



Presenters Rod Holland – SunLink Kate Miller – SunLink Dr. Andreas Schellenberg – R+C





EKENE

I

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

- 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



Alan G. Davenport Wind Engineering Group









California Solar Initiative

CSI R&D Goals

- The overall goal of the RD&D program is to help build a sustainable and selfsupporting 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

Model variations in ballast, connectivity and static load patterns Geometric and material nonlinear elements and hysteretic, Rayleigh and viscous damping

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





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







Procedure Steps – Model and Load Parameters



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







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

	24		r.	ě.	1	}	P	20		ř	
14	+2	+ 16	+4	5	23 +	25 28	- 8	29 + 9	31 + 22	33 + 11	35
+o'' 15	o+ 16 + 17	o+ 18 *	' °+ 20 '	+ 22	-** 0+ 24	+ 43	27 + 44	+ 45	+ 46	47 + 34	
<u>.</u>	38	39	40	53	42	.55	56	57	58	59	
+ 62 63	+ 64 65	o+ 66 67	+ 68	⁵⁹ $^+_{ m o+}$ 70	71 of 72	⁷³ 0+ 74	75 + 76	77 0+ 78	79 ₀ + 80	81 o+ 82	83 0+
+ 00	+ 86	+87	+88	+ 89	+90	+91	+92	93	94+	95+	+
110	98	99	5	101	119	103	123	125 + 105	106	107	131
+o' 10 11	$^{+112}_{+134}$	+135	-0+116	117 + 137		+ 122	+124		128	**0+ 130	1001
	146	147	136	149	138	139	140	141 153	142	143 155	
158 1	596 ⁺ 180 16	16 ⁺ 162 ^{1€}	¹³ 3 ⁺ −164	165+166	167-168	169-170	176+172	173 174	175+176	1770+ 178	178
+	+182	+183	+184	+185	+186	+187	188	189	190	191	
206 2	07 208 209	9, 210 21	1 196	213 1 244	215 1 040	217-199	218 200	221 200	223 - 202	225 200	227
+0	+230	+231	+232	+233	+ 234	+ 235	+236	237+	238+	239+	
	242	243	244	245	246	247	248	249	250	251	
-0254 25	+ + 256 ₂₅₇	·+258 25	o∓260 ²	261 -+ 262	283 - 264	265 + 266	267 - 268	258 + 270	271 + 272	273 + 274	275
+ **	278	+279	280	+281	282	+283	284	285	286	287	
302 3	3 204 30	5 200 30	1 202	309 1 240	311 247	313 014	315 246	317 210	319 200	321 200	323
+	+ 304	+ 300	+ 305	+310	+ 312	+001	+ + 310		- 320	- 322	
	338	339	340	341	342	343	344	345	346	347	
-350 3	516+352 35	3+354 35	5+356	357 + 358	359 360	361-362	363-364	365+366	367-368	369 + 370	371
+	374	375	376	+377	378	379	380	381	382 204	383	-
398 3	99 400 40	1-1 402 40	1 300	405 406	407-1 408	409-410	411 412	413 414	415 416	417 418	419
+	+422	+423	+424	+425	+426	+427	1 428	429	430+	431	+
+	434	435	436	437	438	439	440	441	442	443	
+0 446 447	, o+ 448 ⁴⁴⁹	0+450 45	0+452	103 o+ 454	400 0- 456	457 0+ 458	409 0+460	461 0+462	463 + 464	400+466	46/0-
	470	471	472	473	474	475	476	477 489	478 490	479 491	-
+0494	+ 496 49	70+498 499	+ 500 5	01 + 0+ 502	503 + 504	505 + 0+ 506	507 + 508	509 + 510	511 + 512	513 + 514	515
+ 49	5 + 518	+	1 +520	+521	+522	+523	524	525	526	527	_
+ 640	530	531	532	533	534	535	536	537	538	539	500
+0042 54	3 ⁰⁺ 544	0+548	0+548	-+ 550	01 0+ 552	000 -+ 554	000 0+ 556	557 + 558	009 c+ 560	5010+562	063-
r.	566	567	568	569	570	571	572	573	574	575	





-

Wind Pressure Distribution 4 seconds full scale time, 1/3 speed playback







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

Module -Link Type 1 -Link Type 2 -Link Type 3 -Link Type 4 -Link Type 4 Connection -Link Type 2 Connection -Link Type 4 Ballast Slider -Link Type 3 Slider -

Model Element

elasticBeamColumn as in-plane brace elasticBeamColumn elasticBeamColumn elasticBeamColumn uniaxialMaterial ElasticPPGap fixed flatSliderBearing flatSliderBearing







OpenSees TCL Procedure - File Structure







File Structure - Inputs







File Structure - Outputs







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)
```

"}





TAN

OpenSees TCL Model Interface

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







Wind Load Time Parameter Select Time Slice

1800 seconds

10 seconds



time



time





Wind Load Spatial Parameter

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

50 -	. 982 . 931 . 898 . 847 . 814 . 763 . 730 . 679 . 646	985 934 901 850 817 766 733	987 936 903 852 819 768	• 990 • 939 • 906 • 855 • 822	. 995 . 944 . 911 . 860 . 827	• 997 • 946 • 913	. 999 . 948 . 915	• 1001 • 950	•1006 •955	• 1009 958	.1011	• 1014	-
50 -	.982 .931 .898 .847 .814 .763 .730 .679 .646	985 934 901 850 817 -766 733	. 987 . 936 . 903 . 852 . 819 . 768	990 939 906 855 822	.995 .944 .911 .860 .827	. 997 . 946 . 913	• 999 • 948 • 915	.1001 .950	•1006 •955	• 1009 958	• 1011	. 1014	-
50 -	931 898 847 814 763 730 679 646	934 901 850 817 766 733	936 903 852 819 768	. 939 . 906 . 855 . 822	.944 .911 .860 .827	.946 .913	.948 .915	. 950	. 955	95.8	000	053	-
	. 898 . 847 . 814 . 763 . 730 . 679 . 646	.901 .850 .817 .766 .733	.903 .852 .819 .768	.906 .855 .822	. 911 . 860 . 827	• 913	• 915	017			. 900	. 90.5	
	.847 .814 .763 .730 .679 .646	850 817 766 733	.852 .819 .768	. 855 . 822	. 860 . 827	962		• 917	. 922	. 925	• 927	. 930	
	.763 .730 .679 .646	766 733	.768			. 829	. 864	. 866	. 871	. 874	. 876 . 843	. 879 - 846	
	.763 .730 .679 .646	.766 .733	.768		1000	•	•	•		•	•	•	
40	.679 .646		. 735	• 771 • 738	.743	.745	747	.782	.754	.790	.759	.762	-
	. 646	. 682	. 684	. 687	. 692	. 694	. 696	. 698	.703	.706	• 708	•711	
GL (10		. 649	. 651	• 654	. 659	. 661	. 663	. 665	. 670	. 673	. 675	. 678	
lie	. 595	. 598	. 600	. 603	. 608	. 610	. 612	. 614	. 619	. 622	. 624	. 627	
ate	. 502	• 305	. 307	. 570	. 575	• 5/7	• 3/9	• 201	• 300	. 369	• 291	• 394	
30	. 511	. 514	. 516	. 519	. 524	. 526	. 528	. 530	. 535	. 538	. 540	. 543	-
00	• 4/0	. 401	. 405	. 400		• • • • •	. 435	• • • • •		. 505			
	. 427	· 430	. 432	. 435	. 440	• 442	• 444	. 446	. 451	. 454	. 456	• 459 436	
	• 334	• 337	• 333	. 402	. 407	. 405	• ***	. 415	. 410	. 421	. 42.5	• 420	
	. 343	. 346	· 348	. 351	· 356	. 358	· 360	· 362	· 367	· 370	. 372	· 375	
20 -							• ***					•	-
	. 259	. 262	. 264	. 267	• 272	. 274	. 276	. 278	. 283	. 286	. 288	. 291	
	. 220	. 22.5	• 251	• 2.34	• 235	. 241	. 2.4.5	• 245	• 250	• 235	• 235	• 250	
	• 175	.178	• 180	· 183	. 188	• 190	• 192	• 194	• 199	. 202	. 204	. 207	
	• 142	• 145	• 147	. 150	11105	11106	•155	. 101	.100	. 105	• 1/1	• 1/4	
10	. 91	• 94	• 96	. 99	. 104	. 106	.108	• 110	.115	.118	. 120	• 123	-
	. 36	. 01		. 00	11205	11206	• ' 3	• * *	. 02	. 0.5	. 6/	. 50	
	•7	• 10	• 12	• 15	. 20	• 22	• 24	• 26	• 31	• 34	• 36	• 39	
0		10	2	0	30		40	5	0	60		70	80





1

Wind Load Cp Spatial Envelope



EWA





T

Ballast and Link 2 Geometry







715

Ballast and Link 3,4 Geometry







-

Wind Load Nodal Forces Snapshot Static Load Pattern (2x2 Peak)

60				group p	attern: : step	snap2x2): 28682	V [n GCp	nph]: 60. min: -0.	.0 γ _w 65	: 1.0			
												•	patch uplift
	•14.2 0.3	• 12.8 0.1	• 13.2 0.2	0 -1.0 -0.0	• 6.3 0.1	• 9.7 0.1	•1.9 0.0	• 4.7	•7.4	●- <u>3.0</u> -0.0	• 7.4	• 3.2	
50	14.2 22.4 0.5	12.8 22.3 0.3	13.2 8.4 0.1	•-1.0 • 5.4 0.1	6.3 0.2 0.0	9.7 3.1 0.0	.1.9 0.6 0.0	4.7 9.0 0.1	7.4 4.5 0.1	●-3.0 ●-1.6 -0.0	• 7.4 • 2.6 0.0	•3.2 •1.0 •0.0	
	• 22.4 • 5.2	22.3	• 8.4 • 7.3	• 5.4 • 1.4	0.2 4.0	3.1 10.7	•0.6 •-3.4	9.0 6.9	4.5 6.3	•1.6 •0.4	• 2.6 • -0.5	•-1.0 •5.5	
40	5.2 9.8	1.9	7.3	1.4 10.0	4.0 9.2	10.7 11.3	• ^{-3.4}	•6.9 •2.5	6.3 2.3	0.4	0-0.5 -1.3	5.5 15.6	
	9.8 8.2 0.2	0.1 8.4 5.6 0.1	0.0 3.5 11.7 0.1	0.1 10.0 10.1 0.1	9.2 17.6 0.2	0.1 11.3 19.5 0.2	0.0 0.2 0-0.9 -0.0	-0.0 -2.5 -1.2 0.0	•2.3 •0.5 •0.0	• 3.9 • -2.2 • -0.0	-0.0 -1.3 •1.9 0.0	0.3 15.6 5.2 0.1	
	8.2 4.5 0.1	5.6 16.2 0.2	11.7 13.5 0.2	10.1 8.8 0.1	17.6 17.4 0.2	19.5 11.4 0.1	0-0.9 -5.8 -0.1	•1.2 •-0.5 -0.0	0-0.5 0-1.3 -0.0	●-2.2 ●-5.8 -0.1	• 1.9 • 0.8 • 0.0	5.2 5.9 0.1	
30	4.5 4.2 0.1	16.2 4.7 0.1	13.5 12.0 0.1	8.8 7.7 0.1	•17.4 •-7.3 •0.1	•11.4 •21.2 •0.2	•5.8 •10.6 •0.1	0.5 0.8 0.0	•-1.3 •-4.3 •-0.0	●-5.8 ●-6.0 -0.1	•0.8 •-10.5 -0.1	5.9 14 0.0	
	•4.2 •8.8 •0.2	4.7 6.4 0.1	•12.0 •-3.5 •0.0	•7.7 •7.5 •0.1	-7.3 -13.6 -0.2	•-21.2 •-5.1 -0.1	• 10.6 • 9.5 • 0.1	0.8 12.3 0.1	•4.3 •2.6 0.0	●-6.0 ●-3.8 -0.0	•-10.5 •5.2 •1	•14 •11 0.0	
20	• ^{-8.8} 3.5	• 6.4 • 6.5	-3.5 -5.0	•-7.5 • 5.0	0-13.6 4.6	•5.1 •8.1	9.5 10.8	12.3 21.6	2.6 5.3	•3.8 •3.4	• 5.2 • -2.5	11	
	3.5 13.1 0.3	•-6.5 • 2.5 • 0.0	-5.0 -12.5 -0.1	• 5.0 • -3.1 • 0.0	4.6 5.8 0.1	8.1 25.9 0.3	10.8 7.1 0.1	21.6 2.1 0.0	•5.3 •-2.6 •0.0	.3.4 2.5 0.0	-2.5 -5.4 -0.1	11 2.4 0.1	
	13.1 10.9 0.2	• 2.5 •-14.7 •0.2	•-12.5 •-31.5 •0.4	●-3.1 ●-68.1 -0.8	• 5.8 • -91,1 -1.0	• 25.9 • 45.2 • 0.5	•7.1 •-34.4 •0.4	•2.1 •-11.9 •0.1	-2.6 -17.0 -0.2	•2.5 •4.1 -0.0	●-5.4 ●-6.4 -0.1	2.4 0.0 0.0	
10	10.9 13.1 0.3 13.1	•14.7 •1.9 0.0 •1.9	•-31.5 •-8.5 •0.1 •-8.5	•-68.1 •-20.3 •0.2 •-20.3	-91 1 -30.1 -0.3 -30.1	-45.2 -58.8 -0.7 -58.8	•-34.4 •-34.4 •-34.4 •-34.4	-11.9 -27.9 -0.3 -27.9	•17.0 •8.4 0.1 •8.4	•4.1 •11.4 •11.4 •11.4	•-6.4 •-5.1 •0.1 •-5.1	0.0 4.0 0.1 4.0	
											54		
0		10	2	0	30	X Coo	40 dinate (fe	5 eet)	0	60		70	1





ZAN

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







ZAN

Array Model Displacements Stiffer, Less Ballast, 10 seconds real time, ½ speed playback







ZAN

OpenSees Navigator Procedures



model "12X12-SHP235-20-BALLAST-R08A-1-1.mat" has been loaded successfully

RUTHERFORI

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.





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
- 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

Control Provide Addree Ad	
If the East View Define Auge Delay to Up the Delay Height Park Piece D @ @ @ P P ?? D XY XZ YZ 30 ← → E # [] > + ?? ?? Piece Image: Auge Define Auge Delay Height Park Piece Image: Auge Define Auge Delay Height Park Piece Image: Auge Define Auge Delay Height Park Piece Image: Auge Delay Height Park P	OpenSees Navigator 2.5.1 - 12X12-SHP235-20-BALLAST-R08A-1-1.mat
Deele (PPRO) WZYZ) + → E#IE> + 747 PES	File Edit View Define Assign Analyze Output Design Help MATLAB Menu 🏻
OpenSees Navigator Model	□ ☞ 🖬 吾 ୭ ୭ १७ ○ XY XZ YZ 3D ← → 註 埂 🖽 🕨 ↑ / フ カ パ № 🗠
OpenSees Navigator Model	
model "12X12_SHP235_20_RALLAST_R08A_1_1 mat" has been loaded successfully	OpenSees Navigator Model



SunLink System Components



OpenSees Model Components







Define Element: Line Element

A Materials > D ← → IIII ↓		
Define HingeBeamColumn Element		
Define H	lingeBeamColumn Element	
Element Name :	LINKstart	Add
Section Type Node i (secTagl) :	BOLTConn	
Hinge Length Node i (Lpi) :	1	🔤 relative
Section Type Node j (secTagJ) :	LINK	
Hinge Length Node j (Lpj) :	1	relative
Modulus of Elasticity (E) :	10000	Database
Shear Modulus (G) :	3759.3985	
Cross-Sectional Area (A) :	0.63205	
Torsional Moment of Inertia (J) :	0.19032	
Moment of Inertia (ly) :	0.12578	
Moment of Inertia (Iz) :	0.13948	
Optional Arguments :		
Mass Density (massDens) :	1.5961e-07	
Maximum Iterations (maxIters) :	10	
Tolerance (tol) :	1e-12	





Modeling the Bolted Connection

Define Steelo1 Material		
D	efine 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		
Defi	ne ElasticPPGap Material	
Material Name :	BoltGapPos	Add
Material Name : Modulus of Elasticity (E) :	BoltGapPos	Add
Material Name : Modulus of Elasticity (E) : Yield Stress (Fy) :	BoltGapPos	Add
Material Name : Modulus of Elasticity (E) : Yield Stress (Fy) : Initial Gap (gap) :	BoltGapPos	Add
Material Name : Modulus of Elasticity (E) : Yield Stress (Fy) : Initial Gap (gap) : Optional Parameters :	BoltGapPos	Add
Material Name : Modulus of Elasticity (E) : Yield Stress (Fy) : Initial Gap (gap) : <i>Optional Parameters :</i> Hardening Ratio (eta) :	BoltGapPos	Add
Material Name : Modulus of Elasticity (E) : Yield Stress (Fy) : Initial Gap (gap) : Optional Parameters : Hardening Ratio (eta) : Damage Switch (damage) :	BoltGapPos 10000 10000 0.05 0.99999 off	Add







	Define	Aggregator Section		
Section Name :		BOLTConn		Add
Select Materials :	P :	BoltConnPx	-	
	Vy:	None	•	
	Vz :	None	•	
	т:	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

Defir	e New Analysis Case		
Analysis Case Name :	WindTH		Add
Start from Previous Analysis Case :	InitCond	-	
Load Pattern Name(s) :	PT574	*	
	PT575 PT576	-	
Recorder Name(s) :	DefoShape	*	
	Reactions EigenVectors	-	
Analysis Options Name :	TransientDefault	•	N
User Defined Analysis Script :	TransientOptionVarDeltaT.txt		Browse
Damping Parameters :	Damping Parameters		
Geotechnical Parameters :	Geotechnical Paramete	rs	
Number of Load Steps (numincr) :	10050		
Time Step Increment (dt) :	0.001		





Define Analysis Case: User Defined Script

```
# set default parameters
                                                                                       if {$ok != 0} {
analysis Transient
                                                                                          puts " "
set dt 0.001
                                                                                          puts [format "RegulaFalsi failed (lambda = %1.3e), try Bisection" [getTime]]
set dtMin 1.0e-8
                                                                                          algorithm NewtonLineSearch -type Bisection
set dtMax 0.001
                                                                                          test EnergyIncr 1.0e-8 [expr 2 * StestIter] 0
                                                                                          set ok [analyze 1 $dt]
# set the integrator parameters
                                                                                          algorithm $algoType
integrator HHT 0.9
                                                                                          test $testType $testTol $testIter 0
# set the test parameters
                                                                                       if {$ok != 0} {
set testType NormDispIncr
                                                                                         if {[expr $dt/10.0] >= $dtMin} {
set testTol 1.0e-12;
                                                                                            set dt [expr $dt/10.0]
set testIter 25;
                                                                                            puts " "
test $testType $testTol $testIter
                                                                                            puts [format "Reducing time step size (dtNew = %1.6e)" $dt]
                                                                                            set ok 0
# set the algorithm parameters
                                                                                         }
set algoType Newton
                                                                                       } else {
algorithm $algoType
                                                                                         if {[expr $dt*10.0] <= $dtMax} {</pre>
                                                                                            set dt [expr $dt*10.0]
set ok 0;
                                                                                            puts " "
set tFinal [expr $numSteps * $dt]
                                                                                            puts [format "Increasing time step size (dtNew = %1.6e)" $dt]
set tCurrent [getTime]
                                                                                         }
                                                                                       ٦
while {$ok == 0 && $tCurrent < $tFinal} {</pre>
  puts "$sequ $case $tCurrent"
                                                                                       set tCurrent [getTime]
  set ok [analyze 1 $dt]
  if {Sok != 0} {
                                                                                    if {$ok == 0} {
     puts " "
                                                                                       puts " "
     puts [format "Newton failed (time = %1.3e), try ModifiedNewton" $tCurrent]
                                                                                       puts "-----
     algorithm ModifiedNewton
                                                                                       puts [format "AnalysisCase(%i) completed successfully" $case]
     test $testType $testTol $testIter 0
                                                                                       puts "-----"
     set ok [analyze 1 $dt]
                                                                                       puts " "
     algorithm $algoType
                                                                                    } else {
     test $testType $testTol $testIter 0
                                                                                       puts " "
  3
                                                                                       puts "-----
  if {$ok != 0} {
                                                                                       puts [format "AnalysisCase(%i) failed (time = %1.3e)" $case $tCurrent]
     puts " "
                                                                                       puts "------'
     puts [format "ModifiedNewton failed (lambda = $1.3e), try RegulaFalsi" [getTime]]
                                                                                       puts " "
     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
```

SUNLINK



Define Analysis Case: Damping

Defir	e New Analysis Case		
Analysis Case Name :	WindTH		Add
Start from Previous Analysis Case :	InitCond	-	
Load Pattern Name(s) :	PT574	*	
	P1576		
Recorder Name(s) :	DefoShape	<u>^</u>	
	EigenVectors	-	
Analysis Options Name :	TransientDefault	-	
User Defined Analysis Script :	TransientOptionVarDeltaT.txt		Browse
Damping Parameters :	Damping Parameters	S.	
Geotechnical Parameters :	Geotechnical Paramete	ers	
Number of Load Steps (numincr) :	10050		
Time Step Increment (dt) :	0.001		





Define Analysis Case: Damping



Define Analysis Case: Damping



SUNLINK



Post Processing: Plot Mode Shapes



Post Processing: Plot Response



Post Processing: Plot Element Forces



Post Processing: Animate Response



SUN



OpenSees Run Times

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

2011 commodity workstation ⁽¹⁾

System Information							
CPU Details:	Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz Intel64 Family 6 Model 42 Stepping 7. GenuineIntel						
CPU Family:	6/6 Model: A/2A Stepping: 7						
Instructions:	MMX, SSE, SSE2, SSE3, SSSE3, SSSE4.1, SSSE4.2						
CPU Speed:	3.39 GHz Vendor: Intel Cores: 8						
Memory (RAM):	7.99 GB						
Video Memory:	1024 MB						
OS Version:	"Windows 7" version 6.1.7601 Service Pack 1						
Performance Benchmarks							
CPU Speed:	3.3923788 GHz Run						
CPU FLOPS:	74 Gigaflops Run						
Mem Bandwidth:	10 GB/second Run						
Disk Transfer:	124 MB/second C: ▼ Run						

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

715

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

