Collapse Simulation of RC Frame Buildings

Curt Haselton, PhD Candidate, PE, Stanford (CSUC 1/18/07)

Abbie B. Liel, PhD Candidate, Stanford

Gregory Deierlein, Professor, Stanford, PhD, PE

OpenSees Developers Symposium – August 16, 2006

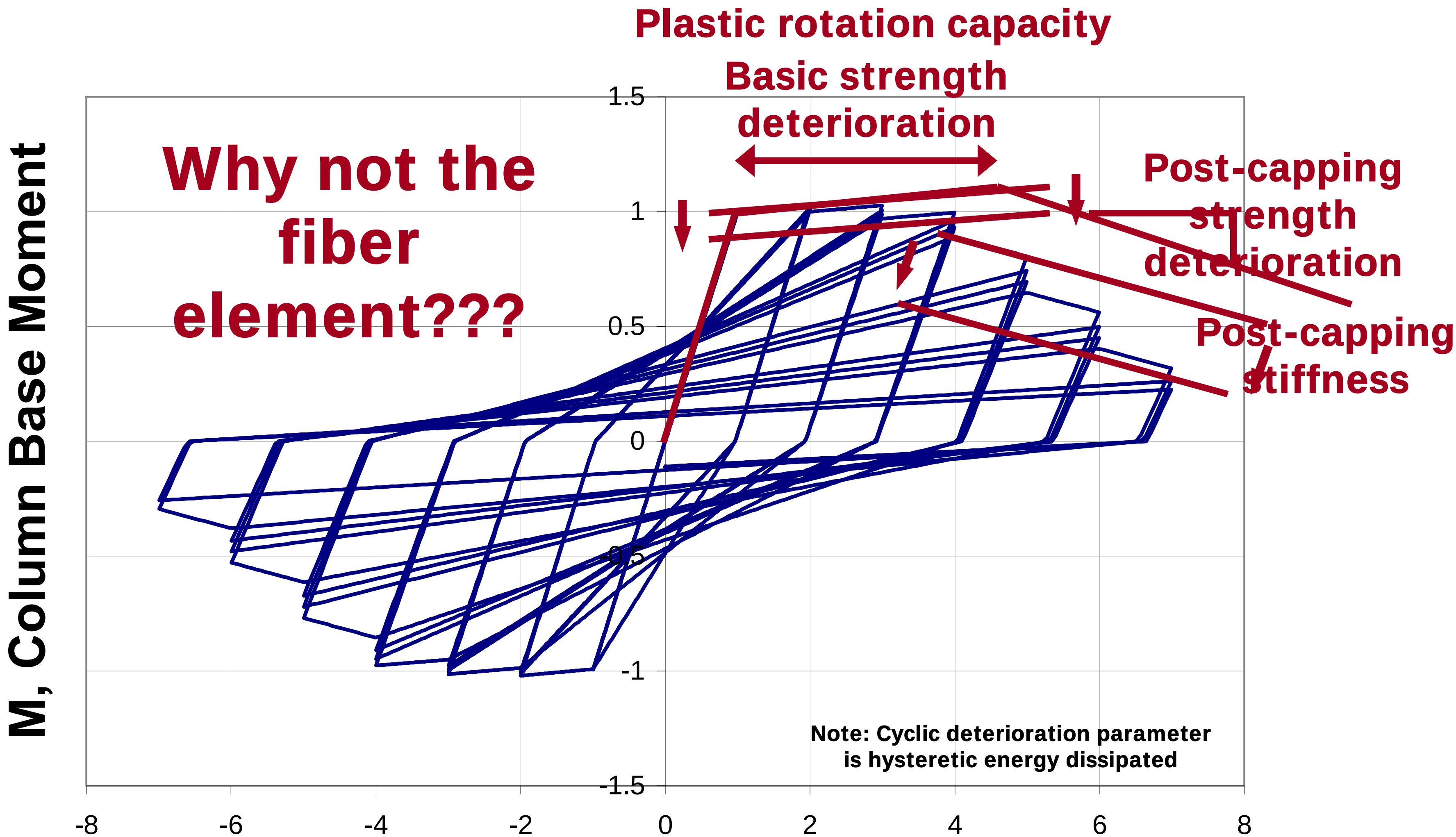
Outline

- Research Overview
  - Purpose
  - Reinforced-Concrete Beam-Column element model
  - Structural model and collapse prediction method
- Collapse Analysis Method and Tools:
  - Calibration of RC Beam-Column element model
  - Program for Automated OpenSees Model Generation
  - Matlab algorithms
- Numerical Considerations:
  - Solution algorithm
  - Treatment of singularity and non-convergence
  - Problem with sparse mass matrix
- Summary

Purpose of Research

- Predict the collapse safety of new code-conforming RC frame buildings (60 buildings designed by 2003 IBC)
- Use these predictions to inform design and Code provisions
- Develop the data and tools necessary for:
  - the collapse predictions to be reliable/defensible
  - others to be able to do collapse analysis with reasonable effort

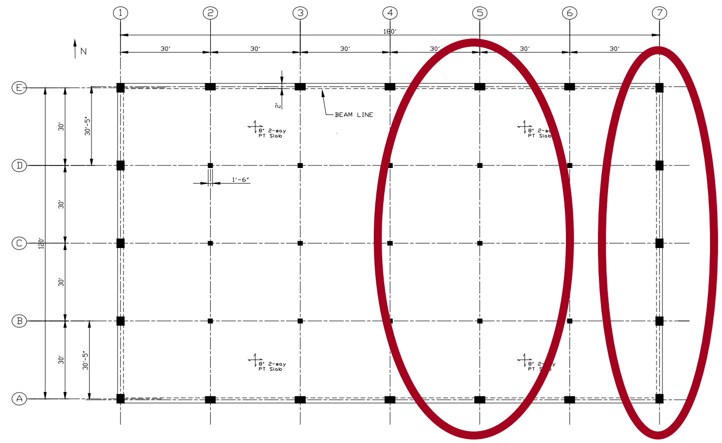
RC Beam-Column Element Model

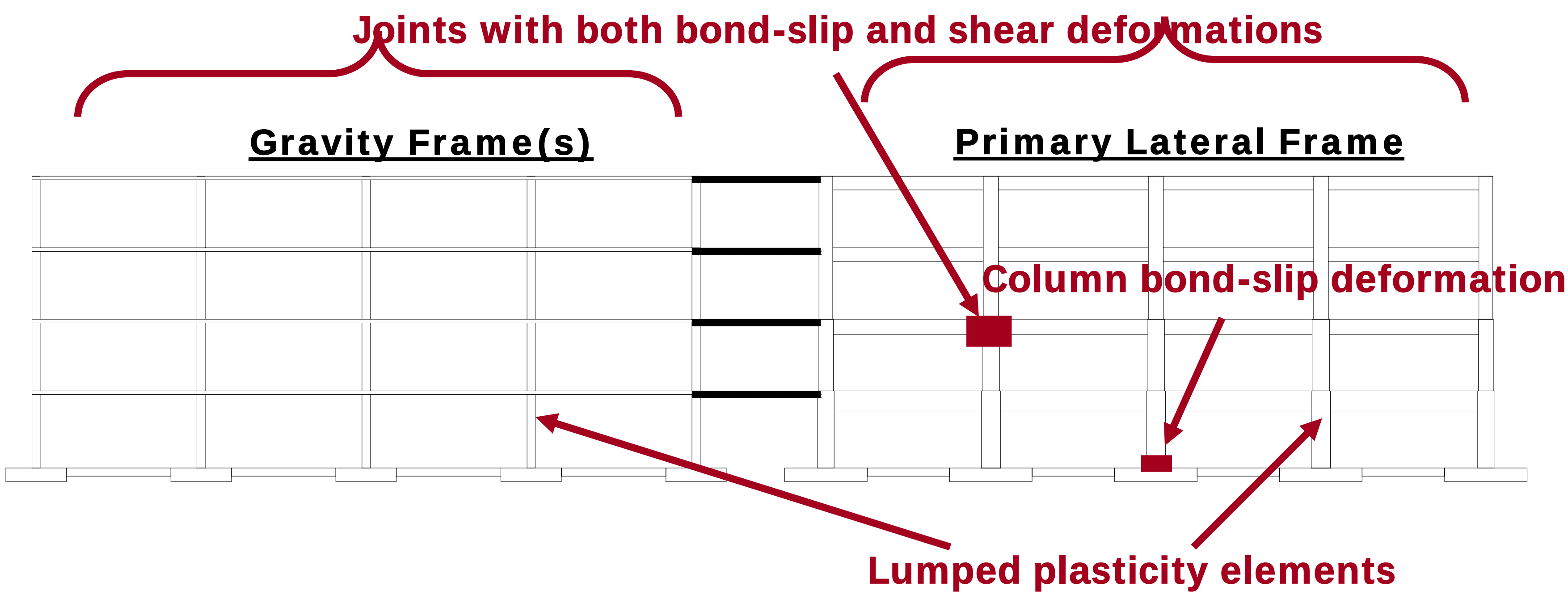


(Material model by Ibarra, Medina, and Krawinkler)

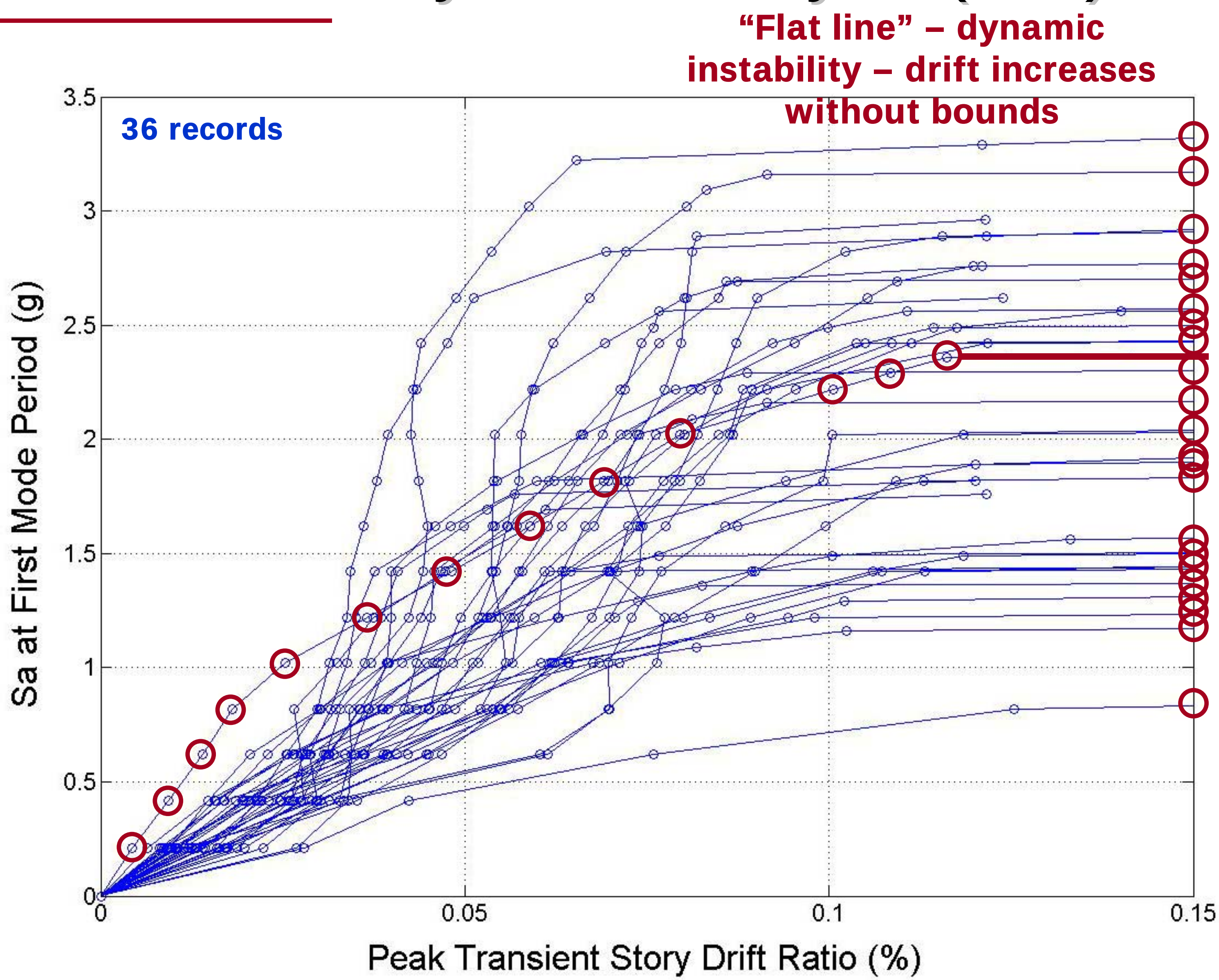
Θ, Chord Rotation

Structural Model





Incremental Dynamic Analysis (IDA)



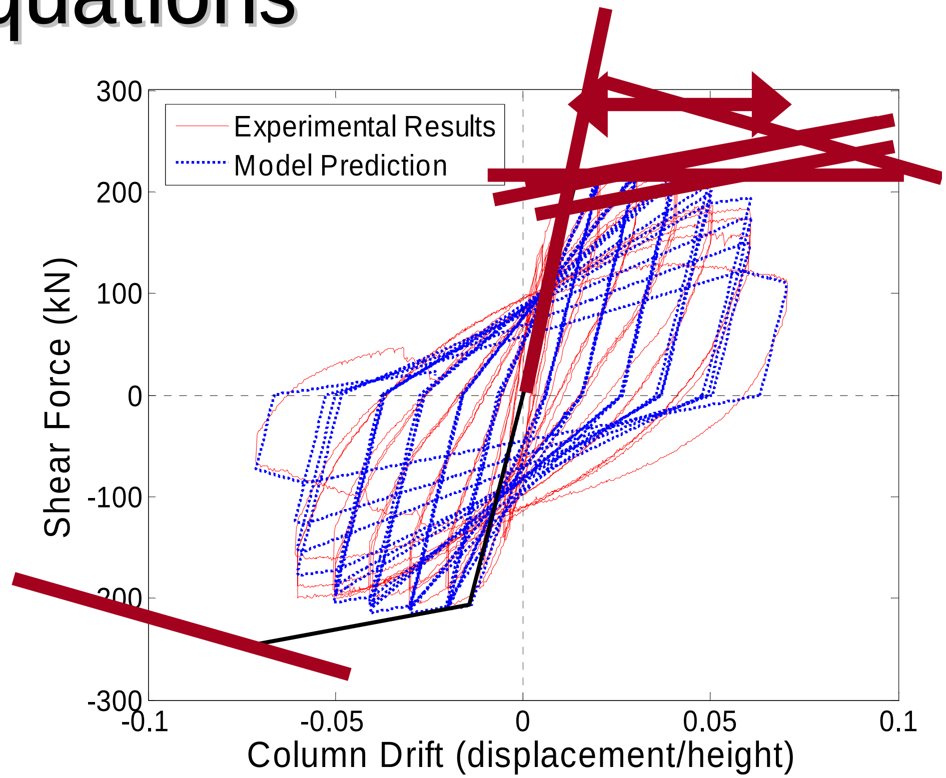
Outline

- Research Overview
  - Purpose
  - Reinforced-Concrete Beam-Column element model
  - Structural model and collapse prediction method
- **Collapse Analysis Method and Tools:**
  - **Calibration of RC Beam-Column element model**
  - **Excel Program for Automated OpenSees Model Generation**
  - **Matlab algorithms**
- Numerical Considerations:
  - Solution algorithm
  - Treatment of singularity and non-convergence
  - Problem with sparse mass matrix
- Summary

Empirical Predictive Equations

Model calibrated to 255 flexurally dominated test from PEER Structural Performance Database (Berry and Eberhard)

Calibration done using a "brute force" approach; Lignos and Krawinkler are developing a more automated approach.



Model Parameters to be Predicted:

- **Strength (easiest)**
- **Initial stiffness**
- **Post-yield stiffness**
- **Plastic rotation capacity**
- **Negative post-cap slope**
- **Cyclic deterioration rate**

Acknowledgements:

- **Sarah Taylor Lange**
- **Abbie Liel**

$$\theta_{cap,pl} = 0.12\left(1+0.55a_{sl}\right)\left(0.16\right)^{\nu}\left(0.02+40\rho_{sh}\right)^{0.43}\left(0.54\right)^{0.01c_{units}f'_c}\left(0.66\right)^{0.1s_n}\left(2.27\right)^{10.0\rho}$$

[Mean and uncertainty both quantified]

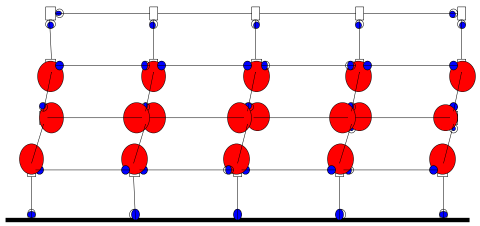
Automated OpenSees Model Generation

ETABS
- Stiffness design

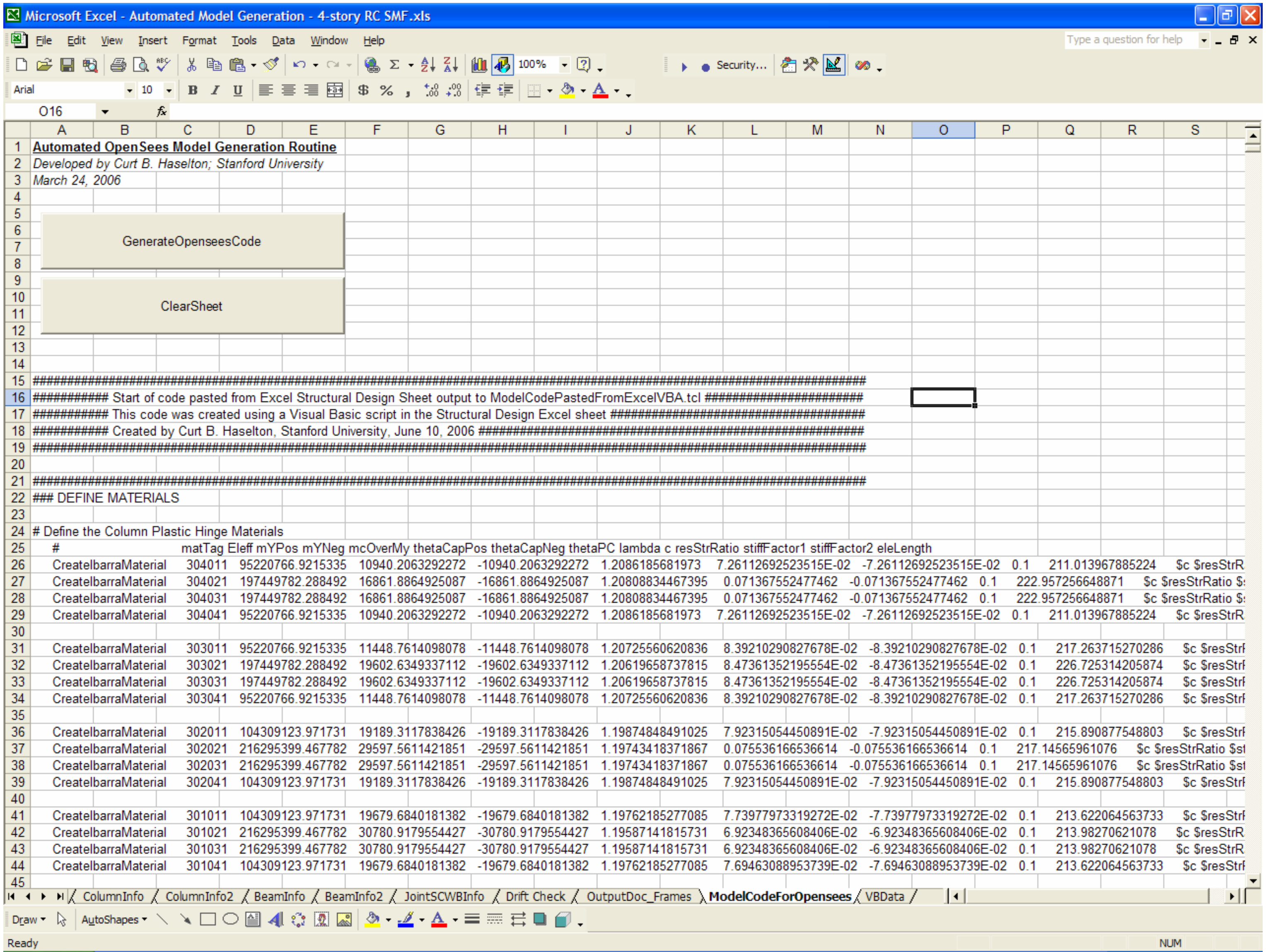
Excel
- All design other than stiffness (2003 IBC)
- Generation of OpenSees model code
- Generation of Matlab code for collapse visualization

Matlab / OpenSees
- Do analysis

Matlab
- Post-process and do collapse visualization



Automated OpenSees Model Generation



Overview of Matlab Collapse Algorithm

- Matlab drives the collapse analysis
  - Automated: Matlab runs analysis, does post-processing, and runs parameter studies when needed
  - Matlab writes a .Tcl file to tell OpenSees what to do, then Matlab executes OpenSees with the following command:
    - “!openSees RunSingleCollapseSensitivityAnalysisMATLAB.tcl”
- After OpenSees analysis, Matlab processes the results, then runs remaining analyses to find collapse point
- With Matlab, it is easier to handle numerical difficulties such as singularity and non-convergence

Outline

- Research Overview
  - Purpose
  - Reinforced-Concrete Beam-Column element model
  - Structural model and collapse prediction method
- Collapse Analysis Method and Tools:
  - Calibration of RC Beam-Column element model
  - Excel Program for Automated OpenSees Model Generation
  - Matlab algorithms
- **Numerical Considerations:**
  - **Dynamic solution algorithm**
  - **Treatment of singularity**
  - **Problem with sparse mass matrix**
- **Summary**

Dynamic Solution Algorithm

- The solution algorithm was a large part of this work and we spent a lot of time refining it.
- "Try Everything" Approach (more details at end of presentation):
  - If Non-Converged:
    - Try all **solution algorithms** (NewtonLineSearch, Newton, Newton –initial, Krylov Newton)
  - If Still Non-Converged:
    - Repeatedly reduce **time step**
  - If Still Non-Converged:
    - Repeatedly reduce **tolerance**
  - If Still Non-Converged:
    - STOP ANALYSIS and report non-convergence (non-convergence does not mean collapse!!!!)
  - Else
    - Output the convergence **tolerance achieved** (user decides if acceptable)
    - Check for **collapse** (based on an interstory drift limit)
    - Check for **singularity**

Treatment of Singularity

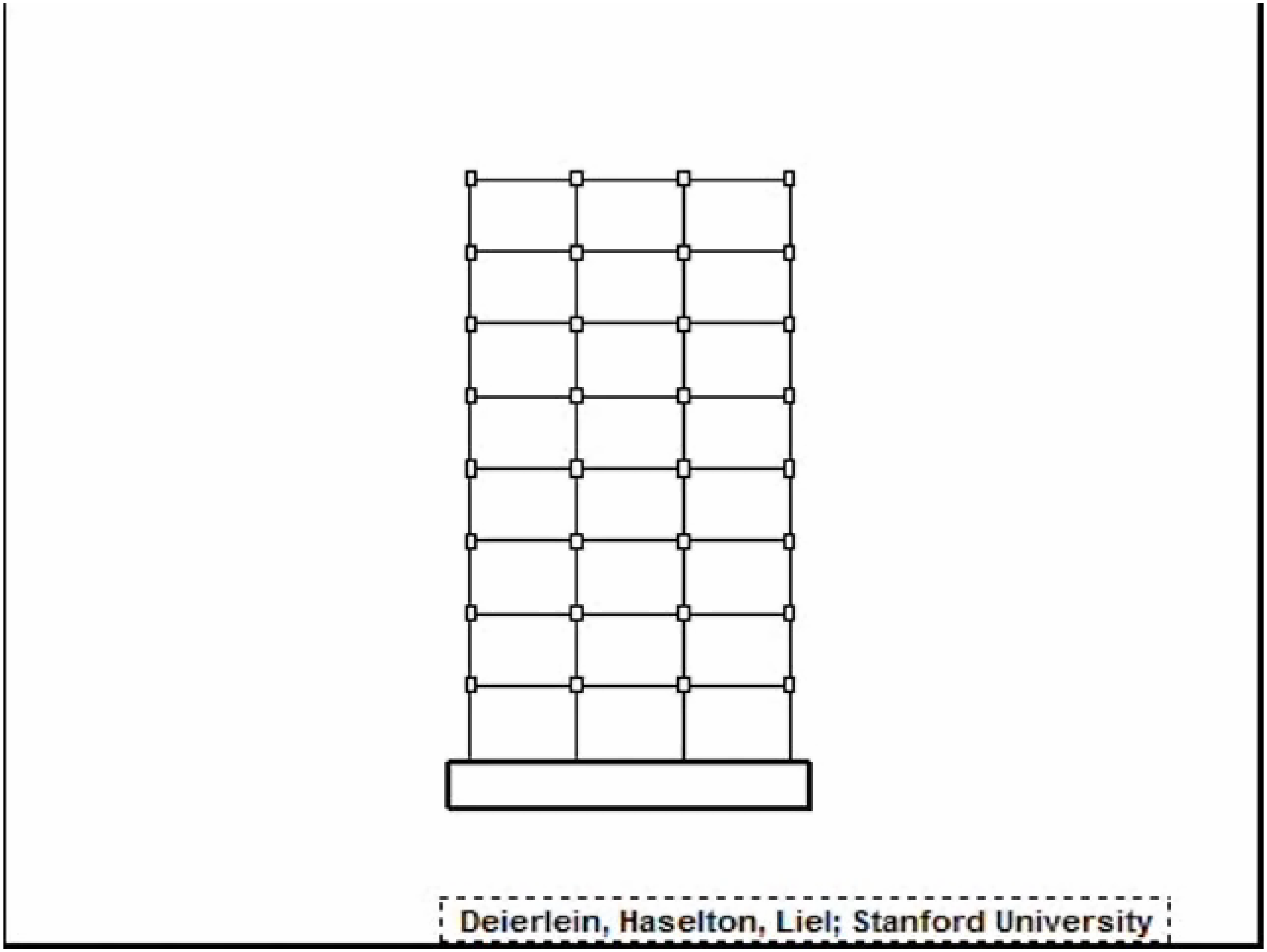
- Singularity is a huge problem for collapse analysis. Singularity causes the current version of opensees.exe to have a runtime error (we need to fix this).
- How do we know that the solution went singular? We get any of the following in *some* (not all) of our nodal recorders:
  - #QNAN, or #IND, or a huge displacement (disp > 1000000.0)
- Code to check for singularity: See reference slides at end of presentation
- What do we do if the solution goes singular?
  - OpenSees stops analysis (to avoid run-time error) and reports the singularity back to Matlab
  - Matlab throws away the results from the singular analysis
  - Matlab then *slightly* alters the Sa level of the earthquake and reruns the analysis
  - Matlab continues these slight alterations to Sa until we get an analysis with either full convergence or collapse (without any singularity)

Problem with Sparse Mass Matrix

- When doing collapse analyses, the dynamic “stiffness” matrix becomes ill-conditioned.
- This can cause the solution to have problems even when it “converges” and does not have singularity problems.
- We can fix this by adding small terms to the mass matrix.

Problem with Sparse Mass Matrix

- Video with sparse mass matrix

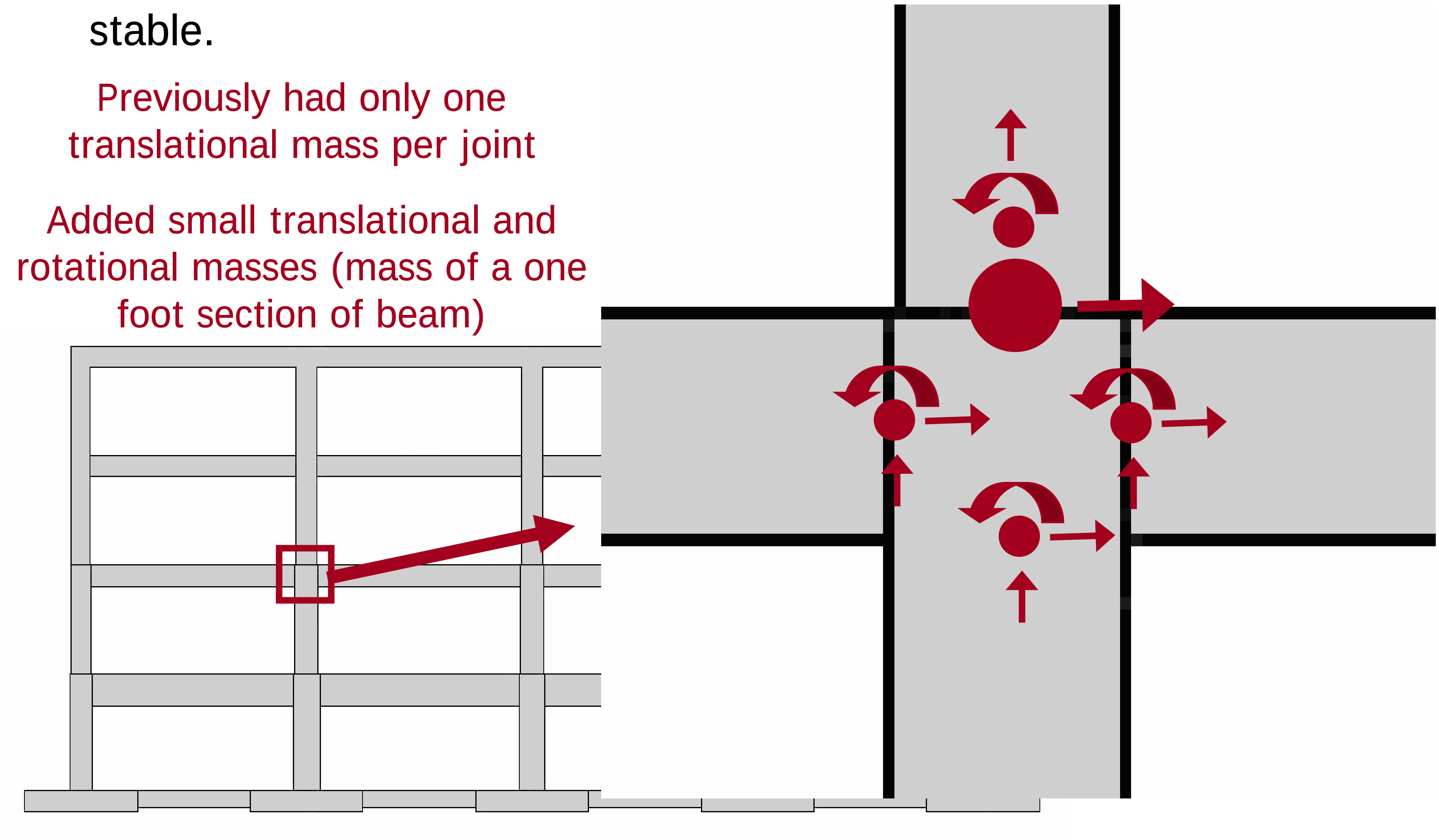


Problem with Sparse Mass Matrix

- Adding small masses to populate mass matrix and make solution more stable.

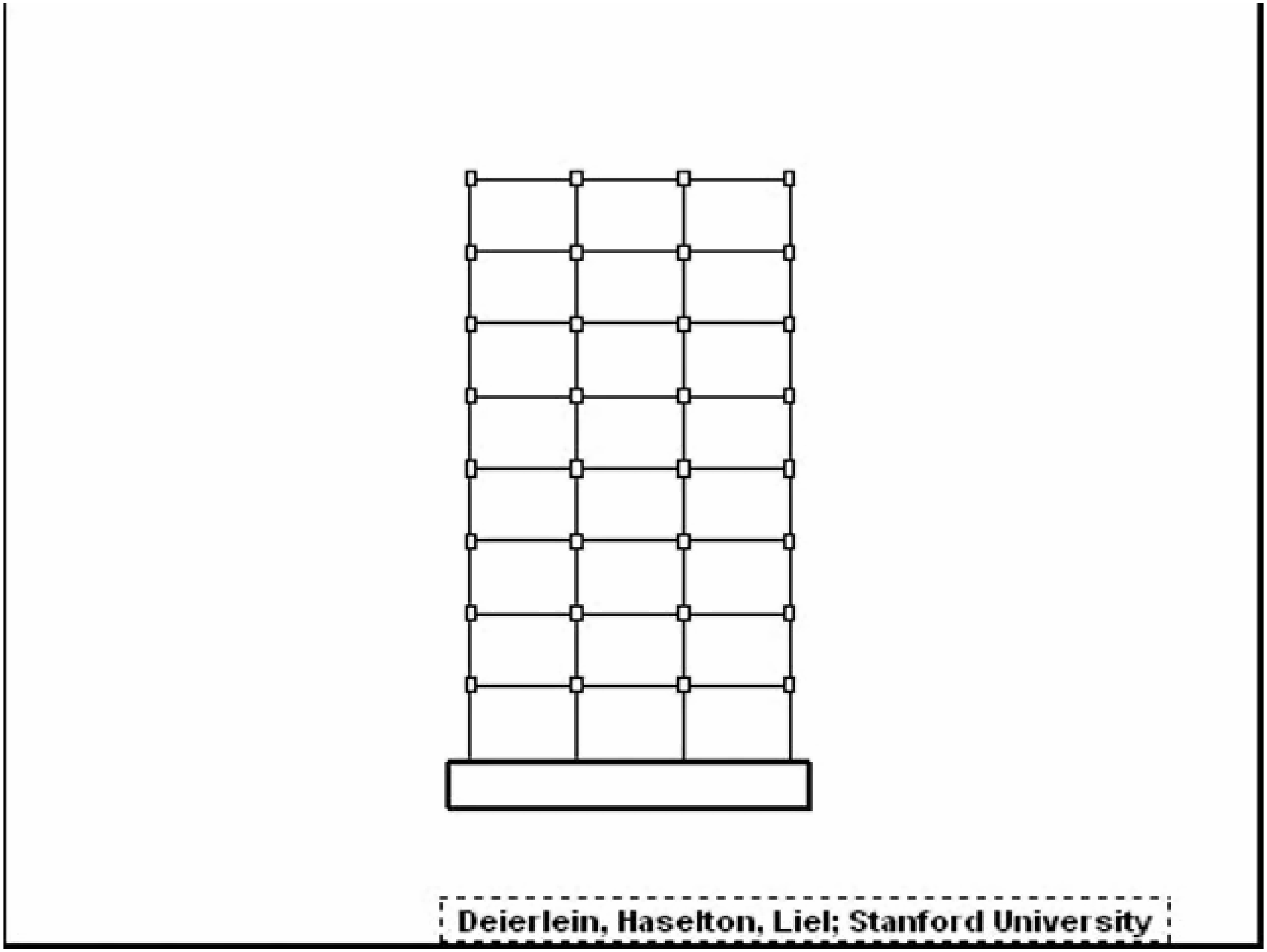
Previously had only one translational mass per joint

Added small translational and rotational masses (mass of a one foot section of beam)



Problem with Sparse Mass Matrix

- Video with small terms added to mass matrix, with higher ground motion intensity



Summary

- It is easy and extremely useful for Matlab to run OpenSees
- For collapse analyses, numerical issues are a large problem
- Using Tcl and OpenSees commands, we can identify these numerical problems and "get around them" in order to get analysis results that are reliable
- Tools Developed (available shortly)
  - Calibrated model for RC beam-columns (PEER report available shortly)
  - Excel program for automated model generation (available as soon as papers are out)
  - Matlab program to run and post-process analysis (available shortly)
- A book that I have found *extremely* useful for programming in Tcl:
  - "Practical Programming in Tcl and Tk", Fourth Edition, by Brent B. Welch and Ken Jones

Reference Slides

Reference: Calib. of Beam-Column Model

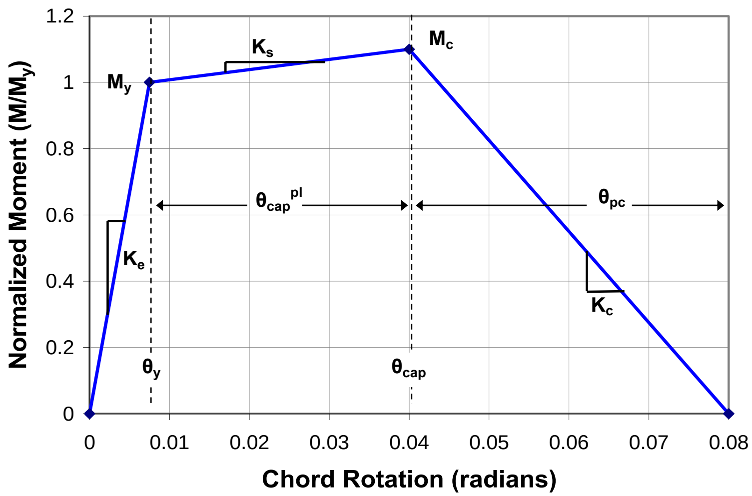
$$\frac{EI_y}{EI_g} = -0.07 + 0.59\left[\frac{P}{A_g f'_c}\right] + 0.07\left[\frac{L_s}{H}\right]$$

$$0.2 \le \frac{EI_y}{EI_g} \le 0.6$$

$$\frac{EI_{stf}}{EI_g} = -0.02 + 0.98\left[\frac{P}{A_g f'_c}\right] + 0.09\left[\frac{L_s}{H}\right]$$

$$0.35 \le \frac{EI_{stf}}{EI_g} \le 0.8$$

$$M_c / M_y = (1.25)(0.89)^{v}(0.91)^{0.01 c_{units} f'_c}$$



$$\theta_{cap,pl} = 0.12\left(1 + 0.55 a_{sl}\right)\left(0.16\right)^{v}\left(0.02 + 40\rho_{sh}\right)^{0.43}\left(0.54\right)^{0.01 c_{units} f'_c}\left(0.66\right)^{0.1 s_n}\left(2.27\right)^{10.0\rho}$$

$$\theta_{pc} = (0.76)(0.031)^{v}(0.02 + 40\rho_{sh})^{1.02} \le 0.10$$

$$\lambda = (127.2)(0.19)^{v}(0.24)^{s/d}(0.595)^{V_p / V_n}(4.25)^{\rho_{sh,eff}}$$

[Mean and uncertainty both quantified]

Haselton, C.B., Liel A.B., Taylor Lange S. and G.G. Deierlein, 2006. *Beam-Column Element Model Calibrated for Predicting Flexural Response Leading to Global Collapse of RC Frame Buildings*, PEER Report 2006, Pacific Engineering Research Center, University of California, Berkeley, California, (in preparation).

Reference: Dynamic Solution Algorithm

```
# Loop for full earthquake unless we detect collapse or until the system goes singular

while { $eqNotFinished == 1 && $isCollapsed == 0 && $isSingular == 0 } {

      test RelativeNormDispIncr 1.0e-6 10 1;      # test testType $tolerance $maxNumIter
      <$printFlag>

      algorithm NewtonLineSearch 0.6;

      set ok [analyze 1 $dT]

      if {$ok != 0} {
              algorithm Newton
              set ok [analyze 1 $dT]
      }

      if {$ok != 0} {
              algorithm Newton -initial
              set ok [analyze 1 $dT]
      }

      if {$ok != 0} {
              algorithm KrylovNewton
              set ok [analyze 1 $dT]
      }

      # Continued on next slide...
```

This algorithm was built starting from the work of other researchers:

- **Frank McKenna**
- **Paul Cordova**

Reference: Dynamic Solution Algorithm

```
# ...continued from last slide


if {$ok != 0} {

        # Repeatedly reduce the time step then the tolerance and try all of the solution
        # algorithms again
}

if {$ok != 0} {

        # STOP ANALYSIS: Give up and let OpenSees return with a convergence error

} else {

        # Output the convergence tolerance achieved for user to decide if that is acceptable

}

# If we get to this line the analysis converged...check for collapse and singularity

if { isCollapsed } { # Stop analysis and report collapse [Details on next slide] }

if { isSingular } { # Stop analysis and report singularity [Details on next slide] }

}
```

Reference: Treatment of Collapse State

- At each step of the analysis we check collapse based on a predefined drift limit (this collapse definition can easily be altered).
- We stop the analysis if the model collapses (dynamically unstable) or else the solution may become singular (which causes an OpenSees error)
- Code to check for collapse:

```
# Get floor displacements from OpenSees at this analysis step
set floor2Displ "[nodeDisp $nodeNumAtFloor2 1]“;
        ...do for all floors

# Compute story drift ratios for all stories
set story2DriftRatio [expr ($floor3Displ - $floor2Displ) / [lindex $floorHeightsLIST 2]]
        ...do for all stories

# Check is collapseDriftLimit is exceeded
if { abs(any story drift) > collapseDriftLimit } {
        set isCollapsed 1;  # Stop the analysis at this time step and report collapse to MATLAB
}
```

Reference: Treatment of Singularity

```
# Code for singularity check

    # Check for singularity in a single nodal recorder
        set floor2Displ "[nodeDisp $nodeNumAtFloor2 1]“
        # Check recorders for IND or QNAN...
        set checkForQNANFlr2  [string first QNAN $floor2Displ 1];
        set checkForINDFlr2     [string first IND $floor2Displ 1];
        # Check recorder for a huge unreasonable displacement...
        if { $ floor2Displ > 1000000.0 } {
                set isSingular 1; # Set singularity flag
        }
    # ...do same checks for many other nodal recorders...

    # Set the singularity flag based on IND or QNAN
        if {($checkForQNANFlr2 != -1) || ($checkForINDFlr2 != -1) || ...check other nodes... } {
                set isSingular 1; # Set singularity flag
        }

    # Stop analysis if singular
    if { isSingular == 1 } {
        # Stop the analysis at this time step and report singularity to MATLAB
    }
```

Thank You

- Thank you for your attention!
- I am a believer in all of us not wasting our time in research, so please contact me if any of this would be useful in your work.
  - haselton@stanford.edu