simple waveform modeling with SW4

Rob Mellors

USArray short course:

Oklahoma wavefields

August 7 – 11, 2017

LLNL-PRES-694105

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

why?

<u>Generate synthetics to test algorithms and models</u>

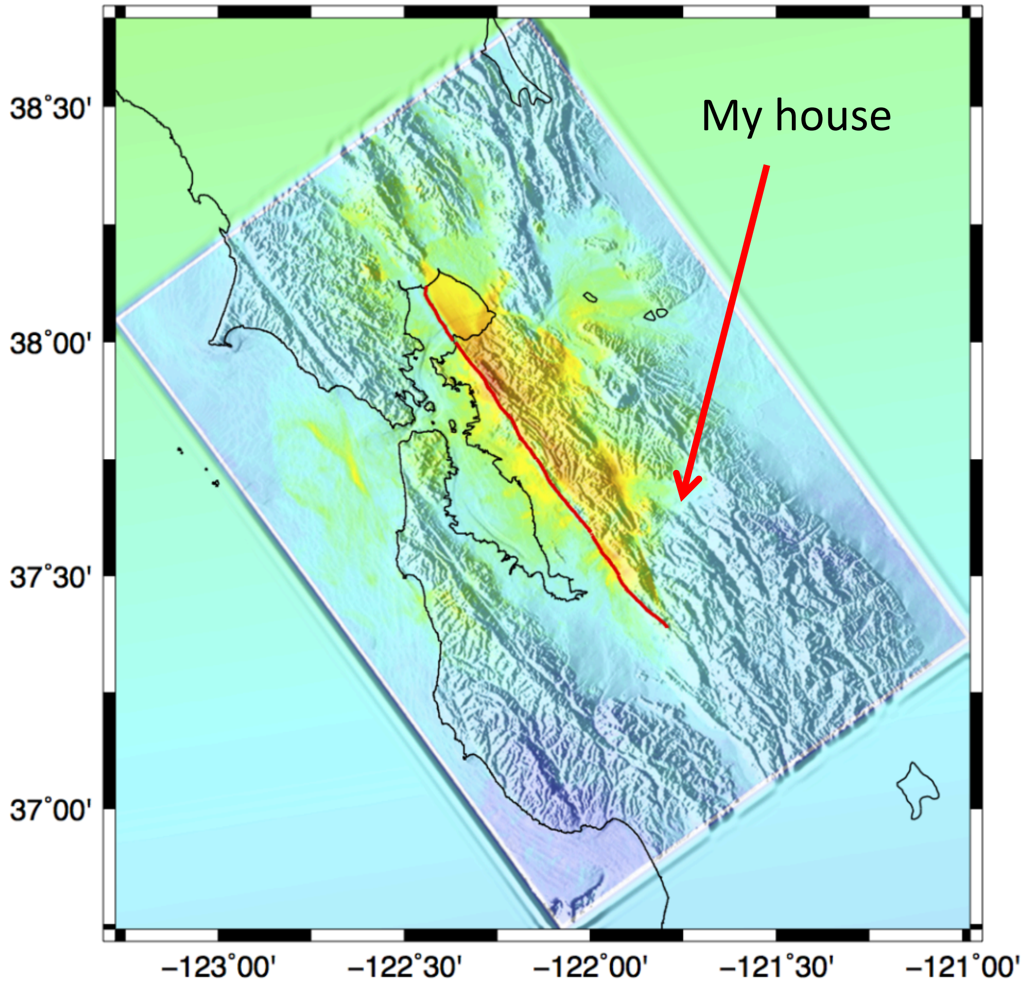
- Test models and compare to data
- Test detection of features

<u>Earthquake hazard</u>

- Estimate ground motion (amplitudes and spectra) for large earthquakes
- Create improved earth models for modeling
- Need 3D capabilities for accurate models

<u>Increasing capabilities</u>

- 3D models used to be at the edge of possibility
- Now, not so much so



An M7 on the Hayward fault

Courtesy A. Rodgers

which code?

- Good documentation
- Well tested, open-source, easy to install, scalable
- Example: CIG 3D full-waveform codes (https://geodynamics.org/cig/software/)
    - SPECFEM
        - Spectral element
        - Acoustic, elastic, poro-elastic
        - Cartesian or global (spherical earth)
        - Meshing required (e.g. Cubit)
    - SW4
        - Finite difference
        - Cartesian only (< ~ 200 km or so)
        - Does not handle internal voids or fluids
        - But no meshing
- Or you can use 2D semi-analytic codes (e.g. reflectivity, fk, etc)

Many exist but we will be testing sw4

SW4

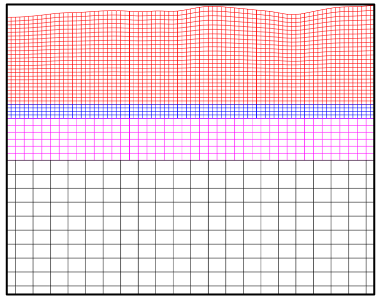
- Designed to run on multiple cores or cluster.
- Requires MPI (even if run on one core).
- Extensive documentation and examples.
- Uses input text files to define source, model, and station locations.
- Handles topography, attenuation, anisotropy, and mesh refinement
- Output is SAC files.
- Can also create images of input model.

components needed for realistic simulations of earthquakes in 3D earth models

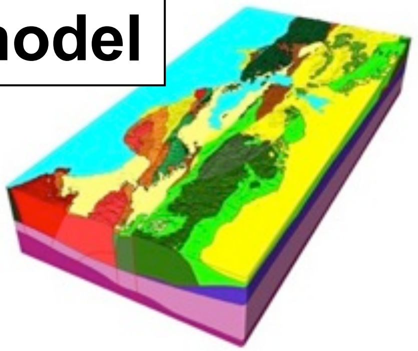
- Computational method (code)
- 3D earth model
- Earthquake source model
- High-performance (parallel) computers
- Validation

SW4

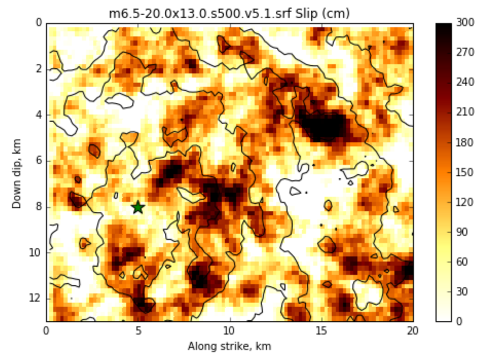
$$\rho\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla\cdot\mathbf{T} + \mathbf{F}(\mathbf{t})$$



3D Earth model

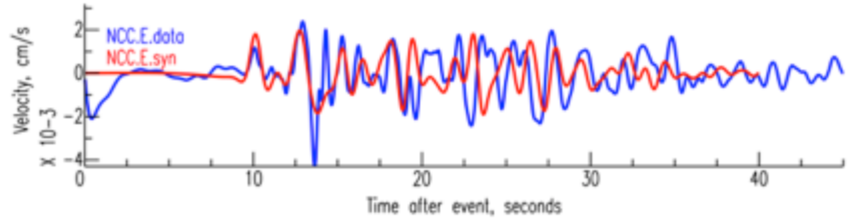


Source model



HPC

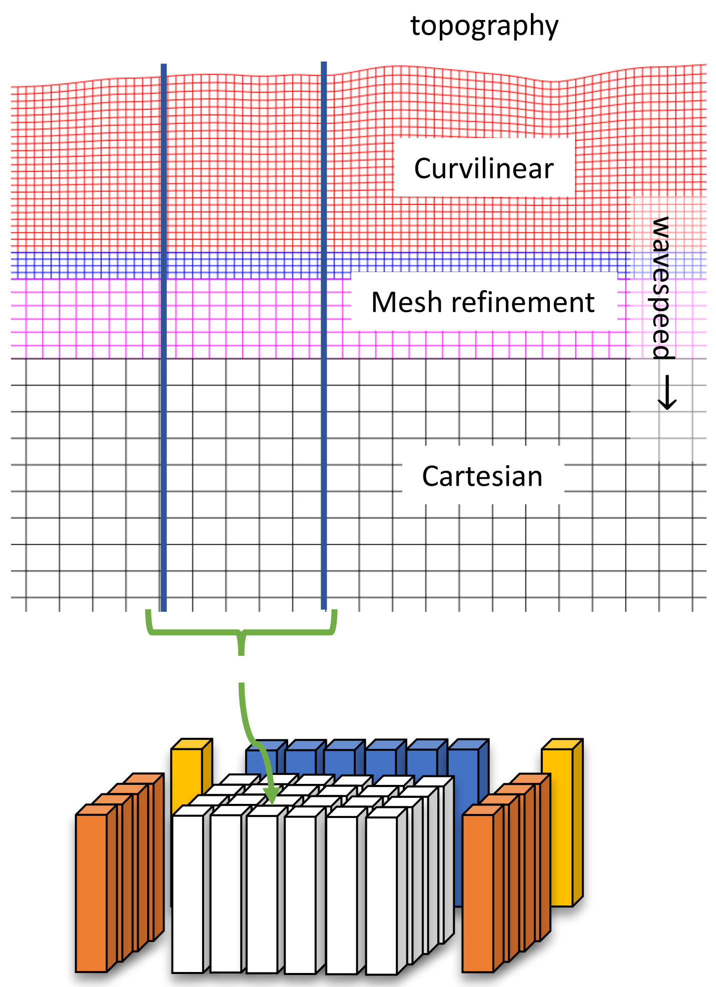
Validate method & model



Courtesy A. Rodgers

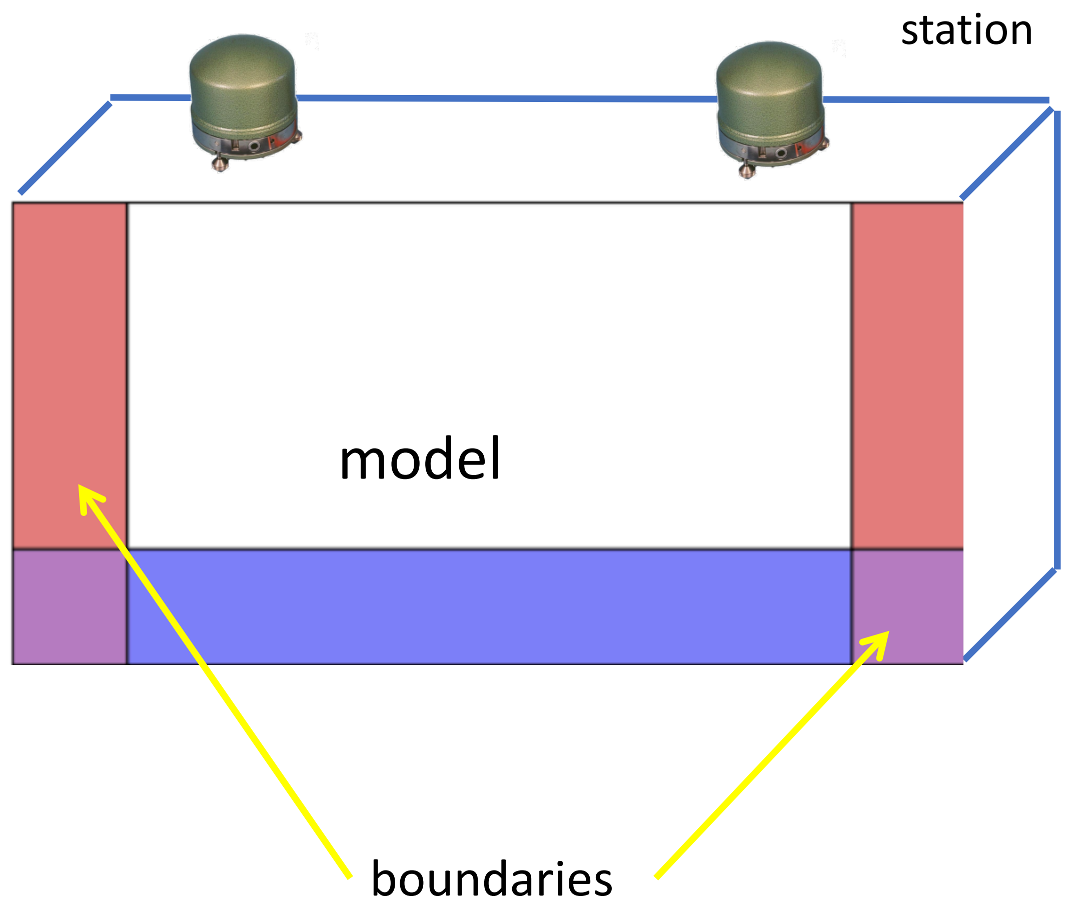
SW4: seismic wave propagation code based on summation by parts FD methods

- SW4 is 4th order accurate
  - Compact FD scheme, explicit time-stepping
  - Automatic mesh generation
- Provably stable for:
  - Heterogeneous materials (iso- and anisotropic)
  - Curvilinear & Cartesian meshes
  - Mesh refinement
- Horizontal MPI-task decomposition
  - Pencil-shaped subdomains
  - Easy load balancing
- Finer meshes and more cores:
  - Pencils get thinner, MPI slows down, I/O slows down, …



Courtesy A. Rodgers

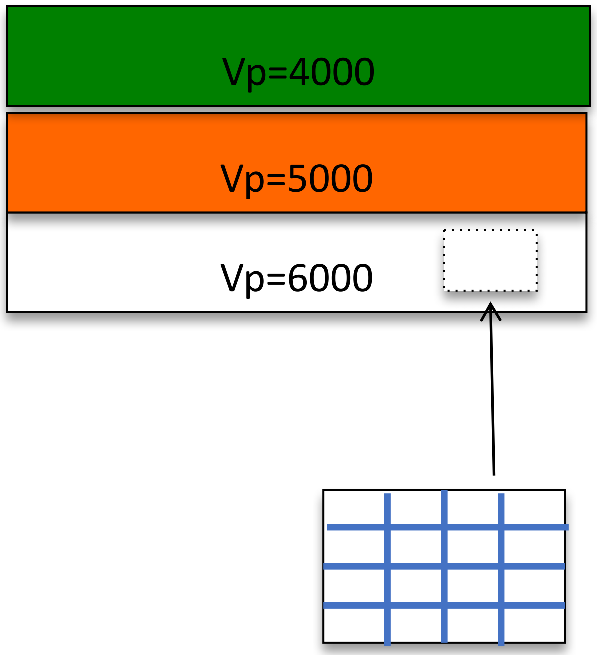
models



Need special layers at edges to avoid reflections from model boundary.
Top edge is the free surface.
Avoid putting source or receivers in boundary layers.
One of the many ways to get weird results.

defining the model

- Grid size
  - defines smallest computational unit
  - frequency dependent on grid size and velocity
- Blocks (rectangles)
  - define Vp, Vs, density, attenuation for each block
- Layers (can have topography)
  - define velocities between layers
- Pfile (set of x,y,z points)
  - define velocities at each point



(units in meters)

```
grid x=2000 y=2000 z=2000 h=25
block vp=4000 vs=2500 rho=2000
block vp=5000 vs=2900 rho=2200 z1=1000
block vp=6000 vs=3500 rho=2400 z1=2000
```

Lots of options!
Very easy to make mistakes
Always check model!

**3d code so models should more complex*

source

- Location of source (x,y,z) or (lat, lon, depth [m])
- Earthquake or moment tensor components
- Type of source (Gaussian, triangle, etc)
- Time and length of source
- Highest frequency [*must match grid size otherwise will cause errors and artifacts*!]

```
source x=1000 y=1000 z=1000 m0=1e13 strike=0 dip=45 rake=90 t0=1 freq=1 type=Gaussian
```

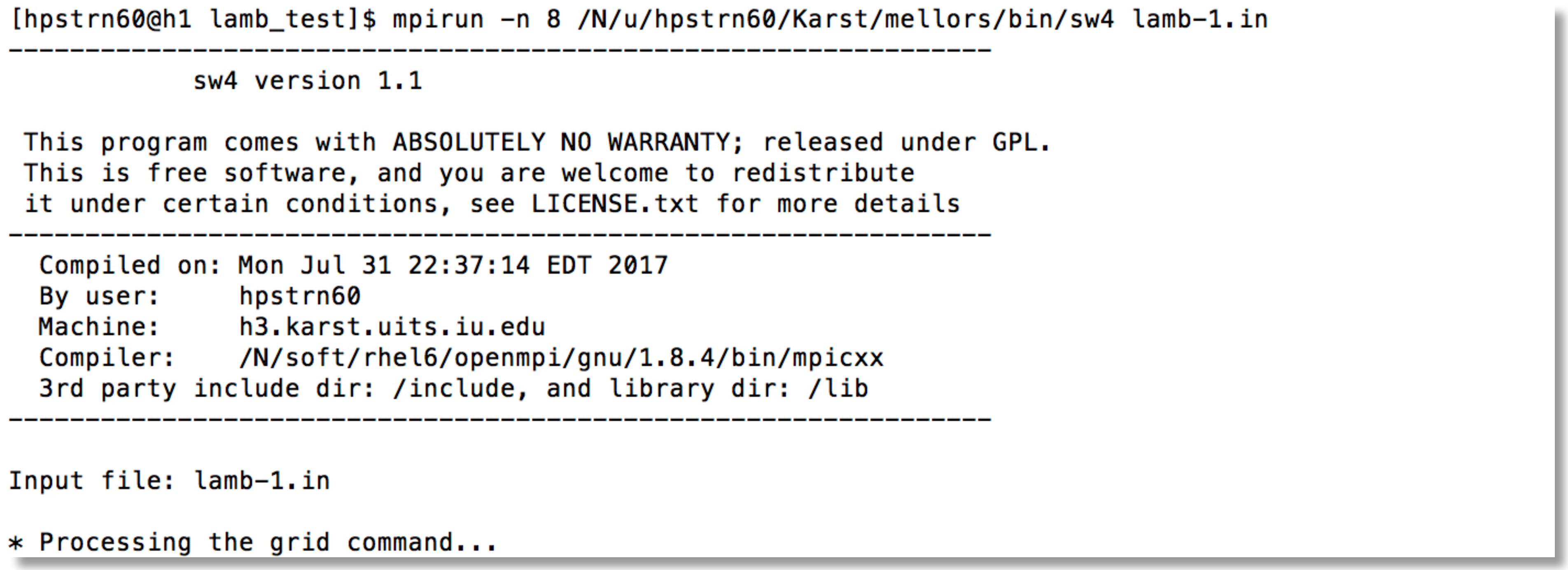
example input (Cartesian)

File named “Hoosier_EQ.in”

```
# this defines location of output files
fileio path=Great_Hoosier_EQ
# grid
grid x=2000 y=2000 z=2000 h=25
# length of simulation
time t=3.0
# source
source x=1000 y=1000 z=1000 m0=1e32 strike=0 dip=45 rake=90 t0=1 freq=1 type=Gaussian
# model
block vp=4000 vs=2500 rho=2000
block vp=5000 vs=2900 rho=2200 z1=500
block vp=6000 vs=3500 rho=2400 z1=1000
# 4 stations
rec  x=1000   y=400 z=0 file=IU1 sacformat=1 nsew=1
rec  x=1000   y=800 z=0 file=kIU2 sacformat=1 nsew=1
rec  x=1000  y=1200 z=0 file=IU3 sacformat=1 nsew=1
rec  x=1000  y=1600 z=0 file=IU4 sacformat=1 nsew=1
```

running sw4 (with mpi)

mpirun –n *8* sw4 *Hoosier_EQ.in*



```
[hpstrn60@h1 lamb_test]$ mpirun -n 8 /N/u/hpstrn60/Karst/mellors/bin/sw4 lamb-1.in
----------------------------------------------------------------
            sw4 version 1.1

 This program comes with ABSOLUTELY NO WARRANTY; released under GPL.
 This is free software, and you are welcome to redistribute
 it under certain conditions, see LICENSE.txt for more details
----------------------------------------------------------------
  Compiled on: Mon Jul 31 22:37:14 EDT 2017
  By user:     hpstrn60
  Machine:     h3.karst.uits.iu.edu
  Compiler:    /N/soft/rhel6/openmpi/gnu/1.8.4/bin/mpicxx
  3rd party include dir: /include, and library dir: /lib
----------------------------------------------------------------

Input file: lamb-1.in

* Processing the grid command...
```

Could do: sw4 lamb-1.in   but would run on one CPU, I think
Batch jobs would use something like moab

and some other output, sometimes useful

```
Input file: lamb-1.in

* Processing the grid command...
* Setting nx to 301 to be consistent with h=4.00000000e-02
* Setting ny to 301 to be consistent with h=4.00000000e-02
* Setting nz to 151 to be consistent with h=4.00000000e-02

*** No topography command found in input file. Using z=0 as free surface boundary ***


Global grid sizes (without ghost points)
Grid         h         Nx        Ny        Nz       Points
   0      0.04        301       301       151     13680751
Total number of grid points (without ghost points): 1.36808e+07

Default Supergrid thickness has been tuned; thickness = 50 grid sizes

   Execution time, reading input file 5.14340401e-02 seconds
----------------------------------------------------
 Making Output Directory: lamb-1/

... Done!
----------------------------------------------------
Geographic and Cartesian coordinates of the corners of the computational grid:
0: Lon= 1.180000e+02, Lat=3.700000e+01, x=0.000000e+00, y=0.000000e+00
1: Lon= 1.180000e+02, Lat=3.700011e+01, x=1.200000e+01, y=0.000000e+00
2: Lon= 1.180001e+02, Lat=3.700011e+01, x=1.200000e+01, y=1.200000e+01
3: Lon= 1.180001e+02, Lat=3.700000e+01, x=0.000000e+00, y=1.200000e+01


       ----------- Material properties ranges ---------------
       1.00000000e+00 kg/m^3 <=  Density <= 1.00000000e+00 kg/m^3
       1.73205081e+00 m/s    <=  Vp      <= 1.73205081e+00 m/s
       1.00000000e+00 m/s    <=  Vs      <= 1.00000000e+00 m/s
       1.73205081e+00        <=  Vp/Vs   <= 1.73205081e+00
       1.00000000e+00 Pa     <=  mu      <= 1.00000000e+00 Pa
       1.00000000e+00 Pa     <=  lambda  <= 1.00000000e+00 Pa
       ------------------------------------------------------
```

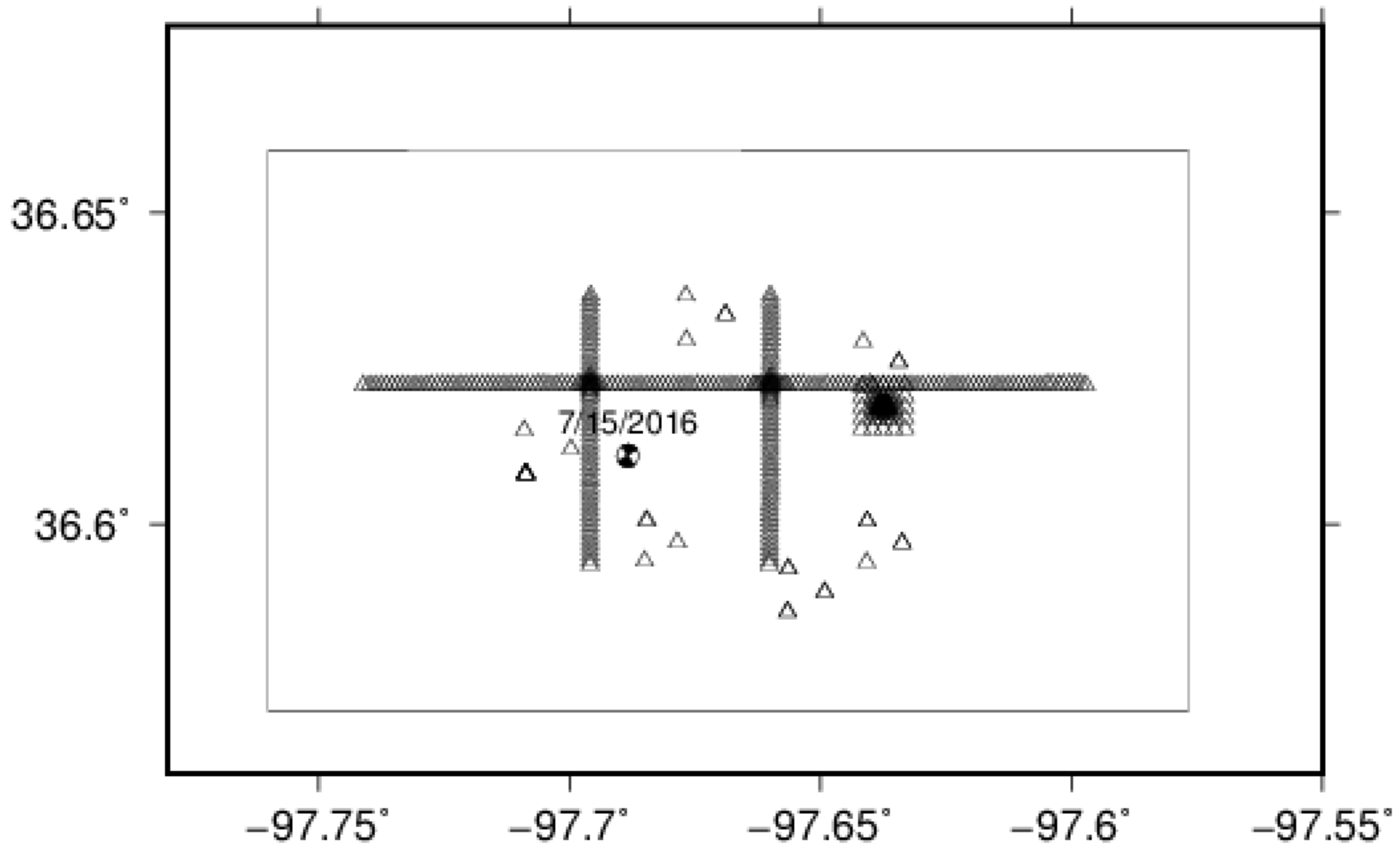
```
***** PPW = minVs/h/maxFrequency ********
g=0, h=4.000000e-02, minVs/h=25 (Cartesian)

   Execution time, start up phase 2.54199982e-01 seconds
Running sw4 on 8 processors...
Writing output to directory: lamb-1/
============================================================
 Running program on 8 MPI tasks using the following data:

 Start Time = 0 Goal Time = 15
 Number of time steps = 645 dt: 0.0232558
-----------------------------------------------------------
Lamb's problem testing
Parameters:
  Cp = 1.73205081e+00
  Cs = 1.00000000e+00
  Rho = 1.00000000e+00
  (xs, ys, zs) = 6.00000000e+00, 6.00000000e+00, 0.00000000e+00
  (fx, fy, fz) = 0.00000000e+00, 0.00000000e+00, 1.00000000e+00
  Source time fcn = C6SmoothBump
-----------------------------------------------------------
  Begin time stepping...
Time step       1  t =   2.3255814e-02
Time step     101  t =   2.3488372e+00
```

Output as sac files in directory

Oklahoma!



Input file (partial)

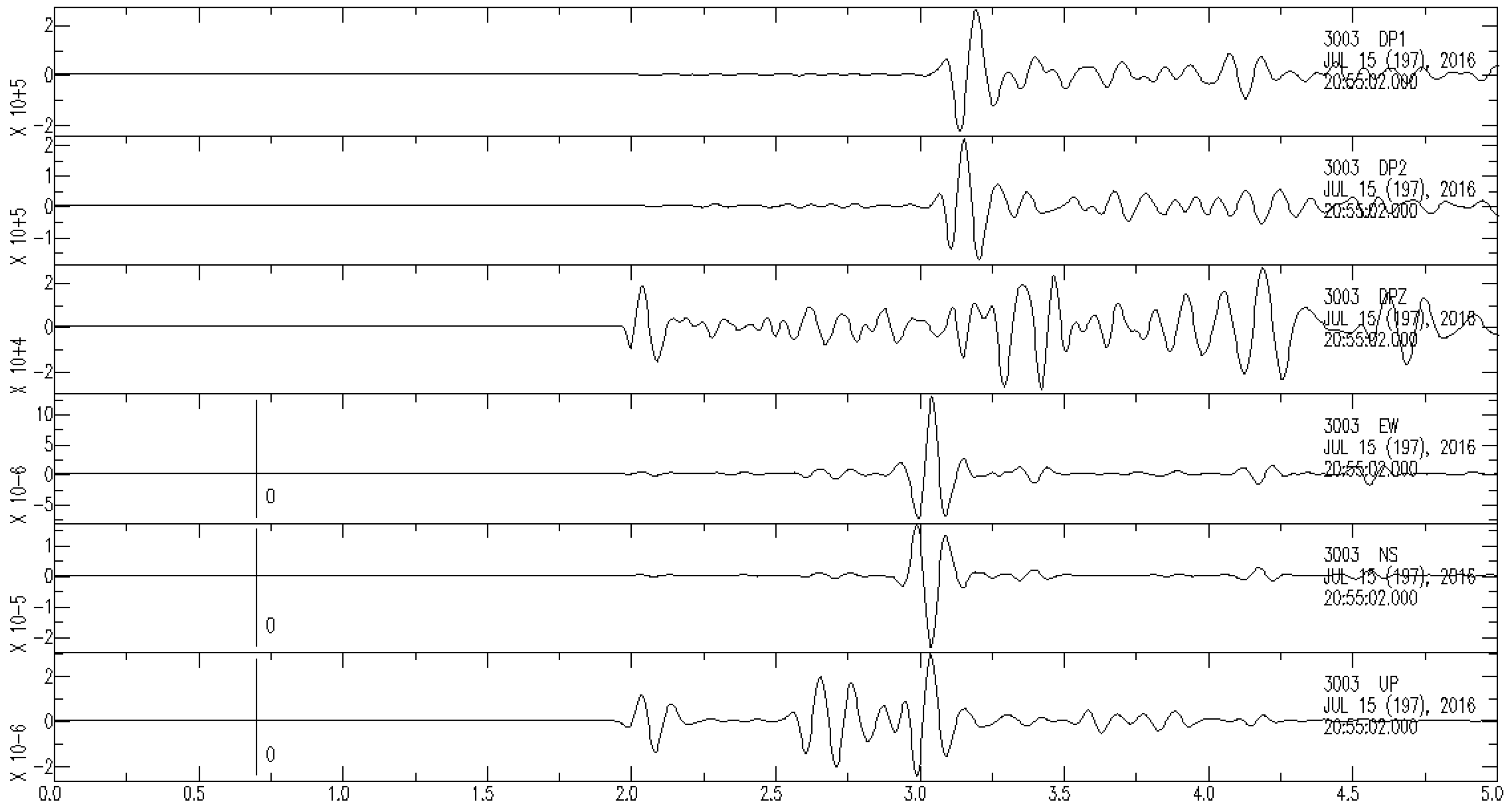
```
fileio path=OK_15July16_h100_freq_1
grid nx=100 ny=165 nz=100 lat=36.57 lon=-97.76 h=100 az=0
time t=5.0 utcstart=7/15/2016:20:55:02.000
source depth=5028.0 lat=36.6110 lon=-97.6883 t0=0.7 freq=1.0 strike=147 dip=84 rake=180 m0=4.0e12 type=Ricker
block vp=2700 vs=1560 rho=2000
block vp=2950 vs=1710 rho=2200 z1=300
block vp=4150 vs=2400 rho=2200 z1=1000
block vp=5800 vs=3350 rho=2400 z1=1500
block vp=6270 vs=3620 rho=2400 z1=8000
#block vp=6410 vs=3710 rho=2800 z1=21000
#block vp=7900 vs=4570 rho=2800 z1=42000
#block vp=8150 vs=4710 rho=2800 z1=50000
#block vp=8500 vs=4910 rho=2800 z1=80000
# Wavefields receivers
rec lat=36.6223 lon=-97.7410 depth=0 file=1001 nsew=1
rec lat=36.6223 lon=-97.7398 depth=0 file=1002 nsew=1
rec lat=36.6223 lon=-97.7387 depth=0 file=1003 nsew=1
```

Full input file with all sensors and comments at:

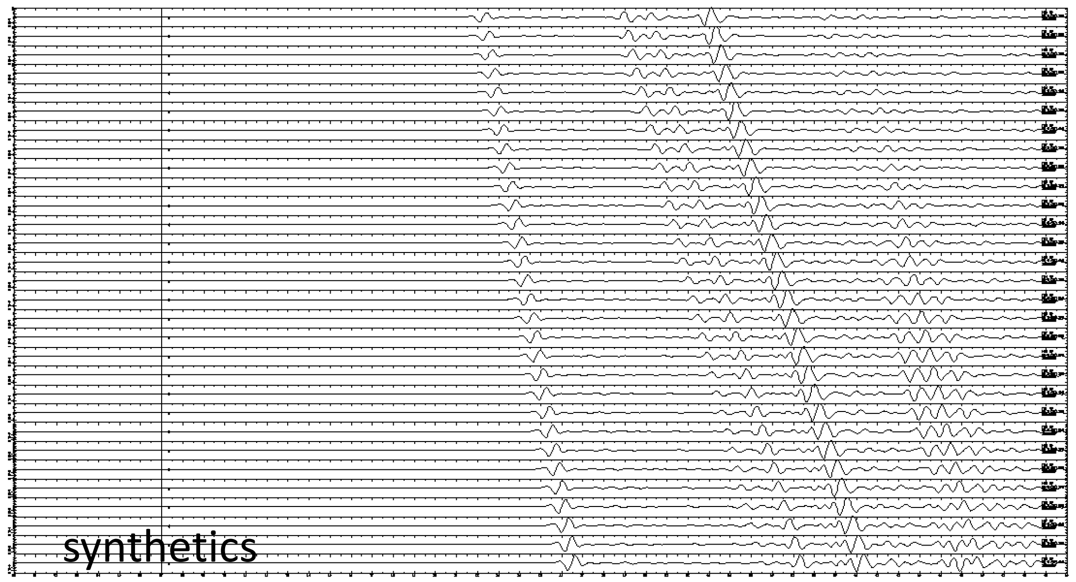
/N/u/hpstrn60/Karst/mellors/OK_synthetics/OK_wf_15July2016.in

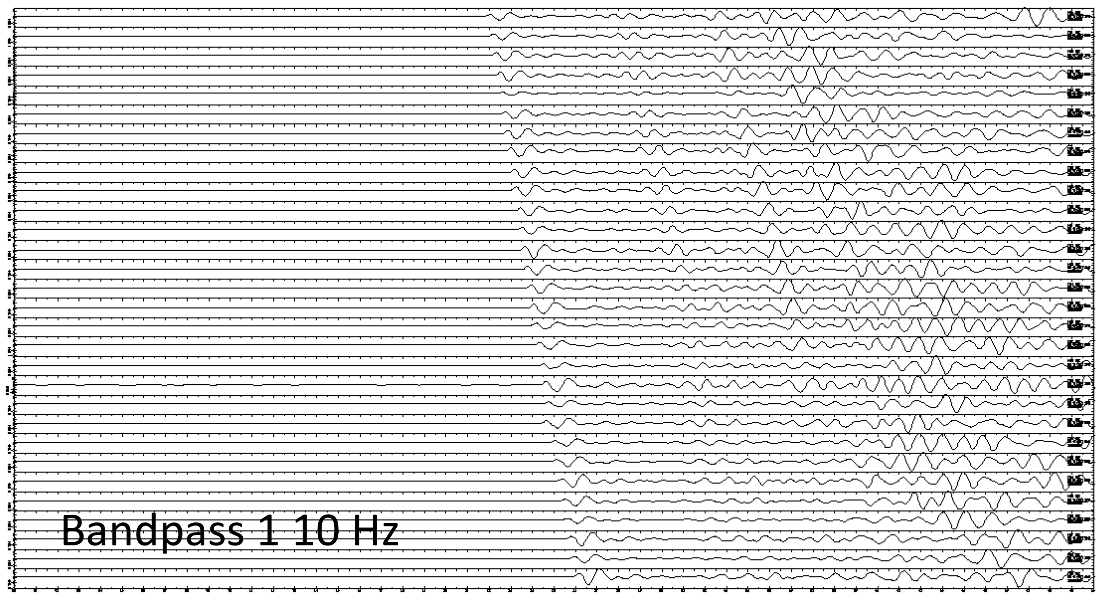
Results for 3003 (bandpassed 1-10 Hz)

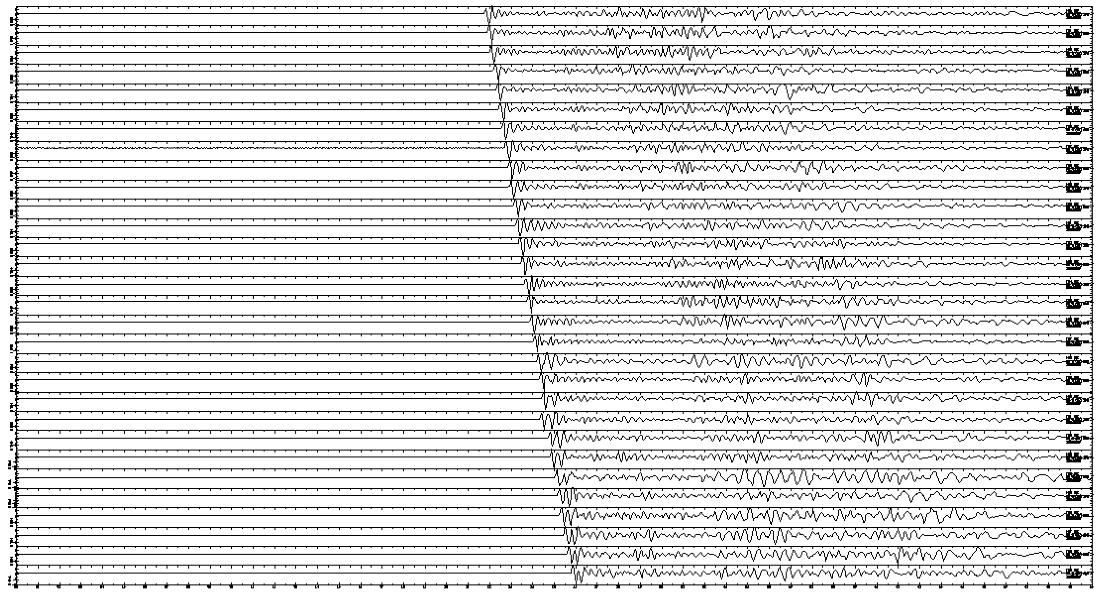
(sorry, this was run at 10 hz)



More synthetics and data







assignments

1) Download the sw4 manual from the CIG site (https://geodynamics.org/cig/software/sw4/)
2) Look at it and read section 4.5: **How to choose the grid size**
3) Run the example input that has the source frequency set at 1 Hz and the grid size (h) at 100
4) Change the frequency to 10 Hz and run. What happened?
    1) What do you need to change to obtain better results? (several options)
    2) Try it
    3) How high a frequency can you go easily on the IU machines…..

5) Other stuff to try:
    1) Output cross-sections of the model
    2) Change the velocity to add a shallow low-velocity layer
    3) Try a high-resolution synthetic of the gradiometer array only
    4) Does anisotropy make a difference? (how much would you expect for layered sediments?)
    5) Generate a Green’s function for comparison with ambient noise cross-correlations

Caveats

- It is very easy to make mistakes and maybe not catch them.
- Mixing up coordinate systems.
- The model not being what you think it is (especially for 3D)

    use the image command to write out the model

    write out images of the wavefield as well
- SW4 not understanding commands and not writing out error messages
- Sending out misleading errors messages
- Having sensors or receivers in the boundary layer
- Incorrect grid spacing or filters
    - Too small a grid => lots of wasted computation
    - Too large => poor frequency recovery
- Source time function too close to start

- <u>Check, double-check and validate as much as possible</u>

For help:
Read manual
Watch youtube video (yes, there is one)
Read wiki, or doxygen manual
(you can try looking at the code also)