OpenSees & DesignSafe: OpenSeesMP

November 2018

Maria Giovanna Durante, Ph.D.
Postdoctoral Research Fellow
University of Texas at Austin
mgdurante@utexas.edu
OpenSees applications on DesignSafe

as well as data analysis and visualization tools including Jupyter, MATLAB, ParaView and VisIt.
OpenSees applications on DesignSafe

The Open System for Earthquake Engineering Simulation (OpenSees) is a software framework for simulating the static and seismic response of structural and geotechnical systems. It has advanced capabilities for modeling and analyzing the nonlinear response of systems using a wide range of material models, elements, and solution algorithms. One sequential (OpenSees EXPRESS) and two parallel interpreters (OpenSeesSP and OpenSeesMP) are available on DesignSafe. Please select the desired interpreter for more details.

Select a version of OpenSees from the dropdown:

- Please Select
- OpenSees-EXPRESS
- OpenSeesMP (V 2.5)
- OpenSeesMP (V 3.0)
- OpenSeesSP (V 2.5)
- OpenSeesSP (V 3.0)
OpenSees applications on DesignSafe

WORKSPACE

Learn About the Workspace.


DATA DEPOT BROWSER

Select data source
My Data

Browsing:
sal

File name

.runb_checkpoints
 Trash
 All_Data_Processed
 applications
 apps
 apps_test
 archive
 empty
 sal

The Open System for Earthquake Engineering Simulation (OpenSees) is a software framework for simulating the static and seismic response of structural and geotechnical systems. It has advanced capabilities for modeling and analyzing the nonlinear response of systems using a wide range of material models, elements, and solution algorithms. One sequential (OpenSees EXPRESS) and two parallel interpreters (OpenSeesSP and OpenSeesMP) are available on DesignSafe. Please select the desired interpreter for more details.

Select a version of OpenSees from the dropdown:
OpenSeesMP (V 3.0)

RUN OPENSEESMP (V 3.0) ver. 3.0.0.6709

OpenSeesMP is an OpenSees interpreter intended for high performance computers for performing finite element simulations with parameteric studies and very large models on parallel machines. OpenSeesMP requires understanding of parallel processing and the capabilities to write parallel scripts. OpenSeesMP runs on up to 12 KNL Nodes on Stampede2, with 64 cores per Node.

OpenSeesMP (V 3.0) Documentation

Inputs

Input Directory
Select

Click to select input data

The directory containing your OpenSees input files as well as your OpenSees TCL script. You can drag the link for the directory from the Data Browser on the left, or click the 'Select Input' button and then select the directory. To try out sample data copy and paste 'agave://designsafe.storage.default/mock/examples/opensees/FreefieldAnalysisEffective' above.

TCL Script

The filename only of the OpenSees TCL script to execute. This file should reside in the input directory specified above.
Why OpenSeesMP?

OpenSeesMP is specifically developed for HPC.

**Pros:**
- Ideal for very large models and parametric studies;
- More control on the parallelization process;

**Cons:**
- Additional script needed for parametric analysis;
- It goes into the queue.
OpenSeesMP: The **Multiple Parallel** OpenSees Interpreter

In case of large models, it works like OpenSeesSP.

The main Processor (P0) interprets the script to build the model and to construct the analysis.

Other processors (P1, P2, P3) are running sub-domains of the model.
OpenSeesMP: The Multiple Parallel OpenSees Interpreter

In case of parametric analysis, this interpreter runs in parallel several slightly modified version of the basic OpenSees interpreter. Each of them runs an independent analysis and provides:

- the total Number of Processors available ($np$);
- Its unique Processor ID number ($pid$).
For large model, changes to the scripts include:

- Change how degrees-of-freedom are numbered (Numberer Command) to one of the following:
  
  - numberer ParallelPlain;
  - numberer ParallelRCM.

- Change the System of Equation and the Solver (System Command) to one of the following:
  
  - system ParallelProfileSPD;
  - system Mumps;
  - system Petsc.
OpenSeesMP: New Commands

In order to allow each running process to determine the processor it is running on, the number of processors that the user started and to allow the inter-process communications, the following additional commands are provided:

- **getNP**: returns the total number of processors assigned to the user for the job;
- **getPID**: returns a unique processor number ranging between 0 and ($\text{getNP}-1$);
- **send –pid $\text{pid}$ $\text{data}$**: to send the data from a local process to a process whose process id is given by the variable pid. Pid must be in the range 0 to \[\text{expr[getNP]-1}\];
- **recv –pid $\text{pid}$ variableName**: to receive data from a remote process and set the variable named variableName to be equal to that data. Pid must be set \{0,..[expr [getNP] -1, ANY}\}. If the value of $\text{pid}$ is ANY, the process can receive data from any process;
- **barrier**: causes all processes to wait until all process reach this point in the code.

Using these commands it is possible for the user to perform their own domain decomposition analysis. The getNP and getPID commands allow the user to specify which nodes and elements are created on which processor.
OpenSeesMP: Additional tcl script for parametric studies

```tcl
set pid [getPID]  # obtains the Processor ID
set np [getNP]  # obtains the number of processors
set par_list [open "variable_list.txt" r]  # defines the list of variables
```
OpenSeesMP: Additional tcl script for parametric studies

```tcl
set pid [getPID]
set np [getNP]
set par_list [open "variable_list.txt" r]
set countP 0

foreach val [split [read $par_list] \n] {
    if {[expr $countP % $np] == $pid} {
        set valuesList [split $val "/"]
        set grade [lindex $valuesList end]
        source Sequential_script.tcl
        wipe
    }
    incr countP 1
}
```

# obtains the Processor ID
# obtains the number of processors
# defines the list of variables
# initiates a count to split the analyses

# loops through the variables

# assigns each analysis to one processor
# reads one value of the variable
# assigns the value to the variable
# calls the sequential script
# cleans the workspace

# increases the count to move to the next analysis
Effective Site Response Analysis

Problem: effective stress site response analysis of a layered deposit of cohesionless soil underlain by an elastic half-space.

Model: A single column of soil is modeled in 2D (with periodic boundary conditions to emulate a 1D analysis) and is subject to an earthquake excitation. Nine node quadrilateral elements with both displacement and pore pressure degrees of freedom enable the model to track changes in pore pressure and effective stress during the earthquake excitation. A Lysmer-Kuhlemeyer (1969) dashpot is utilized to account for the finite rigidity of the underlying elastic medium.

Parametric analysis with OpenSeesMP using three different input motions
Upload files in My Data

Data Depot

Research Workbench - Learning Center - NHERI Facilities - NHERI Community - About - Help

Welcome, Maria Giovanna!

Learn how to Start Using DesignSafe
Browse the Data Depot's Published Data Sets
Join the conversation in DesignSafe’s Slack Channel
Learn more about NHERI, the NCO & DesignSafe
NHERI Five-Year Science Plan

Hurricane Michael Barreling Toward Florida Gulf Coast

Hurricane Michael will make landfall mid-day Wednesday. Oct 10 with life threatening storm surge forecasted up to 12 feet, heavy rainfall up to 12 inches and damaging winds. Researchers from the Florida Coastal Monitoring Program are heading into the field ahead of the storm to set up two 15 meter weather stations.

READ MORE IN THE NEWSROOM
Upload files in My Data
OpenSeesMP analysis: *additional files needed*

Additional files for the parallel analysis!!!
OpenSeesMP analysis: additional files needed

Additional files for the parallel analysis!!!
OpenSeesMP analysis: *additional files needed*

Additional files for the parallel analysis!!!
OpenSeesMP analysis: *Run the analysis*
Each Node assigns to the job 64 cores!
Each Node assigns to the job 64 cores!
OpenSeesMP analysis: *Check job status*
OpenSeesMP analysis: Check job result

File Preview: openseesmp-3802335650284311016-242ac11b-0001-007.err

File name: openseesmp-3802335650284311016-242ac11b-0001-007.err
File size: 11.6 kB
Last modified: 10/24/18 7:53 PM

Modules based on Lua: Version 7.8.3 2018-09-11 13:28 -06:00
by Robert McIay mclay@acc.utexas.edu

module [options] sub-command [args ...]

Help sub-commands:
------------------
help              prints this message
help help         print help message from module(s)

Loading/Unloading sub-commands:
------------------------------
load | add module [...]  load module(s)
try-load | try-add module [...]  Add module(s), do not complain if not found
del | unload module [...]  Remove module(s), do not complain if not found
swap | sw | switch m1 m2  unload m1 and load m2
purge             unload all modules
refresh           reload aliases from current list of modules
update            reload all currently loaded modules.

Listing / Searching sub-commands:
---------------------------------
list               List loaded modules
list s1 s2 ...    List loaded modules that match the pattern
avail | av          List available modules
avail | av string    List available modules that contain "string"
spider            List all possible modules
spider module     List all possible version of that module
spider string     List all module that contain the "string"

Close
### OpenSeesMP analysis: Check job result

**Find in My Data**

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Last modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>accsi_disp_Plot_parallel.m</td>
<td>3.6 kB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>accelerationMotionKobe1985_KobeUniversity.out</td>
<td>796.3 kB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>accelerationMotionLomaPrieta1989_GirroyArray1.out</td>
<td>1.9 MB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>accelerationMotionNorthridge1994_UCLA.out</td>
<td>762.0 kB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>displacementMotionKobe1985_KobeUniversity.out</td>
<td>817.7 kB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>displacementMotionLomaPrieta1989_GirroyArray1.out</td>
<td>2.0 MB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>displacementMotionNorthridge1994_UCLA.out</td>
<td>763.9 kB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>elementInfo.dat</td>
<td>1.0 kB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>freeFieldEffective_flav.a.msh</td>
<td>3.3 kB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>freeFieldEffective.tcl</td>
<td>17.5 kB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>GaccelerationMotionKobe1985_KobeUniversity.out</td>
<td>592.1 kB</td>
<td>10/24/18 7:53 PM</td>
</tr>
<tr>
<td>GaccelerationMotionLomaPrieta1989_GirroyArray1.out</td>
<td>592.1 kB</td>
<td>10/24/18 7:53 PM</td>
</tr>
</tbody>
</table>
Post-processing alternatives

- Work on the Cloud using the tools available on DesignSafe (Data Processing Tab):
  - Jupyter Notebook;
  - Matlab.

- Download all the output and post-process data locally.
Post-processing alternatives

- Work on the Cloud using the tools available on DesignSafe (Data Processing Tab):
  - Jupyter Notebook;
  - Matlab.

- Download all the output and post-process data locally.
Post-processing alternatives: Jupyter Notebook

DATA DEPOT BROWSER

Select data source

File name | Size |
---------|------|
jupyter_checkpoints | 4 kB |
Trash | 16 kB |
Trash-556891 | 4 kB |
archive | 4 kB |
Comparison_stampedv2 | 4 kB |
Examples | 4 kB |
Fixed_base | 4 kB |
real_base_Ureal_params | 4 kB |
Freefield/velocity_field | 4 kB |
Fixed chlor | 4 kB |

DESIGNSAFE - CI JUPYTER NOTEBOOK

The Jupyter Notebook is a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.

NOTE: This Jupyter instance will terminate after being idle for 3 days.
Post-processing alternatives: *Jupyter Notebook – OpenSeesMP* (Community Data/Workspace Applications Examples / OpenSees/freeFieldEffectiveJupyter_postprocessing_parallel.ipynb)
Post-processing alternatives: *Jupyter Notebook – OpenSeesMP* (Community Data/Workspace Applications Examples / OpenSees/freeFieldEffectiveJupyter_postprocessing_parallel.ipynb)

---

**Free Field Analysis Example**

This example shows how to postprocess the output results of OpenSeesMP analyses using Python scripts.

The site response analysis discussed in this article is for a site profile consisting of a 10 m thick layer of loose sand ($D_r = 0.30$) above a 20 m thick layer of medium dense sand ($D_r = 0.50$). A schematic representation of the analyzed site profile is shown in Fig. 1. The site soil profile is underlain by an elastic half-space which represents the finite rigidity of an underlying medium such as bedrock. The groundwater table is located at a depth of 5 m, therefore, saturated unit weights are used for the soil below this point and effective stress analysis is considered through the use of nine-node quadrilateral elements which are able to simulate fluid soil coupling.

The results are presented for each ground motion in terms of:

a) Time history of acceleration at the surface and corresponding response spectra;

b) Profiles of maximum displacement, PGA, maximum shear strain, stress ratio, and stress strain peak for a point near the center of the liquefiable zone;

c) Evolution of pore water pressure for a point near the center of the liquefiable zone.

---

**Postprocess Results**

```
# The raw code for this Python notebook is by default hidden for easier reading. To toggle on/off the raw code, click here.

Import python libraries

'/users/micro/Research/ResearchGuide/webinar'

Change directory

'/users/micro/Research/ResearchGuide/webinar/openSeesMP_postproc'
```
Post-processing alternatives: *Jupyter Notebook – OpenSeesMP* (Community Data/Workspace Applications Examples / OpenSees/freeFieldEffectiveJupyter_postprocessing_parallel.ipynb)
Post-processing alternatives: *Jupyter Notebook – OpenSeesMP* (Community Data/Workspace Applications Examples / OpenSees/freeFieldEffectiveJupyter_postprocessing_parallel.ipynb)
Post-processing alternatives: *Jupyter Notebook – OpenSeesMP* (Community Data/Workspace Applications Examples / OpenSees/freeFieldEffectiveJupyter_postprocessing_parallel.ipynb)
Post-processing alternatives

- Work on the Cloud using the tools available on DesignSafe (Data Processing Tab):
  - Jupyter Notebook;
  - Matlab.

- Download all the output and post-process data locally.
Post-processing alternatives: *Matlab*

You have to submit a ticket to activate your license on DesignSafe.
Recommended for standard post-processing analyses.

Post-processing alternatives: *Matlab*

Run an interactive Matlab 2016a session on a virtual machine. Work directly on your files rather than needing to copy them to and from Stampede.

Run an interactive MATLAB 2017b session on Stampede2.
Recommended for standard post-processing analyses.

**Post-processing alternatives: **Matlab

**WORKSPACE**

- Run an interactive MATLAB 2017b session on Stampede2.
- Run an interactive Matlab 2016a session on a virtual machine. Work directly on your files rather than needing to copy them to and from Stampede.

**RUN MATLAB** ver. 0.1

Run an interactive Matlab 2016a session on a virtual machine. Work directly on your files rather than needing to copy them to and from Stampede.

**Inputs**

- **Desktop Resolution**
  - 1280x800
  - Set the desktop screen size for your visualization session.

**Job details**

- **Maximum job runtime**
  - 01:00:00
  - In HH:MM:SS format. The maximum time you expect this job to run for. After this amount of time your job will be killed by the job scheduler. Shorter run times result in shorter queue wait times. Maximum possible time is 48:00:00 (48 hours).

- **Job name**
  - Matlab
  - A recognizable name for this job.

**Job output archive location (optional)**

- Specify a location where the job output should be archived. By default, job output will be archived at: `<username>/archive/jobs/YYYY-MM-DD/$JOB_NAME-$JOB_ID`.
Post-processing alternatives: Matlab
Post-processing alternatives: Matlab – OpenSeesMP (Community Data/Workspace Applications Examples / OpenSees/ OpenSeesMP /PostProcessing_matlab_parallel.m)
Post-processing alternatives: *Matlab – OpenSeesMP* (Community Data/Workspace Applications Examples / OpenSees/OpenSeesMP/PostProcessing_matlab_parallel.m)
OpenSees & DesignSafe: OpenSeesMP

...Questions?

Maria Giovanna Durante, Ph.D.
Postdoctoral Research Fellow
University of Texas at Austin
mgdurante@utexas.edu