



QuakeCoRE
NZ Centre for Earthquake Resilience

QuakeCoRE OpenSees Training Workshop 2016 Introduction



What is OpenSees?



OpenSees is a finite element analysis **framework** for structural and geotechnical earthquake engineering simulation.

- **Open System for earthquake engineering simulation**
- **Developed at University of California Berkeley through support by the Pacific Earthquake Engineering Research (PEER) Center**

OpenSees is open-source. Anyone can download the source code and see how it works under the hood, make changes, and/or add new features.

- **This open-source nature allows for free exchange of research tools and allows for users to build upon the contributions of others**

OpenSees relies upon community contributions for success.

- **Classes (e.g. elements and materials), documentation, testing, user support, and more are all provided by users in the community**
-

What is OpenSees?

There is no graphical user interface (GUI). The OpenSees “application” is really an interpreter that is extended to include commands for finite element analysis.

- The file(s) you create and submit to OpenSees are actually programs

Wait, what is an interpreter?

- An interpreter is a program that executes commands written in a programming language immediately. MATLAB is an interpreter.
- There is no compilation and linking steps necessary

The OpenSees interpreters are extensions of the tcl interpreters tclsh and wish.

- Tcl is a dynamic programming language with features useful for your OpenSees programs
 - This workshop will cover some basic scripting using tcl
 - An OpenSees Python interpreter is coming soon
-

Working with OpenSees

What should your expectations be for working with the OpenSees framework? (paraphrased from Greg Fenves at UC Berkeley)

- OpenSees is a research tool primarily, but it is reasonably stable and is used in professional practice by some firms/individuals.
 - OpenSees is under continual development by students, faculty, and other researchers.
 - New elements, materials, solvers, etc.... are added all the time
 - Documentation, testing, and user interface typically lags behind inclusion in the main OpenSees source code
 - OpenSees is not perfect. You can (and likely will) get error messages instead of results.
 - In the majority of cases, these errors are directly related to problems in your model/script, not problems with the primary functionality of OpenSees
 - We will discuss troubleshooting techniques in this workshop to help move past such errors in a systematic way
 - As with any analysis tool, working with OpenSees requires careful model preparation and result interpretation.
 - An investment of time/learning is required. This is not unique to OpenSees, but it's also not reasonable to expect to become an advanced user overnight.
 - The OpenSees open-source community relies upon contributions from the entire community to be a success. If you are using OpenSees, you are part of this community.
 - Working with the community goes both ways. The community message boards are a great resource, but if no one is providing answers to posts, then the message boards aren't as useful.
-

Primary Objectives of Workshop



QuakeCoRE
NZ Centre for Earthquake Resilience

Learn the basics of downloading and running the OpenSees interpreters

- Step-by-step instruction to get things set up on your system

Demonstration of basic modeling

- Working with the OpenSees interpreters
- Basic models to demonstrate fundamental tasks

Overview of structural and geotechnical modeling in OpenSees

- Simple to complex structural and geotechnical models with examples

Pre- and post-processing tools/techniques for simple and complex models

- What are the options for creating models and interpreting results?

OpenSees and parallel processing (OpenSeesSP and OpenSeesMP)

- Why would I want to use parallel processing and which tool is right for me?

Running OpenSees in high-performance computing environments

- NeSI computational resources in NZ and international options such as the DesignSafe-ci at the Texas Advanced Computing Center (TACC)

Learn the steps involved in implementing a new material/element in OpenSees

OpenSees Resources



Getting started manual – basic how-to instructions for new users

- http://opensees.berkeley.edu/wiki/index.php/Getting_Started

User documentation for OpenSees commands and classes

- http://opensess.berkeley.edu/wiki/index.php/Command_Manual

OpenSees basic examples manual – best for new users

- http://opensess.berkeley.edu/wiki/index.php/Basic_Examples_Manual

OpenSees advanced examples manual – good content, but hard to follow when new

- http://opensess.berkeley.edu/wiki/index.php/Examples_Manual

OpenSees practical examples – detailed structural and geotechnical examples

- <http://opensess.berkeley.edu/wiki/index.php/Examples>

OpenSees message board – look for answers, post questions, post ANSWERS

- <http://opensess.berkeley.edu/community/index.php>

All of these resources are open tools. The manuals are Wiki pages editable by all registered users (including YOU!), and the message board is a place for users to post questions, post answers to the questions of others, and search for answers.

Running OpenSees



First step is to download the OpenSees executable and Tcl/Tk libraries from

<http://opensees.berkeley.edu/OpenSees/user/download.php>

You will first need to become a registered user by following the links and instructions provided on the download page, then download and install the necessary tools.

Detailed instructions for Windows and Mac OS X users are available on the QuakeCoRE wiki page (address below) among other places.

<https://wiki.canterbury.ac.nz/display/QuakeCore/OpenSees+Development>

Running OpenSees – 3 Options

```
spring1D.tcl (~/QuakeCoRE/OpenSeesWor...
wipe

model BasicBuilder -ndm 1 -ndf 1

node 1 0.0
node 2 1.0

fix 1 1

set E 1000.0
uniaxialMaterial Elastic 1 $E

set A 1.0
element Truss 1 1 2 $A 1

set P 100.0
timeSeries Linear 1
pattern Plain 1 1 {
  load 2 $P
}

constraints Transformation
numberer RCM
test NormDispIncr 1.0e-6 1
algorithm Linear
system ProfileSPD
integrator LoadControl 0.1
analysis Static

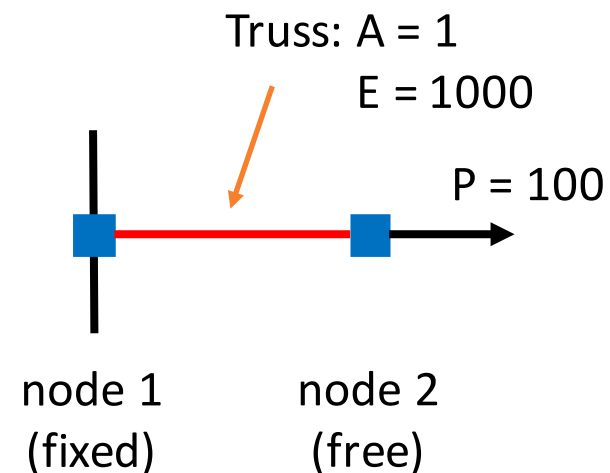
analyze 10

print node 2

wipe
```

A simple 1D spring example using two nodes and a single truss element

The truss is assigned linear elastic response, and a load of 100 units is applied in a linearly-increasing load pattern.



Running OpenSees – 3 Options



Option 1: Run OpenSees Interactively

Commands can be input directly into the prompt.

This isn't typically the most convenient approach, but it can be useful in certain situations.

```
OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center
Version 2.5.0 (rev 6248) 32-Bit

(c) Copyright 1999–2016 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

OpenSees > model BasicBuilder -ndm 1 -ndf 1
OpenSees > node 1 0.0
OpenSees > node 2 1.0
OpenSees > fix 1 1
0
OpenSees > uniaxialMaterial Elastic 1 1000.0
OpenSees > element truss 1 1 2 1.0 1
OpenSees > timeSeries Linear 1
OpenSees > pattern Plain 1 1 {
load 2 100.0
}
OpenSees > constraints Transformation
OpenSees > numberer RCM
OpenSees > test NormDispIncr 1.0e-6 1
OpenSees > algorithm Linear
OpenSees > system ProfileSPD
OpenSees > integrator LoadControl 0.1
OpenSees > analysis Static
OpenSees > analyze 10
0
OpenSees > print node 2

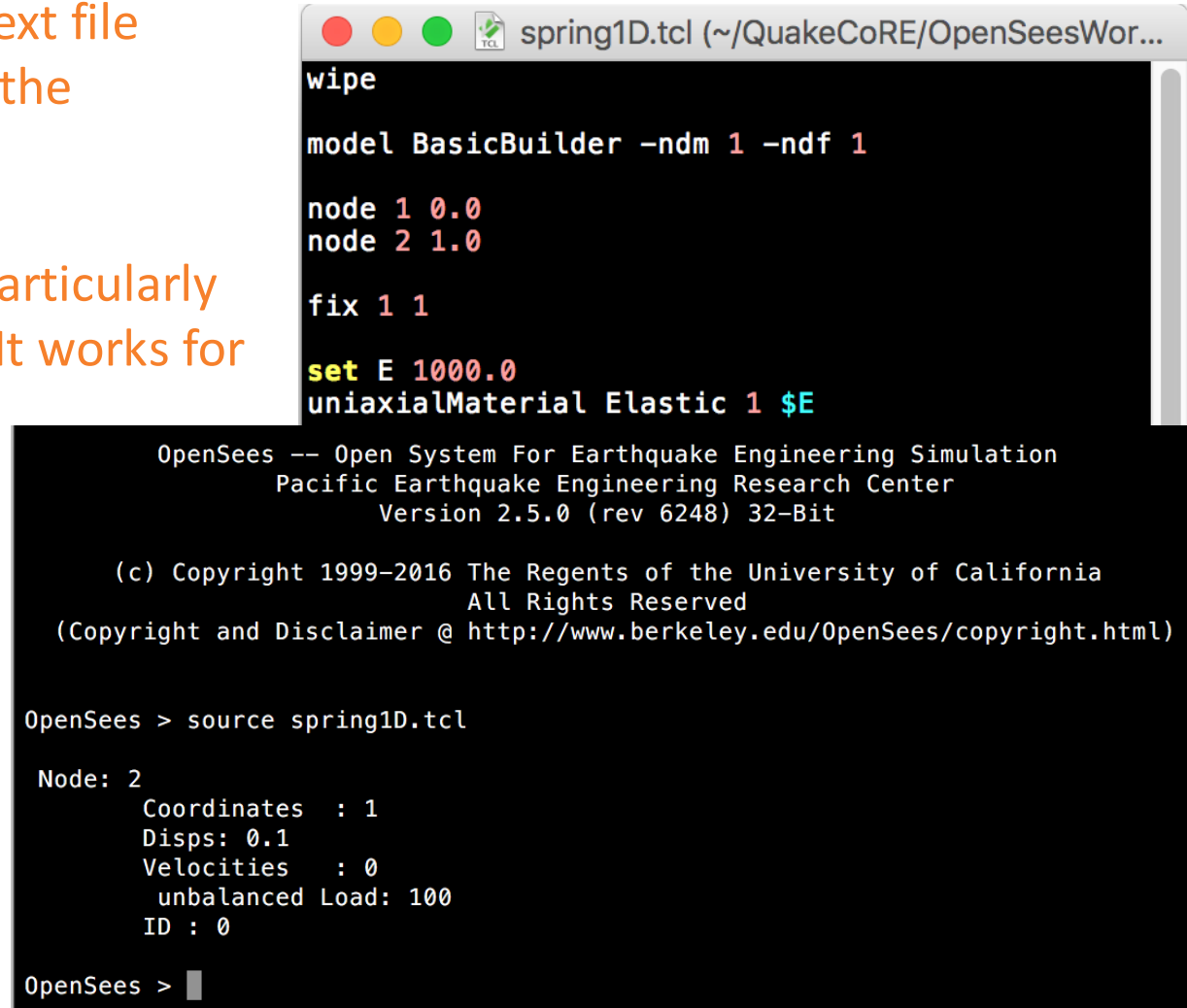
Node: 2
Coordinates : 1
Disps: 0.1
Velocities : 0
unbalanced Load: 100
ID : 0
```

Running OpenSees – 3 Options

Option 2: Source the Commands from File

Commands are placed in a text file that can be sourced in from the OpenSees prompt.

This approach is common, particularly for new users of OpenSees. It works for all model sizes, and is convenient. The only real shortcoming is that it can make debugging more difficult in some cases.



```
spring1D.tcl (~/QuakeCoRE/OpenSeesWor...
wipe
model BasicBuilder -ndm 1 -ndf 1
node 1 0.0
node 2 1.0
fix 1 1
set E 1000.0
uniaxialMaterial Elastic 1 $E

OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center
Version 2.5.0 (rev 6248) 32-Bit

(c) Copyright 1999–2016 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

OpenSees > source spring1D.tcl

Node: 2
Coordinates : 1
Disps: 0.1
Velocities : 0
unbalanced Load: 100
ID : 0

OpenSees > █
```

Running OpenSees – 3 Options



Option 3: Run OpenSees in Batch Mode

Commands are placed in a text file that is executed at startup, directly from the terminal or command prompt.

This approach is arguably the most powerful of the three options.

It allows you to save the runtime information in the stdout and stderr to files (useful for debugging and element testing)

```
enci-crm122@exampleFiles crm122$ OpenSees spring1D.tcl

OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center
Version 2.5.0 (rev 6248) 32-Bit

(c) Copyright 1999–2016 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

Node: 2
Coordinates : 1
Disps: 0.1
Velocities : 0
unbalanced Load: 100
ID : 0

enci-crm122@exampleFiles crm122$
```

It also allows you to take advantage of bash scripting and things like the default variables `argc` and `argv`

Running OpenSees – 3 Options

```
spring1DwInput.tcl + (~...hops/exampleFiles) - VIM1
wipe
if {$argc != 1} {
    puts "Incorrect Usage: OpenSees spring1DwInput.tcl $E"
    exit
} else {
    set E [lindex $argv 0]
}

model BasicBuilder -ndm 1 -ndf 1

node 1 0.0
node 2 1.0

fix 1 1

#set E 1000.0
uniaxialMaterial Elastic 1 $E

set A 1.0
element Truss 1 1 2 $A 1

set P 100.0
timeSeries Linear 1
pattern Plain 1 1 {
    load 2 $P
}

constraints Transformation
numberer RCM
test NormDispIncr 1.0e-6 1
algorithm Linear
system ProfileSPD
integrator LoadControl 0.1
analysis Static

analyze 10

print node 2

wipe
```

Running OpenSees – 3 Options

```
spring1DwInput.tcl + (~...hops/exampleFiles) - VIM1
wipe
if {$argc != 1} {
  puts "Incorrect Usage: OpenSees spring1DwInput.tcl $E"
  exit
} else {
  set E [lindex $argv 0]
}

model BasicBuilder -ndm 1 -ndf 1

node 1 0.0
node 2 1.0

fix 1 1

#set E 1000.0
material Elastic 1 $E

set A 1.0
element Truss 1 1 2 $A 1

set P 100.0
timeSeries Linear 1
pattern Plain 1 1 {
  load 2 $P
}

constraints Transformation
numberer RCM
test NormDispIncr 1.0e-6 1
algorithm Linear
system ProfileSPD
integrator LoadControl 0.1
analysis Static

analyze 10

print node 2

wipe
```

Running OpenSees – 3 Options



```
spring1DwInput.tcl + (~...hops/exampleFiles)
enci-crm122:exampleFiles crm122$ OpenSees spring1DwInput.tcl 2000.0

wipe

if {$argc != 1} {
  puts "Incorrect Usage: OpenSees spring1DwInput.tcl $*"
  exit
} else {
  set E [lindex $argv 0]
}

model BasicBuilder -ndm 1 -ndf 1

node 1 0.0
node 2 1.0

fix 1 1

#set E 1000.0
uniaxialMaterial Elastic 1 $E

set A 1.0
element Truss 1 1 2 $A

set P 100.0
timeSeries Linear 1
pattern Plain 1 1 {
  load 2 $P
}

constraints Transformation
numberer RCM
test NormDispIncr 1.0e-6 1
algorithm Linear
system ProfileSPD
integrator LoadControl 0.1
analysis Static

analyze 10

print node 2

wipe

enci-crm122:exampleFiles crm122$ OpenSees spring1DwInput.tcl 2000.0

OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center
Version 2.5.0 (rev 6248) 32-Bit

(c) Copyright 1999-2016 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

Node: 2
Coordinates : 1
Disps: 0.05
Velocities : 0
unbalanced Load: 100
ID : 0

enci-crm122:exampleFiles crm122$ OpenSees spring1DwInput.tcl 4000.0

OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center
Version 2.5.0 (rev 6248) 32-Bit

(c) Copyright 1999-2016 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

Node: 2
Coordinates : 1
Disps: 0.025
Velocities : 0
unbalanced Load: 100
ID : 0
```

Running OpenSees – 3 Options

```
spring1DwInput.tcl + (~...hops/exampleFiles)
enci-crm122:exampleFiles crm122$ OpenSees spring1DwInput.tcl 2000.0

wipe

if {$argc != 1} {
  puts "Incorrect Usage: OpenSees spring1DwInput.tcl"
  exit
} else {
  set E [lindex $argv 0]
}

model BasicBuilder -ndm 1 -ndf 1

node 1 0.0
node 2 1.0

fix 1 1

#set E 1000.0
uniaxialMaterial Elastic 1 $E

set A 1.0
element Truss 1 1 2 $A 1

set P 100.0
timeSeries Linear 1
pattern Plain 1 1 {
  load 2 $P
}

constraints Transformation
numberer RCM
test NormDispIncr 1.0e-6 1
algorithm Linear
system ProfileSPD
integrator LoadControl 0.1
analysis Static

analyze 10

print node 2

wipe

enci-crm122:exampleFiles crm122$ OpenSees spring1DwInput.tcl 4000.0

OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center
Version 2.5.0 (rev 6248) 32-Bit

(c) Copyright 1999-2016 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

Node: 2
Coordinates : 1
Disps: 0.05
Velocities : 0
unbalanced Load: 100
ID : 0

enci-crm122:exampleFiles crm122$ OpenSees spring1DwInput.tcl 4000.0

OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center
Version 2.5.0 (rev 6248) 32-Bit

(c) Copyright 1999-2016 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

Node: 2
Coordinates : 1
Disps: 0.025
Velocities : 0
unbalanced Load: 100
ID : 0
```



QuakeCoRE
NZ Centre for Earthquake Resilience

Thank you!

www.quakecore.nz

