The Evolution of OpenSees: Is the Open Source Model a Success?

Michael H. Scott

3rd International Conference on Geohazard Information Zonation (GIZ 2014)

October 20, 2014
Medan, Indonesia

Overview of Talk

- Overview of Computational Simulation
- Software for Computational Simulation
- History of OpenSees
- OpenSees Today
- Applications of OpenSees
- Missing Pieces of OpenSees
- Effectiveness of the Open Source Model

The Importance of Computational Simulation

Experiments are the best way to understand the response of a structural system

HOWEVER

- Expensive to design and fabricate
- Limited range of properties
- Controlled loading conditions
- Difficult to perform at full scale
- Must be well-instrumented to capture local response

The Importance of Computational Simulation

The goal of simulation should provide an accurate prediction of structural system response

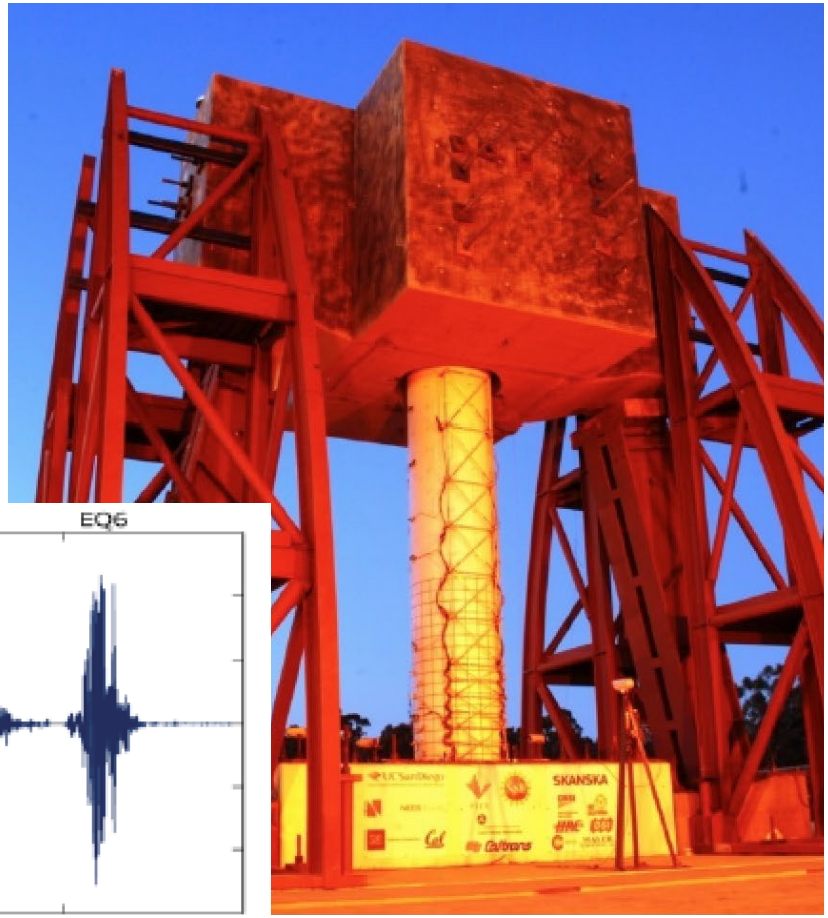
- At low cost
- With high repeatability
- To unanticipated loading conditions
- At any scale
- At high fidelity (if desired)

Simulation and Experiments are not mutually exclusive

- Experiments guide the calibration of simulation models
- Simulation models guide the design of experiments

Current Simulation Capabilities

- RC Bridge Column Test
- Subjected to six EQ motions of increasing intensity and duration
- Blind prediction by 41 analysts



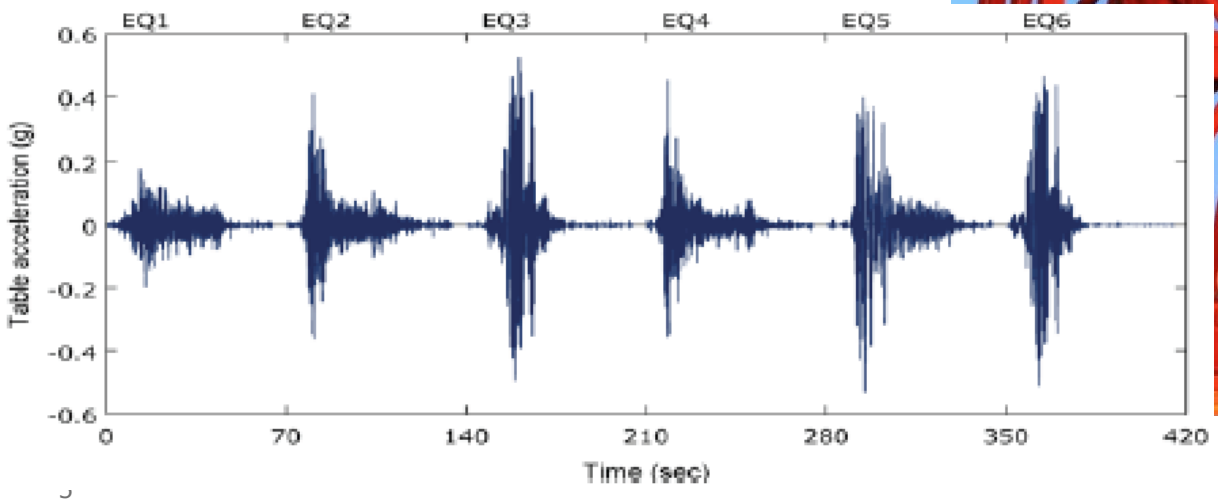
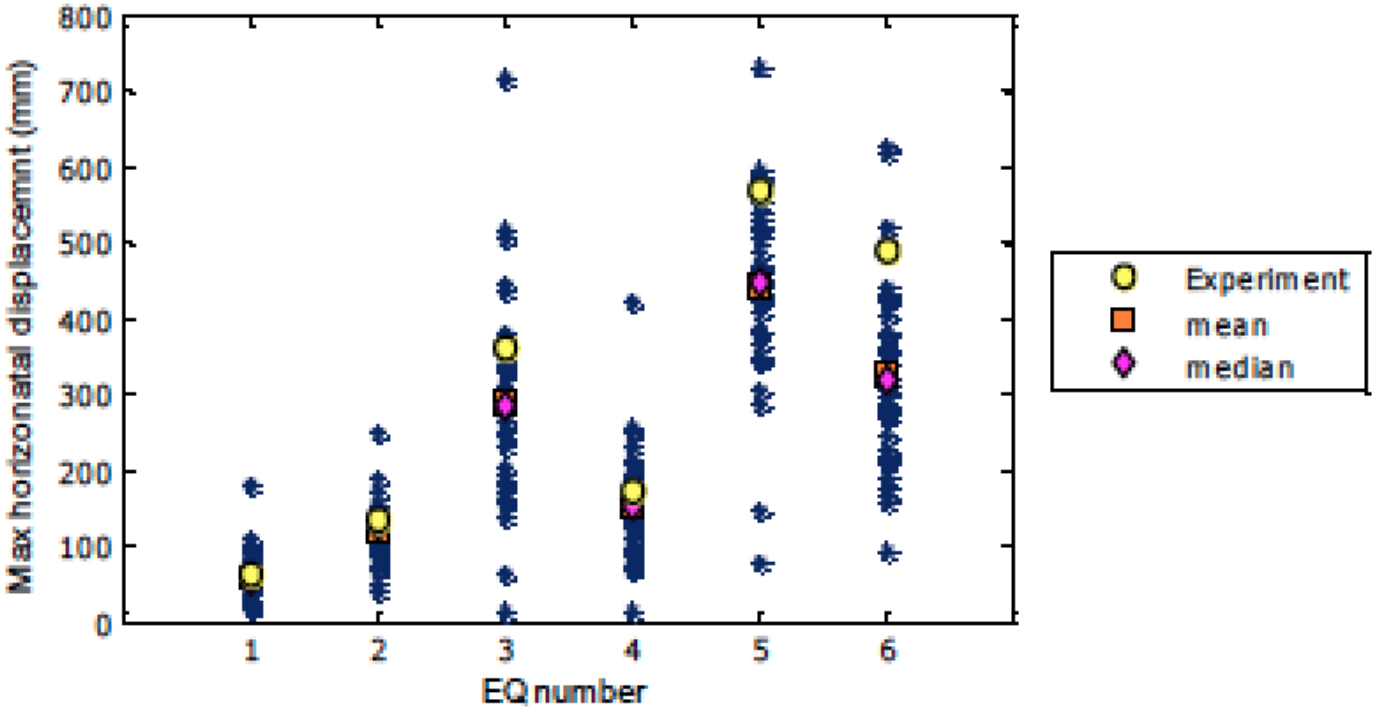


Photo from PEER

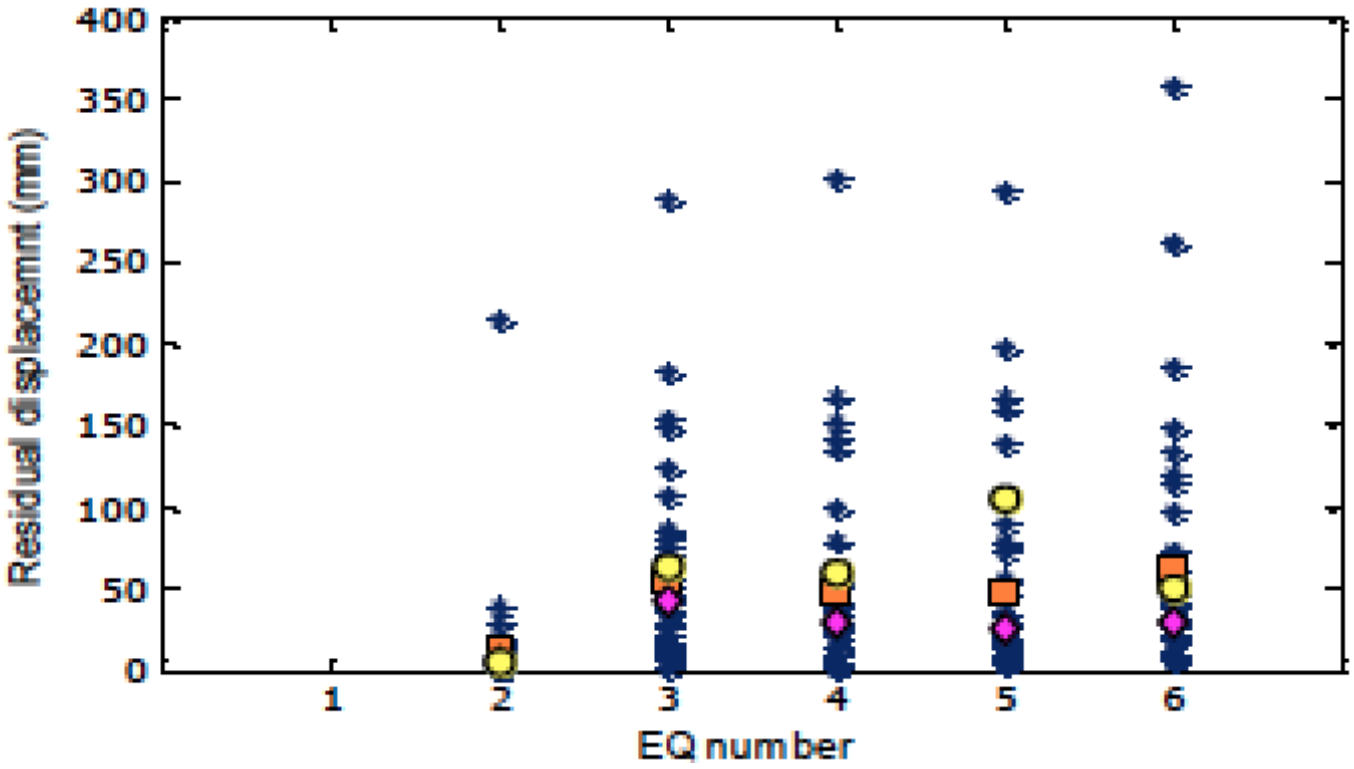
Results for Column Drift



- Bias between experiment and simulation grows with number of EQs
- Average COV of simulated drift over all EQs is 39%

Source: B. Stojadinovic, ETH

Results for Residual Displacement



- Simulations generally underpredict residual displacement
- More scatter in numerical simulation
- Average COV of residual displacement over all EQs is 39%

Source: B. Stojadinovic, ETH

What Can be Done to Improve Simulation?

- Develop new constitutive models that refine simulated response
- Develop new models that account for additional physical effects
- Develop new mathematical formulations of component behavior
- Utilize efficient and accurate numerical solution techniques
- Quantify the uncertainty of the simulated response

Commercial Software for Structural Simulation

- SAP/Perform-3D
- ANSYS
- ABAQUS
- LS-DYNA
- Many more

- Most commercial software allows for user defined element and constitutive models, but the high level software architecture is rigid, limiting the types of new models
- Not possible to share code and build upon others' ideas
- Difficult to employ new high performance computing technologies … have to wait for new release
- Single user software licenses can cost $1000s per year

Open Source Software for Structural Simulation

- OpenSees
- Zeus-NL
- A few others

- Source code can be modified, created, shared by all
- Free to download and use
- Software license prevents commercialization

What is OpenSees?

- OpenSees = Open System for Earthquake Engineering Simulation
- It is a software framework of flexible and extensible modules from which a developer can create custom applications
- The framework defines interfaces for developers to implement their models and methods
- Written in C++, but links to highly efficient equation solvers written in FORTRAN
- Source code maintained centrally and frequently updated

Creation of OpenSees

- Developed from the Ph.D. Dissertation of Frank McKenna

  Object oriented finite element analysis:
  frameworks for analysis algorithms and parallel computing
  University of California, Berkeley (1997)


F. McKenna

- Gregory Fenves
  - Frank McKenna's Ph.D. advisor
  - Research interests in finite element simulation
  - Instrumental in the creation of PEER
- Pacific Earthquake Engineering Research Center (PEER)
  - Started in 1997 with funding from the National Science Foundation (NSF)
  - One of PEER's goals was to provide a common simulation framework for education and research in earthquake engineering
    - Utilize modern software techniques
    - Disseminate to industry for implementation


G. Fenves

Early Days of OpenSees

- Originally called "G3"
  - This was the working group number in PEER that was tasked with creating a common simulation framework
  - Changed to "OpenSees" in 2000
  - Object-oriented MATLAB program called "G2" was written by Prof. Fenves for instructional purposes in graduate courses on finite element analysis and nonlinear structural analysis at UC Berkeley

- Models/algorithms available in 1998
  - Material nonlinear truss, elastic beam
  - Elastic, Elastic Perfectly Plastic, and Parallel uniaxial material models
  - Newton-Raphson and Modified Newton solution algorithms
  - Newmark time integration
  - Profile symmetric, positive definite linear equation solver

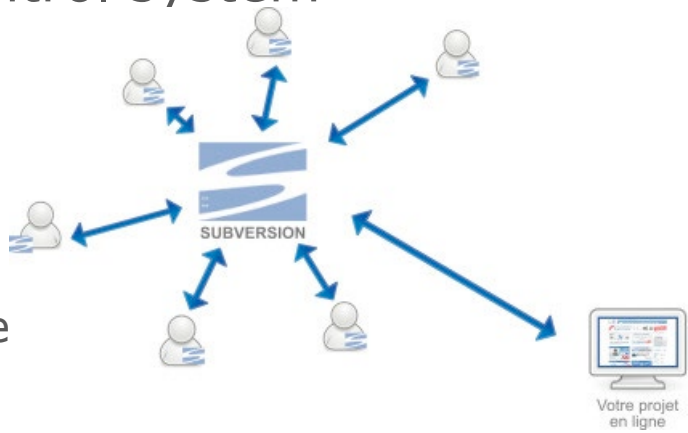
OpenSees Extension of the Tcl Scripting Language

- Fixed format input files, which were common of finite element programs at the time, have limited scope and are prone to errors
- However, writing a main() function in C++ to create and analyze an OpenSees model would be cumbersome and error prone

- The string-based Tcl scripting language was extended with custom commands creating and analyzing an OpenSees model
  - Fully programmable, string-based language
  - Conditionals, looping, procedures, sourcing other scripts, file I/O
  - Quickly build applications
  - Matlab not considered because of its cost
  - Python was considered, but was an unproven commodity in 1998

OpenSees Source Code Repository

- Uses svn (Apache Subversion) revision control system
  - Stores the current version and history of changes to source code on a single server
  - Ensures changes made by one developer do not overwrite \those made by others in a multi-developer environment
  - Users can check-out latest source code at any time
  - Only a handful of developers have check-in access
  - Can get messy when developers don't perform regular syns with svn



- Prior to 2006, OpenSees used the cvs (Concurrent Version System) revision control system
- When it was known as G3, Frank McKenna kept the source code on a zip disk

Early Keys to Wide Use of OpenSees

By the time its name was changed from G3 to OpenSees in 2000, several models critical to widespread adoption by the earthquake engineering community were in place

- Force-based beam-column formulation
  - Non-iterative formulation published in 1997
  - Implemented in OpenSees by Remo de Souza
- General fiber discretization of beam cross-sections
  - Also implemented by Remo de Souza
  - Project for CE 224 class on computer-aided engineering
- Uniaxial material models
  - Steel01, Steel02, Concrete01, Concrete02, Hysteretic material
  - Translated from FORTRAN out of Filip Filippou's FEDEAS program
- Zero length element
  - Implemented by Greg Fenves
  - Basis for p-y models of SSI and base isolation

R. De Souza

F. Filippou

Quantification of Uncertainty

- The development of performance-based earthquake engineering (PBEE) guidelines was central to PEER's mission

- Culminating in 2003, Prof. Armen der Kiureghian and his then Ph.D. student, Terje Haukaas, developed modules in OpenSees for uncertainty quantification
  - Monte Carlo simulation
  - First order reliability
  - Importance sampling
  - Fragility analysis

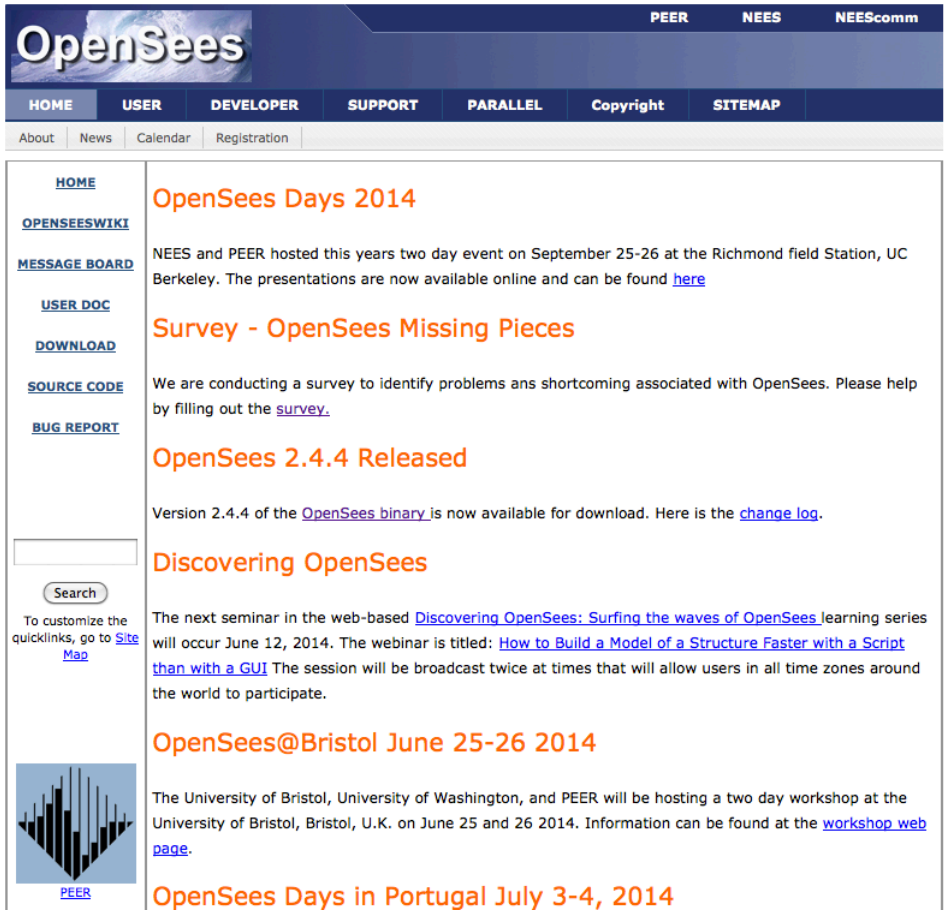
T. Haukaas

A. der Kiureghian

OpenSees Today

- Over 100 uniaxial material models, estimated at
  - 15 concrete models
  - 8 steel models
  - 20 hysteretic models (including DRAIN, SNAP, FEDEAS)
  - 15 p-y, q-z, t-z models
  - 10 utility wrappers for building arbitrarily complex models (series, parallel, min/max, static condensation)
- Various line and solid elements formulated for material and/or geometric nonlinearity
- About a dozen linear equation solvers
  - Matrix topology depends on problem type
  - Sparse/dense, symmetric/non-symmetric, direct/iterative
- Another dozen each of time integration methods and root-finding algorithms

OpenSees Webpage



http://opensees.berkeley.edu

OpenSees International

Visitors to opensees.berkeley.edu in 2013

Visitors

83,663

% of Total: 100.00% (83,663)

Visits

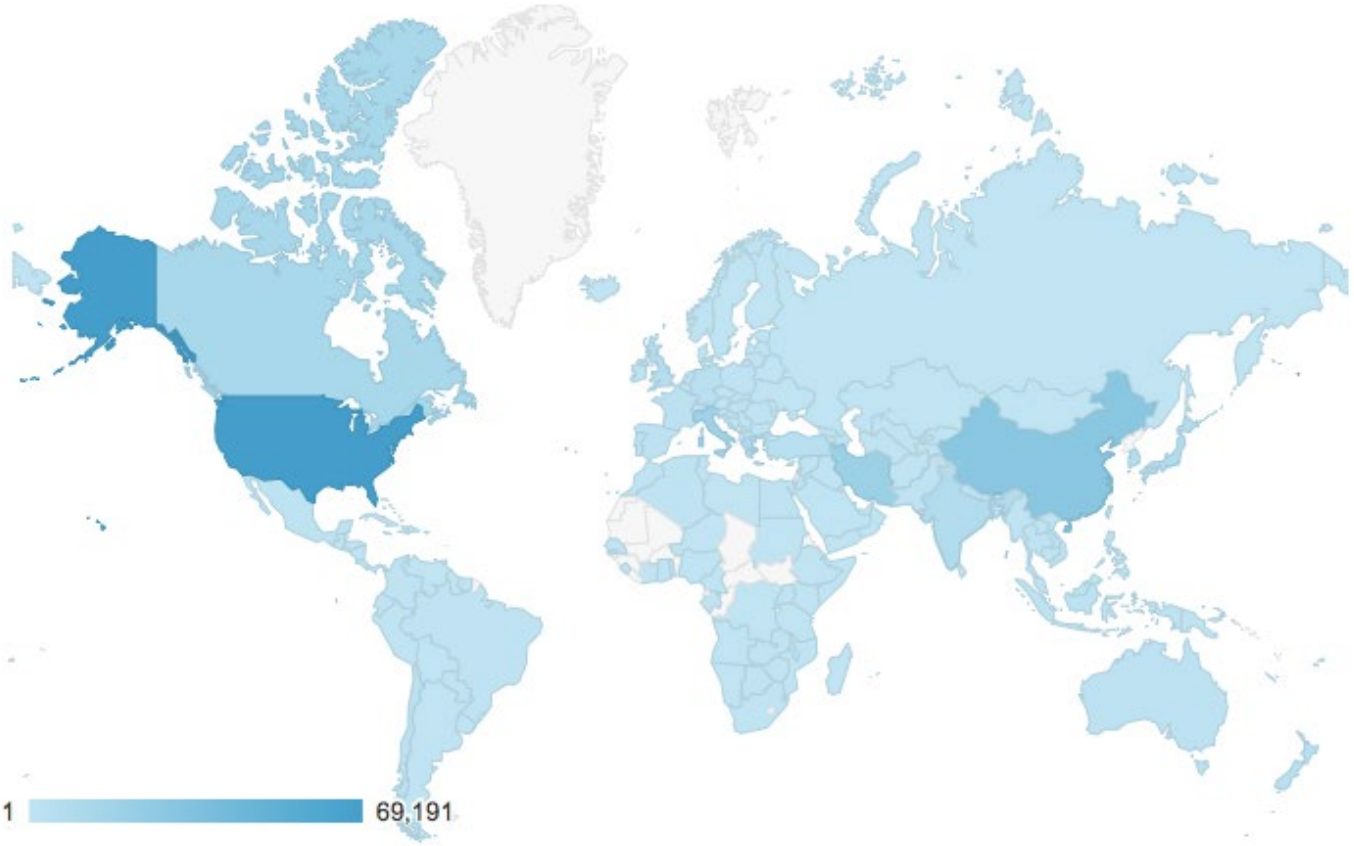
249,630

% of Total: 100.00% (249,630)

Pageviews

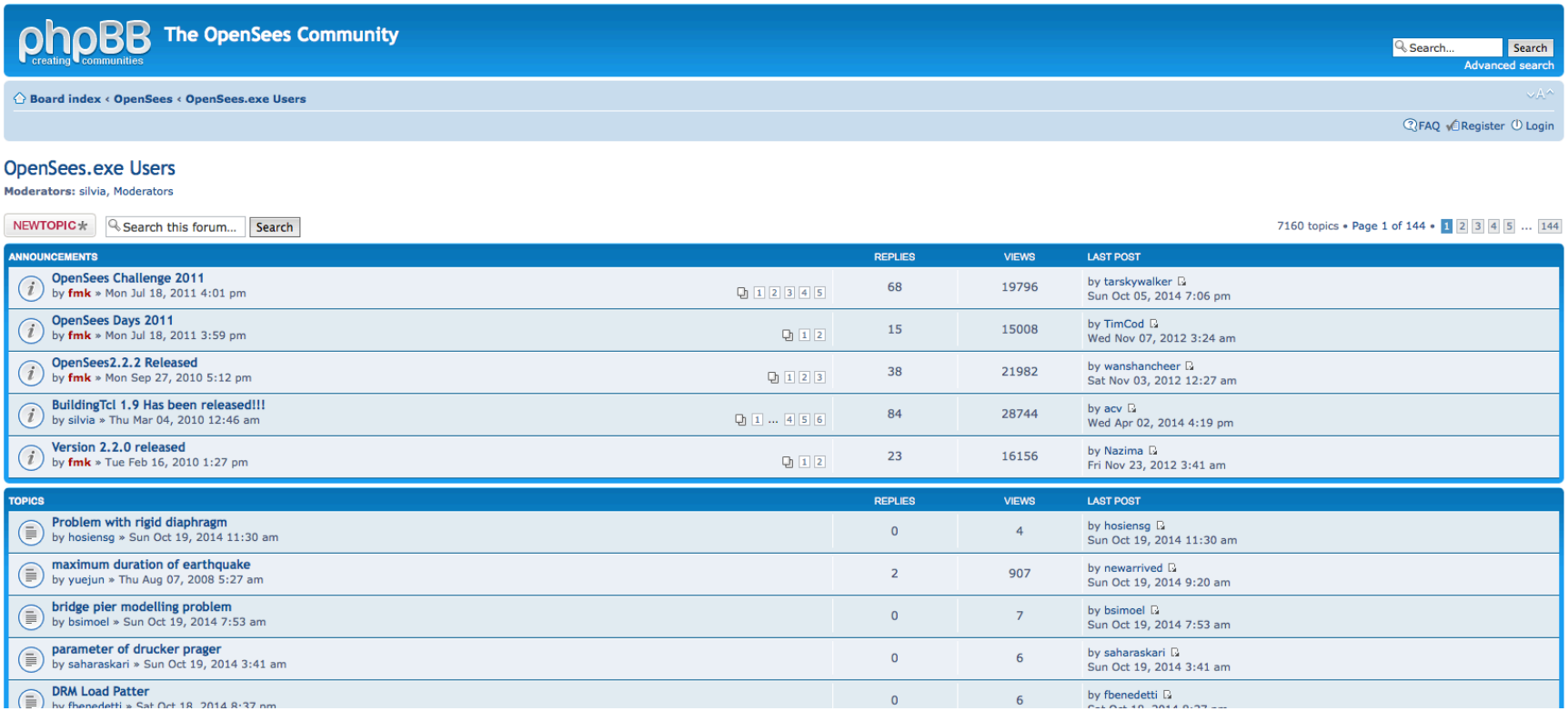
1,492,715

% of Total: 100.00% (1,492,715)



Source: Google Analytics, Compiled by F. McKenna

OpenSees Message Board

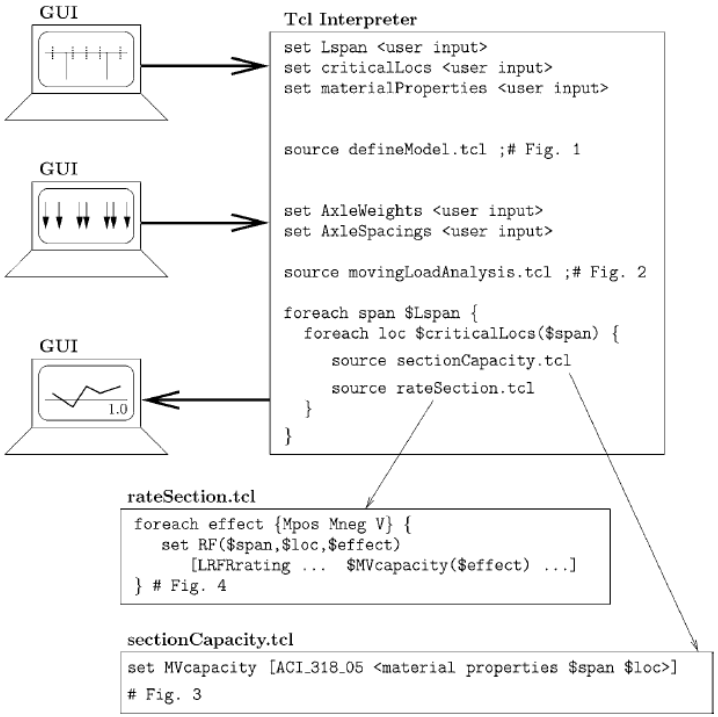


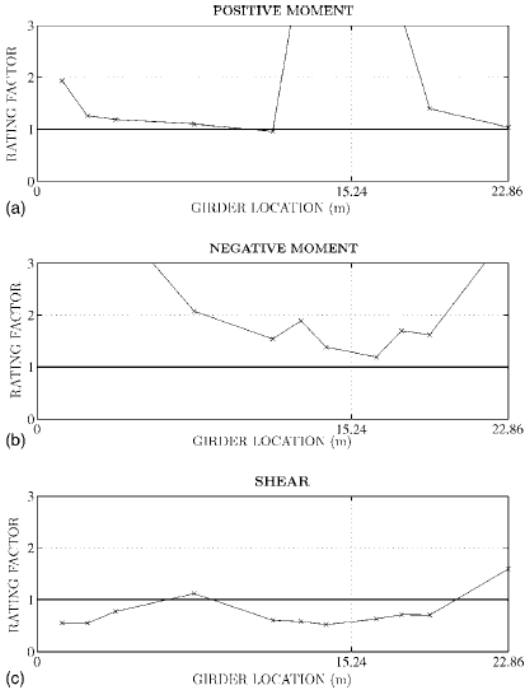
Software License for OpenSees

- Issued by the Regents of the University of California
- Grants permission, without fee and without a written license agreement, for
  a) use, reproduction, modification, and distribution of this software and its documentation by educational, research, and non-profit entities for noncommercial purposes only; and
  b) use, reproduction and modification of this software by other entities for internal purposes only.
- Basically means the executable and source code cannot be modified then repackaged and sold as "new" software

Bridge Rating

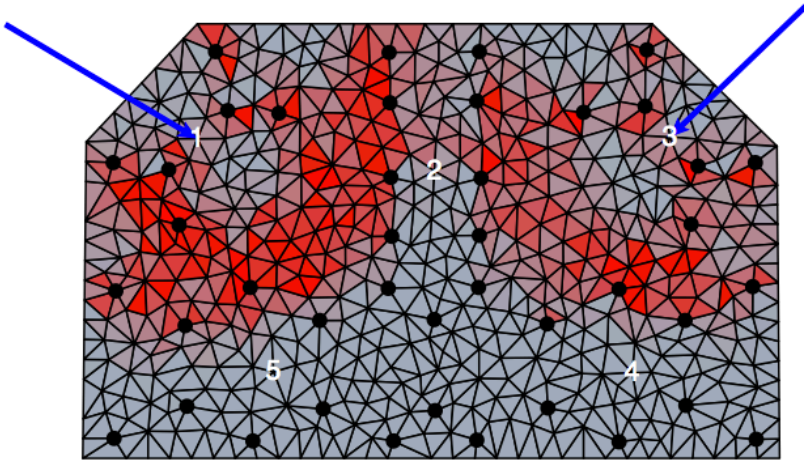
- Relatively easy example of how to use OpenSees and Tcl to build applications beyond earthquake engineering
- Determine rating factors for bridge girders by writing scripts
  - Determine girder capacities at critical locations
  - Move axle loads across bridge, perform demand analysis
  - Compare demand with capacity to get rating factor





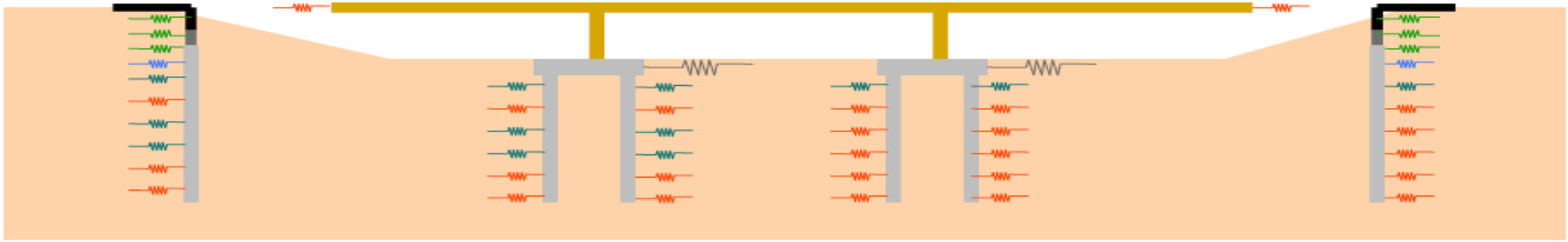
Gusset Plate Evaluation

- Develop mesh of shell elements based on gusset plate geometry and fastener locations
- Write script to load all possible combinations of fastener groups and determine the controlling load case

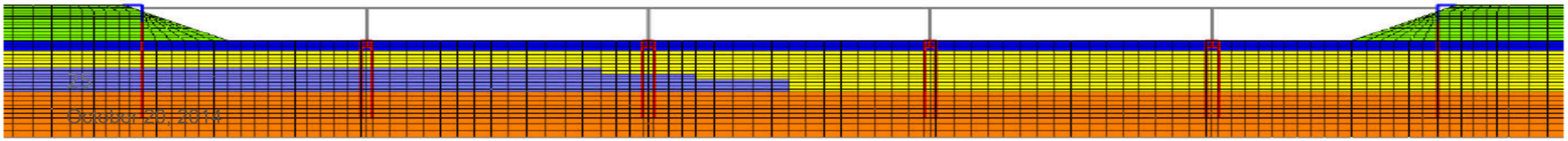


Soil-Structure Interaction

Bridge Idealization



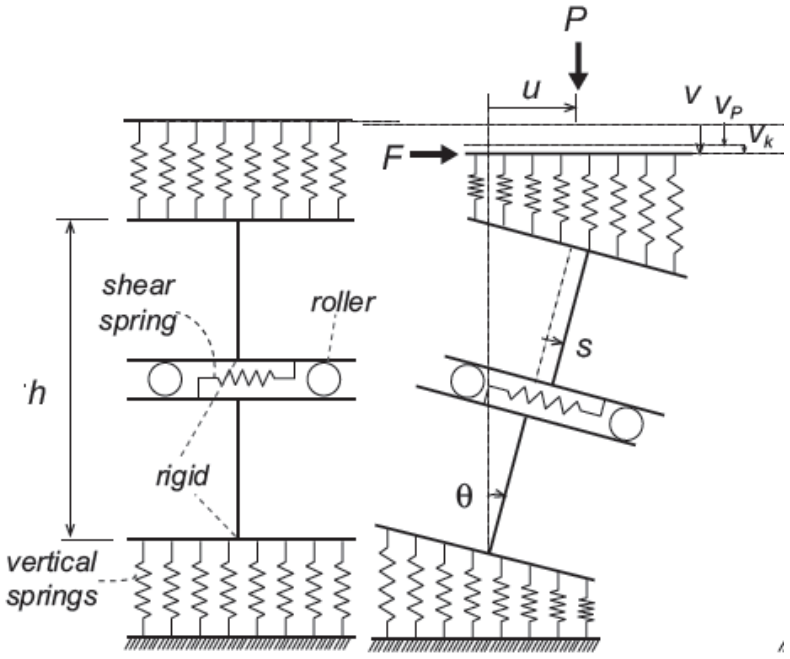
OpenSees model



Base Isolator Models

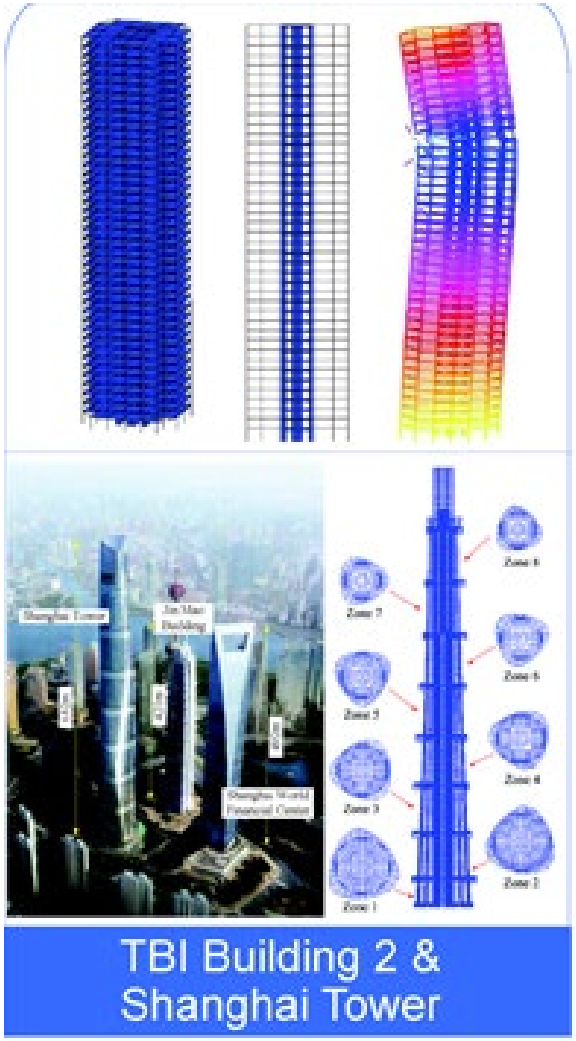
Several base isolator models under development in OpenSees

- Flat slider
- Single friction pendulum
- Triple friction pendulum



Analysis of Tall Buildings

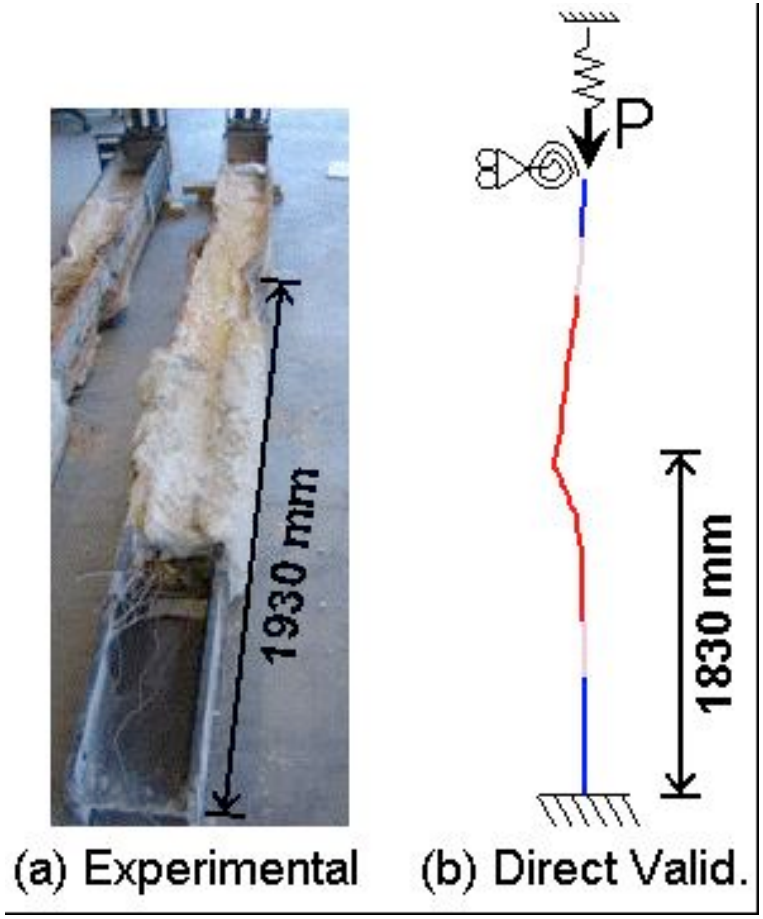
- OpenSees used in conjunction with high performance computing
- Multi-layered shell elements with concrete and reinforcing steel materials
- Custom pre- and post-processors developed



Source: Xinzheng Lu

Fire Attack

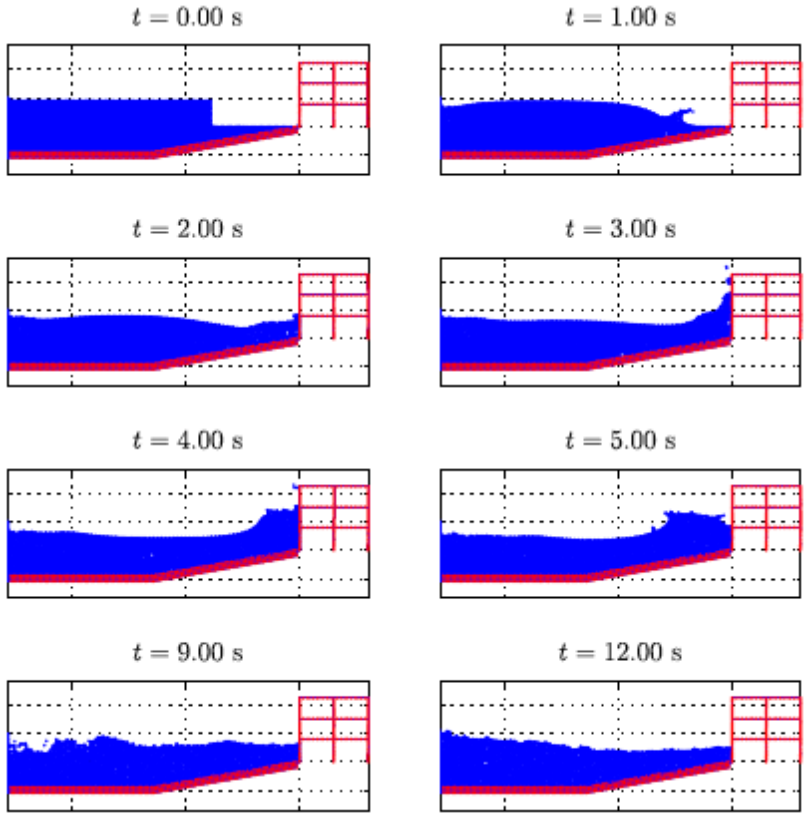
- Thermal effects from fire attack implemented in OpenSees (A. Usmani)
- Numerical verifications and model validations
- Important consideration for multi-hazard event, namely fire following earthquake



Source: M. Garlock

Fluid-Structure Interaction

- Particle Finite Element Method implemented for fluid-structure interaction
- Tsunami loading and storm surge on buildings and bridges
- Build upon existing modules for sequential multi-hazard analysis, e.g., tsunami following earthquake



Scholarly Impact of OpenSees

- OpenSees has been used extensively by researchers to produce publishable results
- There is no standard way to cite OpenSees
- From Google Scholar, the user manual and website are most frequently cited works for OpenSees

☐	**Open system for earthquake engineering simulation** S Mazzoni, F McKenna, MH Scott, GL Fenves, B Jeremic User Command-Language Manual, Pacific Earthquake Engineering Research Center ...	616 *	2006
☐	**Open system for earthquake engineering simulation** F McKenna, GL Fenves, MH Scott University of California, Berkeley, CA	381	2000

- The journal article on OpenSees with the most citations:

☐	**Nonlinear finite-element analysis software architecture using object composition** F McKenna, MH Scott, GL Fenves Journal of Computing in Civil Engineering 24 (1), 95-107	55	2009

What is OpenSees Missing?

- Emanating from research in structural and geotechnical earthquake engineering, OpenSees has expanded to applications in bridge engineering, fire attack, and fluid-structure interaction
- OpenSees has not taken hold in the wind engineering community and it lacks constitutive models for wood
- Its scripting capabilities make OpenSees ideal for linking its detailed structural modeling capabilities with probabilistic hazard assessment modules

Is the Open Source Model Effective?

- It has allowed users of OpenSees to build upon others models and improve the state of knowledge
- State of the art models for earthquake engineering are accessible to anyone around the world

HOWEVER

- Developers are often reluctant to add their source code to the OpenSees repository
  - Waiting for journal articles to be reviewed and published
  - Worried that others will improve upon the model without receiving proper credit for the work
  - Simply don't want to share
- There is a lack of documentation for most models because it is thought that "source code is self-documenting"

Thank You

Questions?