SimCenter NHERI
Center for Computational Modeling and Simulation

EE-UQ

The Earthquake Engineering with Uncertainty Quantification (EE-UQ) Application

Frank McKenna, Wael Elhaddad, Chaofeng Wang and Michael Gardner

NHERI SimCenter, UC Berkeley

Version 1.2.3 Frank, let me know the current version.

March 18, 2019

Licenses and Copyright Notices

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1612843. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Contents

List of Figures

List of Tables

Tool Audience

1) Researchers/Practitioners trying to predict the response of a structure to earthquakes.

About

This open-source research application (https://github.com/NHERI-SimCenter/EE-UQ) provides an application researchers can use to predict the response of a building subjected to earthquake events. The application is focused on quantifying the uncertainties in the predicted response, given the that the properties of the buildings and the earthquake events are not known exactly, and that the simulation software and the user make simplifying assumptions in the numerical modeling of that structure. In the application, the user is required to characterize the uncertainties in the input. The application will after utilizing the selected sampling method, will provide information that characterizes the uncertainties in the response measures. The computations to make these determinations can be prohibitively expensive. To overcome this impediment the user has the option to perform the computations on the Stampede2 supercomputer. Stampede2 is located at the Texas Advanced Computing Center and made available to the user through NHERI DesignSafe, the cyberinfrastructure provider for the distributed NSF funded Natural Hazards in Engineering Research Infrastructure (NHERI) facility.

The computations are performed in a workflow application. That is, the numerical simulations are actually performed by a number of different applications. The EE-UQ backend software runs these different applications for the user, taking the outputs from some programs and providing them as inputs to others. The design of the EE-UQ application is such that researchers are able to modify the backend application to utilize their own application in the workflow computations. This will ensure researchers are not limited to using the default applications we provide and will be enthused to provide their own applications for others to use.

This is Version 1.0 of the tool and as such is limited in scope. Researchers are encouraged to comment on what additional features and applications they would like to see in this application. If you want it, chances are many of your colleagues also would benefit from it.

Installation Instructions

1. The SimCenter is not recognized as either an Apple or a Windows software vendor. As a consequence, the applications are not recognized by the operating system as being signed. As a consequence It is required to start the software in a special way when they invoke it for first time.

   a. Mac: operating system the application must be started by right clicking (ctrl-click) on it.

2. To run the applications locally the external applications OpenSees and Dakota must both be installed and set up to run on your operating system. When both are installed further steps are required so that the EE-UQ application fids them:

   a. Mac: in your home directory there should be a file .bashrc. You need to add the paths to the OpenSees and Dakota executables to your PATH environment variable. For example, I placed the OpenSees executable in my home directory in the opensees-3.0.0/bin and I installed Dakota also in my home directory. I added the following 2 lines in my .bashrc file:

   export PATH=$HOME/opensees-3.0.0/bin:$PATH

   export PATH=$HOME/dakota-6.7.0/bin:$PATH

   b. Windows (Assuming Windows 10): The windows version also requires that the Visual Studio 2017 C++ runtime be installed. Chances are some software you are using has already done this for you. If this is not the case you will be required to install it from the windows website.
   https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads

3. Python

Usage

It is the UI that the user interacts with. The interface itself, as shown in Figure 4.1, provides the user with a selection of input widgets in which, for the application of choosing, the user provides the information needed by the application for that part of the workflow. As seen in Figure 4.1, on the left-hand side of the UI the user is presented with a selection of different applications used in the workflow. The input panel selection area consists of a number of choices:

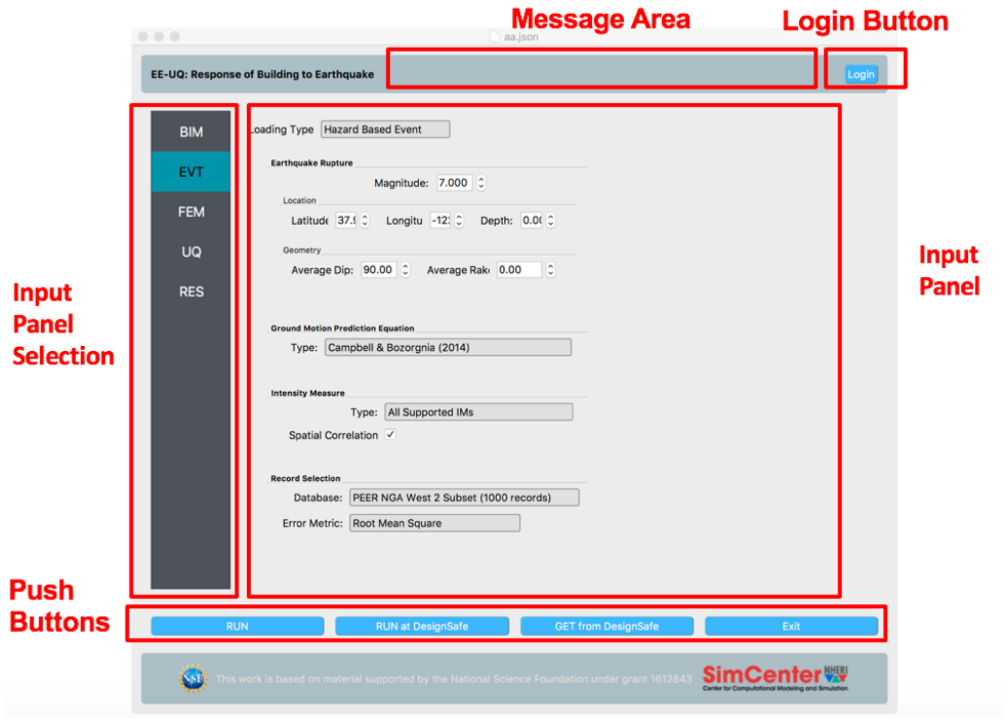
1. BIM: structure information, description and model generation.
2. EVT: earthquake motions specification
3. FEM: finite element application
4. UQ: Uncertainty quantification: specification of the random variable parameters and UQ analysis options
5. RES: results output.

Selecting any of these will change the input panel. It is here that the user selects the actual application to be used and provides input for that application.

Also in the UI is an area where status and error messages will be reported to you, a button to press to login (login is required only if you want to run the job at DesignSafe) and an area with a number of push buttons:

1. RUN – to run the simulation of the user's desktop machine.
2. RUN at DesignSafe – to process the information, and send to DesignSafe where the job will be run on a supercomputer and results stored in your DesignSafe jobs folder.
3. GET from DesignSafe – to obtain from DesignSafe your list of jobs and select from that list a job to download.
4. Exit: to exit the application.

The Screens presented to user when the first 3 of these buttons will be discussed in 2.5.


Message Area
Login Button
Input Panel
Input Panel Selection
Push Buttons
aa.json
EE-UQ: Response of Building to Earthquake
Login
BIM
EVT
FEM
UQ
RES
RUN
RUN at DesignSafe
GET from DesignSafe
Exit
This work is based on material supported by the National Science Foundation under grant 1612843
SimCenter NHERI
Center for Computational Modeling and Simulation

Loading Type: Hazard Based Event

Earthquake Rupture

Magnitude: 7.000

Location

Latitude: 37.9 Longitu: -12: Depth: 0.0(

Geometry

Average Dip: 90.00 Average Rak: 0.00

Ground Motion Prediction Equation

Type: Campbell & Bozorgnia (2014)

Intensity Measure

Type: All Supported IMs

Spatial Correlation ✓

Record Selection

Database: PEER NGA West 2 Subset (1000 records)

Error Metric: Root Mean Square

Figure 4.1: UI

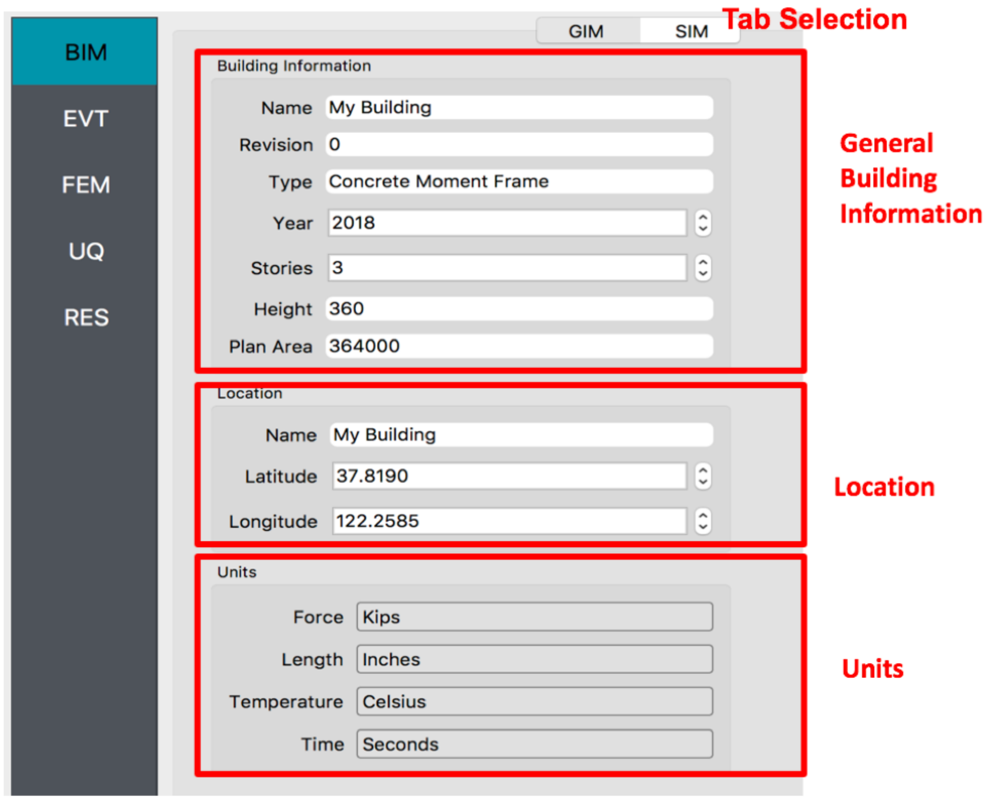
4.1 BIM

The user in the BIM panels defines the building. There are tabbed panels for this:

1. GIM: general Information about the building. The panel presents 3 separate frames. A frame in which the user as shown in Figure 4.2, defines information about the building. This includes the building type, name, year of construction, overall height, plan area, geographic location (needed for certain event types), etc. This information should be provided, as certain applications in the workflow that exist or are planned will need this information.

   In the other frames the user provides the building location and in the last panel the units. this may be needed in applications further down the workflow to ensure units are consistent throughout the applications.

2. SIM: This panel is where the user defines the building. There is a drop-down menu that the user will use to select from a number of different modeling options. At present there is a single option, OpenSees. The panel that presents is as shown in Figure 4.3. The user specifies the main script that contains the building model, a list of nodes that define a column line of interest for which the responses will be determined and an entry for the dimension of the model. Note the column nodes should be in order



Building Information

Name	My Building
Revision	0
Type	Concrete Moment Frame
Year	2018
Stories	3
Height	360
Plan Area	364000

Location

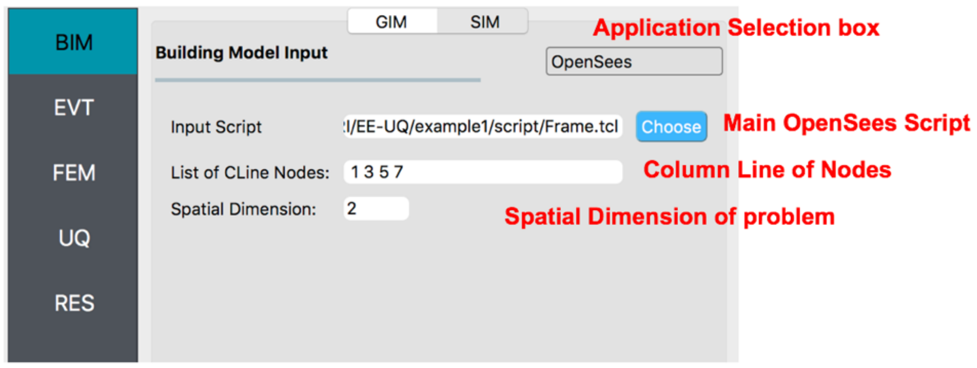
Name	My Building
Latitude	37.8190
Longitude	122.2585

Units

Force	Kips
Length	Inches
Temperature	Celsius
Time	Seconds

Figure 4.2: BIM

from ground floor through to roof (as drifts are calculated based on this). The spatial dimension is used to align the ground motion components in the EVENT file.



Building Model Input

OpenSees

Input Script	:I/EE-UQ/example1/script/Frame.tcl
List of CLine Nodes:	1 3 5 7
Spatial Dimension:	2

Figure 4.3: GIM

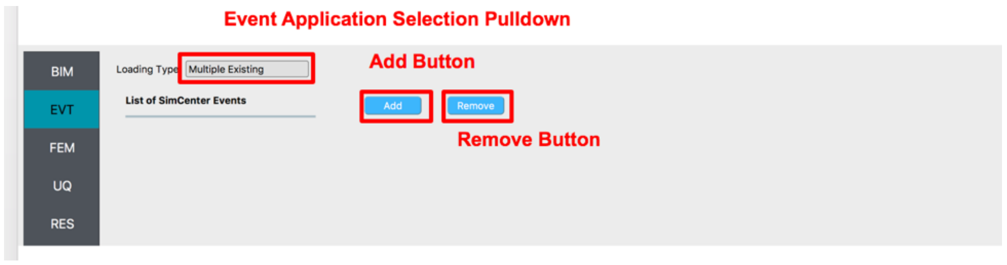
In the worked for version 1.1+ include code to describe the building in most general terms and allow the user to select from expert or machine learning systems that produce the building model file.

4.2 EVT (Event)

The event panel presents the user with a drop-down menu with a list of available applications. Event applications are applications that given the building, and user supplied data to the specific applications input panel will generate a list of events for the building. There are a number of options:

Multiple Event

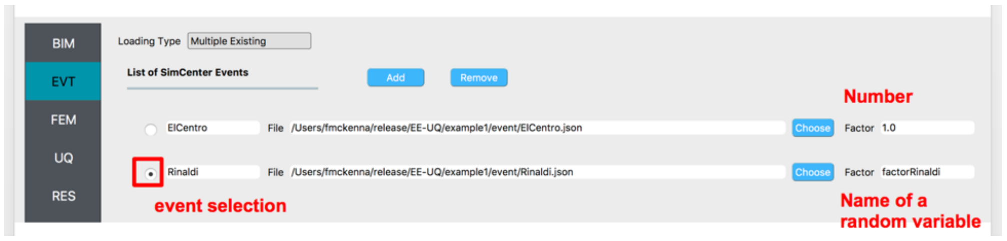
This is provided for the user to specify multiple existing SimCenter Event files. If more than one event is provided it is done to provide the UQ engine with a discrete set of events to choose from. It is not done with the intention of specifying that one event follows another. The panel presented initially to the user is as shown in Figure 4.4.



Loading Type: Multiple Existing

Figure 4.4: EVT

To add a new event, the user presses the Add button. This adds an event to the panel. Pressing the button multiple times will keep adding events to the panel. Figure 4.5 shows the state after the button has been pressed twice, and data entered for the ElCentro and Rinaldi Events.



Loading Type: Multiple Existing

ElCentro File /Users/fmckenna/release/EE-UQ/example1/event/ElCentro.json Factor 1.0

Rinaldi File /Users/fmckenna/release/EE-UQ/example1/event/Rinaldi.json Factor factorRinaldi

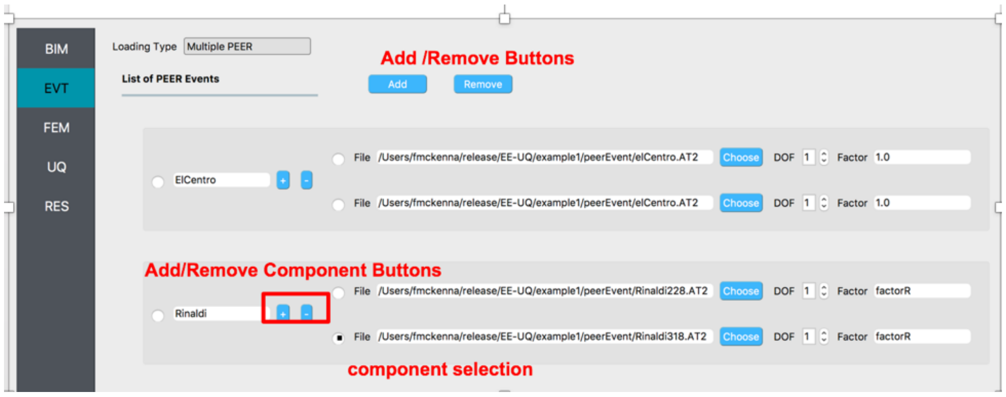
Figure 4.5: Adding new event

The user can enter the full path manually to the file or use the choose button, which brings up your typical file search screen. By default, a scaling factor of 1.0 is assigned to the event. The user can change this to another real value (AT PRESENT DO NOT USE INTEGER) or the user has the option of defining this to be a random variable by entering a name as shown for the second event. Note that this variable name must not start with a number, or contain any spaces or special characters, i.e. -, +,..

The Remove button is pressed to remove events. Once pressed it removes all button whose event selection box is highlighted.

Multiple PEER Event

This is provided for the user to specify multiple existing PEER (http://peer.berkeley.edu) ground motion files. For PEER events the user is required to specify the individual components for the EVENTS. The Add/Remove buttons at the top are to create and remove an event, as per 2.2.1. For the PEER events the user specifies components acting in the individual degree-of-freedom directions. The + and – add and remove components with the remove removing all components selected. Each component in a PEER event can have their own d=scale factor, again a number or a random variable.



Loading Type: Multiple PEER

ElCentro
File /Users/fmckenna/release/EE-UQ/example1/peerEvent/elCentro.AT2 DOF 1 Factor 1.0
File /Users/fmckenna/release/EE-UQ/example1/peerEvent/elCentro.AT2 DOF 1 Factor 1.0

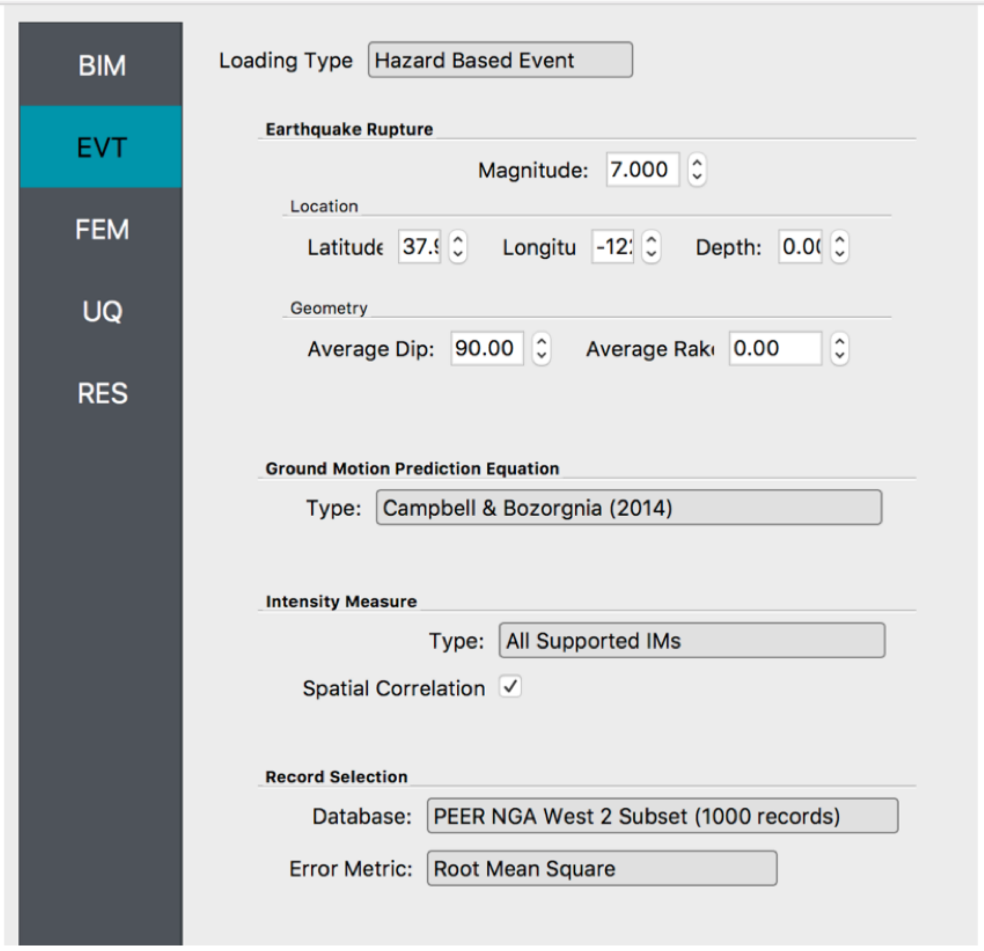
Rinaldi
File /Users/fmckenna/release/EE-UQ/example1/peerEvent/Rinaldi228.AT2 DOF 1 Factor factorR
File /Users/fmckenna/release/EE-UQ/example1/peerEvent/Rinaldi318.AT2 DOF 1 Factor factorR

Figure 4.6: PEER event

Hazard Based Event

The panel for this event application is as shown in Figure 4.7. This application implements a scenario-based (deterministic) seismic event. In this panel the user specifies an earthquake rupture (location, geometry and magnitude), a ground motion prediction equation, a record selection database and the intensity measure used for record selection. In the backend, this application relies on three other applications to perform seismic hazard analysis, intensity measures simulation (to create a simulated target spectrum), and ground motion record selection/scaling. Users interested in learning about those applications are referred to the documentation of the (SimCenter ground motion utilities).

s3hark shows up here



Loading Type: Hazard Based Event

Earthquake Rupture

Magnitude: 7.000

Location

Latitude: 37.9 Longitu: -12 Depth: 0.0(

Geometry

Average Dip: 90.00 Average Rak: 0.00

Ground Motion Prediction Equation

Type: Campbell & Bozorgnia (2014)

Intensity Measure

Type: All Supported IMs

Spatial Correlation ✓

Record Selection

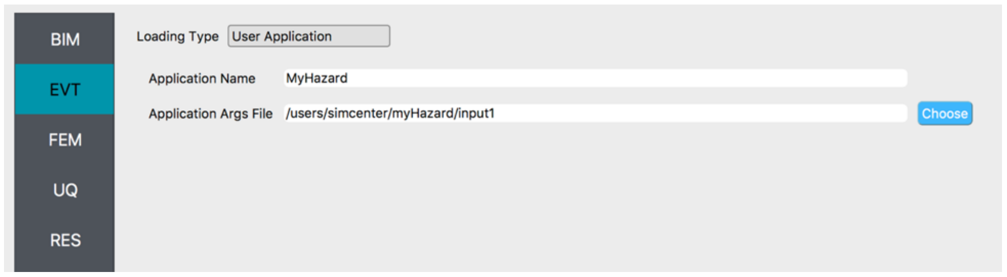
Database: PEER NGA West 2 Subset (1000 records)

Error Metric: Root Mean Square

Figure 4.7: Hazard based event

User Application

The final selection option is a user specific application. The user specifies the application name and the input file containing the specific input information needed by the application when it is running in the backend. As will be discussed, the user is also required when they use an additional application not provided, to edit the tools registry file. Here they must include a new event application with this same name and the location where that application can be found relative to the tools application directory. If running on DesignSafe, that application must of course be built and available on the Stampeded2 supercomputer. NOTE that given how DesignSafe runs the applications through Agave, this applications file permissions must be world readable and executable (as when user running their application through DesignSafe and Agave, they are not running as themselves!)

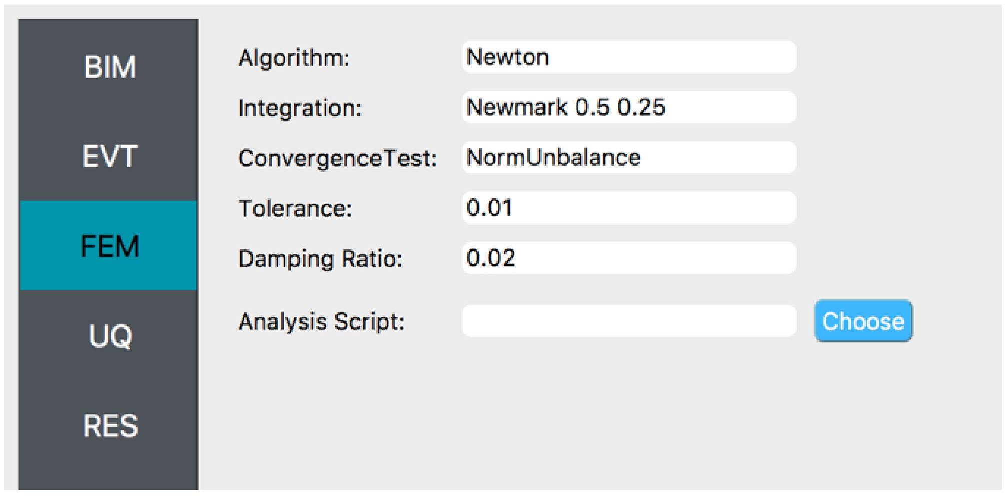


Loading Type: User Application
Application Name: MyHazard
Application Args File: /users/simcenter/myHazard/input1

Figure 4.8: User defined event

4.3 FEM

The FEM panel is intended to present users with a selection of FEM applications that will take a building model generated by the BIM application and the EVENT from the event application and perform a deterministic simulation. At present there is only one application available, OpenSees and there is no application selection box. That will be modified in Version 1.1.0 to allow user to provide their own simulation application. This is not the standard OpenSees executable, but consists of a pre- and post-processor to take the BIM and EVENT file and use OpenSees to determine the response, returning these responses in an EDP. Presently the default EDPs are the relative floor displacements, total accelerations (ground motion + relative response) and inter-story drifts for column lines specified in the BIM panel.



Algorithm: Newton
Integration: Newmark 0.5 0.25
ConvergenceTest: NormUnbalance
Tolerance: 0.01
Damping Ratio: 0.02
Analysis Script:

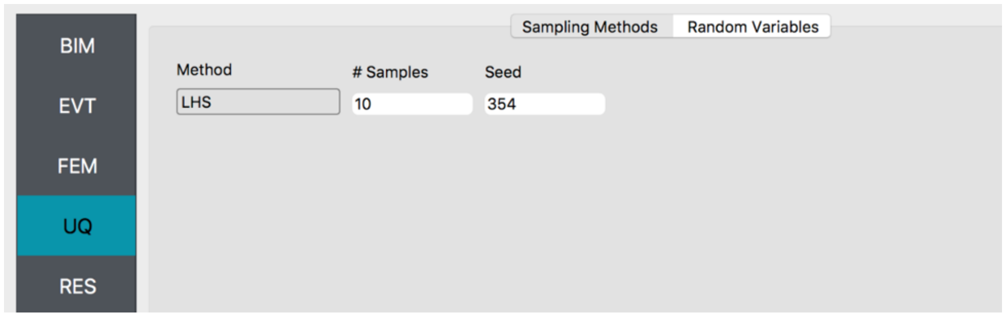
Figure 4.9: FEM

In the OpenSees FEM panel, the user specifies the algorithm, integration strategy, convergence test, tolerance and damping ratio. A default transient analysis script is run with these inputs. It is built for Version 3.0.0+ of OpenSees and uses a divide and conquer algorithm in event of a convergence failure issue. This new algorithm does not always work.

The user is also able to specify their own analysis script to run instead of the default. When chosen the variables numStep and dt that are obtained from the EVENT should be assumed to have been set by the application before the script is run and can be used in your user defined script.

4.4 UQ

Throughout the input specification the user is defining variables. Many of these variables can be specified by the user to be random variables with a distribution on their values. It is in the UQ panel that the user specifies what these distributions are. It is also here that the user specifies what UQ engine, what UQ method and other inputs are for the UQ method. The panel is split into 2 tabs: Sampling Methods and Random Variables as shown in Figure 4.10.



Method: LHS
Samples: 10
Seed: 354

Figure 4.10: UQ

Sampling Methods

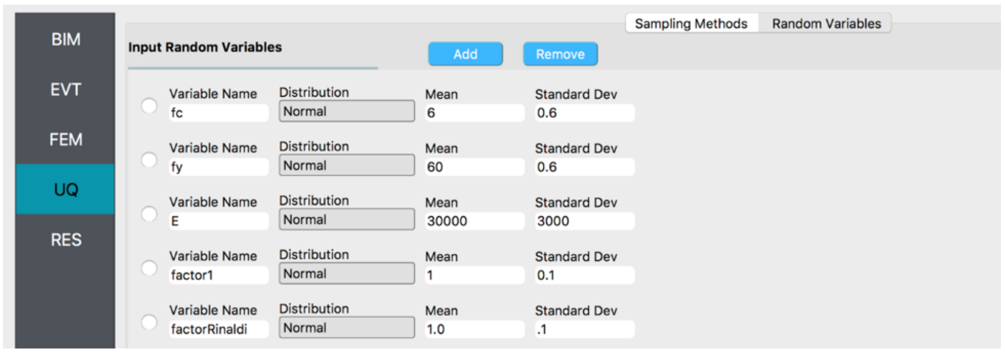
In the sampling methods the user selects the sampling method to use from the method dropdown. Currently this is limited to two options: Monte Carlo and Latin Hypercube Sampling (LHS). For the one selected, the user specifies the number of simulations to perform and the seed.

Random Variables

The Random Variable panel is where the user enters the random variables. Each random variable has a name associated with it, a distribution, and depending on the distribution a number of additional variables to specify. As with the other panels discussed, random variables can be added and removed using the added remove buttons. It should also be noted that the random variables should be created automatically when they are entered in previous screens.

Some Notes:

- A current limitation of the tool is that it does not check for multiple random variables with the same name, nor does it remove a random variable if the random variable is removed, e.g. factor in an earthquake event is set to a number or a different OpenSees file is loaded. The user in such circumstances is required to manually remove the variable from the UQ
- Also, do note that when initially created Random variables are assigned a constant type. The UQ engine will ignore any variables with such a type as they are not random.
- When using OpenSees input files, this application will read any variable set with the pset command as a random variable and create a random variable for it when the user chooses the main file. See the example in example/script/Frame.tcl directory.



Variable Name	Distribution	Mean	Standard Dev
fc	Normal	6	0.6
fy	Normal	60	0.6
E	Normal	30000	3000
factor1	Normal	1	0.1
factorRinaldi	Normal	1.0	.1

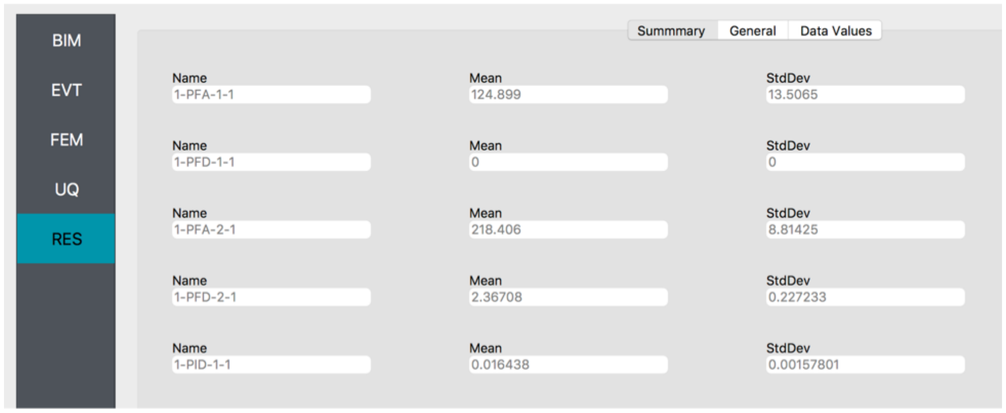
Figure 4.11: Random variables

4.5 RES

When the user hits the Run button, and assuming the results are successful. The results are presented here. A successful run or download of a job that ran successfully will result in 3 tabbed widgets being displayed in this panel. The first panel shows summary statistics: mean and stdDev values or min-max values if discrete set, i.e. multiple events

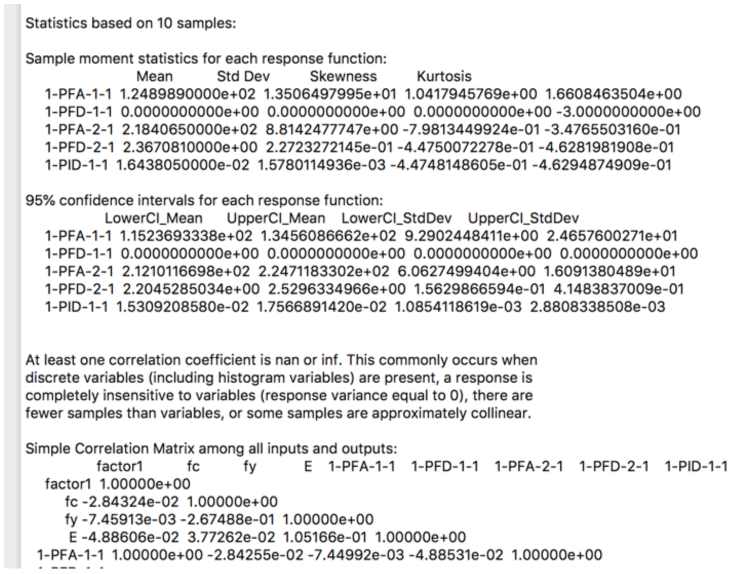
The second panel shows the summary information.

The third panel presents graphically and in tabular form the results. By selecting different columns with left and right mouse buttons in the table below the graphic, the information in the graph is changed. Selecting the left mouse button changes the Y axis, the right mouse changes the X axis. If the same column is selected using both left and right keys, the CDF and PDF is displayed. If last mouse press was with the left button, the PDF and if right the CDF.



Name	Mean	StdDev
1-PFA-1-1	124.899	13.5065
1-PFD-1-1	0	0
1-PFA-2-1	218.406	8.81425
1-PFD-2-1	2.36708	0.227233
1-PID-1-1	0.016438	0.00157801

Figure 4.12: RES

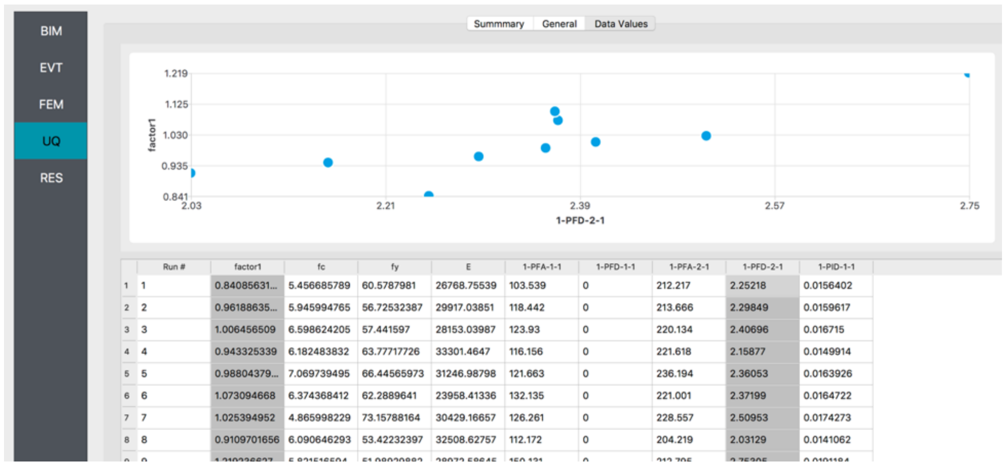


```
Statistics based on 10 samples:

Sample moment statistics for each response function:
                      Mean           Std Dev          Skewness         Kurtosis
   1-PFA-1-1  1.2489890000e+02  1.3506497995e+01  1.0417945769e+00  1.6608463504e+00
   1-PFD-1-1  0.0000000000e+00  0.0000000000e+00  0.0000000000e+00 -3.0000000000e+00
   1-PFA-2-1  2.1840650000e+02  8.8142477747e+00 -7.9813449924e-01 -3.4765503160e-01
   1-PFD-2-1  2.3670810000e+00  2.2723272145e-01 -4.4750072278e-01 -4.6281981908e-01
   1-PID-1-1  1.6438050000e-02  1.5780114936e-03 -4.4748148605e-01 -4.6294874909e-01

95% confidence intervals for each response function:
                 LowerCI_Mean      UpperCI_Mean     LowerCI_StdDev    UpperCI_StdDev
   1-PFA-1-1  1.1523693338e+02  1.3456086662e+02  9.2902448411e+00  2.4657600271e+01
   1-PFD-1-1  0.0000000000e+00  0.0000000000e+00  0.0000000000e+00  0.0000000000e+00
   1-PFA-2-1  2.1210116698e+02  2.2471183302e+02  6.0627499404e+00  1.6091380489e+01
   1-PFD-2-1  2.2045285034e+00  2.5296334966e+00  1.5629866594e-01  4.1483837009e-01
   1-PID-1-1  1.5309208580e-02  1.7566891420e-02  1.0854118619e-03  2.8808338508e-03


At least one correlation coefficient is nan or inf. This commonly occurs when
discrete variables (including histogram variables) are present, a response is
completely insensitive to variables (response variance equal to 0), there are
fewer samples than variables, or some samples are approximately collinear.

Simple Correlation Matrix among all inputs and outputs:
               factor1          fc           fy            E   1-PFA-1-1   1-PFD-1-1   1-PFA-2-1   1-PFD-2-1   1-PID-1-1
   factor1  1.00000e+00
        fc -2.84324e-02  1.00000e+00
        fy -7.45913e-03 -2.67488e-01  1.00000e+00
         E -4.88606e-02  3.77262e-02  1.05166e-01  1.00000e+00
  1-PFA-1-1  1.00000e+00 -2.84255e-02 -7.44992e-03 -4.88531e-02  1.00000e+00
```

Figure 4.13: RES General tab

As for the columns. You will see a column for each random variable the workflow came across. There may be more than you specified if the applications want the UQ engine to consider their own variables in the computation. The outputs at present are limited to:

- PFD peak relative floor displacement $1 - PFD - FLOOR_CLINE$
- PFA peak floor acceleration (relative + ground motion): $1-PFA-FLOOR-CLINE$
- PID peak inter-story drift: $1 - PID - STORY - CLINE$



	Run #	factor1	fc	fy	E	1-PFA-1-1	1-PFD-1-1	1-PFA-2-1	1-PFD-2-1	1-PID-1-1
1	1	0.84085631...	5.456685789	60.5787981	26768.75539	103.539	0	212.217	2.25218	0.0156402
2	2	0.96188635...	5.945994765	56.72532387	29917.03851	118.442	0	213.666	2.29849	0.0159617
3	3	1.006456509	6.598624205	57.441597	28153.03987	123.93	0	220.134	2.40696	0.016715
4	4	0.943325339	6.182483832	63.77717726	33301.4647	116.156	0	221.618	2.15877	0.0149914
5	5	0.98804379...	7.069739495	66.44565973	31246.98798	121.663	0	236.194	2.36053	0.0163926
6	6	1.073094668	6.374368412	62.2889641	23958.41336	132.135	0	221.001	2.37199	0.0164722
7	7	1.025394952	4.865998229	73.15788164	30429.16657	126.261	0	228.557	2.50953	0.0174273
8	8	0.9109701656	6.090646293	53.42232397	32508.62757	112.172	0	204.219	2.03129	0.0141062

Figure 4.14: UQ Data Values

4.6 Push Buttons

There are a number of buttons in the Push Button area of Figure 4.1:

RUN – to run the simulation of the user's desktop machine.

The window that pops up is as shown in Figure 4.15. There are 2 entries and a push button:

- Working Dir Location: specifies where the EE_UQ application can create a "temporary" directory called tmp. SimCenter that the application creates when the submit button is pressed. The application creates this directory, copies files to it that the application needs as a result of your input (e.g. if you are using OpenSees input script, it will to the tmp. SimCenter directory copy that script, ALL FILES IN THAT DIRECTORY AND ALL FILES IN SUBDIRECTORIES OF THAT DIRECTORY GET COPIED SO DON'T PLACE THE SCRIPT IN HOME, DOWNLOADS, DOCUMENTS, ...

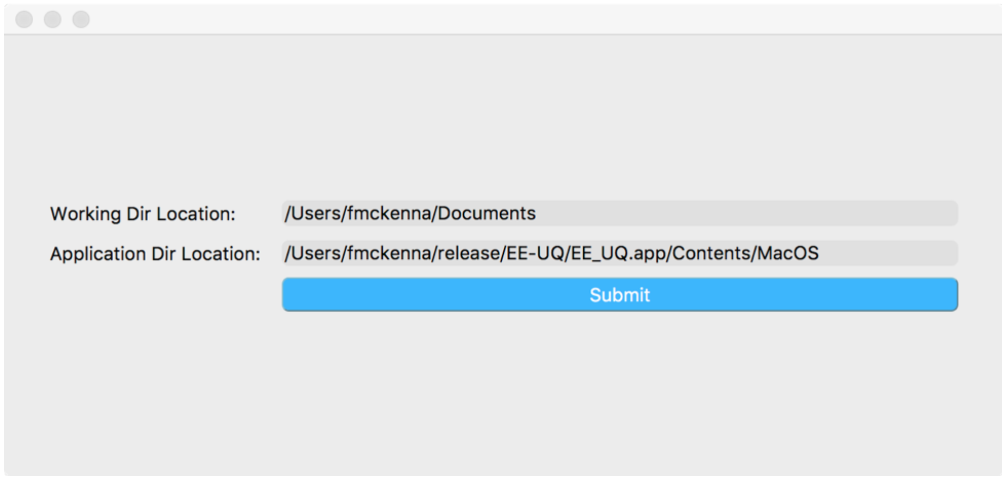

Working Dir Location:
/Users/fmckenna/Documents
Application Dir Location:
/Users/fmckenna/release/EE-UQ/EE_UQ.app/Contents/MacOS
Submit

Figure 4.15: Run button

- Application Dir Location: SHOULD NOT BE TOUCHED unless you are introducing your own applications or want to build and modify the applications provided with the tool. It is this directory the application tool looks to find the applications to run.

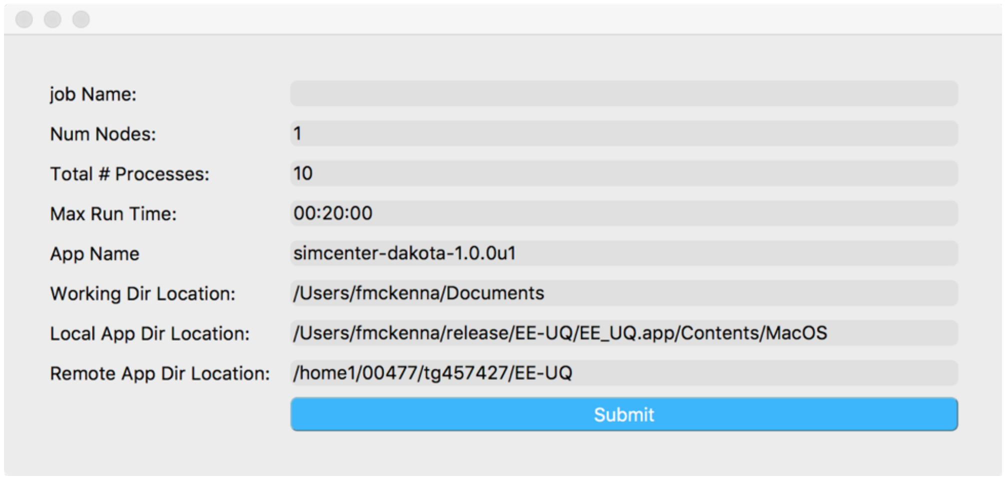
Finally, when inputs are finished the user hits submit button to start the backend job. If it runs the window will close and the RES panel will pop up on successful run. Do not press the submit button multiple times while waiting for it to close. We cannot guarantee what will happen and we did not disable the button in this release.

RUN at DesignSafe

Click this button to process the information, and send to DesignSafe where the job will be run on a supercomputer and results stored in your DesignSafe jobs folder.

A similar bit longer input panel is brought up:

- JobName: The name the user can use to identify the job in Get from DesignSafe.
- NumNodes: The number of compute nodes to use on Stampede2. Using the default App Name the job will run on Stampede2's KNL Landing (KNL) compute nodes. Each node has 68 cores. The actual number of cores the application will use on each of these nodes depends on the total number of processes specified. As per the TACC webpage, for MPI tasks it's best not to specify more than 64-68 processes to run. Depending on the numerical computations and amount of memory each uses, so as to avoid page faulting, for large simulations you may wish to use more nodes and less processes.
- Total Number of Processes: Total number of MPI parallel processes the UQ engine is going to use.



job Name:	
Num Nodes:	1
Total # Processes:	10
Max Run Time:	00:20:00
App Name	simcenter-dakota-1.0.0u1
Working Dir Location:	/Users/fmckenna/Documents
Local App Dir Location:	/Users/fmckenna/release/EE-UQ/EE_UQ.app/Contents/MacOS
Remote App Dir Location:	/home1/00477/tg457427/EE-UQ

Figure 4.16: Remote button

- Max Wall Time: HOURS:MIN:SEC be conservative. Your job is killed after the time limit. On Stampede2 you have a max wall time of 24 hours.
- App Name: Name of Agave app to run. DO not touch unless you know what you are doing.
- Working Dir Location: specifies where the EE_UQ application can create a "temporary" directory called tmp. SimCenter that the application creates when the submit button is pressed. The application creates this directory, copies files to it that the application needs as a result of your input (e.g. if you are using OpenSees input script, it will to the tmp. SimCenter directory copy that script, ALL FILES IN THAT DIRECTORY AND ALL FILES IN SUBDIRECTORIES OF THAT DIRECTORY. (SO, DON'T PLACE THE SCRIPT IN HOME, DOWNLOADS, DOCUMENTS, …). That directory is removed when jib has been successfully submitted.
- Local App Dir Location: SHOULD NOT BE TOUCHED unless you are introducing your own applications or want to build and modify the applications provided with the tool. It is this directory the application tool looks to find the applications it needs.
- Remote App Dir Location: Remote directory on Stampede2 where applications needed by workflow reside. DO not touch unless you know what you are doing.

GET from DesignSafe

Click this button to obtain from DesignSafe your list of jobs and select from that list a job to update status of, download or delete.

Exit

Click this button to exit the application.

Theory and Implementation

The following section describes the workings of the tool. If you intend to use the tool a lot or extend it, it is important that you read this section. Some Definitions before we start:

- Workflow: A sequence of steps involved in moving from a beginning state to an ending state.
- Scientific Workflow Application: An application that automates a workflow process through software, with each step in the workflow being performed by a separate "scientific" software application.
- Scientific Workflow System software providing an infrastructure for the set-up, scheduling, running, and monitoring of a user defined scientific workflow application.

The EE-UQ application is a very limited scientific workflow system that allows users to create scientific workflow applications needed for the characterization of the response of a building subjected to earthquake ground motions. It allows the users to then create and run the workflow application using the data of the users choosing. The application itself is composed of 2 parts:

- Frontend User Interface (UI): This is the application the user interacts with to create a building description, the BIM, and specify the workflow to run, i.e. given the building, the user chooses which applications to use and what data to use for the different applications. The UI is what was explained in section 3. It's purpose as is shown in Figure 5.1 is to create the BIM and start the workflow. Currently the inputs for the workflow are stored in the BIM file to reduce file overhead.
- Backend Application: This is the application that actually creates and runs the workflow. It consists of a script that processes the output file from the UI to determine the applications to run and their data, it invokes these applications using the outputs from one application as the input to another. The application that is run is a python script, EE-UQ.py, that can be found in the /applications/Workflow/ directory. The input and output from each application is in the form of JavaScript Object Notation (JSON) files. JSON is a human readable file format used widely for passing data between your front-end browser application (Safari, Firefox, Internet Explorer) and backend servers.

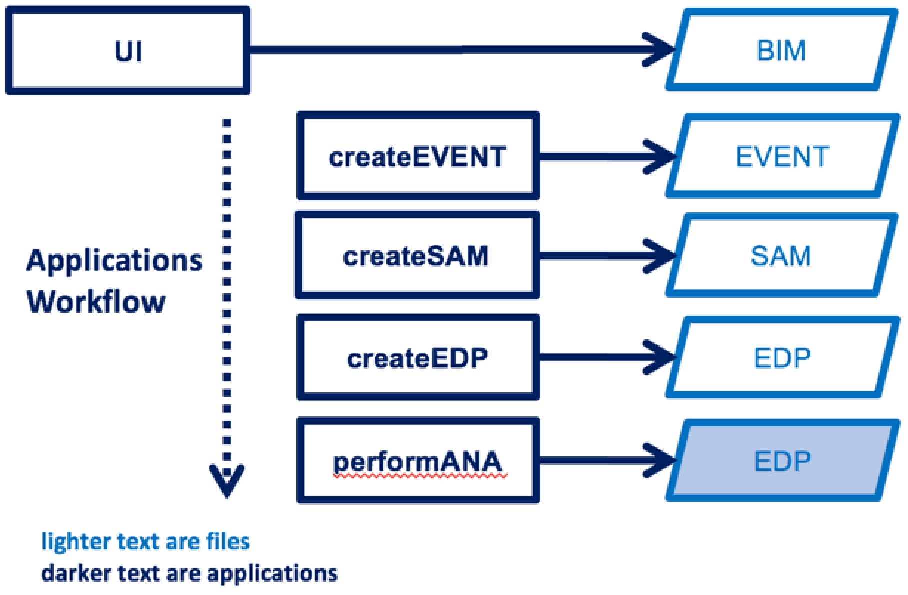


Figure 5.1: Workflow

In the absence of uncertainty, the applications that are invoked by this script are categorized into certain types of applications and are as shown in Figure 5.1:

1. createEVENT: given the structure and the user input for hazard application, define the loadings for the building, i.e. the ground motions for an earthquake event. The output file is an EVENT file.

2. createSAM: given the building description and event, create a finite element model of the building. The output file is a SAM (Structural Analysis Model) file.

3. createEDP: given the building, determine what output quantities are required. The output file is the EDP (engineering Demand Parameters) file.

4. Currely the user has no selection over the EDP's as the StandardEarthquakeEDP application is the built-in default application.

5. performANA: given the finite element model and the event, perform a finite element simulation. The responsibility of the performANA is to fill in the values in the EDP files.

The need to characterize the uncertainties in the computed response complicates this workflow. This is because the uncertainties in the inputs, random variables and random field variables, may exist for each application, e.g. Young's modulus in the building input file, magnitude of event or event ground motion in createEVENT, finite element material

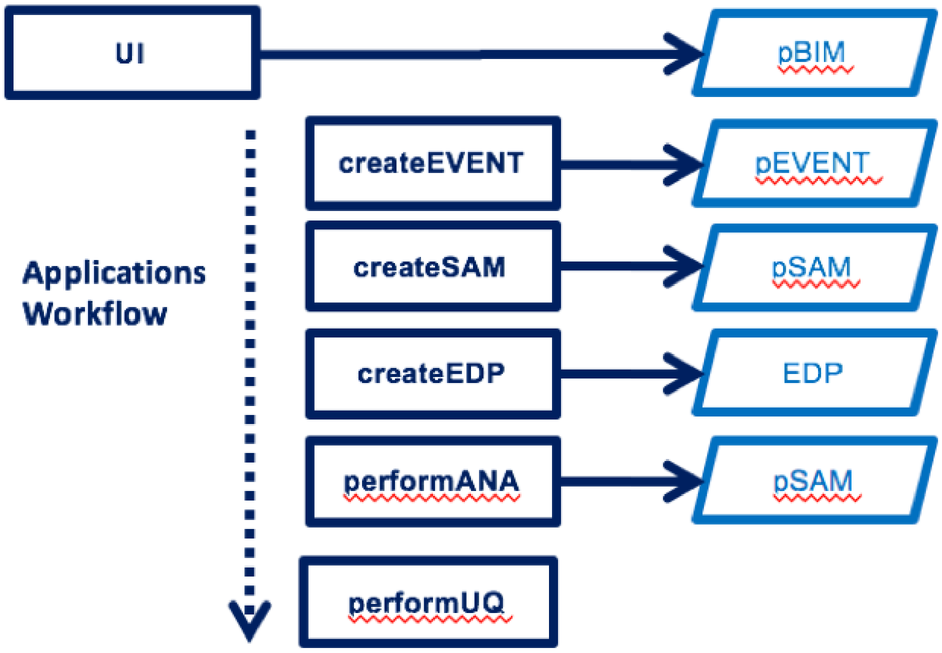


Figure 5.2: Workflow

properties in createSAM, and integration scheme, damping ratio or convergence tolerance in performANA.

As a consequence, each application is called with 2 different sets of input arguments. The first time the application is invoked with a "–getRV" input argument. This tells the application to return information about the random variables inside a "randomVariables" entry in the p output file generated by the application along with other needed data, e.g. the event type. The randomVariables is a JSON array of random variables, each with a field for a name, a type, a value, and other info that depends on the type, e.g.

```
{
  "randomVariables": [
        {
            "distribution": "Normal",
            "mean": 6,
            "name": "fc",
            "stdDev": 0.6,
            "value": "RV.fc",
            "variableClass": "Uncertain"
        },
        {
            "distribution": "Normal",
            "mean": 60,
            "name": "fy",
            "stdDev": 6,
            "value": "RV.fy",
            "variableClass": "Uncertain"
        },
        {
            "distribution": "Normal",
            "mean": 30000,
            "name": "E",
            "stdDev": 3000,
            "value": "RV.E",
            "variableClass": "Uncertain"
        }

}
```

It is during the running of the UQ engine that the value field in these random variables are filled in. Initially as shown, the value field contains RV.variableName. This is a must and what is used by the UQ engine to set the value. It is when the application is called again by the UQ engine during it's running that the application is called without the "-getRV". The application finds these value fields now set to numbers (or strings) in the pFiles that the application uses.

The performUQ application is actually a script that calls 3 applications, as shown below:

1. PreProcessUQ: will first must parse all the pFiles to build the list of all random variables.

2. PeformUQ: It then invokes the UQ engine, which for the number of samples specified will fill in the random variable values, run the applications in the workflow with the new files.

3. PostProcessUQ: will combine all the output results, filling in the EDP's

The computationally expensive part of the simulations is of course the PerformUQ. As discussed earlier, the user has the option of running locally or remotely at DesignSafe. When the user selects to run the job remotely, it is actually the PerformUQ operation that is run locally. The process of setting up the pFiles is done locally. These files are placed in a directory (along with all other needed files) and the files are transferred to DesignSafe and then an Agave application is invoked to run the application on Stampede2.

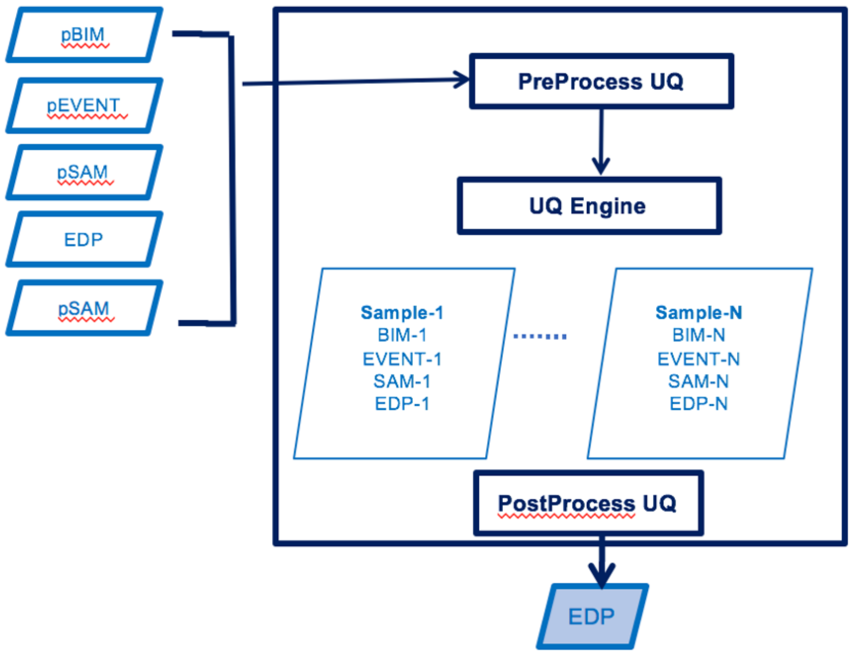


Figure 5.3: UQ sampling

Source Code

This source code for the tool is released under the 2-clause BSD License, commonly called the FreeBSD license. It is available for download from the tools GitHub repository: https://github.com/NHERI-SimCenter/EE-UQ

User Training

User Training consists of an online video available from the tool webpage that demonstrates tool use. The tool will be presented in user workshops hosted by the SimCenter.

Requirements

The following table outlines the user requirements identified for this tool. If you want some additional features added, contact us.

#	Description	Priority	Version
1	Ability to perform UQ on Building with Single Earthquake		
1.1	Run on Local Machine (Mac and Windows)	M	1.0
1.2	Run on Stampede2 through DesignSafe utilizing Agave	M	1.0
2	Motion Selection		
2.1	Ability to select from Multiple Earthquakes and view UQ due to all the discrete events	M	1.0
2.2	Ability to select from list of SimCenter motions	M	1.0
2.3	Ability to select from list of PEER motions.	D	1.0
2.4	Ability to use OpenSHA and selection methods to generate motions	D	1.0
2.5	Ability to Utilize Own Application in Workflow	M	1.0
2.6	Ability to use Broadband	D	1.1
2.7	Ability to use bring motion from rock to surface through soil		
	a) 1d soil effective stress analysis though different soil layers	M	1.1
	b) 2d bidirectional loading	M	1.2
	c) 2d bidirectional with full stochastic characterization of soil layers	M	1.3
3	Building Model Generation		
3.1	Ability to use existing OpenSees model scripts.	M	1
3.2	Ability to define building and use Expert System to generate FE mesh.		
	a) Concrete Shear Walls	M	1.1
	b) Moment Frames	M	1.2
	c) Braced Frames	M	2.0
3.3	Ability to define building and use Machine Learning applications to generate FE mesh for:		
	d) Concrete Shear Walls	M	2.0
	e) Moment Frames	M	2.1
	f) Braced Frames	M	2.3
3.4	Ability to specify connection details for member ends	M	2
3.5	Ability to define a user-defined moment-rotation response representing the connection details	D	2
4	FEM		
4.1	Ability to specify OpenSees as FEM engine and to specify different analysis options.	M	1
4.2	Ability to provide own OpenSees Analysis script to OpenSees engine.	D	1
4.3	Ability to use alternative FEM engine.	M	1.1
5	UQ - Method		
5.1	Ability to Use Dakota UQ engine with the Monte Carlo and LHS methods to perform sampling	M	1
5.2	Ability to Use alternative UQ engines	M	2
6	UQ – Random Variables		
6.1	Ability to Define Variables of certain types: a) Normal b) Lognormal c) Uniform d) Beta e) Weibull f) Gumbel	M	1.0
6.2	User defined Distribution	M	1.1
6.3	Define Correlation Matrix	M	1.1
7	Tool to allow user to store current configuration and reload it later	M	1

Table 8.1: Features (M=Mandatory, D=Desirable, O=Optional, P=Possible Future)

Version	Release	Requirements
1.0	Sept 2018	1.1, 1.2, 2.1, 2.2, 2.3, 2.4, 2.5, 3.1, 4.1, 4.2, 5.1, 5.2, 6.1, 7
1.1	Dec 2018	2.6, 3.2, 4.3, 6.2, 6.3

Table 8.2: Schedule

Verification and Validation

The following section (will) contain examples that verify the functionality of the tool.

Examples This section provides examples of using EE-UQ for uncertainty quantification of structural analysis models used in earthquake engineering. Results of each model is verified against results obtained using other tools. Two-Dimensional Portal Frame subjected to Gravity and Earthquake Loading In this example, a simple 2D portal frame model is used to verify the results of $EE - UQ$. The model is a linear elastic single-bay single-story model of a reinforced concrete portal frame (Figure 9.1). The analysis of this model considers both gravity loading and lateral earthquake loading due to El Centro earthquake (Borrego Mountain 04/09/68 0230, El Centro ARRAY #9, 270). The original model and ground motion used in this example were obtained from example 1b in OpenSees website, and were modified to scale the ground motion record from gravity units (g) to the model units (in/sec2). Files for this example are included with the release of the software and are available in the Examples folder in a subfolder called PortalFrame2D.

To introduce uncertainty in the model, both mass and young's modulus are assumed to be normally distributed random variables with means and standard deviation values shown in Table 1. In this example, the model will be sampled with the Latin Hypercube sampling method using both EE-UQ and a Python script (PortalFrameSampling.py) and response statistics from both analyses will be compared.

Uncertain Parameter	Distribution	Mean	Standard Deviation
Nodal Mass, m [kip]	Normal	5.18	1.0
Young's Modulus, E [ksi]	Normal	4227	500.0

Table 9.1: Uncertain parameters defined in the portal frame model

Modeling uncertainty using EE-UQ can be done using the following steps:

1. Start EE-UQ, click on the simulation tab (SIM) in the left bar to open a building simulation model. Click on choose button in the input script row:
2. Choose the model file Portal2D-UQ.tcl from PortalFrame2D example folder.

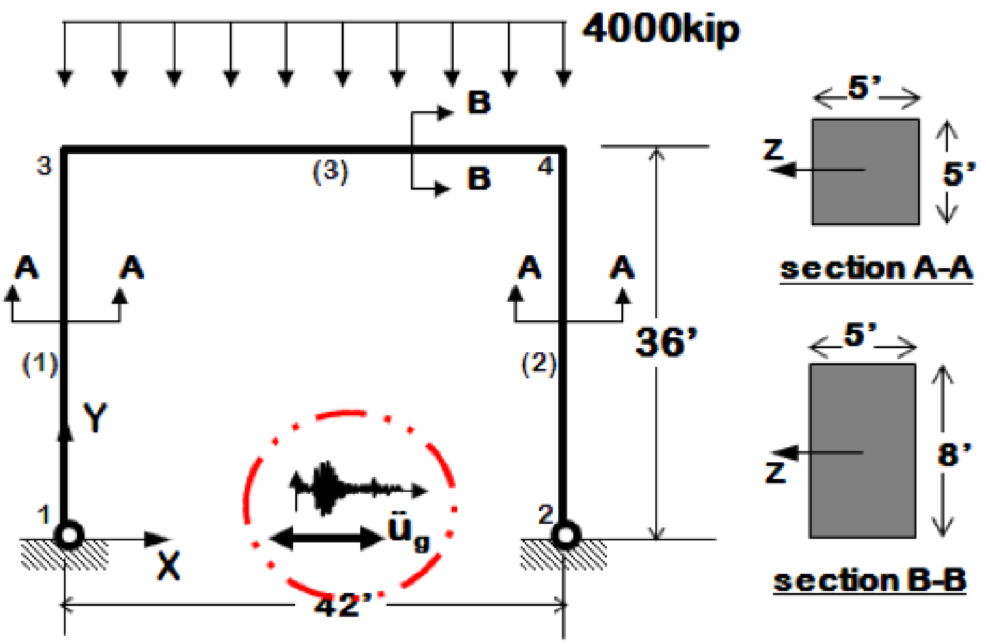
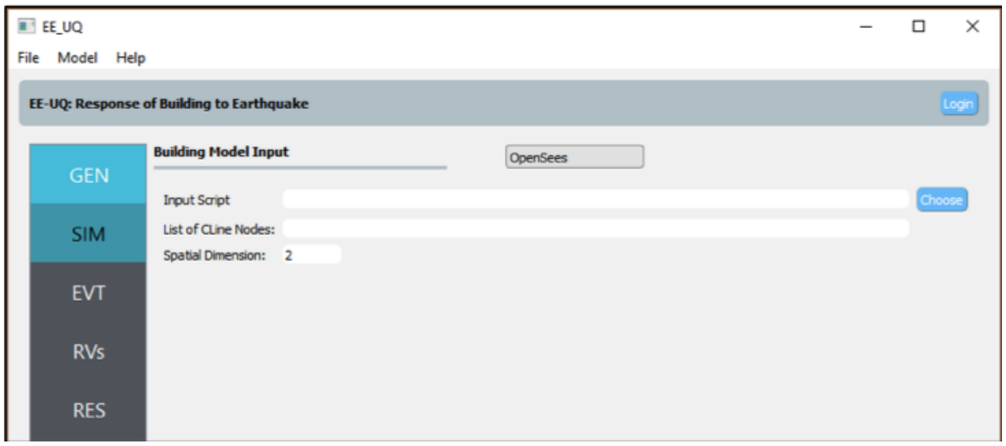


Figure 9.1: Two-dimensional portal frame model subjected to gravity and earthquake loading



Building Model Input

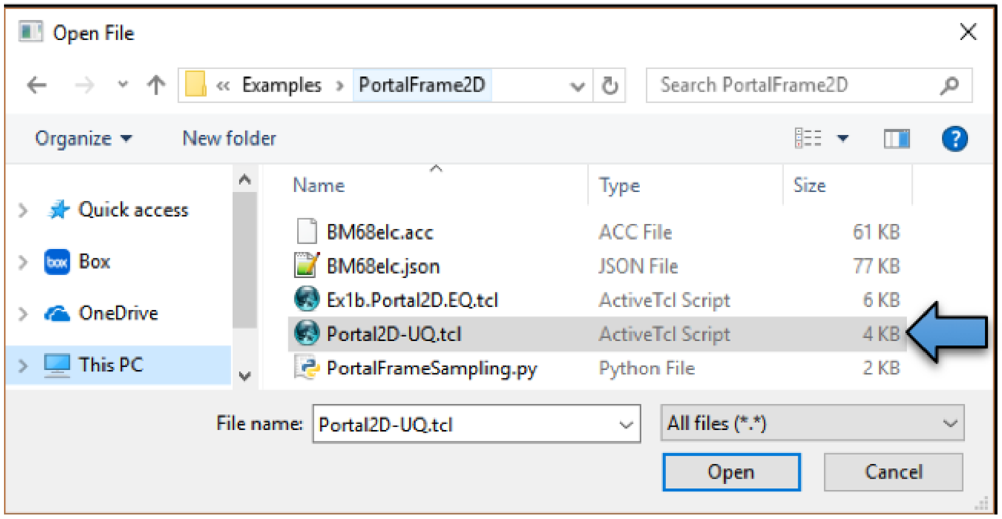
OpenSees

Input Script

List of CLine Nodes:

Spatial Dimension: 2

Figure 9.2: Choose building model



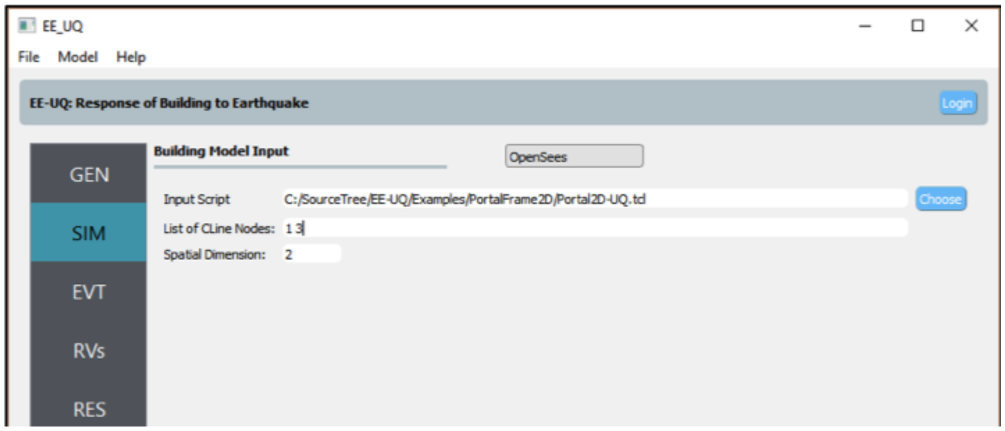
Name	Type	Size
BM68elc.acc	ACC File	61 KB
BM68elc.json	JSON File	77 KB
Ex1b.Portal2D.EQ.tcl	ActiveTcl Script	6 KB
Portal2D-UQ.tcl	ActiveTcl Script	4 KB
PortalFrameSampling.py	Python File	2 KB

File name: Portal2D-UQ.tcl

All files (*.*)

Figure 9.3: Choose tcl file

3. In the list of Clines Nodes edit box, enter “1, 3”. This indicates to EE-UQ that nodes 1 and 3 are the nodes used to obtain EDP at different floor levels (i.e. base and first floor).



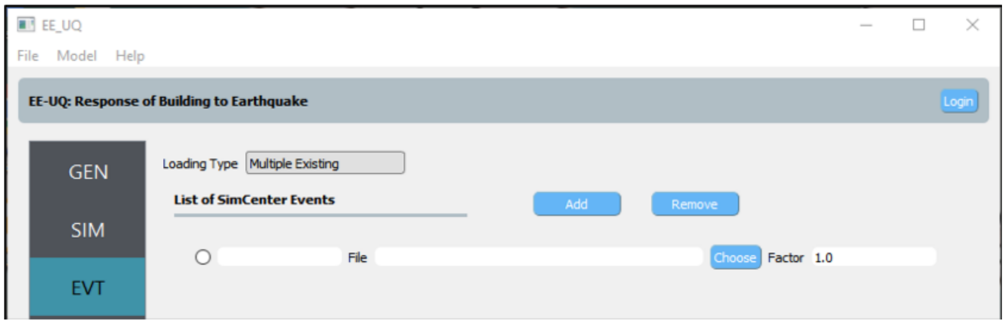
Input Script: C:/SourceTree/EE-UQ/Examples/PortalFrame2D/Portal2D-UQ.tcl

List of CLine Nodes: 1 3

Spatial Dimension: 2

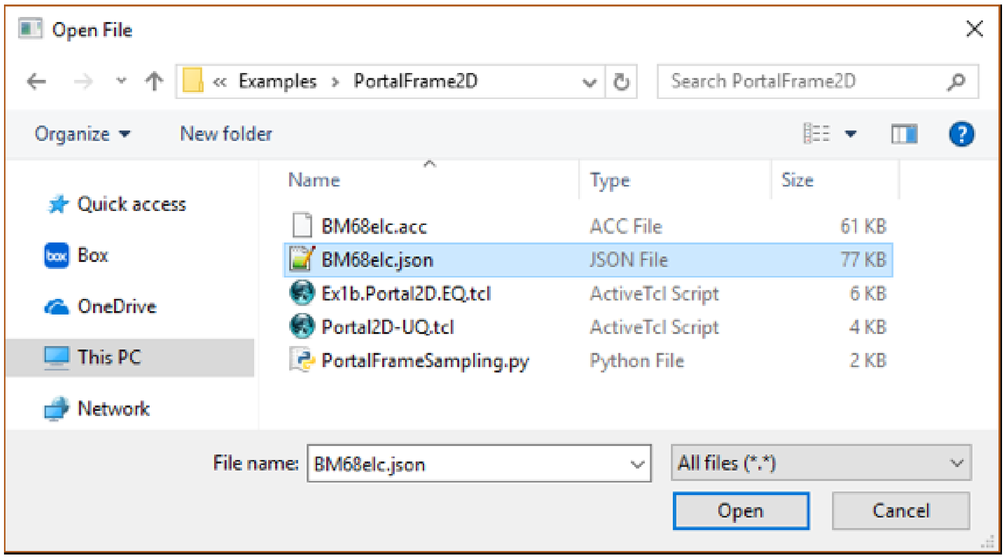
Figure 9.4: Select nodes

4. Click on the event tab (EVT) in the left bar to open the earthquake event specification tab, select Multiple Existing for loading Type. Click on the add button to add an earthquake event. Then click on the choose button to select the event file.
5. Choose the event file (BM68elc.json) for El Centro earthquake provided in the portal frame 2D example folder.
6. Now select the random variables tab (RVs) from the left bar, change the random variables types to normal and set the mean and standard deviation values of the floor



Loading Type: Multiple Existing
List of SimCenter Events
File
Factor 1.0

Figure 9.5: Work on EVT tab



Name	Type	Size
BM68elc.acc	ACC File	61 KB
BM68elc.json	JSON File	77 KB
Ex1b.Portal2D.EQ.tcl	ActiveTcl Script	6 KB
Portal2D-UQ.tcl	ActiveTcl Script	4 KB
PortalFrameSampling.py	Python File	2 KB

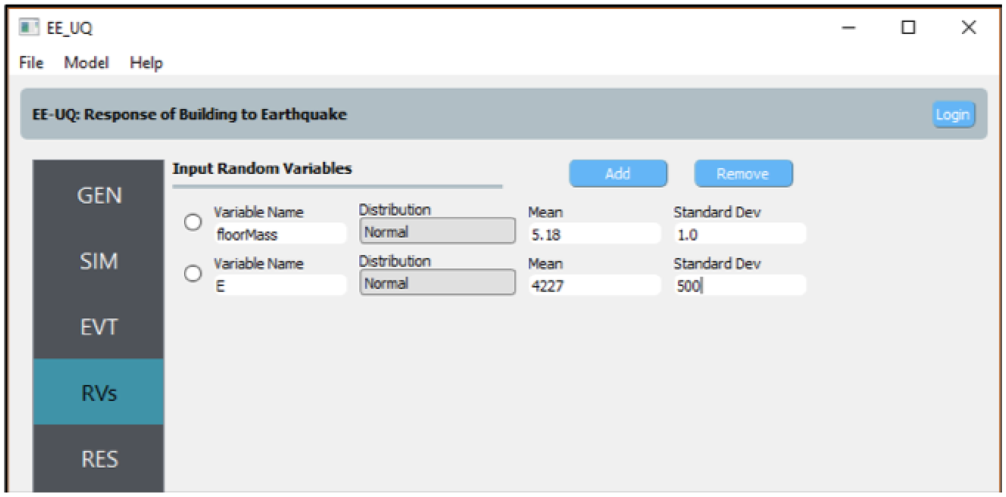
File name: BM68elc.json
All files (*.*)

Figure 9.6: Choose event file

mass and Young's modulus. Notice that EE-UQ has automatically detected parameters defined in the OpenSees tcl file using the pset command and defined them as random variables.

7. Now click on run, set the analysis parameters, working directory and applications directory and click submit to run the analysis. If everything ran successfully the program will automatically open the results tab showing the summary of results (Figure 9.8).

Verification script A verification script (Listing 1) for propagating the uncertainty was developed in Python and is included in the example folder. The script creates 1000 samples for both the Young's modulus and mass values using Latin Hypercube sampling, then modifies the OpenSees model, runs it and stores the output. After all the model samples are processed, the script will compute and output the mean and standard deviation values of the peak floor acceleration and peak drift.



EE-UQ: Response of Building to Earthquake

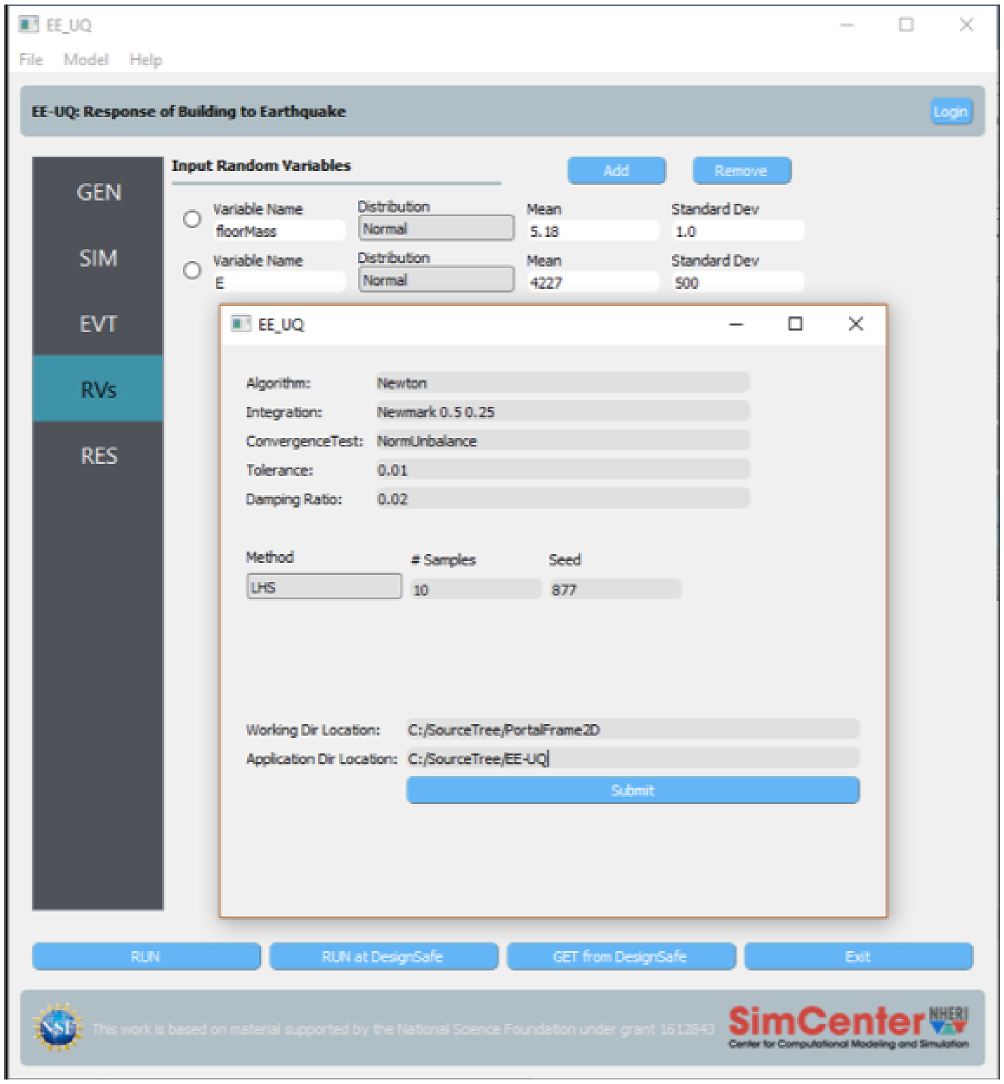
Input Random Variables

Variable Name	Distribution	Mean	Standard Dev
floorMass	Normal	5.18	1.0
E	Normal	4227	500

Figure 9.7: Work on RVs tab

Listing 9.1: Python script for analyzing the portal frame model with uncertain parameters

```
import numpy as np
import os
import shutil
import subprocess
from pyDOE import *
from scipy.stats.distributions import norm

#Setting number of samples
nSamples = 1000

#Creating latin hyper cube designs
design = lhs(2, samples=nSamples)

#Sampling Young's Modulus and Mass
ESamples = norm(loc=4227, scale=500.0).ppf(design[:,0])
mSamples = norm(loc=5.18, scale=1.0).ppf(design[:,1])

#Initializing output arrays
PFA = []
PID = []
#Reading OpenSees Model
with open ("Ex1b.Portal2D.EQ.tcl", "r") as portalFrameFile:
    portalFrameModel = portalFrameFile.read()

    #Looping through the samples and creating modified models
    for i in range(nSamples):
        sampleName = str(i+1)
        if(os.path.exists(sampleName) and os.path.isdir(sampleName)):
            shutil.rmtree(sampleName)

        os.mkdir(sampleName)
        shutil.copy('BM68elc.acc', sampleName)

        #Modifying the model using sample E and m values
        with open (sampleName + '/Ex1b.Portal2D.EQ.tcl' , "w+") as modifiedFile:
            modifiedModel = portalFrameModel.replace('pset floorMass 5.18', 'pset floorMass ' + str(mSampl
            modifiedModel = modifiedModel.replace('pset E 4227', 'pset E ' + str(ESamples[i]))
```



EE-UQ: Response of Building to Earthquake

Input Random Variables

Variable Name	Distribution	Mean	Standard Dev
floorMass	Normal	5.18	1.0
E	Normal	4227	500

Algorithm: Newton
Integration: Newmark 0.5 0.25
ConvergenceTest: NormUnbalance
Tolerance: 0.01
Damping Ratio: 0.02

Method	# Samples	Seed
LHS	10	877

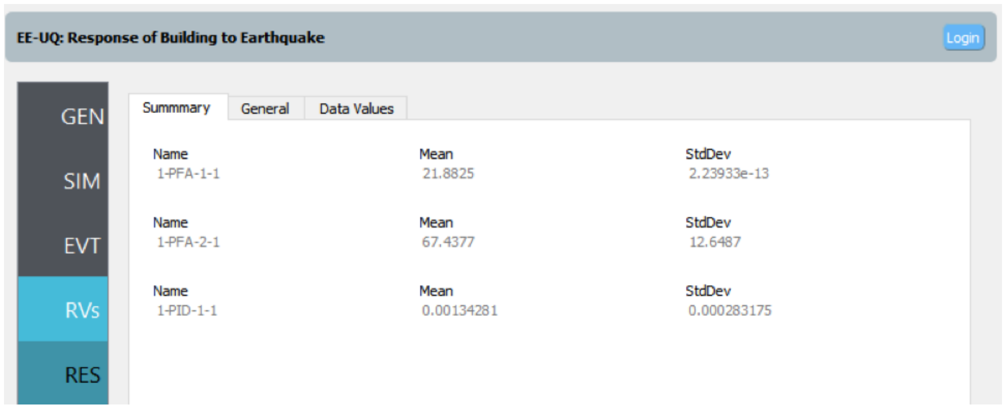
Working Dir Location: C:/SourceTree/PortalFrame2D
Application Dir Location: C:/SourceTree/EE-UQ

This work is based on material supported by the National Science Foundation under grant 1612843

Figure 9.8: Run

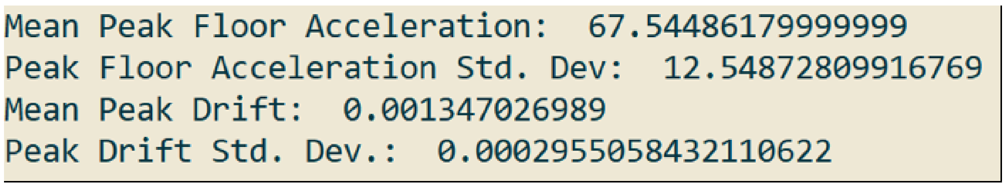
```
            modifiedFile.write(modifiedModel)

        #Running OpenSees
        subprocess.Popen("OpenSees Ex1b.Portal2D.EQ.tcl", shell=True, cwd=sampleName).wait()

        #Reading Peak Floor Acceleration
        with open (sampleName + '/PFA.out' , "r") as pfaFile:
            PFA.append(float(pfaFile.readlines()[2]))

        #Reading Peak Floor Acceleration
        with open (sampleName + '/PID.out' , "r") as pidFile:
            PID.append(float(pidFile.readlines()[2]))

        #Cleaning up
        shutil.rmtree(sampleName)

#Printing results
print 'Mean Peak Floor Acceleration: ', np.mean(PFA)
print 'Peak Floor Acceleration Std. Dev: ', np.std(PFA)

print 'Mean Peak Drift: ', np.mean(PID)
print 'Peak Drift Std. Dev.: ', np.std(PID)
```

Verification of results In this section, the results produced for the portal frame by EE-UQ are verified against the results of running the same problem using the Python script. Running the uncertainty quantification problem on the local computer produces the results shown in Figure 9.9 Running the analysis using the sampling Python script produces the results shown in Figure 9.10. Both results (Mean and standard deviation values of EDPs) are compared in Table 2 and are shown to be in good agreement.



Name	Mean	StdDev
1-PFA-1-1	21.8825	2.23933e-13
1-PFA-2-1	67.4377	12.6487
1-PID-1-1	0.00134281	0.000283175

Figure 9.9: Outputs from EE-UQ



```
Mean Peak Floor Acceleration:  67.54486179999999
Peak Floor Acceleration Std. Dev:  12.54872809916769
Mean Peak Drift:  0.001347026989
Peak Drift Std. Dev.:  0.0002955058432110622
```

Figure 9.10: Outputs from PortalFrameSamplying.py script

Engineering Demand Parameter		EE-UQ	Python Script	Percent Difference [%]
Peak Floor Acceleration [in/s^2]	Mean	67.4377	67.5448	0.16
	Std. Dev.	12.6487	12.5487	0.8
Peak Story Drift [x10-3 in]	Mean	1.3428	1.347	0.3
	Std. Dev.	0.2832	0.2955	4.1

Table 9.2: Features (M=Mandatory, D=Desirable, O=Optional, P=Possible Future)

Bibliography

[1] John Lysmer and Roger L Kuhlemeyer. “Finite dynamic model for infinite media”. In: (1969).