEM, LNODS, COORD, AXIS )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'FILES'
C
DIMENSION
      * AXIS(MCORD),
      * COORD(MCORD, NPOIN),
      * LNODS(MNODE)
...
user code
...
RETURN
END
```

Arguments of the call:

- *AXIS:* Components of the unit vector defining the first direction of the local reference system (output).
- *COORD:* Array of coordinates of the node considered (input).
- *LNODS:* Connectivity of the element (input).
- *MCORD:* Maximum number of coordinates for a node (input from COMMON/CNTR/ ).

- *NPOIN:* Maximum number of nodes in the mesh (input from COMMON/CNTR/ ).
- *MNODE:* Maximum number of nodes in an element (input from COMMON/CNTR/ ).

## 5.14 UMOTIO

The routine UMOTIO allows definition of the velocity of a rigid surface as a function of the time and the position of the surface.

**This routine is not available in the presence version of NOSA-ITACA.**

```
SUBROUTINE UMOTIO ( SURF, TIM, TIMSTP, JSURF )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
C
DIMENSION SURF (8, MPART+1, MSURF)
...
user code
...
RETURN
END
```

Arguments of the call:

- *SURF:* Position and velocity of the rigid surface considered (input/output).
- *TIM:* Total time since the start of the analysis (input).
- *TIMSTP:* Time increment (input/output).
- *JSURF:* Ordinal number of the rigid surface under consideration (input).
- *MPART:* Maximum number of parts making up the rigid surface under consideration (input from COMMON/CNTR/).
- *MSURF:* Maximum number of rigid surfaces (input, from COMMON/CNTR/).

The data relative to the J-th part of the I-th rigid surface are contained in the array SURF as follows:

- SURF(1, 1, I) is the current x coordinate of the surface guide node.
- SURF(2, 1, I) is the current y coordinate of the surface guide node.
- SURF(3, 1, I) is the x component of the current velocity of the surface guide node.
- SURF(4, 1, I) is the y component of the current velocity of the surface guide node.
- SURF(5, 1, I) is the current angular velocity of the surface guide node.
- SURF(6, 1, I) is the initial x coordinate of the surface guide node.
- SURF(7, 1, I) is the initial y coordinate of the surface guide node.
- SURF(8, 1, I) is the current rotation angle of the surface guide node.

If the J-th part is a segment, we have:

- SURF(1, J+1, I) is the current x coordinate of the segment's starting point
- SURF(2, J+1, I) is the current y coordinate of the segment's starting point.
- SURF(3, J+1, I) is the current x coordinate of the segment's end point.
- SURF(4, J+1, I) is the current y coordinate of the segment's end point.

If the J-th part is a circumference arc, we have:

- SURF(1, J+1, I) is the current x coordinate of the arc's starting point.
- SURF(2, J+1, I) is the current y coordinate of the arc's starting point
- SURF(3, J+1, I) is the current x coordinate of the arc's end point.
- SURF(4, J+1, I) is the current y coordinate of the arc's end point.
- SURF(5, J+1, I) is the current x coordinate of the center of the circumference.
- SURF(6, J+1, I) is the current y coordinate of the center of the circumference.
- SURF(7, J+1, I) is the center angle subtended by the arc.
- SURF(8, J+1, I) is the radius of the circumference.

## 5.15 UPLOAD

This routine allows definition of the point loads or fluxes as functions of position and time.

**This routine is not available in the present version of NOSA-ITACA.**

```
SUBROUTINE UPLOAD ( IPOIN, COORD, DISP, PLOAD, SELDIS,
      * LNODS, ICKFR )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'MOTION'
C
DIMENSION
      * COORD(MCORD),
      * DISP(MDOFN),
      * ICKFR(2, MDOFN),
      * LNODS(MNODE, NELEM),
      * PLOAD(MDOFN),
      * SELDIS(MDOFN, MNODE, NELEM)
...
user code
...
RETURN
END
```

Arguments of the call:

- *IPOIN:* Ordinal number of the loaded node (input).

- *COORD:* Array of the initial coordinates of the loaded node (input).
- *DISP:* Array of the total current displacements or temperature change of the loaded node (input).
- *ICKFR: ICKFR(1, I )* indicates the element where the I-th d.o.f of the node IPOIN appears for the first time. *ICKFR(2, I)* indicates the element where the I-th DOF of the node IPOIN can be eliminated from the solution front (input).
- *LNODS:* Array of the connectivities for all elements in the mesh (input).
- *PLOAD:* Array of the point loads or fluxes applied at the node IPOIN (output).
- *MCORD:* Maximum number of coordinates of a node node (input, from COMMON/CNTR/).
- *MDOFN:* Maximum number of degrees of freedom of a node (input, from COMMON/CNTR/).

## 5.16 UPMAS

UPMAS calculates the concentrated masses as a function of the position of the nodes.

```
SUBROUTINE UPMAS ( PMASS, COORD, IUSER.)
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
C
DIMENSION
      * COORD(MCORD),
      * PMASS(MDOFN)
...
user code
...
RETURN
END
```

Arguments of the call:

- *COORD:* Array of the initial coordinates of the loaded node (input).
- *PMASS:* Array of the concentrated masses on a node (output). It should be noted that these are real masses only for translational degrees of freedom, whereas inertia moments are connected to rotational degrees of freedom.
- *IUSER:* Fixed to 1 (input).
- *MCORD:* Maximum number of coordinates of a node (input, from COMMON/CNTR/).
- *MDOFN:* Maximum number of degrees of freedom of a node (input, from COMMON/CNTR/)

## 5.17 URDTEM

Routine for calculating the temperature values at the nodes.

```
SUBROUTINE URDTEM(COORD, DTTMP, DTEMP, TEMPE, NAUTO,
      * ICODE )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'CREAD'
INCLUDE 'ELEM'
INCLUDE 'FILES'
INCLUDE 'TMPPST'
C
DIMENSION
      * COORD(MCORD,NPOIN),
      * DTTMP(ITYRD,NPOIN),
      * DTEMP(ITYRD,NPOIN),
      * TEMPE ITYRD,NPOIN)
...
user code
...
RETURN
END
```

Arguments of the call:

- *COORD:* Array of the initial coordinates of the loaded node (input).
- *DTTMP:* Array of the variation of the nodal temperature at the end of the series of NAUTO increments (output).
- *DTEMP:* Array of the nodal temperature increment (output).
- *TEMPE:* Array of the current nodal total temperature (input).
- *ITYRD:* Indicator of the interpolation type of the temperature across the thickness of a shell element (input, from COMMON/CNTR/).
- *MCORD:* Maximum number of coordinates of a node (input, from COMMON/CNTR/).
- *NPOIN:* Maximum number of nodes in the mesh (input, from COMMON/CNTR/).

## 5.18 UTEMPI

The routine UTEMPI allows definition of initial temperature values. It is called once at the beginning of the analysis.

**This routine is not available in the presence version of NOSA-ITACA.**

```
SUBROUTINE UTEMPI ( TEMPI, COORD, ID, ICODE )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
C
DIMENSION
      * COORD (MCORD)
...
user code
...
```

```
RETURN
END
```

Arguments of the call:

- *TEMPI:* Value of the component of initial temperature (output).
- *COORD:* Coordinates of the node considered (input).
- *ID:* Used for shell elements. Its value varies from 1 to ITYRD (input).
- *ICODE:* Fixed to 1.
- *MCORD:* Maximum number of coordinates of a node (input, from COMMON/CNTR/).

## 5.19 UTIE

Routine for the management of the tying relations (multipoint constraints) among degrees of freedom as functions of position and time. A tying relation means that a degree of freedom (tied) is constrained to assume the value of a linear combination of the values of other degrees of freedom (retained). The routine UTIE is called at the beginning of each load increment.

```
SUBROUTINE UTIE ( ITIE, RTIE, IRTIE, COORD, TDISP, TREAC, M,
      * ITYCH )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
INCLUDE 'FILES'
C
DIMENSION
      * COORD(MDOFN, NPOIN),
      * ITIE(M+2),
      * IRTIE(MDOFN, MRET, MDOFN),
      * RTIE(MDOFN, MRET, MDOFN),
      * TDISP(MDOFN, NPOIN),
      * TREAC(MDOFN, NPOIN)
C
LOGICAL ITYCH
...
user code
...
RETURN
END
```

Arguments of the call:

- *ITIE(1)*: Identifier number of the tying relation (input).
- *ITIE(2)*: Ordinal number of the tied node (input).
- *ITIE(3)*: Ordinal numbers of the retained nodes (input).
- *ITIE*(M+2): Ordinal numbers of the retained nodes (input).

- *RTIE(I, K, J)*: Array of the coefficients of the linear relations among the degrees of freedom of the tied node and the degrees of freedom of the retained node. *RTIE(I,K,J)* links the J-th degree of freedom of the tied node to the I-th degree of freedom of the K-th retained node (output).
- *IRTIE(I, K, J)*: Map of the non-zero coefficients contained in the array RTIE; if IRTIE(I, K, J) is equal to one (1) then the corresponding coefficient RTIE(I, K, J) is different from zero (output).
- *COORD:* Array of the coordinates of all nodes in the mesh (input).
- *TDISP:* Array of the total displacements of all nodes in the mesh (input).
- *TREAC:* Array of the total reactions of all nodes in the mesh.
- *M:* Number of retained nodes relative to the tied node under consideration (input).
- *ITYCH:* Indicator of a change in the overall structure of the tyings (input/output).
- *MDOFN:* Maximum number of degrees of freedom per node (input from COMMON/CNTR/).
- *NPOIN:* Total umber of nodes in the mesh (input from COMMON/CNTR/).
- *MRET:* Maximum number of retained nodes per tied node (input from CPMMON/CNTR/).

*Example of usage* Example 1: Connecting a beam to a shell with pin joint

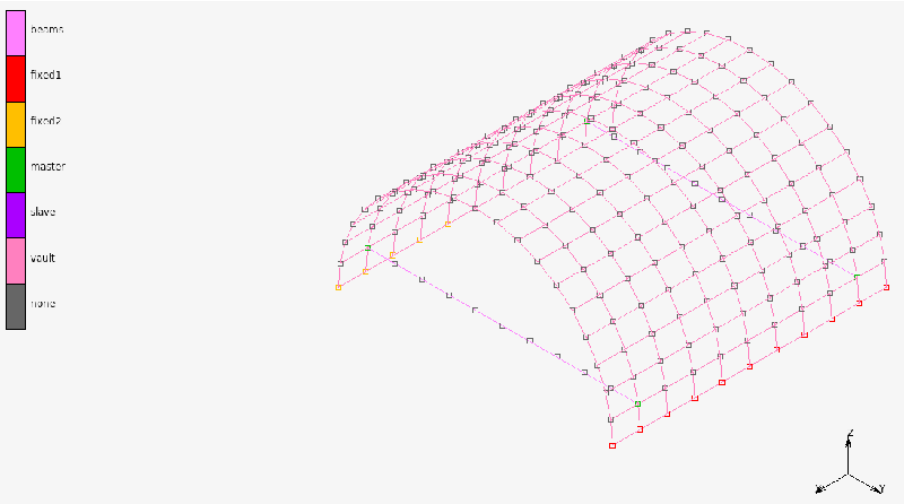



Figure 5.4: beam-shell connection using pin joints

The nodes of the sets "SLAVE" and "MASTER" have the same coordinates, but different numbering: the "SLAVE" nodes are beam nodes, whereas the "MASTER" ones are shell nodes. To set up a pin joint between beams and shells we have to constraint the "SLAVE" nodes to have the same displacements of the "MASTER" ones, whereas the rotations are left free. To do that, we have to use the following UTIE subroutine

```
subroutine utie(itie,rtie,irtie,coord,tdisp,treac,m,ityich)
implicit real*8 (a-h,o-z)
include 'cntr'
include 'files'
dimension
```

```
     *coord(mcord,npoin),
     *tdisp(mdofn,npoin),
     *itie(m+2),
     *rtie(mdofn,mret,mdofn),
     *irtie(mdofn,mret,mdofn),
     *treac(mdofn,npoin)
      logical ityich
c
      do 10 idofn=1,mdofn
      do 10 ir=1,mret
      do 10 jdofn=1,mdofn
      irtie(jdofn,ir,idofn)=0
      rtie(jdofn,ir,idofn)=0.d0
10    continue
c
      rtie(1,1,1)=1.d0
      irtie(1,1,1)=1
      rtie(2,1,2)=1.d0
      irtie(2,1,2)=1
      rtie(3,1,3)=1.d0
      irtie(3,1,3)=1
      return
      end
```

Example 2: Connecting a shell vault and a shell wall with clamped joints

Figure 5.5 shows a vault superimposed on a wall. The thickness of the wall and the vault is 1.0 m and 0.5 m, respectively: so the distance "d" between the middle surfaces of the two structures is 0.25 m. We want to connect the wall and the vault via clumped joints. Denoting by **u** and $\theta$ the displacement and rotation vectors of the nodes on the vault ("BOTVAULT" set) and by **v** and $\phi$ the corresponding vectors of the nodes on the wall ("TOPWALL" set), we have to impose the following set of constraints:

$$u_x = v_x + d \cdot \phi_z,$$

$$u_y = v_y,$$

$$u_z = v_z - d \cdot \phi_x,$$

$$\theta_x = \phi_x,$$

$$\theta_y = \phi_y,$$

$$\theta_z = \phi_z,$$

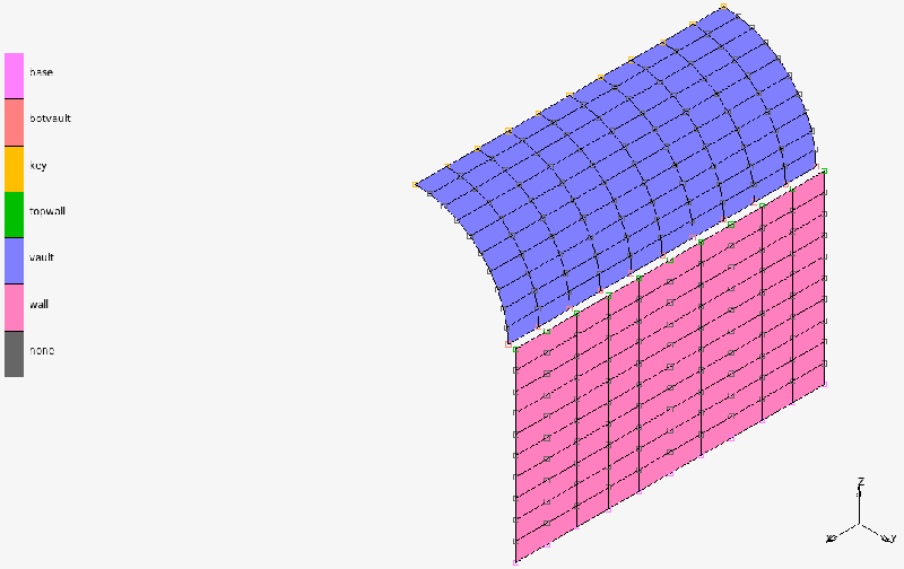



Figure 5.5: Shell-to-shell connections using clamped joints

And we use the following subroutine:

```
      subroutine utie(itie,rtie,irtie,coord,tdisp,treac,m,ityich)
      implicit real*8 (a-h,o-z)
      include 'cntr'
      include 'files'
      dimension
     *coord(mcord,npoin),
     *tdisp(mdofn,npoin),
     *itie(m+2),
     *rtie(mdofn,mret,mdofn),
     *irtie(mdofn,mret,mdofn),
     *treac(mdofn,npoin)
      logical ityich
c
      do 10 idofn=1,mdofn
      do 10 ir=1,mret
      do 10 jdofn=1,mdofn
      irtie(jdofn,ir,idofn)=0
      rtie(jdofn,ir,idofn)=0.d0
10  continue
c
      dist=0.25d0
c
      rtie(1,1,1)=1.d0
      irtie(1,1,1)=1
      rtie(4,1,1)=dist
      irtie(4,1,1)=1
      rtie(2,1,2)=1.d0
      irtie(2,1,2)=1
      rtie(3,1,3)=1.d0
      irtie(3,1,3)=1
      rtie(4,1,3)=-dist
      irtie(4,1,3)=1
      rtie(4,1,4)=1.d0
      irtie(4,1,4)=1
      rtie(5,1,5)=1.d0
      irtie(5,1,5)=1
      rtie(6,1,6)=1.d0
```

```
irtie(6,1,6)=1
return
end
```

Example 3: Connecting a shell vault and a 3D wall with clamped joints

Figures 5.6 - 5.7 show a structure similar to that of the previous example. They differs because the wall is modeled by 3D 8-nodes brick elements. In this case, the nodes of the wall have only 3 degrees of freedom, i.e. the displacements, so the $\phi$ rotations have to calculated by the wall displacements. We recognize that the imposition of a such kind of connection, requires two master DOF for each slave DOF.

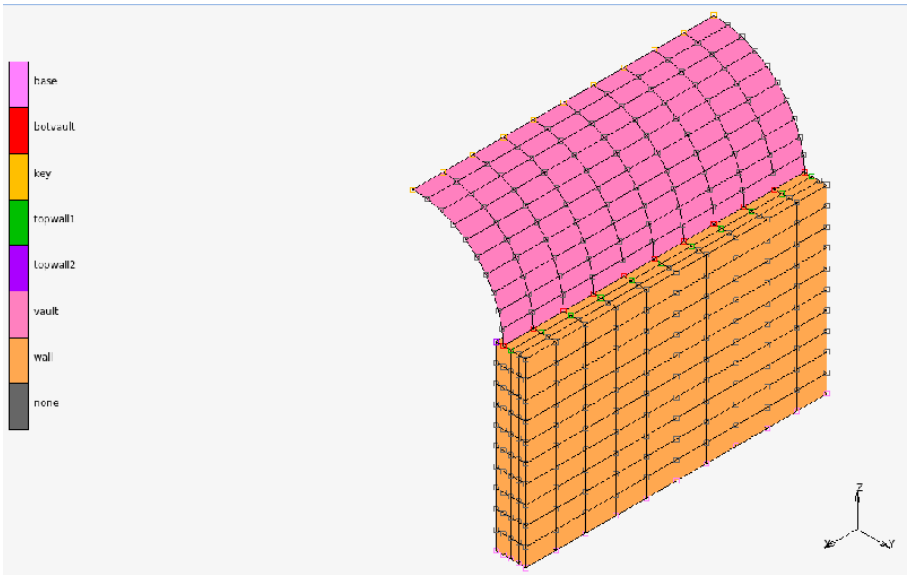



Figure 5.6: Shell-to-solid connections using clamped joints

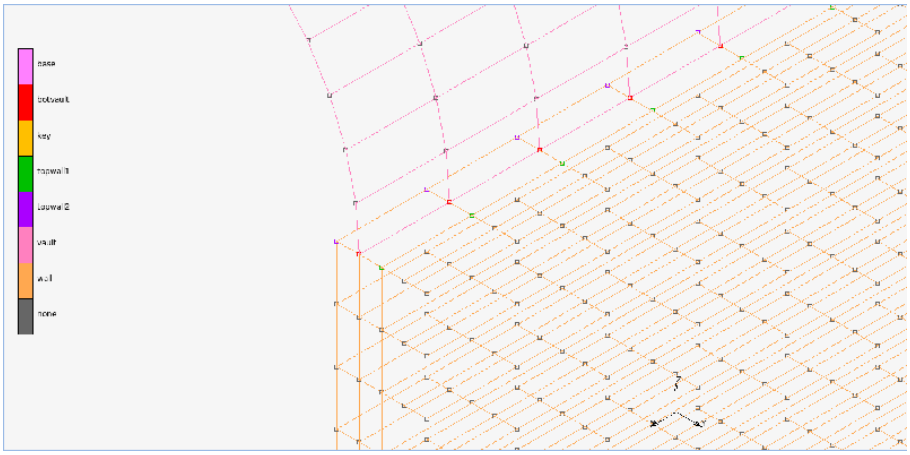


Figure 5.7: Enlarged view of the shell-to-solid connections

In practice, denoting by **u**, $\theta$ the displacements and rotations of the shell nodes ("BOTVAULT" set) and by **v**, **w** the displacements of the brick nodes ("TOPWALL1" and "TOPWALL2" respectively) the rotations $\theta$ are:

$$\phi_x = \theta_x = \frac{v_z - w_z}{2d}$$

$$\phi_z = \theta_z = -\frac{v_x - w_x}{2d}$$

$$\theta_y \; free$$

and the previous constraints became:

$$u_x = \frac{v_x + w_x}{2},$$

$$u_y = v_y,$$

$$u_z = \frac{v_z + w_z}{2},$$

$$\theta_x = \frac{v_z - w_z}{2d},$$

$$\theta_z = -\frac{v_x + w_x}{2d}.$$

And we use the following subroutine:

```
      subroutine utie(itie,rtie,irtie,coord,tdisp,treac,m,ityich)
      implicit real*8 (a-h,o-z)
      include 'cntr'
      include 'files'
      dimension
     *coord(mcord,npoin),
     *tdisp(mdofn,npoin),
     *itie(m+2),
     *rtie(mdofn,mret,mdofn),
     *irtie(mdofn,mret,mdofn),
     *treac(mdofn,npoin)
      logical ityich
c
      do 10 idofn=1,mdofn
      do 10 ir=1,mret
      do 10 jdofn=1,mdofn
      irtie(jdofn,ir,idofn)=0
      rtie(jdofn,ir,idofn)=0.d0
10  continue
c
      dist=0.25d0
c
      rtie(1,1,1)=0.5d0
      irtie(1,1,1)=1
      rtie(1,2,1)= 0.5d0
      irtie(1,2,1)=1
      rtie(2,1,2)=1.d0
      irtie(2,1,2)=1
      rtie(3,1,3)=0.5.d0
      irtie(3,1,3)=1
      rtie(3,2,3)=0.5d0
      irtie(3,2,3)=1
      rtie(3,1,4)=1.d0/(2.d0*dist)
      irtie(3,1,4)=1
      rtie(3,2,4)=-1.d0/(2.d0*dist)
      irtie(3,2,4)=1
      rtie(1,1,6)=-1.d0/(2.d0*dist)
      irtie(1,1,6)=1
```

```
rtie(1,2,6)=1.d0/(2.d0*dist)
irtie(1,2,6)=1
return
end
```

Eaxample 4: Imposition of kinematic constraints along boundaries not aligned with coordinate directions

Figure 5.8 shows a portion (30°) of the cross section of a thick-wall cylinder. On the boundary named BASE0, the y component of displacement is constrained to 0, whereas on the boundary named BASE30 the component of displacement along the direction normal to the boundary is constrained to 0. So, on BASE30 the following constraint holds:

$$\mathbf{u} \cdot \mathbf{n} = u_x n_x + u_y n_y = 0,$$

and then

$$u_x = -\frac{n_y}{n_x} u_y.$$

We have to remark that in this case the slave and master DOF are different DOF of the same node. The corresponding UTIE subroutine is:

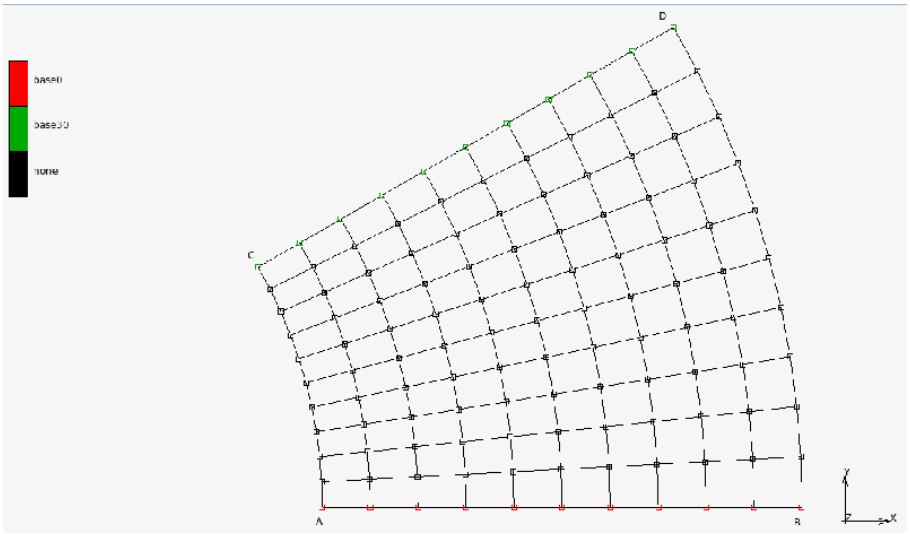


Figure 5.8: Multipoint constraints in polar coordinates

```
 subroutine utie(itie,rtie,irtie,coord,tdisp,treac,m,ityich)
 implicit real*8 (a-h,o-z)
 include 'cntr'
 include 'files'
 dimension
*coord(mcord,npoin),
*tdisp(mdofn,npoin),
*itie(m+2),
*rtie(mdofn,mret,mdofn),
*irtie(mdofn,mret,mdofn),
*treac(mdofn,npoin)
 logical ityich
```

```
      pg=dacos(-1.d0)
c
      do 10 idofn=1,mdofn
      do 10 ir=1,mret
      do 10 jdofn=1,mdofn
      irtie(jdofn,ir,idofn)=0
      rtie(jdofn,ir,idofn)=0.d0
10  continue
c
      angle=30.d0*pg/180.d0
      nx=-dsin(angle)
      ny=dcos(angle)
      coef=-ny/nx
c
      rtie(2,1,1)=coef
      irtie(2,1,1)=1
      return
      end
```

## 5.20 UVELI

The routine UVELI allows definition of initial velocities values; it is called once at the beginning of the analysis.

```
SUBROUTINE UVELI ( VELOI, COORD, IUSER, NODE )
C
IMPLICIT REAL*8 (A-H,O-Z)
C
INCLUDE 'CNTR'
C
DIMENSION
      * COORD (MCORD)
...
user code
...
RETURN
END
```

Arguments of the call:

- *VELOI:* Value of the component of initial velocity (output).
- *COORD:* Coordinates of the node considered (input).
- *IUSER:* Its absolute value is the identifier of the displacement component (input).
- *NODE:* Identifier number of the node (input).
- em MCORD: Maximum number of coordinates of a node (input, from COMMON/CNTR/).

# Chapter 6

# Miscellanea

This documentation deals with miscellaneous subjects; in particular, the following topics are covered:

- Description of the COMMON blocks;
- Description of the NOSA-ITACA output files;
- Error codes.

## 6.1 Description of the COMMON blocks

In this section a brief description of the variables contained in the various COMMON of the NOSA-CODE will be given. Any fortran routine may access the data contained in a COMMON area, through an instruction *INCLUDE* followed by the name of the COMMON. The following common blocks will be described:

- CNTR: Control parameters;
- CONVRG: Convergence check;
- CREAD: Arrays for reading input data;
- DYNA: Dynamic analysis parameters;
- ELEM: Parameters characteristic of the elements;
- ENERG: Energy and energy density on the mesh;
- ERROD: Parameters for error management;
- FILES: Definition of the I/O units;
- FREQPR: Parameters for management of printing output;
- LOADS: Parameters for management of centrifugal loads;
- LOCAL: Characteristics of the element under consideration;
- LSIZE: Dimensions of the elemental arrays;
- MOTION: Time parameters;
- POST: Parameters for management of the post-processing data;
- SHFUN: Shape functions and derivatives;
- SIZE: Pointers to the arrays in the working areas;
- TIMCPU: CPU timing;
- TITL: Title of the analysis;
- TMPPST: Thermal post-processing file data.

### 6.1.1 /CNTR/

This COMMON contains most of the control parameters.

```
 COMMON/CNTR/ICARD, IINCS, IITER, IOUTP, ITYRD, ISAVE, KEAREA,
*            KRESL, LTOTV, MAXBW, MBCASI, LVPRT, MBDT, MBUF1 ,
*            MBUF2, MBUF3, MBPINT, MBINT(2), MBTHCK,MBSHEL(2),
*            MCASI, MCHAN, MCHAN1, MCORD, MDEFO, MDIST, MDIME,
*            MDOFN, MDT, MINT, MPINT, MEAREA, MEVAB, MFAC,
*            MFRON, MGAUS, MGEOM, MHARD, MITER, MNODE, MPART,
*            MPART0, MPROP, MRET, MSET, MSETE, MSETEN, MSETN,
*            MSHEL, MSIDE, MSTIF, MSURF, MTAB, MTHICK, MTIE,
*            MTTEMP, MTYPE,MVSETE, MVSETN, NALGO, NELEM, NMAPH,
*            NMATS, NPOIN, NRAPH, MGASPT, NEIGVAL,
*            IALIA, IBEAM, ICOMP, ICORE, IDAMP, IDYNA, IELAS,
*            IELSH, IIFILM, IFINI, IFOLL, IFRIC, IHEAT, IMASO,
*            IREST, IRSDIS, ISCAL, ISTOP, ITRES, FIRST, LUMP,
*            LENER, IHHT, IELSH5, ITRIA, ITRUSS, IMODAL
 LOGICAL     IALIA, IBEAM, ICOMP, ICORE, IDAMP, IDYNA, IELAS,
*            IELSH, IIFILM, IFINI, IFOLL, IFRIC, IHEAT, IMASO,
*            IREST, IRSDIS, ISCAL, ISTOP, ITRES, FIRST, LUMP,
*            LENER, IHHT, IELSH5, ITRIA, ITRUSS, IMODAL
```

A brief description of the common variables is given in Table 6.1.

Table 6.1: Variables of the COMMON /CNTR/

| *Variable* | *Description* |
|---|---|
| FIRST | If .TRUE., the analysis is at the beginning (the first iteration of first increment). |
| IALIA | If .TRUE., indicates the presence of element aliases (default .FALSE.). |
| IBEAM | If .TRUE., indicates the presence of beam elements (default .FALSE.). |
| ICARD | FORTRAN unit for reading the input data (default ICARD = 5). |
| ICOMP | If .TRUE., indicates that the sections of the shell or beam element have of non-homogeneous thickness and/or possess non-homogeneous mechanical properties. Activation is required from the card COMPOSITE (default .FALSE.). |
| ICORE | If .TRUE., indicates that an out-of-core analysis will be performed (default .FALSE., calculated in the routine CORE). |
| IDAMP | If .TRUE., indicates that the damping terms have to be calculated in a dynamic analysis (default .FALSE.). |
| IDYNA | If .TRUE., indicates a dynamic analysis (default .FALSE.). |
| IELAS | If .TRUE., indicates selection of the option ELASTIC and that a linear elastic analysis will therefore be performed (default .FALSE.). |
| IELSH | If .TRUE., indicates the presence of shell elements in the mesh (default .FALSE.). |
| IELSH5 | If .TRUE., indicates the presence of type 5 shell elements (thin shells) in the mesh (default .FALSE.). |
| IFINI | If .TRUE., indicates selection of the option FINITE STRAIN and that finite deformation will therefore be considered (default .FALSE.). |
| IFOLL | If .TRUE., indicates selection of the option FOLLOWER FORCES for calculation of the local distributed loads in the current configuration (default .FALSE.). |
| IFRIC | If .TRUE., indicates the use of friction coefficients in a contact analysis (default .FALSE.). |
| IHEAT | If .TRUE., indicates a heat transfer analysis (default .FALSE.). |
| IHHT | If .TRUE., indicates that the HHT method will be used in dynamic analyses (default .FALSE.). |

*Table 6.1:continue in the next page*

*Table 6.1:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| IIFILM | If .TRUE. indicates the use of film coefficients in a heat transfer analysis (default .FALSE.). |
| IINCS | Number of the current load increment (calculated in the routine PSTRES). |
| IITER | Number of the current iteration (calculated in the routine PSTRES). |
| IMASO | If .TRUE., indicates selection of the option MASONRY in the control cards (default .FALSE.). |
| IMODAL | If .TRUE., indicates selection of the option MODAL in the control cards and a calculation of the eigenfrequencies an eigenvectors of the structure will be carried out (default .FALSE.). |
| IREST | If .TRUE., indicates selection of the option RESTART (default .FALSE.). |
| IRSDIS | If .TRUE., indicates that, in a dynamic analysis, the iteration convergence criterion is based on the residual displacements instead on the residual forces (default .FALSE.). |
| ISAVE | If different from zero, it indicates selection of the option SAVE INCREMENT in the load cards and its value represents the saving frequency (default 0, i.e. no saving). |
| ISCAL | If .TRUE., indicates selection of the option SCALE in the control cards (default .FALSE.). |
| ISTOP | If .TRUE., indicates selection of the option STOP in the control cards (default .FALSE.). |
| ITRES | If .TRUE., indicates selection of the option ITRESS in the control cards (default .FALSE.). |
| ITRIA | If .TRUE., indicates the presence of triangular elements in the mesh (default .FALSE.). |
| ITRUSS | If .TRUE., indicates the presence of truss elements (type 35) in the mesh (default .FALSE.). |
| KEAREA | Number of double words making up the working area EAREA which contains the elemental data. When the in-core solution is possible, KEAREA is the sum of the dimensions JEAREA for all elements; otherwise it is the maximum value of the elemental JEAREA (calculated in the routine CORE). |
| LENER | If .TRUE., it indicates selection of the option ENERGY in the control cards (default .FALSE.). |
| LTOTV | Is Actual total number of DOF in the mesh. |
| LUMP | If .TRUE., indicates selection of the option LUMPED MASS in the control cards (default .FALSE.). |
| MAXBW | Is Maximum bandwidth calculated for the assembled matrix in the modal analysis. |
| MBCASI | If beam elements are present, it takes the value MBINT(1) + MBINT(2), otherwise it is set to 1 (default value). |
| MBDT | If beam elements are present, it takes the value MBSHEL(1) + MBSHEL(2), otherwise it is set to 1 (default value). |
| MBINT | If beam elements are present, MBINT(1) is the number of section integration points along the first local direction and MBINT(2) is the number of section integration points along the second local direction, otherwise they are set to 1 (default value). |
| MBPINT | If beam elements are present, it takes the value MBSHEL(1) $*$ MBSHEL(2) (i.e. the total number of integration points on the cross section of the beam) , otherwise it is set to 1 (default value). |
| MBSHEL | If beam elements are present, MBSHEL(1) is the number of section fibers along the first local direction and MBINT(2) is the number of section fibers along the second local direction, otherwise they are set to 1 (default values). |
| MBTHCK | If beam elements are present, it takes the value MBSHEL(1) $*$ MBSHEL(2) (i.e. the total number of fibers on the cross section of the beam), otherwise it is set to 1 (default value). |

*Table 6.1:continue in the next page*

*Table 6.1:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| MBUF1 | Dimension of the arrays for the definition of load increments. Its value is NELEM in case of in-core solution, whereas it is calculated in the routine CORE in case of out-of-core solution. |
| MBUF2 | Dimensions of the arrays used for resolution; it is equal to MDOFN∗NPOIN if an in-core solution is performed, whereas it is calculated from the routine CORE for the out-of-core solution. |
| MBUF3 | Indicates how many elemental work-areas are contained in the EAREA array. Its value is NELEM in case of in-core solution, whereas it is calculated in the routine CORE in case of out-of-core solution. |
| MCASI | Maximum number of integration points trough the thickness for shell elements (default 3). |
| MCHAN | Reserved for future development. |
| MCHAN1 | Reserved for future development. |
| MCORD | Maximum number of coordinates per node. |
| MDEFO | Maximum number of deformable bodies. |
| MDIST | Maximum number of different (by type or element) distributed loads or fluxes (default 3). |
| MDIME | Maximum number of dimensions of the structure. |
| MDOFN | Maximum number of degrees of freedom per node. |
| MDT | The maximum between MSHEL and MBDT. |
| MEAREA | Maximum number of double words in an elemental work-area. |
| MEVAB | Maximum number of degrees of freedom per element. |
| MFAC | Maximum number of faces per element. |
| MFRON | Maximum length of the resolution front (calculated in the routine CHKSTI). |
| MGASPT | Maximum number of Gauss integration points for triangular elements. |
| MGAUS | Maximum number of Gauss integration points for each local direction. |
| MGEOM | Number of nodes per element with variable thickness. |
| MHARD | Maximum number of hardening curves which are present. This value is read from the card HARDENING in the control cards (default 0). |
| MINT | The maximum between MCASI and MBCASI. |
| MITER | Maximum number of iterations permitted for each load increment (default 3). |
| MNODE | Maximum number of nodes per element. |
| MPART | Maximum number of parts making up a rigid surface (default 0). |
| MPART0 | Maximum number of post-processing parts making up a rigid surface. |
| MPINT | The maximum between MCASI and MBPINT. |
| MPROP | Maximum number of properties for each material (default 10). |
| MRET | Maximum number of retained nodes for each tied node in the tying relations (default 0). |
| MSET | Maximum number of entity (nodes or elements) making up a set (default 0). |
| MSETE | Maximum number of sets of elements present in the structure (default 0). |
| MSETEN | Maximum number of entities in a compound set (set to MSET as default value). |
| MSETN | Maximum number of sets of nodes present in the structure (default 0). |
| MSHEL | Maximum number of layers for a shell element (default 2). |
| MSIDE | Maximum number of nodes belonging to an element face. |
| MSURF | Maximum number of rigid surfaces present in the structure (default 0). |
| MTAB | Maximum number of tables of temperature dependent properties (default 0). |
| MTHICK | The maximum between MSHEL and MBTHCK. |
| MTIE | Maximum number of tied nodes in the tying relations (default 0). |
| MTTEMP | Maximum number of data pairs in each table of temperature dependent properties. |
| MTYPE | Maximum number of different element types present in the structure (default 1). |

*Table 6.1:continue in the next page*

*Table 6.1:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| MVSETE | Maximum number of items in an element set. |
| MVSETN | Maximum number of items in a node set. |
| NEIGVAL | Number of eigenfrequencies and eigenvectors to be calculated (default 0) |
| NELEM | Maximum number of elements. |
| NMAPH | Maximum number of materials with anisotropic yield stress. |
| NMATS | Maximum number of materials present in the structure. |
| NPOIN | Maximum number of nodes. |
| NRAPH | Maximum number of values for the anisotropic yield stress . |

*Table 6.1:completed*

### 6.1.2 /CONVRG/

This COMMON contains the parameters that control the convergence of increments.

```
 COMMON/CONVRG/PRESD ,DTMAX, PSCMAX,FACTO ,TOLER, FUZTOL,
*              FUZDIS,FUZTMP,TMPFIN,NCHECK,NCHKOU,
*              MAXINC,NUMSTP
 LOGICAL NCHECK,NCHKOU
```

A brief description of the common variables is given in Table 6.2 .

Table 6.2: Variables of the COMMON /CONVRG/

| *Variable* | *Description* |
|---|---|
| DTMAX | Maximum value of the temperature change found in a heat transfer increment. |
| FACTO | Proportionality factor which multiplies the load increment, as declared in the cards PROPORTIONAL INCREMENT (default 1.). |
| FUZDIS | The minimum value of the norm of the incremental displacements for the displacement convergence check in dynamic analysis. The displacement convergence check is used in dynamic analysis if the norm of total forces, including inertia and damping forces, is less than FUZTOL or if this option is explicitly set in the cards CONTROL. |
| FUZTMP | The minimum value of DTMAX in order to allow the step size variation in heat transfer analysis. |
| FUZTOL | The minimum value of the norm of total forces in order that the convergence check be done. |
| MAXINC | Maximum number of increments allowed in an analysis. |
| NCHECK | If .TRUE. the increment satisfies the convergence criteria, and the next increment can be processed. |
| NCHKOU | Output parameter. If .TRUE. the output of the current increment can be done. |
| NUMSTP | Maximum number of steps in a series, for heat transfer analysis. |
| PRESD | Ratio in % between the norm of residual forces/displacements and the norm of the total forces or incremental displacements. |
| PSCMAX | Factor by which the values of elemental quantities must be divided so that the most stressed integration point is brought to the first yield limit. |
| TMPFIN | The maximum value allowed for the temperature change in an increment of the heat transfer analysis. |

*Table 6.2:continue in the next page*

*Table 6.2:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| TOLER | The maximum ratio, in percent, between the norm of residual forces/displacements and the norm of the total external forces or the incremental displacements must be less than TOLER, in order that an increment satisfy the convergence criterion (default 0.01%). |

*Table 6.2:completed*

### 6.1.3 /CREAD/

This COMMON contains the variables for decoding the input data.

```
 COMMON /CREAD/ FLOUT(8000),CARD(160000),IOUT(16000),
*               CHAR1(20,16000),LCHAR1(16000)
 CHARACTER*1 CARD,CHAR1
```

A brief description of the common variables is given in Table 6.3.

Table 6.3: Variables of the COMMON /CREAD/

| *Variable* | *Description* |
|---|---|
| CARD | Array containing an input data card and its continuations, if any, in formatted characters. |
| CHAR1 | Array containing alphanumeric fields (each fields has a maximum number of twenty characters) present in an input data card and its continuations, if any. |
| FLOUT | Array containing the decoding (from formatted characters to double precision real number) of fields present in an input data card and its continuations, if any. |
| IOUT | Array containing the decoding (from formatted characters to integer numbers) of fields present in an input data card and its continuations, if any. |
| LCHAR1 | Array containing the lengths, in characters, of alphanumeric fields present in CHAR1. |

*Table 6.3:completed*

### 6.1.4 /DYNA/

This COMMON contains the parameters characteristic Newmark integration method for the dynamic analysis.

```
COMMON/DYNA/GAMNEW, BETNEW, ALFHHT
```

A brief description of the common variables is given in Table 6.4.

Table 6.4: Variables of the COMMON /DYNA/

| *Variable* | *Description* |
|---|---|
| BETNEW | Weighting factor of the acceleration in the interpolation of the displacement in the Newmark method (default 0.25). |
| GAMNEW | Weighting factor of the acceleration in the interpolation of the velocity in the Newmark method (default 0.5). |
| ALFHHT | "Alfa" coefficient for the HHT method (default 0.0). |

*Table 6.4:completed*

### 6.1.5 /ELEM/

This COMMON contains the parameters characteristic of the different element types (maximum thirty) used in the analysis.

```
 COMMON /ELEM/ NCORD(50) ,NCORN(50) ,NDIME(50) ,NDOFN(20,50),
*              NEVAB(50) ,NFAC(50)  ,NGASP(50) ,NGAUS(50) ,
*              NNODE(50) ,NPRINC(50),NSIDE(50) ,NSTRE(50) ,
*              NSTR1(50) ,NTYPE(50) ,NALIA(50)
```

A brief description of the common variables is given in Table 6.5.

Table 6.5: Variables of the COMMON /ELEM/

| *Variable* | *Description* |
|---|---|
| NALIA(I) | If different from 0, indicates the alias identifier of the I-th element type. |
| NCORD(I) | Number of coordinates for each node of the I-th element type. |
| NCORN(I) | Number of corner nodes of the I-th element type. |
| NDIME(I) | Number of dimensions of the I-th element type. |
| NDOFN(J, I) | Number of degrees of freedom of the J-th node of the I-th element type. |
| NEVAB(I) | Maximum number of degrees of freedom of the I-th element type. |
| NFAC(I) | Number of faces of the I-th element type. |
| NGASP(I) | Total number of Gauss integration points of the I-th element type. |
| NGAUS(I) | Number of Gauss integration points for each local direction of the I-th element type. |
| NNODE(I) | Total number of nodes of the I-th element type. |
| NPRINC(I) | Number of principal stress components in each integration point of the I-th element type. |
| NSIDE(I) | Number of nodes belonging to a face of the I-th element type. |
| NSTRE(I) | Total number of stress components in each integration point of the I-th element type. |
| NSTR1(I) | Total number of components of the deviatoric part of the stress in each integration point of the I-th element type. |
| NTYPE(I) | Identifier of the I-th element type. |

*Table 6.5:completed*

### 6.1.6 /ENERG/

This COMMON contains the values of the energy terms in the mesh.

```
COMMON/ENERG/ENERM, ENERC, ENERP, WORKEX, WORKVI
```

A brief description of the common variables is given in Table 6.6.

Table 6.6: Variables of the COMMON /ENERG/

| *Variable* | *Description* |
|---|---|
| ENERC | The value of the total kinetic energy. |
| ENERM | ENERC+ENERP-WORKEX. |
| ENERP | The value of the total deformation energy. |
| WORKEX | The value of the total work done by the external forces, including the reaction forces. |
| WORKVI | The value of the work done by the viscous forces. |

*Table 6.6:completed*

### 6.1.7 /ERROD/

This COMMON contains the parameters for management of accidental errors in the input data.

```
COMMON /ERROD/ NEROR(200), IFATL, INFTL
```

A brief description of the common variables is given in Table 6.7 .

Table 6.7: Variables of the COMMON /ERROD/

| *Variable* | *Description* |
|---|---|
| IFATL | Number of the errors encountered which prevent the analysis from continuing. |
| INFTL | Number of the errors encountered which do not prevent the analysis from continuing. |
| NEROR(I) | Number of errors of type I found in reading and decoding of the input data. |

*Table 6.7:completed*

### 6.1.8 /FILES/

This COMMON contains the numbers of the I/O FORTRAN units.

```
 COMMON /FILES/ JINP ,JOUT, JFIL1, JFIL2, JFIL3, JFIL4, IPOSB,
*               IPOSF, IPOSM, JREST, JPSTTF, JPSTTB, NFIL2,
*               KREC2, NREC2
```

A brief description of the common variables will be given in Table 6.8.

Table 6.8: Variables of the COMMON /FILES/

| *Variable* | *Description* |
|---|---|
| IPOSB=21 | FORTRAN unit for writing the post-processing data in binary format. The filename will be name.t16. |
| IPOSF = 20 | FORTRAN unit for writing the post-processing data in character format. The filename will be name.t19. |
| IPOSM=22 | FORTRAN unit for writing in character format the results of modal analysis, if any. The filename will be name_modal.t19. |
| JFIL1 = 1 | FORTRAN unit for writing the definition data of distributed loads in case of out-of-core resolution.The filename will be temp/name.1. |
| JFIL2 = 2 | FORTRAN unit for writing the reduced equation in the case of out-of-core resolution.The filename will be temp/name.2. |
| JFIL3 = 3 | direct access FORTRAN unit for writing the data contained in the elemental work areas in the case of out-of-core solution.The filename will be temp/name.3. |
| JFIL4 = 4 | FORTRAN unit for writing the right-hand members of the reduced equations for out-of core back-substitution.The filename will be temp/name.4. |
| JINP = 5 | FORTRAN unit for reading the input data. The filename will be name.crd. |
| JOUT = 8 | FORTRAN unit for printing the results of the analysis. The filename will be name.prt. |
| JPSTTB=25 | FORTRAN unit for reading the binary post-processing file containing temperature values for the thermal loads evaluation. The filename will be name.t25. |
| JPSTTF=26 | FORTRAN unit for reading the formatted post-processing file containing temperature values for the thermal loads evaluation. The filename will be name.t26. |
| JREST=99 | FORTRAN unit for writing and reading analysis data in the case of saving the load increment or restart of an analysis. The filename will be name.rst. |
| KREC2 | Record index into the current file used to store the reduced equations in the case of out-of-core solution. It varies from 1 to NREC2. |
| NFIL2 | No more used. |
| NREC2 | Maximum number of records in each file used to store the reduced equations in the case of out-of-core solution. |

*Table 6.8:completed*

**Remarks:**

The files name.crd, name.t25 and name.t26 if required, must be present in the directory were the run is executed. Moreover, if an out-of-core solution is necessary, a directory temp must be created. The expresssion 'name' is the name of the file containing the input data, and it will be required by the program at the beginning of a run.

### 6.1.9 /FREQPR/

This COMMON contains the parameters which manage output data printing.

```
COMMON /FREQPR/IFREQP, NELPR, NODPR
```

A brief description of the common variables is given in Table 6.9.

Table 6.9: Variables of the COMMON /FREQPR/

| *Variable* | *Description* |
|---|---|
| IFREQP | Printing frequency of incremental data. If printouts are required, the results will be printed every IFREQP increments (default 1). |
| NELPR | If different from zero, indicates that a printout of results is required only for a selected set of elements (default 0). |
| NODPR | If different from zero, indicates that a printout of results is required only for a selected set of nodes (default 0). |

*Table 6.9:completed*

### 6.1.10 /LOADS/

This COMMON contains the parameters for definition of centrifugal loads.

```
COMMON /LOADS/ROTAX(6), ICENT
```

A brief description of the common variables is given in Table 6.10.

Table 6.10: Variables of the COMMON /LOADS/

| *Variable* | *Description* |
|---|---|
| ICENT | If different from zero, indicates that a rotation axis has been defined (default 0). |
| ROTAX(1..3) | Components of the unit vector corresponding to the angular velocity (default 0., 0., 1.). |
| ROTAX(4...6) | Coordinates of a point belonging to the rotation axis (default 0., 0., 0.). |

*Table 6.10:completed*

### 6.1.11 /LOCAL/

This COMMON contains the information relative to the element under consideration. The presence of different values corresponding to different section integration points (up to 441 layers for shells or 21x21 fibers for beams) is meaningful only for non-homogeneous shell/beam elements.

```
 COMMON /LOCAL/YOUNG(441), POISS(441), SHEAR(441), ALPH1(441),
*              BULK(441) , YELD(441) , YELC(441) , ALFA(441) ,
*              BETA(441) , CKE1(441) , CKE2(441) , RHO0(441) ,
*              COND(441) , SPHE(441) , DENS(441) , EMIS(441) ,
*              ALPHA ,ITYPE ,KTYPE ,LCORD ,LCORN ,LDIME ,
*              LDOFN ,LEVAB ,LFAC  ,LGASP ,LGAUS ,LNODE ,
*              LPRINC,LPROP(441)   ,LSIDE ,LSTRE ,LSTR1 ,
*              LDELTA,LINS  ,LINT  ,LPINT ,LTHCK
```

A brief description of the common variables is given in Table 6.11.

Table 6.11: Variables of the COMMON /LOCAL/

| *Variable* | *Description* |
|---|---|
| ALFA(I) | Value of the parameter used in kinematic hardening model for infinitesimal elasto-plasticity, corresponding to the I-th section integration point of the element[1]. |
| ALPHA | Reserved for future developments. |
| ALPH1(I) | Value of the coefficient of thermal dilatation corresponding to the I-th section integration point of the element. |
| BETA(I) | Value of the parameter b used in isotropic hardening model for infinitesimal elasto-plasticity, corresponding to the I-th section integration point of the element[1]. |
| BULK(I) | Value of the bulk modulus corresponding to the I-th section integration point of the element. |
| CKE1(I) | Value of the parameter e1 used in kinematic hardening model for infinitesimal elasto-plasticity, corresponding to the I-th section integration point of the element[1]. |
| CKE2(I) | Value of the parameter e2 used in kinematic hardening model for infinitesimal elasto-plasticity, corresponding to the I-th section integration point of the element[1]. |
| COND(I) | Value of the heat conductivity corresponding to the I-th section integration point of the element. |
| DENS(I) | Value of the density corresponding to the I-th section integration point of the element. |
| EMIS(I) | Value of the emissivity corresponding to the I-th section integration point of the element. |
| ITYPE | Ordinal number of the element identifier as specified in cards SIZING and/or ELEMENTS. |
| KTYPE | Identifier of the element type. |
| LCORD | Number of coordinates for each node of the element. |
| LCORN | Number of corner nodes in the element. |
| LDIME | Number of element dimensions. |
| LDOFN | Maximum number of degrees of freedom for each node of the element. |
| LEVAB | Total maximum number of degrees of freedom of the element. |
| LFAC | Number of faces of the element. |
| LGASP | Total number of Gauss integration points of the element. |
| LGAUS | Number of Gauss integration points of the element for each local direction. |
| LNODE | Total number of element nodes. |
| LPRINC | Number of principal stress components for each integration point of the element. |
| LPROP(I) | Identifier of the material at the I-th section integration point of the element. |
| LSIDE | Number of nodes belonging to an element face. |
| LSTRE | Number of stress components for each integration point of the element. |
| LSTR1 | Number of components in the deviatoric part of the stress for each integration point of the element. |
| POISS(I) | Value of the Poisson ratio corresponding to the I-th section integration point of the element. |
| RHO0(I) | CKE1(I) + CKE2(I). |
| SHEAR(I) | Value of the shear stiffness modulus corresponding to the I-th section integration point of the element. |
| SPHE(I) | Value of the specific heat per unit mass corresponding to the I-th section integration point of the element. |
| YELC(I) | For masonry-like materials, the value of the compressive strength for the I-th section integration point of the element. |
| YELD(I) | For elastic-plastic materials, the limit value of first shear yield for the I-th section integration point of the element. For masonry-like materials the value of the tensile strength for the I-th section integration point of the element. |

*Table 6.11:continue in the next page*

*Table 6.11:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| YOUNG(I) | Value of the Young modulus corresponding to the I-th section integration point of the element. |

*Table 6.11:completed*

[1]: These parameters are valid only for 2d and 3d elements, so only the value in the position 1 is meaningful

### 6.1.12 /LSIZE/

This COMMON contains the lengths in double words of the arrays making up the work areas EAREA containing the elemental data.

```
 COMMON /LSIZE/
                KACCA(50) ,KBMAT5(50),KBMAT9(50),KBMA10(50),
*               KCBASE(50),KCDERV(50),KCDER0(50),KCENTR(50),
*               KCMAT(50) ,KCOSTR(50),KDMATG(50),KDSTRA(50),
*               KDSTRE(50),KDVOLU(50),KEMASS(50),KENERG(50),
*               KGEI(50)  ,KGPBAS(50),KGPCOD(50),KGRADT(50),
*               KIFBE(50) ,KRSPTF(50),KRSPTI(50),KSDC(50)  ,
*               KSDEPL(50),KSDH(50)  ,KSDUPL(50),KSDZ(50)  ,
*               KSECT(50) ,KSHAP5(50),KSTRAP(50),KSTRUP(50),
*               KTHER(50) ,KTHICK(50),KTSTRA(50),KTSTRE(50),
*               KTTHEI(50),KTTHER(50),KZETA(50) ,KZETSH(50)
```

A brief description of the common variables is given in Table 6.12.

Table 6.12: Variables of the COMMON /LSIZE/

| *Variable* | *Description* |
|---|---|
| KACCA(I) | Dimensions of the array ACCA defining the exponential part of the kinematic hardening for each integration point of the I-th type of element. |
| KBMAT5 | Dimensions of the array BMAT5 containing the strain-displacement matrix for each integration point of the thin shell element. |
| KBMAT9 | Dimensions of the array BMAT9 containing the strain-displacement matrix for each integration point of the beam element. |
| KBMA10 | Dimensions of the array BMA10 containing the strain-displacement matrix for each integration point of the thick shell element. |
| KCBASE | Dimensions of the array CBASE containing the centroidal basis vectors of the shell/beam elements. |
| KCDERV(I) | Dimensions of array CDERV containing the Cartesian derivatives of the shape functions for each integration point of the I-th type of element. |
| KCDER0(I) | Dimensions of array CDER0 containing the Cartesian derivatives of the shape functions calculated at the centroid of the I-th type of element. |
| KCENTR(I) | Dimensions of array CENTR containing the current values of components of the center of elastic range for each integration point of the I-th type of element. |
| KCMAT | Dimensions of the array CMAT containing the damping matrix used in dynamic analysis. |

*Table 6.12:continue in the next page*

*Table 6.12:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| KCOSTR(I) | Dimensions of array COSTRE containing the corrective term to the stiffness matrix due to the presence of anisotropic yield materials for each integration point of the I-th type of element. |
| KDMATG(I) | Dimensions of array DMATG containing the constitutive matrix for each integration point of the I-th type of element. |
| KDSTRA(I) | Dimensions of array DSTRA containing the incremental values of the total deformation components for each integration point of the I-th type of element. |
| KDSTRE(I) | Dimensions of array SDSTRE containing the incremental values of the Cauchy stress components for each integration point of the I-th type of element. |
| KDVOLU(I) | Dimensions of array DVOLU containing the elements of volume or surface used in the Gauss quadrature for each integration point of the I-th type of element. |
| KEMASS(I) | Dimensions of the array EMASS containing the mass matrix terms of the I-th type of element. |
| KENERG | Dimensions of the array ENERG containing the densities of energy and external work. |
| KGEI | Dimensions of the array GEI containing the cross section moments of inertia for beam elements. |
| KGPBAS | Dimensions of the array GPBASE containing the basis vectors for each integration point of the shell elements. |
| KGPCOD(I) | Dimensions of the array GPCOD containing the values of the Cartesian coordinates of the Gauss integration points of the I-th type of element. |
| KGRADT(I) | Dimensions of the array GRADT containing the temperature gradient values for each integration point of the I-th type of element. |
| KIFBE | Dimensions of the array IFBE containing, in case of beam element, the map of the actual fibers of the cross section. |
| KRSPTF(I) | Dimensions of array RSPTF containing the current values of the rotation matrix from global to principal reference system for each integration point of the I-th type of element. |
| KRSPTI(I) | Dimensions of array RSPTI containing the values, at the beginning of the increment, of the rotation matrix from global to principal reference system for each integration point of the I-th type of element. |
| KSDC(I) | Dimensions of array SDC containing the incremental value of the deformation tensor center of the elastic range for each integration point of the I-th type of element. |
| KSDEPL(I) | Dimensions of array SDEP containing the incremental values of the plastic deformation for each integration point of the I-th type of element. |
| KSDH(I) | Dimensions of array SDH containing the incremental values of tensor which defines the exponential part of the kinematic hardening for each integration point of the I-th type of element. |
| KSDUPL(I) | Dimensions of array SDUP containing the incremental values of the crushing deformation for each integration point of the I-th type of element. |
| KSDZ(I) | Dimensions of array SDZ containing the incremental values of the accumulated equivalent plastic strain (Odquist parameter) for each integration point of the I-th type of element. |
| KSECT | Dimensions of the array SECT containing the cross section areas of beam elements. |
| KSHAP5 | Dimensions of the array SHAP5 containing the shape function values for each integration point of the thin shell element. |
| KSTRAP(I) | Dimensions of array TSTRAP containing the total values of the plastic strain components for each integration point of the I-th type of element. |
| KSTRUP(I) | Dimensions of array TSTRUP containing the total values of the crushing strain components for each integration point of the I-th type of element. |

*Table 6.12:continue in the next page*

*Table 6.12:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| KTHER(I) | Dimensions of array THERM containing the values of temperature increments for each integration point of the I-th type of element. |
| KTHICK | Dimensions of the array DELTAH containing the thickness of the layers/fibers of shell/beam elements. |
| KTSTRA(I) | Dimensions of array TSTRA containing the values of the total deformation components for each integration point of the I-th type of element. |
| KTSTRE(I) | Dimensions of array TSTRE containing the total values of the Cauchy stress components for each integration point of the I-th type of element. |
| KTTHEI(I) | Dimensions of the array TTHERI containing the initial temperature values for each integration point of the I-th type of element. |
| KTTHER(I) | Dimensions of array TTHERM containing the values of the total temperature for each integration point of the I-th type of element. |
| KZETA(I) | Dimensions of array ZETA containing the total values of the accumulated equivalent plastic strain (Odquist parameter) for each integration point of the I-th type of element. |
| KZETSH | Dimensions of the array ZETSH containing the position of the integration point along the thickness of shell/beam element types. |

*Table 6.12:completed*

### 6.1.13 /MOTION/

This COMMON contains the parameters for time stepping in contact problems, dynamic analyses or heat transfer problems.

```
 COMMON/MOTION/TOTTIM, DELTIM , TOTTII, DELTII, PERIOD,
*              DIAG  , AUTTIM , IRDINC, IUTLOA, ITYICH,
*              IUSURF, IUFLUX , IUBND , IDYNCH, IACCI ,
*              ITRANS, NAUTO  , INCINI
 LOGICAL AUTTIM, IRDINC, IUTLOA, ITYICH, IUSURF, IUFLUX,
*        IUBND , IDYNCH, IACCI , ITRANS
```

A brief description of the common variables is given in Table 6.13.

Table 6.13: Variables of the COMMON /MOTION/

| *Variable* | *Description* |
|---|---|
| AUTTIM | If .TRUE., indicates that the automatic time stepping procedure will be used in heat transfer analysis. |
| DELTII | Initial time step of a series of increments. |
| DELTIM | Time increment corresponding to the current load increment. |
| DIAG | Estimate of the total mesh dimensions. |
| IACCI | If .TRUE., indicates that the initial acceleration will be calculated in a dynamic analysis. |
| INCINI | Not used. |
| IDYNCH | Not used. |

*Table 6.13:continue in the next page*

*Table 6.13:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| IRDINC | Not used. |
| ITRANS | Not used. |
| ITYICH | If set to .TRUE. in the user routine UTIE, it indicates that the structure of the tying relations is changed and a recalculation of the front-width will be necessary. |
| IUBND | Not used. |
| IUFLUX | Not used. |
| IUSURF | If .TRUE., indicates that the user routine UMOTIO will be called to redefine the velocity of rigid surfaces in a contact analysis. |
| IUTLOA | Not used. |
| NAUTO | Indicates how many increments remain to complete a series. |
| PERIOD | Duration of a series of increments. |
| TOTTII | Initial total time at the beginning of a series of increments. |
| TOTTIM | Total time elapsed from the beginning of the analysis |

*Table 6.13:completed*

### 6.1.14 /POST/

This COMMON contains the parameters managing the post-processing data file.

```
 COMMON/POST/ WRTIM, IBINI , ICONT, NPOST , MVAR, INDVAR(200),
*             INCSB, LSTINC,IREV  , NODREL, CHVAR(48,200)
 CHARACTER*1 CHVAR
 LOGICAL IBINI,ICONT
```

A brief description of the common variables is given in Table 6.14 .

Table 6.14: Variables of the COMMON /POST/

| *Variable* | *Description* |
|---|---|
| CHVAR(I) | Name (maximum 48 characters) of the I-th elemental variable to be written to the post-processing file. |
| IBINI | If .TRUE., the post-processing file will be written in binary, without any format control, otherwise the file will be written in alphanumeric characters in card format. |
| ICONT | If .TRUE., in the case of restart, the initial part of the post-processing file containing the connectivity and nodal coordinates is written. |
| INCSB | Write frequency of iteration data on the post-processing file (Default 0, i.e. only the end-of-increment data will be written). |
| INDVAR(I) | Identifier (as specified in the card POST) of the I-th elemental variable to be written to the post-processing file. |
| LSTINC | Number of the lastly written increment. |
| MVAR | Number of the elemental variables which will be written to the post-processing file for each integration point. |
| NODREL | If different from 0, indicates that relative displacements with respect to node NODREL will be written in the post file. |

*Table 6.14:continue in the next page*

*Table 6.14:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| NPOST | Write frequency for the post-processing file. This file will be written for the first increment and every NPOST increment starting from the increment LSTINC. |
| WRTIM | Write frequency in time, i.e. data will be written if the time increment is a integer multiple of WRTIM (Default 0.0 and the write frequency is controlled by NPOST). |

*Table 6.14:completed*

### 6.1.15 /SHFUN/

This COMMON contains the shape function and their derivatives for each integration point of the element types present in the mesh.

```
 common /shfun/sh1d1(2),sh1d2(2,2),sh2d4(4,5),sh2d8(8,9),
*              sh3d8(8,9),sh3d20(20,27),shsh5(9,4),
*              shf2d4(4,2),shf2d8(8,3),shf3d8(8,4),
*              sf3d20(20,9),shfsh5(4,2),
*              dr1d1(2),dr1d2(2,2),dr2d4(2,4,5),dr2d8(2,8,9),
*              dr3d8(3,8,9),dr3d20(3,20,27),drsh5(2,9,4),
*              drf2d4(2,4,2),drf2d8(2,8,3),drf3d8(3,8,4),
*              df3d20(3,20,9),drfsh5(2,4,2),
*              wg1d1,wg1d2(2),wg1d3(3),wg2d4(4),wg2d8(9),
*              wg3d8(8),wg3d20(27),gp2d4(2,4),
*              shtr3(3,4),shtr6(6,4),shth4(4),shth10(10,5),
*              shftr3(2,2),shftr6(3,3),shfth4(3),shfth10(6,4),
*              drtr3(2,3,4),drtr6(2,6,4),drth4(3,4),
*              drth10(3,10,5),drftr3(2,2),drftr6(3,3),
*              drfth4(2,3),drfth10(2,6,4),
*              wgfth1,wgfth10(4),wgftr2(2),wgftr3(3),
*              wgtr(4),wgth4,wgth10(5)
```

A brief description of the common variables is given in Table 6.15 .

Table 6.15: Variables of the COMMON /SHFUN/

| *Variable* | *Description* |
|---|---|
| DF3D20(I, J, K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the face of the 20-nodes 3D element. |
| DRFSH5(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the edge of the thin shell element. |
| DRFTH10(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the face of the 10-nodes thetraedral element. |
| DRFTH4(I, J) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the integration point of the face of the 4-nodes thetraedral element. |
| DRFTR3(J, K) | Value of the derivative along the local direction of the J-th shape function calculated at the K-th integration point of the edge of the 3-nodes triangular element. |
| DRFTR6(J, K) | Value of the derivative along the local direction of the J-th shape function calculated at the K-th integration point of the edge of the 6-nodes triangular element. |
| DRF2D4(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the edge of the 4-nodes 2D element. |

*Table 6.15:continue in the next page*

*Table 6.15:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| DRF2D8(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the edge of the 8-nodes 2D element. |
| DRF3D8(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the face of the 8-nodes 3D element. |
| DRSH5(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the thin shell element. |
| DRTH10(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the 10-nodes thetraedral element. |
| DRTH4(I, J) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the integration point of the 4-nodes thetraedral element. |
| DRTR3(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the 3-nodes triangular element (the case K=1 indicates the element centroid). |
| DRTR6(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the 6-nodes triangular element (the case K=1 indicates the element centroid). |
| DR1D1(I) | Value of the derivative of the I-th shape function calculated at the integration point of the beam element. |
| DR1D2 | Not used. |
| DR2D4(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the 4-nodes 2D element (the case K=5 indicates the element centroid). |
| DR2D8(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the 8-nodes 2D element. |
| DR3D20(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the 20-nodes 3D element. |
| DR3D8(I,J,K) | Value of the derivative along the I-th local direction of the J-th shape function calculated at the K-th integration point of the 8-nodes 3D element (the case K=9 indicates the element centroid). |
| GP2D4(I, J) | I-th coordinate of the J-th integration point for the 4-nodes 2D elements. |
| SF3D20(I, J) | Value of the I-th shape function calculated at the J-th integration point of the face of the 20-nodes 3D element. |
| SHFSH5(I, J) | Value of the I-th shape function calculated at the J-th integration point of the edge of the thin shell element. |
| SHFTH10(I,J) | Value of the I-th shape function calculated at the J-th integration point of the edge of the 10-nodes thetraedral element. |
| SHFTH4(I) | Value of the I-th shape function calculated at the integration point of the edge of the 4-nodes thetraedral element. |
| SHFTR3(I,J) | Value of the I-th shape function calculated at the J-th integration point of the edge of the 3-nodes triangular element. |
| SHFTR6(I,J) | Value of the I-th shape function calculated at the J-th integration point of the edge of the 6-nodes triangular element. |
| SHF2D4(I, J) | Value of the I-th shape function calculated at the J-th integration point of the edge of the 4-nodes 2D element. |
| SHF2D8(I, J) | Value of the I-th shape function calculated at the J-th integration point of the edge of the 8-nodes 2D element. |
| SHF3D8(I, J) | Value of the I-th shape function calculated at the J-th integration point of the face of the 8-nodes 3D element. |
| SHSH5(I, J) | Value of the I-th shape function calculated at the J-th integration point of the thin shell element |

*Table 6.15:continue in the next page*

*Table 6.15:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| SHTH10(I, J) | Value of the I-th shape function calculated at the J-th integration point of thetraedral 10-nodes element |
| SHTH4(I) | Value of the I-th shape function calculated at the integration point of thetraedral 4-nodes element |
| SHTR3(I, J) | Value of the I-th shape function calculated at the J-th integration point of triangular 3-nodes element |
| SHTR6(I, J) | Value of the I-th shape function calculated at the J-th integration point of triangular 6-nodes element |
| SH1D1(I) | Value of the I-th shape function calculated at the integration point of the beam element. |
| SH1D2 | Not used. |
| SH2D4(I, J) | Value of the I-th shape function calculated at the J-th integration point of the 4-nodes 2D element (the case J=5 indicates the element centroid). |
| SH2D8(I, J) | Value of the I-th shape function calculated at the J-th integration point of the 8-nodes 2D element. |
| SH3D20(I, J) | Value of the I-th shape function calculated at the J-th integration point of the 20-nodes 3D element. |
| SH3D8(I, J) | Value of the I-th shape function calculated at the J-th integration point of the 8-nodes 3D element (the case J=9 indicates the element centroid). |
| WGFTH1 | Gauss integration factor calculated at the integration point of the face of the 4-node thetraedral element. |
| WGFTH10(I) | Gauss integration factor calculated at the I-th integration point of the face of the 10-node thetraedral element. |
| WGFTR2(I) | Gauss integration factor calculated at the I-thintegration point of the edge of the 3-node triangular element. |
| WGFTR3(I) | Gauss integration factor calculated at the I-thintegration point of the edge of the 6-node triangular element. |
| WGTH10(I) | Gauss integration factor calculated at the I-th integration point of t the 10-nodes thetraedral element. |
| WGTH4 | Gauss integration factor calculated at the integration point of t the 4-nodes thetraedral element. |
| WGTR(I) | Gauss integration factor calculated at the I-th integration point of t the triangular element. |
| WG1D1 | Gauss integration factor calculated at the integration point of the beam element. |
| WG1D2(I) | Gauss integration factor calculated at the I-th integration point of the edge of 4-nodes 2D elements. |
| WG1D3(I) | Gauss integration factor calculated at the I-th integration point of the edge of the 8-nodes 2D element. |
| WG2D4(I) | Gauss integration factor calculated at the I-th integration point of the 4-nodes 2D element. |
| WG2D8(I) | Gauss integration factor calculated at the I-th integration point of the 8-nodes 2D element. |
| WG3D20(I) | Gauss integration factor calculated at the I-th integration point of the 20-nodes 3D element. |
| WG3D8(I) | IGauss integration factor calculated at the I-th integration point of the 8-nodes 3D element. |

*Table 6.15:completed*

### 6.1.16 /SIZE/

This COMMON contains the values of the storage used by the principal arrays.

```
 common /size/ mcor1i, mcor1r, mcorei, mcorer, msizi , msizr ,
*              mrec  , lacce , laccei, lasdis, laxis , lboun ,
*              lcaric, lcfric, lcoord, lcpres, lcsete, lcsetn,
*              lcsvlo, ldamp , ldispi,ldtemp, ldttmp,  leload,
*              leloaf, leloat, leqrhs, lequat, lestif, lfext ,
*              lfext0, lffric, lfilm , lfint , lfint0, lfixea,
*              lfixed, lgeom , lgload, lgstif, lhaiso, lhakin,
*              lhard , iboun , lickfr, licpre, lidefo, ldfpst,
*              lidist, lielpr, liffia, liffix, lifilm, lifpre,
*              liglue, lilump, lindnod,linodp, lipbw , liplan,
*              liprea, liraph, lirtie, lisele, lishbm, lish5 ,
*              lisurf, lisur0, litie,  littem, liuaxi, liwset,
*              ljarea, ljpcod, lkfric, llese , llesn , llnods,
*              lltype, lmaso , lmatno, lnacid, lnacva, lnamev,
*              lndest, lndfeq, lndfro, lnhard, lnodof, lnodth,
*              lnpivo, lnsete, lnsetn, lpmass, lpoint, lposgp,
*              lpresc, lprops, lprsca, lraph , lrdist, lrefor,
*              lrload, lrplan, lrsp0 , lrtie , lrttem, lseldi,
*              lsink , lsurf , lsurf0, ltdisp, ltempe, ltempi,
*              ltload, ltreac, lvecrv, lvelo , lveloi, lversn,
*              lverst,lweigp

 integer*8 msizi,msizr,mcor1i,mcor1r,mcorei,mcorer
```

A brief description of the common variables is given in Table 6.16.

Table 6.16: Variables of the COMMON /SIZE/

| *Variable* | *Description* |
|---|---|
| LACCE | Lenght of array ACCE. |
| LACCEI | Lenght of array ACCEI. |
| LASDIS | Lenght of array ASDIS. |
| LAXIS | Lenght of array AXIS. |
| LBOUND | Lenght of array BOUND. |
| LCARIC | Lenght of array CARIC. |
| LCFRIC | Lenght of array CFRICT. |
| LCOORD | Lenght of array COORD. |
| LCPRES | Lenght of array CPRES. |
| LCSETE | Lenght of array CSETE. |
| LCSETN | Lenght of array CSETN. |
| LCSVLO | Lenght of array CSVLOA. |
| LDAMP | Lenght of array DAMP. |
| LDFPST | Lenght of array IDFPST. |
| LDISPI | Lenght of array DISPI. |
| LDTEMP | Lenght of array DTEMP. |
| LDTTMP | Lenght of array DTTMP. |
| LELOAD | Lenght of array ELOAD. |
| LELOAF | Lenght of array ELOAF. |
| LELOAT | Lenght of array ELOAT. |
| LEQRHS | Lenght of array EQRHS. |
| LEQUAT | Lenght of array EQUAT. |

*Table 6.16:continue in the next page*

*Table 6.16:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| LESTIF | Lenght of array ESTIF. |
| LFEXT | Lenght of array FEXT. |
| LFEXT0 | Lenght of array FEXT0. |
| LFFRIC | Lenght of array FFRICT. |
| LFILM | Lenght of array FILM. |
| LFINT | Lenght of array FINT. |
| LFINT0 | Lenght of array FINT0. |
| LFIXEA | Lenght of array FIXEA. |
| LFIXED | Lenght of array FIXED. |
| LGEOM | Lenght of array GEOM. |
| LGLOAD | Lenght of array GLOAD. |
| LGSTIF | Lenght of array GSTIF. |
| LHAISO | Lenght of array HAISO. |
| LHAKIN | Lenght of array HAKIN. |
| LHARD | Lenght of array HARD. |
| LIBOUN | Lenght of array IBOUND. |
| LICKFR | Lenght of array ICKFR. |
| LICPRE | Lenght of array ICPRES. |
| LIDEFO | Lenght of array IDEFO. |
| LIDIST | Lenght of array IDIST. |
| LIELPR | Lenght of array IELPR. |
| LIFFIA | Lenght of array IFFIA. |
| LIFFIX | Lenght of array IFFIX. |
| LIFILM | Lenght of array IFILM. |
| LIFPRE | Lenght of array IFPRE. |
| LIGLUE | Lenght of array IGLUED. |
| LILUMP | Lenght of array ILUMP. |
| LINDNOD | Lenght of array INDNOD. |
| LINODP | Lenght of array INODPR. |
| LIPBW | Lenght of array IPBW. |
| LIPLAN | Lenght of array IPLANE. |
| LIPREA | Lenght of array IPREA. |
| LIRAPH | Lenght of array IRAPH. |
| LIRTIE | Lenght of array IRTIE. |
| LISELE | Lenght of array ISELE. |
| LISHBM | Lenght of array ISHBM. |
| LISH5 | Lenght of array ISH5. |
| LISURF | Lenght of array ISURF. |
| LISUR0 | Lenght of array ISURF0. |
| LITIE | Lenght of array ITIE. |
| LITTEM | Lenght of array ITTEMP. |
| LIUAXI | Lenght of array IUAXI. |
| LIWSET | Lenght of array IWSET. |
| LJAREA | Lenght of array JEAREA. |
| LJPCOD | Lenght of array JPCOD. |
| LKFRIC | Lenght of array KFRICT. |
| LLESE | Lenght of array LESE. |
| LLESN | Lenght of array LESN. |
| LLNODS | Lenght of array LNODS. |
| LLTYPE | Lenght of array LTYPE. |

*Table 6.16:continue in the next page*

*Table 6.16:continue from the previous page*

| *Variable* | *Description* |
|---|---|
| LMASO | Lenght of array MASO. |
| LMATNO | Lenght of array MATNO. |
| LNACID | Lenght of array NACID. |
| LNACVA | Lenght of array NACVA. |
| LNAMEV | Lenght of array NAMEV. |
| LNDEST | Lenght of array NDEST. |
| LNDFEQ | Lenght of array NDFEQ. |
| LNDFRO | Lenght of array NDFRO. |
| LNHARD | Lenght of array NHARD. |
| LNODTH | Lenght of array NODTH. |
| LNPIVO | Lenght of array NPIVO. |
| LNSETE | Lenght of array NSETE. |
| LNSETN | Lenght of array NSETN. |
| LPMASS | Lenght of array PMASS. |
| LPOINT | Lenght of array POINT. |
| LPOSGP | Lenght of array POSGP. |
| LPRESC | Lenght of array PRESC. |
| LPROPS | Lenght of array PROPS. |
| LPRSCA | Lenght of array PRSCA. |
| LRAPH | Lenght of array RAPH. |
| LRDIST | Lenght of array RDIST. |
| LREFOR | Lenght of array REFOR. |
| LRLOAD | Lenght of array RLOAD. |
| LRPLAN | Lenght of array RPLANE. |
| LRSP0 | Lenght of array RSP0. |
| LRTIE | Lenght of array RTIE. |
| LRTTEM | Lenght of array RTTEMP. |
| LSELDI | Lenght of array SELDIS. |
| LSINK | Lenght of array SINK. |
| LSURF | Lenght of array SURF. |
| LSURF0 | Lenght of array SURF0. |
| LTDISP | Lenght of array TDISP. |
| LTEMPE | Lenght of array TEMPE. |
| LTEMPI | Lenght of array TEMPI. |
| LTLOAD | Lenght of array TLOAD. |
| LTREAC | Lenght of array TREAC. |
| LVECRV | Lenght of array VECRV. |
| LVELO | Lenght of array VELO. |
| LVELOI | Lenght of array VELOI. |
| LVERSN | Lenght of array VERSN. |
| LVERST | Lenght of array VERST. |
| LWEIGP | Lenght of array WEIGP. |
| MCOR1I | Total length of the integer arrays calculated by CORE1 |
| MCOR1R | Total length of the real∗8 arrays calculated by CORE1 |
| MCOREI | MCOR1I + total length of the integer arrays calculated by CORE |
| MCORER | MCOR1R + total length of the real∗8 arrays calculated by CORE |
| MREC | Number of records necessary for the file JFIL3. |
| MSIZI | Total length of the integer variables. |
| MSIZR | Total lenght of the real variables. |

*Table 6.16:completed*

### 6.1.17 /TIMCPU/

This COMMON contains the CPU time spent by the analysis.

```
common /timcpu/ cput0, cput
```

A brief description of the common variables is given in Table 6.17.

Table 6.17: Variables of the COMMON /TIMCPU/

| *Variable* | *Description* |
|---|---|
| CPUT | CPU time, in seconds, used by a program run. |
| CPUT0 | Initial CPU time, in seconds. |

*Table 6.17:completed*

### 6.1.18 /TITL/

This COMMON contains the title of the analysis.

```
common /titl/ title(70)
character*1 title
```

A brief description of the common variables is given in Table 6.18.

Table 6.18: Variables of the COMMON /TITL/

| *Variable* | *Description* |
|---|---|
| TITLE | Alphanumeric string (maximum 70 characters) which identifies the analysis. |

*Table 6.18:completed*

### 6.1.19 /TMPPST/

This COMMON contains the parameters which control the reading of a thermal post-processing file needed for calculating thermal loads..

```
 common/tmppst/ irdtmp, inctmp, lastin, ntvar, ntnod, ntele,
*               ntdof , ntgp  , ntnodv
```

A brief description of the common variables is given in Table 6.19 .

Table 6.19: Variables of the COMMON /TMPPST/

| *Variable* | *Description* |
|---|---|
| INCTMP | Number of the increment to be read from post-processing file. |
| LASTIN | Number of the last increment read from the post-processing file. |
| NTDOF | Maximum number of degrees of freedom for each node in the thermal post-processing file. |
| NTELE | Total number of elements present in the thermal post-processing file. |
| NTGP | Maximum number of integration points for each element in the thermal post-processing file. |
| NTNOD | Total number of nodes present in the post-processing file. |
| NTNODN | Number of nodal variables present in the thermal post-processing file. |
| NTVAR | Number of elemental variables present in the thermal post-processing file. |

*Table 6.19:completed*

## 6.2 Description of the NOSA-ITACA files

This section describes all input/output files required/generated by NOSA-ITACA software, when an analysis job is going to run; these files are the following:

- **jobname.crd**;
- **jobname.prt**;
- **jobname.rst**;
- **jobname.t19**;
- **jobname.t16**;
- **jobname.t25**;
- **jobname.t26**;
- **jobname.med**;
- **jobname.sta**;
- **jobname_modal.t19**;
- **jobname.1**;
- **jobname.2**;
- **jobname.3**;
- **jobname.4**.

where *"jobname"* is the name of the job currently in execution. The last four files are generated by NOSA fem code only when an analysis requires an amount of memory large enough to be not supported by the NOSA core; in such a case (**out-of-core solution**), NOSA writes analysis data on disk, and a subdirectory **temp** (located in the directory of the job's execution) must be provided, where these files are written.

### 6.2.1 File "jobname.crd"

This is an ASCII file containing the input data describing the current run, in accordance with the specifications given in Nosa Keywords Reference Guide

### 6.2.2 File "jobname.prt"

This is the ASCII file where some results of the current run are printed. The file is structured into two parts; the first one contains a structured echo of the input data, that is:

- the principal characteristics of the run,
- the connectivities of the elements,
- the coordinates of the nodes,
- the list of the kinematic boundary conditions,
- the description of the element/node sets,
- the values of the termomechanical properties,
- the list of elemental variables to be written on the post-processing files,
- the description of the composition and of the geometry of the elements.

The second part of the file deals with the results of load increments; for each load increment and each iteration of every load increment, the following information can be printed:

1. information about applied loads; incremental and total loads summed over the model are printed.
2. information about CPU time; CPU time at the beginning and end of assembly, at the end of the solution of the system and after the stress recovering is printed.
3. information about solution convergence; values of the external loads, reaction forces, and residual loads are printed. Furthermore, the flag *NCHEK* is also printed, indicating whether the convergence criterion is fullfilled or not.
4. if the node printout is enabled, the values of nodal variables, such as generalized displacements, reaction forces and moments are printed; for dynamic analyses, velocities and accelerations are printed too.
5. if the element printout is enabled, the values of the elemental variables, such as stresses and strains for each integration point of the elements are printed.

### 6.2.3 File "jobname.rst"

This is a binary file containing all data required to restart a run; when a restart analysis has to be performed, as continuation of a previous analysis or a new analysis starting from the final configuration obtained from another analysis, model data (mesh, boundary conditions, loads and so on) and deformed shape configuration are read from this file; therefore, it represents a dump of all the common areas, followed by a dump of all arrays use by the current run.

### 6.2.4 Files "jobname.t16" and "jobname.t19"

Together with the "jobname.med", these files contain the output data required for the job and post-processing analysis results. Jobname.t16 is a binary file, whereas jobname.t19 is an ASCII file. The main contents of these files, in order of writing, are:

1. The header with the principal parameters characterizing the run.
2. The list of the codes and names of the elemental variables to be writte on the file.
3. The type and the connectivity of the elements.
4. The coordinates of the nodes.
5. The nodal codes (not used by SALOME).
6. The information about the sets of elements and nodes.
7. The data on the contact geometry. After these data blocks (valid for any job analysis), the writing of the results of each load increment begins.
8. The header of the increment.
9. The results for the rigid bodies in a contact analysis (if any).
10. The values of the elemental variables, for each integration point of all elements; if the mesh model involves different element types with different number of integration points, the values of a field output will be written on *Ngp* gauss points for any element, with *Ngp* the maximum number of integration points. Clearly, for those elements with the number of integration points less than *Ngp* a null value of the output field is associated with the remaining integration points.
11. The values of the nodal variables, i.e. displacements, rotations , velocities, rotational velocities, accelerations, rotational accelerations, external forces, external moments, reaction forces, reaction moments, temperatures, external heat flux, reaction heat flux. Of course only the nodal variable consinstent with the current analysis are available.

### 6.2.5 Files "jobname.t25" and "jobname.t26"

These files are the ASCII and the binary form, respectively, of the temperature post-file, generated by a previous heat transfer run and to be read in the current run. The files are a simplified version of the normal post- files because, after the common parts, they contain only the values of the nodal temperature.

### 6.2.6 File "jobname.med"

This binary file represents the main output file used for post-processing analysis results and viewing them via NOSA-ITACA software.

### 6.2.7 File "jobname.sta"

The file contains all summary information about the job's run; it is an ASCII file reporting the status of the job execution, such as the initial time of the job, all iterations of every load increment, the residual force for every iteration and the corresponding convergence check, the finishing time of the run and the analysis job execution status (i.e. whether the run is completed or aborted due to errors).

### 6.2.8 File "jobname_modal.t19"

This file is a simplified version of the ASCII post-file written in a modal analysis run. The file contains only one increment subdivided into n subincrement, where n is the required number of eigenfrequencies. In a single subincrement only the nodal values (displacements and rotations) of an eigenvector are written.

### 6.2.9 File "jobname.1"

This file is a binary dump of the arrays defining the load increments RDIST and IDIST in case of out-of-core solution; it is located in the **temp** subdirectory.

### 6.2.10 File "jobname.2"

This file is a binary dump of the solution arrays EQUAT, EQRHS, NAMEV, NPIVO, NDFEQ in case of out-of-core solution; it is located in the **temp** subdirectory.

### 6.2.11 File "jobname.3"

This is a direct access binary file; it has a number of records equal to the number of elements, and its i-th record is a dump of the array EAREA corresponding to the i-th element. The file is located in the **temp** subdirectory.

### 6.2.12 File "jobname.4"

This file is a binary dump of the solution array EQRHS used in case of out-of-core resolution done without a complete elimination process. The file is located in the **temp** subdirectory.

## 6.3 Error Codes

This section describes the error codes associated with a job analysis abortion.

A brief description of these errors is given in Table 6.20.

Table 6.20: Error codes associated with an abnormal job execution

| *Error code* | *Description* |
|---|---|
| 0 | the list of errors encountered is required; |
| 1 | the total number, NPOIN, of nodes of the structure is less than or equal to zero; |
| 2 | the total number, IELEM, of elements of the structure is less than or equal to zero; |
| 3 | the number of constrained loads is less than or equal to zero; |
| 4 | the total number NMATS of different types of materials constituting the structure is less than or equal to zero; |
| 6 | the element type considered does not fall within the range 1 to 10; |
| 7 | the maximum number of different element types making up the structure is not within range (1 to 10), or else the first component of the vector NTYPE is equal to zero; |
| 8 | the element type considered has not been yet implemented; |

*Table 6.20:continue in the next page*

*Table 6.20:continue from the previous page*

| *Error code* | *Description* |
|---|---|
| 9 | the SIZING card has not been defined; |
| 10 | the TITLE card has not been assigned; |
| 15 | a node number equal to zero has been found; |
| 16 | the nodes numbering is erroneous, the matrix LNODS has an element which is negative or greater than the maximum number of points contained in the mesh; |
| 18 | the number corresponding to a constrained node is less than zero or greater than the maximum number of nodes of the mesh; |
| 19 | the number relative to the constrained node currently under consideration is the same as the number of a previous node; |
| 21 | the interpretation of a card is impossible; |
| 95 | the number of the current element is not contained in the list of elements; |
| 110 | the coordinates of the current node are the same as those of a previous node; |
| 111 | a degree of freedom of the current node is not contained in the list; |
| 112 | a node has not been found in the matrix of the mesh connectivity; |
| 113 | the sum of the absolute values of the coordinates of an unused node is different from zero; |
| 114 | there is a unused node in the list of constrained nodes; |
| 115 | the number of a constrained nodes is zero. |

*Table 6.20:completed*