Automated Installation Verification of COMSOL via LiveLink™ *for* MATLAB®

Michael W. Crowell
Oak Ridge National Laboratory, PO Box 2008 MS6423, Oak Ridge, TN 37831
crowellmw@ornl.gov

Abstract: Verifying that a local software installation performs as the developer intends is a potentially time consuming but necessary step for nuclear safety related codes. Automating this process not only saves time, but can increase reliability and scope of verification compared to 'hand' comparisons. While COMSOL does not include automatic installation verification as many commercial codes do, it does provide tools such as LiveLink™ *for* MATLAB® and the COMSOL API for use with Java® through which the user can automate the process. Here we present a successful automated verification example of a local COMSOL 5.0 installation for nuclear safety related calculations at the Oak Ridge National Laboratory's High Flux Isotope Reactor (HFIR).

Keywords: automation, software verification, software quality assurance.

1. Introduction

In the past, installation verification of COMSOL for HFIR nuclear safety related calculations has been performed by hand and in an ad-hoc manner. This has limited the precision of the individual verifications, the consistency of the verifications over time, the number of versions of COMSOL verified, and the number of machines on which any version of COMSOL has been verified. In this current effort to verify the installation of COMSOL v5.0.1.276 on our local computer cluster, we also seek to automate and standardized the verification process somewhat. We expect this automated process to yield increased precision and consistency, and dramatically speed up the verification process such that updated COMSOL releases can be verified more frequently, and on as many local machines as desired, with minimal personnel effort.

2. Methods

We have chosen to work via the LiveLink™ *for* MATLAB® functionality offered in COMSOL, which allows full access to the COMSOL Application Programming Interface (API): creating, editing, running, and extracting results, through the MATLAB® command line and scripting interfaces. Identical access to the COMSOL API is provided through the Java® programming language with the current COMSOL API for use with Java® support. Although this verification effort could have been completed with either the MATLAB® or Java® interfaces to COMSOL, MATLAB® was selected due to the author's familiarity with it, and its ease of use for technical computing.

In order to verify the installation of COMSOL, it is necessary to have models with results already provided by COMSOL ('included' results) which can be rerun on the local machine where the COMSOL installation is being verified. This requirement is satisfied by the model library that is included with a COMSOL installation wherein many of the models are provided with already computed results. Once these models are rerun on the local machine, these 'local' results can be compared with the included results and if the differences are sufficiently small the COMSOL installation will be considered verified. Note that due to differences in machine architecture and mathematical libraries, the limitations of machine precision, and the relative error tolerance convergence criteria implemented in COMSOL, the included and local results are not expected to be identical. We must therefore decide which differences are 'sufficiently small' and which are not. To do this, we must choose both a difference metric and a threshold. For the metric, we have defined a relative difference per dependent variable, $d_r^{u_i}$, that is the maximum difference between the local and included results, normalized by the largest value in the included results for that particular dependent variable:

$$d_r^{u_i} = \max\left(\frac{\left|\bar{u}_i^{loc} - \bar{u}_i^{inc}\right|}{\max\left(\left|\bar{u}_i^{inc}\right|\right)}\right)$$

Here u_i is a particular dependent or 'solution' variable, $\bar{u}_i$ is the vector of the values for that variable at all the model node points for each parameter sweep, if any, and the 'inc' and 'loc' superscripts refer to the included solution and locally computed solution respectively. Then we take the maximum relative difference over all the dependent variables in a model, and that is the relative difference for the model, d_r^m:

$$d_r^m = \max_i\left(d_r^{u_i}\right)$$

Our approach was to define the threshold for 'sufficiently small' from a practical standpoint as being less than typical measurement uncertainty for our problems of interest. We determined this to be a per model relative difference of less than 1.0×10^{-3}. This equates to the maximum difference of any dependent variable at any node point being less than 1/10th of 1 percent of the maximum or 'full scale' value for that variable across the model. Now this per model relative difference must obviously be bounded by the relative error tolerance implemented in COMSOL. The exact relationship is nontrivial to determine due to differences in formulation of the two metrics, but if the relative error tolerance is set too large we could be unable to meet our per model relative difference threshold even with a correctly functioning COMSOL installation. Therefore part of this verification process will necessarily be to determine if the default COMSOL relative error tolerance of 1.0×10^{-3} is sufficient to guarantee that the per model relative difference, as we have defined it here, is less than our prescribed threshold of 1.0×10^{-3}.

Finally note that calculating d_r^m as we have defined it requires calculating the differences between the included and locally computed model results for all dependent or 'solution' variables at all model node points for all model parameter sweeps. This approach is desirable in that even very minor discrepancies will be identified, but it would not be feasible without an automated approach such as the one we have employed. However, with automation this approach can be both faster and more reliable/repeatable than previous verification efforts.

3. Models

In selecting which models to consider, we first searched the model library included with our local COMSOL installation for models with included results. From the models with included results we determined what physics features were being used and compiled a master list, organized by COMSOL module, of the physics features available for verification (Table 1). From this list we down-selected to the physics features most relevant/needed for HFIR related calculations, which can be found in Table 2, again organized by COMSOL module. Finally, from the list of models with included results we determined a minimal set which utilized the desired physics features from Table 2. The resulting model/physics matrix is found in Table 3.

One model, in addition to the 11 found in Table 3, was also considered due to a known error in the current COMSOL release. This error is an incorrect implementation of the periodic boundary condition for fluid domains where there are wall functions applied to the periodic boundary. To establish that this error does not occur when there are no wall functions on the periodic boundary, the model 'circuit board forced 3d' was also considered.

4. Results

For the 12 models considered in this verification study, the range of d_r^m values calculated was from a minimum of 7.4359×10^{-16} for the 'disk stack heat sink' model to a maximum of 9.3887×10^{-05} for the 'naca0012 airfoil' model. Machine precision for double precision floating point numbers is $\approx 1.11 \times 10^{-16}$, so the minimum d_r^m value was on the order of machine precision while the maximum was still well below our practical threshold of 1.0×10^{-3}. In general, models with turbulent fluid flow (naca0012 airfoil, ahmed body, displacement ventilation, circuit board forced 3d) or nonlinear solid mechanics (arterial wall mechanics) had the highest d_r^m values while models with only heat transfer and/or linear solid mechanics had significantly lower d_r^m values.

The d_r^m values for all 12 models can be found below in graphical form in Figure 1 and in tabular form in Table 4. The relative difference calculation was also performed on the mesh coordinates of each model, and the maximum value across all 12 models was 5.3291×10^{-16} which is on the order of machine precision, as expected. This value is shown graphically by the green horizontal line in Figure 1.

5. Conclusions

All d_r^m values were well below our threshold of 1.0×10^{-3} and thus we may conclude that the current COMSOL installation on our local computer cluster is verified for the physics of interest as listed in Table 2. We may further conclude that the default COMSOL relative error tolerance of 1.0×10^{-3} is sufficient for meeting our current verification metric/threshold combination. Finally, with the MATLAB® scripts now in place, the above verification study can be repeated on any local computer desired with minimal user input and with the time required governed only by how long the models take to run on the given machine (~1/2 day on our local computer cluster). The above study can also be expanded to include any or all of the additional physics features available in COMSOL (by using more of the available models) in a straightforward manner, with the time required again governed only by how long it takes to run all the models on the given local machine.

6. Notes

As the conference paper length restrictions preclude inclusion of the MATLAB® automation scripts herein, please feel free to contact the author if interested in the scripts themselves.

Table 1. Available physics in COMSOL database models with included results.

AC/DC:	**Heat Transfer:**
Charged Particle Tracing	Bioheat Transfer
Electric Currents	Heat Transfer
Electrostatics	Heat Transfer in Fluids
Magnetic Fields	Heat Transfer in Pipes
	Heat Transfer in Porous Media
Acoustics: Pressure Acoustics, Frequency Domain	Heat Transfer in Solids
	Heat Transfer in Thin Shells
Chemical Species Transport:	Heat Transfer with Radiation in Participating Media
Reacting Flow	Heat Transfer with Surface-to-Surface Radiation
Transport of Diluted Species	Radiation in Participating Media
Fluid Flow:	**Mathematics:**
Brinkman Equations	Boundary ODEs and DAEs
Creeping Flow	Coefficient Form Boundary PDE
Euler-Euler Model, Laminar Flow	Coefficient Form PDE
Fluid-Structure Interaction	Convection-Diffusion Equation
Free and Porous Media Flow	Curvilinear Coordinates
High Mach Number Flow	Deformed Geometry
Laminar Flow	General Form Boundary PDE
Laminar Two-Phase Flow, Level Set	General Form PDE
Laminar Two-Phase Flow, Phase Field	Global ODEs and DAEs
Mixture Model, Laminar Flow	Laplace Equation
Mixture Model, Turbulent Flow	Mathematical Particle Tracing
Non-Isothermal Pipe Flow	Moving Mesh
Particle Tracing for Fluid Flow	Optimization
Pipe Flow	Phase Field
Rotating Machinery, Laminar Flow	
Rotating Machinery, Turbulent Flow, k-ε	**Microfluidics:** Slip Flow
Thin-Film Flow, Edge	
Thin-Film Flow, Shell	**RF:** Electromagnetic Waves, Frequency Domain
Turbulent Bubbly Flow	
Turbulent Flow, SST	**Structural Mechanics:**
Turbulent Flow, k-ε	Beam
Turbulent Flow, k-ω	Beam Cross Section
Water Hammer	Membrane
	Shell
	Solid Mechanics
	Truss

Table 2. Subset of available physics desired for HFIR related calculations.

Fluid Flow:	**Mathematics:**
Fluid-Structure Interaction	Boundary ODEs and DAEs
Turbulent Flow, SST	Curvilinear Coordinates
Turbulent Flow, k-ε	Deformed Geometry
Turbulent Flow, k-ω	Global ODEs and DAEs
Heat Transfer:	**Structural Mechanics:**
Heat Transfer	Shell
Heat Transfer in Fluids	Solid Mechanics
Heat Transfer in Solids	
Heat Transfer in Thin Shells	

Table 3. Model/desired physics matrix.

Physics:	Fluid Flow				Heat Transfer				Mathematics				Struct. Mech.	
Models:	Fluid-Structure Interaction	Turbulent Flow, SST	Turbulent Flow, k-ε	Turbulent Flow, k-ω	Heat Transfer	Heat Transfer in Fluids	Heat Transfer in Solids	Heat Transfer in Thin Shells	Boundary ODEs and DAEs	Curvilinear Coordinates	Deformed Geometry	Global ODEs and DAEs	Shell	Solid Mechanics
oscillating fsi	x													
naca0012 airfoil		x												
ahmed body			x											
displacement ventilation				x		x								
heat sink surface radiation					x									
disk stack heat sink							x	x						
cohesive zone debonding									x					x
arterial wall mechanics										x				x
electrochemical polishing											x			
diffuse double layer												x		
bracket shell													x	x

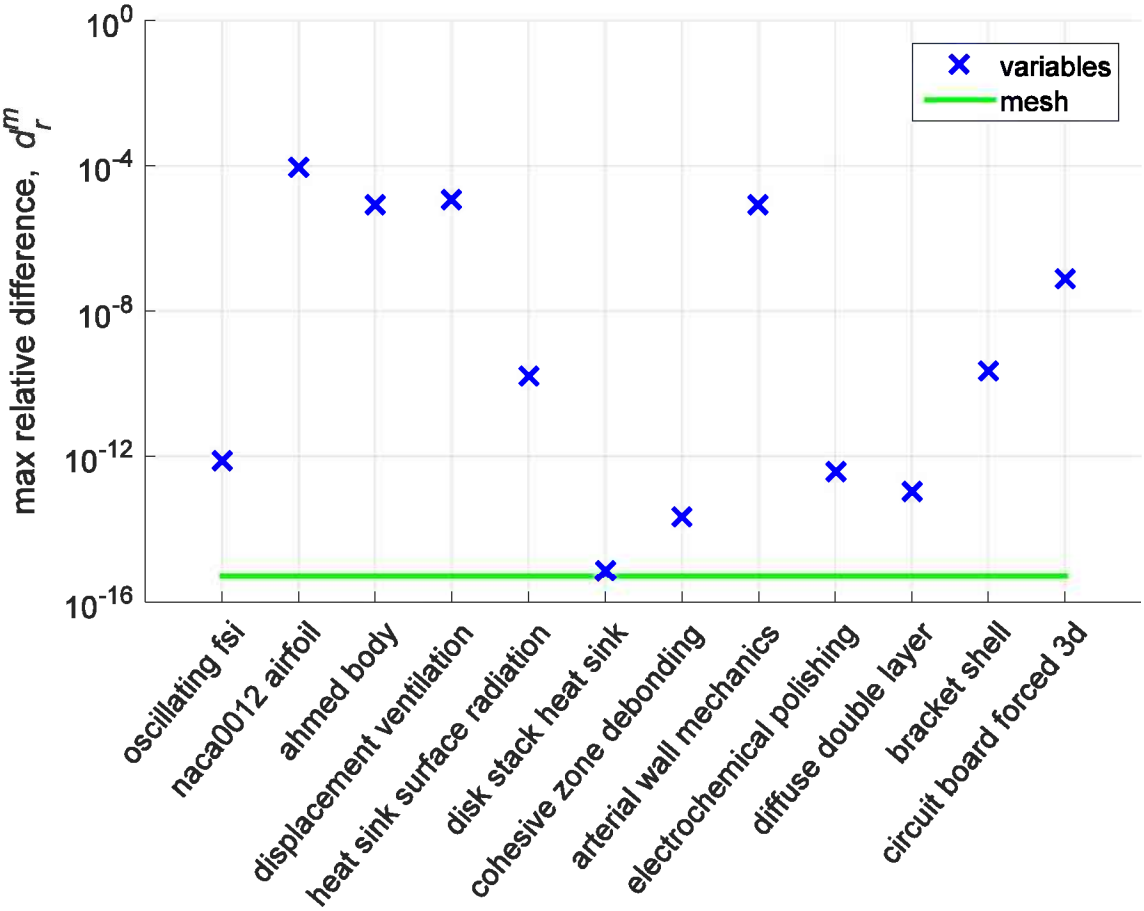

Figure 1. COMSOL provided versus local computer cluster computed results.

Table 4. Model maximum relative differences.

model	d_r^m
oscillating_fsi	8.0845e-13
naca0012_airfoil	9.3887e-05
ahmed_body	8.8960e-06
displacement_ventilation	1.1608e-05
heat_sink_surface_radiation	1.5937e-10
disk_stack_heat_sink	7.4359e-16
cohesive_zone_debonding	2.2603e-14
arterial_wall_mechanics	8.5540e-06
electrochemical_polishing	3.9108e-13
diffuse_double_layer	1.1128e-13
bracket_shell	2.3681e-10
circuit_board_forced_3d	8.1790e-08