<u>Presenters</u>

Rod Holland – SunLink

Kate Miller – SunLink

Dr. Andreas Schellenberg – R+C

Approach to Parametric Analysis of Wind Load Effects on PV Solar Roof Bearing Arrays Using OpenSees

OpenSees Days August 16, 2012

SunLink R & D

Long relationship with University of Western Ontario

- Boundary layer wind tunnel
- Modeling, testing, analysis

Alan G. Davenport Wind Engineering Group

Long relationship with R+C

- Analytical methods and process
- Seismic testing planning, execution, and analysis
- Joint authorship of papers

California Solar Initiative Grant

- 2 year effort 2011-2012
- Matching funds
- Focus on automation, analysis, process integration, seismic testing

California Solar Initiative

CSI R&D Goals

- The overall goal of the RD&D program is to help build a sustainable and self-supporting industry for customer-sited solar in California. The research funding should measurably support these basic goals:
- Reduce the current retail solar price to levels that are comparable to the retail price of electricity.
- Install increasing volumes of solar distributed generation
- Facilitate greatly expanded market penetration of cost-effective solar applications in California.

An OpenSees Approach

In order to reduce system cost and risk, SunLink continually works to improve its understanding of wind uplift effects on linked, flexible, roof bearing PV arrays.

A cost effective way to explore array response, and improve designs and design procedures, is through parametric static and dynamic analysis of structural models.

OpenSees – Modeling

Design Need	OpenSees Capability
Model nonlinear response and energy dissipation	Geometric and material nonlinear elements and hysteretic, Rayleigh and viscous damping
Model variations in ballast, connectivity and static load patterns	Programmatic scripting of components and load patterns including simulated loads

OpenSees – Execution Speed

Design Need	OpenSees Capability
Efficient analysis of static and dynamic cases	Shorter solution time compared to other programs and can run on HPC cloud machines
Because wind loads are stochastic many runs are needed to understand the range of responses	Many separate programs can economically run simultaneously

SUNLINK

R+C RUTHERFORD + CHEKENE

OpenSees – Data and Control

Design Need	OpenSees Capability
Large amounts of data require efficient reduction and management	Output control and file formats accommodate straightforward pre- and post-processing
Monitor model response and control execution	Programmatic control of analysis

Analysis Procedure Steps

University of Western Ontario Boundary Layer Wind Tunnel

Record Wind Pressure Time Series from Rigid Model

TCL or Navigator

Build Models or Model Templates

Build Load and Analysis Environment

Komodo + Python
or
Navigator + MATLAB

Execute Analysis

Next page

Procedure Steps – Model and Load Parameters

All Analysis

A

1. Array tilt, framing stiffness and strength, and connectivity patterns
2. Foot/ballast friction
3. Ballast patterns

B

4. Wind Direction
5. Wind speed

Analysis and Analysis Dependent Parameters

C

6. Dynamic
7. Time slice (explicit /spatially selected)

6. Static
7. Static load pattern (Patch or Snapshot)

D

8. Structural damping (Rayleigh)
9. Connection hysteresis (or elastic plus viscous)
10. Contact damping (viscous or Hertz)
11. Air damping (viscous)

8. Connection slip

A. Capacity
B. Demand
C. Demand
D. Capacity

SUNLINK

R+C RUTHERFORD + CHEKENE

R & D Objective

Given the stochastic nature of the loading, improve estimates of gust effects on array structures considering variations in geometry, stiffness, energy dissipation, connection and ballast patterns, tilt, roof location, and building geometry.

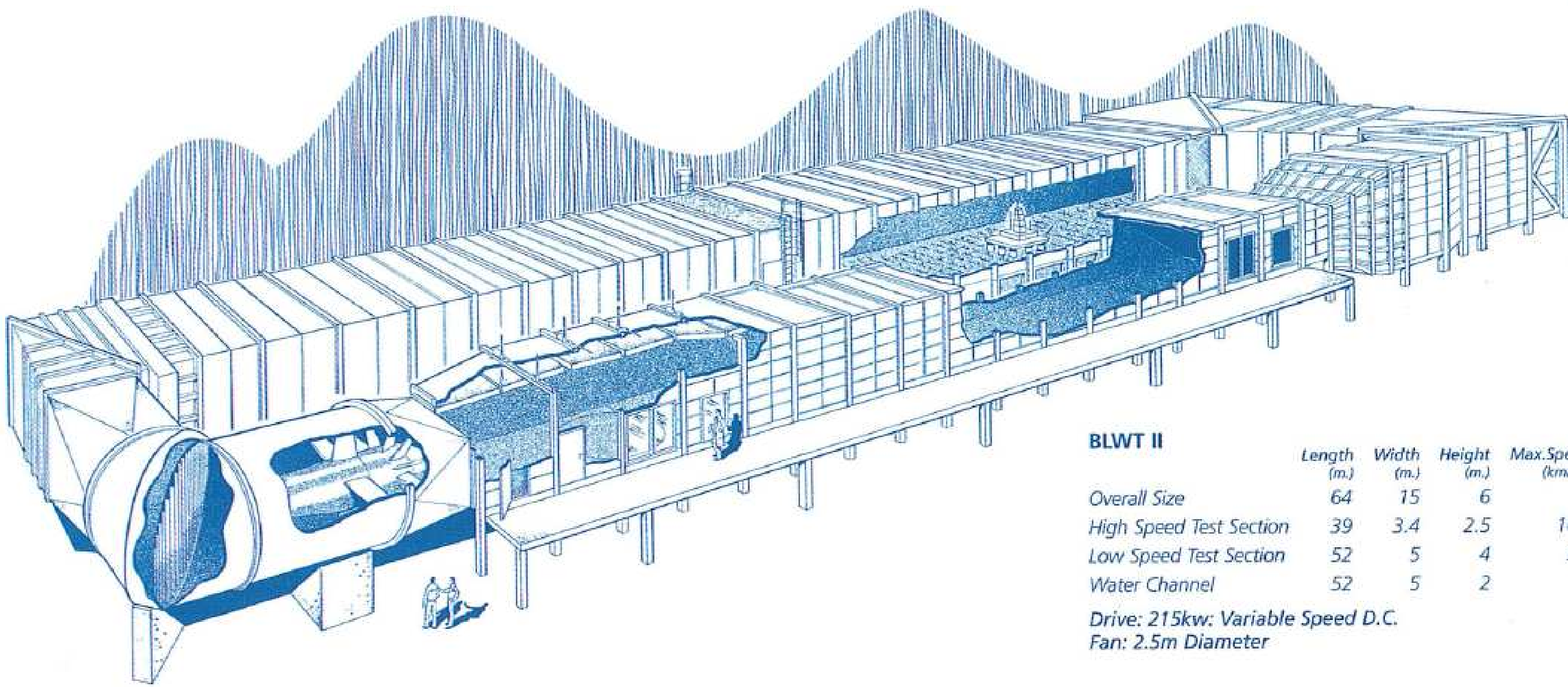
Investigate the following loading techniques:

- **Full length time-series (dynamic)**
- **Peak time slices based on targeted array areas (dynamic)**
- **Snapshots of peak loads at target array areas (static)**
- **Patch loading based on Cp vs. EWA curves (static)**

UWO Boundary Layer Wind Tunnel Schematic

High Speed Test Section:

Length = 128 ft   Width = 11ft   Height = 8ft   Max speed = 62 mph   (SunLink tests: ~34 mph)



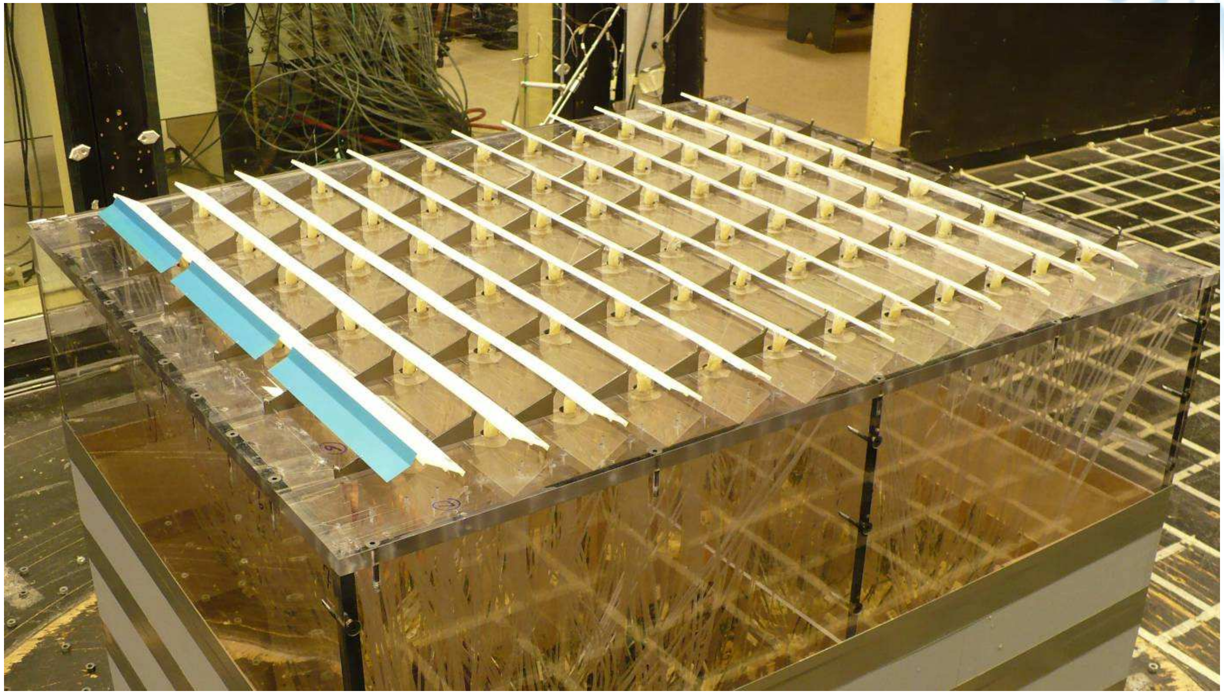
BLWT II

	Length (m.)	Width (m.)	Height (m.)	Max.Speed (km/hr.)
Overall Size	64	15	6	-
High Speed Test Section	39	3.4	2.5	100
Low Speed Test Section	52	5	4	36
Water Channel	52	5	2	-

Drive: 215kw: Variable Speed D.C.
Fan: 2.5m Diameter

Representative Wind Tunnel Model

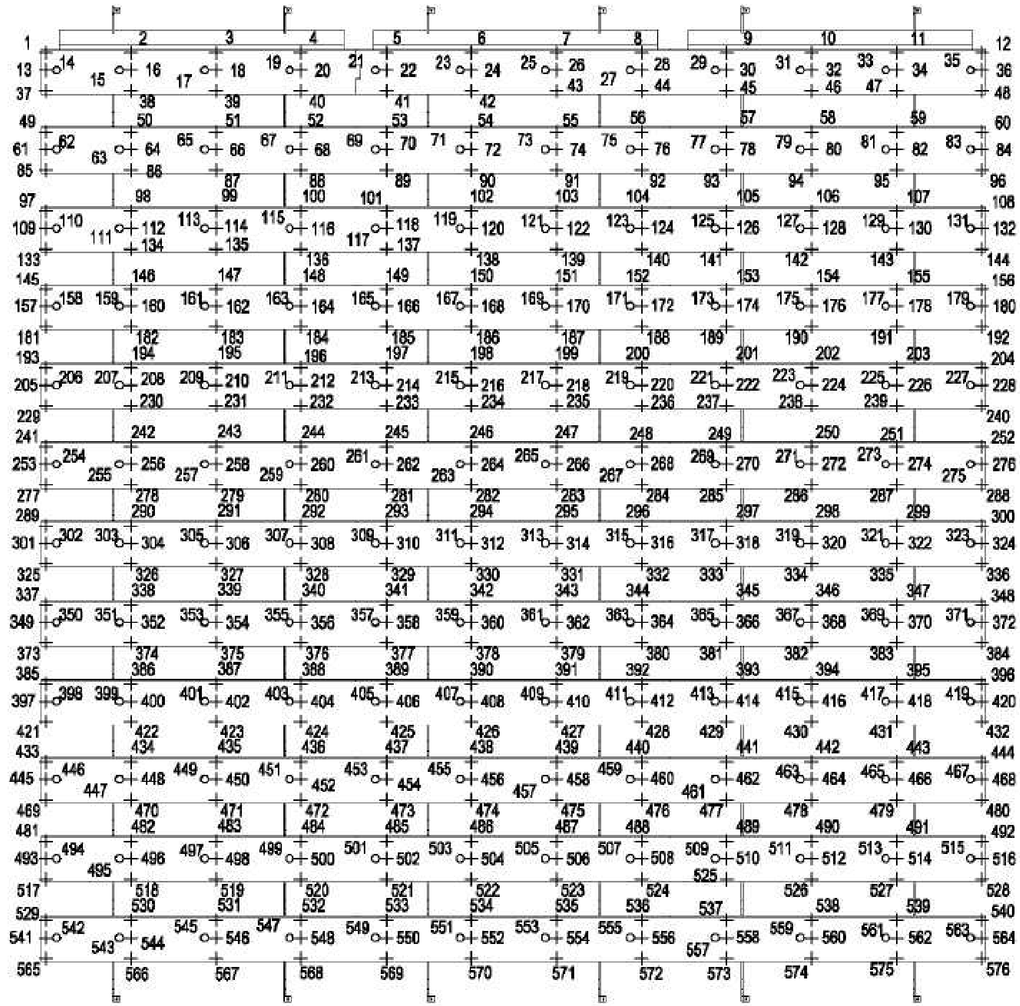
Full Scale Parameters
132 PV modules [~25 KW]
Array dimensions ~[80’ x 50’]
Test duration [~30 minutes]



Model Parameters
Rigidly attached
Time scale - 1:5 to 1:10
Length scale - 1:30
Pressure sampling at 400 Hz
576 pressure taps
65,0000 samples at each tap per test

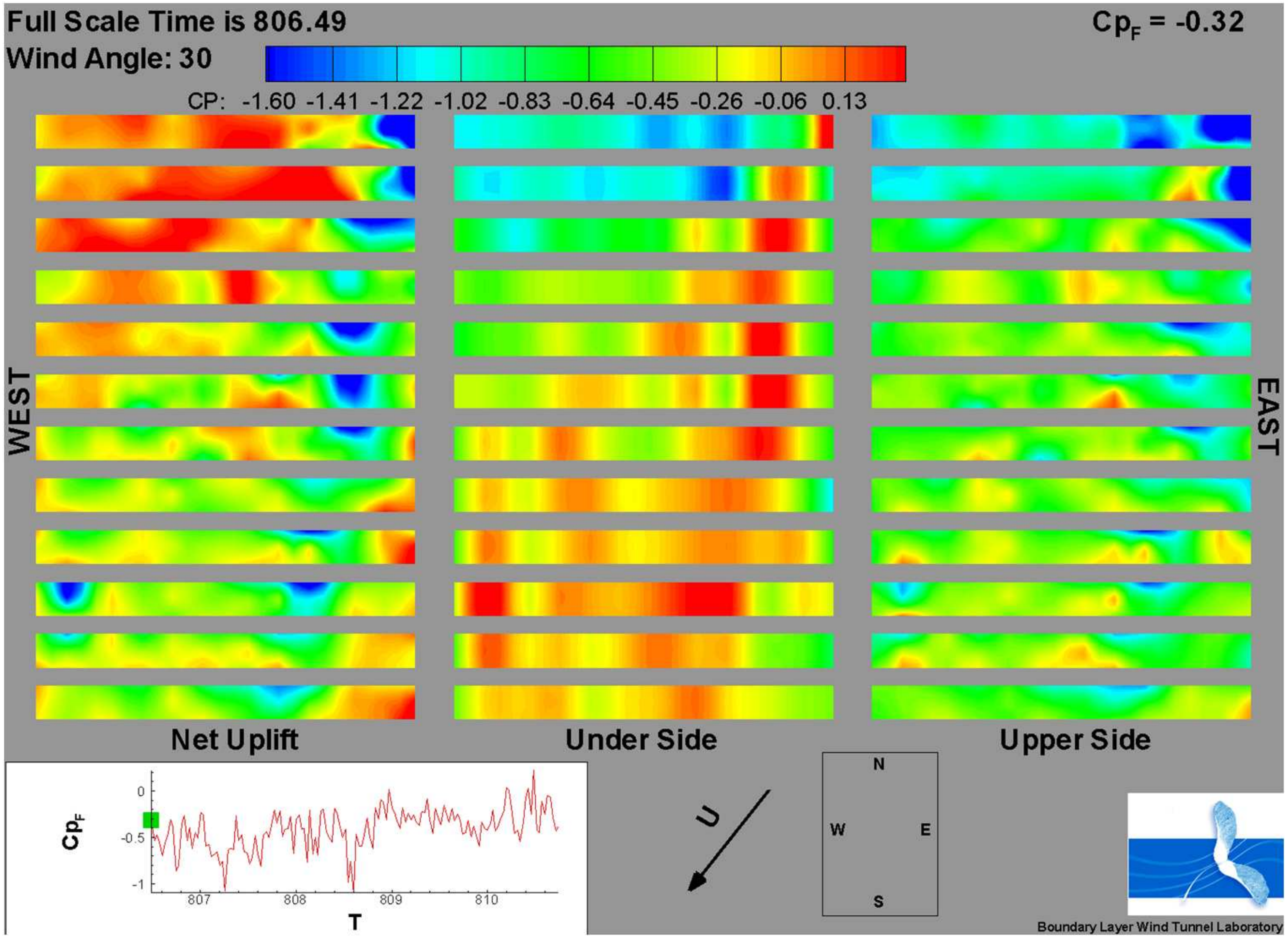
Test Procedures and Data
UWO performed ASCE 7 research for low-rise buildings
ASCE7-05 procedures used for array tests
1000+ tests on 75+ SunLink array configurations (tilt, setback, panel spacing, gaps to roof)
20+ GB of tap data
500+ hours of wind

Array Model Pressure Tap Layout



Wind Pressure Distribution

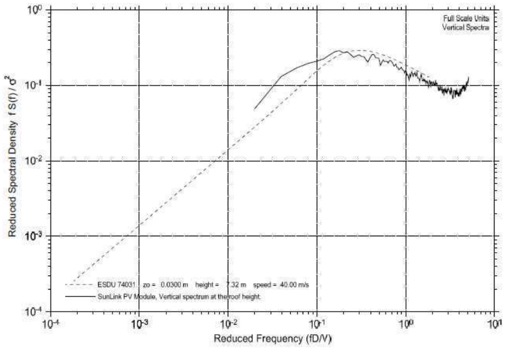
4 seconds full scale time, 1/3 speed playback



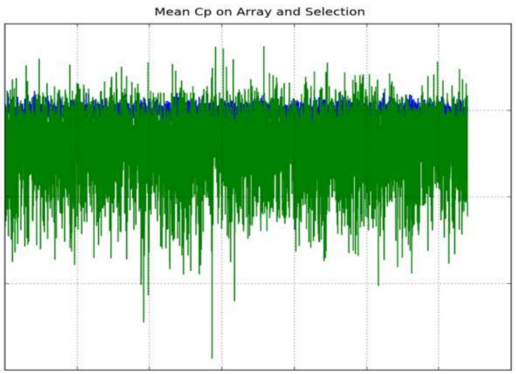
SUNLINK

R+C RUTHERFORD + CHEKENE

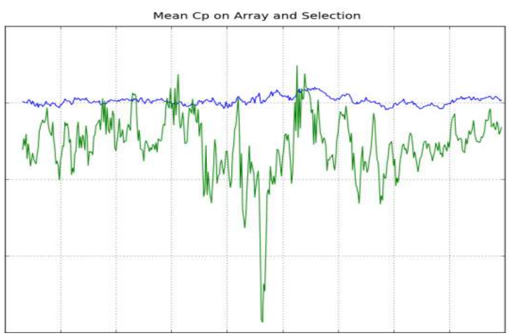
Wind Time Series Characteristics



Vertical Reduced Spectral Density for wind tunnel turbulent flow at 24 feet. Peak at about f = 1 sec.

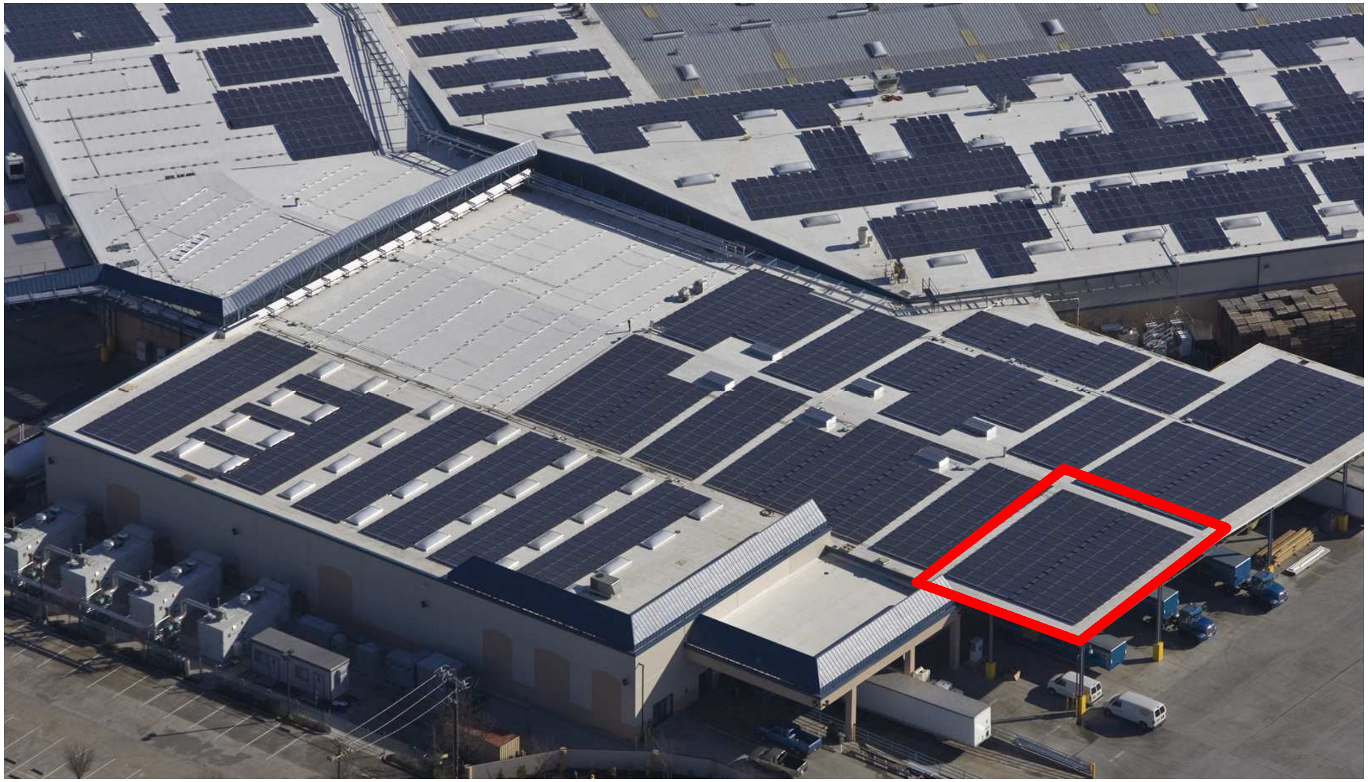


Average array Cp values over entire array (blue) and over 5% of the array (100 sf), for 1800 seconds



Average array Cp values over entire array and over an area 5% of the array, for 10 seconds

General Type of Sub-Array Represented by Model

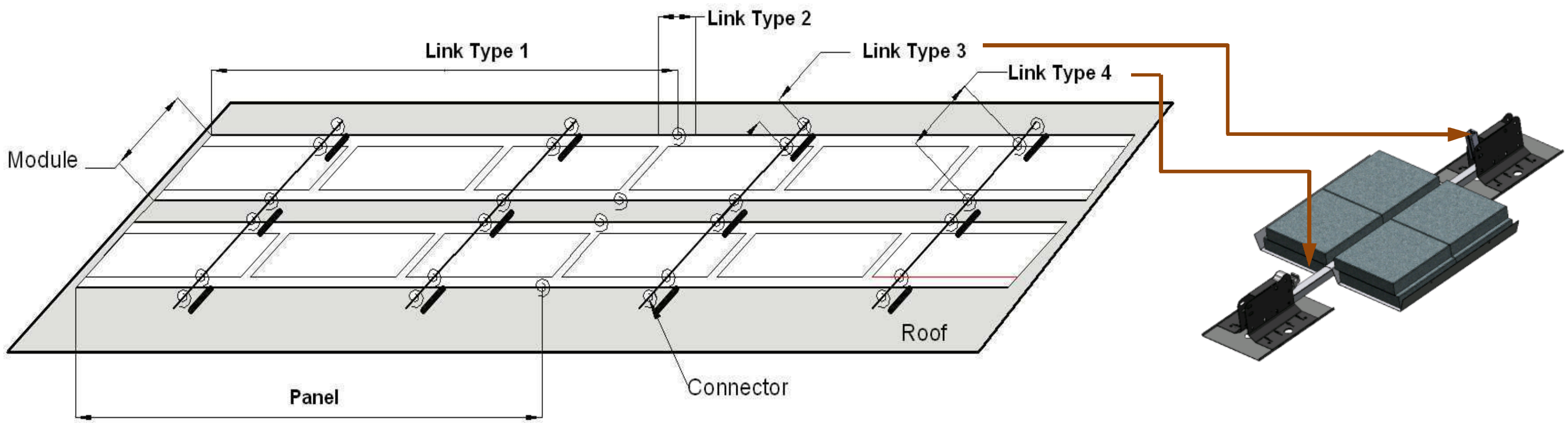


Typical Sub-Array Panels with Linked Aluminum Components

Reduction of Components to In-Plane Model

note: normal pressure loads transformed by cos, sin of tilt to nodal loads

Component	Model Element
Module -	elasticBeamColumn as in-plane brace
Link Type 1 -	elasticBeamColumn
Link Type 2 -	elasticBeamColumn
Link Type 3 -	elasticBeamColumn
Link Type 4 -	elasticBeamColumn
Link Type 4 Connection -	uniaxialMaterial ElasticPPGap
Link Type 2 Connection -	fixed
Link Type 4 Ballast Slider -	flatSliderBearing
Link Type 3 Slider -	flatSliderBearing



OpenSees TCL Procedure - File Structure

Program

Model Execution

program folder
- OpenSees.exe
- commonprocedures.tcl
- initialloads.tcl
- openstatic.tcl
- openth.tcl

Load Generation and Analysis Post-Processing

- directioncp.py
- gents.py
- gencp.py
- genlds.py
- plotdisp.py
- plotgeom.py
- anidisp.py
- anicp.py
- tapwtcp.py
- batchcp.cmd
- batchcmp.cmd

Model

Model Definition

model folder
- c01_k001_t05_bp1_r05_f08.tcl

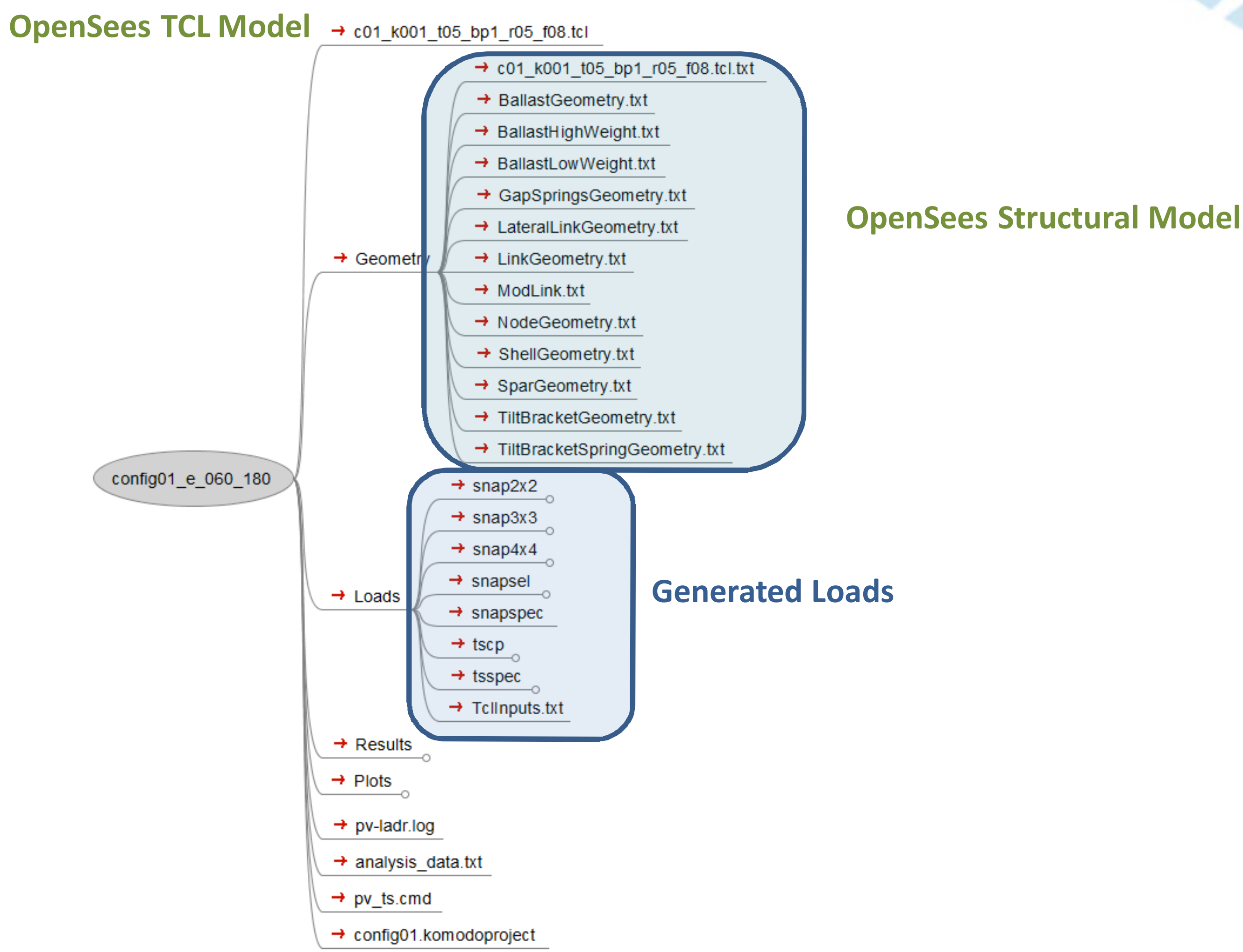
Pre-Processed Inputs

- Geometry
- Loads

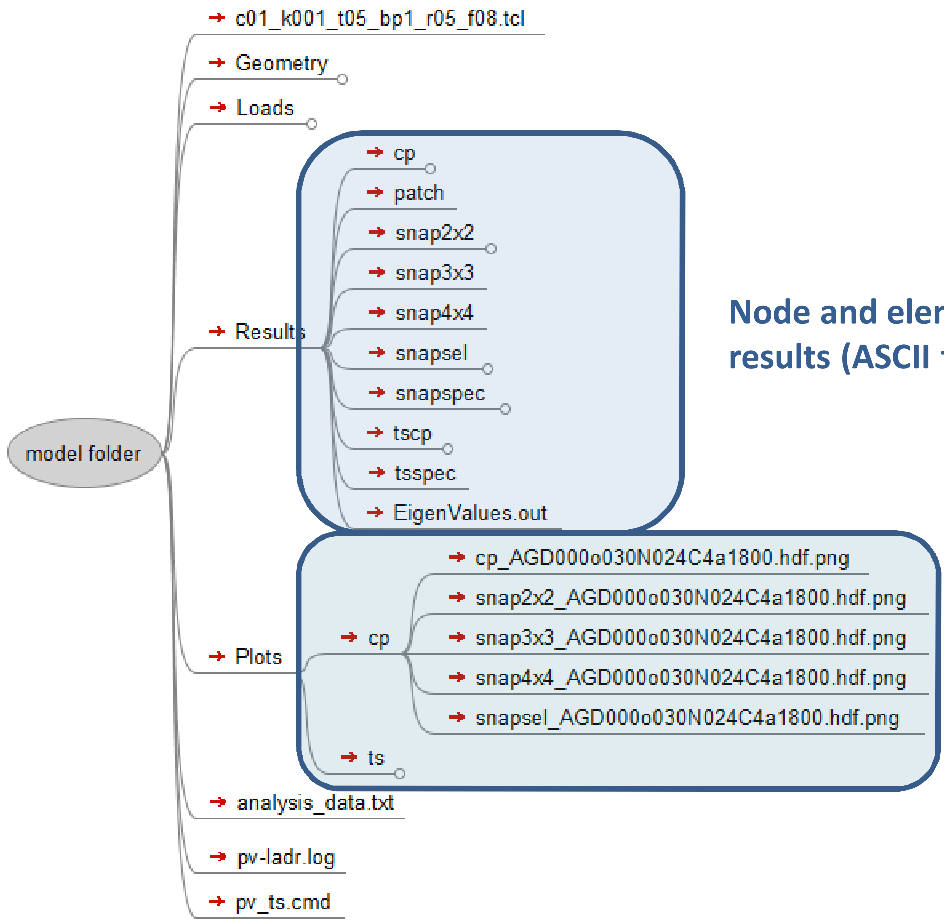
Results

- Results
- Plots
- analysis_data.txt
- pv-ladr.log
- pv_ts.cmd

File Structure - Inputs



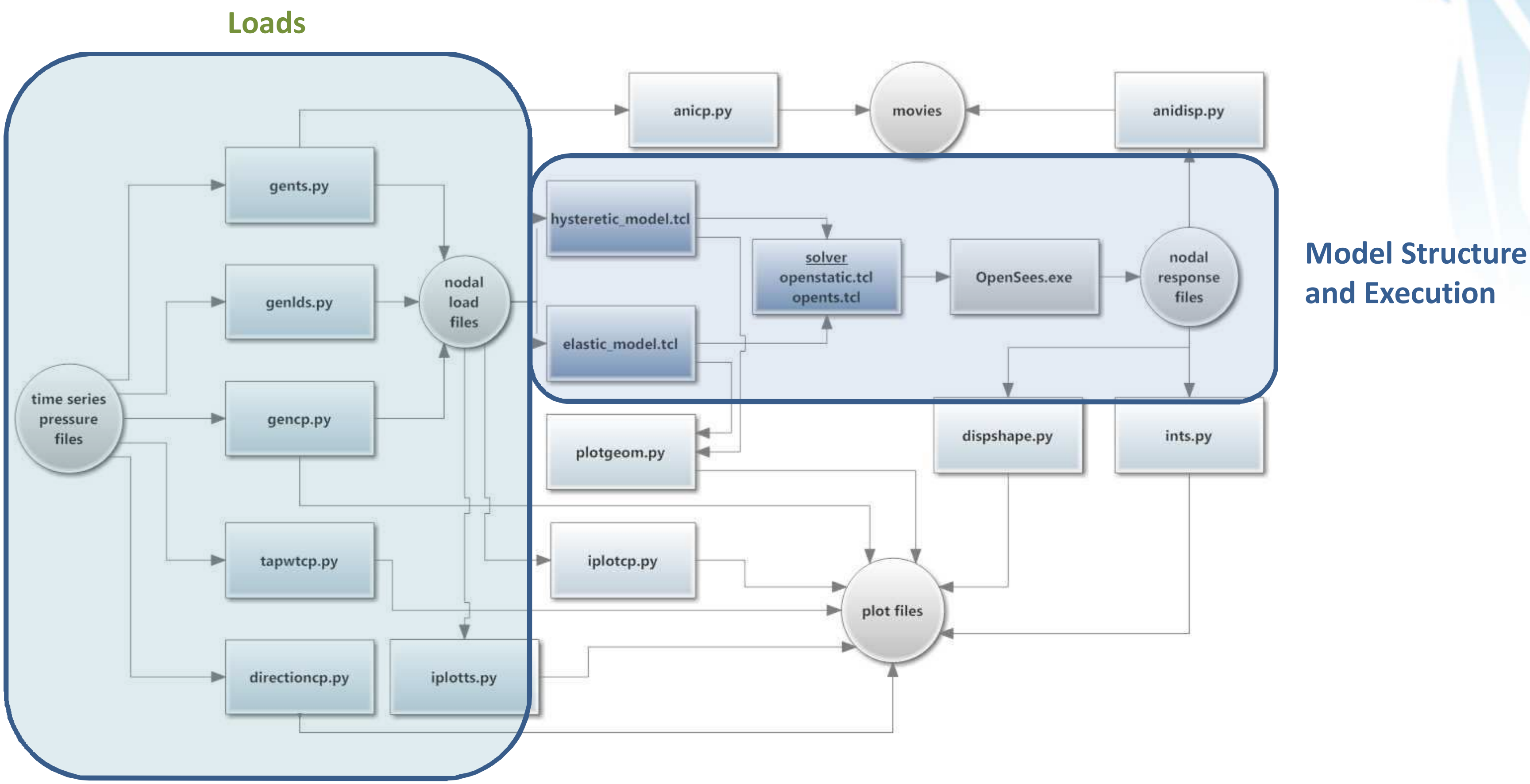
File Structure - Outputs



Node and elements results (ASCII files)

Plots and animations

TCL Procedure



OpenSees TCL Model

Table of Contents - Model Navigation

```
##  PV-LADR Hysteretic Model Template
##  Mon May 07 00:15:46 2012
##  GENERAL INFORMATION:
##  - user inputs are in sections 1.1, 1.2 and 1.3 and 4.
##  - for static analysis run genop.py to generate static load pattern
##  - for dynamic analysis run gents.py to generate nodal load time series
##  - geometry is based on the wind tunnel 12x12 array, 5 and 20 degree tilt
##  - global coordinates are x-y in the plane of the PV shells
##  - positive z out of the page, origin at array center
##  - units: kip, in, sec

set comment {
"TABLE OF CONTENTS
   1 USER INPUTS
    1.1   SOURCED LIBRARIES
    1.2   GEOMETRY PARAMETERS
    1.3   ANALYSIS PARAMETERS
    1.4   SETUP FOLDER STRUCTURE
      1.4.1  FINISH INITIALIZATION
   2 DEFINE GEOMETRY
    2.1   NODE STRUCTURE
      2.1.1  DEFINE GEOMETRY - TILT ANGLE
      2.1.2  CREATE TOP, "STRUCTURAL" NODES
      2.1.3  CREATE BOTTOM, "GROUND" NODES
      2.1.4  FINISH NODE CREATION
    2.2   SUPPORT CONDITIONS
      2.2.1  GENERAL NODE CONDITIONS
      2.2.2  CREATE SUPPORT CONDITION MATERIALS
      2.2.3  CREATE TILTBRACKET SLIDER SPRINGS
      2.2.4  CREATE BALLAST SLIDER SPRINGS
      2.2.5  CREATE GAP SPRINGS
    2.3   ELASTIC FRAME BEAM-COLUMN ELEMENTS
      2.3.1  DEFINE ELEMENT TRANSFORMATIONS
      2.3.2  CREATE SPAR ELEMENTS
      2.3.3  CREATE TILTBRACKET ELEMENTS
      2.3.4  CREATE LATERAL LINK ELEMENTS
    2.4   HYSTERETIC ELEMENTS
      2.4.1  CREATE LONGITUDINAL LINK MATERIALS
      2.4.2  CREATE LONGITUDINAL LINK SECTIONS
      2.4.3  CREATE LONGITUDINAL LINK ELEMENTS
    2.5   SHELL ELEMENTS (not included in model but used in geometry routines)
      2.5.1  CREATE SHELL SECTION
      2.5.2  CREATE SHELL ELEMENTS
3 DEFINE MASS, WEIGHTS AND DAMPING
        3.1   SET FRAME NODAL MASS and weights (read from init file)
        3.2   SET BALLAST MASS & LOADS
4 ANALYSIS TYPE  (set analysis type in this section)
"}
```

SUNLINK

R+C RUTHERFORD + CHEKENE

OpenSees TCL Model Interface

Komodo - customizable, Open Source, cross platform, multi-language IDE

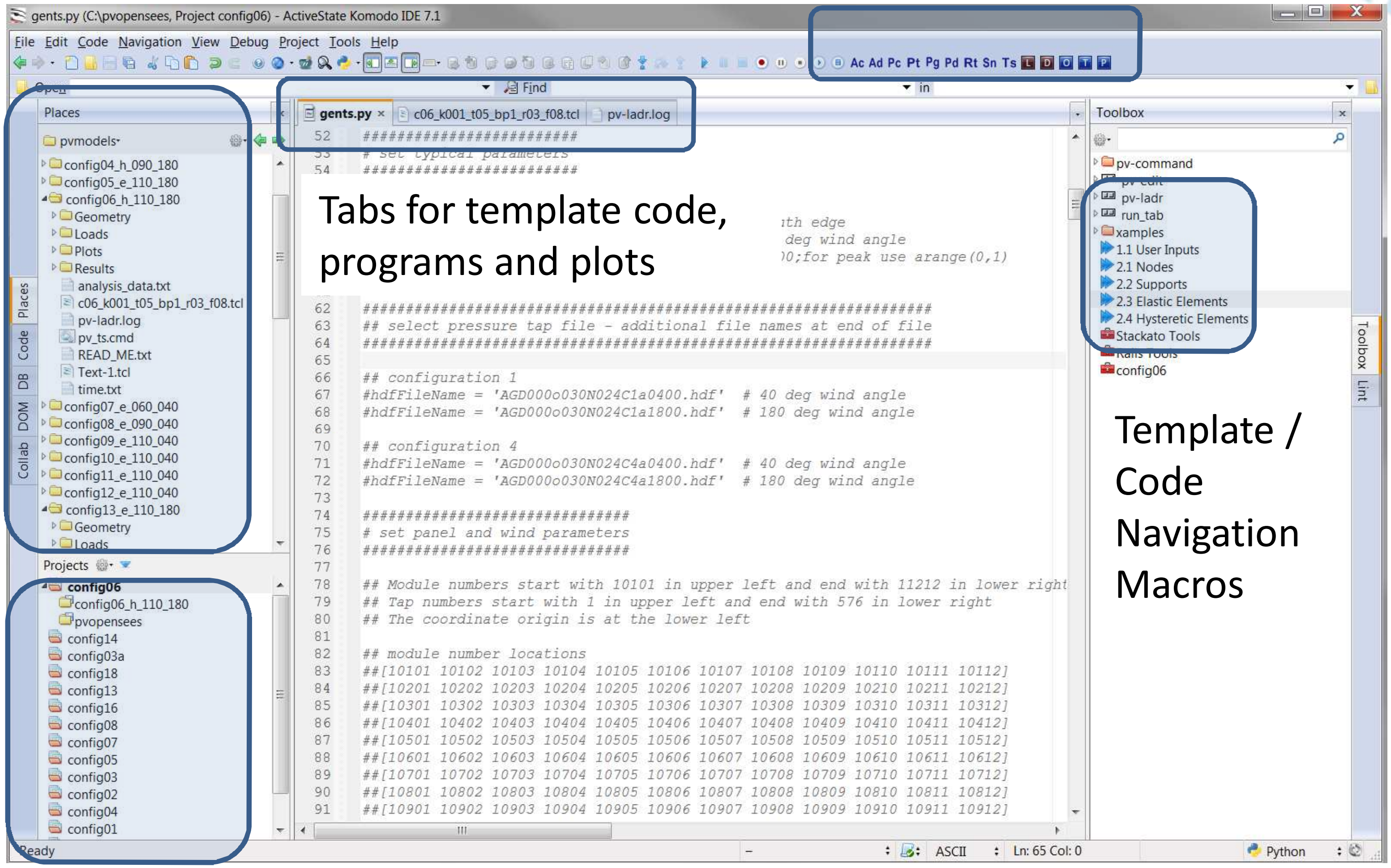
Program Execution Buttons

Mode Files

Models

Tabs for template code, programs and plots

Template / Code Navigation Macros



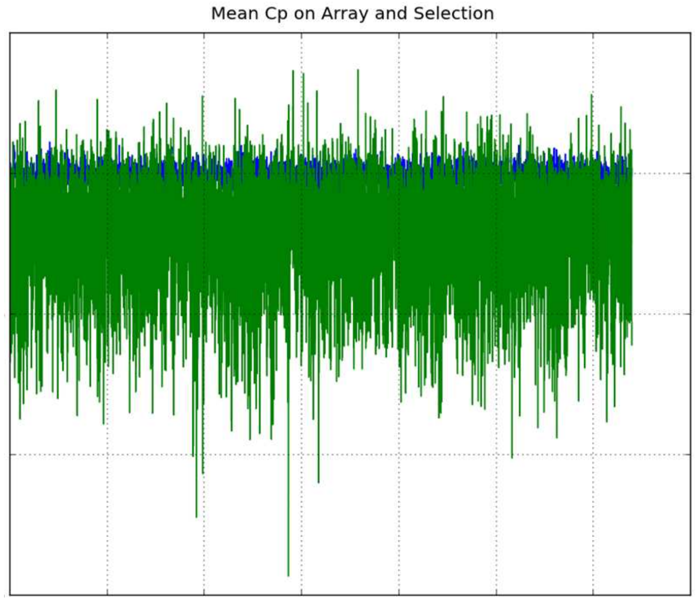
SUNLINK

R+C RUTHERFORD + CHEKENE

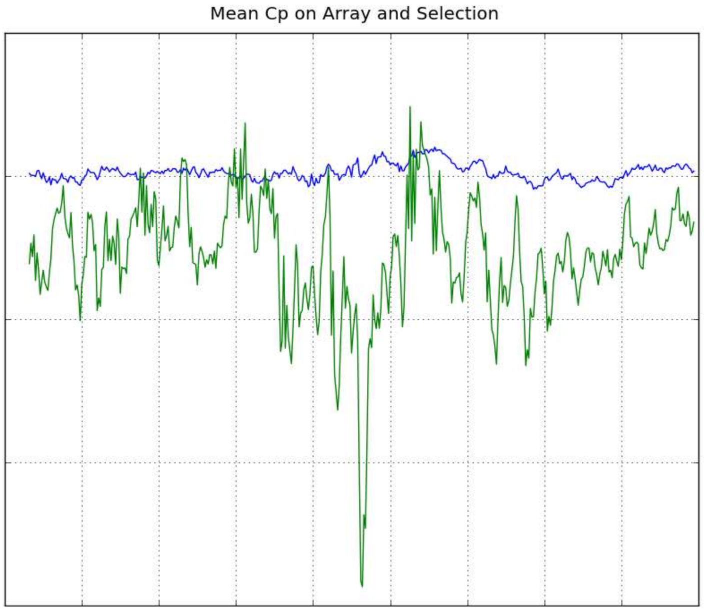
Wind Load Time Parameter

Select Time Slice

1800 seconds



10 seconds

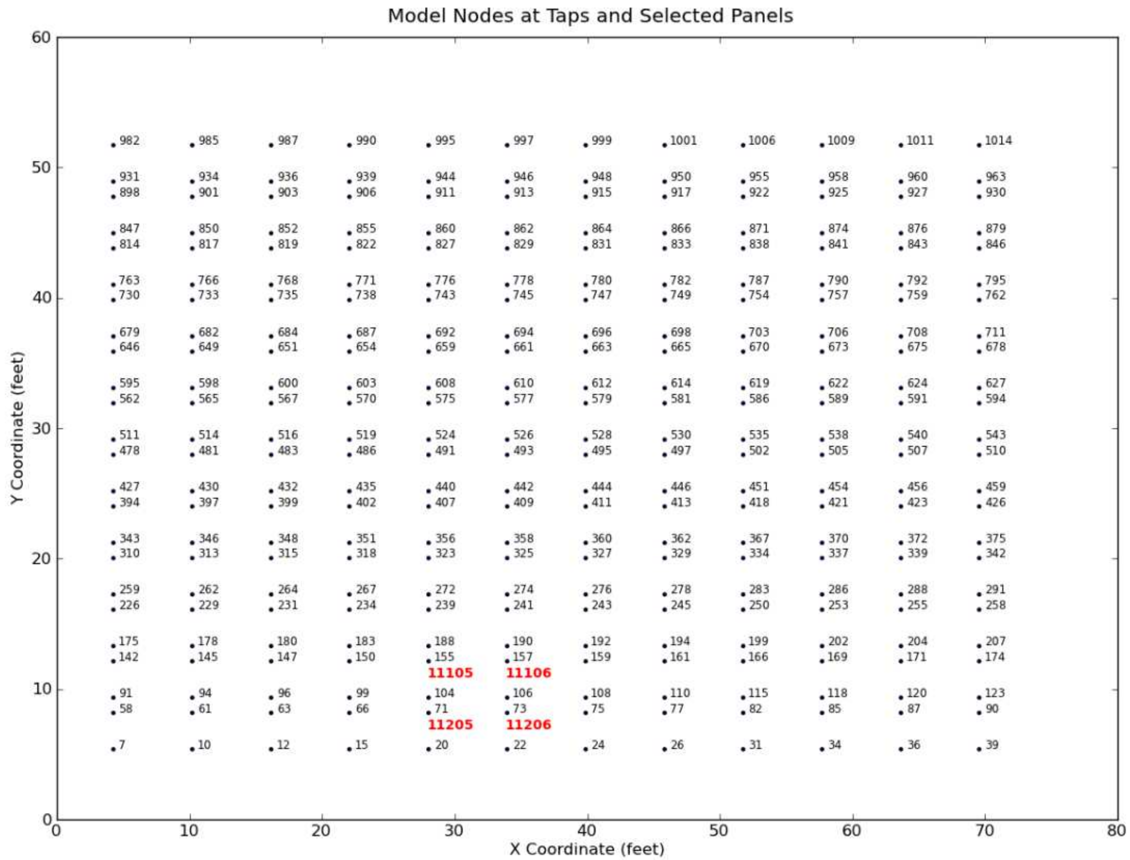


SUNLINK

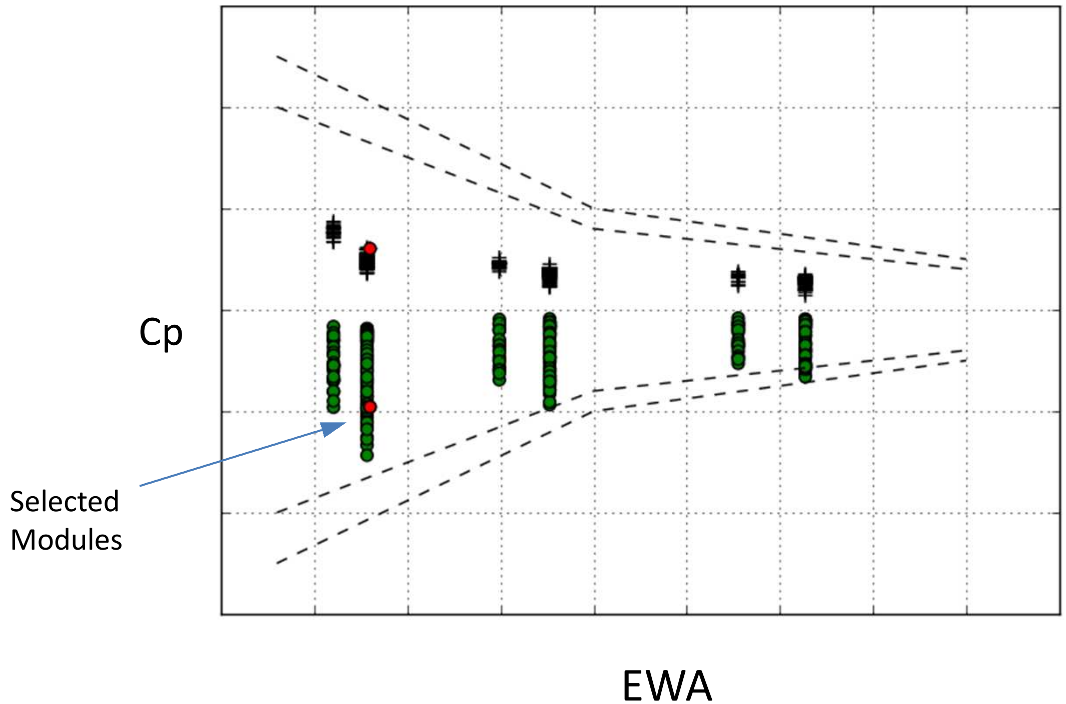
R+C RUTHERFORD + CHEKENE

Wind Load Spatial Parameter

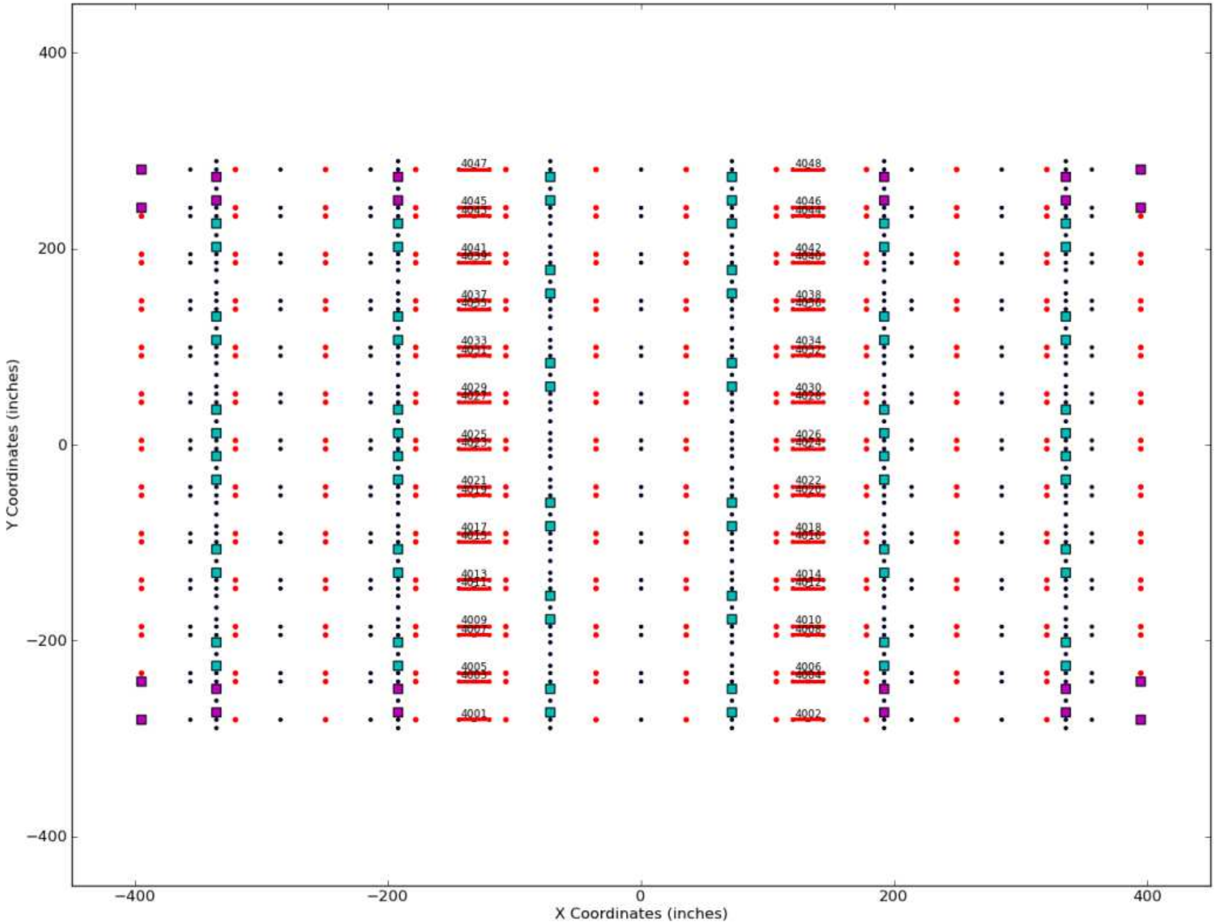
Time slice can be associated with the peak response at a location



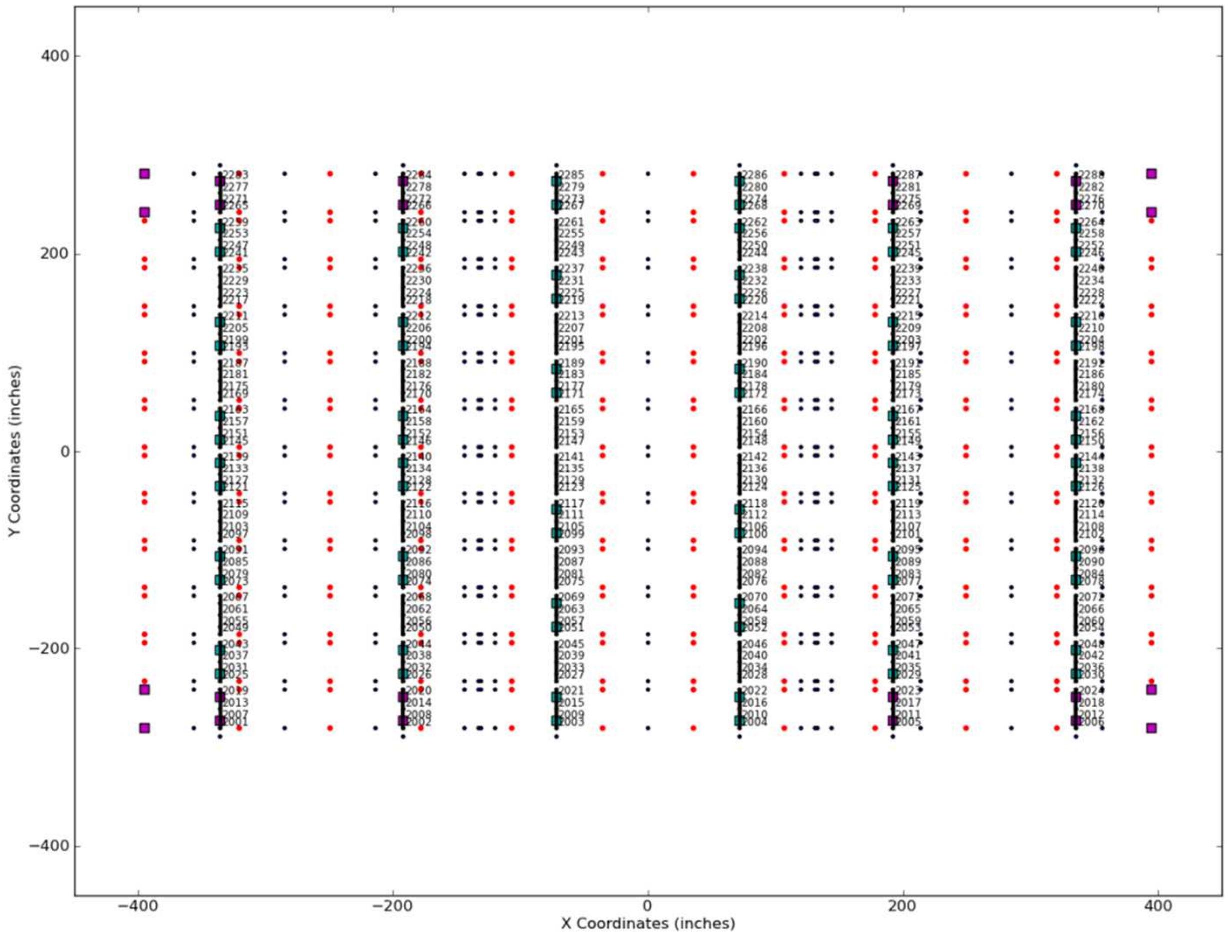
Wind Load Cp Spatial Envelope



Ballast and Link 2 Geometry



Ballast and Link 3,4 Geometry

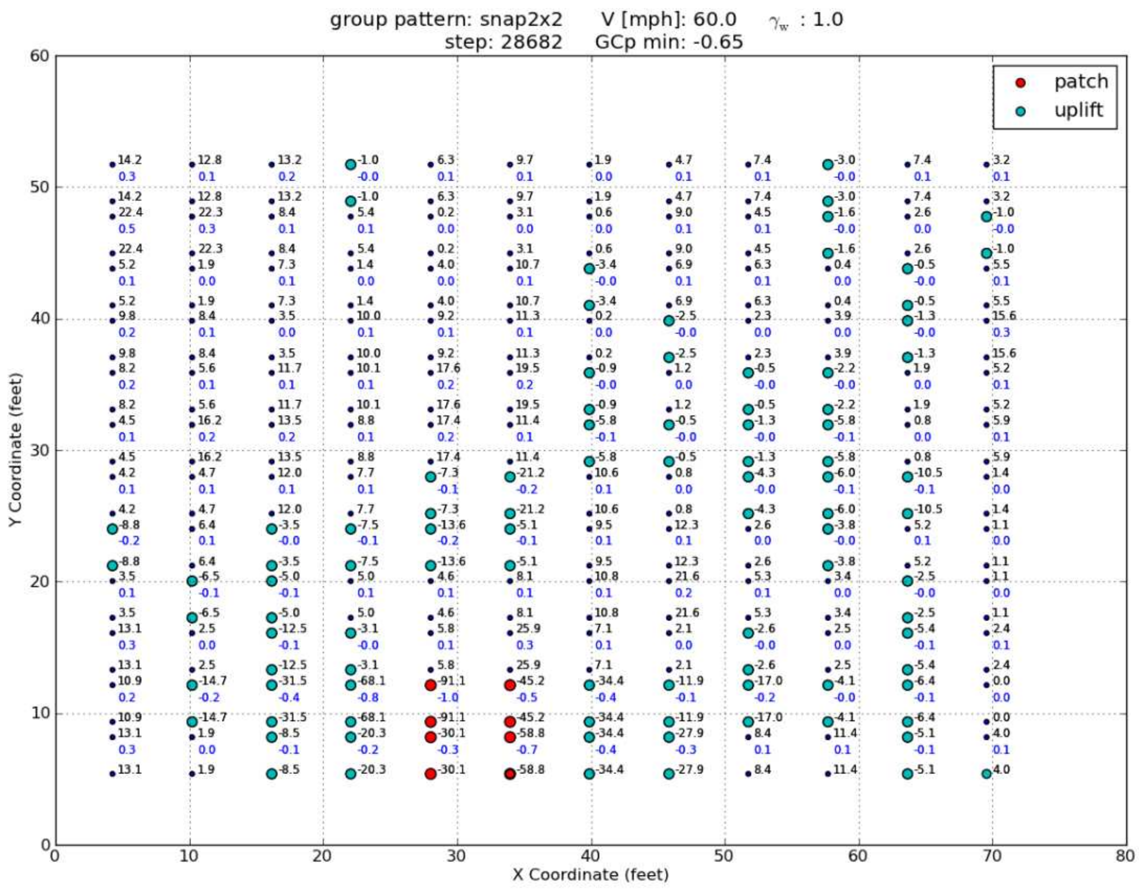


SUNLINK

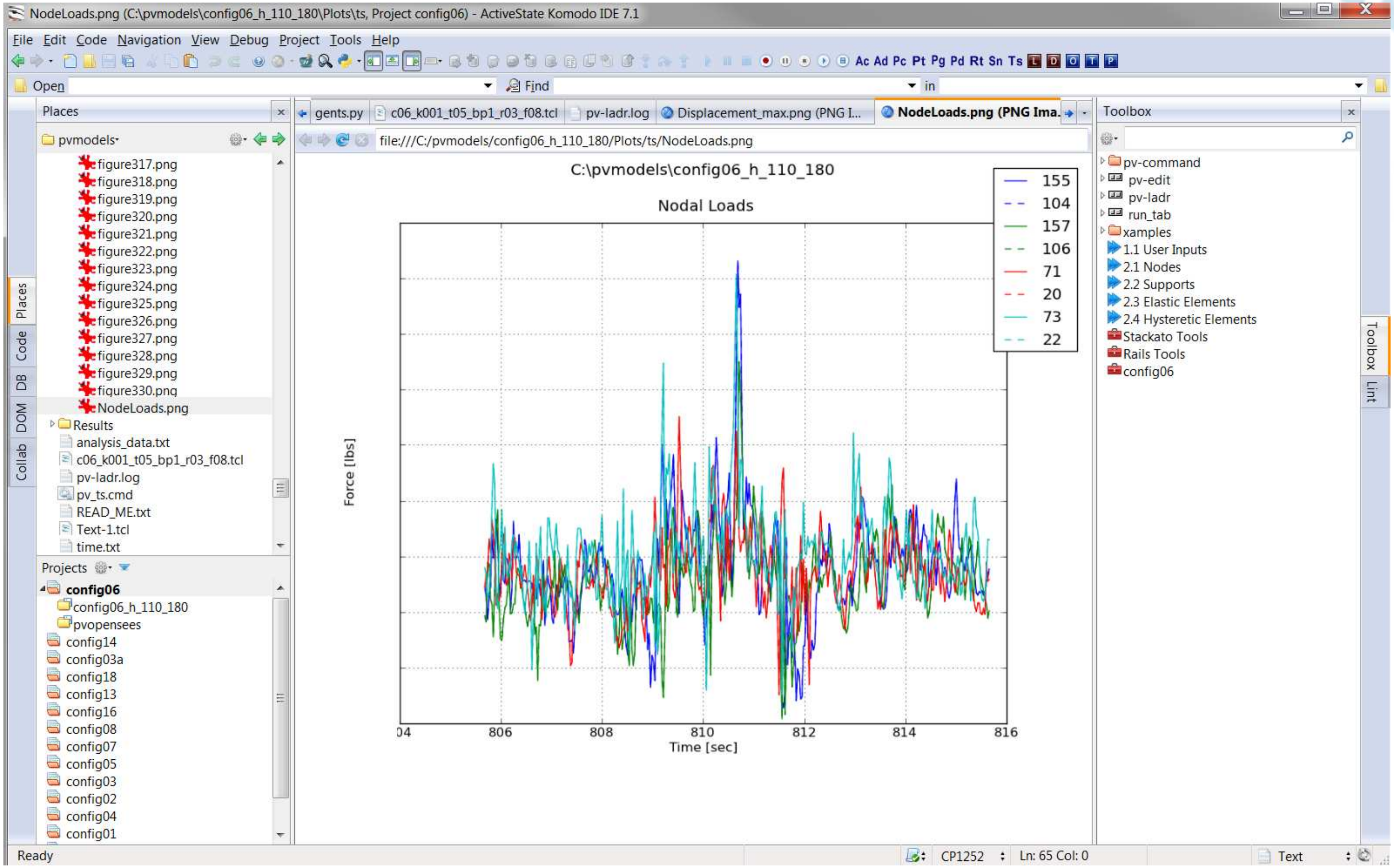
R+C RUTHERFORD + CHEKENE

Wind Load Nodal Forces Snapshot

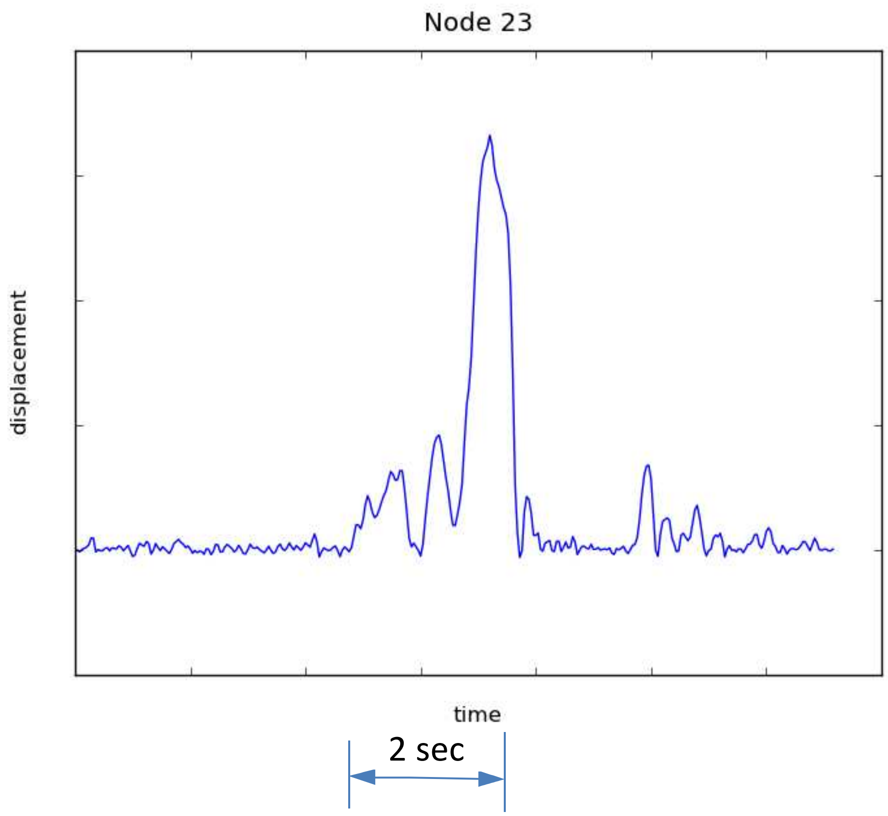
Static Load Pattern (2x2 Peak)



Wind Time Series Nodal Loads

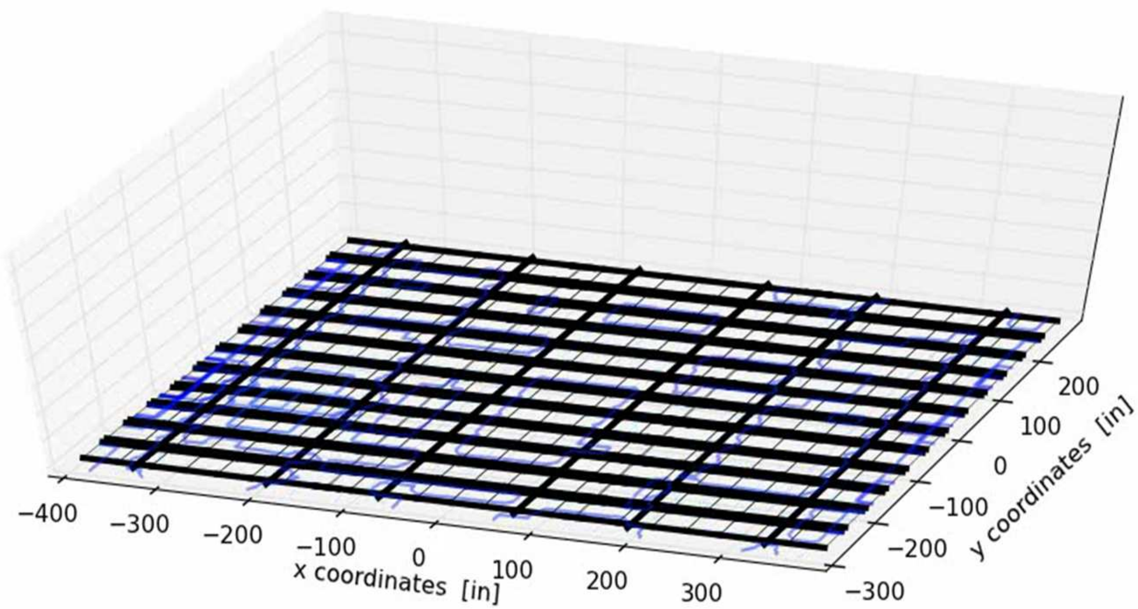


Common Response History Prior to Peak Displacements



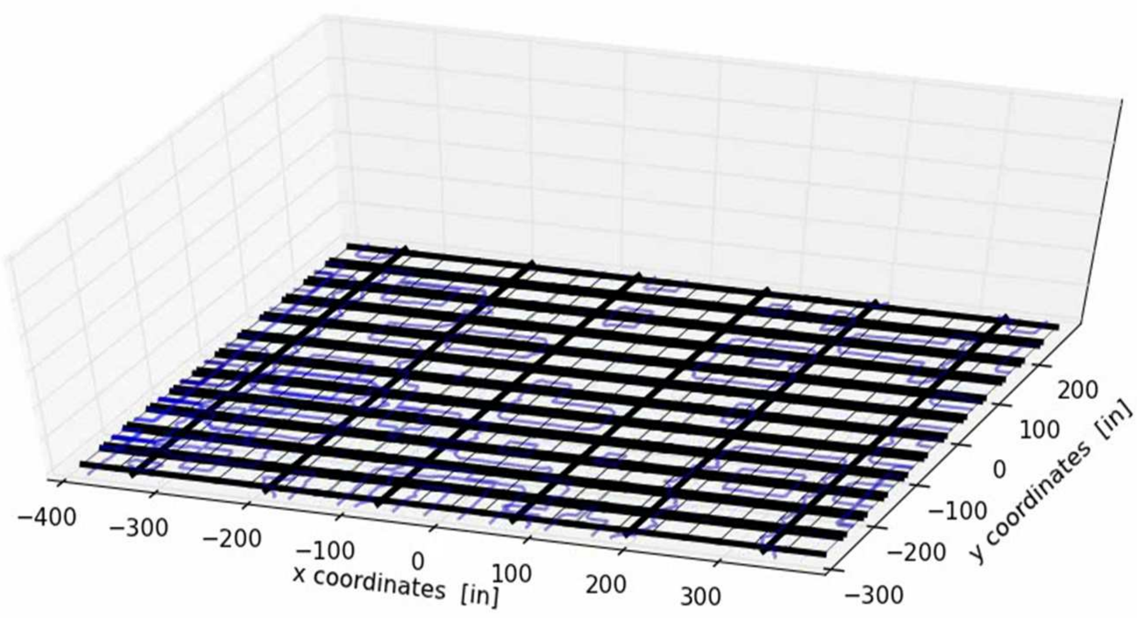
Array Model Displacements

Less Stiff, More Ballast, 30 seconds real time , ¾ speed playback



Array Model Displacements

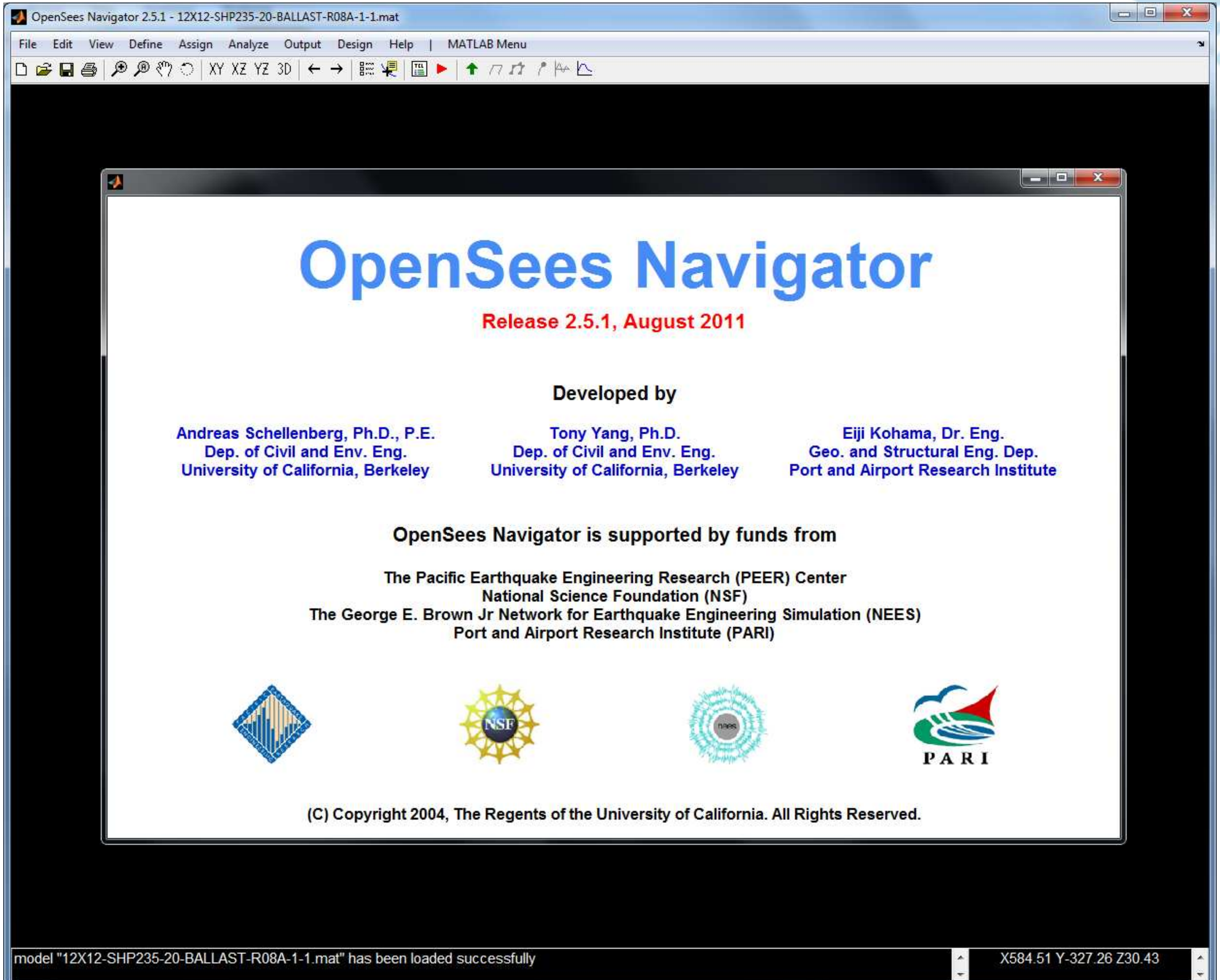
Stiffer, Less Ballast, 10 seconds real time, ½ speed playback



SUNLINK

R+C RUTHERFORD + CHEKENE

OpenSees Navigator Procedures



What is OpenSees Navigator ?

- 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.

SUNLINK

R+C RUTHERFORD + CHEKENE

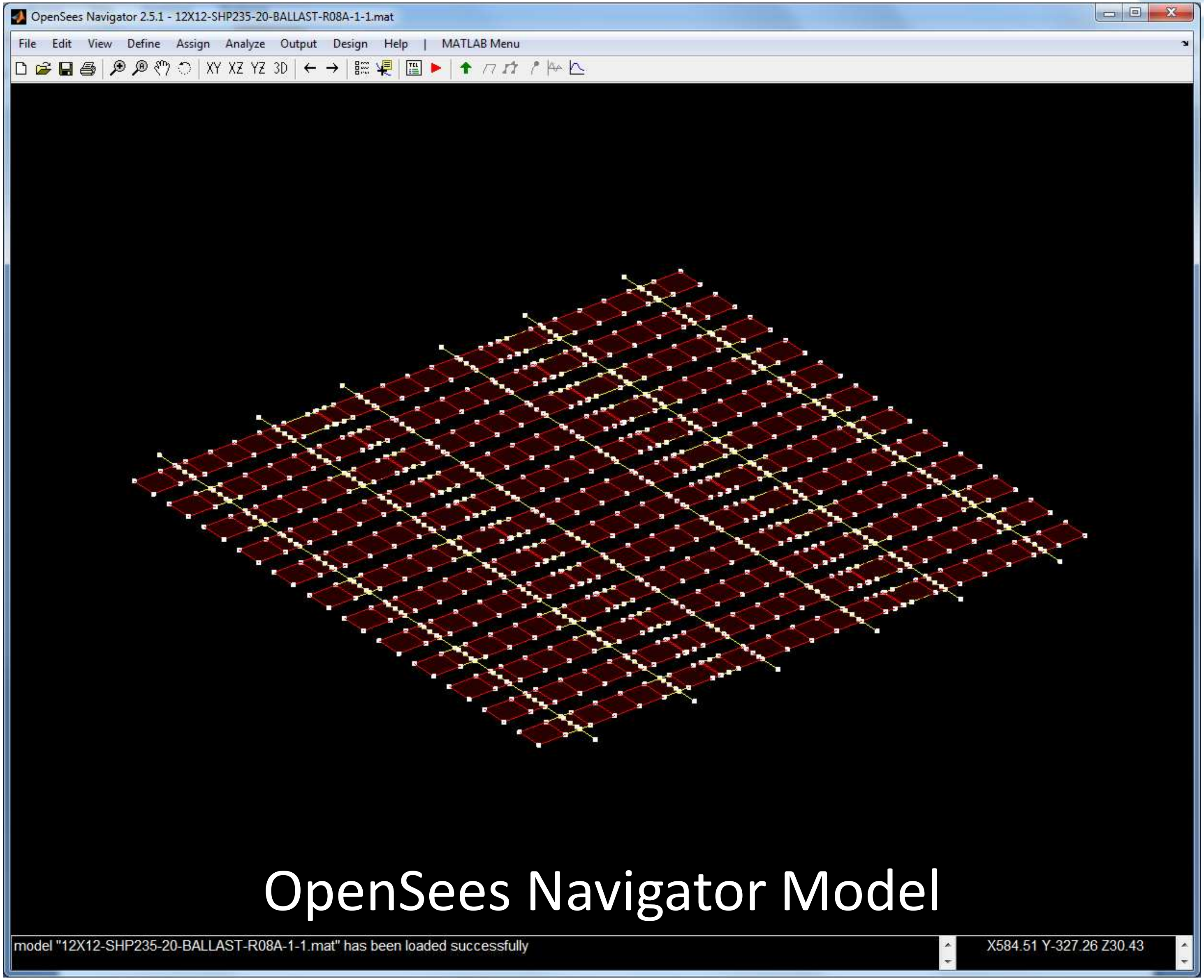
Motivation for using OpenSees Navigator

- Replace the TCL text input with graphical 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.
- Provides many robust plotting algorithms and is very effective in generating the plots for engineering applications.
- Flexible to use and requires no programming skill.

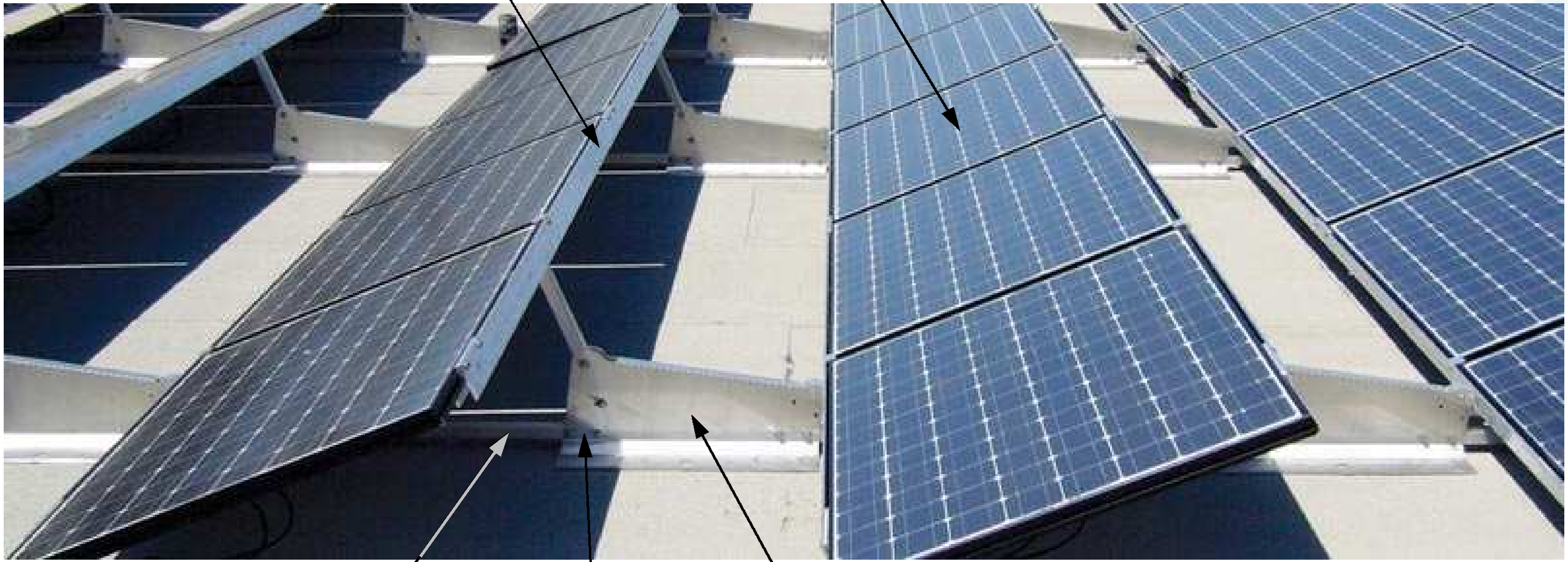
Setting Up and Analyzing the Model

1. Create SAP2000 model to define geometry of PV array
2. Run custom Matlab functions that use the SAP2000 API to generate the loads
3. Export the SAP2000 model to an .s2k file and import the .s2k file into OpenSees Navigator
4. Modify the model and analysis parameters inside OpenSees Navigator using the GUI or custom Matlab functions
5. Run the analyses
6. Use OpenSees Navigator post-processing capabilities to plot results and generate movies

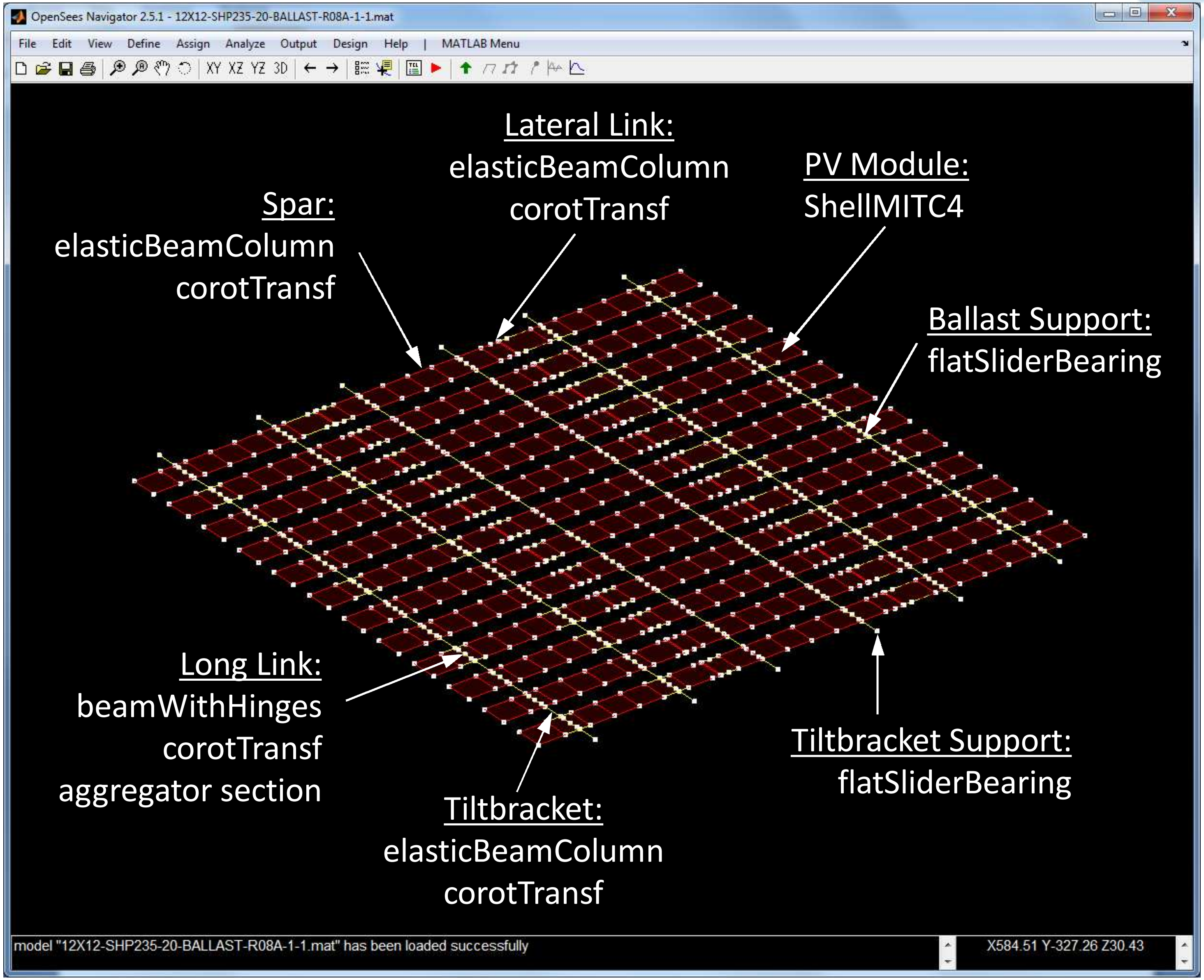
Converting the Model from SAP to OSN



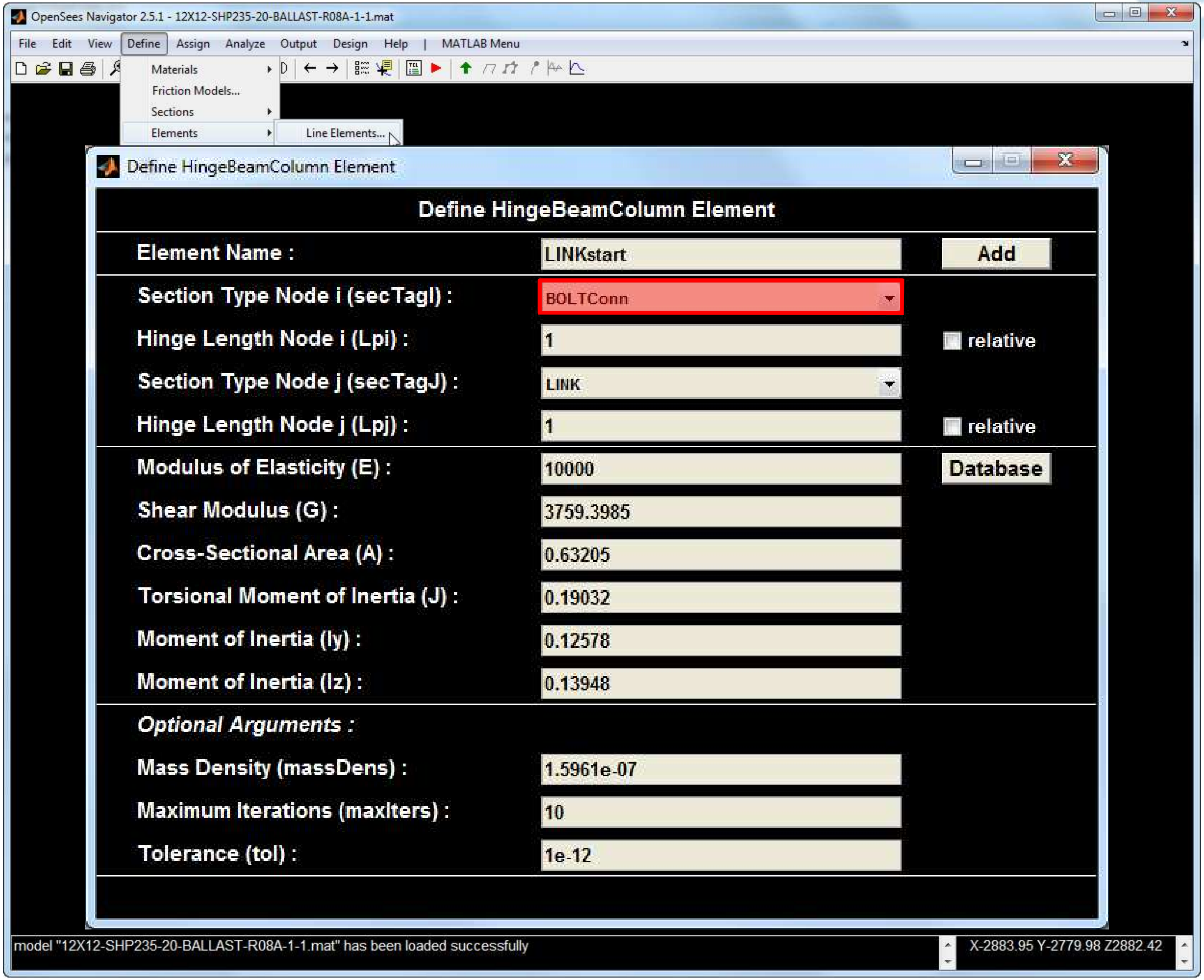
SunLink System Components



OpenSees Model Components



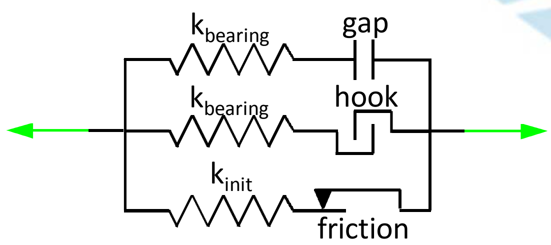
Define Element: Line Element



Modeling the Bolted Connection

Define Steel01 Material

Material Name :	BoltHyst	Add
Yield Stress (Fy) :	3.65	
Modulus of Elasticity (E) :	1394.8	
Hardening Ratio (b) :	0.0001	
Optional Parameters :		
Iso Hardening Parameter (a1) :	0	
Iso Hardening Parameter (a2) :	1	
Iso Hardening Parameter (a3) :	0	
Iso Hardening Parameter (a4) :	1	



Define ElasticPPGap Material

Material Name :	BoltGapPos	Add
Modulus of Elasticity (E) :	10000	
Yield Stress (Fy) :	10000	
Initial Gap (gap) :	0.05	
Optional Parameters :		
Hardening Ratio (eta) :	0.99999	
Damage Switch (damage) :	off	

Define Parallel Material

Material Name :	BoltConnMz	Add
Add Material :	BoltHyst BoltGapPos BoltGapNeg BoltConnPx BoltConnMz BoltConnMy BoltConnTx	

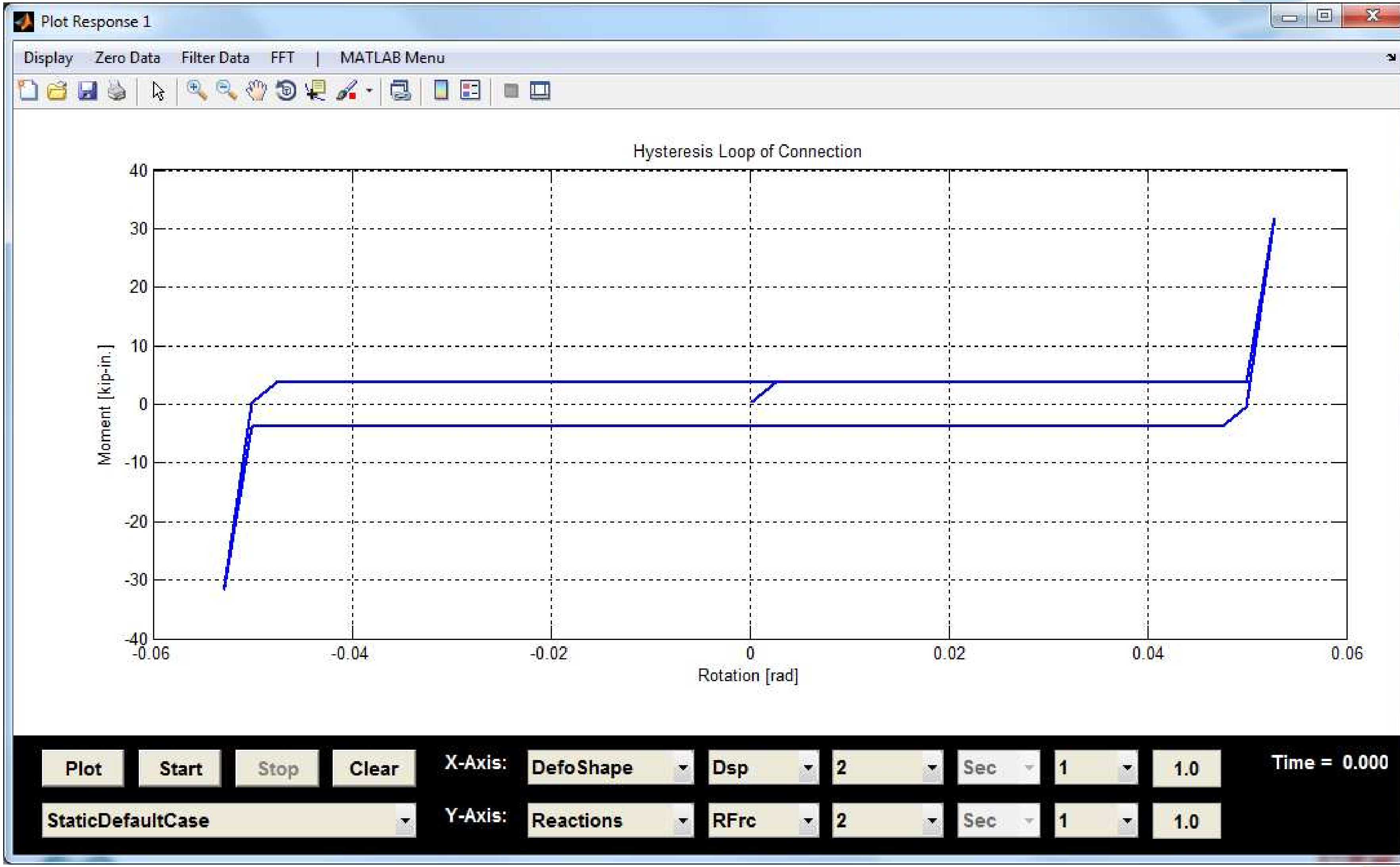
Define ElasticPPGap Material

Material Name :	BoltGapNeg	Add
Modulus of Elasticity (E) :	10000	
Yield Stress (Fy) :	-10000	
Initial Gap (gap) :	-0.05	
Optional Parameters :		
Hardening Ratio (eta) :	0.99999	
Damage Switch (damage) :	off	

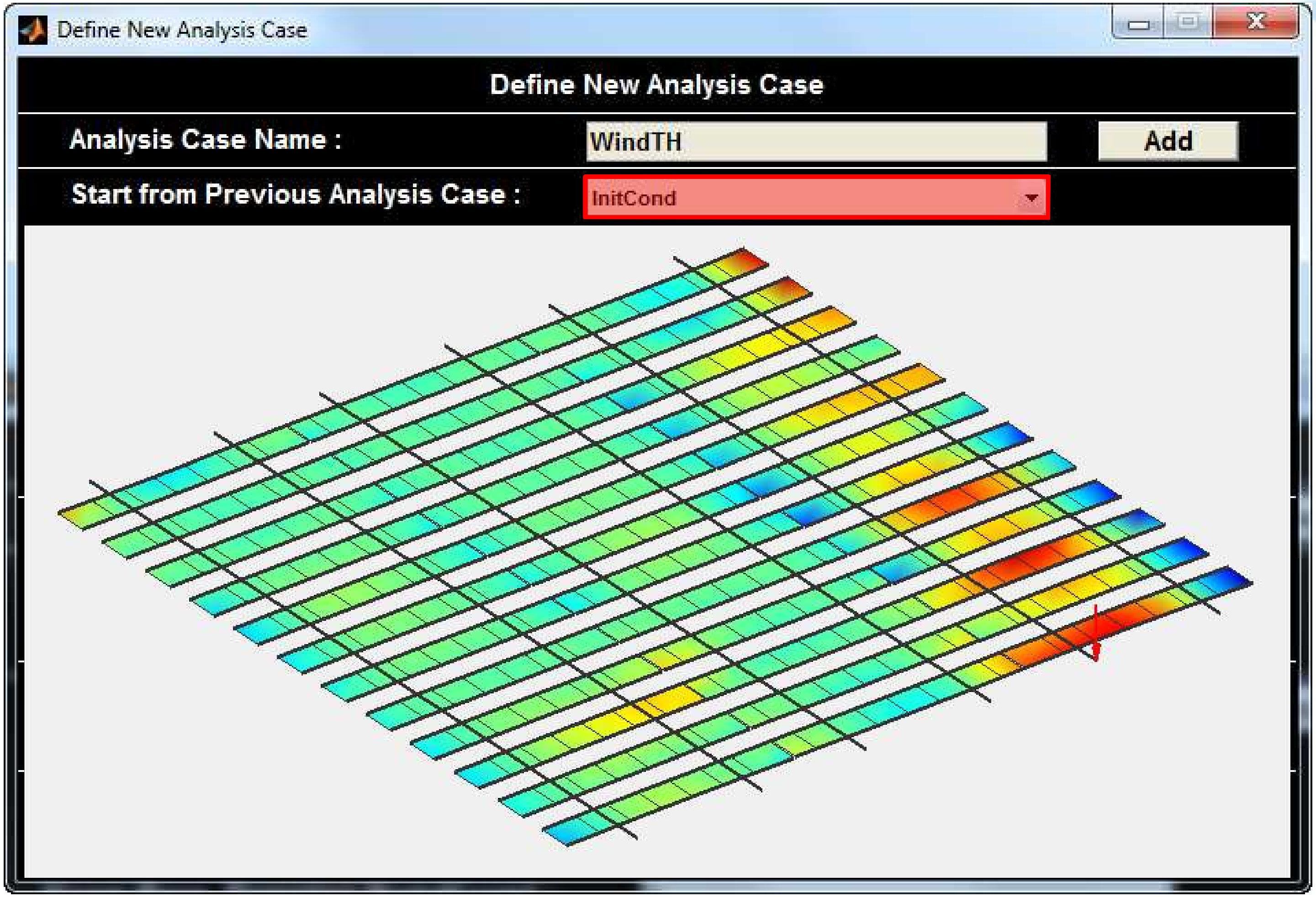
Define Aggregator Section

Section Name :		BOLTConn
Select Materials :	P :	BoltConnPx
	Vy :	None
	Vz :	None
	T :	BoltConnTx
	My :	BoltConnMy
	Mz :	BoltConnMz
Add Existing Section :		None

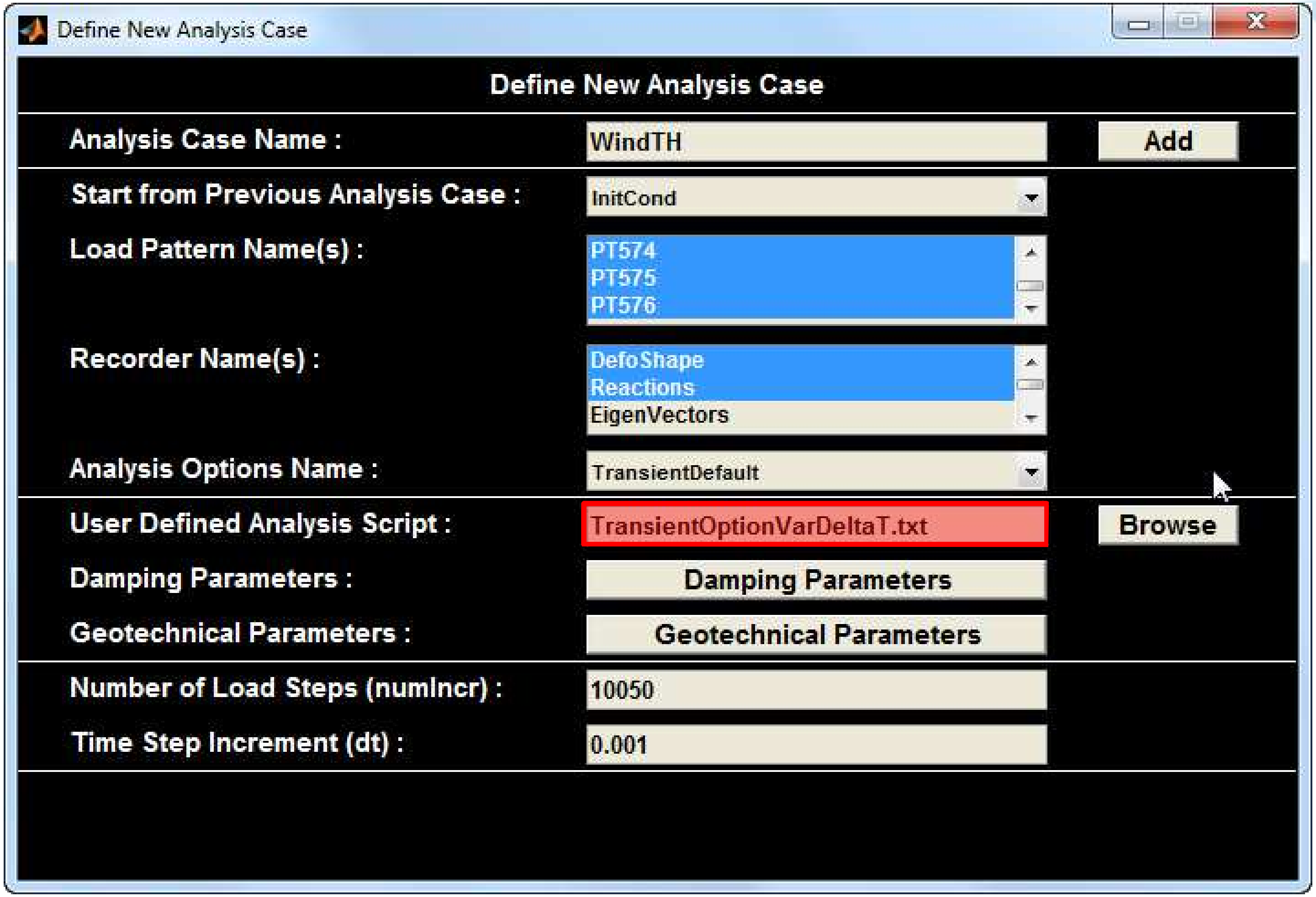
Post Processing: Plot Hysteresis Loop



Define Analysis Case: Initial Condition



Define Analysis Case: User Defined Script



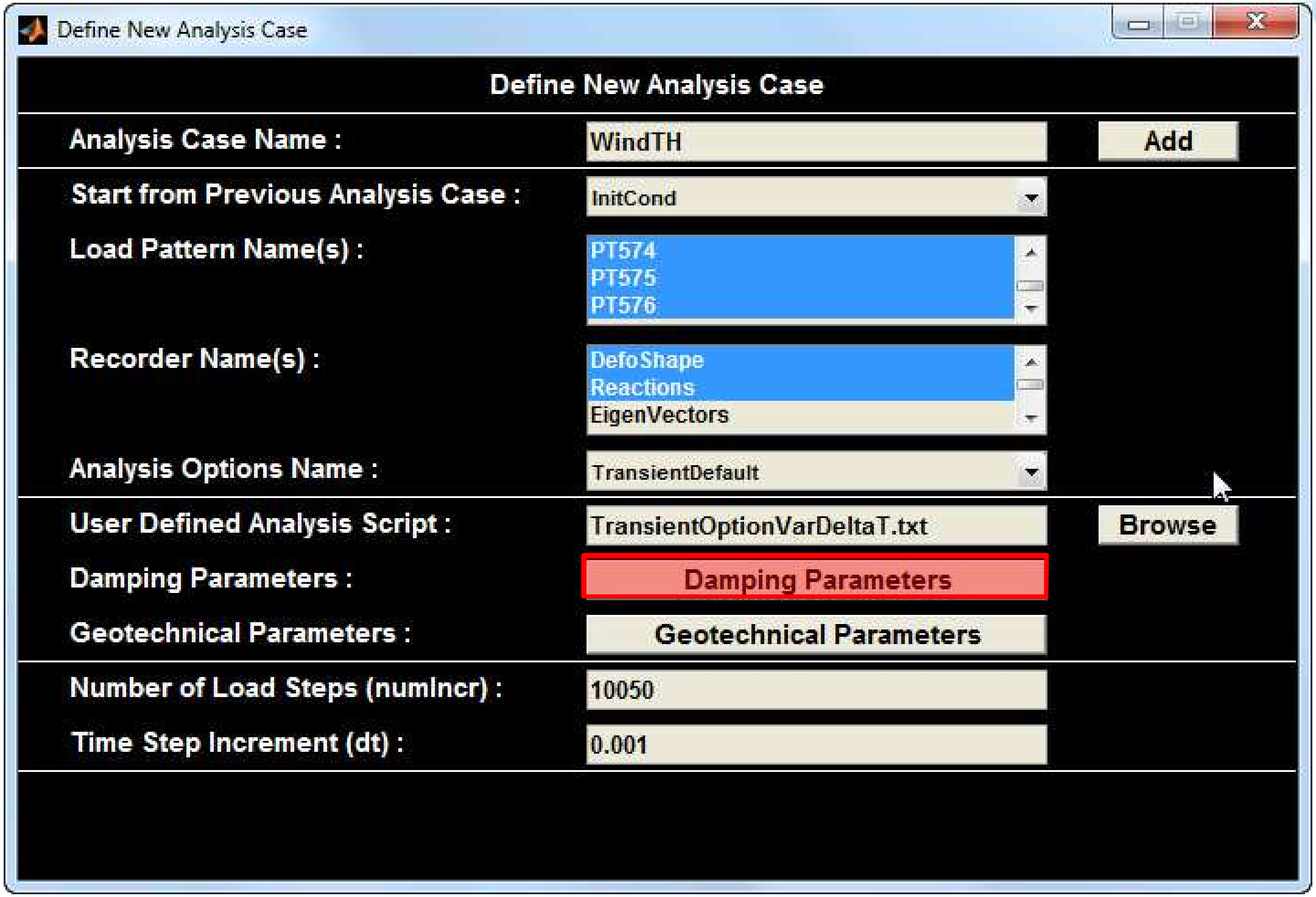
Define Analysis Case: User Defined Script

```
# set default parameters
analysis Transient
set dt 0.001
set dtMin 1.0e-8
set dtMax 0.001

# set the integrator parameters
integrator HHT 0.9

# set the test parameters
set testType NormDispIncr
set testTol 1.0e-12;
set testIter 25;
test $testType $testTol $testIter

# set the algorithm parameters
set algoType Newton
algorithm  $algoType

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

while {$ok == 0 && $tCurrent < $tFinal} {
   puts "$sequ $case $tCurrent"
   set ok [analyze 1 $dt]

   if {$ok != 0} {
      puts " "
      puts [format "Newton failed (time = %1.3e), try ModifiedNewton" $tCurrent]
      algorithm ModifiedNewton
      test $testType $testTol $testIter 0
      set ok [analyze 1 $dt]
      algorithm $algoType
      test $testType $testTol $testIter 0
   }
   if {$ok != 0} {
      puts " "
      puts [format "ModifiedNewton failed (lambda = %1.3e), try RegulaFalsi" [getTime]]
      algorithm NewtonLineSearch -type RegulaFalsi
      test EnergyIncr 1.0e-8 [expr 2 * $testIter] 0
      set ok [analyze 1 $dt]
      algorithm $algoType
      test $testType $testTol $testIter 0
   }
   if {$ok != 0} {
      puts " "
      puts [format "RegulaFalsi failed (lambda = %1.3e), try Bisection" [getTime]]
      algorithm NewtonLineSearch -type Bisection
      test EnergyIncr 1.0e-8 [expr 2 * $testIter] 0
      set ok [analyze 1 $dt]
      algorithm $algoType
      test $testType $testTol $testIter 0
   }
   if {$ok != 0} {
      if {[expr $dt/10.0] >= $dtMin} {
         set dt [expr $dt/10.0]
         puts " "
         puts [format "Reducing time step size (dtNew = %1.6e)" $dt]
         set ok 0
      }
   } else {
      if {[expr $dt*10.0] <= $dtMax} {
         set dt [expr $dt*10.0]
         puts " "
         puts [format "Increasing time step size (dtNew = %1.6e)" $dt]
      }
   }

   set tCurrent [getTime]
}

if {$ok == 0} {
   puts " "
   puts "------------------------------------------------"
   puts [format "AnalysisCase(%i) completed successfully" $case]
   puts "------------------------------------------------"
   puts " "
} else {
   puts " "
   puts "---------------------------------------------------------"
   puts [format "AnalysisCase(%i) failed (time = %1.3e)" $case $tCurrent]
   puts "---------------------------------------------------------"
   puts " "
}
```

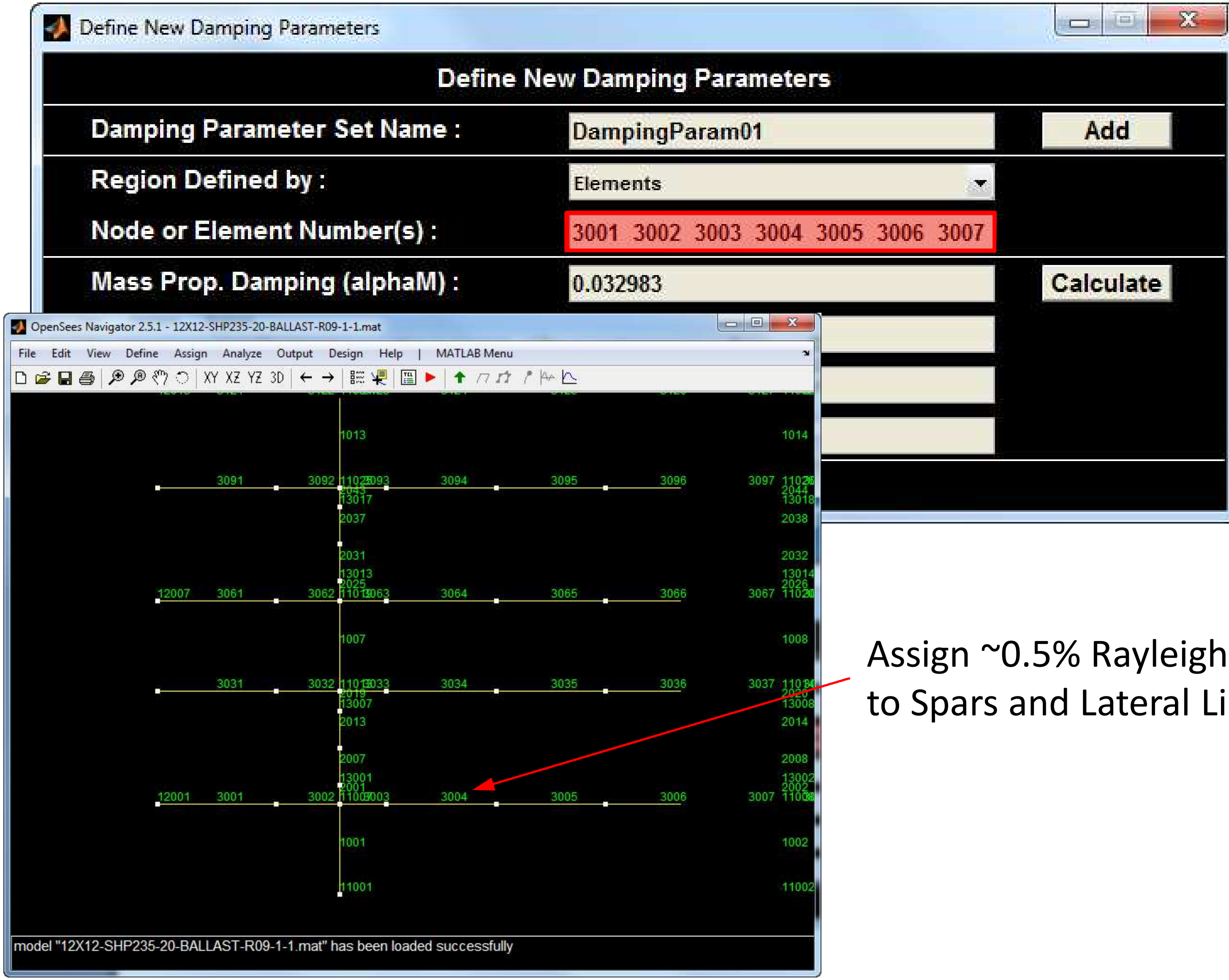
SUNLINK

R+C RUTHERFORD + CHEKENE

Define Analysis Case: Damping

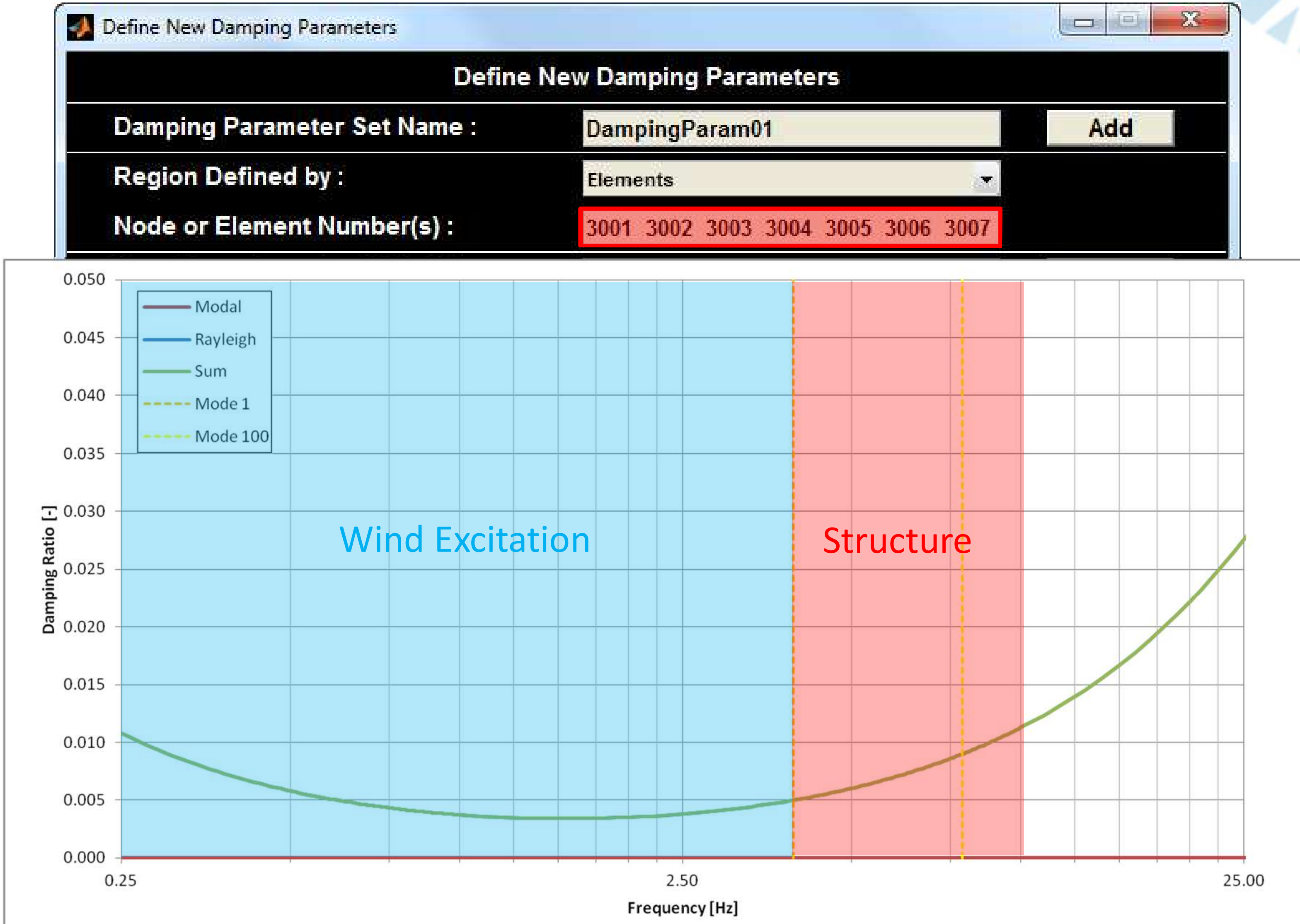


Define Analysis Case: Damping

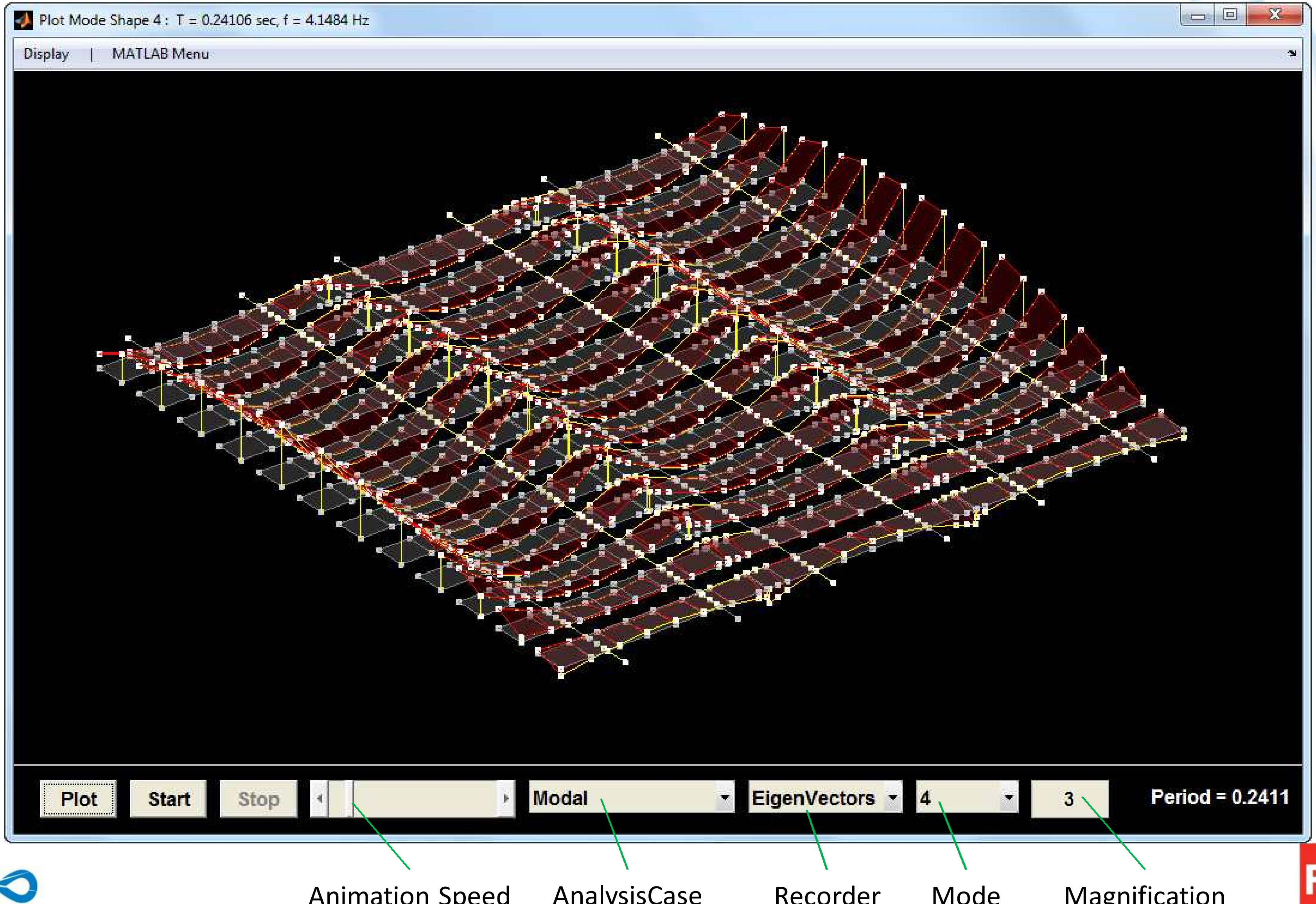


Assign ~0.5% Rayleigh damping to Spars and Lateral Links only

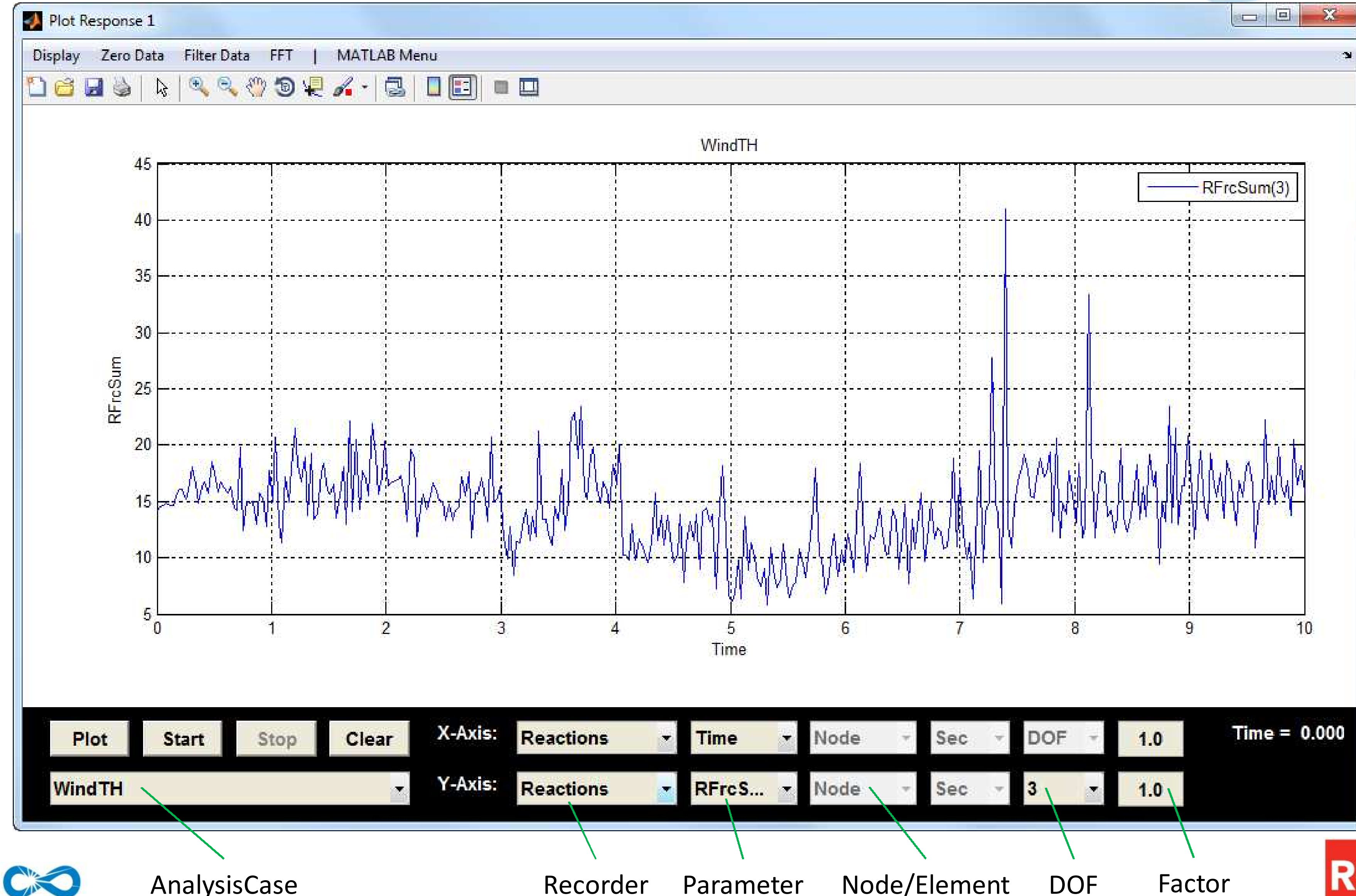
Define Analysis Case: Damping



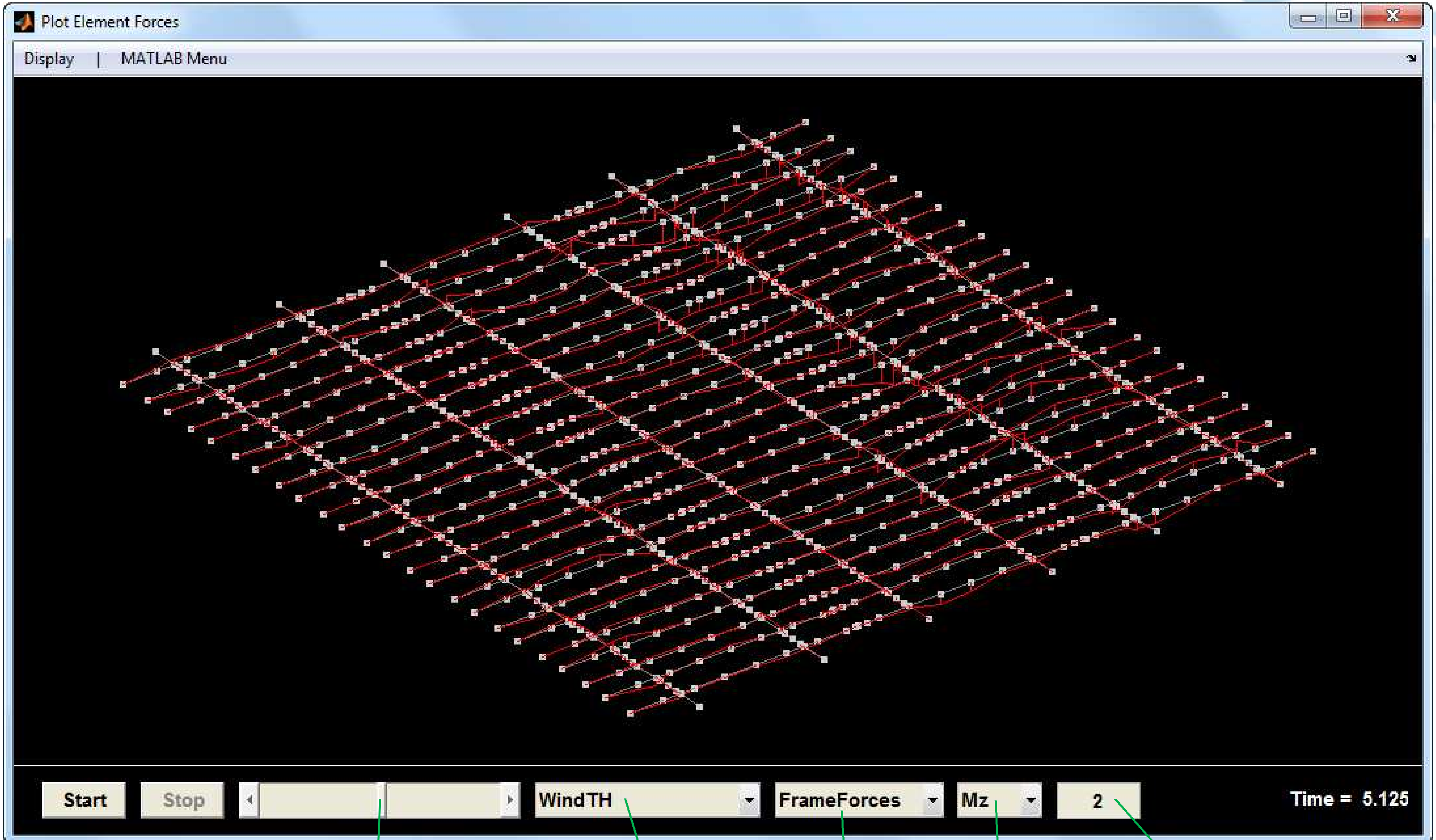
Post Processing: Plot Mode Shapes



Post Processing: Plot Response



Post Processing: Plot Element Forces



Time Step

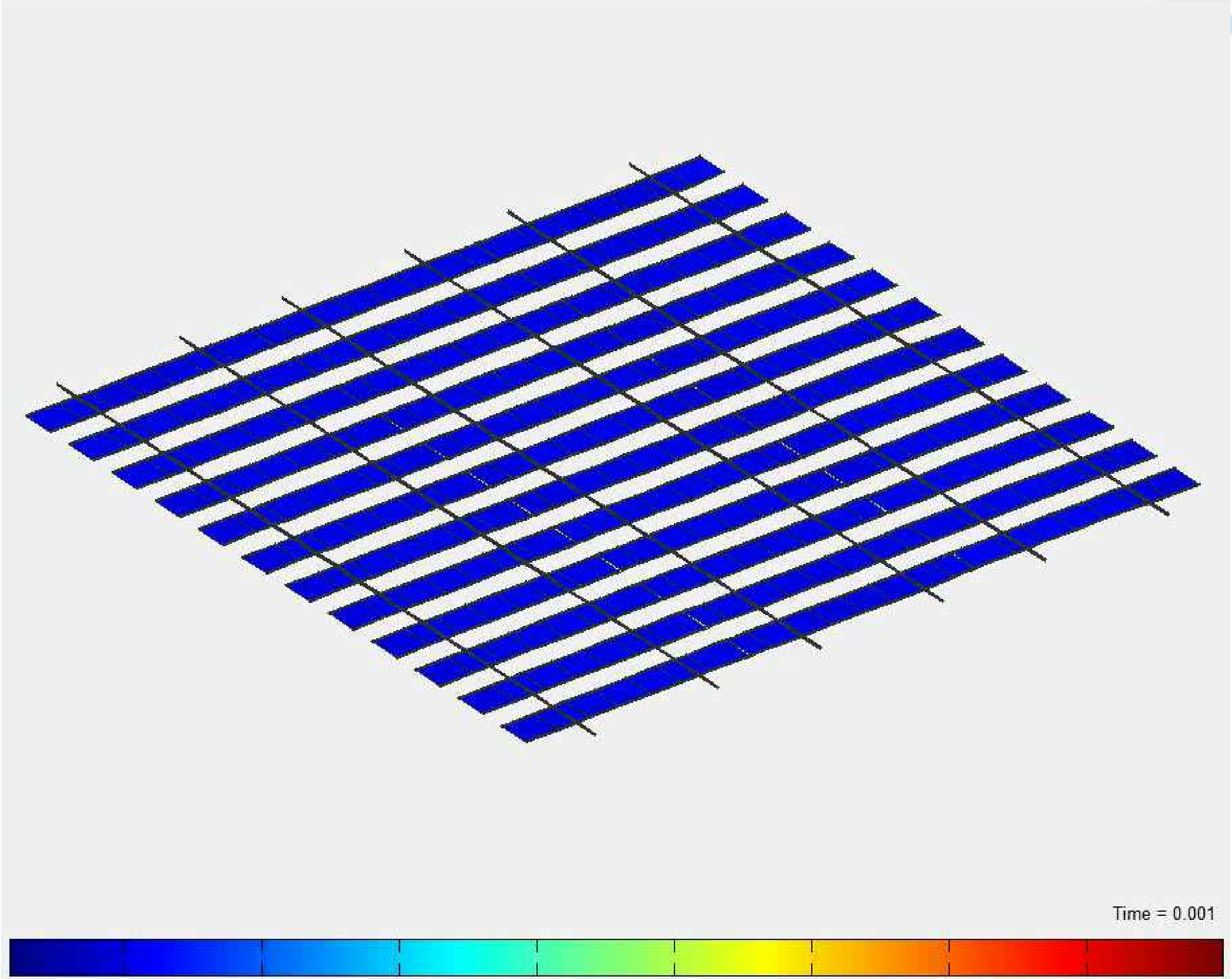
AnalysisCase

Recorder

Response

Magnification

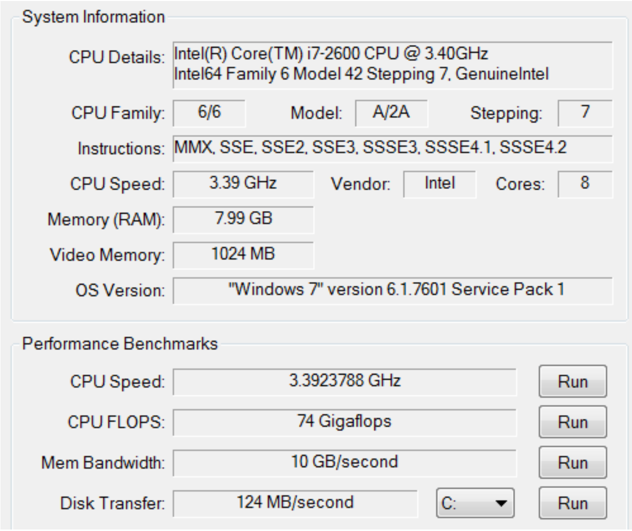
Post Processing: Animate Response



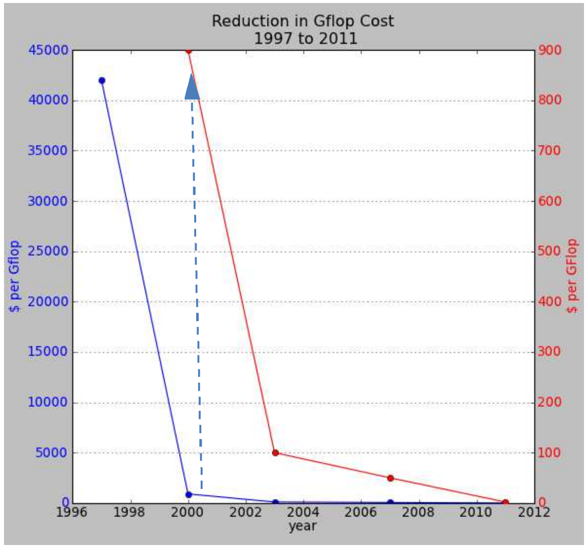
OpenSees Run Times

Static Analysis: about 2-10 minutes
Dynamic Analysis: about 2-10 minutes per second of wind

2011 commodity workstation [1]



Gflop trend over last decade and a half [1]



Another order of magnitude increase in speed would be very helpful for this type of analysis

Summary

- OpenSees ran much faster compared to SAP 2000 and is easy and economical to deploy for simultaneous runs on many PC's or High Performance Computing (HPC) platforms such as Amazon Web Services (AWS).
- OpenSees provides a sufficient range of options for modeling the complex, nonlinear behavior of the arrays.
- The Open Source Komodo IDE is an effective interface for managing files and heterogeneous scripts, and automating execution.
- Navigator fits with MATLAB, used by most researchers to do post-processing and hybrid simulation.
- Coupling OpenSees with Navigator, MATLAB, TCL and Python programs and scripts provides a straightforward way to build interactive and batch execution processes and visualizations for parametric analysis.

An effective OpenSees "cluster" can be a simple collection of networked PC's