ZEUS NL

A System for Inelastic Analysis of Structures

User Manual

Version 1.8.9

Amr S. Elnashai

Vassilis K. Papanikolaou

Do Hyung Lee

1 INTRODUCTION

Zeus Nonlinear (ZeusNL) provides an easy and efficient way to run accurate nonlinear dynamic time-history, conventional and adaptive pushover, and eigenvalue analysis. Unlike other similar analysis packages, dynamic analysis is now a matter of basic simple steps, using a completely visual approach. This means that the user can create a structural model just by point-n-click and then let the program take care of all the analysis details.

1.1. Technical Capabilities

ZeusNL can be used to predict the large displacement behavior of plane and space frames under static or dynamic loading, taking into account both geometric and material nonlinear behavior. Concrete and steel material models are available, together with a large library of 3D elements that can be used with a wide choice of typical pre-defined steel, concrete and composite section configurations. The applied loading can be constant or variable forces, displacements and accelerations.

ZeusNL has the ability to perform eigenvalue, static pushover, static time-history and dynamic analysis, as follows:

- **Eigenvalue analysis.** The efficient Lanczos algorithm is used for the evaluation of the structural natural frequencies and mode shapes.
- **Static pushover analysis (conventional and adaptive).** In the conventional pushover analysis, the applied loads (displacement, forces or a combination of both) vary proportionally according to a predefined pattern. The post-peak response is obtained with different displacement control procedures. With the new adaptive pushover, the applied load pattern is not constant, but varies throughout the loading procedure, in order to more accurately describe the stiffness degradation and the period elongation of the system.
- **Static time-history analysis.** The applied loads (displacement, forces or a combination of both) vary independently in the pseudo-time domain, according to a prescribed load pattern.
- **Dynamic analysis.** The applied load is usually acceleration at the supports (although forces can also be used). Both synchronous and asynchronous excitation can be modeled. The Hilber-Hughes-Taylor or Newmark integration algorithms may be employed.

1.2. How to Use this Manual

This manual explains the operation and features of ZeusNL. It has been separated into a number of sections, which allows it to be used it as a reference guide, as well as an initial tutorial on using the program.

To get started using this software, follow the instructions, later in this chapter, which explain how to install ZeusNL. Once the software is installed, the rest of the program can be explored, referring to Section 3 when necessary or by following the step-by-step Tutorials in Section 2. It is strongly suggested to follow the Tutorials, as it will get the user up and running in the quickest time possible.

1.3. Conventions

There are a number of terms and conventions used in this manual that the user should become familiar with:

- **Menu commands :** For example, **Menu Name > Command Name**, such as **File > Save**, means 'open the File menu and click the Save command'.
- **Model :** The model of the structure that is created with ZeusNL. It includes the complete description of the structure from the materials and sections types, to the nodes, elements and restraints.
- **Windows® :** This refers to the Microsoft Windows® product line; that is Windows® 95/98, Windows® Me, Windows NT®, Windows® 2000 and Windows® XP. Note that the program is not supported by Windows® 3.1 systems.

- **Project :** This refers collectively to the files and options that are used in ZeusNL for a particular analysis. ZeusNL input data files are saved and loaded with the extension **.dat**. However, there are other files created during the formation of the model (e.g., the input curve files **.crv** that are files that describe the loading of the modeled structure). A complete description of all the files created by ZeusNL will be given in Section 4.
- **Dialog boxes :** These are the windows that open for data input. The user either enters the required data and clicks **OK** to accept the entries or clicks **Cancel** to cancel the operation.
- **Pop-up menus :** The shortcut menus that appear when the user right-clicks on parts of the ZeusNL windows. These menus allow for fast and easy execution of the most commonly used functions. Pop-up menus are attached to all the ZeusNL tables and diagrams.

1.4. What is Included

The program is delivered as an installation executable ZeusNL <version number> Setup.exe. There is a time limit on the program until the end of the current year.

1.5. System Requirements

The following table depicts the requirements for using ZeusNL.

Part	Requirement
Processor	Pentium III or higher
RAM	64 MB or higher (128 MB recommended)
Hard Disk	11 MB of free space for the installation; however, running projects with large models (hundreds of nodes), especially in time-history analysis may result in extremely large output files, sometimes more than 100MB
Video adapter	Windows® operating system supported graphic adapter; minimum resolution 1024x768
Operating System	Windows® 95/98, Windows® Me, Windows® NT, Windows® 2000, Windows® XP.

1.6. Installing ZeusNL

1. Start up the Windows® operating system.
2. Insert ZeusNL installation CD-ROM into the CD-drive.
3. Normally the setup program will start automatically. If not, open the CD-ROM contents and run the Setup executable.
4. Follow the instructions and the automated installation program will proceed to copy ZeusNL onto the hard drive. Normally, the default settings suggested by the installation should work well. ZeusNL shortcuts will be added to the desktop and to the **Start Menu** under **Programs > ZeusNL**.

1.7. Program Features

ZeusNL represents a revolution in Finite Element packages. Quite simply there are not many FE tools that put as much power into the user's hands as easily as ZeusNL.

Some of the many features of ZeusNL:

- Completely visual interface. No input or configuration files or programming scripts.
- Full control over adding, modifying and deleting material models, section types, nodes, elements, restraints and loads.
- Six different types of analysis: dynamic and static time-history, conventional and adaptive pushover, eigenvalue and static with non-variable loading.
- The program accounts for both material and geometrical nonlinearities.
- Accurate and thoroughly tested concrete and steel material models.
- A large variety of RC, steel and composite sections.
- The spread of inelasticity along member length and across section depth is explicitly modeled in ZeusNL allowing for accurate estimation of damage accumulation. This feature sets ZeusNL apart from most of the similar tools that use lamped inelasticity to model the members' non-linear behavior.
- High stability and accuracy at very high strain levels enabling precise determination of the collapse load of structures.
- The applied loading may consist of constant or variable forces, displacements and accelerations at the nodes. The variable loads can vary proportionally or independently in the pseudo-time or time domain.
- The innovative Adaptive Pushover Procedure. In this pushover method, the lateral load distribution is not kept constant, but is continuously updated during the analysis according to the modal shapes and participation factors derived by eigenvalue analysis carried out at the current step. In this way, the current stiffness state and the period elongation of the structure at each step, as well as higher mode effects, are accounted for.

- Integration with the Windows® operating system environment. Data can be pasted to the ZeusNL input tables from spreadsheet programs, such as Microsoft® Excel; whereas everything that may appear in the program windows can be copied back (e.g., to word processing programs, such as Microsoft® Word), including input and output data, high quality graphs, the models' deformed and undeformed shapes and more. Moreover, AVI movie files can be created describing the sequence of the structures' deformed shapes.
- With the new **Template** facility, the user can create regular or irregular, 2D or 3D models and run all types of analyses in a few seconds.
- Advanced post-processing facilities for deriving graphs and deformed shapes, easily and efficiently.

2 TUTORIALS

This chapter will walk the user through the first analysis types of ZeusNL. ZeusNL was designed with both ease-of-use and flexibility in mind. The goal is to run analyses (even dynamic time-history analysis) in just a few minutes. It is actually much easier to use ZeusNL than it is to describe. Once the user has become familiar with a few important concepts, the entire process is quite intuitive.

Although the whole process will last no longer than a few minutes, the model that is created has many features and can efficiently and accurately simulate real structures. For many people, this is all the functionality they will ever need by ZeusNL. Section 3 goes into further detail about all of the powerful features of ZeusNL and more ways to increase productivity.

2.1. Tutorial 1 – Dynamic Analysis

It is assumed hereinafter that the user is trying to model a 3D, four-story RC structure and run dynamic time-history analysis for a specified record. It is also assumed that the structure is regular, has two bays and consists of two parallel frames. The bay lengths are 5m, the story heights are 3m and the distance between the two frames is 4m.

2.1.1. Structural configuration

To open the **Template** window, select **File > Create from Template** menu command or click on the **Template** button of the toolbar. The user will be presented with a screen full of options about the structural configuration :

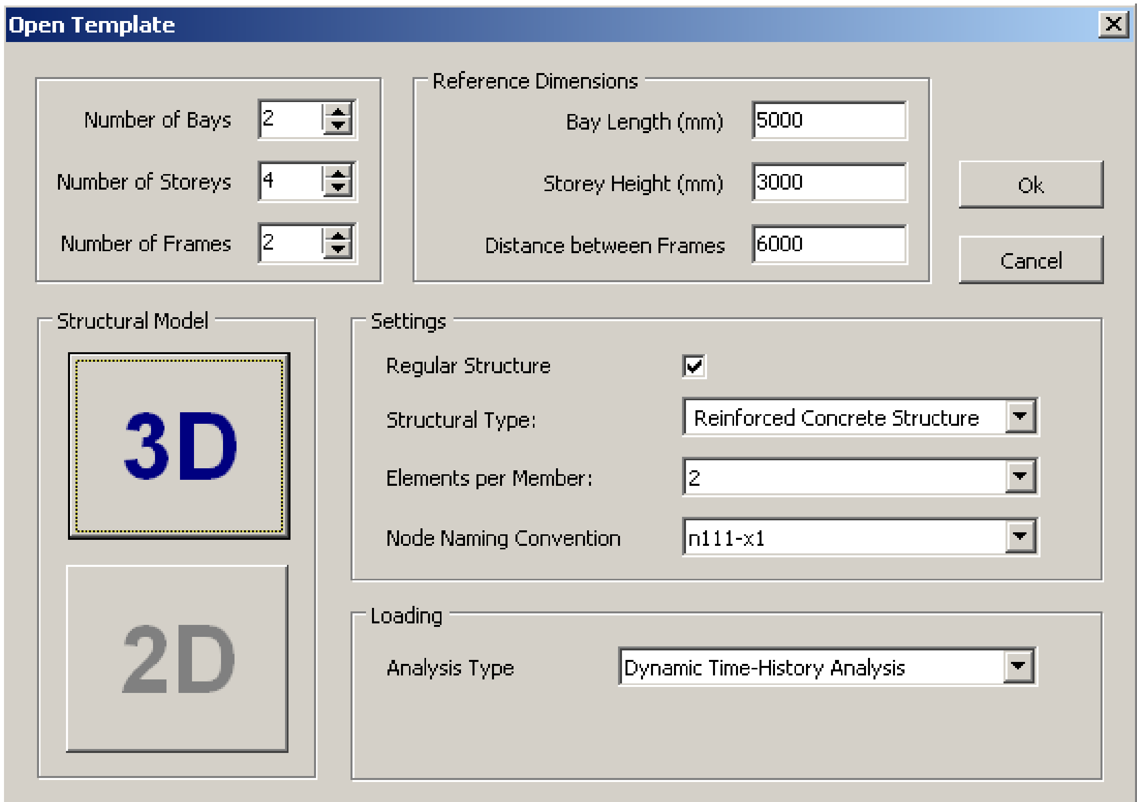


Fig.1 Template screen one (structural configuration).

- **3D-frame or 2D–frame.** Choose a 3D frame.
- **Number of bays, stories, frames.** Select two bays, four stories and two frames.
- **Regular structure.** For the time being, a regular model will be used. In Section 3, the option for modeling structures is discussed.
- **Bay length, story height, distance between frames.** ZeusNL is using mm for length units; choose 5000mm, 3000mm or 6000mm respectively.

In ZeusNL, length units are always millimeters and force units are Newtons.

- **Structural type (RC or steel structures).** Select RC.
- **Elements per member.** This option determines how many elements each member (column and beam) will be subdivided into. Select two elements per member.
- **Node naming convention.** This determines the way in which the nodes are to be named. The first option (default) yields node names easier to read. Select the default.
- **Analysis type.** The user can select one of the six analysis types of ZeusNL. Choose dynamic time-history analysis.

Click on the **Ok** button to proceed to the next step.

2.1.2. Applied loading

On this screen, the accelerogram that will be used for the dynamic analysis is specified. The program assumes that the time and acceleration values are given in a text file, in table format, such as:

Loma Prieta Earthquake 17 OCT 1989

2.000000E-02	-4.534576E-04
4.000000E-02	-8.691271E-04
6.000000E-02	9.069152E-04
8.000000E-02	7.255322E-03
1.000000E-01	7.255322E-03
1.200000E-01	2.569593E-03
1.400000E-01	-8.653483E-03
1.600000E-01	-2.191712E-02
1.800000E-01	-4.394760E-02
2.000000E-01	-4.039552E-02
2.200000E-01	-8.955788E-03
2.400000E-01	-1.900743E-02
2.600000E-01	1.549314E-03
2.800000E-01	-2.191712E-03
3.000000E-01	2.494017E-03
3.200000E-01	-1.012722E-02

. . . .
. . . .
. . . .

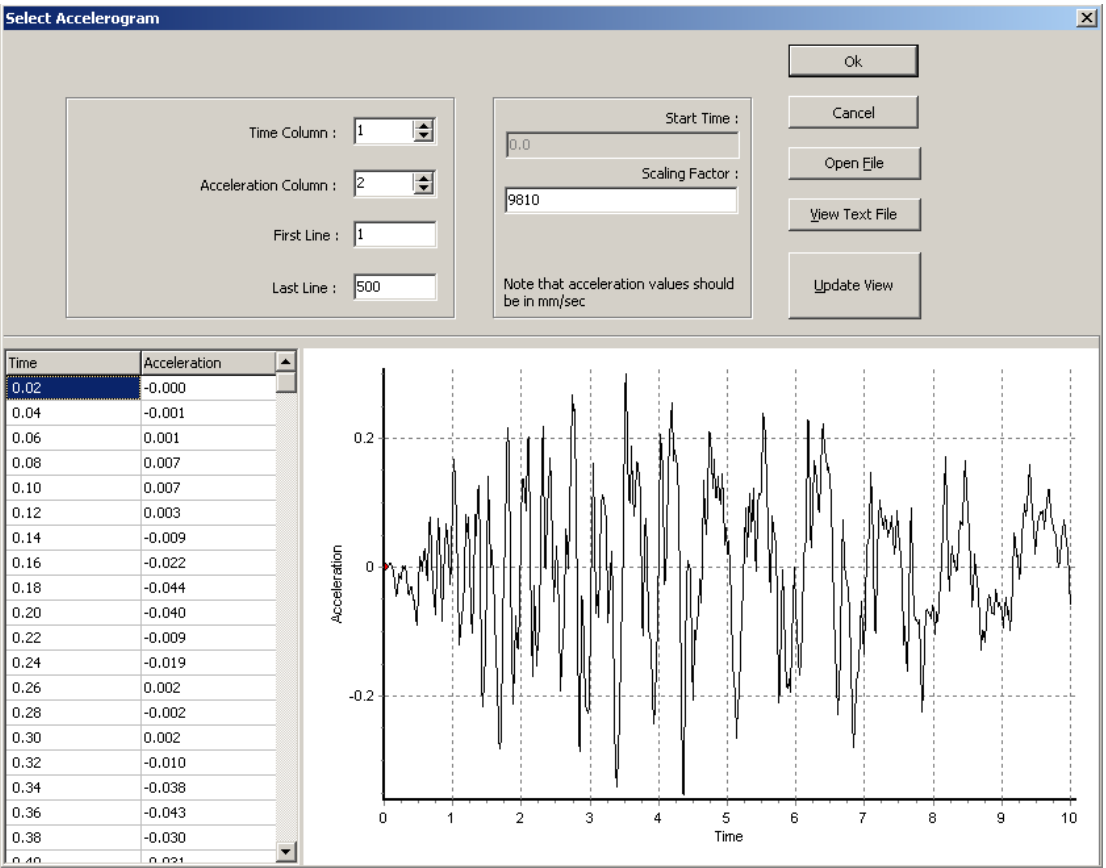


Fig.2 Template screen two (selection of acceleration input).

The user can specify the columns of the time and acceleration values, the first and last lines to be read and the scaling factor. The file can be selected with the **Select File** button, whereas if the user wants to view the contents of a specific file, the **View Text File** button opens a text file reader. The **Update View** button simply updates the input data if one of the parameters (i.e., the last line) has been changed.

There are a couple of important things to note:

- Time values should be in ascending order and larger than zero. Values less or equal to zero are simply neglected by the reader. Moreover, non-numerical input is not accepted.
- If the given acceleration values are in g, then a scaling factor of 9810 is required to change the units to mm/sec^2.
- The accelerogram, by default, will be applied in the x-direction.
- By right clicking on the table or the graph, a very useful pop-up menu appears. The user can copy or print the selected time and acceleration values and the graph to other applications, word-processing (e.g., Microsoft® Word) or spreadsheet (e.g., Microsoft® Excel). Furthermore, the user can change numerous options of the graph (line color or thickness, background, axes values,, etc.) before actually copying or printing. In ZeusNL, almost every table or chart has a pop up menu.

After selecting the input accelerogram, click the **Ok** button. A 3D structural model of a four-story building, which consists of more than 100 elements, has been created. The static (gravity loads) and the dynamic (earthquake) loading have also been applied.

2.1.3. Program modules

Apart from the menu and the toolbars with the buttons that are normally found on any application for the Windows® operating system, in ZeusNL there is a series of pages (modules). On every page, different input data are specified. After the completion of the model, the program focuses on the **Nodes** page and the structure appears. The screen will look like Fig.3.

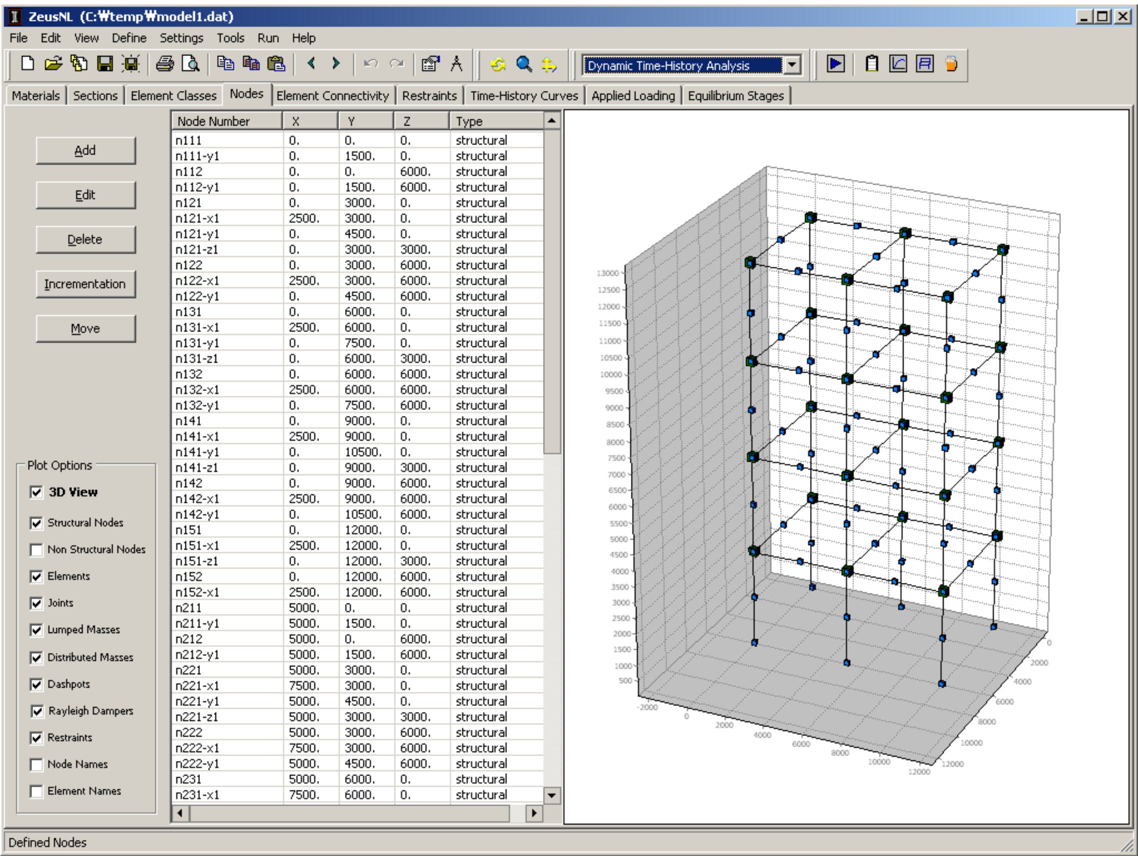


Fig.3 The model together with a list of the created nodes.

Depending on the type of analysis that is running, different modules will appear. For example, in dynamic analysis, there is a page called **Time-History Curves** for the description of the loading (acceleration) curve applied to the supports. Apparently, this module is not needed in pushover (conventional or adaptive) or eigenvalue analysis. In the same way, in pushover analysis there is a page called **Loading Phases** that is not needed for dynamic analysis. For a complete description of the available ZeusNL modules, refer to Section 3. Save the project with **File > Save As**. Note, the input data files of ZeusNL should always have the extension **.dat.**

2.1.3.1. Analysis

Select the type of analysis: dynamic time-history, static time-history, conventional pushover, adaptive pushover, eigenvalue or static with non-variable loading.

2.1.3.2. Materials

In the **Materials** module, the user can specify the different materials available for the current project. These materials are then used to define sections at the **Sections** module. Each material has a material type (stl0, stl1, stl2 for steel, con1, con2, con3, con4 or frp1 for concrete - refer to Appendix A for a detailed description of the material types), and specific material properties, i.e., strength, The Young's modulus, strain-hardening parameter, etc. Each material also has a distinct name with which it is specified in the **Sections** module. Do not confuse the materials defined in this module with the material types available in ZeusNL libraries. In this case, there are two concrete materials, conf and unconfined for the confined and unconfined concrete respectively, with different properties but of the same material type con2. For a comprehensive description of the materials' properties, select one material and click the **Edit** button or simply double-click on the material. A window similar to Fig.4 will appear, where the user can change the properties or even the type or the name of the selected material. Apart from editing the existing materials, it is easy to add a new material by clicking the **Add** button and selecting a name, material type and the corresponding properties. Moreover, the user can remove the selected material(s) with the **Remove** button.

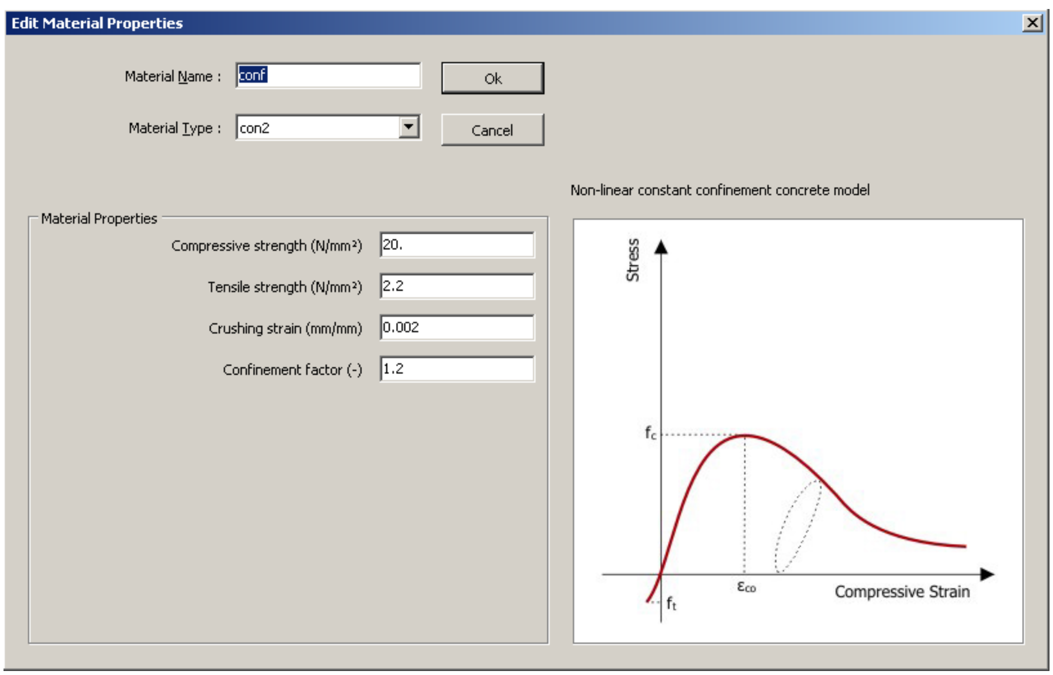


Fig.4 The Material Properties dialog box.

If, by mistake, the user removes one material, well there is an **Undo-Redo** facility in ZeusNL. Simply select the **Edit > Undo** menu command (or the corresponding toolbar button) and the material will be restored. **Edit > Redo** restores the last undone action.

Note, there are some limitations in the names that ZeusNL uses for materials (and sections, element classes, nodes and elements).

> In ZeusNL, material, section, element class, node and element names may be up to eight characters long. Moreover, they should not contain spaces or the characters # or &.

Also, note that cut materials can be copied and pasted (**Edit > Copy and Edit > Paste** or using the pop-up menus by right-clicking on the table). When a material (meaning the entries for the description of a material in ZeusNL) is pasted on the materials table, if the name is the same with that of an existing material, a star '*' is added at the end of the pasted material name. The material properties can be copied and pasted to and from other programs, such as Microsoft® Excel, as long as the entries are consistent with ZeusNL format.

2.1.3.3. Sections

In this module, the different sections of the model are specified. There are 14 available section types, including steel, RC and composite. For a complete description of the sections, refer to Appendix B. Each section is described by a set of sectional dimensions and if it is an RC section by the area and location of the reinforcing bars. Like in the **Materials** module, each section has a unique name and can be copied, pasted and edited. In the example, ZeusNL has created two RC sections: one rectangular called 'scol' for the columns (rcrs type) and one T-shaped called 'sbeam' for the beams (rcts type). For this example, the program assumes that all the columns/beams have identical sections. If the user edits one of the sections, for example, 'sbeam' (again with the **Edit** button or by double-clicking), a **Section Properties** dialog box will appear, which is similar to Fig.5. In this dialog box, the section's name, type, materials, dimensions and reinforcement may be modified.

Depending on the selected section type different numbers of materials (one for steel sections, three to four for RC and composite sections) and dimensions (one to nine) are specified. The materials available are those defined in the **Materials** module (reinf, conf and unc). The program has selected reinf for the reinforcement, conf for the confined region and unc for the unconfined region. There is a description of the dimensions needed, but also notice that whenever the user focuses on a dimension textbox, a red line is drawn on the sample section picture that explicitly shows the edited dimension.

For RC sections, the area and the location of the reinforcing bars has to be defined. Adding, removing and editing bars is easy with the corresponding buttons and is done in the same way materials were added, edited and removed. Note, all the bars have to be within the confined concrete region. Moreover, since the sections are symmetrical, only the bars of the positive 1-3 quadrant have to be specified for the rectangular section and only the bars in the positive (1) side for the T-section. The program generates the rest of the bars. Finally, if the user clicks the **Ok** button, the reinforcing bars are arranged on the **Section Reinforcement** table entry in trinities of (A_s, d_3, d_1).

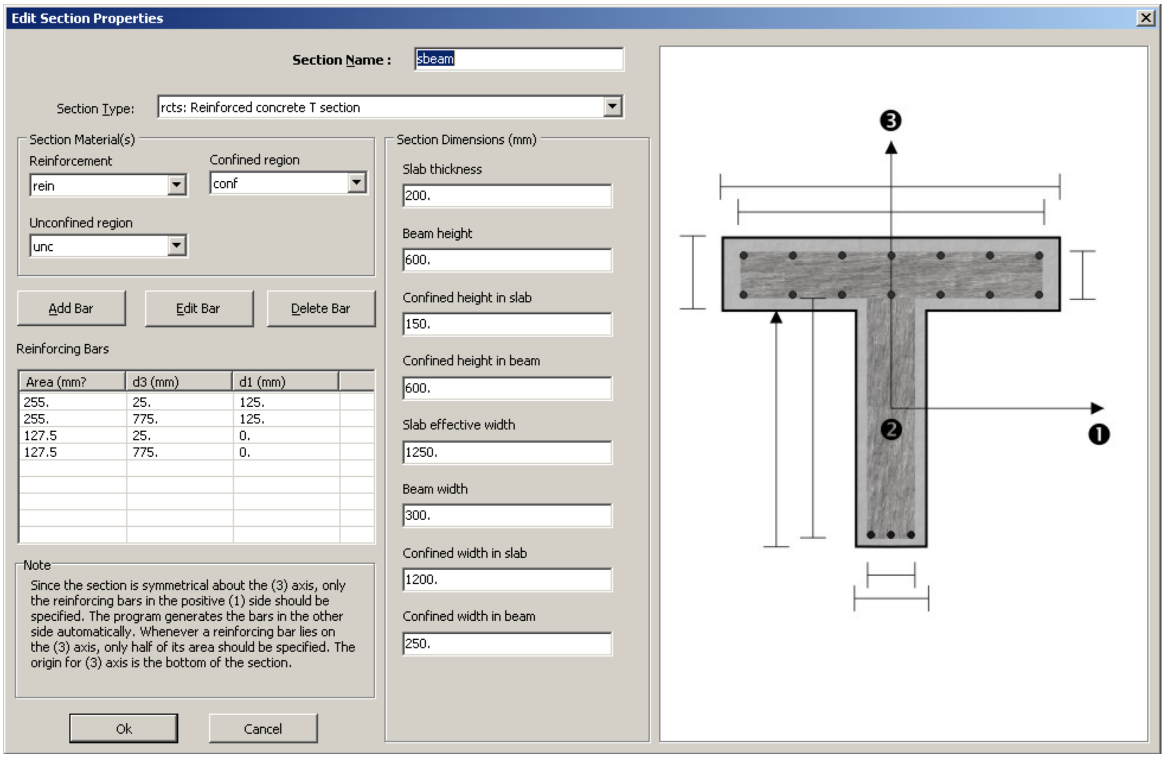


Fig.5 The Section Properties dialog box.

2.1.3.4. Element Classes

The ZeusNL element library contains a set of elements used to model structural elements (beams and columns), non-structural elements (mass and damping) and boundary conditions (supports and joints):

- **Cubic.** Cubic elasto-plastic 3D beam-column element. It is used for detailed inelastic modeling, making use of the uniaxial inelastic material models described above. It accounts for the spread of inelasticity along the member length and across the section depth.
- **Joint.** 3D joint element with uncoupled axial, shear and moment actions.
- **Lmass.** Lumped (concentrated) mass element used in dynamic and eigenvalue analysis.
- **Dmass.** Cubic distributed mass element.
- **Ddamp.** Dashpot (concentrated) viscous damping element used in dynamic analysis.
- **Rdamp.** Element that models Rayleigh damping for dynamic analysis.

These element types are used to define element classes. An element class is a group of properties referring to a particular element category. The element types (different types of elements available in the ZeusNL libraries) should not be confused with the element classes. In each ZeusNL project, there may be many different element classes of the same element type. For example, in the model, there are two element classes (col and beam) of the cubic element type and three element classes of the lmass element type (mass1, mass2, mass4). The element classes defined here are used in the **Element Connectivity** module to define the connectivity of the elements in the mesh configuration. A complete description of the element types of ZeusNL is found in Appendix C.

2.1.3.5. Nodes

After defining the element classes, the user needs to address the mesh configuration of the model. In the **Nodes** module, apart from a list of the nodes, there is a 3D plot of the structure allowing for the better visualization and understanding of the model.

Most of the nodes are structural, although there are some non-structural ones. The question arises: What is the difference between a structural and a non-structural node? For some element types (cubic, joint, dmass and rdamp), extra nodes, apart from the end-nodes, should be specified. The extra nodes define the orientation of the local axes of the elements. In particular, cubic, dmass and rdamp require a third node to define the local (1)-axis and joint requires a third and fourth node to define the (2) and (1)-local axes. For a comprehensive explanation of the use of non-structural nodes as extra nodes for the definition of elements, refer to Appendix E.

By default, the non-structural nodes are not visible on the plot. To make them visible, simply check non structural nodes plot option in the **Nodes** module. Alternatively select **Settings > 3D Plot Settings**. This opens the dialog box, from which the user can change the display settings of the 3D plot of the model. Go to the Non-Structural Nodes tab and click on the **Visible** checkbox, as in Fig.6. The non-structural nodes appear in pink all around the model.

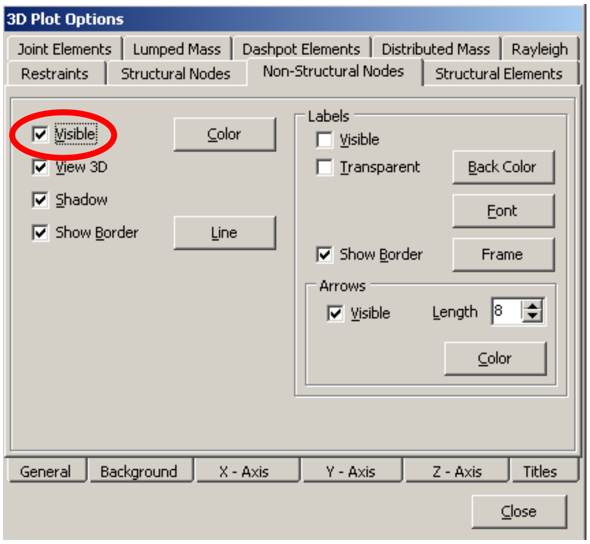


Fig.6 3D Plot Options dialog box. Non-Structural Nodes.

Try to navigate around the 3D Plot Options tabs. There are numerous settings for the appearance of the model. The user can change the color, thickness and style of the lines, size of the nodes and mass cubes, insert titles and footers, change the scale and show or hide the axes and the walls around the model. (The details of these settings are beyond the scope of this tutorial.) The only options that deserve a special attention are those on the General tab. The 3D properties change the rotation, location, zoom and perspective of the plot. Whereas the user may wish to uncheck the **Automatic 3D Plot Update** checkbox in cases where there are really large structures (hundreds or thousands of nodes and elements) and it takes several seconds for the program to update the view. The view will update automatically every time something is changed. If the option is unchecked, the user can manually update the view from the pop-up menu of the plot. After having a look at the options, close the dialog box. Since, the non-structural nodes are not going to be needed any further, the user may wish to make them invisible again in order to enlarge the actual structure.

The **Nodes** module contains the standard **Add**, **Remove** and **Edit** buttons, but there is also an **Incrementation** button. This activates the **Nodes Incrementation** facility of ZeusNL. The user can select one or more nodes and generate new ones in a repetitive manner.

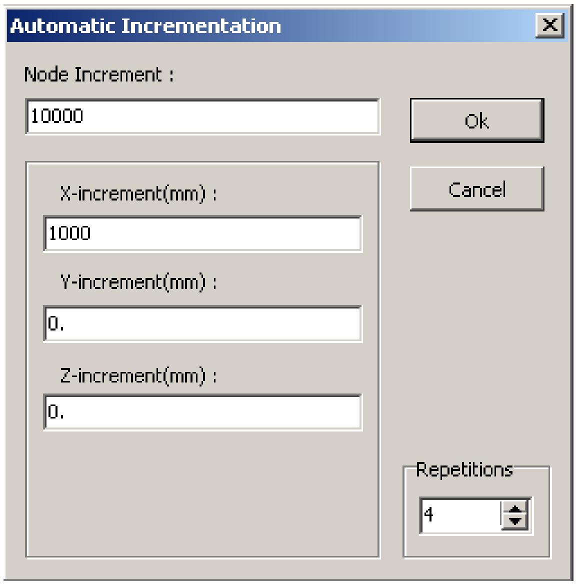


Fig.7 Nodes automatic incrementation facility.

For example, select both the n111 and n112 nodes (to make multiple selections on the ZeusNL tables, click the table items the user wants to be kept, holding the Ctrl key down). Then, with the two nodes selected, click the **Incrementation** button. Select a node increment of 10000. This means that the generated nodes will have names n10111, n20111 (from n111) and n10112, n20112 (from n111). Also, select an x-increment of 1000 and y- and z-increments of zero. Choose four repetitions (this means that for each selected node, four new nodes will be generated, that makes 8 [2x4] new nodes). Click **Ok** and see the new nodes on the 3D plot. It should look something like Fig.8.

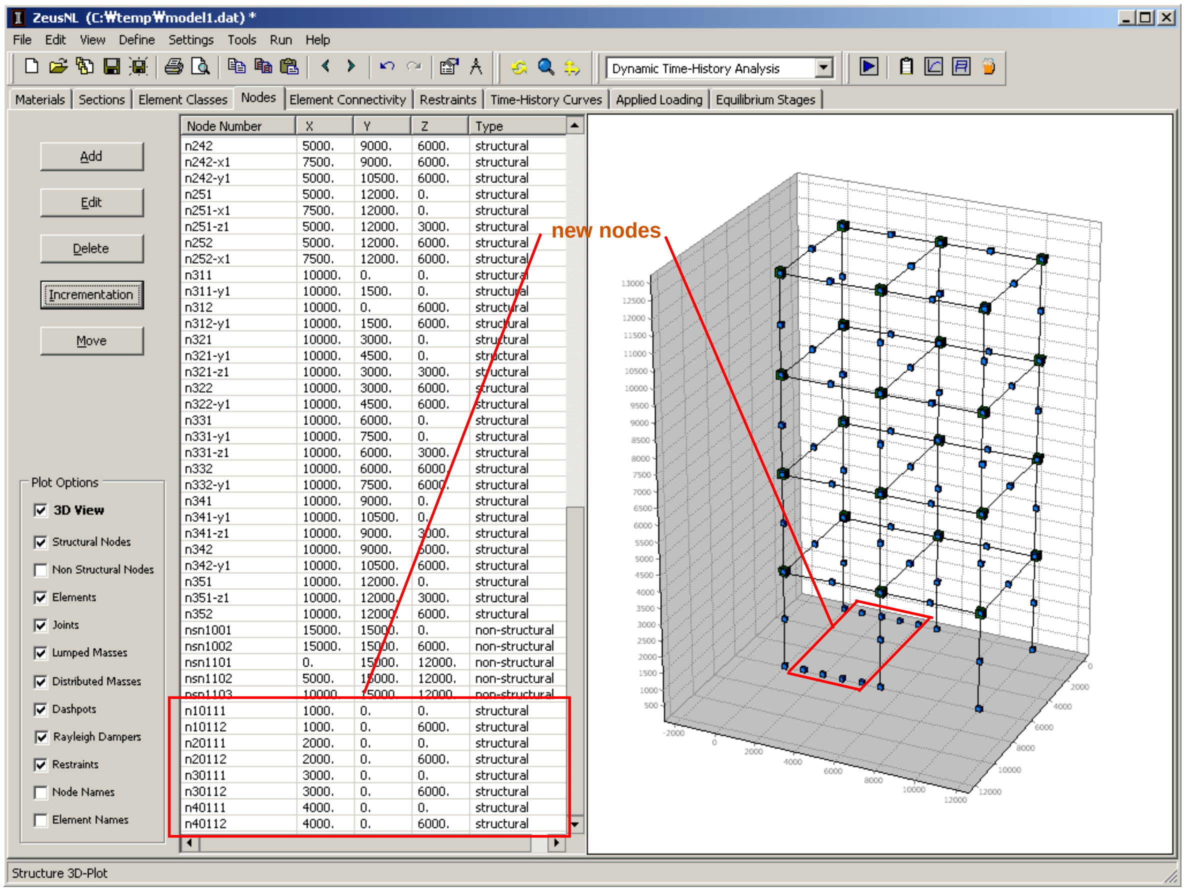


Fig.8 Nodes generated with the automatic incrementation facility.

The **Automatic Incrementation** facility for the **Nodes**, **Element Connectivity** and **Applied Loading** modules is a very powerful ZeusNL tool that enables the user to generate a structure easily and efficiently within minutes. However, there is one restriction with the identifiers of the nodes or elements that can be incremented: they have to be in the format (word)+(number), e.g., n111 and nod20; or the word can be omitted and only numbers may be used for identifiers, e.g., 22 and 44. If the user tries to increment the n111-y1 node, there will be an error message indicating that the node cannot me incremented, since it's not on the correct format.

Node and element identifiers have to be in the correct (word)+(number) [or simply (number)] format to be incremented.

For the time being, since the extra nodes aren't needed, either remove them or **Edit > Undo** the last action to return to the previous state, otherwise the program will not run with some nodes unconnected to the structure.

There are some 3D plot options on the left of the screen. These are the options most frequently used. There is also a **Reset** button that returns the plot to the default state and an **Animate** checkbox that enables the user to see the model from different perspectives. The user can also rotate the model if the plot is dragged with the mouse.

One very interesting feature is the ability to sort the nodes by their number or their x-, y- or z- coordinates. The user has to click on the header of the node number or the coordinate columns (x, y or z) to sort them in an ascending way. If the clicked again, the node will be sorted in a descending way.

2.1.3.6. Element Connectivity

The different elements of the structure are defined here. Each element belongs to a specific element class and, depending on its element type, may have one (lmass, ddamp), three (cubic, dmass, rdamp) or four nodes (joint). The model has only cubic structural elements and lmass mass elements. A complete description of all the element types of ZeusNL libraries can be found in Appendix C. As mentioned in the previous section, the cubic elements are defined, apart from their two end nodes, with a third node that can be structural or non-structural. A more comprehensive explanation can be found in Appendix E. For example, the end nodes of the first column element 'col1111', are 'n111' and 'n111-y1'. They accurately define the geometry of the element, but what about its section and its orientation? The section is 'scol' defined in the **Sections** module. It's clear that the (2) local axis of the section coincides with the (y) global axis. But what about the other two axes? Which global axis (x- or z-) coincides with the (1) local axis of the section? This is why a third node is required to accurately define the element. The element-end nodes, together with the third node nsn1001, define a plane in the 3D space. The section's strong axis [i.e., axis (3)] lies on that plane and, for the model, coincides with the (x) global axis.

Like the other modules, the user can add, remove and edit the selected elements. However, there is also an **Incrementation** and a **Subdivision** facility. The automatic incrementation of the elements works more or less in the same way as the incrementation of the nodes. However, here the increment of the element number together with the increments of the node numbers, have to be specified. The subdivision can only be applied to linear elements and permits the fast and easy subdivision of each element in two by creating a new node at the middle of it. The user may want to subdivide an element in the critical areas of the structure, in order to increase the accuracy of the analysis.

One feature that the user will find very useful, is the ability to change the element class of a large number of elements in one step by making a multiple selection and clicking **Edit**. For example, this is very useful when the user wants to the change the beams element of one story from one element class to another.

2.1.3.7. Restraints

The user can easily specify the restrained nodes by selecting them and clicking the **Edit** button. The entire process is straightforward however, please note something very important about restraints:

> In dynamic analysis, the restrained DOF at the supports, in the direction of the earthquake, must be released.

That is why the restrained DOF of the supports in the model are y+z+rx+ry+rz, but not x (x is the direction of the earthquake).

2.1.3.8. Time-history curves

This module specifies piecewise linear curves for dynamic (or time-history) analysis. One has been created with the template by selecting the record. The defined curve will be applied with certain rules to the structure at the next module, **Applied Loading**.

There is a box at the left of the main window with the **Start Time** of the analysis. This is the time when the analysis starts (zero for the model).

If the user double-clicks on the curve, the **Edit Curve** dialog box opens. Here the applied curve can be change, i.e., its duration or even the selected file. For every curve, there is a delay parameter. The delay (which should always be positive) is the time after the start time that the curve is applied to the structure. In this way, it is very easy to simulate asynchronous excitation by specifying the same curve with different delays.

2.1.3.9. Applied Loading

This module specifies the applied loads. There are four different types of loads:

- **Initial.** These are static loads that are applied prior to any variable load. They can be forces or prescribed displacements applied at nodes.
- **Proportional.** These are static loads that have proportional variation. The magnitude of a load at any step is given by the product of its nominal value (which is constant) and the current load factor (which varies). Proportional loads may be forces or prescribed displacements applied at nodes.
- **Time-history.** These are static loads varying according to different load curves in the pseudo-time domain. The magnitude of a load at any time-step is given by the product of its nominal value (which is constant) and the variable load factor obtained from its load curve. Time history loads may be forces or prescribed displacements applied at nodes.
- **Dynamic.** These are accelerations or forces varying according to different load curves in the real time domain. The product of its constant nominal value and the variable load factor obtained from its load curve at that time gives the magnitude of the load. Dynamic loads can be forces or accelerations applied at the nodes in the global directions.

For this dynamic analysis, initial loads (applied at the structural masses) and dynamic loads (applied at the supports in the x-direction) will be used. Again, the user can add, remove, edit or increment the loads. Note, for this module the user can add the same load (value, direction, type) to more than one nodes at a single step, specifying many nodes at the **Add Load** dialog box (use the Ctrl key for multiple selections).

2.1.3.10. Equilibrium Stages

This module specifies the stages of time intervals at which structural equilibrium is established. In other words, the user specifies the time steps at which the forces and displacements of the structure are equilibrated.

The user can have different stages with different time-steps, depending on the difficulties in convergence that may arise at different times of the dynamic analysis (for more demanding analyses a smaller time-step is required). This is done by specifying the end–time of the stage (that should be larger than the end-time of the previous step) and the number of steps.

The program calculates the time step value for each stage. It is equal to the difference between the end time of the current stage and that of the previous stage, divided by the number of steps of the current stage. For the first stage, the time-step is equal to the difference between the end-time and the **Start Time** defined at the **Time-History Curves** module divided by the number of steps.

2.1.4. Running the analysis

After reviewing the different modules, it is time to run the analysis. Save the project (**File > Save**) and select **Run > Run** or the corresponding toolbar button. After performing some integrity checks, the program starts running. Depending on the size of the structure, the applied loads and the PC processor, the analysis may last for up to several hours. This is significantly higher than the time required by other similar FE packages and it is attributed to the way the spread of inelasticity along member length and across section depth is modeled. However, the results have increased stability and significantly better accuracy.

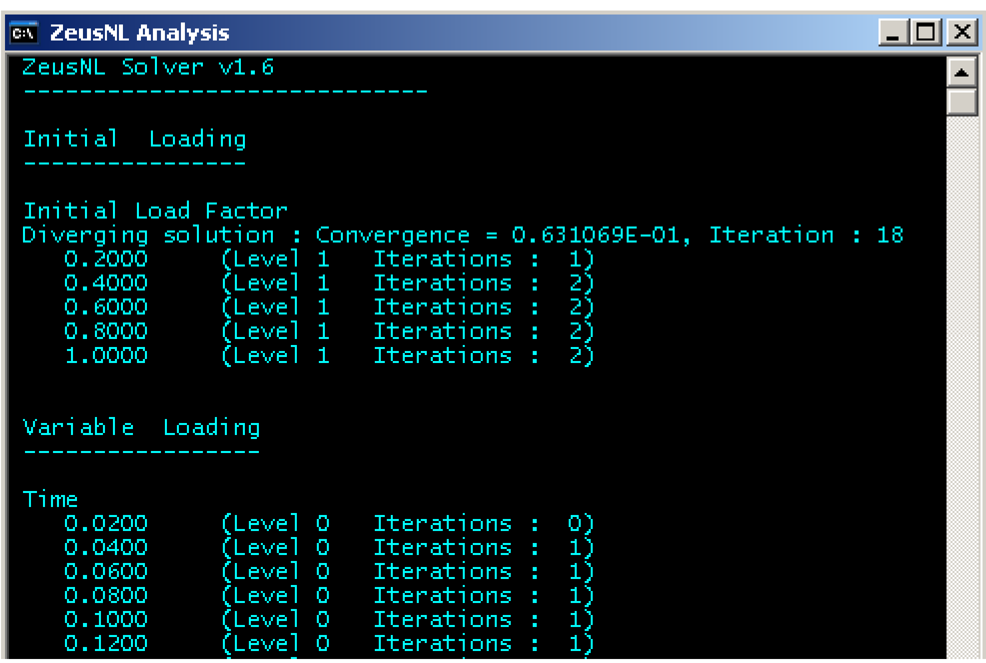


Fig.9 ZeusNL running dynamic analysis.

2.1.5. Getting the results

ZeusNL has two different post-processing facilities that supplement each other. The Post-Processor, which creates curves with the action effects of the analysis (displacement, forces, stresses, etc.) and the deformed shape viewer, with which the deformed structural shape can be viewed at the different time-steps of the analysis. Both open the basic results file type of ZeusNL (**.num** extension), read them and display the data. The user can run them from the corresponding commands of the Tools menu.

2.1.5.1. Post-processor

The **Post-Processor** is a facility used to easily derive diagrams with the results of the analysis.

For example, the user is trying to plot the interstory drift between the top left node and the bottom left node of the first frame (n151 and n111 respectively) vs. time. Run the Post-Processor with **Tool > Post-Processor** and open the project **.num** file.

Select from the drop-down lists on the top-left of the window: **Time** for the x-axis and **Interstory Drift** for the y-axis. On the dialog box that opens, choose the nodes n151 and n111, select the appropriate direction (U_x that is the direction of the earthquake) and click **Ok**.

> In many cases, the user will have to find an item (e.g., a node) in a drop-down list with hundreds of items. If the name of the item (node) is known, start typing it when the drop-down list is highlighted and ZeusNL will locate it.

Click on the **View Curve** button and the plot is created. The values of the plot are shown on the table at the bottom-left corner.

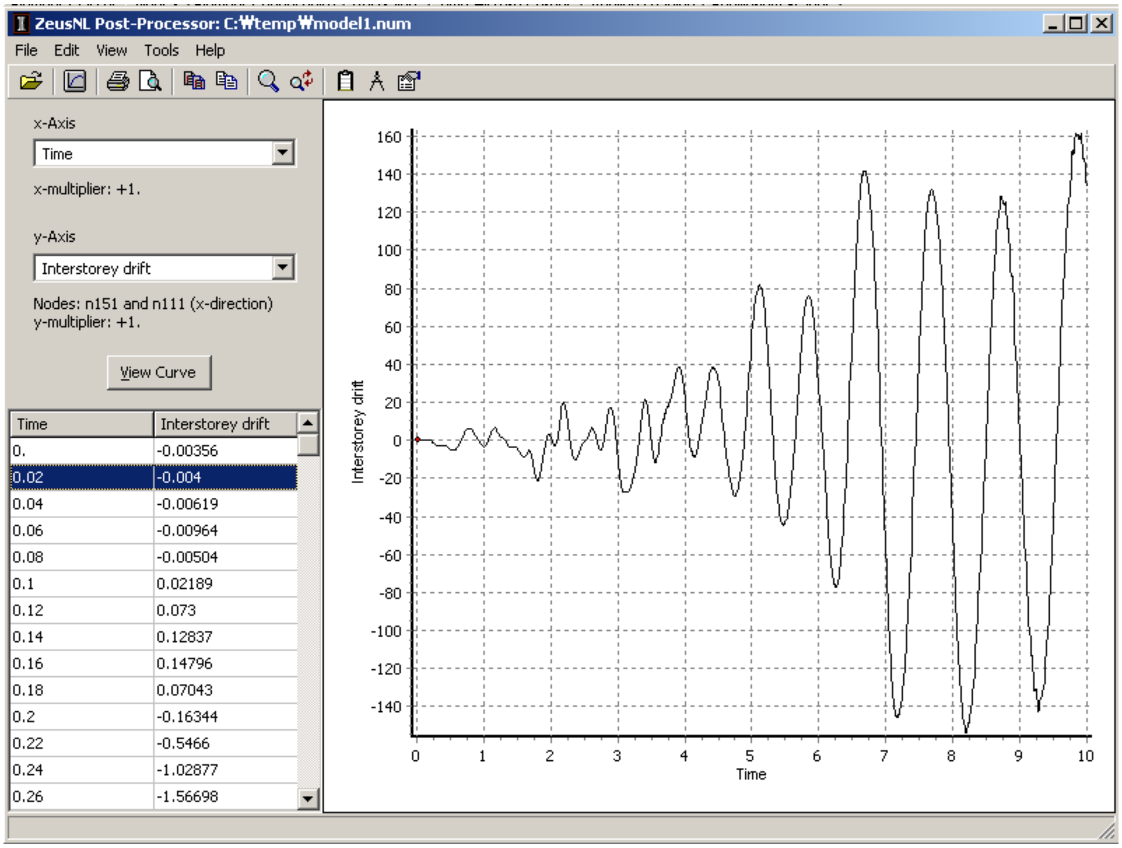


Fig.10 ZeusNL Post-Processor.

To convert the displacement values from mm to cm, multiply all of the y-values of the plot by 0.1. Select **Tools > Settings** and change the y-multiplier to 0.1. The diagram is then re-plotted with all the y- values multiplied with 1/10. Note, the user can choose any value (positive, negative or even zero) for the x- and y- multipliers.

With the Post-Processor, there are several diagrams that the user can easily create including nodal displacement and rotations, interstory drifts, support reactions, element shear forces, etc.

The diagram and values of the tables can be copied to other applications, such as Microsoft® Word and Microsoft® Excel. There are many options (**Tools > Graph Options**) for the diagram and the user can change almost everything in its appearance, including the background, color, thickness and style of the line, axes labels and ticks, etc. Note, the user can zoom in and out with the menu commands or the corresponding toolbar buttons. To zoom in on a specific area, select it by moving the mouse pointer from top-left to bottom-right. By making a selection, moving the mouse to the opposite direction (from bottom-right to top-left), the diagram zooms out to the initial state.

To remove the gradient of the graph, go to the **Panel** tab of the **Options** window and uncheck the **Gradient-Visible** checkbox. The result is a plain graph with a white background ready to copy to a word-processing application.

2.1.5.2. Deformed Shape Viewer

With the Post-Processor, the user can quickly and easily create selected diagrams from an analysis and copy or print them. However, questions arise. For example: What happens when the user actually wants to see how the structure looks like at 6.02sec, when the largest top displacement of node n151 occurs? Is there an easy way to identify soft-stories? What will the second eigenmode of the structure would look like?

The Deformed Shape Viewer answers these questions. Close the Post Processor (this takes the user to the **Main Program** window) and open **Tools > Deformed Shape Viewer**.

From the shape viewer, open the **.num** file that has been created with the analysis. After the file has been loaded, a list of the time-steps of the analysis appears on screen. Select time step 6.68 sec and click the **View Shape** button. In a couple of seconds, the deformed shape appears and it is ready to be copied or printed.

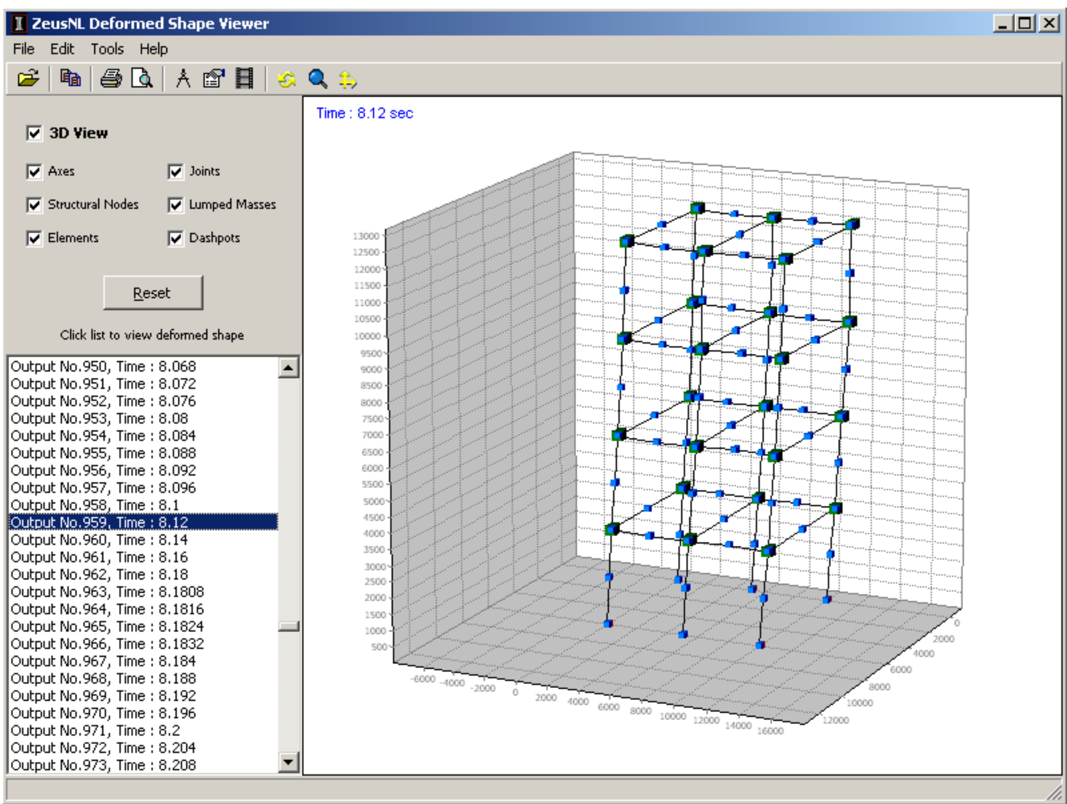


Fig.11 Deformed Shape Viewer (dynamic analysis).

The options on the top-left corner of the window are already familiar, as well as the **Tools > 3D Plot Options** dialog box. They allow the user to quickly change the appearance of the plot before printing or copying it. One interesting parameter is that the Deformation Multiplier (**Tools > Settings**), is the parameter by which the nodal deformations are multiplied before the plot is derived.

2.2. Tutorial 2 – Eigenvalue analysis

The dynamic analysis has been run with the four-story structure, but in order to obtain the dynamic characteristics of it, i.e., eigenperiods and mode shapes, an eigenvalue analysis needs to be run.

There are two possible ways of doing this. The more vigorous would be to alter the existing model for the needs of eigenvalue analysis. However, the potential problems that could arise, when changing from one analysis type to another, will be dealt with in Section 3. The second way, is to create a new identical structural model from the template.

Select **File > Create from Template** and specify the same structural characteristics with the previous example, but now choose eigenvalue rather than dynamic analysis.

There are a couple of changes in this current model, in comparison with the first (dynamic) analysis model:

- There are some modules missing: **Time-History Curves**, **Loads** and **Equilibrium Stages**. All of these are related to the applied loading and therefore, are not needed for eigenvalue analysis.
- The x-DOF of the supports that has been released for the dynamic analysis is now restrained.

Apart form these differences, the models should be identical. Save the project and run it.

After the analysis has been completed (because its eigenvalue analysis, it should not last more than a few seconds), open the deformed shape viewer and open the analysis' **.num** file. A list of the converged eigenmodes will appear. Select each one of them and click the **View Shape** button to see the mode shapes (Fig.12). Again, the user can easily copy, print or change the appearance of the created 3D plot.

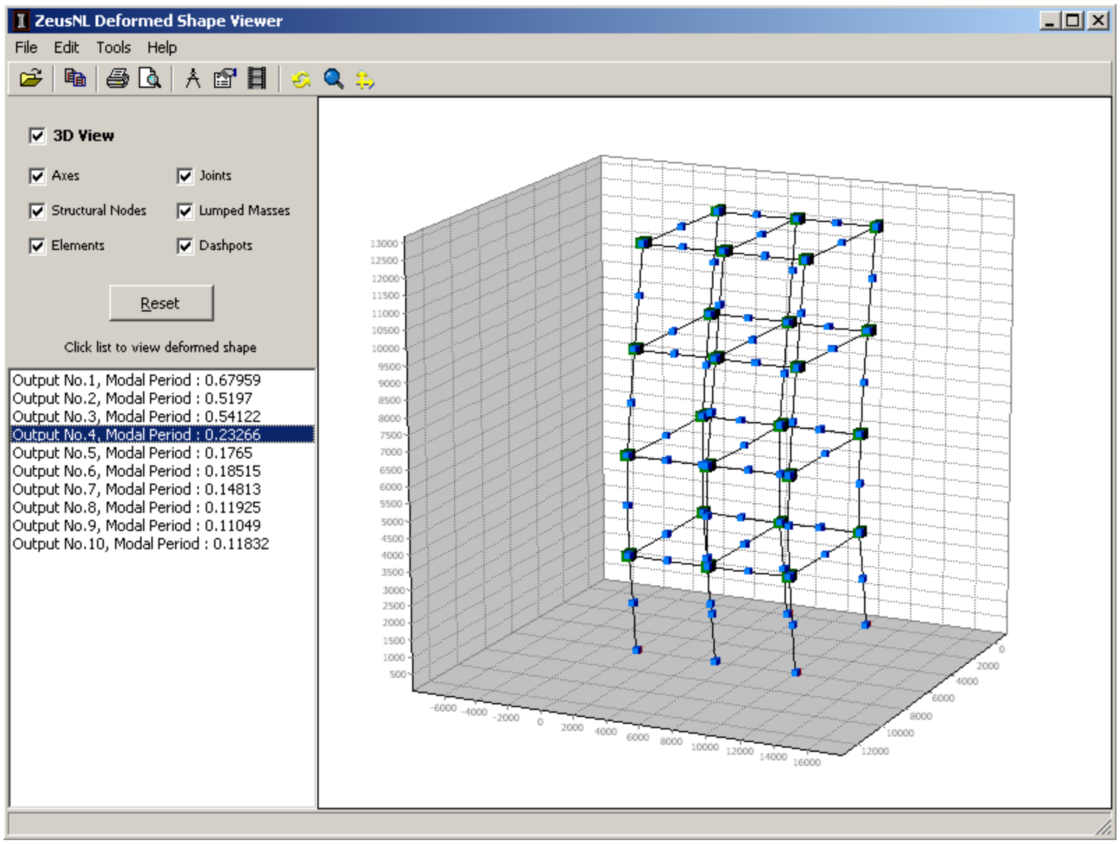


Fig.12 Deformed Shape Viewer (eigenvalue analysis).

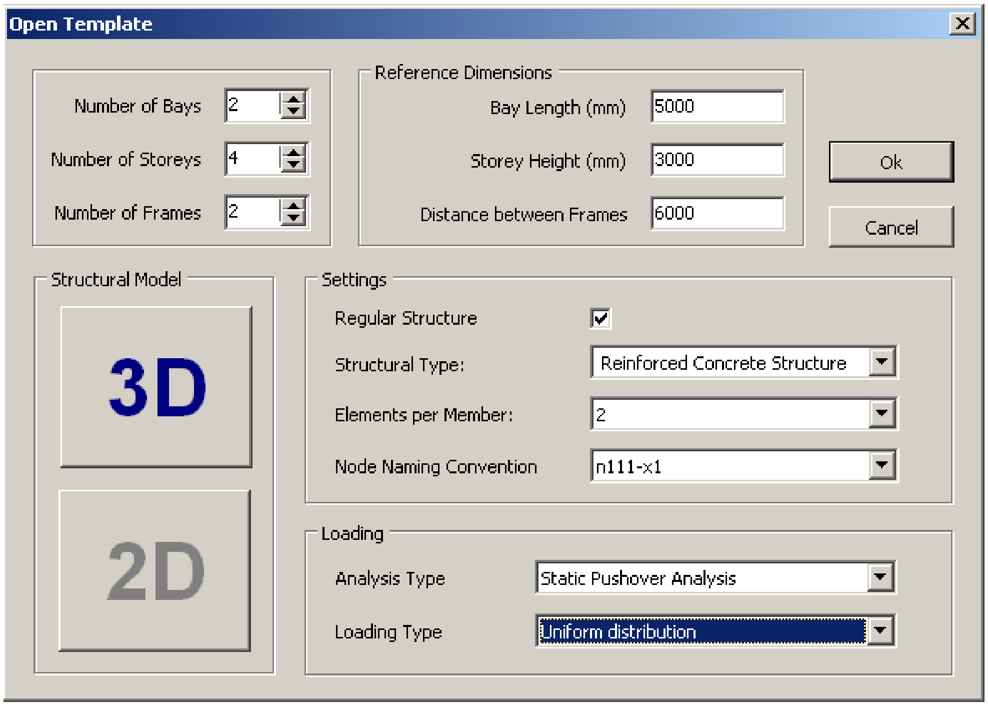


Fig.13 Template screen one (pushover analysis).

Fig.14 Template screen two (pushover analysis).

2.3. Tutorial 3 – Pushover Analysis

The ZeusNL template will be used to create a model of the building for pushover analysis. Selecting **Static Pushover Analysis** as the analysis type, a new drop-down list appears asking the user for the distribution of the proportional loads (uniform or triangular). Select the default, which is **Uniform** (Fig.13). Click **Ok** and a new dialog box appears where the target value of the proportional loads applied at the different stories is specified (Fig.14).

The main differences between the current model and the model previously created for dynamic analysis are:

- There are no masses. Masses are not needed since there are no inertia forces in static pushover analysis (Fig.15).
- The x-DOF is restrained at the supports (Fig.15).
- The **Time-history Curves** and **Equilibrium Stages** modules are missing. There is an extra module instead called **Loading Phases**. This module defines the control phases used to trace the load deflection curve for proportional loading. Three types of control are available:
    - **Load control** refers to the situation where the load factor λ is directly incremented and the global structure displacements are determined at each load factor level. The applied load can be either forces or displacements.
    - **Response control** refers to the situation where the response (displacement/rotation) of a node, specified by the user, is incrementally increased. The loading applied and the deformations of the other nodes are determined by the solution of the program.
    - **Automatic response control** refers to a procedure in which a new DOF is automatically chosen for response control, whenever convergence difficulties arise during the analysis. The chosen node is the one having the highest rate of nominal tangential response.

There are different possible schemes that can be applied. For the time being, the default created by the template will be used: one load control phase and one automatic response control phase. For a complete description of the module, refer to Section 3. For the time being, it's only worth mentioning the target displacement of the automatic control (which is the target at which the analysis stops, if it hasn't stopped before due to divergence problems) and the direction of the controlled freedom (here it is the x-direction).

After the termination of the analysis, like in the two previous tutorials, the results can be processed with the Post-Processor and the Deformed Shape Viewer (Fig.16 and 17).

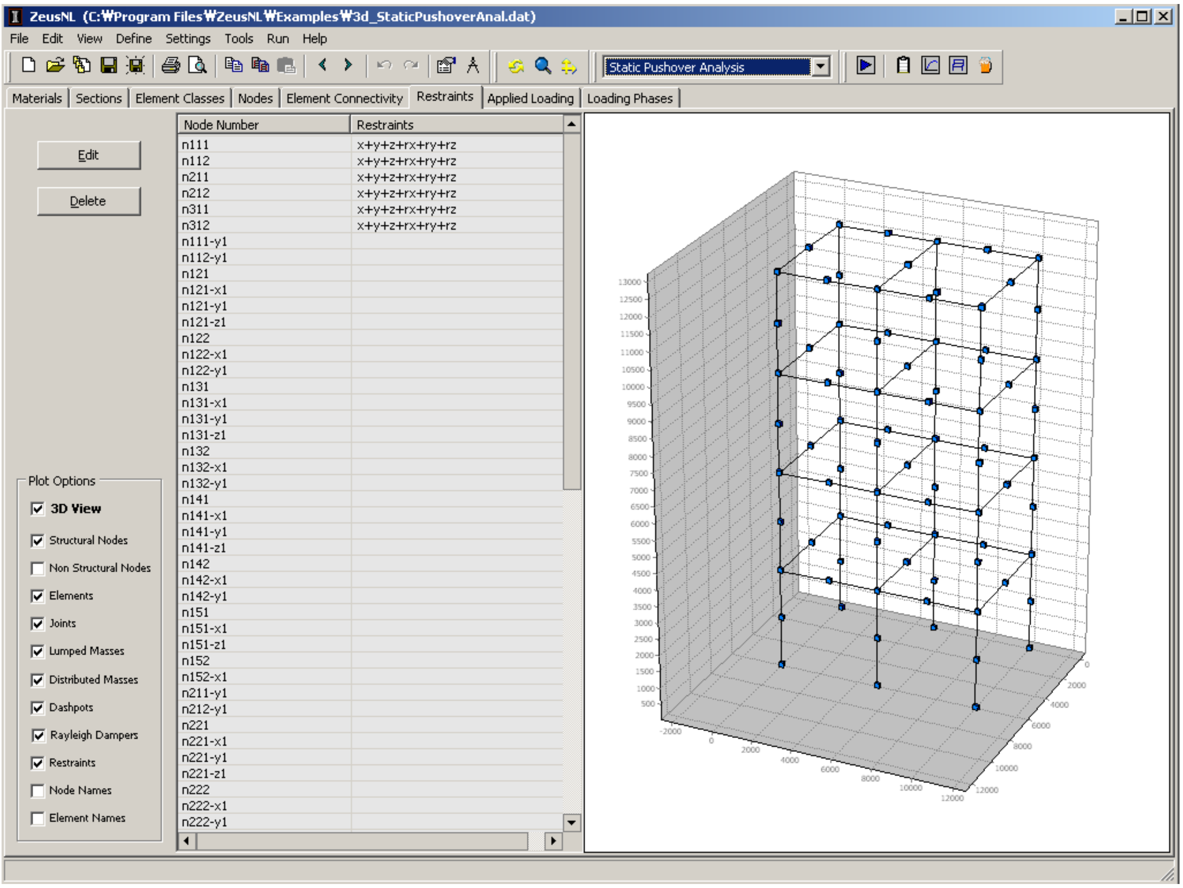


Fig.15 The model created for pushover analysis.

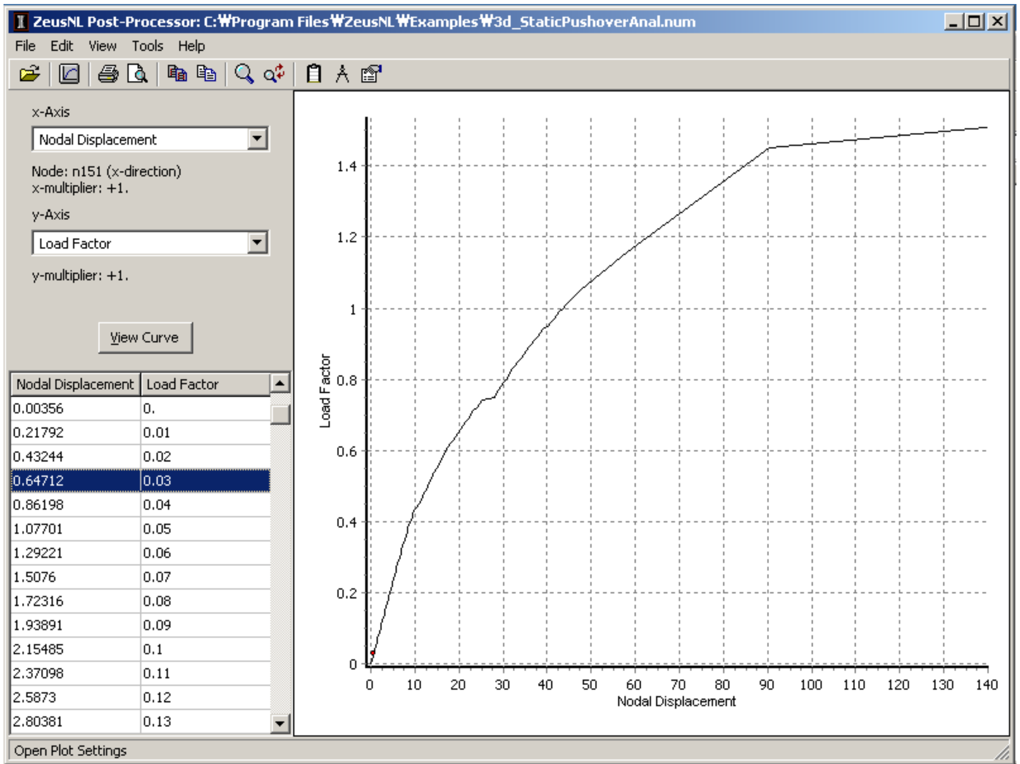


Fig.16 The top displacement vs. load factor pushover curve.

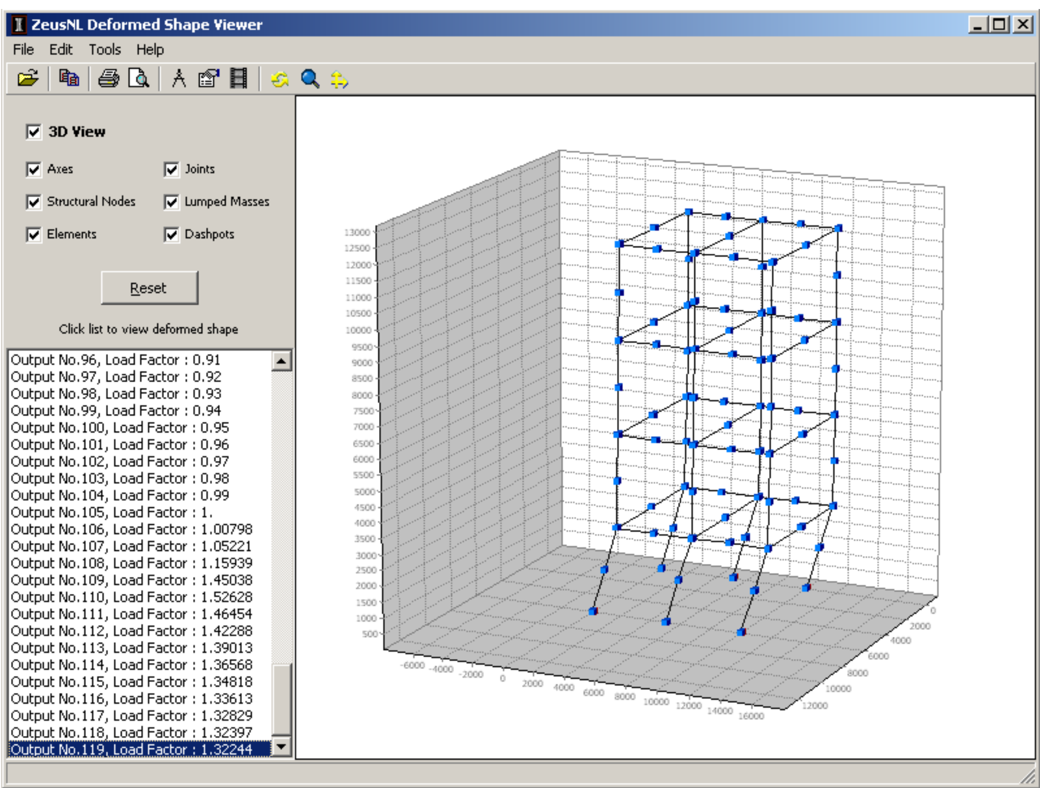


Fig.17 The deformed shape at the last step of the pushover procedure.

2.4. Tutorial 4 – Static Time-History Analysis

Static time-history analysis is usually used for the simulation of experimental tests on specimens. For example, the user can model an imaginary cyclic test on an RC wall. Assume that the loading consists of a constant axial compressive force and a variable displacement applied on the top of the wall (Fig.18), according to a pre-defined pattern.

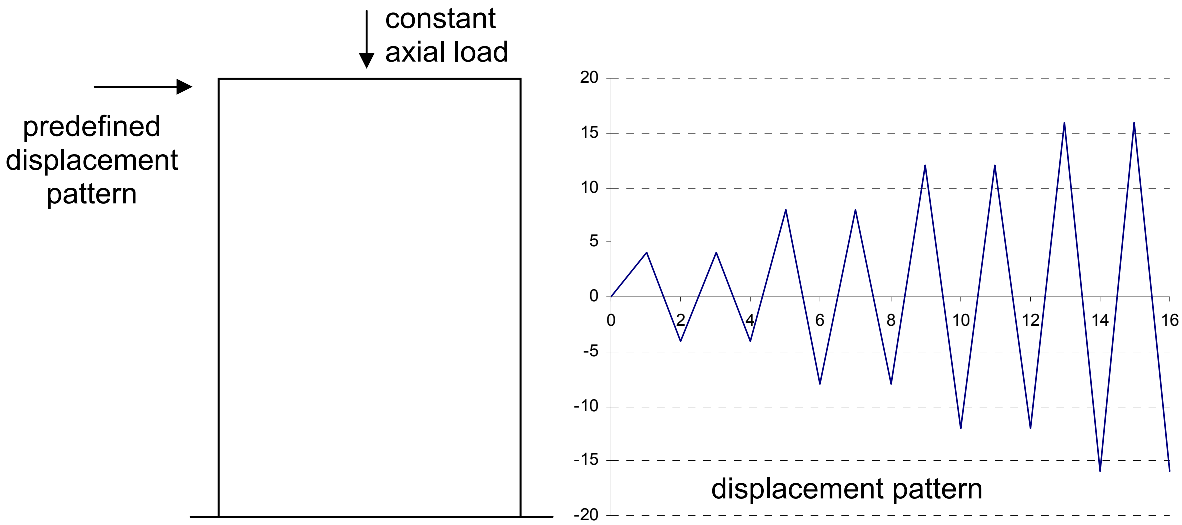


Fig.18 Cyclic test on an RC wall.

This time a template will not be used. The model will be created from scratch, module by module. Even without the **Template** facility, creating a model with ZeusNL is simple.

2.4.1. Structural Configuration

From the **Analysis** module, select **Static Time-History** analysis. In the **Materials** module, the user will have to define four materials for the project, one for the reinforcing bars and three for concrete, since the RC wall section rcfws consists of four different materials (see detailed description in Appendix B).

Use the stl1 material type for the steel material long-rei and the con2 type for the three concrete materials (unconfined: conc1; partially confined: conc2; fully confined: conc3). Note, the parameters of conc1, conc2 and conc3 are exactly the same (compressive and tensile strength, crushing strain), apart from the confinement factor.

Material Name	Material Type	Material Properties
long-rei	stl1	200000. 500. 0.05
conc1	con2	20. 2.2 0.002 1
conc2	con2	20. 2.2 0.002 1.03
conc3	con2	20. 2.2 0.002 4

Fig.19 Materials used in the static time-history analysis.

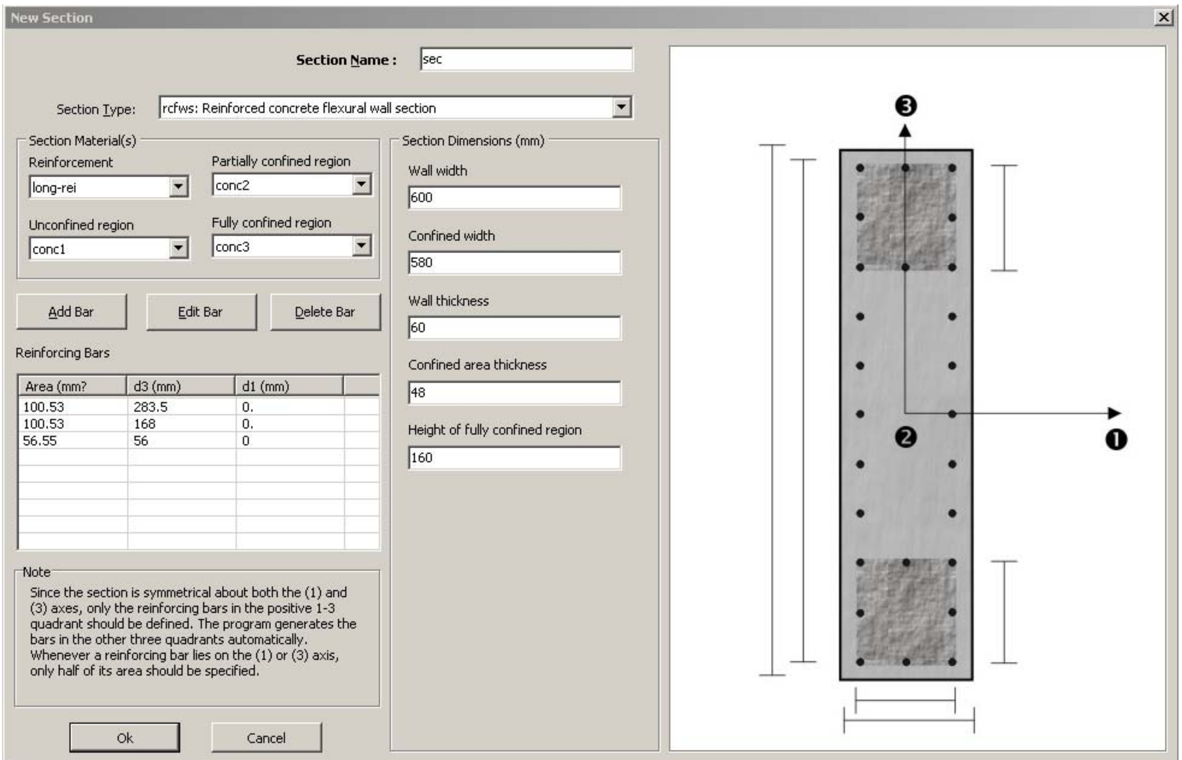


Fig.20 The wall section used in the static time-history analysis.

With these four materials, the user can now define the wall section (Fig.20) using the rcfws type and also define a cubic element class called wall, with the new section and (for example) 250 monitoring points. The model's nodes, elements and restraints are depicted in Fig.21, 22 and 24. For their derivation, it is advisable to use ZeusNL **Incrementation** facility. For example, after defining the first element 1, the user can easily derive the other elements as in Fig.23. Also, note the non-structural node 100 that is used for the definition of elements and that all the nodes are restrained for the out-of-plane deformations (z+rx+ry).

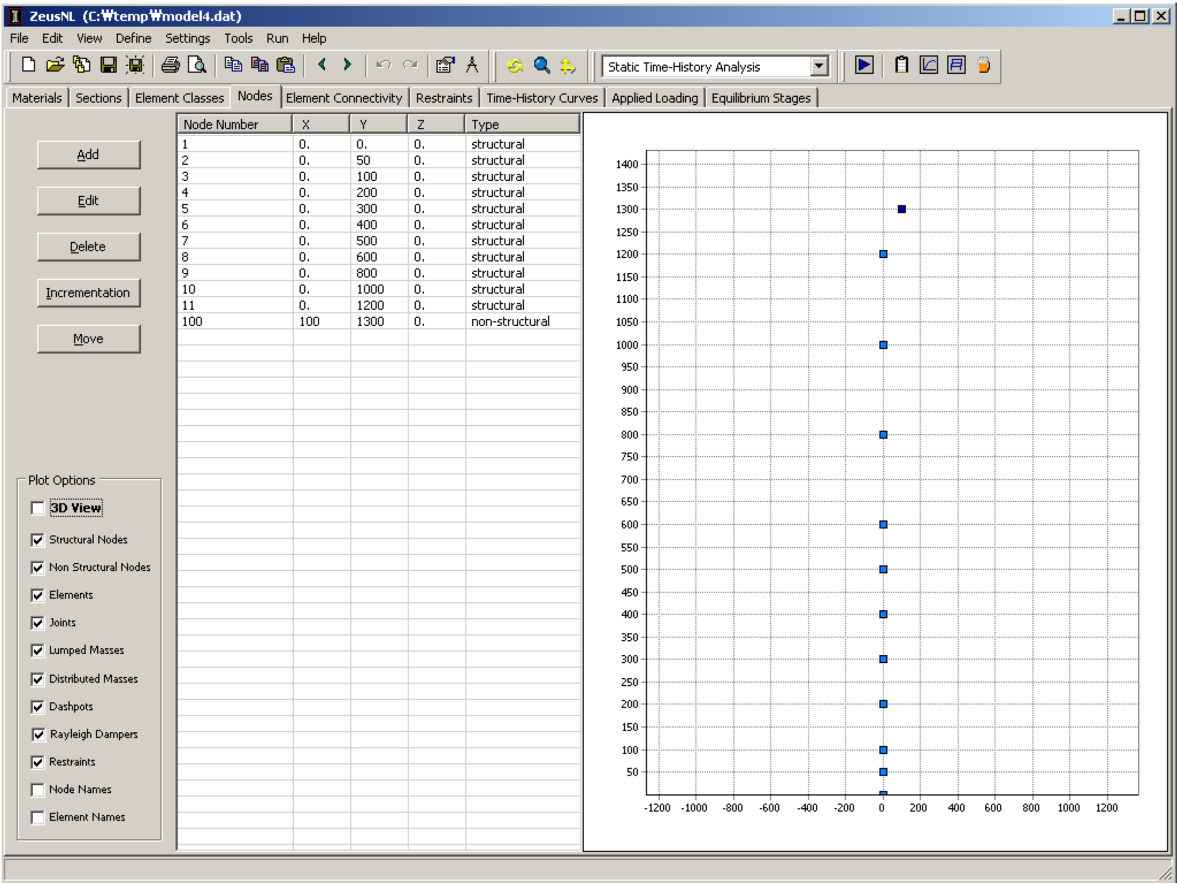
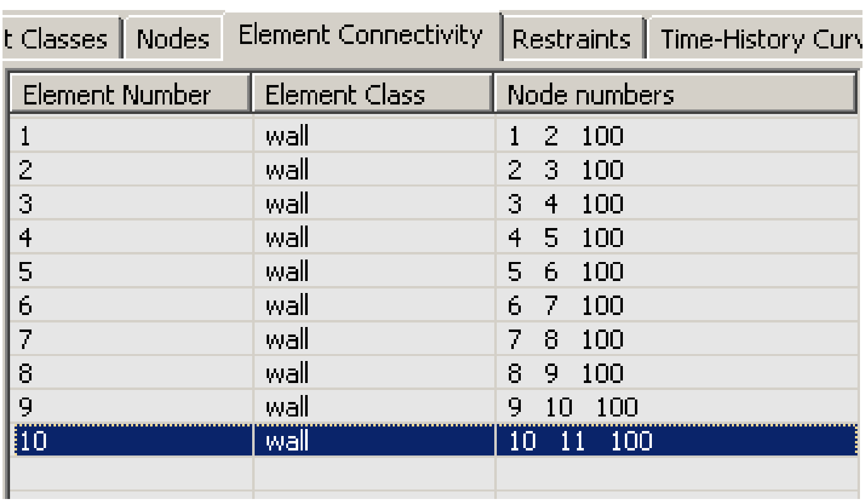


Fig.21 Model nodes.



Element Number	Element Class	Node numbers
1	wall	1 2 100
2	wall	2 3 100
3	wall	3 4 100
4	wall	4 5 100
5	wall	5 6 100
6	wall	6 7 100
7	wall	7 8 100
8	wall	8 9 100
9	wall	9 10 100
10	wall	10 11 100

Fig.22 Model elements.

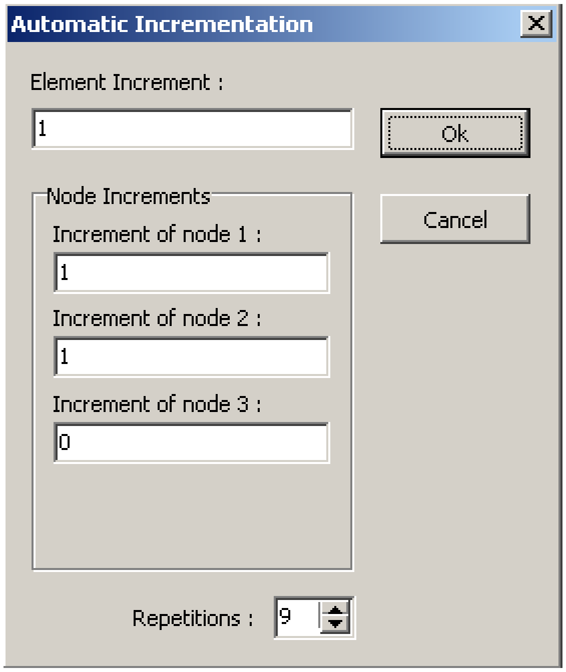
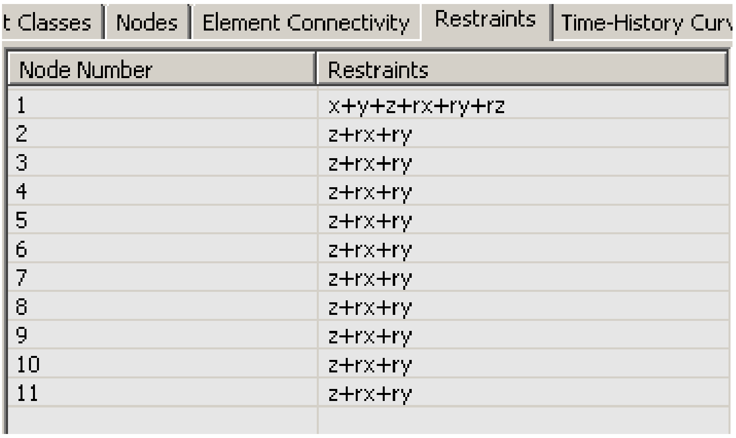


Fig.23 Using automatic incrementation to derive elements 2-10 from element 1.



Node Number	Restraints
1	x+y+z+rx+ry+rz
2	z+rx+ry
3	z+rx+ry
4	z+rx+ry
5	z+rx+ry
6	z+rx+ry
7	z+rx+ry
8	z+rx+ry
9	z+rx+ry
10	z+rx+ry
11	z+rx+ry

Fig.24 Model restraints.

The next step is to determine the load applied to it, which consists of a constant vertical force and a variable horizontal displacement at node 11. First, describe the pattern for the horizontal load. This is done in the **Time-history Curves** module. Select **Create a Curve**, which takes the user to the **New User-Defined Curve** dialog box. Specify the pseudo-time and load factor coordinates on the table, as well as the name of the curve (Fig.25) and click **Ok**.

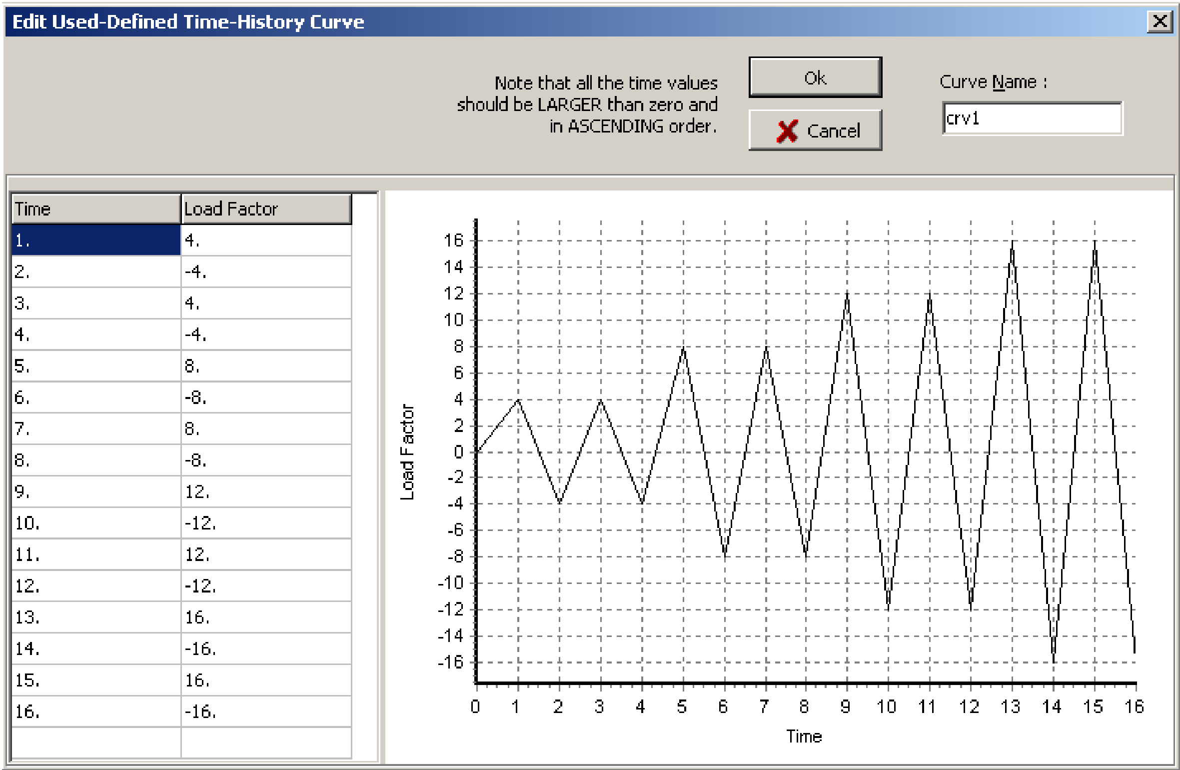
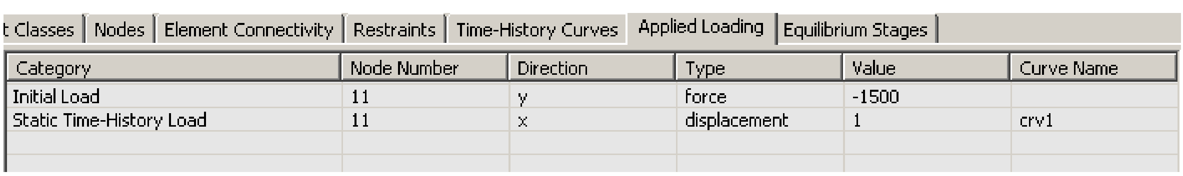


Fig.25 New user-define curve.

The applied loads can be defined in the **Applied Loading** module as an initial load and a static time-history load (Fig.26). The process is straightforward. Note, the vertical is applied downwards and therefore, should be negative and that, in order to apply double or triple the variable horizontal load, the user would only have to change its Value parameter from 1.0 to 2.0 or 3.0, respectively.



Category	Node Number	Direction	Type	Value	Curve Name
Initial Load	11	y	force	-1500	
Static Time-History Load	11	x	displacement	1	crv1

Fig.26 Applied loading.

Finally, the equilibrium stages need to be defined. One stage for the 16sec of pseudo-time of our test will be defined with 320 steps. The time-step is 0.05 sec.

The model is ready to run. Click **Run > Run** and wait for the analysis to finish. After the termination, the hysteretic curve for the wall needs to be plotted. Run the Post-Processor **(Tools > Post-Processor**) and open the project's **.num** file. Choose to plot the x-displacement of node 11 for the x- graph axis and the support moment M_z of node 1 for the y- axis. Click the **View Curve** button to view the resulting hysteretic curve. At this point, it needs to be formatted.

The moment units are currently in Nmm. Go to the **Tools > Settings** dialog box and change the y-multiplier to 1e-6 or 0.000001. Now select **Tools >Graph Options** and from the **Panel** tab uncheck the **Gradient-Visible** checkbox. In the **Axis** tab, go to the **Title** sub-tab for the left and the bottom axes and change the titles to 'Support Moment (kNm)' and 'Top Displacement (mm)', respectively. As a result, there is a graph ready to copy to a word-processing program. It is not necessary to open a spreadsheet for the derivation of diagrams.

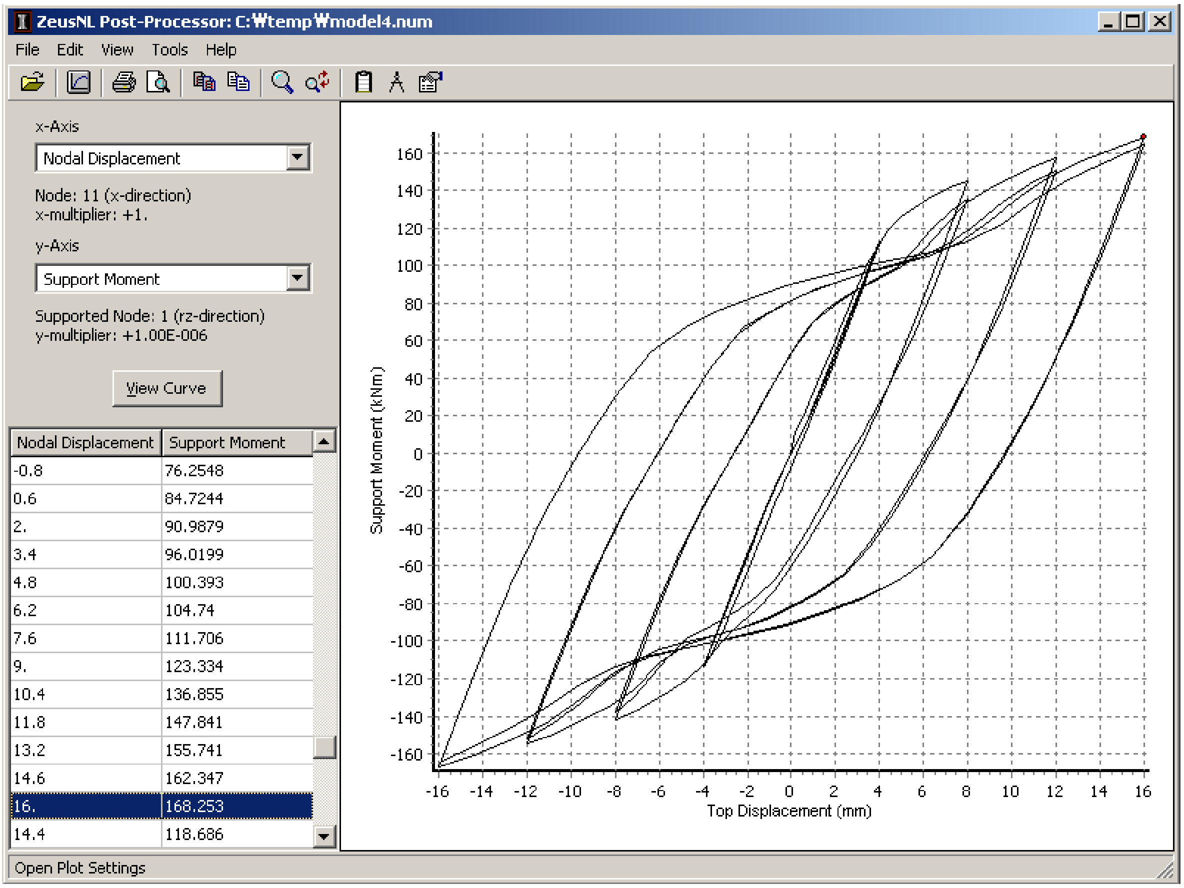


Fig.27 Wall hysteretic curve.

3 RUNNING ZeusNL

This section presents advanced features of ZeusNL that makes the user take advantage of the full potential of the program. It will also help the user build a better understanding of the procedures and the theoretical background behind them.

3.1. Analysis Types

3.1.1. Eigenvalue analysis

The Lanczos algorithm is used for the evaluation of the structural natural frequencies and mode shapes. The number of required modes and a range of frequencies of interest are specified by the user in the program settings (**Tools > Settings**).

3.1.2. Static analysis (constant loading)

The applied load *P* is kept constant. The program performs the solution of the analysis in a single step and outputs the nodal displacements and the support and element forces.

3.1.3. Static Pushover analysis

The applied variable load P is kept proportional to the pattern of nominal loads P^0, initially defined by the user. The load factor λ is automatically increased by the program until a user-defined limit or structural failure is reached:

$$P = \lambda \cdot P^0$$

3.1.4. Adaptive pushover analysis

In this revolutionary development, the load distribution of the procedure is not kept constant but is continuously updated to take into account the stiffness degradation and period elongation of the system and higher mode effects. This is achieved by carrying out eigenvalue analysis at the different steps, considering the current stiffness distribution at that step. The subject will be covered in Section 4.

3.1.5. Static time-history analysis

In static time-history analysis, the applied loads can vary independently in the pseudo-time domain. The applied load P_i in a nodal position i is given by $P_i = \lambda_i(t) \cdot P_i^0$, as a function of the time-dependent load factor $\lambda_i(t)$ and the nominal load P_i^0. This type of analysis is typically used to model static testing of structures under various force or displacement patterns (e.g., cyclic loading).

3.1.6. Dynamic time-history analysis

In dynamic analysis, non-structural mass and damping elements are added to the FE model, as required to solve the dynamic equation of motion. Modeling of seismic action is achieved by introducing acceleration loading at the supports. Further, the ability to employ different loading curves at each support allows for representation of asynchronous ground excitation.

3.1.7. Switching between analysis types

- Whereas the cubic and joint elements can be used for every analysis type, mass elements (dmass and lmass) are not needed in static analyses. Therefore, they can be used only in dynamic, eigenvalue and adaptive pushover analysis. Moreover, damping elements (ddamp and rdamp) are only needed in dynamic analysis.
- In dynamic analysis, the DOF of the support in the direction of the earthquake should be released so that the acceleration input can be applied. For example, if a node is fully supported (x+y+z+rx+ry+rz) and the earthquake is applied in the x-direction, the x-restraint should be released (y+z+rx+ry+rz). If there is earthquake input in both x- and y-directions, the supported DOF become z+rx+ry+rz.

- Whenever the user tries to change analysis type, the program notifies the user of the changes that happen. For example, if the there is a model for dynamic analysis and the analysis type is changed to static pushover, ZeusNL asks the user to remove the mass and damping elements and change the boundary conditions.

3.2. Basic Table Functions

Most of the input data are arranged in tables in the different modules. There are some standard functions that can be used with tables to increase the productivity. Most of these functions were discussed in Section 2. Below is a complete list of the functionality tables that are offered:

- **Copying and pasting.** The user can copy data from or paste data to all the tables. In this way, ZeusNL can interact with other applications (mainly spreadsheet programs, such as Microsoft® Excel). Copying and pasting can be done either by the main menu (**Edit > Copy Selection** and **Edit > Paste Selection**) or by the pop-up menus of the tables (right-click on the tables). Note, if the user try to paste data that is not in the correct format, ZeusNL will generate an error message.
- **Sorting.** If the user clicks on the column headings, ZeusNL will sort the list of items of the table according to the clicked column. For example, if the user clicks on the section **Names Headings**, ZeusNL will sort the sections alphabetically. If the nodal x-coordinates is selected, it will sort them according to their x-value. A first click sorts the data in ascending order. Clicking a second time will sort them in descending order. Note, for node and element identifiers, the sorting is done according to the node number only if the nodes are in the format (word)+(number). For identifiers in different format, the sorting is alphabetical.
- **Adding items.** If the user clicks on the **Add** button, a dialog box appears and the properties of the new table item can be selected. The procedure is straightforward. Note, the same load can be added to many nodes at a single step since the **Loads** dialog box permits multi-selection (keep the Ctrl key pressed to select more than one node). Also note, for drop-down lists with many items (e.g., large models lists with nodes may contain hundreds of nodes), if the user starts typing the name of the item desired, ZeusNL will find the item.
- **Removing items.** The user can remove one or more items by selecting them and clicking the **Remove** button.
- **Editing.** Select one table item and click the **Edit** button (alternatively, double-click on the item). A dialog box similar to the corresponding **Add** dialog box opens and allows the user to change the properties. Multiple editing is not allowed, except for two cases: editing the boundary conditions (restraints) of many nodes and changing the element class of many elements.

The user can change the element class of many elements by selecting them in the **Element Connectivity** page and clicking the **Edit** button.

The user can change the boundary conditions of many nodes by selecting them in the **Restraints** page and clicking the **Edit** button.

- **Incrementation.** The user can easily create new nodes, elements and loads incrementing the existing ones. This was shown in an example in Section 2. The process is straightforward. However, it is important to specify the correct increment so that there are no conflicts between the new and the existing items (e.g., creating a new node with the same number as an existing node or applying a load to a node that does not exist).
- **Subdivision.** If the user selects one or more linear elements and subdivides them, the program creates a new node in the middle of them and two new elements of the same element class replace the existing one. Note, this procedure applies only to linear elements (cubic, dmass or rdamp).

Note, each material, section, element class, node and element should have a unique identifier (word or number or word+number) that should be in a valid format.

Valid ZeusNL identifiers should be up to eight characters long. Moreover, they should not contain spaces or the characters: # , or &.

If the user pastes one of the tables where an item with an identifier already exists, ZeusNL adds a star '*' at the end of it so that the new name is unique.

3.3. Materials

A selection of four material types is available in ZeusNL libraries. Based on these types, the user can create an infinite number of materials that will be used to define sections. The four material types are:

- **stl0.** Linear elastic model. This model is applied for the uniaxial modeling of mild steel.

  One parameter is required: The Young's Modulus.

- **stl1.** Bilinear elasto-plastic model with kinematic strain-hardening. This model is applied for the uniaxial modeling of mild steel.

  Three parameters are required: The Young's Modulus, yield strength and strain-hardening.

- **stl2.** Ramberg-Osgood model (Power-Law) with Masing type hysteresis curve. This model is applied for the uniaxial modeling of mild steel.

  Four parameters are required: The Young's Modulus, Three other material constants determined by a best-fit procedure using the available experimental data.

- **con1.** Trilinear concrete model. This is a simplified uniaxial concrete model. Tension resistance and confinement effects are not included.

  Four parameters are required: initial stiffness, compressive strength, degradation stiffness and residual strength.

- **con2.** Nonlinear concrete model with constant (active) confinement modeling. Accurate uniaxial concrete model based on the work by Mander *et al.,* [1988]. A constant confining pressure is assumed, taking into account the maximum transverse pressure from confining steel. This is introduced on the model through a constant confinement factor, used to scale up the stress-strain relationship throughout the entire strain range. Further, the cyclic rules were significantly improved by Martinez-Rueda and Elnashai [1997] to enable the prediction of continuing cyclic degradation of strength and stiffness, as well as better numerical stability under large displacements analysis.

  Four parameters are required: compressive strength, tensile strength, crushing strain and confinement factor.

- **con3.** Nonlinear concrete model featuring variable (passive) confinement modeling; and uniaxial concrete model, similar to con2, including the advanced variable confinement model developed by Madas and Elnashai [1992]. The latter calculates and continuously updates the transverse confinement stress for a given applied axial strain of an RC member under cyclic or transient loading. Thus, in addition to concrete compressive strength, the characteristics of confinement detailing such as diameter of stirrups, their spacing and yield strength, confined core area and Poisson ratio have also to be introduced to fully define the material model.

  Ten parameters are required: concrete compressive strength, concrete tensile strength, concrete crushing strain, Poisson's ratio of concrete, yield stress of transverse steel, Young's modulus of transverse steel, strain hardening parameter of transverse steel, diameter of stirrups, spacing of stirrups and diameter of concrete core.

- **con4.** Sheikh-Uzumeri nonlinear concrete model. This model can consider the effect of effectively confined concrete core as well as the effect of tie spacing and confining pressure. It is recommended to use for the simplified uniaxial concrete model for square sections with uniformly distributed longitudinal steel.

  Eight parameters are required: concrete compressive strength, steel compressive strength, strain corresponding to maximum stress in plain concrete, ratio of the volume of total lateral reinforcement to the volume of core, center-to center distance of outer tie, tie spacing, number of longitudinal bars and area of one longitudinal bar.

- **frp1.** Uniaxial constant fiber-reinforced plastic confined concrete model

For a comprehensive description of the material types, refer to Appendix A.

3.4. Sections

Fourteen steel, RC and composite section types are available in ZeusNL libraries:

rss	Rectangular solid section
css	Circular solid section
chs	Circular hollow section
sits	Symmetric I- or T-section
alcs	Asymmetric L- or C-section
pecs	Partially encased composite I-section
fecs	Fully encased composite I-section
rcrs	RC rectangular section
rccs	RC circular section
rcts	RC T-section
rcfws	RC flexural wall section
rchrs	RC hollow rectangular section
rchcs	RC hollow circular section
rcjrs	RC jacket rectangular section

For a complete description of the section types, refer to Appendix B.

Each section is described by a set of sectional dimensions (1 through 9 depending on the section type) and materials defined in the **Materials** module (1 for steel sections and 3 through 4 for RC and composite sections).

The user can define an infinite number of sections to be used to define element classes. Each section has a unique name, can be copied, pasted and edited.

Reinforcing bars may be added only to RC sections. The bars should be positioned within the confined region of the section. The reinforcing bars are arranged on the **Section Reinforcement** table of the **Main Program** window in trinities of (A_s, d_3, d_1).

Since the sections are symmetrical, only the bars of the positive 1-3 quadrant have to be specified (especially for T-sections the bars in the positive [1] side of the section should be specified). The rest of the bars are generated by the program.

3.5. Element Classes

The ZeusNL element library includes a set of element types used to model structural elements (beams and columns), non-structural elements (mass and damping) and boundary conditions (supports and joints):

- **Cubic.** Cubic elasto-plastic 3D beam-column element. It is used for detailed inelastic modeling, making use of the uniaxial inelastic material models described above. It accounts for the spread of inelasticity along the member length and across the section depth.
- **Joint.** 3D joint element with uncoupled axial, shear and moment actions. This element is used to model pin joints, inclined supports, elasto-plastic joint behavior, soil-structure interaction and structural gaps.
- **Lmass.** Lumped (concentrated) mass element used in dynamic and eigenvalue analysis.
- **Dmass.** Cubic distributed mass element. It models uniformly distributed mass in dynamic and eigenvalue analysis.
- **Ddamp.** Dashpot (concentrated) viscous damping element used in dynamic analysis.
- **Rdamp.** Rayleigh damping element. It allows the use of Rayleigh damping modeling (proportional to mass and stiffness) in dynamic and eigenvalue analysis.

These element types are used to define element classes. For a complete description of ZeusNL element types refer to Appendix C.

3.5.1. Adding element classes

As mentioned in Section 2, an element class is a number of properties referring to a particular element category. Each element class has different properties, depending on its type. For example, for cubic element classes, the section of the elements should be specified together with the monitoring points in which the section is divided. For lmass elements, the lumped mass value should be specified; for dmass elements, the distributed mass value, etc.

Adding an element class is accomplished with the **Add** button. However, the procedure is a bit more complicated than adding a section. The **New Element Class** dialog box is similar to Fig.28.

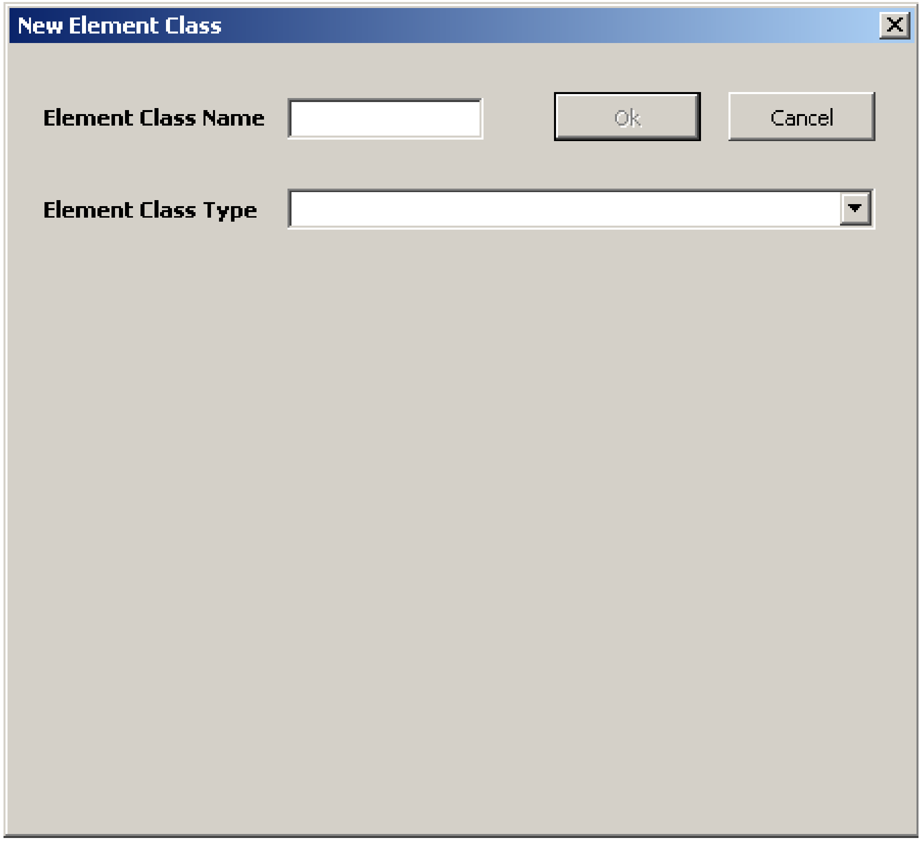


Fig.28 New Element Class dialog box.

The drop-down menu contains a list of the available ZeusNL element types. Note, some element types are not available for particular analysis types (e.g., mass and damping element types [lmass, dmass, ddamp, rdamp] are not available in static analyses).

Selecting an element type from the drop-down menu opens the appropriate lists and textboxes for the user to specify the element properties. For example, if a cubic type is selected, the **New Element Class** dialog box will appear as in Fig.29 and the user has to specify the section for the new cubic element class and the monitoring points in which the section is divided.

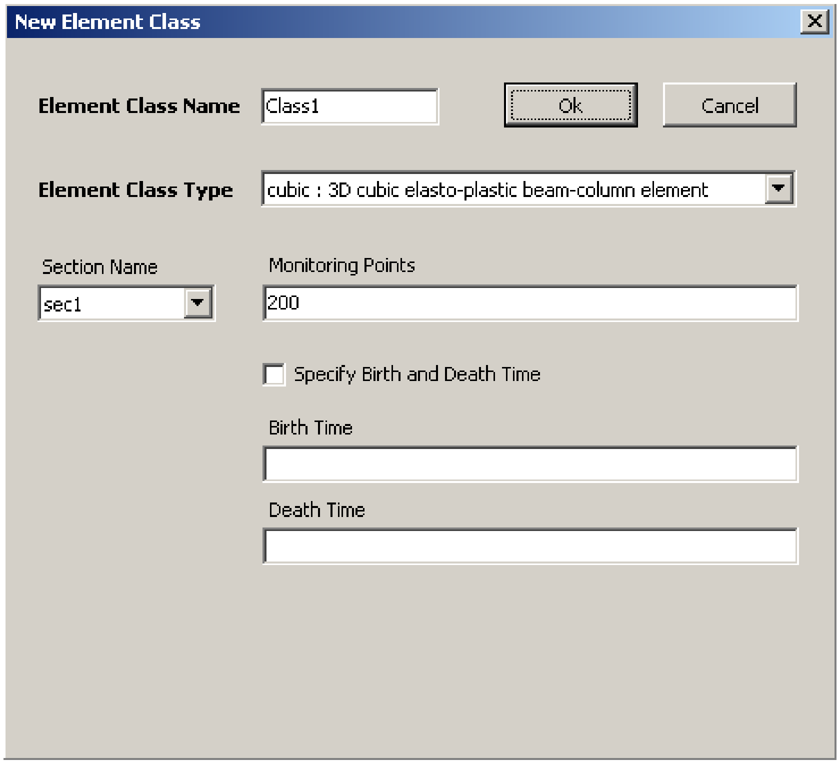


Fig.29 New 'cubic' Element Class.

For lmass, the concentrated mass must be specified; for dmass, the distributed mass must be specified; for ddamp, the six damping parameters C_x,C_y,C_z, C_{xx},C_{yy},and C_{zz} must be specified; and for rdamp, a section name, the mass/length value and the two damping parameters a1 and a2 must be specified. The situation is more complicated for the joint element (Fig.30).

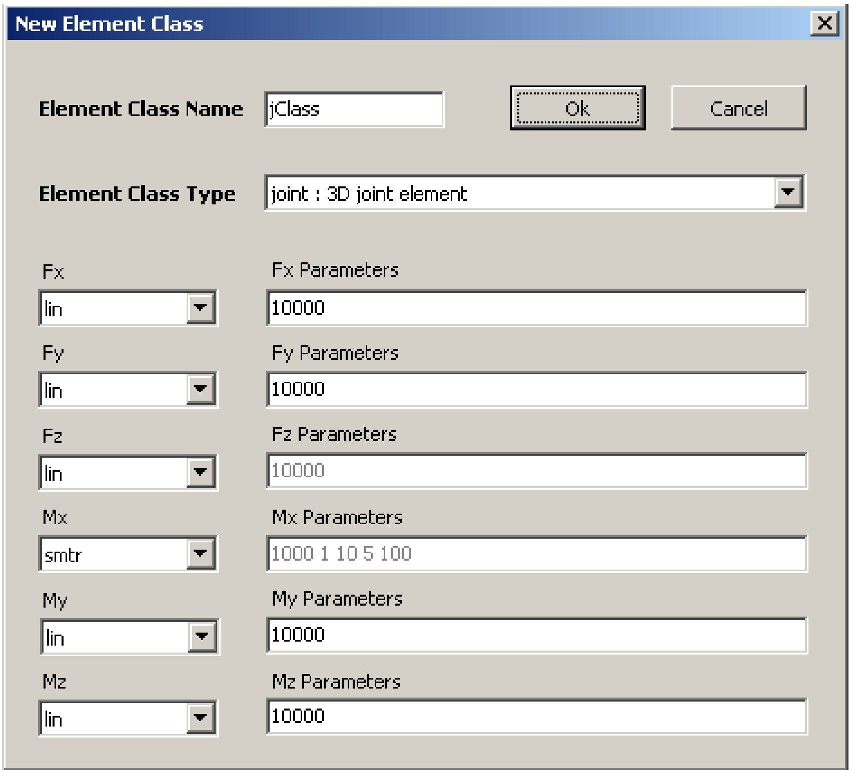


Fig.30 New 'Joint' Element Class.

Six curves have to be defined for the 6 DOF of the joint (F_x,F_y,F_z,M_x,M_y,M_z). There are currently seven curve types in ZeusNL libraries:

- **lin.** Elastic linear curve. Number of parameters: one.
- **smtr.** Tri-linear symmetrical elasto-plastic curve type. Number of parameters: five.
- **astr.** Tri-linear asymmetric elasto-plastic curve type. Number of parameters: ten.
- **hsc.** Hysteretic shear model under constant axial force. Number of parameters: twelve.
- **hsv.** Hysteretic shear model under axial force variation. Number of parameters: forty five.
- **hfc.** Hysteretic flexure model under constant axial force. Number of parameters: twelve.
- **hfv.** Hysteretic flexure model under axial force variation. Number of parameters: forty five.

For a complete description of the curves, refer to Appendix D.

For each of the six curves, there is a drop-down list of the available curve types. Selecting each one of them opens a new dialog box for defining the parameters of each curve (Fig.31).

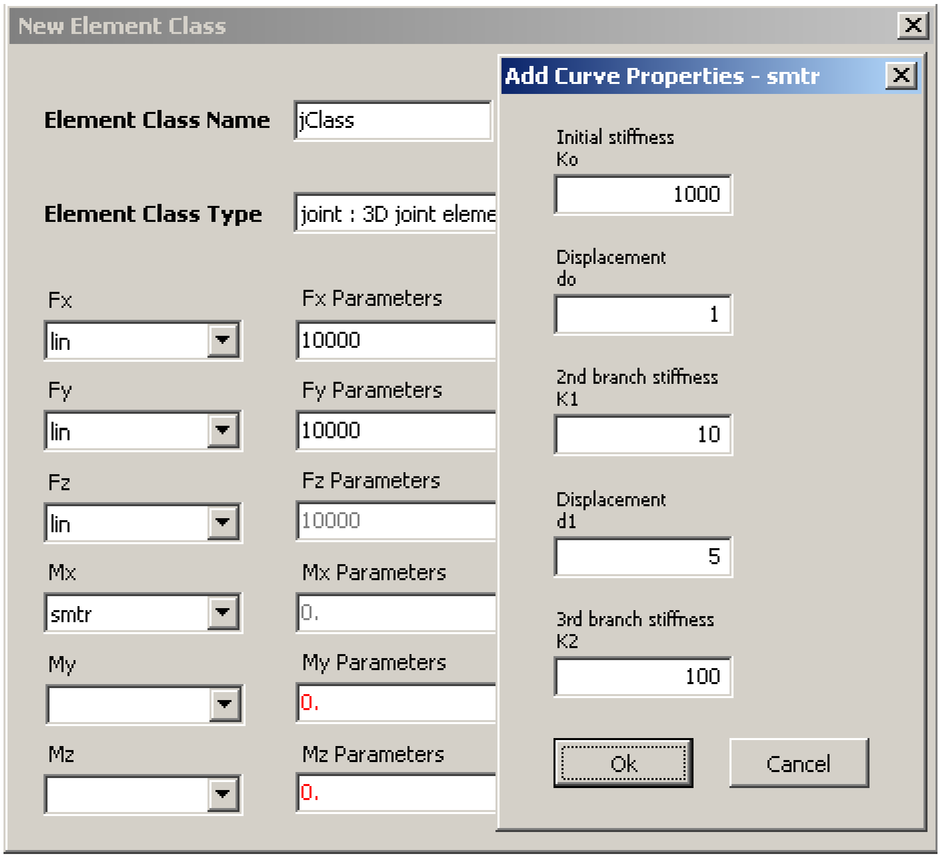


Fig.31 New 'Joint' Element Class. Defining curves.

If the user adds a new element class, the details of the class will be added to the appropriate **Element Class** table (there is a table for each of the element types available for the particular analysis type that is being run).

The element classes defined here are used in the **Element Connectivity** module to define the connectivity of the elements in the mesh configuration.

3.6. Nodes

Non-structural nodes were discussed in Section 2. For some element types (cubic, joint, dmass and rdamp), extra nodes, apart from the end-nodes, should be specified to define the orientation of local axes of the elements. This is the only purpose that non-structural nodes serve. However, structural nodes can also be used as the extra node. It is much more simple and clear to use non-structural nodes for this.

For a comprehensive explanation of the relation of the local element axes with the global axes, refer to Appendix E.

Adding, editing and incrementing nodes are straightforward and shouldn't be difficult or complicated. However, take care during incrementation not to specify nodes that already exist. Also, note that the identifiers of the nodes to be incremented should be in the format: *(word)+(number)*, e.g., n111 and nod20. The word can be omitted and only numbers can be used for identifiers, e.g., 22 and 44. If the user tries to increment a node called ‘n111-y1’, an error message will appear indicating that the node cannot me incremented, since it’s not on the correct format.

> Node and element identifiers have to be in the correct [*(word)+(number)* or *(number)*] format to be incremented.

3.7. Element Connectivity

Each element defined here belongs to a specific element class and, depending on element type of the class, it may have one (lmass, ddamp), three (cubic, dmass, rdamp) or four nodes (joint). The third (for cubic, dmass, rdamp) or fourth (for joint) node may be a non-structural node.

Again, adding, editing, incrementing or subdividing nodes is not difficult. Remember:

- in incrementation, the new element identifiers are unique and the end-nodes of the elements already exist.
- only linear elements (cubic, dmass, rdamp) are subdivided.

One feature that the user will probably find very useful is the ability to change the element class of a large number of elements in one step by making a multiple selection and clicking **Edit**. This is very handy, when for example the user want to the change the beams element of one story from one element class to another.

3.8. Restraints

To change the boundary conditions, select one or more nodes and in the **Restraints** dialog box, specify the restrained freedoms.

Note, in order to run 2D analysis, all the nodes should have z=0 and should be restrained in the z+rx+ry directions. For these models (z-0 and z+rx+ry restrained for **all** of the nodes), the z+rx+ry restraints are not shown on the 3D plot for reasons of clarity.

Also note:

> In dynamic analysis, the restrained DOF at the supports in the direction(s) of the earthquake must be released.

When changing analysis type to and from dynamic analysis, ZeusNL will remind the user to change the boundary conditions at the supports.

3.9 Applied Loading

Depending on the selected analysis type, different load may be applied to the structure:

- **Eigenvalue analysis.** No loads are applied. The stiffness and mass distribution of the structure are needed.
- **Static analysis with non-variable loading.** Only initial loads are allowed.
- **Static pushover analysis (conventional and adaptive).** Initial and proportional loads may be applied.
- **Static-time history analysis.** Initial and static time-history loads.
- **Dynamic time-history analysis.** Initial and dynamic loads.

Definition of load types:

- **Initial loads.** Static loads that are applied prior to any variable load. They can be forces or prescribed displacements applied at nodes.
- **Proportional loads.** Static loads that have proportional variation. The magnitude of a load at any step is given by the product of its nominal value (which is constant) and the current load factor (which varies). Proportional loads may be forces or prescribed displacements applied at nodes
- **Time history loads.** Static loads varying according to different load curves in the pseudo-time domain. The magnitude of a load at any given pseudo-time is given by the product of its nominal value (which is constant) and the variable load factor obtained from its load curve at that pseudo-time. Time history loads may be forces or prescribed displacements applied at nodes
- **Dynamic loads.** Dynamic loads varying according to different load curves in the real time domain. The product of its constant nominal value and the variable load factor obtained from its load curve at that time gives the magnitude of the load. Dynamic loads can be forces or most commonly accelerations applied at the nodes in the global directions.

3.9.1. Applying initial loads

In the **Applied Loading** module, click the **Add** button to show the **Add Loads** dialog box (Fig.32). More than one node can be specified (keeping the Ctrl key down). This adds a load of the specified type, direction and value to all the selected nodes.

Initial loads are usually gravity loads applied to the structure before the variable loading. Note, gravity loads should be applied downwards, which means that they should have a negative value.

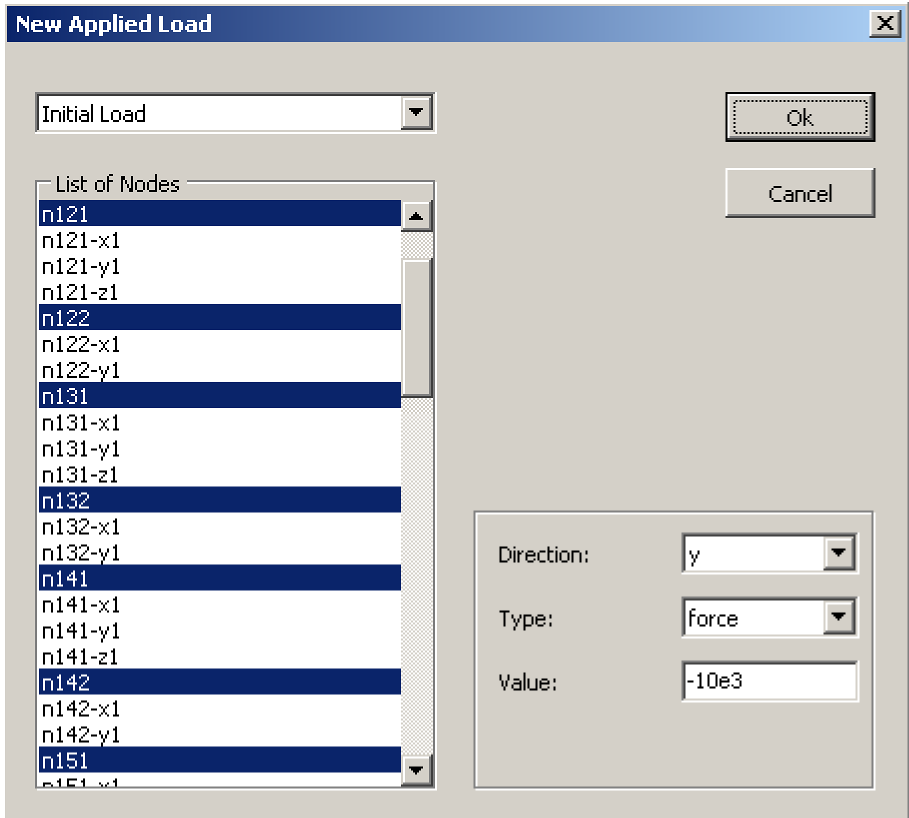


Fig.32 Adding a gravity initial load at many nodes of the model.

3.9.2. Applying loads for pushover analysis

In pushover analysis, the applied loading usually consists of initial (constant) gravity loads in the y-direction and proportional loads (forces or displacements) in the x-direction.

The procedure for adding proportional loads is similar to the procedure for adding initial loads (Fig.33).

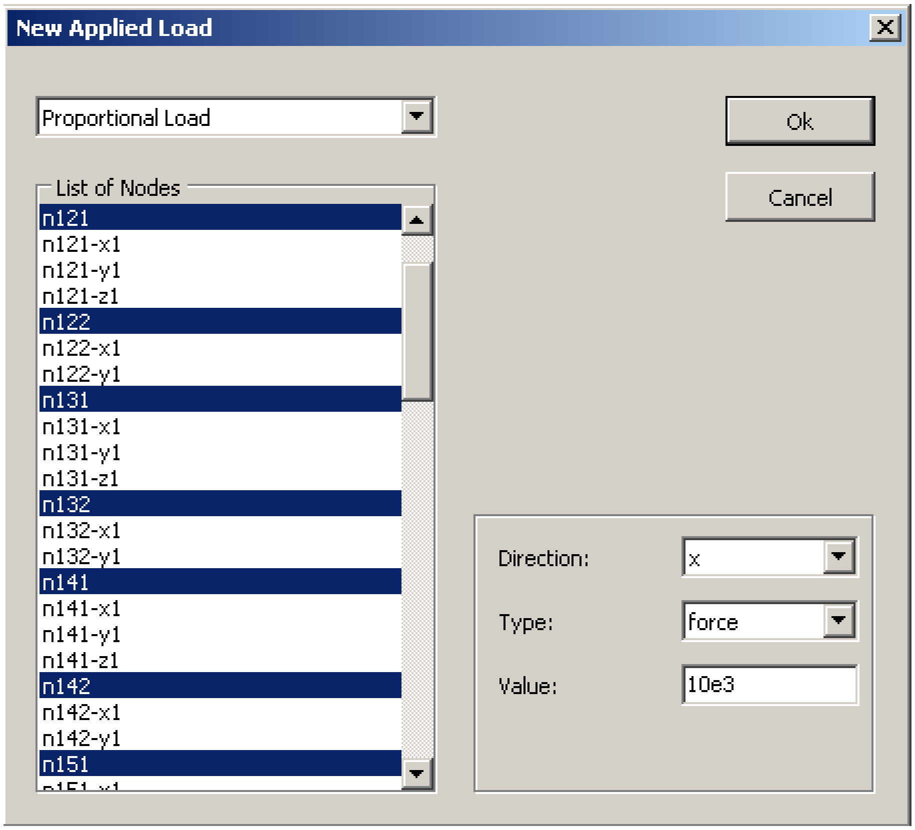


Fig.33 Adding proportional loads at different nodes.

This process indicates the nominal values of the proportional loads. After doing so, however, the user should specify some kind of rules to define how these loads will be increased, in how many steps, etc. This is done in the **Loading Phases** module.

This module defines the control phases used to trace the load deflection curve for proportional loading. Three types of control are available:

- **Load control** is where the load factor λ is directly incremented and the global structural displacements are determined at each load factor level. For the load control phases only an increment value (the factor by which all nominal loads are multiplied to get the target loads) and the number of steps, in which this target load is applied, are required. Loads can be either forces or displacements. For example, assume that the nominal proportional loads applied to two nodes are 5mm and 10mm (Note, do not confuse loads with forces. Loads can be either forces or displacements), the increment is 3 and the number of steps 100. The total loads applied to the nodes are 15mm (5x3) and 30mm (10x3) respectively. These loads will be applied in increments of 0.15mm (15/100) and 0.3mm (30/100).
- **Response control** refers to direct incrementation of the global displacement of one node. This displacement is being controlled by the program and at every step is equal to the value:

(displacement increment)x(number of current steps)/(total number of steps).

The parameters that should be specified in this control type are the controlled node, the controlled direction, the displacement that will be incrementally applied to the node and the number of steps that this displacement will be applied in (Fig.34).

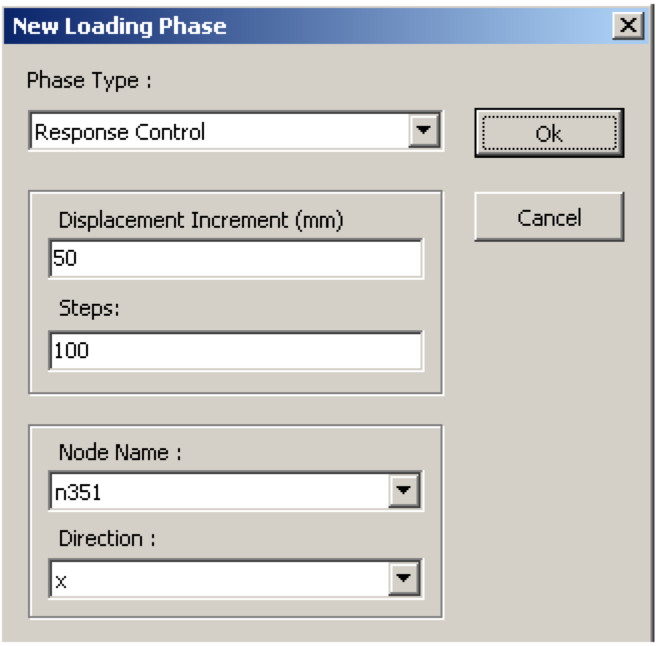


Fig.34 Response control in ZeusNL.

To clarify, assume that there is a very simplified structure with three nodes: n1, n2 and n3.

- If a load control phase is applied with force loading, the program simply increments these forces and calculates the resulting displacements.
- If a load control phase is applied with displacement load, the program controls all three displacements of n1, n2 and n3, applies the displacement increments and calculates the forces generated in the structure, due to the displacements.
- However, if response control is used, controlling the displacement of n3 and applying forces rather then displacements, the program applies forces but calculates the load factor of the forces so that the displacement of n3 is equal to the displacement value specified by the response control parameters (displacement increment and steps). The program actually controls only the n3 displacement and calculates the displacement of n1 and n2 from structural equilibrium.
- **Automatic response control** refers to a procedure in which a new DOF is automatically chosen by the program for response control, whenever convergence difficulties arise during the analysis. The chosen node is the one having the highest rate of nominal tangential response.

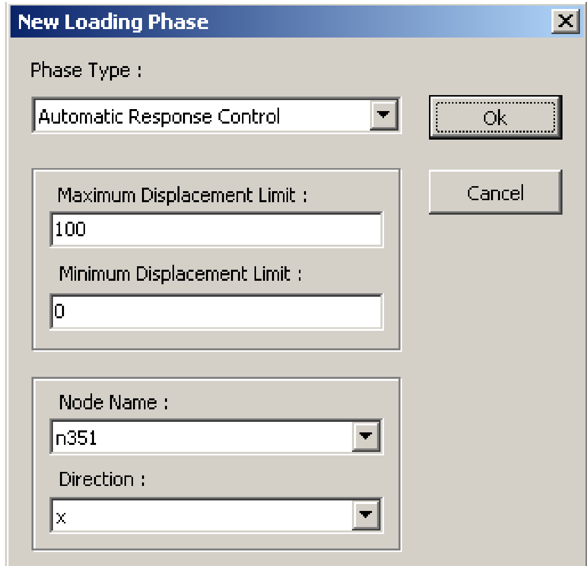


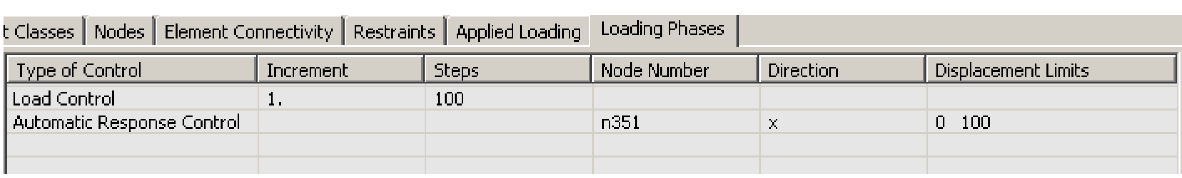
Fig.35 Automatic response control in ZeusNL.

The parameters required for this control type define a termination condition for the procedure. The procedure terminates if the displacement of the selected node in the selected direction exceed the specified limits. Do not confuse the DOF selected by the program for the response control with the DOF specified by the user to define the end condition of the procedure. Note, automatic control cannot be the starting phase of a pushover analysis.

Apart from the above three types of control, there is actually a fourth one available only for the adaptive pushover. This will be covered in Section 4.

There are different possible control schemes that can be efficiently applied:

- **One Load Control phase (forces applied) and one Automatic Response Control phase.** Applying forces rather than displacements seems more attractive because force-based pushover tends to identify much better structural deficiencies, such as soft-stories. However, force-based pushover diverges at the peak of the curve and cannot describe the descending branch. This is the reason that automatic control is used for the second phase. Instead of one load control phase, two or more may be used in order to apply the forces in the inelastic range in smaller increments (Fig.36).



Type of Control	Increment	Steps	Node Number	Direction	Displacement Limits
Load Control	1.	100			
Automatic Response Control			n351	x	0 100

Fig.36 Load control and Automatic control scheme.

- **One Load Control phase (displacements applied).** If the user is interested in keeping the ratio of the displacements at the story heights fixed, the displacement loading strategy may be used. The loading consists of controlled displacement at selected nodes and the ratio of these displacements remains constant during the pushover analysis. With displacements rather than forces applied, the descending branch of the curve may be derived.
- **One Response Control phase (forces applied).** As mentioned earlier, the applied variable load is forced on selected nodes, but the analysis checks convergence, taking into account the displacement at the controlled node.

3.9.3. Applying loads for static time-history analysis

The variable loading consists of displacement, forces or a combination of both which vary independently in the pseudo-time domain, according to a prescribed load pattern.

The load pattern is defined in the **Time-History Curve** module. The user can either load or create a new curve. Usually, static time-history analysis is used to model simple cyclic tests on specimens. In these cases the loading curve is fairly simple, so the user will need to define it rather than load it. By contrast, in dynamic analysis, the applied curve is usually an accelerogram that is loaded in ZeusNL.

To create a new curve, click the **Create** button. This brings the user to the **New User-Defined Curve** dialog box (Fig.37). Input the pseudo-time and load factor values in the table on the left. Use **Enter** or **Tab** to move to the next cell or line and specify a name for the curve. The name will be used in the **Applied Loading** module to use the defined load pattern. Note, the values and the graph can be copied or printed with the pop-up menus (right-clicking on the table and plot).

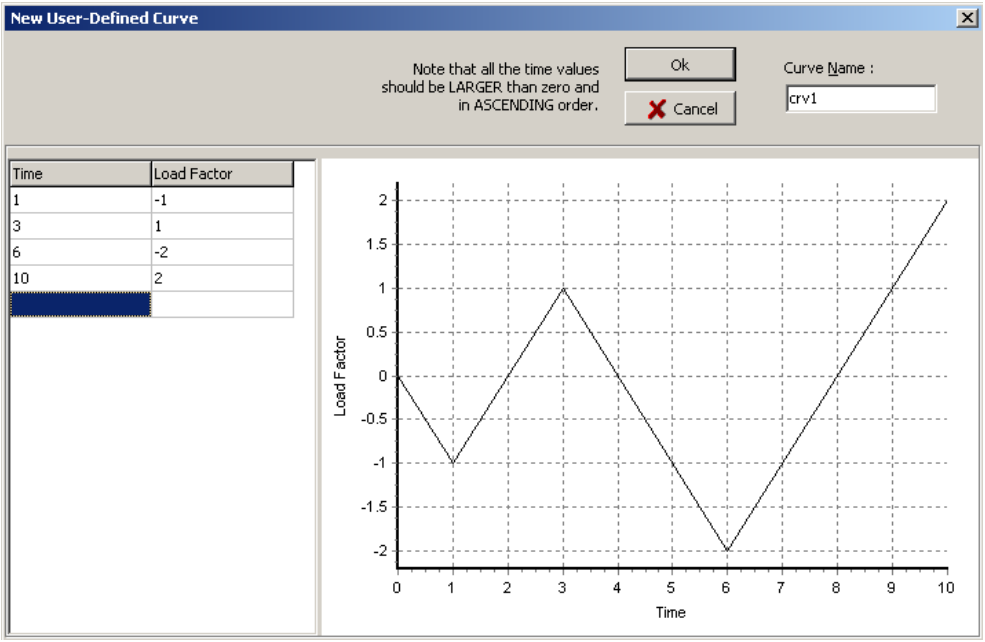


Fig.37 The New User-Defined Curve dialog box.

The **Start Time** value is at the left of the main window. It is the time that the analysis starts.

All of the time entries of the curves (either loaded or user-defined) should be larger than (not equal to) the start time.

To edit the Start Time, there shouldn't be any curves defined. Usually the default value (zero) is fine.

After defining the loading pattern, the applied loading that uses this pattern should be specified. In the **Applied Loading** module, select **Add** and define the node(s), where the pattern will be applied, the direction, the type (force/displacement), the value, with which the pattern values will be multiplied and the curve name (Fig.38).

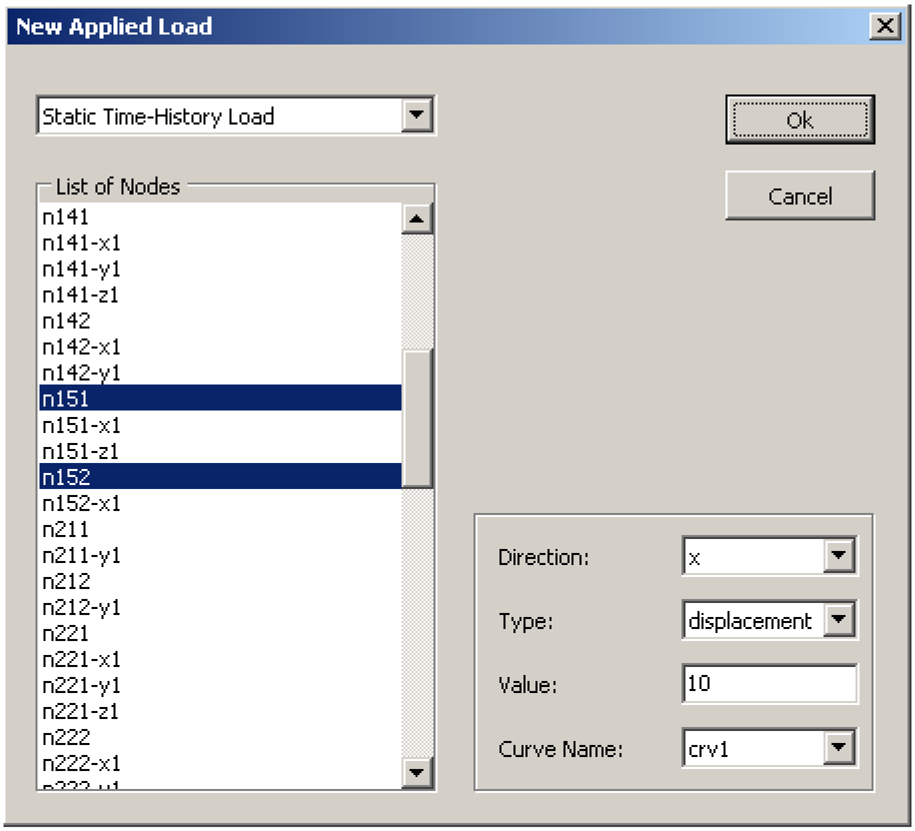


Fig.38 New static time-history load.

The last thing to do before running the project is to define at which time-steps structural equilibrium is sought. This is done in the **Equilibrium Stages** module. More than one stage can be added, if the user wants to have smaller time-steps in demanding phases of the analysis. To add a stage, enter the time of the end of the stage and the number of steps. The time-step is calculated by ZeusNL as the difference between the end time of the current stage and that of the previous stage, divided by the number of steps of the current stage. For the first stage, the difference between the end-time of it and the **Start Time**, defined at the **Time-History Curves** module, is utilized.

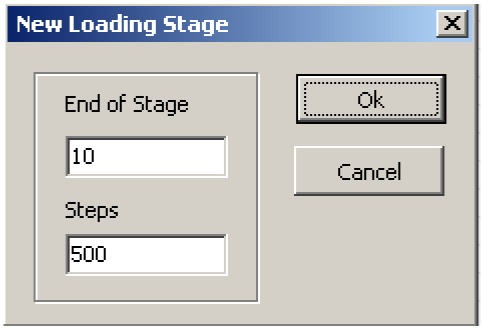


Fig.39 Adding a new stage.

3.9.4. Applying loads for dynamic time-history analysis

As for the static time-history analysis, the user must define a new curve. Usually, this curve is an accelerogram. In the **Time-history Curves** module, select **Load** which opens the **New Curve** from **File** dialog box. Choose the reading parameters (columns of time and acceleration in the file, first line and last line to be read) and load the curve with the **Select File** button.

Note, accelerograms that are not in table format (data in columns) are not supported and have to be transformed before being used (99% of the existing accelerograms are in table format)

If the user is not sure about the reading parameters, the **View Text File** button opens a text file for examination. After the accelerogram has been loaded, the user can copy or print the values and the plot (pop-up menus). Moreover, to change one of the input parameters (e.g., if the user decides that only the first 1500 lines are needed and not all the 2500 lines of the accelerogram) simply click the **Update View** button for the changes to take effect (Fig.40).

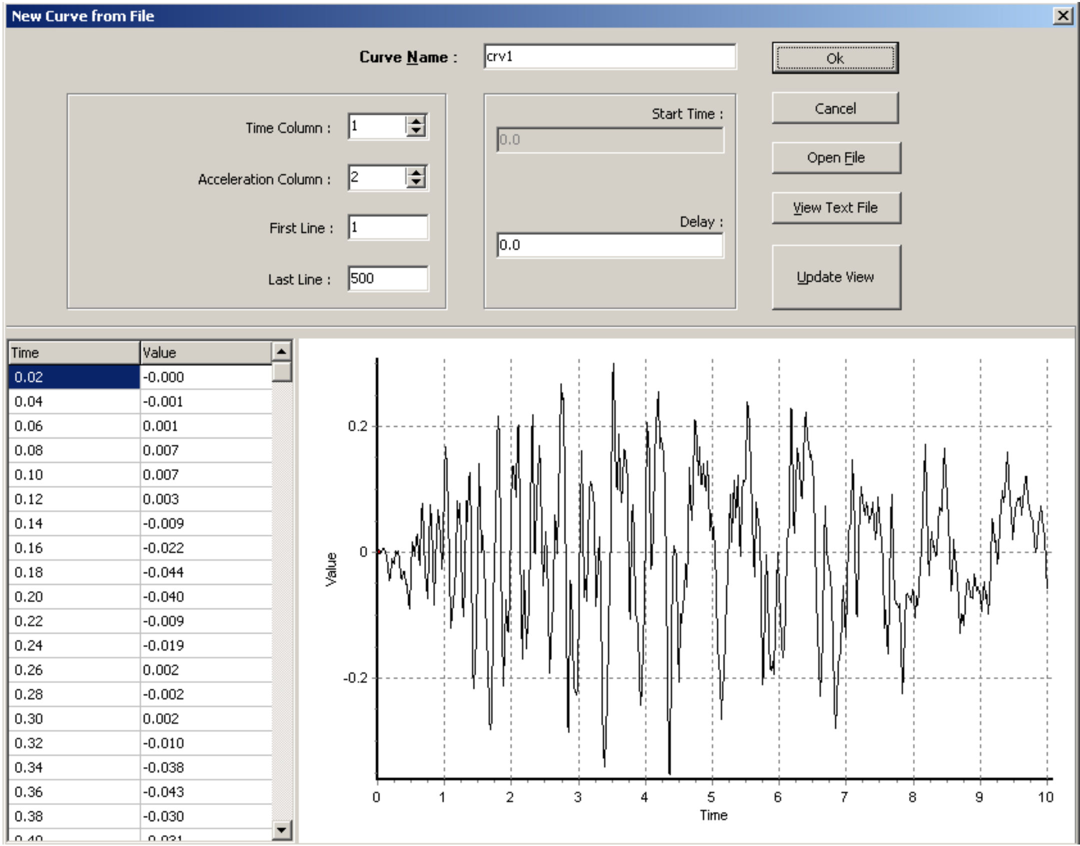


Fig.40 The New Curve from File dialog box.

There is also a Delay parameter which is the time (in seconds), after the Start Time, that the curve starts being applied. At first, this does not seem to have a purpose. For the majority of cases, it is not needed and should be kept to zero. However, it can be used efficiently to run dynamic analysis with *asynchronous earthquake loading* by defining curves that are exactly the same but have different delay parameters (Fig.41).

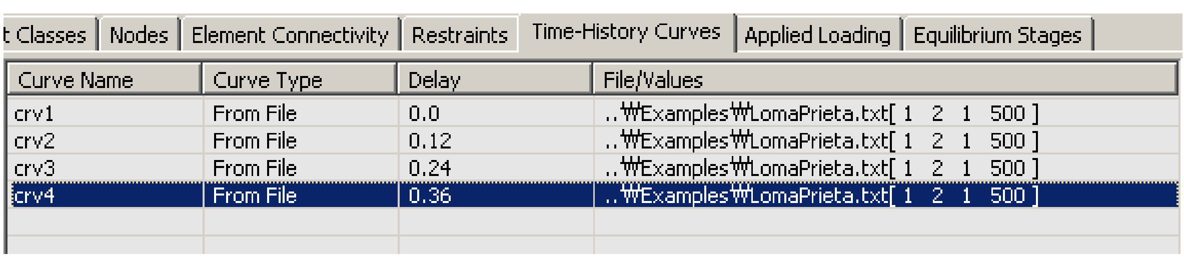


Fig.41 Curves for asynchronous earthquake input.

Adding the dynamic loads in the **Applied Loading** module is very similar to adding static time-history loads (Fig.42). The user has to specify the node(s), the direction, the value (that is the scaling factor) and the curve name. If the accelerogram is in g (most common), the value should be 9810.

The scaling factor for transforming g to mm/sec^2 is 9810.

The dynamic loads are usually accelerations applied at the supports (they can also be forces). The restraint at the support in the direction of the earthquake should be released (x+y+z+rx+ry+rz becomes y+z+rx+ry+rz).

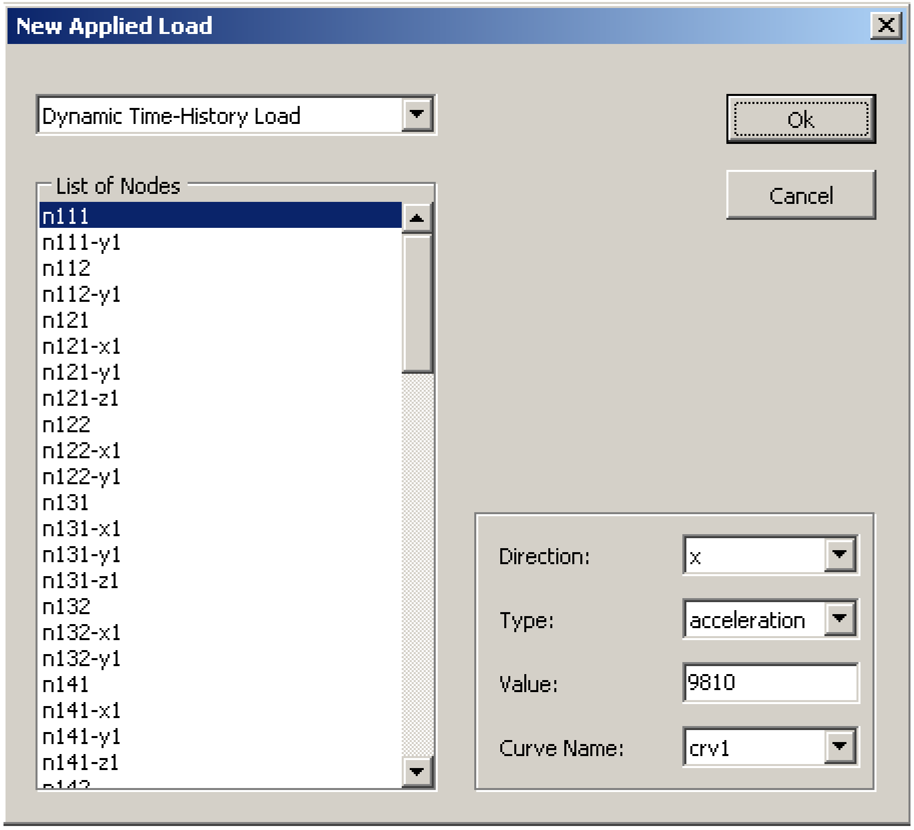


Fig.42 New dynamic time-history load.

Finally, the equilibrium stages of the analysis should be defined, as in the previous section.

3.10. ZeusNL Settings

ZeusNL has a rich set of program settings that help the user optimize the efficiency and the performance of the analysis. To open the **Settings** dialog box, select **Tools > Settings**.

3.10.1. General tab

The General tab contains general settings for the program:

- **Set as Default/Program Defaults.** ZeusNL has a set of default program settings. After installation, these settings are loaded each time the user runs ZeusNL. These settings have been thoroughly tested and proven to result in an optimal performance. However, the user may wish to use personal settings. This is the objective of the **Set as Default** button that saves the current program settings as default. The **Program Defaults** button restores the default program settings.
- **Save Settings.** Another option is to keep the current settings for the next run each time the user closes the program. Check the **Save Settings** checkbox.
- **Tab Position.** Determines the position of the tabs in the main window.
- **Multiple Tabs.** Determines whether the tabs appear on single or multiple rows when there is lack of space.
- **Autosave.** ZeusNL saves a backup of the input file at regular intervals (the default is 5min). The backup files have a **.bak** extension. If a zero value is specified, no backup is kept.

3.10.2. Template

When four-element members are selected from the template, the user may want to specify the exact length of these elements. The default is that the member is divided in elements of length 15%-35%-35%-15%. Changing the option for the end-element of the member allows for changing these percentages.

3.10.3. Integration scheme

These settings are useful only for dynamic analysis and allow for the determination of the integration algorithm and their parameters (alpha, beta and gamma for HHT - beta and gamma for Newmark). Two algorithms are available: Newmark (default) and Hilber-Hughes-Taylor.

Note, the default values (Newmark, beta = 0. 25 and gamma = 0.5) are optimal and the user does not need to change them under normal circumstances.

3.10.4. Iterative strategy

The settings below determine the iterative strategy employed during the solution procedure:

- **Number of Iterations.** Specifies the maximum number of iterations to be performed at each increment.
- **Number of Initial reformations.** Specifies the number of initial reformations of the tangent stiffness matrix to be performed at each increment.

- **Step Reduction.** Specifies the step reduction factor when convergence is not achieved. When the solution diverges or fails to converge within the maximum specified iterations, the increment is reduced by the step reduction factor. The increment can be reduced for up to three times, resulting in an increment (step reduction)^3 smaller than the original value.
- **Divergence Iteration.** The iteration, at which divergence checks are performed.
- **Divergence Criterion.** The reference value used to check for divergence of the solution.

The Newton-Raphson strategy is employed by using a number of initial reformations equal to the number of iterations. Using a number of initial reformations equal to zero is equivalent to the modified Newton-Raphson strategy.

3.10.5. Convergence criteria

The settings determine the convergence criteria for the iterative procedures. There are two different convergence criteria in ZeusNL. The first is based on the norm of the out-of-balance forces. Convergence is attained when the norm is smaller than the tolerance defined in Settings:

$$\sqrt{\sum_{i=1}^{n_t}\left(\frac{G_i^F}{F_{ref}}\right)^2+\sum_{i=1}^{n_r}\left(\frac{G_i^M}{M_{ref}}\right)^2} \le tolerance \Rightarrow convergence$$

Where:

G_i^F = out-of-balance forces

G_i^M = out-of-balance moments

F_{ref} = reference force (defined in **Settings > Convergence criteria**)

M_{ref} = reference moment (defined in **Settings > Convergence criteria**)

n_t = number of translational freedoms

n_r = number of rotational freedoms

The second criterion, which is the default, is based on the maximum iterative increment of displacements which requires the definition of displacement and rotation reference values:

$$\max\left[\left|\frac{\delta d_i}{d_{ref}}\right|_{i=1}^{n_t},\left|\frac{\delta \rho_i}{\rho_{ref}}\right|_{i=1}^{n_r}\right] \le tolerance \Rightarrow convergence$$

Where:

δd_i = iterative displacement i

$\delta\rho_i$ = iterative rotation i

d_{ref} = reference displacement (defined in **Settings > Convergence criteria**)

ρ_{ref} = reference rotation (defined in **Settings > Convergence criteria**)

n_t = number of translational freedoms

n_r = number of rotational freedoms

3.10.6. Output

The output settings determine:

- **Output Frequency.** Specifies the numerical data to be output.
  - When frequency=0, output is printed at all the equilibrated steps, including the step reduction.
  - When frequency=1, output is printed at all the equilibrated steps, without step reduction levels.
  - When frequency=n, output is printed every n equilibrated steps.
- **Stress/Strain Output.** Specifies whether the stresses of all the monitoring points, of the two Gauss points, of each element, are printed to the output file. Use this function only when absolutely necessary. It may result in huge output files (hundreds of Mb for very large structures).

3.10.7. Eigenvalue

These settings specify the number of required eigenvalues, the range of natural frequencies of interest and other parameters:

- **Number of Eigenvalues.** The number of required eigenvalues.
- **Maximum Number of Steps.** The maximum number of steps required to converge to the solution.
- **Minimum Natural Frequency of Interest and Maximum Natural Frequency of Interest.** The default values (zero and extremely large value) mean that all the natural frequencies are of interest.
- **Frequency Shift during the solution of the eigenproblem.** There is no reason to change the default (zero).

> The algorithm does not necessarily output the eigenmodes in an ascending or descending order.

3.11. Other facilities

3.11.1. Template

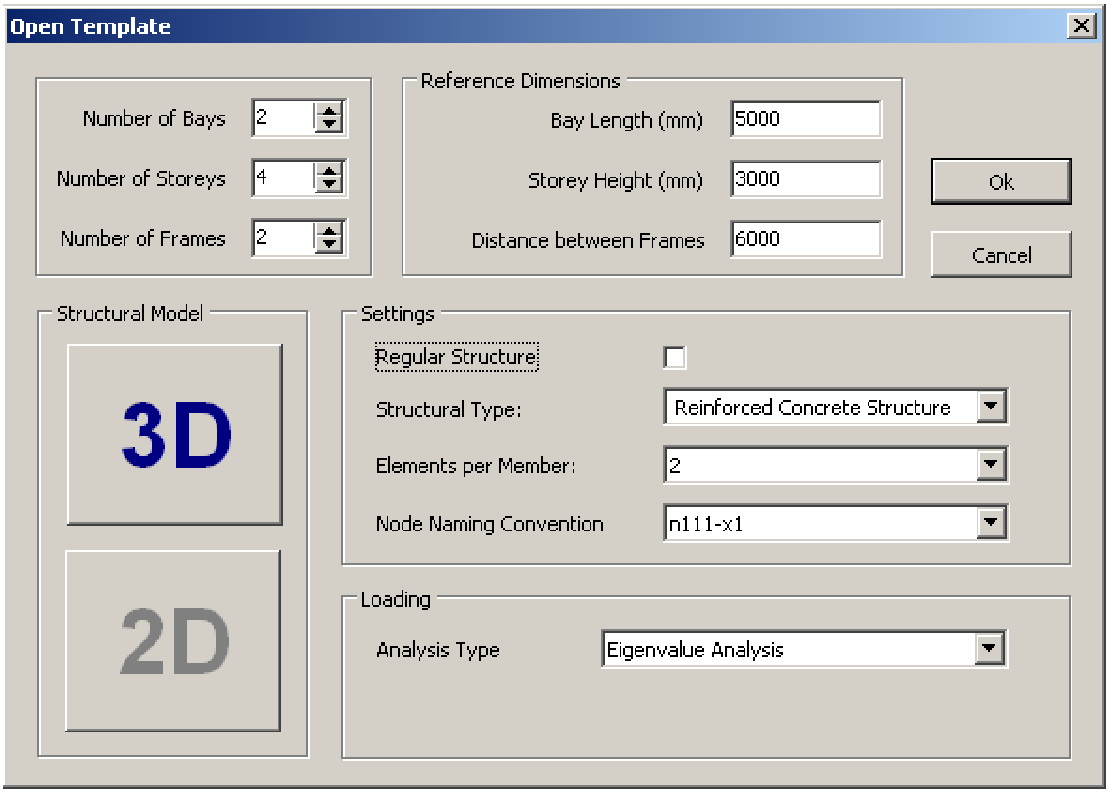


Fig.43 Template screen.

There are many options that the user can choose from:

- **Define the geometry of the structure.** 3D or 2D; number of bays; stories and frames; reference length of bays; height of stories and distance between frames; and regular or irregular structure. Everything is straightforward, apart from the regular/irregular option. In the tutorials, only used regular models were used. Choosing an irregular structure and clicking **Ok**, takes the user to a window similar to Fig.44. The length of each bay is equal to the reference bay length times the bay length ratio. Therefore, the length of the first bay is 9,000mm (5,000x1.8) and that of the second bay is 4,000mm (5,000x.8). The irregular model is shown in Fig.45. The procedure is similar for different story height or distances between frames.
- **One, two or four elements per structural member.** When two elements are selected the members are divided in two equal elements. For four elements, the length of the end elements is determined in the ZeusNL settings (**Tools > Settings**).

- **Node naming convention.**
  - **n111-x1.** All the node numbers at the beam-column joints are of the format 'n'+i+j+k, where i is the column number (starting from the left); j is the story number (starting from the bottom; ground nodes have a j=1 not j=0) and k is the frame number (starting from the front). For example, n132 is the node at the left column of the model (i=1), at the second story (j=3, third level of nodes) and at the second frame (k=2). The nodes on the x-beam, starting from node n121 are n121-x1, n121-x, etc.; the nodes on the z-beam, starting from node n121 are n121-z1, n121-z2, etc.; and the nodes on the y-column, starting from node n121 are n121-y1, n121-y2,etc. This convention is clear, but it has the disadvantage that the nodes of the columns and the beams are not in the format (word)+(number) and therefore, cannot be incremented. However, the template is so powerful and flexible that the user probably won't need incrementation and this convention is the default.
  - **n101011.** All the node numbers at the beam-column joints are of the format 'n'+10i+10j+10k, where i is the column number (starting from the left); j is the story number (starting from the bottom; e.g., n102010 is the node n121, according to the previous convention); and k is the frame number (starting from the front). The nodes on the x-beam, starting from node n121 are n112010, n122010, etc.; the nodes on the z-beam, starting from node n121 are n102011, n102012, etc.; and the nodes on the y-column, starting from node n121 are n102110, n102210. This convention is not very clear, especially when the user has a large number of nodes. However, it allows for incrementation of the nodes since they are in the correct (word)+(number) format.

  The choice of the naming convention is completely up to the user and the needs of the particular project that is running. However note, if one element per member is selected, this option has no meaning since in both cases the node identifiers are derived by the 'n'+i+j+k formula.

- **Analysis and Loading Type.**
  - **Eigenvalue.** The model is derived ready for eigenvalue analysis. Masses are added at the beam-column connections. No loading is applied.
  - **Static analysis with non-variable loads.** No masses are added, just the initial load in applied.
  - **Static pushover analysis.** The user chooses between uniform or triangular proportional loading and is asked to specify the nominal value of the proportional load at the top nodes of the structure. Initial gravity loads are also applied. No masses are added. Two phases are created; one of load control and the second of automatic response control. The proportional loads are applied in the x-direction.
  - **Adaptive static pushover analysis.** Although it is a static analysis, the mass distribution is required for the eigenvalue analysis. As a result, masses are added. Two phases are created; one of adaptive load

control and the second of automatic response control. The proportional loads are applied in the x-direction.

- **Static time-history analysis**. The user chooses between applied displacements or forces, and inputs the loading curve. No masses are required. The load is applied in the x-direction.
- **Dynamic time-history analysis.** The user is asked for an earthquake input (accelerogram). The accelerogram is applied to the supports on the x-direction. The x-DOF of the supports is released (restraints: y+z+rx+ry+rz).

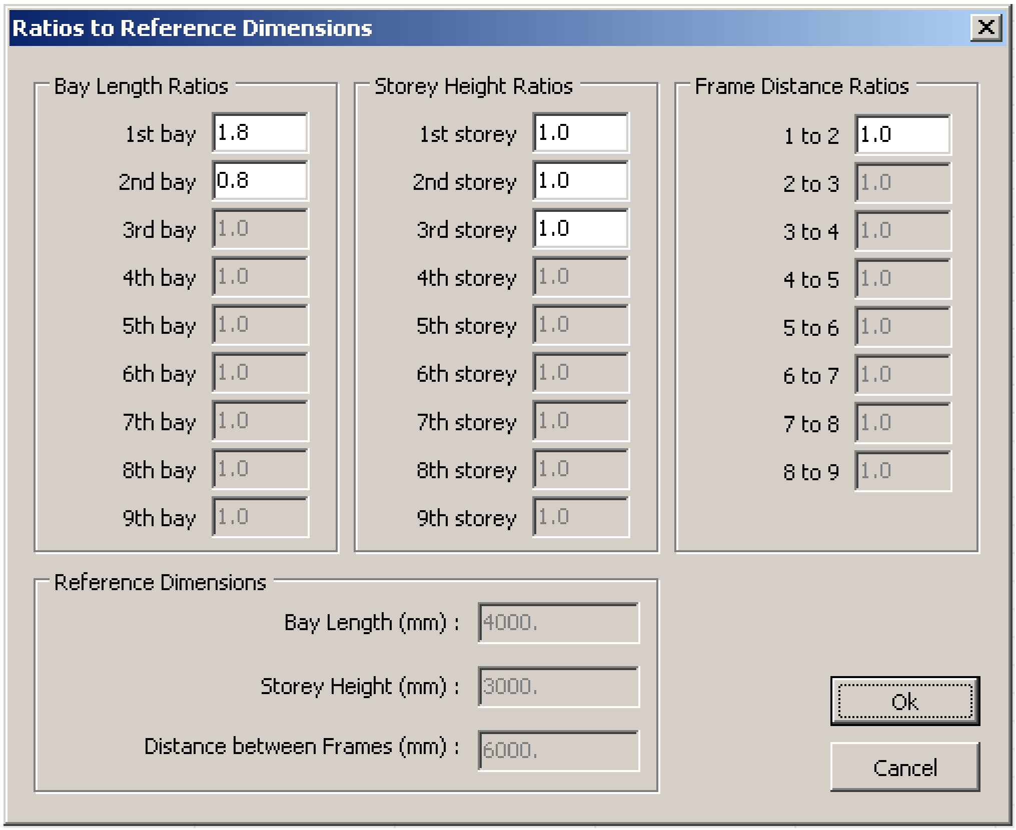


Fig.44 Template - Determination of the dimensions of irregular models.

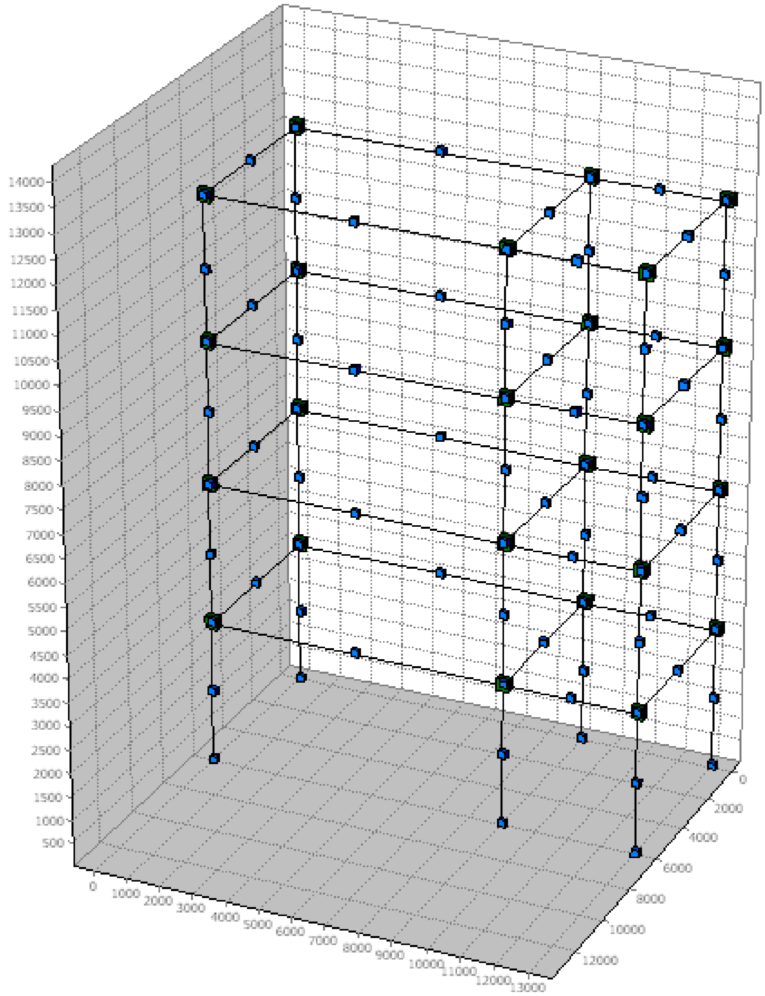


Fig.45 An irregular model created with the template.

3.11.2. Data Entry table

ZeusNL offers a fully functional graphical user interface that permits fast and easy entry of the required parameters. However, there may be some experienced ZeusNL users that know the tables' format and prefer to add entries directly on a table. For these users, there is the **Data Entry Table** facility (**Tools > Open Data Entry Table**).

Opening the **Data Entry** table will open a table with cells that can be edited. The columns and the headings of the table are similar to the table of the currently active module. For example, if the users open the table for the **Nodes** module it will look like Fig.46.

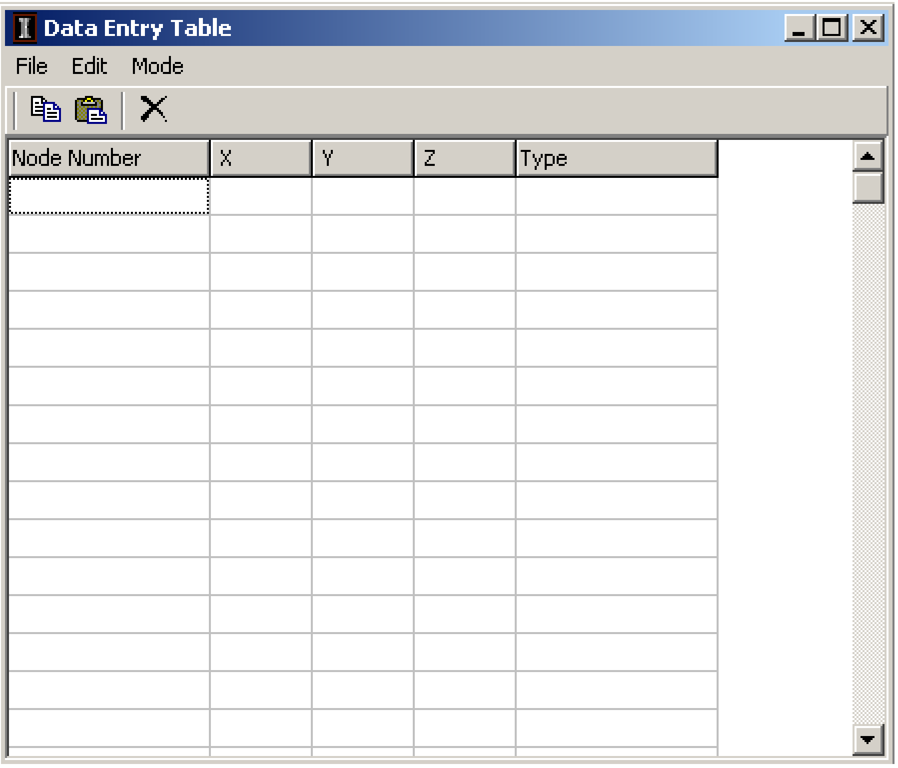


Fig.46 Data Entry Table.

There are two modes on the Data Entry Table: the **Editor** mode that allows the user to enter, change or delete data and the **Selection** mode that allows the user to make a selection and copy it.

Note, data can be pasted to the table, edited and then copied back.

After making the entries, the data can be copied and pasted to ZeusNL main tables. However, make sure that the format is *exactly* the same with the required format.

3.12. Getting the results

3.12.1. Post-Processor

The Post-Processor allows the user to easily derive diagrams from the results of the analysis. It runs with **Tool > Post-Processor** and allows the user to open the project results (.**num**) file. Note, eigenvalue and static non-variable loading analysis files cannot be read by the Post-Processor.

After opening a project, the user can select what is to be plotted in the x- and y- axes from the drop-down lists on the left:

- Nodal displacements
- Nodal rotations
- Interstory drifts
- Support force
- Support moment
- Element shear force
- Time (dynamic time-history analysis): Pseudo-time (static time-history analysis) or Load factor (pushover analysis)

The nodes and elements are selected from dialog boxes that open after a selection is made from the drop-down list.

> In many cases, the user will have to find an item (e.g., a node) in drop-down lists with hundreds of items. If the name of the item (node) is known, start typing the name when the list is active and ZeusNL will find it.

After selecting both (x) and (y), the user can view the curve with the **View Curve** button. The values of the plot are shown on the table at the bottom-left corner.

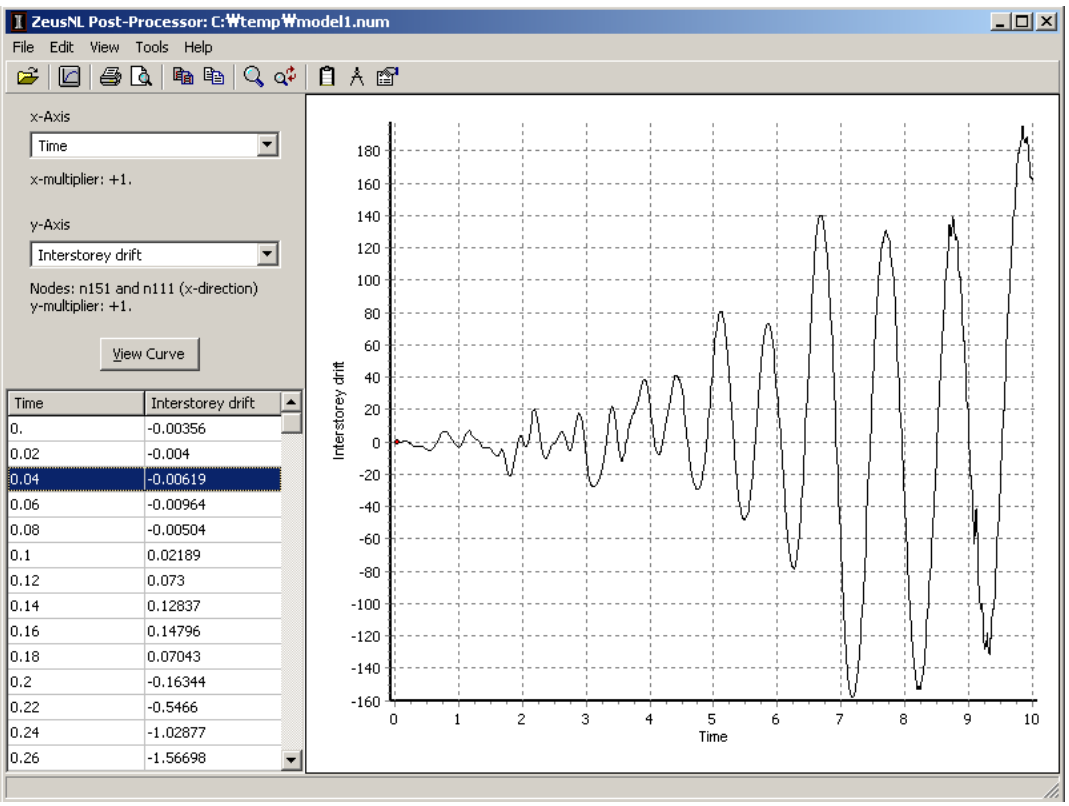


Fig.47 ZeusNL Post-Processor.

The diagram and corresponding values can be copied to other word-processing or spreadsheet programs. To change the appearance of the diagram before copying it (line color, thickness, background, axes values, etc.), open the **Options** dialog box (**Tools > Options**). The process is very straightforward. The user can also zoom in and out with the menu commands, toolbar buttons or by selecting a specific area (a top-left to bottom-right selection zooms in; whereas a bottom-right to top-left selection zooms out).

3.12.1.1. Post-Processor settings

Select **Tools > Settings** to display the settings. The user can choose the value of the x- and y- multipliers, which are the value with which the actual results are multiplied to derive the curve (e.g., multiply everything with 0.001 to get displacements in m rather than mm). The user can choose any value (positive, negative or even zero) for the multipliers.

After selection, the diagram is re-plotted with the new multiplier values.

3.12.1.2. Table Output facility

If the user wants to obtain the model vs. time in a table and there are more than 10 nodes, the **Table Output** facility should be used (Fig.48).

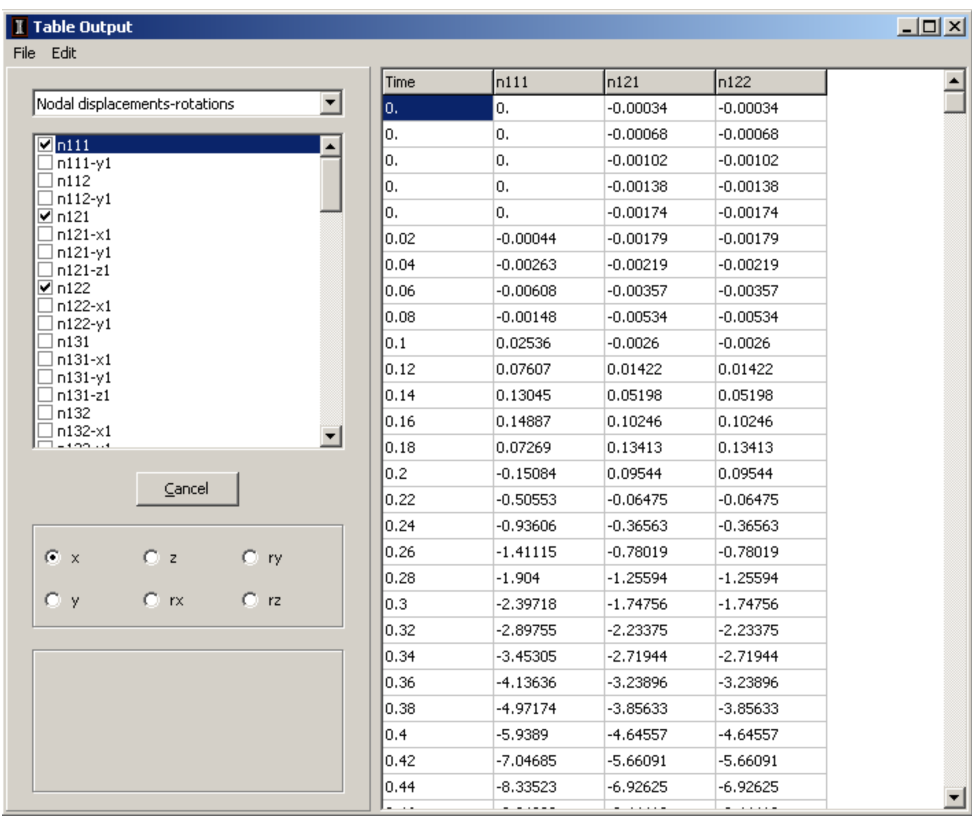


Fig.48 Table Output facility of the Post-Processor.

The **Table Output** facility simply allows the user to obtain the nodal displacements, velocities or accelerations, the cubic or joint element forces, the stresses of the monitoring points and the support forces of a large number of nodes, elements or supports on a table. Then, the read data can easily be copied to spreadsheet programs (e.g., Microsoft® Excel).

Select the type of data to read and the nodes or elements of interest, and click the **Read Data** button. The program starts reading the **.num** output file and prints the required values on the table on the right.

3.12.2. Deformed Shape Viewer

With the Post-Processor, the user can easily create diagrams from an analysis. With the deformed shape viewer (**Tools > Deformed Shape Viewer**), the user can see the deformed shape of the model at different steps of the analysis.

Load the results **.num** file of the project. A list of the steps (time-steps for time-history analysis or loading steps for pushover analysis) or the modes (eigenvalue analysis) appears on the window. To display the deformed shape at one of these steps, select it and click the **View Shape** button. The derived, deformed shape can be then easily copied or printed.

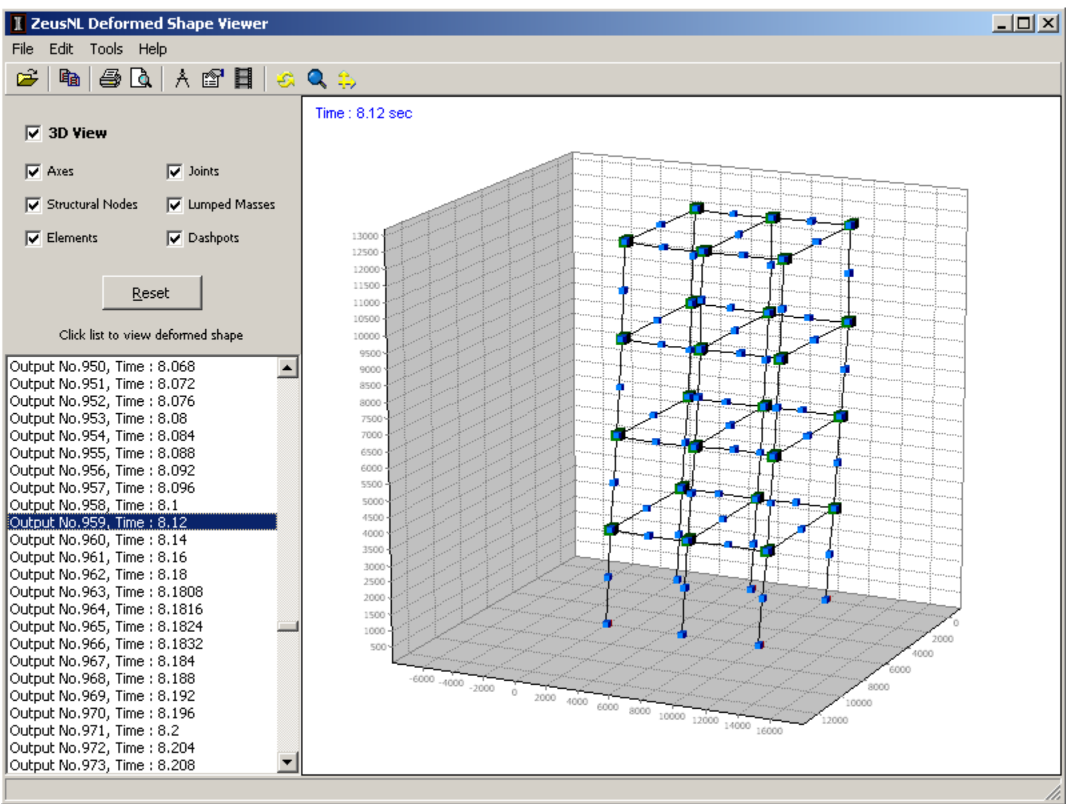


Fig.49 Deformed Shape Viewer.

With the options on the top-left corner, the user can quickly change the appearance of the plot. For more advanced options and full control over the diagram, select **Tools > 3D Plot Options**.

3.12.2.1. Deformed Shape Viewer settings

Select **Tools > Settings** to display the **Settings** dialog box. The two available options are:

- **Deformation Multiplier.** This is the value with which the nodal displacements are multiplied. The purpose of this setting is to exaggerate the deformation in order to have a better insight of the deformed shape.
- **Fix position of the first node.** This setting is useful for dynamic analysis. If checked, the first node is always fixed to the same position. The purpose of this setting is that the DOF of the supports in the direction(s) of the earthquake should be released. This means that the support is free to move according to the displacement of the record. If the record is corrected, there is no problem. However, if it is not corrected, the cumulative displacement may become extremely large. In most of the cases, the nodal relative deformations are of interest rather than their absolute displacements. Fixing the position of the first node eliminates the large absolute displacements of the structure, but keeps their relative displacements unaffected to derive a correct shape.

3.12.2.2. Create a movie of the analysis

ZeusNL allows the user to create a movie of a part of a dynamic, static time-history or pushover analysis. It also allows the user to create a movie with the deformed shapes derived by eigenvalue analysis.

Open a **.num** file and display the deformed shape at a particular step. Change some 3D plot settings, such as the colors, the background color, the axes value, the titles, etc. All these settings are kept in the movie file. Select **Tools > Create AVI File**. First, the user will be asked to specify the name of the file (AVI type) that will be created. If the analysis type is eigenvalue, the user will be asked for the number of frames per half cycle and the number of cycles. Usually the defaults give a good animation result. If it is time-history or pushover analysis, the user will be asked to specify the start and end steps, as well as the frequency with which the steps will be read (Fig.50). Values up to 3-4 usually yield a smooth animation. Changing the settings, results in an approximate size of the derived AVI file.

> When creating AVI animation files, try to keep their size as small as possible. Try to keep the file size up to 50% of the RAM memory.

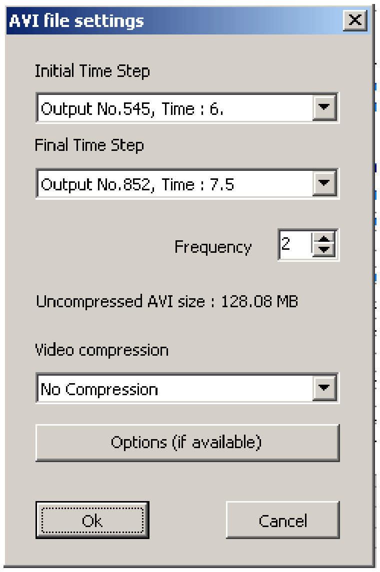


Fig.50 AVI file settings.

The whole process could last up to some minutes, depending on the size of the derived file. The movie can be viewed with the **File > Show AVI File** command.

Note, AVI files can also be opened by other applications such as the Windows® Media Player™ or Microsoft® PowerPoint®, to be inserted into presentations (Fig.51).

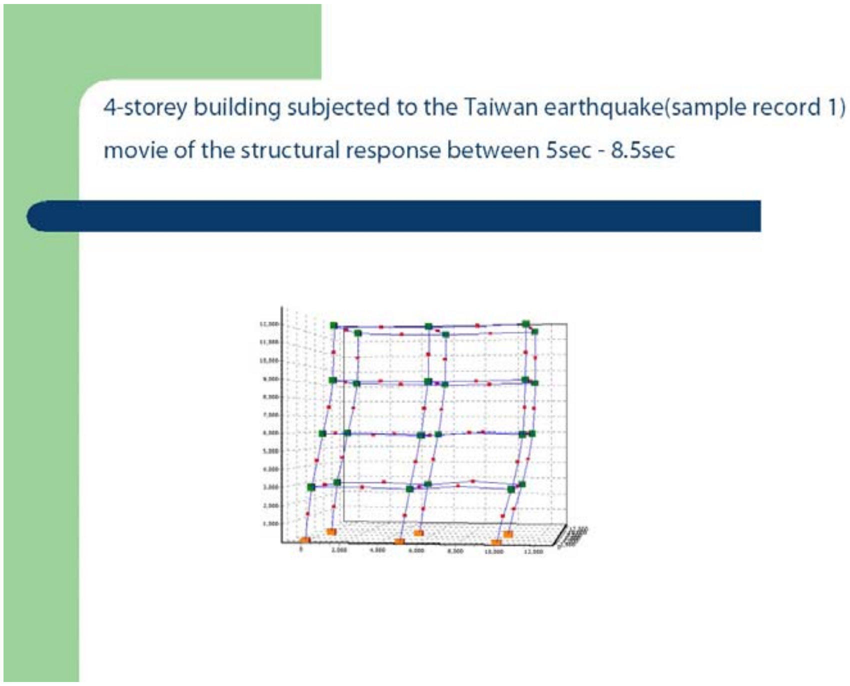


Fig.51 Microsoft® PowerPoint® presentation playing a dynamic analysis movie created by ZeusNL.

4 ADVANCED SUBJECTS

This chapter will present some more advanced subjects, such as the new Adaptive Pushover Procedure, the modeling of structural gaps and the input and output ZeusNL files. The user will need to be familiar with the subjects described in the previous chapters, in order to understand and use the advance features presented hereafter.

4.1. Adaptive Pushover Procedure

4.1.1. Theoretical background

One of the main deficiencies of conventional pushover analysis is its inherent inability to account for the progressive stiffness degradation that occurs during the cyclic non-linear earthquake loading. Consequently, the changes in the modal characteristics, the period elongation and the different spectral amplifications cannot be considered. The fixed nature of the load distribution applied to the structure, which ignores the potential redistribution of forces during the procedure, does not allow for the capturing of these characteristics that are of great significance in an inelastic time-history analysis. Moreover, the deformation estimates obtained from a pushover analysis may be highly inaccurate for structures where higher mode effects are significant.

A new refined approach, which takes into account the current stiffness state of the structure at each step and higher mode effects, is expected to yield more accurate results than the conventional pushover. Such procedure is ZeusNL Adaptive Pushover Procedure.

In the adaptive pushover approach, the lateral load distribution is not kept constant but is continuously updated during the analysis, according to the modal shapes and participation factors derived by eigenvalue analysis carried out at the current step. The new method is fully multi-modal and accounts for period elongation, spectral amplification (through the introduction of a site-specific spectrum), spread inelasticity and geometric nonlinearity of the members. It performs better than the existing conventional methods, especially in cases where strength or stiffness irregularities exist in the structure and higher mode effects are of importance.

A typical analysis involves the following:

1. At each step, before applying the extra load, perform an eigenvalue analysis considering the stiffness state at the end of the previous step and calculate the periods and eigenvectors of the system. The Jacobi method is used for this purpose.
2. Based on the eigenvalue results and the shape of the selected spectrum, the patterns of story forces for each mode are determined as follows:

   $$F_{ij} = \Gamma_j \phi_{ij} M_i S_a(j) \qquad (1)$$

   Where:

   i = story number

   j = mode number

   n = total number of modes considered

   Γ_j = modal participation factor for the j^{th} mode

   φ_{ij} = mass normalized mode shape value for the i^{th} story and the j^{th} mode

   M_i = mass of the i^{th} story

   $S_\alpha(j)$ = spectral amplification of the j^{th} mode

   Whenever the spectral amplification is not considered, the $S_\alpha(j)$ factor in (1) is replaced by the unity and (1) becomes:

   $$F_{ij} = \Gamma_j \phi_{ij} M_i \qquad (1a)$$

   In this case, the lateral load pattern becomes spectrum independent and is defined only by the modal shapes of the system.
3. After defining the lateral load profiles for each mode, the values of the force distribution at each story level F_i are calculated using SRSS or CQC.
4. Update (increase) the load factor. The forces applied at each story i are evaluated as the product of the updated load factor, the nominal load at that story and F_i.

5. Apply the new calculated forces to the model and calculate the members' forces, displacements and rotations, the interstory drifts, the new base-shear and top-displacement, etc., at the new equilibrium state.
6. Calculate the updated stiffness matrix K_{TOT} of the structure.
7. Return to step one for the next step of the pushover analysis.

The procedure is depicted as a flowchart in Fig.42.

Note, before the pushover procedure starts, an eigenvalue analysis is carried out to determine the initial load distribution that is applied at the first step.

The main advantage of the algorithm is that it permits the application of the exact forces profile derived by the eigenvalue analysis at every step, without stability and convergence problems.

The algorithm is able to accept many different options, such as:

- Neglect any spectral amplification and scale according to the modal properties of the structure only.
- Scale according to a user-defined or code-specified spectrum instead of a spectrum derived by a given record.
- Scale only the increment of forces applied at each step and not the total forces already applied to the structure in previous steps. Then, add the scaled increment to the already applied forces that remain unchanged.

Depending on the parameters given, the algorithm yields slightly different results. However, the algorithm of Fig.52 (inclusion of spectral amplification and total scaling according to a spectrum derived by a record) is superior to all the other existing alternatives, in terms of accuracy, without losing in stability.

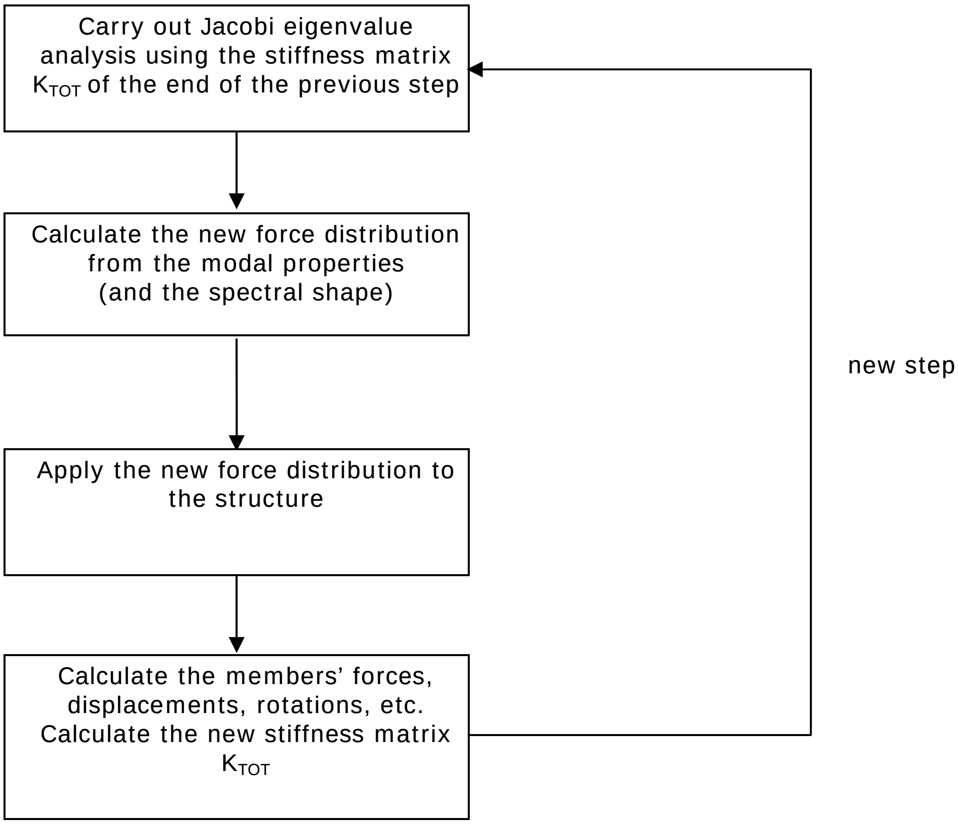


Fig.52 Adaptive Pushover algorithm.

4.1.2. Running Adaptive Pushover

Running adaptive pushover is very similar to running conventional pushover. However, there are some important differences:

- The analysis type should be Adaptive Static Pushover rather then Static Pushover.
- The mass distribution of the structure should be modeled for the eigenvalue analysis carried out at each step.
- The proportional loads input is defined in the same way as in the conventional pushover (Fig.53). However, although it is permitted to use different nominal values for the loads at different nodes, it is preferable that the loads have equal nominal values. In this way, the load applied at every node is determined by the modal characteristics of the structure and the spectral shape. The proportional loads have to be applied to nodes with masses, otherwise they will not be considered in the analysis. Moreover, proportional loads cannot be displacements.

t Classes | Nodes | Element Connectivity | Restraints | Applied Loading | Loading Phases | Adaptive Parameters

Category	Node Number	Direction	Type	Value	Curve Name
Proportional Load	n121	x	force	100000.	
Proportional Load	n122	x	force	100000.	
Proportional Load	n131	x	force	100000.	
Proportional Load	n132	x	force	100000.	
Proportional Load	n141	x	force	100000.	
Proportional Load	n142	x	force	100000.	
Proportional Load	n151	x	force	100000.	
Proportional Load	n152	x	force	100000.	

Fig.53 Proportional loads for Adaptive Pushover analysis.

- There is an extra control type in the **Loading Phases** module: **Adaptive Load Control**. This should be considered as a replacement of **Load Control**. The input is the increment and the number of steps (Fig.54).

: Classes | Nodes | Element Connectivity | Restraints | Applied Loading | Loading Phases | Adaptive Parameters

Type of Control	Increment	Steps	Node Number	Direction	Displacement Limits
Adaptive Load Control	1.	100			
Automatic Response Control			n151	x	0. 600.

Fig.54 Loading phases in adaptive pushover.

- All the parameters needed in the method are defined in a new module called **Adaptive Parameters**. These parameters are:
  - **Frequency.** Determines when the load distribution will be updated. The default is 1, which means that the load distribution is updated at every step.
  - **Type of loading.** Total or incremental loading. Incremental loading means that only the increment of forces applied at each step are scaled. This increment is added to the existing forces that remain unchanged. In contrast, total loading means the forces already applied to the structure are scaled, as well. Total loading yields slightly better results and is the default.
  - **Modal Combination method.** SRSS, CQC or absolute summation (absolute summation could be very inaccurate and should be avoided).
  - **Displacement limit condition.** The adaptive pushover phase finishes when the displacement of the selected node in the specified direction exceeds the maximum or minimum limits.
  - **Spectral amplification.** There are three options:
    - Do not consider spectral amplification. In this case the scaling depends on the modal characteristics of the structure only.
    - Given accelerogram. The scaling takes into account the elastic spectrum of a specified record. Loading the accelerogram is straightforward. The user can use the **Accelerogram** button to see the shape of the record.

- User-defined spectrum. The coordinates of the spectrum are given in a table by the user. This option can be used to introduce code-defined spectra.

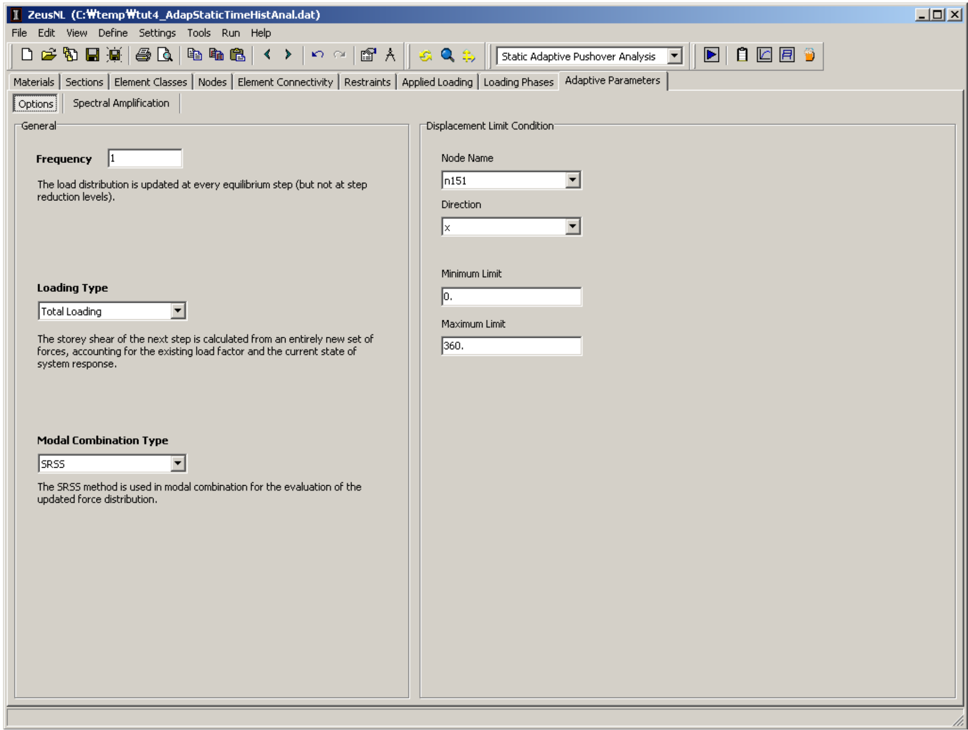


Fig.55 Adaptive parameters (page 1).

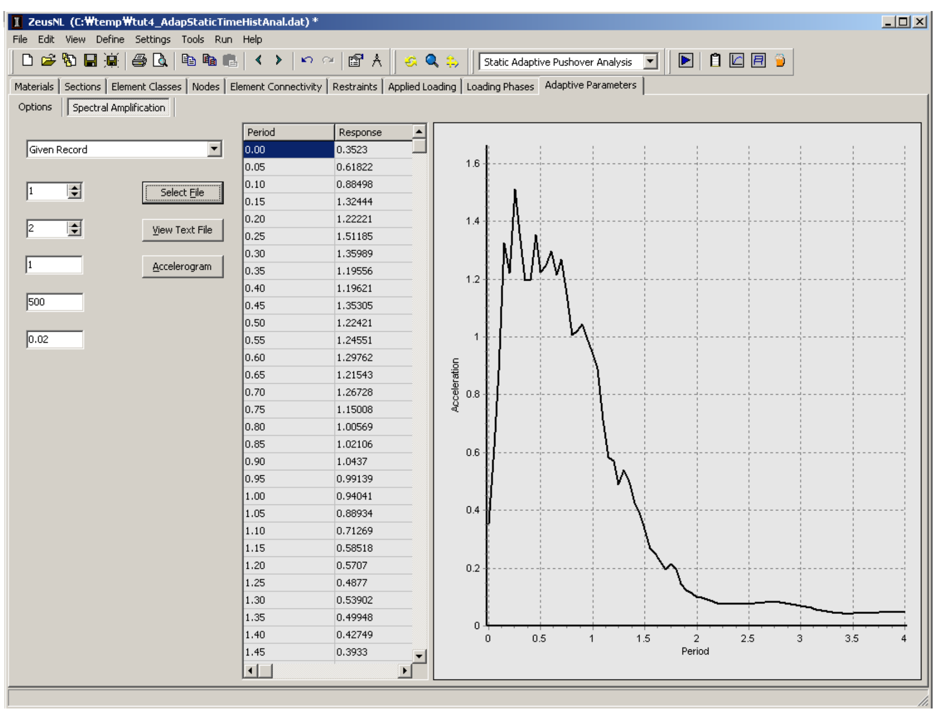


Fig.56 Adaptive parameters (page 2).

After implementing the changes the adaptive pushover is ready to run. The **.num** file is very similar to a conventional pushover **.num** file. However, note that three more output files are created:

- **.pat.** Contains the loading patterns applied to the structure at every step
- **.per.** Contains a list of the modal periods at every step
- **.mpf.** Contains a list of the modal participation factors at every step

4.2. Structural Gaps

In order to model a structural gap in ZeusNL, the joint element with the astr curve should be employed. To model the gap specify zero resistance in the direction of the gap until a certain displacement is reached. At that point, the resistance should have a very large value i.e., theoretically infinite. Also, specify zero resistance in all the other directions. For example, the following parameters represent a curve with zero resistance until a negative displacement –D is achieved:

K^+_0 = arbitrary value (it is not important because d^+_1=0)

d^+_1 = 0

K^+_1 = 0

d^+_2 = arbitrary value (it is not important because both K^+_1=0 and K^+_2=0)

K^+_2 = 0

K^-_0 = arbitrary value (it is not important because d^-_1=0)

d^-_1 = 0

K^-_1 = 0

d^-_2 = -D

K^-_2 = the stiffness of the curve after –D is reached

[K^+_0, d^+_1, K^+_1, d^+_2, K^+_2, K^-_0, d^-_1, K^-_1, d^-_2, K^-_2 are the ten parameters of the astr curve]

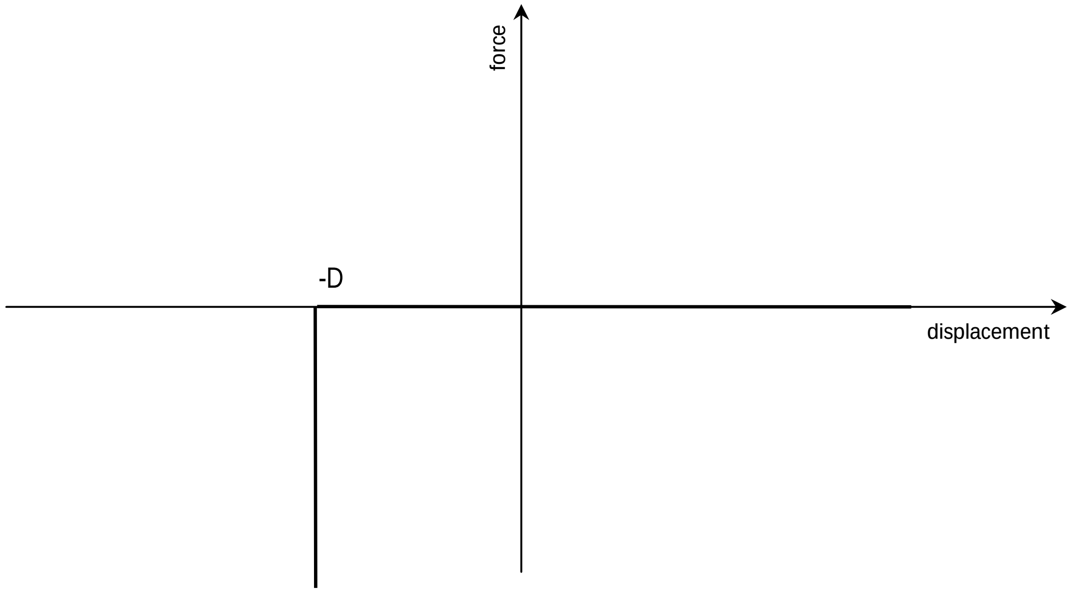


Fig.57 Joint Curve to define a structural gap

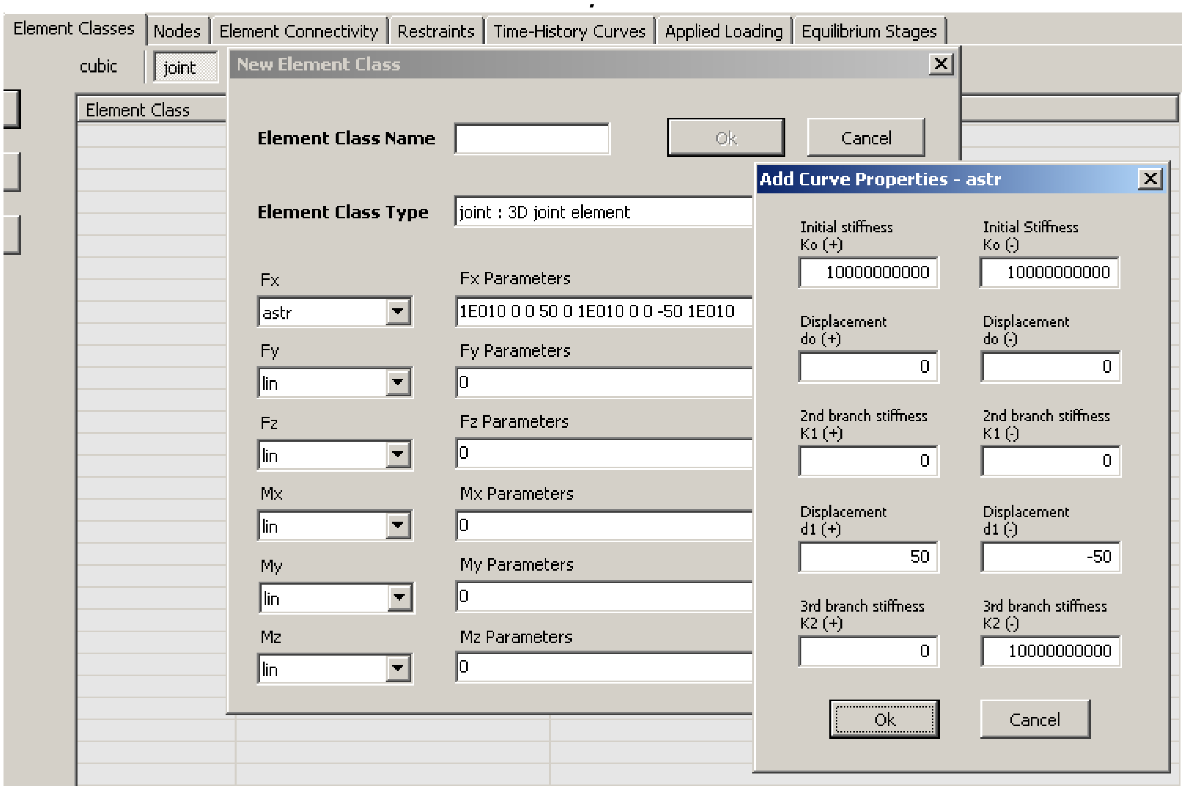


Fig.58 Defining a structural gap from ZeusNL graphical environment.

4.3. Background processing

4.3.1. ZeusNL input data files (.dat)

It is important to have an idea of what happens beyond the tables and graphics: how the input data are saved, how they are restored, how the program runs,, etc.

ZeusNL is based on a text input file to run. This file has a **.dat** extension and is also used to save the data. The user can get an impression of what this file looks like if any file created by ZeusNL is opened with a simple text editor, like Microsoft® WordPad. The user can also see the **.dat** file of the project that is being built from the **View > Open Data File** menu command.

The user will notice many similarities between ZeusNL tables and the **.dat** file. All the data are organized in modules (different module for the section types, the nodal co-ordinates, the element connectivity, etc.). These modules are identified by the program by a unique header (e.g., **Sections, Materials**, **Element Classes**, **Structural Nodal Coordinates**, **Element Connectivity**, **Restraints**, **Applied Loading**, etc.). The headers correspond very closely to the headers of ZeusNL modules. The user should have no problems in understanding how the input data are arranged in the **.dat** file. There are, however, a couple of things that the user should pay attention to:

- Some of the secondary **.dat** file modules are defined in ZeusNL Settings, e.g., the **Integration Parameters** or the **Iterative Strategy** modules.
- The **Phases** module has an unusual path parameter. Moreover, the automatic.control phase is of nod.control translation type and there is a condition.name parameter. The condition is defined in a completely new module called **Conditions**, with no correspondence in the graphical environment. The path parameter specifies the sign of the applied load increments (keep: keeps the same sign of increment, continue: follows the previous loading path). The user will never need to use the path, so there is no need to pay any attention to it. The **Conditions** module specifies the stopping condition of the automatic control phase. Simply, what is specified in the **Loading Phases** module of the graphical environment is divided into the two modules: conditions and phases. Finally, the automatic control type is always nod control translation; meaning control of the translation (rather than rotation) of the node defined in condition cnd1.

```
#--------------------------------------------------------------
               Conditions
#--------------------------------------------------------------
conditions
  response.cnd.name     nod.name     direction     limits
  cnd1                  n151         x             0.   600.

#--------------------------------------------------------------
              Phases
#--------------------------------------------------------------
phases
 load.control
   increment     path       steps
   1             keep       100

 automatic.control
   type                        path       cnd.name
   nod.control translation     continue   cnd1
```

Note: The user will never have to use them in any way and the user will never be asked to edit the **.dat** file directly.

4.3.2. What happens when a project is running?

After a model has been created and saved (that is the **.dat** file), when it is run, two MS-DOS® windows will appear, one after the other. Although only one application is running, in reality two different programs, called by the graphical environment, are running. The first reads the data, arranges them in a certain way (understood by the second) and makes some initial calculations. After that, if everything is correct, the graphical environment is minimized and the second program, which is the actual finite element analysis program, runs.

However, under certain circumstances an error may exist in the input data (e.g., the file with the input earthquake motion may not exist or be corrupt). In this case the user is informed of the occurrence of an error and is asked if they want to see a log file. Answer yes and find the error message (indicated with a distinctive red color). Correct it and run the analysis again.

4.3.3.List of ZeusNL input and output files

Apart from the **.dat** file, there are two other file types that hold input data. The first type is the **.crv** file that holds the data of loaded time-history curves. The second type is the **.adt** file that holds the records which will be used to derive the elastic spectrum used for scaling of the forces in adaptive pushover analysis. These files should not be deleted.

When running a project, ZeusNL creates a number of temporary files that are deleted after the completion of the analysis (.res, .cnd, .res, .lod, .phs, .ref, .rpr, .plt, .sbd, .stg, .spr, .tmp and .eig). Due to their temporary nature, these files are of no importance to the user.

The output files are:

- **.num.** The file that holds the results of the analysis.
- **.nod, .log and .out.** Log files that hold data about the modeled structure and the analysis itself.
- **.pat, .per, mpf.** Files created during adaptive pushover analysis. They hold data about the loading patterns, the modal periods and the modal participation factors at every step. These files can be useful in many ways to the user.

Also note, if the autosave function is activated (this is the default) a back up (**.bak**) file of the **.dat** is being saved at regular intervals.

Appendix A- Materials

In this Appendix, a list of the available ZeusNL material types is presented :

- **stl1** Linear elastic model
- **stl1** Bilinear elasto-plastic model with kinematic strain-hardening
- **stl2** Ramberg-Osgood model with Masing type hysteresis curve
- **stl3** Menegotto-Pinto model with isotropic strain-hardening
- **con1** Trilinear concrete model
- **con2** Uniaxial constant confinement concrete model
- **con3** Uniaxial variable confinement concrete model
- **con4** Sheikh-Uzumeri model
- **ecc** Model for Engineered Cementitious Composite (ECC) materials
- **frp1** Uniaxial constant fiber-reinforced plastic confined concrete model

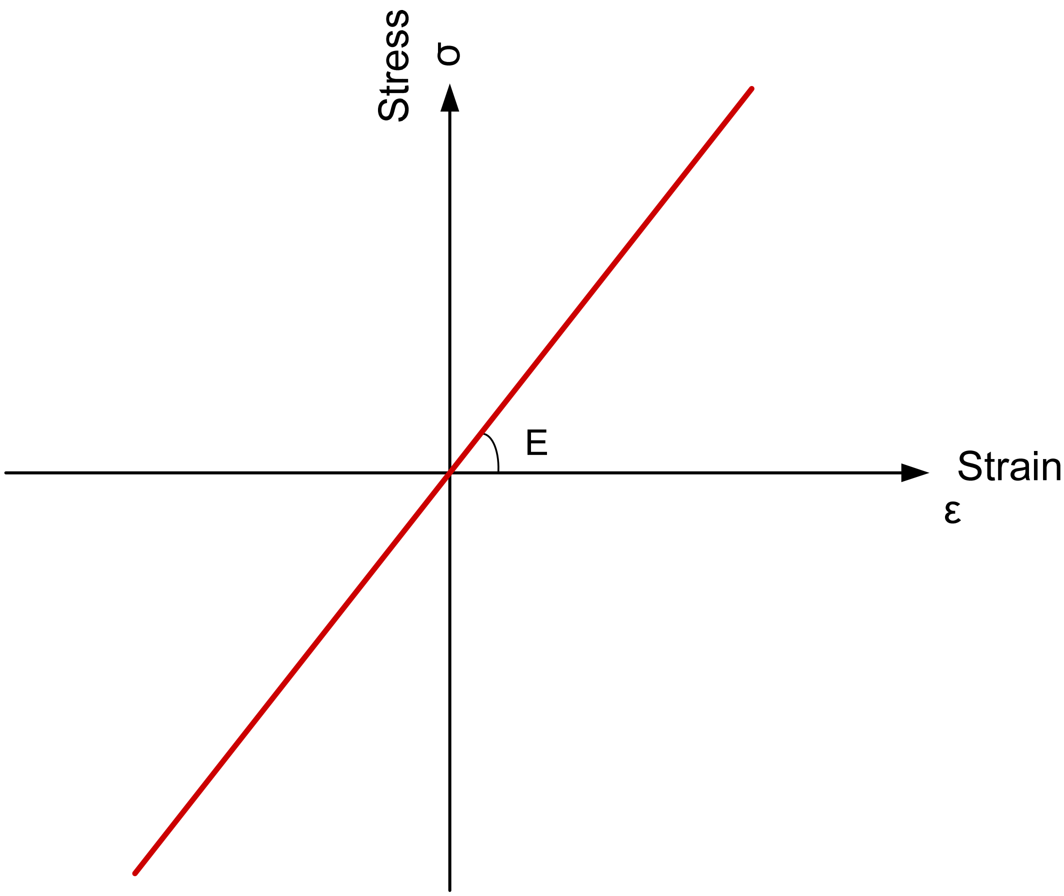
stl0

Linear elastic model

Number of properties: 1

This model is applied for the uniaxial modeling of mild steel.

property	description	typical value
E	The Young's Modulus	200000



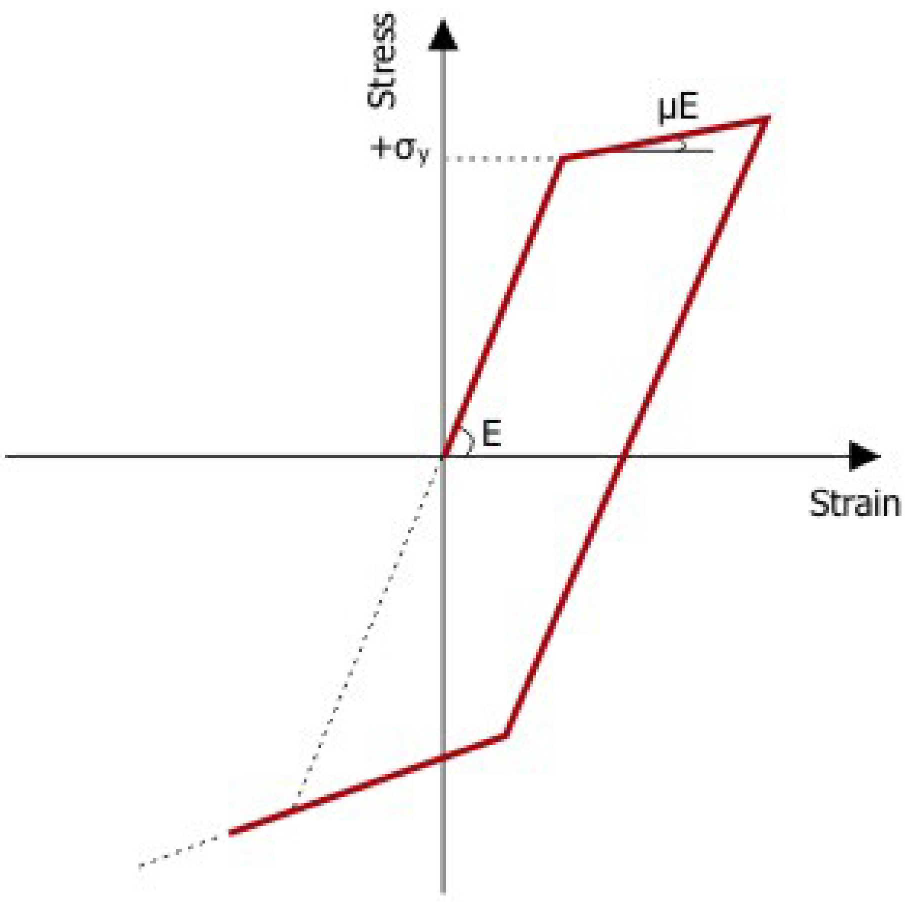
stl1

Bilinear elasto-plastic model with kinematic strain-hardening

Number of properties: 3

This model is applied for the uniaxial modeling of mild steel.

property	description	typical value
E	The Young's Modulus	200000
σ_y	Yield Strength	500
μ	Strain-hardening parameter	0.005



stl2

Ramberg-Osgood model with kinematic strain-hardening

Number of properties: 4

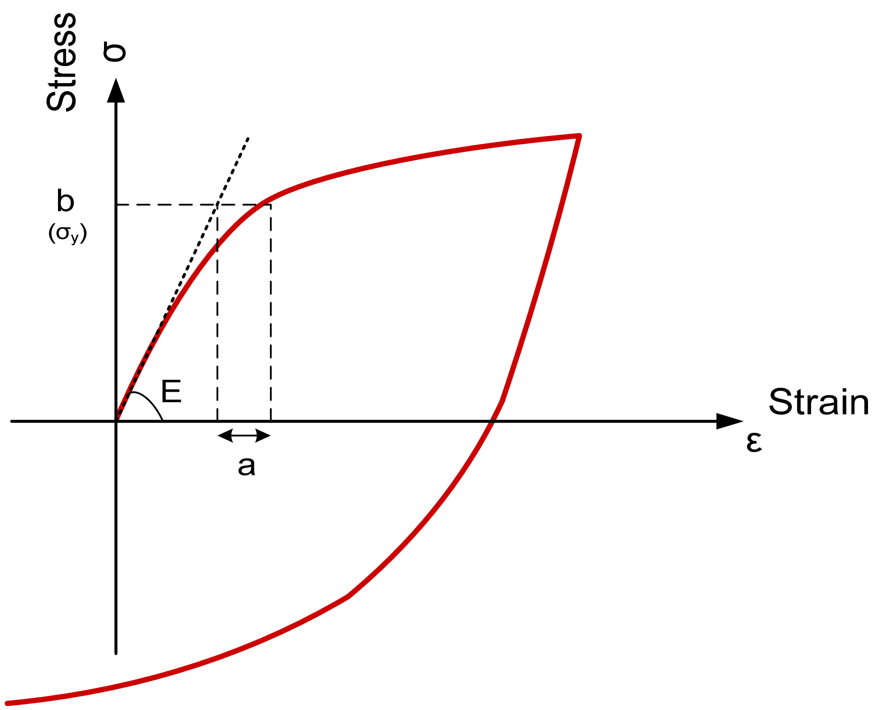
This model is applied for the uniaxial modeling of mild steel.

Stress-Strain Relationship

$$\varepsilon = \frac{\sigma}{E} + a\left(\frac{\sigma}{b}\right)^n$$

property	description	typical value
E	The Young's Modulus	200000
a b n	Material constants determined by a best-fit procedure using the available experimental data.	

* The parameter 'b' is assumed as a yield stress. After this yield point, the model will follow the Masing type unloading or reloading hysteresis curves.



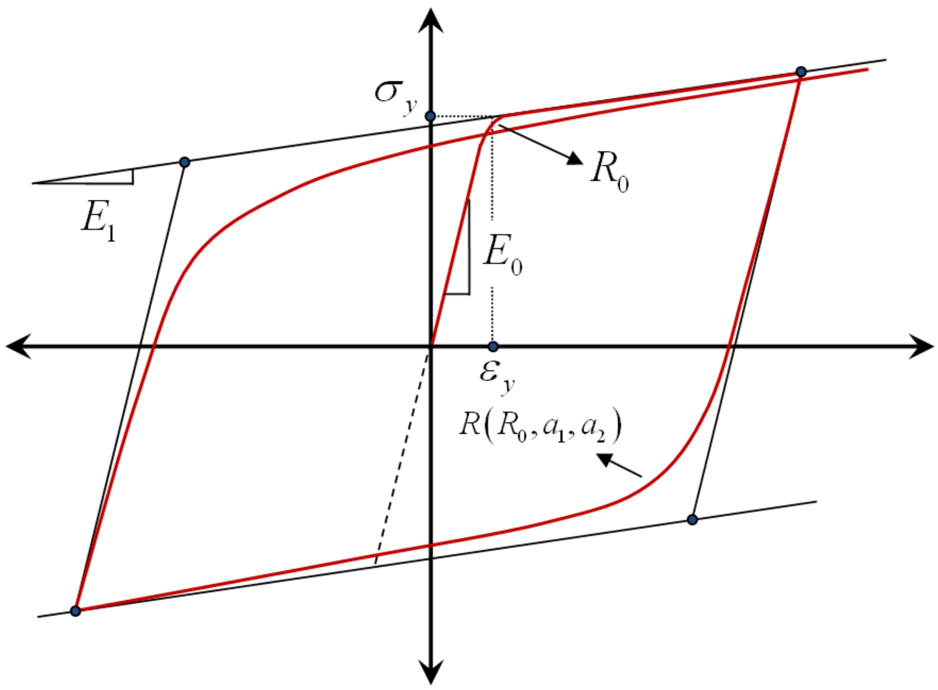
stl3

Menegotto-Pinto model with isotropic strain-hardening

Number of properties: 8

This model is applied for the uniaxial modeling of mild steel.

property	description	typical value
σ_y	Yield stress	500
E_0	Initial elastic modulus	200000
E_1	Strain-hardening modulus	2000
R_0	Parameter defining the initial loading curvature	20
a_1	Experimentally determined parameters controlling the curvature in subsequent cycles	18.5
a_2		0.15
a_3	Experimentally determined parameters controlling the isotropic strain hardening in subsequent cycles	0.01
a_4		7



Using the default values for the parameters R_0, a_1, a_2, a_3 and a_4 is highly recommended unless user has experimental data to determine these parameters.

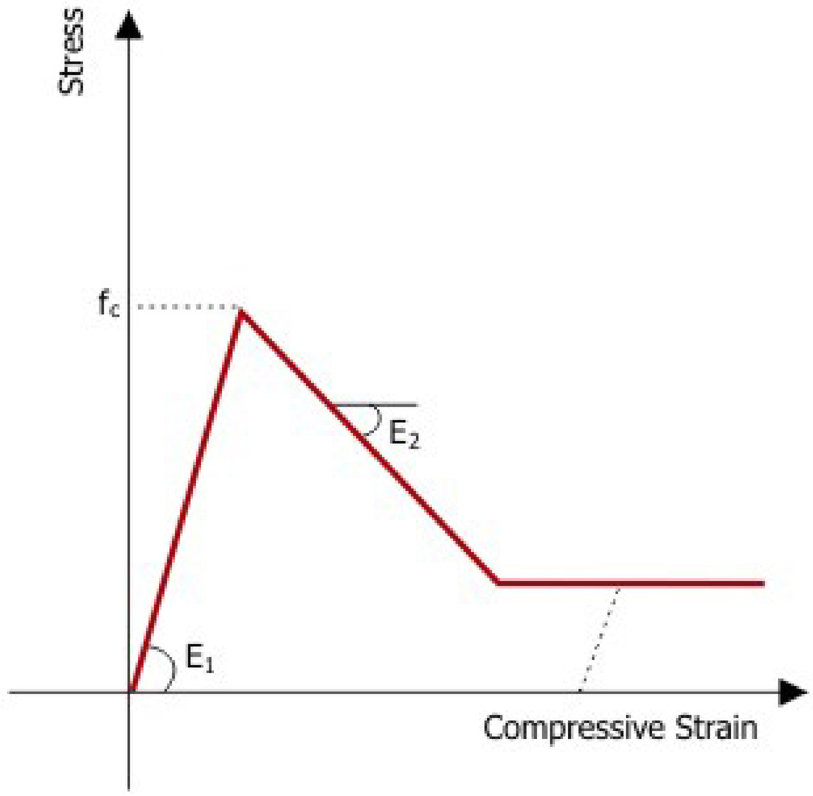
con1

Trilinear concrete model

Number of properties: 4

It is a simplified concrete model for uniaxial modeling.

property	description	typical value
E_1	Initial stiffness	29000
f_{c1}	Compressive strength	20
E_2	Degradation stiffness	-29000
f_{c2}	Residual strength	15



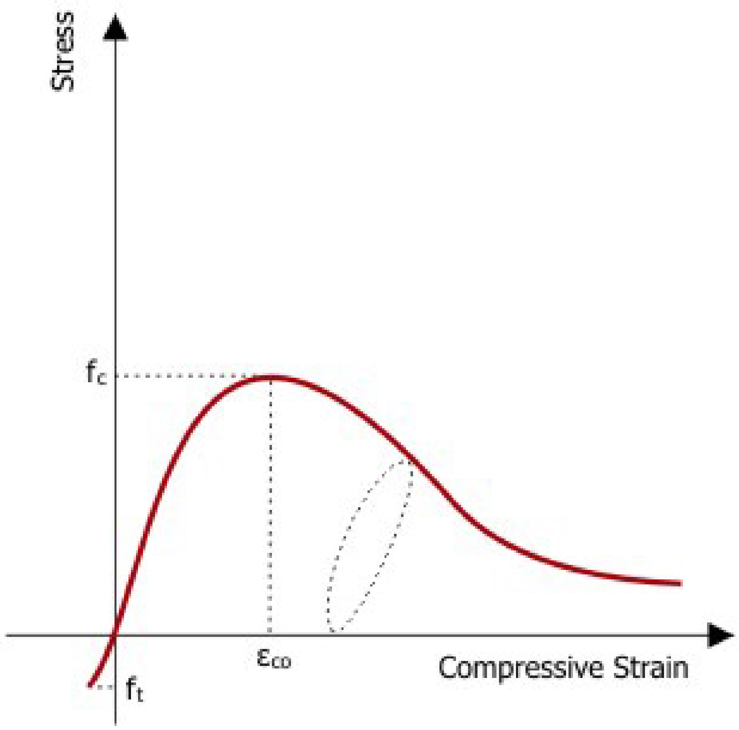
con2

Uniaxial constant confinement concrete model

Number of properties: 4

Uniaxial modeling of concrete assuming constant confinement. It is considerably more accurate than con1.

property	description	typical value
f_c	Compressive strength	20
f_t	Tensile strength	2.2
e_{co}	Crushing strain	0.002
k	Confinement factor	1.2



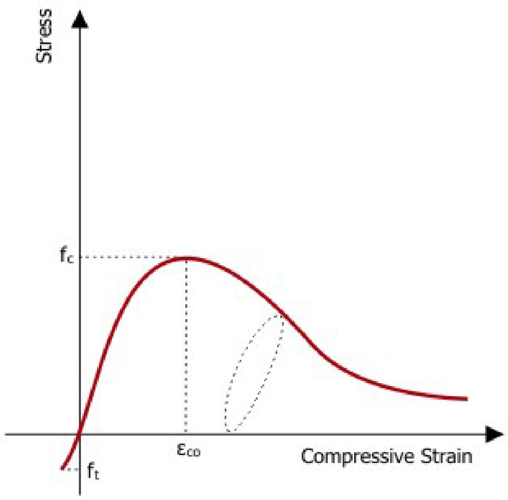
con3

Uniaxial variable confinement concrete model

Number of properties: 10

Uniaxial modeling of concrete. It accounts for the variable confinement effects, which are influenced by the core area within the stirrups, the stirrup size and material and the stirrup spacing.

property	Description	typical value
f_c	Concrete compressive strength	20
f_t	Concrete tensile strength	2.2
e_{co}	Concrete crushing strain	0.002
ν	Poisson's ratio of concrete	0.2
σ_y	Yield stress of stirrups	500
E	The Young's modulus of stirrups	200000
μ	Strain hardening parameter of stirrups	0.005
φ	Diameter of stirrups	10
S	Stirrup spacing	100
Φ_c	Diameter of concrete core	300



con4

Sheikh-Uzumeri nonlinear concrete model

Number of properties: 8 (concrete compressive strength, steel compressive strength, strain corresponding to maximum stress in plain concrete, ratio of the volume of total lateral reinforcement to the volume of core, center-to center distance of outer tie, tie spacing, number of longitudinal bars and area of one longitudinal bar)

This model is applied for the simplified uniaxial concrete model for square sections with uniformly distributed longitudinal steel.

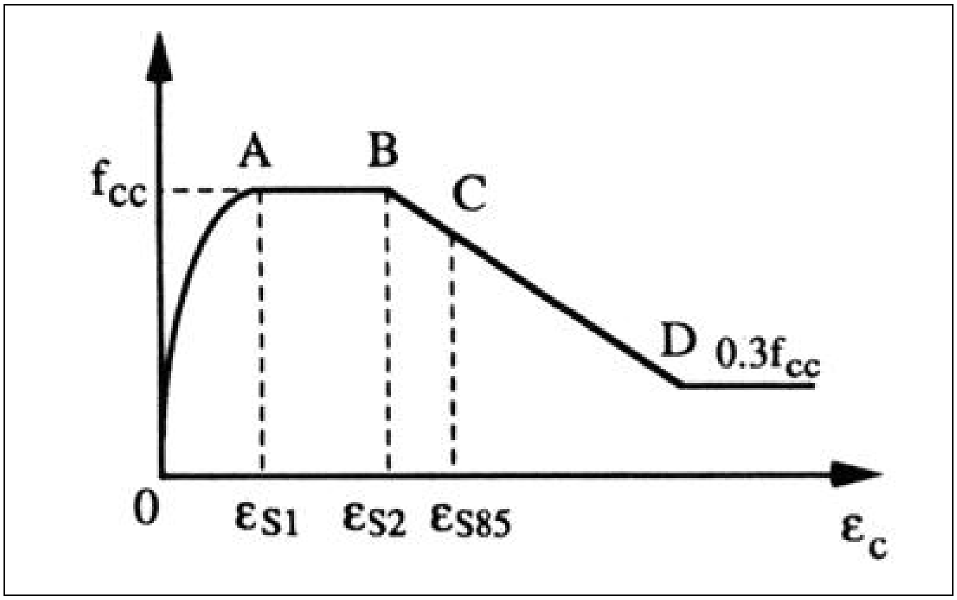
Stress-Strain relationship

<u>OA</u> → $$\sigma = f_{cc} - \frac{f_{cc}}{\varepsilon_{s1}^{\;2}}(\varepsilon - \varepsilon_{s1})^2$$

<u>AB</u> → $$\sigma = f_{cc} \qquad \varepsilon_{s1} \le \varepsilon \le \varepsilon_{s2}$$

<u>BC</u> → $$\sigma = f_{cc} - \frac{0.15 f_{cc}}{\varepsilon_{s85} - \varepsilon_{s2}}(\varepsilon - \varepsilon_{s2})$$

<u>CD</u> → $$\sigma = 0.3 f_{cc}$$

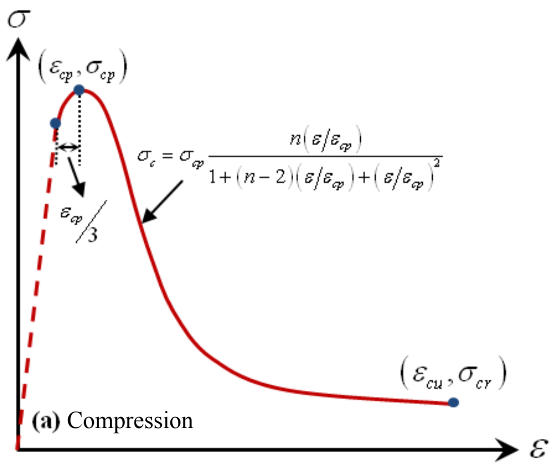


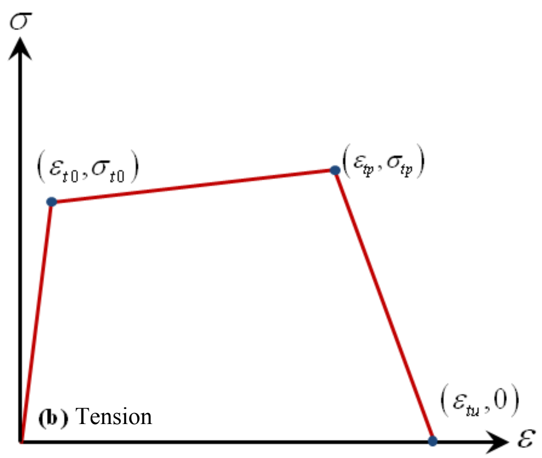
ecc

Model for Engineered Cementitious Composite (ECC) materials

Number of properties: 9

property	Description	typical value
E	Young's modulus	16000
ε_{t0}	First cracking strain	2.5E-4
ε_{tp}	Strain at peak stress in tension	0.038
σ_{tp}	Strength in tension	6
ε_{tu}	Tensile strain capacity	0.06
ε_{cp}	Strain at peak stress in compression ($\varepsilon_{cp} < 0$)	-0.005
σ_{cp}	Strength in compression ($\sigma_{cp} < 0$)	-80
ε_{cu}	Ultimate strain in compression ($\varepsilon_{cu} < 0$) This value should always be less than the maximum compressive strain expected during analysis	-0.012
σ_{cr}	Stress on the compression envelope corresponding to ε_{cu} ($\sigma_{cr} < 0$)	-25



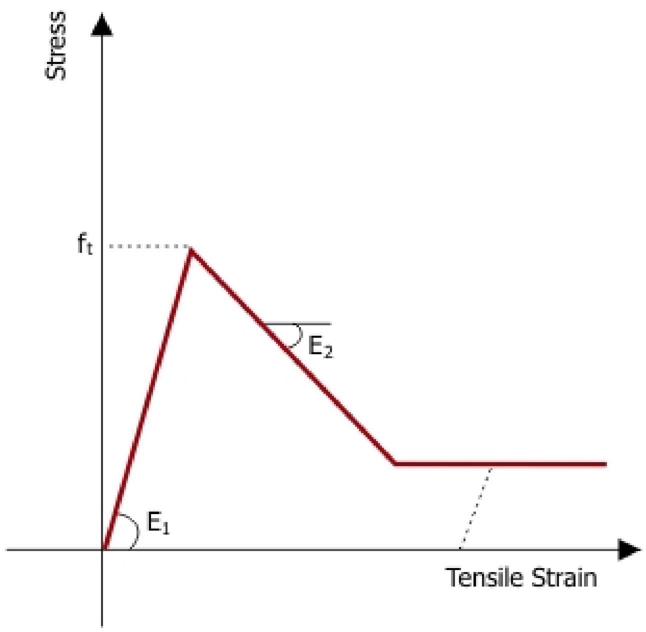


frp1

Trilinear FRP model

Number of properties: 4

property	Description	typical value
E_i	Initial stiffness	65000
σ_t	Tensile strength	6500
E_d	Degradation stiffness	-65000
σ_r	Residual strength	1500



Appendix B- Sections

In this Appendix a list of the available ZeusNL section types is presented:

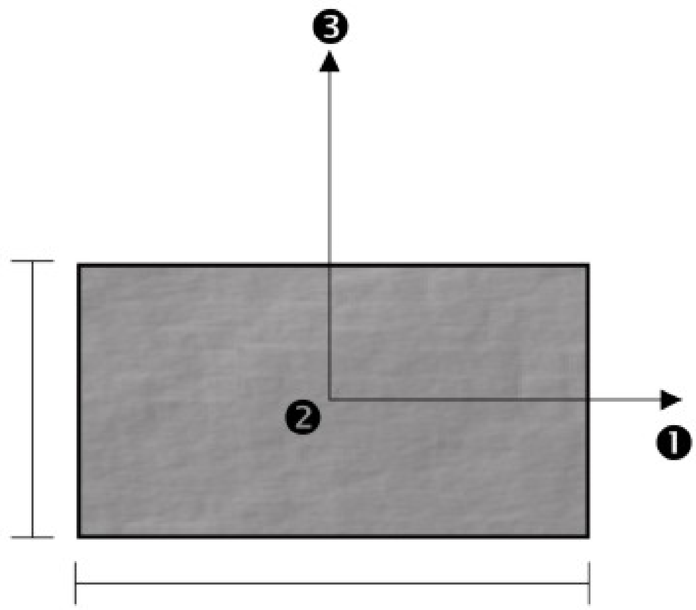
- **rss** Rectangular solid section
- **css** Circular solid section
- **chs** Circular hollow section
- **sits** Symmetric I- or T-section
- **alcs** Asymmetric L- or C-section
- **pecs** Partially encased composite I-section
- **fecs** Fully encased composite I-section
- **rcrs** RC rectangular section
- **rccs** RC circular section
- **rcts** RC T-section
- **rcfws** RC flexural wall section
- **rchrs** RC hollow rectangular section
- **rchcs** RC hollow circular section
- **rcjrs** Reinforce concrete jacket rectangular section

rss

Rectangular solid section

Number of materials : 1

Number of dimensions : 2 (Width, Height)

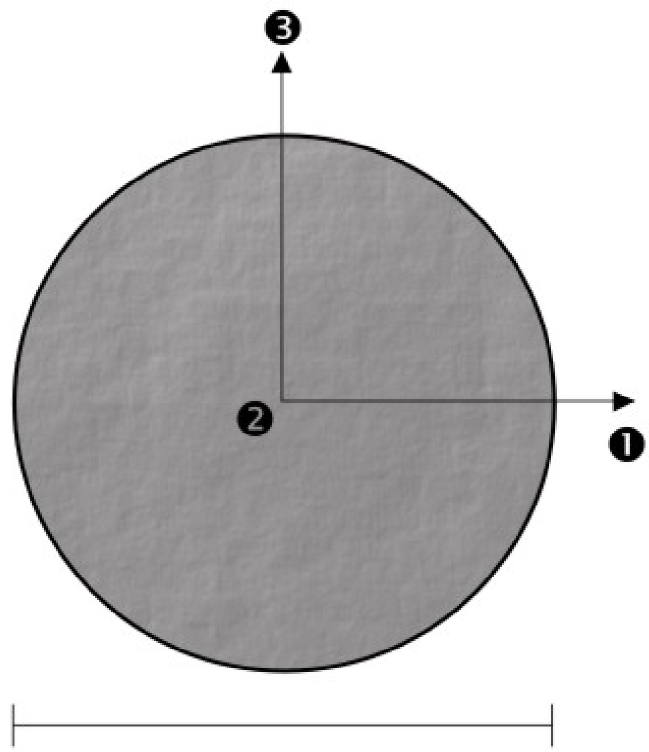


CSS

Circular solid section

Number of materials : 1

Number of dimensions : 1 (Diameter)

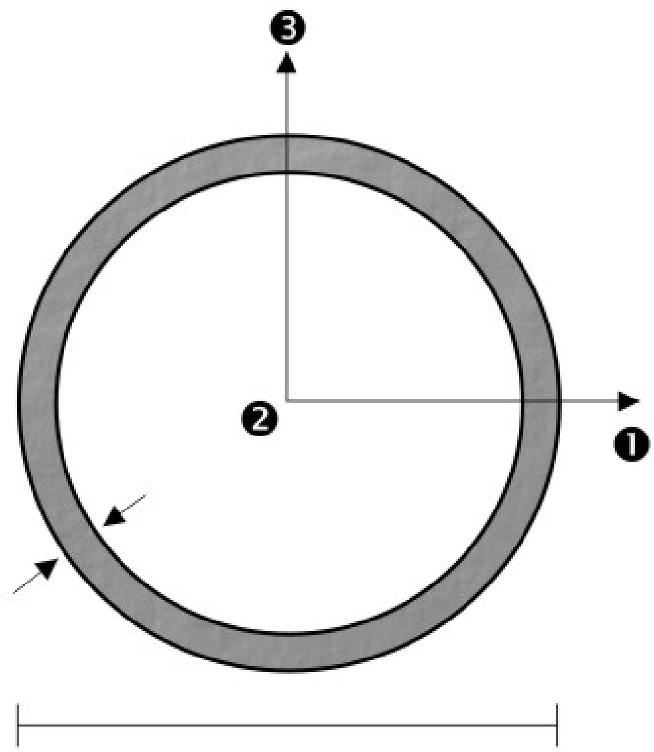


chs

Circular hollow section

Number of materials : 1

Number of dimensions : 2 (Diameter, Thickness)

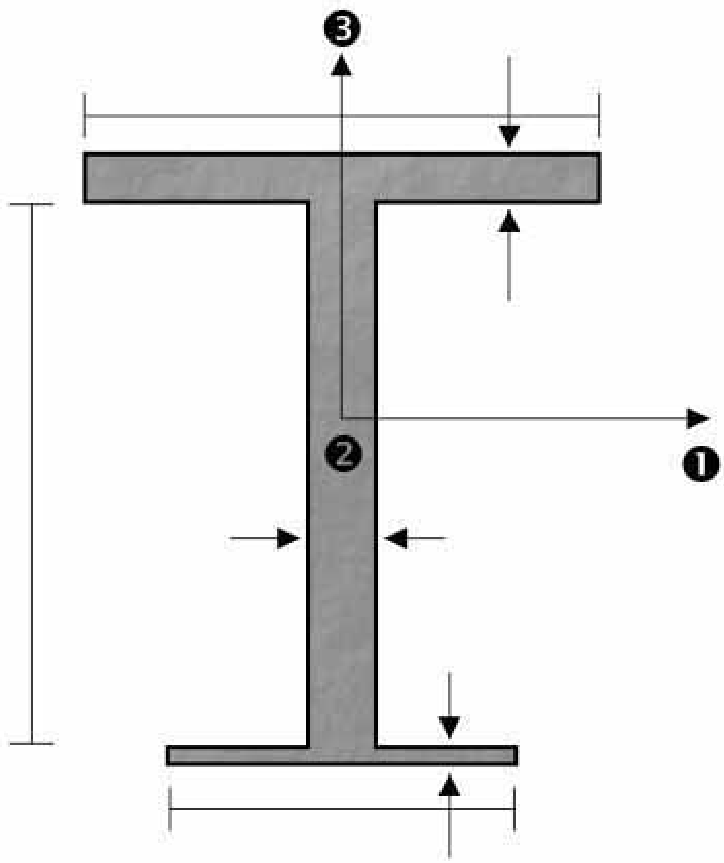


sits

Symmetric I- or T-section

Number of materials : 1

Number of dimensions : 6 (Bottom flange width, Bottom flange thickness, Top flange width, Top flange thickness, Web height, Web thickness)

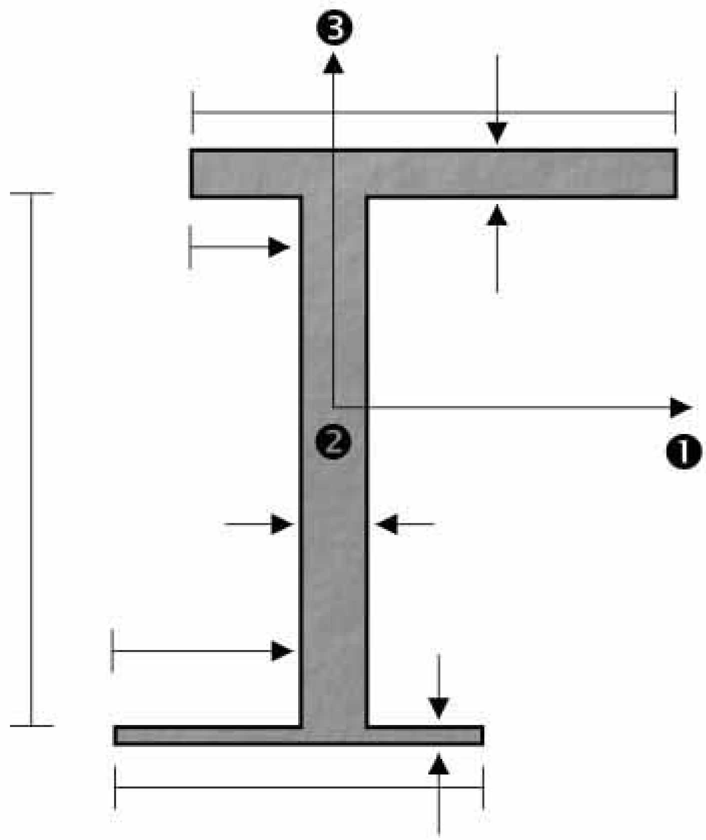


alcs

Asymmetric L- or C-section

Number of materials : 1

Number of dimensions : 8 (Bottom flange width, Bottom flange thickness, Top flange width, Top flange thickness, Web height, Web thickness, Bottom flange eccentricity, Top flange eccentricity)

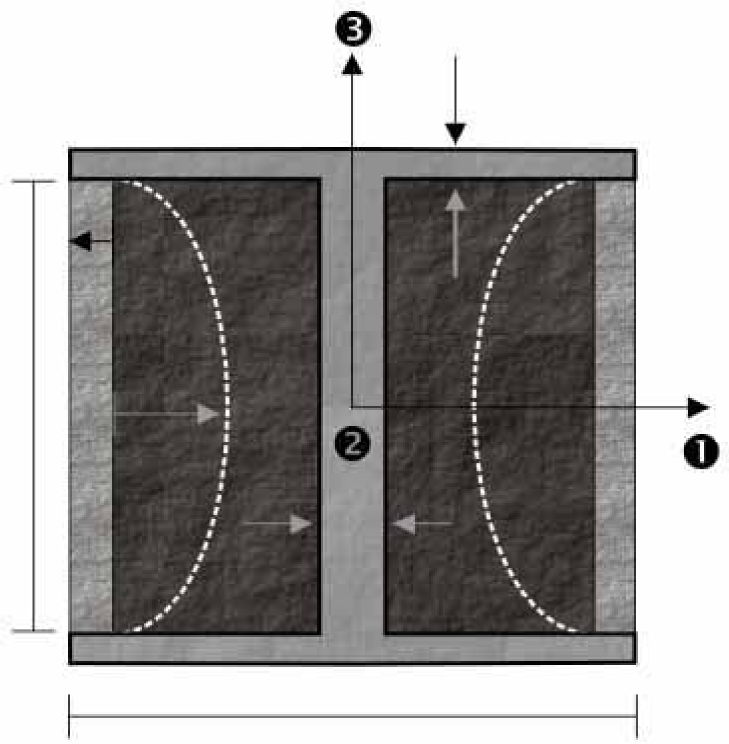


pecs

Partially encased composite I-section

Number of materials : 4 (I-section, Unconfined region, Partially confined region, Fully confined region)

Number of dimensions : 6 (Flange width, Flange thickness, Web height, Web thickness, Unconfined concrete thickness, Max thickness of partially confined concrete)

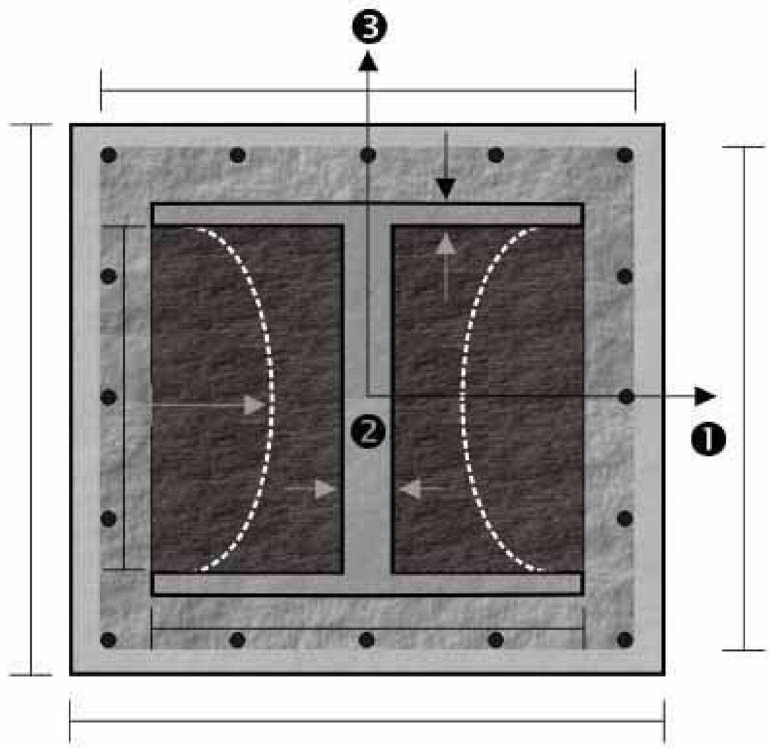


fecs

Fully encased composite I-section

Number of materials : 4 (I-section, Unconfined region, Partially confined region, Fully confined region)

Number of dimensions : 9 (Flange width, Flange thickness, Web height, Web thickness, Max thickness of partially confined concrete, Stirrup width, Section width, Stirrup height, Section height)

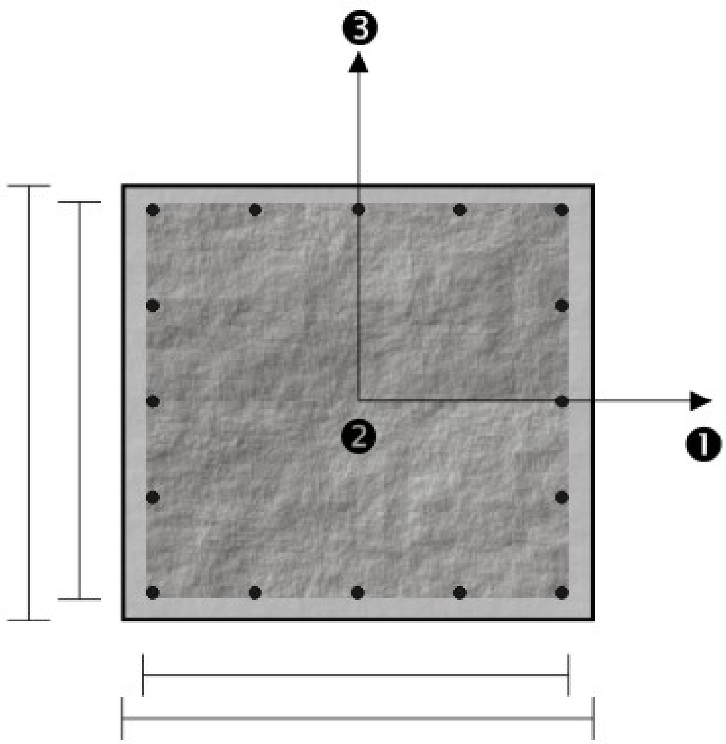


rcrs

RC rectangular section

Number of materials : 3 (Reinforcement, Unconfined region, Confined region)

Number of dimensions : 4 (Section height, Stirrup height, Section width, Stirrup width)

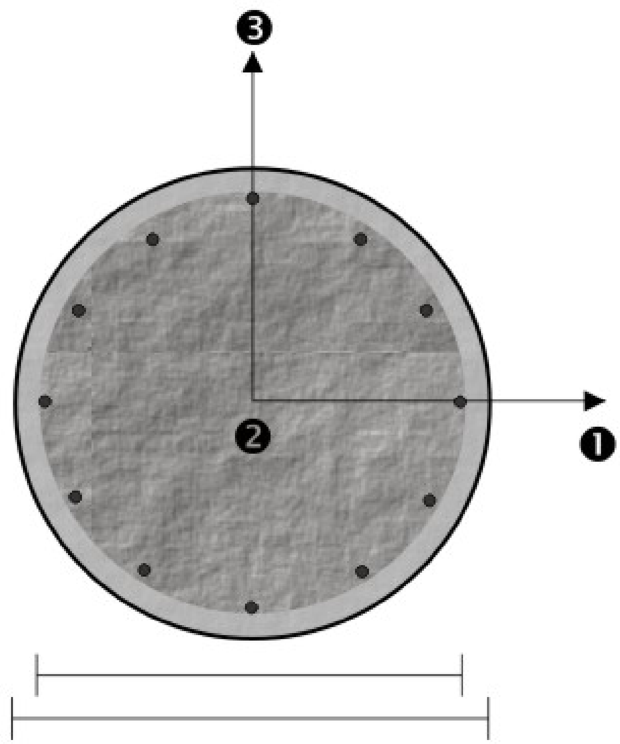


rccs

RC circular section

Number of materials : 3 (Reinforcement, Unconfined region, Confined region)

Number of dimensions : 2 (Section diameter, Stirrup diameter)

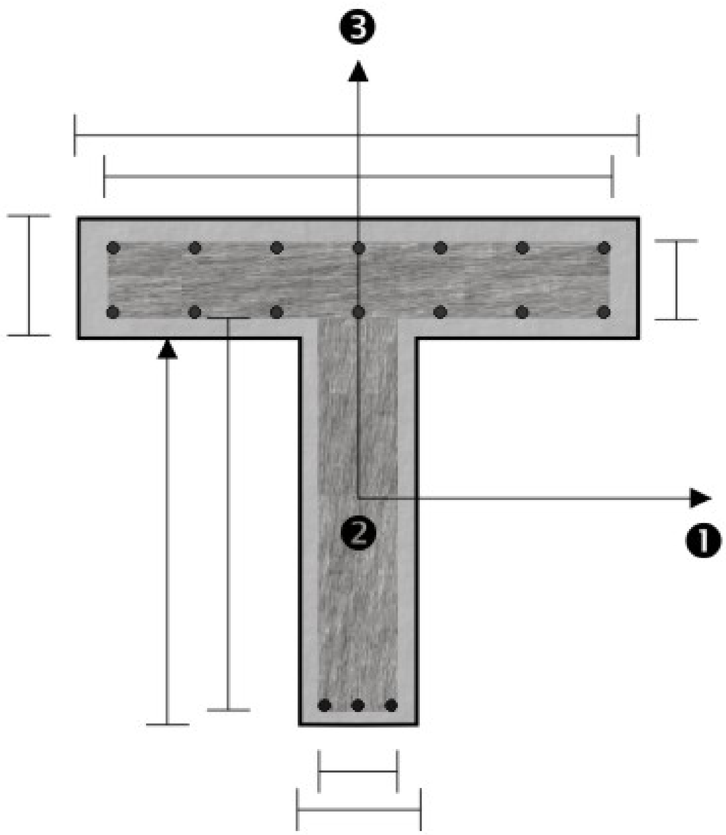


rcts

RC T-section

Number of materials : 3 (Reinforcement, Unconfined region, Confined region)

Number of dimensions : 8 (Slab thickness, Beam height, Confined height in slab, Confined height in beam, Slab effective width, Beam width, Confined width in slab, Confined width in beam)

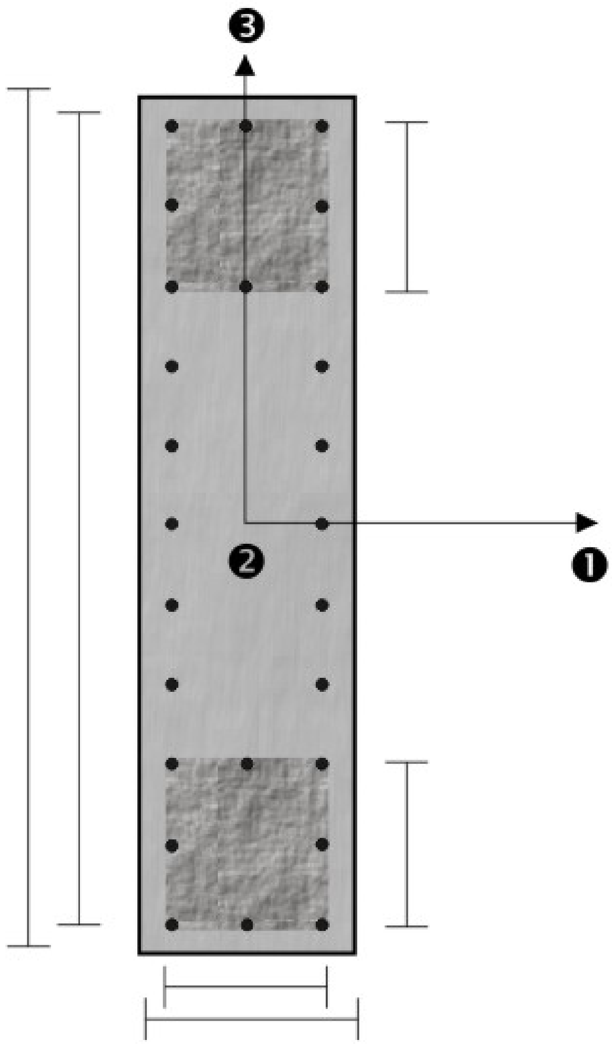


rcfws

RC flexural wall section

Number of materials : 4 (Reinforcement, Unconfined region, Partially confined region, Fully confined region)

Number of dimensions : 5 (Wall width, Confined width, Wall thickness, Confined area thickness, Height of fully confined region)

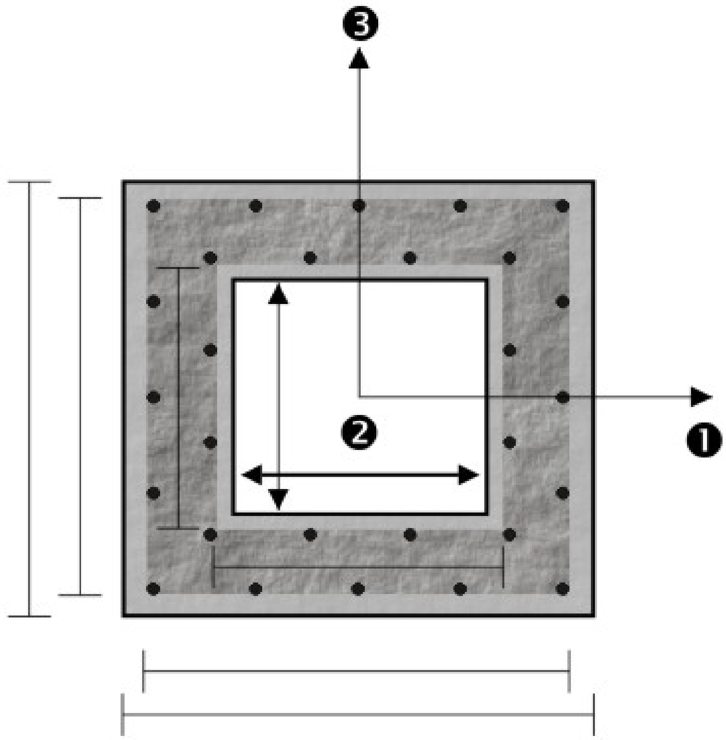


rchrs

RC hollow rectangular section

Number of materials : 3 (Reinforcement, Unconfined region, Confined region)

Number of dimensions : 8 (External section height, External stirrup height, Internal stirrup height, Internal section height, External section width, External stirrup width, Internal stirrup width, Internal section width)

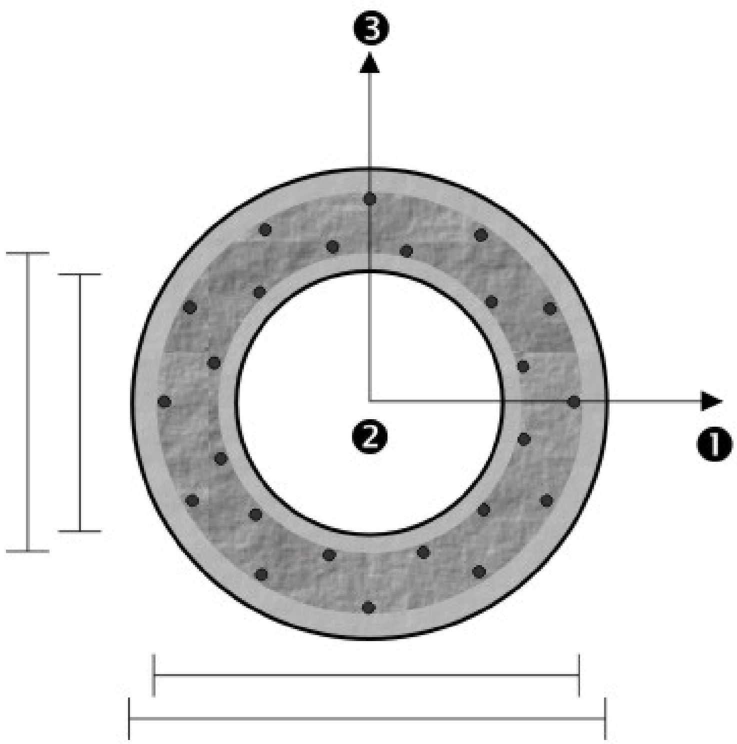


rchcs

Reinforced concrete hollow circular section

Number of materials : 3 (Reinforcement, Unconfined region, Confined region)

Number of dimensions : 4 (External section diameter, External stirrup diameter, Internal stirrup diameter, Internal section diameter)

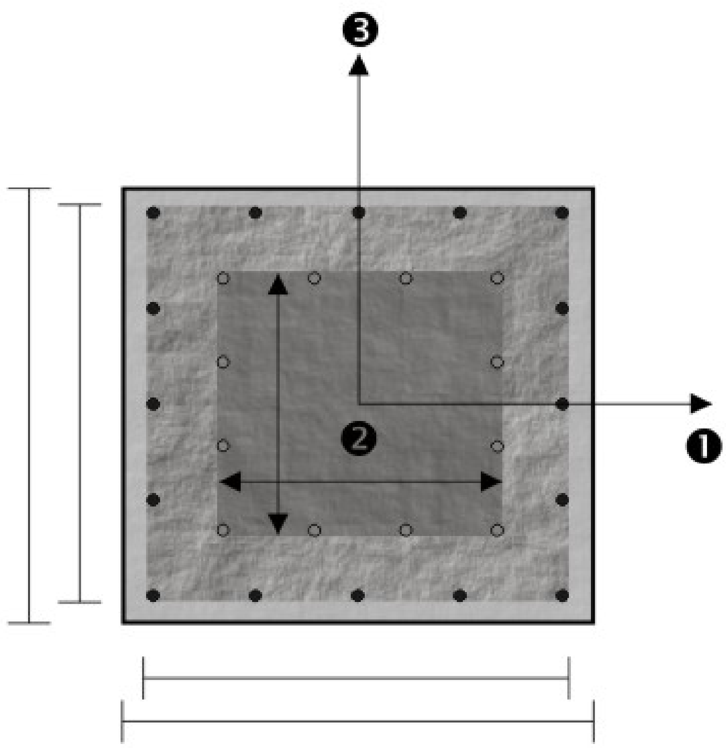


rcjrs

Reinforce concrete jacket rectangular section

Number of materials : 4 (Reinforcement, Unconfined region, Partially confined region, Fully confined region)

Number of dimensions : 6 (Section height, External stirrup height, Internal stirrup height, Section width, External stirrup width, Internal stirrup width)



Appendix C - Elements

The ZeusNL elements library contains a set of elements used to model the elasto-plastic structural behavior, the boundary conditions and the dynamic characteristics of the models:

- **Cubic.** Cubic elasto-plastic 3D beam-column element. It is used for detailed elasto-plastic modeling, making use of the available uniaxial inelastic material models. In accounts for the spread of inelasticity along the members' length and across the section depth, by dividing the cross-section at the two Gauss points in a number of monitoring areas to which the material models apply.
- **Joint.** 3D joint element with uncoupled axial, shear and moment actions. The joint element is used to model pin joints, inclined supports, elasto-plastic joint behavior, soil-structure interaction and structural gaps, through employing appropriate joint curves.
- **Lmass.** Lumped (concentrated) mass element. It models lumped masses. It is used in dynamic and eigenvalue analysis.
- **Dmass.** Cubic distributed mass element. It models uniformly distributed mass for dynamic and eigenvalue analysis.
- **Ddamp.** Dashpot (concentrated) viscous damping element.
- **Rdamp.** Rayleigh damping element. It models Rayleigh damping effects in dynamic analysis of space frames.

Cubic

Cubic elasto-plastic 3D beam-column element

Number of nodes: 3

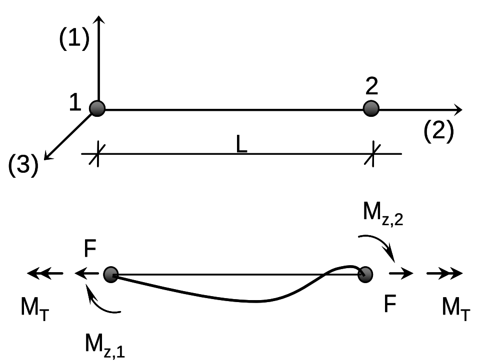
The cubic-elasto-plastic element can adequately model members of space frames with geometrical and material nonlinearities.

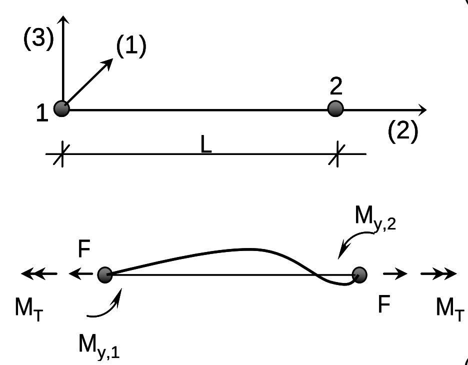
For the evaluation of the element forces, numerical integration is performed at the two Gauss points. For this purpose, the section at each Gauss point is divided into a number of monitoring points (monitoring areas); the stress-strain relations of which are considered during the integration. For single-material sections (sits, rss), 100 monitoring points are usually enough. For more complicated sections (fecs, rcts, rcfws), this number should be increased to about 200 or more.

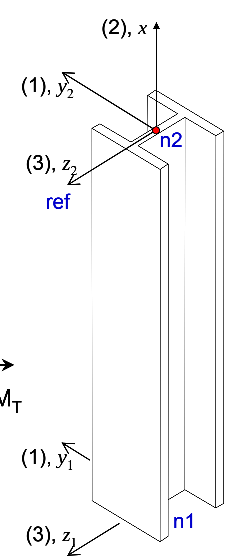
For accurate inelastic modeling, it is advisable to use more than one cubic element per member.

Nodes (1) and (2) are the end-nodes of the element. The element local x-axis lies on the line defined by them.

> Node (3) is required to define the (local) x-y plane and can be a non-structural node. It is possible (and advisable) to use one non-structural as the third node for all the cubic elements that lie on the same plane of the model.







Joint

3D joint element with uncoupled axial, shear and moment actions

Number of nodes: 4

The joint element is used in space frame analysis to model pin joints, inclined supports, elasto-plastic joint behavior, soil-structure interaction and structural gaps through employing appropriate joint curves.

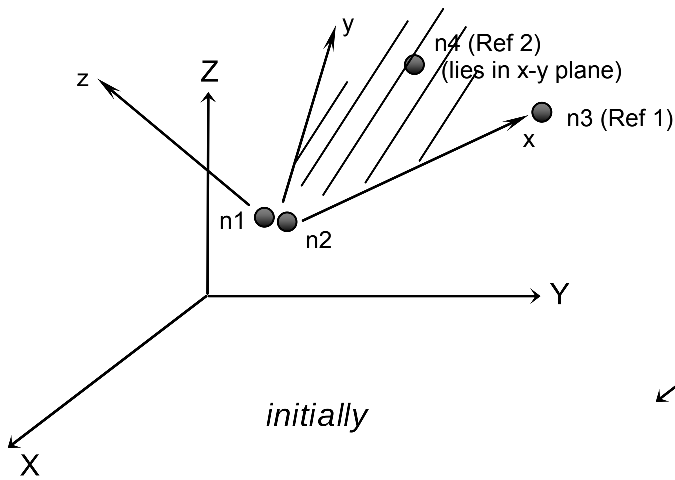
For the complete definition of joint, four nodes are required. Nodes 1 and 2 are the end-nodes of the element and must be initially coincident. Node 3 is only used to define the x-axis of the joint and can be either a structural or a non-structural node.

Node 4 is required to define (together with the already defined x-axis) the x-y plane and can be a non-structural node. After deformation, the orientation of the joint x-axis is determined by its initial orientation and the global rotations of node 1.

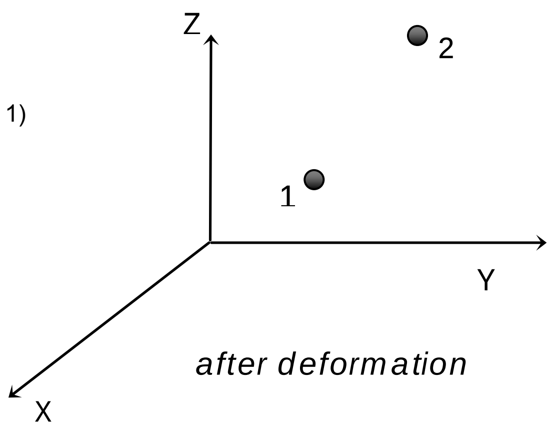
The force-displacement characteristics for the axial F_x, the shear forces F_y and F_z and the moments M_x, M_y and M_z, are determined by curves included in ZeusNL libraries (lin, smtr, astr).

The input parameters are a list of parameters required for the definition of the curves and should be given in the following order: F_x – F_y - F_z - M_x - M_y - M_z

Element has a zero initial length since nodes 1 and 2 are coincident. The joint element cannot be used to model coupled axial, shear and moment actions.



initially



after deformation

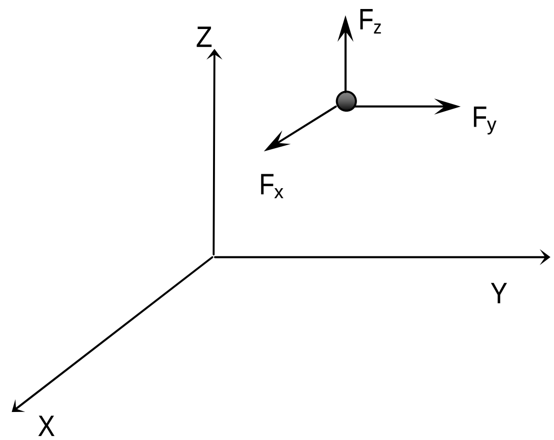
Lmass

Lumped (concentrated) mass element

Number of nodes: 1

Lmass models lumped masses. It is used in dynamic and eigenvalue analysis.

Mass units should be N/(mm/sec2)



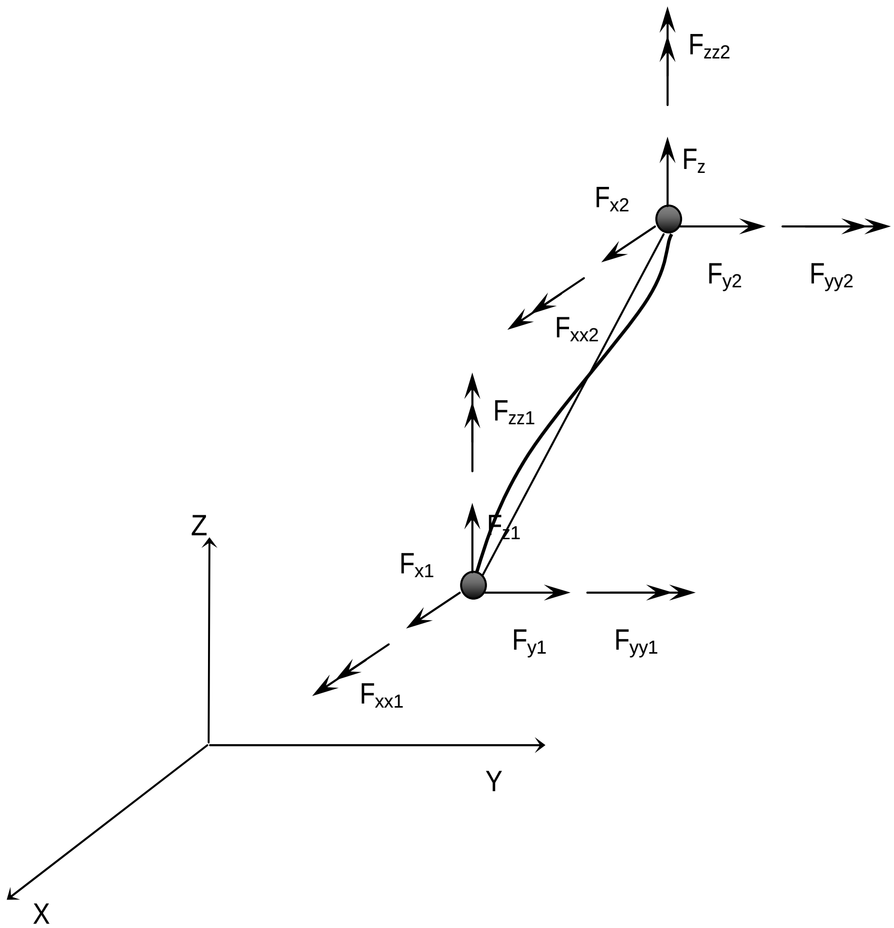
Dmass

Cubic distributed mass element

Number of nodes: 2

Dmass models uniformly distributed mass for dynamic and eigenvalue analysis. It uses an updated Lagrangian formulation and a cubic shape function for the transverse displacement and a linear distribution for the axial displacement.

The mass/length units should be in N/(mm/sec2)/mm



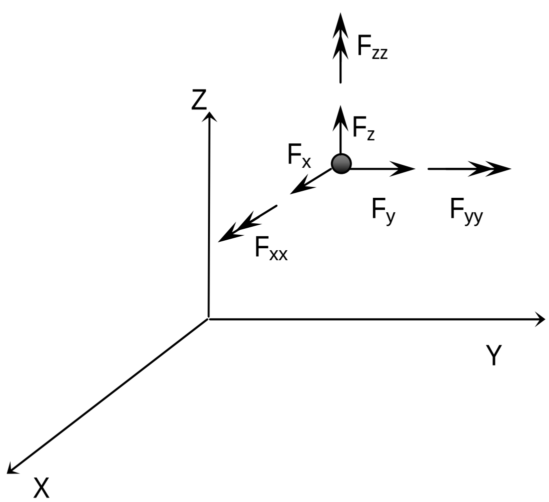
Ddamp

Dashpot (concentrated) viscous damping element

Number of nodes: 1

Six (three translational and three rotational) parameters must be specified: C_x, C_y, C_z, C_{xx}, C_{yy} and C_{zz}.

The elements models nodal viscous damping in dynamic analysis.



Rdamp

Rayleigh damping element

Number of nodes: 3

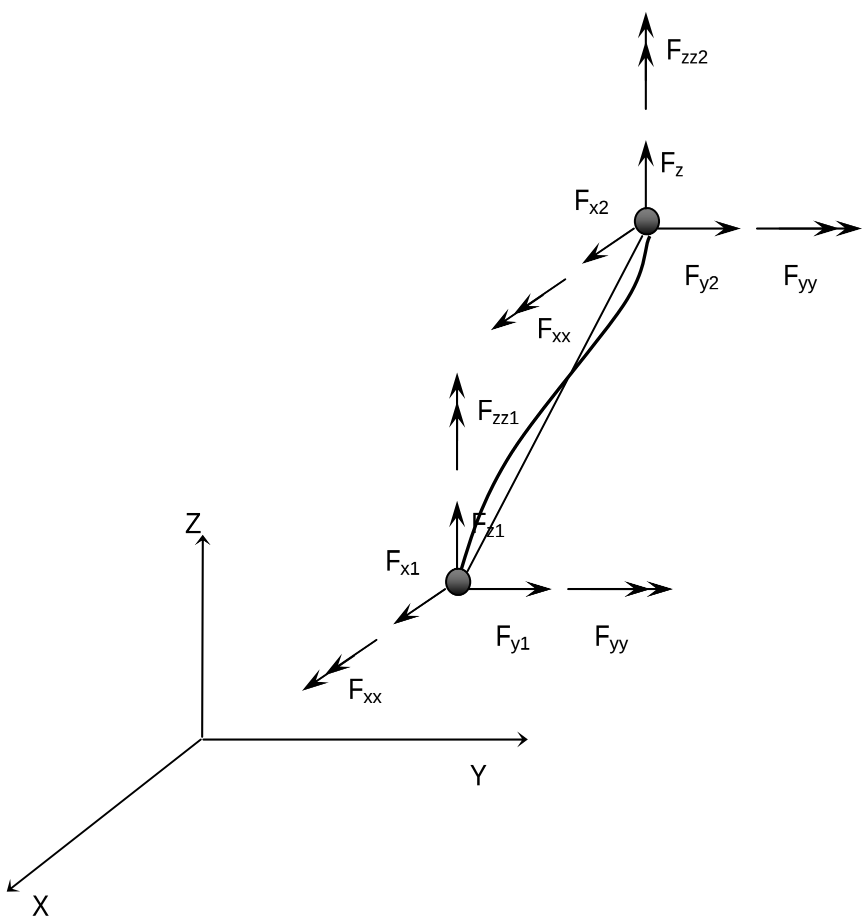
Rdamp models Rayleigh damping effects in dynamic analysis of space frames.

Two parameters must be given: the proportionality constants (al and a2) of mass and stiffness respectively.

Nodes 1 and 2 define the element connectivity and its local x- axis. Node 3 is required to define the x-y plane and can be a non-structural node.

a1 should be set to zero for dynamic analysis involving ground excitation, otherwise damping would be proportional to absolute rather than relative frame velocity.

All rdamp elements must have the same constant (al and a2) to model conventional Rayleigh damping.



Appendix D – joint curves

Appendix D describes the force-displacement curves available to be used with the joint element:

- **lin** Elastic linear curve
- **smtr** Tri-linear symmetrical elasto-plastic curve
- **astr** Tri-linear asymmetric elasto-plastic curve
- **hsc** Hysteretic shear model under constant axial force
- **hsv** Hysteretic shear model under axial force variation
- **hfc** Hysteretic flexure model under constant axial force
- **hfv** Hysteretic flexure model under axial force variation

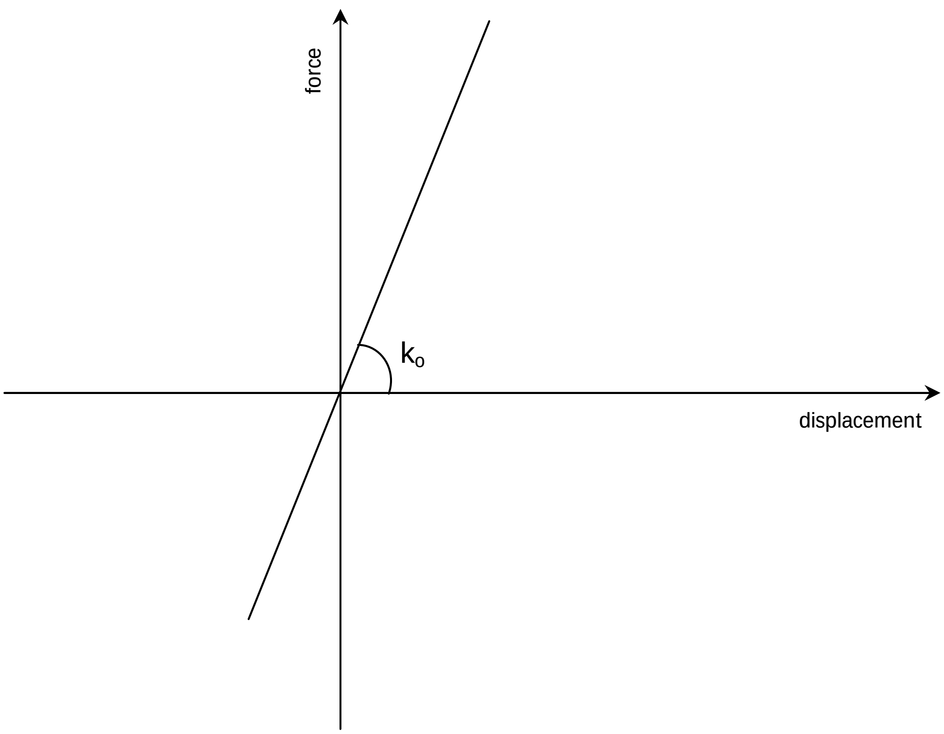
lin

Elastic linear curve

Number of parameters: 1

This curve describes the elastic joint action characteristics.

parameter	description	typical value
k_o	Stiffness	1e5



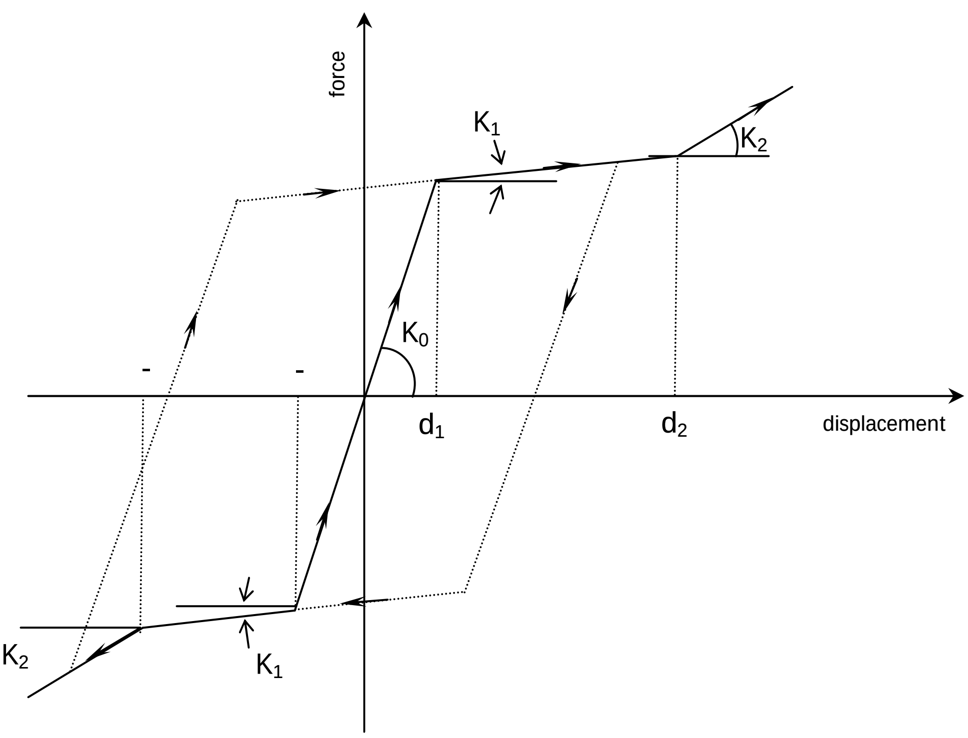
smtr

Tri-linear symmetrical elasto-plastic curve type

Number of parameters: 5

It is a typical tri-linear symmetrical elasto-plastic curve used to model the elasto-plastic joint action. Unloading is done kinematically to the extension of the second branch of the curve. The stiffnesses K_0, K_1 and K_2 must be positive, whereas K_1 and K_2 should be less than K_0.

parameter	Description	typical value
K_0	Initial stiffness	1e5
d_1	Displacement where the stiffness changes from K_0 to K_1	1
K_1	Stiffness of second branch	10
d_2	Displacement where the stiffness changes from K_1 to K_2	50
K_2	Stiffness of third branch	100



astr

Tri-linear asymmetric elasto-plastic curve type

Number of parameters: 10 (=2*5)

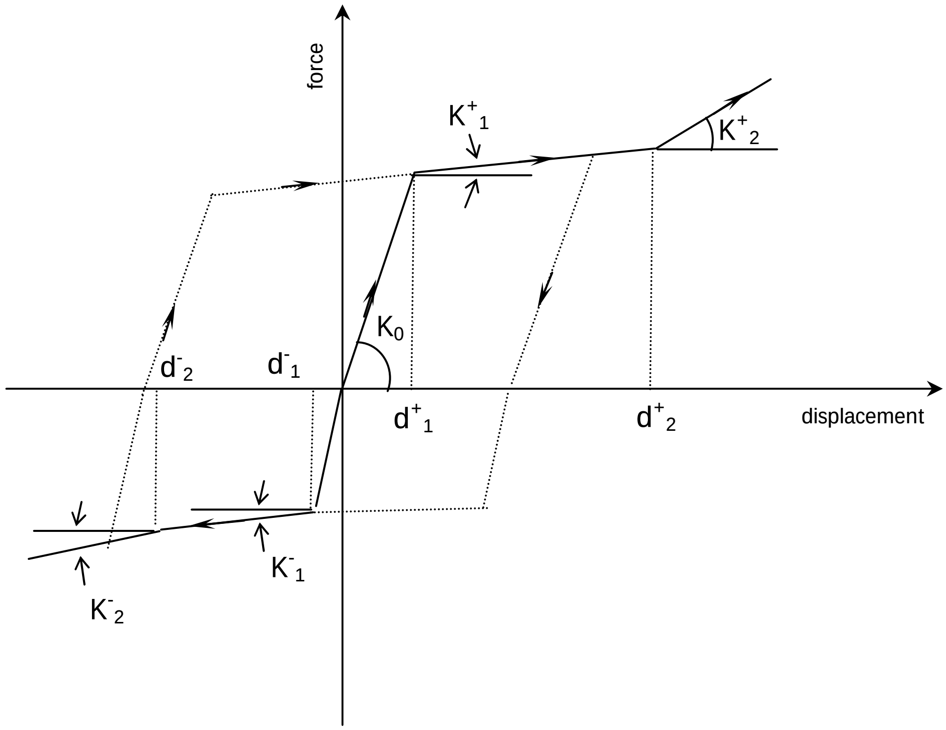
It is similar to the smtr tri-linear elasto-plastic curve but it is asymmetric. Hence, 10 parameters are required for the complete description of the curve.

Unloading is done kinematically to the extension of the second branch of the curve.

All the stiffnesses K^+_0, K^+_1, K^+_2 and K^-_0, K^-_1, K^-_2 must be positive and K_1 and K2 should be less than K_0, both for the positive and negative displacement region.

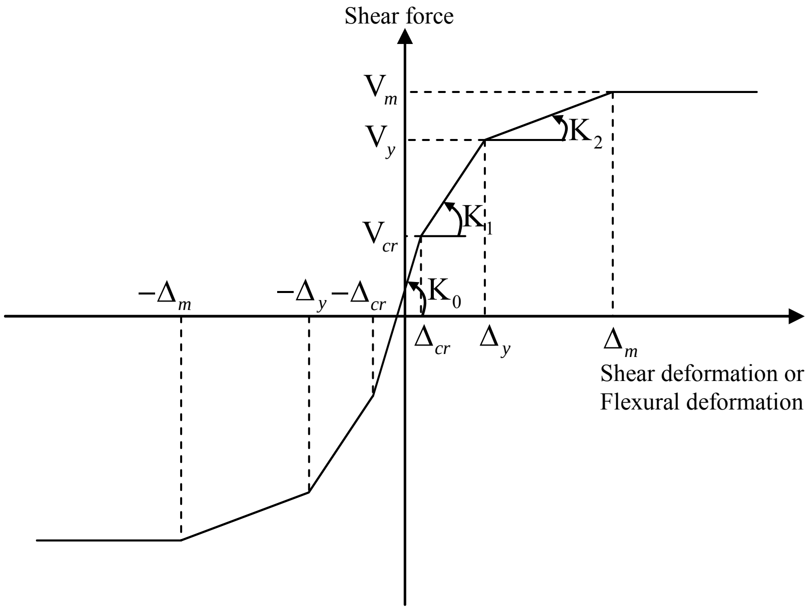
The curve models the elasto-plastic joint action and, because it is asymmetric, it can also model structural gaps.

parameter	Description	typical value
K^+_0	Initial stiffness (positive displacement region)	1e5
d^+_1	Positive displacement where the stiffness changes from K^+_0 to K^+_1	1
K^+_1	Stiffness of second branch (positive displacement region)	10
d^+_2	Positive displacement where the stiffness changes from K^+_1 to K^+_2	50
K^+_2	Stiffness of third branch (positive displacement region)	100
K^-_0	Initial stiffness (negative displacement region)	1e5
d^-_1	Negative displacement where the stiffness changes from K^-_0 to K^-_1	-1
K^-_1	Stiffness of second branch (negative displacement region)	10
d^-_2	Negative displacement where the stiffness changes from K^-_1 to K^-_2	-25
K^-_2	Stiffness of third branch (negative displacement region)	50


force
displacement
K+1
K+2
K0
d-2
d-1
d+1
d+2
K-1
K-2

hsc

Hysteretic shear model under constant axial force



Initial shear stiffness: K_0

Shear displacement at cracking: Δ_{cr}

Shear force at cracking: V_{cr}

Shear stiffness after cracking: K_1

Shear displacement at yielding: Δ_y

Shear force at yielding: V_y

Shear stiffness after yielding: K_2

Shear displacement at ultimate: Δ_m

Shear force at ultimate: V_m

Shear stiffness after ultimate: 0.0

Applied axial force: Whereas compressive axial force is negative, tensile axial force is positive

Compressive axial force capacity: Axial capacity based on ACI 318(should be negative)

hsv

Hysteretic shear model under axial force variation

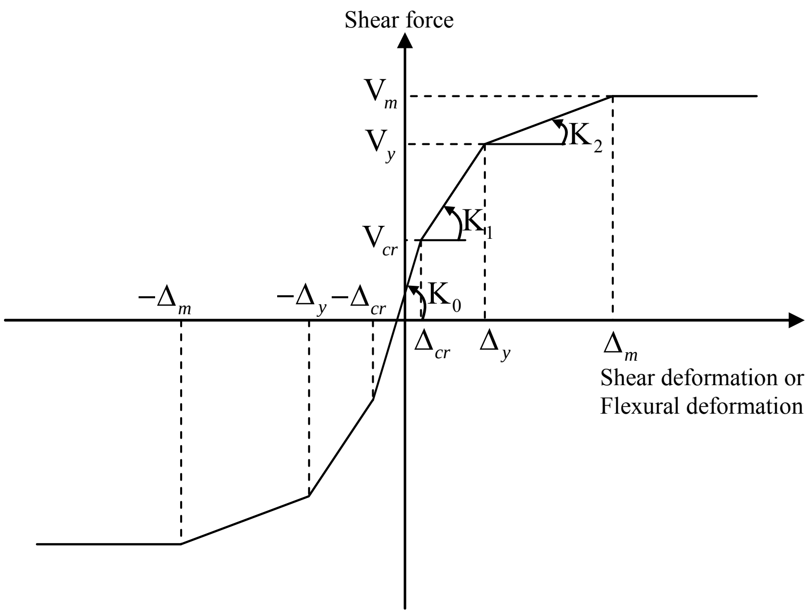
Parameters required for the above ZeusNL curve type is described as below from left column to right column.

1. The first column represents the level of axial force of interest (zero axial force level, three levels of compressive axial force and two levels of tensile axial force) which can be defined by user. Say for instance, if you choose 10%, 20%, and 30% of axial capacity in compression, and 10% and 30% of axial capacity in tension, you can define parameters in the first column as 0.0 0.1, 0.2, 0.3, 0.1 and 0.3 from top to bottom. Hence, level of axial force of interest can freely be defined by user.
2. The first row of the second column is an identifier of each hysteretic curve. The 'curve number' stands for a numbering in sequence in order to trace each hysteretic curve assigned to each direction of a member. For example, if a bridge structure has three piers, user has to define six curves (three in longitudinal and three in transverse direction). In this case, user can define the 'curve number' as 1 to 6 corresponding to each direction of each pier. The second and the third row of the second column represent the axial force capacity in compression and in tension, respectively (these values should be positive)
3. The third column represents shear displacement at cracking corresponding to each level of axial force defined in the first column.
4. The fourth column represents shear displacement at yielding corresponding to each level of axial force defined in the first column.
5. The fifth column represents shear displacement at ultimate corresponding to each level of axial force defined in the first column.
6. The sixth column represents shear force at crack corresponding to each level of axial force defined in the first column.
7. The seventh column represents shear force at yielding corresponding to each level of axial force defined in the first column.
8. The eighth column represents shear force at ultimate corresponding to each level of axial force defined in the first column.

* For the parameters of the curve type 'hsv', monotonic shear force-shear displacement curve subjected to each level of axial force defined has to be evaluated in advance.

hfc

Hysteretic flexure model under constant axial force



Initial flexural stiffness: K_0

Flexural displacement at cracking: Δ_{cr}

Shear force at cracking: V_{cr}

Flexural stiffness after cracking: K_1

Flexural displacement at yielding: Δ_y

Shear force at yielding: V_y

Flexural stiffness after yielding: K_2

Flexural displacement at ultimate: Δ_m

Shear force at ultimate: V_m

Flexural stiffness after ultimate: 0.0

hfv

Hysteretic flexure model under axial force variation

Parameters required for the above ZeusNL curve type is described as below from left column to right column.

1. The first column represents the level of axial force of interest (zero axial force level, three levels of compressive axial force and two levels of tensile axial force) which can be defined by user. Say for instance, if you choose 10%, 20%, and 30% of axial capacity in compression, and 10% and 30% of axial capacity in tension, you can define parameters in the first column as 0.0 0.1, 0.2, 0.3, 0.1 and 0.3 from top to bottom. Hence, level of axial force of interest can freely be defined by user.
2. The first row of the second column is an identifier of each hysteretic curve. The 'curve number' stands for a numbering in sequence in order to trace each hysteretic curve assigned to each direction of a member. For example, if a bridge structure has three piers, user has to define six curves (three in longitudinal and three in transverse direction). In this case, user can define the 'curve number' as 1 to 6 corresponding to each direction of each pier. The second and the third row of the second column represent the axial force capacity in compression and in tension, respectively (these values should be positive)
3. The third column represents Flexural displacement at cracking corresponding to each level of axial force defined in the first column.
4. The fourth column represents Flexural displacement at yielding corresponding to each level of axial force defined in the first column.
5. The fifth column represents Flexural displacement at ultimate corresponding to each level of axial force defined in the first column.
6. The sixth column represents shear force at crack corresponding to each level of axial force defined in the first column.
7. The seventh column represents shear force at yielding corresponding to each level of axial force defined in the first column.
8. The eighth column represents shear force at ultimate corresponding to each level of axial force defined in the first column.

* For the parameters of the curve type 'hfv', monotonic shear force-flexural displacement curve subjected to each level of axial force defined has to be evaluated in advance.

Appendix E – Local and Global axes

Appendix E is a discussion about the extra node for defining the orientation of the element. Examples will clarify how the section local axes 1-2-3 relate to the global axes X-Y-Z and depict the details that one should pay attention to, in order to have a correct modeling.

Assume there is a simple beam with a T-section (Fig.E1) that the user tries to model. Obviously, the user will create two new nodes n1 and n2 that define its end-nodes. However, n1 and n2 give no information for the T-section and its orientation (Fig.E2).

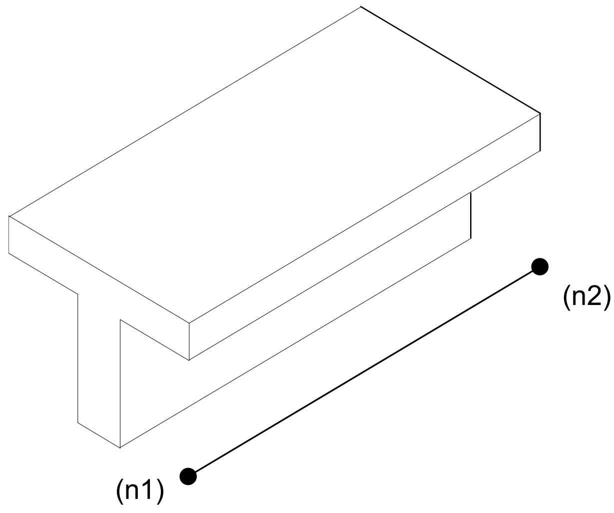


Fig.E1 Beam with a T-section to be modeled.

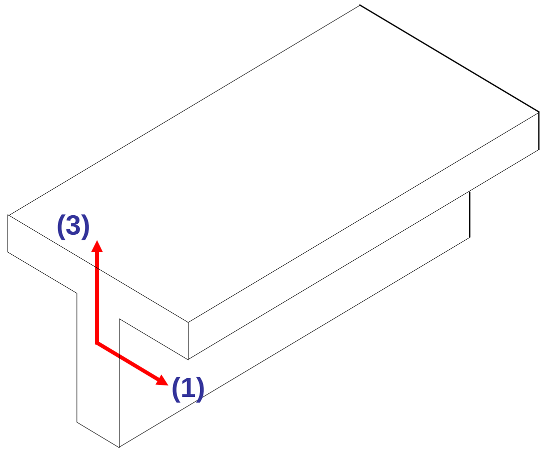


Fig.E2 Orientations of the T-section of the beam.

The third node of the element n3 serves exactly this purpose: to define the orientation of the section. Depending on the position of n3, the three nodes define one plane (Fig.E3). The rule is that the strong axis of the section (3) should lie within this plane.

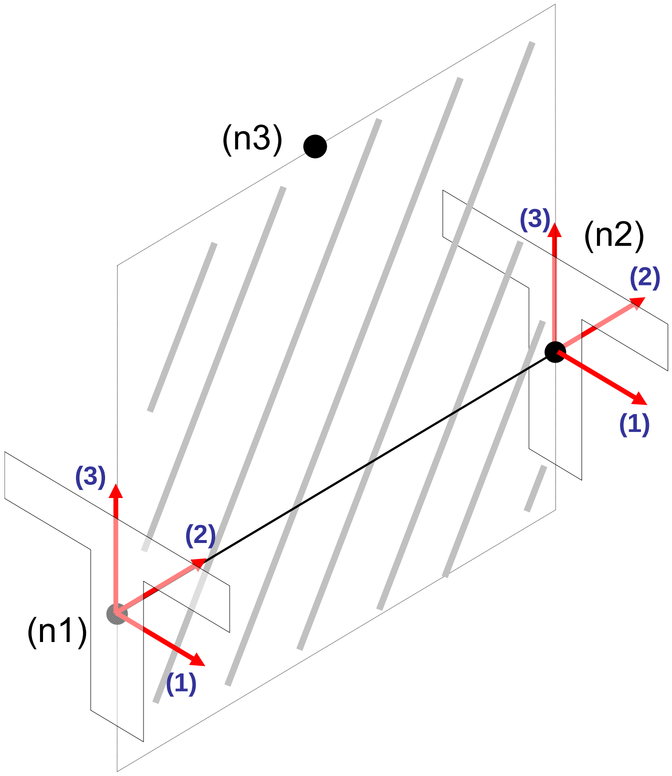
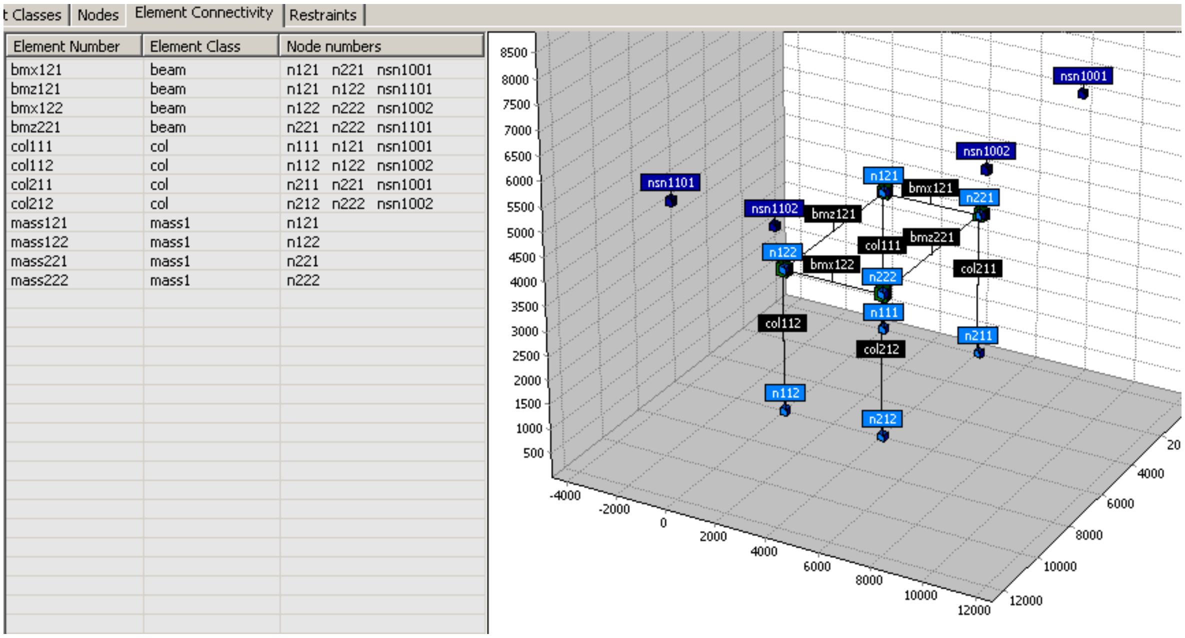


Fig.E3 The correct position of node n3 for modeling the orientations of Fig.E2.

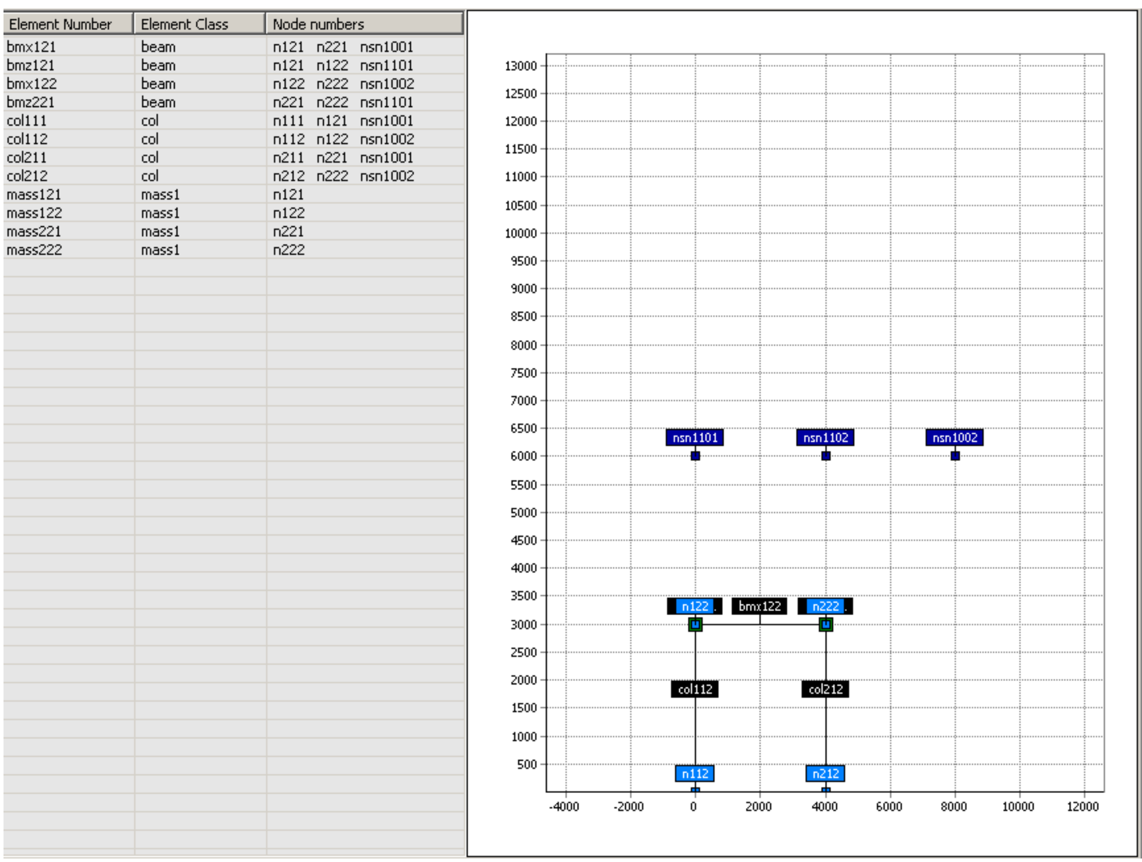
In practice, the user should follow this rule of thumb: The vast majority of modeled structures are formed in plane frames. For every frame in the x-y or z-y planes, the user should create a non-structural node that is not in the same line with any of the elements on the frame. A good tactic would be to place the non-structural node outside the limits of the structure. The third node of every element of this frame will be this non-structural node.

Good examples of this tactic are the models derived with the template, as in Fig.E4. The elements of the front frame, namely col111, bmx121 and col211, all use nsn1001 as the third node. In the same way, the elements of the back frame (col112, bmx122, col212), use nsn1002. The beams in the z-direction, bmz121 and bmz221, use ns1101 and ns1102, respectively. In the special case of 2D analysis, only one non-structural node is required, as in Fig.E5.



Element Number	Element Class	Node numbers
bmx121	beam	n121 n221 nsn1001
bmz121	beam	n121 n122 nsn1101
bmx122	beam	n122 n222 nsn1002
bmz221	beam	n221 n222 nsn1101
col111	col	n111 n121 nsn1001
col112	col	n112 n122 nsn1002
col211	col	n211 n221 nsn1001
col212	col	n212 n222 nsn1002
mass121	mass1	n121
mass122	mass1	n122
mass221	mass1	n221
mass222	mass1	n222

Fig.E4 Example of the use of non-structural nodes.



Element Number	Element Class	Node numbers
bmx121	beam	n121 n221 nsn1001
bmz121	beam	n121 n122 nsn1101
bmx122	beam	n122 n222 nsn1002
bmz221	beam	n221 n222 nsn1101
col111	col	n111 n121 nsn1001
col112	col	n112 n122 nsn1002
col211	col	n211 n221 nsn1001
col212	col	n212 n222 nsn1002
mass121	mass1	n121
mass122	mass1	n122
mass221	mass1	n221
mass222	mass1	n222

Fig.E5 Example of the use of one non-structural node in 2D analysis.

Appendix F – The **ZBeer** Utility

Does the idea of running hundreds of dynamic or static pushover analyses by the press of one button sound appealing ? But what happens with the gigabytes of outcoming results, when only a few response parameters are of importance ? Shouldn't they be automatically filtered, calculated and stored, ready for plotting ? If these all sound visionary, the ZBeer utility brings them to reality, uncovering free time and ... justifying its name !

Originally developed under DOS and Linux platforms to interact with the ancestors of ZeusNL, ADAPTIC and INDYAS, the ZBeer utility has been written from scratch for Windows, featuring automatic running of static pushover and dynamic pushover analysis (also referred as Incremental Dynamic Analysis - IDA), for multiple files and various response monitoring parameters.

F.1 Overview

The ZBeer utility is activated by **Tools > ZBeer** in the main ZeusNL window, or by pressing its associated button in the Run/Tools toolbar. Figure F1 shows the main window of ZBeer. It is split into three regions, the Input region at the top left, the Status region at the top right and the Chart region at the bottom.

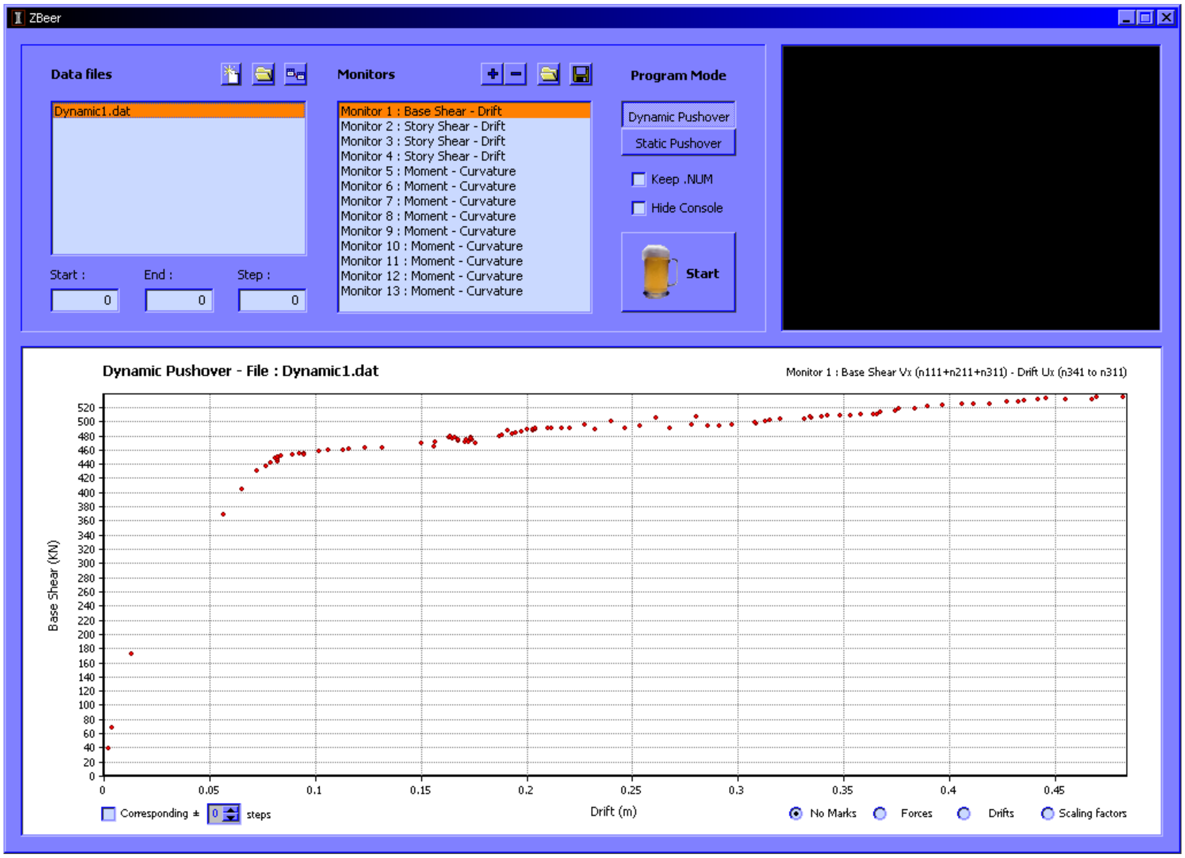


Fig.F1 The ZBeer main window

• **Input region** : It includes the type of analysis, a list of data files scheduled to run, a list of response parameters to monitor, scaling factors for dynamic pushover analysis, the calculation of the CCDF value and some other secondary options. All these features will be presented in detail in the subsequent paragraphs.

• **Status region** : The status window shows online information about the number of files run, record scaling factors for dynamic pushover analysis, and elapsed times.

• **Chart region** : This chart depicts the analysis results, shows response numeric values and features data exporting when right-clicked.

F.2. Theoretical background

There are two types of analysis supported by the ZBeer utility. Inelastic static (pushover) analysis, in any form (conventional or adaptive), and dynamic pushover analysis, also referred as Incremental Dynamic Analysis (IDA).

F.2.1 Static pushover analysis

This type of analysis has been already presented in detail in previous chapters both in its conventional and adaptive form. The difference that ZBeer provides is the ability to run *multiple* (practically unlimited) static pushover files (instead of one at a time) and collect the user-selected response parameters in separate files.

F.2.2 Dynamic pushover analysis

The dynamic pushover approach is a special analysis technique where the structural system under consideration is excited by the same strong motion input, scaled to different PGA values. For every scaling factor, the **maximum** response parameters (shear-drift, moment-curvature etc.) are plotted on a 2D plot just like static pushover curves. The difference is that now each point represents a *full run inelastic dynamic analysis*, whereas each point of the static pushover curve is simply a load step.

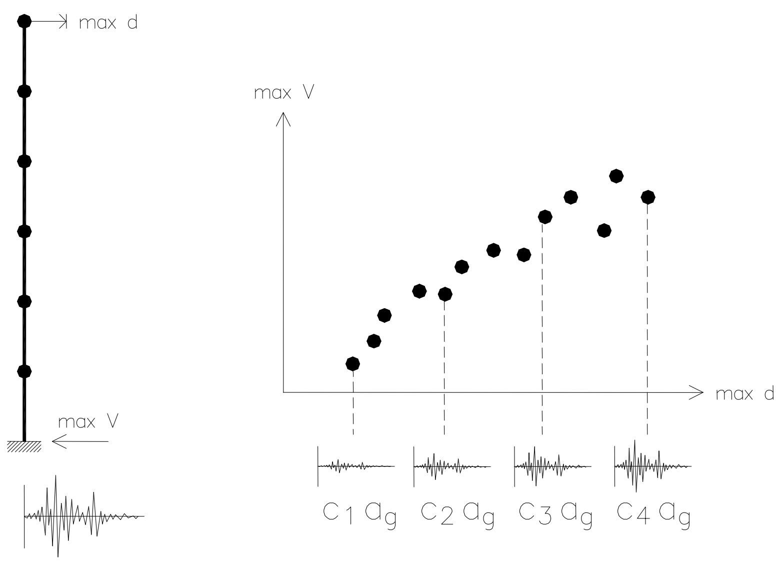


Fig.F2 Implementation of the 'Dynamic Pushover' approach

The selection of the dynamic response absolute maxima is an issue which requires further discussion, regarding the fact that they *may not occur in the same time instant*. For this reason, ZBeer has the ability to collect not only the pair of absolute maxima (X_{max}, Y_{max}) for a response parameter, but also the **corresponding** values of both response maxima (Y_{cor}, X_{cor}), with an optional **time step window** (up to ±3 time steps) inside which the maximum corresponding values are sought. Figure F3 shows the above concept in detail.

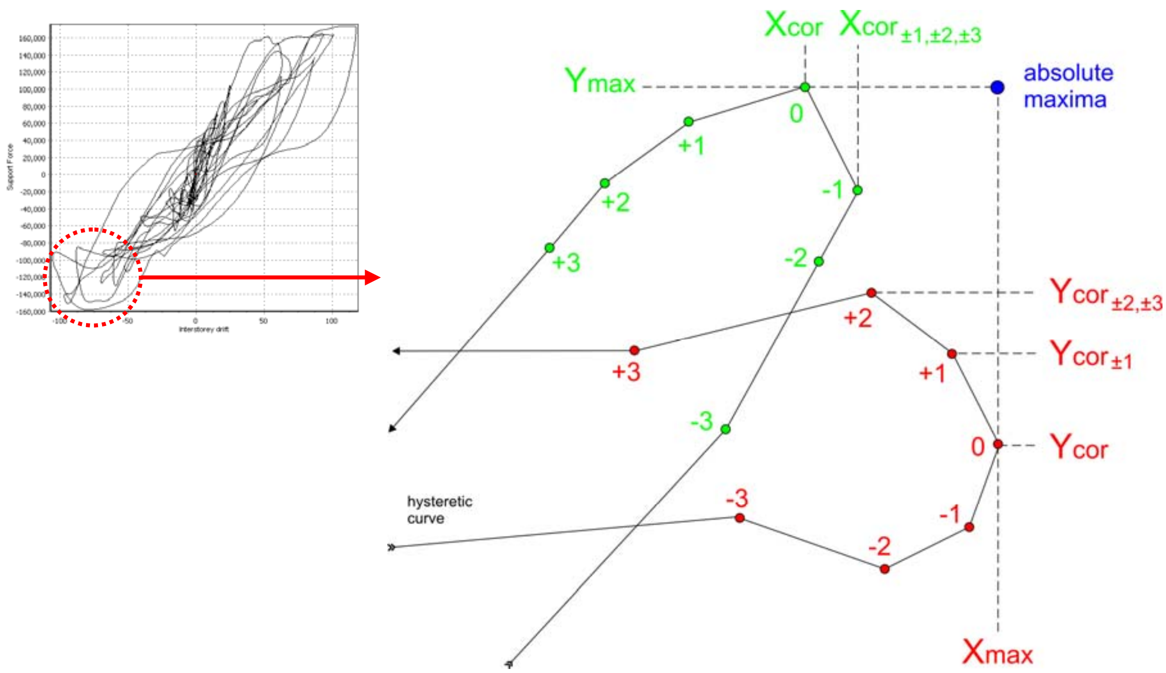
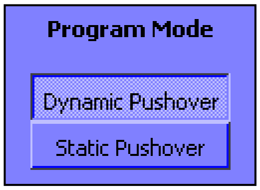


Fig.F3 Response maxima and corresponding values including a time step window

In the above figure, X_{max} has a corresponding Y_{cor} , when no step window is taken into account. If a step window of ±1 is considered, the corresponding Y_{cor} is the maximum ordinate of points 0,+1 and -1, resulting to point Y_{cor}±1, and so forth. The use of response maximum with corresponding values (instead of both absolute maxima) results into **two** series of dynamic pushover points, one for the maximum X with corresponding Y values and the other for just the opposite. It is noted here that in cases where the response maxima actually occur at the *same time instant*, the use of corresponding values does not make any difference.

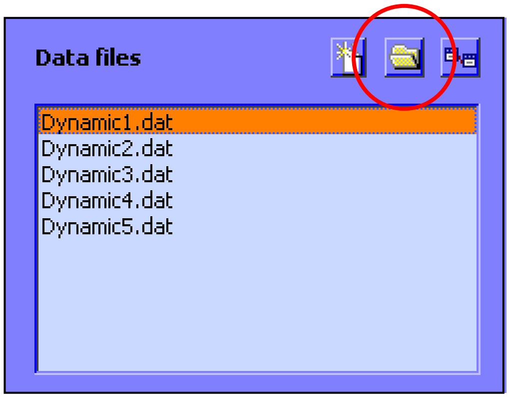
Running dynamic pushover analysis is a time consuming process, but ZBeer simplifies the whole procedure by automatically scaling the input record for a series of scaling factors, running the dynamic analysis for the structure under consideration, collecting the requested response parameters and plotting the dynamic pushover points. Moreover, the above procedure can be run for many structures at a time, just like in the static pushover case, resulting to even thousands of dynamic analyses by the press of one button !

F.3 Using ZBeer



After starting ZBeer, the user has to select between **Dynamic Pushover** and **Static Pushover** mode, by pressing the corresponding button. If any monitors or results exist from previous analyses, they will be deleted.

F.3.1 The File List



After selecting the type of analysis, data files have to be added in the files list. This is done by pressing the '**Open data file**' button and select the data files for analysis, which of course have to comply with its type (static or dynamic). Adding files from different directories is also possible by following the above procedure as many times as needed. Extra attention has to be paid with dynamic analysis though, because all the associated record files have to be present in their original directories. If the original directory of the record file needs to be changed, this can be done either from the ZeusNL core program (edit the time-history curve) or by manually editing the following line inside the dynamic analysis data file from :

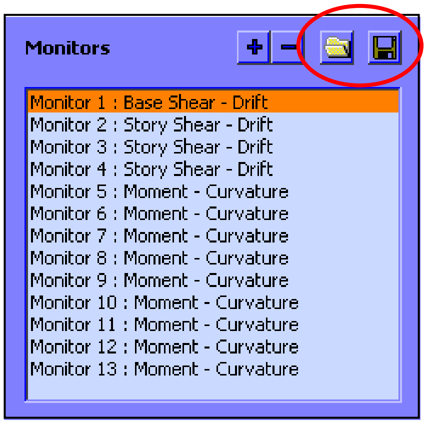
```
# Do NOT change the commented line below!
# C:\Program Files\ZeusNL\ZBeer Examples\LomaPrieta.rec[ 1   2   1   500 ]
```

to :

```
# Do NOT change the commented line below!
# <new path>\LomaPrieta.rec[ 1   2   1   500 ]
```

In case of any mistake during the file addition procedure, the program can be reset by pressing the '**New session**' button on the left of the open button. The button on the right is for calculation of the Capacity Curve Discrepancy Factor (CCDF) and will be discussed in detail later on.

F.3.2 The Monitor List



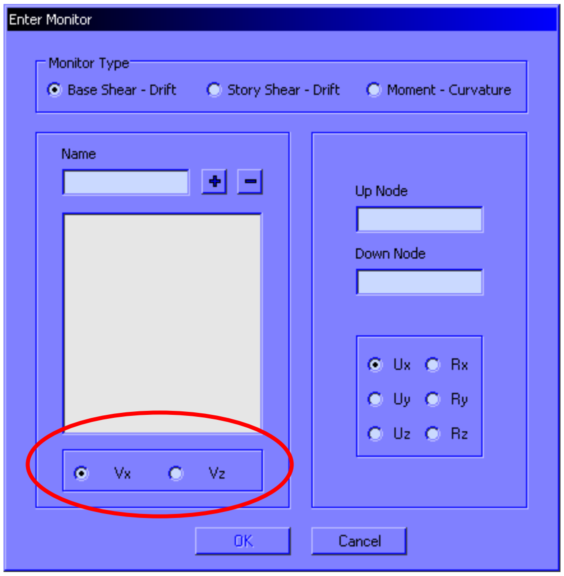
After the file selection, comes the most important procedure before the fully automated analysis provided by ZBeer : The selection of response monitors and creation of the monitor list. Before presenting all different monitor types and creation steps in detail, it is noted that at this stage, the user can load an already prepared list of monitors stored in the form of a .mon file. This can be done by pressing the '**Open monitor list**' button and select the monitors file. In the same way, when a monitor list is ready, the user can store it in the disk by pressing the '**Save monitor list**' button and enter a preferred filename. The monitors file (.mon) is written in plain text format and can be edited with any text editor (like notepad) for easy corrections or modifications. However, it has a strict syntax and extensive editing may result in serious errors.

New monitors can be added by pressing the '**Add monitor'** button, indicated by the plus (+) sign. Similarly, removing an existing monitor from the list can be done by pressing the '**Delete monitor**' button indicated by the minus (-) sign. Three different types of response monitors are available. These are :

1) Base shear – Drift monitor

2) Story shear – Drift monitor

3) Element moment – Section curvature monitor



These three types are explained in the subsequent paragraphs.

F.3.2.1 Base shear vs Drift

The first monitoring option is the base shear (V) versus drift (d) (figure F4). Horizontal forces (V_i) from support nodes are added and plotted against the displacement (or rotation) difference between two nodes, usually between a top story node and one at the base of the structure (global drift). Note here that whereas a base node is fixed for static pushover analysis, it is displaced (and/or rotated) during dynamic analysis.

After pressing the 'Add Monitor' button, a new dialog window appears, where the user must select the **Base Shear – Drift** monitor type in the top of the dialog window.

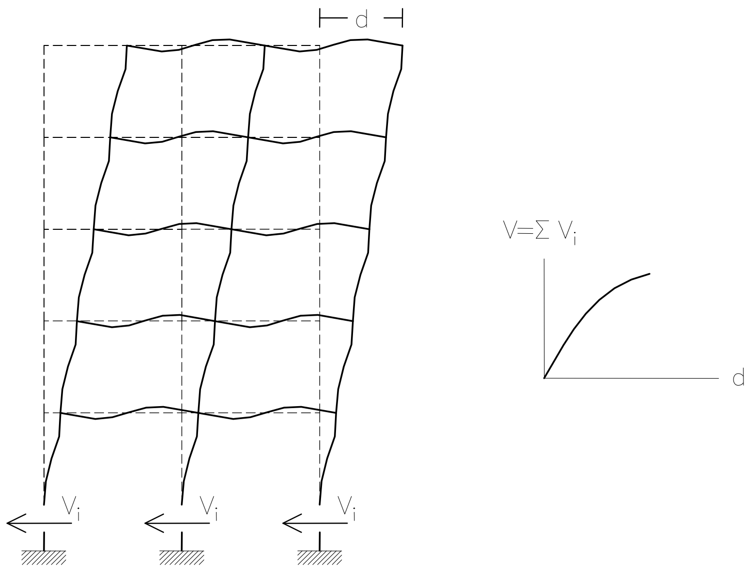
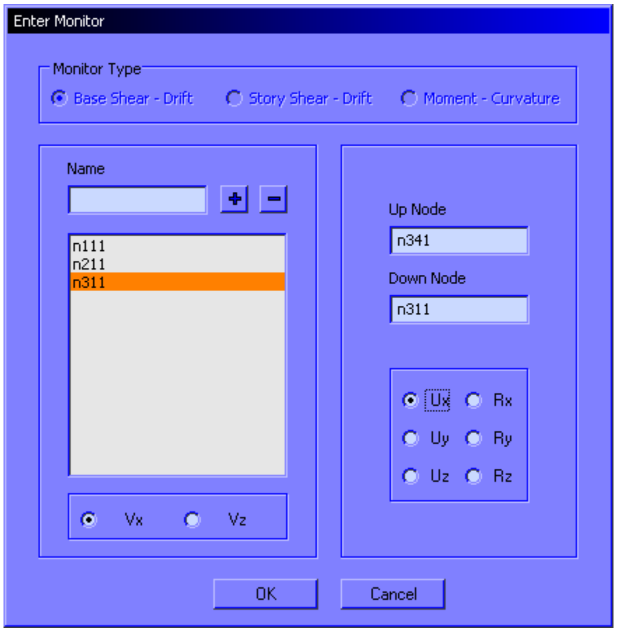


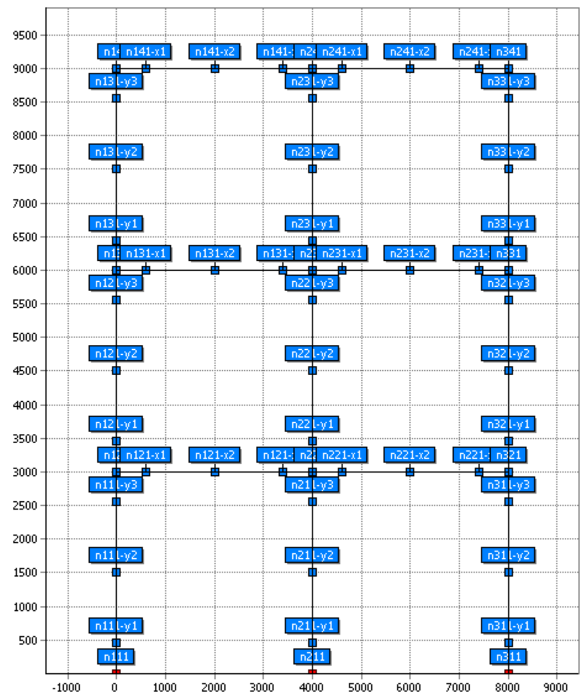
Figure F4 Base shear versus global drift monitoring

Next, the base shear has to be defined by adding all base nodes in a list. This is done by entering each node name in the 'Name' editbox and pressing the '**Add node**' button. Accidentally entered nodes can be removed by pressing the '**Delete node**' button. Finally, the base shear direction must be specified; for 2D structures the direction is always Vx, whereas for 3D structures it can also be Vz.

The drift is specified by entering the up and down node names in the corresponding editboxes, along with the displacement (or rotation) direction. The resulting drift is d = d_{up} - d_{down} where d_{up} and d_{down} are the up and down node displacements (or rotations) respectively. The snapshot at the left shows a completed monitor dialog.



By pressing ‘**OK**’ the monitor is added in the list and its full description is displayed in the chart window. Figure F5 shows an example of the correct definition of this monitor type.



Base shear :

n111+n211+n311

Direction : Vx

Global drift :

Up node : n341

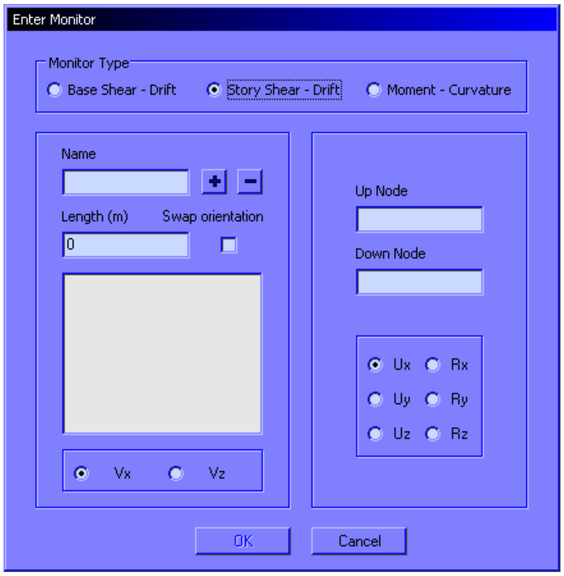
Down node : n311

Direction : Ux

Figure F5 Definition example of base shear vs global drift monitor

F.3.2.2 Story shear vs Drift

The second monitoring option is the story shear (V^S) versus drift (d) (figure F6). Horizontal shear forces (V^S_i) from columns of the same story are added and plotted against the displacement (or rotation) difference between two nodes, usually at the top and the bottom of the same story (interstory drift). The element shear is calculated by subtracting the end moments of the element and dividing by the element length $V = (M_2\text{-}M_1)/\ell$, and hence the element length has to be specified as well.



After pressing the ‘Add Monitor’ button, a new dialog window appears, where the user must select the **Story Shear – Drift** monitor type in the top of the dialog window.

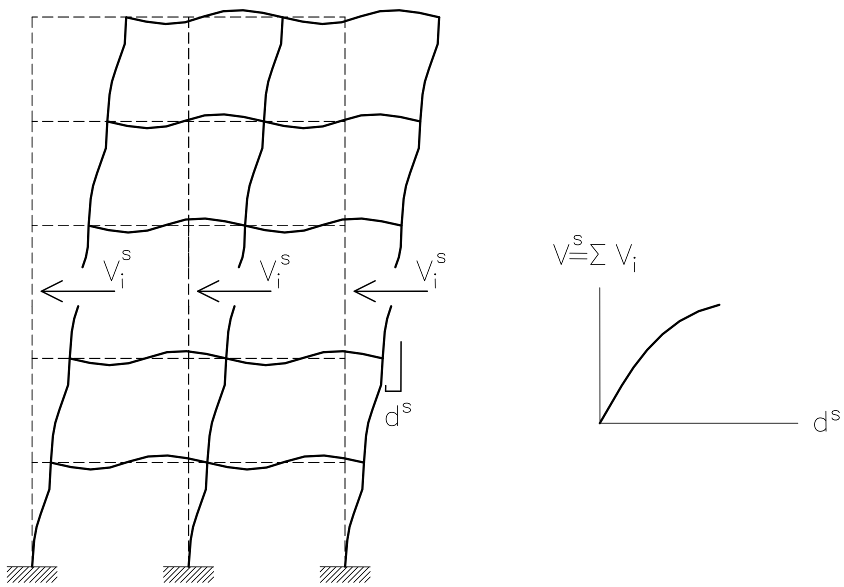
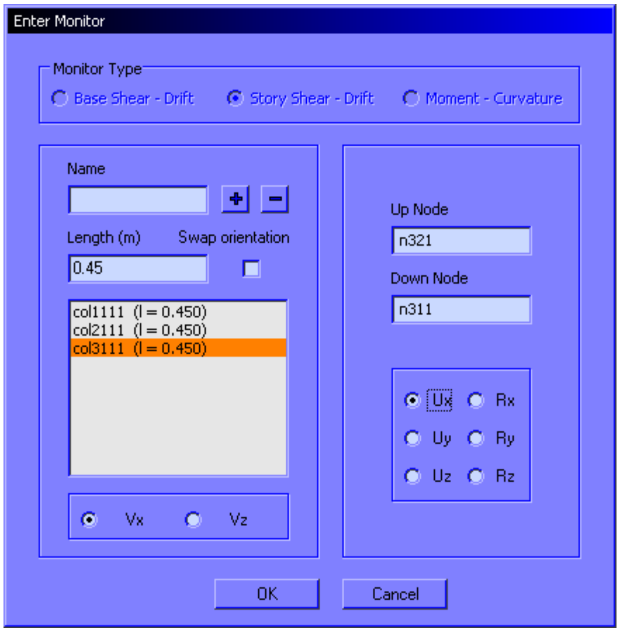


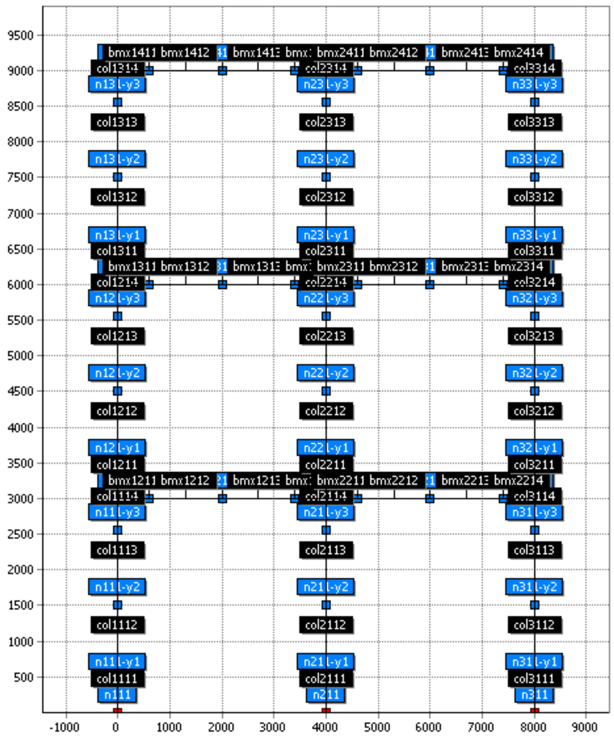
Figure F6 Story shear versus interstory drift monitoring



Next, the story shear has to be defined by adding all story columns in a list. This is done by entering each element name along with its length in the corresponding editboxes and pressing the ‘**Add element** button. Accidentally entered columns can be removed by pressing the ‘**Delete element** button. The ‘**Swap orientation**’ check box has to be checked in the rare case in which the 2-axis of the column points downwards instead of upwards (default). Refer to appendix E for more details on the element orientation. Finally, the story shear direction must be specified; for 2D structures the direction is always Vx, whereas for 3D structures it can also be Vz.

The drift is specified by entering the up and down node names in the corresponding editboxes, along with the displacement (or rotation) direction. The resulting drift is $d = d_{up} - d_{down}$ where d_{up} and d_{down} are the up and down node displacements (or rotations) respectively. The snapshot at the left shows a completed monitor dialog.

By pressing '**OK**' the monitor is added in the list and its full description is displayed in the chart window. Figure F7 shows an example of the correct definition of this monitor type.



Story shears :

Story 1 : col1111+col2111+col3111

Story 2 : col1211+col2211+col3211

Story 3 : col1311+col2311+col3311

Element length : 0.45 m

Direction : Vx

Interstory drifts :

Story 1 : Up n321 , Down n311

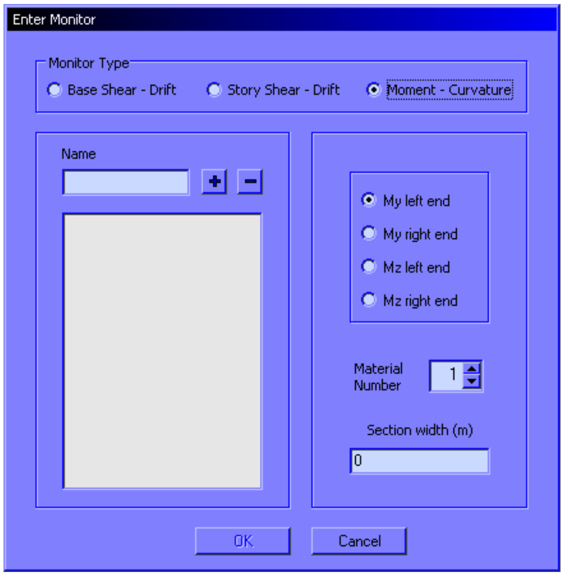
Story 2 : Up n331 , Down n321

Story 3 : Up n341 , Down n331

Direction : Ux

Figure F7 Definition example of story shear vs interstory drift monitor

F.3.2.3 Element moment vs Section curvature



The third and final monitoring option is the element moment versus section curvature (figure F8). The curvature is calculated by the formula $(\varepsilon_t - \varepsilon_b)/h$ where ε_t and ε_b are the top and bottom layer strains respectively and h is the height of the section. Various strain layers can be selected for each material that constitutes the section, but then attention must be paid to the definition of the layer width (h).

After pressing the 'Add Monitor' button, a new dialog window appears, where the user must select the **Moment – Curvature** monitor type in the top of the dialog window.

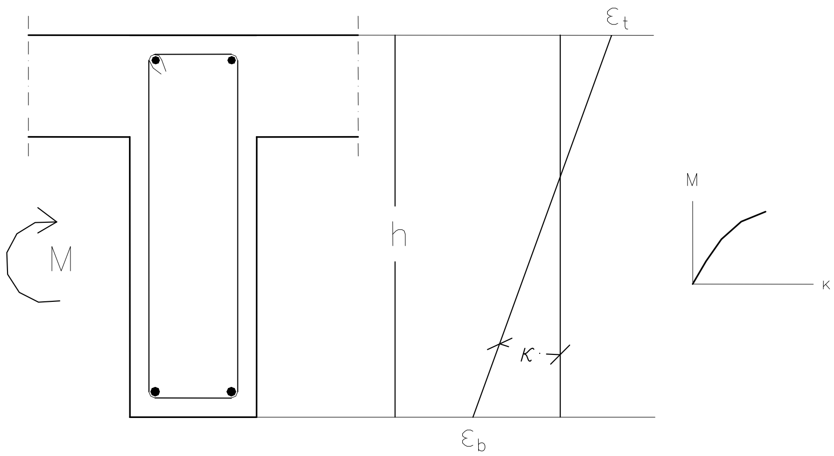
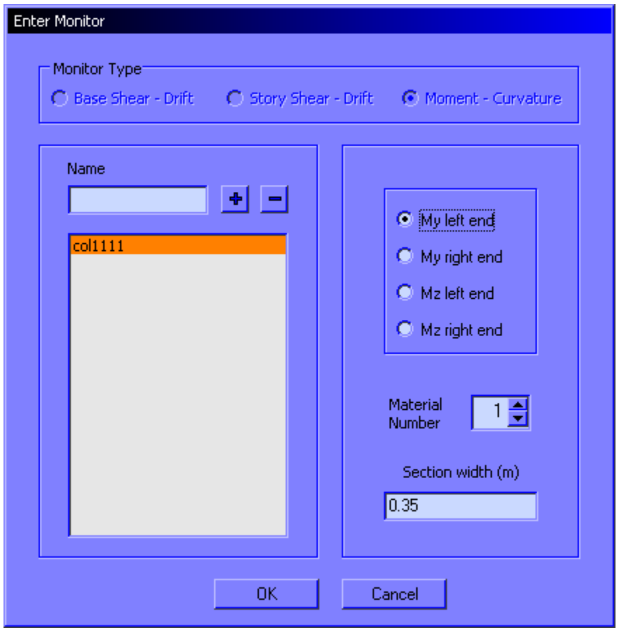


Figure F8 Element moment versus section curvature monitoring

Next, the element to be monitored must be defined by entering its name and pressing the '**Add element** button. If it is wrongly entered, it can be removed by pressing the '**Delete element** button. Note that only one element can be monitored in this monitor type.



There are four different moments that can be selected. My and Mz both for left and right element ends. My stands for the moment around the 1-axis (usually strong axis) of the element and Mz around the 3-axis of the element (not applicable for 2D structures). The left end coincides with the element start node (or the origin of its 2-axis) and the right end with the destination node (or the end of its 1-axis). Refer again to appendix E for more details on the element orientation.

The final step is the selection of the strain level where the curvature will be measured. Every section type consists of one up to four different materials. Table F1 lists all the available sections in ZeusNL with the corresponding material number. This material number, for which the strain level will be monitored, has to be selected in the provided editbox.

Extreme attention must be paid to the '**Section Width**' editbox. The user should provide the distance between the top and bottom strain level corresponding to the material number selected. For instance, if the material selected is steel, the 'Section Width' should be the maximum distance between the steel bars. If the material selected is the confined region of a concrete column, the width of the confined region must be entered and so forth.

Section type	Material #1	Material #2	Material #3	Material #4
rss	Steel	N/A	N/A	N/A
css	Steel	N/A	N/A	N/A
chs	Steel	N/A	N/A	N/A
sits	Steel	N/A	N/A	N/A
alcs	Steel	N/A	N/A	N/A
pecs	Steel	Unconfined	Partially Confined	Fully Confined
fecs	Steel	Unconfined	Partially Confined	Fully Confined
rcrs	Steel	Unconfined	Confined	N/A
rccs	Steel	Unconfined	Confined	N/A
rcts	Steel	Unconfined	Confined	N/A
rcfws	Steel	Unconfined	Partially Confined	Fully Confined
rchrs	Steel	Unconfined	Confined	N/A
rchcs	Steel	Unconfined	Confined	N/A
rcjrs	Steel	Unconfined	Partially Confined	Fully Confined

Table F1 ZeusNL sections and material numbers

By pressing '**OK**' the monitor is added in the list and its full description is displayed in the chart window.

F.3.2.4 Tips and tricks

- The monitors list **will apply to ALL files specified in the file list**. Therefore, all structures in the file list must have the same node numbering, element numbering and element lengths, in order to obtain comparable results. If this is not possible, the user needs to specify different monitors for each group of different structures, and ignore the non-corresponding monitors for each group of structures after the analysis.

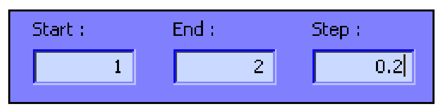
• Node and element names which do not exist in the structure are ignored, and their corresponding values are zeroed. Therefore, extra attention should be paid during the definition of base shear, story shear and drift.

• Monitoring absolute displacements and rotations instead of drifts can be realized by assigning the ‘Up Node’ to the node of interest and the ‘Down Node’ to a non-existing node name, such as ‘dummy’ (the non-existing ‘dummy’ node yields to zero displacement or rotation).

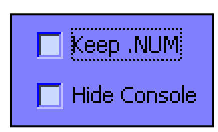
• Selecting material number 1 (Steel) is the best solution when monitoring curvature because it gives more stable results and it is common for all sections. Section width has to be the maximum distance between steel bars, along the direction of interest, for concrete sections and the section width, along the direction of interest, for steel sections.

F.3.3 Running the analysis

After completing the file and monitor list (possibly saving the latter too), analysis is ready to run. If the program mode is set to Dynamic pushover though, one last parameter has to be specified, which is the scaling factors.



Three numbers are needed : the **Start** scaling factor, the **End** scaling factor and the scaling factor **Step**. For instance, if numbers 1.0, 2.0 and 0.2 are entered respectively, six dynamic analyses will run for **EACH file specified in the file list**, with record scaling factors of 1.0, 1.2, 1.4, 1.6, 1.8 and 2.0. In order to run only one dynamic analysis for each file, Start end End numbers must be the same (Step is ignored). If Start and End numbers are 1.0, then the dynamic analysis will be the same as if it was run under the ZeusNL core program.



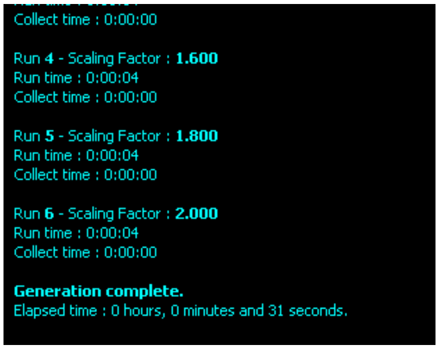
Two extra options are also available. The first one is to show or hide the console window during the analysis and the second one is to keep the bulk output of the analysis (.NUM files). Enabling the second option should be used only if special post processing of the analysis results beyond the ZBeer capabilities is intended. However, keeping the .NUM files would result in creating numerous large files and hence the available free disk space should be checked in advance. When dynamic pushover is selected, the serial number of each run is automatically appended to the .NUM filename for consistency.

By pressing

the analysis will start. Let now ZBeer do all the job ! Depending on the number of analyses scheduled to run, the program will finish in seconds, minutes, hours, or even days ! Closing the console window by pressing its ☒ button will interrupt the analysis procedure.

During the analysis, the status window is continually updated, showing the current file analyzed, the current scaling factor of the record (in dynamic pushover) and the elapsed times of analysis and collection of results. The total running time is finally displayed at the end.

It is highly recommended not to run any other programs in the background during the analysis procedure in order to prevent possible interference. It's better to switch off the monitor and leave the computer alone !



F.3.4 Getting the results

When all the scheduled analyses have been finished, results are depicted in the chart window at the bottom of the ZBeer application. Actually, results are updated during analysis too, but the user cannot yet interact with the chart window. The chart window always shows the results corresponding to the *active* data file and the *active* monitor. The term 'active' refers to the currently selected data file in the file list and the currently selected monitor in the monitor list (figure F9). By left-clicking into these two lists, the chart window is updated, and the new set of results is displayed (figures F10 and F11 for dynamic and static pushover respectively).

Figure F9 Active data file and active monitor

Zooming in the chart window can be activated by left-clicking and dragging the mouse down-rightwards ↘ . The opposite move (up-leftwards ↖) will reset any previous zoom. Panning can be activated by clicking and dragging the mouse wheel. Left clicking will activate a popup menu which will be described later on.

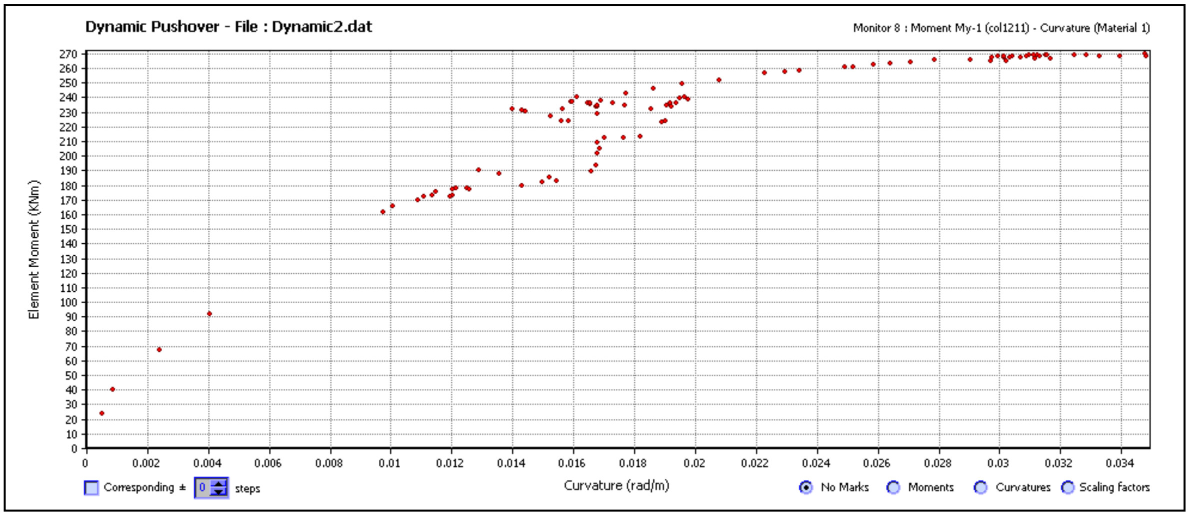


Figure F10 Dynamic Pushover results (absolute maxima)

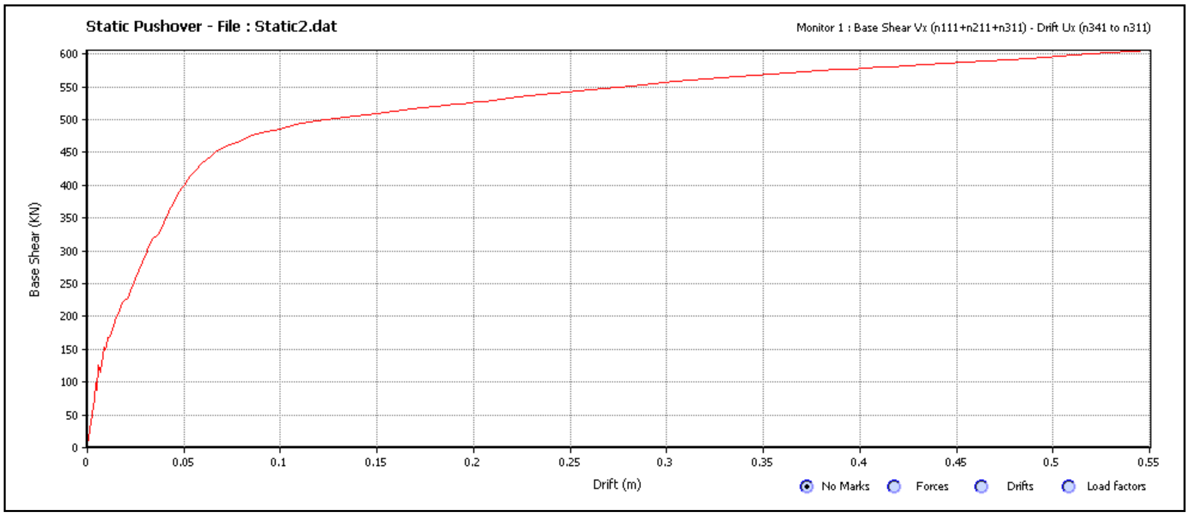


Figure F11 Static Pushover results

On the bottom and left side of the chart window there are some radio buttons which activate chart marks, such as Forces/Moments, Drifts/Curvatures and Scaling/Load factors, depending on the type of analysis and the type of the active monitor. These marks are depicted upon each dynamic pushover point or static load step, and can be better read by zooming in the chart (figure F12).

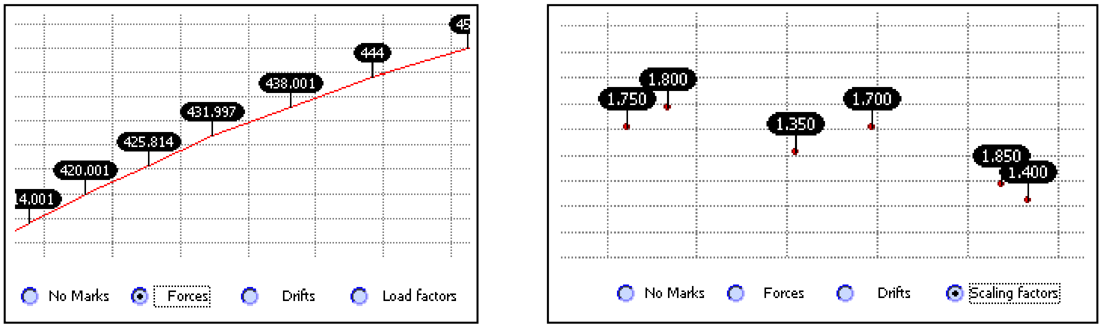


Figure F12 Chart marks

For Dynamic Pushover only, there are two available views of the outcoming results. The default view is the pair of absolute maxima for each scaling factor (red dots) and the alternative is the maximum versus corresponding values, including a user defined time step window (refer to paragraph F.2.2 and fig. F3 for more details). As already explained in the theoretical part of this chapter, when corresponding values are used, two result series are calculated, instead of one. The first is the maximum X versus corresponding Y values (X stands for displacement or curvature and Y for force or moment), which is depicted by a green normal triangle (▲) and the second is just the opposite, depicted by a green inverted triangle (▼) (Figure F13).

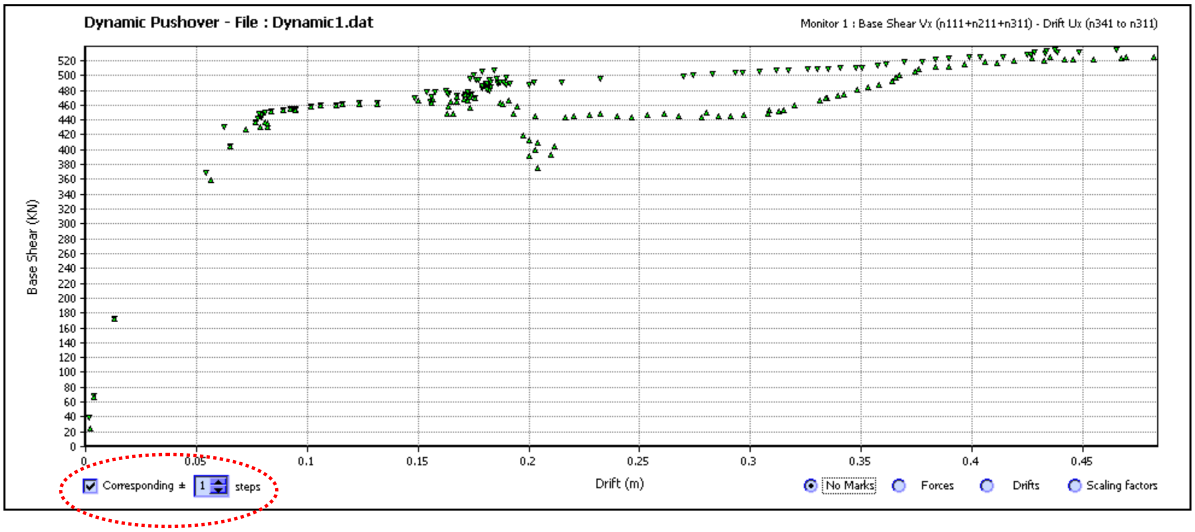
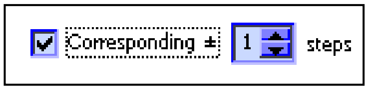
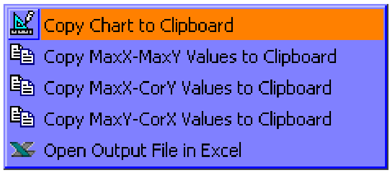


Figure F13 Dynamic Pushover results (maximum vs corresponding)

Switching between these two available views is done by pressing the checkbox an the bottom-left of the chart window and also select the size of the time step window (up to ±3 time steps). The chart is automatically updated.



All the results depicted in the chart window can be exported for further processing in spreadsheet programs (like Microsoft Excel). For this reason, right-clicking in the chart window activates a popup menu which provides the user with the following options :



- **Copy Chart to Clipboard**

  With this option, the chart is copied to the Clipboard as a bitmap image and can be pasted into any program which supports image objects (Microsoft Word, Adobe Photoshop, CorelDRAW).

- **Copy Values to Clipboard**

  By clicking this option, the numeric values of the displayed chart (active data file and active monitor) are copied to the clipboard and can be directly pasted into any spreadsheet program (like Microsoft Excel) for further processing. Three columns are created, the first with scaling or load factors (depending on the type of analysis), the second with X values (displacements/curvatures) and the third with Y values (forces/moments). In dynamic pushover analysis, the 'Copy Values' option is tripled : absolute maxima, maximum X versus corresponding Y and the opposite, with the latter two referring to the currently specified time step window.

- **Open Output File in Excel**

  The last export option is to open the Output File in Microsoft Excel. The output file is automatically created for EACH file included in the file list, under the same directory, the same file name, and the **.psh** extension. Each output file contains results for all the monitors included in the monitor list. It is tab delimited and can be easily dragged and dropped in Microsoft Excel for further processing like creating comparative plots from different analyses. By selecting this menu option, the output file of the currently *active* file (which is currently selected in the file list), is automatically opened in Excel.

It is highly recommended that after a time consuming analysis, copies of all data files (.dat), output files (.psh) and monitor files (.mon) should be kept together in a safe place, preferably in a compressed format.

ZBeer has the ability to plot past analysis results in the chart window, when the corresponding output files are present in the same directory with the data files, just after the creation of the file list. Moreover, the SAME monitors that had been used in the past analysis should also be present in the monitors list. If both these happen, the program environment will be just the same as if the analysis had just finished ! But beware : if the start button is pressed, a new analysis will start and all the output files will be reset !

F.3.5 The Capacity Curve Discrepancy Factor (CCDF)

Latest studies by the authors of ZeusNL have surfaced the need for defining a measuring quantity for the difference between inelastic static and dynamic analysis, in the form of a simple percentage number. The Capacity Curve Discrepancy Factor (CCDF) is a numerically simple but yet efficient way to define the difference between the ordinates (forces or moments) of a single pushover curve compared to a set of dynamic pushover points, both emerging from the analysis of the same structural system.

Consider the pushover curve S0-S1-S2-S3-S4 and the set of dynamic points (D1,D2,D3,D4,D5) of figure F14. Coordinates of each point are given in the parentheses. X values stand for displacements or curvatures and Y values for forces or moments, as already described earlier.

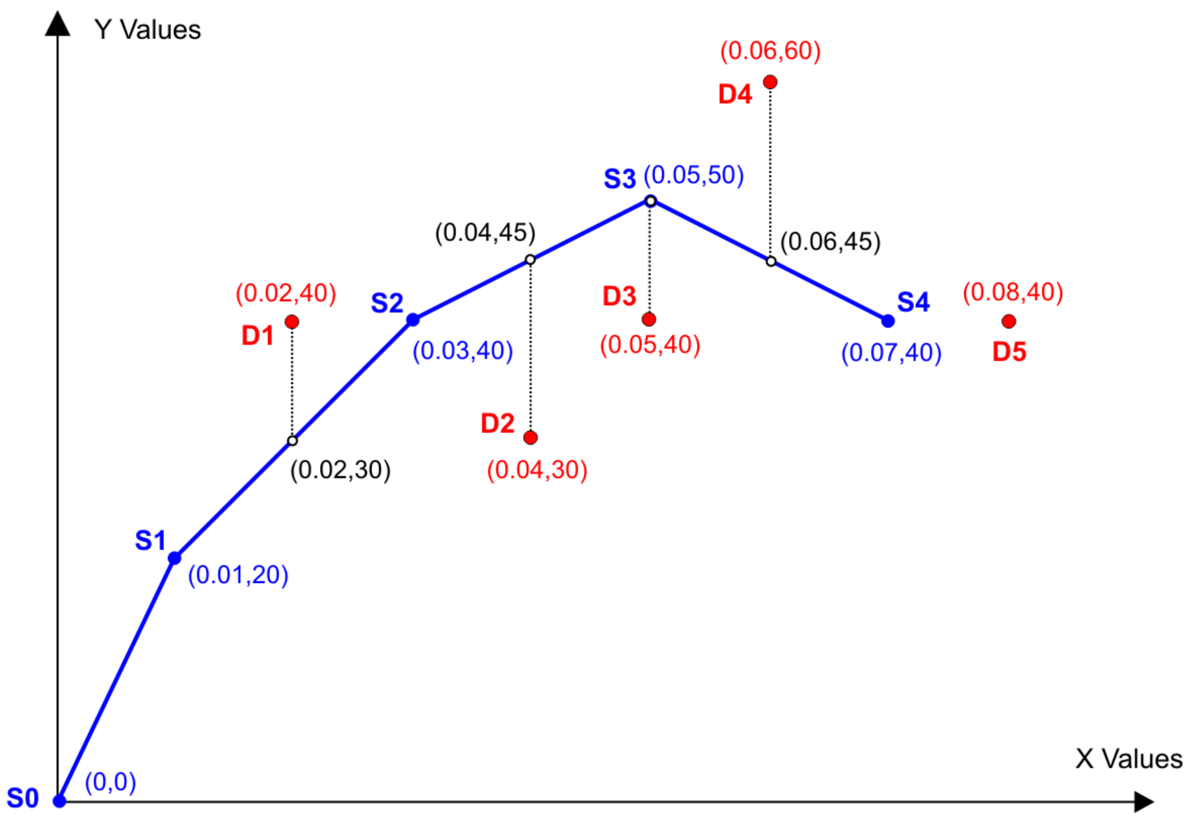


Figure F14 Definition of the Capacity Curve Discrepancy Factor (CCDF)

First of all, the coordinates of the vertical projection of each dynamic pushover point on the static pushover curve are defined (shown in black). Dynamic points with no projections on the static pushover curve are ignored, like D5 in the above example. Then, the difference between each projection point and the corresponding dynamic point (in other words the difference of static with respect to dynamic analysis) is calculated as follows :

Point D1 : d_{D1} = abs ($Y_{P1} - Y_{D1}$) / Y_{D1} = abs (30-40)/40 = 0.25

Point D2 : d_{D2} = abs ($Y_{P2} - Y_{D2}$) / Y_{D2} = abs (45-30)/30 = 0.50

Point D3 : d_{D3} = abs ($Y_{P3} - Y_{D3}$) / Y_{D3} = abs (50-40)/40 = 0.25

Point D4 : d_{D4} = abs ($Y_{P4} - Y_{D4}$) / Y_{D4} = abs (45-60)/60 = 0.25

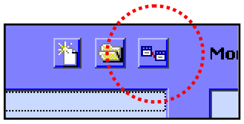
Point D5 : Ignored

The final PP value is just the average of all above difference values :

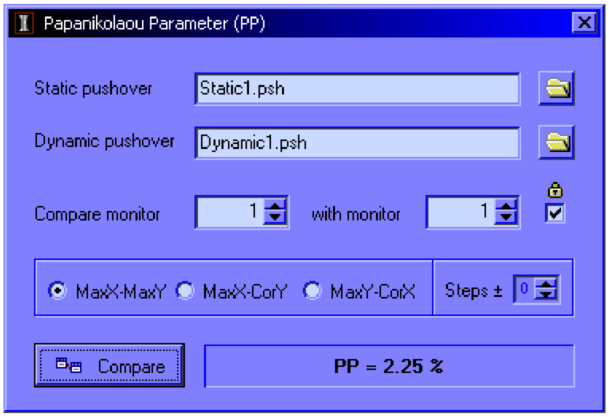
PP = (0.25+0.50+0.25+0.25) / 4 = 0.3125 = **31.25 %**

Note again that point D5 does not participate in the above procedure because it lies beyond the horizontal span of the static pushover curve. Moreover, averaging all the differences from each pair of points means that equal weight has been assigned to each pair. This is deemed to be realistic only when the scaling step of the record remains *constant* throughout the analysis, which is always the case in ZBeer. Finally, the formula ($Y_{P1} - Y_{D1}$) / Y_{D1} was selected, because the *reference* response (denominator Y_{D1}) is the dynamic response (in other words, comparing *static to dynamic*, not the opposite).

The calculation of PP was implemented as an extra utility in ZBeer. This utility is activated by pressing the PP button, in the right of the open data file button.



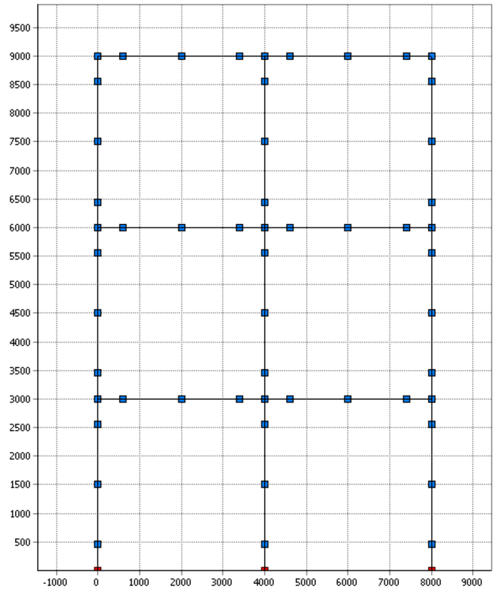
The new dialog requests the output files (.psh) of the static and dynamic pushover analysis to be entered in the corresponding edit boxes. The monitor numbers to be compared can be changed by clicking the up-down buttons. The 'locking' checkbox on the right makes the two monitor numbers (for static and dynamic pushover analysis) to be the same, which is mostly the case. Moreover, the user can select to compare either the response maxima, or the maximum versus corresponding values, with or without a time step window.



Finally, by pressing the '**Compare**' button the PP value is automatically calculated and displayed, for the current monitor number and dynamic pushover scheme.

F.3.6 Case study

After the installation of ZeusNL, a special directory under the name 'ZBeer Examples' is created inside the installation directory (usually C:\Program Files\ZeusNL). Inside this directory, there are five dynamic and five pushover analysis data files, already prepared, featuring the same three-story, two-bay concrete structure, with variable concrete strength as follows :



Static1.dat, Dynamic1.dat : Concrete **C12**

Static2.dat, Dynamic2.dat : Concrete **C20**

Static3.dat, Dynamic3.dat : Concrete **C30**

Static4.dat, Dynamic4.dat : Concrete **C40**

Static5.dat, Dynamic5.dat : Concrete **C50**

Moreover, a record file (LomaPrieta.rec) used by dynamic analysis and an already prepared monitors file for ZBeer (Monitors.mon) are included. Monitors include a base shear - global drift monitor, story shear – interstory drift monitors for all three stories and nine moment – curvature monitors of all column bases.

The user can test the performance of ZBeer by running several static and dynamic pushover analyses for these structures, collect the output files in the form of :

Dynamic1.psh – Dynamic5.psh , Static1.psh – Static5.psh

and create comparative plots using a spreadsheet program like Microsoft Excel. Finally, comparison between static and dynamic analysis can be carried out using the PP utility of ZBeer. Figure F15 shows a completed comparative plot of all ten structures, for the base shear – global drift monitor.

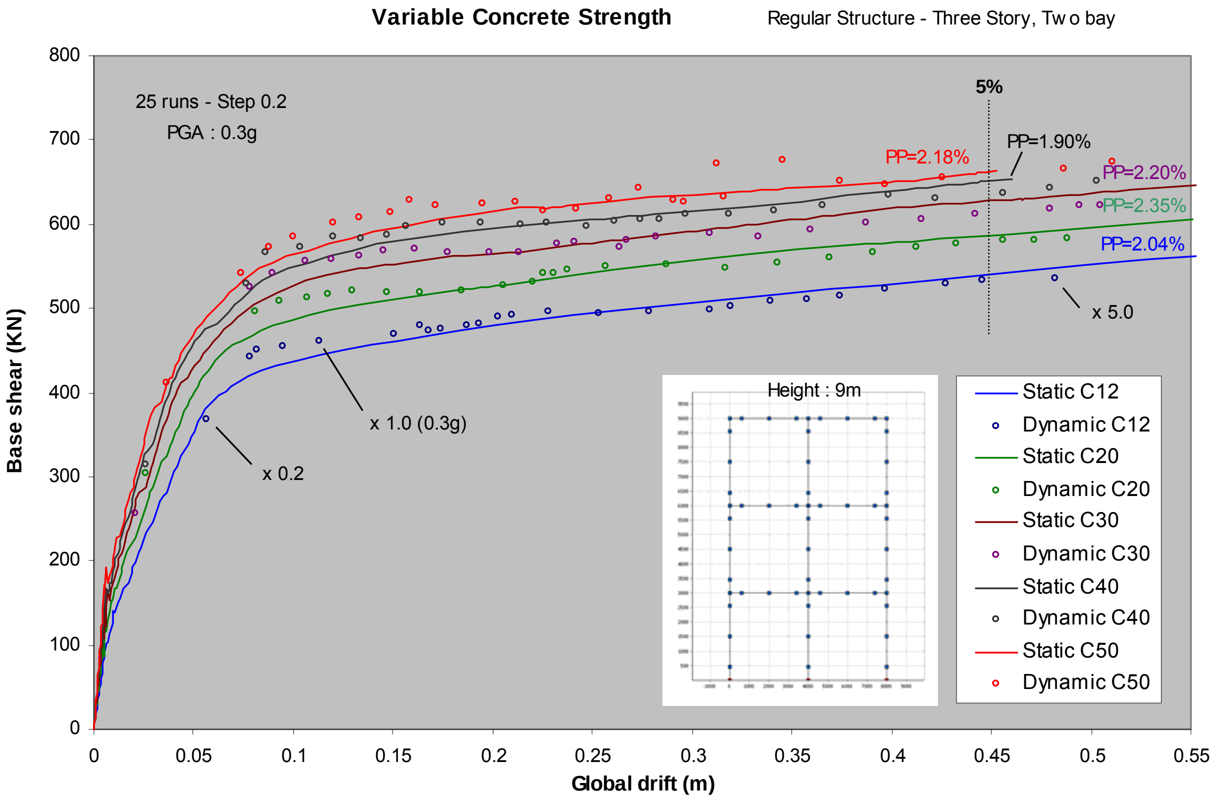


Figure F15 Comparative plot of structures with variable material strength