HEEDS Technical Tip – Quickly Identifying Potential Runtime Errors in a HEEDS Setup

Level: Intermediate

Last revision: 12/09/2009

Introduction

Before launching a HEEDS study, it is important to make sure that there are no errors in the model setup. Any unresolved errors can result in the loss of valuable time. For example, a HEEDS study with undetected runtime errors, executed to run unmonitored over the weekend, could result in the loss of up to three days of computational time.

Although HEEDS provides various built-in features (such as color coding, data validation, and value extraction) to help you identify and fix the majority of setup errors, some errors cannot be detected until at least a few designs have been evaluated. These errors fall into the "runtime error" category.

The only way to identify and resolve runtime errors is to allow the HEEDS study to complete at least one full design iteration, but preferably a few design iterations. This is practical in studies where the design evaluations are short (only a few minutes). However, for studies with lengthy design evaluations, using full evaluations is not the most efficient way to identify and fix potential setup errors.

This tip will describe a few different techniques you can use to quickly identify potential runtime errors without performing complete design evaluations.

Various Techniques for Runtime Error Checking

The key to reducing the time involved in runtime error testing is to find a way to shorten the evaluation time for the purposes of testing. If the evaluations can be shortened, any errors can be detected and resolved quickly, saving significant time as compared to running the unaltered, long evaluations.

Some methods for shortening evaluations include

1. **Using mass scaling.** A high value for mass scaling can be used to speed up long explicit finite element analyses. The mass scaling level can be defined as a parameter in the HEEDS project, so you can easily change it back to the correct value once the testing phase has been completed.
2. **Using shorter total analysis time.** The total time for the simulation model can be reduced to shorten the evaluation time. For example, a 120 ms analysis can be set up to run for 2 ms, reducing the evaluation time by a factor of 60.
3. **Using a linear static step instead of a nonlinear step.** For non-linear implicit finite element jobs, the load step definition can be changed from the non-linear definition to a linear definition to shorten the evaluation time.
4. **Using a coarse mesh in a shape optimization problem.** In studies where the shape is being changed at the CAD level, and the part is being remeshed for each design, the mesh can be coarsened significantly by using a larger mesh seed. A coarser mesh will result in a smaller model, which would then run more quickly.

Important! With all of these approaches, remember to return the parameters you used to shorten the evaluation time to the correct values before you execute the final HEEDS study.

Also, the results of these simplified analyses will potentially be very inaccurate. The accuracy of the analysis results does not matter, however, because the only purpose of the simplified run is to test the process automation steps.

Example

This example uses Abaqus as the analysis solver. The default time step for the analysis is 1.0e-6.

Step 1

Define a parameter in HEEDS for tagging the time step in the mass scaling definition.

6	bead3_H	Continuous	-1	0	1	
7	bead4_W	Continuous	-2	0	1	
8	bead4_H	Continuous	-1	0	1	
9	bead_Depth	Continuous	-0.25	0	1	
10	thickness	Continuous	1	1	4	
▶	Time_Step	Parameter		1.0e-6		

Figure 1. The new parameter "Time_Step" has been added to control the mass scaling used in the analysis model in this project.

Step 2

Tag the time step in the mass scaling definition of the Abaqus input file, so that it can be updated for every design to the value defined in the parameter definition.

171	**	Mass	Scaling:	Semi-Automat				
172	**	Whole	Model					
▶	*Variable	Mass	Scaling	dt	Time_Step	type	below	min
174	**							
175	**	OUTPUT	REQUESTS					
176	**							

Figure 2. The new parameter "Time_Step" has been tagged in the input file.

Step 3

To get ready for the testing phase, change the value of the "Time_Step" parameter to the value for larger mass scaling. This will result in shorter evaluation times for quicker evaluation of any runtime errors in the setup. In this example the value was changed to 1.0e-4. This will produce a speedup factor of 100 in the evaluation time.

6	bead3_H	Continuous	-1	0	1		
7	bead4_W	Continuous	-2	0	1		
8	bead4_H	Continuous	-1	0	1		
9	bead_Depth	Continuous	-0.25	0	1		
10	thickness	Continuous	1	1	4		
▶	Time_Step	Parameter		1.0e-4			

Figure 3. The value of the "Time_Step" parameter has been changed to the larger value for use in the runtime error testing setup.

Step 4

Execute the study, and post process the results to make sure there are no errors. If no errors are detected, you can proceed to step 5. Otherwise, the errors should be resolved before continuing to step 5.

Step 5

Change the value of the "Time_Step" parameter back to the value that should be used in the final study. Save the project and execute the study.

6	bead3_H	Continuous	-1	0	1	
7	bead4_W	Continuous	-2	0	1	
8	bead4_H	Continuous	-1	0	1	
9	bead_Depth	Continuous	-0.25	0	1	
10	thickness	Continuous	1	1	4	
▶	Time_Step	Parameter		1.0e-6		

Figure 4. The value of the "Time_Step" parameter has been changed back to the original value for use in the final setup.