**Documentation: Regional Reinforced Concrete Frame Simulations**

By: Maha Kenawy

University of Nevada, Reno

Updated: November 24, 2020

This repository includes scripts that generate structural designs and simulation models of reinforced concrete (RC) special moment frames, and conduct multiple nonlinear time history simulations on a given frame either locally (in series) or remotely on parallel computers.

The general steps to use this computational tool are:

1. Use the design notebooks to design RC buildings (currently the design notebooks are tuned to special moment frame designs according to ASCE 7-16 and RC member checks according to ACI 318-14)
2. Populate the building input excel sheets using the design building and structural member output from the design notebooks.
3. Run the simulation scripts according to the following instructions in this document. The MATLAB-based simulation scripts will read the design excel sheets, generate the building simulation models, and run the analysis using the Opensees framework.

Note: The following Opensees files are copied/modified from Tcl scripts available in the Opensees examples manual: LibAnalysisDynamicParameters.tcl; LibUnits.tcl; LibUnitsNmm.tcl; rotSpring2DModIKModel.tcl. These files are used in the simulations workflow described in this repository.

**RC Frame Ground Motion Simulations**

**Instructions**

This following instructions describe the process for running simulations of reinforced concrete moment frames subjected to ground motion acceleration series using the scripts provided in this repository. The first set of instructions covers running in-series response history dynamic simulations of any RC frame subjected to multiple ground motions; the second set covers running the same type of simulations remotely in parallel on NERSC’s CORI computer. Running the scripts in this repository requires a MATLAB license and access to the structural analysis platform Opensees.

**Instructions for running multiple GMs in the EQSIM scenario format**

FOR RUNNING MULTIPLE SIMULATIONS

The following files must be in the directory for creating the model, running the simulation and processing the output (this is run in /multipleGM\_simulations)

* Opensees.exe
* Directory with ground motion .data files (If .txt files with a single component are available, skip step 1)
* Excel sheets with building and structural member information
* MATLAB scripts for preparing the ground motions: (prepareGMs.m) – unless the ground motions are already prepared in .txt format.
* MATLAB scripts for generating the structural model and other simulations input (main\_multipleGMs.m, generateModel.m, readBuildingInfo.m, colSectionInfo.m, beamSectionInfo.m, framInfo.m, writeFrameInfo.m, memParameters.m, readcoldata.m, readIMKParameters.m, readbeamdata.m, beamRftGeometry.m, colRftGeometry.m, flexuralStrength.m, shearStrength.m, IMKCalibration.m, processIMKParameters.m)
* Tcl scripts for running the simulations

(multipleGMs.tcl, runGMs.tcl, LumpedModel.tcl, rotSpring2DModIKModel.tcl, LibUnitsNmm.tcl, LibUnits.tcl, LibAnalysisDynamicParameters.tcl)

* MATLAB scripts for getting ground motion intensity measures:

(specDriver.m, integrate.m, spectrum.m, parseDATA\_mk.m, spectra\_T1)

* MATLAB scripts for getting the pulse classification: *classification scripts by Jack Baker and collaborators.*
* MATLAB scripts for postprocessing

(output\_multipleGMs.m, extractMaxDrift.m, plot\_drift\_envelope\_vs\_distance.m, plot\_drift\_vs\_distance\_color.m, hazMap.m, generic\_map.m, plot\_drift\_IMs.m, plot\_drift\_envelopes.m, plotLimits.m, bar\_ticks)

* **STEP 1: Prepare the ground motion files** (this needs to be done once for each ground motion set)
  + Open prepareGMs.m (this file will generate the ground motion time series needed for simulation - if the GMfiles already exist, skip this step)
    - set name of GMset – this is the name of the directory containing the ground motions .data files and should be in the same directory as prepareGMs.m
    - set the component to FN or FP
  + run prepareGMs.m
  + output:
    - GMfiles\_GMsetComp - contains the records for all GMs in .txt format with only the selected acceleration component present.
    - numPts.txt – contains number of points in each record
    - timeincr.txt – contains the time increment of each record
    - pathToGMs.tcl – contains the path to the ground motion files, which will be called by the analysis script runGMs.tcl
* **STEP 2: Generate the structural model**
  + main\_multipleGMs.m (this script generates the actual building model, and spectral ordinates for the building)
    - Set the building name and fundamental period; excel files with the input info for the building and its members must exist first.
    - Set the GMset
    - Set the component (FN or FP)
    - Set a unique name for the simulation set (optional)
  + Run main\_multipleGMs.m
  + Output:
    - frameInfo.tcl – contains the building information
    - memParameters.tcl – contains the structural member information
    - a .mat data file for each column and beam
    - a .mat data file for saving the building info
    - a .mat file for saving SA(T1) and SV(T1)
    - outputDir.tcl – contains the name of the data directory for saving the simulations output, which will be called by runGMs.tcl
* **STEP 3: Run nonlinear response history simulations on Opensees**
  + Run Opensees and source multipleGMs.tcl
  + Output: An output directory named driftOutput\_buildingName\_GMsetComponentSimulation – contains all the building and inter-story drift histories
* **STEP 4: Process output and create plots**
  + output\_multipleGMs.m – this output processing script is separated from the workflow to facilitate extraction of results from previous simulations – the script will call other scripts to generate the SA, SV spectra and PGV, and classify pulse ground motions (if they don’t exist already)
    - Edit user input section appropriately (set buildingName, structPeriod, GMset, component, sim)
    - Set ground motion subset for plots, if desired. The options are:
      * ‘near’: creates plots for GM stations within 10 km normal to fault
      * ‘far’: creates plots for GM stations beyond 10 km normal to fault
      * ‘’: default – creates plots for all stations
    - Set newSim = 1 if extracting results of new simulations; set newSim = 0 if using results extracted and saved previously.
    - Edit the number of x and y ground motion stations and location of the hypocenter if needed
    - Edit plot\_ind to specify indices of specific stations at which drift envelopes will be plotted.
  + Run output\_multipleGMs.m
  + Output:
    - maxDrifts\_buildingName\_GMsetComp.mat – contains maximum drifts
    - GMsetComponent\_SA\_SV\_PGV.mat – contains acceleration and velocity response spectra, and peak ground velocities
    - Pulse\_class\_GMset – contains pulse classification of all ground motions
    - Various plots

**Instructions for running multiple GMs in the EQSIM scenario format on CORI**

STEP 1 is performed in $CSCRATCH directory: The following files must be in that directory for generating the needed ground motion files:

* unzipped directory with ground motion .data files
* MATLAB scripts for preparing the ground motions (prepareGMs.m,) – unless the ground motions are already prepared in .txt format.
* **STEP 1: Prepare the ground motion files** (this step needs to be done once for each ground motion set)
  + Open prepareGMs.m (this file will generate the ground motion time series needed for simulation - if the GMfiles already exist, skip step 1)
    - set name of GMset – this is the name of the directory containing the ground motions .data files and should be in the same directory as prepareGMs.m
    - set the component to FN or FP
  + Open matlab:

salloc -q interactive -N 1 -c 32 -C haswell -t 30:00

module load matlab

matlab

* + prepareGMs
  + output:
    - GMfiles\_GMsetComponent - contains the records for all GMs in .txt format with only the selected acceleration component present.
    - numPts.txt – contains number of points in each record
    - timeincr.txt – contains the time increment of each record
    - pathToGMs.tcl – contains the path to the ground motion files, which will be called by the analysis script runGMs.tcl

STEP 2 is performed on CORI in the $CSCRATCH directory: The following files should be in that directory for running the simulations:

* The directory containing the ground motion files, named as: GMfiles\_GMsetComponent – this is generated in the previous step
* The time increment and number of step files: (timeincr.txt, numPts.txt)
* Tcl scripts containing the building properties

(frameInfo3st.tcl, memParameters3st.tcl, frameInfo12st.tcl, memParameters12st.tcl)

* Tcl scripts for running the simulations

(runGM.tcl, LumpedModel.tcl, rotSpring2DModIKModel.tcl, LibUnitsNmm.tcl, LibUnits.tcl, LibAnalysisDynamicParameters.tcl)

* run.sh – calls opensees
* gen.sh – creates the parallel simulation files and the simulations task list
* batch.sh – allocates resources on CORI
* createSimFiles.py – python file which creates the parallel simulation files
* user\_input.py – to specify the building and GMset and component
* **STEP 2: run the simulations**
* edit user\_input.py - specify the buildingName, GMset, component, sim
* in command line, type *./gen.sh* - this generate the inputlist.txt (this also generates parallelized files and other needed input by running createSimFiles.py)
* change the number of nodes if needed in batch.sh
* *sbatch batch.sh* – submits the job which includes running the parallel simulations

STEP 3 is performed on CORI in the $CSCRATCH directory: The following files should be in that directory for running the simulations:

* Matlab files to extract the drift output and save the maximum drifts: (output\_CORI.m, extractMaxDrift.m (*a function*))
* **STEP 3: extract the output**
  + Load and run the matlab postprocessing script to extract the output files

module load matlab

srun -n 1 -c 32 matlab -nodisplay -r < output\_CORI.m -logfile output\_CORI.log

* download maxDrifts\_buildingName\_GMsetComponentSim.mat to /Opensees\_CORI
* Output:
* maxDrifts\_buildingName\_GMsetComponentSim.mat – contains maximum building and inter-story drifts

STEP 4 is performed locally in /CORI\_simulations, and requires the following files:

* directory with all ground motion .data files (for generating the spectra and pulse classification)
* .mat file containing maximum drifts: maxDrifts\_buildingName\_GMsetComponentSim – downloaded from $CSCRATCH
* MATLAB scripts for post-processing:
* (output\_multipleGMs.m, plot\_drift\_envelope\_vs\_distance.m, plot\_drift\_vs\_distance\_color.m, hazMap.m, generic\_map.m, plot\_drift\_IMs.m, plot\_drift\_envelopes.m, plotLimits.m, bar\_ticks)
* MATLAB scripts for getting ground motion intensity measures:

(specDriver.m, integrate.m, spectrum.m, parseDATA\_mk.m, spectra\_T1)

* MATLAB scripts for getting the pulse classification: *classification scripts by Jack Baker and collaborators.*
* **STEP 4: Process output and create plots** (same as STEP 4 in running simulations locally)
* output\_multipleGMs.m
  + Edit user input section appropriately (set buildingName, structPeriod, GMset, component, sim)
  + Set ground motion subset for plots, if desired. The options are:
    - ‘near’: creates plots for GM stations within 10 km normal to fault
    - ‘far’: creates plots for GM stations beyond 10 km normal to fault
    - ‘’: default – creates plots for all stations
  + Set newSim = 0 (because results have already been extracted on CORI)
  + Edit the number of x and y ground motion stations and location of the hypocenter if needed
  + Edit plot\_ind to specify indices of specific stations at which drift envelopes will be plotted.
* Run output\_multipleGMs.m

Note: to generate new buildings for CORI simulations (locally in CORI\_simulations):

* Set the buildingName in generateModel\_CORI.m – must match name on design excel sheets in the same directory
* Run generateModel\_CORI.m
* Move frameinfoBuildingName.tcl and memParametersBuildingName.tcl to $CSCRATCH/Opensees\_models
* Output:
  + frameinfoBuildingName.tcl – defines building properties
  + memParametersBuildingName.tcl – defines structural member properties
  + buildingName\_info.mat – saves some building properties for output processing