Promoting and Enabling International Collaboration

OpenFresco: An open source framework for performing geographically distributed and local hybrid simulations

Andreas Schellenberg, Hong Kim
Stephen A. Mahin, Gregory L. Fenves

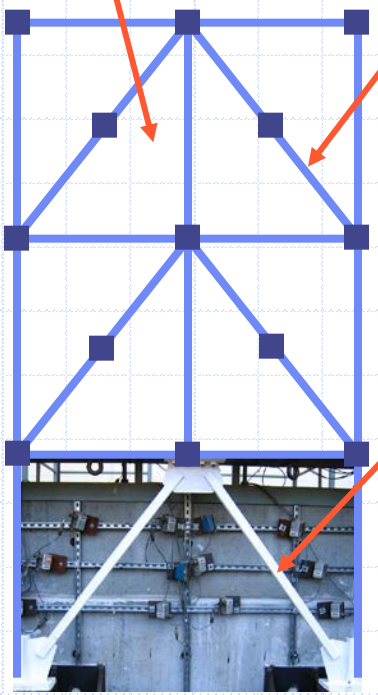
Department of Civil and Environmental Engineering
University of California, Berkeley

Yoshikazu Takahashi
Kyoto University

Hybrid Simulation

$$\mathbf{M}\cdot\ddot{\mathbf{u}}+\mathbf{C}\cdot\dot{\mathbf{u}}+\mathbf{P}_{\mathbf{r}}(\mathbf{u})=\mathbf{P}(t)$$

Dynamic Loading
- Seismic
- Wind
- Blast/Impact
- Wave
- Vehicle



analytical model of structural energy dissipation and inertia

physical model of structural resistance

Versatile implementation and support for:
- Slow or fast execution
- Local or geographically distributed
- Displacement, force & mixed control
- Collaborative computing

OpenFresco

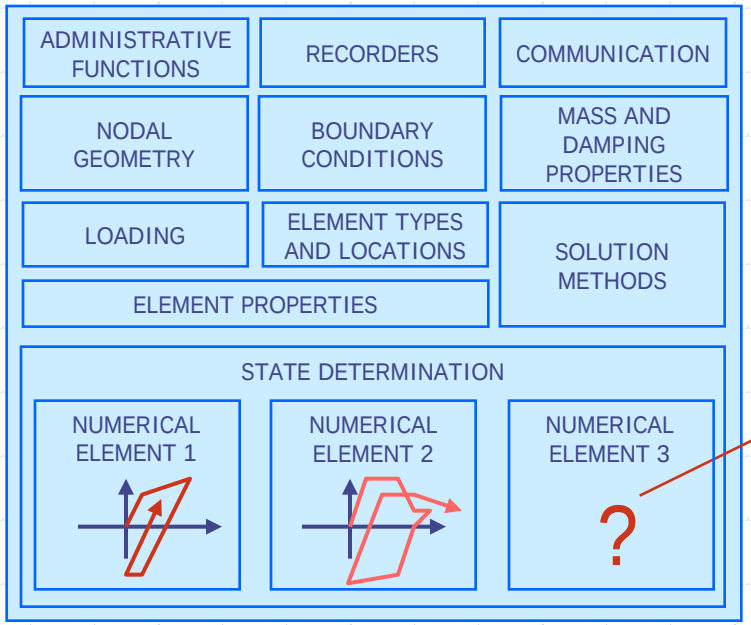
Open-source Framework for Experimental Setup and Control

Secure, object oriented, network enabled "middleware" -- Pairs computer analysis software with laboratory control systems and other software to enable hybrid and collaborative computing:

- Software
  - Abacus
  - Ansys (soon)
  - JumpStart (soon)
  - LS-Dyna
  - Matlab
  - OpenSees
  - SimCor
  - Simulink
- Control Systems
  - dSpace
  - MTS
    - STS family
    - Flextest/CSI
    - Flextest/Scramnet (soon)
  - National Instruments
  - Pacific Instruments
  - Shore Western

Implementation strategy

Embed test specimen(s) in an existing computational framework of users choice

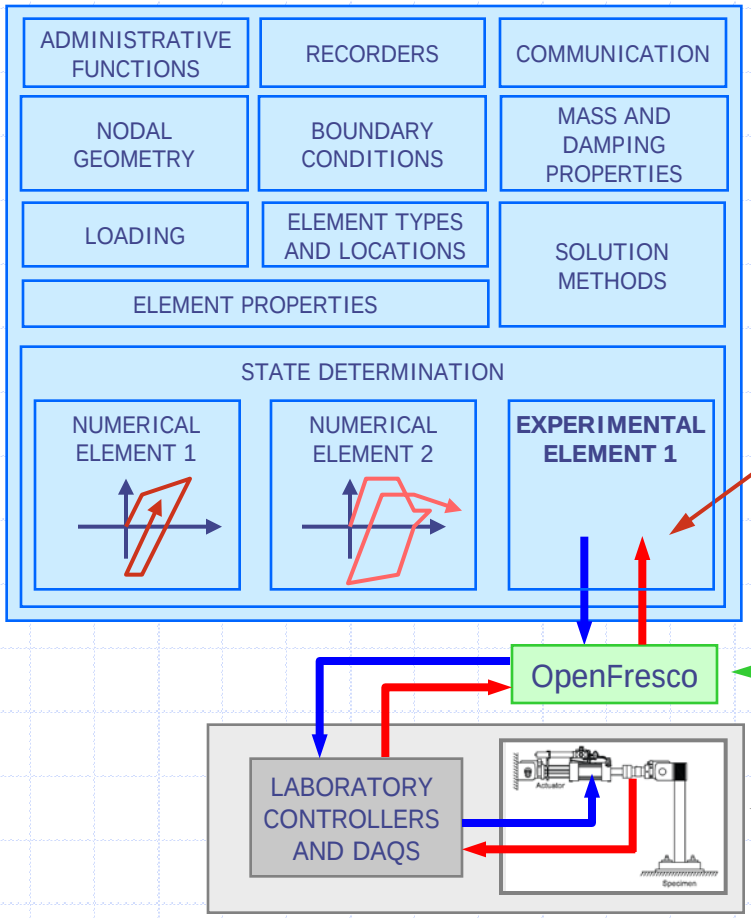


Typical features of an analysis framework

Proper numerical model uncertain

Implementation strategy

Embed test specimen(s) in an existing computational framework of users choice



Typical features of an analysis framework

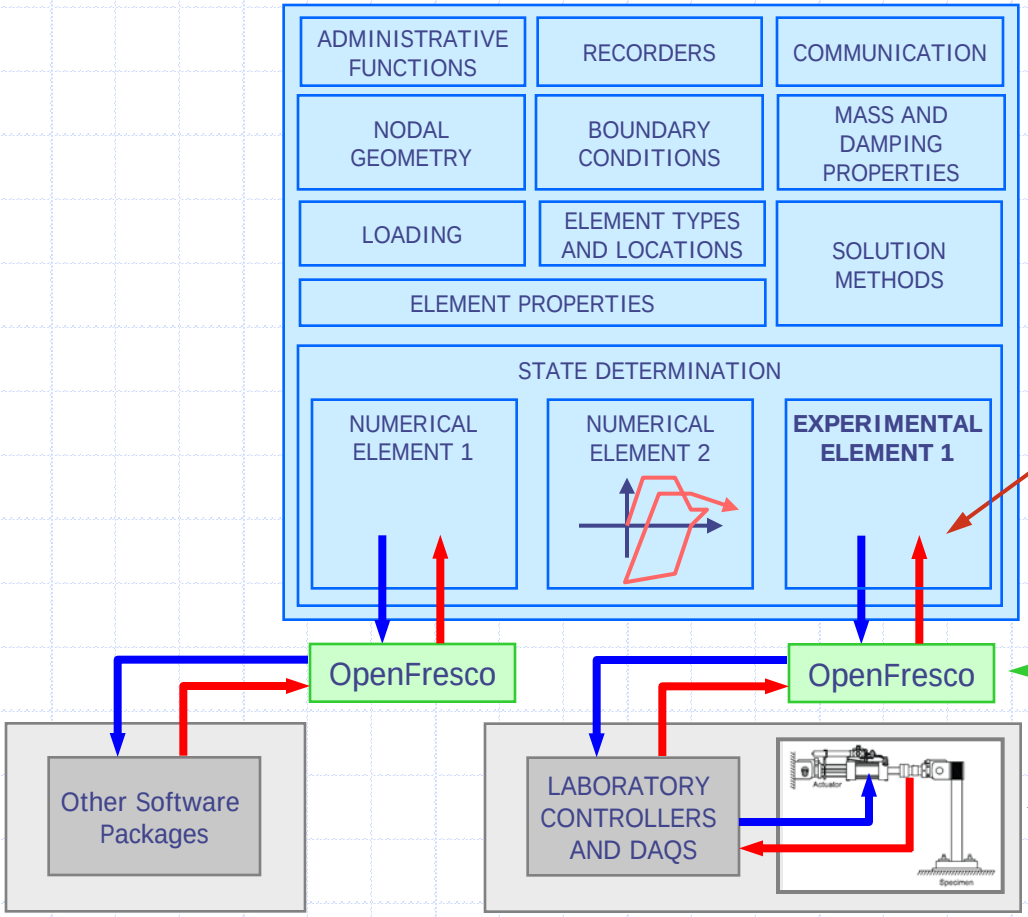
Define element as an "Experimental Element"

OpenFresco

Laboratory

Implementation strategy

Embed test specimen(s) in an existing computational framework of users choice



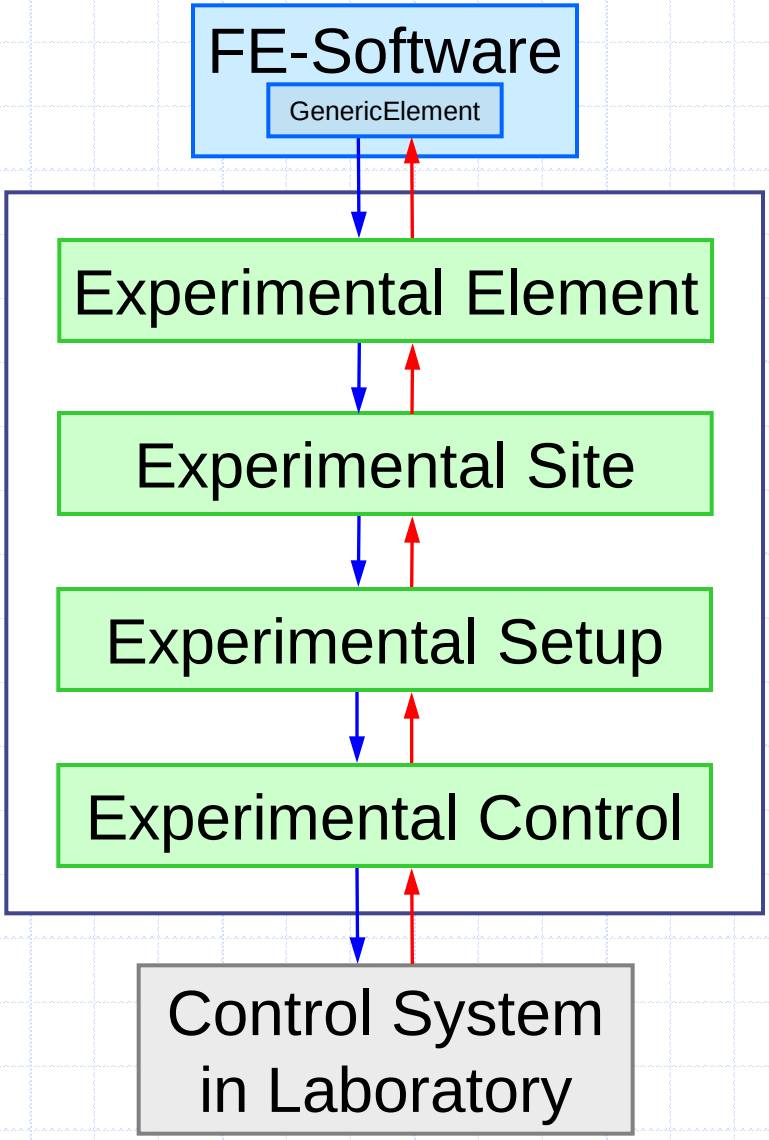
Typical features of an analysis framework

Define element as an "Experimental Element"

OpenFresco

Laboratories

OpenFresco Components



provides all features of unmodified computational framework, including parallel and network computing

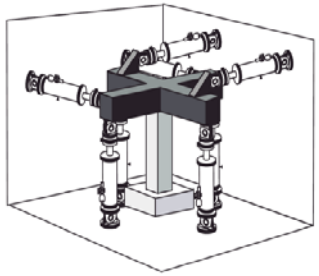
represents the part of the structure that is physically tested and provides the interface between the FE-software and the experimental software framework

stores data and provides communication methods for distributed testing

transforms between the experimental element degrees of freedom and the actuator degrees of freedom (linear or non-linear transformations)

interfaces to the different control and data acquisition systems in the laboratories

provides control of physical actuators as well as data acquisition using physical instrumentation devices



OpenSees Navigator

Andreas Schellenberg and Tony Yang
Department of Civil and Environmental Engineering
University of California, Berkeley

Eiji Kohama
Port and Airport Institute, Japan
Earthquake Engineering Research Center (EERC)
University of California, Berkeley

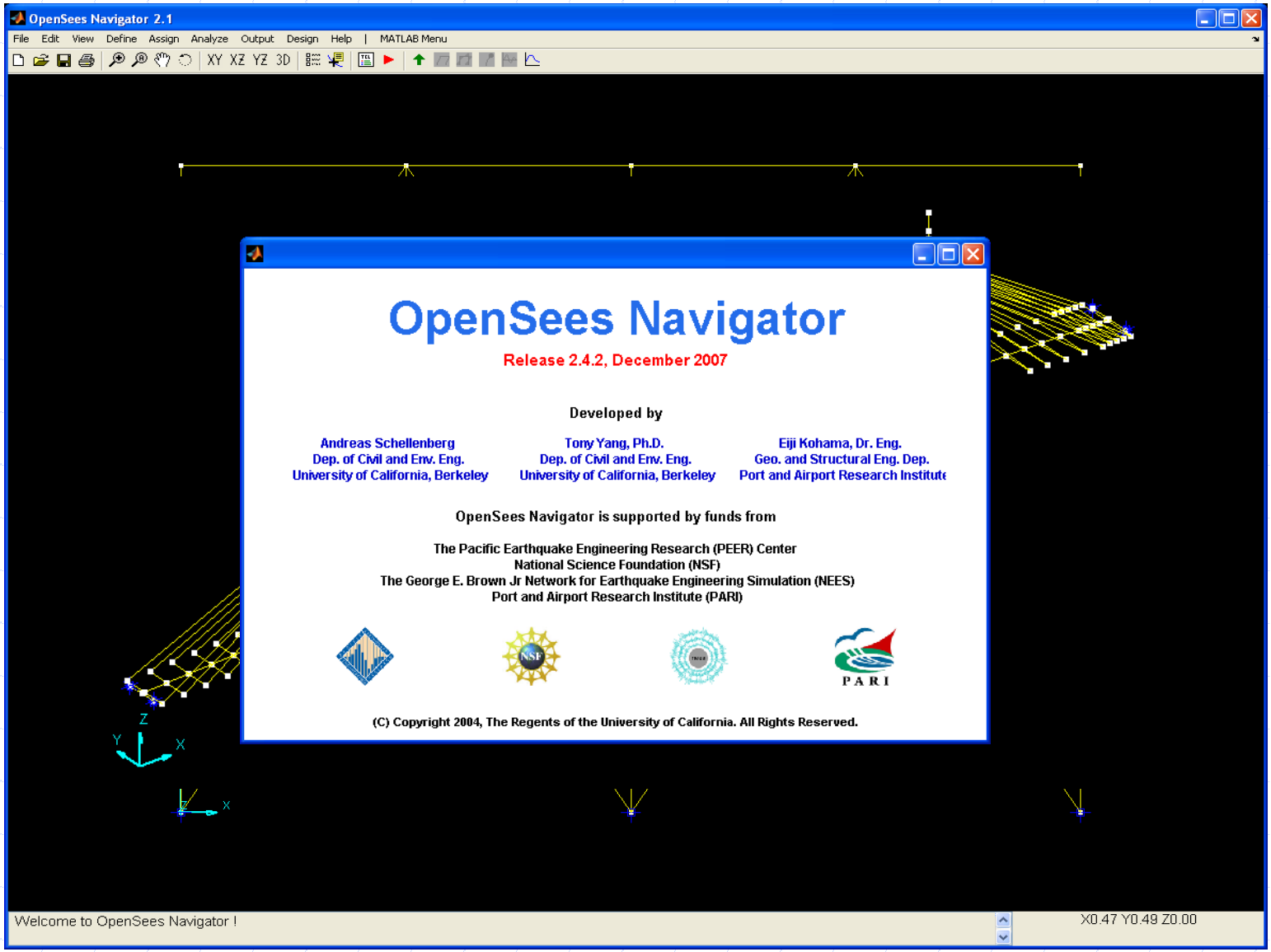
Introduction

- MATLAB based graphical user interface (GUI).
- Pre- and post-processing for OpenSees and OpenFresco.
- Design toolboxes: NSP, PBEE, AISC design checks, AISC database, response spectra for linear and bilinear systems and signal filtering.
- Both MATLAB Pcode and self-executable versions are available for Windows & Mac.
- Being used by researchers from Asia, US, Canada, south America and Europe.

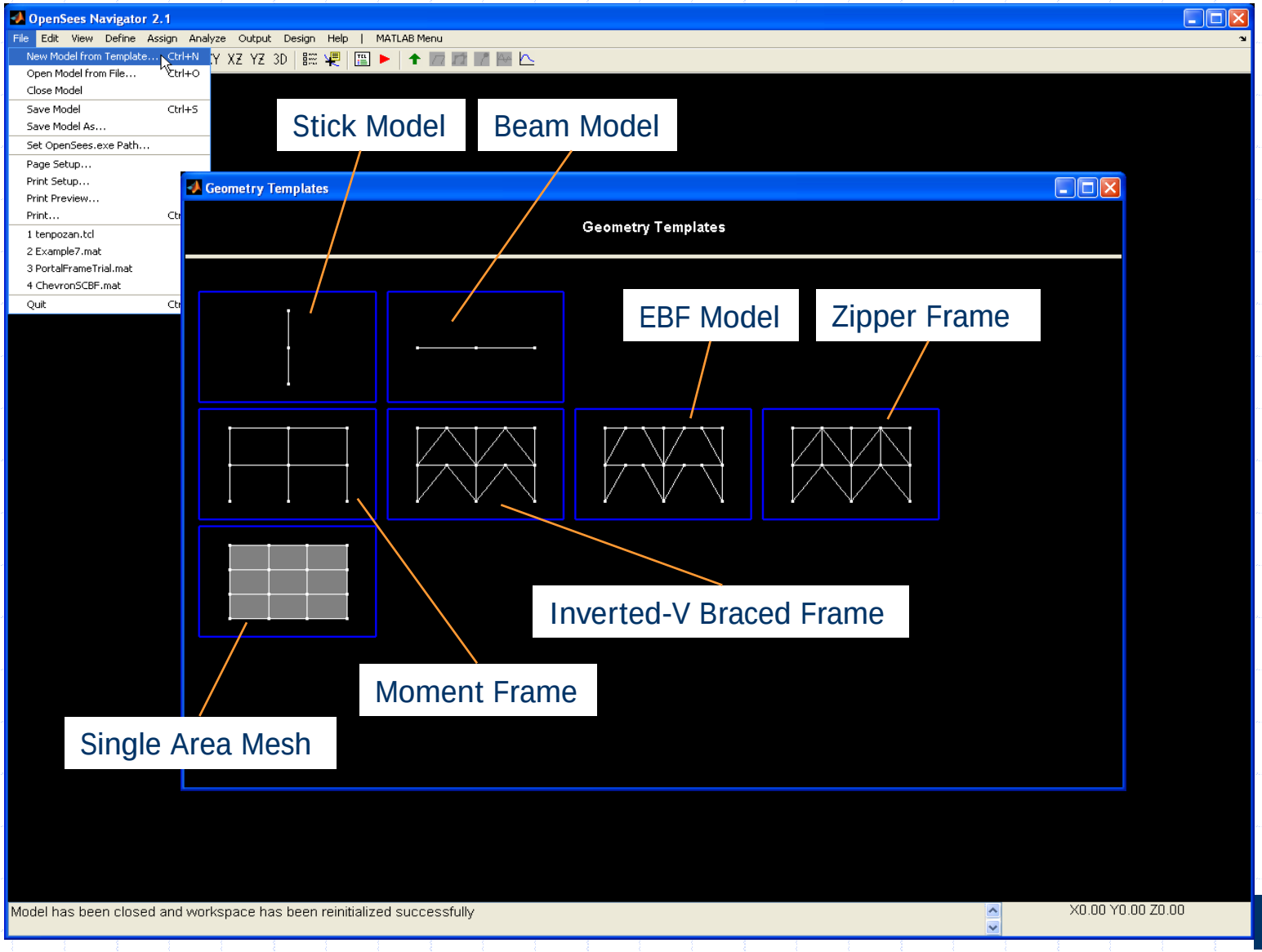
Motivations

- Graphical input is more flexible than TCL text input.
- Most researchers use MATLAB to do the post-processing, and MATLAB/Simulink is the typical framework for implementing hybrid simulation tests.
- OpenSees Navigator will create the OpenSees (analytical/hybrid) model and graphically display the results before, during or after a test.
- Flexible to use and requires no programming skill.

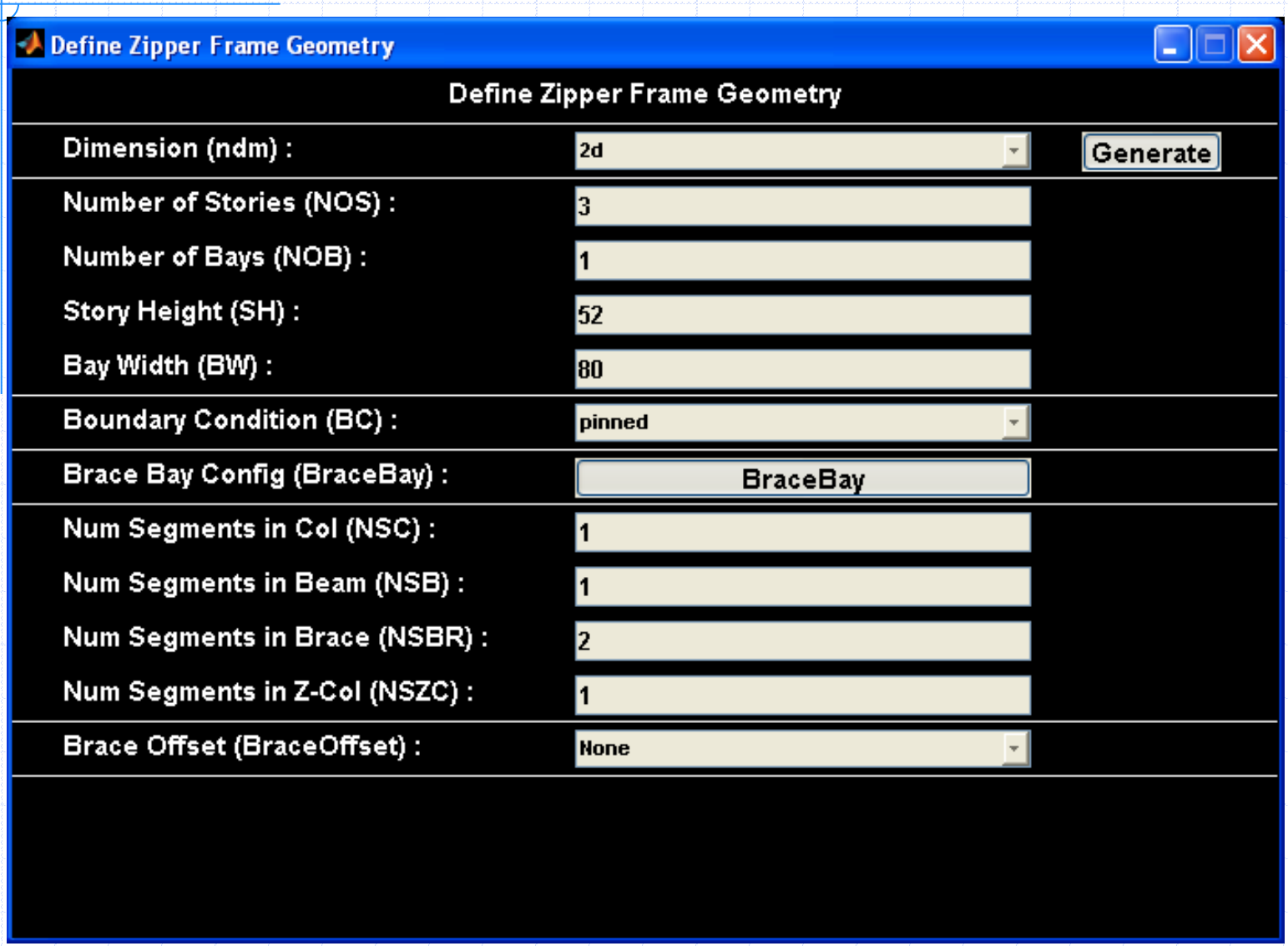
OpenSees Navigator



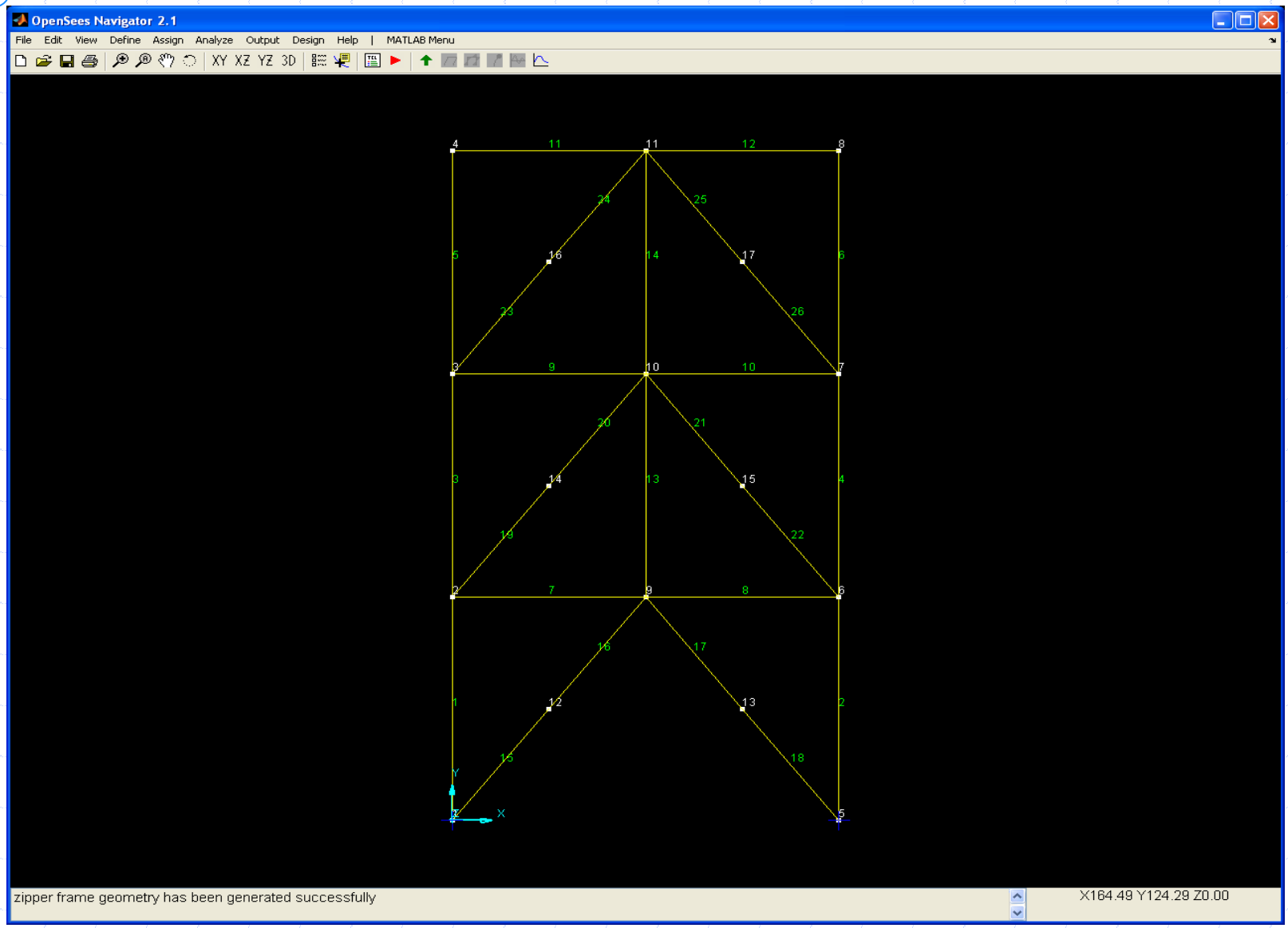
Define geometry: new model template



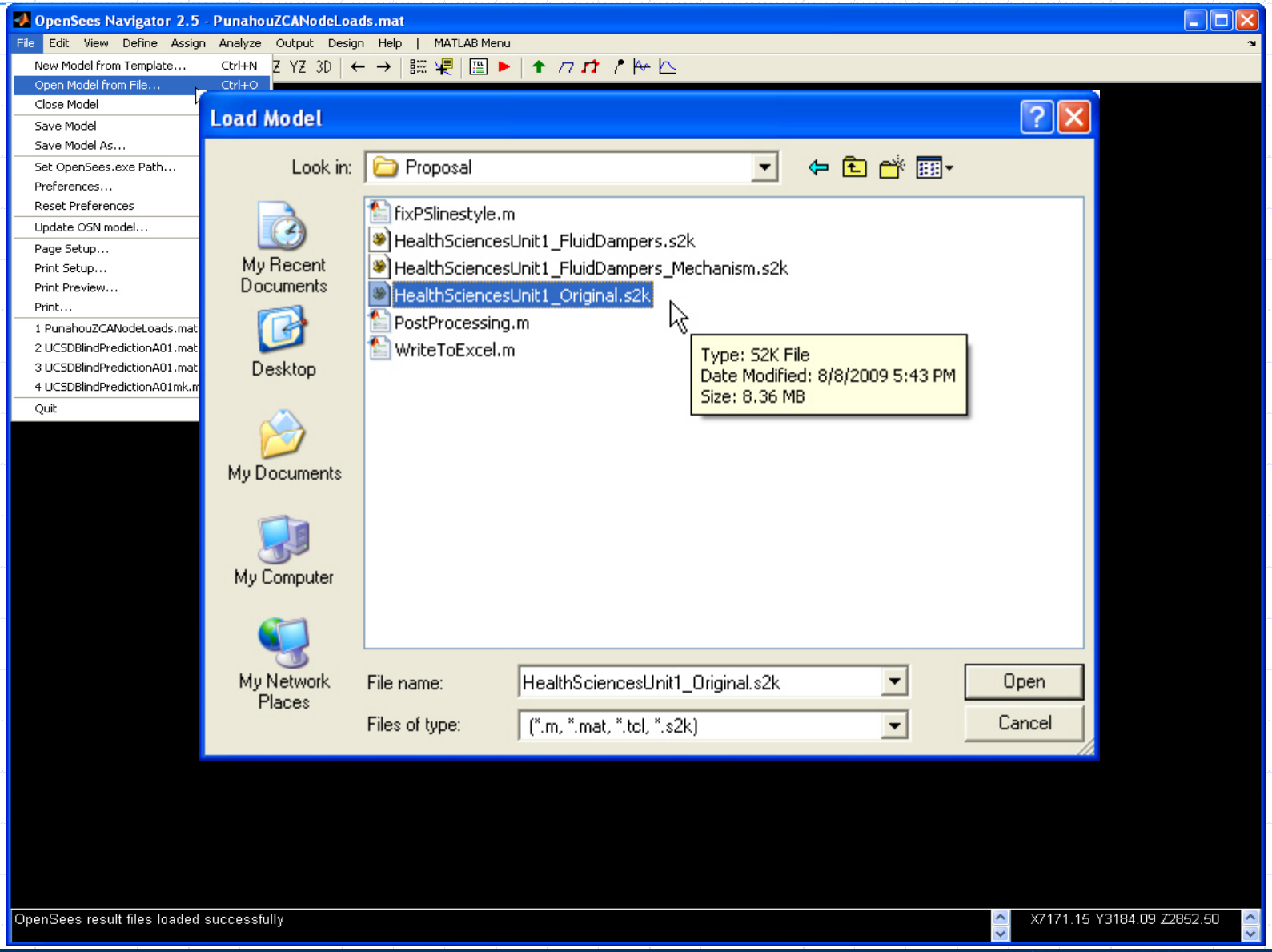
Define geometry: Zipper braced frame



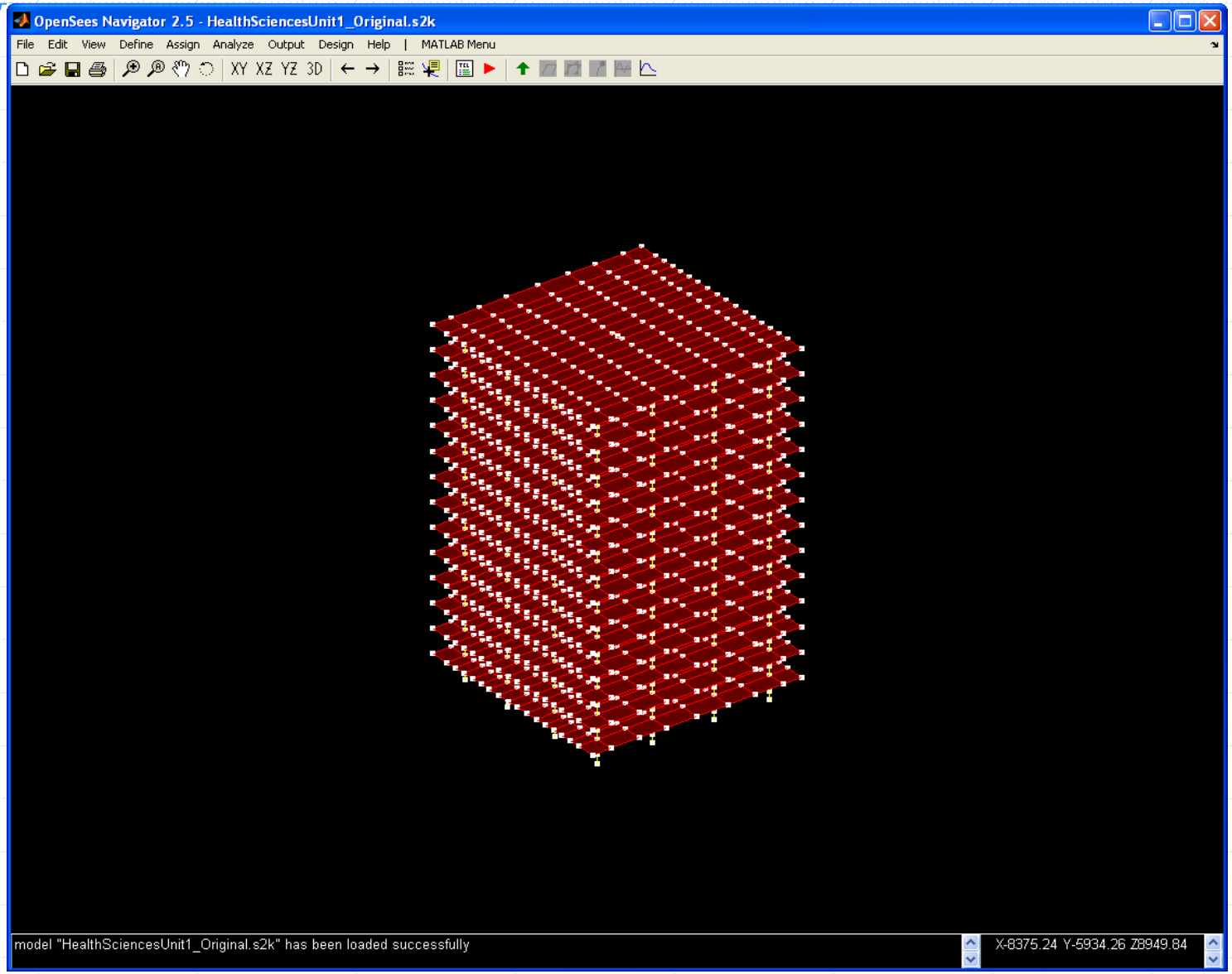
View geometry: display



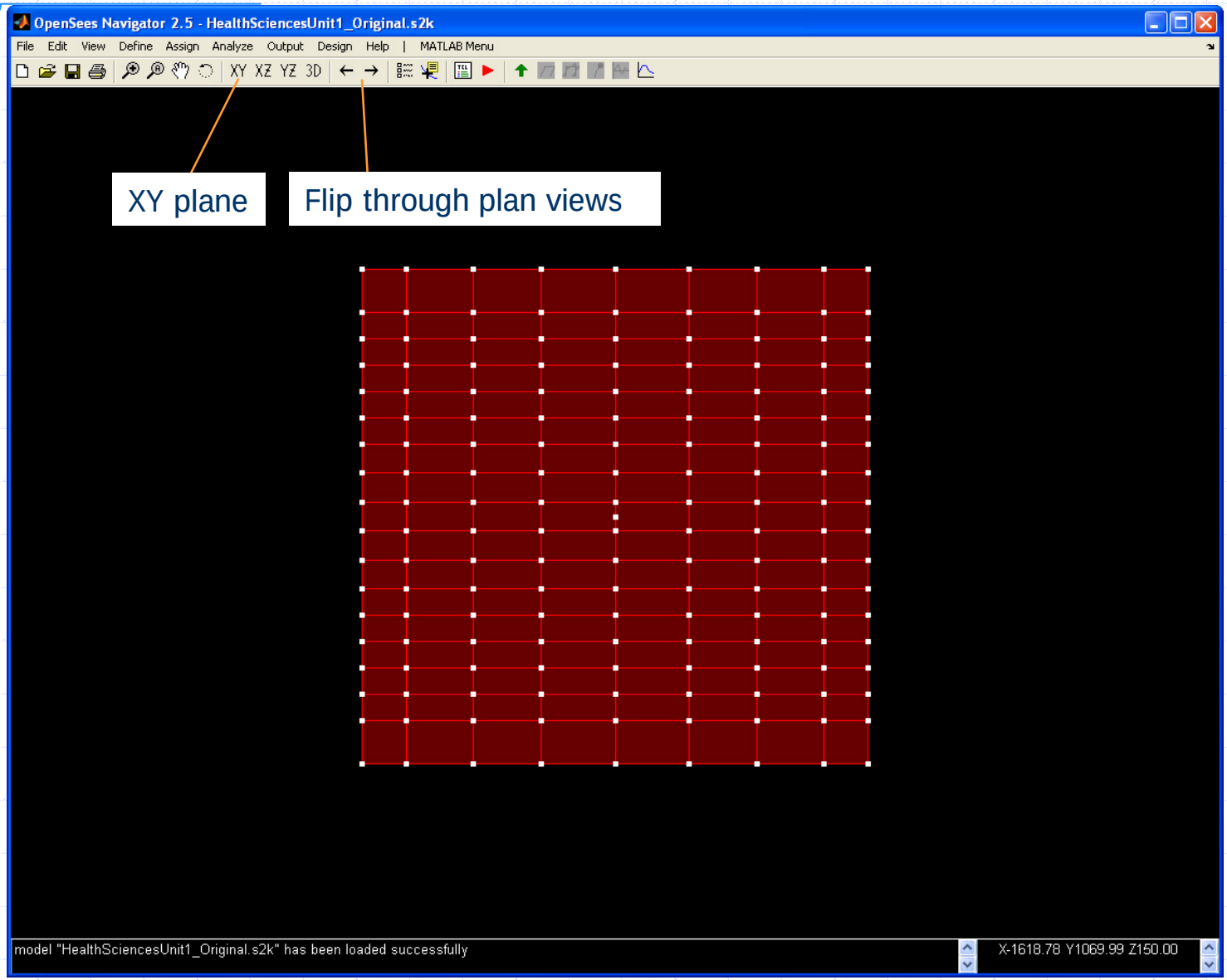
Define geometry: import from SAP2000



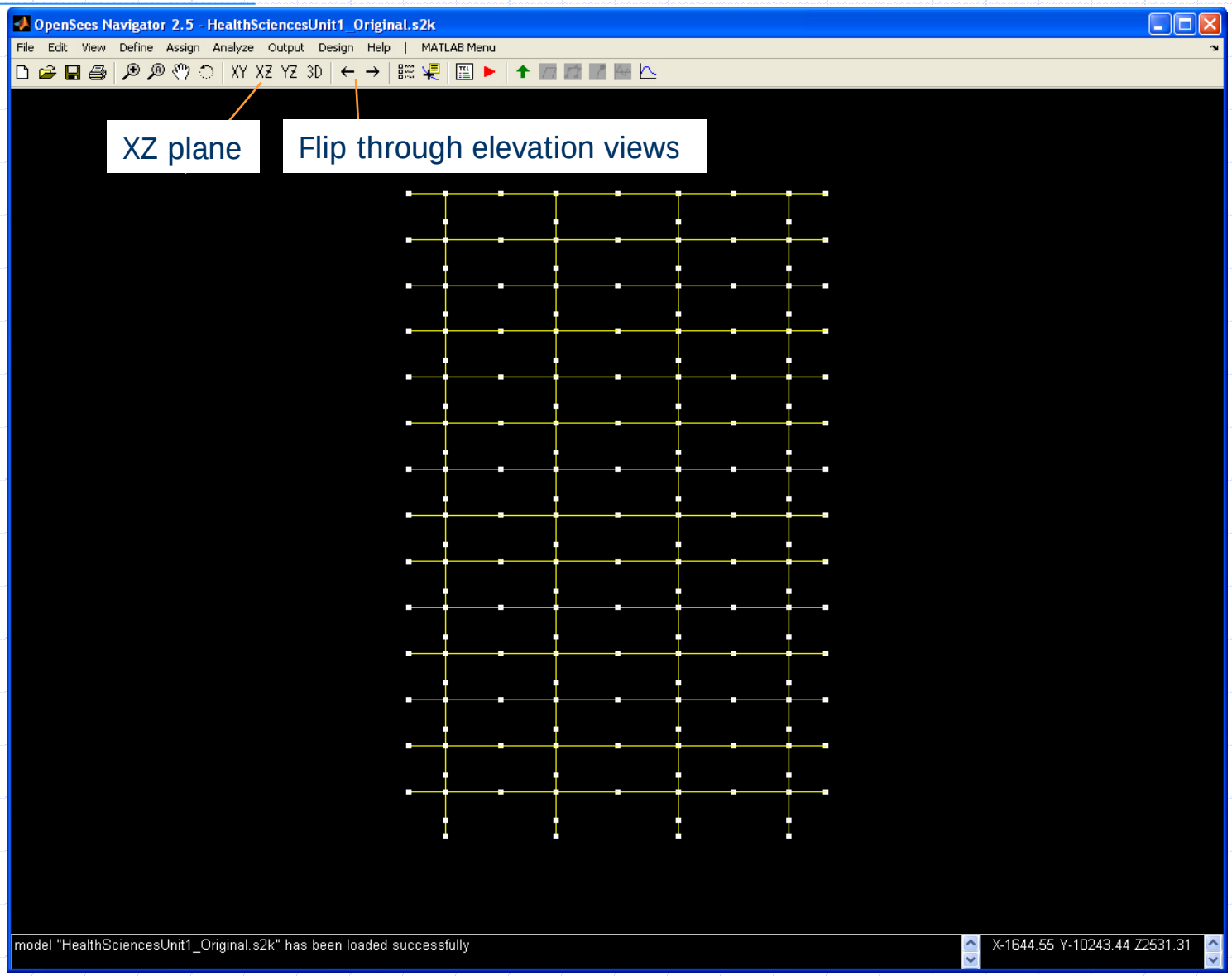
Define geometry: import from SAP2000



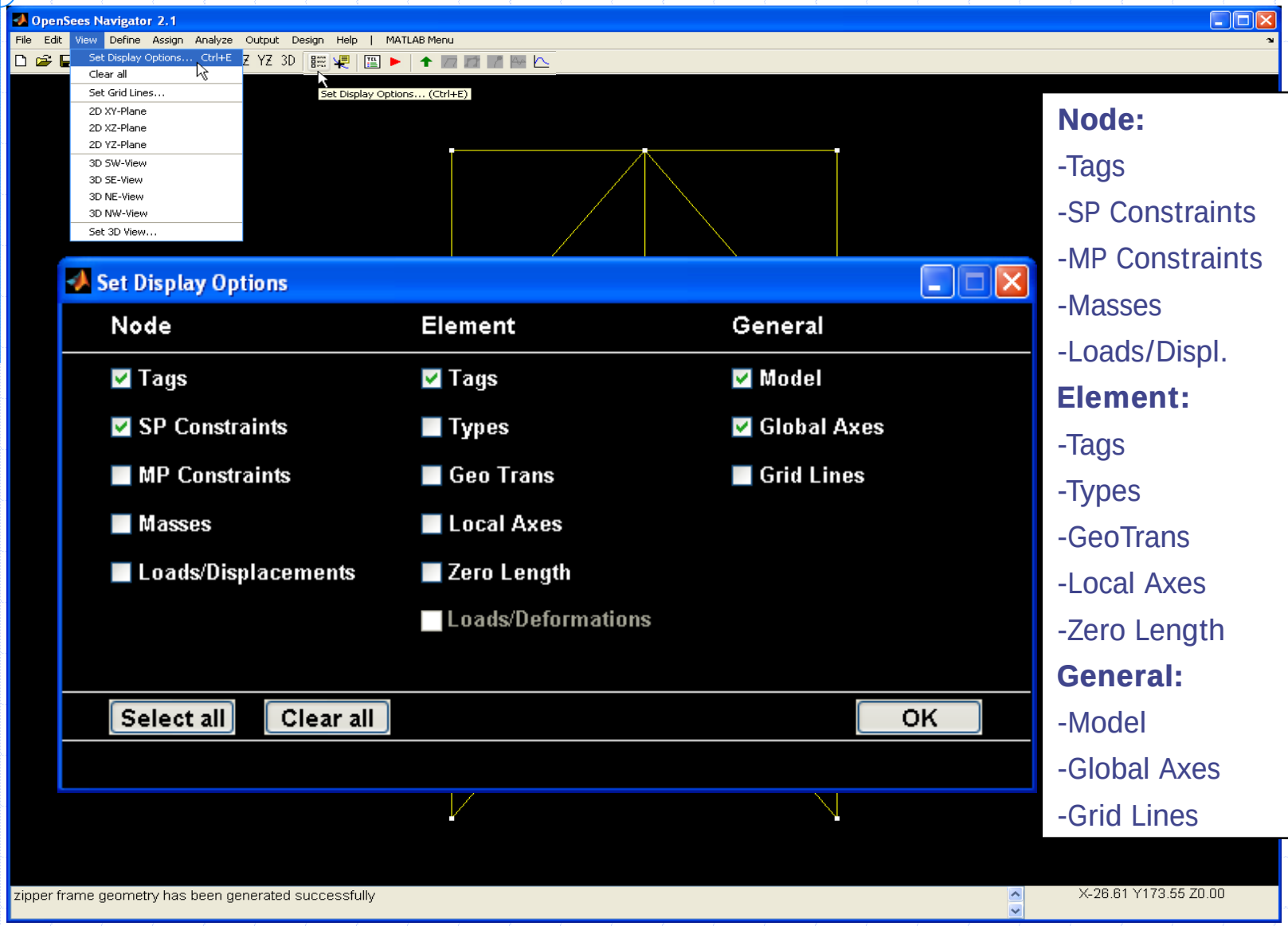
Define geometry: import from SAP2000



Define geometry: import from SAP2000



View geometry: set display options



Node:

-Tags

-SP Constraints

-MP Constraints

-Masses

-Loads/Displ.

Element:

-Tags

-Types

-GeoTrans

-Local Axes

-Zero Length

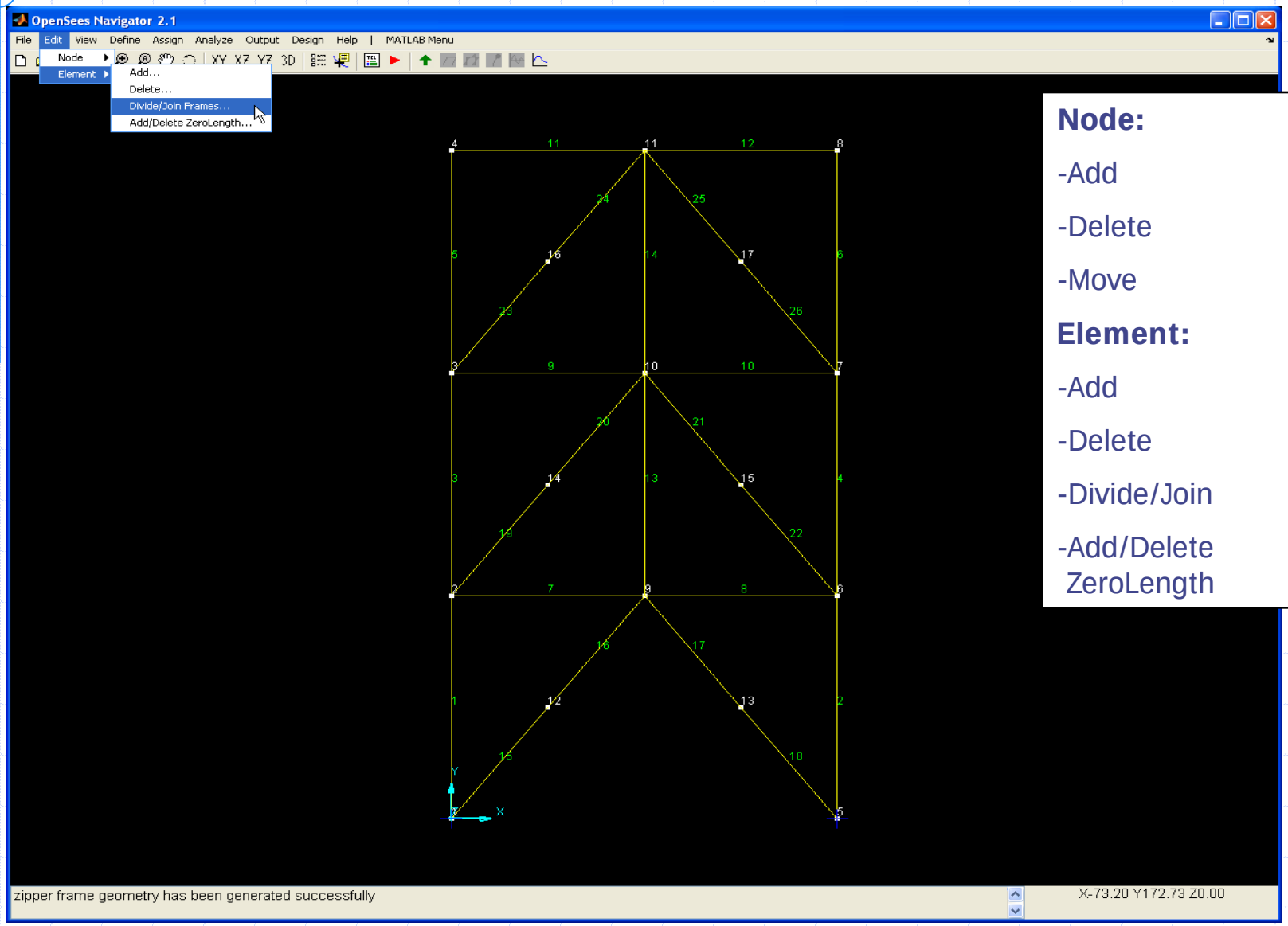
General:

-Model

-Global Axes

-Grid Lines

Edit geometry



Node:

-Add

-Delete

-Move

Element:

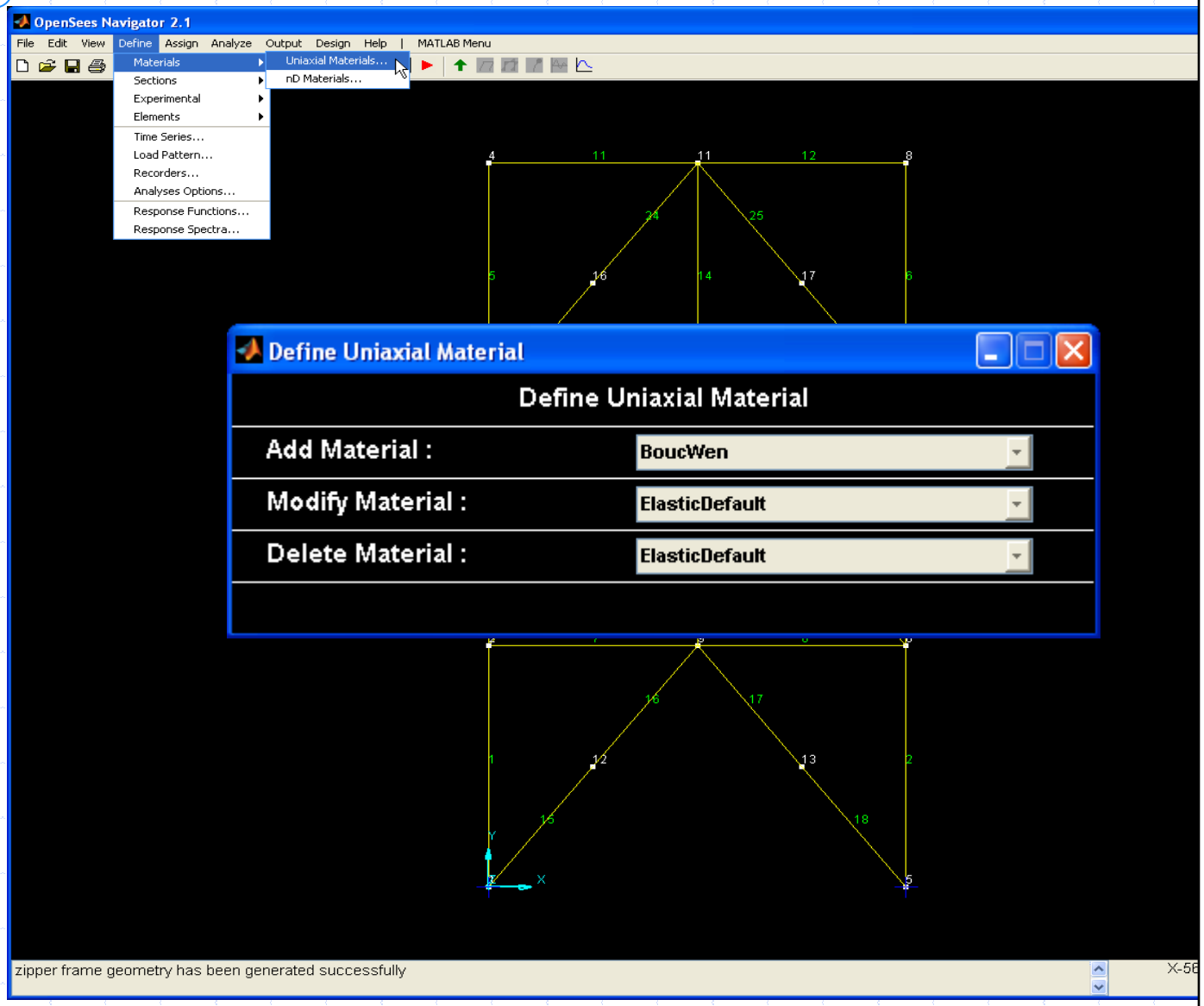
-Add

-Delete

-Divide/Join

-Add/Delete ZeroLength

Define material: uniaxial materials



Templates:

-BoucWen

-Concrete01

-Concrete02

-Concrete03

-Concrete04

-Elastic

-ElasticNoTension

-ElasticPP

-ElasticPPGap

-Fatigue

-Hardening

-Hysteretic

-MinMax

-Parallel

-ReinforcingSteel

-Series

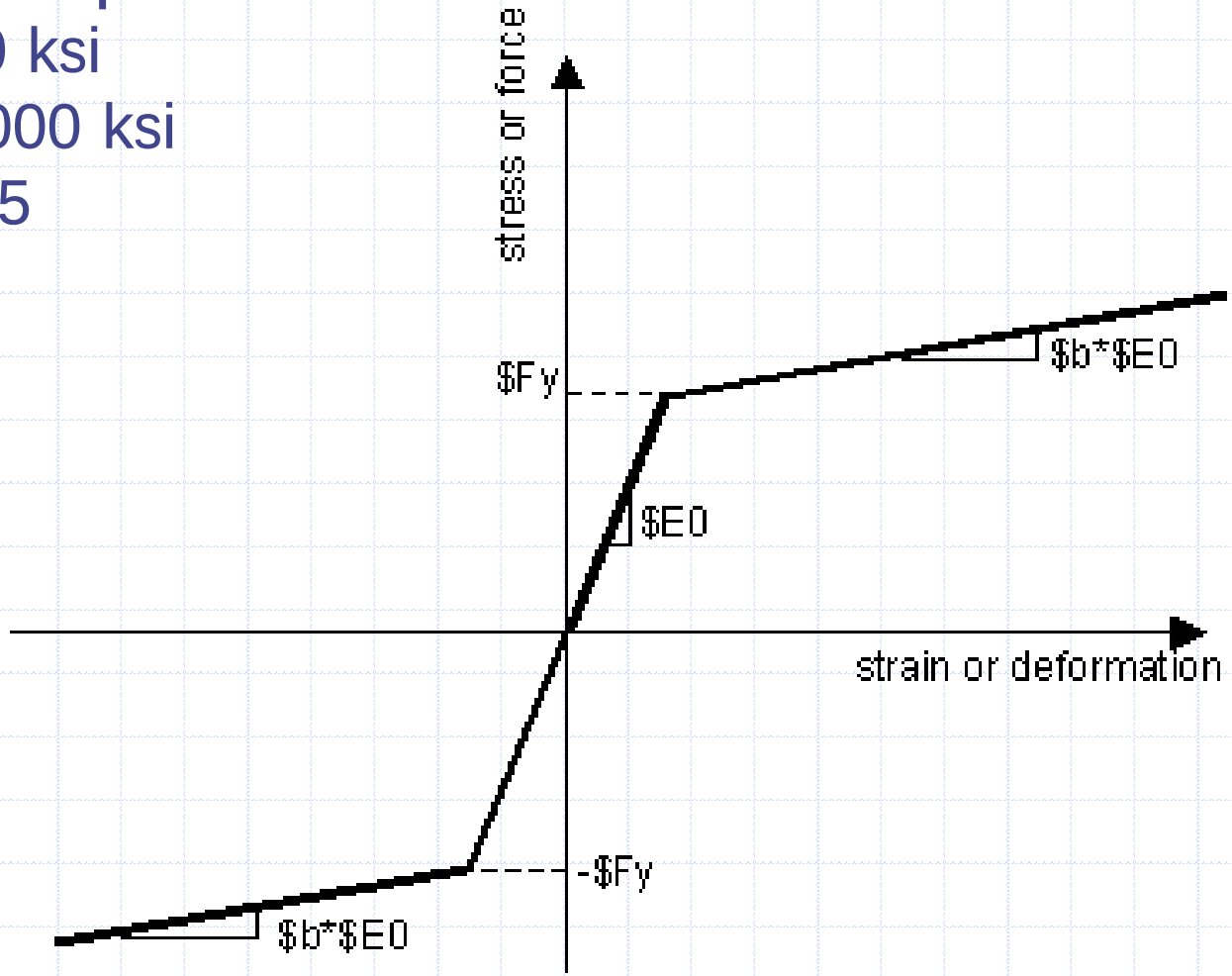
-Steel01

-Steel02

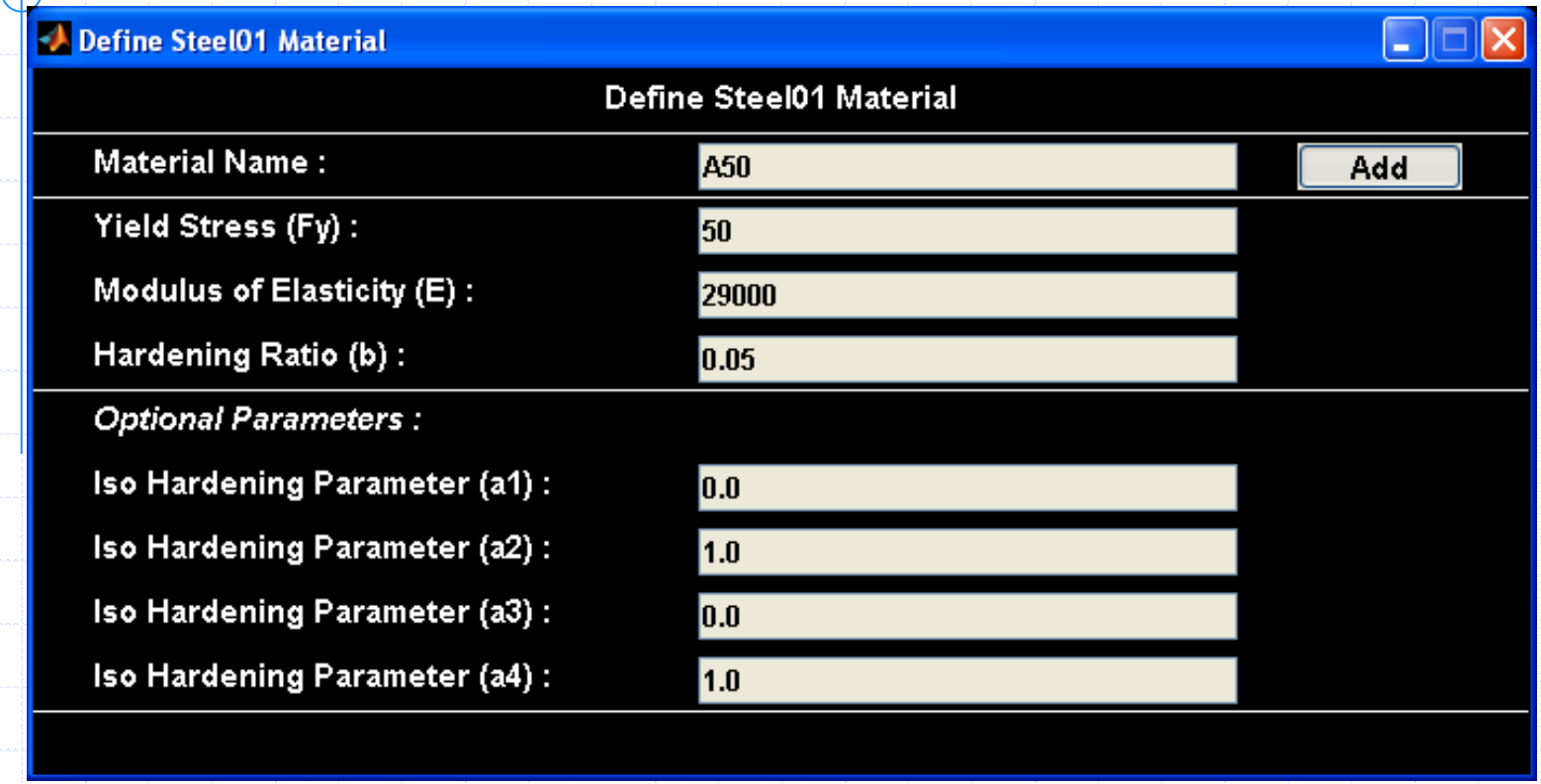
-Viscous

Define uniaxial material: Steel01

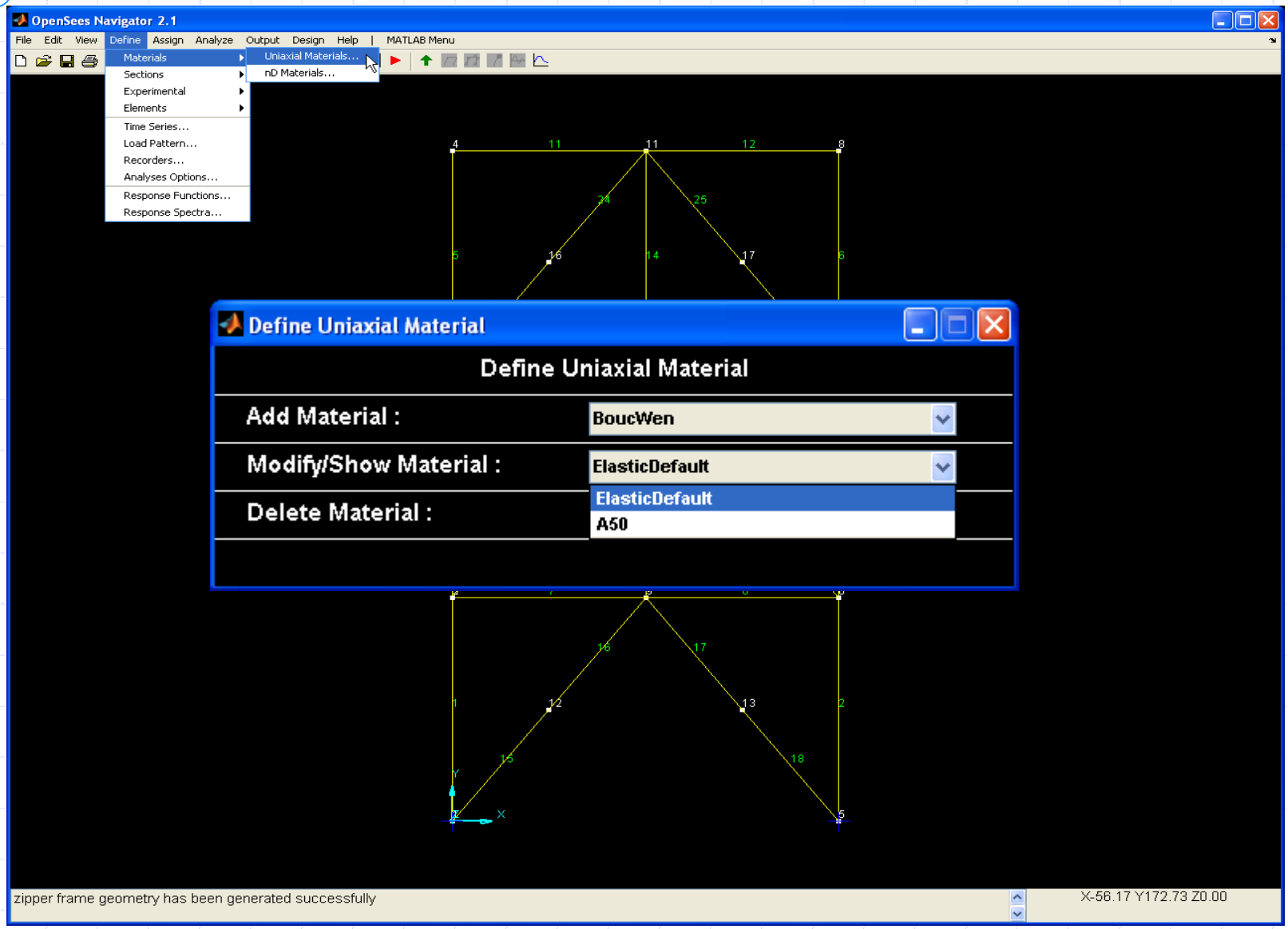
- Material properties
  - Fy = 50 ksi
  - E = 29000 ksi
  - b = 0.05



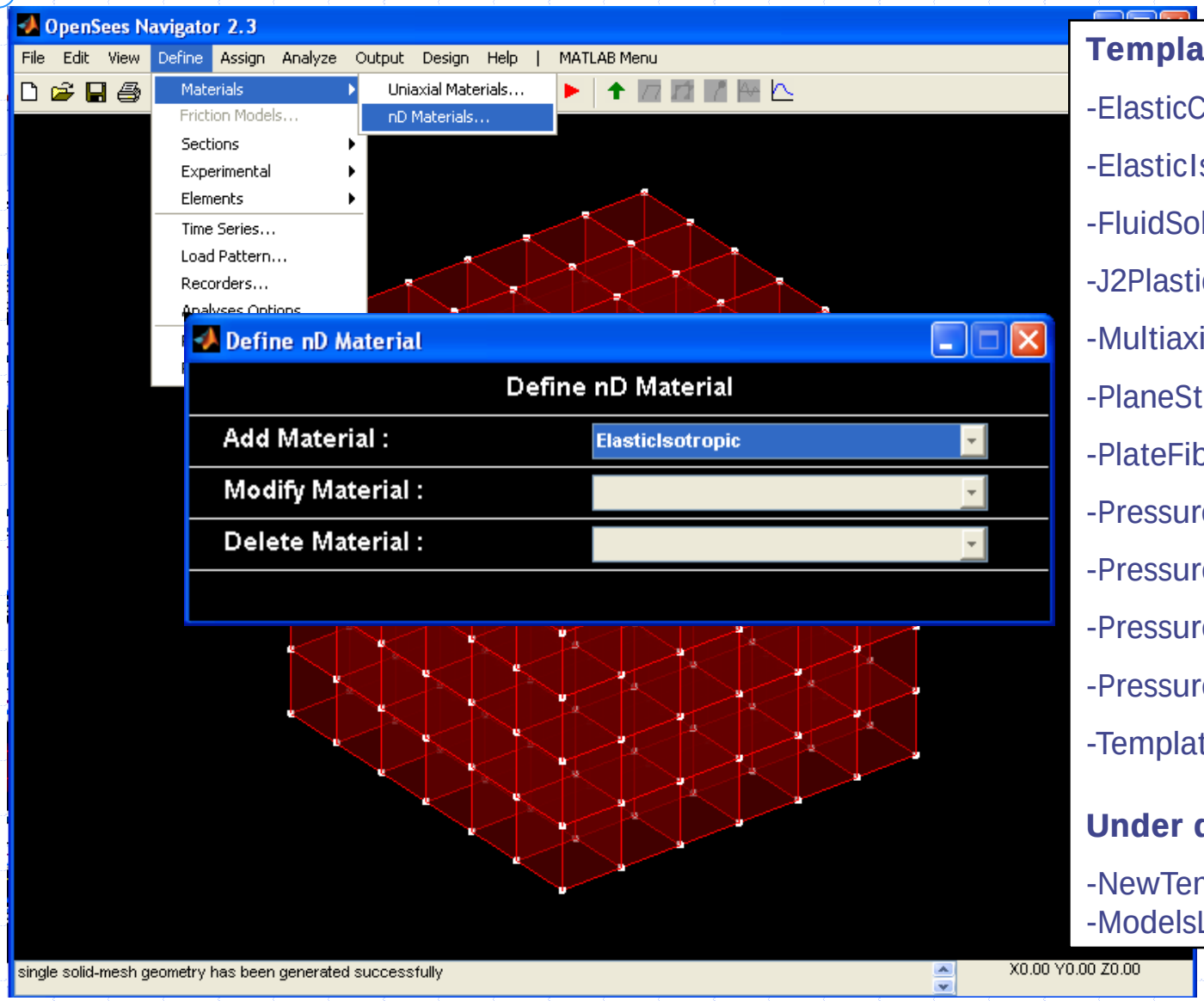
Define uniaxial material: Steel01



Define material: uniaxial materials



Define material: nD materials



Templates:

-ElasticCrossAnisotropic3D

-ElasticIsotropic

-FluidSolidPorous

-J2Plasticity

-MultiaxialCyclicPlasticity

-PlaneStress

-PlateFiber

-PressureDependMultiYield

-PressureDependMultiYield02

-PressureDependentElastic3D

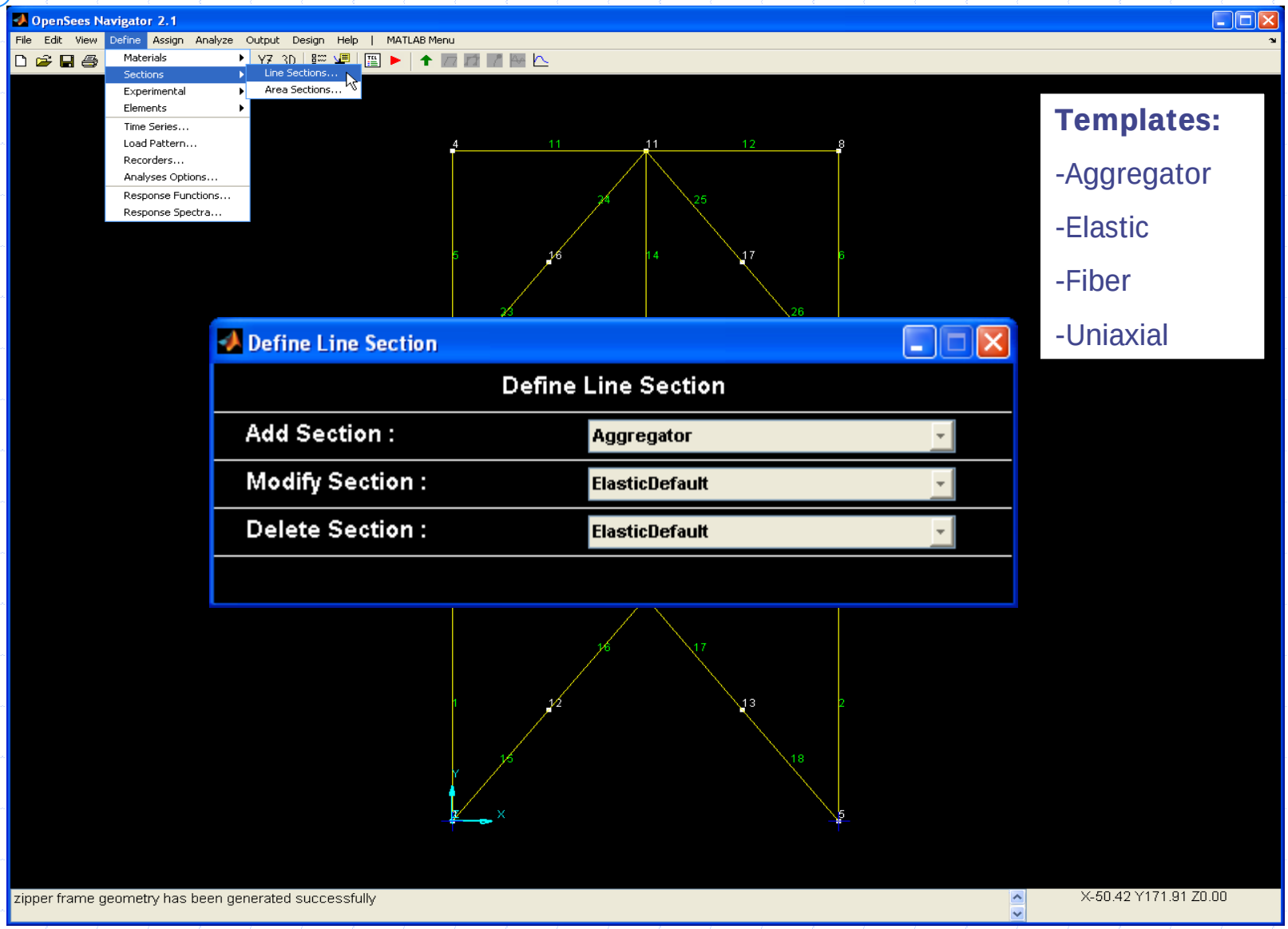
-PressureIndependMultiYield

-Template3DElastoPlastic

Under development:

-NewTemplate3DElastoPlastic
-ModelsLargeDeformation

Define section: line sections



Templates:

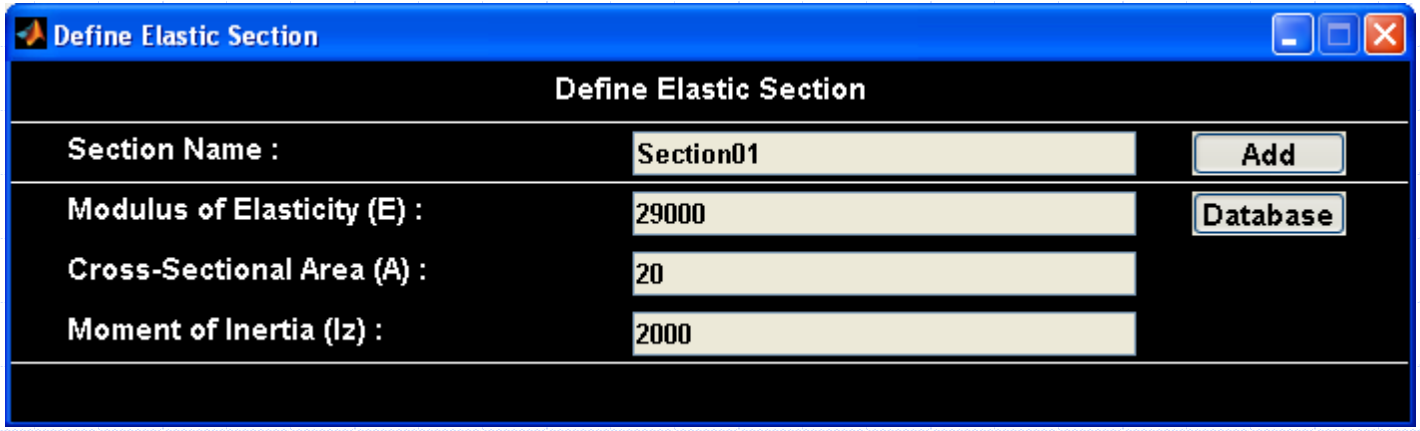
-Aggregator

-Elastic

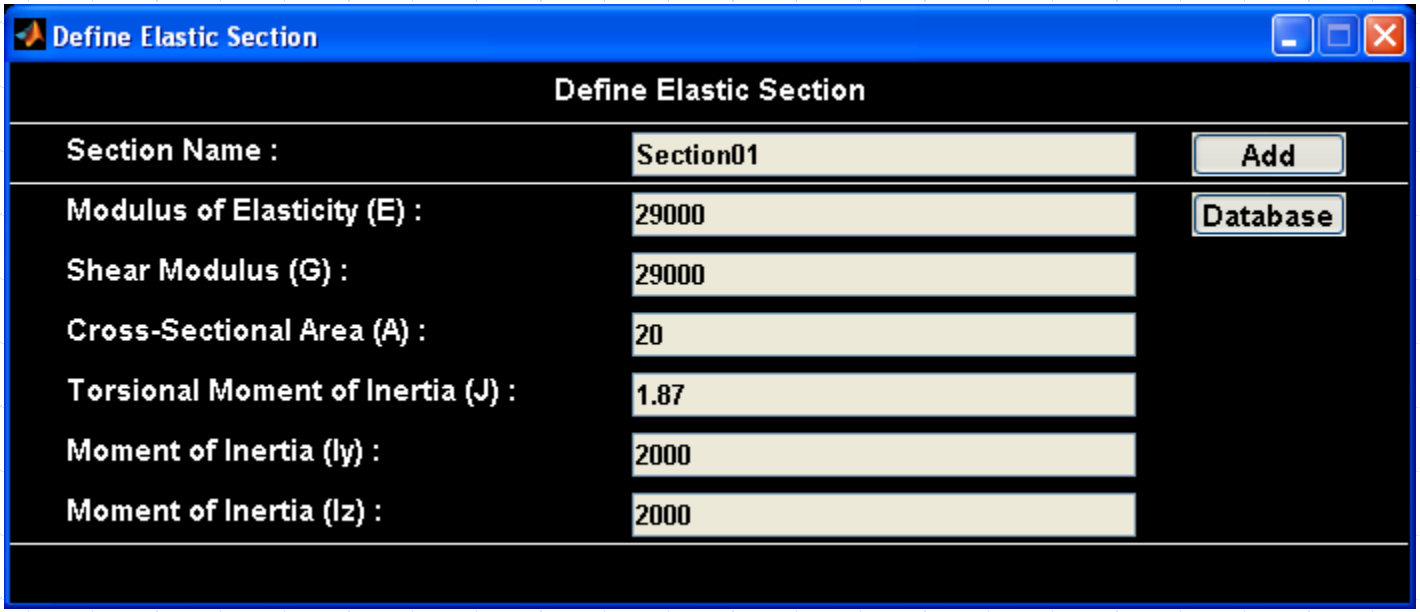
-Fiber

-Uniaxial

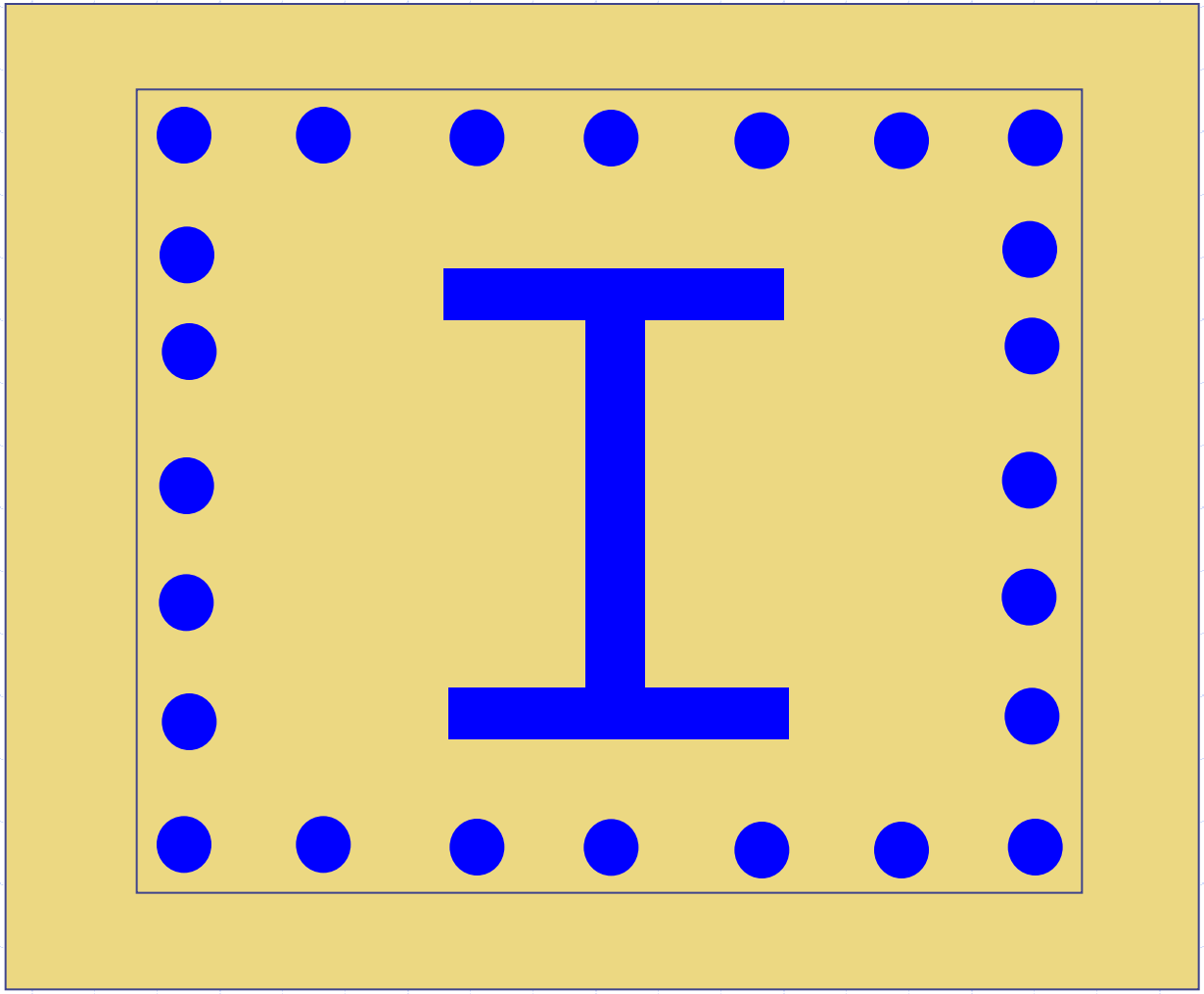
Define line section: elastic section



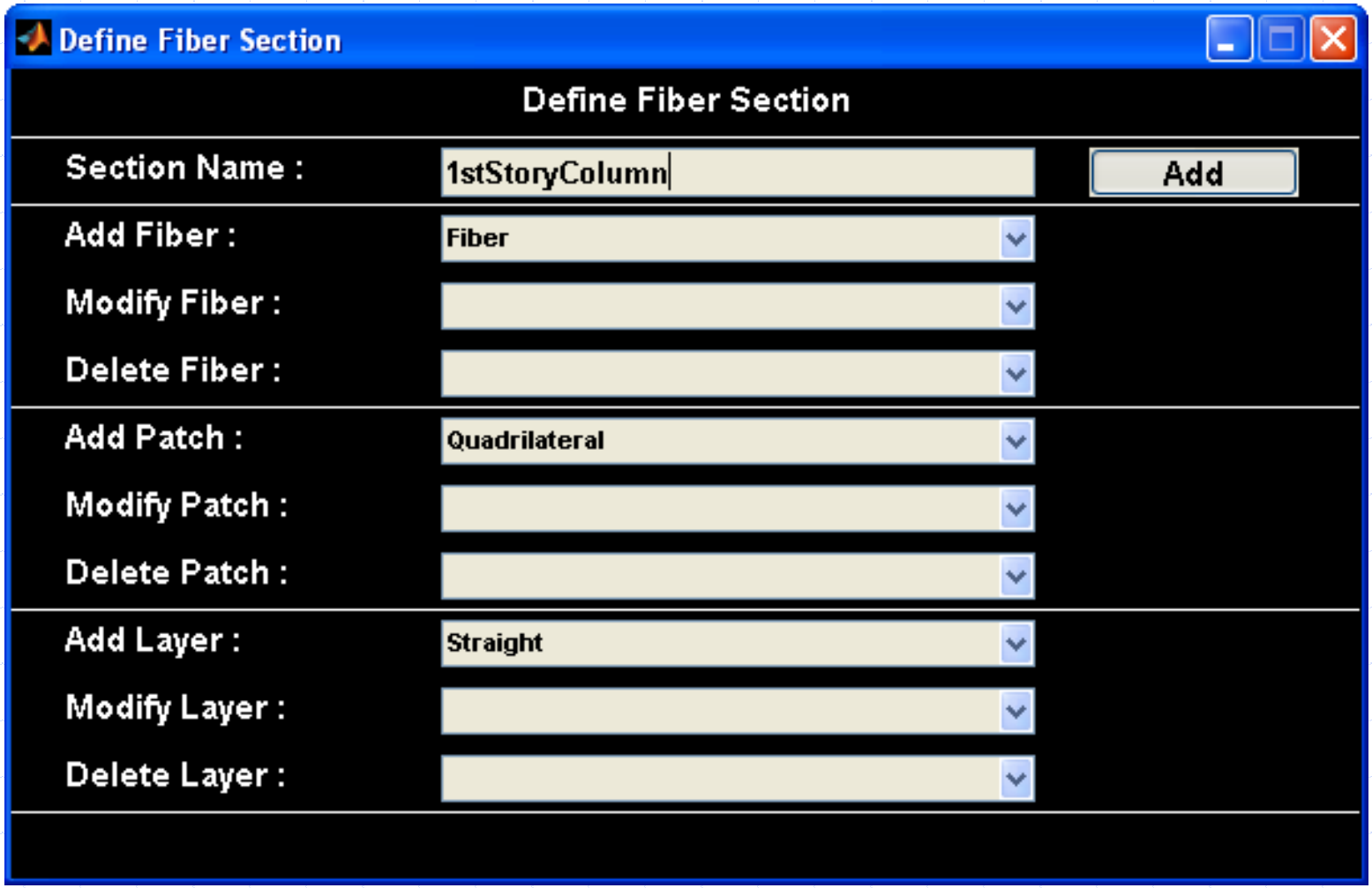
If the model is 3D



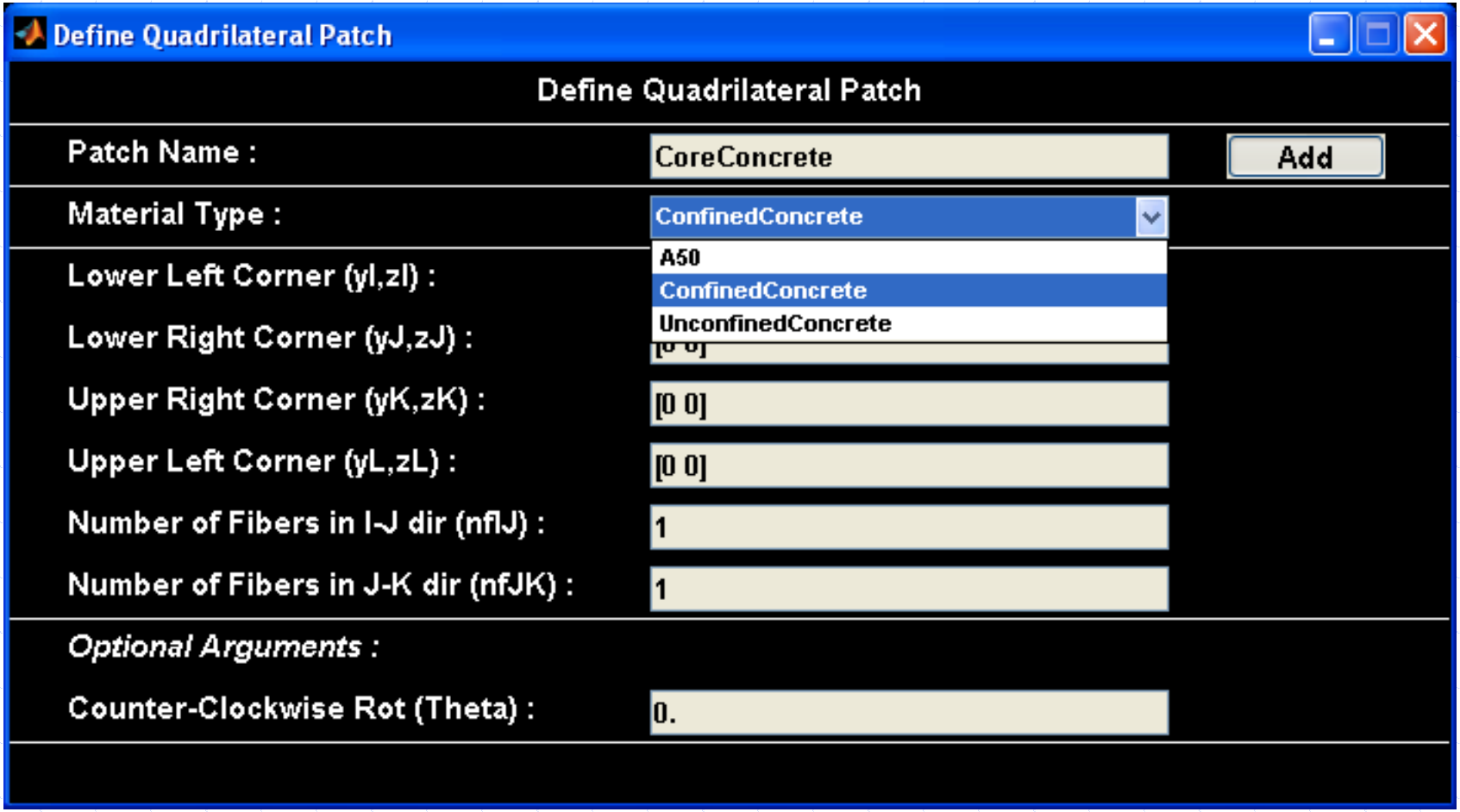
Define fiber section: Composite patch



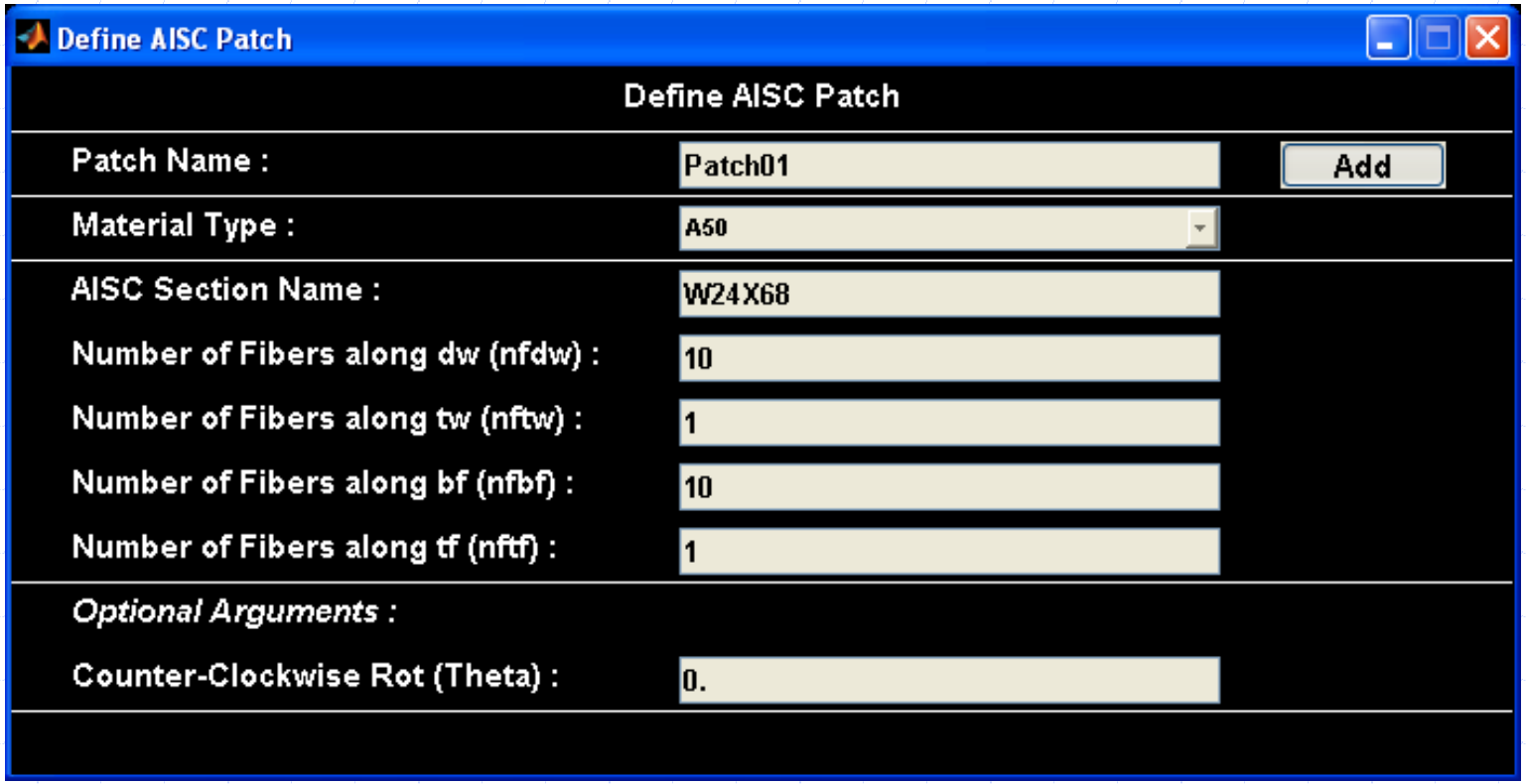
Define line section: fiber section



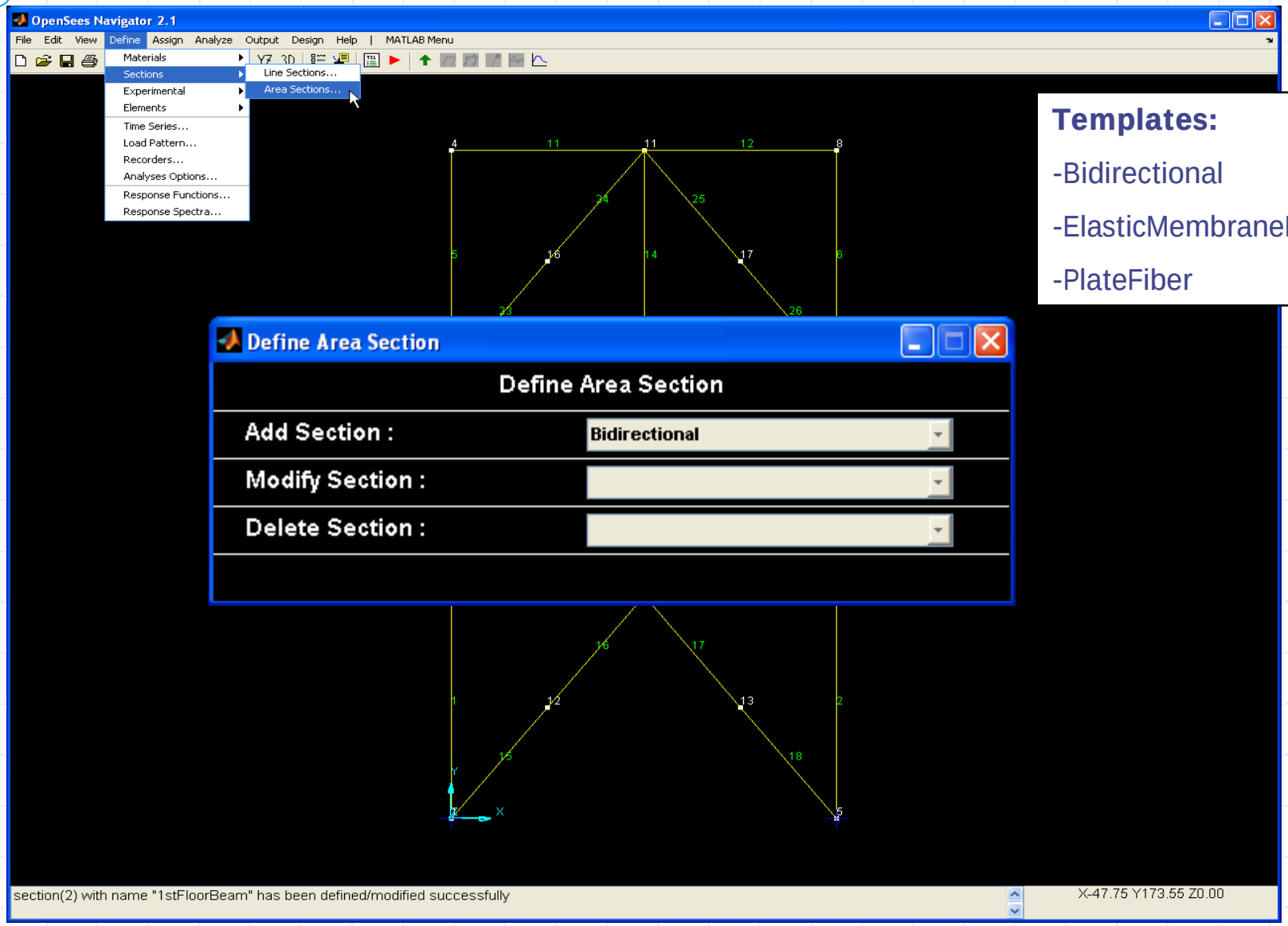
Define line section: quadrilateral patch



Define fiber section: AISC patch



Define section: area section



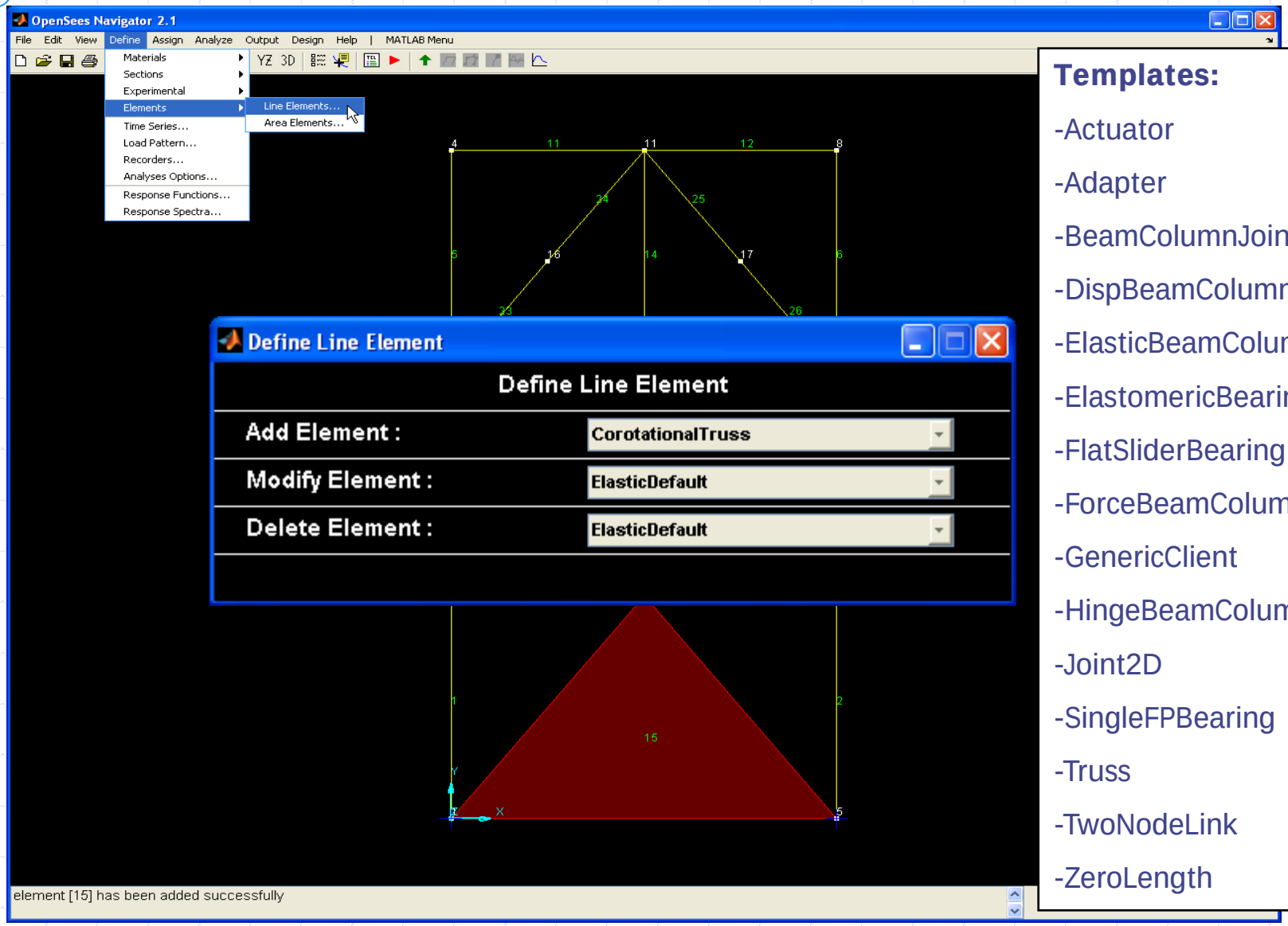
Templates:

-Bidirectional

-ElasticMembranePlate

-PlateFiber

Define element: line element



Templates:

-Actuator

-Adapter

-BeamColumnJoint

-DispBeamColumn

-ElasticBeamColumn

-ElastomericBearing

-FlatSliderBearing

-ForceBeamColumn

-GenericClient

-HingeBeamColumn

-Joint2D

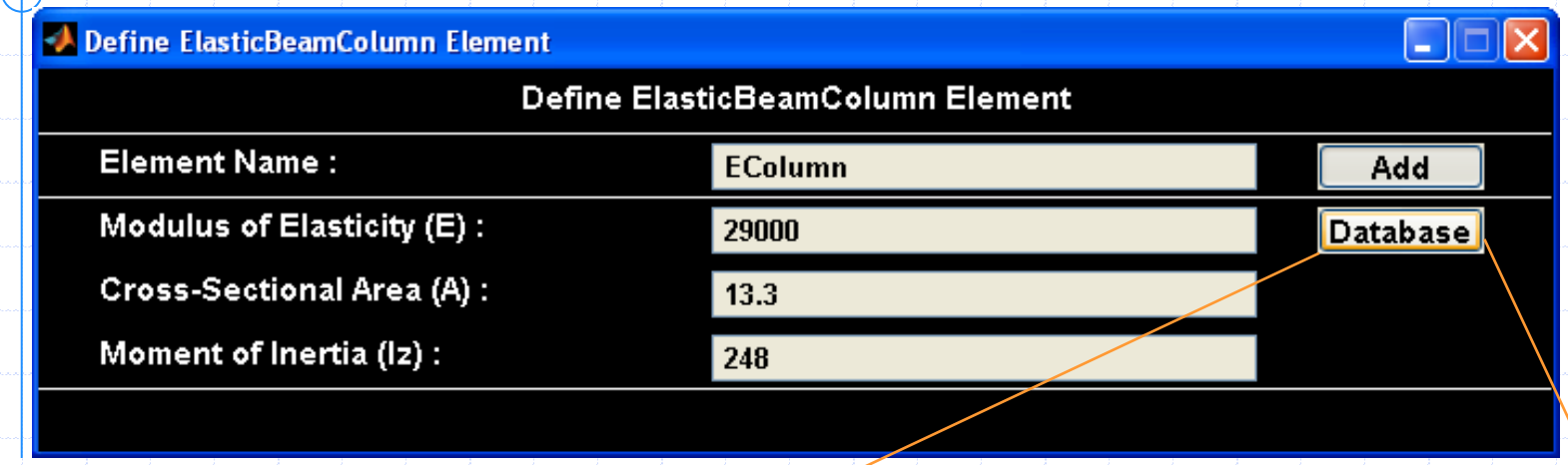
-SingleFPBearing

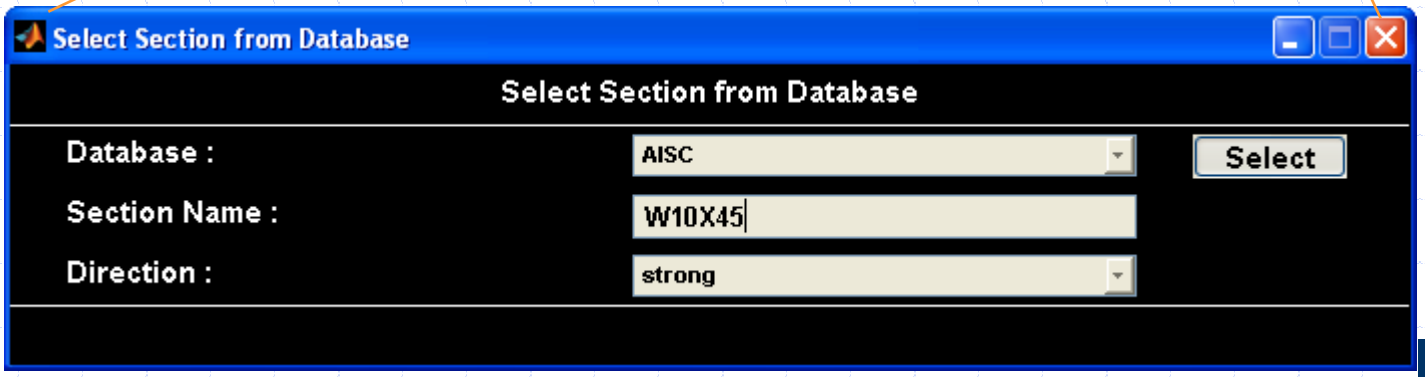
-Truss

-TwoNodeLink

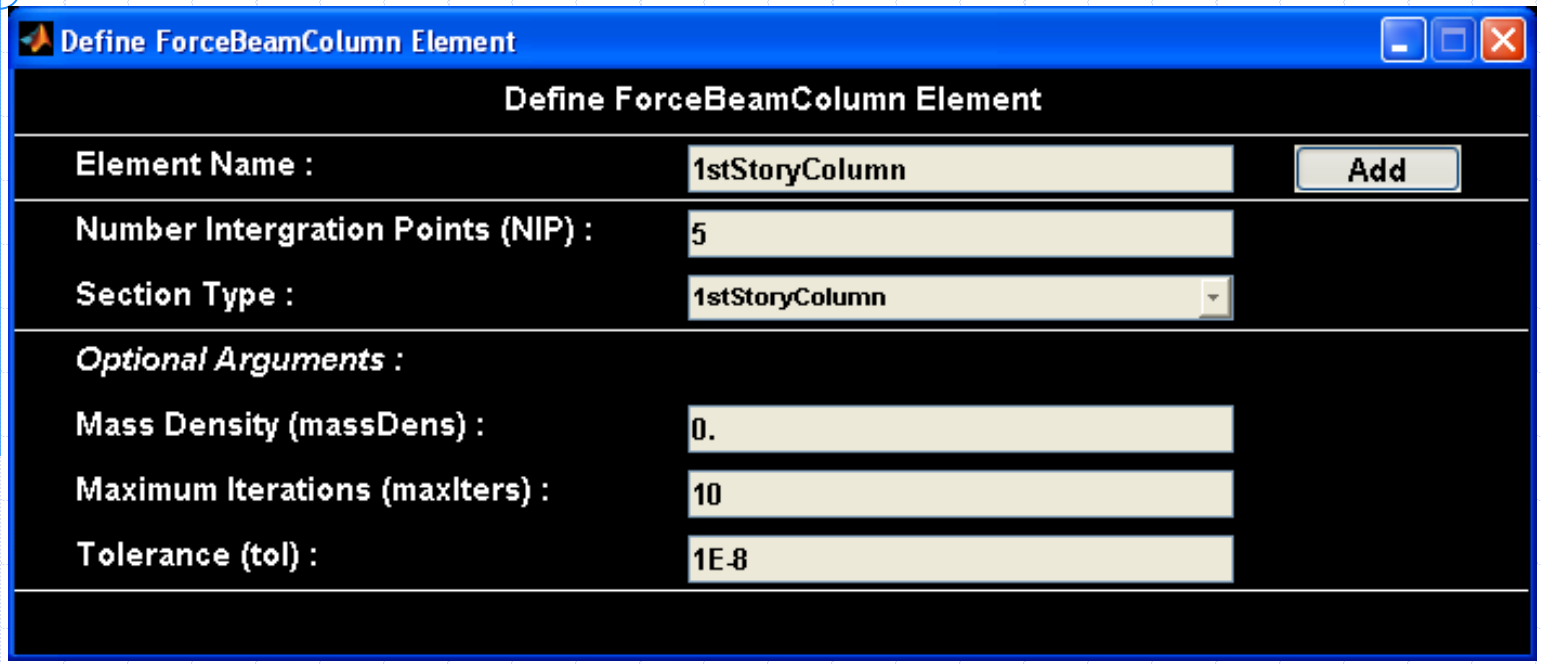
-ZeroLength

Define line element: ElasticBeamColumn

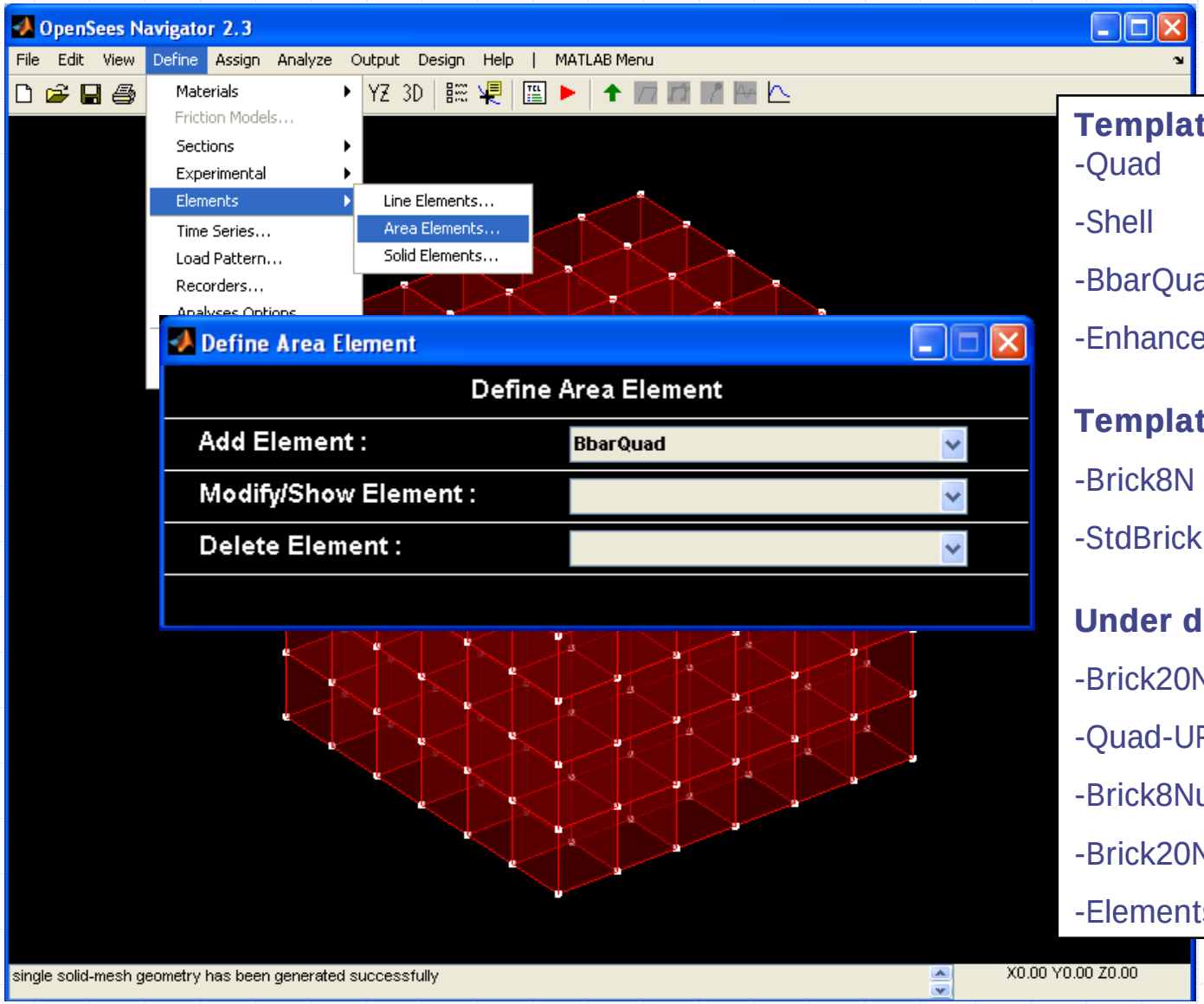




Define line element: ForceBeamColumn



Define element: area and solid elements



Templates (area):

-Quad

-Shell

-BbarQuad

-EnhancedQuad

Templates (solid):

-Brick8N

-StdBrick

Under development:

-Brick20N

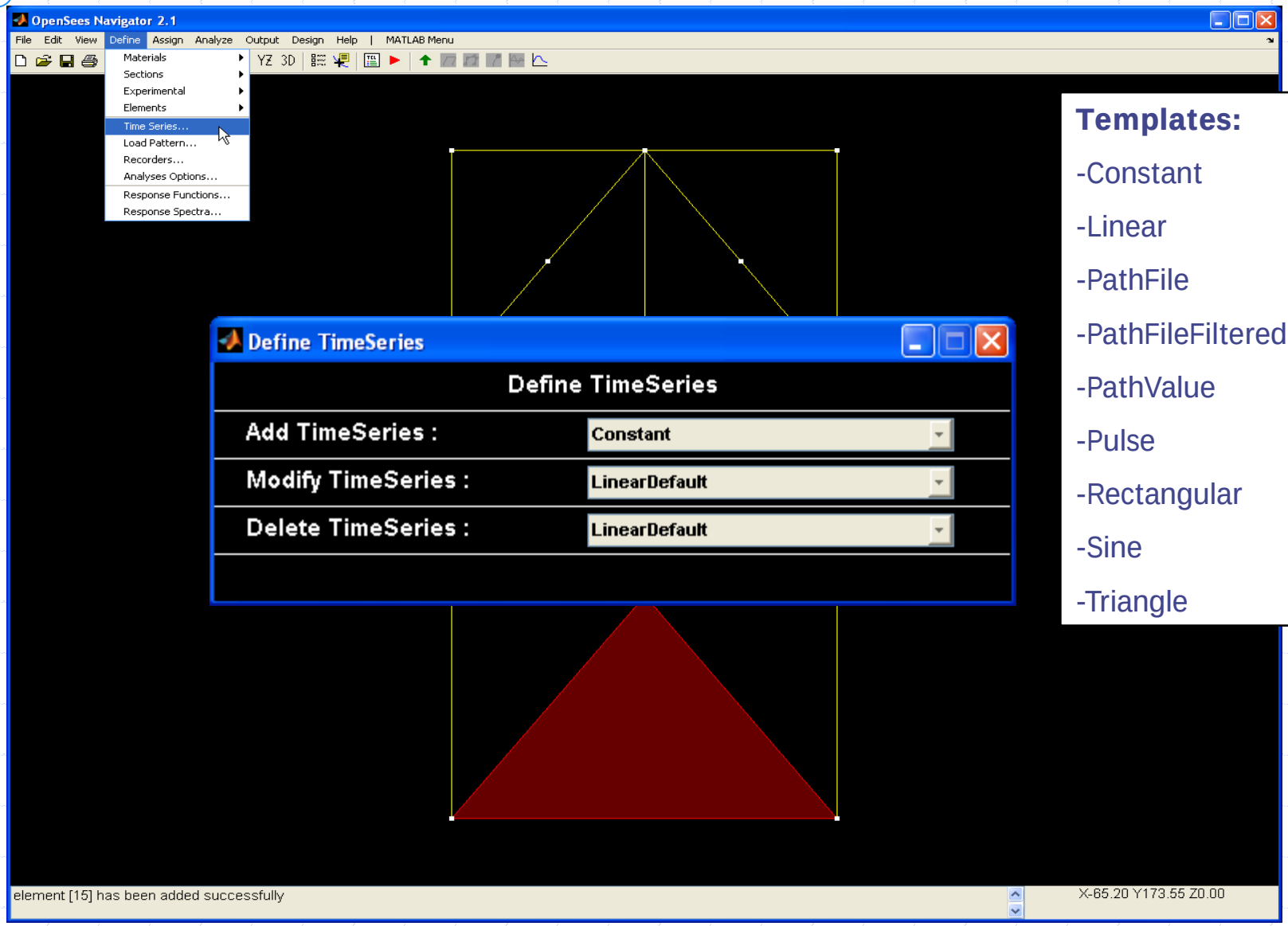
-Quad-UP

-Brick8Nu-p-U

-Brick20Nu-p-U

-ElementsLargeDeformation

Define TimeSeries:



Templates:

-Constant

-Linear

-PathFile

-PathFileFiltered

-PathValue

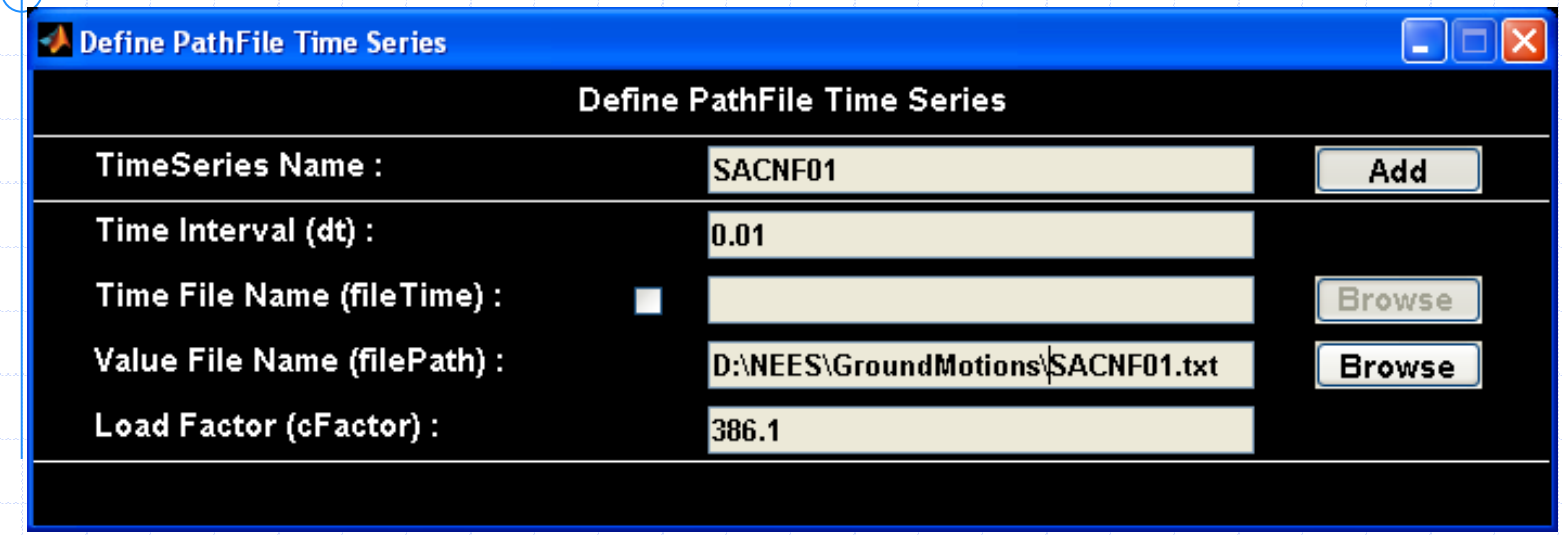
-Pulse

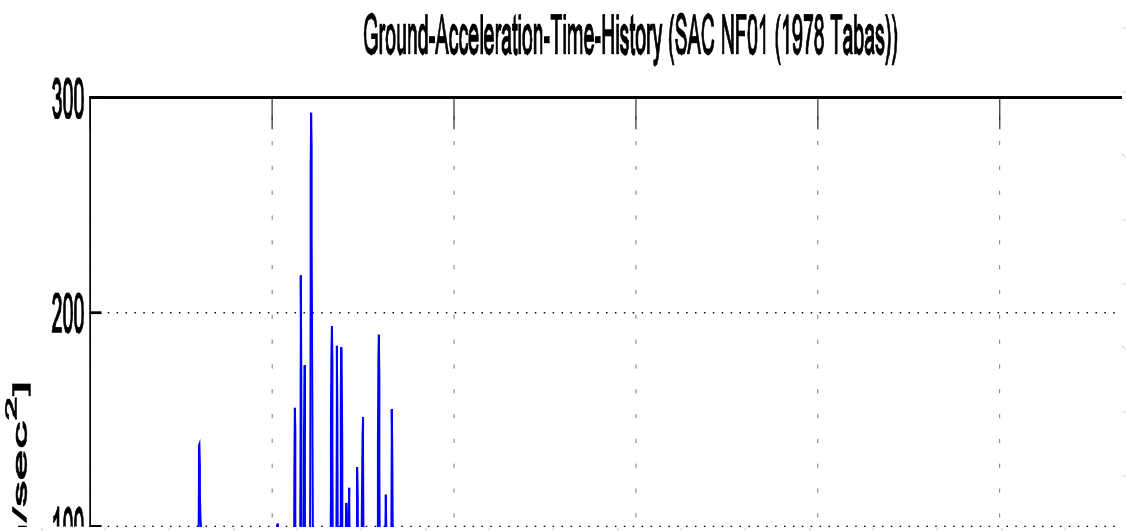
-Rectangular

-Sine

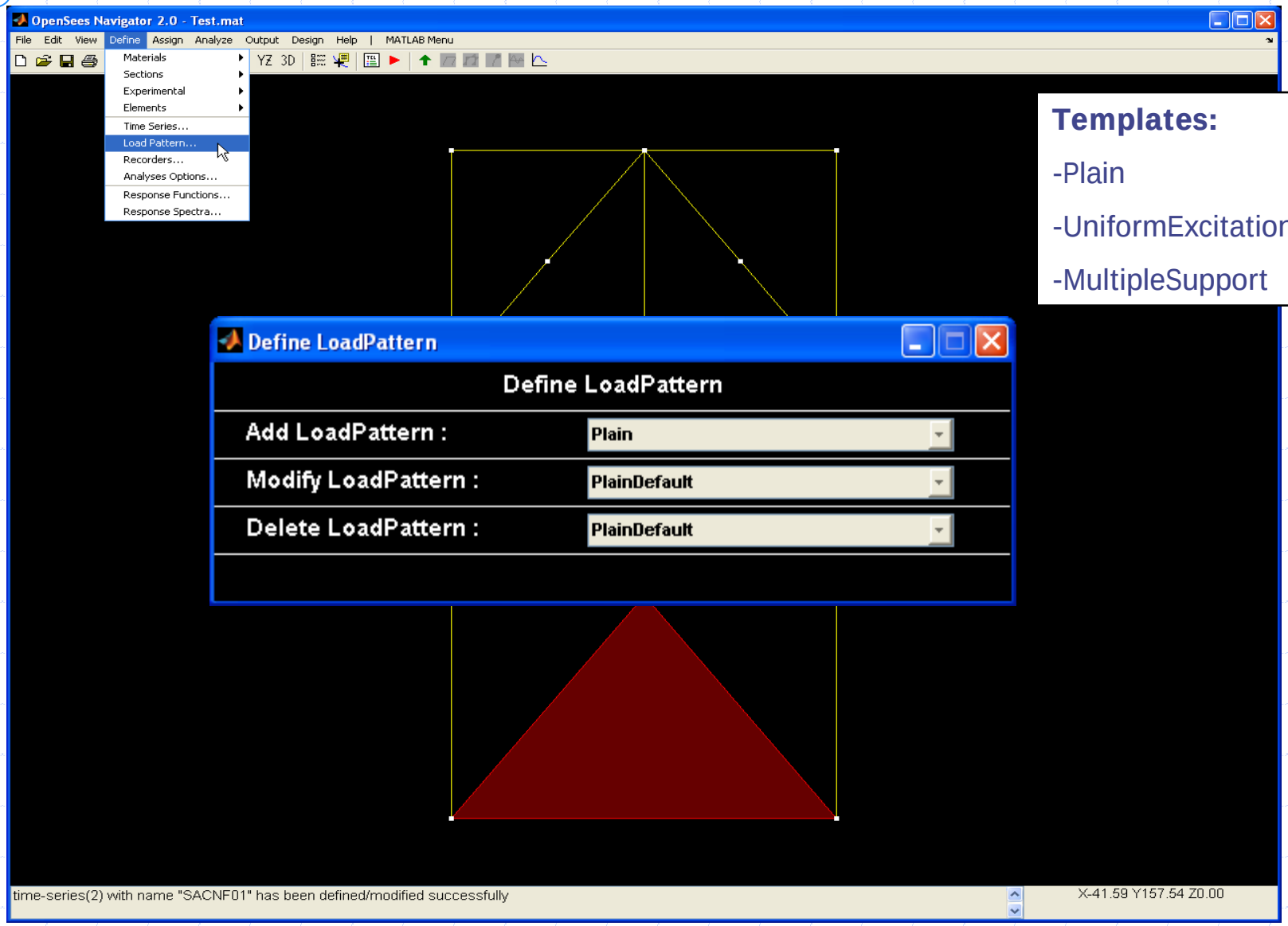
-Triangle

Define TimeSeries: PathFile





Define LoadPattern:



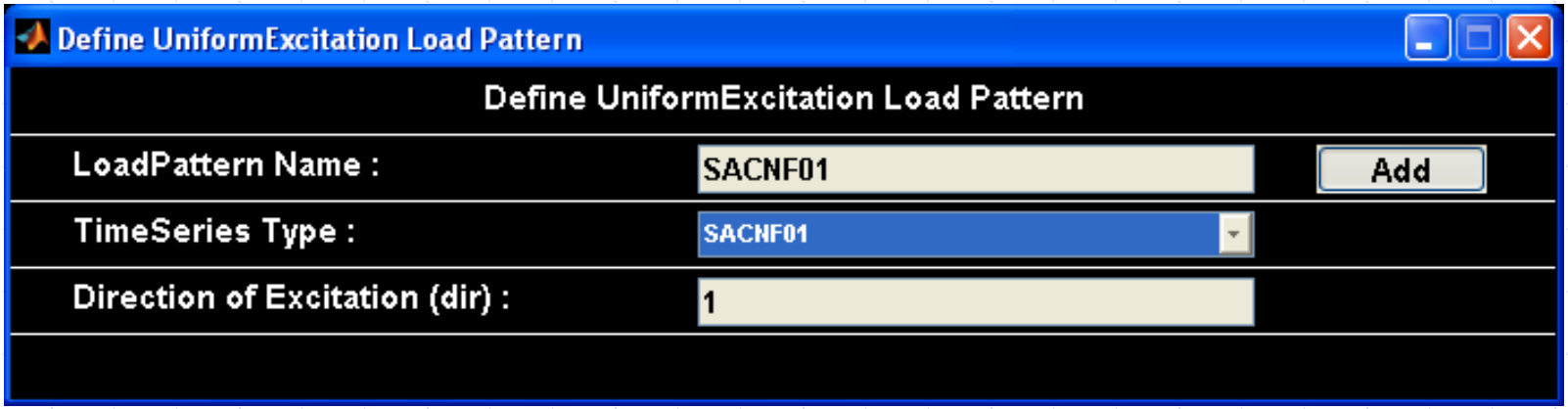
Templates:

-Plain

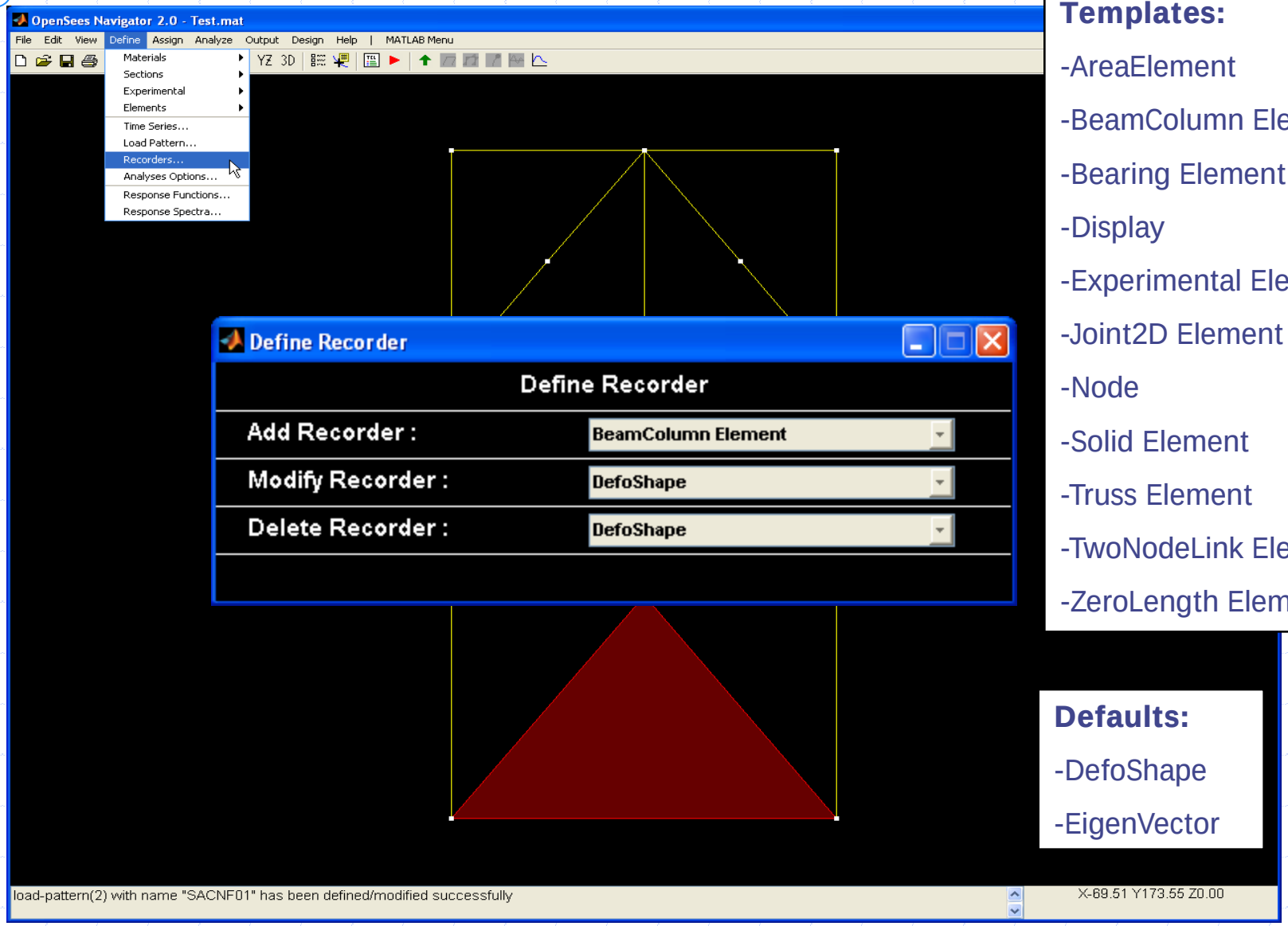
-UniformExcitation

-MultipleSupport

Define LoadPattern: UniformExcitation



Define recorder



Templates:

-AreaElement

-BeamColumn Element

-Bearing Element

-Display

-Experimental Element

-Joint2D Element

-Node

-Solid Element

-Truss Element

-TwoNodeLink Element

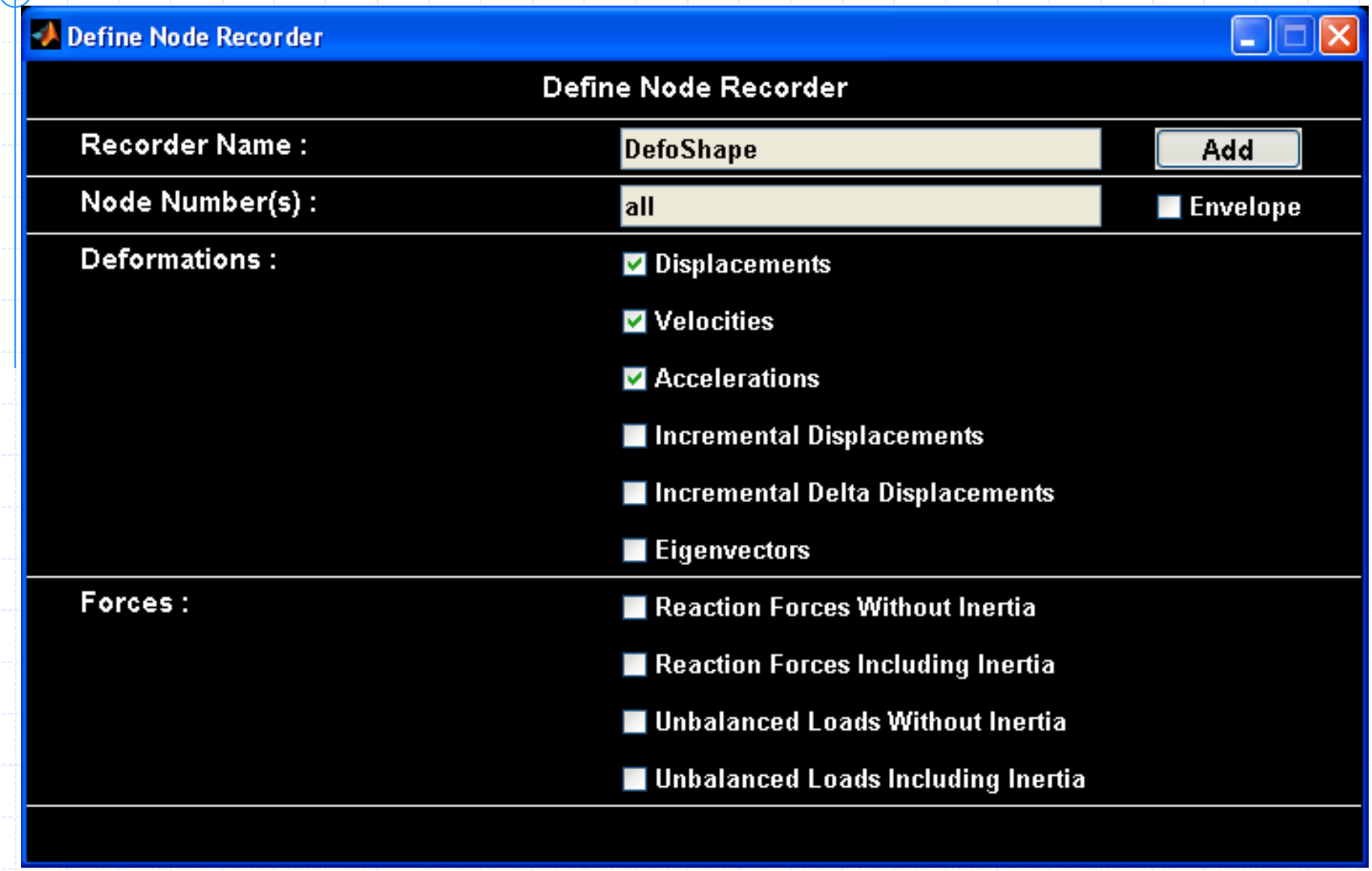
-ZeroLength Element

Defaults:

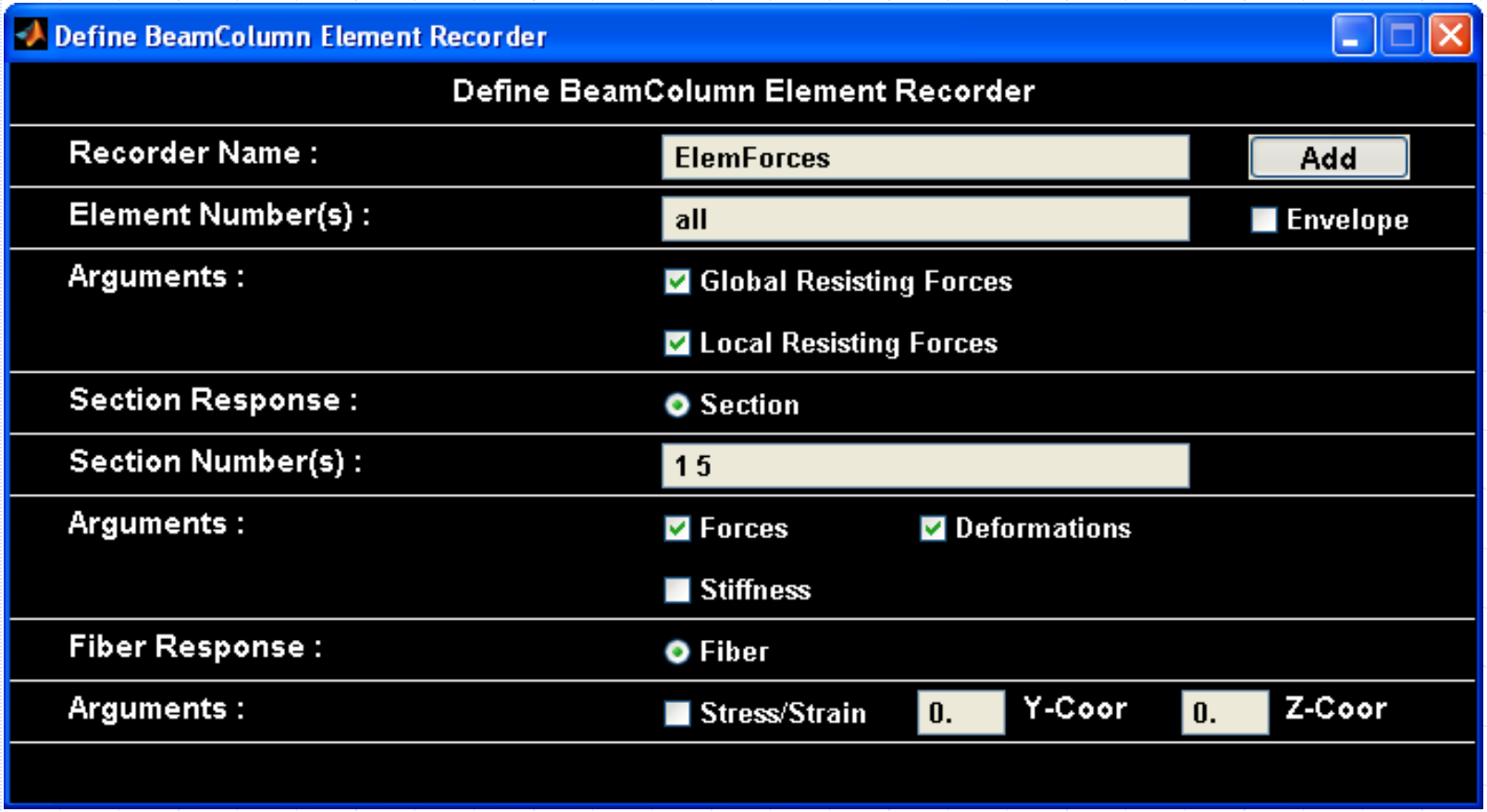
-DefoShape

-EigenVector

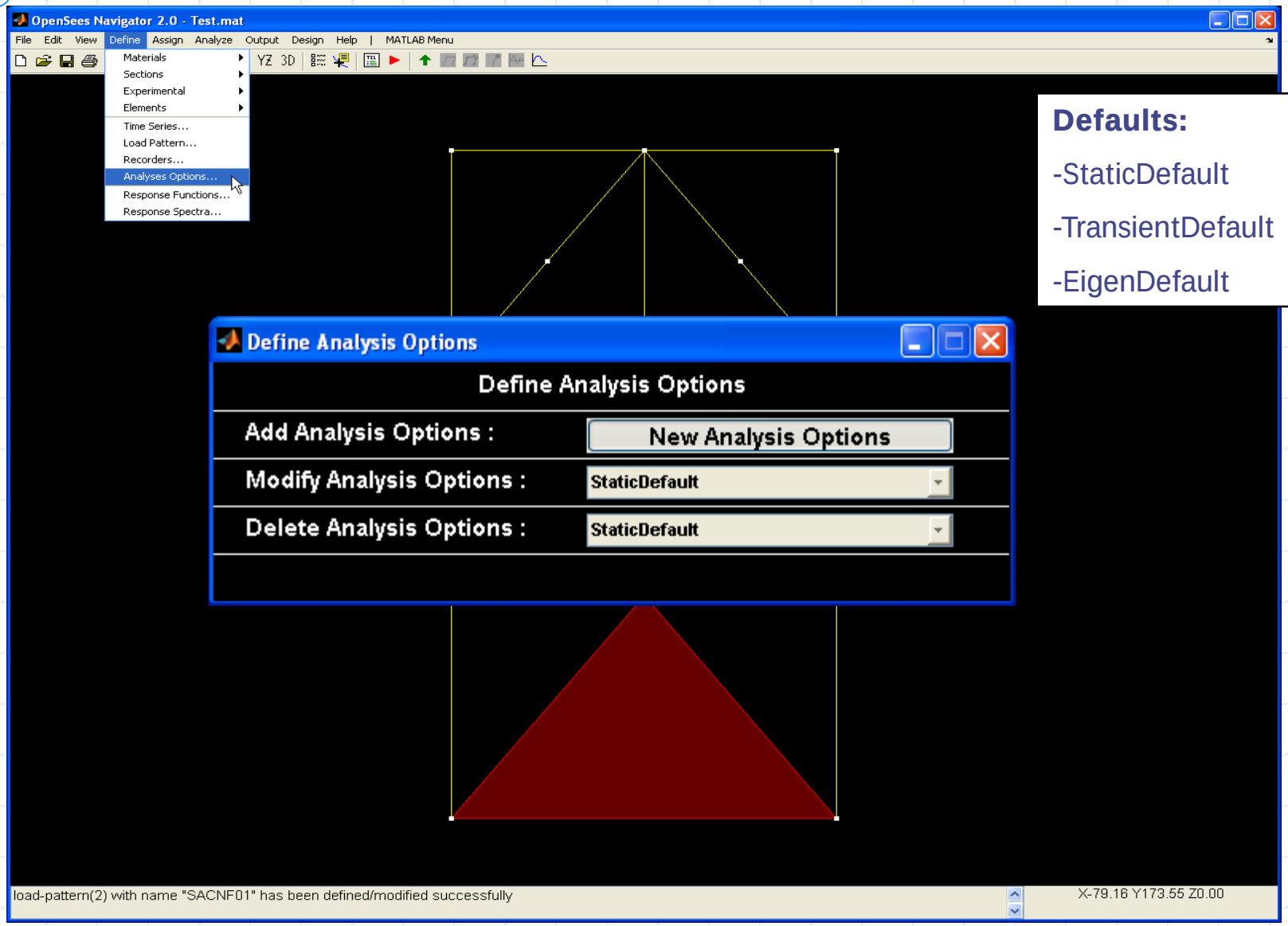
Define recorder: node recorder



Define recorder: BeamColumn recorder



Define analysis options



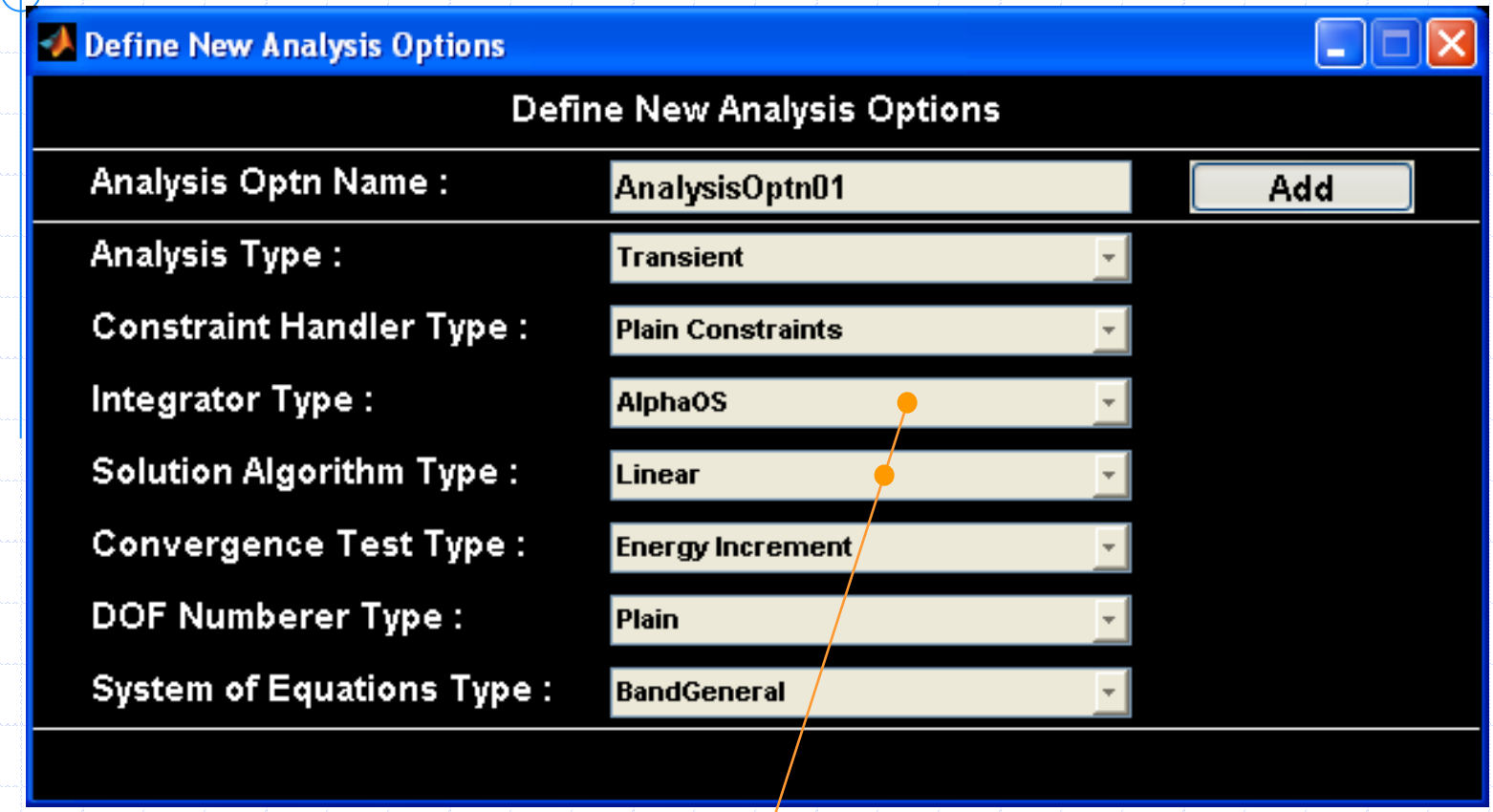
Defaults:

-StaticDefault

-TransientDefault

-EigenDefault

Define analysis options: new analysis



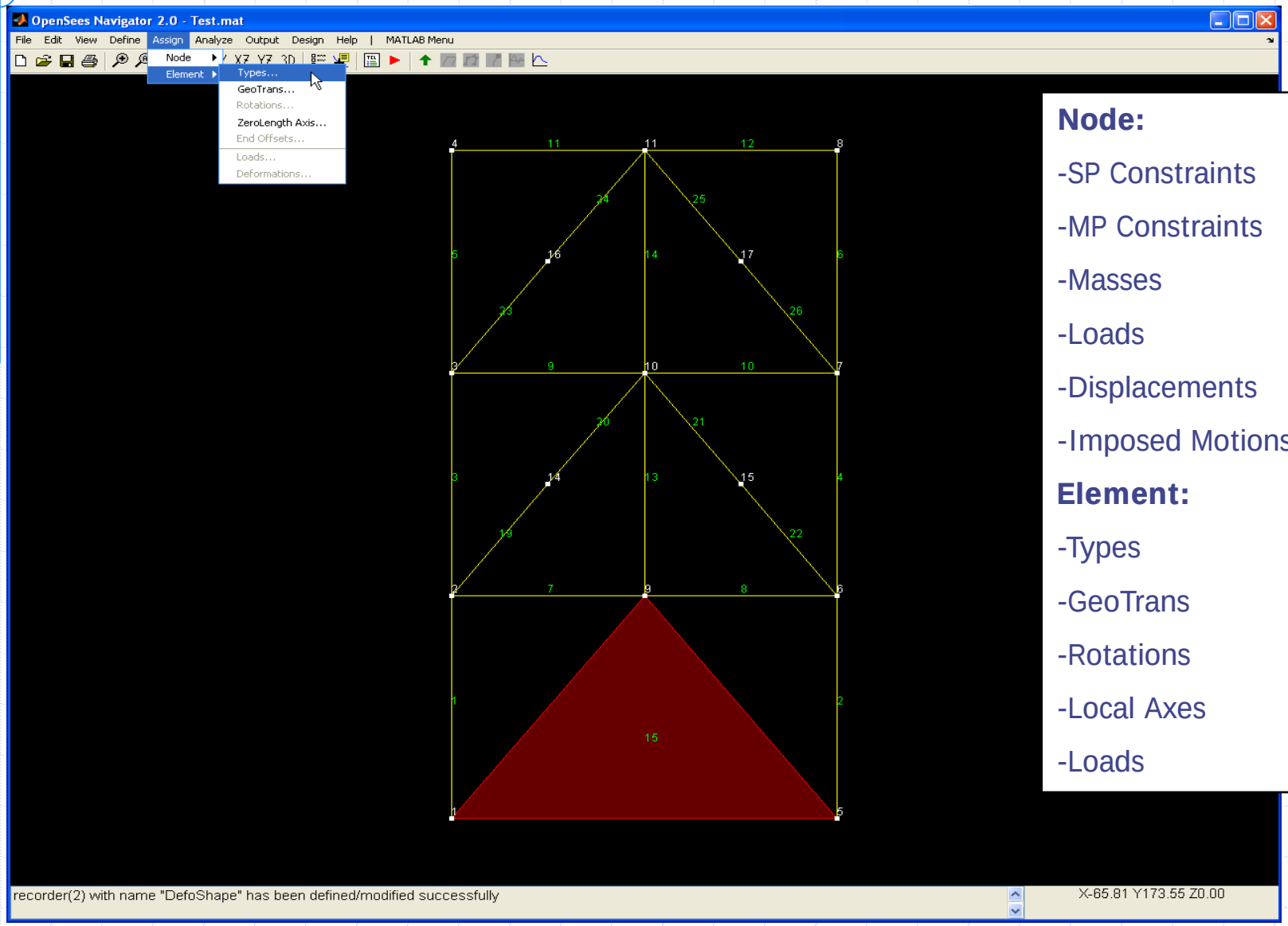
Integrator Type:

For example use AlphaOS Method for Hybrid Simulation

Solution Algorithm:

The AlphaOS Method requires a Linear solution algorithm

Assign menu



Node:

-SP Constraints

-MP Constraints

-Masses

-Loads

-Displacements

-Imposed Motions

Element:

-Types

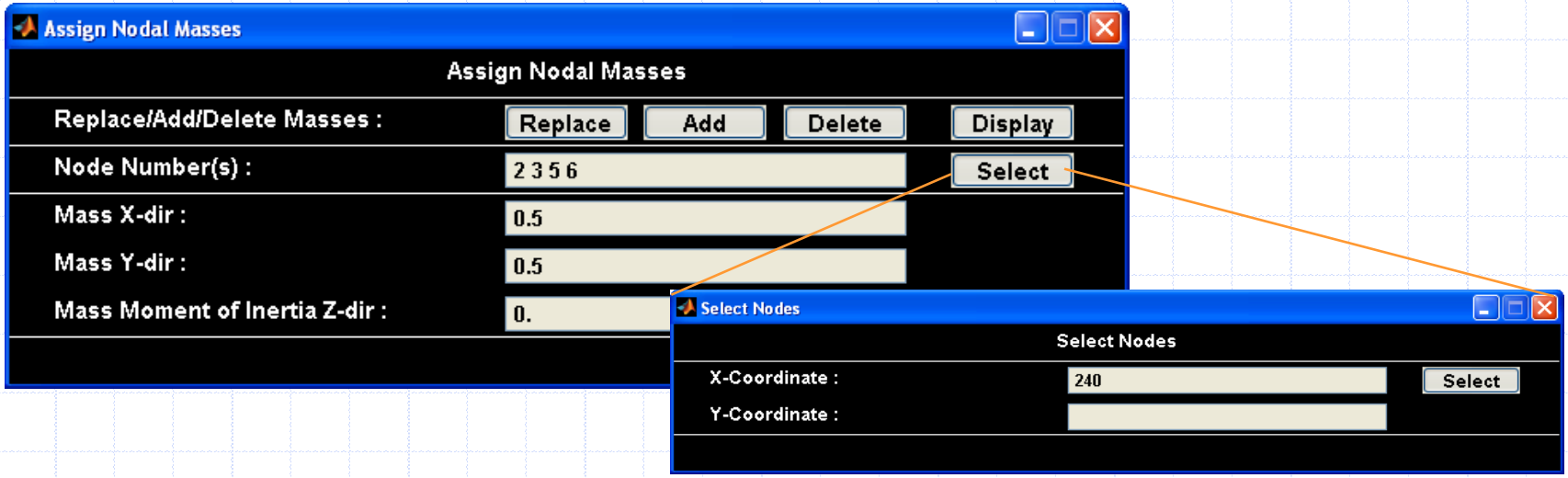
-GeoTrans

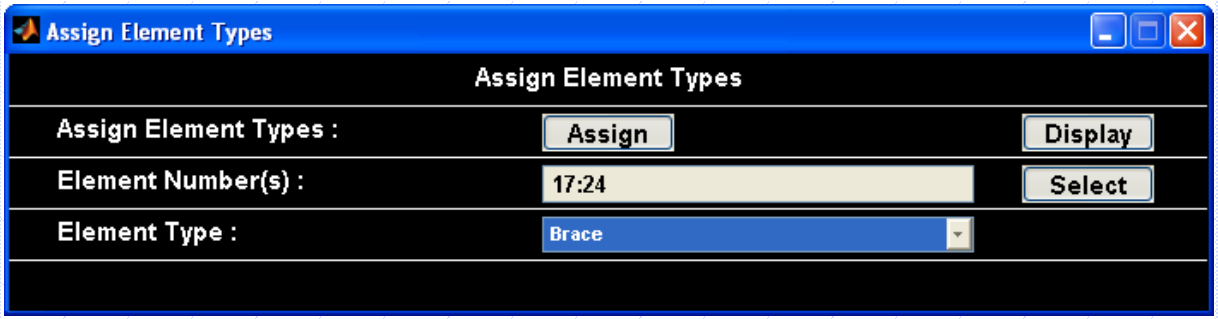
-Rotations

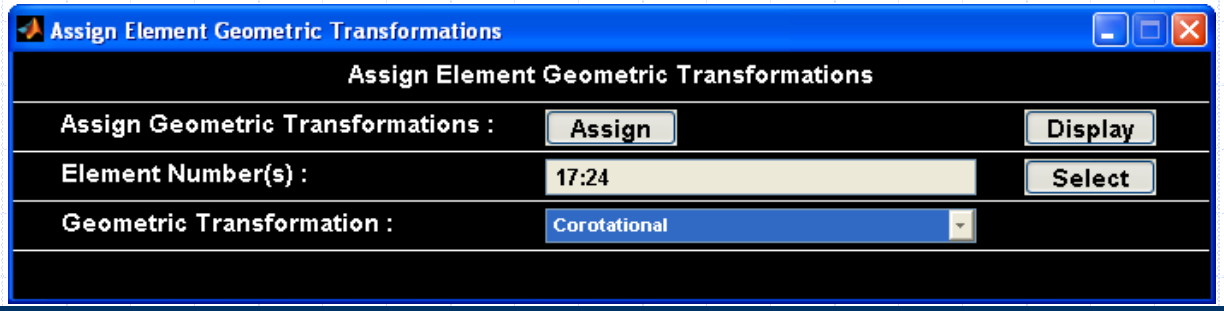
-Local Axes

-Loads

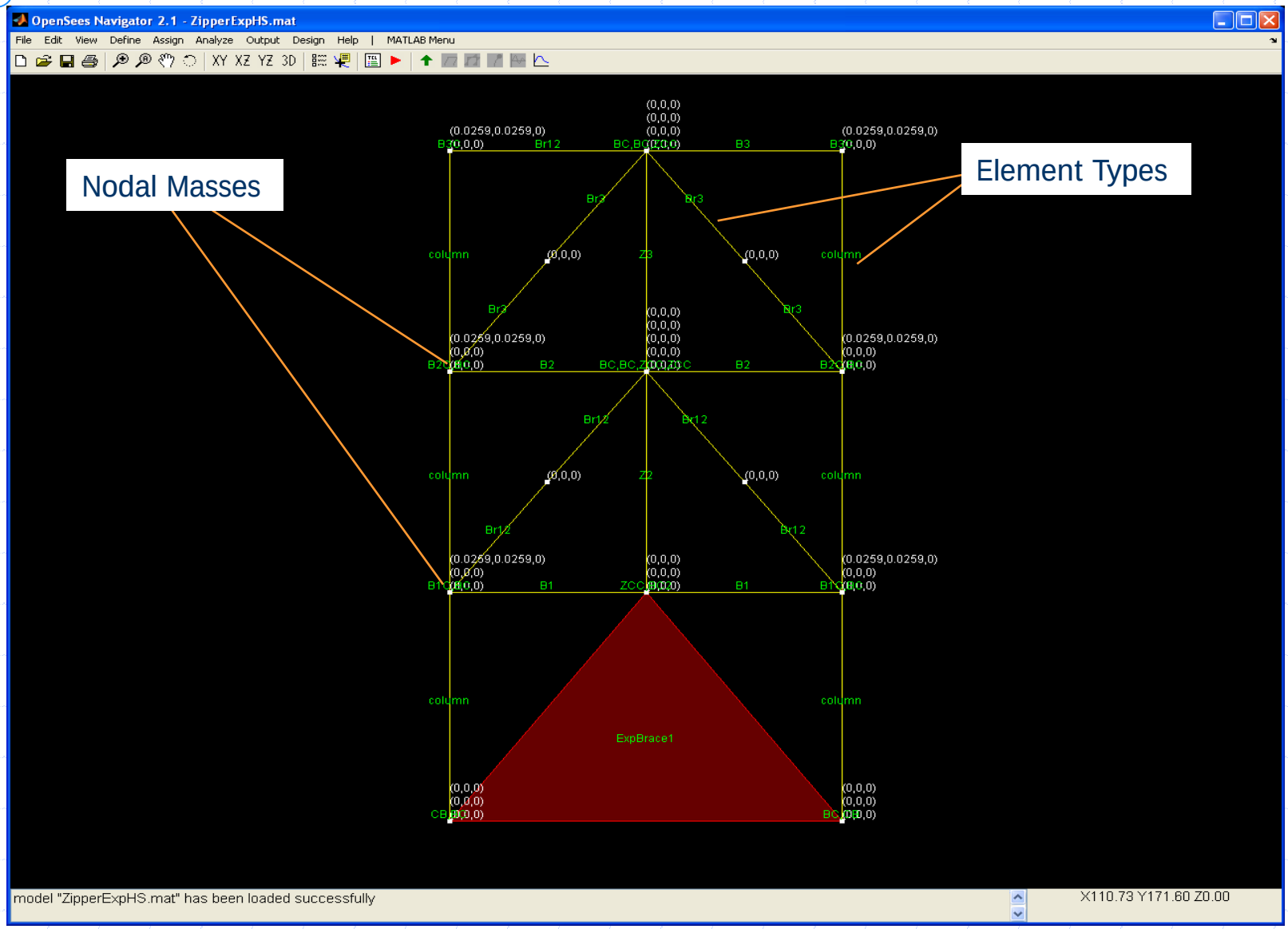
Assign menu


Assign Nodal Masses
Assign Nodal Masses
Replace/Add/Delete Masses :
Replace
Add
Delete
Display
Node Number(s) :
2 3 5 6
Select
Mass X-dir :
0.5
Mass Y-dir :
0.5
Mass Moment of Inertia Z-dir :
0.
Select Nodes
Select Nodes
X-Coordinate :
240
Select
Y-Coordinate :

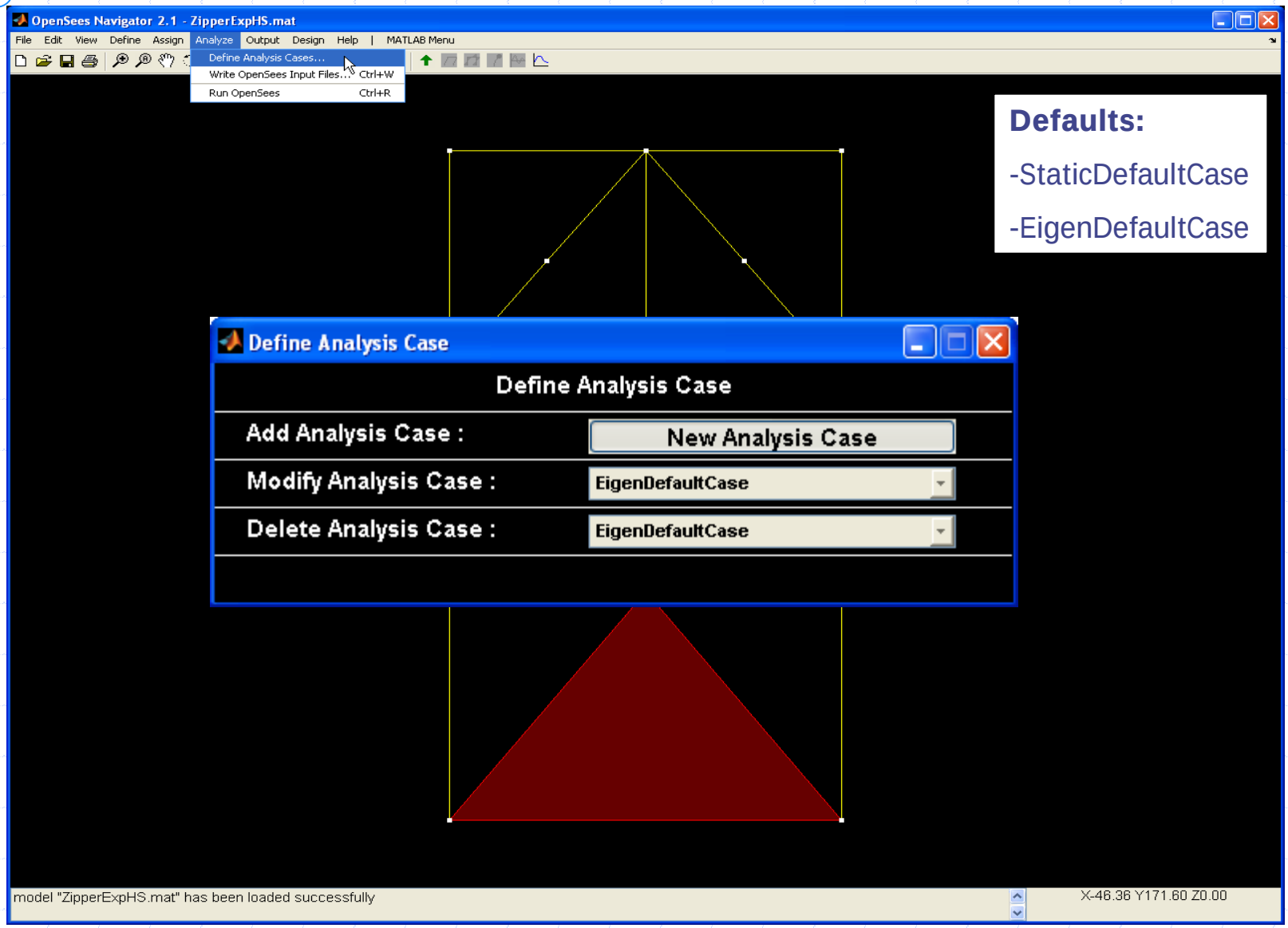

Assign Element Types
Assign Element Types
Assign Element Types :
Assign
Display
Element Number(s) :
17:24
Select
Element Type :
Brace


Assign Element Geometric Transformations
Assign Element Geometric Transformations
Assign Geometric Transformations :
Assign
Display
Element Number(s) :
17:24
Select
Geometric Transformation :
Corotational

Display assigned properties



Define analysis case

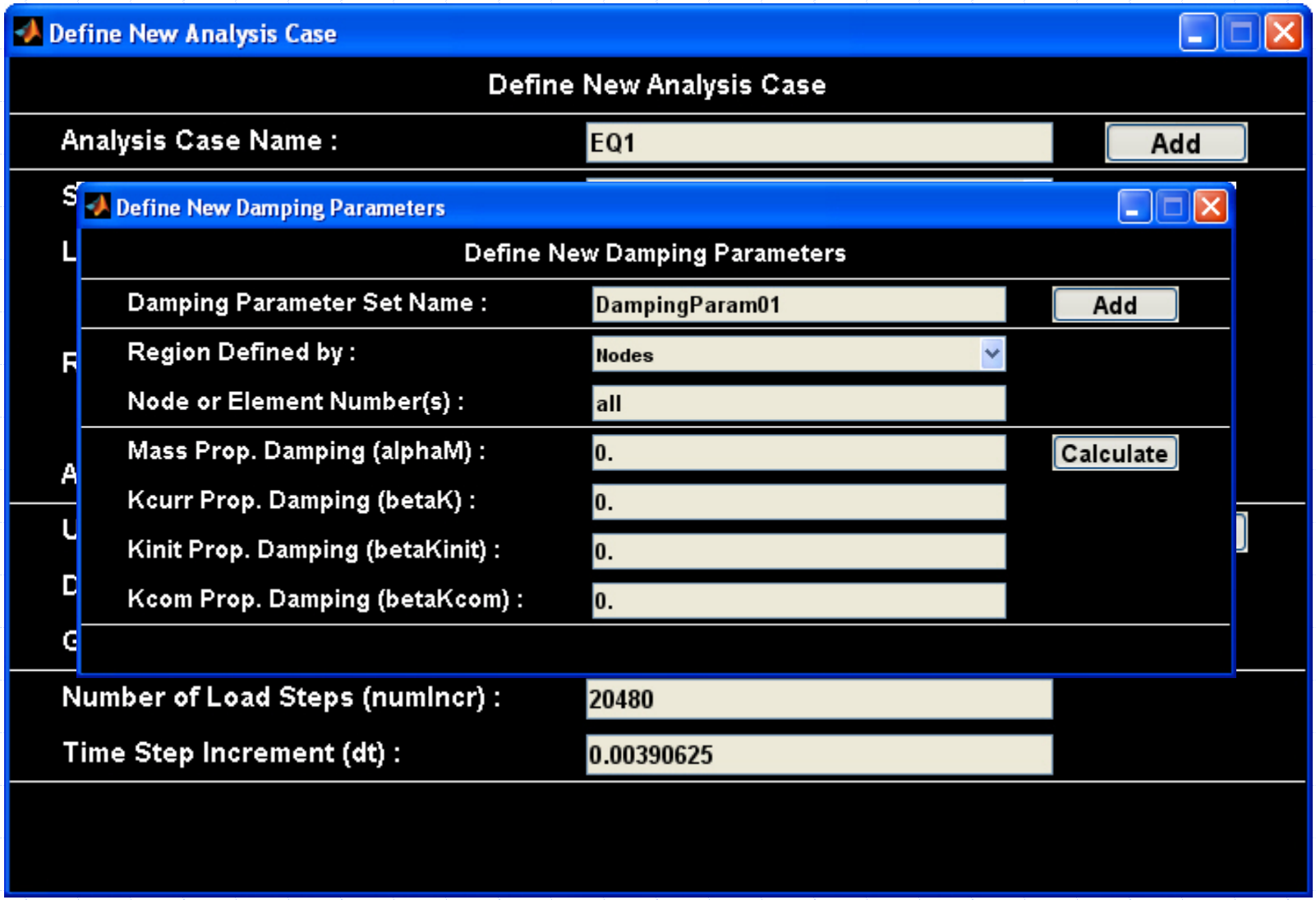


Defaults:

-StaticDefaultCase

-EigenDefaultCase

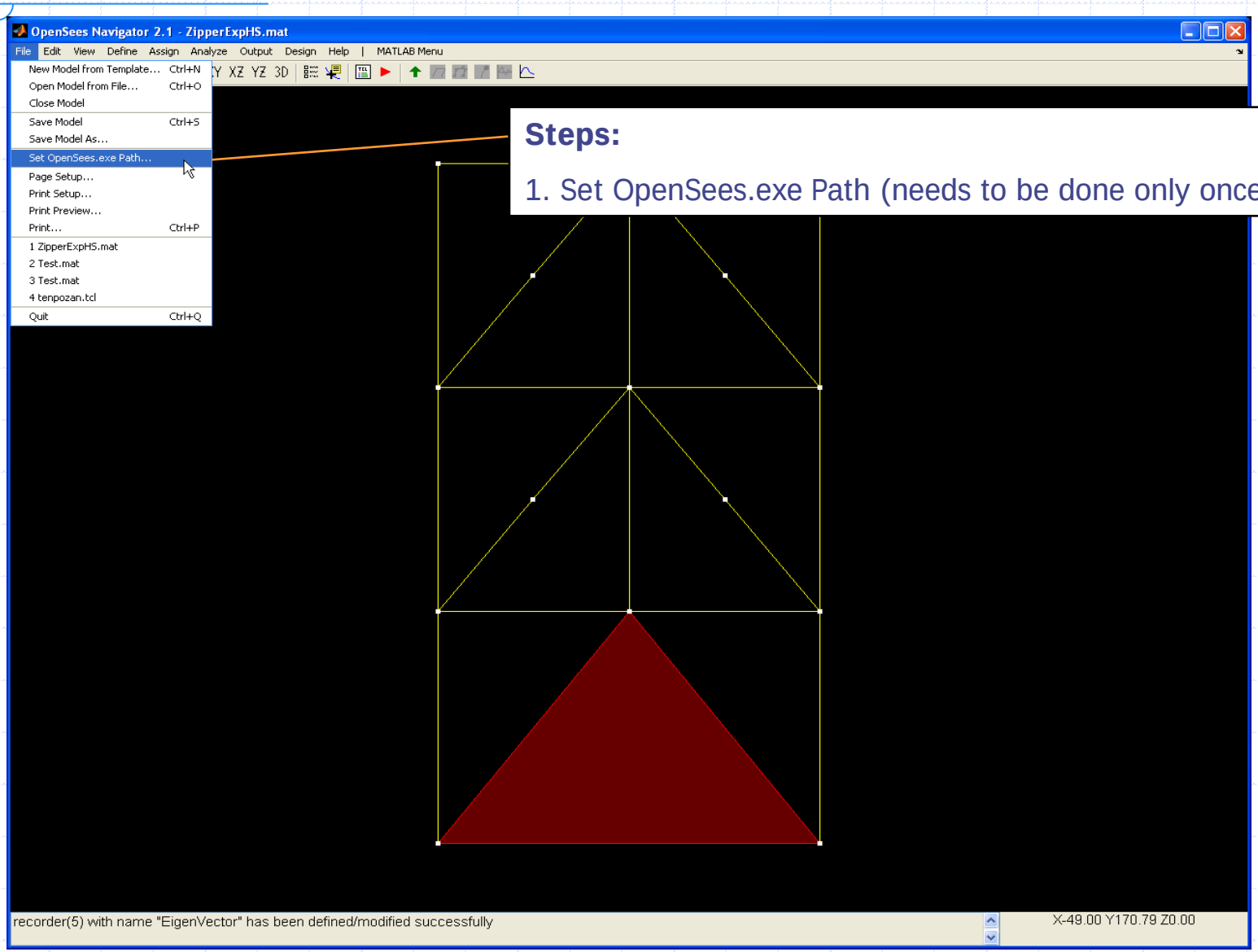
Define analysis case: new analysis case



User Defined Analysis Script

```
# set the test parameters
set testType NormDispIncr
set testTol 1.0e-8;
set testIter 25;
test  $testType $testTol $testIter

# set the algorithm parameters
set algoType KrylovNewton
algorithm  $algoType

set ok 0;
set tFinal [expr $numSteps * $dt]
set tCurrent [getTime]

while {$ok == 0 && $tCurrent < $tFinal} {
   if {fmod($tCurrent,1) < 1.0E-16} {
          puts "$i $tCurrent"
        }
        set ok [analyze 1 $dt]

   if {$ok != 0} {
      puts " "
      puts [format "KrylovNewton failed (time = %1.3e), try Newton" $tCurrent]
               algorithm Newton
      test $testType $testTol $testIter 0
      set ok [analyze 1 $dt]
      algorithm $algoType
   }

   if {$ok != 0} {
      puts " "
               puts [format "Newton failed (time = %1.3e), try Newton  w/ iniCurrent" $tCurrent]
               algorithm Newton -initialCurrent
      test $testType $testTol $testIter 0
      set ok [analyze 1 $dt]
      algorithm $algoType
   }

     if {$ok != 0} {
      puts " "
               puts [format "Newton  w/ iniCurrent failed (time = %1.3e), try Newton  w/ ini" $tCurrent]
               algorithm Newton -initial
      test $testType $testTol [expr 500 * $testIter] 0
      set ok [analyze 1 $dt]
      algorithm $algoType
      test $testType $testTol $testIter 0
```

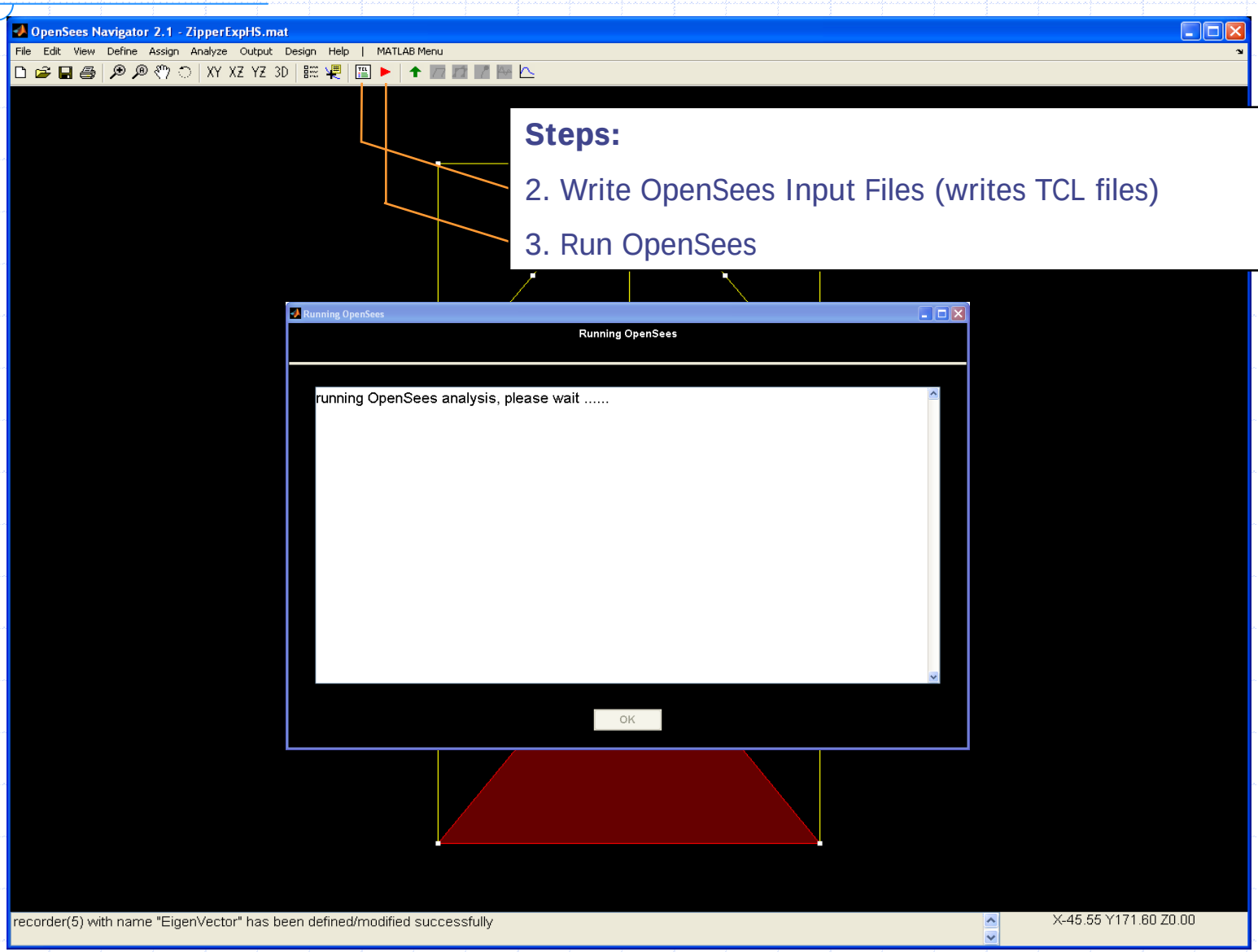
Run OpenSees: set OpenSees.exe path



Steps:

1. Set OpenSees.exe Path (needs to be done only once)

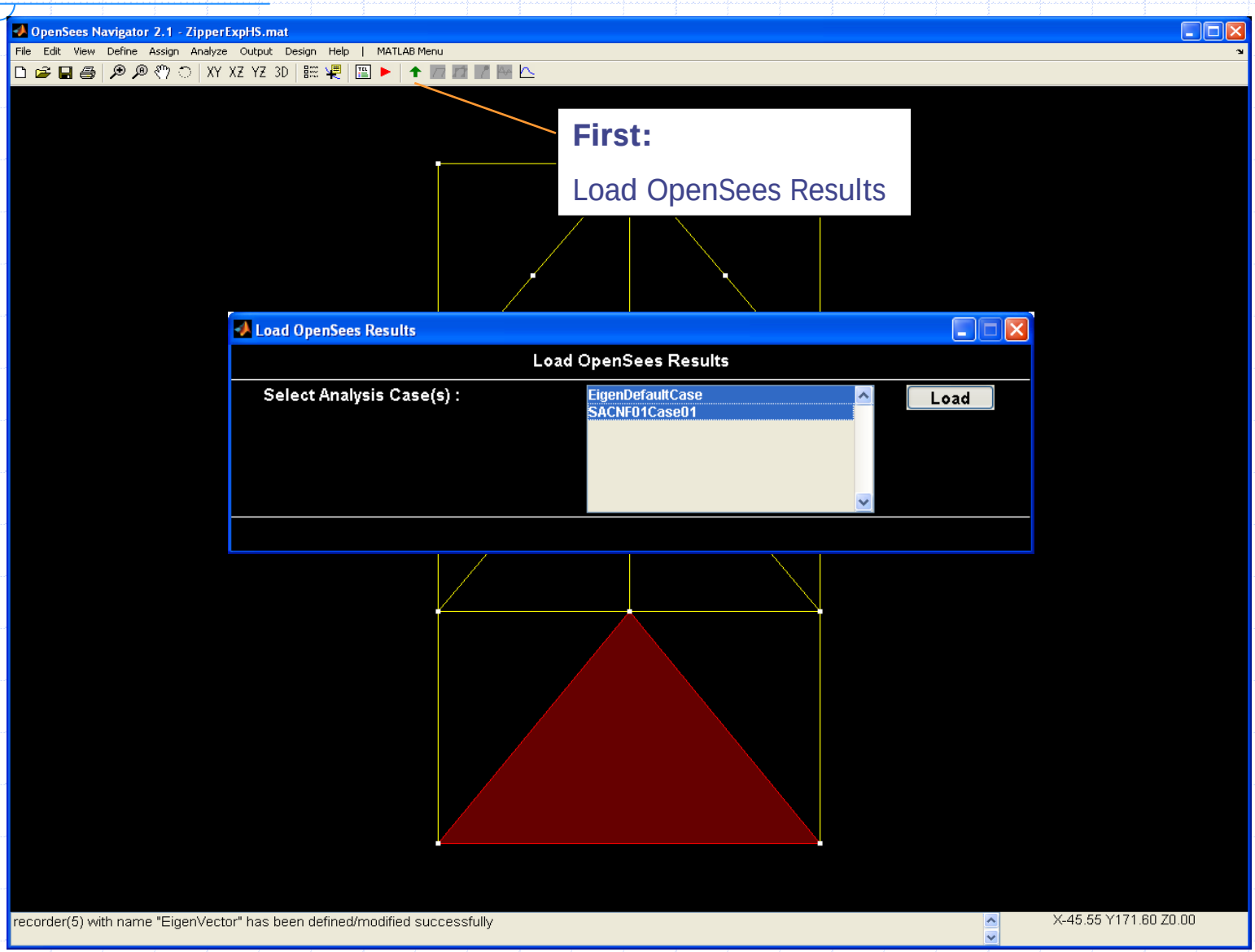
Run OpenSees: write TCL files



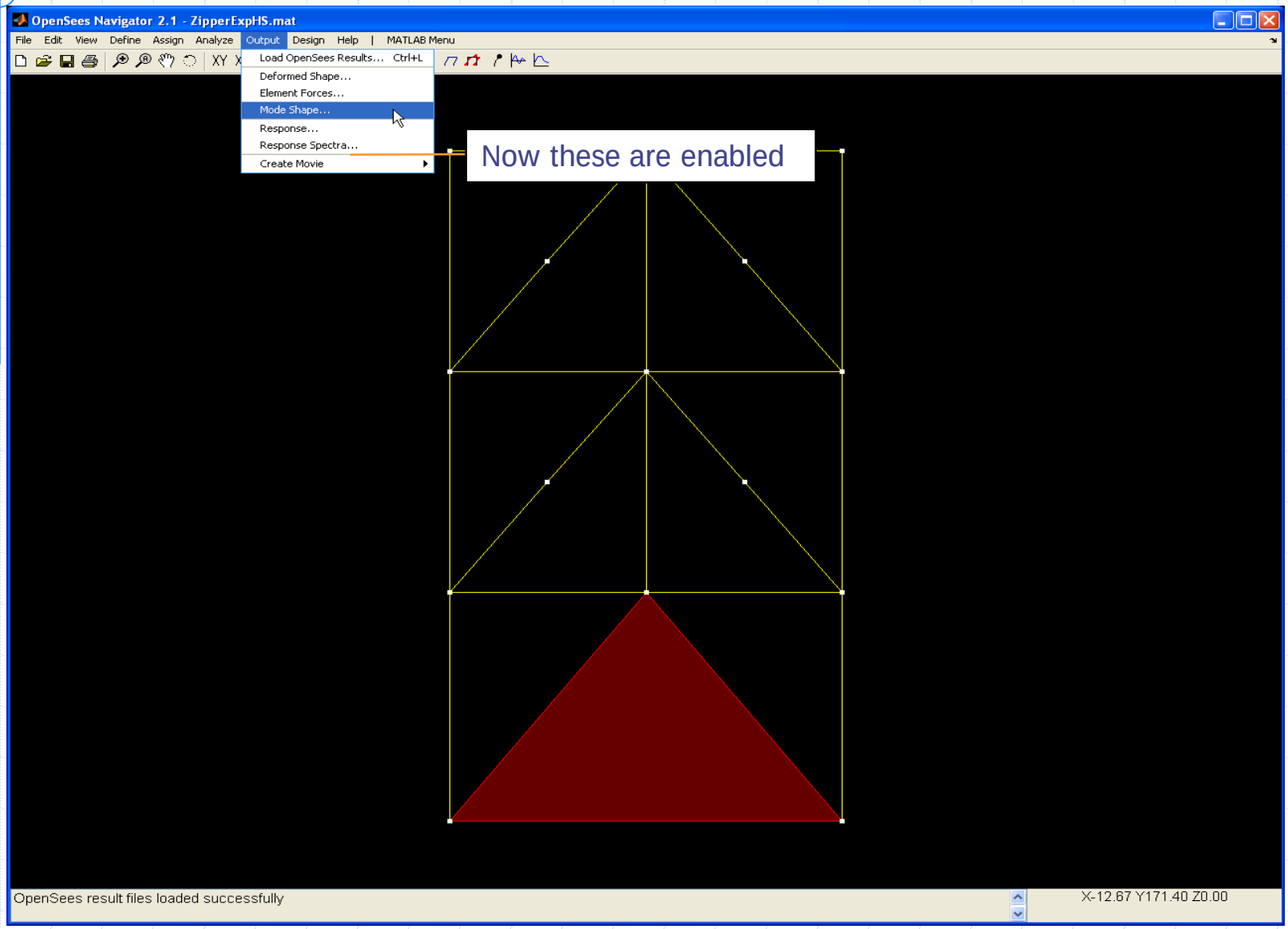
Steps:

2. Write OpenSees Input Files (writes TCL files)
3. Run OpenSees

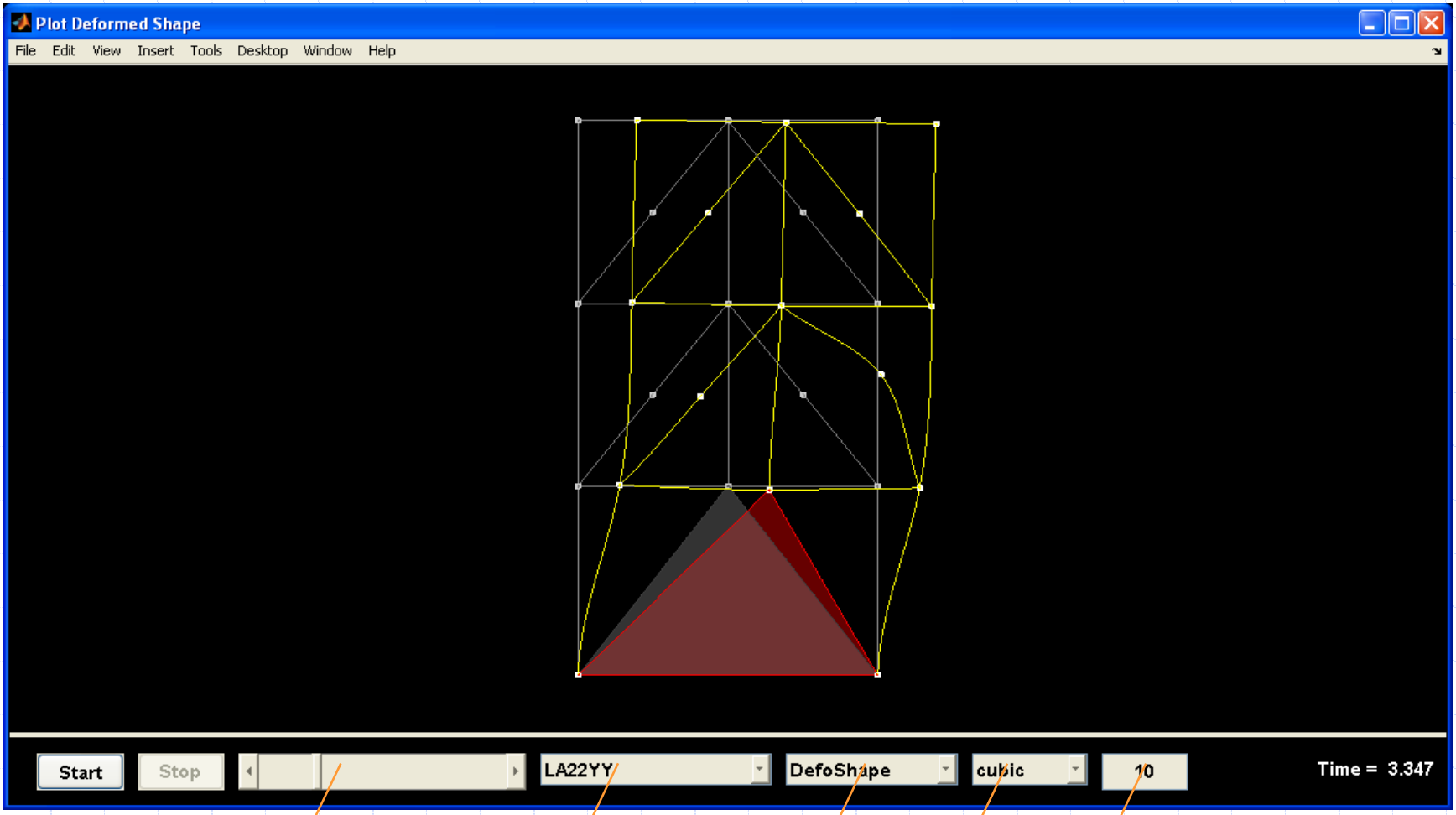
Post processing: load results



Post processing: output

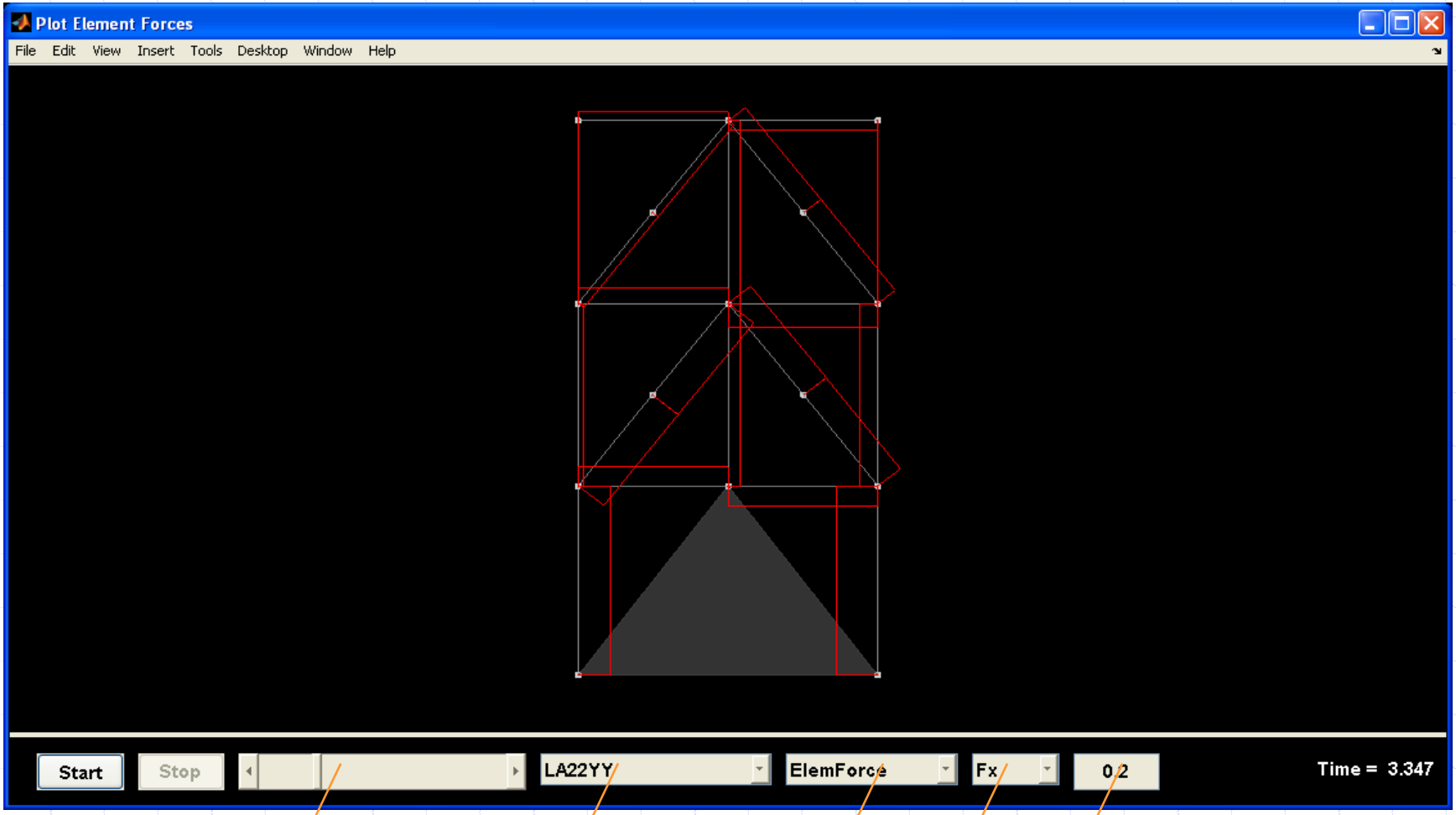


Post processing: plot deformed shape



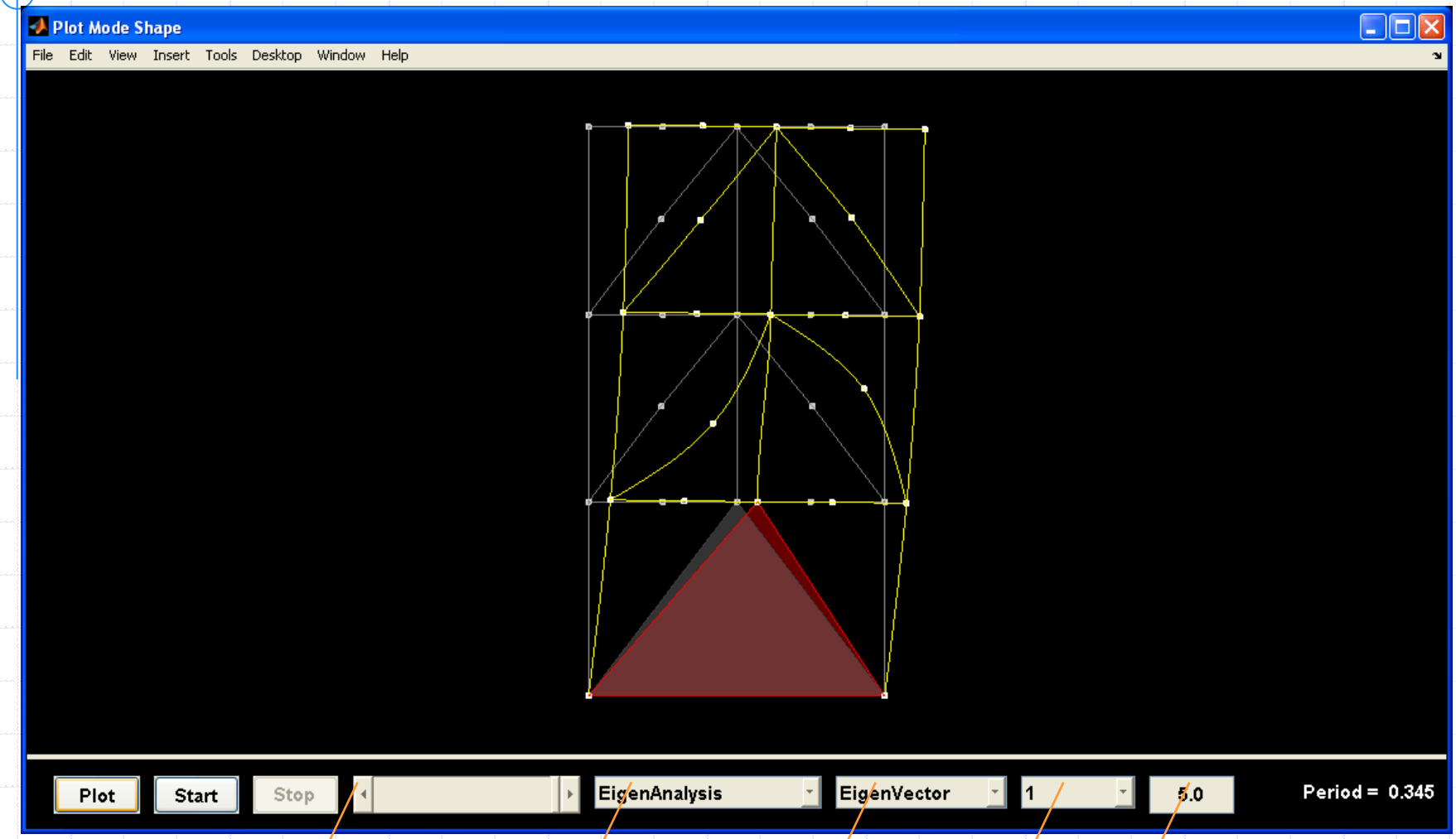
Time Step AnalysisCase Recorder Order Magnification

Post processing: plot element forces



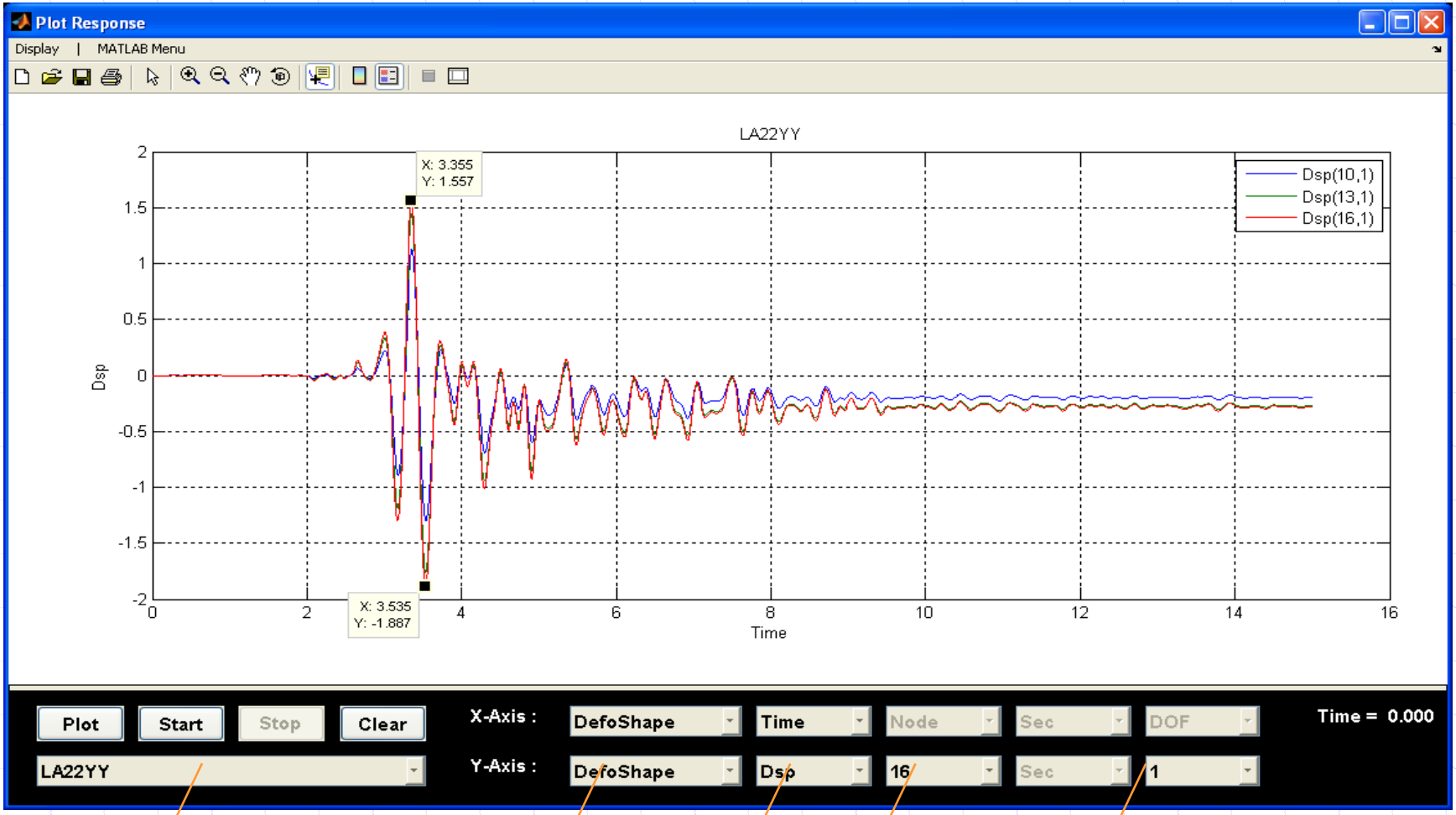
Time Step AnalysisCase Recorder Response Magnification

Post processing: plot mode shape



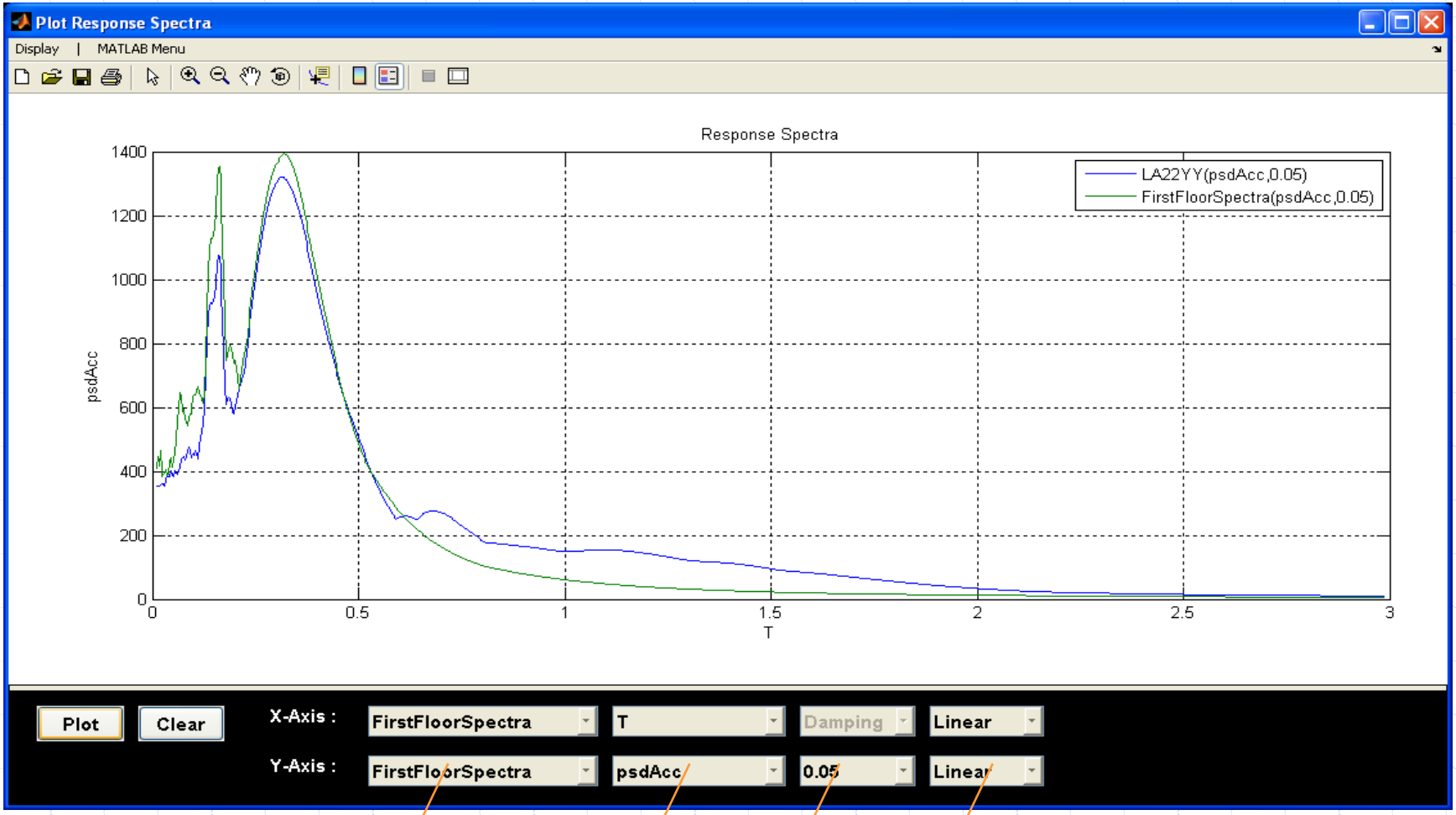
Animation Speed AnalysisCase Recorder Mode Magnification

Post processing: plot response histories



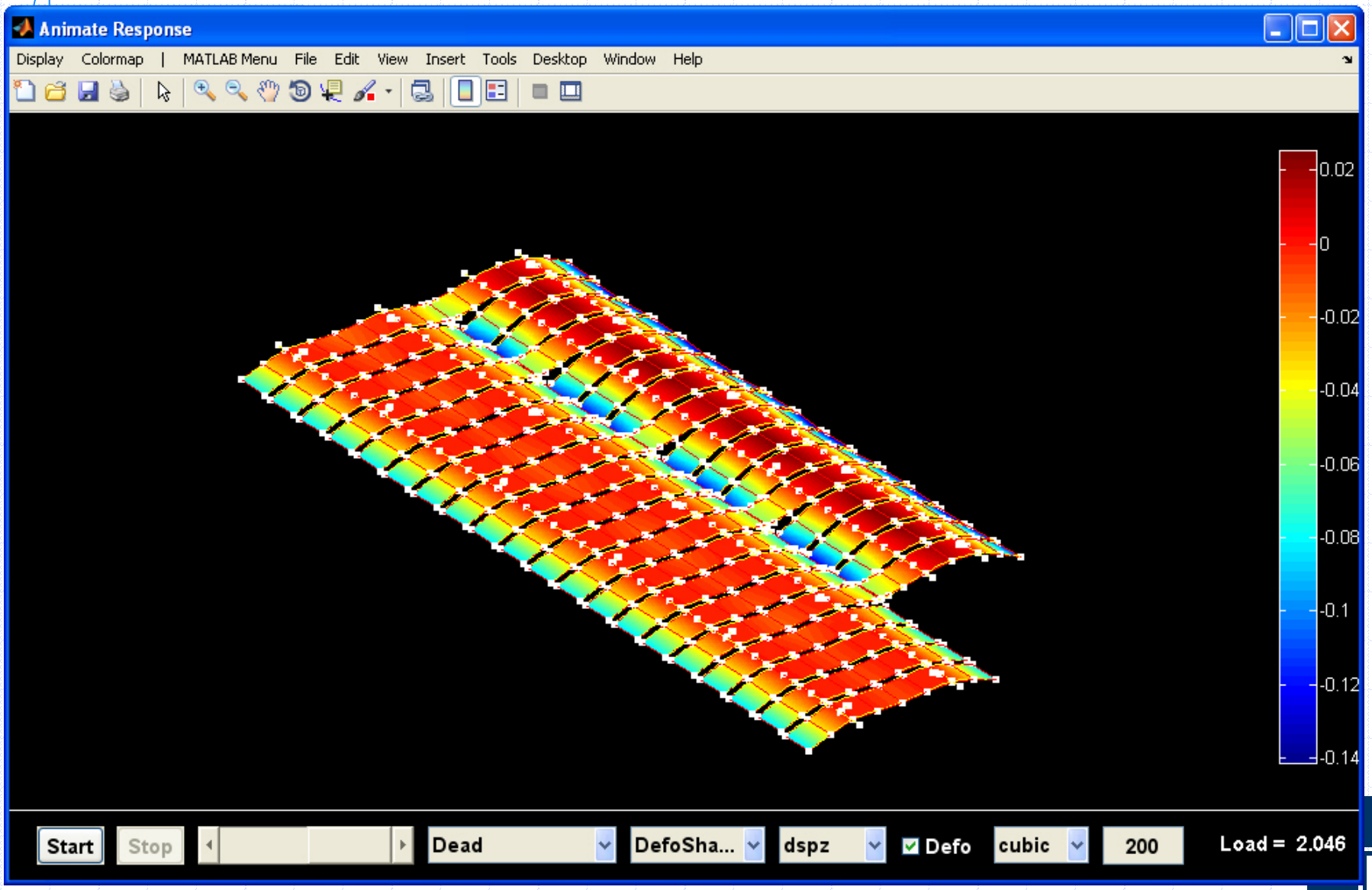
AnalysisCase Recorder Parameter Node/Element DOF

Post processing: plot response spectra

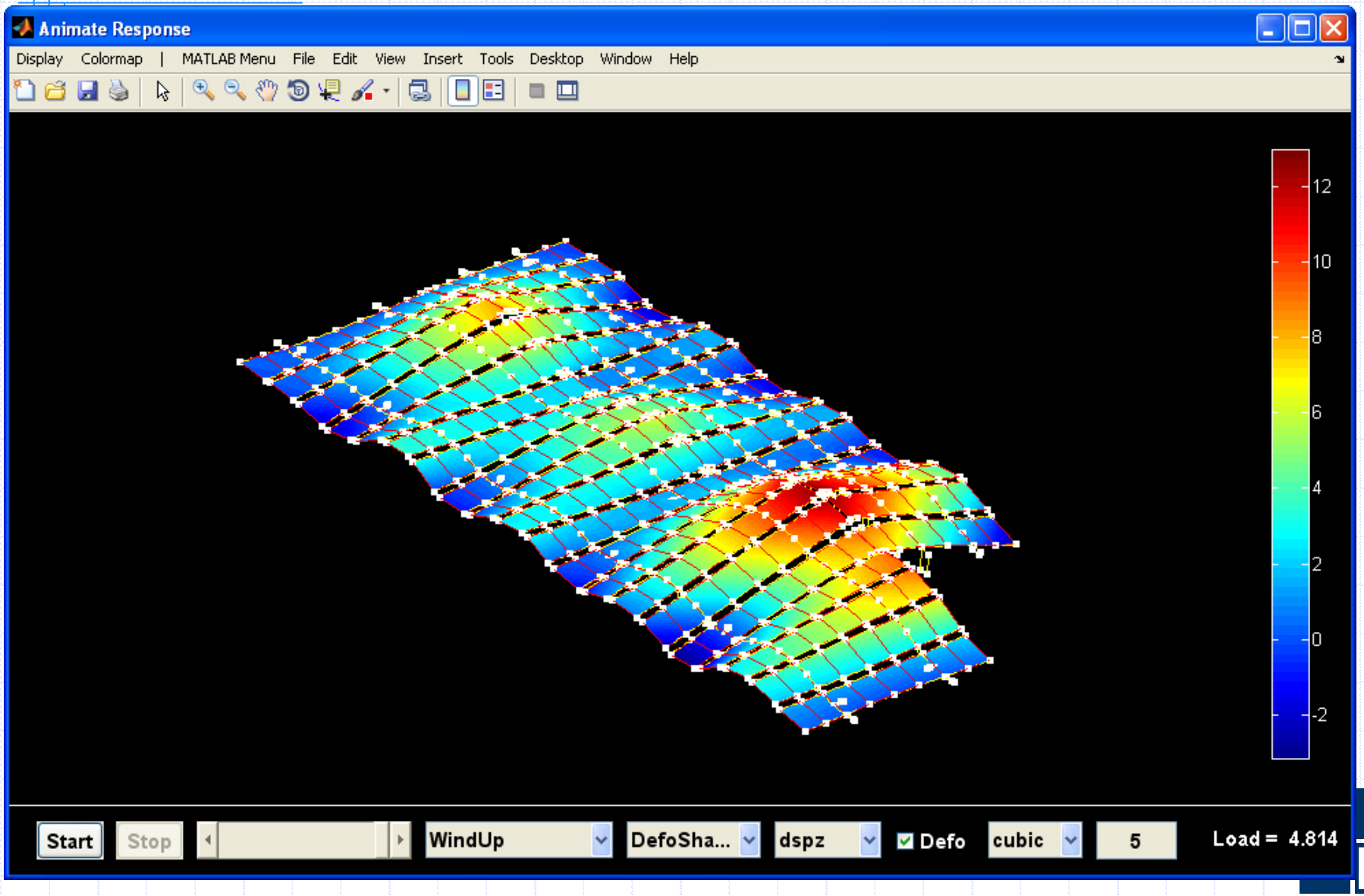


Spectra Name Response Damping Axis Scale

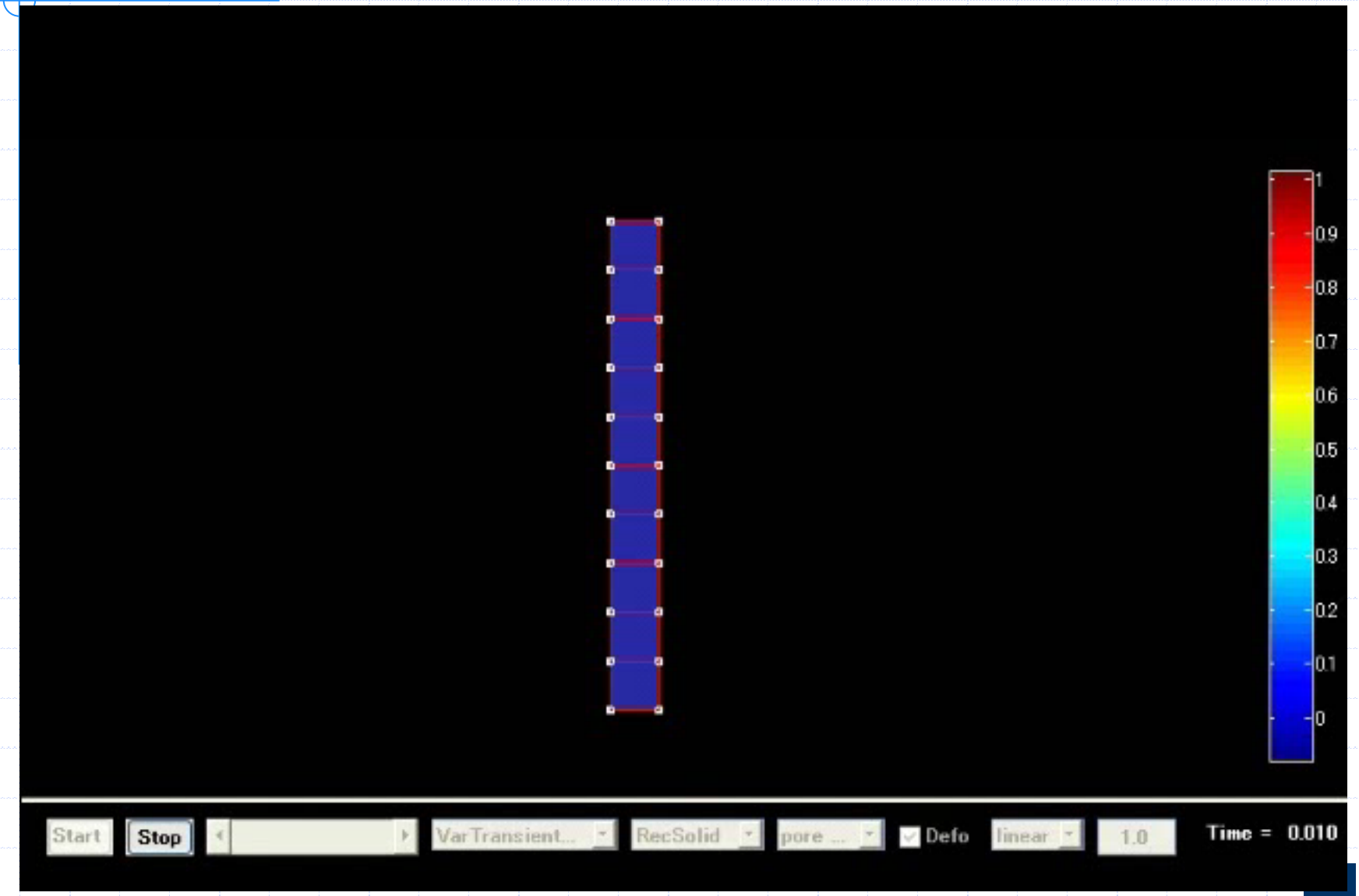
Post processing: animate response



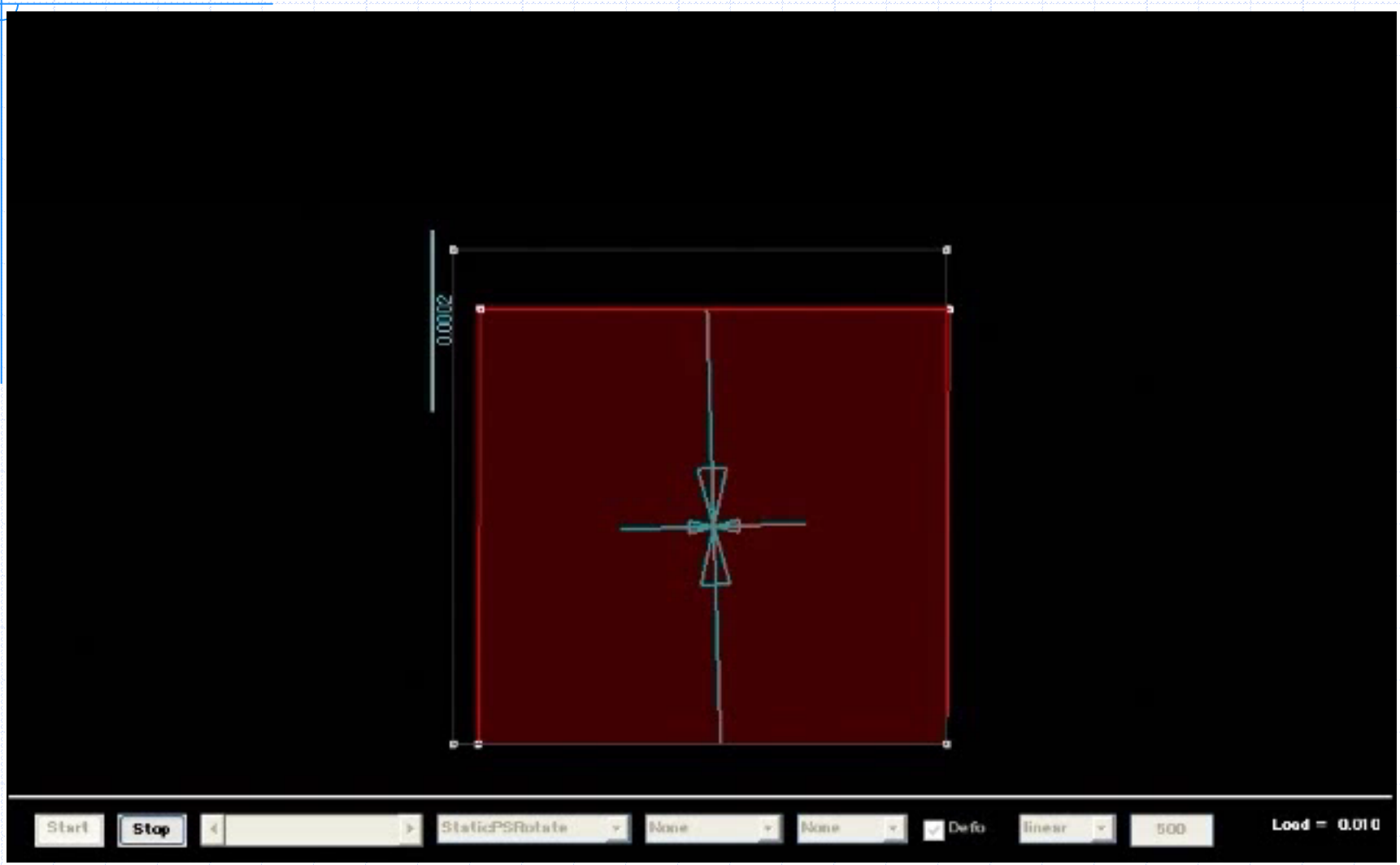
Post processing: animate response



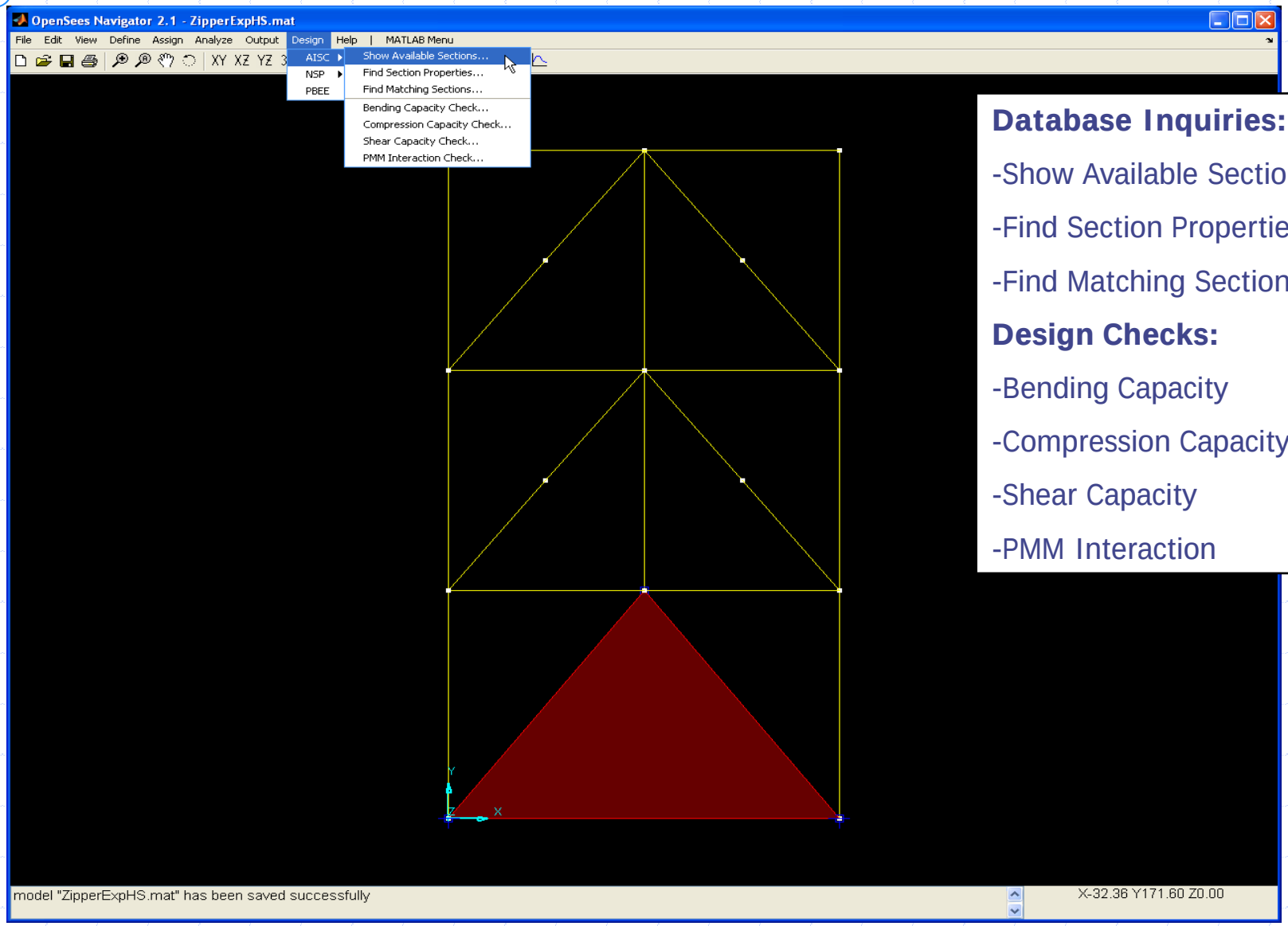
Post processing: animate response



Post processing: principal σ and ε



Design: AISC design toolbox



Database Inquiries:

-Show Available Sections

-Find Section Properties

-Find Matching Sections

Design Checks:

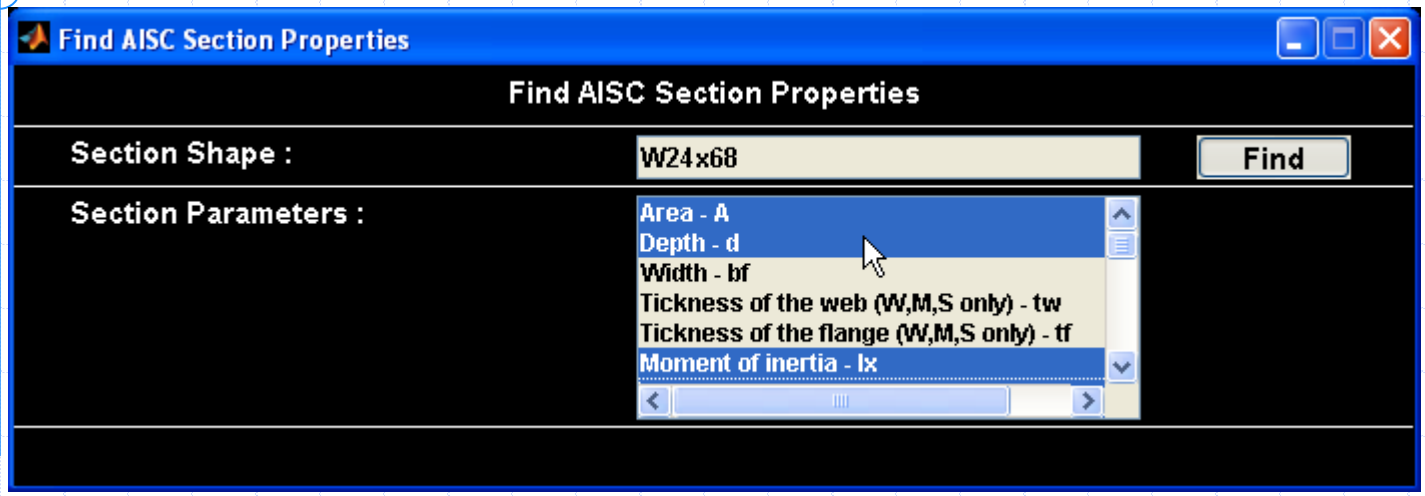
-Bending Capacity

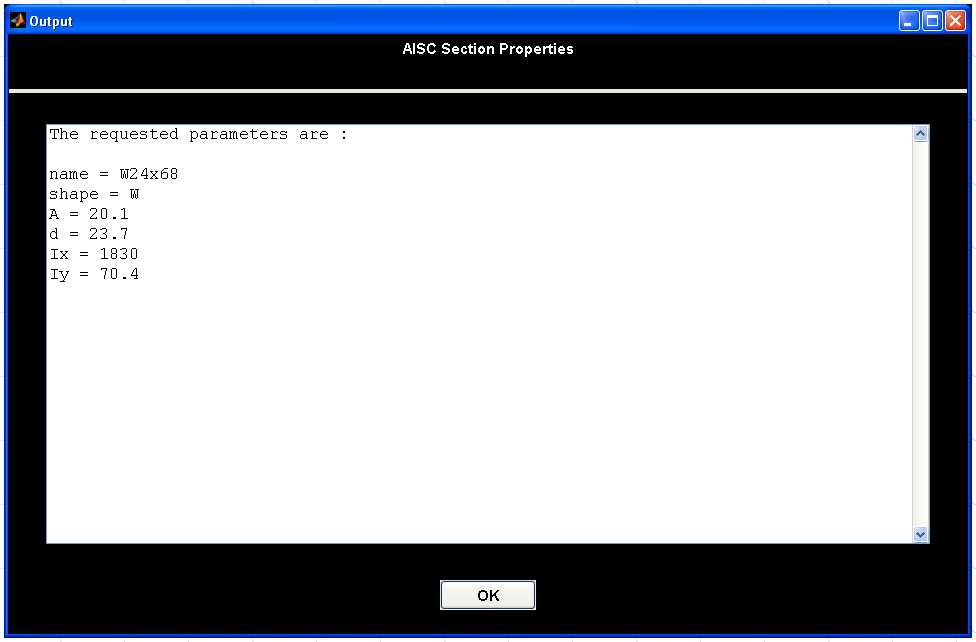
-Compression Capacity

-Shear Capacity

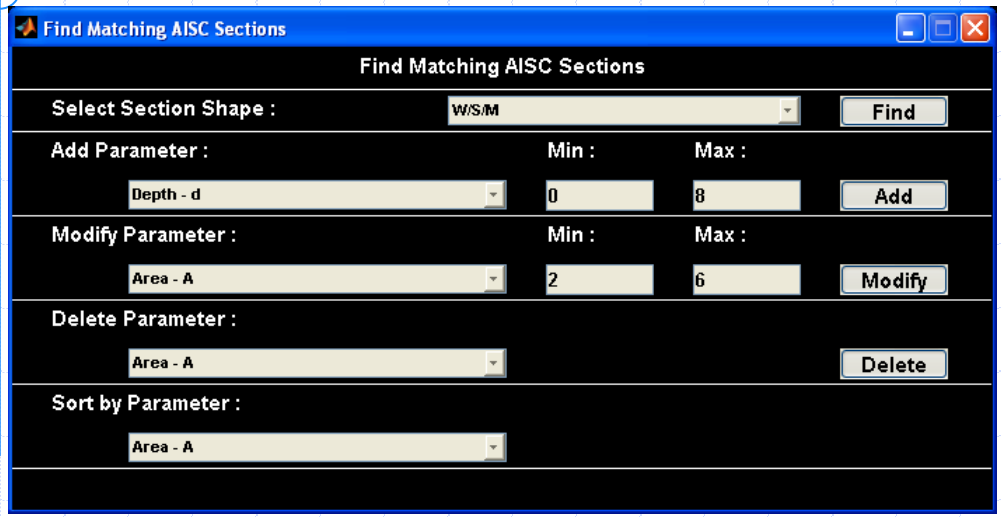
-PMM Interaction

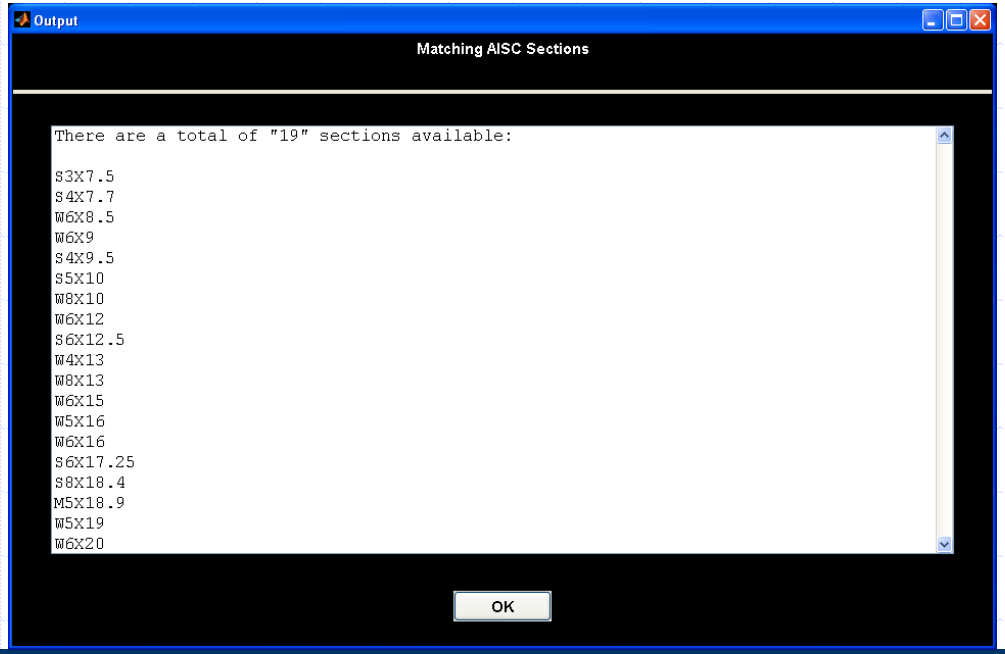
AISC design toolbox: section properties



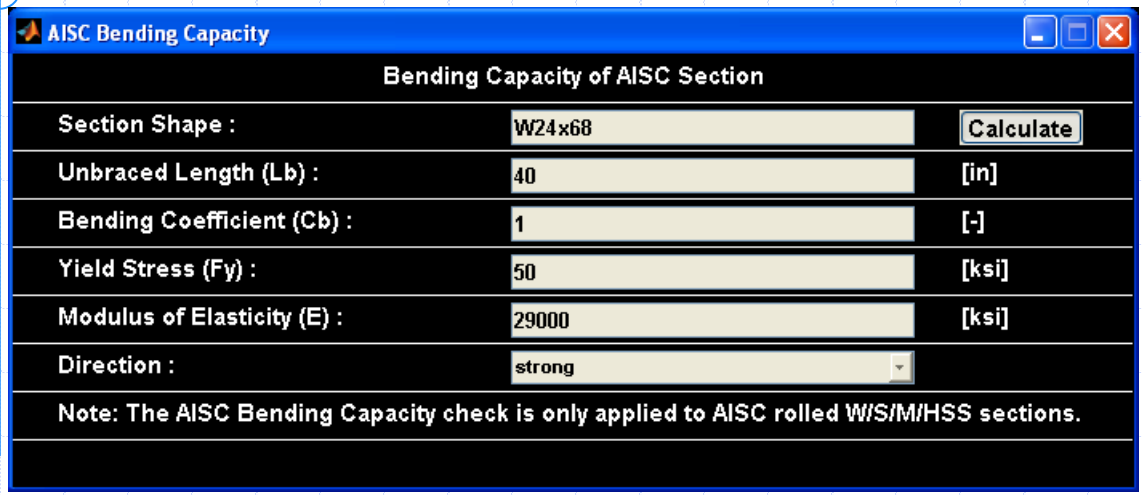


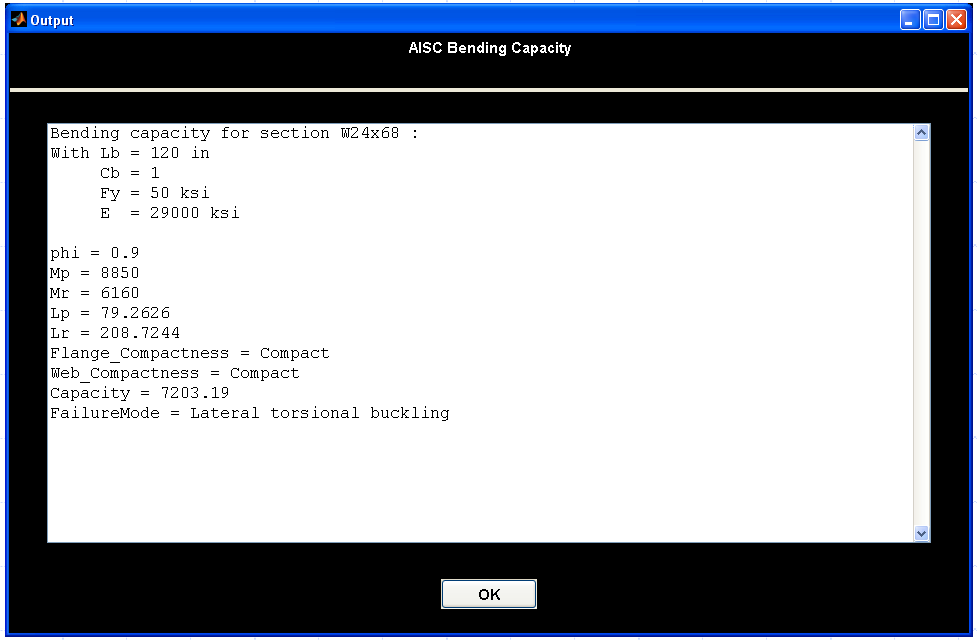
AISC design toolbox: matching sections



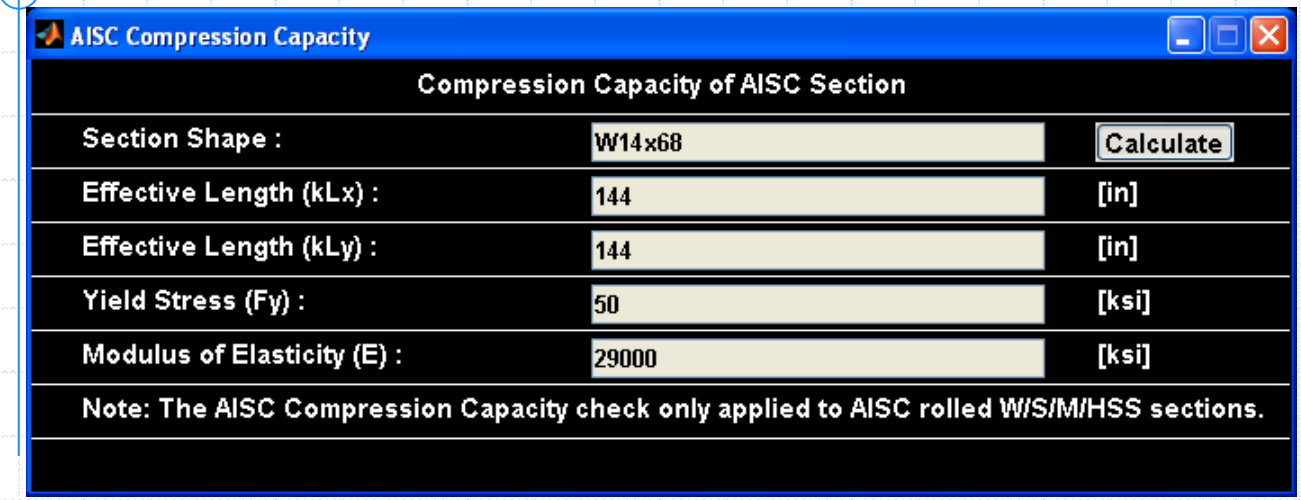


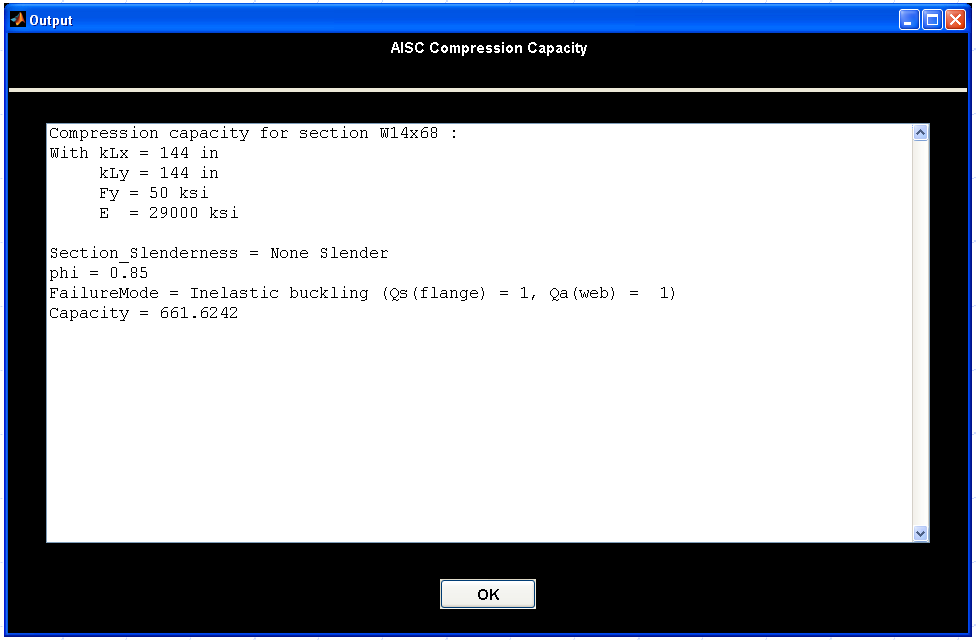
AISC design toolbox: bending capacity



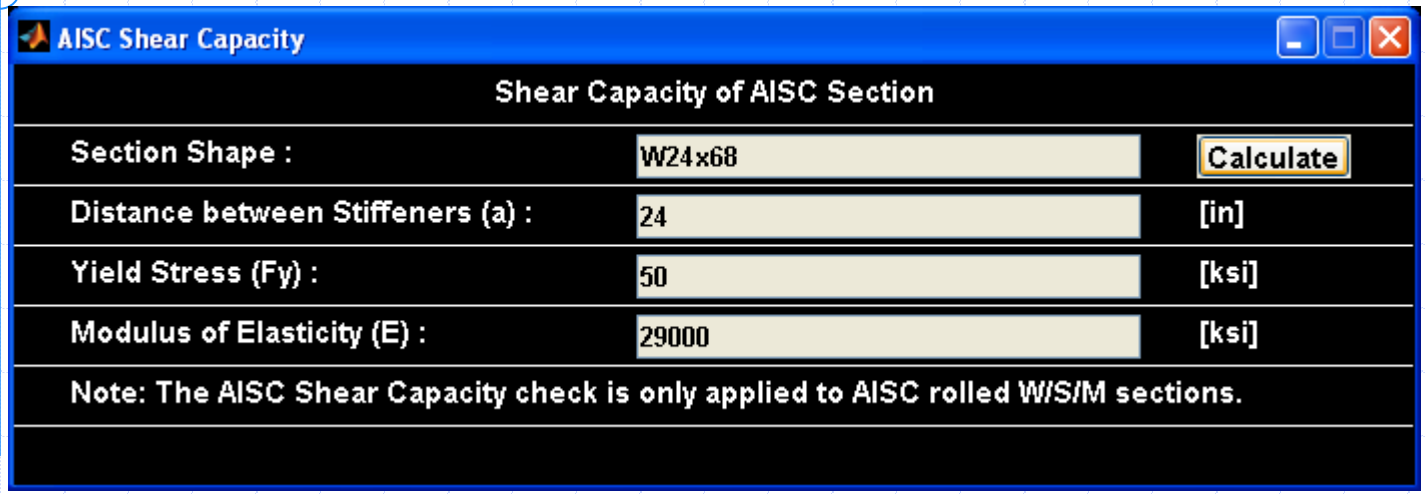


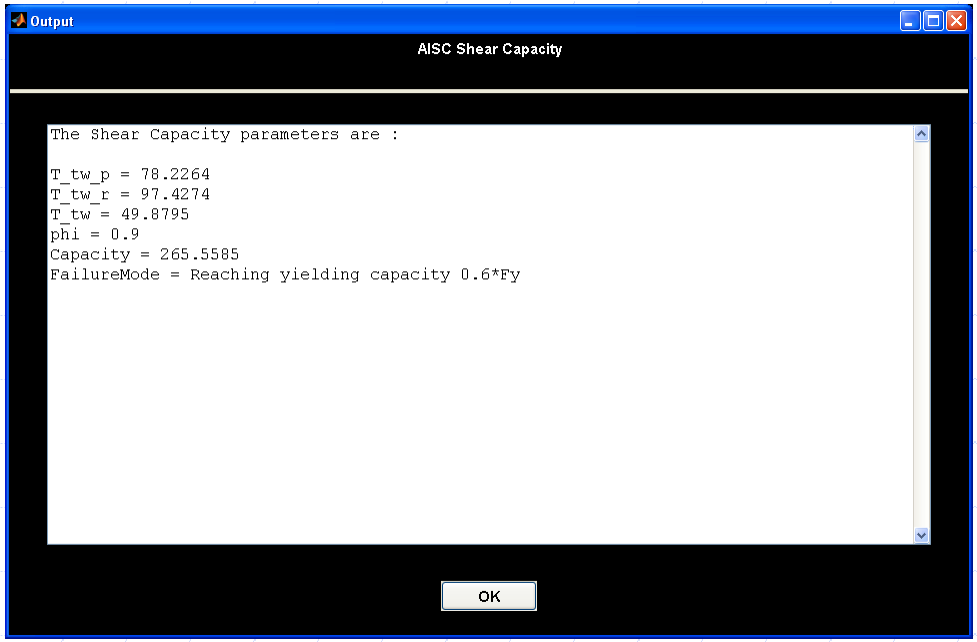
AISC design toolbox: compression cap.



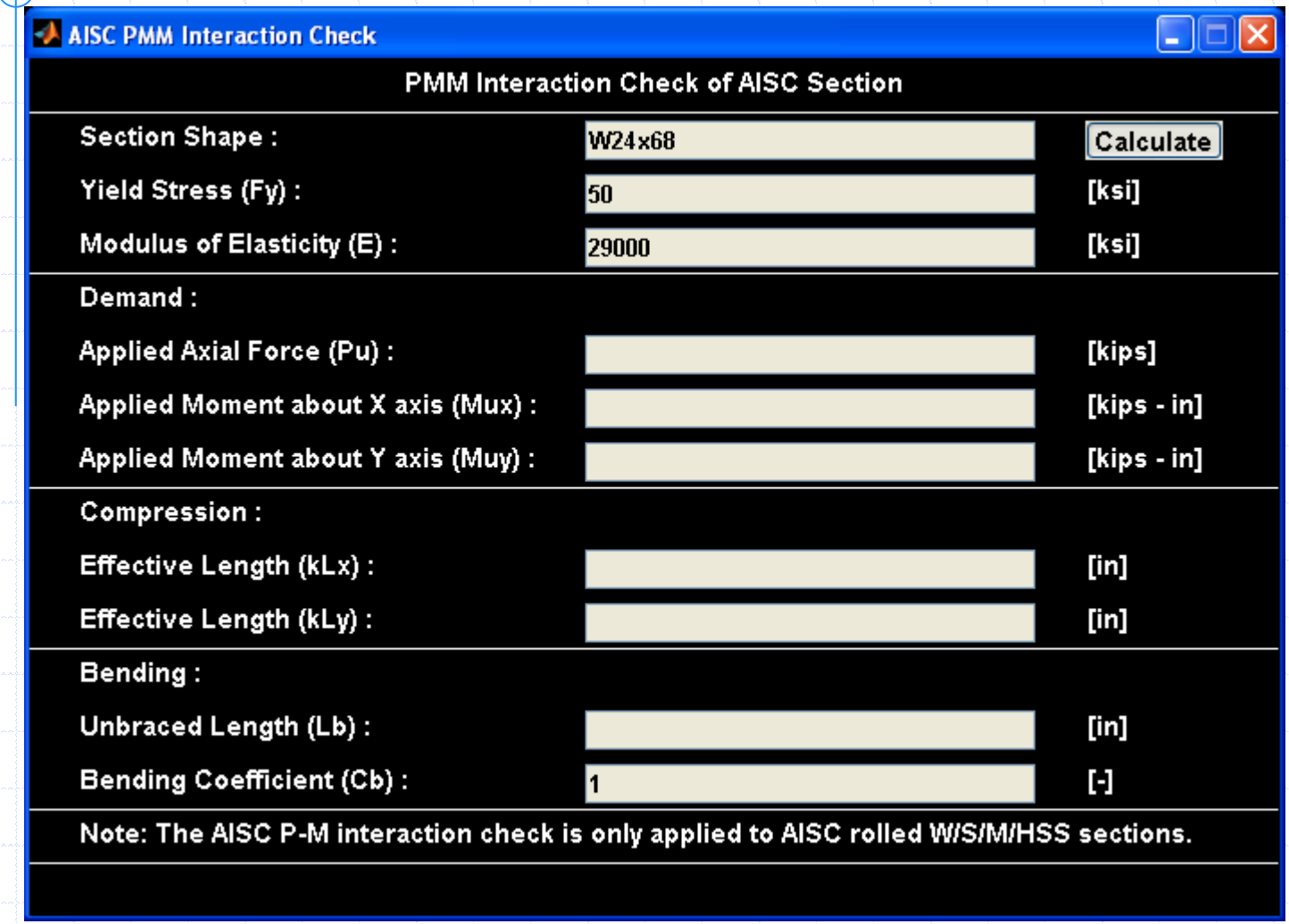


AISC design toolbox: shear capacity





AISC design toolbox: PMM interaction



Summary

- OpenSees Navigator provides
  - Flexible and user friendly graphical user interface.
  - Great tool to visualize structural behavior.
  - Easy way to study material, section, element or system behavior.
- Hybrid simulation interface (OpenFresco).
- Many design toolboxes: NSP, PBEE, AISC design checks, AISC database, response spectra for linear and bilinear systems and signal filtering.
- Both MATLAB Pcode and self-executable versions are available.

Website: home

OpenSees Navigator

Home | Introduction | Manuals | Tutorials | Presentations | Discussion | Updates | Downloads | Links

Home> Search

- Introduction
- Manuals
- Tutorials
- Presentations
- Discussion
- Updates
- Downloads
- Links

Dear OpenSees Navigator users,

Thanks for your interest in OpenSees Navigator. This program is intended to be self-explanatory, nevertheless a basic user manual will be added to the website shortly. We are very happy to have the opportunity to distribute this software for OpenSees Navigator users. We encourage everyone to try out all of the functions of the program and send us criticism, corrections or suggestions to improve future versions. We also encourage users to e-mail us at either andreas.schellenberg@gmail.com or yangtony2004@gmail.com so that we can add the e-mail addresses to the OpenSees Navigator user list. We will use such list to contact everyone about new releases or major updates. We will try our best to improve the next release.

Thank you.

Please feel free to visit our websites to discover in what other fun research we are involved:

Andreas Schellenberg & Tony Yang

Hit Counter

Website: download

OpenSees Navigator

Home | Introduction | Manuals | Tutorials | Presentations | Discussion | Updates | Downloads | Links

Home> Downloads> Stand-Alone Windows>

Search

- MCRInstaller.exe
- OpenSeesNavigator.zip

Installation Instructions:

1. Download the two files on the left.
2. Install the Matlab component runtime libraries by executing MCRInstaller.exe and following the on screen instructions (this has only to be done once).
3. Extract OpenSeesNavigator.zip in any folder of your choice and then execute OpenSeesNavigator.exe.
4. If you like you can create a shortcut to OpenSeesNavigator.exe on your Desktop.

Hit Counter