



LAB OF R/C AND MASONRY STRUCTURES
SCHOOL OF CIVIL ENGINEERING
ARISTOTLE UNIVERSITY OF THESSALONIKI
GREECE

GiD+OpenSees Interface
User Manual

Kartalis-Kaounis Theocharis

February 2018

Table of Contents

Introduction.....	1
General	2
Installing/Uninstalling the Interface	2
Loading the OpenSees problem type.....	3
Pre-Processor	6
General	6
Theme.....	6
Units.....	6
Problem Type Transformation.....	7
Reset to new Problem Type.....	7
Geometry Modeling.....	8
<i>Geometrical entities</i>	8
<i>Point Definition</i>	8
<i>Line Definition</i>	9
<i>Surface Definition</i>	9
<i>Volume Definition</i>	9
<i>Modeling Tools</i>	9
Materials/Sections F-D Types	13
Section F-D models	15
<i>Elastic Section</i>	15
<i>Fiber Section</i>	16
<i>Fiber Custom Section</i>	21
<i>FiberInt Section</i>	22
<i>Plate Fiber</i>	23
<i>Elastic Membrane Plate</i>	23
<i>LayeredShell</i>	24
Steel Uniaxial Materials.....	25

<i>Steel01</i>	26
<i>Steel02</i>	28
<i>Reinforcing Steel</i>	29
<i>Hysteretic</i>	30
<i>Ramberg-Osgood</i>	32
Concrete Uniaxial Materials	34
<i>Concrete01</i>	34
<i>Concrete02</i>	35
<i>Concrete04</i>	36
<i>Concrete06</i>	37
<i>ConcreteCM</i>	39
Other Uniaxial Materials	41
<i>Elastic</i>	41
<i>Elastic Perfectly Plastic</i>	42
<i>Elastic Perfectly Plastic with Gap</i>	44
<i>Viscous</i>	45
<i>Viscous Damper</i>	46
<i>Hyperbolic Gap</i>	47
<i>PySimple1</i>	48
<i>TzSimple1</i>	49
<i>QzSimple1</i>	50
Combined Materials	51
<i>Series</i>	51
<i>Parallel</i>	51
<i>Section Aggregator</i>	52
<i>Initial Strain</i>	53
<i>Initial Stress</i>	53
nD Materials	54

<i>Elastic Isotropic</i>	54
<i>Elastic Orthotropic</i>	55
<i>J2Plasticity</i>	56
<i>PressureIndependMultiYield</i>	58
<i>PressureDependMultiYield</i>	62
Element Types	66
Beam Column Elements	68
<i>Elastic Beam Column – Timoshenko Beam Column</i>	68
<i>Force-Based Beam Column</i>	69
<i>Displacement-Based Beam Column</i>	70
<i>Flexure-Shear Interaction Displacement-Based Beam Column</i>	70
Beam-Column Local Axes.....	71
Truss Elements	72
<i>Truss</i>	72
<i>Corotational Truss</i>	73
Surface Elements.....	73
<i>Quad</i>	73
<i>Quad U-P</i>	73
<i>Tri31</i>	74
<i>ShellDKGQ</i>	74
<i>ShellMITC4</i>	75
Volume Elements.....	75
Standard Brick Element	75
Conditions	76
Zero Length Elements.....	77
Records.....	78
Restraints.....	81
Constraints.....	82

<i>Equal Constraints</i>	82
<i>Rigid Link</i>	83
<i>Rigid Diaphragm</i>	83
Loading Parameters	84
<i>Nodal Forces</i>	85
<i>Uniform Forces</i>	85
<i>Nodal Displacements</i>	86
<i>Ground Motion from Record</i>	87
<i>Sine Ground Motion</i>	87
<i>Masses</i>	88
<i>Rayleigh Damping</i>	89
<i>Rayleigh Damping-Elements</i>	91
<i>Dead Loads</i>	91
<i>General Data</i>	93
<i>Intervals Data</i>	95
<i>Output Options</i>	99
<i>Meshing Options</i>	100
<i>Analysis Options</i>	103
<i>GiD+OpenSees menu</i>	106
Processor	111
OpenSees to GiD converter	111
Post Processor	112
View Results and Deformation	114
Plot Graphs	118
Animation	120
Quick Start	123
Tutorial 1 – Static and Modal Analysis of a Three-Story Building	123
<i>Description</i>	123

<i>Problem solution</i>	123
<i>Results</i>	130
<i>Alternative Simulation</i>	133
Tutorial 2 – Pushover Analysis of a Three-Story Building	137
<i>Description</i>	137
<i>Problem solution</i>	138
<i>Results</i>	145
Tutorial 3 – Dynamic Analysis of a Three-Story Building	148
<i>Description</i>	148
<i>Problem solution</i>	148
<i>Results</i>	151
Tutorial 4 – Nonlinear Reversed Cyclic Static Analysis on Beam-Column Joint .	153
<i>Description</i>	153
<i>Problem solution</i>	153
<i>Results</i>	158
Tutorial 5 – Site Response Analysis of a Layered Soil Column	160
<i>Description</i>	160
<i>Problem solution</i>	161
<i>Results</i>	170
Tutorial 6 – Pushover Analysis on 2D Frame considering Soil sustainability	173
<i>Description</i>	173
<i>Problem Solution</i>	174
<i>Results</i>	178
REFERENCES	182

Introduction

GiD+OpenSees Interface is a user-friendly graphical interface for the open source finite element software OpenSees, suitable for modeling and analysis of structural as well as geotechnical problems. It is an add-on (problem type) for GiD and is available as open source, under the GNU General Public License. It consists of a collection of utilities that allows the user to interact by means of a Graphical User Interface. It features the Pre and Post graphical capabilities provided by GiD in addition to the powerful text-only tool for numerical simulation of structural and geotechnical systems, OpenSees solver, through the file transfer connection between them.

This user manual aims at the quick and easy understanding of the necessary tools that GiD offers, and of the various additional tools provided by the add-on (also known as **problem type**) **OpenSees.gid**, by the users. The goal is that the user will be able to efficiently use this problem type to resolve all possible types of problems supported so far. Therefore, there is reference to GiD Pre–Processor, in conjunction with figures, so that user can perceive the planning method of the modeling, the creation of various materials, elements, boundary conditions etc. Finally, the way in which the user can display the results in GiD Post-Processor is described.

The GiD+OpenSees software consists of four main parts:

- **Pre-Processor (GiD)**
- **Processor (OpenSees)**
- **OpenSees to GiD converter (OpenSeesPost.exe)**
- **Post-Processor (GiD)**

In Pre-Processor, we can define all the necessary information (elements, materials, conditions etc.) that will consist the input data by the solver module. After that, OpenSees is invoked in the role of Processor, in which the analysis is taking place, and finally the results are exported in OpenSees text format. OpenSeesPost.exe is then invoked for transferring the results to the GiD Post-Processor in which the results are observed in numerous ways. All this is handled through GiD graphical interface, without programming or code-editing required. In case that the interface does not support some features so far, text-editing is also available in the input tcl file that is sourced for the analysis process by the solver. Post-Processor features some advanced facilities including deformed shaper viewer with animation capabilities, force/stress/strain vectors and contours, line diagrams and x-y graphs.

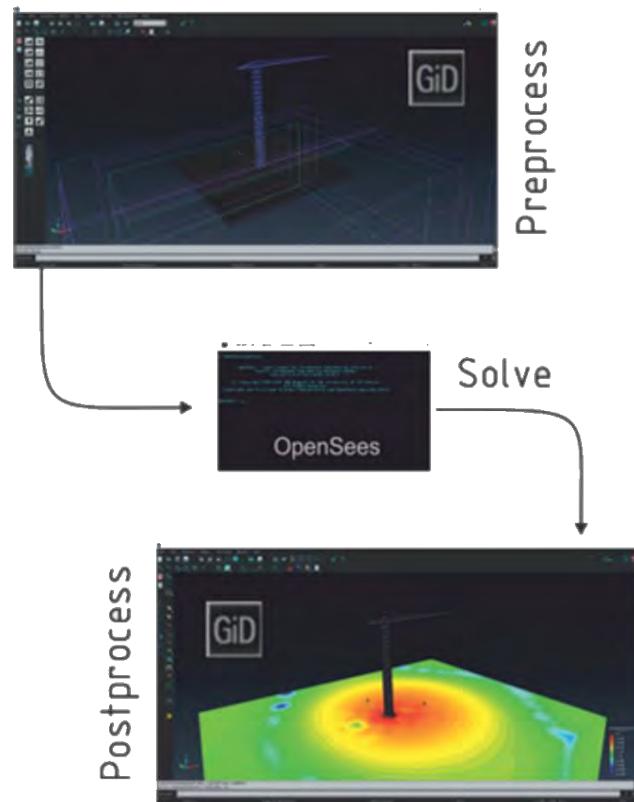


Figure 1: GiD+OpenSees Interface Structure

General

Installing/Uninstalling the Interface

Installing the Interface

1. Download OpenSees and the associated tcl framework through official OpenSees website. Install the tcl framework and copy OpenSees executable in any folder.
2. Download and install GiD through official GiD website.
3. Download the latest version of the GiD+OpenSees Interface (problem type) through your web browser from: <https://github.com/rclab-auth/gidopensees/releases>. You can also find it through the main website <http://gidopensees.rclab.civil.auth.gr> on the *Download* option.
4. Save and launch the setup application.
5. After accepting the agreement, you will be asked to select the OpenSees executable application path, so that it is invoked when analysis is taking place.

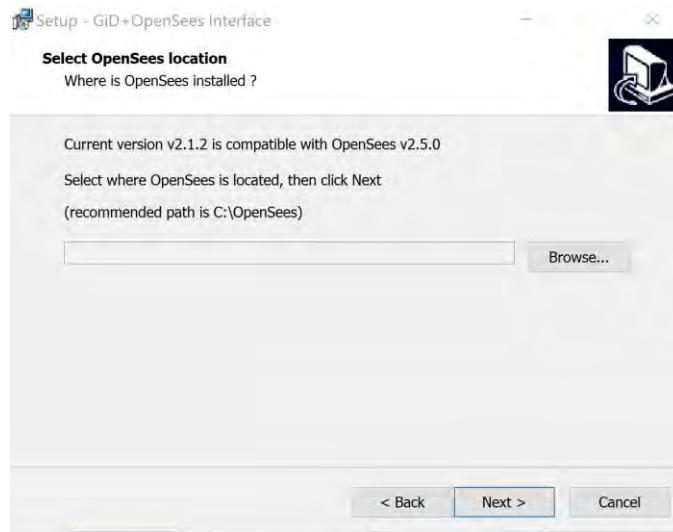


Figure 2 :Installing – Giving the OpenSees location

6. Click the *Next* button and you are ready to go. The problem type files will be transferred to the **latest GiD version** program files installed in your computer.

Uninstalling the Interface

To remove the problem type from the computer:

You can launch the uninstall application inside the OpenSees problem type folder, which can be found in *problemtype* directory, located in installed GiD directory of the computer. Otherwise the files can be removed manually.

Loading the OpenSees problem type

To load the OpenSees facilities within GiD program:

Select *Data > Problem type > OpenSees*.

After that, a toolbar and a menu tab named *GiD+OpenSees* should appear inside GiD Pre-Processor as depicted in Fig. 3.



Figure 3: GiD+OpenSees utilities

Toolbar commands are separated into 6 main groups. The first two are about Material and Element definition, respectively. The third one contains some conditions like constraints and loads assigning as well as Records (time history) definition. The forth corresponds to analysis features and the fifth one contains some general utilities. The last group is placed in the third column of the toolbar and consists of some macros for displaying some specific applied conditions. These commands are presented in the following table beside their button icons.

Table 1: Toolbar commands (v2.3.0)

Command	Toolbar icon	Command	Toolbar icon
Define Standard/Other Uniaxial Materials		Assign Zero Length Elements	
Define Concrete Uniaxial Materials		Define/Assign Truss Elements	
Define Steel Uniaxial Materials		Define/Assign Beam-Column Elements	
Define nD Uniaxial Materials		Define/Assign Surface Elements	
Define F-D Sections		Define/Assign Volume Elements	

Define Combined Materials/Sections		Set General Data	
Define Records		Set Output Options	
Assign Boundary Conditions (Restraints)		Set Interval Data	
Assign Constraints		Generate Mesh	
Assign Masses/Damping		Show/Hide Frame Local Axes	
Assign Loads		Show/Hide Assigned Elements	
Create tcl file, run analysis and go to Post-Processor		Show/Hide Assigned Conditions	
Select Active Interval		Show equal constraint master nodes' IDs	
Show equal constraint slave nodes' IDs		Show rigid link master nodes' IDs	
Show rigid link slave nodes' IDs		Show rigid diaphragm master nodes' IDs	
Show rigid diaphragm slave nodes' IDs		Show zeroLength elements' IDs	
Clear macros		Show line mesh divisions	
Show line mesh sizes		Show point masses	

In addition to the OpenSees Toolbar, a GiD+OpenSees menu tab is created with some analysis as well as general options available. These options are shown in the figure below.

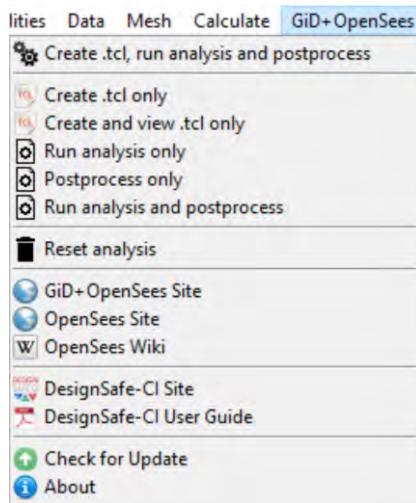


Figure 4: GiD+OpenSees menu options

The GiD+OpenSees menu options are further analyzed in detail later in the relevant section.

Pre-Processor

General

GiD+OpenSees projects are modeled inside the GiD Pre-Processor, which includes various utilities that are used in modeling the structural or geotechnical systems, setting boundary conditions, loading as well as setting analysis options. There are a lot of tools for modeling the required geometrical entities, but also for generating the meshed model, which is actually the representative system for the final numerical analysis.

Theme

GiD+OpenSees supports two different themes that can be set inside the GiD Pre-Processor, **Classic** and **Dark**. The desired theme can be set through *Utilities > Preferences > Graphical > Appearance* choice. The OpenSees problem type toolbars then, will be adapted to the selected GiD theme.

Units

Since OpenSees like many other powerful FEA solvers, does not define specific unit system, it is up to the user what units are used in the input tcl script. Although GiD+OpenSees supports both SI as well as English unit systems and hence the units that are preferred to be used in the input tcl file for OpenSees can be defined from the GiD Data Menu: *Data > Data Units*, which contains some standard units' systems but also supports user-defined systems. Inside this dialog window it is also possible to define the model units that are used for the geometry design.

Whatever property units user uses during the pre-process within dialog windows, all units will be converted as defined in the *Data Units* menu as mentioned above.

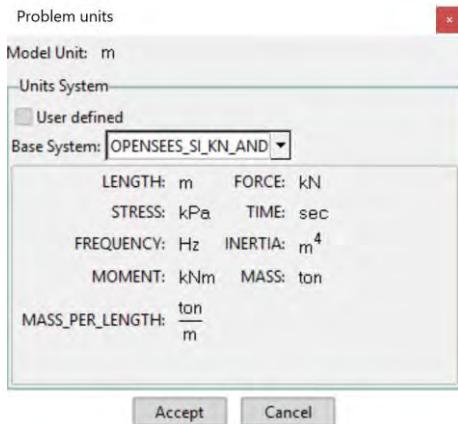


Figure 5: Data units

Problem Type Transformation

Problem type transformation can be found from the Data Menu *Data > Problem type > OpenSees > Transform to new problemtype*. In this way, user can update a model from an old problem type to a newer one that is similar to the first. This option is automatically displayed through a dialog window when the project was saved with a different problem type version than the current. When transforming, all the similar conditions, materials, elements assigned to the geometry as well as the other general and analysis data are maintained if possible. Some data may be lost depending on the type of changes between the two versions, although any changes are handled considering this disadvantage.

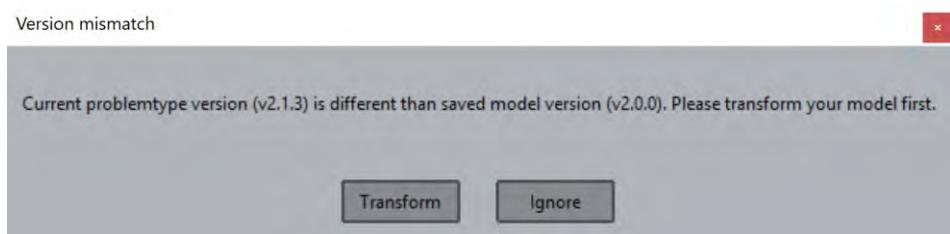


Figure 6: Version mismatch dialog window

Reset to new Problem Type

Reset to new problemtype can be also found from the Data Menu: *Data > Problem type > OpenSees > Reset to new problemtype*. When resetting to the new problemtype, **all data information is lost** and only the geometrical objects are maintained as before.

Geometry Modeling

First, the geometry of the model should be defined before assigning any element or conditions to that. Geometry consists of the geometrical entities which can include points, lines, surfaces and volumes. Any assignment on them will be later transferred to the corresponding (body) element entities or to lower element entities that they consist of, as the mesh is generated.

It is important to deeply understand the difference between Geometry model and Meshed Model. Keep in mind that every existing point in the Geometry model will necessarily correspond to a node at the same location, and as a result the number of nodes (meshed model) will be greater or equal to the number of points (geometry model). This means that an existing node is does not necessarily correspond to a point, and this depends on the user-defined mesh options.

Geometrical entities

A geometrical entity (or object) can consist of individual geometrical entities that follow a hierarchy as follows from the higher to the lowest one:

- **Volumes. They consist of surfaces, line and points**
- **Surfaces. They consist of lines and points**
- **Line. They consist of points**
- **Point. The lowest level of the geometrical objects**

Geometrical entities are not the final deliverable model as input for OpenSees solver. OpenSees solver gets the meshed model as input geometry data after generating the mesh of the geometrical entities. The meshing options are explained in the relevant subparagraph later.

Point Definition

Point is the **lowest level** of the geometrical entities that can be defined. For activating the point definition, the user must select *Geometry > Create > Point*. After that, there are several ways to create points:

1. Use the Coordinates windows

Entering the point coordinates and applying, will create the point. If an existing point with same coordinates is found, user will be asked for joining or not.

2. Use the command line

Point coordinates can be typed in the command line. There are two available formats:

- x,y,z: The components are separated by comma.
- x y z: The components are separated by whitespace.

In case that z coordinate is omitted, it is assumed as zero (X-Y Plane).

3. Use the grid

Grid nodes can be used to create points if *Activate snap* is enabled as shown in Fig. 8.

Line Definition

In general, we can draw lines, which consist of existing or non-existing points. After activating the lines definition from *Geometry > Create > Straight Lines* or the corresponding toolbar icon .

For using existing points, the *Join* command can be enabled via *Right click > Contextual > Join* or *Ctrl+a*.

Otherwise the *Coordinates window*, the *command line* or the *grid* nodes can be used as shown previously. The lines are created by the sequentially point coordinates definition.

Surface Definition

Surfaces can be created from existing points, line points, or lines. These options are available via the Geometry menu: *Geometry > Create > NURBS Surface*. The most usual function is the

creation of a surface using existing lines. The toolbar icon  can enable this function. Once a closed shape of lines is formed, user can select the lines that form the closed shape. A new pink inner color indicates the surface creation.

Volume Definition

Volumes are created from existing surfaces that form a closed 3D shape. The corresponding options are available via the Geometry menu: *Geometry > Create > Volumes*. Beside the manual way (By contour), two automatic functions can also be selected.

Once a closed shape of surfaces is formed, user can select the surfaces that form the closed shape via the toolbar icon . A new light blue inner color indicates the volume creation.

Modeling Tools

A set of useful tools are given to design the geometry of the model easily by GiD.

1. Coordinates window

One of these tools is the Coordinates window which is accessed via the *Utilities menu: Utilities > Tools > Coordinates window.*

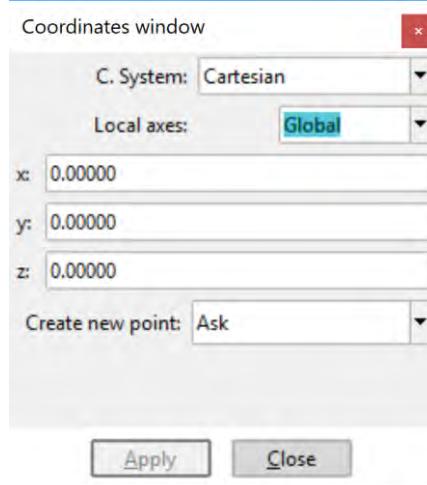


Figure 7: Coordinates window utility

In this way, user can define points, lines or higher geometrical entities giving the points.

2. Grid

GiD Pre-Processor also features a working grid option via the *Utilities menu: Utilities > Preferences* or via the Toolbar icon .

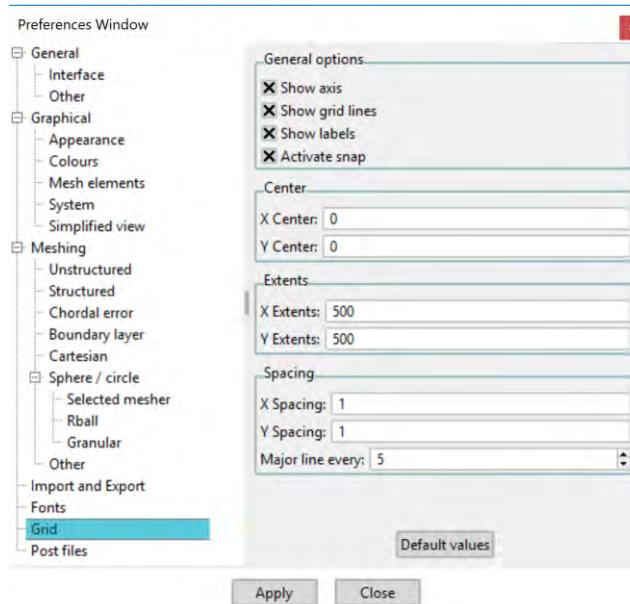


Figure 8 Grid Options

The grid lays on the x-y plane and it is helpful on defining the point coordinates. Setting on the *Activate snap* options allows the user to use the grid nodes. Combining the grid with other utilities such as *copy*, *move*, (shown below) the user can efficiently design in the 3D space.

3. Copy – Move utilities

Two more, basic tools that can be accessed through the *Utilities* menu are the *Copy* and *Move* commands. They are usually extremely necessary tools especially for modeling large systems.

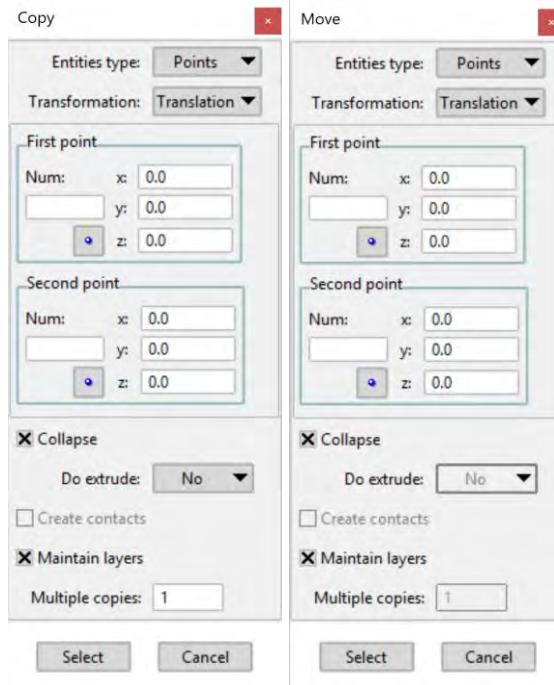


Figure 9 Copy and move utilities

Firstly, the user has to select which geometrical object type is to be moved/copied or select all the geometrical objects, as well as the transformation type:

- *Translation*: Entering the coordinates of two points (existing or not), defines the relative movement to be caused.
- *Rotation*: Initially, the user must define the angle of rotation in degrees. In 2D model, one point's (existing or not) coordinates are asked to define the axis of rotation. The axis direction is towards the Global Z axis. In 3D model, two points' coordinates are required to define the rotation axis and its orientation.
- *Mirror*: In 3D model, three points' coordinates are required to define the mirror planar plane. In contrast, in 2D model, 2 points' coordinates are required to define the mirror line.
- *Scale*: A center point's coordinates, as well as scale factors for every axis are required.

- *Offset*: Only one positive or negative scalar magnitude is required. Each object is then moved/copied in the direction of its normal, by the magnitude given. This option can be selected only for lines and surfaces.
- *Sweep*: This is available only for copying objects. This option is used to copy entities along a path line.
- *Align*: Three source and destination coordinates are required for setting the new desired location from a generic position. The first point defines the exact destination of the source point. In contrast, the second and third points are not necessarily the exact destination. The Second point is just a point over the destination straight line, and the third one is a point over the destination plane.

Some additional options are also available:

Collapse: If it is checked, entities with same position as an existing one that **does not** belong to a frozen layer, are merged.

Do extrude: It creates extruded objects from lower geometrical objects. Lines are generated from copy points, surfaces are created from copying lines, and volumes are created from copying surfaces.

Create contacts: This option is available only when copying surfaces. It generates separated contact volumes for every copied surface.

Maintain layers: If it is checked, the new objects are assigned to the same layer as their originals. Otherwise, they are assigned to the using layer.

Multiple copies: User enters the number of copies and the operation takes place this number of times. It is not combined for Mirror option.

4. Layers and groups

A very often phenomenon is that geometry models consist of multiple geometrical entities (for example nodes) existing at the same coordinates, e.g. when zero length elements are constructed, or SSI problems are simulated. For the best user-friendly modeling, the user may take advantage of GiD layer tools. This utility can be accessed via the *Utilities Menu Utilities > Layers and groups*. This feature allows the user to face a lot of geometry difficulties such as the above. The *On/Off* and *Freeze/Unfreeze* options make the geometry modeling an easy job.

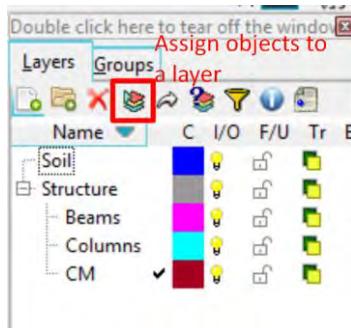


Figure 10: Layer and groups tools

Materials/Sections F-D Types

The Interface Materials and Section F-D features are shown in the following table:

Table 2: Interface Material features (v2.3.0)

Standard Uniaxial Materials	Section Force-Deformation models
<ul style="list-style-type: none"> - Elastic - Elastic Perfectly Plastic - Elastic Perfectly Plastic with Gap 	<ul style="list-style-type: none"> - Elastic - Fiber - Fiber custom (tcl code) - FiberInt - Plate Fiber - Elastic Membrane Plate - LayeredShell
Uniaxial Steel Materials	Other Uniaxial Materials
<ul style="list-style-type: none"> - Steel01 - Steel02 - Reinforcing Steel - Hysteretic - Ramberg-Osgood 	<ul style="list-style-type: none"> - Viscous - Viscous Damper - Hyperbolic Gap - PySimple1 - TzSimple1 - QzSimple1
Uniaxial Concrete Materials	Combined Materials
<ul style="list-style-type: none"> - Concrete01 (Zero tensile strength) - Concrete02 (Linear tensile softening) - Concrete04 (Popovics) - Concrete06 - ConcreteCM (Chang & Mander) 	<ul style="list-style-type: none"> - Parallel - Series - Section Aggregator - Initial Strain - Initial Stress

nD Materials

- Elastic Isotropic
- Elastic Orthotropic
- J2Plasticity
- PressureIndependMultiYield
- PressureDependMultiYield

Initially, Interface includes the above Materials as **template** materials. **These can be modified or copied** by the user through the window dialogs that can be accessed from the OpenSees toolbar.

Note: It is recommended to use names **not including** whitespaces due to a GiD issue. Use underscore or dash instead.

Materials cannot be assigned to the geometrical entities. They are used from the OpenSees finite Element types, which are finally assigned to the geometry model. Thus, the preferred materials or Section F-D models can be defined and stored so that can be used later by the Element types.

A typical Material window is shown below including the various available options.

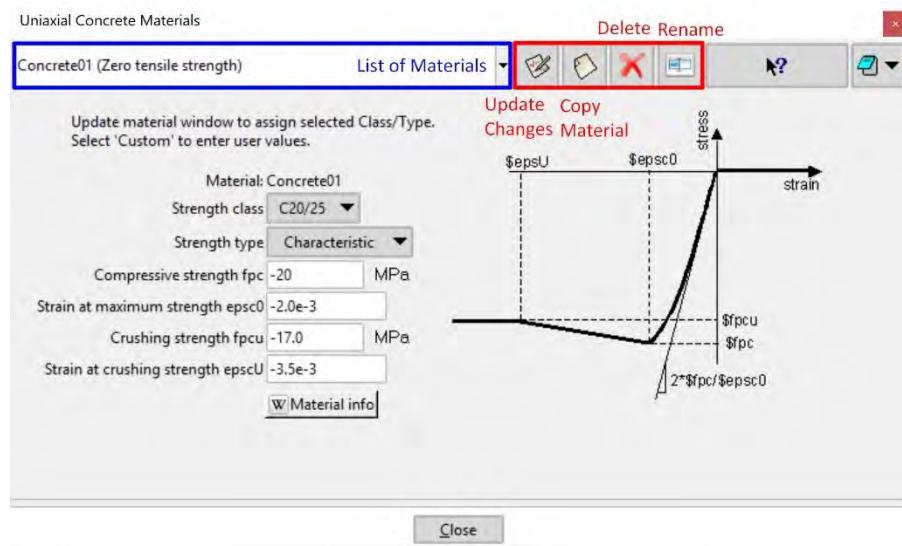


Figure 11: Typical Material dialog window

A few uniaxial materials are often used for Zero Length Elements, so a formulation field option is available to prevent logical units' error conversion at the input tcl file. The formulation options are:

- **Stress – Strain**
- **Force – Deformation**

- **Moment – Rotation**

All Materials dialog windows include a *Material info* button that links to the corresponding command information in official OpenSees Wiki. All parameters required, can be found there as well as a few notes and references for the best understanding from the OpenSees users.

Section F-D models

Elastic Section

This is an elastic section model for being used from nonlinear elements, which is very useful in the initial stages of constructing a large complex model. In this way, the user does not need to create additional elastic beam column elements, but only to create elastic sections.

The geometrical, inertial and material characteristics are either exclusively user-defined or not. Elastic modulus is user-defined, and shear modulus is user-defined or calculated by elastic modulus and Poisson's ratio which is additionally asked to be entered. The cross-sectional characteristics user options are similar to the Elastic Timoshenko Beam-Column element and as a result the user can use the template cross sections type among **Rectangular**, **Tee** and **Circular**, or choose **General** type, where all geometrical and inertial properties are user-defined. Finally, modification factors are also available for modifying specific properties if desired.

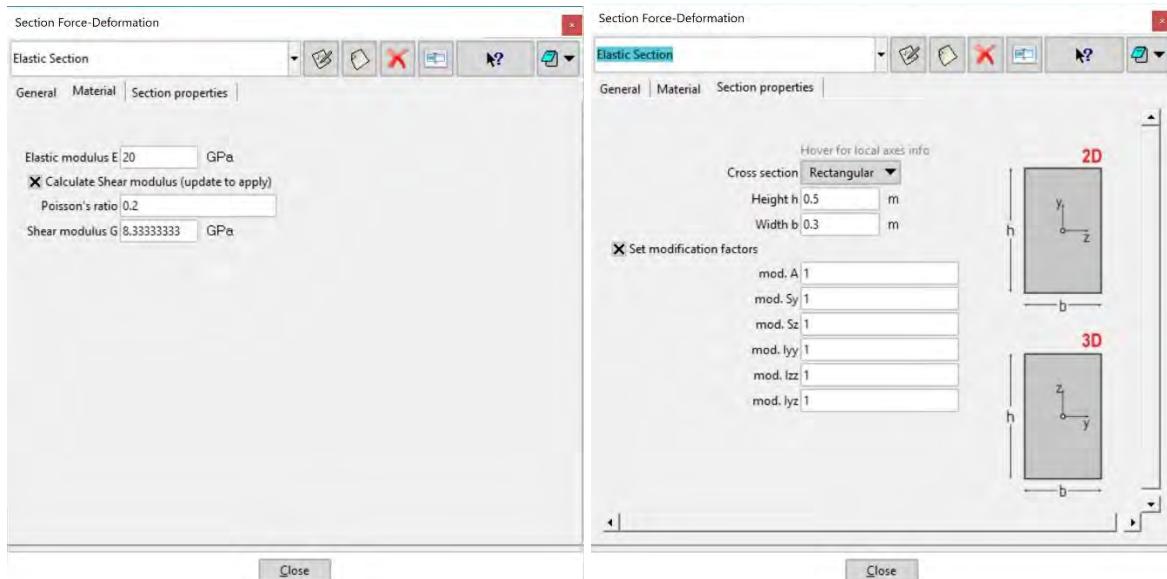


Figure 12: Elastic Section model options

Fiber Section

Fiber Sections are used by the nonlinear fiber elements for the nonlinear analysis of reinforced concrete members. Distributed non-linearity models are assumed to possibly provide a more accurate description of the inelastic performance of reinforced concrete members. In this case the cross section is subdivided into fibers, which consist of the location in the local reference system as well as the fiber area. Each fiber follows a particular uniaxial stress-strain relation (uniaxial material). Consequently, this discretization of the cross section into fibers explicitly derives the cross section constitutive behavior.

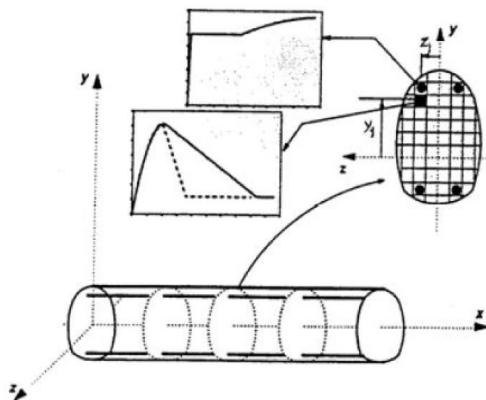


Figure 13: Subdivision of cross-section into fibres

The interface supports some standard cross section types, for the efficient and convenient construction of fiber sections based on the user specified parameters and materials as well. The cross-section types that are available are:

- **Rectangular Column**
- **Rectangular Beam**
- **Tee Beam**
- **Circular Column**
- **Bridge Deck**

In any case the torsional stiffness is **user input** (optional).

- **Rectangular Column**

If rectangular column is selected, except the height and width of the cross section, user can define the cover depth, the number of reinforcing bars along the two faces, the bar areas, and finally the total fibers in both local directions.

Height refers to local y if it is a 2D problem or to local z if it is a 3D problem. So, width refers to local z if it is a 2D problem or to local y if it is a 3D problem.

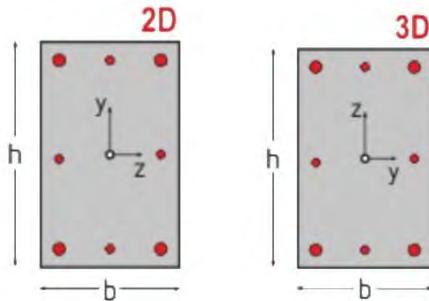


Figure 14: Rectangular Column for Fiber Section with reference local system

Reinforcing bars areas can be different between the corner and intermediate ones.

There are three types of materials that can be chosen: core, cover and reinforcing bar material. Core material corresponds to the confined concrete fibers and cover material to the external unconfined concrete fibers. Cover depth is measured from the outside side to the center of bars.

The number of fibers for each local axis direction in the cover concrete areas is calculated as proportional to the ratio of cover depth with the total relevant dimension.

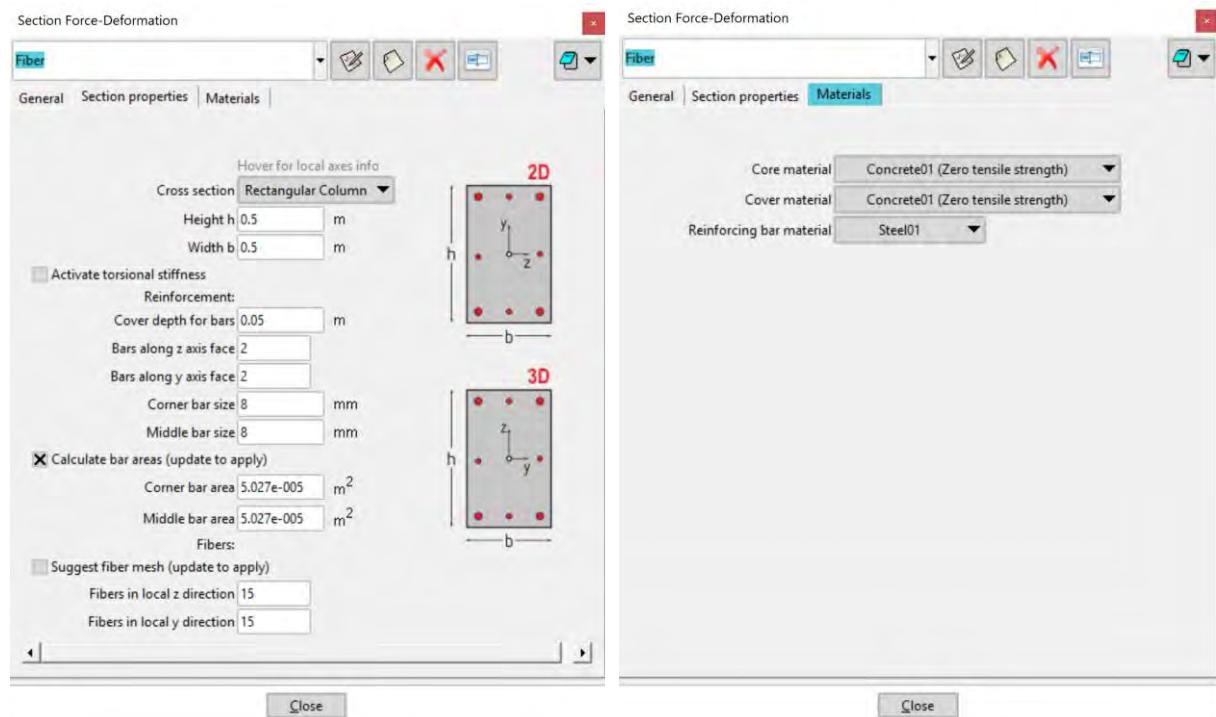


Figure 15: Fiber model – Rectangular Column options

- **Rectangular Beam**

The only difference here is on the reinforcement definition. Reinforcing bars are specified on two faces, which are in the positive and negative direction of the local axis that is parallel to the height. **Top bars are assumed to be in the positive direction and bottom bars in the negative direction.**

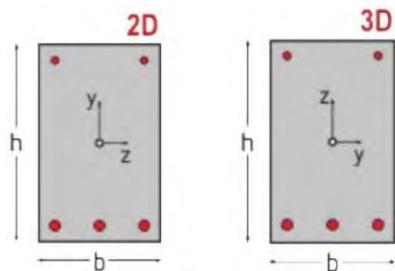


Figure 16: Rectangular Beam for Fiber Section with reference local system

• Tee Beam

In addition to Rectangular Beam, Tee Beam includes a reinforced concrete slab. Slab bars must be an **even** number, so that they are bisected in both sides. Top web bars, bottom web bars and slab bars can be of different size (diameter). Fibers along local y and z axis refer to the total external dimensions. The number of fibers in both local axes for core, cover and slab concrete areas is calculated as proportional to the corresponding dimension divided by the total external one (h or bf).

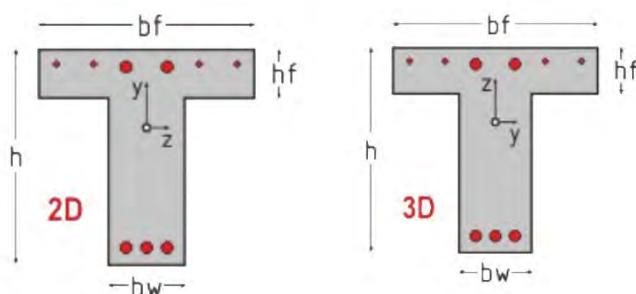


Figure 17: Tee beam for Fiber Section with reference local system

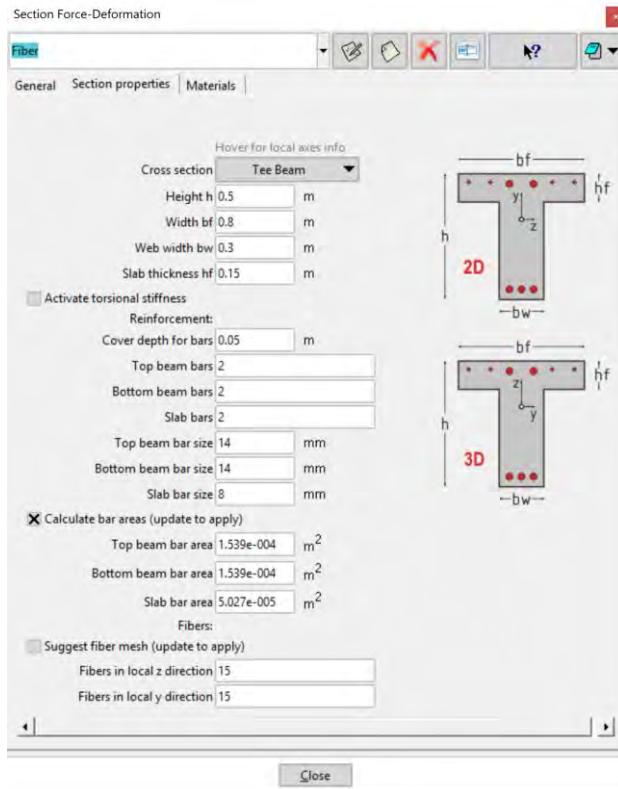


Figure 18: Fiber model – Tee Beam options

- **Circular Column**

For a circular column, user should enter the diameter, the concrete cover depth, the number of reinforcing bars along the arc as well as the bars' area. In this case, number of fibers is given in the circumferential and in the radial direction and refers to the total dimensions. For concrete cover area, the number of fibers in the radial direction is proportional to the cover depth divided by the radius. In this case, the local reference system does not matter at all due to the circular shape.

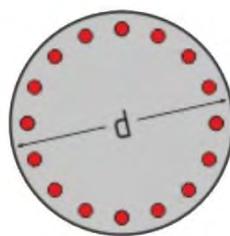


Figure 19: Circular column for Fiber Section

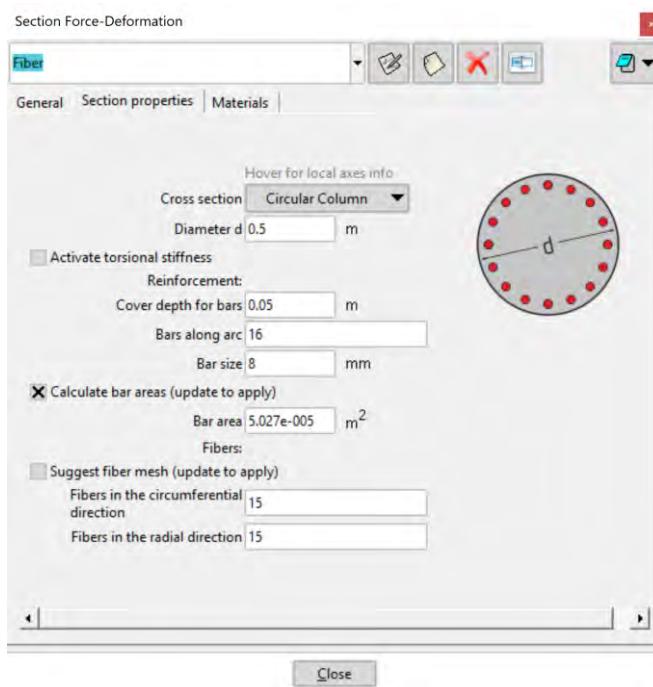


Figure 20: Fiber model – Circular column options

- **Bridge Deck**

Since v2.2.0 bridge deck fiber section is supported for bridge modeling. At the moment, it consists of two parts: main section and additional section, which is **optional**. Main section consists of two reinforced concrete slabs and a rectangular concrete area between them. The rectangular area is either solid or includes voids. Additional part consists of a reinforced slab, two sidewalk slabs and two reinforced concrete beams. For each reinforced part, the bars' area and material can be different. Also, Main section and additional part can use different concrete materials. Finally, the number of fibers is given along both local axes in terms of width, thickness or height for every subpart of the main section like top/bottom slab and internal/external webs. On the other hand, in case that additional part is included, only the number of fibers along width and height for the beams is given as input. Additional slab fibers are assumed to be identical with the top slab's fibers of main section. Sidewalk slabs' fibers along the thickness are also identical with the top slab's fibers of main section and fibers along the width are equal to the number of top slab fibers along the width, divided by 3.

Note: In case that a solid section is selected, the rectangular concrete area between top and bottom slab is assumed to consist of two external webs.

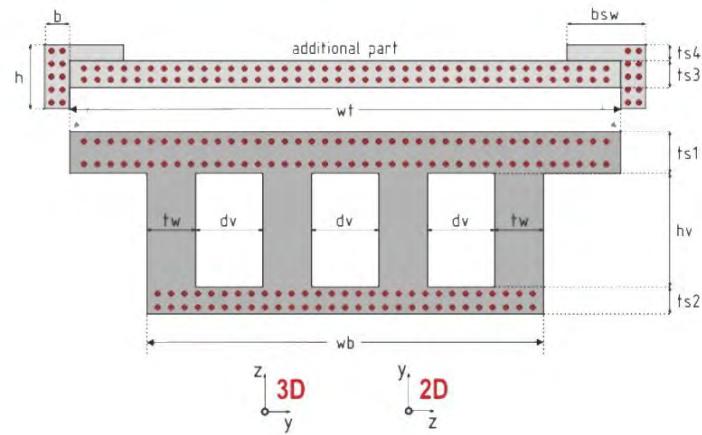


Figure 21: Bridge deck for Fiber Section

Fiber Custom Section

This is a Fiber section model inherited by non-linear fiber elements for the non-linear analysis of general composite cross sectional structural members. It allows the user to enter tcl code through the interface windows and hence can construct any kind of fiber section. The only limitation is that the fiber section must consist of the available uniaxial steel/concrete materials inside the Interface. Inside GiD Pre-Processor, the user can use tcl code up to 10 region-scripts. Each region is made of fibers of a specified uniaxial steel or concrete material. Finally, the cross-sectional Area and linear torsional stiffness are user-defined. Area is considered **only** for the mass of the elements, if mass density is given as a non-zero value and/or for dead loads if activated.

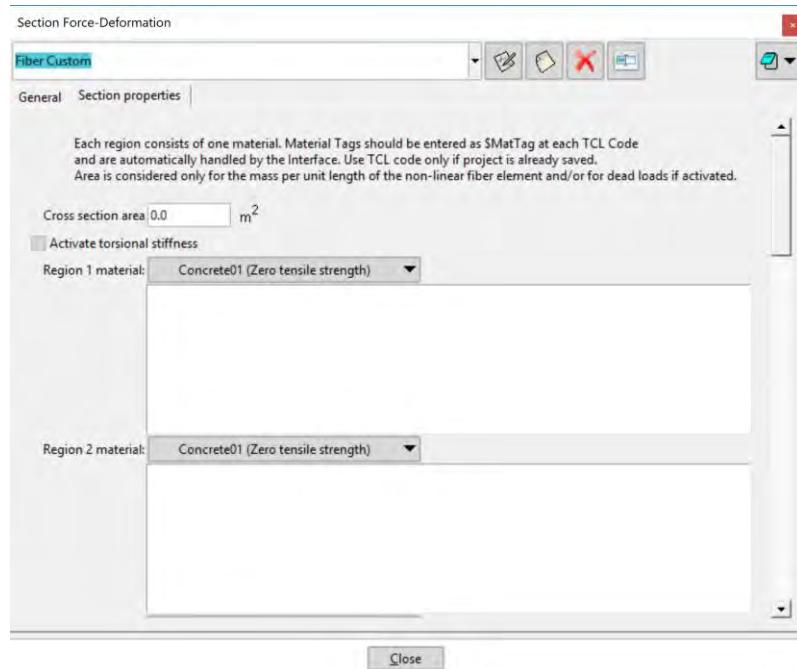


Figure 22: Fiber Custom options

It is not possible from user to know each material tag in advance, and as a result material tag in each region is asked to be entered as “\$*MatTag*” (value of the variable *MatTag* in tcl). Variable *MatTag*’s value is set automatically before the import of each region script.

The tcl code used in the Interface through Fiber Custom windows, is automatically saved in tcl scripts in *Scripts* folder inside project directory (*projectname.gid*) from where it is sourced by the main tcl script (*projectname.tcl*). These scripts use a format *Fiber\$SecTag-\$RegionNumber.tcl*, where *\$SecTag* is the Fiber section Tag and *\$RegionNumber* the number of region.

FiberInt Section

This is a kind of Fiber Section model, which is exclusively used by Flexure-Shear Interaction Displacement-based Beam Column Elements and is based on **uniform** steel and concrete distribution. The section consists of three sub-sections with different thicknesses and widths. Each subsection may be associated to several strips. Strips consists of concrete and steel uniaxial materials and the location of each of them is defined by the fiber coordinates. **Concrete and steel areas inside each strip are located at one same point.** The Interface uses the center of area of concrete by default so far. Finally, horizontal steel reinforcement is included, assumed uniformly distributed in the section, represented by one only fiber.

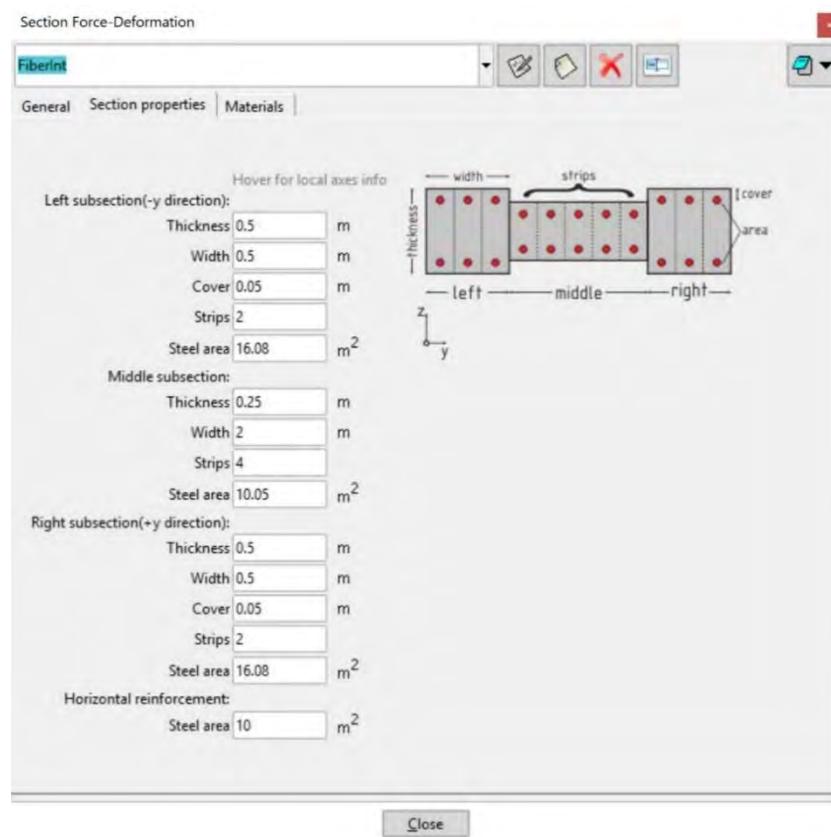


Figure 23: FiberInt Section model options

User may enter the dimensions of each sub-section, from which the concrete area is calculated, as well as the steel area. Cover depth can be given only for the left and right sub-section. Finally, user is asked to enter the total Horizontal steel area of reinforcement.

Plate Fiber

This is a section model that uses a pre-defined multi-dimensional material which numerically integrates through the plate thickness with fibers. It is suitable for plate and shell analysis. The only essential parameter it takes is the section thickness.

Elastic Membrane Plate

This is a section model, which numerically integrates through the plate thickness with fibers and is also suitable for plate and shell analysis.

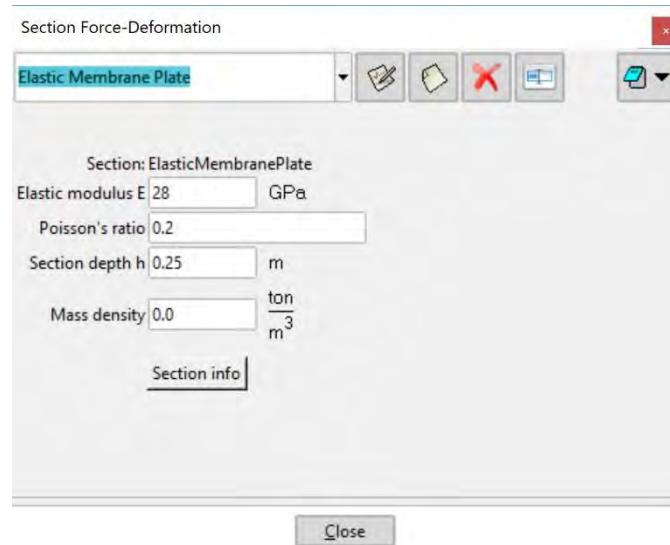


Figure 24: Elastic Membrane Plate section window

The parameters it takes are Young's modulus, Poisson's ratio, thickness of the section and the mass density.

LayeredShell

The LayeredShell section model is implemented by Shell elements (ShellMITC4 or ShellDKGQ) for simulating a multi-layer element, which is based on the principles of composite material mechanics. In the Interface the number of layers is **fixed to ten** and are characterized by different thicknesses and different material properties. It is appropriate and capable of simulating coupled in-plane/out-of-plane bending as well as in-plane direct shear and coupled bending/shear behavior of reinforced concrete shear walls.

Concrete is assumed to be in planar stress. Cracks are assumed to form when principal tensile stress exceeds the specified concrete tensile strength, and as a result concrete is considered as an orthotropic material.

The reinforcing bars are simulated as smeared steel layers of equivalent thicknesses, derived from a user-defined uniaxial steel material for rebars in conjunction with their corresponding angles. [3]

Layers are shown below from the outside to the inside layer (symmetric).

- 2 cover concrete layers (one for each side)
- 2 transverse rebar layers (one for each side)
- 2 longitudinal rebar layers (one for each side)
- 4 core concrete layers (middle layers)

The user is asked to enter the cross-sectional properties (height, total thickness and width) of the part (e.g. wall web or wall end column) including its reinforcement in order to determine the equivalent thicknesses for the rebar layers. Cover layer thickness is the real cover depth entered by the user. Longitudinal rebar layer thickness is resulted from the total longitudinal rebar area divided by two (two layers) and then divided by the **width**. Likewise, transverse rebar layer thickness is calculated the total transverse rebar area divided by two and then divided by the **height**. Total transverse rebar area is determined by the reinforcement space along the height, the rebar size (diameter) and the Height as follows:

$$\text{Transverse rebar layer thickness} = \left(\text{int}\left(\frac{\text{Height}}{\text{space}}\right) + 1 \right) * \frac{\text{rebarArea}}{\text{Height}}$$

Where:

Int: implies the integer part of the division

Height: the total height of the structural member

For instance, a single RC shear wall, which consist of two end-columns and a wall web (different reinforcement) would be simulated with 3 surfaces as shown in the following Figure.

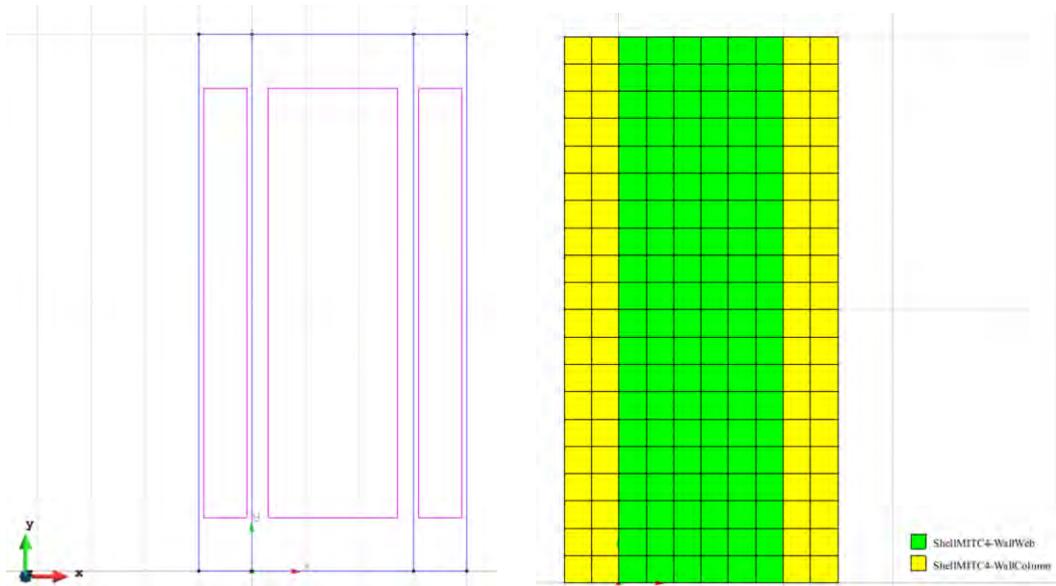


Figure 25: Geometry model with 3 surfaces (left); Meshed model with 2 different shell elements using 2 different LayeredShell section models (right)

Steel Uniaxial Materials

Some of the steel uniaxial materials as you will see, include isotropic hardening parameters. Their default hardening behavior is set to kinematic type. It is important to understand the difference between these two behaviors.

Isotropic hardening is characterized by the transfer of the hardening in tension mode to the compression one. The yield stress at point B' (in compression) depends on the stress at point B (in tension) and as a result the yield envelope can be increased.

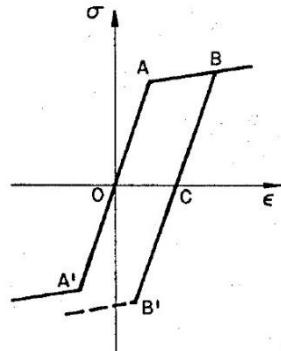


Figure 26: Isotropic hardening

Kinematic hardening is characterized by the **constant elastic range** and as a result the yield envelope is constant and its slope equal to the hardening ratio (Plastic tangent divided by the Elastic tangent).

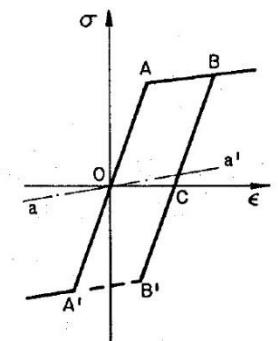


Figure 27: Kinematic hardening

Steel01

This is a uniaxial bilinear steel material with kinematic hardening behavior and optional isotropic hardening behavior determined by four isotropic parameters. The elastic range remains constant until a specific strain/deformation Fy/E_0 and the strain hardening is a linear function of the increment of plastic strain/deformation.

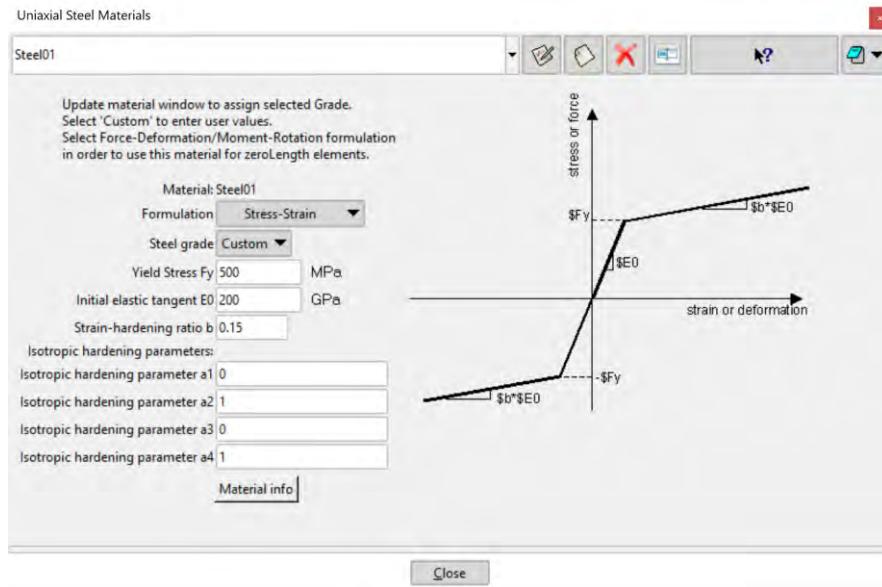


Figure 28: Steel01 material options

Steel01 material can be used in modelling reinforced concrete structures as the reinforcing bar material for Fiber elements' sections. It is also possible to be used for zero length elements, thus the formulation should be *Force-Deformation* or *Moment-Rotation* in order to avoid logical error from the units' conversion.

Isotropic parameters a_1 and a_2 correspond to compression. On the other hand, isotropic parameters a_3 and a_4 correspond to tension.

a_1 parameter is the coefficient of the product $a_2 \cdot \frac{F_y}{E_0}$ for the increase or decrease of the compression yield envelope. Positive value of a_1 corresponds to increase and negative value to decrease.

a_3 parameter is the coefficient of the product $a_4 \cdot \frac{F_y}{E_0}$ for the increase or decrease of the tension yield envelope. Positive value of a_3 corresponds to increase and negative value to decrease.

Material Properties	Default values
Yield stress Fy	500 MPa
Initial Elastic tangent E0	200 GPa
Strain-hardening ratio b	0.15
Isotropic hardening parameter a1	0
Isotropic hardening parameter a2	1

Isotropic hardening parameter a3	0
Isotropic hardening parameter a4	1

Steel02

This is a uniaxial steel material model with kinematic hardening proposed by Menegotto and Pinto (1993), combined with the isotropic hardening rules proposed by Filippou et al. (1983). This material also contains parameters to control the transition from elastic to plastic ranges. It is also possible to be used for zero length elements, thus the formulation should be *Force-Deformation* or *Moment-Rotation* in order to avoid logical error from units' conversion.

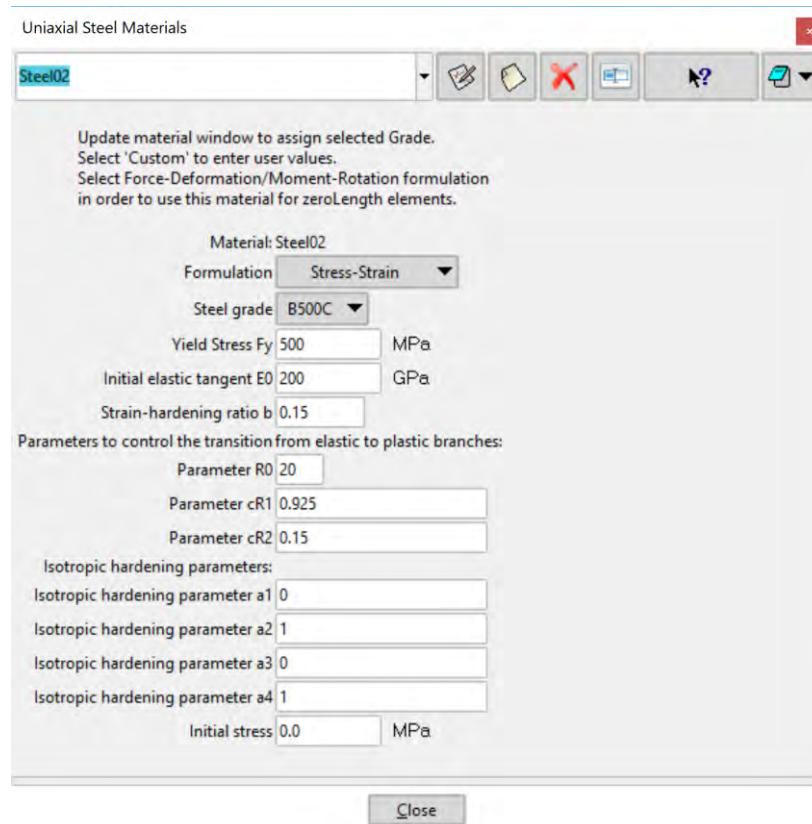


Figure 29 Steel02 material

Recommended parameters to control the transition from elastic to plastic ranges:

R_0 between 10 and 20

$cR_1 = 0.925$

$cR_2 = 0.15$

Isotropic parameters a_1 and a_2 correspond to compression. On the other hand, isotropic parameters a_3 and a_4 correspond to tension.

$a1$ parameter is the coefficient of the product $a2 \cdot \frac{Fy}{E0}$ for the increase or decrease of the compression yield envelope. Positive value of $a1$ corresponds to increase and negative value to decrease.

$a3$ parameter is the coefficient of the product $a4 \cdot \frac{Fy}{E0}$ for the increase or decrease of the tension yield envelope. Positive value of $a3$ corresponds to increase and negative value to decrease.

Zero values of the parameters $a1$ and $a3$ indicate a totally isotropic behavior.

Material Properties	Default values
Yield stress Fy	500 MPa
Initial Elastic tangent E0	200 GPa
Strain-hardening ratio b	0.15
Parameter R0	20
Parameter cR1	0.925
Parameter cR2	0.15
Isotropic hardening parameter a1	0
Isotropic hardening parameter a2	1
Isotropic hardening parameter a3	0
Isotropic hardening parameter a4	1

Reinforcing Steel

This is a steel uniaxial material based on the Chang and Mander (1994) model, which is intended to be used in a reinforced concrete fiber section for the steel reinforcing bar material.

It is not used by zero length elements.

Elastic range remains constant until strain is equal to $\frac{f_y}{E_s}$

Where:

F_y : Yield stress

E_s : Elastic tangent (Young's modulus)

After that, perfectly plastic region (yield plateau) takes place until strain is equal to ϵ_{sh} , where strain hardening behavior is performed. At strain ϵ_{su} the peak stress is reached before the degradation of strength until material failure is performed.

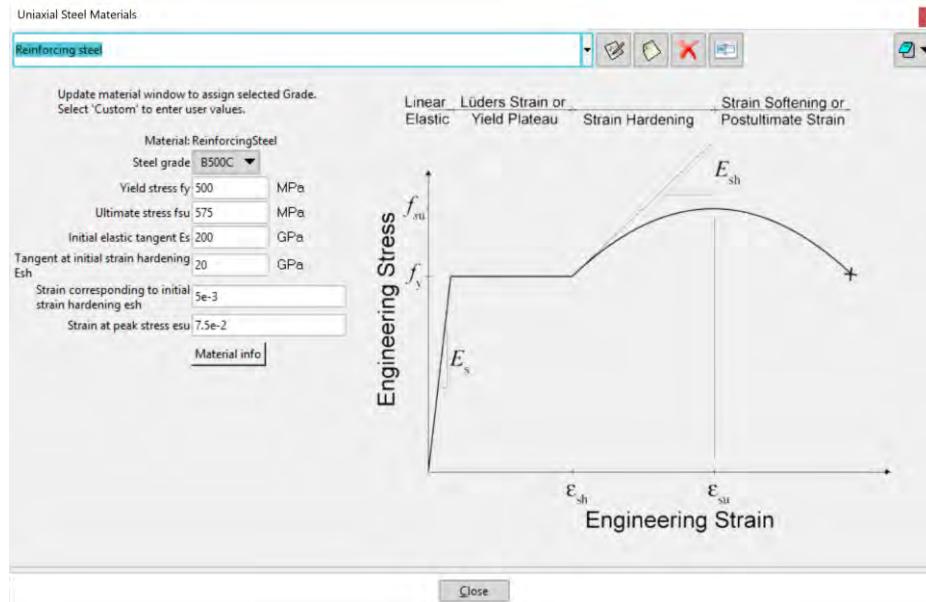


Figure 30: Reinforcing Steel material options

Material Properties	Default values
Yield stress fy	500 MPa
Ultimate stress fsu	575 MPa
Initial elastic tangent Es	200 GPa
Tangent at initial strain hardening Esh	20 GPa
Strain corresponding to initial strain hardening esh	0.005
Strain at peak stress esu	0.075

Hysteretic

This is a uniaxial bilinear hysteretic material with **pinching** of force and deformation, damage due to ductility and energy, and degraded unloading stiffness based on ductility. If you want to develop a thorough knowledge of the parameters of the hysteretic material it is recommended to experiment with them by simulating simple stress-strain analyses using displacement control, for instance using a truss or zero length element. Although more information can be accessed in the following link: <http://opensees.berkeley.edu/OpenSees/manuals/usermanual/4052.htm>,

where the hysteretic behaviour is observed for various parameter values and given strain histories imposed.

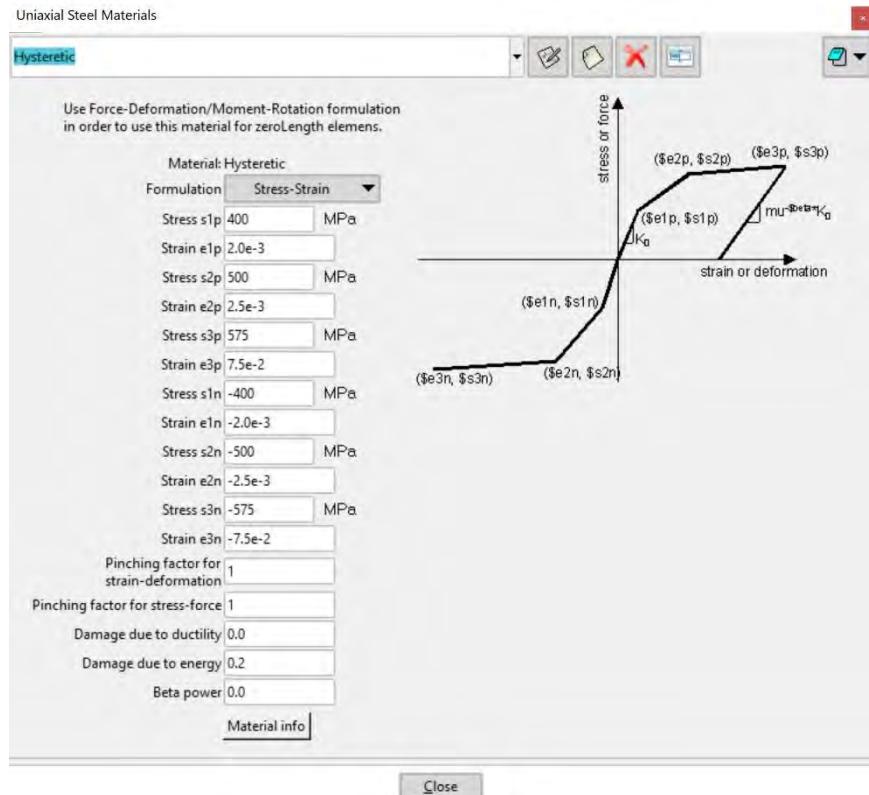


Figure 31: Hysteretic material options

The three-point stress-strain or force-deformation values of the envelope in each direction are user-defined. It is also possible to be used for zero length elements, thus the formulation should be Force-Deformation or Moment-Rotation in order to avoid logical error from the units' conversion.

Note: If $s_{3p} > s_{2p}$ and $|s_{3n}| > |s_{2n}|$ the envelope after e_{3p} or s_{3p} is a flat line with constant stress/force equal to e_{3p} or s_{3p} . Otherwise the envelope follows the slope defined by the second and third point of envelope.

Material Properties	Default values
Stress s1	400 MPa
Strain e1	2.0e-3
Stress s2	500 MPa
Strain e2	2.5e-3

Stress s3	575 MPa
Strain e3	7.5e-2
Pinching factor for strain/deformation	1
Pinching factor stress/force	1
Damage due to ductility	0.0
Damage due to energy	0.2
Beta power	0.0

Ramberg-Osgood

This is a uniaxial steel material, using the Ramberg-Osgood functions, which are usually used to evaluate the behavior of structural steel materials and components.

The Ramberg-Osgood function is expressed as:

$$\varepsilon = \frac{\sigma}{E_0} + \alpha \cdot \left(\frac{\sigma}{\sigma_0} \right)^n$$

Where E_0 is the initial elastic modulus and σ_0 is equal to $E_0 \cdot \varepsilon_0$

ε_0 is an arbitrary strain, for which the plastic component of strain is not zero. The yield strain is used for normalization of strain.

The α factor determines the yield offset, because of the following equation.

$$\varepsilon = (1 + \alpha) \cdot \frac{\sigma_0}{E}$$

Exponent n parameters is used for controlling the transition from elastic to plastic branches as well as the hardening ratio. The usual value used are approximately 5 or greater.

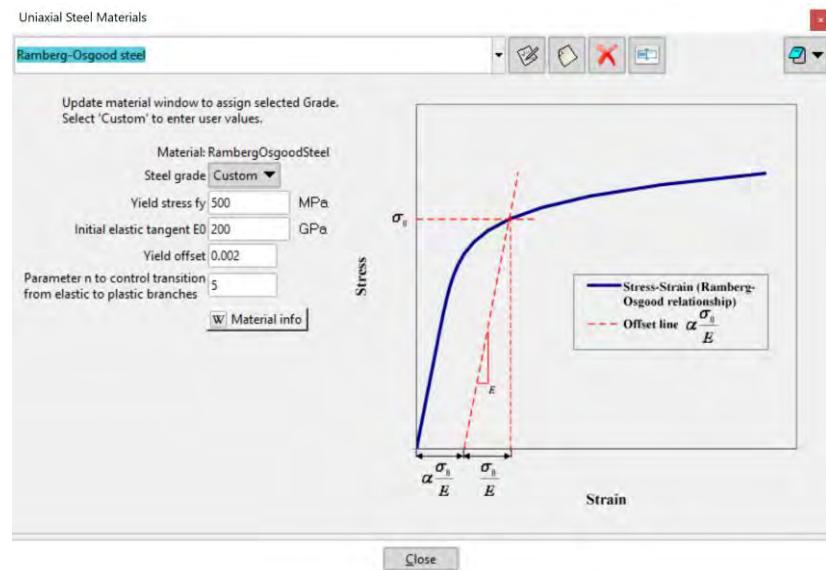


Figure 32: Ramberg-Osgood material options

Material Properties	Default values
Yield stress fy	500 MPa
Initial elastic tangent E0	200 GPa
Yield offset	0.002
Exponent n	2.5e-3

Concrete Uniaxial Materials

All concrete uniaxial material windows contain two special fields, *Strength class* and *Strength type*, for generating some fast properties of concrete. These fast-recommended values are mainly generated according to Eurocode 2 (EN-1992-1-1).

Table 3: Strength and deformation characteristics for concrete according to EN-1992-1-1

	Strength classes for concrete														Analytical relation / Explanation
f_{ck} (MPa)	12	16	20	25	30	35	40	45	50	55	60	70	80	90	
$f_{ck,cube}$ (MPa)	15	20	25	30	37	45	50	55	60	67	75	85	95	105	
f_{cm} (MPa)	20	24	28	33	38	43	48	53	58	63	68	78	88	98	$f_{cm} = f_{ck} + 8 \text{ (MPa)}$
f_{ctm} (MPa)	1,6	1,9	2,2	2,6	2,9	3,2	3,5	3,8	4,1	4,2	4,4	4,6	4,8	5,0	$f_{ctm} = 0,30 \cdot f_{ck}^{(2)} \leq C50/60$ $f_{ctm} = 2,12 \cdot \ln(1 + (f_{cm}/10))$ $> C50/60$
$f_{ctk,0,05}$ (MPa)	1,1	1,3	1,5	1,8	2,0	2,2	2,5	2,7	2,9	3,0	3,1	3,2	3,4	3,5	$f_{ctk,0,05} = 0,7 \times f_{ctm}$ 5% fractile
$f_{ctk,0,95}$ (MPa)	2,0	2,5	2,9	3,3	3,8	4,2	4,6	4,9	5,3	5,5	5,7	6,0	6,3	6,6	$f_{ctk,0,95} = 1,3 \times f_{ctm}$ 95% fractile
E_{cm} (GPa)	27	29	30	31	33	34	35	36	37	38	39	41	42	44	$E_{cm} = 22(f_{cm}/10)^{0,3}$ (f_{cm} in MPa)
ε_{c1} (%)	1,8	1,9	2,0	2,1	2,2	2,25	2,3	2,4	2,45	2,5	2,6	2,7	2,8	2,8	see Figure 3.2 $\varepsilon_{c1}^{(0)} = 0,7 \cdot f_{cm}^{0,31} < 2,8$
ε_{cu1} (%)						3,5				3,2	3,0	2,8	2,8	2,8	see Figure 3.2 for $f_{ck} \geq 50 \text{ MPa}$ $\varepsilon_{cu1}^{(0)} = 2,8 + 27[(98 - f_{cm})/100]^4$
ε_{c2} (%)						2,0				2,2	2,3	2,4	2,5	2,6	see Figure 3.3 for $f_{ck} \geq 50 \text{ MPa}$ $\varepsilon_{c2}^{(0)} = 2,0 + 0,085(f_{ck} - 50)^{0,63}$
ε_{cu2} (%)						3,5				3,1	2,9	2,7	2,6	2,6	see Figure 3.3 for $f_{ck} \geq 50 \text{ MPa}$ $\varepsilon_{cu2}^{(0)} = 2,6 + 35[(90 - f_{ck})/100]^4$
n						2,0				1,75	1,6	1,45	1,4	1,4	for $f_{ck} \geq 50 \text{ MPa}$ $n = 1,4 + 23,4[(90 - f_{ck})/100]^4$
ε_{c3} (%)						1,75				1,8	1,9	2,0	2,2	2,3	see Figure 3.4 for $f_{ck} \geq 50 \text{ MPa}$ $\varepsilon_{c3}^{(0)} = 1,75 + 0,55[(f_{ck} - 50)/40]$
ε_{cu3} (%)						3,5				3,1	2,9	2,7	2,6	2,6	see Figure 3.4 for $f_{ck} \geq 50 \text{ MPa}$ $\varepsilon_{cu3}^{(0)} = 2,6 + 35[(90 - f_{ck})/100]^4$

Strength class can be chosen from C12/15 to C90/105 or *Custom*. For user modifications after generating fast values, select *Custom* class.

Strength type can be either *Characteristic* or *Mean*, if a specific strength class is selected. Otherwise this field is not visible.

Concrete01

This is a uniaxial concrete material based on Kent-Park envelope with degraded linear unloading/reloading stiffness according to the work of Karsan-Jirsa and with **zero tensile strength**. When updating the window and a particular class is selected, fpc, epsc0, epscU values are updated according to Eurocode 2. Crushing strength fpcu is calculated as the 85% of fpc.

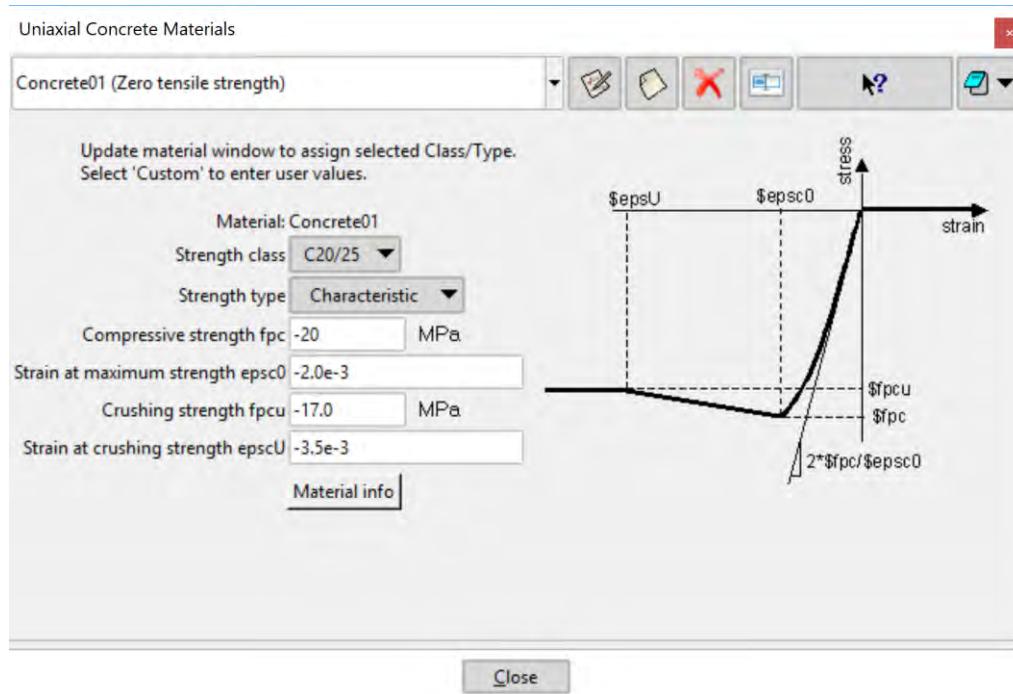


Figure 33: Concrete01 material options

Material Properties	Default values
Compressive strength fpc	-20 MPa
Strain at maximum strength epsc0	-0.002
Crushing strength fpcu	-17 MPa
Strain at crushing strength epscU	-0.0035

Concrete02

This is a uniaxial concrete material with **linear tension softening defined by a tangent E_{ts}** .

When updating the material window and a particular class is selected, compressive strength and the corresponding strain strepsc0, the tensile strength f_t , as well as the strain at crushing strength epscU values are updated according to Eurocode 2.

Crushing strength f_{pcu} is calculated as the 85% of f_{pc} .

Tension softening stiffness E_{ts} is calculated as $E_{ts} = \frac{2f_{pc}}{epsc0} * \frac{1}{10} = Eo/10$.

Ratio between unloading slope at epscU and initial slope, ***lambda* is always user-defined**.

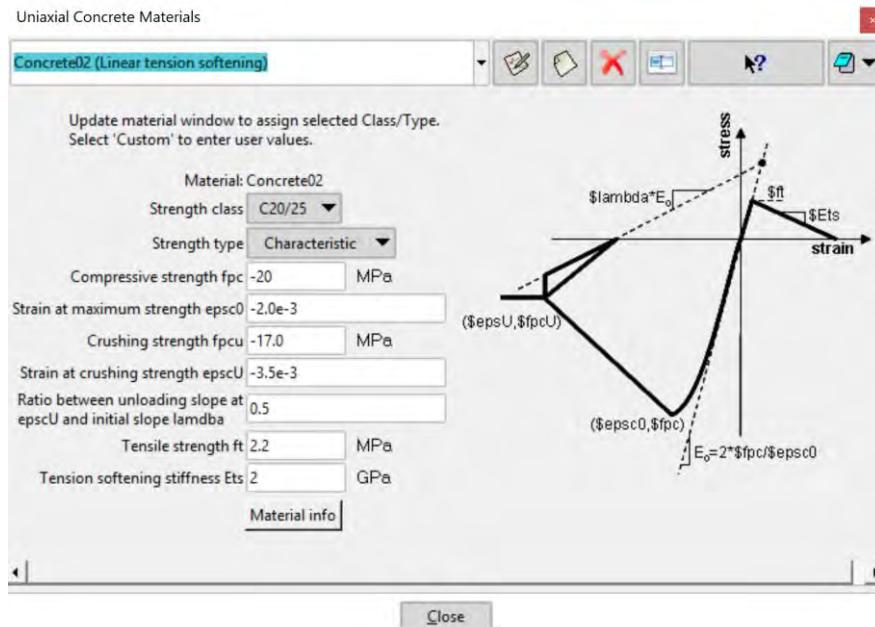


Figure 34 Concrete02 material

Material Properties	Default values
Compressive strength fpc	-20 MPa
Strain at maximum strength epsc0	-0.002
Crushing strength fpcu	-17 MPa
Strain at crushing strength epscU	-0.0035
Ratio between unloading slope at epscU and initial slope, lambda	0.5
Tensile strength ft	2.2 MPa
Tension softening stiffness Ets	2 GPa

Concrete04

This is a uniaxial Popovics concrete material with **degraded linear unloading/reloading stiffness** according to the work of Karsan-Jirsa and tensile strength with exponential decay.

When updating the material window and a specific class is selected, compressive strength, and the corresponding strain, strain at crushing strength, initial stiffness and maximum tensile strength are updated according to table 3. However, ultimate tensile strain is not changed and remains user-defined.

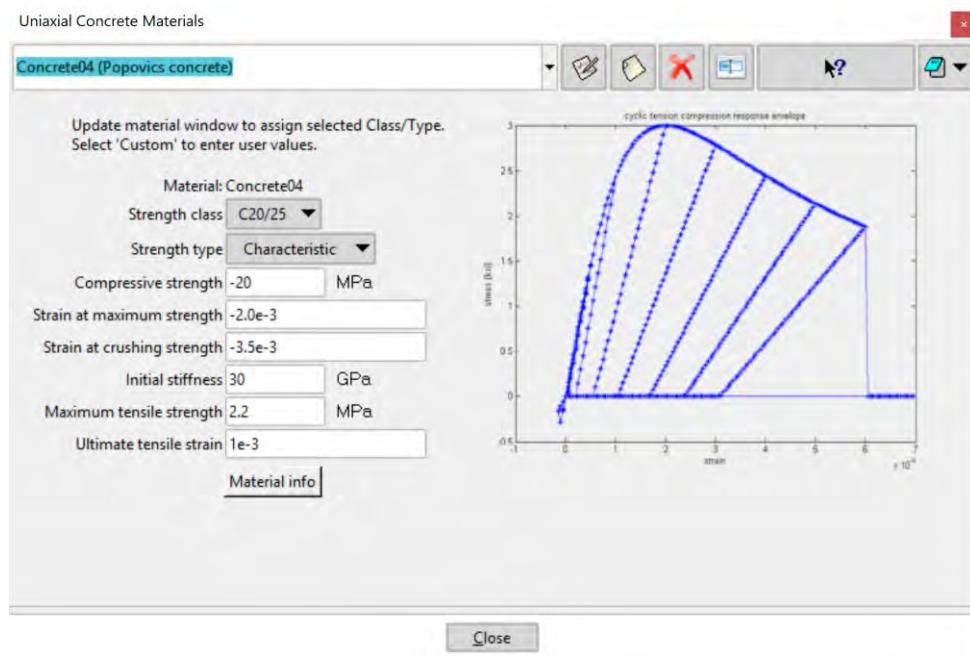


Figure 35 Concrete04 material options

Material Properties	Default values
Compressive strength	-20 MPa
Strain at maximum strength	-0.002
Strain at crushing strength	-0.0035
Initial stiffness	30 GPa
Maximum tensile strength	2.2 MPa
Ultimate tensile strain	0.003

Concrete06

This is a uniaxial concrete material with tensile strength, nonlinear tension softening, and compressive behavior based on Thorenfeldt curve, which is similar to Popovics (1973) definition. It appropriate for providing an accurate simulation of concrete structures subjected to reversed loading. The backbone curve for the compression response follows the monotonic response that is based on Popovics formulation and includes softening effects in compression according Modified Compression Field Theory.

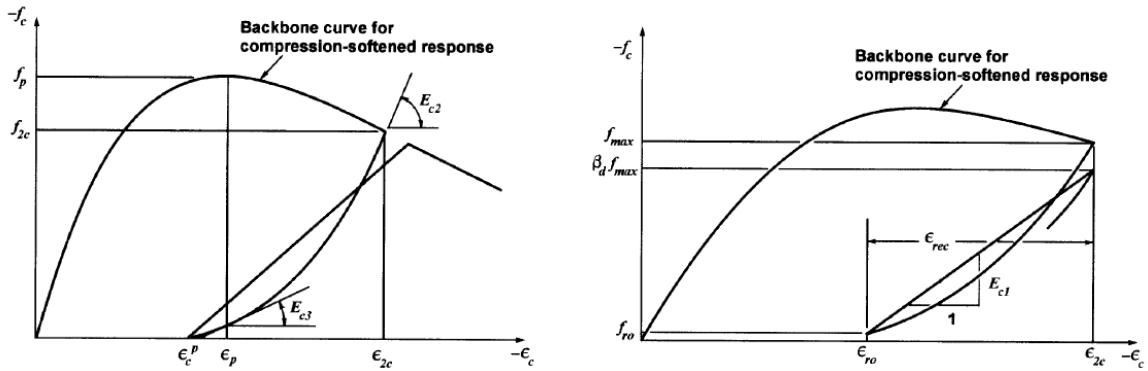


Figure 36: Hysteresis models for concrete06 in compression for unloading (left) and reloading (right)

The backbone curve for the tension response is shifted such that its origin coincides with the compressive plastic offset strain. [4]

When updating the material window and a strength class is selected, compressive strength f_c , strain at compressive strength e_0 and tensile strength f_{cr} are updated according to table 3.

Parameter $a1$ is calculated as:

$$a1 = ecu1 - \frac{f_c}{2 \cdot \frac{f_c}{e_0}}$$

where $ecu1$ is strain at crushing strength according to EC2.

Parameter $a2$ is calculated as:

$$a2 = 7 \cdot ecr$$

where ecr is calculated as:

$$ecr = \frac{f_{cr}}{2 \cdot \frac{f_c}{e_0}}$$

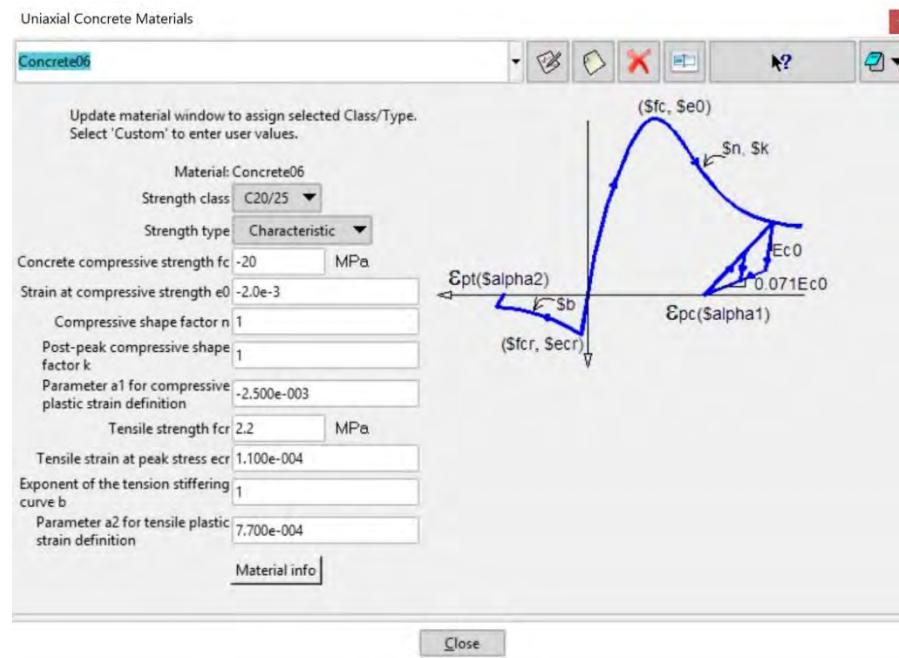


Figure 37: Concrete06 material options

Material Properties	Default values
Compressive strength f_c	-20 MPa
Strain at compressive strength e_0	-0.002
Compressive shape factor n	1
Post-peak compressive shape factor k	1
Parameter for compressive plastic strain definition a_1	-0.0025
Tensile strength f_{cr}	2.2 MPa
Tensile strain at peak stress e_{cr}	0.00011
Exponent of the tension stiffening curve b	1
Parameter for tensile plastic strain definition	0.00077

ConcreteCM

This is a uniaxial concrete material representing the hysteretic constitutive model for concrete proposed by Chang and Mander (1994). This model allows the user to simulate the monotonic

and/or hysteretic behavior of confined and unconfined concrete in both tension and compression states. It features interesting behavioral parameters including continuous hysteretic behavior in cyclic tension and compression, dynamic degradation combined with smooth unloading/reloading curves at developing strains, as well as gradual crack closure effects.

The tension envelope curve shape is similar to compression's one, but its origin is dependent on the unloading strain from the compression. Note also that, the strain ductility performed on the compression is reflected on the tension envelope.

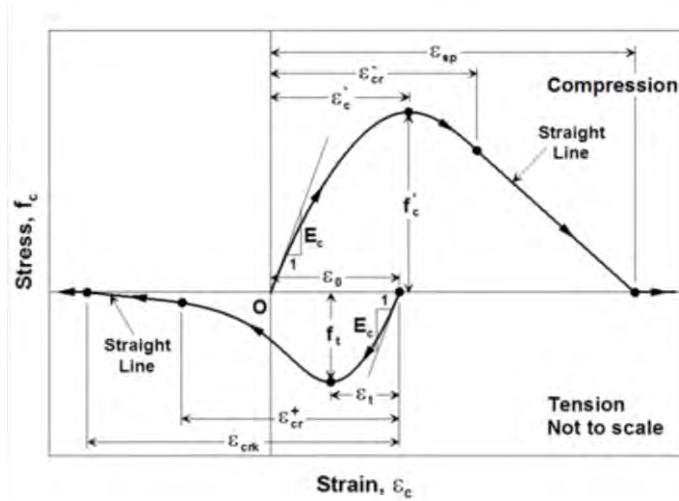


Figure 38: ConcreteCM envelope curves

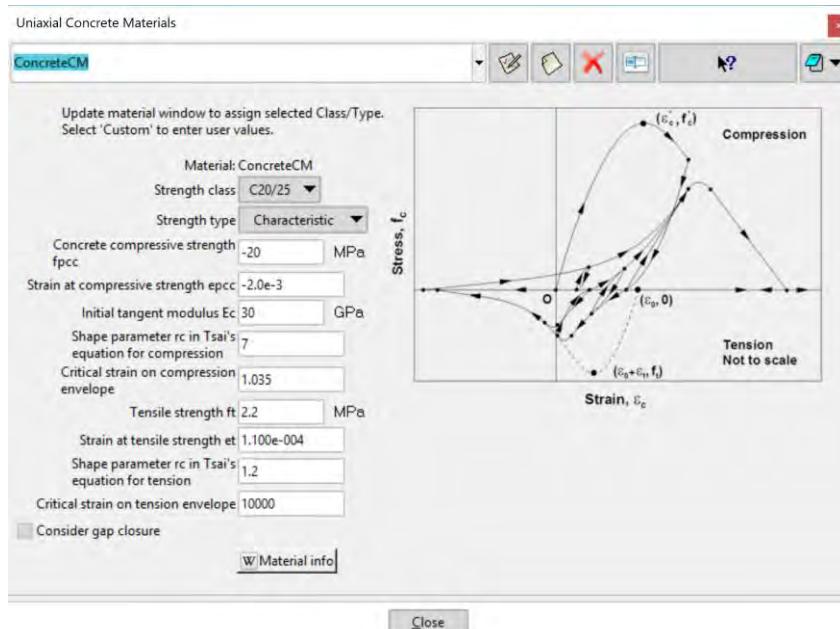


Figure 39: ConcreteCM material options

The original Chang and Mander (1994) proposed model is simulated by giving the gap parameter the value 1, which represents a non-zero tangent stiffness at zero stress level upon unloading from the tension envelope. Setting gap parameter equal to zero (default value) results a zero tangent stiffness at zero stress level upon unloading from the tension envelope.

When updating the material window and a specific class is chosen, compressive strength, and the corresponding strain (ec1), initial tangent modulus and maximum tensile strength are updated according to Eurocode 2. However, ultimate tensile strain is not changed and remains user-specified.

Material Properties	Default values
Compressive strength fpcc	-20 MPa
Strain at compressive strength epcc	-0.002
Initial tangent modulus Ec	30 GPa
Shape parameter rc in Tsai's equation for compression	7
Critical strain on compression envelope	1.035
Tensile strength ft	2.2 MPa
Strain at tensile strength et	0.00011
Shape parameter rc in Tsai's equation for tension	1.2
Critical strain on tension envelope	10000
Gap parameter	0

Other Uniaxial Materials

Elastic

This is a simple uniaxial elastic material model with symmetric behavior in tension and compression. It can also be used for zero length elements. If so, a *Force-deformation* or *Moment-Rotation* formulation is recommended through the corresponding field.

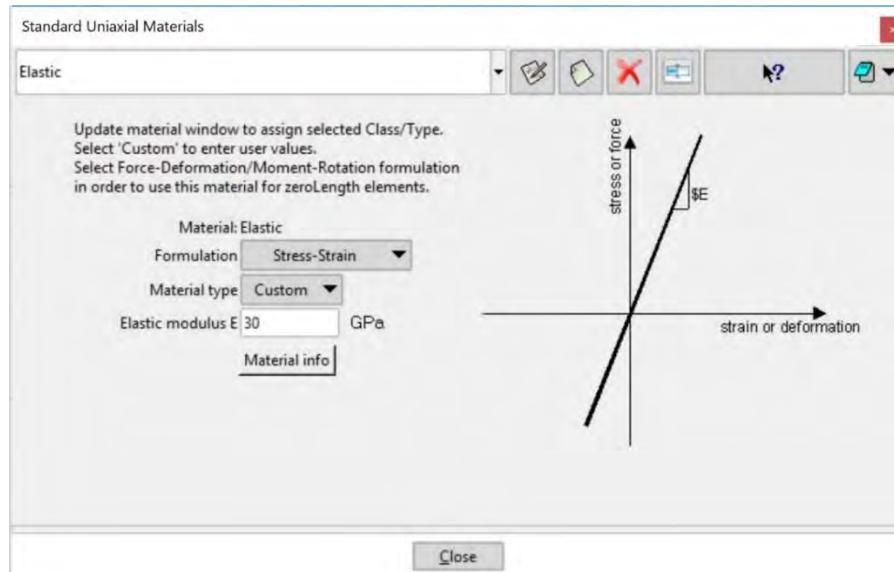


Figure 40: Elastic material options

This material takes only one parameter: Elastic modulus or Stiffness or Moment per rotation unit, depending on the selected formulation.

If *stress-strain* formulation is chosen, user can select material type as concrete, steel or custom. For concrete material type, user can choose the concrete strength class. Thus, updating the material window, the Elastic modulus is changed according to Eurocode 2. For steel material type elastic modulus is updated to 200 GPa.

Elastic Perfectly Plastic

This is a uniaxial material, which consists of an elastic range and a perfectly plastic one. The behavior is not necessarily symmetric for tension and compression. It can also be used for zero length elements. If so, a *Force-deformation* or *Moment-rotation* formulation is recommended through the corresponding field.

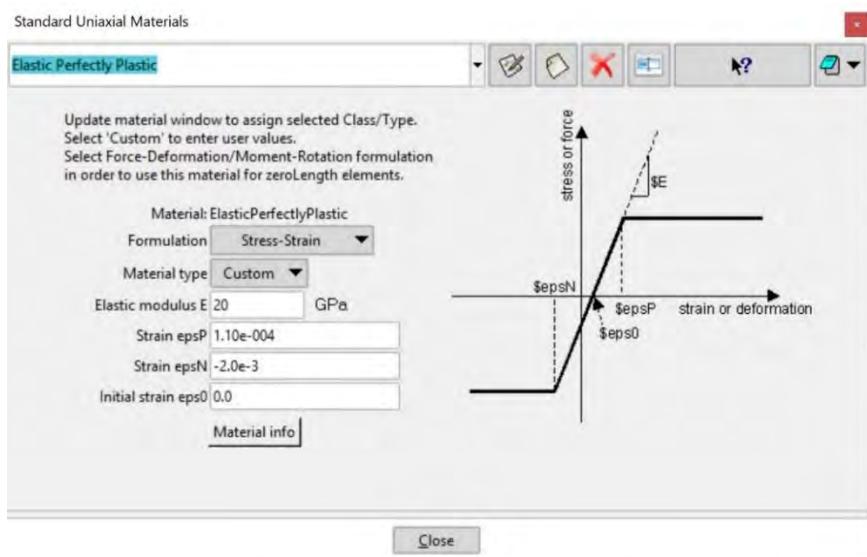


Figure 41 Elastic perfectly plastic material options

It takes three more parameters than Elastic material. Two strains/deformations/rotations ϵ_{sp} and ϵ_{sn} to define when the transition from elastic range to perfectly plastic one takes place in tension and compression respectively. Finally, an initial strain can be set.

When updating the material window and formulation is Stress-Strain, the properties are updated as follows:

For a particular concrete class, elastic modulus E is calculated as:

$$E = 2 \cdot \frac{f_c}{\epsilon_{sn}}$$

Where:

ϵ_{sn} gets the value of $ec1$ according to Eurocode 2.

Strain ϵ_{sp} is also calculated as:

$$\epsilon_{sp} = \frac{f_{ctm}}{E}$$

where f_{ctm} is the tension strength according to Eurocode 2.

For a particular steel grade, elastic modulus E is updated to 200 GPa and, as a result, Yield stress F_y depends on the steel grade and strains ϵ_{sp} and ϵ_{sn} are calculated as $\pm \frac{F_y}{E}$, respectively.

Material Properties	Default values
Elastic modulus E	20 GPa

Strain epsP	0.00011
Strain epsN	-0.002
Initial strain esp0	0.0
Stiffness K	4000 kN/m
Deformation epsP	1 m
Deformation epsN	-1 m
Moment per rotation unit	10000
Rotation epsP	0.5 rad
Rotation epsN	-0.5 rad
Initial rotation esp0	0.0 rad

Elastic Perfectly Plastic with Gap

This is a uniaxial material, which consists of an elastic range and a perfectly plastic one, that works **only in tension or compression**. The sign of yield stress determines if it works in tension or compression. It can also be used for zero length elements. If so, a *Force-deformation* or *moment-rotation* formulation is recommended to be used through the corresponding field.

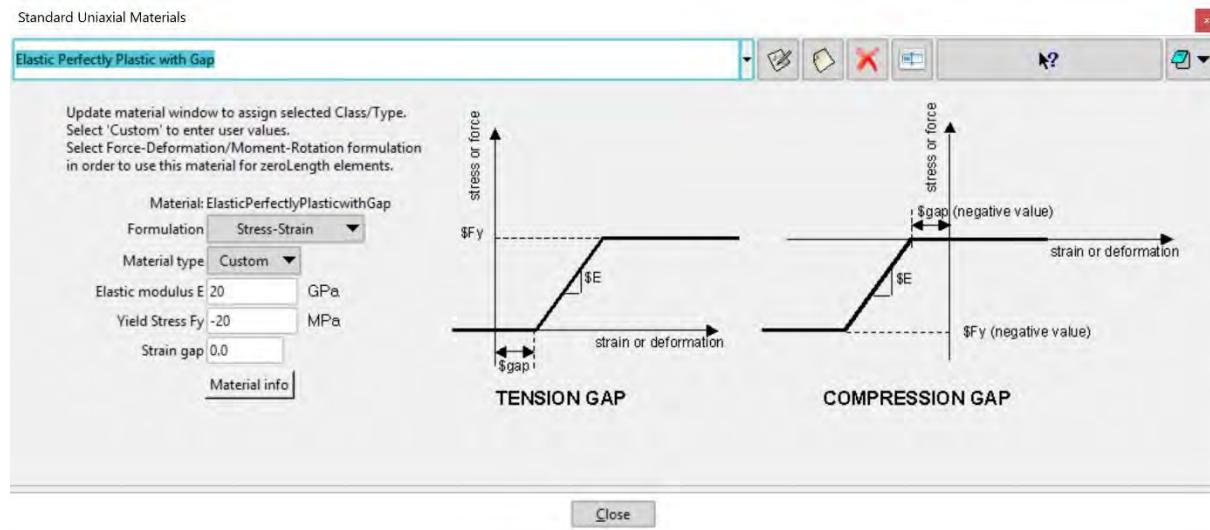


Figure 42 Elastic Perfectly plastic with gap material window

When updating the material window and formulation is *Stress-Strain*, the properties are updated as follows:

For a particular **concrete** class, elastic modulus E is calculated as

$$E = 2 \cdot \frac{fc}{ec1}$$

where strain $ec1$ is generated from Eurocode 2 and yield stress is changed to concrete strength fc .

For a particular **steel** grade, elastic modulus E is changed to 200 GPa and as a result, Yield stress Fy is updated depending on the steel grade selected.

Material Properties	Default values
Elastic modulus E	20 GPa
Yield stress Fy	-20 MPa
Strain gap	0.0
Stiffness K	4000 kN/m
Force Fy	1000 kN
Deformation gap	0.0 m
Moment per rotation unit	10000 kNm/rad
Moment My	150 kNm
Rotation gap	0.0 rad

Viscous

This is a uniaxial viscous material, which can be used directly from truss and zerolength elements or in combination with other materials through a series or parallel uniaxial material. It requires the input of the damping coefficient C and the α power. It should be clarified that small values of α power ($\alpha < 0.1$) changes the mechanism from viscous to Coulomb friction type ($\alpha \rightarrow 0$).

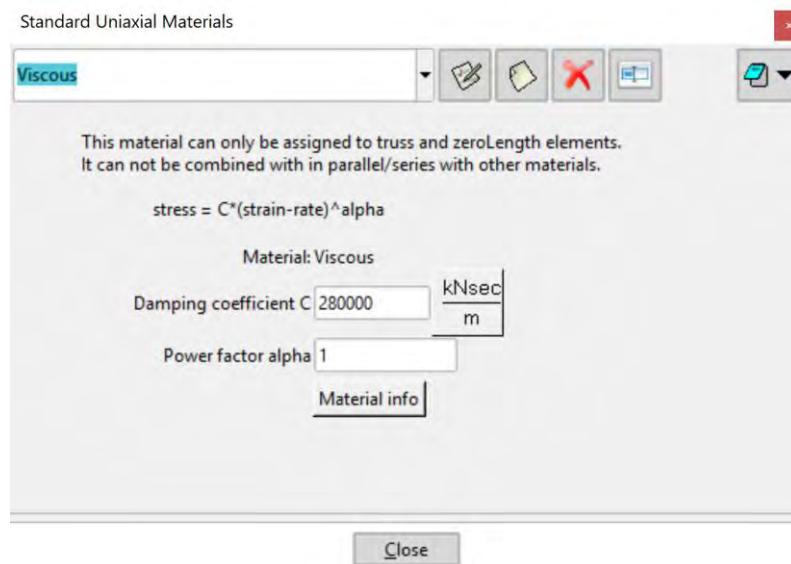


Figure 43: Viscous material options

In case that a viscous material is used for connecting two nodes through a zerolength element (or truss), the development of relative velocity between them is required to develop a viscous damping force.

Viscous Damper

This is a uniaxial material, which represents the Maxwell material, proposed by James Clerk Maxwell (1867), which is a viscoelastic material including the properties both of elasticity and viscosity and is suitable for simulating the hysteretic response of nonlinear viscous dampers.

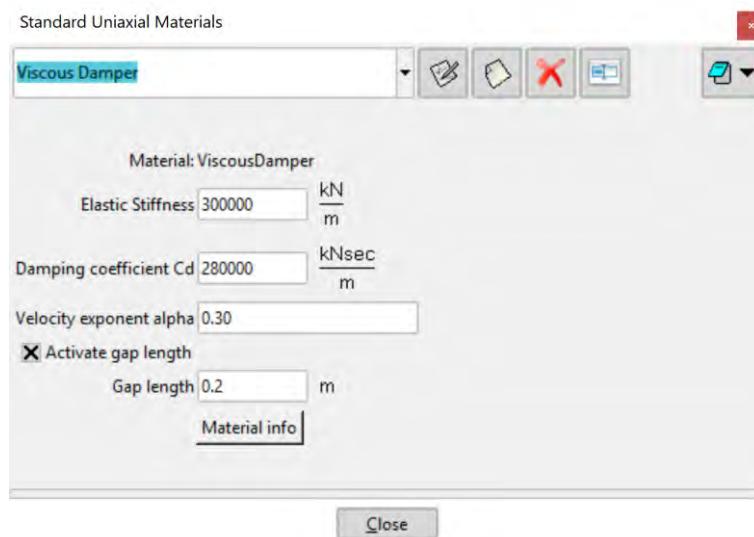


Figure 44: Viscous Damper material options

Material Properties	Default values
Elastic Stiffness	300000 kN/m
Damping coefficient Cd	280000 kNsec/m
Velocity exponent alpha	0.30
Gap length	0.02 m

Hyperbolic Gap

This is a uniaxial compression only gap material. It is based on abutment stiffness models for bridge simulation proposed by Patrick Wilson and Ahmed Elgamal at UCSD. The hyperbolic force-displacement model is based on work by Duncan and Mokwa (2001) and Shamsabadi et al. (2007) with calibrated parameters from UCSD abutment tests.

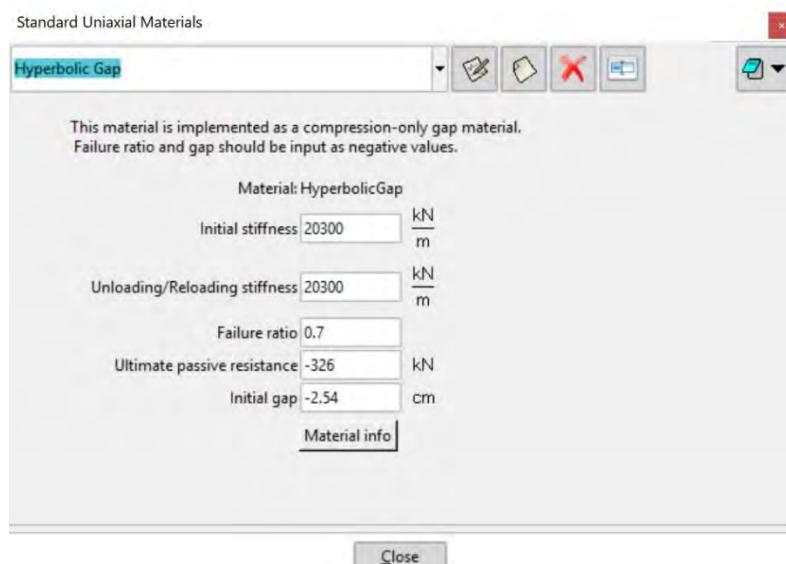


Figure 45: Hyperbolic Gap material options

The force is calculated from the following formula:

$$F(x) = \frac{x}{\frac{1}{K_{max}} + R_f \cdot \frac{x}{F_{ult}}}$$

Where:

$F(x)$ is the force at displacement x

K_{max} is the initial stiffness

F_{ult} is the ultimate passive resistance

Rf is the failure ratio

Note: Ultimate passive resistance and initial gap should be input as negative values

Material Properties	Default values
Initial Stiffness	20300 kN/m
Unloading/Reloading stiffness	20300 kN/m
Failure ratio	0.70
Ultimate passive resistance	-326 kN
Initial gap	-2.54 cm

PySimple1

This is a uniaxial spring material for implementing the p-y method of analyzing the ability of deep foundations to resist loads applied in the lateral direction. P-y curves relate the force applied to soil to the lateral deflection of the soil and hence non-linear springs (zeroLength elements) are attached to the foundation in place of the soil, greatly reducing the computational time at the same time.

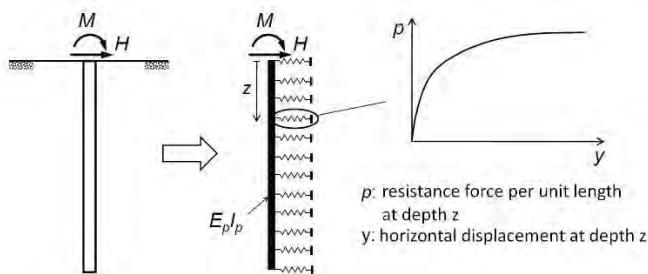


Figure 46: P-y springs

The p-y curves vary depending on soil type. Each one is characterized by an ultimate resistance force p_{ult} . The user is also asked to enter the displacement at which the 50% of p_{ult} is mobilized in monotonic loading, the coefficient C_d for drag resistance within a fully-mobilized gap ($C_d \cdot p_{ult}$) and the viscous damping coefficient for representing radiation damping effect. Finally, the user must select the type of p-y backbone curve between Soft clay and sand. Soft clay corresponds to Matlock (1970) relation and Sand corresponds to API (1993) relation.

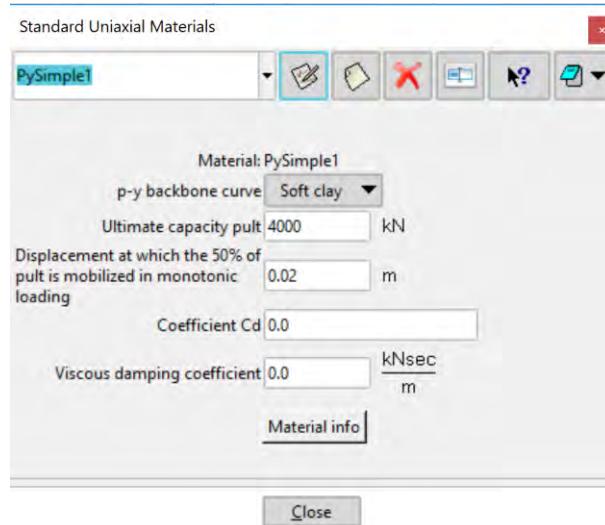


Figure 47: PySimple1 material options

TzSimple1

This is a uniaxial spring material for considering the t-z soil curves which relate the shear force applied to soil to the vertical deflection of the soil (soil skin friction along the shaft, Fig. 49) and hence non-linear springs (zeroLength elements) are attached to the foundation in place of the soil.

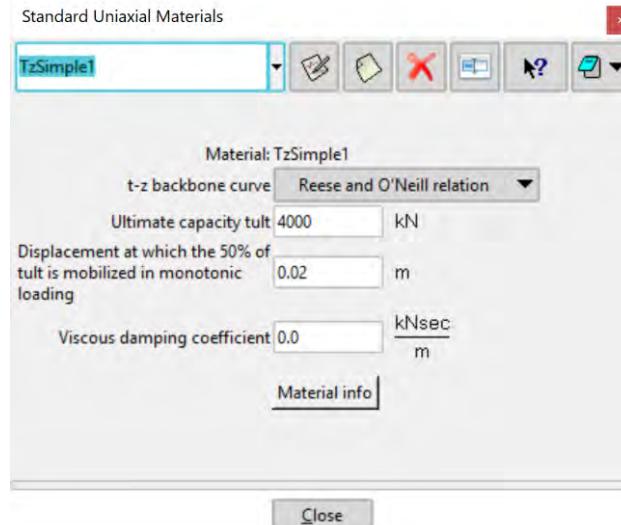


Figure 48: TzSimple1 material options

The t-z curves vary depending on soil type. Each one is characterized by an ultimate shear resistance force t_{ult} . The user is also asked to enter the displacement at which the 50% of t_{ult} is mobilized in monotonic loading and the viscous damping coefficient for representing radiation damping effects. Finally, the user must select the type of t-z backbone curve between Reese and O'Neill's relation (1987) and Mosher's relation.

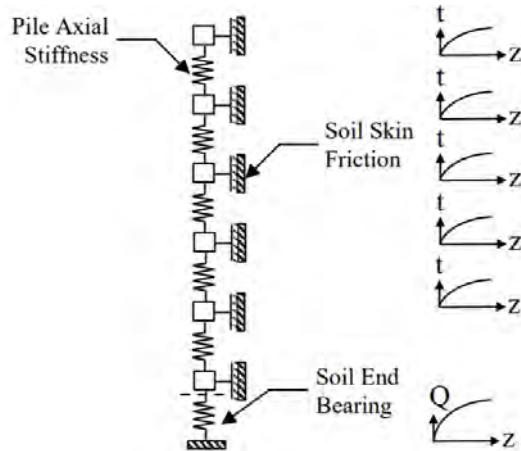


Figure 49: Spring model of load transfer mechanisms in an axially loaded pile

QzSimple1

This is a uniaxial spring material for considering the q - z soil curves which relate the force applied to soil to the vertical deflection of the soil for simulating soil end bearing resistance (Fig. 49) using zero length elements.

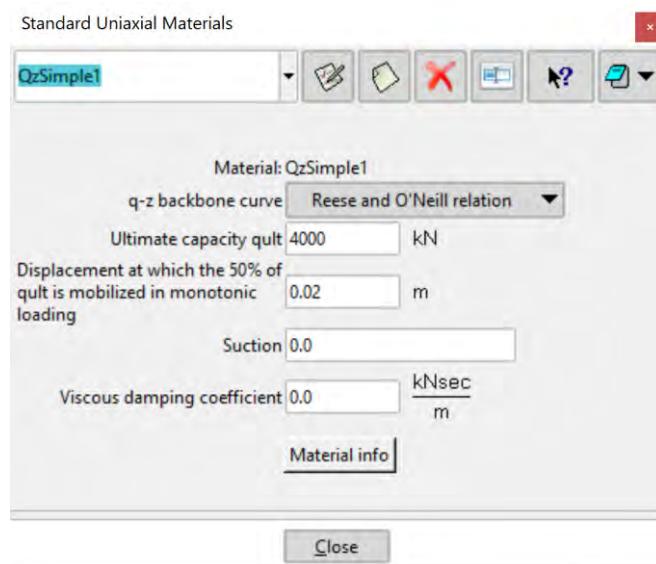


Figure 50: QzSimple1 material options

The t - z curves vary depending on soil type. Each one is characterized by an ultimate shear resistance force q_{ult} . The user is also asked to enter the displacement at which the 50% of q_{ult} is mobilized in monotonic loading, the viscous damping coefficient for representing radiation damping effects and the suction parameter for simulating the uplift resistance, which is equal to $suction \cdot q_{ult}$. Finally, the user must select the type of t - z backbone curve between Reese and O'Neill's relation (1987) and Vijayvergiya relation (1977).

The value of suction must be between 0.0 to 0.1.

Combined Materials

By mentioning combined materials, we mean either uniaxial materials or section models that determine Force-Deformation behavior, which use at least one other material or section model.

Series

This is a uniaxial material that is made up of a number of other uniaxial materials. At this moment, the Interface allows the user to use up to five uniaxial materials. **For the series material, stresses are equal, and strains and flexibilities are additive.**

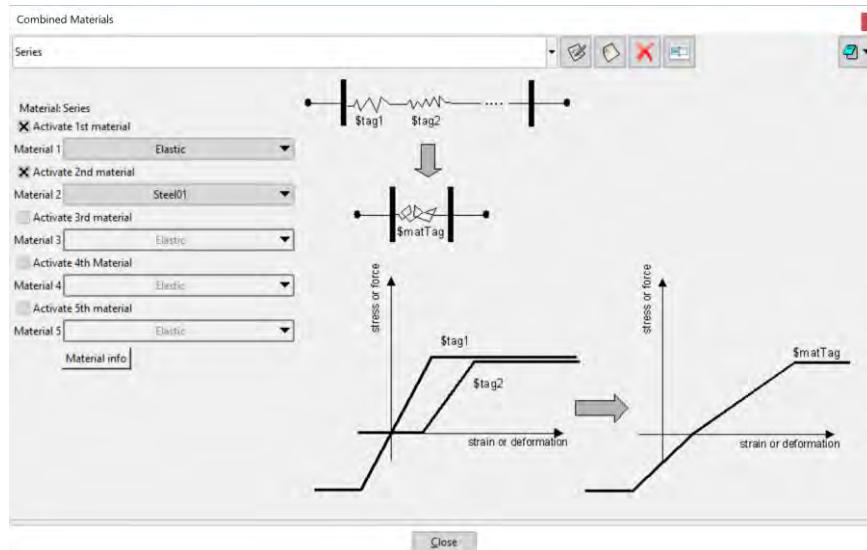


Figure 51: Series uniaxial material options

Parallel

This is a uniaxial material that is also made up of a number of other uniaxial materials. At this moment, the Interface allows you to use up to five individual uniaxial materials. **For a parallel material, strains are equal and stresses and stiffnesses are additive.**

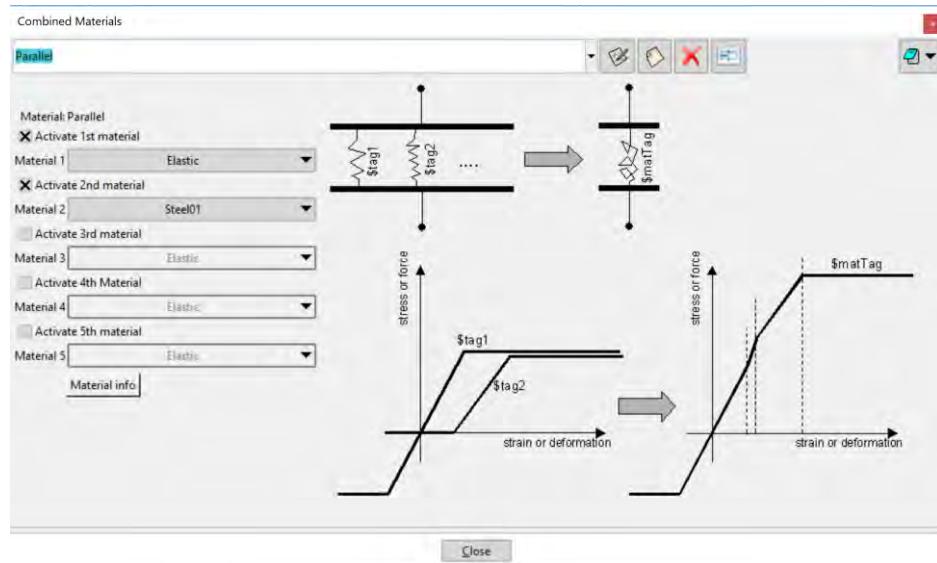


Figure 52: Parallel uniaxial material options

Section Aggregator

This is a section force-deformation model, which aggregates uniaxial materials into another section force-deformation model. Each uniaxial material that is activated and used, represents the section force-deformation response for a particular degree of freedom of the section and **no interaction takes place among them**.

Only Fiber section can be selected for the section F-D model. Moreover, all uniaxial materials except viscous materials and hyperbolic gap are available for the response of the section's degrees of freedom.

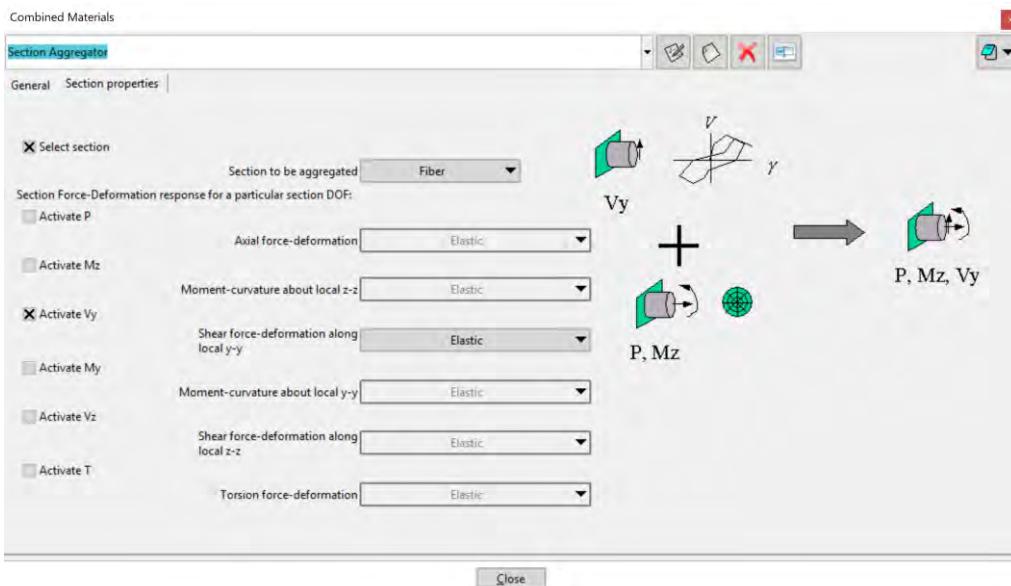


Figure 53: Section aggregator options

Initial Strain

This is a uniaxial material, which uses the stress-strain relationship of another uniaxial material, but it enables the definition of initial strains. The stress that corresponds to the initial strain is calculated from the other (selected) material.

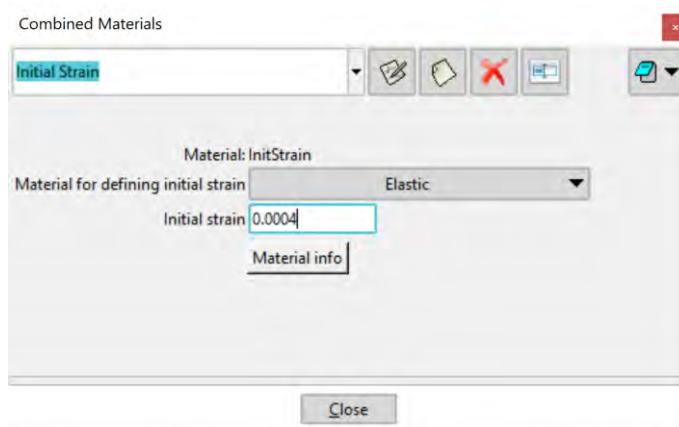


Figure 54: Initial strain options

Initial Stress

This is a uniaxial material, which uses the stress-strain relationship of another uniaxial material, but it enables the definition of initial stress. The strain which corresponds to the initial stress is calculated from the other (selected) material.

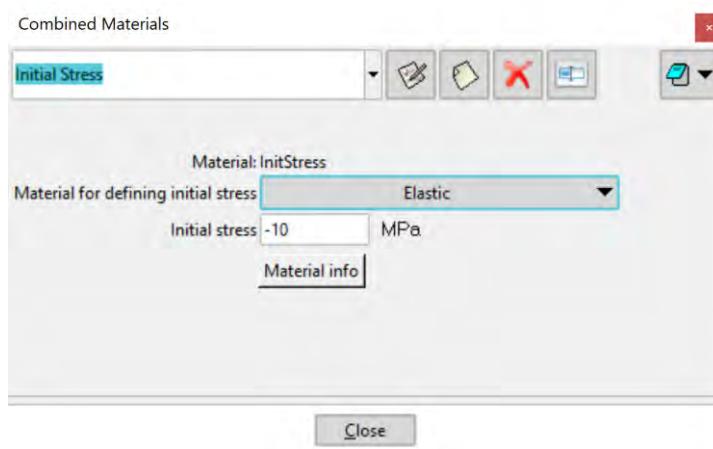


Figure 55: Initial stress options

MinMax

This is a uniaxial material, which uses another uniaxial material for defining its stress-strain/force-deformation/moment-rotation relationship with the addition of the determination of the upper and lower limit of the strain/deformation/rotation. So, if the strain/deformation/rotation

falls below or above the minimum or maximum values, the material is assumed to have failed, and as a result, zero values are returned for the tangent and stress/force/moment.

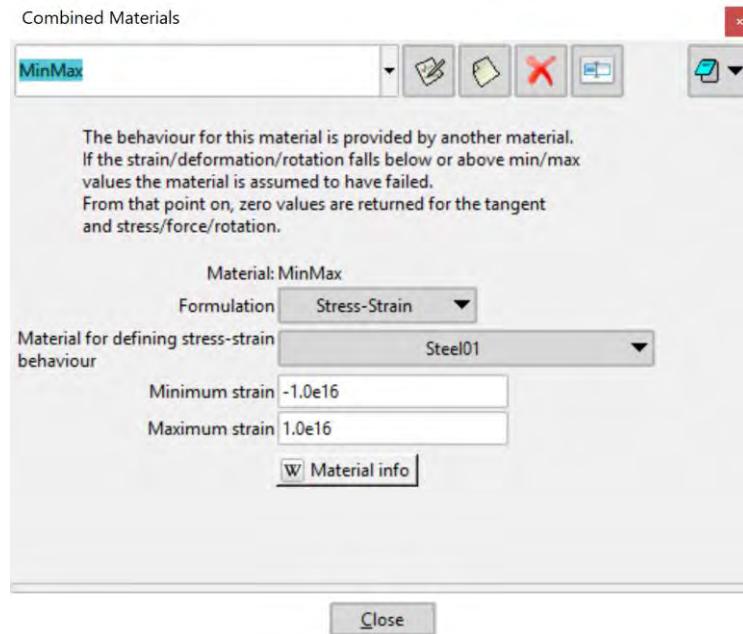


Figure 56: MinMax material options

nD Materials

nD materials represent the stress-strain relationship at the gauss point of a **continuum** element such as Quad, Brick etc.

Elastic Isotropic

This is a simple elastic isotropic material characterized just from elastic modulus E and Poisson's ratio. In addition, user can enter the mass density of the material.

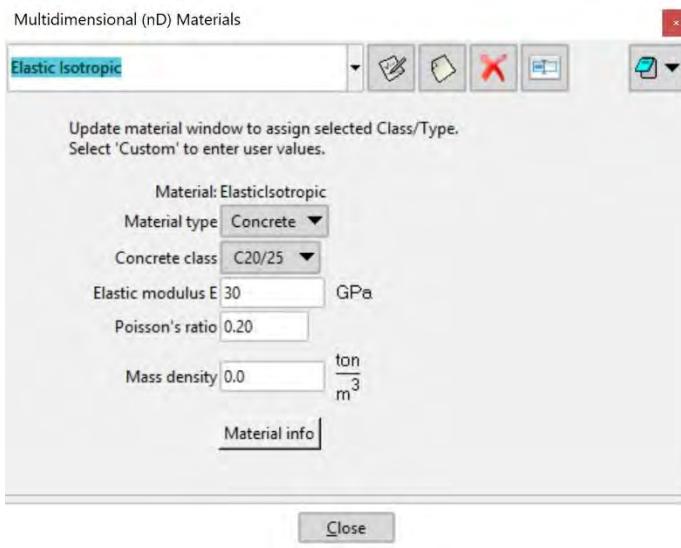


Figure 57: Elastic isotropic material options

Updating the material window, properties' values are updated depending on the selected material type (Concrete or Steel). For concrete classes, the Young modulus is updated according to Eurocode 2. Poisson's ratio for all concrete classes is changed to 0.20. For steel material type, the Young modulus is updated to 200 GPa and Poisson's ratio to 0.30. Mass density is not updated and always remains user-customized.

Note: Except continuum elements, Elastic Beam Column and Elastic Timoshenko Beam elements, use this material for the elastic modulus and mass density.

Material Properties	Default values
Elastic modulus E	30 GPa
Poisson's ratio	0.20
Mass density	0.0 ton/m ³

Elastic Orthotropic

This is an elastic orthotropic multi-dimensional material, which consists of 3 elastic moduli in all three perpendicular directions, 3 Poisson's ratios, three Shear moduli and the mass density of the material.

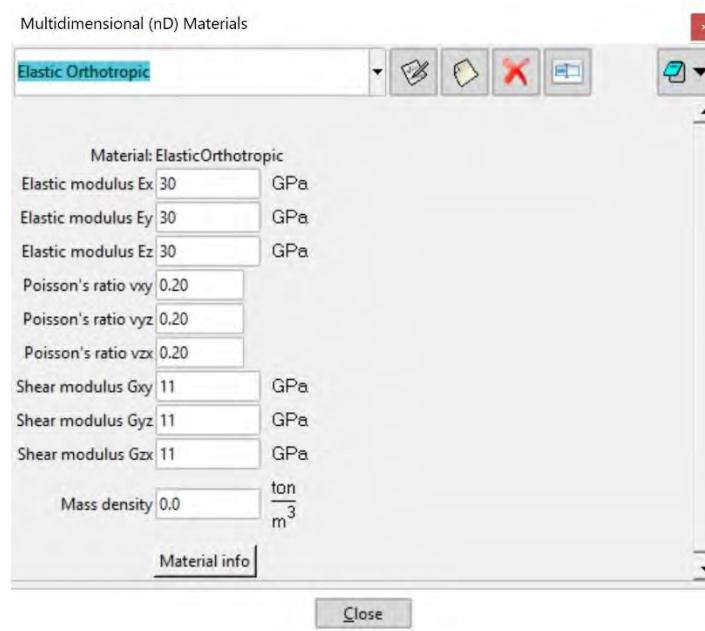


Figure 58: Elastic orthotropic material options

Material Properties	Default values
Elastic modulus Ex	30 GPa
Elastic modulus Ey	30 GPa
Elastic modulus Ez	30 GPa
Poisson's ratio vxv	0.20
Poisson's ratio vyv	0.20
Poisson's ratio vzv	0.20
Shear modulus Gxy	11 GPa
Shear modulus Gyz	11 GPa
Shear modulus Gzx	11 GPa

J2Plasticity

This is a multi-dimensional material, which follows the Von Mises yield criterium and has isotropic hardening.

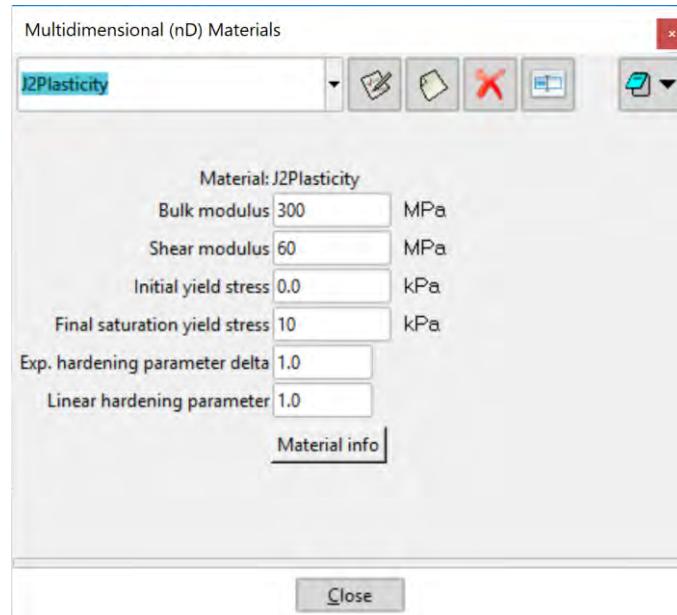


Figure 59: J2Plasticity material options

Yield function:

$$\varphi(\sigma, q) = |dev(\sigma)| - \sqrt{\frac{2}{3}} \cdot q(xi)$$

Saturation isotropic hardening with linear term:

$$q(xi) = \sigma_0 + (\sigma_{inf} - \sigma_0) \cdot e^{-delta \cdot \xi} + H \cdot \xi$$

Where:

σ_0 : Initial yield stress

σ_{inf} : Final saturation yield stress

Delta: exponential hardening parameter

H: linear hardening parameter

Material Properties	Default values
Bulk modulus	300 MPa
Shear modulus	60 MPa
Initial yield stress	0.0 kPa
Final saturation yield stress	10 kPa

Exponential hardening parameter	1
delta	
Linear hardening parameter	1

PressureIndependentMultiYield

This is an elastic-plastic material in which plasticity exhibits only in the deviatoric stress-strain response. The volumetric stress-strain response is linear-elastic and is **independent** of the deviatoric response. This material is for simulating monotonic or cyclic response of materials whose shear behaviour is **insensitive to the confinement change**. Such materials include, for example, **organic soils or clay under fast (undrained) loading conditions**.

Soil type can be:

- **Soft Clay**
- **Medium Clay**
- **Stiff Clay**
- **Custom, for user-customized values**

If a specific type of clay is selected and material window is updated, the field values are updated to the recommended values proposed by the OpenSees code developers of this material (Dr. Zhaohui Yang, UC San Diego). These values are provided as a quick reference for selecting parameter values, and can be modified later, selecting *Custom* soil type. Default Soil Type is Medium Clay.

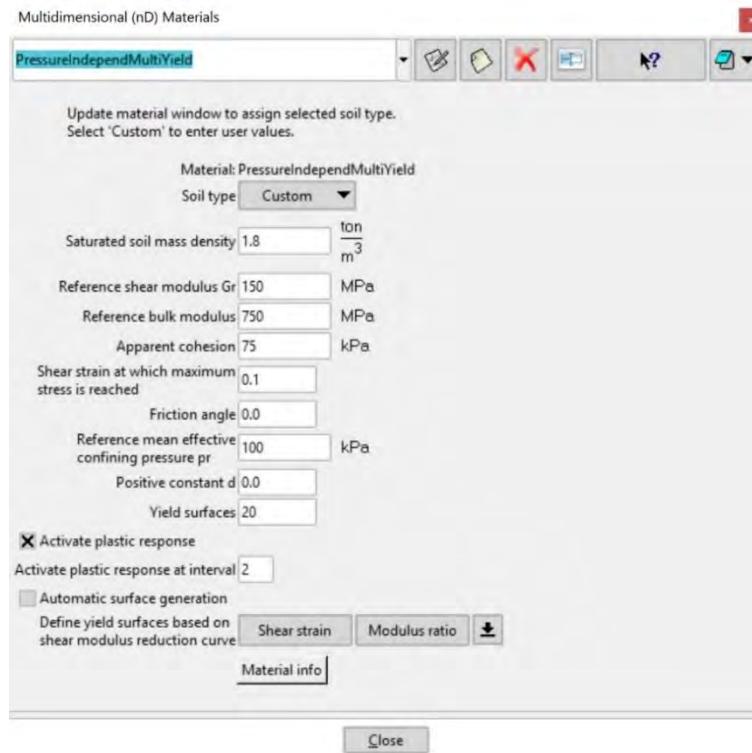


Figure 60: PressureIndependMultiYield material window

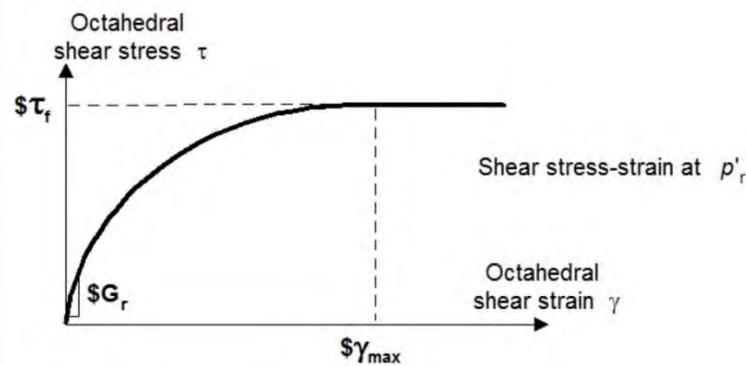


Figure 61: Constitutional law of shear stress-shear strain of PressureIndependMultiYield material

The friction angle Φ and cohesion c define the variation of peak shear strength τ_f as follows:

$$\tau_f = \frac{2 \cdot \sqrt{2} \cdot \sin\varphi}{3 - \sin\varphi} \cdot p'i + \frac{2\sqrt{2}}{3} \cdot c$$

The constant d parameter defines the variation of shear and bulk moduli G and B , respectively, as a function of instantaneous effective confinement p' as follows:

$$G = G_r \cdot \left(\frac{p'}{p'_r} \right)^d$$

$$B = B_r \cdot \left(\frac{p'}{p'_r} \right)^d$$

In case that Friction angle Φ is zero (usual for clays), the constant d is automatically reset to 0.0 (default value).

In case of **automatic surface generation**, the hyperbolic backbone curve (shear stress-strain relation) is defined as follows:

$$\tau = G \cdot \frac{\gamma}{1 + \frac{\gamma}{\gamma_r} \cdot \left(\frac{p'_r}{p'}\right)^d}$$

Where γ_r satisfies the following equation at the reference confinement p' :

$$\tau_f = \frac{2 \cdot \sqrt{2} \cdot \sin\varphi}{3 - \sin\varphi} \cdot p'_r + \frac{2\sqrt{2}}{3} \cdot c = G_r \cdot \frac{\gamma_{max}}{1 + \frac{\gamma_{max}}{\gamma_r}}$$

It is possible either to select automatic surface generation or to define yield surfaces directly based on the desired shear modulus reduction (G - γ curve) by providing the shear strain γ and modulus ratio G_s pairs. For this purpose, a special tkwidget is used, where the pair values can be easily copied. The G - γ curve then can be displayed as shown in the following figures.

Shear strain	Modulus ratio
0.0001	1
0.0002	1
0.0005	1
0.001	0.992
0.002	0.965
0.005	0.898
0.01	0.818
0.02	0.719
0.05	0.549
0.1	0.408
0.2	0.287
0.5	0.158
1	0.096
2	0.055
5	0.028
10	0.014

Figure 62: $G/Go-\gamma$ values

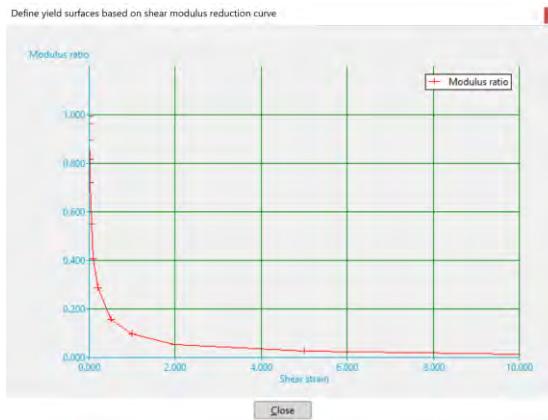


Figure 63: G/Go- γ curve

In case that **the user defines the yield surfaces**, cohesion c is ignored if Φ is defined as 0.0. Consequently, c is calculated as

$$c = \sqrt{3} \cdot \frac{\sigma_m}{2}$$

Where σ_m is the product of the last modulus and strain pair in the G/Go- γ curve.

Thus, it is worth noting that an appropriate cohesion depends on the backbone curve in case that $\Phi = 0.0^\circ$.

On the other hand, if Φ is greater than zero, this Φ will be ignored and is dictated by:

$$\sin\varphi = \frac{3 \cdot \frac{\sqrt{3} \cdot \sigma_m - 2c}{p'_r}}{6 + \frac{(\sqrt{3} \cdot \sigma_m - 2c)}{p'_r}}$$

If it results $\Phi < 0^\circ$, the Φ is set to 0° and cohesion is defined as shown above.

Material Properties	Default values
Saturated soil mass density	1.5 ton/m ³
Reference shear modulus Gr	60 MPa
Reference bulk modulus	300 MPa
Apparent cohesion	37 kPa
Shear strain at which maximum stress is reached	0.1
Friction angle	0°
Reference mean effective confining pressure pr	100 kPa

Positive constant d	0.0
Yield surfaces	20

PressureDependMultiYield

This is an elastic-plastic material for simulating the essential response characteristics of pressure sensitive soil materials under general loading conditions. Such characteristics include dilatancy (shear-induced volume contraction or dilation) and non-flow liquefaction (cyclic mobility), typically exhibited in **sands** or **silts** during monotonic or cyclic loading. The yield function is a conical surface in principal stress space (Prevost 1985; Lacy 1986). Material elasticity is assumed to be linear and isotropic, and that nonlinearity and anisotropy comes from plasticity.

To simulate **drained soil conditions**, use this material in combination with solid element (Quad, Tri31 etc.). In contrast, use it with one solid-fluid fully coupled element (QuadUP) with **low** permeability values to simulate fully undrained conditions. Finally, for partially drained soil response you should use this material with **suitable** permeability values.

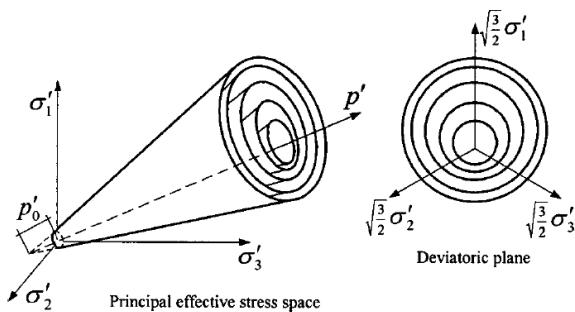


Figure 64: Conical yield surfaces in principal stress space and deviatoric plane (after Prevost 1985, Parra 1996 and Yang 2000)

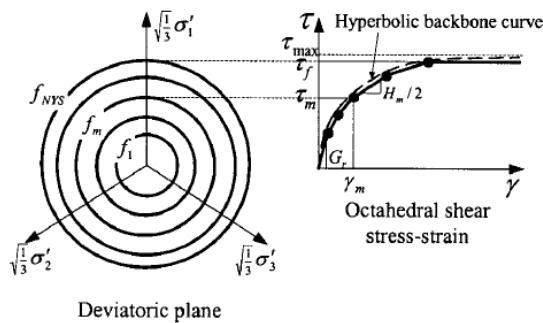


Figure 65: Hyperbolic backbone curve for soil nonlinear shear stress-strain response (after Prevost 1985 and Parra 1996)

The friction angle Φ defines the peak shear strength τ_f as a function of current effective confinement p' :

$$\tau_f = \frac{2 \cdot \sqrt{2} \sin\varphi}{3 - \sin\varphi} p'$$

The octahedral shear stress is calculated as:

$$\tau = \frac{1}{3} \cdot \left[(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{xx} - \sigma_{zz})^2 + 6 \cdot \sigma_{xy}^2 + 6 \cdot \sigma_{yz}^2 + 6 \cdot \sigma_{xz}^2 \right]^{0.5}$$

In case of **automatic surface generation**, the hyperbolic backbone curve (shear stress-strain relation) is defined as follows:

$$\tau = G \cdot \frac{\gamma}{1 + \frac{\gamma}{\gamma_r} \cdot \left(\frac{p'_r}{p'} \right)^d}$$

Where γ_r satisfies the following equation at p'_r :

$$\tau_f = 2 \cdot \sqrt{2} \cdot \frac{\sin\varphi}{3 - \sin\varphi} \cdot p'_r = G_r \cdot \frac{\gamma_{max}}{1 + \frac{\gamma_{max}}{\gamma_r}}$$

Where d is a positive constant defining variations of G and B as a function of instantaneous effective confinement p' as follows:

$$G = G_r \cdot \left(\frac{p'}{p'_r} \right)^d$$

$$B = B_r \cdot \left(\frac{p'}{p'_r} \right)^d$$

Within the framework of elasticity theory, the bulk modulus B can be calculated as:

$$B = 2 \cdot G \cdot \frac{1 + \nu}{3 - 6 \cdot \nu}$$

Soil type can be one of the following options:

- **Loose Sand**
- **Medium Sand**
- **Medium-dense Sand**
- **Dense**
- **Custom, for user-customized values.**

If a specific type of sand is selected and material window is updated, the field values are updated to the recommended parameter values proposed by the OpenSees code developers of this material (Dr. Zhaojun Yang, UC San Diego). These values are provided as a quick reference for selecting parameter values, and can be modified later, selecting *Custom* soil type. The default Soil Type is Loose Sand.

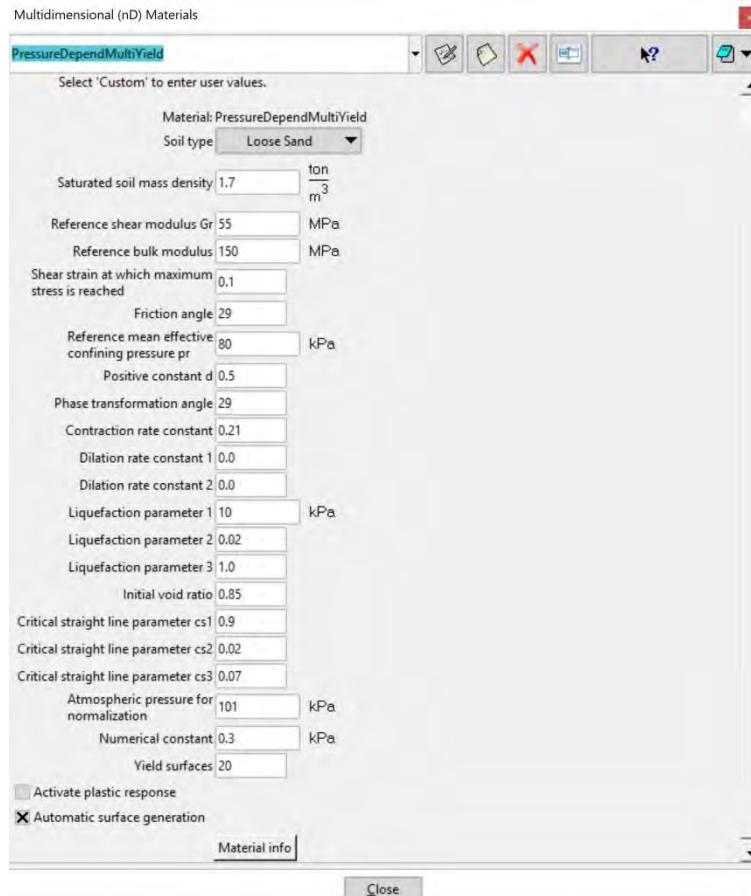


Figure 66: PressureDependMultiYield material window

The octahedral shear strain that corresponds to the maximum shear strength, is specified at reference mean effective confining pressure p'_r , as follows:

$$\gamma = \frac{2}{3} \cdot \left[(\varepsilon_{xx} - \varepsilon_{yy})^2 + (\varepsilon_{yy} - \varepsilon_{zz})^2 + (\varepsilon_{xx} - \varepsilon_{zz})^2 + 6 \cdot \varepsilon_{xy}^2 + 6 \cdot \varepsilon_{yz}^2 + 6 \cdot \varepsilon_{xz}^2 \right]^{0.5}$$

Like in PressureIndependentMultiYield material, it is possible either to select automatic surface generation or to define yield surfaces directly based on the desired shear modulus reduction ($G-\gamma$ curve) by providing the shear strain γ and modulus ratio G_s pairs. In that case, the user-defined friction angle Φ is ignored, and is calculated as:

$$\sin\varphi = \frac{3 \cdot \sqrt{3} \cdot \frac{\sigma_m}{p'_r}}{6 + \sqrt{3} \cdot \frac{\sigma_m}{p'_r}}$$

Where:

σ_m : product of the last shear modulus G and shear strain γ in the corresponding defined reduction curve.

Note: If the yield surfaces are defined by the user, it is important to the backbone curve so as to render a proper friction angle Φ .

If the above equation results a Φ smaller than phase transformation angle Φ_{PT} , then Φ_{PT} is set equal to Φ .

Dilation constants and liquefaction parameters are needed only when critical-state response (flow liquefaction) is expected. While reaching the critical-state, material dilatancy is set to zero.

Dilation constants are non-negative constants defining the rate of shear-induced volume decrease or pore pressure buildup. **Larger values mean faster contraction rate.**

Liquefaction parameter 1 defines the effective confining pressure below which the mechanism is in effect. Liquefaction parameter 2 defines the maximum amount of perfectly plastic shear strain developed at zero effective confinement during each loading phase. Liquefaction parameter 3 defines the maximum amount of biased perfectly plastic shear strain γ_b accumulated at each loading phase under biased shear loading conditions.

Material Properties	Default values
Saturated soil mass density	1.7 ton/m ³
Reference shear modulus Gr	55 MPa
Reference bulk modulus	150 MPa
Shear strain at which maximum stress is reached	0.1
Friction angle	29 degrees
Reference mean effective confining pressure pr	80 kPa
Positive constant d	0.5
Phase transformation angle	29 degrees
Contraction rate constant	0.21
Dilation rate constant 1	0.0

Dilation rate constant 2	0.0
Liquefaction parameter 1	10 kPa
Liquefaction parameter 2	0.02
Liquefaction parameter 3	20
Initial void ratio	0.85
Critical straight line parameter cs1	0.9
Critical straight line parameter cs2	0.02
Critical straight line parameter cs3	0.07
Atmospheric pressure for normalization	101 kPa
Numerical constant	0.3
Yield surfaces	20

Element Types

The element classes that are supported by the GiD+OpenSees Interface are shown in the following table and they consist of both structural and continuum elements.

Table 4: Interface Element type features (v2.3.0)

Beam-Column Elements <ul style="list-style-type: none"> - Elastic Beam-Column - Elastic Timoshenko Beam-Column - Force-Based Beam-Column - Displacement-Based Beam-Column - Flexure-Shear Interaction Displacement-based Beam-Column 	Surface Elements <ul style="list-style-type: none"> - Quad - QuadUP - Tri31 - ShellMITC4 - ShellDKGQ
Truss Elements <ul style="list-style-type: none"> - Truss - Corotational Truss 	Volume Elements <ul style="list-style-type: none"> - Standard Brick Elements (8node)

Zero Length Elements

- Zero Length

A characteristic Element type window is shown below.

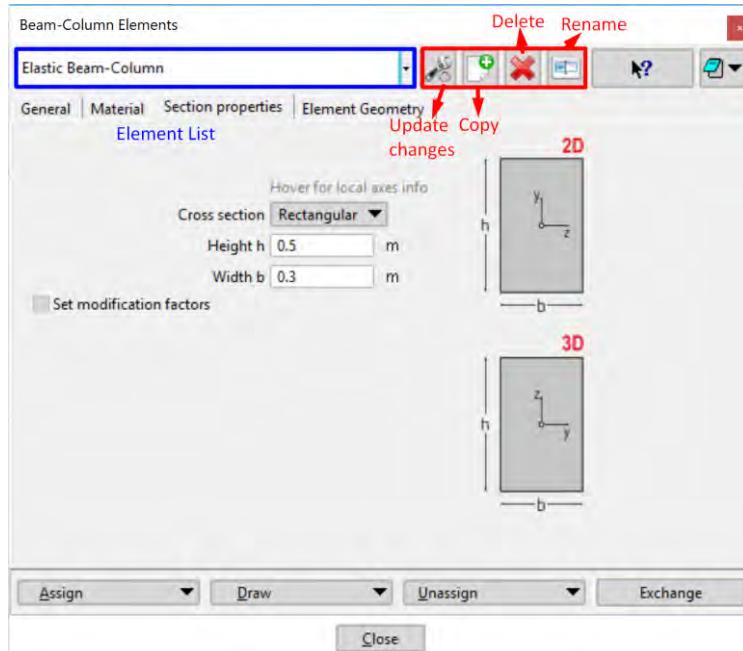


Figure 67: Typical Element dialog window

Four command options are available on the bottom of this window type:

Assign command: OpenSees Finite Elements are assigned to geometrical objects.

After the discretization, the finite elements will inherit the Element type assigned to the geometrical entity that they have been generated from. For instance, if you assign an Elastic Beam-Column element type to a line object, and then this line is meshed into 10 linear finite elements (10 cells per line), all these ten elements will be of Elastic Beam-Column type.

Draw command: OpenSees Finite Elements can be displayed with colors.

Unassign command: OpenSees finite Elements are unassigned from the geometrical objects.

Exchange command: OpenSees Elements or Materials can be imported/exported from/to a GiD+OpenSees project to/from another.

Every Element window contains a *Wiki info button* in General tab that links to the OpenSees wiki for further information and references regarding to each element type.

Note: Zero Length Elements are not included in this type of windows because they are assigned to points. See condition windows in Conditions section.

Beam Column Elements

For 2D models, Beam-Column Elements consist of two nodes, which have 3 degrees of freedom (u_i, v_i, θ_i), two translational and two rotational. On the other hand, for 3D models, their nodes have 6 degrees of freedom ($u_i, v_i, w_i, \theta_{xi}, \theta_{yi}, \theta_{zi}$), three translational and three rotational.

Elastic Beam Column – Timoshenko Beam Column

These element types are almost identical in user input level. Timoshenko beam column considers shear deformations, in comparison with Elastic beam column which does not. In user input level, Timoshenko requires the shear modulus G , even in a 2D analysis.

In every occasion, Shear Modulus G is calculated as

$$G = \frac{E}{2 \cdot (1 + \nu)}.$$

Where:

E : The Elastic Modulus

ν : Poisson's ratio

The Elastic Modulus E and Poisson's ratio is determined from a **pre-defined Elastic Isotropic** material through the corresponding *Material* field in *Material* tab.

In *Section properties* tab, the user can define a particular *Cross Section* shape or a general one. For general cross section, user enters all the geometrical and inertial properties. The other types are Rectangular, Circular and Tee, for which, user defines just the geometrical characteristics (dimensions) and all the necessary properties (Area etc.) are automatically calculated through the analysis process. Moreover, user can set modification factors, which will be then multiplied by the corresponding properties.

Note: For 2D analysis, only Area and Moment of Inertia about local z axis are considered from the *Section properties* tab.

In the *Element Geometry* tab, the user can choose the geometric transformation among Linear, P-Delta or Corotational to consider geometric non-linearities or not.

Finally, the specific weight is input through the *Weight* tab. It is considered only for Dead loads. Specifically, it is multiplied by the Section Area, and the dead load is applied as uniform distributed external force per unit length.

Force-Based Beam Column

This element type is based on flexibility-based formulation which provides a more accurate description of the force distribution within the element, offering significant advantages over stiffness-based elements. The element state determination is based on a non-linear iterative algorithm, like Newton-Raphson method, accepting both residual displacements and unbalanced section forces, which violate compatibility and equilibrium along the element, respectively. These violations are limited to a specified numerical tolerance. This element is based on the Euler-Bernoulli beam theory, considering linear geometry. Thus, plane sections remain plane and normal to the local-x (longitudinal) axis of the element during its' deformation history. It is worth noting that, **effects like cracking and bond-slip**, which are usual characteristic type of failures in reinforced concrete elements, are **not considered**. The phenomenon of cracking can only be simulated by modifying the stress-strain behavior of reinforcing steel or concrete fibers. Moreover, the user should be aware that **shear effects are not accounted**. Consequently, such behavior matches better for medium to large span to depth ratios of the structural member [5] [6].

This element type uses a pre-defined Section Force-Deformation model which is selected in the *Section* tab. The available compatible Section model is **Fiber**. The number of integration points per element is currently fixed to three points per element. To increase analysis accuracy, a finer mesh is recommended for modeling structural members. Moreover, the integration type is set to Lobatto, which is the most common approach for evaluating the response of this type of elements because it places an integration point at each end of the element, where the bending moments are largest especially for seismic loads [7].

In the *Element Geometry* tab, user can choose the geometrical transformation among Linear, P-Delta or Corotational to consider geometric non-linearities or not.

In the *Element Compatibility* tab, user can enable the iterative form of the flexibility formulation, and determine the number of iterations as well as the tolerance to satisfy element compatibility.

Finally, Weight and Mass density can be defined in *Mass-Weight* Tab. Weight density is considered only if dead loads are activated through Intervals Data options.

Note: It has been observed that mass density in force-based beam columns is only considered in eigenvalue analysis by the OpenSees solver. For other kind of dynamic analysis, it is recommended to use nodal masses. It may be an issue in OpenSees source code.

Displacement-Based Beam Column

This element is based on the common stiffness-based approach and is also based on the assumption of geometrically linear behavior. Thus, plane sections remain plane and normal to the local-x (longitudinal) axis of the element during its' deformation history. Disadvantages linked with effects such as cracking and bond-slip as well as shear effects also exist for this type of element.

It uses a pre-defined Section Force-Deformation model which is selected in the *Section* tab. The available compatible Section models are **Fiber** and **Section Aggregator**. The number of integration points per element is currently fixed to three points per element. To increase analysis accuracy, a finer mesh is recommended for modeling structural members. Moreover, the numerical integration type is set to Legendre by default. [7]

In the *Element Geometry* tab, user can choose the geometric transformation among Linear, P-Delta or Corotational to consider geometric non-linearities or not.

Finally, Weight and Mass density can be defined in *Mass-Weight* Tab. Weight density is considered only if dead loads are activated through *Intervals Data* options and Mass Density only if dynamic or/and eigenvalue analysis is going to be executed.

Flexure-Shear Interaction Displacement-Based Beam Column

This element type uses a pre-defined Section Force-Deformation model, which is selected in the *Section* tab. The only compatible Section model is FiberInt. The number of integration points per element is currently fixed to three points per element. To increase analysis accuracy, a finer mesh is recommended for modeling structural members.

By default, it uses a particular geometric transformation *LinearInt*, which is based on the geometric Linear transformation. Thus, user cannot define other geometric transformation. For the local axes orientation, see Beam-Column Local Axes section.

Like Displacement-based Beam Column, Weight and Mass density can be defined in *Mass-Weight* Tab. Weight density is considered only if dead loads are activated through Intervals Data options and Mass Density only if dynamic or/and eigenvalue analysis is going to be executed.

Beam-Column Local Axes

Beam-Column elements (Frame elements) use a geometric transformation, which transforms the element stiffness and resisting force from the local-coordinate system to the global one and vice versa.

Three types of geometric transformation are available:

- **Linear**
- **P-Delta: Considers second-order P-Delta effects**
- **Corotational: For large displacement-small strain problems**

The interface uses some standards local axes orientation depending on the model dimensions, the defined vertical axis, as well as the in-length orientation of the frame element.

In any case the local x axis always coincides with the positive element direction, which depends on the corresponding line object design direction. The line direction can always be changed through *Utilities > Swap normal > Lines*.

In a 2D problem (X-Y Plane):

The local axes are defined by OpenSees itself: Local z axis coincides with the global Z axis direction and local y axis is found by the right-hand rule ($V_x \times V_y = V_z$). So, for a horizontal element, if it is aligned with the positive or negative global X axis, the local y axis is aligned with the positive or negative global Y axis respectively. Similarly, for a vertical element, if it is aligned with positive or negative global Y axis, the local y axis is aligned with the negative or positive global X axis respectively.

In a 3D problem:

Vertical axis is user-specified (Y or Z) in *General Options* () menu.

In general, a vector Vec_{xz} is specified, which in combination with the local x axis, define the local xz plane. Then the local y axis is defined by taking the cross product of the x-axis vector and the Vec_{xz} vector, $V_y = V_{xz} \times V_x$. Finally, the local z axis is found by taking the cross product of the y and x local axis vectors, $V_z = V_x \times V_y$.

For horizontal and vertical elements, depending on the specified vertical axis, the local axes can be easily determined:

For a horizontal frame element: local z axis positive direction coincides with the direction of the defined vertical axis (Y or Z). Local y axis is then found by the right-hand rule.

For a vertical element: Local z axis positive direction coincides with the global X axis negative direction. Local y axis is found by the right-hand rule.

Truss Elements

Truss elements' stress-strain behavior is considered along the longitudinal axis of the element. Thus, they cannot be loaded with transverse forces, consequently it resists only by axial loads. In the context of GiD+OpenSees Interface, their nodes have 2 degrees of freedom or 3 degrees of freedom for 2D geometry models or 3D geometry models, respectively. For equivalent simulation of truss element with 1 degree of freedom per node, you should design the truss element along the global X axis (inside XY global plane) and fix all nodes' y translation.

Truss

This element type can use any uniaxial material except concrete ones through the Interface. The material can be selected in the *Material* tab.

Truss elements require just an Area since length is determined by the end-nodes. The area can be calculated or directly given by the user. In *Sections properties* tab, the user can select either a particular cross section shape or *General* cross section and as a result, the area is automatically calculated or input, respectively. The available Cross Sections are Rectangular, Circular, Tee and General.

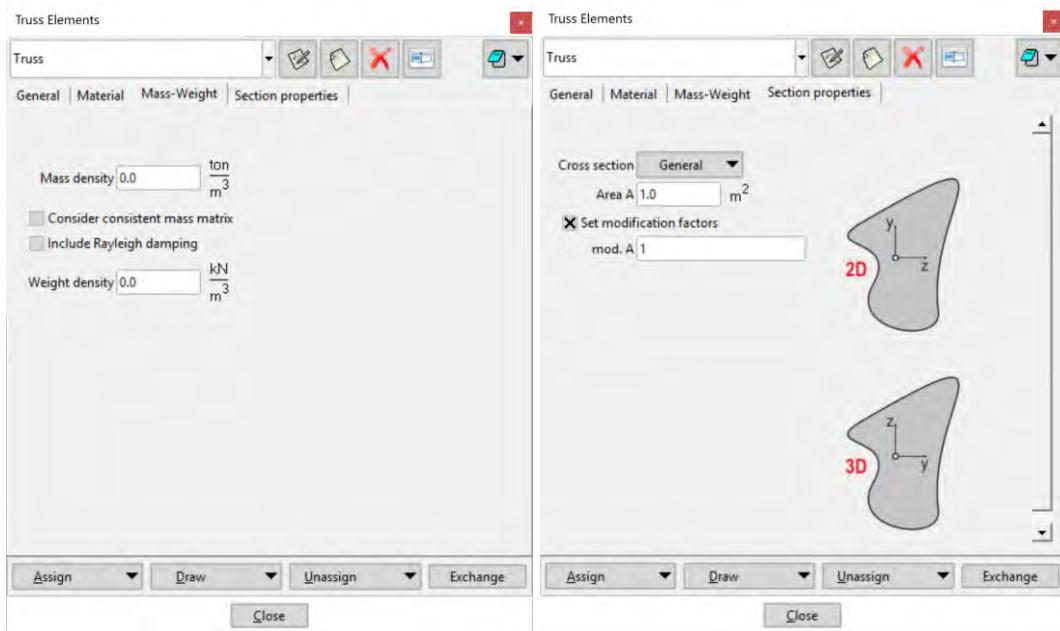


Figure 68: Truss element options

In the *Mass-Weight* tab, the user can determine the mass and weight density in case that a dynamic or/and eigenvalue analysis is going to be executed and if dead loads are going to be applied, respectively. Considering Dead loads, weight density is multiplied by the element length and cross section area and the total force is bisected and applied on the two end nodes of the element. Finally, the user can assume consistent mass matrix instead of lumped, as well as include Rayleigh damping for truss elements through checking the corresponding boxes.

Note: This element does **not** include geometric nonlinearities.

Corotational Truss

The only difference between this element type and simple truss element, is that Corotational truss **includes** geometric nonlinearities, where corotational formulation tries to separate rigid body motions (rotations and translations) from strain producing deformations at the local element level [8].

Surface Elements

Quad

This is a continuum four-node plane element, which is based on bilinear isoparametric formulation. The Interface automatically handles the order of the nodes definition, as GiD also uses counter-clockwise order. Quad Elements can be simulated **only in 2D models** and each node has 2 translational degrees of freedom.

Quad elements use a pre-defined nD material, which can be selected inside the *Material* tab. Moreover, in the same tab user can determine the material plane behavior (Plane Stress or Plane Strain). For completing the element geometry definition, user is asked to input the thickness of the section. Finally, surface pressure and body forces can be applied, from where consistent nodal loads are computed.

Quad U-P

This is a continuum four-node plane strain element based on bilinear isoparametric formulation. QuadUP Elements can be simulated **only in 2D models** and each node has 3 degrees of freedom, 2 translational and one that corresponds to fluid pressure.

It is proper for simulating dynamic response of solid-fluid fully coupled material, based on Biot's theory of porous medium. These elements also use a pre-defined multi-dimensional material through the *Material* tab and their plane behavior is either plane strain or plane stress.

In the *Properties* tab the user can enter the thickness of the plane element, the combined undrained bulk modulus Bc , and the fluid mass density.

Bc can be calculated as:

$$Bc = \frac{Bf}{n}$$

Where:

Bf : the bulk modulus of fluid phase (2.2 GPa)

n : the initial porosity

Permeability coefficients in both horizontal and vertical directions can also be input by the user, as well as the interval in which these parameters are going to be updated.

In the *Pressure* tab, the user is asked to enter a uniform normal traction if any.

In the *Body Forces* tab, the user may enter the gravity acceleration components in both X and Y axes. Thus, the user can implement even a slope analysis, analyzing the gravity acceleration in two components.

Tri31

This is a three-node constant strain triangular element, which contains only one integration point. The order of the nodes definition is automatically handled by the Interface. It can be simulated **only for 2D models** and each node has 2 translational degrees of freedom.

The required parameters are the same as for the Quad element.

ShellDKGQ

This is a four-node quadrilateral shell element based on the theory of generalized conforming element. It has 6 degrees of freedom (3 translational and 3 rotational) per node and can be used **only for 3D analyses**. For using it in a 2D model, design your 2D model outside the global XY plane (parallel or normal to it) and define the proper vertical axis from the *General Options* menu.

Two Section types can be used for ShellDKGQ elements from the *Section* tab: *Plate Fiber* and *Elastic Membrane Plate* Section models. Inside Weight tab, the specific weight can be entered, in order to be used for dead loads. The total force is calculated multiplying the weight density by the thickness and the area of each element and then it is subdivided to the four element nodes. The thickness is Thus, finer mesh is required for the better dead load distribution.

ShellMITC4

This is a four-node quadrilateral shell element based on the theory of mixed interpolation of tensorial components proposed by Dvorkin et al. It uses a bilinear isoparametric formulation in combination with a **modified shear interpolation to improve thin-plate bending performance**. It is used **only for 3D analyses** and has 6 degrees of freedom (3 translational and 3 rotational) per node. For using it in a 2D model, design your 2D model outside the global XY plane (parallel or normal to it) and define the proper vertical axis from the *General Options* menu.

ShellMITC4 is identical to ShellDKGQ on user-input level.

Volume Elements

Standard Brick Element

This is an eight-node brick element object, which is based on a trilinear isoparametric formulation. Obviously, it can be used only for 3D analyses and has 3 translational degrees of freedom per node.

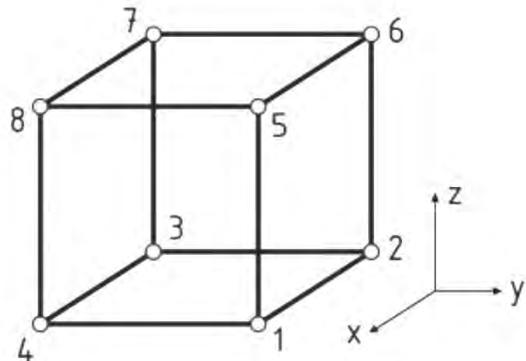


Figure 69: Standard Brick element and its node order definition

The only parameters it takes, is the material and body forces. Material can be any pre-defined multi-dimensional material and body forces are given in all three **global** directions. The node definition order is also handled by the Interface.

Conditions

In the context of the Interface, some features are called conditions and use a dialog window type, which is called **Condition window**. The modification of the fields in a Condition window **cannot be saved** and field values are initialized with the default values each time the window is accessed. A typical condition window is shown in Fig. 70.

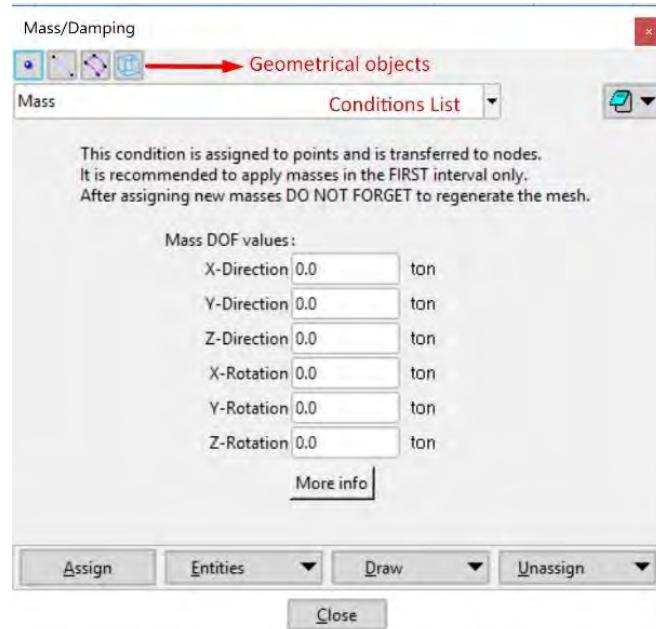


Figure 70: Typical Conditions window

Note: Conditions should be assigned on the geometry model. Otherwise, if the user transits from the meshed model to the geometry model, all conditions applied to meshed model will be lost.

The features that use Condition windows are the following:

- Loads
- Mass
- Damping
- Restraints
- Constraints
- Zero Length Elements

There are four options available on the bottom area of the condition window:

Assign command: Assign the condition to geometrical objects.

Entities command: See the applied conditions so far.

Draw command: Applied conditions are displayed with colours and labels.

Unassign command: Unassign the condition from the geometrical objects.

Moreover, some geometrical object icons () are displayed at the top of a condition window. Each one enables the conditions, that are applicable to the corresponding geometrical entity, in the conditions list below. **At this point, it is important to clarify that a condition can be transferred to the body element, to lower element entities or to higher elements after generating the mesh.** For instance, you can apply a condition to a line object and then it will be transferred on nodes of the linear elements that are generated from the initial object. Condition windows include comments that explain how the condition are transferred at the transition to the meshed model.

Zero Length Elements

Zero Length elements can be set through the Toolbar icon  . Zero length elements are used to define the multiple (optional) Force-Deformation behavior between two nodes at the same location. They use uniaxial materials for each degree of freedom. The dialog window with the available fields is demonstrated in Fig. 71.

This condition is applicable only to points and is transferred to the corresponding nodes. It should be assigned to two or more points with same ID. If more than two points are selected with same ID, $(nodes - 1)$ zero length elements are created in node-pairs within tcl script. This could be observed in a three-joint internal hinge simulation. In Addition, they can be used usually to simulate the soil-structure-interaction behavior and the soil sustainability (Springs).

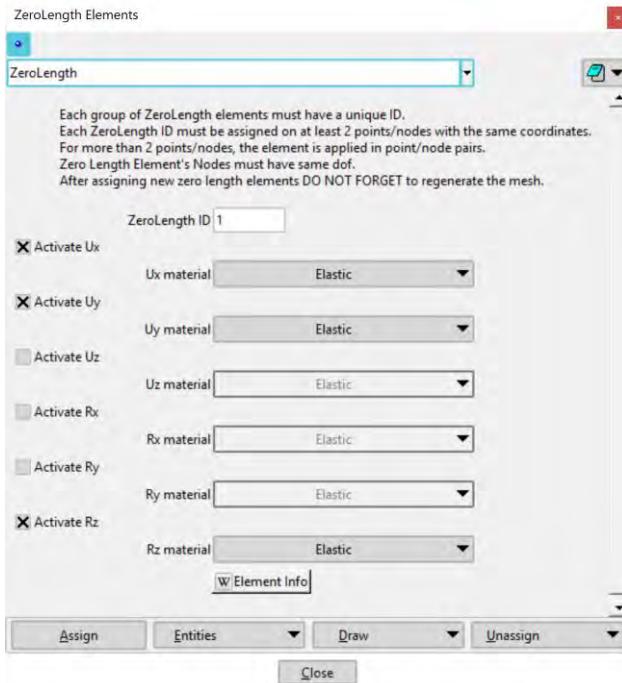


Figure 71 Zero length elements condition

Note: It is important that the zero length elements are assigned in **the first interval**, otherwise they are omitted.

Records

Record files that correspond to ground motions or not can be stored and used in case of a dynamic analysis. This can be set through the toolbar icon .

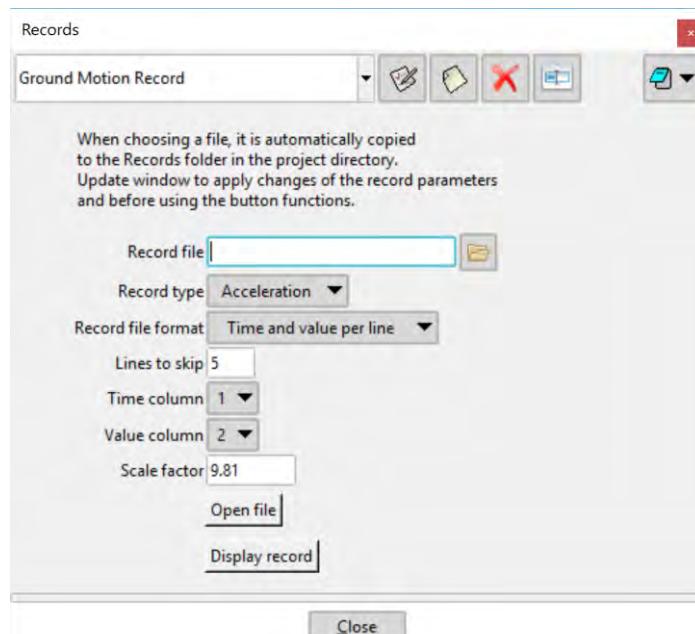


Figure 72: Records window

This window is like a material window. User can make copies of this with different names and different properties. In this way, all the desired ground motion records can be stored.

Record type can be one of the following:

- **Acceleration**
- **Displacement**
- **Velocity**
- **Function (Force)**

Acceleration should be selected for analysis using uniform excitation pattern or multiple support excitation pattern.

Displacement should be selected for analysis using multiple support excitation pattern.

Velocity or Function may be selected for analysis using Function loading type (Plain Pattern using Path Timeseries).

3 Record file formats are supported:

- **PEER format**
- **Single value per line**
- **Time and value per line**

If Single value per line is selected for Record file format, the lines to be skipped (headers) and the time step per value are user-defined. If Time and value per line is selected, the lines to be skipped as well as the time and value columns are user-defined. **Finally, the scale factor should be carefully input which depends on the record values' unit as well as the Data units of the project.**

When a record file is chosen through the *file button*, it is automatically copied to the *Records* folder inside the project directory *projectname.gid*, from where it is used for the analysis module (source command inside main tcl script).

Note: To load a record file through the *file button*, the user must have already saved the project, otherwise is warned by a proper message.

Note: While saving an existing project to a new one using v2.2.5 or earlier, the *Records folder* is not copied. Thus, you should either manually copy it or reselect and store the records through the new project. Otherwise an error will be caused during analysis. **Since v2.3.0 it is transferred automatically while saving a new project.**

Since v2.3.0 two new button functions were added. One for opening the chosen file for observing the proper parameters to be entered in the Interface and one for displaying it. The *open file* button opens the chosen file using **the default user text editor** for the selected type of file. For displaying the record an external program was developed (*RecordViewer.exe*) which can be find in the *exe* folder inside the *problem types* folder in Program files where GiD has been installed. This program can be used either **independently or in combination with GiD+OpenSees Interface**. Pushing the *display record* button, the *RecordViewer.exe* is called taking as input data the file including its parameters (format, lines to skip etc.) and the record is automatically displayed graphically. Moreover, it provides Fast Fourier Transformation formula and hence you can see the Fourier Spectrum. Finally, the scale factor can be modified either manually or determining the desired maximum value (e.g. acceleration) and the record is updated. Any modifications can be then saved to a new/existing file and may be used by the GiD+OpenSees Interface. It is worth noting that full open/save file capabilities are available, and hence *RecordViewer.exe* can be used to apply any modifications in the record file which is going to be implemented by GiD+OpenSees Interface. Note also that a file is saved in a time/value column format with one header line, and as a result, the record parameters may be also modified in the Interface in order to work efficiently.

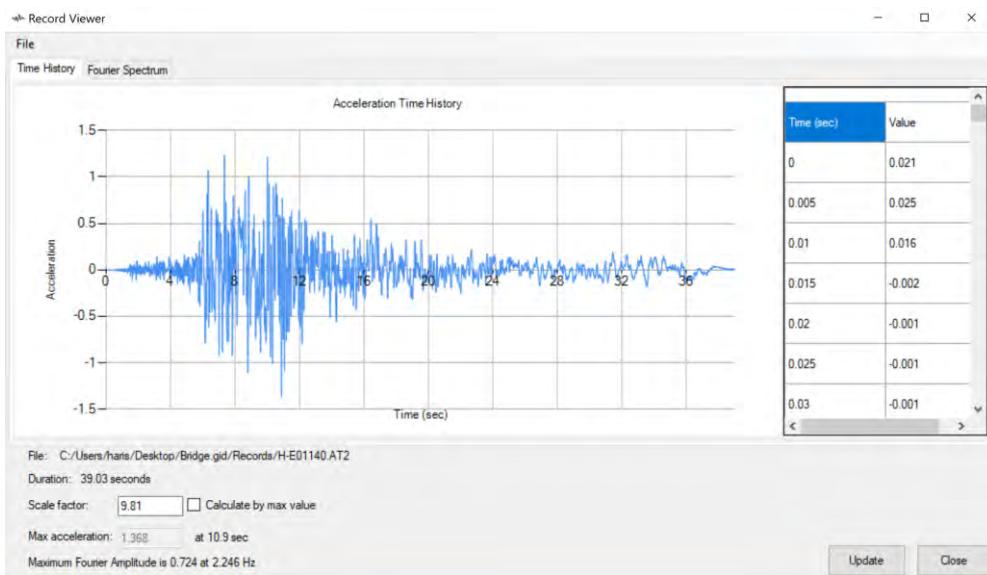


Figure 73: RecordViewer.exe; Time history

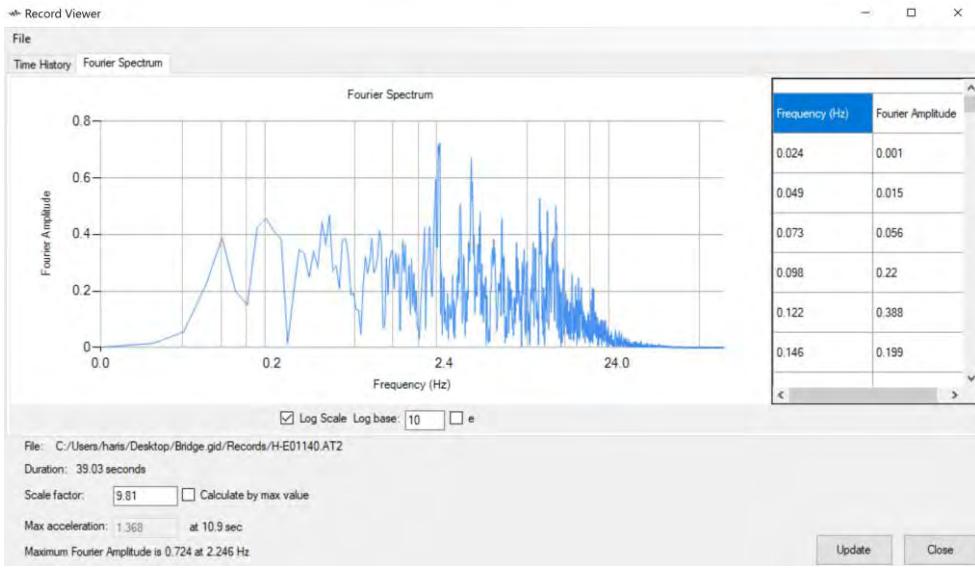


Figure 74: RecordViewer.exe; Fourier Spectrum

Note: Any changes must be followed by window updating before using the button functions.

Restraints

Single-Node Constraints can be set through the corresponding toolbar icon . They can be applied on points, lines or surfaces and transferred to the corresponding nodes of the selected geometrical entity. In this way you can determine which degrees of freedom are known in advance (fixed) or unknown.

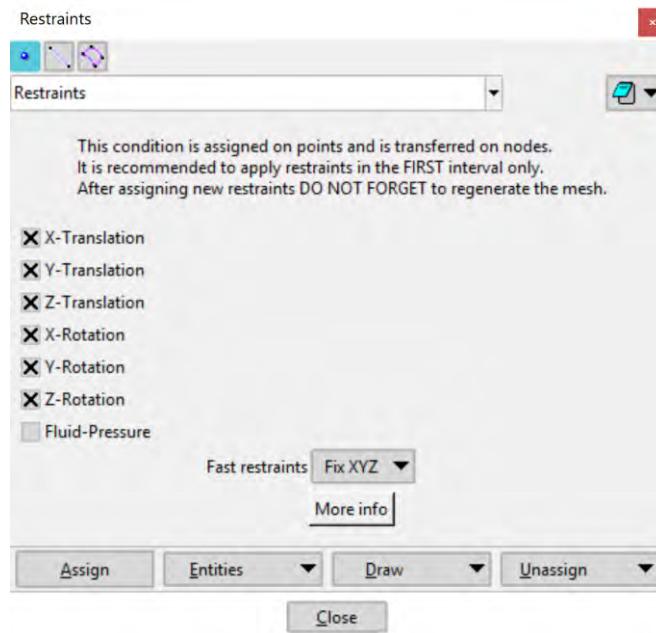


Figure 75: Restraints condition options

Note: It is important that the boundary conditions must be applied in the **first interval**, otherwise they are omitted.

Constraints

Multi-Node constraints can be set through the corresponding toolbar icon . There are two types of constraints:

- **Equal Constraint**
- **Rigid Link (Body Constraint)**
- **Rigid Diaphragm**

Constraints conditions consists of four condition types

- Equal constraint master node
- Equal constraint slave nodes
- Rigid Link master node
- Rigid Link slave nodes
- Rigid diaphragm master node
- Rigid diaphragm slave nodes

Equal Constraints

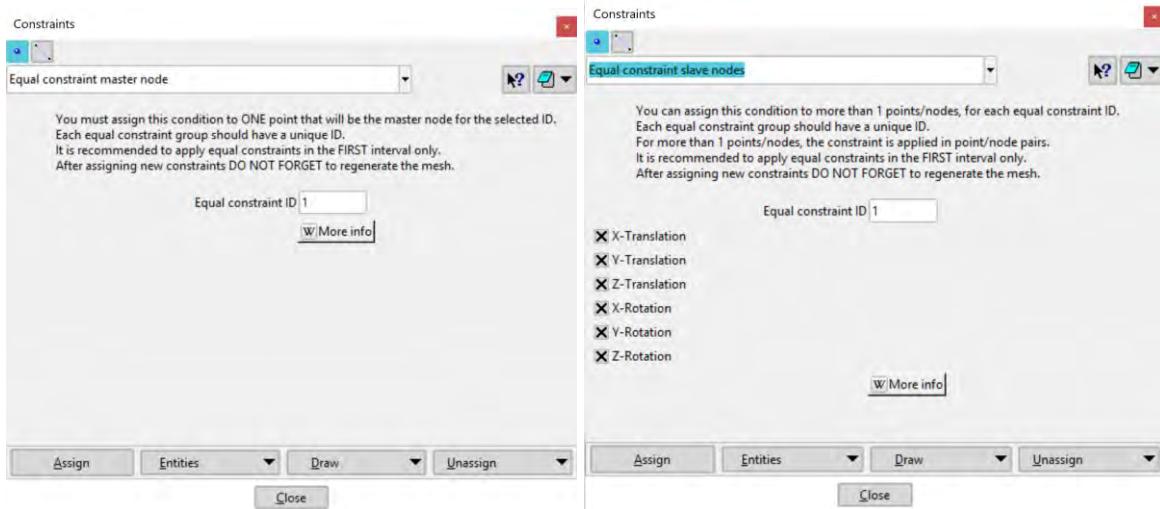


Figure 76: Equal constraint conditions

Equal constraint results equal displacements along the translational degree of freedoms among the nodes which share an identical ID. **There is no coupling between translation and rotation.**

Each slave node's degree of freedom that is checked, is equal to the master node's one that has the same equal constraint ID. The ID links the master node with the slave nodes and the *equalDOF* command is implemented in node pairs (Master node and slave node) in the tcl OpenSees input file.

The Master node condition is applied only to points and is transferred to the corresponding node when mesh is generated. On the other hand, slave node condition can be also applied on line objects and is transferred on nodes of the linear elements when mesh is generated. For instance, in a 2D frame analysis, assuming rigid diaphragms on each floor, and a fine mesh is desired, the slave node condition could be assigned to the horizontal (Beams) line objects for including all nodes and not the end nodes only.

Note: This condition can be applied among nodes with different degrees of freedom, but constraining common degrees of freedom.

Rigid Link

Rigid links translate and rotate nodes, which share an identical ID, together as a **rigid body** and hence the distance between master and each slave node influences the total behavior and the results, in contrast to equal constraint.

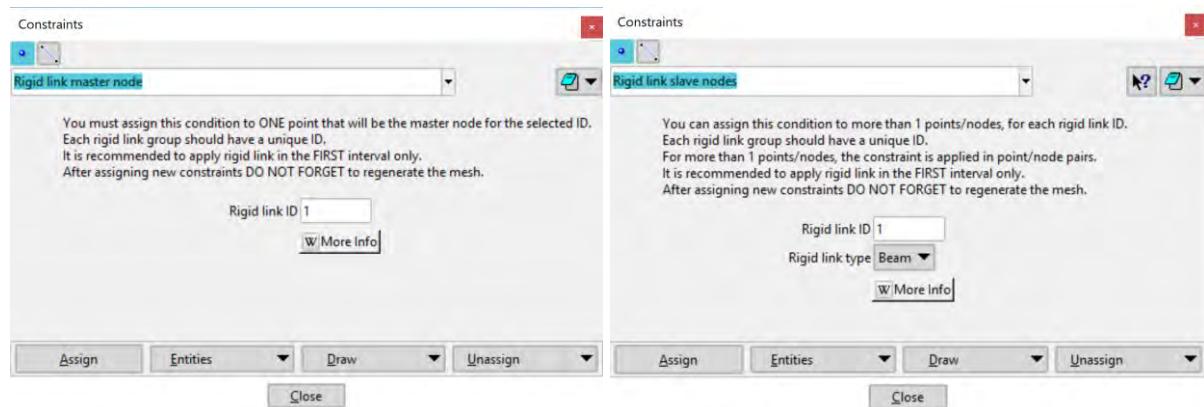


Figure 77: Rigid link constraint conditions

Note: Rigid link connections can be applied only between nodes with **same degrees of freedom**.

Rigid Diaphragm

Master node condition can be applied only to points and is transferred to the generated node after the discretization. Slave nodes condition is also applicable to line objects and is transferred to the corresponding nodes of the linear elements after the discretization. Each slave node

follows the movement in a specific plane of the Rigid Diaphragm Master node with the identical ID.

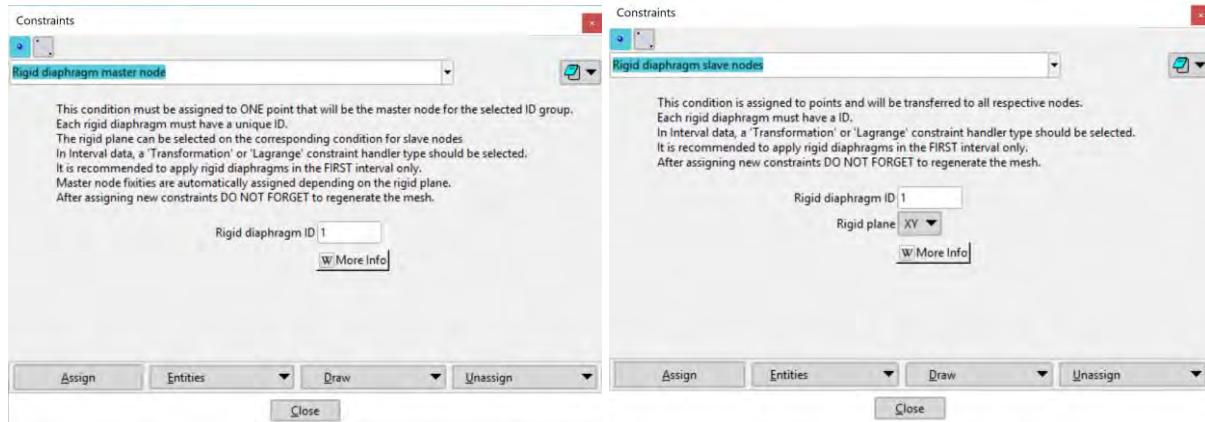


Figure 78 Rigid diaphragm conditions

For instance, in a 3D frame analysis, if a finer mesh is desired for the beams of a rigid floor, the slave node condition should be assigned on line objects (beams).

Note: The proper degrees of freedom of the master node are automatically fixed, depending on the rigid plane.

Note: It is important that the constraints must be applied in the **first interval**, otherwise they are omitted.

Loading Parameters

Load conditions can be set through the Toolbar icon . There are five types of Load conditions available:

- Nodal Forces
- Nodal Displacements (imposed)
- Uniform line Forces
- Ground Motion from Record
- Sine Ground Motion

Note: Every condition is always applied in the active interval and as result the proper load parameters must be assigned in each interval, depending on the analysis options in *Interval Data*.

Nodal Forces

Nodal Forces (or Moments) can be applied to points, lines or surfaces and be transferred to the corresponding nodes of the elements that are generated from the geometrical object after the discretization.

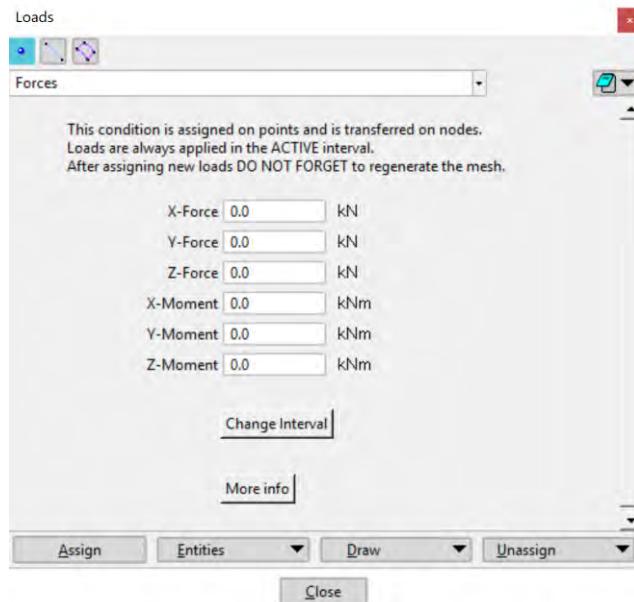


Figure 79: Nodal Forces condition

For this condition to be considered, the Loading type must be ***Constant, Linear or Function*** in *the Interval Data* options in the **active interval**, where it was applied.

Uniform Forces

Uniform Forces are applicable only to line objects and transferred to body elements (linear elements) when mesh is generated. This condition should be applied to frame elements only, otherwise an error may be occurred during the analysis. So far, it follows the local axes reference system, which was explained before. See frame Beam-Column local axes section for further information.

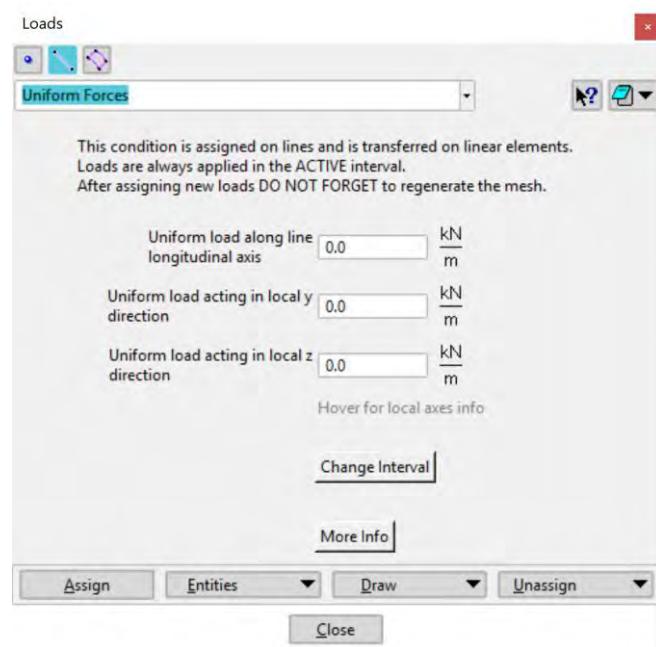


Figure 80: Uniform Forces condition

For this condition to be considered, the Loading type must be **Constant, Linear or Function** in the **Interval Data** options in the **active interval**, where it was applied.

Nodal Displacements

Nodal Displacements are applicable only to points and transferred to the corresponding nodes.

Zero values are similar to restraining the corresponding degree of freedom, therefore **zero values are omitted by the interface**. User can use restraint condition instead.

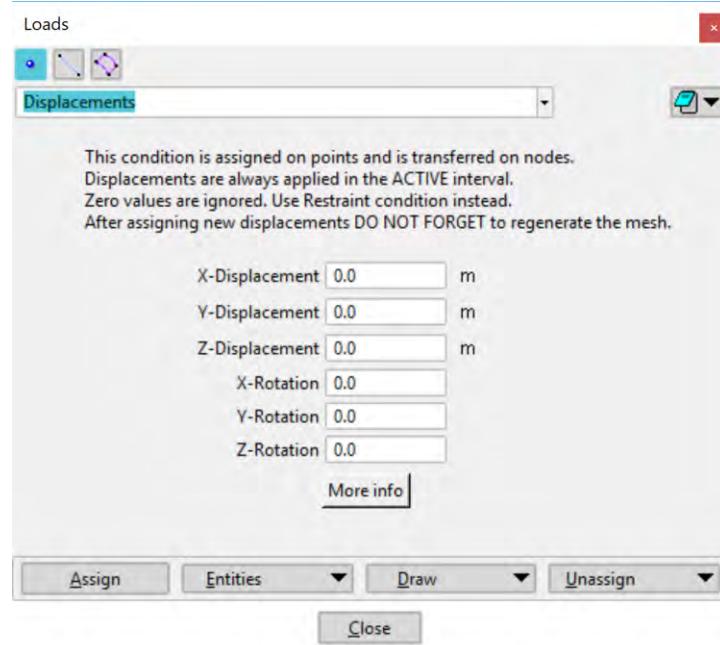


Figure 81: Nodal displacement condition

For this condition to be considered, the Loading type (Load Pattern) must be **Constant, Linear or Function** in the *Interval Data* options in the **active interval**, where it was applied.

Ground Motion from Record

Ground motion from Record is applicable only to points and is transferred to the corresponding nodes after mesh is generated. The record, which is used, is imported from the Record dialog window that can be displayed by the corresponding button. The user can select the degree of freedom at which the ground motion is imposed. A displacement ground time history is recommended to be used in contrast to acceleration.

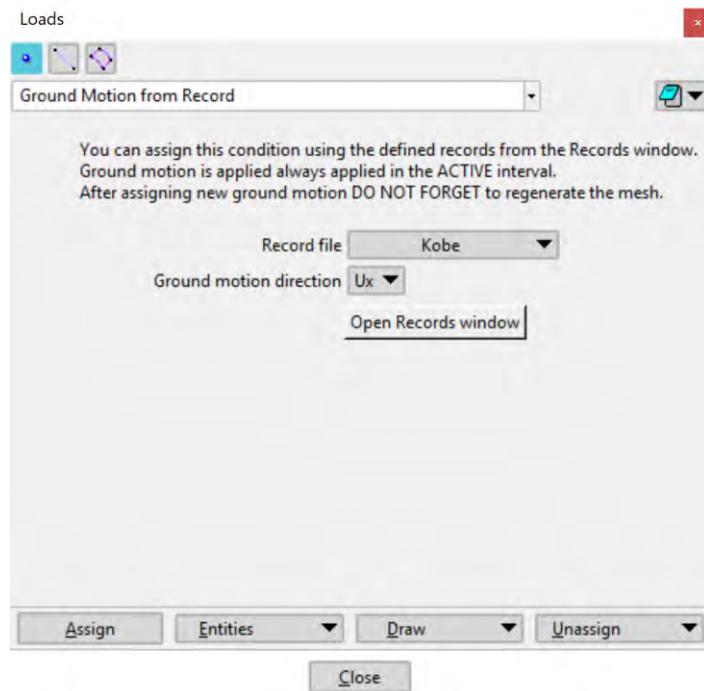


Figure 82: Ground Motion from Record condition

For this condition to be considered, **Multiple Support Excitation** must be selected for the *Loading type* in the *Interval Data* options in the active interval, where it was applied. Otherwise it is omitted by the Interface.

Sine Ground Motion

Sine Ground Motion is applicable to points and is transferred to the corresponding nodes after mesh is generated. The user can select the degree of freedom at which the ground motion is imposed, as well as the type of the excitation between *Acceleration* and *Displacement*. The parameters of the sine excitation are also user-defined such as Amplitude, Period, duration and the shift. The dialog window is shown below.

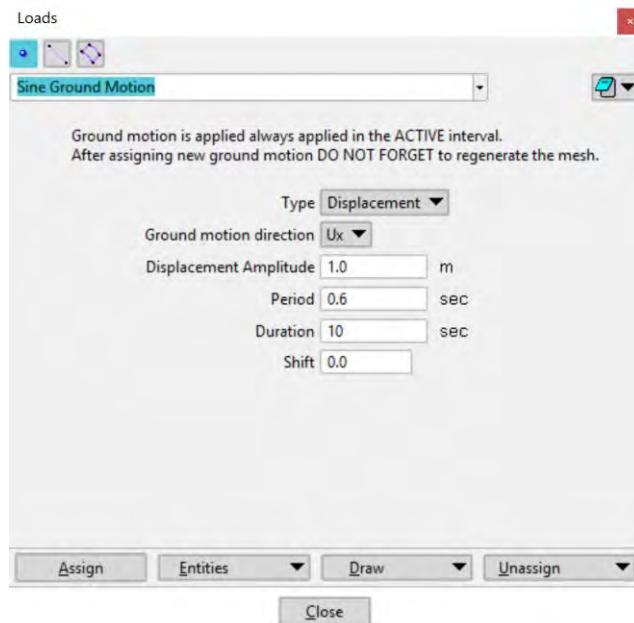


Figure 83: Sine Ground Motion

For this condition to be considered, ***Multiple Support Excitation*** should be selected for the *Loading type* in the *Interval Data* options in the active interval, where it was applied. Otherwise it is omitted by the Interface.

Masses

Mass options can be set through the Toolbar icon . Mass conditions are applicable to all geometrical entities and are transferred to the corresponding nodes. These conditions must be applied in the **first Interval** only, otherwise they are omitted.

Masses are necessary only for Transient or eigenvalue analysis and not for dead loads. Dead loads use only weight density properties or are applied as an external load pattern using force conditions.

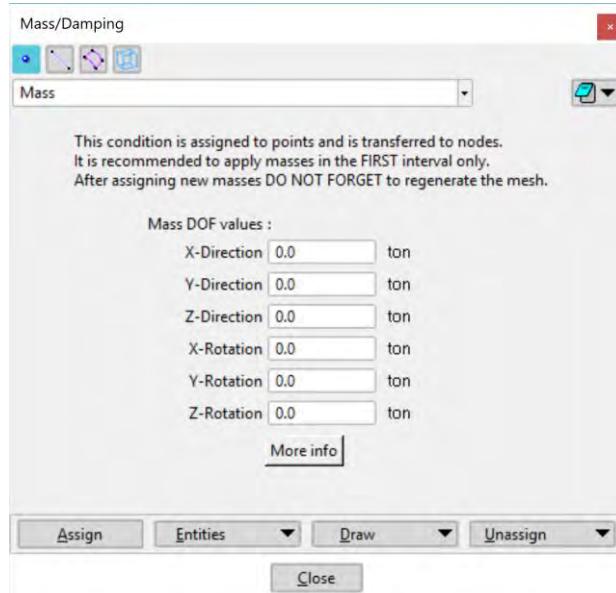


Figure 84: Point Mass condition

Rayleigh Damping

Rayleigh damping is one of the classical procedures for constructing the damping matrix for a structure from modal damping ratios.

Rayleigh damping matrix is calculated as:

$$\mathbf{c} = a_0 \cdot \mathbf{m} + a_1 \cdot \mathbf{k}$$

Where:

m: the mass matrix

k: the stiffness matrix

a_0 : The mass-proportional damping coefficient which is constant and has unit of sec^{-1}

a_1 : The stiffness-proportional damping coefficient which is constant and has unit of sec

The coefficients a_0 and a_1 can be determined from the specified damping ratios ζ_i and ζ_j for the i_{th} and j_{th} modes, respectively, from the following equation:

$$\frac{1}{2} \begin{bmatrix} 1/\omega_i & \omega_i \\ 1/\omega_j & \omega_j \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} \zeta_i \\ \zeta_j \end{Bmatrix}$$

Considering the same damping ratio ζ , which is reasonable based on experimental data, the coefficients are calculated using the formulas:

$$a_0 = \zeta \left(\frac{2\omega_i \omega_j}{\omega_i + \omega_j} \right) \quad a_1 = \zeta \left(\frac{2}{\omega_i + \omega_j} \right)$$

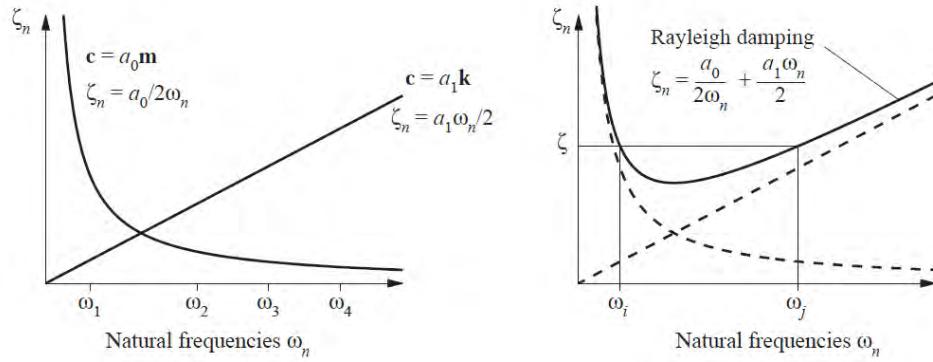


Figure 85: Variation of modal damping ratios with cyclic frequency; (a) mass-proportional damping and stiffness-proportional damping; (b) Rayleigh damping [9]

OpenSees allows the user to use the current or committed stiffness matrix, which can lead to some interesting damping in inelastic response problems in contrast to initial stiffness which is recommended by Chopra [9] and can lead to high damping forces [10]. Committed stiffness matrix is the stiffness matrix at the last time step in contrast to current which is the stiffness matrix at the current trial step in the nonlinear iteration.

Global Rayleigh damping (applied to the whole system) is a convenient and appropriate idealization if similar damping mechanisms are distributed throughout the structure. This is common in practice using Rayleigh damping for exclusively RC building or Steel buildings. Global Rayleigh damping can be set through the *General Data*. However, there are cases that a model consists of more than one systems having different damping mechanisms. In such cases, Rayleigh damping may be used separately in each system, using the corresponding condition as described below.

To assign damping to only certain elements you should use the Rayleigh damping conditions,

which can be set through the toolbar icon .

There are two types of Rayleigh damping conditions:

- Rayleigh Damping on Elements
- Rayleigh Damping on Nodes

Note: Mass proportional damping assigns damping to nodes with mass and hence OpenSees ignores mass proportional damping assigned for a region of elements. So, the mass proportional damping and stiffness proportional damping must be assigned separately.

Rayleigh Damping-Elements

These conditions are applicable to all geometrical entities except points and are transferred to the corresponding body elements after the discretization. Rayleigh damping is applied to all these elements **including the connected nodes that belong to the elements**.

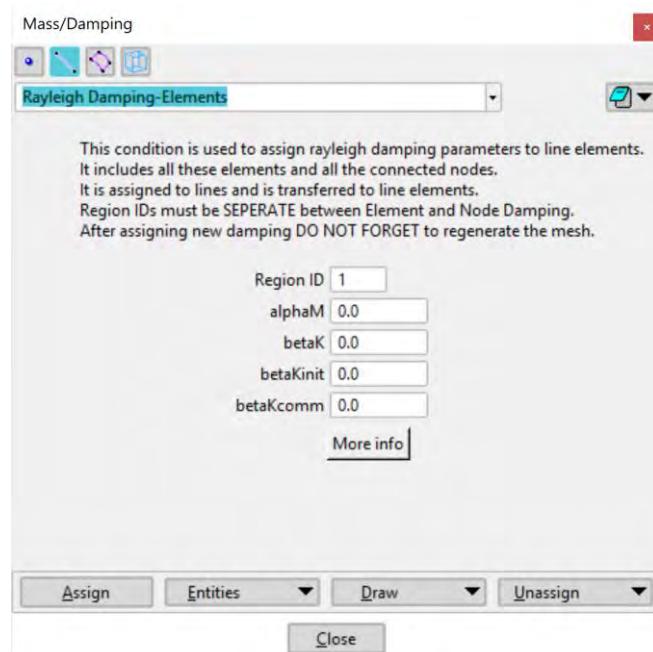


Figure 86: Rayleigh damping to linear elements condition

Rayleigh Damping-Nodes

These conditions are applicable to all geometrical entities and are transferred to the corresponding nodes after the meshing. Rayleigh damping is applied to all these nodes **including the elements of which all nodes are selected**.

Dead Loads

Since v2.2.0, dead loads can be applied to some element types as external forces **automatically**. This feature is activated through the *Interval Data* dialog window through a checkbox. It is available only for the following elements:

- Elastic Beam Column
- Elastic Timoshenko Beam
- Force Based Beam Column
- Displacement Based Beam Column
- Flexure-Shear Interaction displacement based Beam Column
- Truss
- Corotational Truss

- ShellMITC4
- ShellDKGQ

Other element types contain special properties (Body forces etc.) in their graphical window, so the dead load can be given through their definition instead of applying external forces. Dead loads for Beam Column elements are applied as uniform forces, in contrast with truss and shell elements, on which they are applied as nodal forces to their connecting nodes.

Dead loads are applied in the **negative direction of the vertical axis**. Therefore, in a 2D problem they are aligned with the negative global Y axis. In a 3D problem, they are aligned with the negative direction of the global Vertical axis that is chosen through *General Data* window.

All the above elements, contain a weight density (body force; force per unit volume) property in their dialog window and hence the uniform dead forces are calculated by the product of weight density times the cross-sectional area of the element and the nodal dead forces are calculated by the product of the weight density times the volume of the element divided by the number of element's nodes.

Fiber elements' area is automatically found either by the fiber section model they use or by the fiber section that is aggregated in case that they use a Section Aggregator. If the used section aggregator does not use any defined fiber section, the section area is assumed as zero, so zero dead loads are applied on these elements and user should apply them as external forces.

Elastic Beam Column and Truss elements' area is directly calculated from the section properties tab in their dialog windows. Truss elements' length is also easily calculated since we can know the end nodes' coordinates.

Quadrilateral Shell element's volume is calculated by the product of the section thickness with the face area. The thickness is directly found by the Section F-D they use, PlateFiber, ElasticMembrane or LayeredShell. The quadrilateral area is calculated using vectors since we can know the end nodes' coordinates. The quadrilateral is imaginary divided into two triangles. The area of triangle is calculated as:

$$A_{triangle} = \frac{|\vec{a}| \cdot |\vec{b}| \cdot \sin(\hat{\theta})}{2}$$

Where:

\vec{a}, \vec{b} are two vectors connecting two nodes each.

$\hat{\theta}$ is the angle between \vec{a} , \vec{b} .

So, the quadrilateral area is calculated as:

$$A_{quadr} = \frac{|\vec{a}| \cdot |\vec{b}| \cdot \sin(\hat{\theta}\hat{1})}{2} + \frac{|\vec{c}| \cdot |\vec{d}| \cdot \sin(\hat{\theta}\hat{2})}{2}$$

Where a , b , c , and d the vectors connecting the four nodes of the quadrilateral element.

General Data

General data can be set through the toolbar button  . In contrast with elements and conditions windows, the general data is about general information of the problem which does not concern any specific geometrical object, and is about the definition of data that is considered for the entire analysis process.

Specifically, it contains 5 main tabs. In the first one, user can use a frame for keeping notes of the project. When clicking the accept button, a copy of these notes is created inside the project directory in file *projectname.txt*. Also, any changes to the *projectname.txt* are automatically copied to the Project info tab in General Data window. The *Modeling Options* tab contains information about project dimensions and the Vertical axis that user can define (if 3D problem). Project dimensions are automatically found by the Interface searching for any point that is out of XY plane. If the geometry lays on XY plane the project is assumed as a 2D problem and the vertical axis is Y by default. Otherwise the problem is characterized as 3D and the vertical axis is user-defined between Y or Z axis.

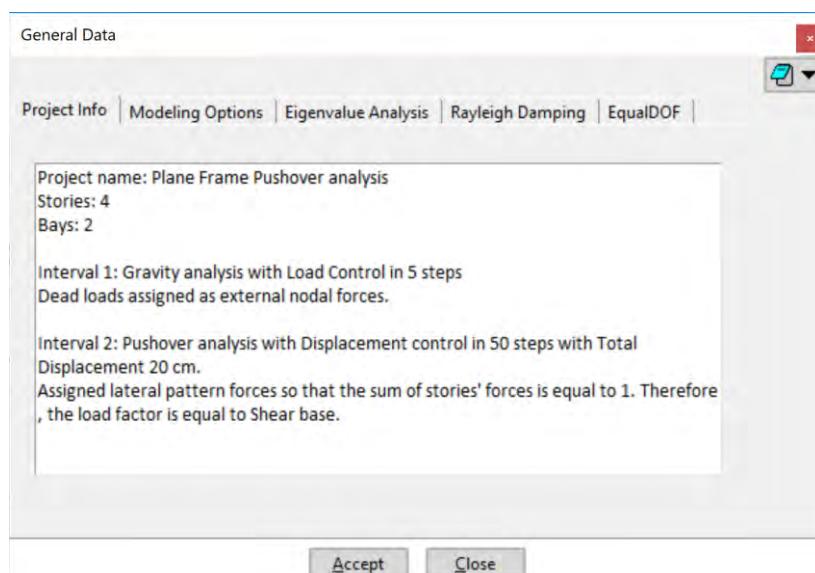


Figure 87: General Data window

Moreover, to perform eigenvalue analysis, the user should check the *Activate eigen value analysis* box in the *Eigenvalue Analysis* tab, type the number of eigenvalues he/she wants to evaluate as well as the solver to be used. Default eigensolver is the Full General Lapack solver. In the *Rayleigh Damping* tab user can define Rayleigh damping coefficients for assign them to **all defined elements and nodes**. For the stiffness-proportional damping user can choose between **current, initial or committed** stiffness matrix.

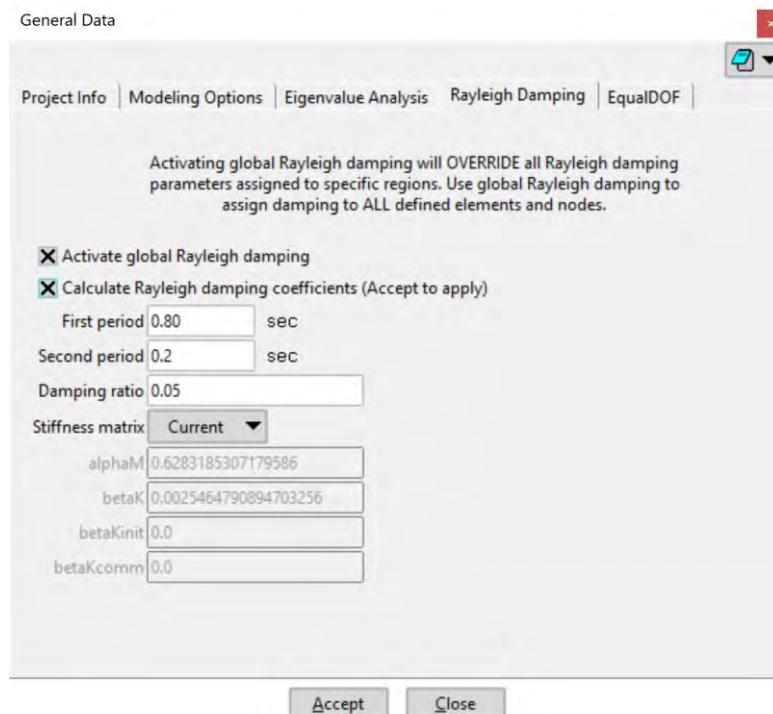


Figure 88: Global Rayleigh damping options

Note: When using Rayleigh Damping from the General Data, all Rayleigh damping parameters that has been assigned to specific regions through the corresponding conditions will be overwritten. In order to use different Rayleigh damping parameters among particular regions, user may use the relevant conditions as shown before.

Finally, the last tab is about the automated implementation of equalDOF commands between nodes which share the same location, but belong to different model domain. For instance, in a 2D problem a truss node has 2 degrees of freedom, contrary to a beam-column element's node which has 3 degrees of freedom, and as a result these two nodes should be different (not common) even in the same location. If you decided to connect them, you would assign equal constraint condition for x and y translation. This is done automatically through *General Data*, saving time for the user, especially when there a lot of this type of connection.

Note: For efficiently handling models with different model domains, that you want to connect them, use *Layer tools*.

Intervals Data

Intervals data can be set via the toolbar button . By mentioning interval, we mean a subdivision of the general problem including its analysis options and hence by mentioning **interval data**, we mean the analysis and loading options of a particular part (interval) of the whole problem. For instance, Pushover analysis consists of two parts (intervals): gravity analysis with vertical loads and lateral force analysis using displacement control, which could take place in two successive intervals.

Through the *Intervals Data* options, the user can set up a few analysis features to be implemented:

Table 5: Main Intervals options (v2.3.0)

Equations System	DOF Numberer
<ul style="list-style-type: none"> - BandGeneral - BandSPD - ProfileSPD - UfmPack - FullGeneral - SparseSYM 	<ul style="list-style-type: none"> - Plain - Reverse Cuthill-McKee - Alternative Minimum Degree - Parallel Plain
Constraint Handler	Solution Algorithm
<ul style="list-style-type: none"> - Plain - Lagrange - Transformation - Penalty 	<ul style="list-style-type: none"> - Linear - Full Newton-Raphson - Modified Newton-Raphson - Newton-Raphson with line search - Broyden - BFGS - KrylovNewton
Analysis Type	Integrator Type
<ul style="list-style-type: none"> - Static - Transient 	<ul style="list-style-type: none"> - Load Control (Static) - Displacement Control (Static)

	<ul style="list-style-type: none"> - Newmark (Transient) - Hilber-Hughes-Taylor (Transient)
Loading Type (Pattern)	Convergence Criterion <ul style="list-style-type: none"> - Norm unbalance - Norm displacement increment - Energy Increment - Relative Norm unbalance - Relative Norm displacement increment - Total Relative Norm displacement increment - Relative Energy increment - Fixed number of iterations

The constraint handler determines how the constraint equations are implemented in the analysis. The numberer determines the mapping between equation numbers and degrees of freedom. The equation system determines how to store and solve the system of equations in the analysis. Convergence criterion (test command) is used for determining if convergence has been achieved at the end of an iteration step. The algorithm determines the sequence of steps taken to solve the non-linear equation. Finally, the Integrator determines the meaning of the terms in the system of equation, and more specifically, it determines the predictive step for time $t + dt$, specifies the tangent matrix and residual vector at any iteration step and determines the corrective step based on the displacement increment dU [11]. A few other options are available depending on the user options among the main options shown above. These are given below for each analysis type.

For **Static Analysis**, the user can also select the loading path between *Monotonic* and *Cyclic* and the Integrator type between *Load Control* and *Displacement Control*. The Loading type can be *Constant*, *Linear* or *None*. At this moment, Cyclic path is considered only for Displacement Control. In every occasion, the number of analysis steps is always user-defined.

In the case that Displacement Control is selected, user must also enter the control node, the control direction as well as the total displacement/rotation to be evaluated. Moreover, if the loading path is *Cyclic*, user can enter the pair values of Displacement ratio and Number of cycles. Displacement ratios refer to the ratio between the desired displacement and the total displacement, and for every displacement the number of cycles can be set. Finally, if user wants to keep this load of the current interval for the following interval analyses, a related box *Keep this loading active until the end of analysis* is available. Finally, a special *Activate dead load* checkbox is available. Checking this, translates the dead loads to external loads and apply them on the elements according to the specific weight, the section area and the element length (if necessary). This feature is considered only for the elements, that dead load cannot be input through their definition in Elements' window options. Obviously, this is available only in case that integrator type is *Load Control*.

For **Transient analysis**, the user can select between *Newmark* or *Hilber-Hughes-Tayler* for Integrator. The Loading type can be chosen among *Uniform Excitation*, *Multiple Support Excitation*, *Function* and *None*. For the *Uniform Excitation*, user can define the type of excitation, either *Sinusoidal* or from *Record*. In order to use Records, up to 3 directions as well as the ground motions to the corresponding directions are asked to be entered. For sinusoidal excitation, the Interface supports only one direction so far and a *Sine Excitation Parameters* tab is enabled where the user can enter the required parameters for this type of excitation.

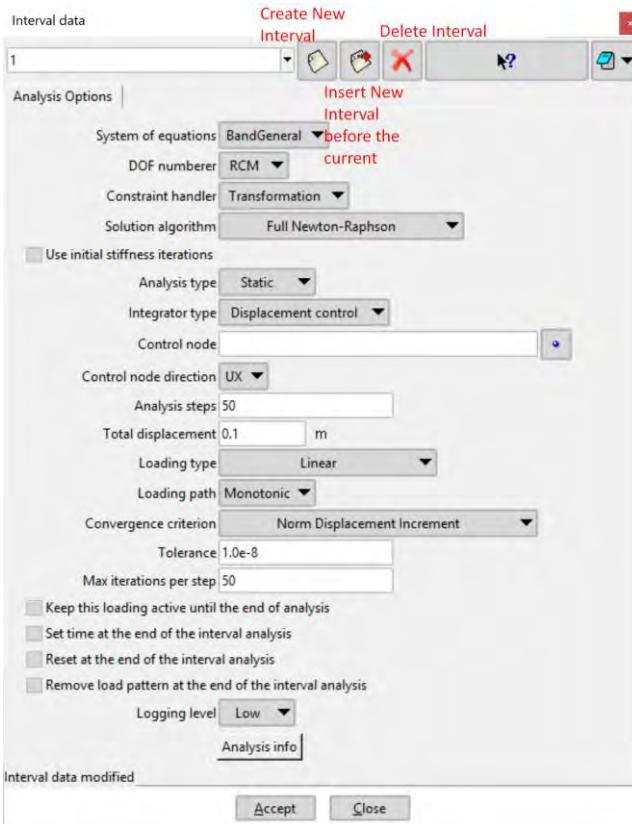


Figure 89 Interval Data options

At this point, it is important to explain the use of the boxes at the bottom of the window, where the last three of them are supported since v2.2.5.

Keep this loading active until the end of analysis: This option transforms the predefined load pattern into constant and sets the pseudo-time to zero. This is useful when the gravity loads need to remain constant for the remaining of the total analysis process. Setting time to zero is required in case that a transient analysis is following in the next interval.

Set time at the end of the interval analysis: This option allows the user to determine the time to be set after the end of this interval. This is useful when you have successive transient interval analyses. Remember that in OpenSees terms, Load patterns for dynamic analysis (e.g. Uniform excitation) use a timeseries which starts at a specific pseudo-time point. In the context of GiD+OpenSees Interface these timeseries are set to start at 0.0 sec, by default.

Reset at the end of the interval analysis: This option resets all objects to the initial state. For instance, all deformations and internal forces are set to zero. **Note that previous load patterns still exist and hence you may want to remove them enabling the next option.**

Remove load pattern at the end of the interval analysis: This option removes the pattern of this interval since interval analysis is completed and before continuing the next one.

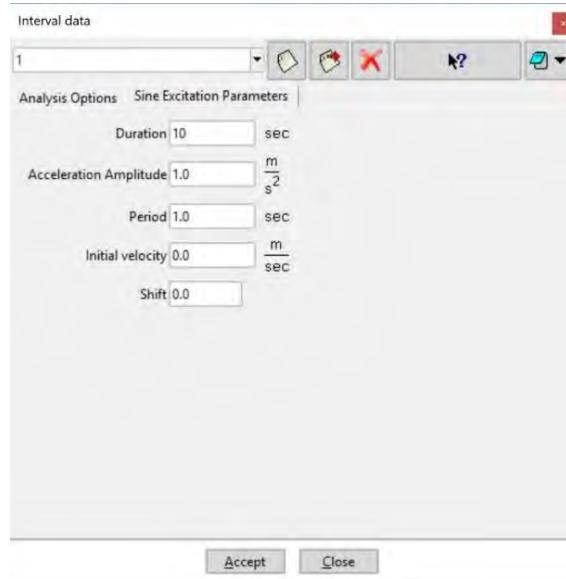


Figure 90: Sine excitation options

Output Options

Output options can be set through the toolbar (). Considering these user options, the proper recorder commands will be printed in the tcl file. After that, the output files containing the results will be created through the interpreter. Output options are separated in tabs depending on Element type as follows:

- Nodes
- Beam-Column elements
- Truss elements
- Surface element
- Solid elements

In addition, binary format is available to be used for the transformation of results to GiD Post-Processor. It is efficient in case that results' data is huge and GiD loads them a lot of faster (after the first time). Finally, user can select the output step frequency, which indicates how many step results will be transferred to GiD Post-Processor from OpenSeesPost.exe program. For instance, in dynamic analysis that analysis time step is equal to 0.005 seconds, you may want the results in every 5 (every 0.025 seconds) or 10 (every 0.05 seconds) steps in order to greatly reduce the results data size.

Meshing Options

Before creating tcl file or running the analysis, the user must first generate the mesh, which will be used for the numerical analysis later. Model body is divided into an equivalent system of many smaller bodies or units (finite elements) interconnected at nodes. GiD offers various of functions to generate various kinds of meshes. GiD uses some default options regarding what is meshed and how it is meshed, which can be easily changed through the *Mesh* Menu by the user. By default, GiD produces an unstructured mesh made of triangles and tetrahedra.

Note: To define the element type of the generated mesh, select *Mesh > Element type > Triangle/Quadrilateral* for surface elements. The proper element type depends on the Element in OpenSees terms that have been assigned for each geometrical object. For instance, a surface on which the user has assigned Quad element properties must be meshed into quadrilateral elements (4 nodes for each element). Otherwise, an error will be caused, and a proper message will be displayed.

Meshing options can be either unstructured or structured and these two categories are described as in the next subparagraphs.

Unstructured Mesh

If unstructured mesh is selected, the size of the element is given by the average side length of the corresponding mesh element. The desired size can be assigned on point, lines, surfaces or volumes and is possible to assign different sizes to different objects of the mesh. Thus, in the vicinity of these objects, the size of the elements will be pretty much of the size assigned, but not equal. Assigning the size 0.0 to an object is like setting the default size. To create an unstructured mesh, select *Mesh > Unstructured > Assign size on points/lines/surfaces/volumes*.

For instance, assume having a line object 5 meters length which will be a fiber beam element and you want a finer mesh at the edges where the inelasticity is concentrated but you do not care so much about the middle part. Assigning a size of 0.1m to points and 1.0m to the line object, will have the following result (labels are referring to nodes):

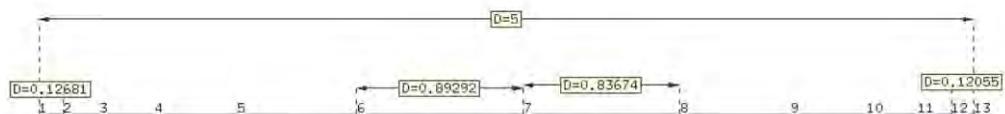


Figure 91: Unstructured mesh of linear elements

The mesh of the line object, **is not necessarily symmetric**.

Similarly, the user can generate the mesh for a surface object, so that the mesh will be finer at the sides and edges, and sparser at the core.

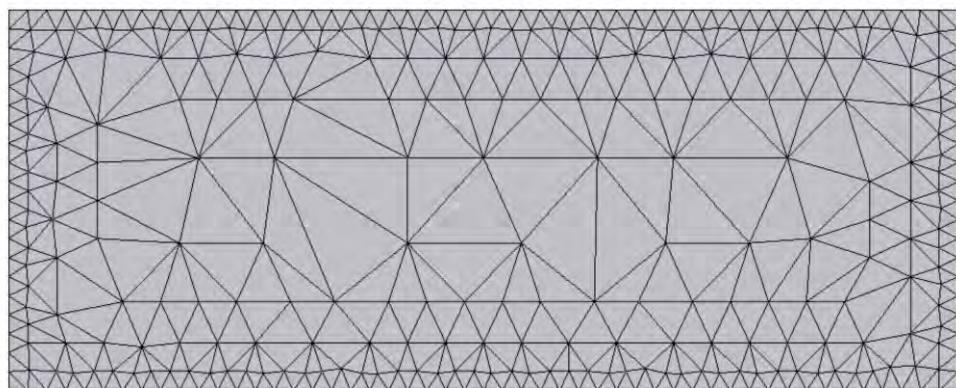


Figure 92: Unstructured mesh of surface elements

Structured Mesh

A structured mesh is characterized by a mesh where the inner nodes have the same number of elements around them. Beside the size of the elements, user can define the number of elements that are required on every line object. The line can either belong to a higher geometrical entity or not. To assign structured mesh, select *Mesh > Structured > Lines/Surfaces/Volumes > Assign number of cells* or *Assign Size*.

In addition to these options, a few additional interesting options are available. For structured mesh on lines, the user can additionally select the *concentrate elements* function, which allows to choose the start and end weight and hence the intended distribution can be easily achieved in combination with the desired number of elements. Weight takes either positive or negative values. If it is positive, the elements will be concentrated towards the extremities of the line object, otherwise they will be concentrated towards the middle and repelled from the extremities.

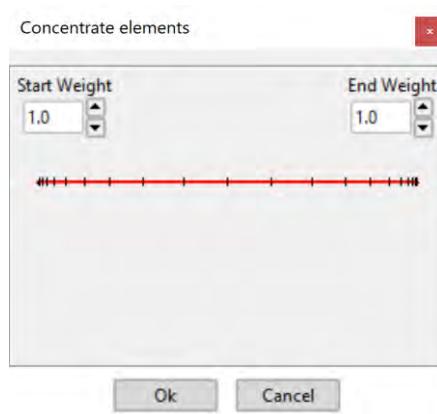


Figure 93: (Line) Concentrate elements options

So, like the previous example, a line object with 10 elements per cell and start/end weights equal to 2.0 is discretised as shown below (labels are referring to nodes):

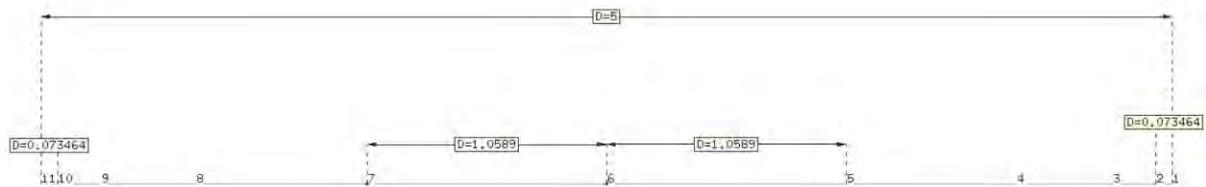


Figure 94: Structured mesh of linear element

In this case, the mesh is symmetric.

Concerning the surface objects, GiD also offers a *center structured* mesh function, which allows the user to determine the intended distribution of elements between the center and the boundaries. For this purpose, the user enters the number of concentric and perimetric divisions, as well as the two weights to concentrate the elements either in the center of structure or in the boundaries.

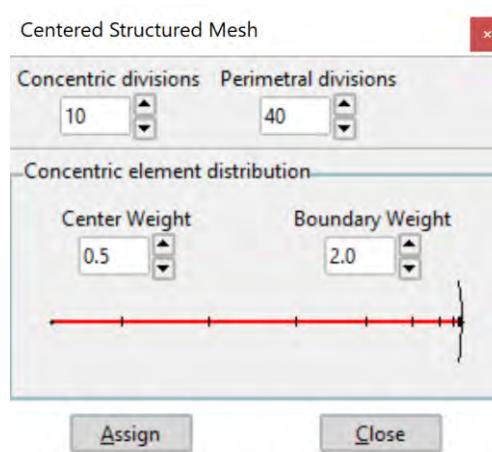


Figure 95: Centered structured mesh options

In the following figure you can see the difference between a discretised system of surface elements concentrated in the center, and another one with concentrated elements in the boundaries, for a given concentric and perimetric division.

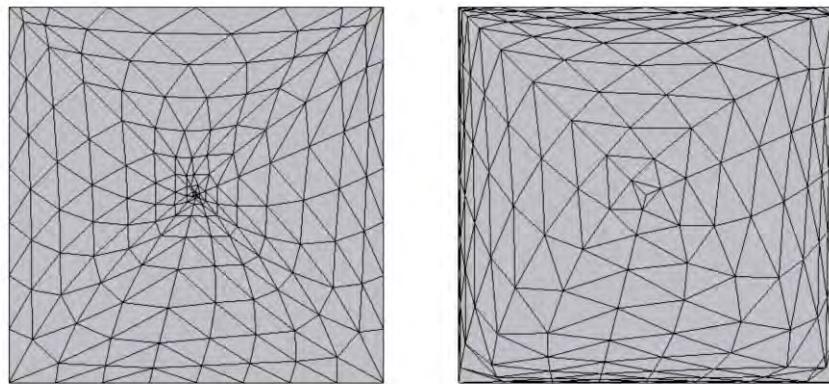


Figure 96: Structured mesh with concentrated elements in the center (left) and in the boundaries (right)

Analysis Options

The following analysis types are supported so far:

- Static Analysis
 - Linear
 - Non-linear monotonic (Pushover)
 - Non-linear reversed cyclic
- Eigenvalue Analysis
- Transient Analysis
 - Uniform excitation from records, up to three directions (Uniform excitation Pattern / Path Timeseries)
 - Multiple support excitation from records (Multiple support excitation Pattern / Path timeseries)
 - Uniform sine excitation (Uniform excitation Pattern / Sine Timeseries)
 - Multiple support sine excitation (Multiple support excitation Pattern / Sine Timeseries)
 - Function (Plain pattern / Path Timeseries)

Since version v2.2.0, Static or Transient Analysis can also be combined with ***None Loading type***. This is because, some element types include body forces through their definition, and as a result there are cases that additional loads are not required.

To perform these analysis types, the proper analysis options can be set within *General Data* and mainly within ***Interval Data*** that can be accessed via the toolbar icons and respectively.

Eigenvalue analysis

Eigenvalue analysis can be set within *General Data* in the corresponding tab, where the number of eigenvalues are selected as well as the type of solver to be used. This analysis type can be combined with all other analysis types.

All other analysis options should be worked out within *Interval Data* specifying the following fields:

- System of equations
- DOF numberer system
- Constraint handler
- Solution algorithm
- Analysis type (Static or Transient)
- Integrator type
- Loading type
- Loading path
- Convergence criterion and error tolerance
- Number of iterations per step

Some tips are following for inexperienced users for the required options that may be selected for each type of analysis.

Static Linear Analysis

To perform static linear analysis **Solution algorithm should be Linear or not**. If *Linear* solution algorithm is selected, the analysis steps should be 1 and be combined with linear elastic elements (e.g. Elastic Beam Column Elements). **Loading type** could be *constant* or *linear* and **Integrator** could be of *Load Control* type. **If more than one steps are selected, the loading type must be linear**. In cases where an interval corresponds to gravity analysis and Static Non-Linear or Transient analysis is following in the next interval, the *keep this loading active until the end of analysis* field should be probably checked. This option will keep the gravity loads constant during the remaining analysis process and set the pseudo-time to zero.

Note: Remember that many load pattern types use a starting and ending pseudo-time. Starting pseudo-time is set to zero by default.

Static Non-Linear Analysis

To evaluate nonlinear static analysis, solution algorithm should be **not** Linear (i.e. Full Newton-Raphson algorithm). Integrator type should be *Load* or *Displacement Control* and Loading Type must be *Linear*. Finally, loading path could be either *Monotonic* or *Cyclic*.

Note: Until version v2.2.5, Cyclic loading path is combined only with Displacement Control Integrator.

In case of **Monotonic Static** analysis with **Load Control** (Pushover analysis with Load Control, e.g. gravity analysis), only the number of *analysis steps* is additionally required and obviously should be greater than one (usually 5 or 10 steps are enough).

In case of **Monotonic Static** analysis with **Displacement Control** (Pushover with Displacement Control) the user must also select the *control node* for the displacement control, the *control node direction* (ux, uy, uz, rx, ry or rz), the *total displacement/rotation*, as well as the number of *analysis steps* which must be also greater than one (depending on the total displacement/rotation). The displacement/rotation of each analysis step results from division of the total one by the number of steps. The smaller displacement/rotation of each step, the more accurate results.

Note: Do not use Nonlinear elements in combination with Linear solution algorithm. The results may be incorrect and unexpected.

Note: *Control node* can be selected after mesh is generated. Otherwise, an error may be occurred during OpenSees analysis. It is worth pointing out that any changes on the geometry model or on meshing options results a different meshed model. Consequently, the Control node must be selected again.

Note: For *Constant* and *Linear* Loading Type, all Load Conditions except *Ground Motion from Record* and *Sine Ground Motion* are considered. These two conditions are obviously omitted.

Dynamic Analysis – Uniform Excitation

To perform dynamic analysis, the analysis type should be Transient, and the Integrator type should either *Newmark* or *Hilber-Hughes-Taylor*. The *analysis duration* and *analysis time step* is up to the user and depends on the sample frequency if a record file is going to be used. *Loading type* must be *Uniform Excitation*. After that, if Excitation type selected is *Record*, user can use up to three directions for ground motions. The **pre-defined** Record files should be selected as well as the directions of the ground motions. On the other hand, if *Sine excitation*

type is selected the corresponding parameters are user-defined in the enabled *Sine Excitation Parameters* tab.

Note: If Uniform Excitation Loading Type is selected, whatever Load conditions have been assigned through the condition windows are omitted.

Dynamic Analysis – Multiple Support Excitation

The only difference with Uniform excitation is the *Loading type*, which should be *Multiple Support Excitation* for this analysis type. If so, the Load pattern will contain only the Loads from the relevant conditions *Ground Motion from Record* and/or *Sine Ground Motion*. All other load conditions are omitted.

Dynamic Analysis – Function

This analysis type uses a **Plain pattern** which consists of forces and/or imposed displacement in combination with **Path Timeseries**. A proper Path timeseries can be pre-defined within Records window using a file. All other loads are ignored. For instance, this analysis type is usually observed in geotechnical engineering problems (SFSI, site response etc.) where the excitation is simulated as dynamic nodal force represented from the velocity ground time history times a proper damping coefficient.

Logging level

During the Analysis, a log file is created which contains some warning and error messages. In Interval Data, there are three logging levels:

- **Low:** print nothing
- **Medium:** print information on norms and number of iterations at end of successful convergence error test
- **High:** print information on norms each time convergence error test is evaluated.

Note: To efficiently reduce the analysis time, low logging level is recommended.

GiD+OpenSees menu

The available options in GiD+OpenSees menu are the following:

- **Create .tcl, run analysis and postprocess:** Analysis is executed straightforward in one step. A new tcl file is created based on user options inside GiD Pre-Processor (overwrites any existing). Furthermore, **OpenSees** is automatically invoked for the analysis process, so that results output files are created (overwrites any existing with same name from previous analyses). Finally, **OpenSeesPost.exe** is invoked for the

transforming of the results in a GiD-format file *projectname.post.res* in the project directory and GiD automatically loads it, asking then user to go over GiD Post-Process environment.

- **Create .tcl only:** A new .tcl file is created according to user actions inside GiD Pre-Process environment. Existing .tcl script is then overwritten.
- **Create and view .tcl only:** In addition to the above option, .tcl script is automatically opened by the user-defined editor program (e.g. Notepad++) after being created.
- **Run analysis only:** OpenSees is invoked to run the **current** .tcl script in *OpenSees* folder inside the project directory.
- **Postprocess only:** OpenSeesPost.exe is invoked and transforms the **current** results output files for GiD Post-Processor.
- **Run analysis and postprocess:** This is a combination of the two options aforementioned.
- **Reset analysis:** *OpenSees* folder in project directory, that may include the .tcl script, the log file as well as the results output files, is deleted. Furthermore, the *projectname.post.res* file in project directory, which is used for post processing is deleted.
- **GiD+OpenSees Site:** This option links to the main Website of the GiD+OpenSees Interface using the default browser. General information is provided there as well as links connecting user to YouTube channel, GitHub repository, Interface Wiki and so on.
- **OpenSees Site:** Through this option, user can link to the home of the official OpenSees Website. All the newsletters about OpenSees can be found there, as well as links to the OpenSees source code, OpenSees Wiki and more.
- **OpenSeesWiki:** This option directly links user to the OpenSees Wiki. All the details for using or developing in OpenSees are provided there.
- **DesignSafe-CI Site:** This option links user to DesignSafe-CI Website. DesignSafe is a web-based research platform of the NHERI Network that provides the computational tools need to manage, analyze and understand critical data for natural hazards research.
- **DesignSafe-CI User Guide:** This is a guide for OpenSees users on how to combine the GiD+OpenSees Interface with DesignSafe-CI platform especially for large models that are computationally expensive.

- **Check for Update:** An external program *CheckForUpdate.exe* is invoked, and compares the current user GiD+OpenSees Interface version with the last one released on official GiD+OpenSees Interface GitHub repository. A proper window is displayed informing user about either it is up to date or not. If it is not, a Download button is provided for linking user to the Download Website. **This option needs internet connection in order to compare versions.**
- **About:** The GiD+OpenSees logo is displayed, including the Interface version number, the Project Coordinator, as well as the initial Development Team.

Editing the tcl file

It is worth noting that, the tcl script is **editable** from user and hence it is tried to be created in a hand-made way for easy manual input data addition in case of features that are not supported so far or to overleap Interface weaknesses, if any. The tcl script is pursued to be simple and easy-to-read containing guidelines (comments) that would be helpful for OpenSees beginners to better learn simulating with OpenSees. For further modifications of the tcl file the *GiD+OpenSees* menu tab should be accessed. After *creating the tcl file* (without running analysis immediately), any manual changes or additions can take place and then the *Run Analysis* or *Run analysis and post process* options can be chosen through the same menu tab. In this way, the tcl script will not be overwritten and the current one (modified) will be executed for the analysis purpose.

The tcl file is created following a particular structure, which is further discussed in the following section.

Structure of tcl input file

We remind that the OpenSees tcl input file is obtained from the modeling, the various loading and analysis and the discretization that take place within GiD Pre-Processor. Although the tcl file contents depend absolutely from the user activity, it follows a concrete structure that is explained here.

The tcl file structure contains **4 main parts** with the following order:

- **Model definition (Geometry, Materials, Elements)**
- **DOF's equations (If necessary)**
- **Recorders definition**
- **Loading and Analysis Options**

Model Definition

Each Element type may have different number of degrees of freedom per node, consequently in OpenSees and FEA terms, they should be defined in different model domains. Thus, the first action is to determine how many model domains are going to be defined in the tcl file. For example, in a 2D Soil-Foundation-Structure Interaction problem we usually use Beam Column Elements and solid surface elements, such as Quad that represent the structure and soil, respectively. In this case, two model domains would be defined: 2D-3Dof's and 2D-2Dof's respectively. The connections between the nodes of different Element types would be later with ***equalDOF*** commands (see below).

For each Model domain, the following features are implemented (if selected by the user):

- Nodes definition
- Restraints
- Constraints
- Mass definition
- Materials/Sections F-D definition
- Geometric transformation definition (if frame elements exists in model domain)
- Elements definition (belong on this model domain)

DOF's equations

After the definitions of model domains and their contents, *equalDOF* commands (Equal constraints) are implemented once. Equal constraints are either used for nodes that belong in the same model domain (same degrees of freedom) or for nodes that belong in different ones. It is independent of model domains and that is why it is implemented after all model domains have been defined.

Since v2.2.5, two new features are included about *equalDOF* commands. The first one is enabled within *General Data* options and triggers the automatic generated *equalDOF* commands between nodes which share the same location, but are part of elements with different number of degrees of freedom per node. The second one, is activated within Quad/QuadUP elements' windows for generating *equalDOF* commands between the end-nodes which share the same vertical (Y) location. **Note that the user should define proper mesh options in order to make this feature work as expected.**

Recorders definition

In order to record the intended results during the analysis and they be printed to output files, recorder commands are used. The user chooses what results he/she wants to be monitored via the Pre-Processor within *Output options* dialog window. Recorder definitions are implemented before the loading and analysis parameters definition.

Loading and Analysis Options

For each interval: Loading and Analysis options are implemented in the tcl file. **Every interval consists of a set of analysis options and/or a load Pattern.** If None Loading type is chosen in Interval Data options, only Analysis options are considered.

Processor

OpenSees is invoked for the analysis process after user-defined options and geometry simulation within GiD Pre-Processor. Depending on the user's options, a tcl file is generated (exported) from GiD that is given as input data for OpenSees solver. It is all about a file exchange (connection) of the two independent software mentioned above.

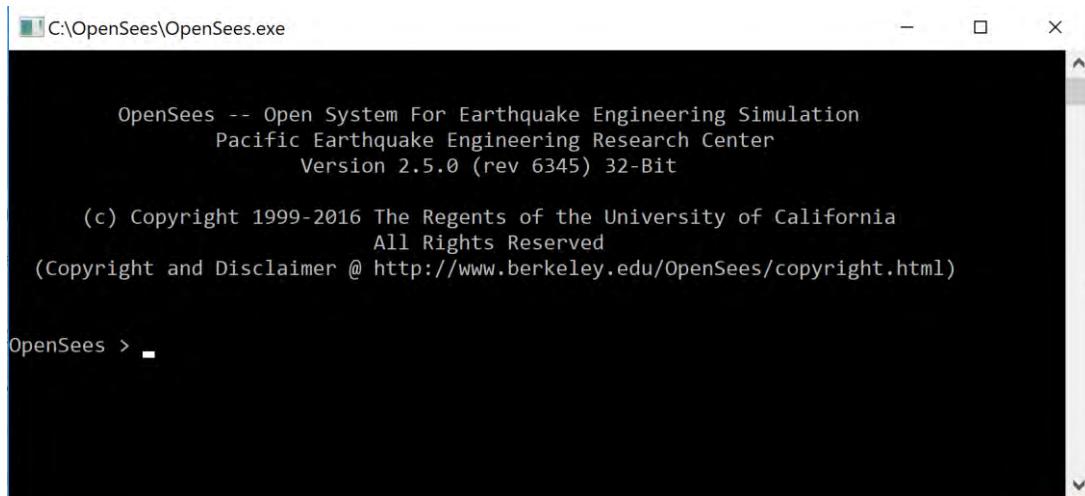


Figure 97: OpenSees DOS shell

OpenSees to GiD converter

The output files containing the results are automatically translated to the required GiD compatible format and imported in the GiD Post-Processor environment, where all standard visual and graphic tools are available. For further information about Post-Processor, see the next subparagraph.

GiD can import the results produced by the OpenSees, so GiD and OpenSees communicate through the transfer of files. In order to achieve this file communication, an external program has been developed (*OpenSeesPost.exe*) which translates the produced numerical results to a file that has the extension (file tail) *post.res* in the project directory, which is subsequently read by GiD and results are imported in the post-processor environment. Specifically, when analysis is completed, *OpenSeesPost.exe* searches for output files in the *OpenSees folder* inside the project directory *projectname.gid* and therefore it translates and transforms them in the *projectname.post.res* file.

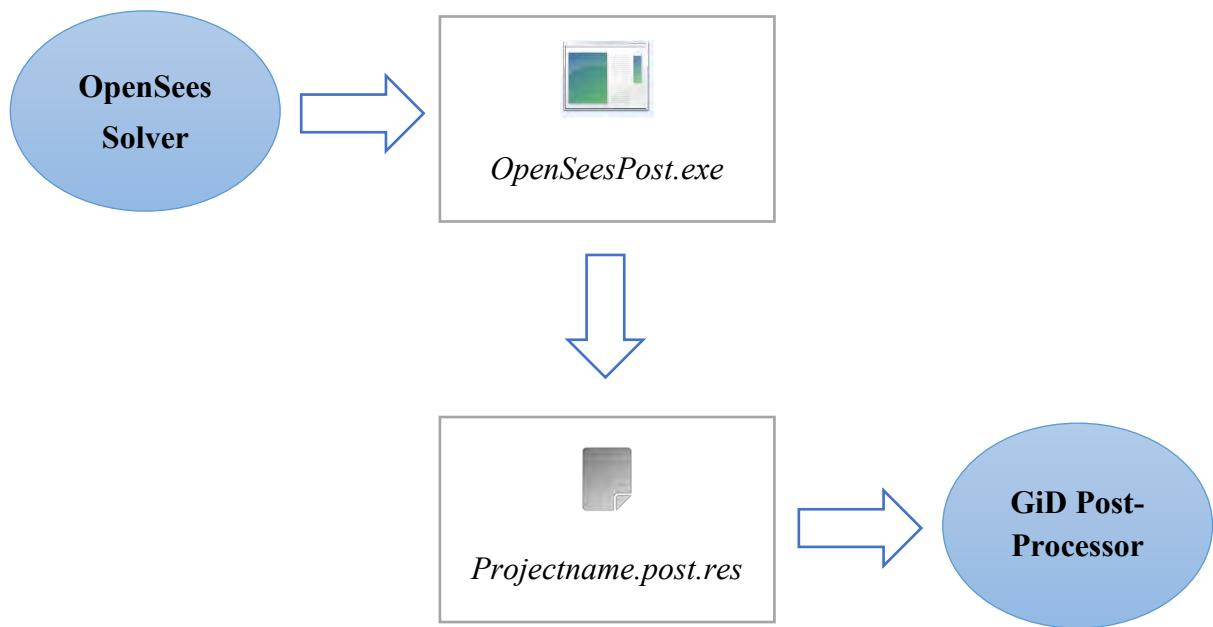


Figure 98: Results conversion and transfer

```

C:\Program Files\GiD\GiD 13.0.1\problemtypes\OpenSees.gid\exe\OpenSeesPost.exe
OpenSeesPost - OpenSees to GiD results converter
https://github.com/rclab-auth/gidopensees

Model file : Example - Plane Frame on Elastic Soil - Dynamic Analysis.tcl

Reading frame element local axes
Reading mode 1
Reading mode 2
Reading mode 3
Reading load factors / time (4007)
Reading nodal displacements (4007)
Reading quad stresses (2766)
  
```

Figure 99: OpenSees to GiD results converter

Post Processor

GiD Post-Processor receives the mesh information from the GiD Pre-Processor. That means model geometry must be strictly created in Pre-Processor environment and not manually because errors will be occurred, and results will may not match in the mesh model properly. On the other hand, the results are received from the solver module through a specific file *projectname.post.res* that is produced by the external program *OpenSeesPost.exe*.

For the transition to the Post-Process environment, you can select the related button function at the upper right corner. GiD automatically searches for the *projectname.post.res* file in the

project directory for importing the results and hence for an existing project with existing produced results, you do not need to run the analysis again of course.



Figure 100: Toggle between pre and postprocess

Similarly, a related button option is available for the transition to the Pre-Processor from the Post-Processor.

So far, the element results that can automatically be generated and transformed by the Interface are given in the following table:

Table 6: Element results (v2.3.0)

Elastic Beam Column	Elastic Timoshenko Beam Column
- Local Forces	- Local Forces
Force-Based Beam Column	Displacement-Based Beam Column
<ul style="list-style-type: none"> - Local forces - Basic deformation - Plastic deformation 	<ul style="list-style-type: none"> - Local forces - Basic deformation - Plastic deformation
Flexure-Shear Interaction Displacement-based Beam Column	Truss
<ul style="list-style-type: none"> - Local forces 	<ul style="list-style-type: none"> - Axial force
Corotational Truss	Quad
<ul style="list-style-type: none"> - Axial force 	<ul style="list-style-type: none"> - Forces - Stresses - Strains
QuadUP	Tri31
<ul style="list-style-type: none"> - Forces 	<ul style="list-style-type: none"> - Forces - Stresses
ShellMITC4	ShellDKGQ
<ul style="list-style-type: none"> - Forces 	<ul style="list-style-type: none"> - Forces

<ul style="list-style-type: none"> - Stresses 	<ul style="list-style-type: none"> - Stresses
Standard Brick	
<ul style="list-style-type: none"> - Forces - Stresses - Strains 	

At the moment, any other results that are related with elements, such as stress-strain of some fiber in a fiber-element should be handled manually by editing properly the tcl file that is generated by the GiD Pre-Processor.

The nodal results that can automatically be generated and transformed by the Interface are:

- Nodes
 - Relative/Absolute Displacements
 - Rotations
 - Nodal Reactions
 - Relative/Absolute accelerations
 - Relative/Absolute velocities
 - Eigenvectors (if eigenvalue analysis is activated)

View Results and Deformation

To view the results in several ways including Contour fill, display vectors, line diagrams etc.

user should select the *View results* menu tab or the *Results Window* (). In the first tab *View Results* user can choose the View type, the interval number and the step for the specific result to be displayed in the graphical environment. Inside the *Main Mesh* tab, the user can define the model deformation at which the results will be displayed on as well as a scale factor for the best view experience. These options are shown at the following figures.

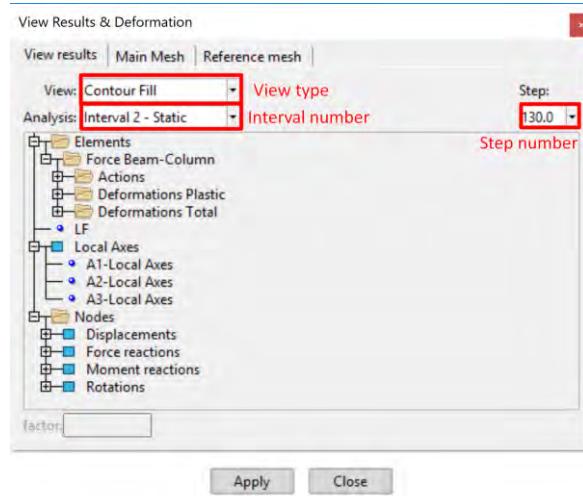


Figure 101: View Results window – View Results tab

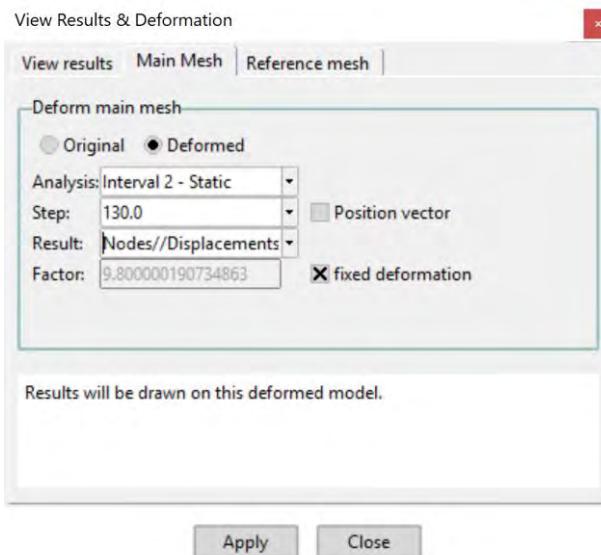


Figure 102: View Results window – Main Mesh tab

If you want to display results on the deformed shape comparing to the initial shape at the same time, you handle it either through the reference mesh tab of the *View Results & Deformation* or through the *View style* options (). View style function will enable you the following window shown in Fig. 103.

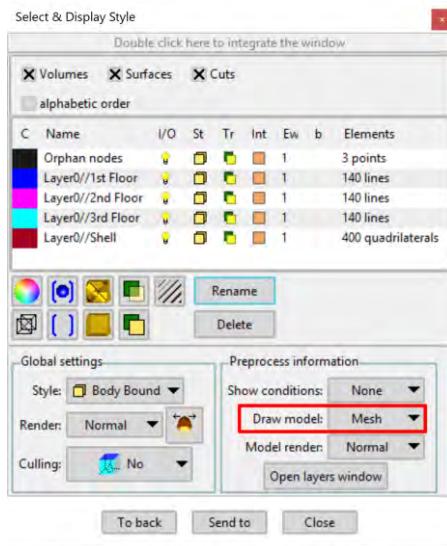


Figure 103: View style options

On the *Draw model* option, you can choose the *Geometry* or *Mesh* model to be displayed. *Model render* option is also available for modification as well as showing the conditions applied on the Pre-Processor. The other options are not important to be described for the purposes of the Interface.

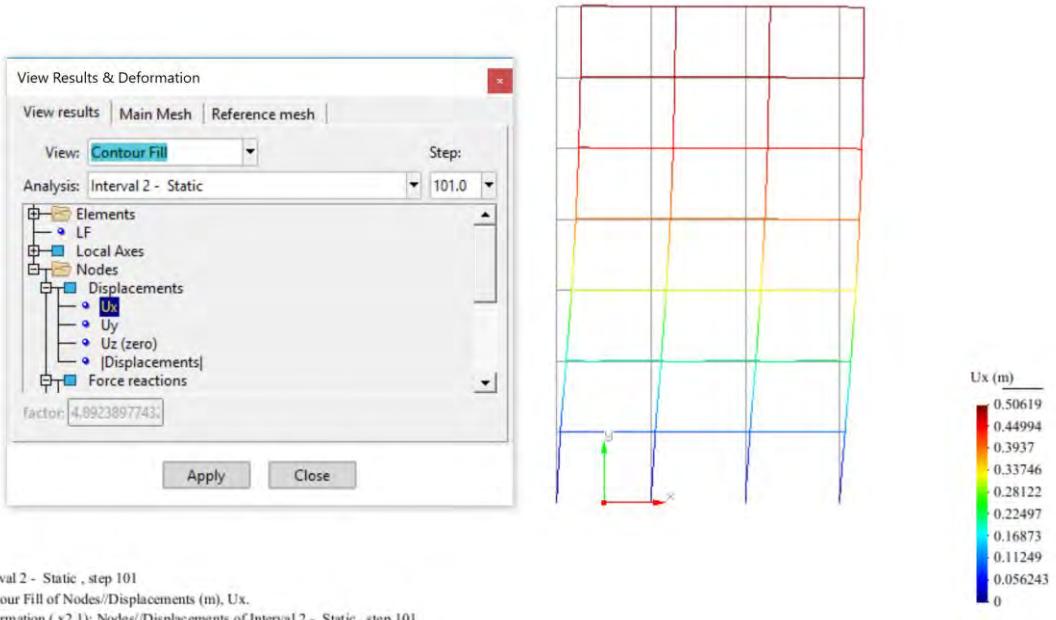


Figure 104: 2D frame – Deformed shape

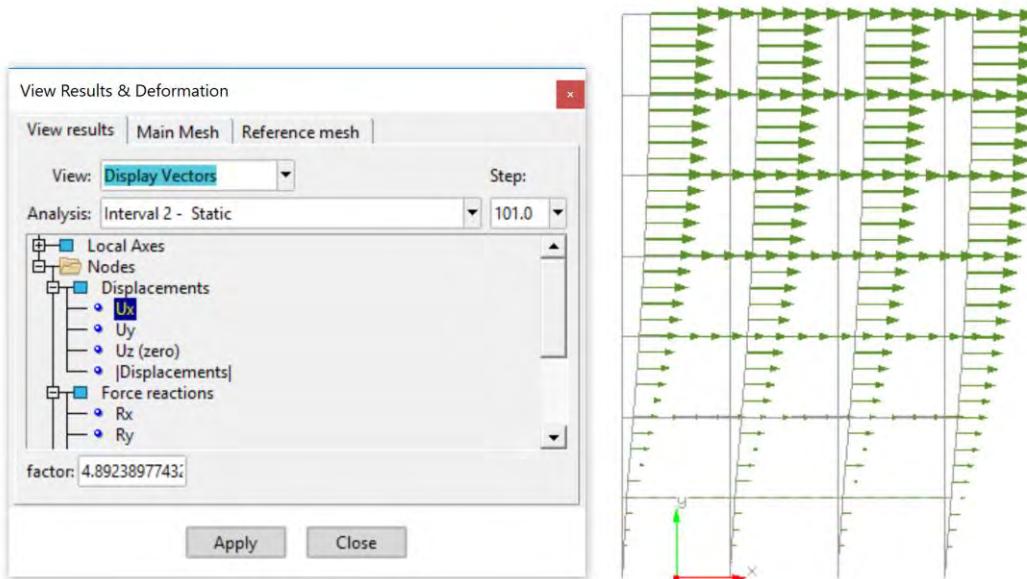
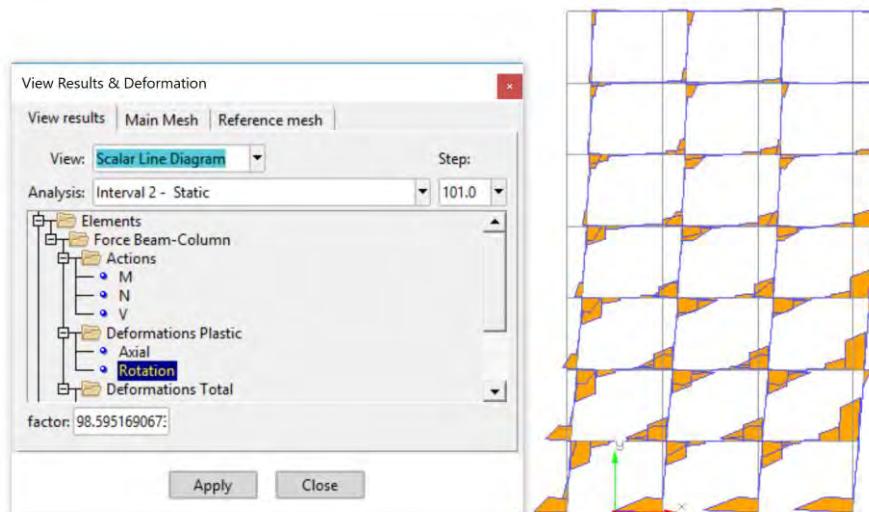


Figure 105: Display displacement vectors – ux component



Interval 2 - Static , step 101
 Scalar Line Diagram of Elements//Force_Beam-Column//Deformations_Plastic//Rotation factor 98.595.
 Deformation (x2,1): Nodes//Displacements of Interval 2 - Static , step 101.

Figure 106: Line diagram; Plastic rotation deformation

So far, the internal actions are not displayed as we know from the structural analysis theory which is based on the side that is in tension. Through the *OpenSeesPost.exe*, some internal force values' signs are changed on one element's end-node for every linear element so that the relative values of the elements' end-nodes are correct. That is, if at two frame elements' ends the moment is opposite, as we know from Structural analysis theory, OpenSees prints them with the same sign which is correct based on the FEA terms and local axes' orientation. In addition,

GiD displays the diagram based on the direction it was designed, making the procedure complex. Thus, through the *OpenSeesPost.exe* the internal forces' signs undergo a conversion:

- Moment and Shear forces at the beginning of the element **hold** sign
- Moment and Shear forces at the end of the element **change** sign
- Axial and Torsion forces at the beginning of the element **change** sign
- Axial and Torsion forces at the end of the element **hold** sign

The axial and Torsion forces are always easy to understand. You can know if axial is compressive or tensile by the sign and similarly how the torsion is applied using the right-hand rule. As far as Shear and Moment forces are concerned, since you know the local axes orientation and hence the start node and end node, we can easily understand how they are applied based on structural analysis theory.

Plot Graphs

A very useful visualization tool that GiD provides is Graphs, within which we can visualize results in several ways. *Graphs window* can be accessed from the toolbar command  . The most useful functions for Civil Engineering problems are the *Point Graph*, *Point Evolution*, *Border graph* and *Line graph* view types. Graphs can be managed into graph sets and each set shares the same units for each axis. You can combine more than one graphs of the same type in the same graph set for a specific nodal result. Graphs are supported only for results defined over nodes.

Using *Point Evolution* type, the user can observe the evolution of the selected nodal result along all the steps of a specific Interval (i.e. the x-displacement of node 2 during the Interval 2 Static analysis).

Using *Point Graph* type, the user can observe a nodal result against another specified nodal result, in a single step or including all steps in a specific interval analysis. For example, you should use Point Graph including all steps to observe response time-history or Pushover curve. In the first case, the x component is the Load Factor and in the second the Time.

Line graph displays a graph defined by the line connecting two chosen nodes. The horizontal axis corresponds to the x, y, z variation (distance of the component of the two selected nodes) or line variation (distance between the selected nodes along the line connecting them). The vertical axis corresponds to the specified nodal result.

Border graph displays a graph of the nodal results on the selected border (line path) using the x, y, z or line variation (distance) for the horizontal graph axis. The vertical axis corresponds to the specified nodal result.

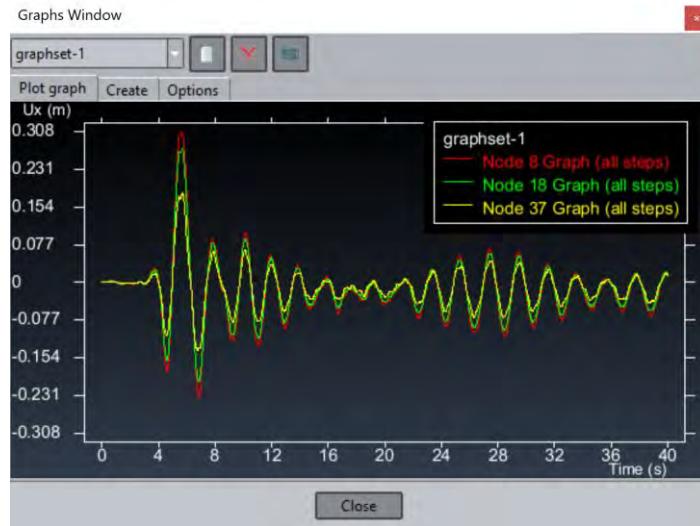


Figure 107: Point graph (includes all steps); Displacement time history response of 3 stories

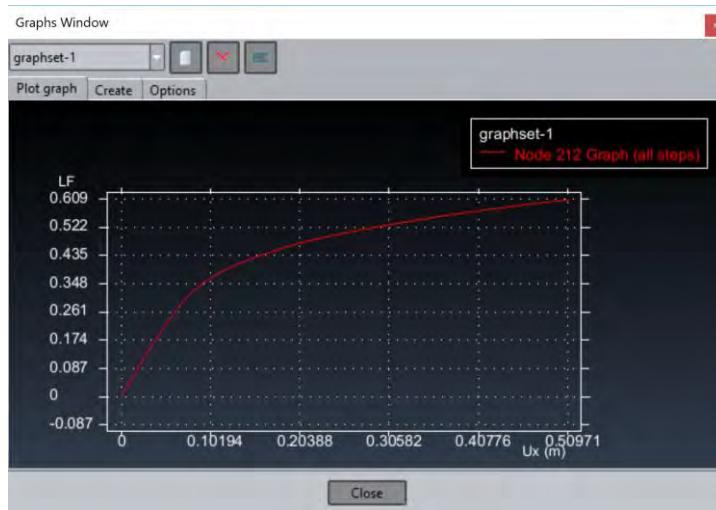


Figure 108: Point graph (includes all steps); Pushover curve

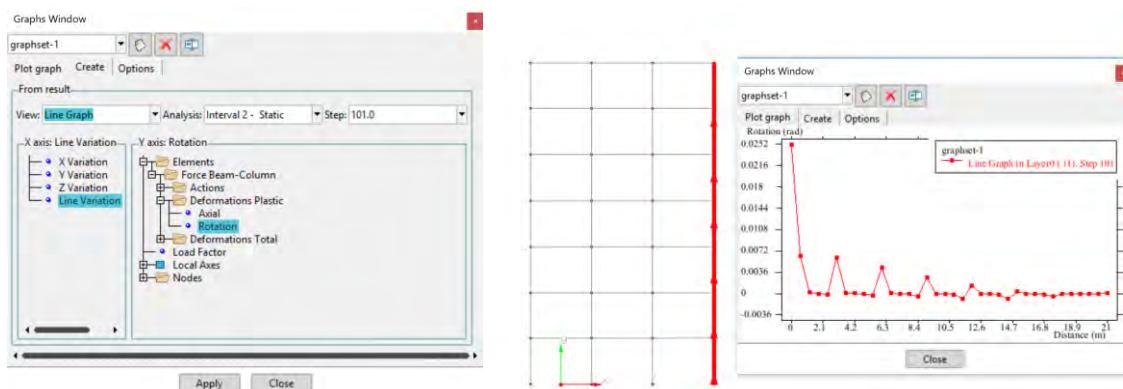


Figure 109: Line graph - Plastic rotations

Through the *Options* tab, the user can handle the results in a table format, and subsequently to copy the column values to any external software such as an excel sheet for further processing. After choosing a specific Graph, a *Show table* function is enabled. This option transforms the graph results in a convenient table format.

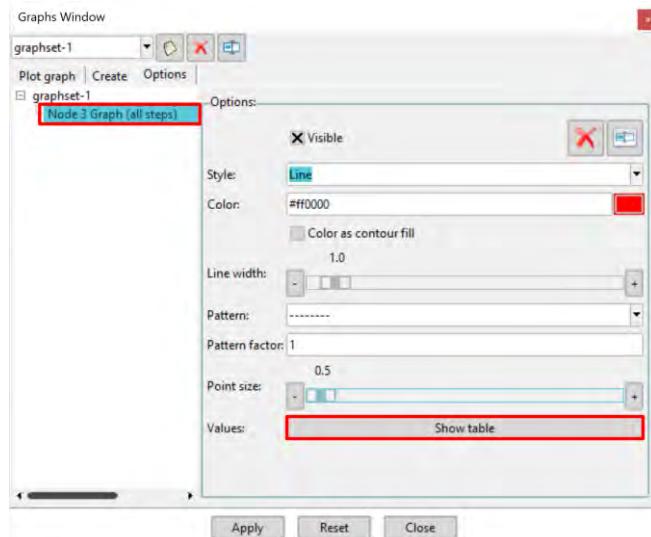


Figure 110: Graphs window - Show table option

Graph values		
	Node 3 Graph (all steps)	
	Time	Ux
	s	m
1	0	0
2	0.005	2.08416e-007
3	0.01	1.24421e-006
4	0.015	3.94038e-006
5	0.02	9.14774e-006
6	0.025	1.7705e-005
7	0.03	3.04343e-005
8	0.035	4.81686e-005
9	0.04	7.17493e-005
10	0.045	0.000101993
11	0.05	0.000139709
12	0.055	0.000185743
13	0.06	0.000240956

Figure 111: Results in table format

Animation

GiD Post-process environment also includes animation capabilities *within Result animation window* (). This feature allows the user to create an animation of the current Results View,

where the limits can be fixed during the animation and/or an animation of the Deformation of the mesh.

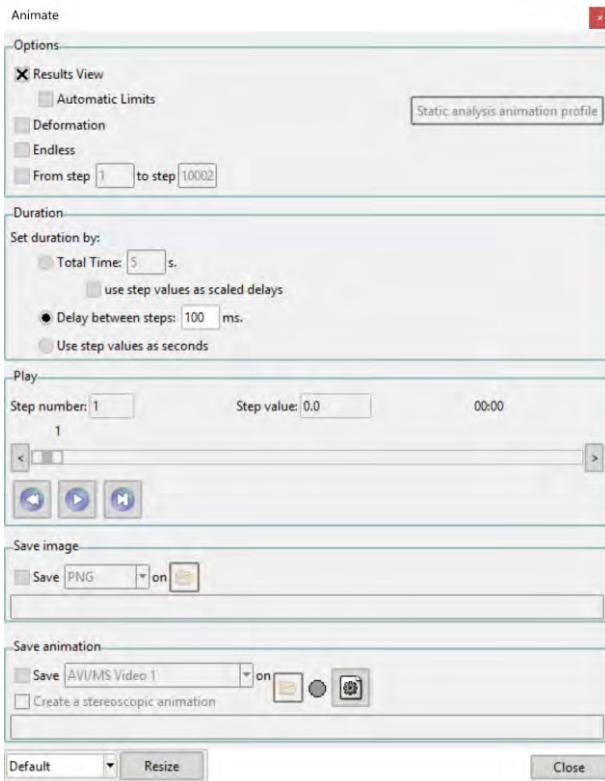


Figure 112: Results animate window

Results view: GiD shows the results pre-defined by the user (e.g. Displacement ux with contour filling)

Automatic Limits: It is accessible only if Results view option is checked. GiD searches for the maximum and minimum value of the results along all the steps of the analysis and uses them to draw the results view during whole the animation (all steps).

Deformation: Record animation with a specific deformation that is pre-activated by the user.

Endless: The animation continues indefinitely.

From step I to step J: The animation is limited between two specified step numbers, both included.

Total time: The user can specify the clip total duration.

Delay: The user can specify a delay time between the steps. It is input in milliseconds.

Use step values as seconds: The number of steps are used for the duration of the clip.

There are 3 main buttons under the slide bar in *Play* group options; The first one Rewinds the animation, the second one Plays/Stops the animation, and the third one advances the animation to the next step.

Save image: The user can save snapshots of the animation, in PNG, TIFF, JPEG or GIF format, of **each step** when the play button is pushed

Save animation: A movie clip is save by pressing play with selected compression method.

Quick Start

Tutorial 1 – Static and Modal Analysis of a Three-Story Building

Description

This tutorial example describes the GiD+OpenSees implementation of modal and Static analysis of a three-story building. The floor plan is identical along the height of the building and is depicted in Fig. 113. The ground floor height is 4 meters, while the rest of the floors' heights are 3 meters long. The concrete strength class is considered as C20/25 according to Eurocode 2. The requested task is to evaluate the 9 first modes as well as the deformation and internal member forces for lateral static forces representing seismic forces.

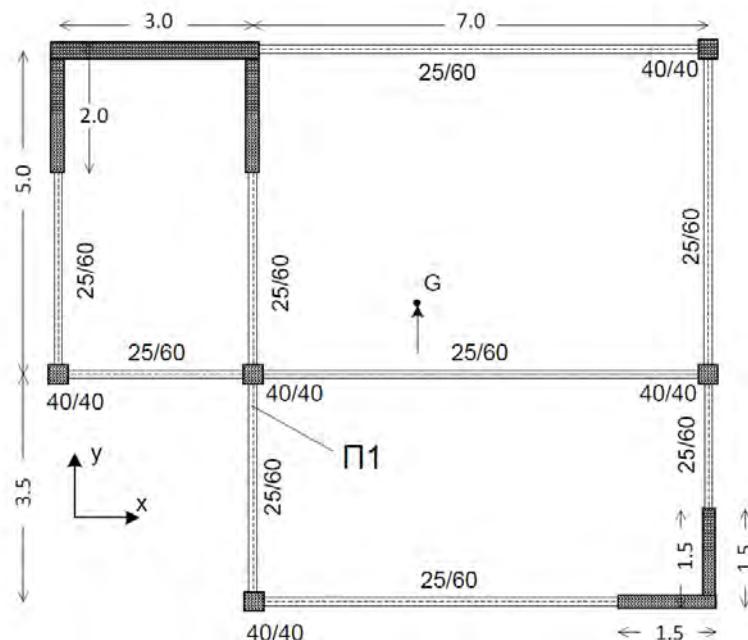


Figure 113: Floor plan view

Problem solution

First, the plan view is created in the XY plane, using the grid lines. Grid options are enabled and modified through the *Preferences* options as shown below.

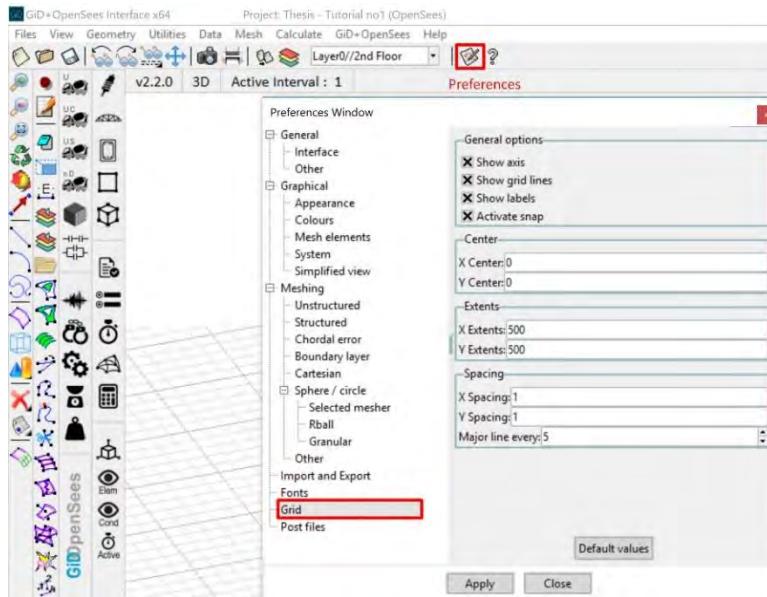


Figure 114: Tutorial 1 - Grid options

The Beams are designed with line objects (■) on grid lines. The geometric center point is created through *Geometry > Create > Point*. The two other plans are created using the *Copy* command from *Utilities* menu with relative translation 3 meters. Columns are created using Copy command with *Collapse* option enabled as well as *extrude lines* is chosen. In this way, column lines are creating from points, and intersection points are merged.

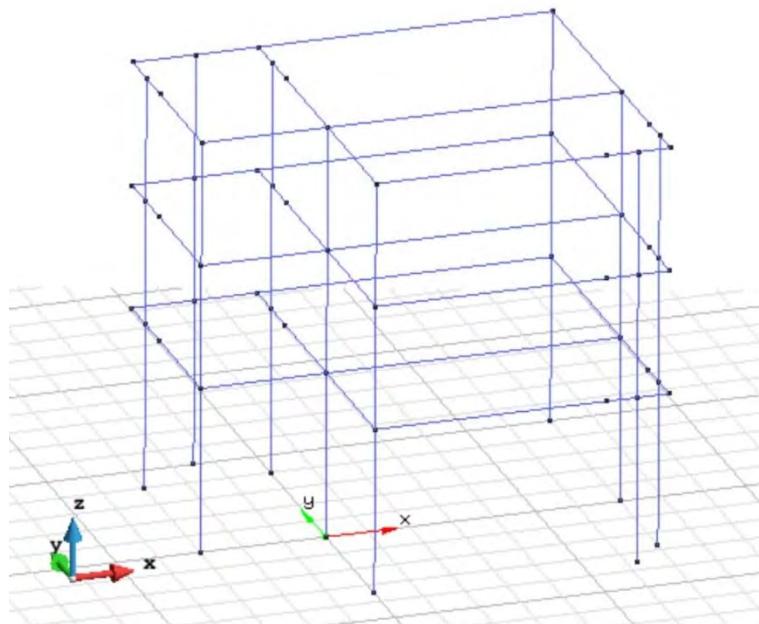


Figure 115: Tutorial 1 – Geometry model

You should access *General Data* options and select *Vertical Axis Z* in *Modeling options* tab.

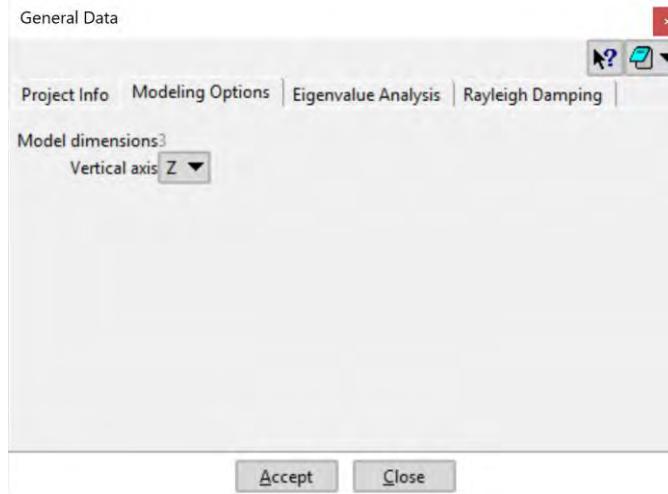


Figure 116: Tutorial 1 – Setting Z as vertical axis

All the base nodes are supposed to be built-up. Thus, you should fix all 6 degrees of freedom using the Restraints condition applied on points (☞).

After that, you need to create the points that are going to be the master nodes for the rigid diaphragms on each floor level. You could create the points (*Geometry > Create > Point*) at the geometric centre of each floor.

Remember that Elastic beam column elements use the Young Modulus from an Elastic Isotropic material. So, before creating the elements with their cross-sectional characteristics, you should define the elastic Concrete material (砼) as shown below.

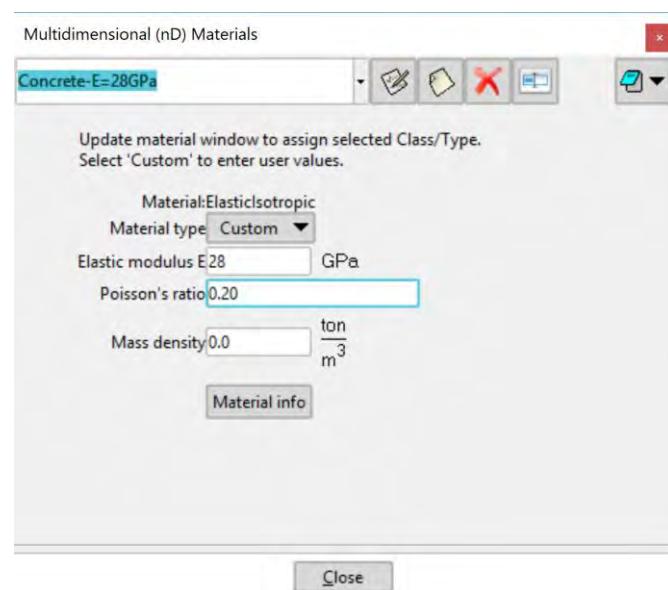


Figure 117: Tutorial 1 – Concrete material options

For Linear Elastic analysis, you may use Timoshenko beam column elements (), which account also shear deformations in comparison with simple Elastic Beam Column elements. The walls can be simulated as equivalent frames with rigid zones at the floor levels.

Note: You could use Shell elements as well for walls, which are supported for elastic analysis.

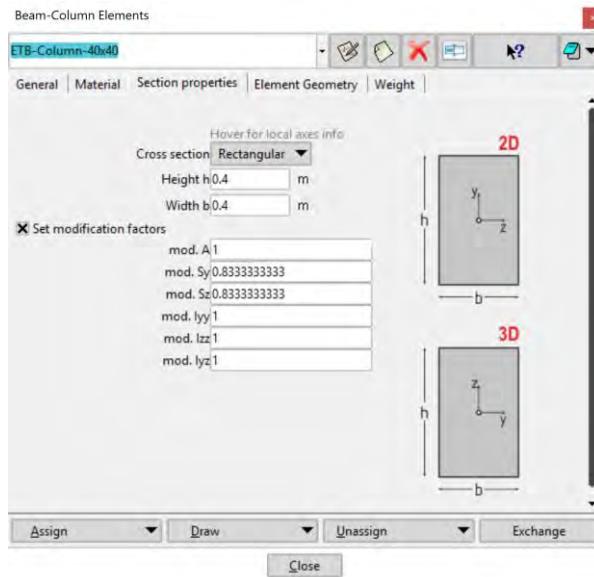


Figure 118: Tutorial 1 – Creating Timoshenko Beam Column element

For rectangular cross sections, the shear area is usually assumed to be equal to the 5/6 of the total area.

Note: Remember that for vertical elements the height fields refer to the global X directions.

When all necessary OpenSees finite elements are defined, you can assign them to the line objects. After that you can see how you assigned the elements on the geometry model ().

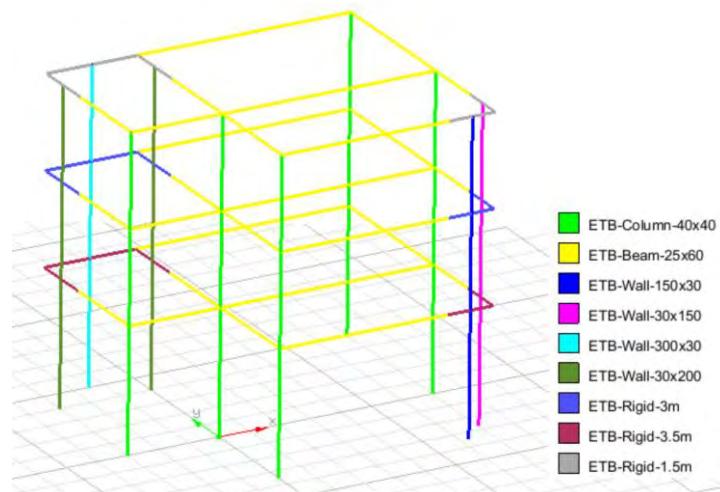


Figure 119: Tutorial 1 – Displaying applied elements

To consider rigid diaphragms at each floor level, you need to specify (⚙) the master node (centre of mass) of each rigid diaphragm with a unique ID number, which is located at each floor level. From the same window, you can then determine the slave nodes of each rigid diaphragm just choosing the slave node condition from the conditions combo box. Master node and slave nodes for each rigid diaphragm are linked together through the ID number. Because the mesh will be finer than 1 element per line object, you may should assign the slave nodes condition to the line objects, so to be transferred to the generated nodes of linear elements. Moreover, be careful to select the right rigid plane.

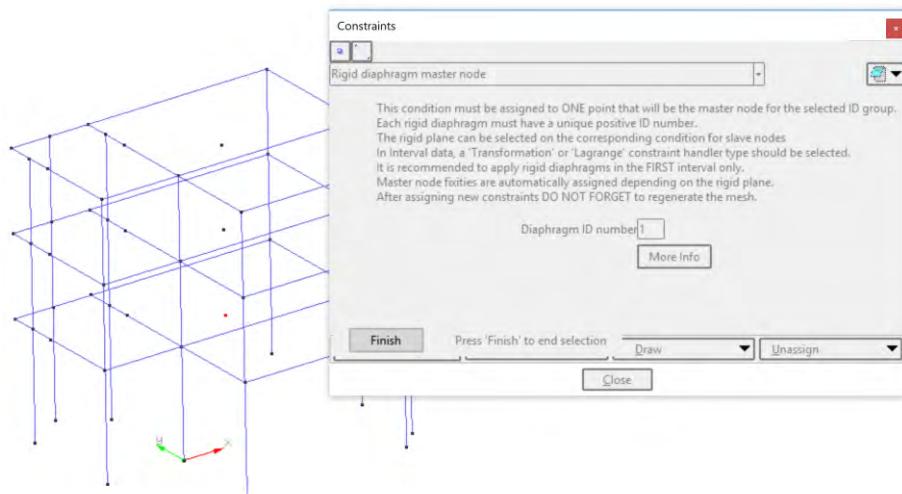


Figure 120: Tutorial 1 – Master node assignment

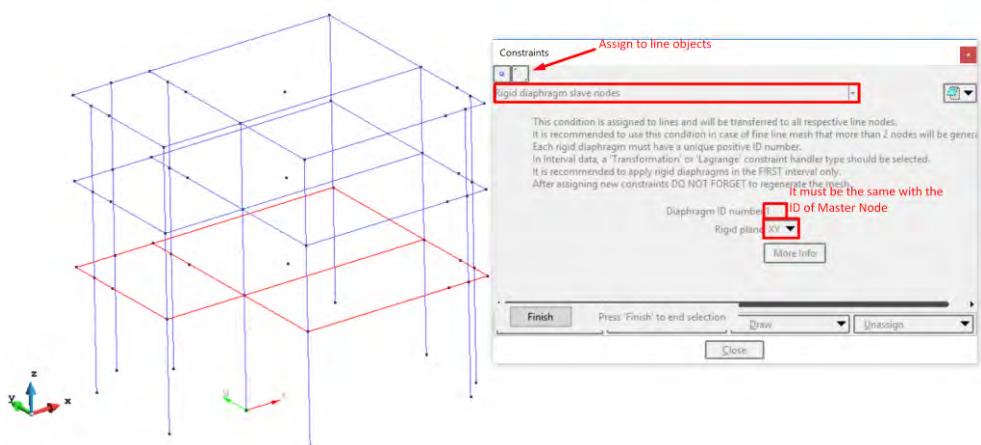


Figure 121: Tutorial 1 – Slave nodes assignment

After calculating the mass and the mass moment of inertia, you can assign these values to the center of mass of each floor level.

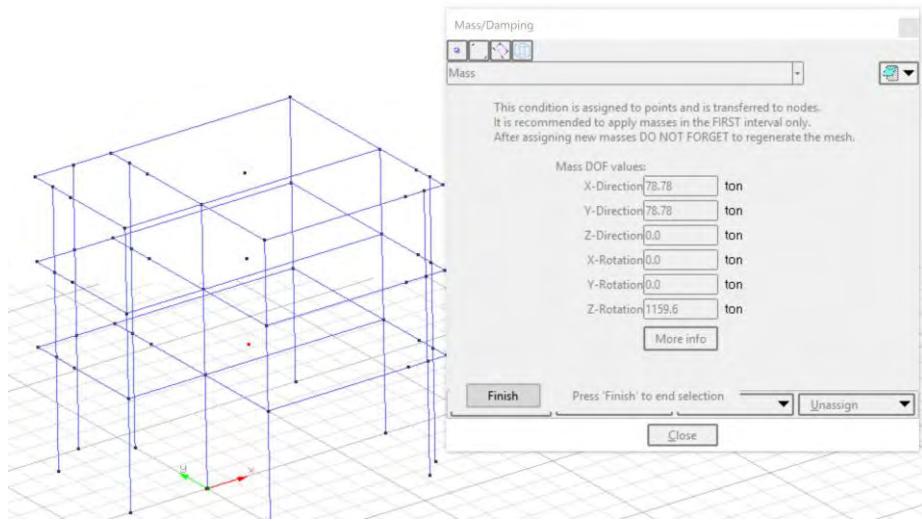


Figure 122: Tutorial 1 – Masses assignment

As you consider for the vertical loads, they consist of the dead load and the loads coming from the slabs. As you do not simulate the slabs with shell elements like in this example, you have to calculate the loads from the slab to the beams manually. After that, you can assign these forces as they are distributed uniformly (枰) along the beam spans. **The loads must be applied on the negative direction of local z axis.**

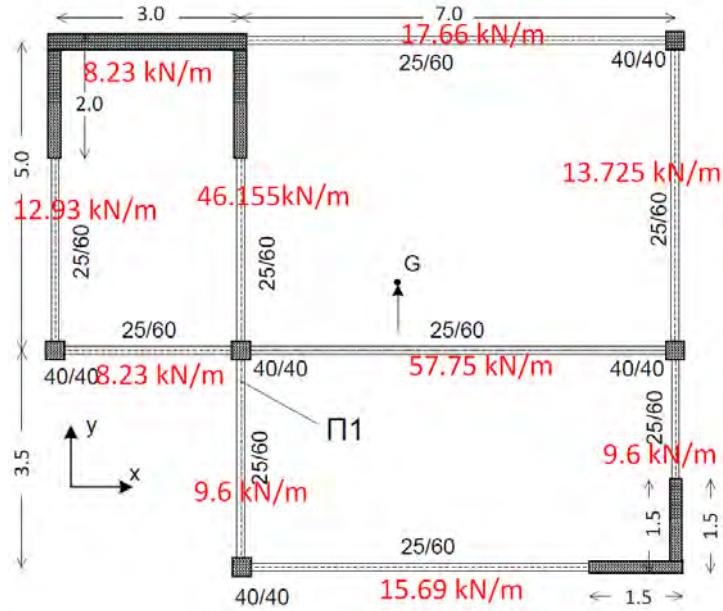


Figure 123: Tutorial 1 – Slab Uniform loads

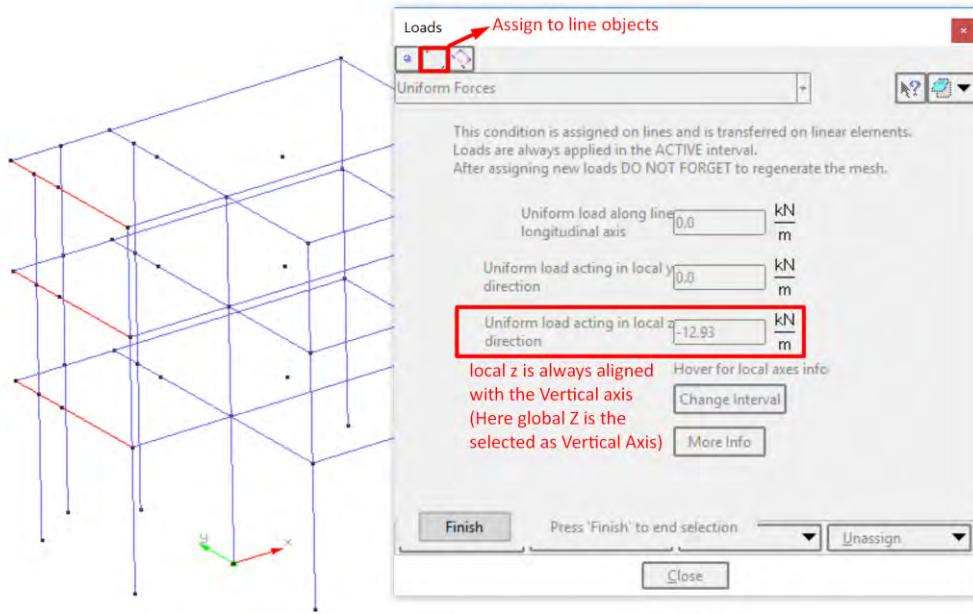


Figure 124: Tutorial 1 – Beam Dead loads assignment

For activating Dead Loads, you should access the *Interval Data* options (⌚) and check the relevant box option. Inside the elements' windows, you can determine the specific weight of the members through the *Weight* tab. In this example, the reinforced concrete specific weight is assumed to be equal to 25 kN/m^3 .

As for the lateral seismic static loads, base Shear is assumed to be equal to the 16% of the total vertical loads and the distribution is specified as triangular along the height.

$$Fb = 0.16 * 5273.98 = 843.83 \text{ kN}$$

$$F1 = \frac{4 * Fb}{3 * 4 + 2 * 3 + 3} = 160.73 \text{ kN}$$

$$F2 = \frac{7 * Fb}{3 * 4 + 2 * 3 + 3} = 281.27 \text{ kN}$$

$$F3 = \frac{10 * Fb}{3 * 4 + 2 * 3 + 3} = 401.827 \text{ kN}$$

Finally, in order to execute modal analysis, we must check the field *Activate eigenvalue analysis* for 9 eigenvalues through the *General Data* option (💾).

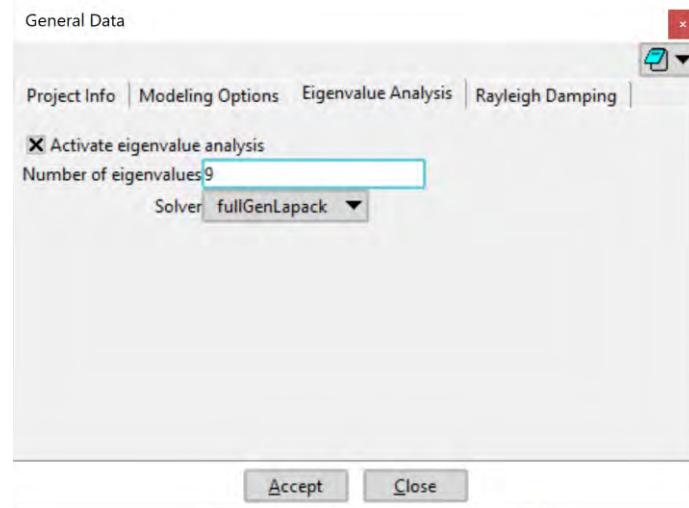


Figure 125: Modal analysis options

The last necessary action is to determine the mesh of elements. The finer the discretization you choose, the more accurate and multiple the results. At this example, line objects are transformed into 5 linear elements (*Mesh > Structured > Lines > Assign number of cells*).

After all that, the analysis is ready to run ( or *GiD+OpenSees > Create .tcl, run analysis and postprocess*).

Results

You can display the desired results as well as the deformed shapes using the *View Results & Deformation* option (). The evaluated modes are shown in the following Figure. The fundamental modal period of the structure is only 0.165 seconds, obviously due to the existence of the walls.

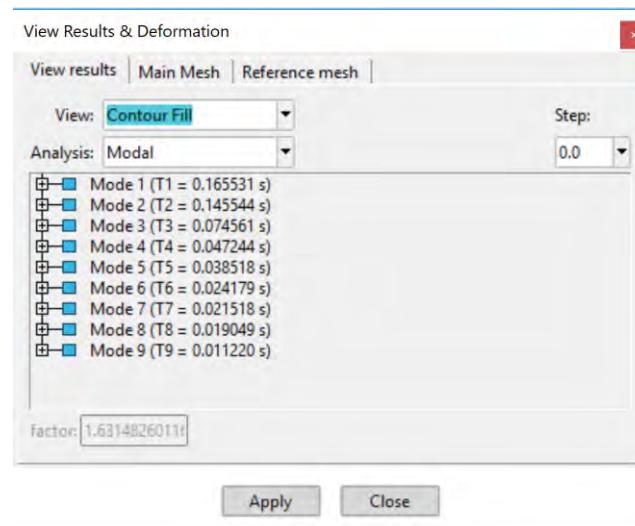


Figure 126: Tutorial 1: Modal periods

The first mode (deformed shape) () is shown in Fig. 127. You can observe that it is mostly rotational about global z axis. The initial shape can be displayed at the same time through the *view style* options ().

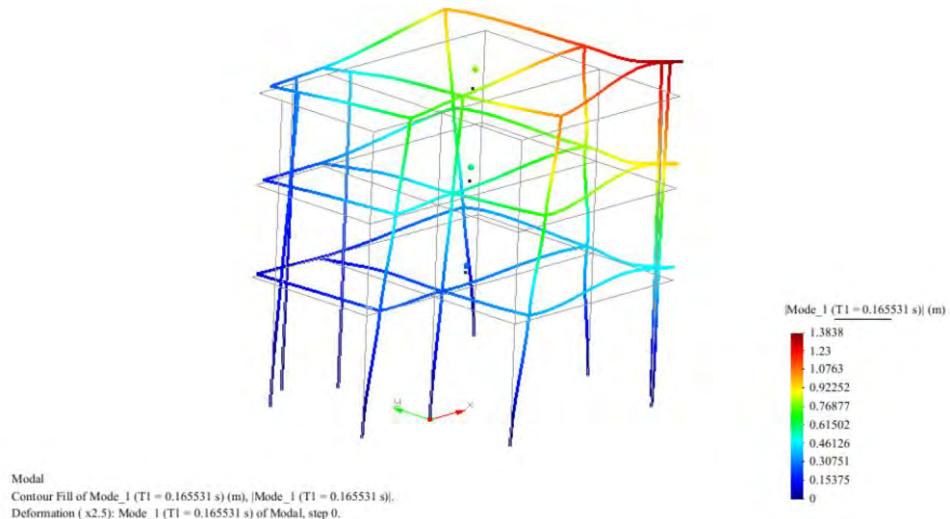


Figure 127: Fundamental mode

Similarly, line diagrams can also be displayed, unfortunately without value labels. Values can be shown using contour filling. For displaying the result of a specific node, you can select *Right click > label > select > Result*.

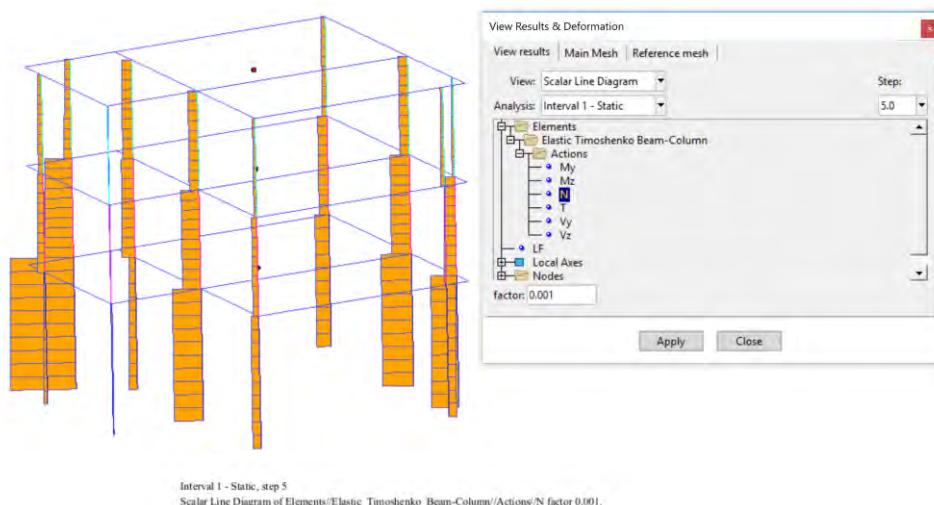


Figure 128: Line diagram; Axial forces

You can see the way that base shear force is distributed among the vertical elements at the base level. Obviously, the walls of which the long dimension is along the force direction (in-plane behaviour) receive the largest part of the total base shear force.

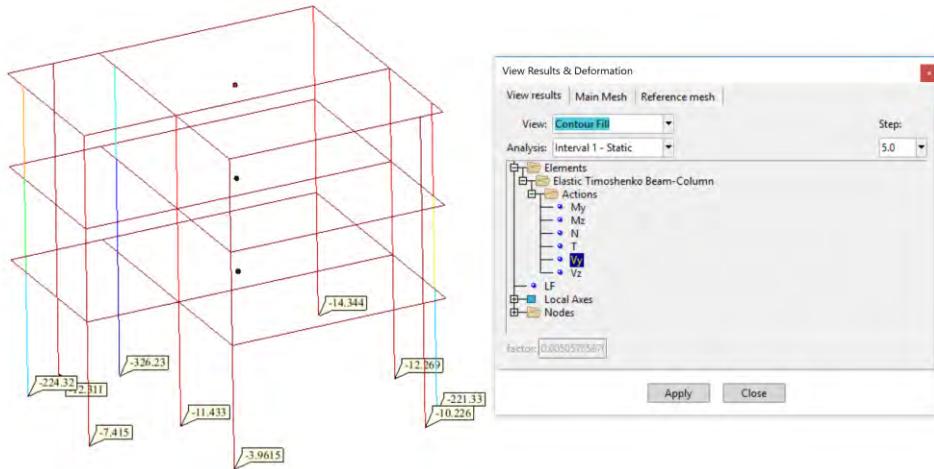


Figure 129: Shear forces along local y axis

Finally, you can display the deformed shape of the elastic static analysis at the final state (5^{th} step). As you can see (Fig. 130) the greatest y-displacement is equal to 0.411 cm, and the floor plan is rotated about the region of the big reinforced concrete core. This somehow validates the results since we can evaluate and assume that the elastic stiffness center is concentrated near the core.

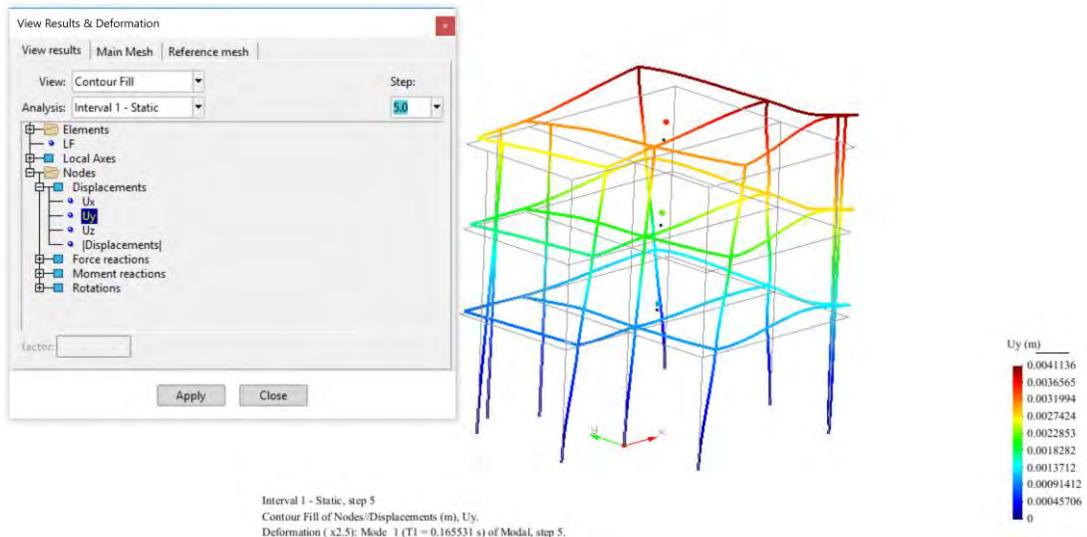


Figure 130: Elastic static analysis: Deformed shape

Alternative Simulation

You could simulate shear walls with Shells elements as an alternative solution to the Beam-Column Elements, with the knowledge the produced results will not be in M-N-Q terms but in stress terms, and forces must be found by the integration of the stresses along the cross-section.

For this example, ShellDKGQ elements were chosen for simulating shear walls, using a Plate Fiber model with 30 cm thickness and the same elastic concrete material as before (Elastic isotropic with $E=28$ GPa). The mesh was determined as structural with a fixed 0.5 m size of the surface boundaries.

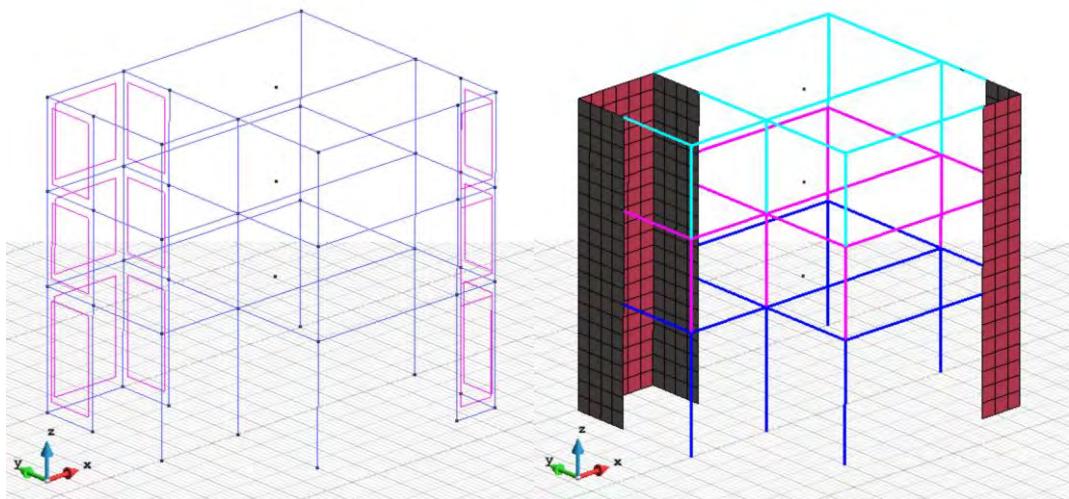


Figure 131:Tutorial 1 with shell elements - Geometry model (left) and Meshed model with layer colours (right)

Note: Remember that a line that belongs to a higher entity (surface or volume) is not transformed to a linear element after discretization.

Having the above note on mind, the uniform forces conditions that come from the slab weight cannot be applied on the lines that belong wall surfaces. Remember that uniform forces condition should be applied only on lines that simulate a frame element (Beam-Column). Given this limitation, you could apply the uniform forces using the Forces condition (apply to lines / transfer to line nodes after discretization) by doing the maths (uniform load times the length divided by the number of nodes).

In the following Figure (Fig. 132), the modal periods are depicted. The first 3 modal periods are close to the ones produced using exclusively beam-column elements, but for the following modes they are getting larger.

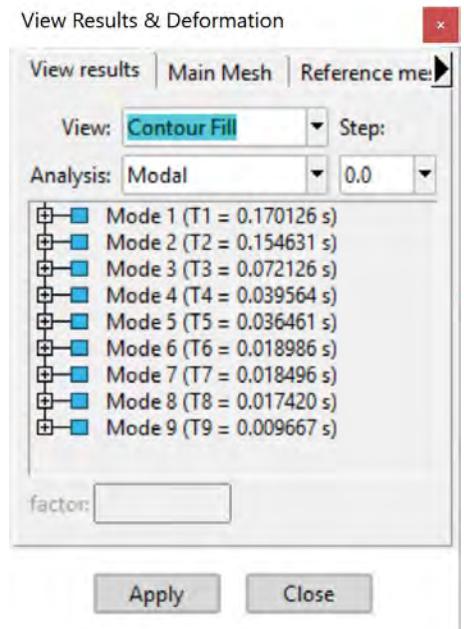


Figure 132: Tutorial 1 with shell elements – Modal periods

Table 7: Comparison of modal periods simulating walls as Beam-Column or Shell elements

Mode	Period T (sec)		Difference (%)
	Beam-Column only	Beam-Column plus Shell	
1	0.165531	0.170126	2.78
2	0.145544	0.154631	6.24
3	0.074561	0.072126	-3.27
4	0.047244	0.039564	-16.26
5	0.038518	0.036461	-5.34
6	0.024179	0.018986	-21.48
7	0.021518	0.018496	-14.04
8	0.019049	0.01742	-8.55
9	0.01122	0.009667	-13.84

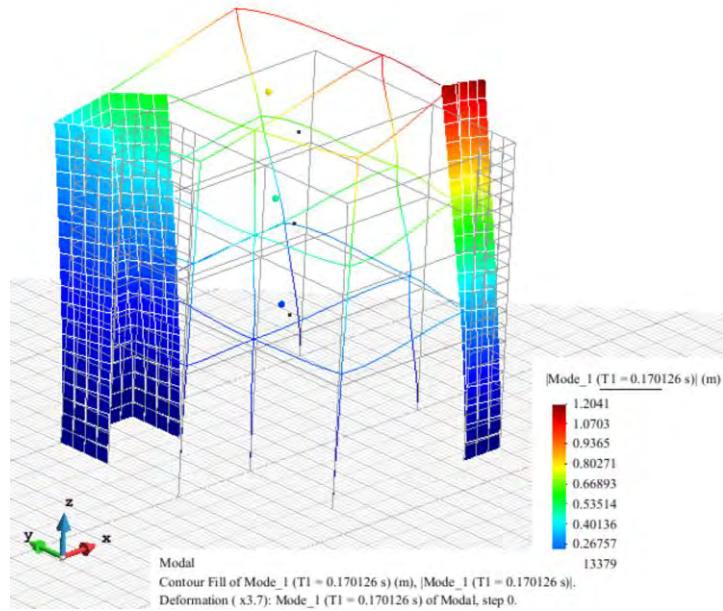


Figure 133: Tutorial 1 with shell elements – Mode 1

The normal stresses along the vertical axis can also be displayed for the static analysis in Fig. 134. For the better understanding of the stress distribution, maximum and/or minimum values can be set (,).

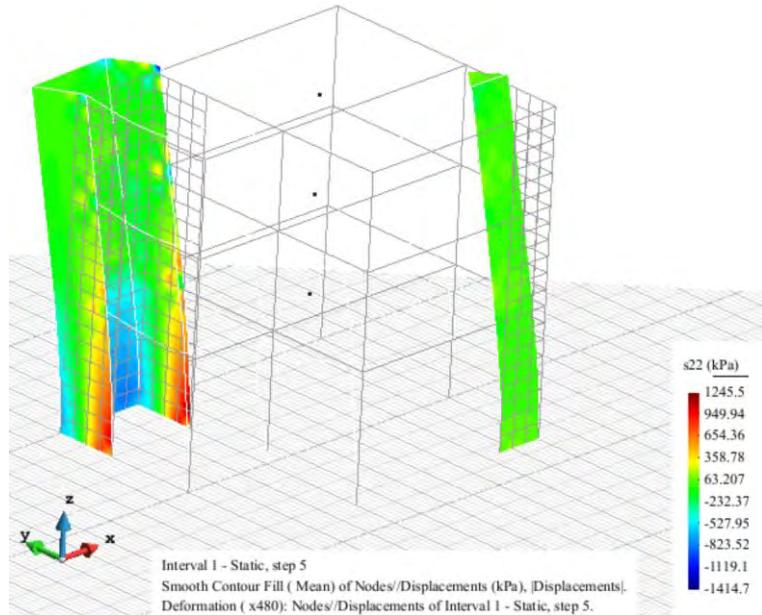


Figure 134: Tutorial 1 with shell elements – Stresses s_{22} of ShellDKGQ elements

To compare with previous simulation, the deformed shape from the static analysis is shown in Fig. 135. As you can see the displacements haven been increased, which was expected considering that this model is more flexible, as a result the fundamental modal period which

was found increased. Generally, it has been observed that equivalent models including Shell elements (for walls) are more flexible than others using only frame elements.

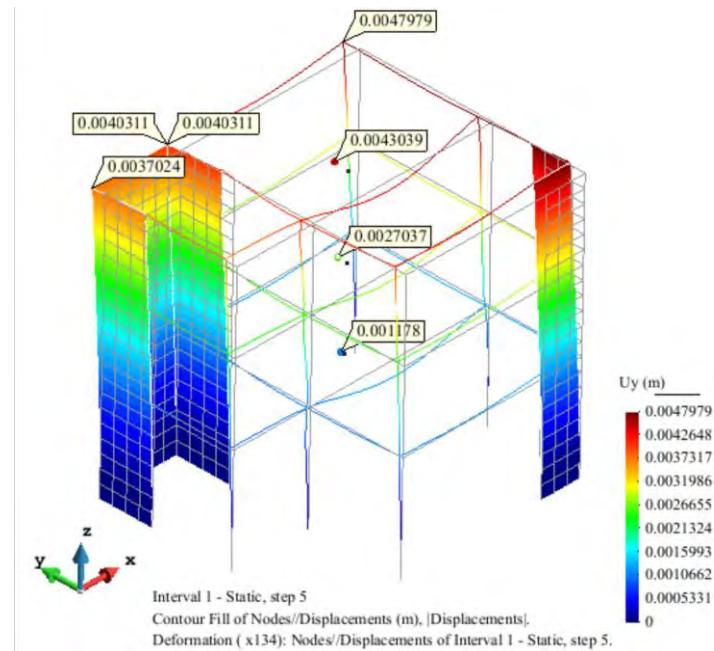


Figure 135: Tutorial 1 with shell elements – Deformed shape with displacements along Y axis

Tutorial 2 – Pushover Analysis of a Three-Story Building

Description

This tutorial example describes the GiD+OpenSees implementation of Static nonlinear analysis (Pushover static analysis) of a three-story frame reinforced concrete building. The floor plan is identical along the height of the building and is depicted in Fig. 136. The ground floor height is 3.5 meters, while the rest of the floors heights are 3 meters. The concrete strength class is considered as C20/25 according to Eurocode 2. It is requested to evaluate the pushover curve of the building along the x-x axis.

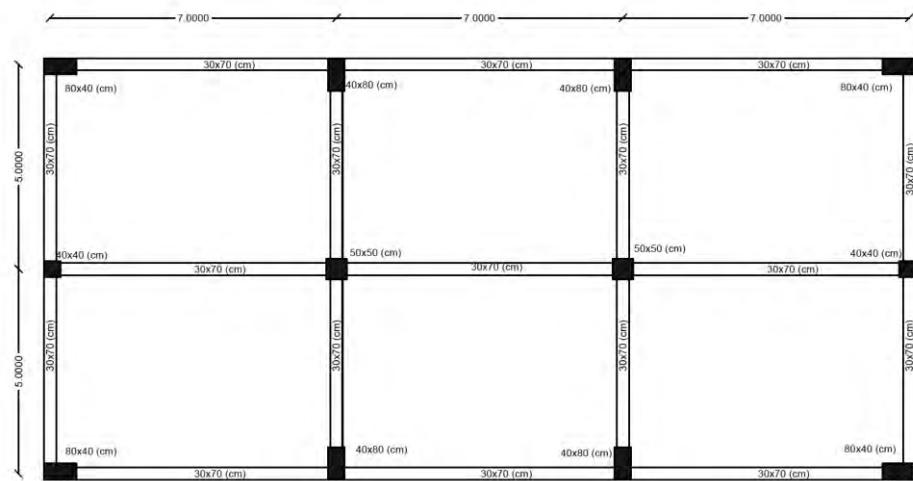


Figure 136:Tutorial 2 - Floor plan view

There are five different sections created as shown in table 8.

Table 8: Longitudinal reinforcement of structural members

Member	Height (m)	Width (m)	Longitudinal reinforcement
Column	0.5	0.5	4Ø20
50x50			+ 12Ø16
Column	0.4	0.4	16Ø16
40x40			
Column	0.8	0.4	16Ø20
80x40			

Column	0.4	0.8	16Ø20
40x80			
Beam 70x30			Top: 4Ø16
(1st floor)	0.7	0.3	Bottom: 4Ø16
Beam 70x30			Top: 4Ø14
(2nd -3rd floor)	0.7	0.3	Bottom: 4Ø14

Problem solution

In this example, you will need to use two intervals, in contrast with the previous tutorial paradigm. The first one refers to the gravity analysis and the second one to the lateral seismic forces using displacement control.

Similarly, you should first create the plan view in the XY plane using the *grid* function. After that, we copy the floors using the *Copy* function (*Utilities > Copy*) along the Z axis using 3 meters distance. Finally, the columns are created using again the *Copy* command, and the *Extrude Lines* options, as well as the *Collapse* option checked.

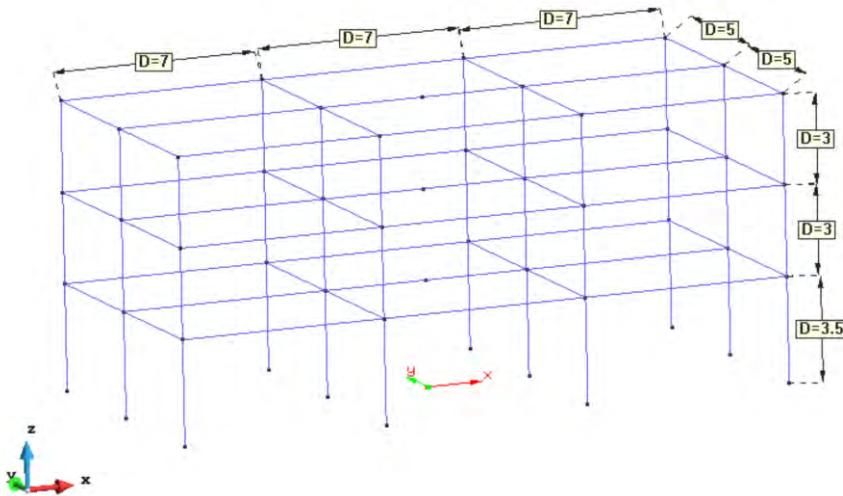


Figure 137: Tutorial 2 – Geometry model

Remember to determine the Vertical axis as global Z ().

The concrete strength class is C20/25. You should define two different concrete uniaxial materials () that will be used by the fiber elements, one for the confined concrete (core)

and one for the unconfined material (cover). Specifically, the Popovics Concrete type (Concrete04) is used. The difference between the aforementioned materials is only in the strain at crushing strength. After updating the window for generating the values of C20/25 strength class, you should select *Custom* class, to change the field properties.

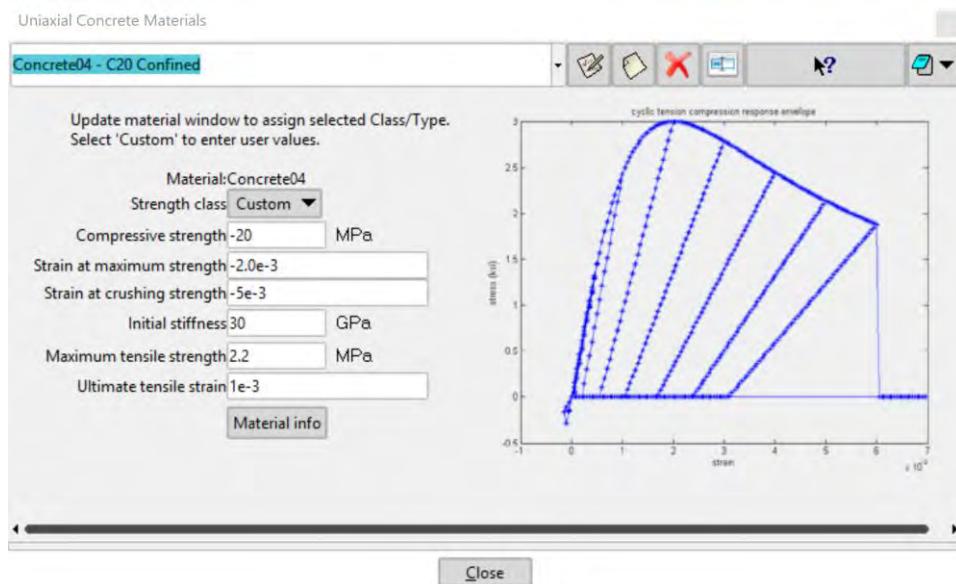


Figure 138: Tutorial 2 – Confined concrete material

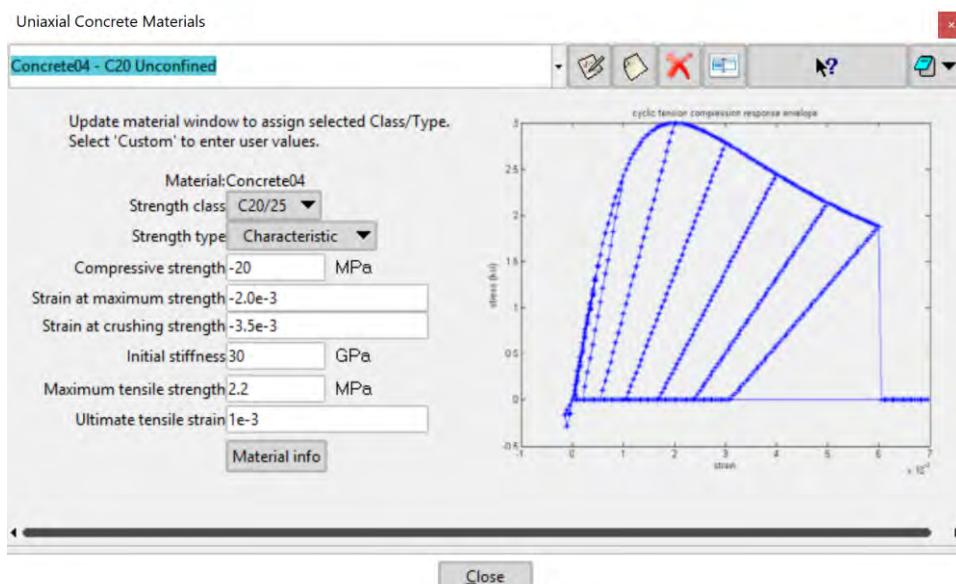


Figure 139: Tutorial 2 – Unconfined concrete material

Likewise, you should create the uniaxial steel material (), which will be used by the steel fibers. We choose the *Steel02* material for B500C steel grade with a strain hardening rate 15%. Moreover, the parameters for the transition from elastic to plastic branches are defined as proposed by the OpenSees developers.

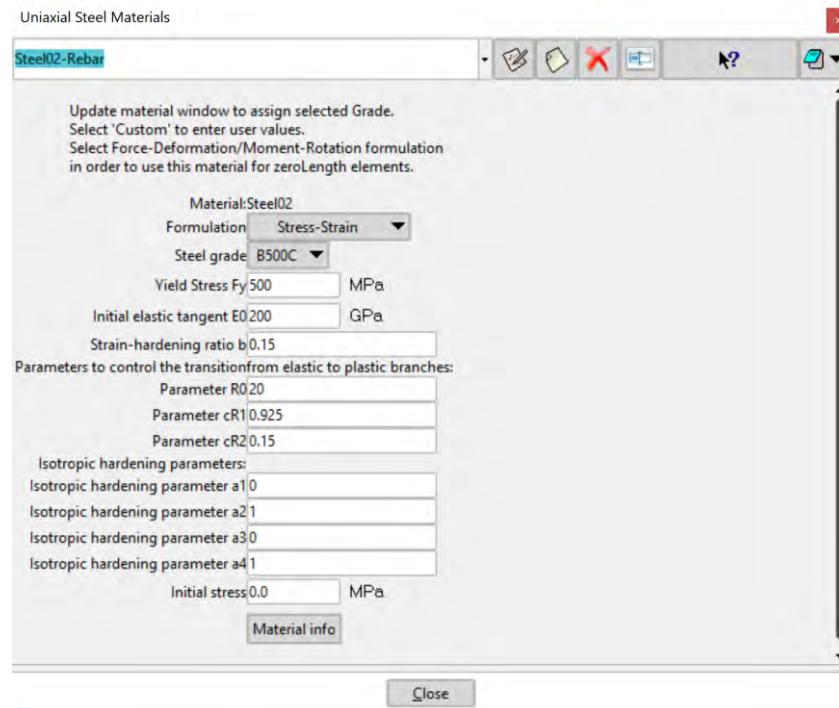


Figure 140: Tutorial 2 – Reinforcing bar material options

Since the uniaxial materials are defined, you can continue creating the Fiber Section Force-

Deformation models (), which will be used by the Force-Based Beam Column elements.

Note: Obviously, in case that reinforcement is differentiated along the member length, more fiber elements as well as more fiber section models are necessary to be created.

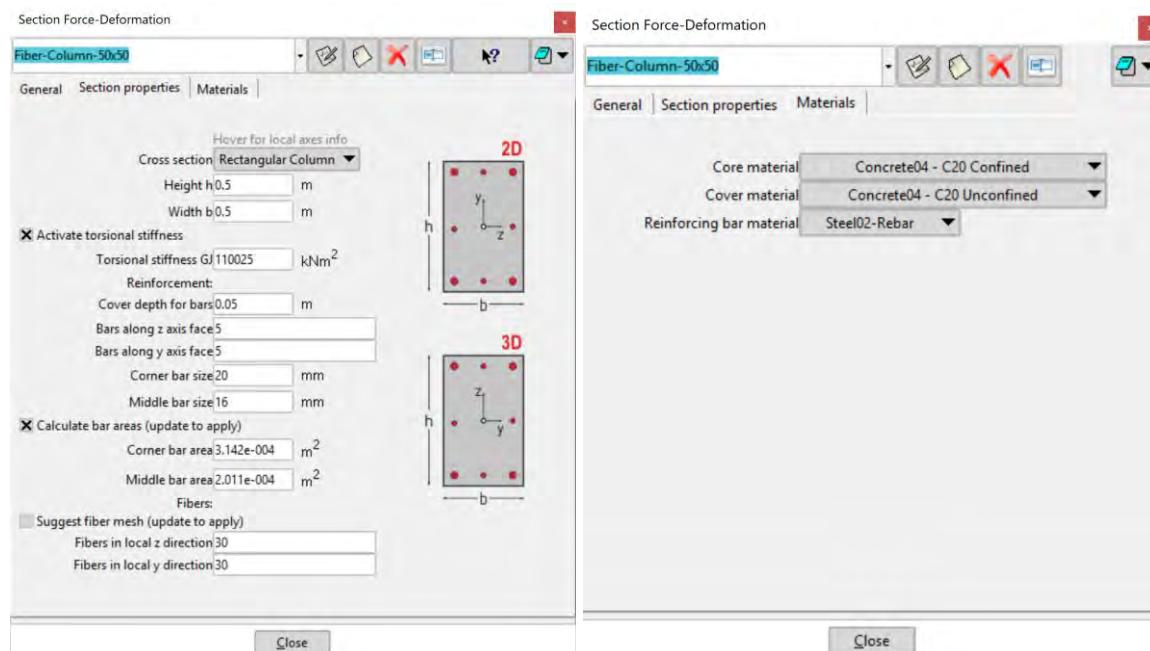


Figure 141: Tutorial 2 – Fiber Section options

After that, you need to define the Force-based beam column elements (), which consist of the above fiber modes. Each fiber section model corresponds to a Force-based beam column through the *Section* property at each element window dialog. Finally, you can assign them to the geometrical objects as shown below ().

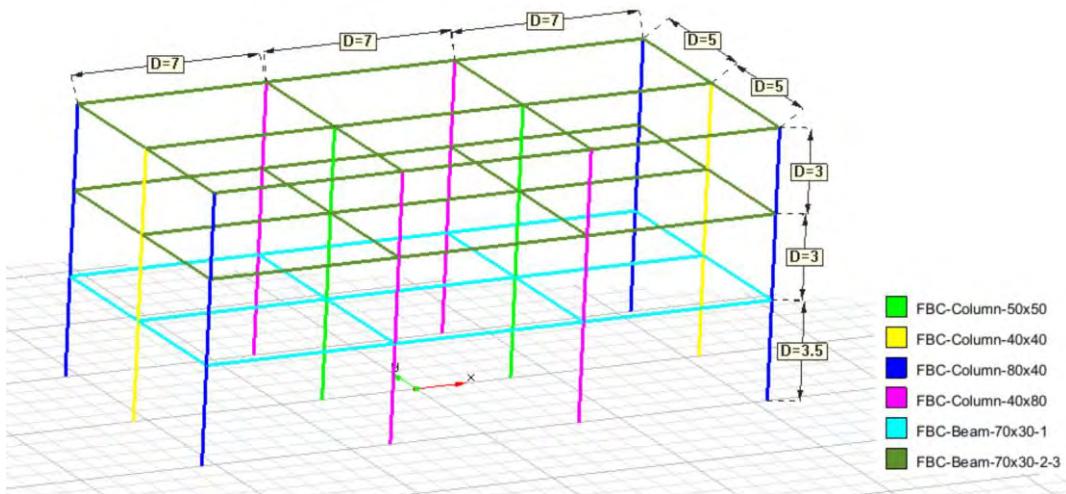


Figure 142: Tutorial 2 – Elements assigned to the geometry model

Note: Restraints, rigid diaphragms and gravity forces can be implemented in analogous way, such as in the previous example. These conditions must be applied in the first interval only.

For simplicity, interior and exterior beams were loaded with uniform forces $\frac{20 \text{ kN}}{\text{m}}$ and $\frac{15 \text{ kN}}{\text{m}}$ () respectively, simulating the weight of the slabs which are not designed. In addition to these loads, the dead loads of beams and columns will be included through Interval Data options.

Before we continue with the pushover analysis, you should determine the gravity analysis parameters (). First, Gravity analysis is defined as *Static* with *Load Control integrator*, as the total forces of gravity are estimated and the 100% of them must be applied on the structure. At this example, gravity loads are applied in 5 steps. *Loading type* must be *Linear*, so that an ascending presence of the load pattern (gravity) can be used in each analysis step. For 5 steps, solver will try to apply 20%, 40%, 60%, 80% and 100% of the load pattern at each step, respectively. **It is worth noting that in case of analysis failure, different algorithms are tried, as well as more steps, if necessary.**

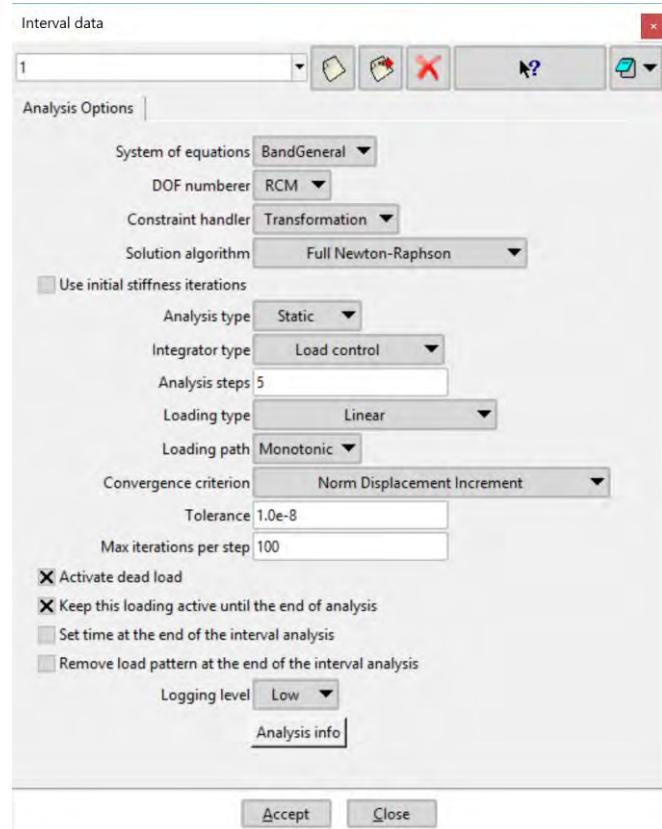


Figure 143: Tutorial 2 – Gravity analysis options

Do not forget checking the *Keep this loading active until the end of analysis* option. In this way, nonlinear lateral static analysis (Interval 2) will continue from the end of the gravity analysis and the gravity loads will be remaining constant (Transformation of Linear Timeseries to Constant one) for the rest of the analysis.

Logging level is recommended to be low, so that analysis process becomes greatly faster.

The next step is to create a new interval () through the *Interval Data* window, without copying the conditions (you will be asked when creating a new interval) from interval 1 to interval 2.

Using this interval, you should construct the load pattern, which will be used by the OpenSees solver, in order to push the structure over, and specifically the selected control node, to the desired lateral displacement at each step. Thus, you must assign forces to each floor mass center, that will constitute the desired, force distribution along the height, regardless the force value. It is up to the user to decide which distribution is suitable, regarding the structure type. It may be used triangular, uniform, based on the first mode, distribution, or even more complicated distributed based on the floor lateral displacements considering higher modes.

For simplicity, the applied force distribution is expressed by the following equation (triangular distribution):

$$F_i = F_b * \frac{zi}{\sum zi}$$

Where:

F_b : The base shear force.

zi : The height of the story

Assuming that $Fb = 1 kN$, we have:

$$F1 = 0.179 kN$$

$$F2 = 0.3333 kN$$

$$F3 = 0.487 kN$$

Note: The above assumption is convenient because OpenSees prints the load factor for the load pattern at each step, so in this way the load factor is equal to the base shear.

Therefore, you should apply the above forces () to each floor master points at the direction which is under consideration.

The last step, before determining the mesh options, is to define the analysis options for the pushover analysis. *Integrator* must be *Displacement Control*, as you input the desired total displacement of the top floor. Depending on the selected number of *steps*, OpenSees solver tries to attain part of the total displacement multiplying the load pattern with a load scale factor. Thus, the *loading type* must be selected as *Linear*, so that the load pattern can be multiplied properly.

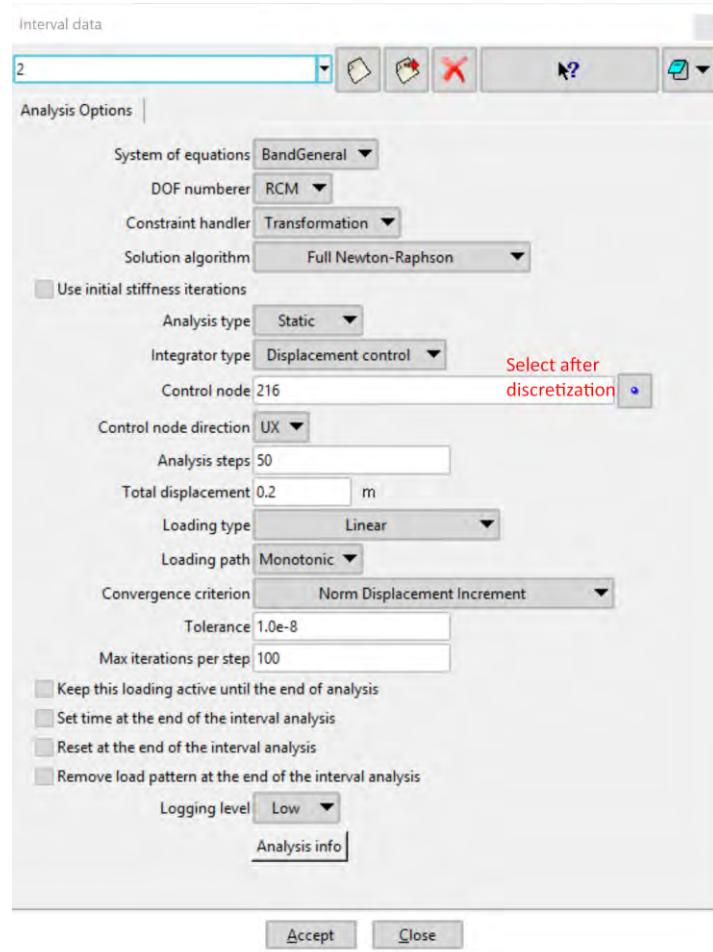


Figure 144: Tutorial 2 – Pushover analysis options

It is worth of noting that **Control node** can and must be selected after discretization. **OpenSees solver receives as input nodes and elements, not points and geometrical objects (lines, surfaces, volumes), consequently it is not possible to know the correct node label in advance.** Moreover, regenerating the mesh with different mesh options means that you must reselect the Control node, whose label is very likely to has been differentiated.

Before running the analysis, you must generate the mesh. From the menu *Mesh > Structured > Lines > Assign number of cells/Assign size*, you can determine the level of refinement of the mesh. In this example, 5 elements per line and 1.5 weight at the line ends (*Mesh > Structured > Lines > Concentrate elements*) is chosen. In this way, you have better results near to the member edges, where plasticity is firstly and mainly concentrated, as well as we keep the number of elements low for reducing the computation time.

Results

The pushover curve can be displayed () by choosing point graph (*Create > Point graph*) including all steps (checkbox on the upper right corner; if not displayed select point graph again). Y component must be Load factor and the X component the displacement of the control node at the direction we pushed the structure.

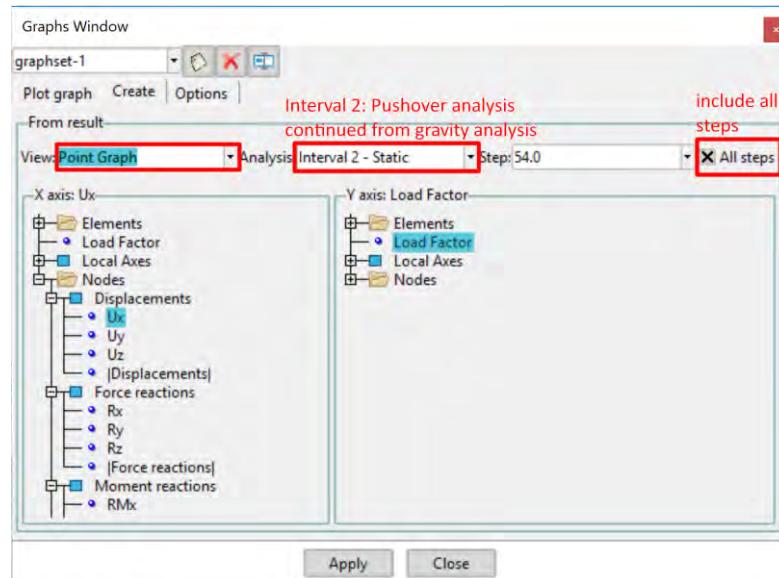


Figure 145: Tutorial 2 – Point graph options for pushover curve

Note: When applying to the Control node (usually the top of the structure) DO NOT forget to activate the *Join* option, so that existing node can be selected (*Right click > contextual > Join ctrl-a* or just press *ctrl+a*).



Figure 146: Tutorial 2 – Pushover curve

Obviously, degradation of the strength is not performed because of the constitutive law of the reinforcing bars, where strength is always increasing and hence the stiffness reduction is mostly resulted by the concrete damage.

Similarly, any result vs another can be displayed in a plot graph such as Moment vs Rotation.

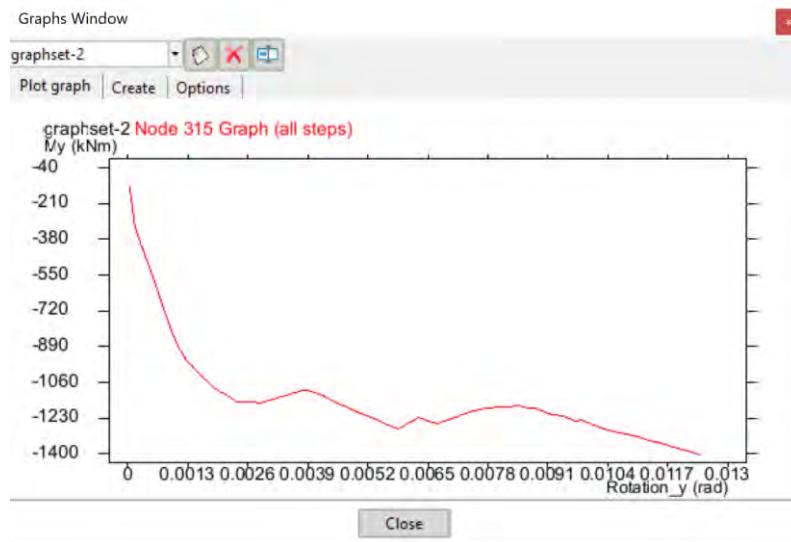


Figure 147: Tutorial 2 – Moment-rotation about local y diagram

When creating a plot graph, the values can be observed in a table format through the Options tab as shown in the following figure. These values can be easily then copied to another external program.

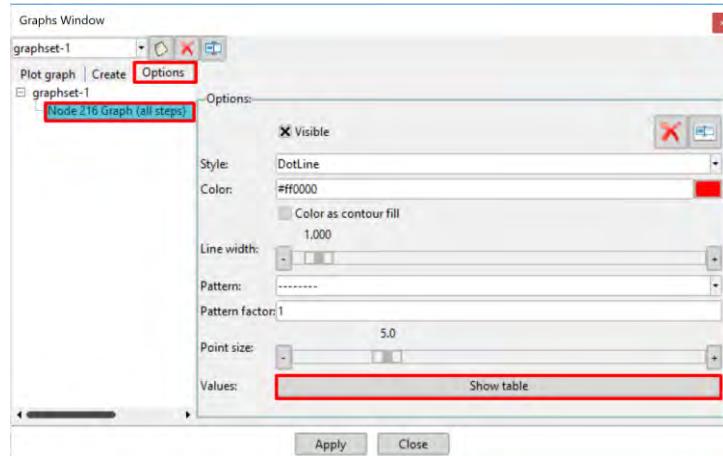


Figure 148: Tutorial 2 – Show table option

Results can also be displayed with Contour filling, displaying vectors, Line diagrams as well as the deformed shape () such as in the previous example.

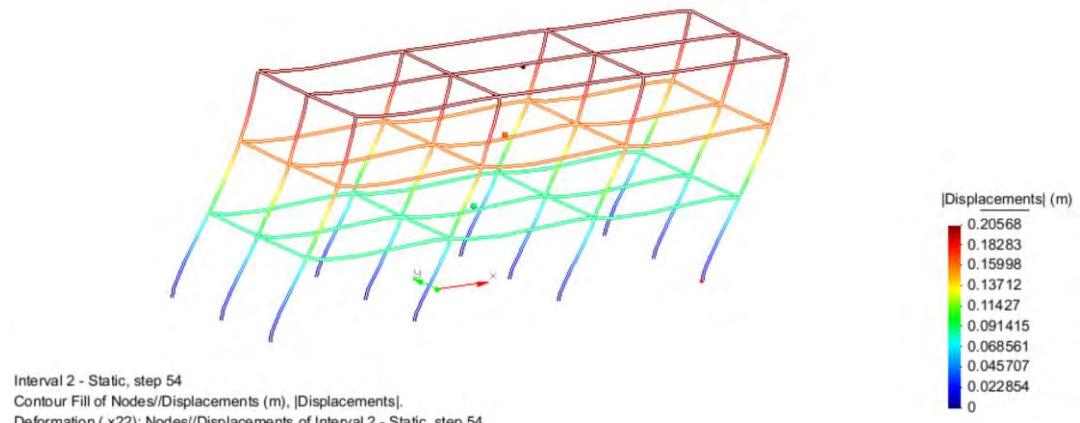


Figure 149: Tutorial 2 – Deformation at final state

Tutorial 3 – Dynamic Analysis of a Three-Story Building

Description

This example illustrates the dynamic analysis of the same building as the previous tutorial example and will mostly focus on the differences between them. The gravity analysis is executed first, and then uniform excitation in the 2 normal horizontal directions (X and Y) using ground motion records. The same ground motion record will be used for both horizontal directions with the suitable scale factors, so that the earthquake direction is 45 degrees rotated to both horizontal axes. It is requested to evaluate the response of each floor during the ground motion in displacement, acceleration and velocity terms.

Problem solution

In this example, two intervals are required; The first one corresponds to the gravity analysis and the second one to the Uniform excitation using ground motion records. Restraints (🔗), Rigid diaphragms (⚙️) and gravity loads (⬇️) are applied like in the previous examples **in the first interval only**. In addition to them, you need to apply masses (🏋️) to the center of mass of each floor. It is important to apply them in the first interval, otherwise the Interface will omit them.

For the dynamic analysis purposes, the Rayleigh damping parameters should be evaluated, and as a result a modal analysis is first required, unless you do not want to include material damping contribution. The modal periods are shown in Fig. 150, with which the Rayleigh damping coefficients can be calculated.

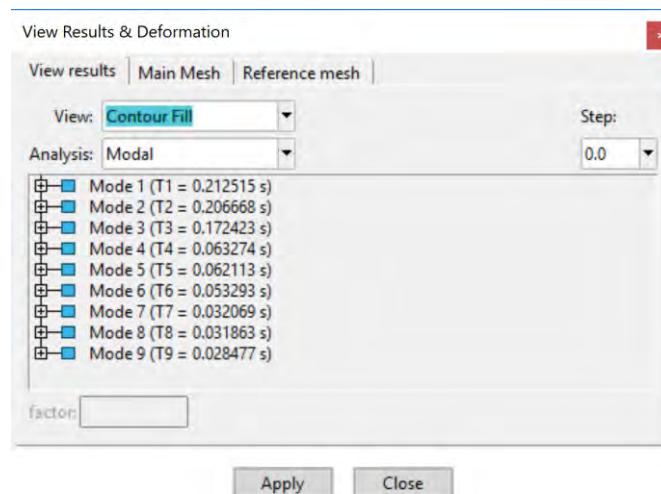


Figure 150: Tutorial 3 – Modal periods

Through the *General Data* options, you can determine the Rayleigh damping parameters. In this example they are determined based on the first and the third mode and assuming that the material damping ratio is equal to 5%, which is widely used for reinforced concrete structures.

The first modal period is $T_1 = 0.215 \text{ sec} \rightarrow \omega_1 = 29.20 \frac{\text{rad}}{\text{s}}$.

The third modal period is $T_3 = 0.172 \text{ sec} \rightarrow \omega_3 = 36.44 \frac{\text{rad}}{\text{s}}$.

Rayleigh damping is given by:

$$\mathbf{c} = a_0 \cdot \mathbf{m} + a_1 \cdot \mathbf{k}$$

Where \mathbf{c} is the Rayleigh damping matrix, \mathbf{m} is the mass matrix and \mathbf{k} is the stiffness matrix. Mass proportional and stiffness proportional coefficients are calculated as:

$$a_0 = \zeta \cdot \frac{2 \cdot \omega_i \cdot \omega_j}{\omega_i + \omega_j} \text{ and } a_1 = \zeta \cdot \frac{2}{\omega_i + \omega_j}$$

Consequently,

$$a_0 = 1.6211 \text{ and } a_1 = 0.001523$$

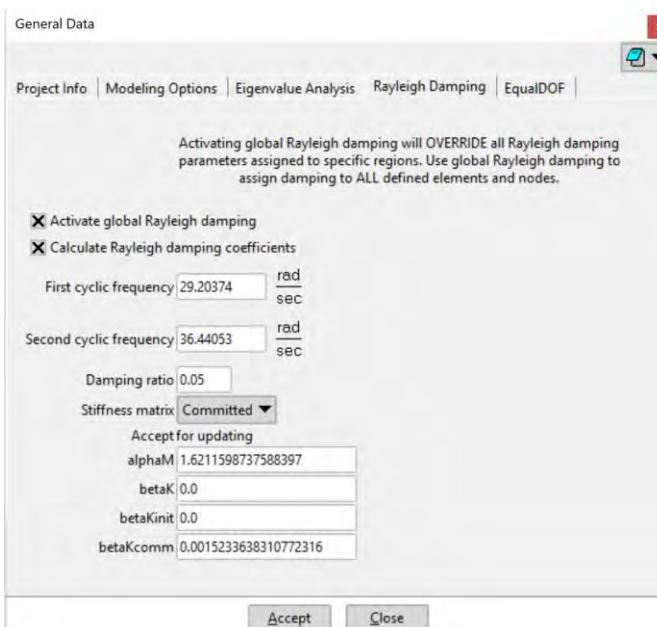


Figure 151: Tutorial 3 – Rayleigh damping parameters

Before continuing with Interval 2 options, you should define the ground motion records (), which will be used for the dynamic analysis. After selecting a record file through the file  button (), it is automatically copied to the project directory (*projectname.gid*) inside the

Records folder and OpenSees solver sources it from there. Be careful with the Record parameters (Type, format etc.) in order to avoid logical errors during the analysis process.

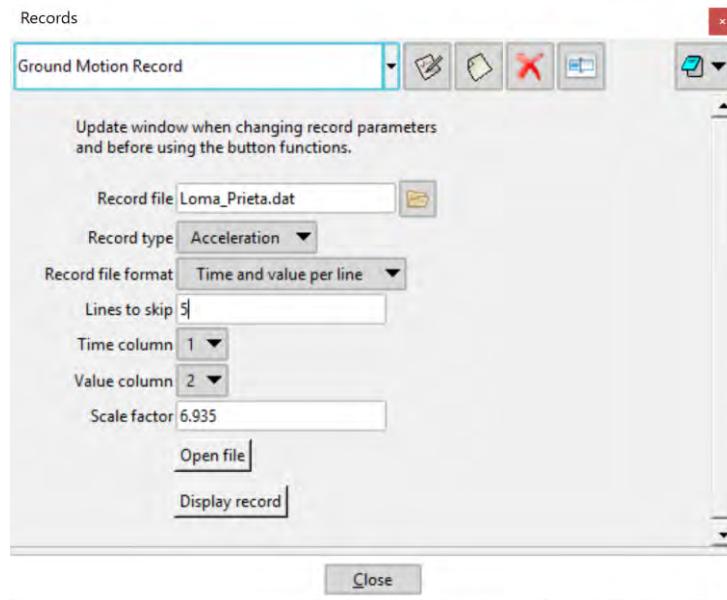


Figure 152: Tutorial 3 – Record file options

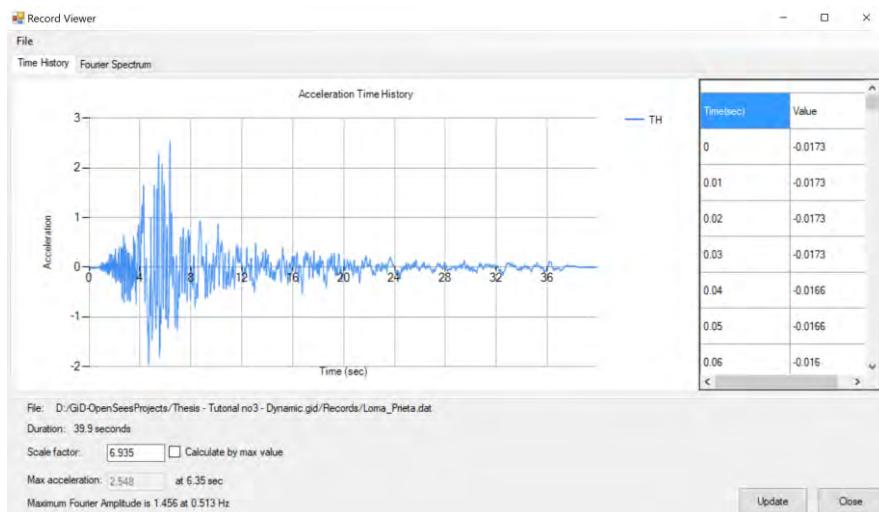


Figure 153: Tutorial 3 – Acceleration Time history used

The mentioned record file in this example uses g units, so we define a scale factor $9.81 \cdot \frac{\sqrt{2}}{2} \approx 6.935$ to transform it to m/s^2 and to be consistent with the desired ground motion direction, as we will apply it to both horizontal directions.

After that, you can proceed to the interval 2 options. You should obviously select transient (dynamic) analysis with Newmark or Hilber-Hughes-Taylor integrator and set the Loading type as *Uniform excitation*. After selecting the number of directions (2), you can define which

records correspond to each direction. It is recommended to use a small analysis time step, such as 0.01 seconds which is the same as the time step of the recorded motion.

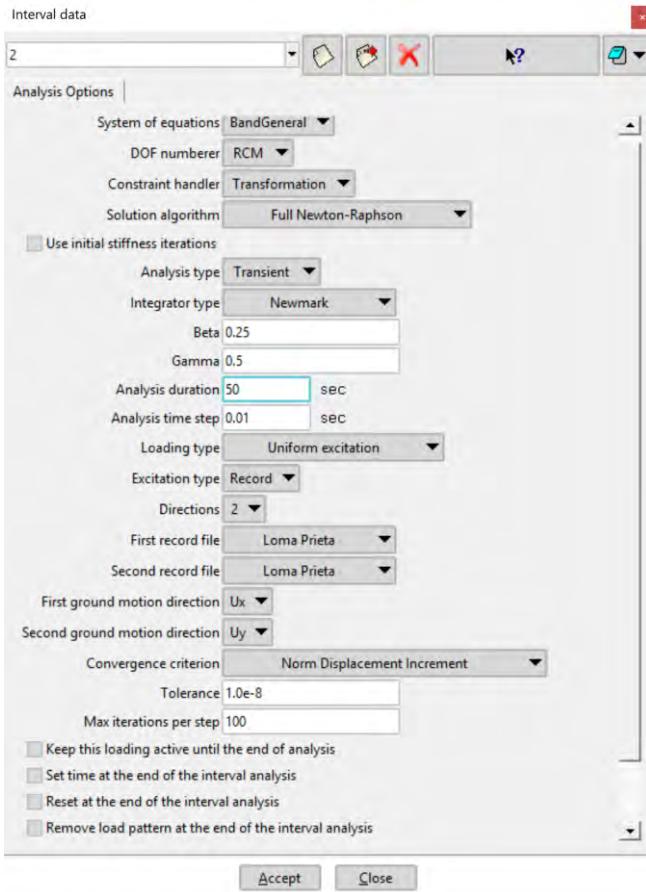


Figure 154: Tutorial 3 – Dynamic analysis with uniform excitation options

Considering dynamic analysis, the bulk of results is extremely huge and consequently GiD needs a lot of time to import and visualize them. In *Output Options* menu (≡), you can determine the output step frequency with which, the numerical results are going to be transferred to GiD Post-Process environment as well as the desired results to be printed. Besides the above options, you can select for binary format of the results, so that GiD can read and import the bulk of results to the Post-Processor a lot of faster.

Results

In the following figures you can see the response of each floor, which is relative to the ground motion. The higher the floor, the greatest the response, as expected. It is also worth noting the permanent (plastic) deformations that are developed resulting permanent displacement.

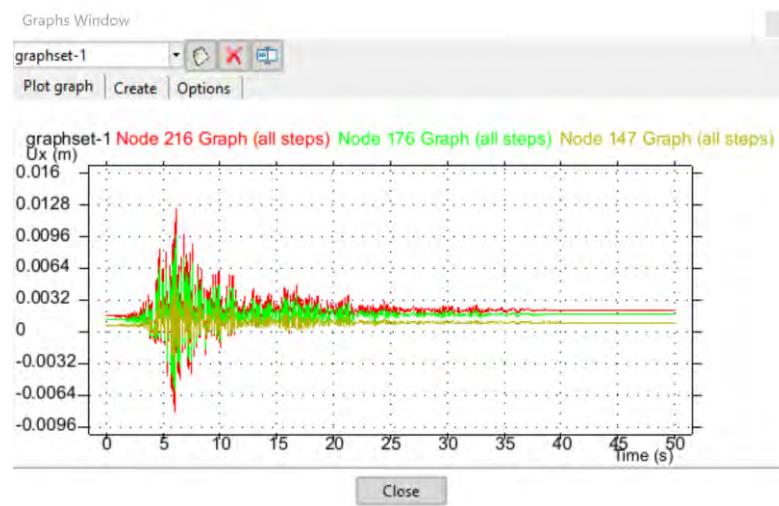


Figure 155: Tutorial 3 – Relative X-Displacement response time histories for each floor

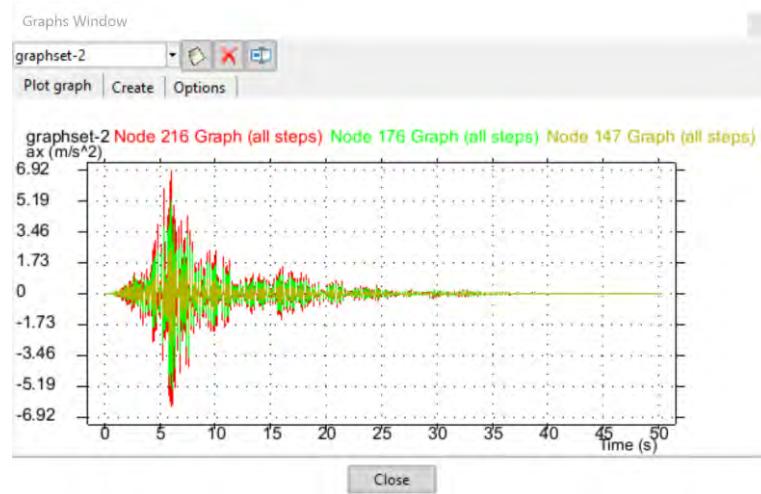


Figure 156: Tutorial 3 – Relative X-Acceleration response time histories for each floor

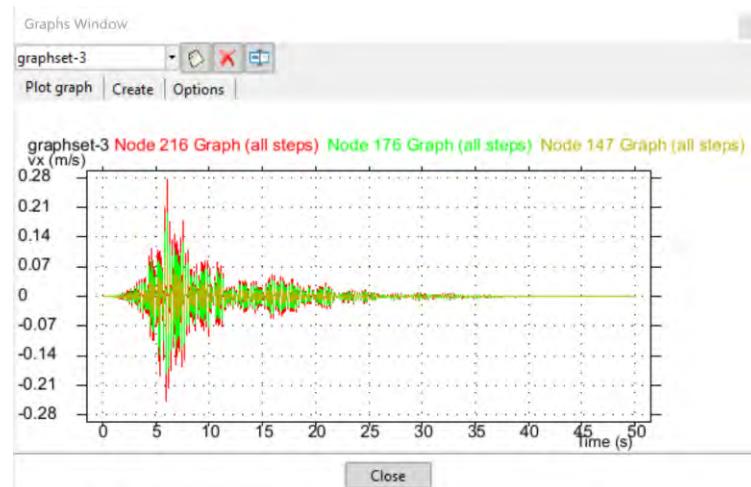


Figure 157: Tutorial 3 – Relative X-Velocity response time histories for each floor

Tutorial 4 – Nonlinear Reversed Cyclic Static Analysis on Beam-Column Joint

Description

This example describes the simulation of an exterior beam-column joint and tests the behavior under reversed cyclic loads. The dimensions of the structural members are shown in Fig. 158. The cross-sectional dimensions of the columns and beam as well as the reinforcement details are shown in Fig. 159. The strength class of concrete is C20/25 according to Eurocode 2 and the mean strength is used (28 MPa). Grade 60 (420 MPa yield stress) reinforcing steel is specified for all members. The load-deflection curve of the beam end is requested for this example.

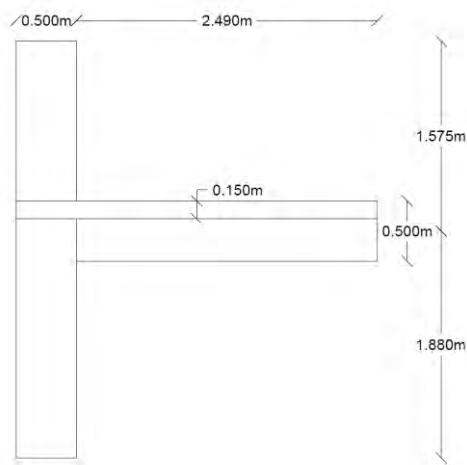


Figure 158: Tutorial 4 - Elevation view of exterior beam-column joint

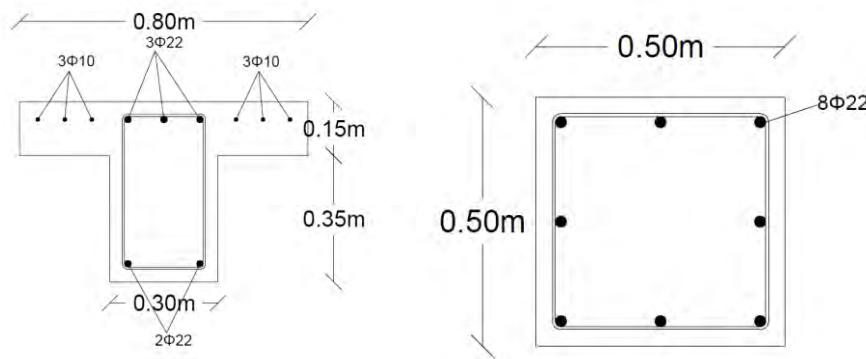


Figure 159: Tutorial 4 - Cross Sectional details: Beam (left) and column (right)

Problem solution

Again, you need to use two analysis intervals: the first one corresponds to the gravity analysis and the second one to the reversed cyclic static analysis with displacement control.

Two Popovics concrete uniaxial materials are defined: Confined and Unconfined. Confined concrete is characterized by a larger strain at crushing strength ($-5.5\text{e-}3$ m/m instead of $-3.5\text{e-}3$ m/m). They are going to be used for the concrete fibers, for core and cover material, respectively.

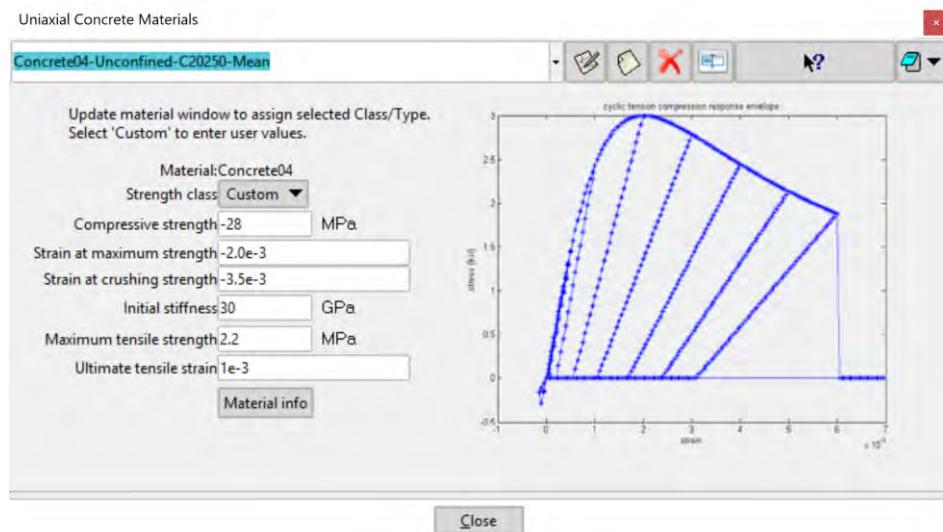


Figure 160: Tutorial 4 - Unconfined concrete material options

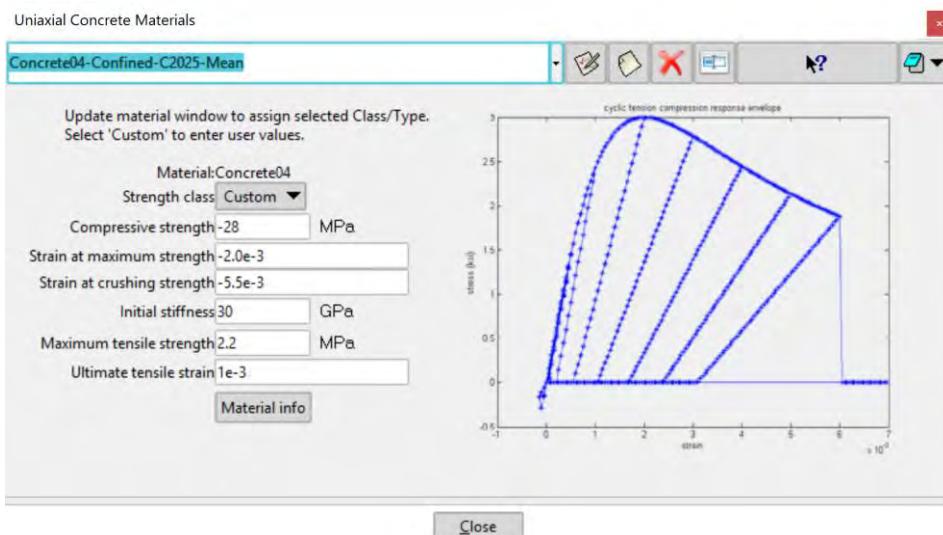


Figure 161: Tutorial 4 – Confined concrete material options

Reinforcing steel uniaxial material is defined for the reinforcing bar fibers' material. Ultimate stress is assumed to be 600 MPa and the corresponding strain to 0.075 m/m.

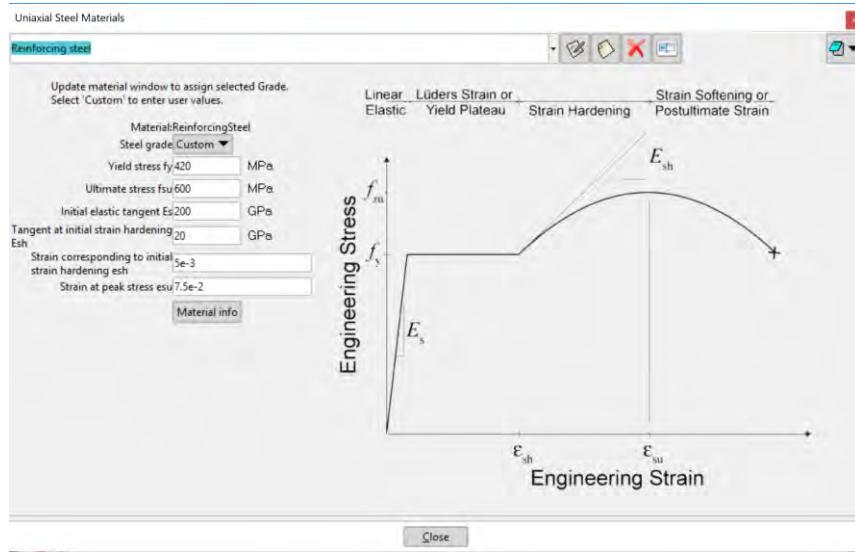


Figure 162: Tutorial 4 – Reinforcing bar material options

Two Fiber section models are created, one for the Column and one for the Tee Beam. The dimensions and reinforcements options are shown in the following Figure.

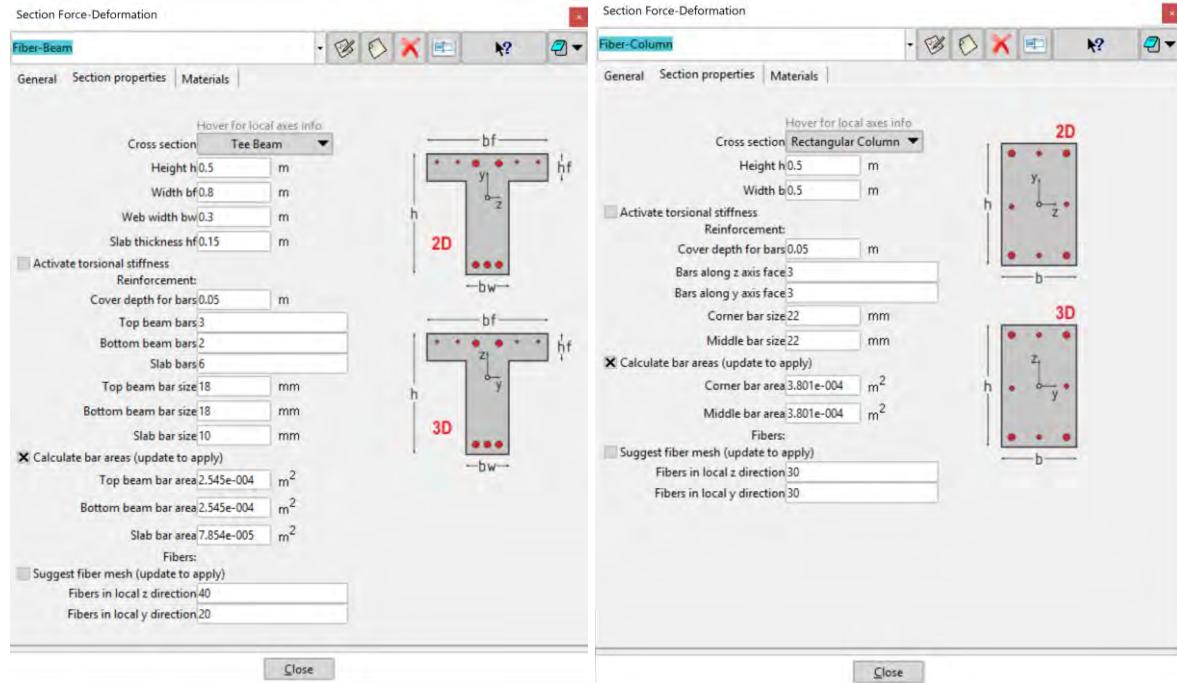


Figure 163: Tutorial 4 – Fiber section model options

Column is reinforced with $8\varnothing 22$ steel bars equally spaced. Beam web is reinforced with $3\varnothing 18$ steel bars at the top and $2\varnothing 18$ steel bars at the bottom. In addition to this reinforcement, $6\varnothing 10$ reinforcing slab bars are added inside the effective width of the beam member.

Both Fiber sections use the pre-defined confined concrete material for the core, as we assume that we efficient transverse reinforcement that allows the concrete to develop larger strain at

crushing strength. Cover material is the pre-defined unconfined Popovics material, which corresponds also to the slab concrete material for the beam. Finally, all reinforcing bars are of the same pre-defined steel uniaxial material. Material options, are common and shown below.

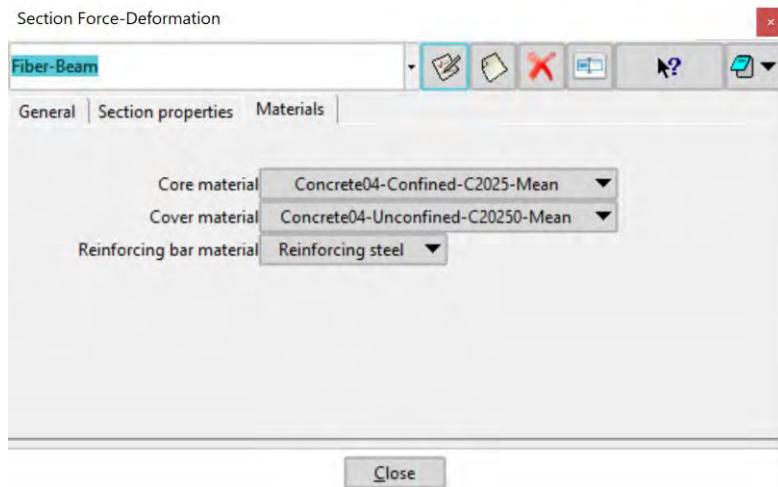


Figure 164: Tutorial 4 - Materials of Fiber Sections

The corresponding Force-based Beam Column elements are then defined () and assigned to the line objects as shown in Fig. 165.



Figure 165: Tutorial 4 – Assigned elements

Pinned connections () are assigned (in Interval 1) to simulate **zero moment conditions** at the top and bottom of the column. As a 2D problem, you need to fix (check in window dialog) x and y translation and not fix the z rotation. The Interface automatically omits any other options.

For gravity analysis, the dead loads are considered through the *Interval Data* options. The analysis is Static monotonic with Load Control and is applied in 5 steps.

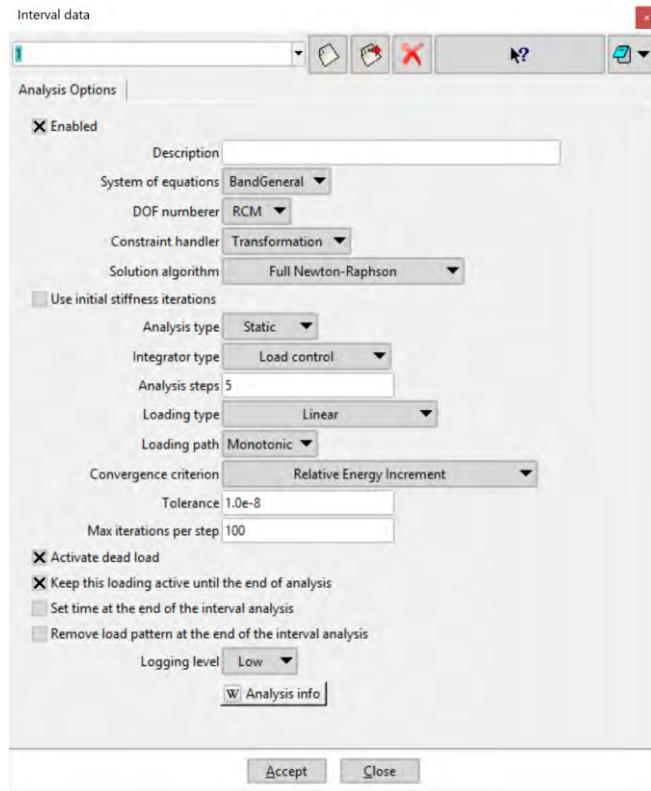


Figure 166: Tutorial 4 – Gravity analysis options

Using **interval 2**, you should create a load pattern, which is used for reaching the desired displacement of the control node using the displacement control integrator. Thus, you can assign a point force () 1 kN along the Y direction to the end of the beam.



Figure 167: Tutorial 4 – Force assigned to beam end

You need to define the *Total displacement* as the maximum displacement to be imposed. Selecting *Cyclic Loading Path*, you can define pairs of displacements peaks and number of cycles. *Displacement peaks* need to be entered as the **ratio** between the displacement and the

total displacement. At this example, the deflection to be impose is 1.2", 2.4" and 3.6", respectively. Consequently, given the total displacement equal to 3.6 inches, the displacement ratios are defined as 0.333, 0.666 and 1, respectively. Three cycles are imposed for each displacement peak.

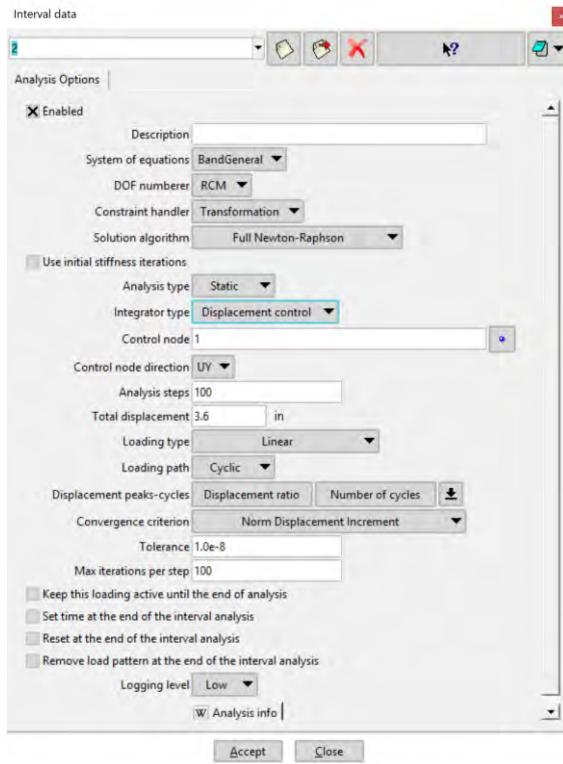


Figure 168: Tutorial 4 – Static reversed cyclic analysis options

Ten elements per line object is selected for the discretization (*Mesh > Structured > Assign number of cells*). After that, the analysis is ready to run ().

Results

The load-deflection curve is possible to be visualized using the graphs function () by creating a point graph using all steps of Interval 2. Component x corresponds to the vertical displacement and component y to the Load Factor. The asymmetry of the curve is mostly a result of the cross-sectional as well as the reinforcement asymmetry of the beam. The strength and stiffness during the negative vertical displacement is expected due to the existence of the effective beam width and the more reinforcement including the slab reinforcing bars as well.

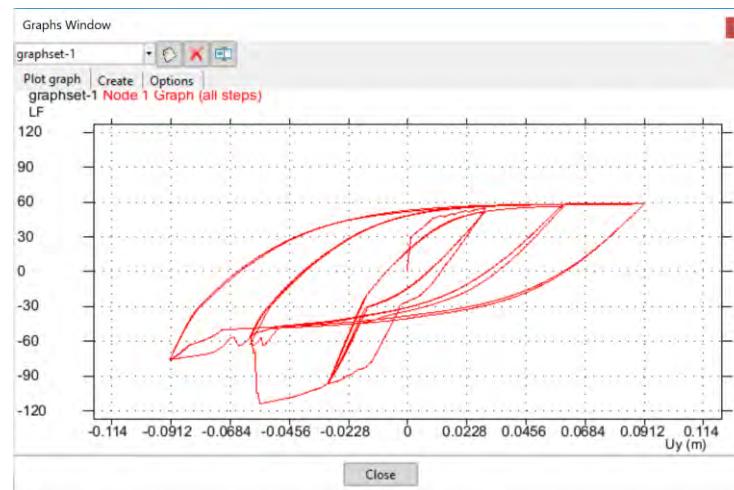


Figure 169: Tutorial 4 – Load – deflection curve at the beam end

Tutorial 5 – Site Response Analysis of a Layered Soil Column

Description

This example describes the GiD+OpenSees implementation of a site response analysis of a layered soil column located above an elastic half-space. The earthquake ground motion is imposed in a manner which accounts for the finite rigidity of the underlying medium. The soil type is assumed to be stiff clay, and the soil characteristics are according to the OpenSees developers' recommended parameter values.

Table 9: Soil column characteristics

Soil Characteristics	Stiff Clay
V_{s,30} (m/s)	359.99
Friction angle φ (°)	0
Apparent cohesion c (kPa)	75
Mass density ρ (tn/m³)	1.8
Poisson ratio ν	0.3

Soil column is thirty meters deep, and each layer has one-meter thickness, which is characterized by a different shear wave velocity Vs, which results the differentiation of Shear Modulus G and Bulk Modulus K at each layer. **For the calculation of K as a function of G, Poisson's ratio ν is assumed to be zero on the concept of simulating a one-dimensional analysis.**

Table 10: Layer characteristics

Layer	Gi (Mpa)	Bi (Mpa)	Vs,i
1	52.57226	35.04817	170.9
2	91.04402	60.69601	224.9
3	117.5964	78.39763	255.6
4	139.1112	92.7408	278
5	157.7088	105.1392	296
6	174.4338	116.2892	311.3
7	189.5405	126.3603	324.5
8	203.6969	135.798	336.4
9	216.7362	144.4908	347
10	229.1512	152.7675	356.8
11	240.9891	160.6594	365.9

12	252.1809	168.1206	374.3
13	262.9383	175.2922	382.2
14	273.2187	182.1458	389.6
15	283.1248	188.7499	396.6
16	292.7716	195.1811	403.3
17	301.9899	201.3266	409.6
18	311.0517	207.3678	415.7
19	319.7921	213.1947	421.5
20	328.3459	218.8973	427.1
21	336.7013	224.4675	432.5
22	344.8463	229.8975	437.7
23	352.7699	235.1799	442.7
24	360.4613	240.3075	447.5
25	368.0727	245.3818	452.2
26	375.4348	250.2899	456.7
27	382.8698	255.2465	461.2
28	389.8749	259.9166	465.4
29	396.9435	264.629	469.6
30	403.905	269.27	473.7

Shear modulus G_i for each layer was calculated as follows:

$$G_i = V_{si}^2 \cdot \rho$$

Bulk modulus K_i for each layer was calculated as follows:

$$K_i = 2 \cdot G_i \cdot \frac{(1 + \nu)}{3 \cdot (1 - 2 \cdot \nu)}$$

Where:

ν : Soil Poisson's ratio

V_{si} : Shear wave velocity for layer i

Problem solution

Stiff clay is simulated as **PressureIndependMultiYield** material type. The properties used, except shear and bulk moduli, were chosen according to the recommended values by the OpenSees Developers. The soil elements are represented by four-node quad elements using the plane strain formulation.

The proper mesh options are based upon a concept of ensuring that an adequate number of elements fit within the wavelength of a shear wave. This shear wave is calculated as the ratio

of the lowest shear wave velocity and the highest frequency desired to be resolved. An adequate number of elements in this wavelength is at least 8.

$$H < \frac{Vs_{min}}{8 \cdot f_{max}}$$

Where:

H : the element size

Vs_{min} : The lowest shear wave velocity

f_{max} : The highest frequency to be resolved

Thus, before starting the Geometry model, we can calculate the desired mesh geometry in order to save computational time.

The highest frequency is assumed to be $f_{max} = 100 \text{ Hz}$, so the element size must be 0.2 m maximum. The final mesh model is 5 elements per layer (each one with 0.2m thickness) in vertical direction. In horizontal direction, the element size must be the lower of the vertical size. Thus, the element size in horizontal direction must be equal to 0.2m too.

First, the analysis will include **3 intervals**. The first two are for gravity analysis and the third and last one is for the dynamic analysis. It is worth mentioning that gravity analysis can be applied as **transient analysis** with very large steps. In this way, you can avoid conflicts by mixing static and transient analyses. The first interval is for entirely elastic behavior, and the second is for elastoplastic behavior and the gravity is repeated.

The first step is to make the grid options () fit to your needs. Make sure that x spacing is equal to 0.2 and y spacing is equal to 1. After that you can make one surface, which correspond to one layer and copy it (Utilities > Copy) number of layers minus one, with relative y distance equal to 1m. Make sure that the *Collapse* box is checked in order to handle the connections of the surfaces properly (merge nodes sharing the same location).

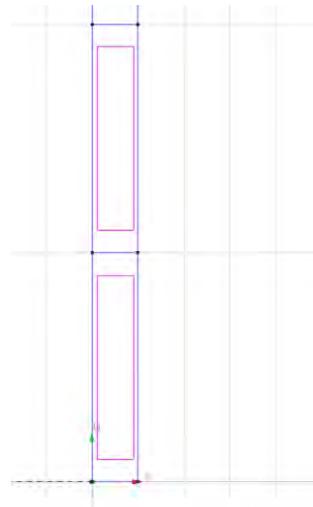


Figure 170: Geometry model of soil column (left) and Meshed model of soil column (right)

Obviously, you could make your model in a larger scale, as shown in the following figure, but the computational time would be increased dramatically. Differentiation of colours correspond to different quad elements (soil layers) which use a different material for each soil layer's properties. However, this is not recommended because of the increase of computational "cost".



Figure 171: Tutorial 5: Meshed model of an alternative simulation (larger) of soil column

The soil base nodes are restrained (☞) in uy direction only, so that they can move in ux direction only.

You need to create the PressureIndependMultiYield materials (☞) that correspond to each layer. You can use the recommended parameters for stiff clay, and then modify the Shear Modulus G and Bulk Modulus K using *Custom* soil type.

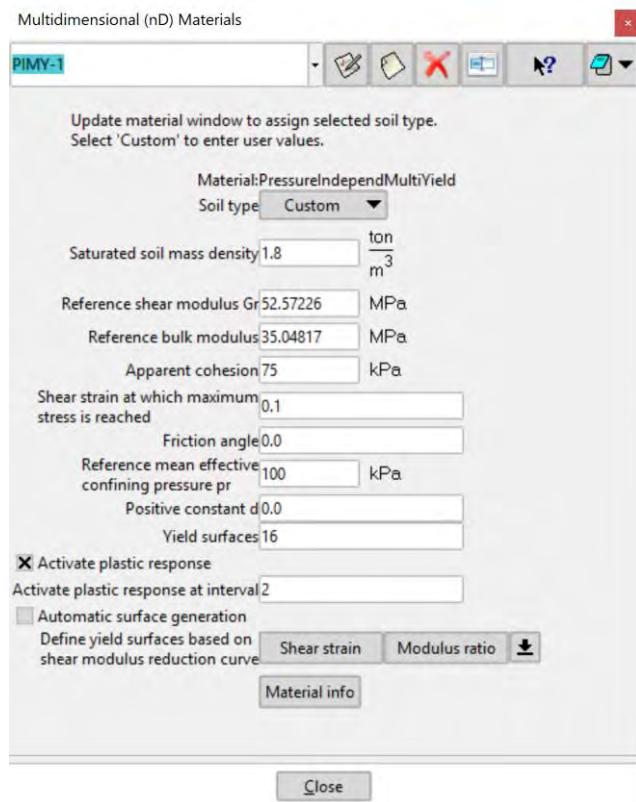


Figure 172: Soil layer material options

You can select that plastic response can be activated at interval 2 (Plastic gravity analysis).

In the last field, you can input the G/Go- γ Curve of the soil material, copying the values from an excel sheet for example.

Shear strain	Modulus ratio
0.000001	1
0.000002	1
0.000005	0.966
0.00001	0.984
0.00002	0.975
0.00005	0.922
0.0001	0.85
0.0002	0.734
0.0005	0.532
0.001	0.367
0.002	0.224
0.005	0.139
0.01	0.085
0.02	0.051
0.05	0.027
0.1	0.021

Figure 173: Shear modulus-strain curve

The next step is to define the element types (Quad elements) that corresponds to each layer (they use different material). In *General* tab you should check the option *Define equalDOF between nodes of the same vertical location*, so that soil nodes are tied together in order to achieve a simple shear deformation pattern.

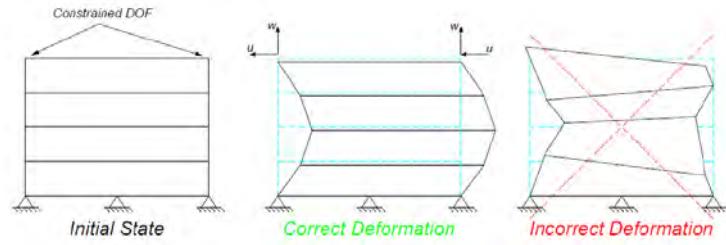


Figure 174: Schematic illustrating the equalDOF command relating to soil column deformation (Chiaramonte, 2011)

In the Material field you can select the previously defined PressureIndependMultiYield material. The plane behavior is set up to Plane Strain. Thickness should be equal to 1 meter. Finally, the Body forces should be input for the gravity. You should enter the mass density times the g acceleration. In this example, it is equal to $-17.658 \frac{kN}{m^3}$. Mass density field is not necessary to be defined, because the mass density of the PressureIndependMultiYield material is inherited, and hence mass density of quad element would be additional to that, influencing the results in a wrong way.

For this type of analysis, in this example the Lysmer-Kuhlemeyer (1969) dashpot is defined. Two extra nodes are required in order to create a single zeroLength element between them. These two nodes must have the same location as the node of the base of the soil column. In order to efficiently handle geometrical objects, which share the same location, we recommend using the *layer and group tools* (*Utilities > Layer and groups*).

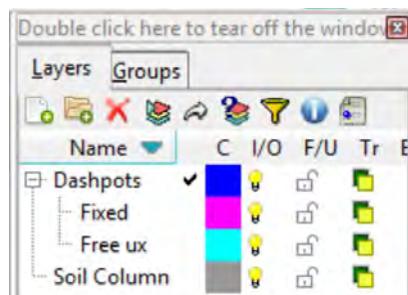


Figure 175 Tutorial 5 – Layer options

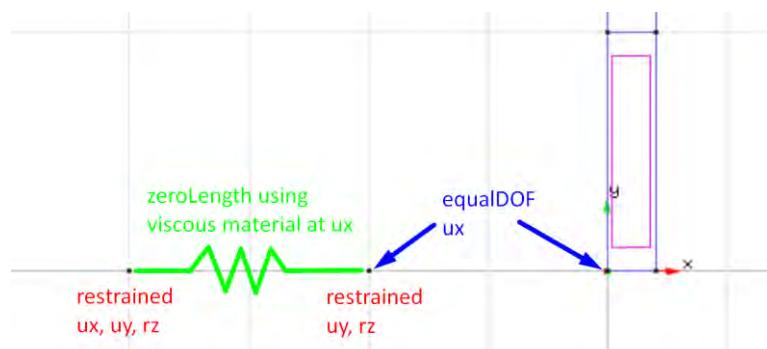


Figure 176: Tutorial 5 - Lysmer-Kuhlemeyer dashpot implementation

In this way, you can turn on/off each layer you desire to apply changes.

Remember that nodes which do not belong to any higher entity, are in defined in domain with 3 degrees of freedom. Thus, all 3 (ux, uy, rz) degrees of freedom of the one dashpot are fixed. The other one is free only at the ux direction. These two nodes are connected with zeroLength element using a viscous material (). Viscous uniaxial material should be defined with a damping coefficient that is calculated as

$$c = \text{colArea} \cdot \text{rockVs} \cdot \text{rockDens}$$

Where:

colArea: the area of the soil column

rockVs: the bedrock shear wave velocity

rockDens: the bedrock mass density

In this example:

$$\text{colArea} = 0.2 \text{ m}^2$$

$$\text{rockVs} = 700 \frac{\text{m}}{\text{s}}$$

$$\text{rockDens} = 2.5 \frac{\text{ton}}{\text{m}^3}$$

Thus, the damping coefficient is equal to 350 kNsec/m.

After that, the zeroLength element () can be assigned to the two dashpots using the Viscous material in the ux direction only. To incorporate the dashpot element into the total model, you should use equal constraint () in ux direction only between the partially fixed dashpot node and the one of the nodes at the base of the soil column.

Rayleigh damping is defined within *General Options*. It is a function of the damping ratio and the circular frequencies of two modes of vibration. The damping ratio for soils is usually set to $\zeta=2\%$. You can use the two frequencies that are the limits of the frequencies in interest for common structures. In this example, the chosen frequencies are 0.2 and 20 Hz. So that Rayleigh parameters are calculated as

$$\begin{aligned} a_0 &= \frac{2 \cdot \zeta \cdot (2 \cdot \pi) \cdot f_1 \cdot f_2}{f_1 + f_2} = \frac{2 \cdot 0.02 \cdot 2 \cdot 3.1415 \cdot 0.2 \cdot 20}{0.2 + 20} \\ &= 0.04976780441980199 \end{aligned}$$

$$a1 = \frac{2 \cdot \zeta}{(2 \cdot \pi) \cdot (f1 + f2)} = \frac{2 \cdot 0.02}{2 \cdot 3.1415 \cdot 20.2} = 0.00031515830311111683$$

Where:

$a0$: the mass-proportional damping coefficient

$a1$: the stiffness proportional damping coefficient

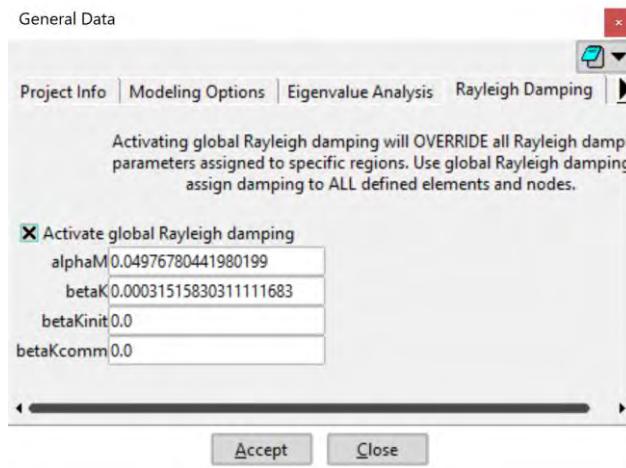


Figure 177: Tutorial 5 – Rayleigh damping parameters

For the dynamic analysis, the load pattern must be defined for the horizontal loading using the Joyner and Chen method (1975), with which the dynamic excitation is applied as a force time history to the base of the soil column at the particular node which is connected (equal constraint) with the Lysmer-Kuhlemeyer (1969) dashpot node. The force time history is resulted by multiplying the velocity ground motion time history by the damping coefficient (350 kNsec/m), which was previously discussed.

It is necessary then to define the Ground motion record file, which is used for the force time history (). The record file corresponds to velocity ground motion (m/s), so the scale factor is equal to 350 (such as the damping coefficient of the viscous material).

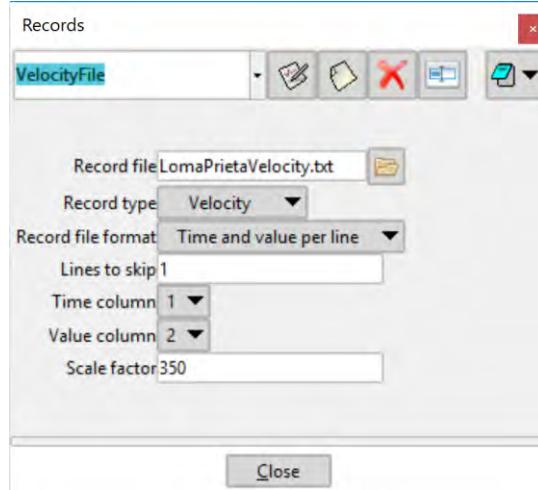


Figure 178: Tutorial 5 – Record file used

The first two interval options are shown on the following figures.

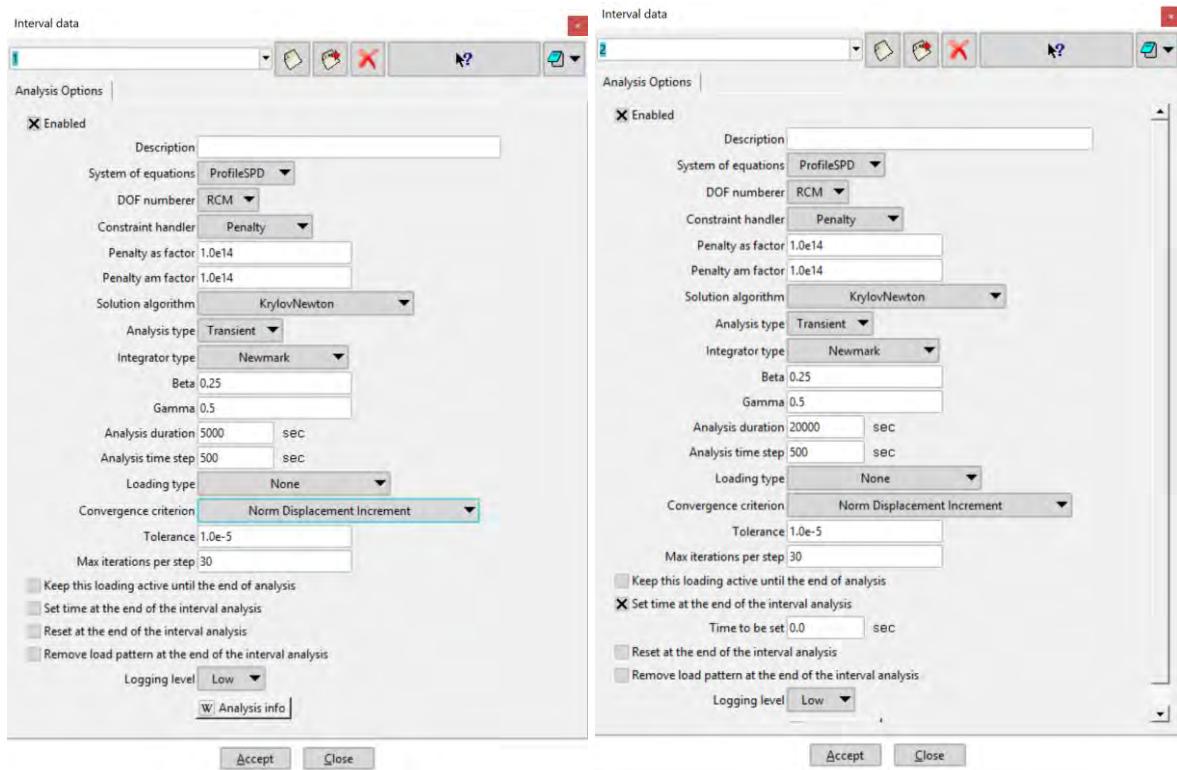


Figure 179: Tutorial 5 – Elastic (left) and Plastic (right) Gravity analysis options

No pattern is needed to be selected as you have defined the body forces in the Quad elements definition, so the loading type can be *None*. In the second interval, you should check the *Set time* option and set the time to zero, which is necessary for considering the record file properly.

In the third interval, you should apply a unit force () on the corner base node in x direction. In combination with setting the *Loading type as Function* and choosing the proper record file, the desired load pattern will be created.

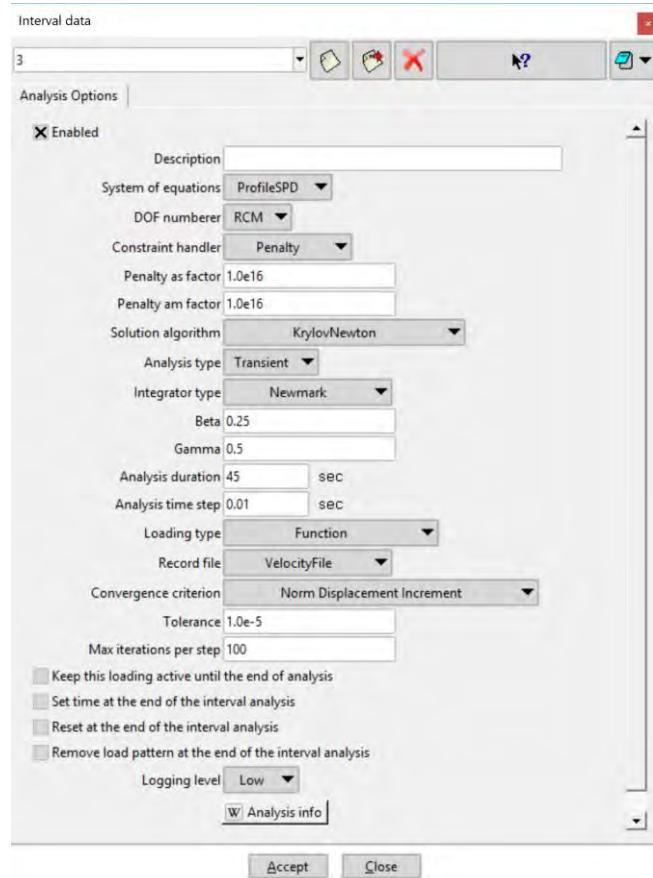


Figure 180: Tutorial 5 – Dynamic analysis options

The analysis is then ready to run ().

Results

You can create the acceleration/velocity/displacement time history response of the ground surface using the graph window features (). Selecting point graph including all steps (in interval 3), you can select (*right click > contextual > join or select nodes*) existing nodes (surface and/or base nodes). Node 1 corresponds to the surface and node 301 to the soil column base in the following figures.

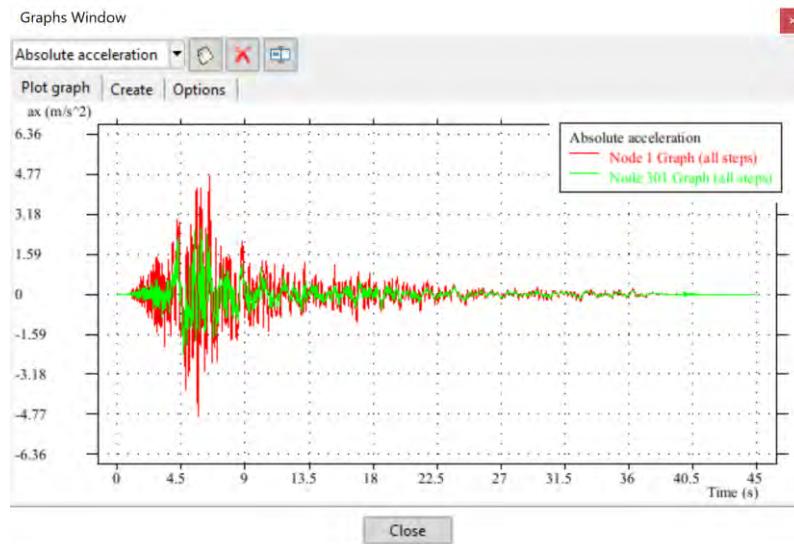


Figure 181: Tutorial 5 – Absolute Acceleration time histories at surface (red) and base (green)



Figure 182: Tutorial 5 – Absolute displacements of soil base (green) and surface (red)

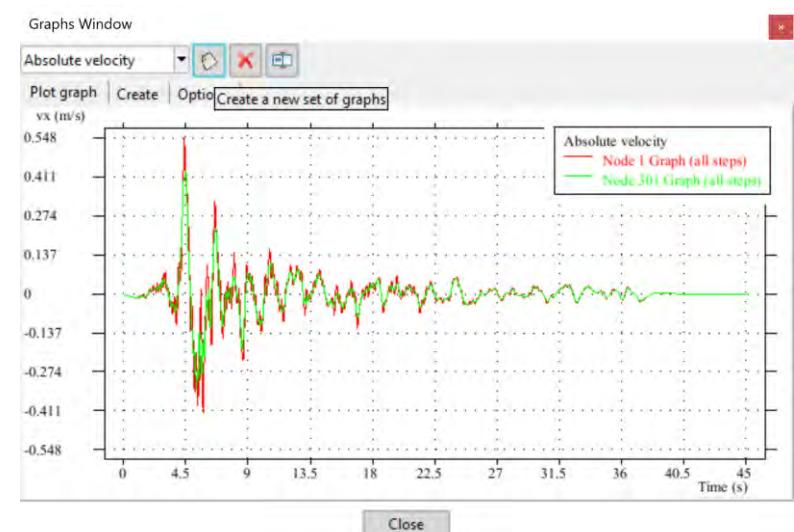


Figure 183: Tutorial 5 – Absolute velocities time histories at soil base (green) and surface (red)

From the options tab you can copy the results to an excel sheet and compare the surface response with the original acceleration time history of the ground motion.

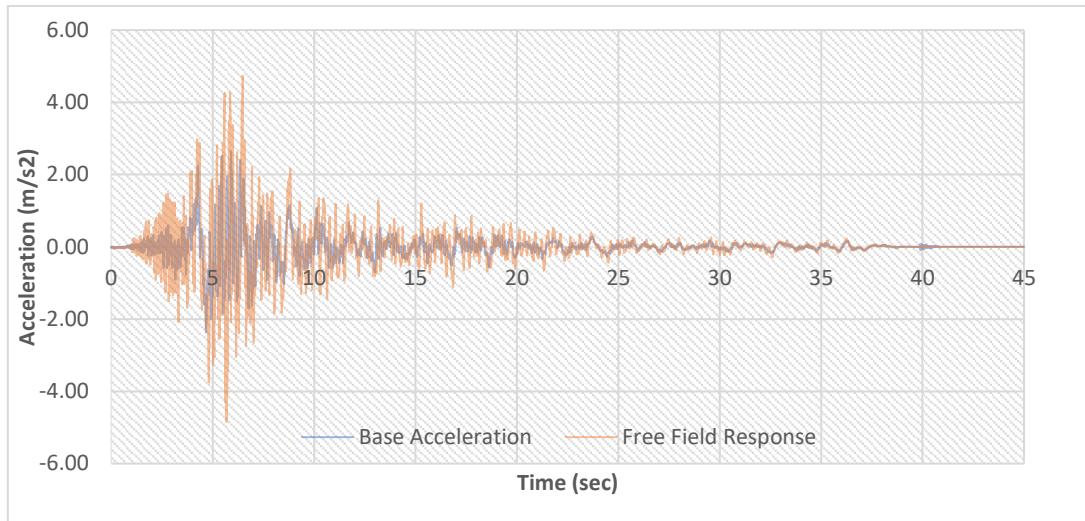


Figure 184: Comparison of surface acceleration response versus original acceleration ground motion recorded

In the following chart, the acceleration response spectrum is observable, which was obtained from the time history response of the surface, using the SeismoSignal Software package.

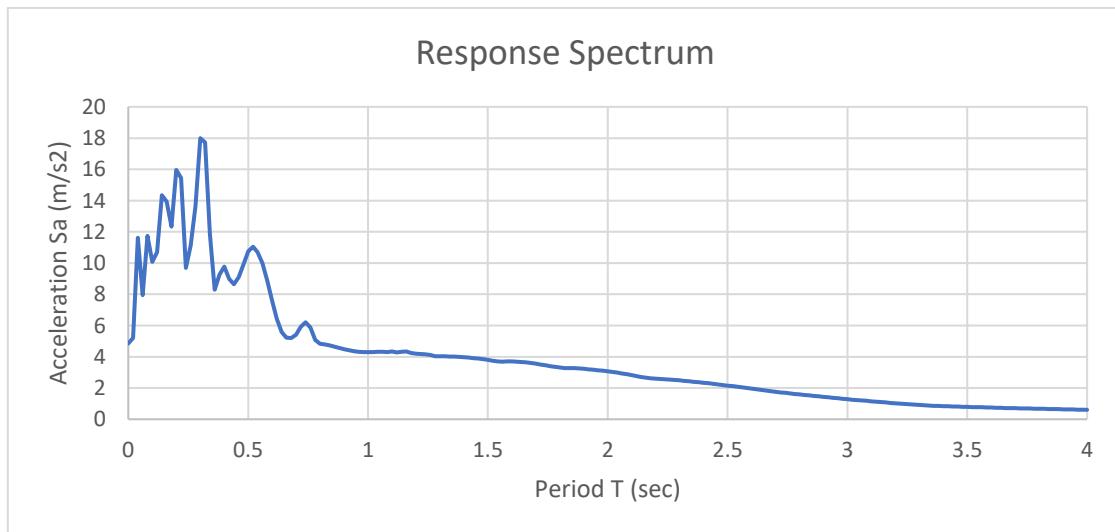


Figure 185: Tutorial 5 - Response Spectrum of surface response time history

Tutorial 6 – Pushover Analysis on 2D Frame considering Soil sustainability

Description

This example describes the GiD+OpenSees implementation of a two-dimensional Pushover analysis on a 3-story frame structure above an elastic soil column for concerning the influence of the soil sustainability as well. The soil type is assumed to be a loose sand and gravel with elastic behavior.

The geometrical and cross-sectional dimensions of the structure are demonstrated in tables 11 and 12.

Table 11: Frame characteristics

Bay length (m)	5
Ground level height (m)	4
Story height (m)	3

Table 12: Structural member properties

Column cross section (cm x cm)	40x40
Beam cross section (cm x cm)	25x60
Footing cross section (cm x cm)	200x80

The longitudinal reinforcement of the frame structure is shown in table 13:

Table 13: Frame reinforcement

Ground level columns	12Ø20
Other columns	8Ø20
Beam – 1st floor	5Ø18 top - 4Ø16 bottom
Beam – 2nd floor	4Ø18 top - 3Ø16 bottom
Beam – 3rd floor	3Ø16 top - 3Ø16 bottom

Problem Solution

Frame members are simulated with Force-based beam column elements (). Two concrete04 uniaxial materials () are defined for the confined and unconfined region, respectively. Confined region is assumed to be capable of developing a maximum 0.006 strain in compression in contrast with unconfined, where maximum strain is equal to 0.0035.

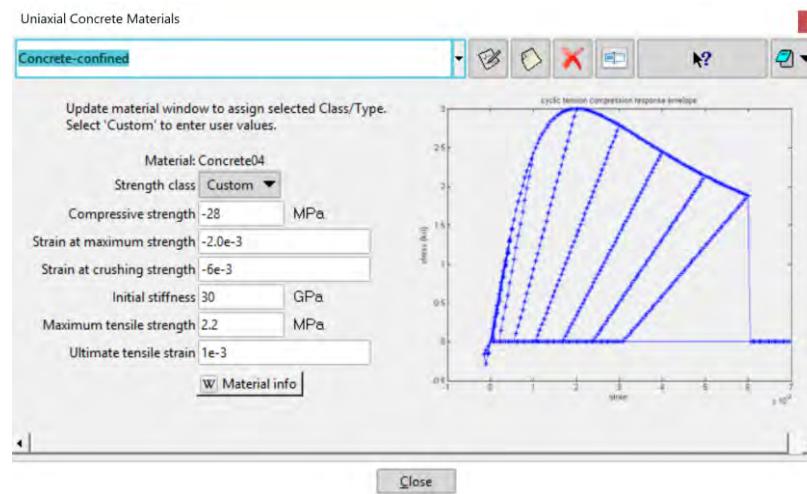


Figure 186: Tutorial 6 – Confined concrete material options

Reinforcing bars are simulated with steel02 uniaxial material () with its default values (B500C).

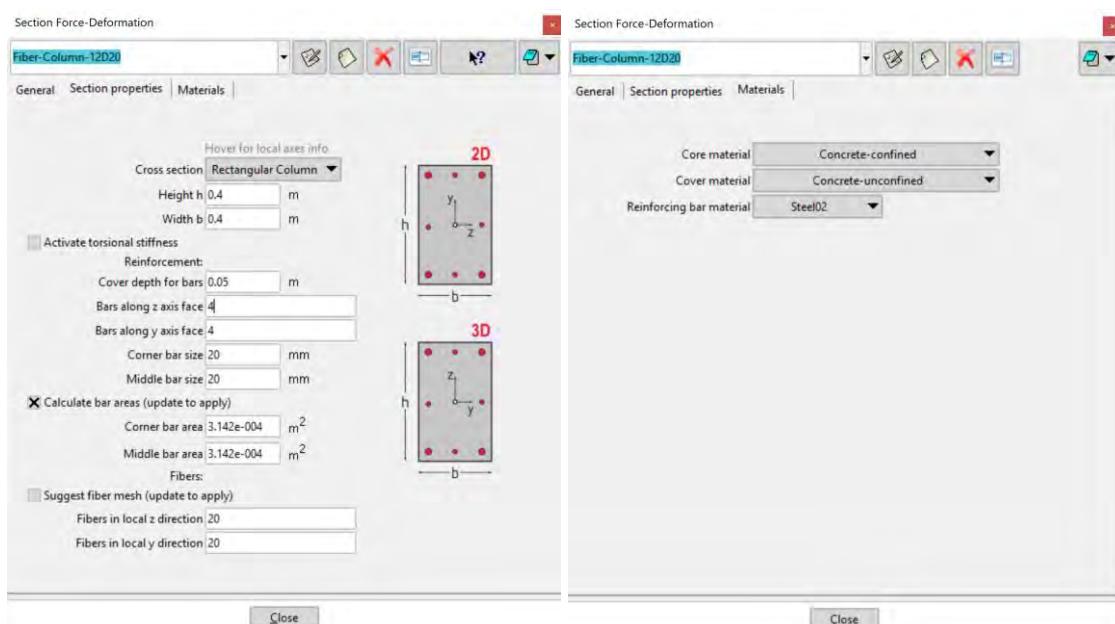


Figure 187: Tutorial 6 – Ground level column: Fiber section options

Soil elements are simulated with Quad elements (□) using the material properties of an Elastic Isotropic nD material (nD) with Young's modulus and Poisson's ratio equal to 100 MPa and 0.3, respectively. Note that an alternative simulation could be using an equivalent Single degree of freedom system for more simplicity.

Frame is simulated with Force-based Beam Column elements and as a result its nodes have 3 degrees of freedom (ux, uy and rz). On the other hand, Soil is simulated with quad elements and their nodes have 2 degrees of freedom (ux and uy). Consequently, **to make these two systems work together you need to apply equal constraint conditions** between the nodes which share the same location. However, they can be designed with some distance between them, without influencing the total behaviour. Otherwise you should use the layer tools (*Utilities > Layer and group tools*) separating the structure from the soil and manage efficiently one system each time.

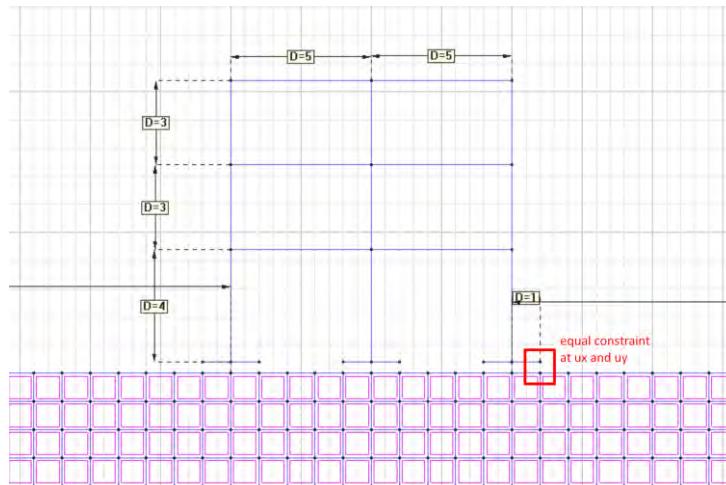


Figure 188: Connecting structure with the soil using equal constraints

Equal constraints are also required between nodes of the lateral limits, which share the same vertical (Y) location, at both translational directions.

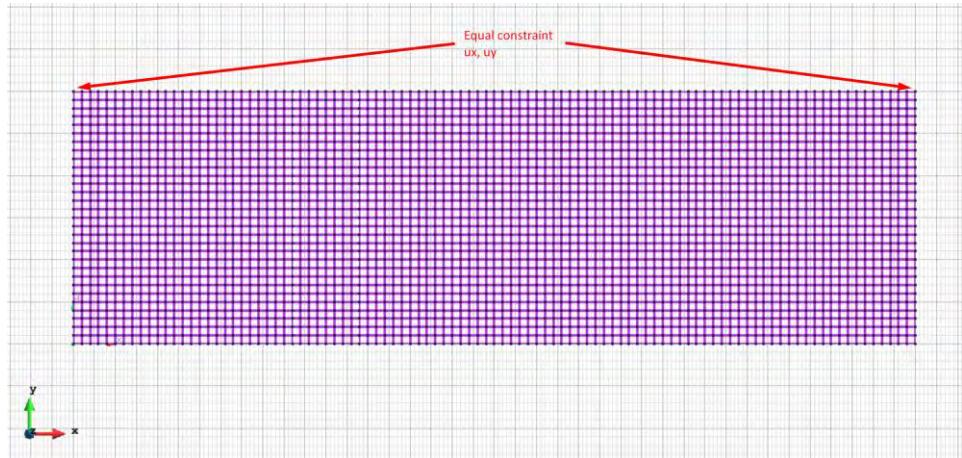


Figure 189: Tutorial 6 - Equal constraints between lateral soil column limits

The soil base nodes are restrained (🔗) in both translation direction, so that they cannot move. All the previous actions mentioned must take place at the first interval. The first interval corresponds to gravity analysis.

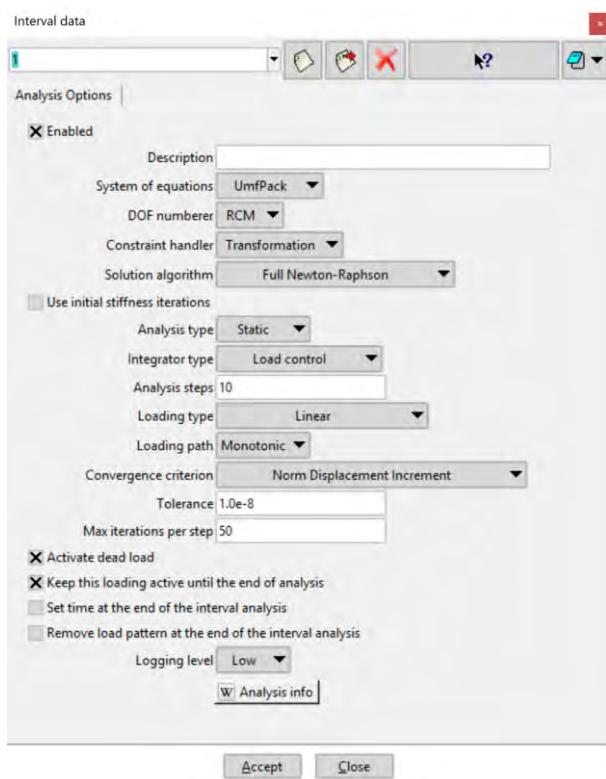


Figure 190: Tutorial 6 - Interval 1 and 2 (gravity) options

Gravity is simulated as 200 kN to each floor (⬇️) in addition to the dead loads coming from the cross-sectional properties and the specific weight of the frame elements (25 kN/m³). Dead loads are activated through Intervals data as depicted in Fig. 190. The loading type chosen is Linear with 10 analysis steps, so that the 10% of the total gravity loads are applied in each time

step until the total 100% is applied at the final step. The body forces of soil elements are given in their element definition window in *Body Forces* tab (-17.658 kN/m³).

Interval 2 corresponds to the lateral pushover analysis, so you need to create the load pattern used for pushing the structure, using the interval 2.

For simplicity, the applied force distribution is expressed by the following equation (triangular distribution along the height):

$$F_i = F_b * \frac{zi}{\sum zi}$$

Where:

F_b : The base shear force.

zi : The height of the story

Assuming that $Fb = 1 \text{ kN}$, we have:

$$F1 = 0.190476 \text{ kN}$$

$$F2 = 0.3333 \text{ kN}$$

$$F3 = 0.47619 \text{ kN}$$

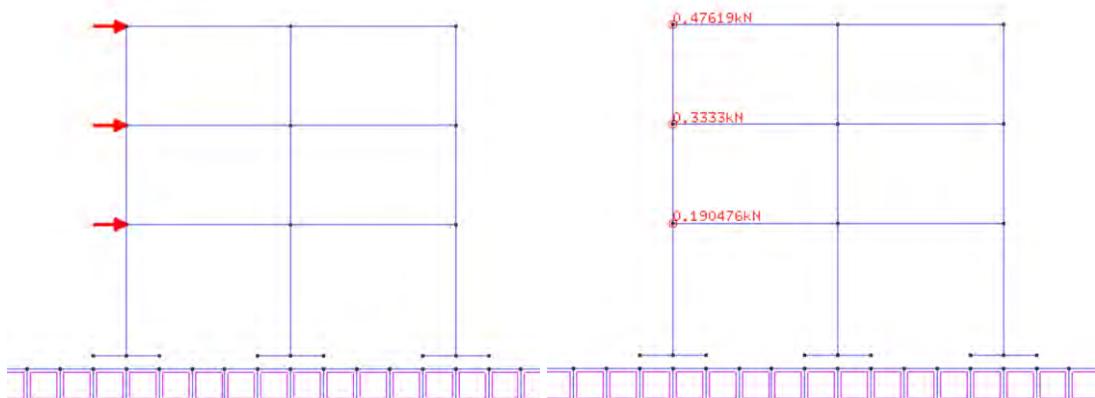


Figure 191: Load pattern for lateral pushover analysis

Soil elements were divided into 1m x 1m element size like the geometry surfaces. It is worth mentioning that the same meshed model could be generated using one only surface because soil column consists of one only soil layer. Frame elements were divided into 10 parts per line, using weights to the end of line, concentrating the nodes near the beam and column ends, where inelasticity is firstly and mostly developed.

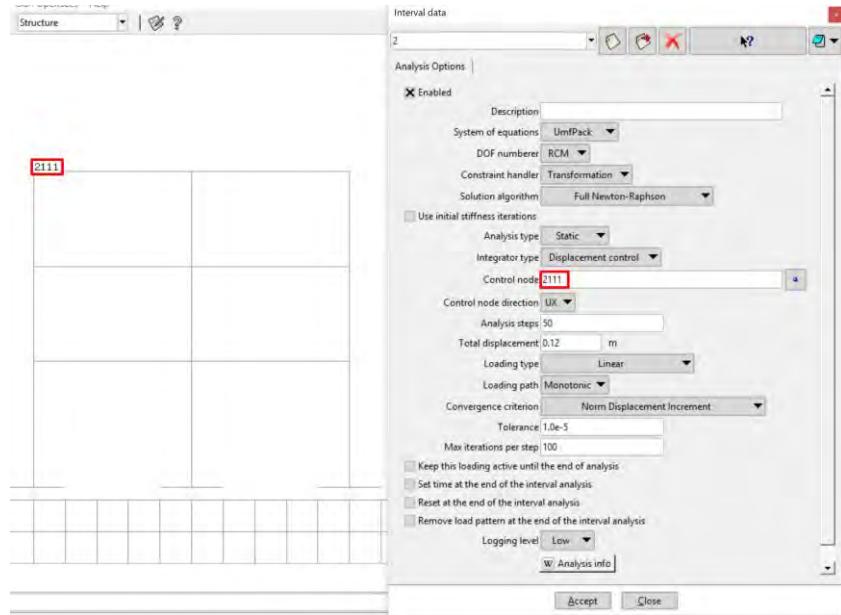


Figure 192: Tutorial 6 - Interval 2 Pushover analysis options

Results

The vertical displacements using contour filling on the deformed shape are depicted in Fig. 193.

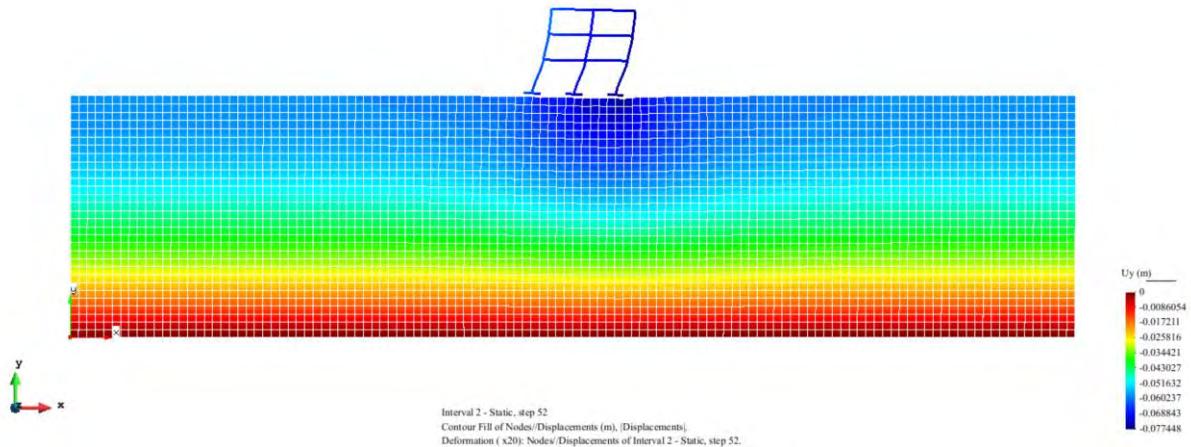


Figure 193: Tutorial 6 – Vertical nodal displacements (m)

The horizontal floor displacements using contour filling at the final stage are depicted in Fig. 194.

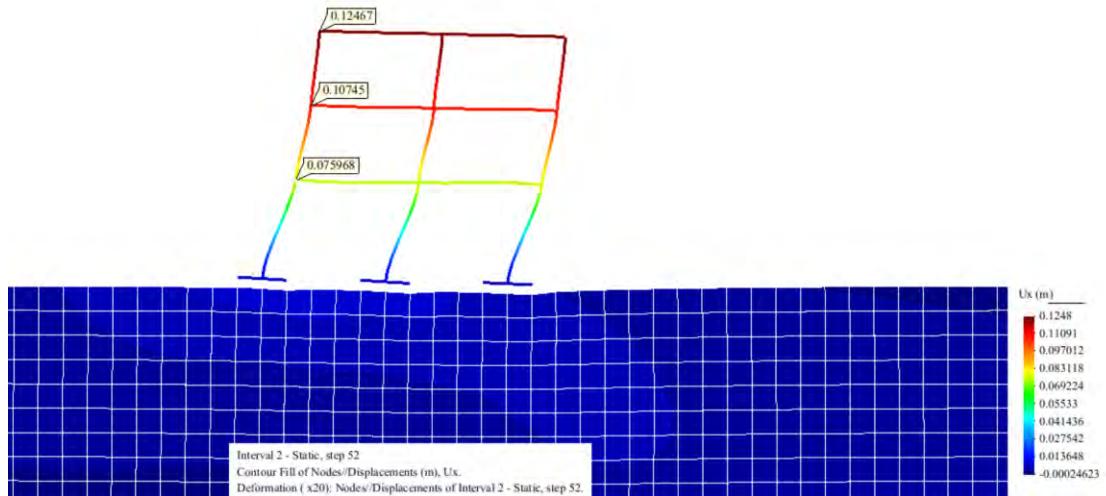


Figure 194: Tutorial 6 - Horizontal nodal displacements (m)

The normal vertical stresses s_{22} are demonstrated in the following Figure, where the influence of interaction between structure and soil is observable. The initial meshed model is also visible here.

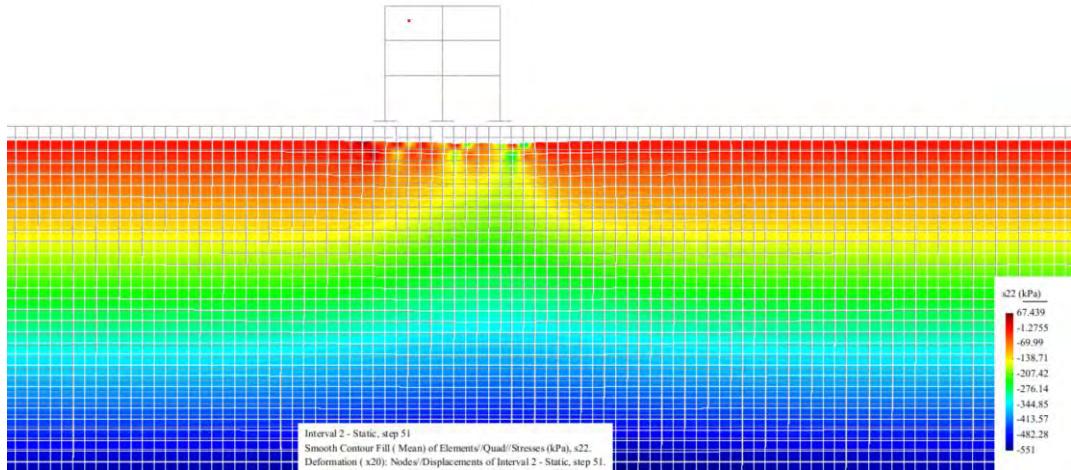


Figure 195: Tutorial 6 - Normal vertical stresses s_{22} (kPa)

Similarly, the shear strains e_{12} can be obtained as shown in Fig. 196.

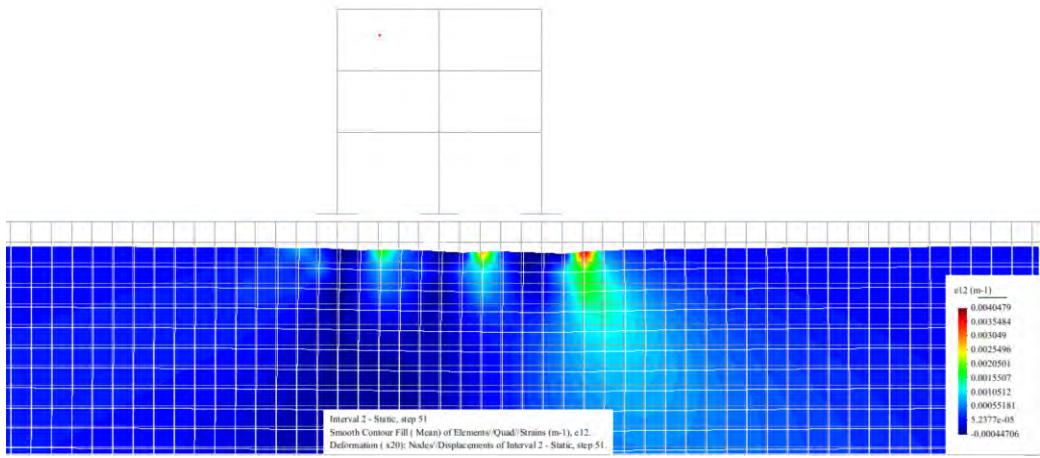


Figure 196: Tutorial 6 - Soil shear strains e_{12} near the structure

The plastic rotations are depicted through a line diagram in Fig. 197. The largest plastic rotations are developed at the ground level especially at the bottom of the ground level columns, as expected. It shows the distribution of the inelasticity in the structure.

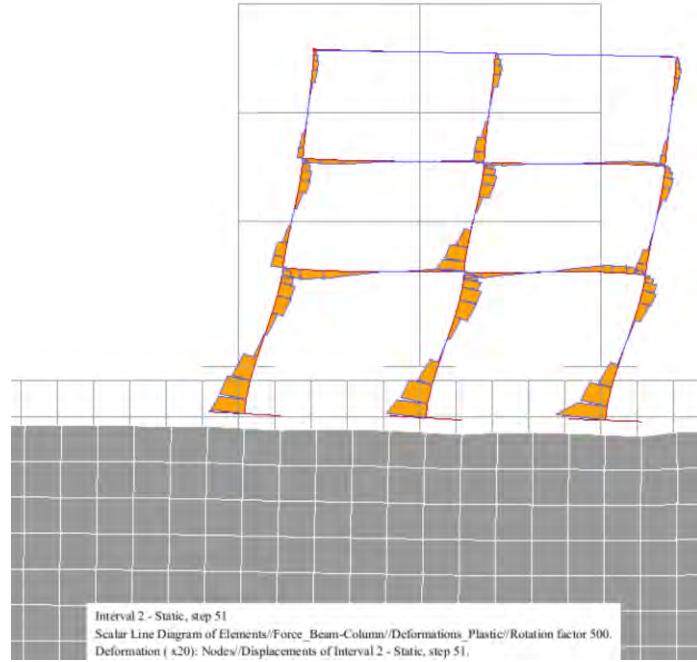


Figure 197: Tutorial 6 - Plastic rotations

Finally, the pushover curve () can be obtained as depicted in Fig. 198. Load factor is equal to shear base because the sum of the lateral forces applied is equal to 1 kN. The displacement corresponds to the top floor.

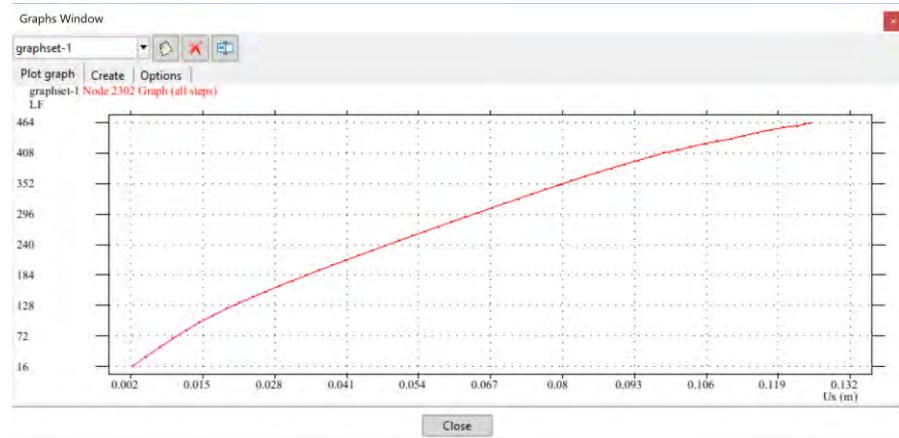


Figure 198 Tutorial 6 – Pushover curve

REFERENCES

- [1] F. McKenna, G. Fenves and M. Scott, "Open System for Earthquake Engineering Simulation," *PACIFIC EARTHQUAKE ENGINEERING RESEARCH CENTER*, 2000.
- [2] V. K. Papanikolaou, T. Kartalis Kaounis, E. K. Protopapadakis and T. Papadopoulos, "A New Graphical User Interface for OpenSEES," 2017.
- [3] X. Lu, L. Xie, H. Guan, H. Yuli and X. Lu, "A shear wall element for nonlinear seismic analysis of super-tall buildings using OpenSees," *Finite Elements in Analysis and Design*, June 2015.
- [4] D. Palermo and F. J. Vecchio, "Compression Field Modeling of Reinforced Concrete Subjected to Reversed Loading: Formulation," *ACI Structural Journal*, September 2003.
- [5] A. Neuenhofer and F. C. Filippou, "Geometrically Nonlinear Flexibility Based Frame Finite Element," *Journal of Structural Engineering*, 1998.
- [6] A. Neuenhofer and F. F. Filippou, "Evaluation of Nonlinear Frame Finite-Element Models," *Journal of Structural Engineering*, 1997.
- [7] M. Scott, "Numerical Integration Options for Force-Based Beam-Column Element in OpenSees," 2011.
- [8] L. L. Yaw, "3D Co-rotational Truss Formulation," 2011.
- [9] A. K. Chopra, *Dynamics of Structures*, Pearson Prentice Hall, 2012.
- [10] F. A. Charney, "Unintended Consequences of Modeling Damping in Structures," *Journal of Structural Engineering*, 2008.
- [11] S. Mazzoni, F. McKenna, M. H. Scott and G. L. Fenves, "The Open System for Earthquake Engineering Simulation (OpenSEES) User Command-Language Manual," 2006.

- [12] V. Cervenka, L. Jendele and Cervenka J., "ATENA program Documentation Part 1: Theory," 2017.
- [13] Coll, A.; Ribó, R.; Pasenau, M.; Escolano , Enrique; Perez, J. S.; Melendo, A.; Monros, A.;, "GiD v.13 Customization Manual," CIMNE, 2017.
- [14] A. Elgamal, Z. Yang and E. Parra, "Computational modeling of cyclic mobility and post-liquefaction site response," *Geotechnical and Geoenvironmental Engineering, ASCE*, 2003.
- [15] F. F. Taucer, E. Spacone and F. F. Filipou , "A Fiber-Beam-Column Element for Seismic Response Analysis of Reinforced Concrete Structures," *Earthquake Engineering Research Center*, 1991.
- [16] E. Spacone and F. C. Filippou, "Fiber Beam-Column Model for Nonlinear Analysis of R/C Frames," *Earthquake Engineering & Structural Dynamics*, July 1996.
- [17] J. W. Wallace, "Displacement-Based Design of Slender Reinforced Concrete Structural Walls - Experimental Verification," *Journal of Structural Engineering*, 2004.