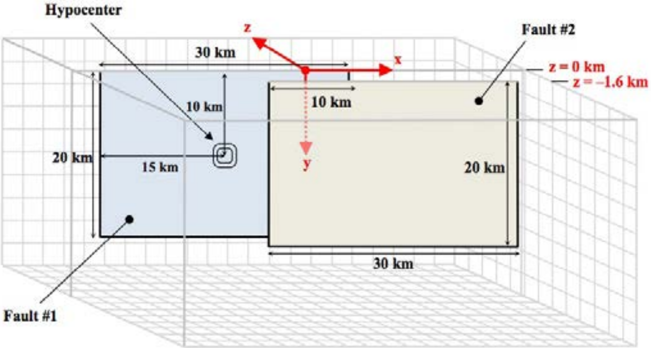
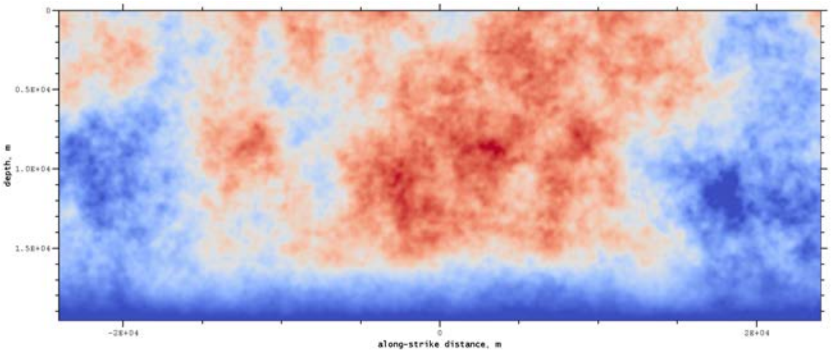
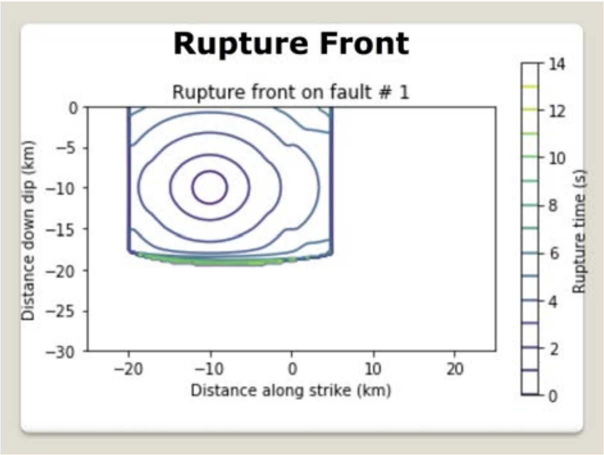
SCEC Dynamic Rupture Benchmarks in the Classroom

Eric G. Daub
Center for Earthquake
Research and Information
University of Memphis

THE UNIVERSITY OF MEMPHIS
Center for Earthquake Research and Information

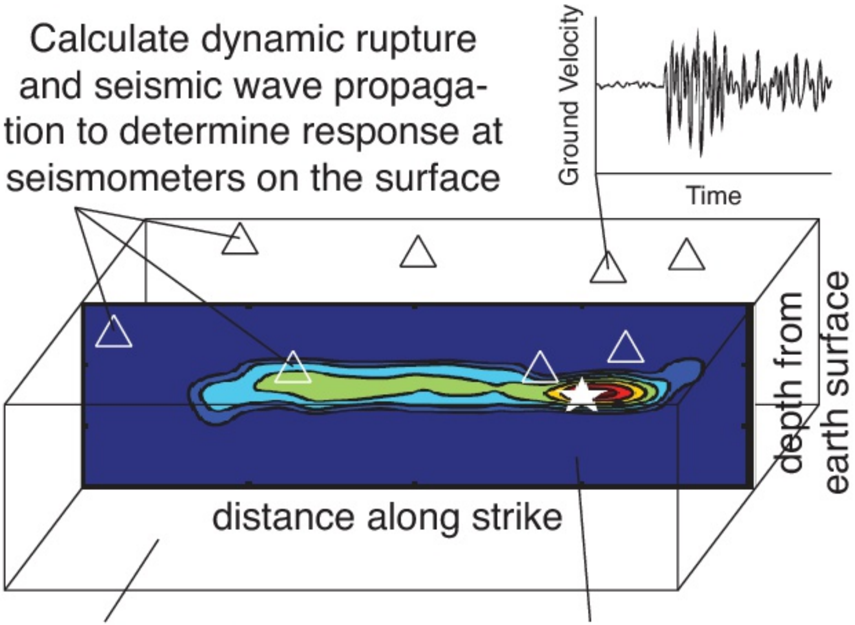






Earthquake Rupture Modeling

Finite Difference Code: at CERI, we use my finite difference code for research and teaching purposes



Calculate dynamic rupture and seismic wave propagation to determine response at seismometers on the surface

Rock: treated as elastic, elastic-plastic, visco-elastic, etc.

Fault: obeys a friction law that describes failure process

Relevant features for teaching:

-- Goal is to make the code easy for novice use

-- Interface with Python for problem setup

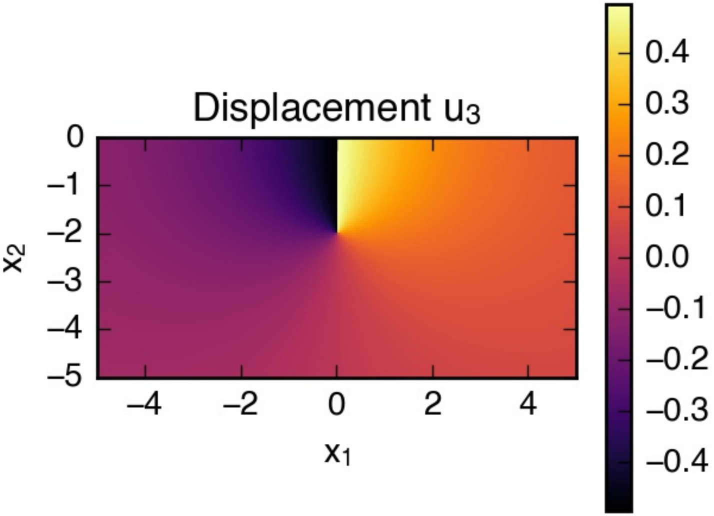
-- No external mesh generator (finite difference)

-- Available on Github

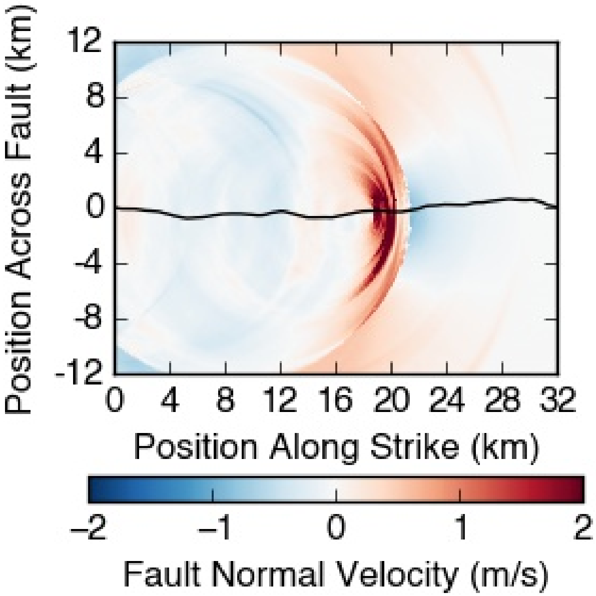
Teaching: Earthquake Source Physics

Graduate course at CERI focusing on quantitative modeling of earthquake slip (analytical and numerical methods)

Continuum Mechanics and Dislocation Models



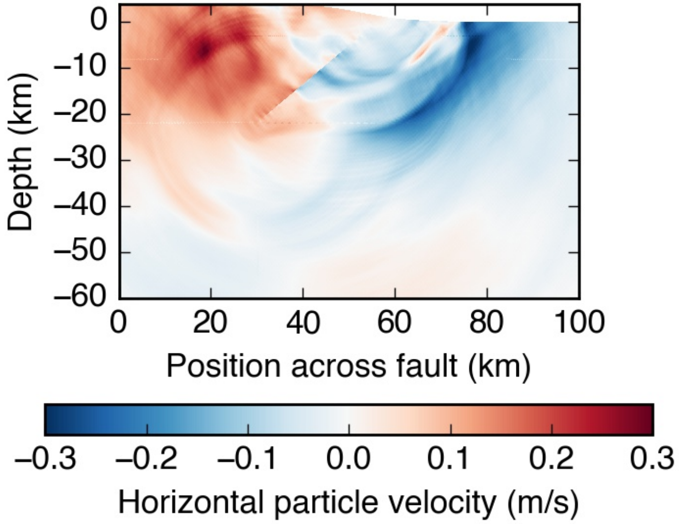
Wave Propagation and Dynamic Rupture

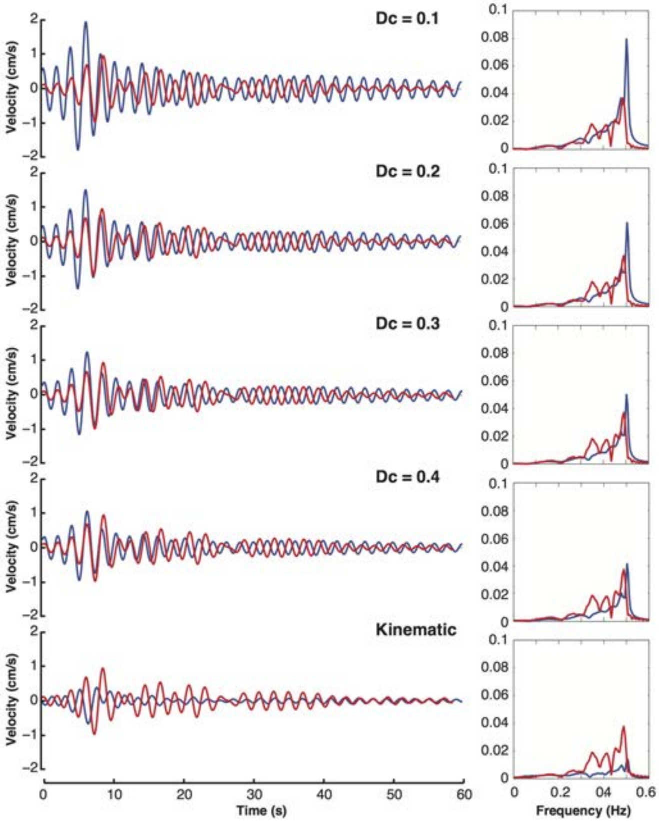


Final Term Project

Spring 2015 (first iteration): Students designed a research project on earthquake rupture. Results were mixed: some excellent, others less so.

Yang Yang -- Lushan earthquake wave propagation modeling

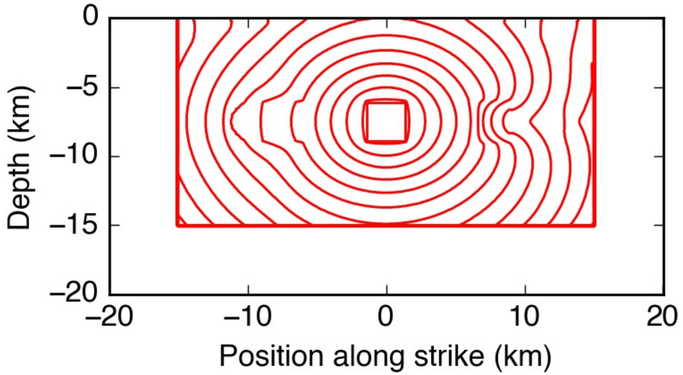




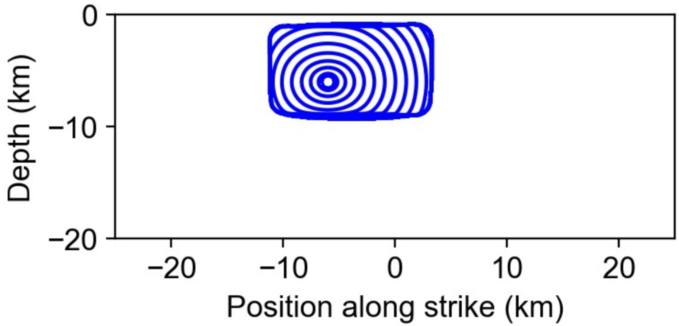
Naeem Khoshnevis -- Dynamic model of Chino Hills Earthquake

Rupture Benchmarks

TPV5

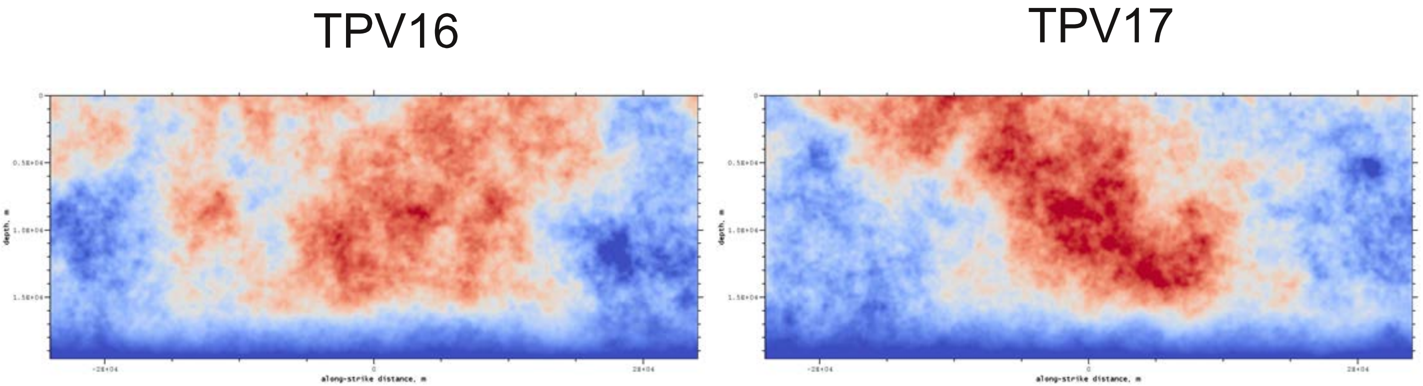


TPV33



Spring 2017: I decided to make term project more structured by using rupture benchmarks as target problems for students. Students would set up and run benchmarks, present results to class, and PhD students were required to do a 30 minute oral exam with the instructor.

Rupture Benchmarks



TPV16

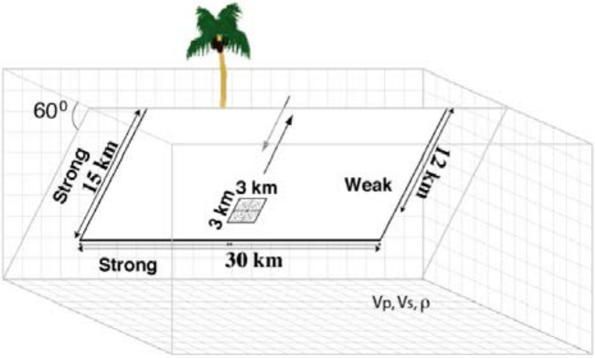
TPV17

Initial heterogeneous shear stresses for TPV16 and TPV17

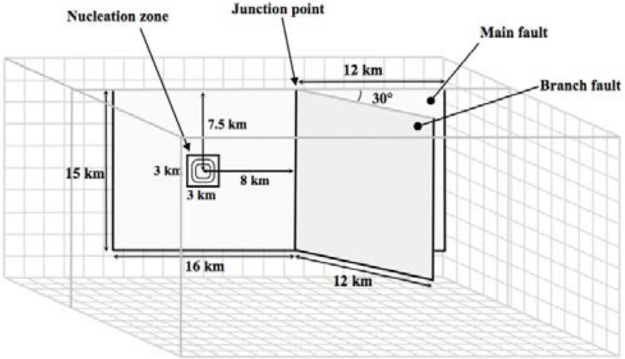
Students were assigned in pairs to related problems (i.e. TPV16 and TPV17). Paired problems could be set up using a single input file to describe both simulations with minor modifications. Encourage collaborative work, but required each student to be able to explain results on their own.

Rupture Benchmarks

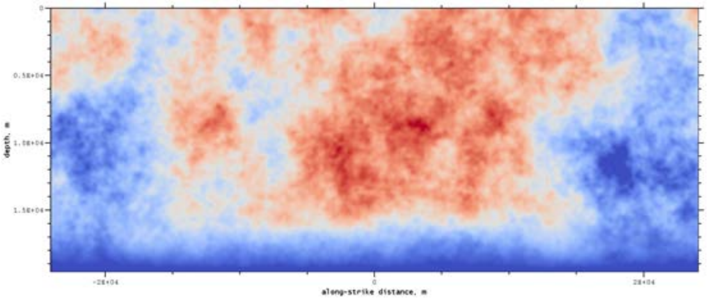
TPV10/11



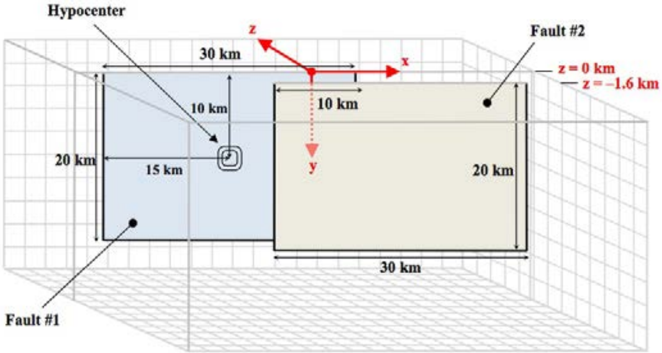
TPV14/15



TPV16/17

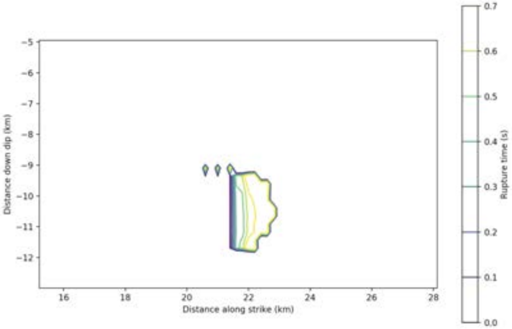


TPV22/23

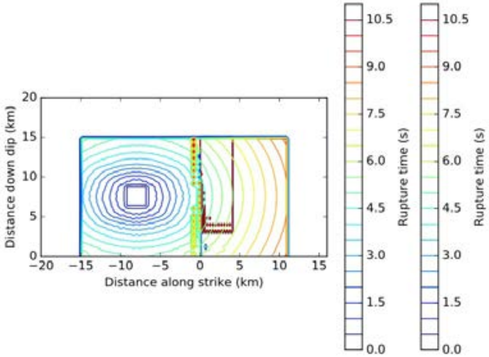


Benchmark Results

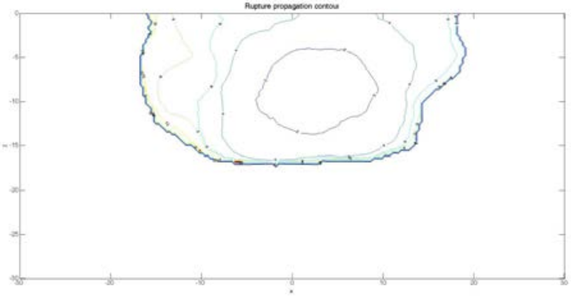
TPV10/11



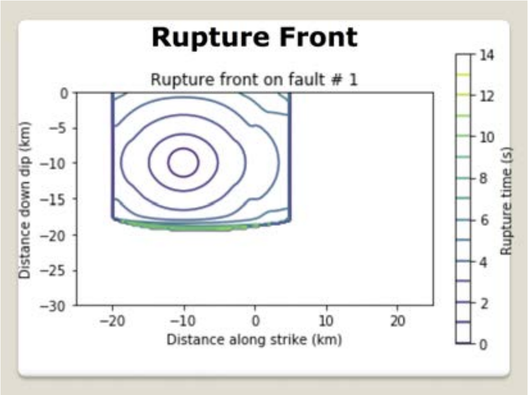
TPV14/15



TPV16/17



TPV22/23



Lessons Learned (Software Side)

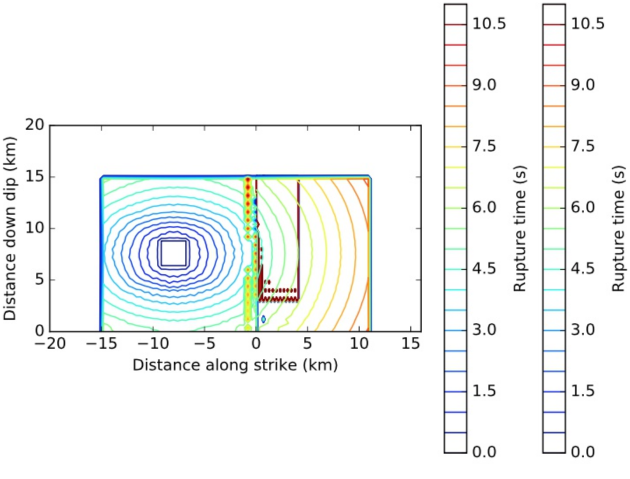
-- Student users are tremendously helpful in finding bugs

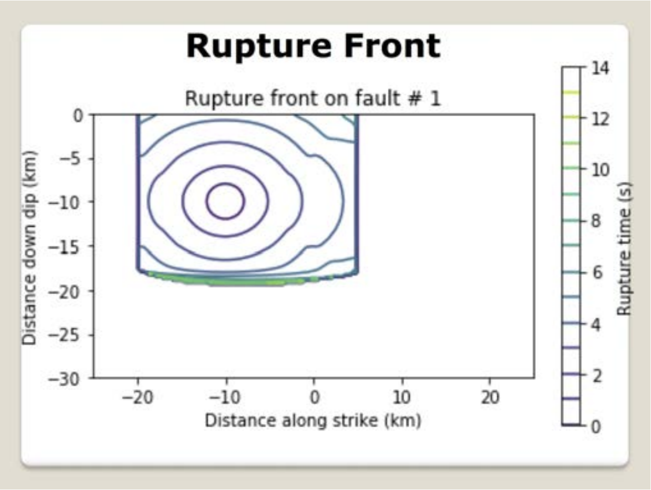
-- Students do not always read the documentation

-- No matter how clear you think the documentation is, not everyone will understand

-- Users will not always use the code as you intended

-- Give many examples!



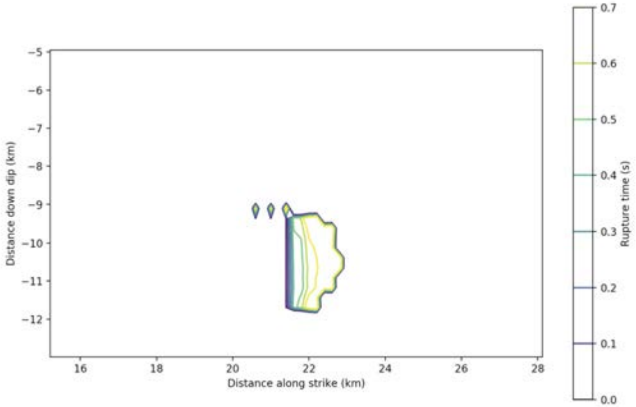


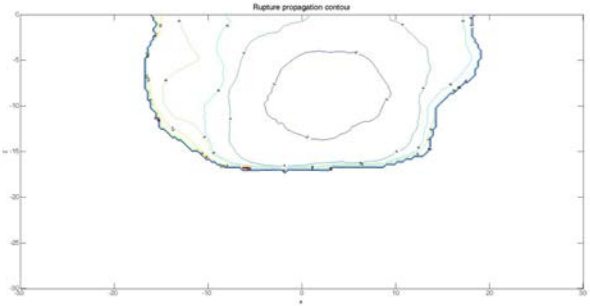
Lessons Learned (Simulation Side)

-- Specifying non-rectangular domains was far and away the biggest challenge. Led to some significant code improvments on my end to simplfy this process

-- Sign and array ordering conventions can be tricky

-- Surprisingly, I thought students handled heterogeneous friction and stress fairly easily with little confusion (many examples)



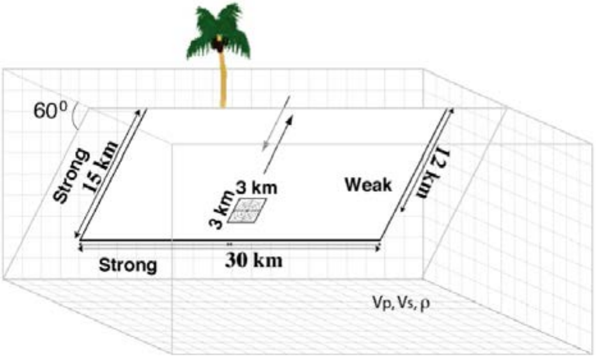


Lessons Learned (Benchmark Side)

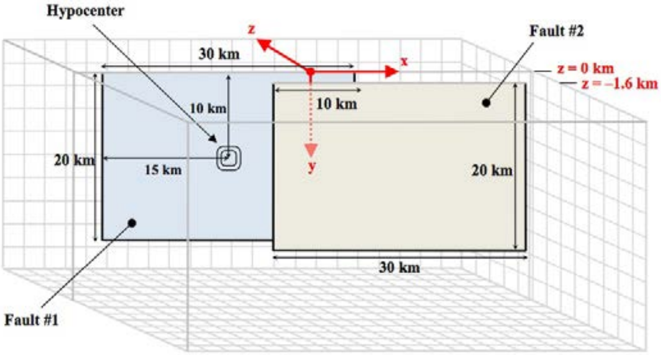
Benchmarks were generally easy for novice modelers to understand.

Questions were most often for clarification on topics that we had only discussed briefly in class, or code-specific details.

Students generally ran the benchmarks at a relatively coarse resolution (~1 hour on a node of UM cluster), but still obtained results useful for the purposes of the course (came up in most recent group paper)



60°
Strong
15 km
3 km
3 km
Weak
12 km
Strong
30 km
Vp, Vs, ρ



Hypocenter
z
x
30 km
Fault #2
z = 0 km
z = −1.6 km
10 km
10 km
20 km
15 km
y
20 km
30 km
Fault #1