User vs. COMSOL® Developed Automated Installation Verification of COMSOL Multiphysics® Software

Michael Crowell
crowellmw@ornl.gov
High Flux Isotope Reactor
Research Reactors Division
Neutron Sciences Directorate
Oak Ridge National Laboratory

Optimization and Simulation Methods Session

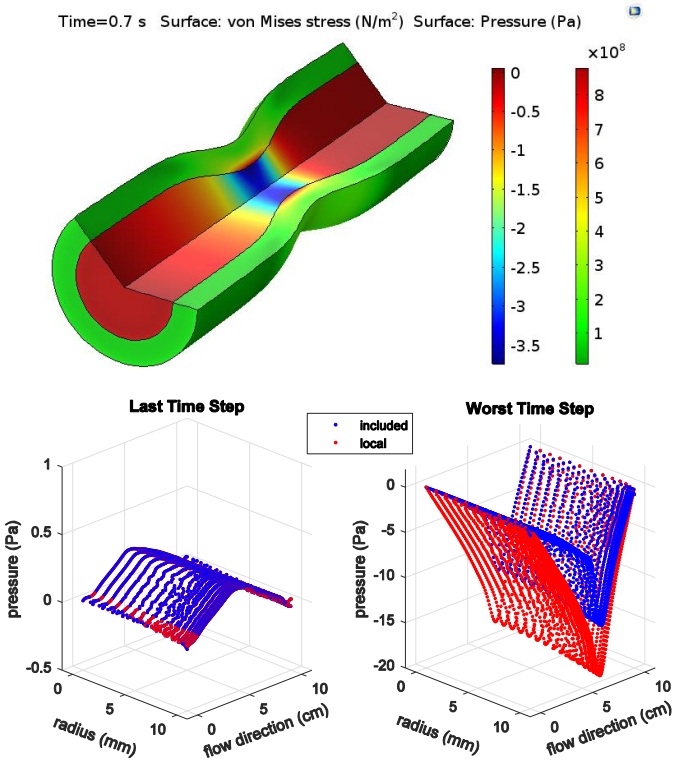
ORNL is managed by UT-Battelle
for the US Department of Energy

COMSOL CONFERENCE 2016 BOSTON

OAK RIDGE National Laboratory

Overview

- What is Software [Installation] Verification and Validation?
- Why Should *You* Care?
- What Should Software Installation Verification Look Like?
- User Scripts vs. COMSOL App
- Conclusions

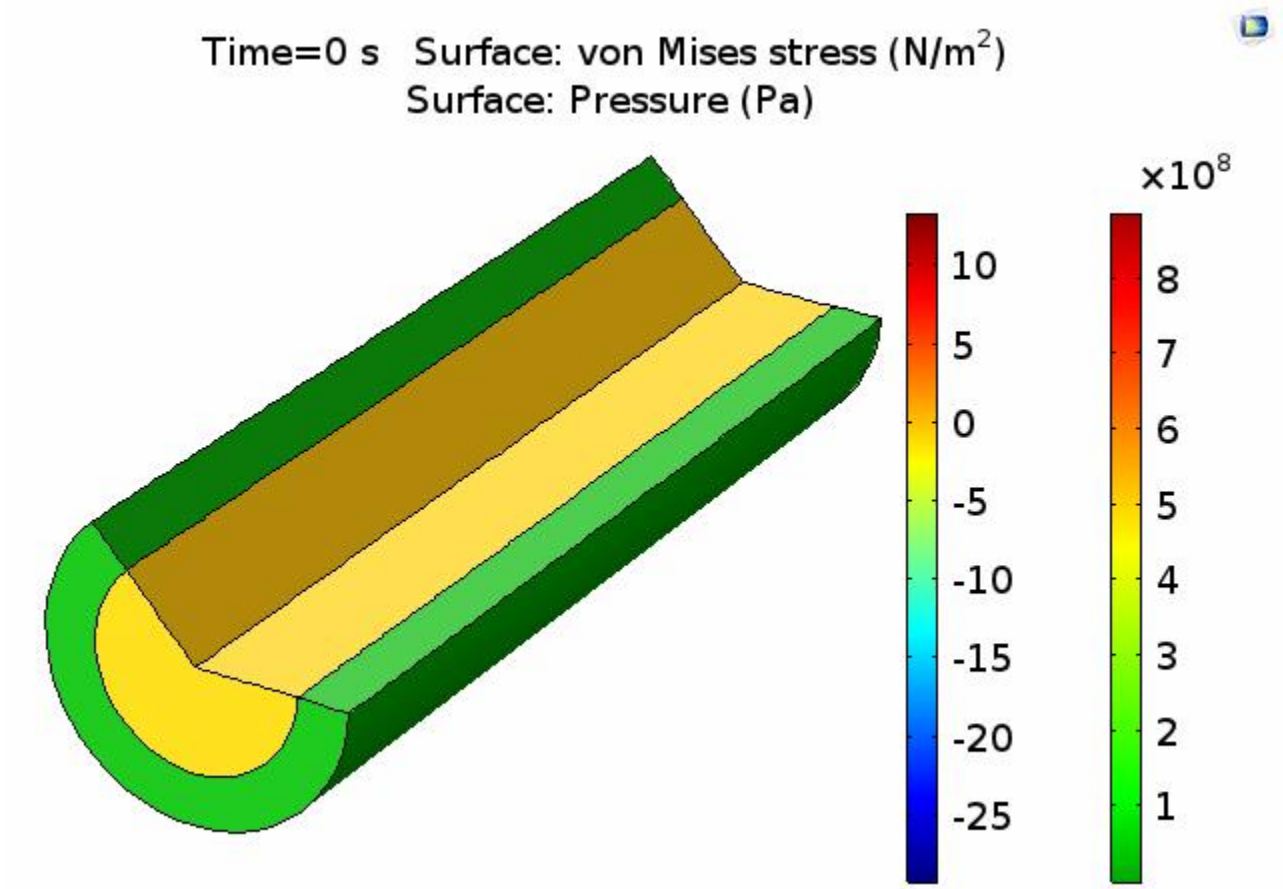


What is Software [Installation] Verification & Validation?

- Verification:
  - Solve given equations ‘correctly’
  - Installation Verification:
    - software performs on your machine as it did on the developer’s machine (i.e. as the developer intends).
  - Automation is very desirable and valuable
- Validation:
  - Show that equations/parameters are applicable to problem of interest
  - Ideally, compare to “real” data
  - Automation generally not as important

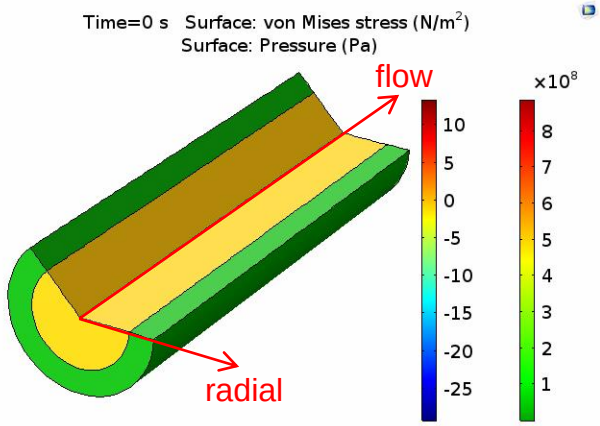
Why Should *You* Care?

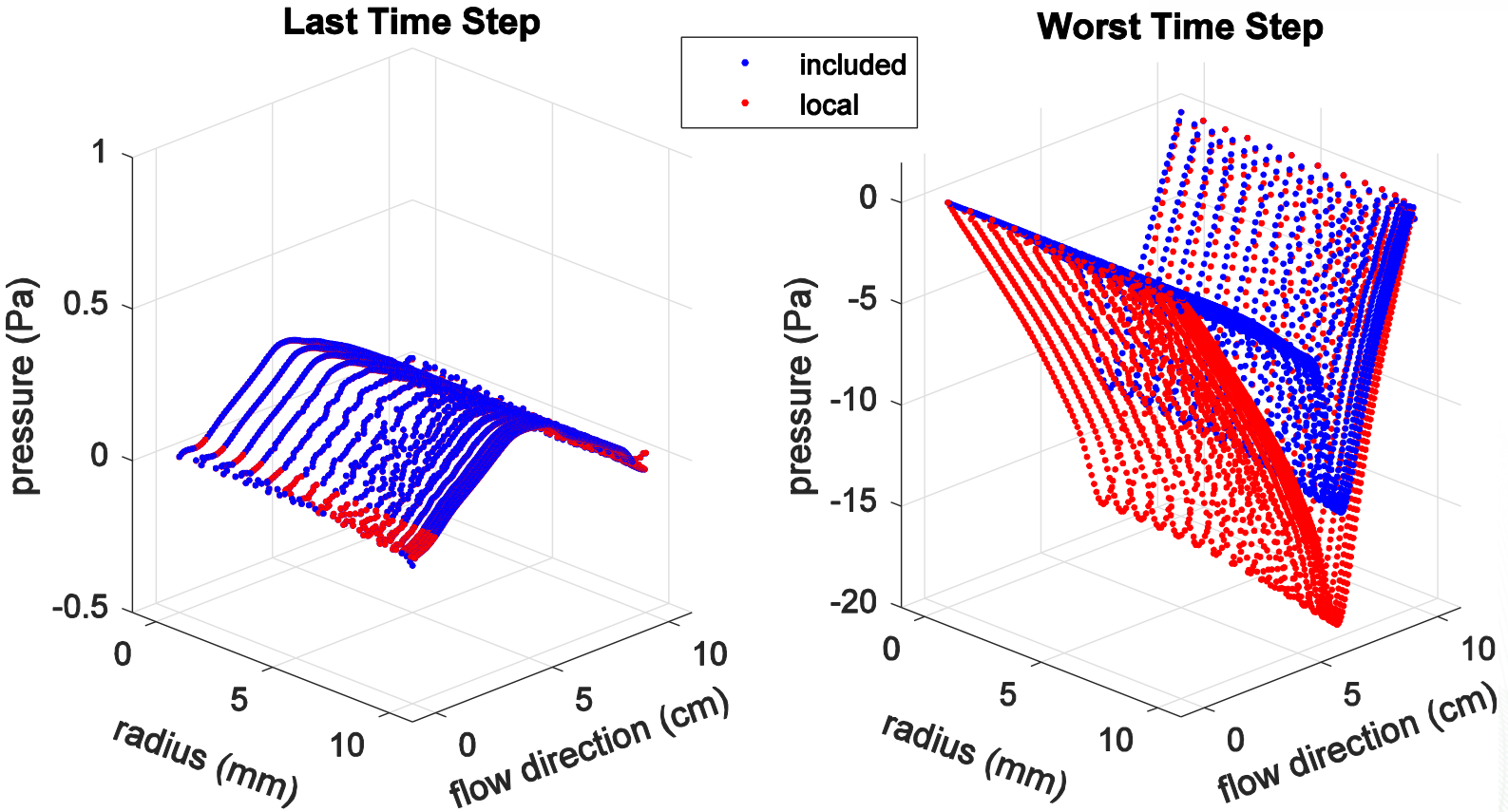
- Peristaltic Pump Model



Why Should *You* Care?

- Peristaltic Pump Model





What Should Software Installation Verification Look Like?

- Compare solutions* computed on a verified software installation with those computed on the new (to be verified) software installation.
  - Solutions from a verified software installation (usually the developer)
  - Solutions from the new software installation (your local installation)
  - Difference metric (absolute and/or relative)
  - Acceptance criteria/threshold

* Should include all solution variables at all node/integration points at all solution steps. This is why you want automation.

What Should Software Installation Verification Look Like?

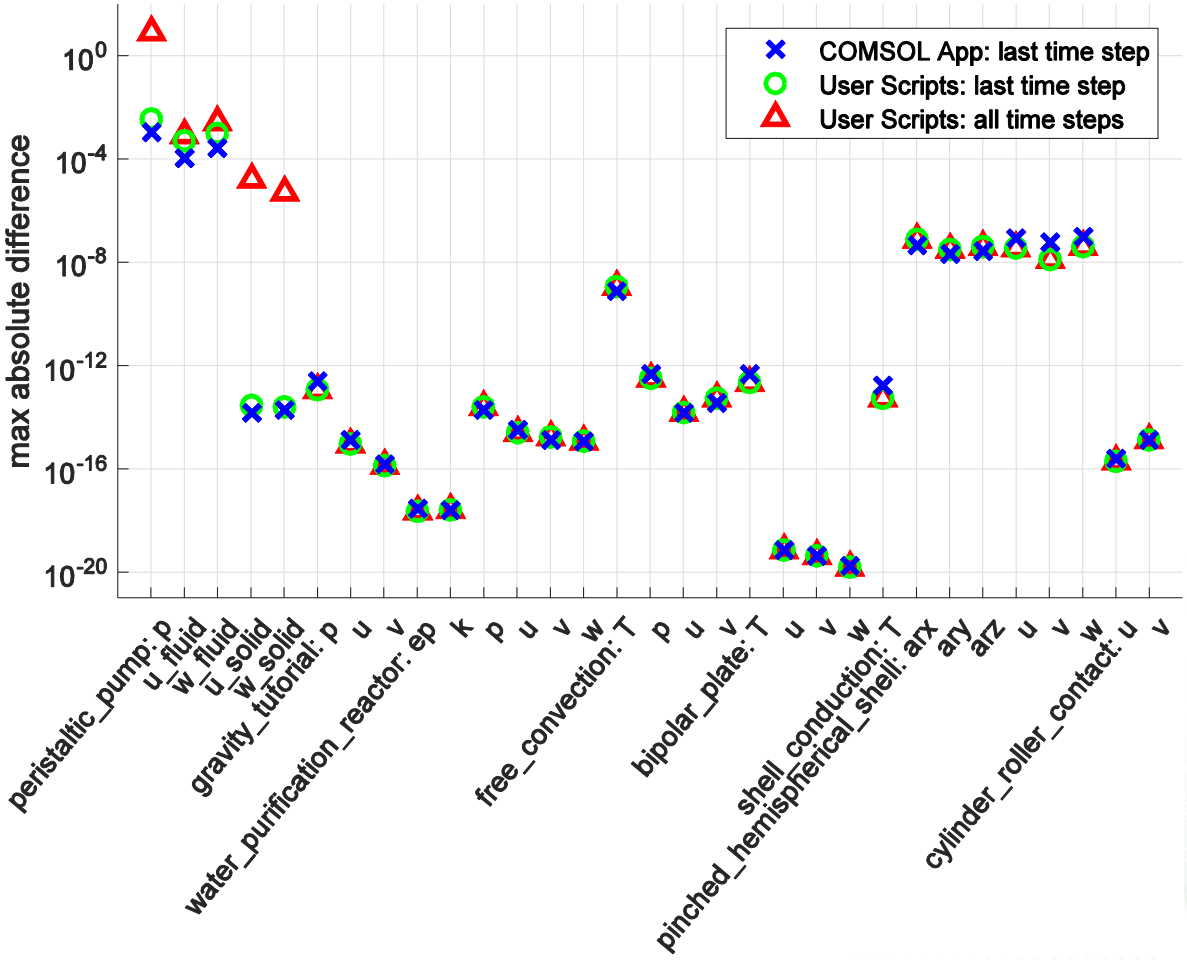
- COMSOL 5.2 at HFIR/ORNL example

Physics:	Fluid Flow					Heat Transfer			
Models:	Fluid-Structure Interaction	Laminar Flow	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
peristaltic_pump	x								
gravity_tutorial		x							
naca0012_airfoil			x						
water_purification_reactor				x					
pipe_elbow					x				
potcore_inductor		x				x			
free_convection		x					x		
bipolar_plate*								x	
shell_conduction									x
fluid_damper*		x				x			
tin_melting_front*		x				x			
circuit_board_forced_3d**		x				x			

Physics:	Mathematics								Struct. Mech.	
Models:	Coefficient Form Boundary PDE	Coefficient Form PDE	Curvilinear Coordinates	General Form Boundary PDE	General Form PDE	Global ODEs and DAEs	Deformed Geometry	Moving Mesh	Shell	Solid Mechanics
shell_diffusion	x									
black_sholes_put		x								
orthotropic_container			x							x
transport_and_adsorption				x						
spherically_symmetric_transport					x					
lorenz_attractor						x				
tin_melting_front*							x			
fluid_damper*								x		
pinched_hemispherical_shell									x	
cylinder_roller_contact										x
bipolar_plate*										x

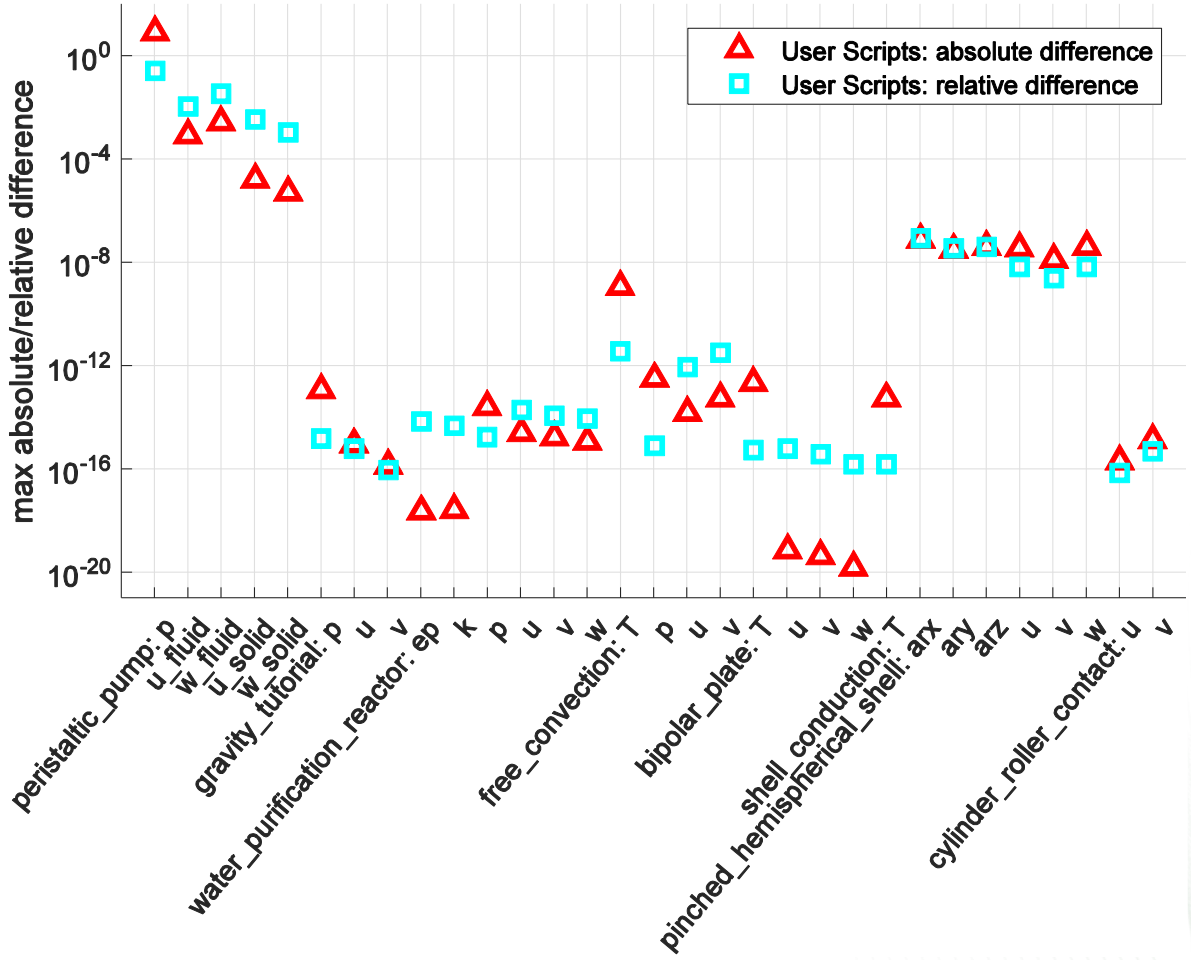
What Should Software Installation Verification Look Like?

- COMSOL 5.2 at HFIR/ORNL example



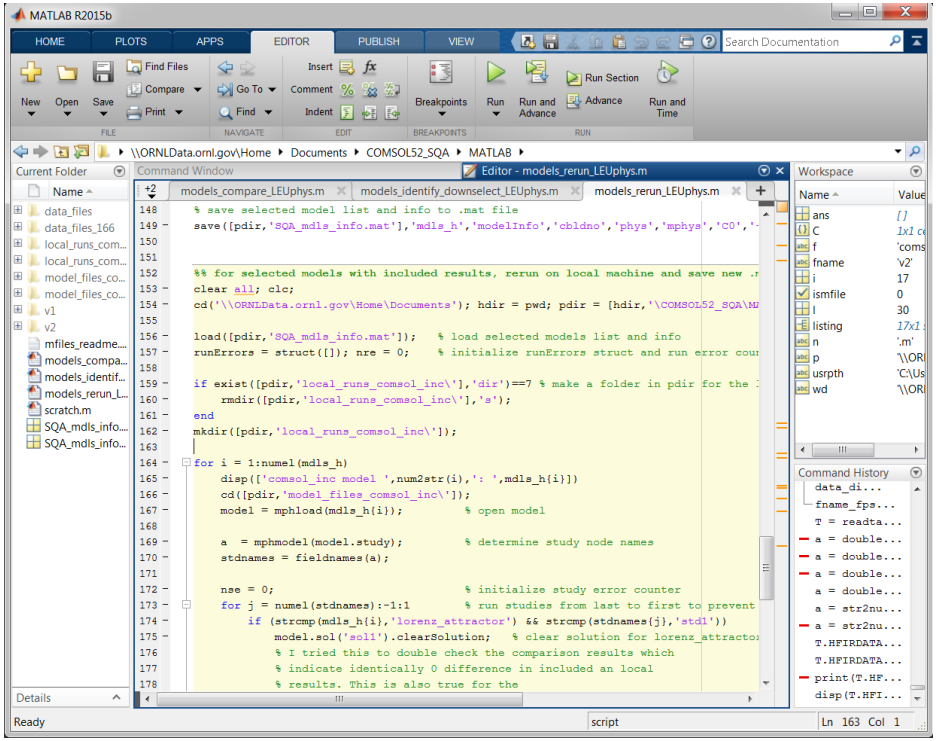
What Should Software Installation Verification Look Like?

- COMSOL 5.2 at HFIR/ORNL example

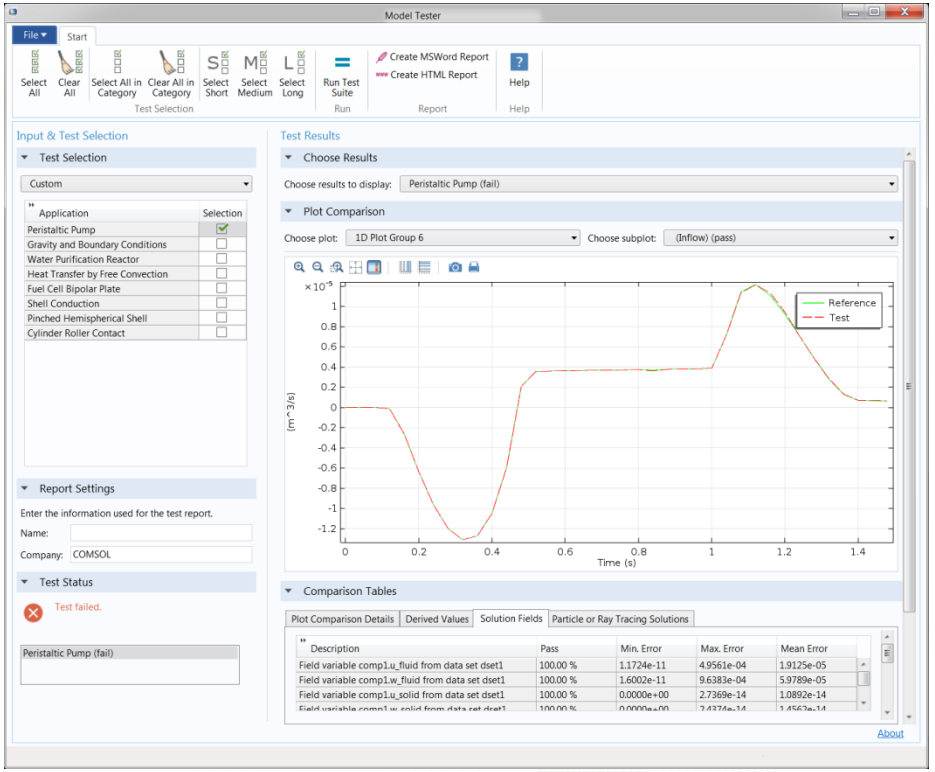


User Scripts vs. COMSOL App

User Scripts



COMSOL App



User Scripts vs. COMSOL App

COMSOL App: Pros	COMSOL App: Cons
More user friendly	More difficult to customize
Developed/Maintained by COMSOL	Outputs absolute but not relative differences
Auto Report Generation	Looks at results from final instead of all steps

User Scripts: Pros	User Scripts: Cons
Easy to customize	Less user friendly
All calculations accessible	Requires expert user development/maintenance
Looks at results from all steps	May require additional software (like Matlab)

Conclusions

- Installation Verification is for *everyone*
- It does not need to be onerous (if you use automation)
- I learn something new each time
- Stay tuned for COMSOL Verification/Validation App