Transactions, **SMiRT-25**
Charlotte, NC, USA, August 4-9, 2019
Division III

NONLINEAR SOIL-STRUCTURE INTERACTION ANALYSIS USING CO-SIMULATION

Robert Spears[1] and Justin Coleman[2]

[1] Seismic Engineer, Idaho National Laboratory, Idaho Falls, ID (robert.spears@inl.gov), USA
[2] Seismic Engineer, Idaho National Laboratory, Idaho Falls, ID (justin.coleman@inl.gov), USA

ABSTRACT

Nonlinear soil-structure interaction (NLSSI) analysis methods are currently being developed at Idaho National Laboratory. Appendix B in ASCE (2016) includes a framework for performing NLSSI analysis. For NLSSI analysis to be widely implemented in industry, it is important to demonstrate that accurate response in dynamically loaded, nonlinear models can be achieved in an efficient manner.

Both explicit and implicit dynamic solvers are applicable to NLSSI. When considering CPU time, an explicit dynamic solver is often desirable for large models with no features (e.g. small elements, stiff elements, or stiffness damping) that can drive the time step unreasonably small. An implicit dynamic solver may be desirable in models not well suited for an explicit solver. A further option is co-simulation where a portion of the model is run with an explicit dynamic solver and the rest of the model is run with an implicit dynamic solver. This study explores NLSSI examples where co-simulation could be the most CPU efficient modelling strategy.

INTRODUCTION

In performing NLSSI analysis, one approach is to consider vertically propagating shear and compressive waves that pass through a nonlinear soil column and into a structure. Using finite element techniques, the structure can be placed on a finite volume of nonlinear soil. Then rock outcrop seismic time series can be applied to the bottom of the nonlinear soil. The mesh for the soil volume can be built using solid brick elements that model simple soil layers. The size of the brick elements can then be managed to be just small enough to pass the highest frequency of interest with ten elements per wavelength. Generally there is no need for the soil to have elements significantly smaller than the optimal size. Also, considering the hysteresis in the nonlinear soil, it is likely that added damping is unnecessary or only necessary in a very small amount. The mesh for the structure, however, may require a variety of elements (including bricks, shells, beams …). The geometry and materials in the structure may make it necessary to have relatively small and stiff elements. Additionally, significant damping (such as Rayleigh damping) may be necessary to produce accurate results.

Consider the mesh in Figure 1. The soil mesh contains about 1,800,000 brick elements while the structure contains about 9,000 brick, shell, and beam elements. The soil elements adjacent to the structure are of various sizes to fit the structure but the remaining mesh is uniformly sized (i.e. each soil layer has uniformly sized elements according to its stiffness so that the whole soil mesh accurately passes the same range of seismic wave frequencies). The soil is modelled with nonlinear hysteretic soil material properties and no Rayleigh damping. The structure is modelled with a variety of element shapes, stiffnesses, and sizes to address important geometry. Also, the structure is modelled with elastic material properties and Rayleigh damping.

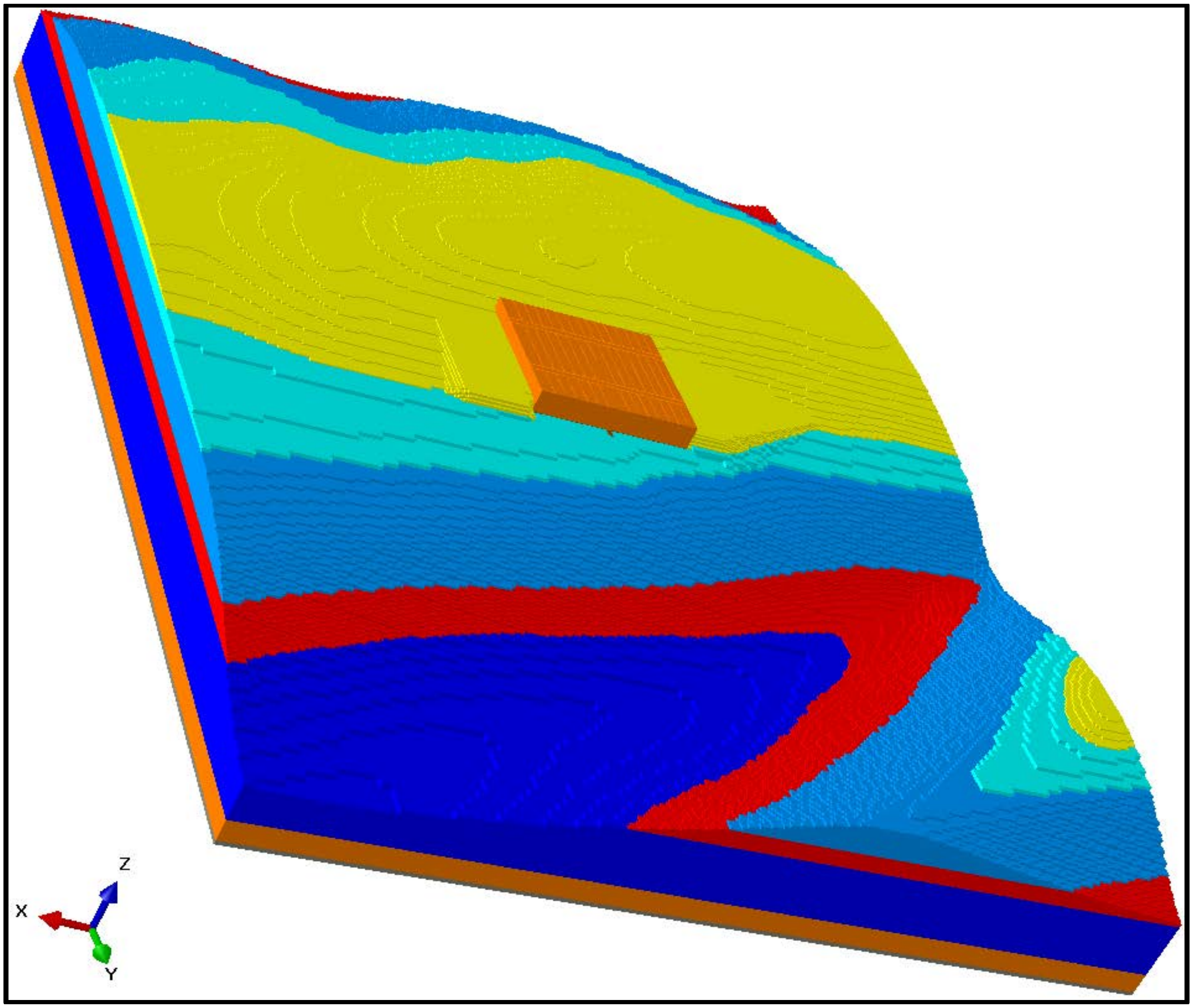

Figure 1. Example NLSSI model.

Due to the potential CPU efficiency of a dynamic explicit solver, a dynamic explicit approach should be considered for the whole model in Figure 1. If the modelled features in the vicinity of the structure cause a dynamic explicit solver to have a very small time step, then there is a possibility that a dynamic implicit solver could be more CPU efficient for the whole model. However, there is only a small portion of the mesh in Figure 1 (less than 1%) that has features that could cause a small time step in a dynamic explicit solver. Consequently, if the small portion of the mesh causes a significant reduction in the dynamic explicit time step (relative to that required for the uniform soil mesh), then this model should also be considered for the use of co-simulation. The co-simulation in this study is done with Abaqus (2018) where Abaqus/Explicit is used for the dynamic explicit solver and Abaqus/Standard is used for the dynamic implicit solver. A logical way to separate the Abaqus/Explicit and Abaqus/Standard portions of the model is shown in Figure 2. For this example, all of the uniformly sized soil elements are evaluated with Abaqus/Explicit and the remaining elements are evaluated with Abaqus/Standard. Consequently, the maximum dynamic explicit time step is realized.

For the example NLSSI model in Figure 2, coincident soil nodes exist at the boundary between the Abaqus/Explicit and Abaqus/Standard models. Both models step through time on the same time step and data is shared through their coincident nodes. Other model boundary conditions include constraints at the side boundaries of the model that cause each set of nodes at a given elevation to translate together. Additionally, there is a single layer of elastic soil elements at the base of the model. A seismic load time history is applied to its upper surface and dashpots tuned to generate non-reflective boundary conditions are attached to nodes on the bottom surface.

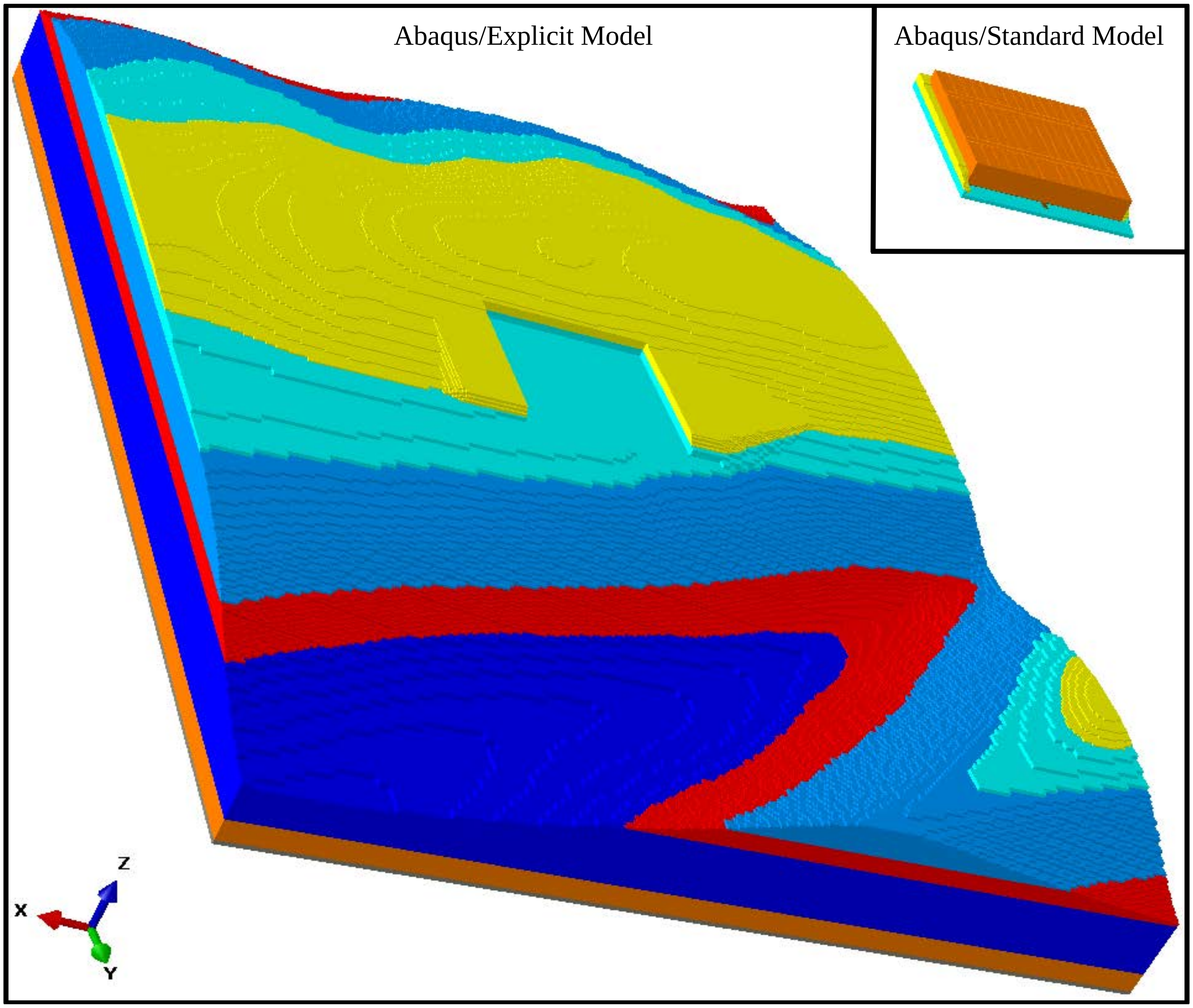

Figure 2. Example NLSSI co-simulation model.

CASE STUDY

To demonstrate the decision process between explicit only, implicit only, or co-simulation approaches, a case study is performed. The model for the case study is shown in Figure 3. The structure in the case study model is a simple box shape (72 ft on each edge) formed using 1.5 ft thick shell elements with elastic concrete material properties and Rayleigh damping. Half of the structure is embedded in the soil. The soil consists of elastic, mudstone brick elements (9 ft on each edge).

For co-simulation, the case study model is broken apart as shown in Figure 4. For this model, only the soil elements that touch the structure are included in the Abaqus/Standard model.

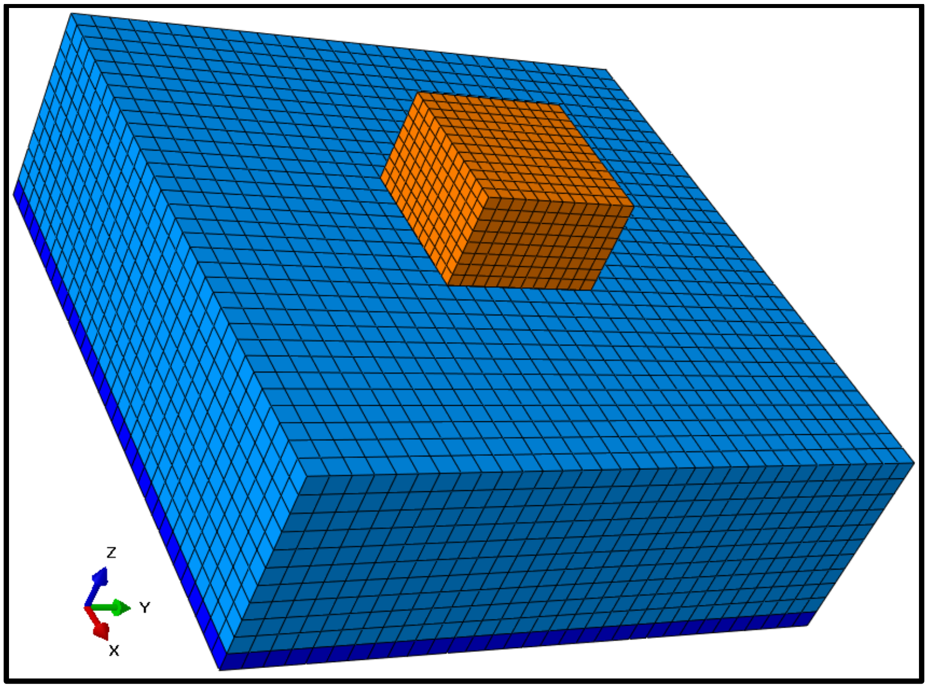

Figure 3. Case study model.

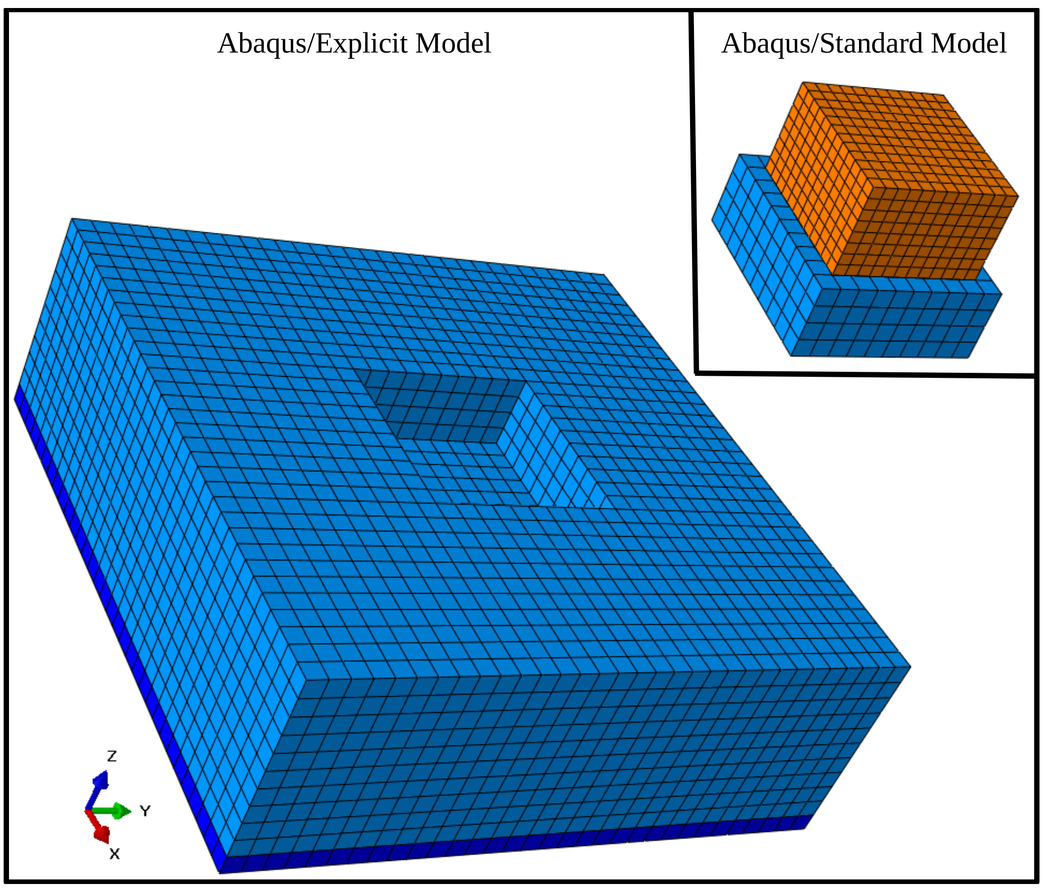

Figure 4. Case study co-simulation model.

The case study considers three versions of the structure that differ only in the addition of a steel beam element (shown in Figure 5). The steel beam is intended to represent a situation where the structural geometry may require small elements. The additional of the steel beam causes the structure to significantly affect the explicit time step.

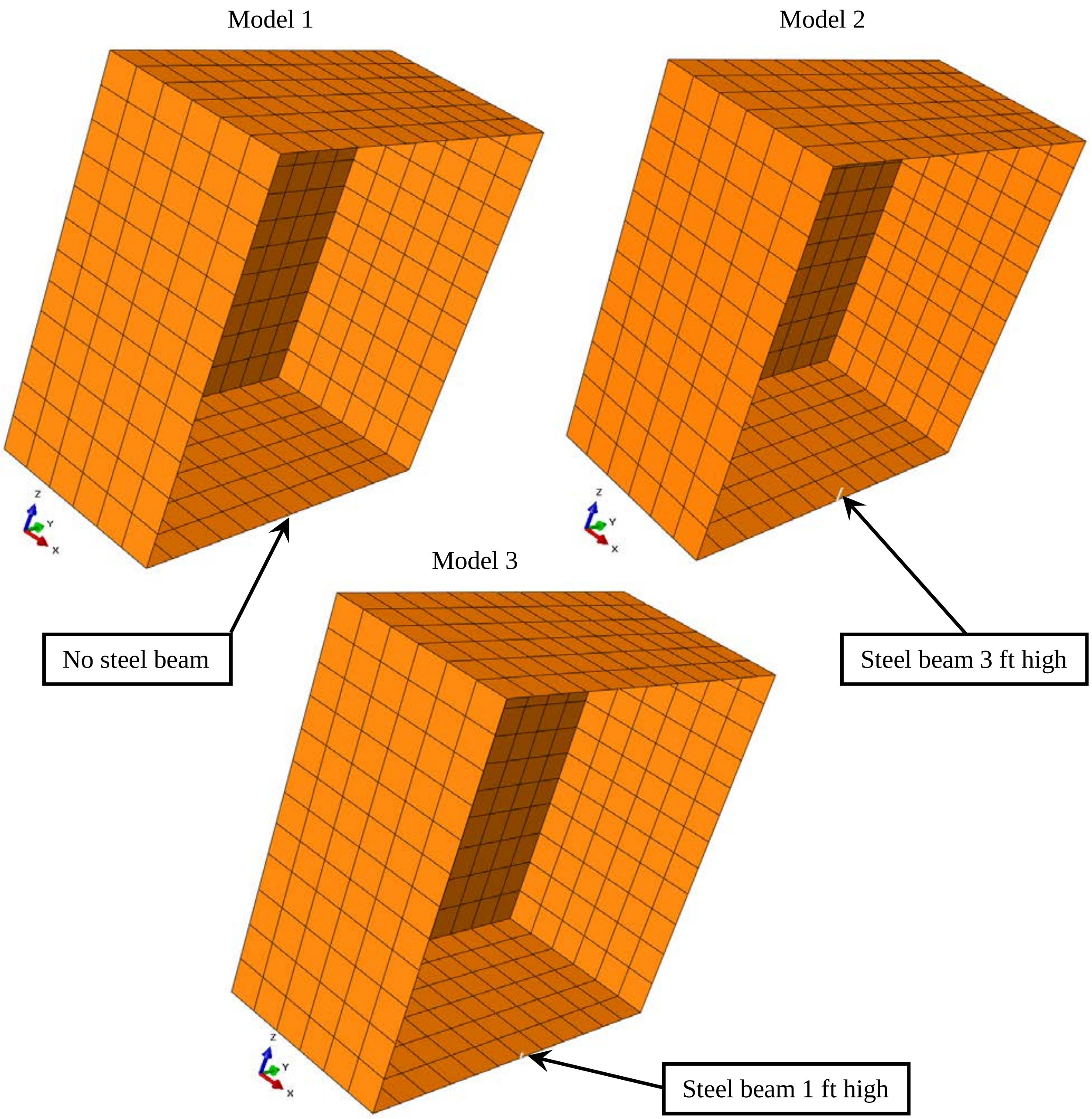

Figure 5. Cut-away view of the three structure models.

Other model boundary conditions include constraints at the side boundaries of the model that cause each set of nodes at a given elevation to translate together (to mimic and infinite continuum). Additionally, there is a single layer of elastic soil elements at the base of the model. A seismic load time history lasting 20 seconds is applied to its upper surface of the elastic soil elements. Dashpots tuned to generate non-reflective boundary conditions are attached to nodes on the bottom surface of the elastic soil elements.

MODEL SETUP AND CONFIGURE FILE

The co-simulation models in this study are all performed with a file named “SSI_exp.inp” for the Abaqus/Explicit portion and a file named “SSI_std.inp” for the Abaqus/Standard portion. The soil mesh is initially modelled as a continuous mesh for the Abaqus/Explicit only and Abaqus/Standard only model runs. Therefore, when the model is separated into two models for co-simulation, there are coincident nodes at the boundaries of the Abaqus/Explicit and Abaqus/Standard portions. These nodes are placed in a node set named “COSIM_EXP_ND” for the Abaqus/Explicit portion and “COSIM_STD_ND” for the Abaqus/Standard portion. The following commands are then included in the step cards to perform the co-simulation:

Commands included in the step portion of “SSI_exp.inp”:

```
*CO-SIMULATION CONTROLS, NAME=CTRL, TIME INCREMENTATION=LOCKSTEP
*CO-SIMULATION, NAME=EXP_STD, PROGRAM=ABAQUS, CONTROLS=CTRL
*CO-SIMULATION REGION
 COSIM_EXP_ND
```

Commands included in the step portion of “SSI_std.inp”:

```
*CO-SIMULATION CONTROLS, NAME=CTRL, TIME INCREMENTATION=LOCKSTEP
*CO-SIMULATION, NAME=EXP_STD, PROGRAM=ABAQUS, CONTROLS=CTRL
*CO-SIMULATION REGION
 COSIM_STD_ND
```

With the input files defined, the command line for running the co-simulation can be as follows:

```
abaqus cosimulation cosimjob=NLSSI job=SSI_std,SSI_exp cpus=1,1
```

Using the command line above, the model will proceed on the time step of the Abaqus/Explicit portion. In many cases (including all the models presented in this study), this is sufficient for a stable model run. However, if at any point the Abaqus/Standard portion requires a smaller time step for convergence, the model will crash. A simple modification to the configuration file for the co-simulation can be used to prevent this crash. Upon initial use of the command line above, a configure file will be automatically generated (named “elaborated_NLSSI_config.xml” for the study models). This configure file is a text file and can be modified to remove the commands for a locked step. By doing so, the configure file tells the co-simulation to use the minimum time step of either solver. To test this process, a scoping model was performed where the Abaqus/Standard portion required a smaller time step than the Abaqus/explicit portion over some of the model run. The initial run (using the command line above) started out stable but crashed as expected when the Abaqus/Standard portion required a smaller time step than the Abaqus/explicit portion. A second run was performed where the initial run configure file was copied to “modified_config.xml” and the following modification was performed:

Text in “elaborated_NLSSI_config.xml”:

```
<componentInstances>
   <componentInstance modelIdentifier="SSI_std" name="STD">
     <component>Abaqus/Standard</component>
     <timeIncrementation>
       <lockstep>true</lockstep>
     </timeIncrementation>
   </componentInstance>
   <componentInstance modelIdentifier="SSI_exp" name="XPL">
     <component>Abaqus/Explicit</component>
```

```
        <timeIncrementation>
          <lockstep>true</lockstep>
        </timeIncrementation>
      </componentInstance>
    </componentInstances>
```

Modified text in “modified_config.xml”:

```
<componentInstances>
      <componentInstance modelIdentifier="SSI_std" name="STD">
        <component>Abaqus/Standard</component>
      </componentInstance>
      <componentInstance modelIdentifier="SSI_exp" name="XPL">
        <component>Abaqus/Explicit</component>
      </componentInstance>
    </componentInstances>
```

Given the modified configure file, the following command line could be used to perform the second run:

```
abaqus cosimulation cosimjob=NLSSI job=SSI_std,SSI_exp cpus=1,1 configure=modified_config.xml
```

This approach used the Abaqus/Explicit portion time step until the Abaqus/Standard portion required a smaller time step. Then the model run continued to converge (beyond the first model run) using the time step required by the Abaqus/Standard portion. Once the Abaqus/Explicit portion again required the smaller time step, the model run continued on its time step.

RESULTS

Figure 6 shows the case study in-structure response spectra data for the top, centre of the structure. The data shown in Figure 6 is the superimposed response spectra data for all of the model runs and it demonstrates that reasonable agreement can be achieved with all of the tested models.

The maximum response shown in Figure 6 is a little over 10 g. This is a very high seismic response but it makes a good test for model stability.

Considering the z-direction response between 20 Hz and 30 Hz in Figure 6, there is some notable discrepancy in the results. In this region, the higher amplitude data represents the Abaqus/Standard only and co-simulation results while the lower amplitude data represents the Abaqus/Explicit only results. In initial scoping models where a more coarse structural mesh was used, the response discrepancies were much greater. The results for the selected mesh density (shown in Figure 5) were considered sufficiently accurate while leaving the structure coarse enough to demonstrate how fast an explicit only solver can be.

Table 1 shows a summary of the case study results. All of the models were run on the same computer and an attempt was made to ensure that plenty of memory and CPUs were available for each.

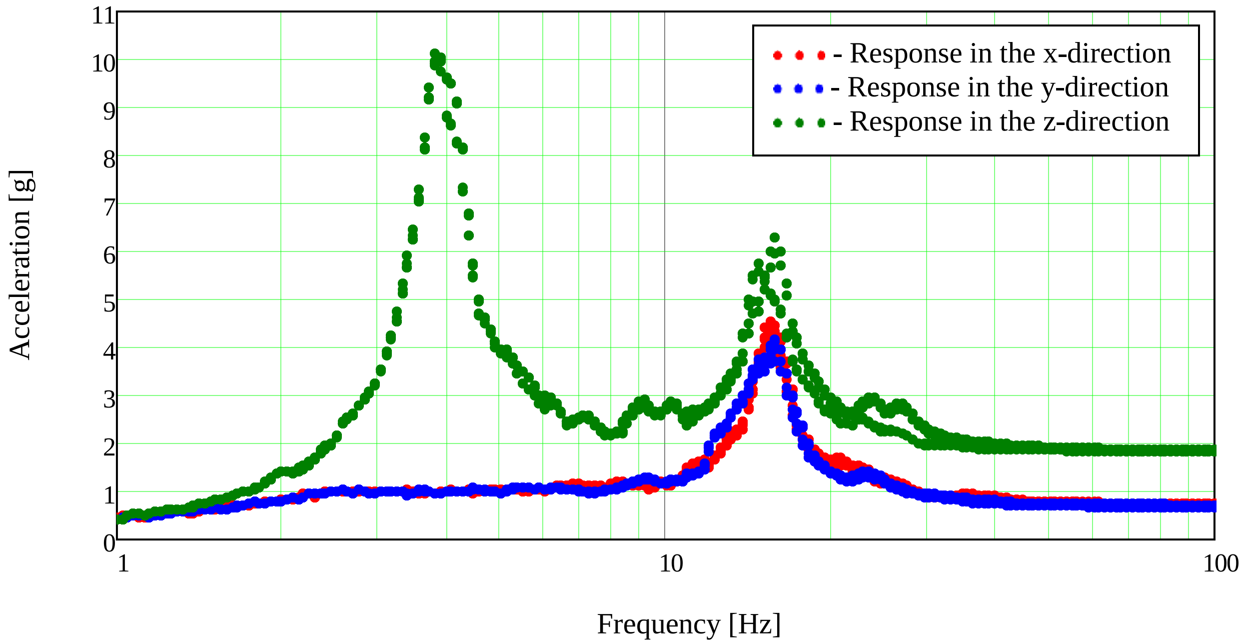

Figure 6. Case study 5% damped response spectra data for the top, centre of the structure.

Table 1: Summary of the case study results.

	Model 1	Model 2	Model 3
	CPU Time/Time Step [hr]/[sec]	CPU Time/Time Step [hr]/[sec]	CPU Time/Time Step [hr]/[sec]
Abaqus/Explicit Only (2 CPUs)	0.35/0.000144[1]	7.91/0.00000603	31.93/0.00000104
Abaqus/Standard Only (2 CPUs)	14.21/0.00125[2]	14.48/0.00125[2]	14.46/0.00125[2]
Co-simulation (1 CPU Abaqus/Explicit and 1 CPU Abaqus/Standard)	3.82/0.000437	4.01/0.000437	4.03/0.000437

1.) Structural elements (sized to increase response spectra agreement) forced the time step in this model to be approximately three times smaller than would be required by the soil elements. Consequently, with a more coarse structural mesh, the Model 1 CPU time could be reduced to as much as one third of the CPU time shown.
2.) Response spectra results were provided to a frequency of 100 Hz. Therefore, given the discussion in Mertz et al. (2015), the maximum time step should not exceed 0.00125 seconds.

Table 1 shows that a large variety of CPU times are possible with using explicit only model runs. In the case study, the range of CPU times varies from significantly faster than implicit only or co-simulation model runs to significantly slower than implicit only or co-simulation model runs. The CPU time for the implicit only and co-simulation model runs varied minimally for the case study and generally showed co-simulation as a better option than implicit only relative to CPU time. Consequently, the case study showed that an explicit only model has the potential of being the most efficient relative to CPU time but co-simulation is the best in some cases.

For information, the model shown in Figure 1 was run for 72 hours as an Abaqus/Explicit only run using 28 CPUs. It achieved a model time of 0.0809 seconds (or 809 hours per second of model time). The model shown in Figure 2 was run for 22.3 hours as a co-simulation run using 2 CPUs for Abaqus/Explicit and 26 CPUs for Abaqus/Standard. It achieved a model time of 27.5 seconds (or 0.811 hours per second of model time). For this model, nothing was done to intentionally make it run poorly in an Abaqus/Explicit only model run yet, as a co-simulation, it ran almost 1000 times faster using the same number of CPUs.

CONCLUSION

For NLSSI analysis to be widely implemented in industry, it is important to demonstrate that accurate response in dynamically loaded, nonlinear models can be achieved in an efficient manner. The case study provided in this paper demonstrates that equivalent response can be achieved with a dynamic solver using explicit, implicit, or co-simulation formulation. It goes on to show that a wide variety of CPU times can be expected depending on the dynamic solver that is used. In a situation where the structural model does not significantly change the required explicit time step from that required by the soil, an explicit solver is likely the best choice relative to CPU time. In a situation where the structural model does significantly change the required explicit time step from that required by the soil, a co-simulation solver is likely the best choice relative to CPU time.

Additionally, a modification to the configure file is presented that increases the stability of co-simulation with Abaqus (2018) when the minimum stable time step could be determined by either Abaqus/Explicit or Abaqus/Standard.

REFERENCES

American Society of Civil Engineers (ASCE) (2016). *Seismic Analysis of Safety-Related Nuclear Structures and Commentary*, ASCE/SEI 4- 2016, Reston, VA, USA.

Mertz, G. E., Spears, R. E., and Houston, T. W. (2015). “The Effects of Discretization Errors on the High Frequency Content of In-Structure Response Spectra,” PVP2016-63679, Proceedings of the ASME 2016 Pressure Vessels and Piping Conference, PVP2016, Vancouver, British Columbia, Canada, July 2016.

Abaqus (2018), Version 2018.HF3, Dassault Systèmes Simulia Corporation, Providence, Rhode Island.