**Gallery of Teaching Applications**

**Module: Uncertainty quantification (UQ) for structural models**

**Title:** Uncertainty quantification (UQ) for structural models

**Target audience:** Graduate students in Structural Engineering with basic knowledge of programing, linear and nonlinear structural analysis.

**Learning objectives:**

* Calibrate an OpenSees structural model based on test data
* Quantify the impact of the model calibrations based on modeling decision such as including or excluding axial load effects.
* Estimate the uncertainty in the model parameters.

**List of tools:**

* OpenSees
* SimCenter quoFEM

**Supplemental material:**

* Download link quoFEM: <https://www.designsafe-ci.org/data/browser/public/designsafe.storage.community//SimCenter/Software/quoFEM>
* Training videos quoFEM: <https://simcenter.designsafe-ci.org/research-tools/quofem-application/>.

**Activity description:**

In this problem, you will address the different questions that arise while calibrating a nonlinear structural model using experimental results. quoFEM will be used to calibrate a concentrated plastic hinge model of a rectangular concrete column tested by Soesianawati in 1989. The model is presented in the figure below. This tests only provides information to moderate levels of displacement; thus, you will only calibrate the cracking factor (*crack*\_*factor)*, the yielding moment (*Mp*), and the capping moment (*M*c) of the plastic hinge. The test results (displacement and force time histories), the OpenSeesPy model of the column, and the Google Colab notebooks for postprocessing are available to support your work.



The problem and the steps in this module follow the examples presented during class lecture. The model shown in lecture is similar to this problem, except that the model in this problem includes axial load. Hence, the results for the problem are not going to be identical to those of the example shown in the lecture. Please note that you don’t need to modify the OpenSeesPy model since the files attached to this problem already include the gravity load.

The following four steps are required to successfully complete this problem:

1. Perform a **parametric study** in quoFEM to investigate the influence of the *crack*\_*factor*, *Mp*, and *Mc* parameters on the calibration error of the test.
	1. Set the UQ Engine to Dakota with Forward Propagation using Latin Hypercube Sampling (LHS). Use 500 samples for the parametric study and any seed value (the seed is only to reproduce the exact random samples).
	2. Choose “Pyhton” as FEM application and choose the appropriate model script provided in the StarterCode folder. Choose the parameter script folder that defines the random variables.
	3. Assign a uniform distribution to the three parameters of calibration using the following bounds:

*crack*\_*factor* = 0.05 – 0.80

*Mp* = 100 – 500 kN-m

*Mc* = 100 - 500 kN-m

* 1. Specify “force” with a length of 548 as the quantity of interest. (Verify that the “experimentalDisp.csv” and the “experimentalForce.csv” files have 548 values).
	2. Run the parametric study and save the table. For compatibility with the postprocessing notebook, save the table as “results\_parametric\_study.csv”.
1. Perform a **deterministic** **calibration** in quoFEM to find the *crack*\_*factor*, *Mp*, and *Mc* parameter vector that yields a numerical model that accurately simulates measured response of the column.
	1. Set the UQ Engine to Dakota, select the Dakota Method Category as Deterministic Calibration and use the NL2SOL method under the Method category. Use a maximum of 1000 iterations and a convergence tolerance of 0.0001. Choose the calibration data file as the measured force response in the file “experimentForce.csv”.
	2. Choose “Pyhton” as FEM application and choose the appropriate model script provided in the StarterCode folder. Choose the parameter script file that defines the random variables.
	3. Assign a uniform distribution to the three parameters of calibration using the following bounds:

*crack*\_*factor* = 0.05 – 0.80.

*Mp* = 100 – 500 kN-m

*Mc* = 100 - 500 kN-m

Use the initial value that you prefer within these ranges.

* 1. Specify “force” with a length of 548 as the quantity of interest.
	2. Run the calibration and save the table with all the iterations. For compatibility with the postprocessing notebook, save the table as “results\_det\_cal.csv”.

Provide two scatter plots showing the samples of the parametric study color-coding the calibration error (*crack*\_*factor* Vs. *Mp*, and *Mc* Vs. *Mp*). Use the post-processing notebook: post\_process\_quoFEM.ipynb. This plot should also include the path of the deterministic calibration.

Comment on the differences between the calibrated parameters that you obtained (considering axial load) and those presented during the lecture (ignoring axial load). What are the major differences? Why?

Provide a force-displacement plot that overlaps the experimental and simulated results of the experimental test using the calibrated values of *crack*\_*factor*, *Mp*, and *Mc.* Use the post-processing notebook column\_model.ipynb.

1. Perform a **Bayesian calibration** problem in quoFEM to find the distribution of *crack*\_*factor*, *Mp*, and *Mc* that represents the model uncertainty corresponding to the measured response.
	1. Set the UQ Engine to UCSD-UQ and select the Transitional Markov chain Monte Carlo method. A Bayesian calibration could take a lot of time to run depending on the number of samples, in this case, use 100 samples to achieve a reasonable representation of the parameter distributions with limited running time. Choose the calibration data file as the measured force response in the file “experimentForce.csv”.
	2. Choose “Python” as FEM application and choose the appropriate model script provided in the StarterCode folder. Choose the parameter script file that defines the random variables.
	3. Assign a uniform distribution to the three parameters of calibration using the following bounds:

*crack*\_*factor* = 0.05 – 0.80

*Mp* = 100 – 500 kN-m

*Mc* = 100 - 500 kN-m

* 1. Specify “force” with a length of 548 as the quantity of interest.
	2. Run the calibration and save the table with all the iterations. For compatibility with the postprocessing notebook, save the table as “results\_bayesian\_cal.csv”.

Provide a histogram for each parameter including a vertical dashed line depicting the mean value of the Bayesian calibration and the values of the deterministic calibration. Generate the histogram with the post-processing noteook: post\_process\_quoFEM.ipynb.

Provide two scatter plots showing the joint distributions of *Mp* vs *Mc* and *crack\_factor* vs *Mp.* Use the post-processing notebook: post\_process\_quoFEM.ipynb. Do the observed relationships make sense to you? Why?

**Starter code**

**Model and test files**

The folder “StarterCode/ model\_files\_wpdelta” includes the following files in the proper format to use in quoFEM:

ColumnModel.py = OpenSeesPy model of the column

ColumnParams.py = Parameters to calibrate

experimentDisp.csv = Measured displacement time history

experimentForce.csv = Measured force time history

**SimCenter quoFEM**

Please download and install the quoFEM application following the instructions in the online documentation:

<https://simcenter.designsafe-ci.org/research-tools/quofem-application/>

<https://nheri-simcenter.github.io/quoFEM-Documentation/>

**Postprocessing notebooks**

To facilitate your work, you can use the following notebooks to produce the plots required in this problem.

column\_model.ipynb: notebook to run the OpenSeesPy model for a set of assumed values of *crack*\_*factor*, *Mp*, and *Mc*and generate the plots of Force Vs. Displacement that compare the experimental and the simulated data in part (2). Note that you need to manually select the experimental results by clicking the “Choose Files” button in the notebook:



post\_process\_quoFEM.ipynb: notebook to produce the plots in part (2) and (3) from the quoFEM output files (the scatter plots and histograms). As with the experimental results, you should also manually select the files with the quoFEM outputs.