



Software Testing 2

Example for Building a Test-Suite

Anshu Dubey
Argonne National Laboratory

Better Scientific Software Tutorial, SC20, November 2020



See slide 2 for
license details

License, Citation and Acknowledgements

License and Citation



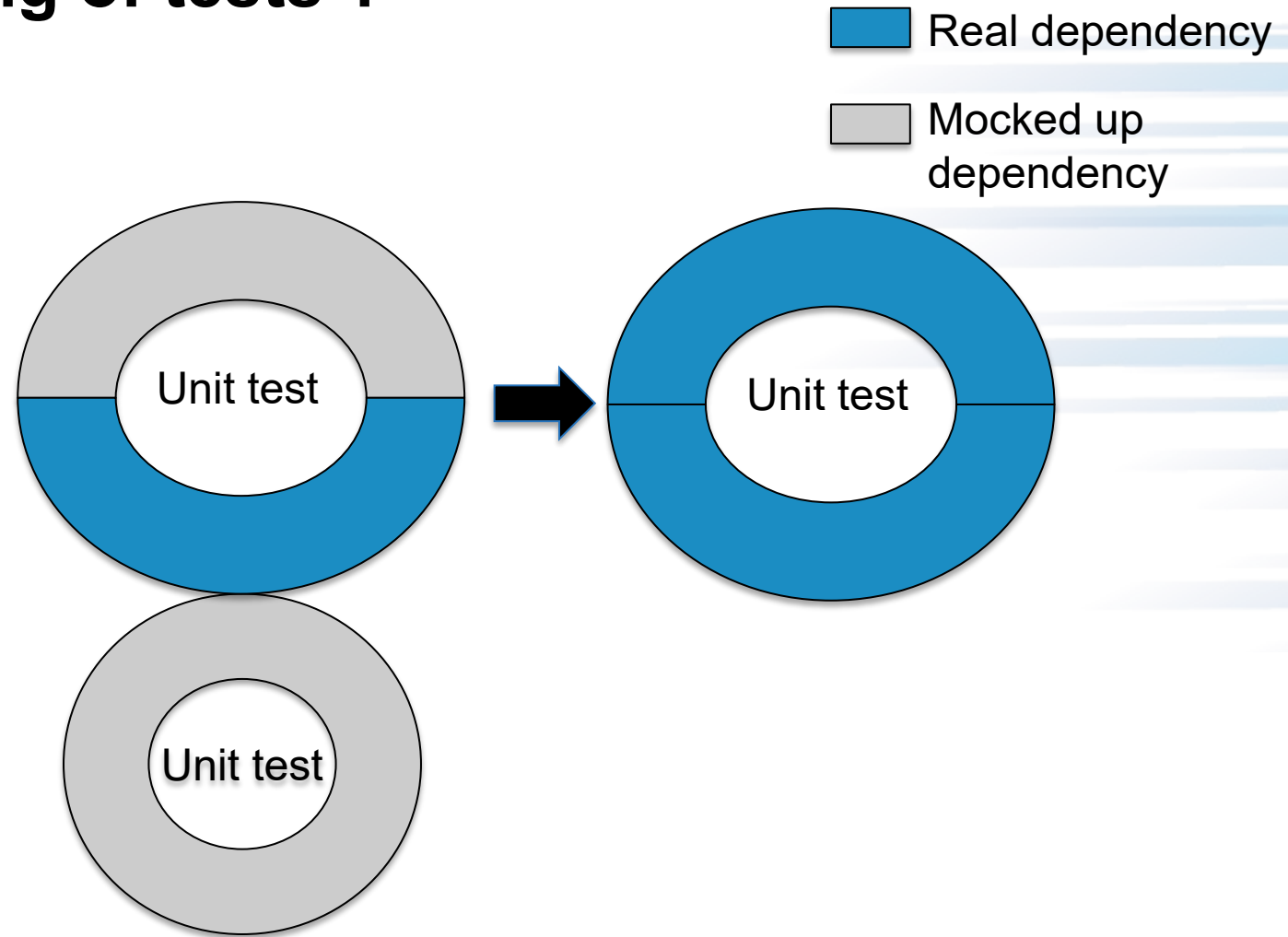
- This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/) (CC BY 4.0).
- **The requested citation the overall tutorial is: David E. Bernholdt, Anshu Dubey, Patricia A. Grubel, Rinku K. Gupta, Better Scientific Software tutorial, in SC '20: International Conference for High Performance Computing, Networking, Storage and Analysis, online, 2020. DOI: [10.6084/m9.figshare.12994376](https://doi.org/10.6084/m9.figshare.12994376)**
- Individual modules may be cited as *Speaker, Module Title*, in Better Scientific Software tutorial...

Acknowledgements

- Additional contributors include: Mike Heroux, Alicia Klinvex, Mark Miller, Jared O'Neal, Katherine Riley, David Rogers, Deborah Stevens, James Willenbring
- This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.
- This work was performed in part at the Argonne National Laboratory, which is managed by UChicago Argonne, LLC for the U.S. Department of Energy under Contract No. DE-AC02-06CH11357.
- This work was performed in part at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.
- This work was performed in part at the Lawrence Livermore National Laboratory, which is managed by Lawrence Livermore National Security, LLC for the U.S. Department of Energy under Contract No. DE-AC52-07NA27344.
- This work was performed in part at the Los Alamos National Laboratory, which is managed by Triad National Security, LLC for the U.S. Department of Energy under Contract No. 89233218CNA000001
- This work was performed in part at Sandia National Laboratories. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

How do you build a scaffolding of tests ?

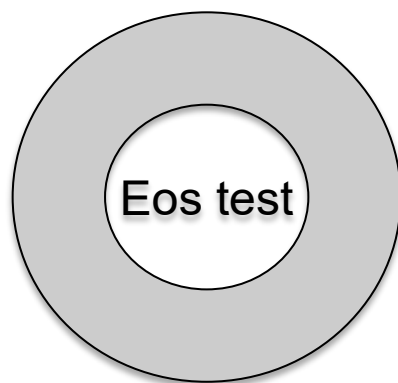
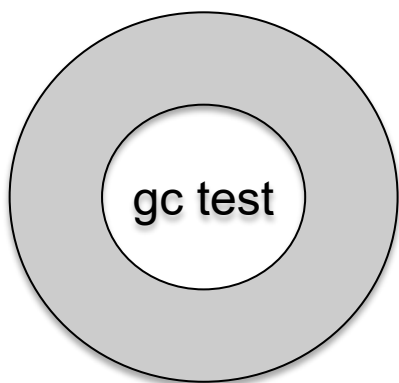
- Approach the problem sideways
 - Components can be exercised against known simpler applications
 - Same applies to combination of components
- Build a scaffolding of verification tests to gain confidence



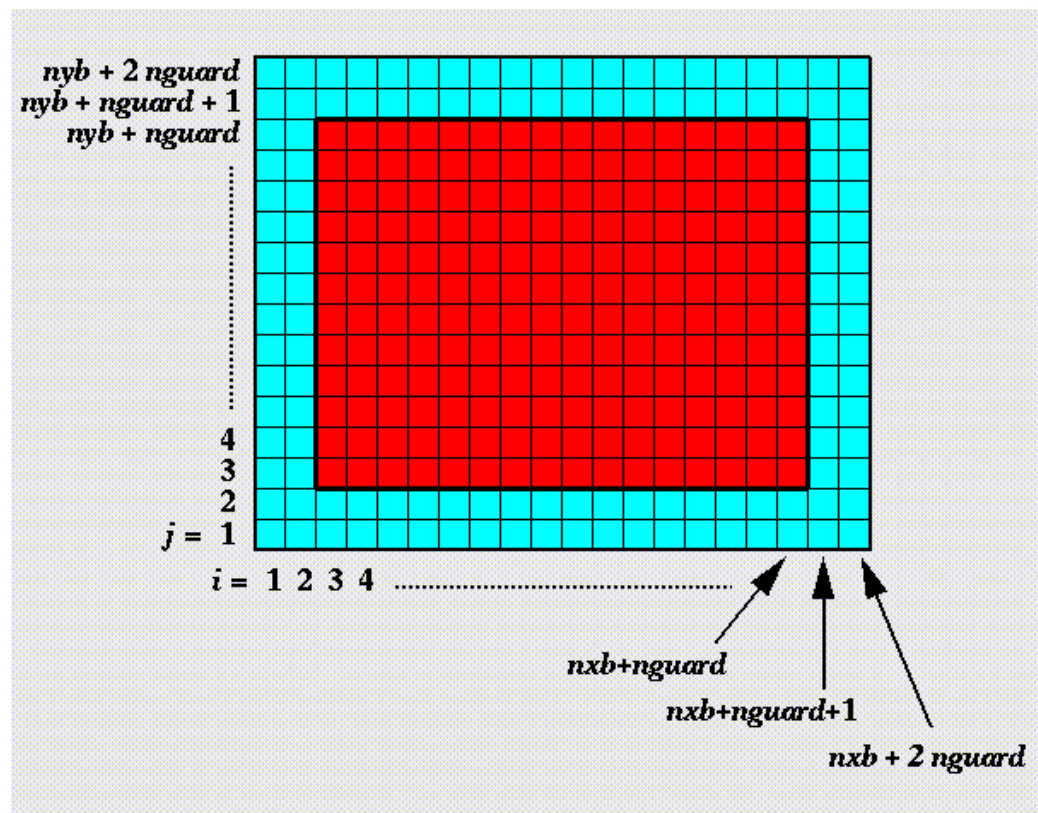
Scaffolding Example from FLASH

Unit test for Grid halo cell fill

- Verification of guard/ghost/halo cell fill
- Use two variables A & B
- Initialize A in all cells and B only in the interior cells (red)
- Apply guard cell fill to B



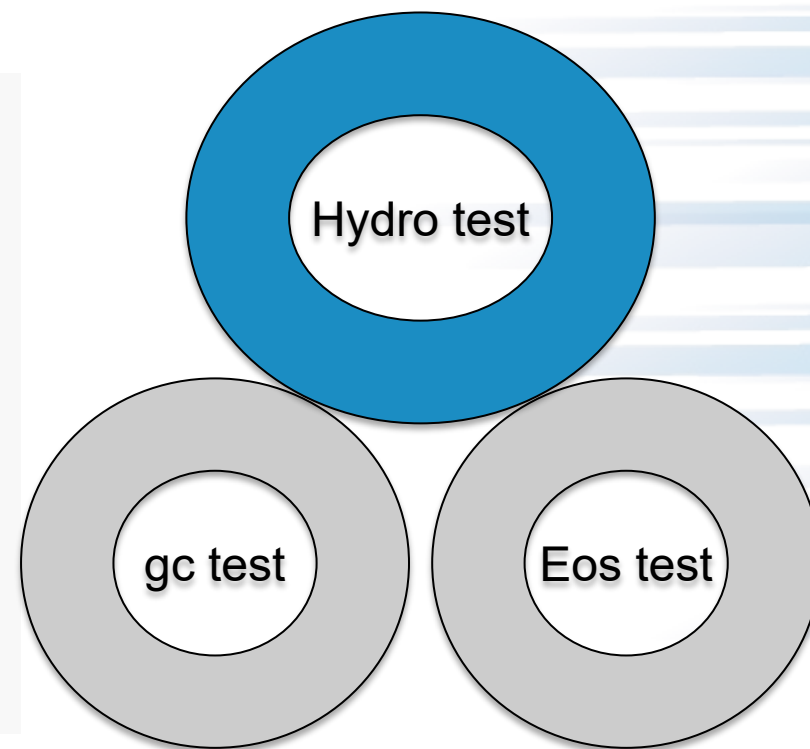
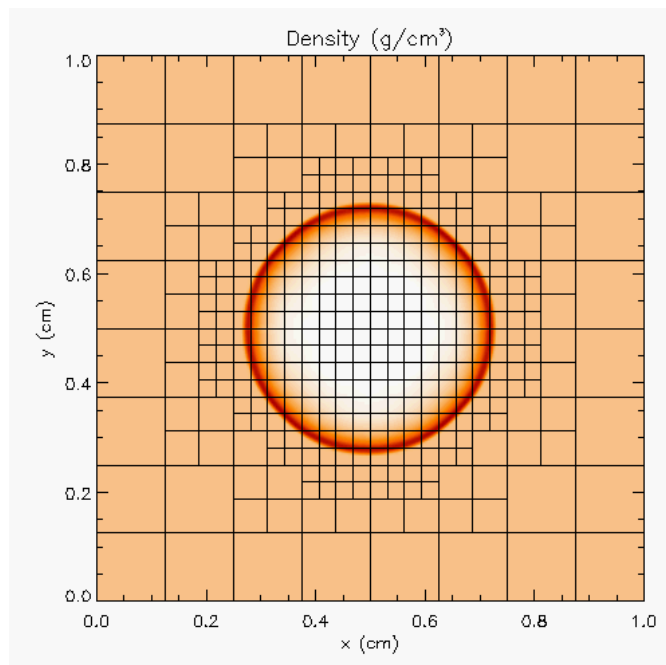
Similarly, it is possible to build Eos Test



Scaffolding Example from FLASH

Unit test for Hydrodynamics

- Sedov blast wave
- High pressure at the center
- Shock moves out spherically
- FLASH with AMR and hydro
- Known analytical solution



Though it exercises mesh, hydro and eos, if mesh and eos are verified first, then this test verifies hydro

More testing needed for Grid using AMR
Flux correction and regridding

Scaffolding Example from FLASH

For AMR, correct behavior of flux conservation and regridding should also be verified.

Reason about correctness for testing Flux correction and regridding

IF Guardcell fill and EOS unit tests passed

- Run Hydro without AMR
 - If failed fault is in Hydro
- Run Hydro with AMR, but no dynamic refinement
 - If failed fault is in flux correction
- Run Hydro with AMR and dynamic refinement
 - If failed fault is in regridding

Test Selection

First line of defense – code coverage tools

- Code coverage tools necessary but not sufficient
- Do not give any information about interoperability

	Hydro	EOS	Gravity	Burn	Particles
AMR	CL	CL		CL	CL
UG	SV	SV			SV
Multigrid	WD	WD	WD	WD	
FFT			PT		

- Build a functionality matrix
- Mark $\langle i,j \rangle$ if test covers corresponding features, and is a valid combination
- Follow the order
 - All unit tests – including full module tests
 - Tests sensitive to perturbations
 - Most stringent tests for solvers
 - Least complex test to cover remaining spots

TAKEAWAYS

- TESTING AT VARIOUS GRANULARITIES HELPS BUILD CONFIDENCE
 - DEVISE TESTS TO ENABLE QUICK PINPOINTING OF ERRORS THROUGH REASONING ABOUT THEIR BEHAVIOR
 - SAME APPLIES TO ENSURING COVERAGE
 - MAKE SURE THAT YOUR TESTS FAIL WHEN THEY SHOULD
-QUESTIONS ?