

SW4 for Chandan

Kindly do it

Sean R. Ford

2025-04-02

1 Introduction

Chandan asked, “Can you run MZZ, MXX, MYY seismograms for a layered medium in your SW4 and send me the output with the model?”

We obtained [SW4 v3.0](#) from [GitHub](#) and compiled on macos with [MPICH](#) using the following `make.inc` file:

```
proj = yes
# homebrew version of proj is fine
SW4ROOT = /opt/homebrew
fftw = yes
# homebrew version uses openmpi so installed fftw for mpich
FFTWHOME = /Users/ford17/Applications/fftw-3.3.10
FC = mpif90
CXX = mpicxx
# need Accelerate for blas on macos
EXTRA_LINK_FLAGS = -framework Accelerate -L/opt/homebrew/lib/gcc/current -lgfortran
```

2 Method

The requirements are:

- Station at $(x,y,z) = (60, 0, 0)$
- Source at $(x,y,z) = (10, 0, 10)$
- WUS model
- PPW for 5 Hz
- Dirac (delta) source

So we create an SW4 input files for each source. Below is the input file for `mxw.sw4in`:

```
fileio pfs=1 nwriters=16 path=mxw.dir printcycle=1000
grid x=70e3 y=20e3 z=50e3 h=50
time t=45
block vp=7900.0 vs=4620.0 rho=3276.0 qp=60976 qs=27027
block vp=6407.5 vs=3768.0 rho=2822.3 qp=901 qs=402 z2=40000
block vp=6270.8 vs=3739.6 rho=2781.2 qp=658 qs=293 z2=21000
block vp=5544.5 vs=3295.3 rho=2608.9 qp=287 qs=128 z2=8000
block vp=3406.5 vs=2008.9 rho=2215.0 qp=331 qs=147 z2=1900
source x=10e3 y=10e3 z=10e3 mxw=1e18 type=Dirac t0=0
#source x=10e3 y=10e3 z=10e3 mxw=1e18 type=Triangle t0=0 freq=1
#source x=10e3 y=10e3 z=10e3 mxw=1e18 type=C6SmoothBump t0=0 freq=1
rec x=60e3 y=10e3 z=0 file=rec50 sacformat=1
image mode=s y=10e3 cycle=1
```

Note that we could have used refinement at the lowest layer for a more efficient calculation.

And to calculate the results we could use the command:

```
> mpirun -np 16 sw4 mxw.sw4in
```

In practice we used many more processors available on ruby from [LC](#) with a slurm script, `runruby.sw4run`:

```
#!/bin/tcsh
#SBATCH -N 50
#SBATCH -J run
#SBATCH -t 500
#SBATCH -p pbatch
#SBATCH --license=lustre1
#SBATCH -A gmp
#SBATCH -o run.sw4out
#SBATCH -e run.sw4err

# Max: 1440 minutes (24 hours) on 520 nodes
# 50 nodes took 15 minutes per run (9 runs = 135 minutes)

# Set CPUS/nodes for RUBY (limit 520 nodes for 24 hours)
@ CPUSPERNODE = 56
```

```
# Compute number of CPUs
@ NCPUS = ( $SLURM_JOB_NUM_NODES * $CPUSPERNODE )

srun -n$NCPUS /usr/workspace/ford17/sw4/optimize_ruby_mp/sw4 mxx.sw4in
srun -n$NCPUS /usr/workspace/ford17/sw4/optimize_ruby_mp/sw4 myy.sw4in
srun -n$NCPUS /usr/workspace/ford17/sw4/optimize_ruby_mp/sw4 mzz.sw4in
srun -n$NCPUS /usr/workspace/ford17/sw4/optimize_ruby_mp/sw4 mxy.sw4in
srun -n$NCPUS /usr/workspace/ford17/sw4/optimize_ruby_mp/sw4 mxz.sw4in
srun -n$NCPUS /usr/workspace/ford17/sw4/optimize_ruby_mp/sw4 myz.sw4in
srun -n$NCPUS /usr/workspace/ford17/sw4/optimize_ruby_mp/sw4 fx.sw4in
srun -n$NCPUS /usr/workspace/ford17/sw4/optimize_ruby_mp/sw4 fy.sw4in
srun -n$NCPUS /usr/workspace/ford17/sw4/optimize_ruby_mp/sw4 fz.sw4in
```

And then we execute the script with:

```
> sbatch runruby.sw4run
```

Our goal for an accurate calculation is to have a minimum points per wavelength (PPW) of between 6 and 10. The PPW is related to the smallest wave velocity v [m/s] divided by the grid spacing h [m] divided by the maximum frequency represented f [1/s]. For the grid used here ($v = 2008.9$ m/s, $h = 50$ m) the points per wavelength for a 5 Hz maximum frequency is:

$$\text{PPW} = v/h/f = 2008.9/50/5 \approx 8$$

3 Results

The shear wave velocity model is shown in Figure 1.

Calculated velocities bandpassed between 20 s and 5 Hz are shown in Figure 2.

4 Comparison with CPS

Displacements are also calculated using CPS and shown in Figure 3.

[Bob Herrmann](#) produced an [excellent tutorial](#) that guided our work here. In that tutorial, Bob found “excellent agreement” between SW4 and CPS.

We use the script `D0IT.sh` to run the CPS codes:

DB: image.cycle=0001.y=0.s.sw4img

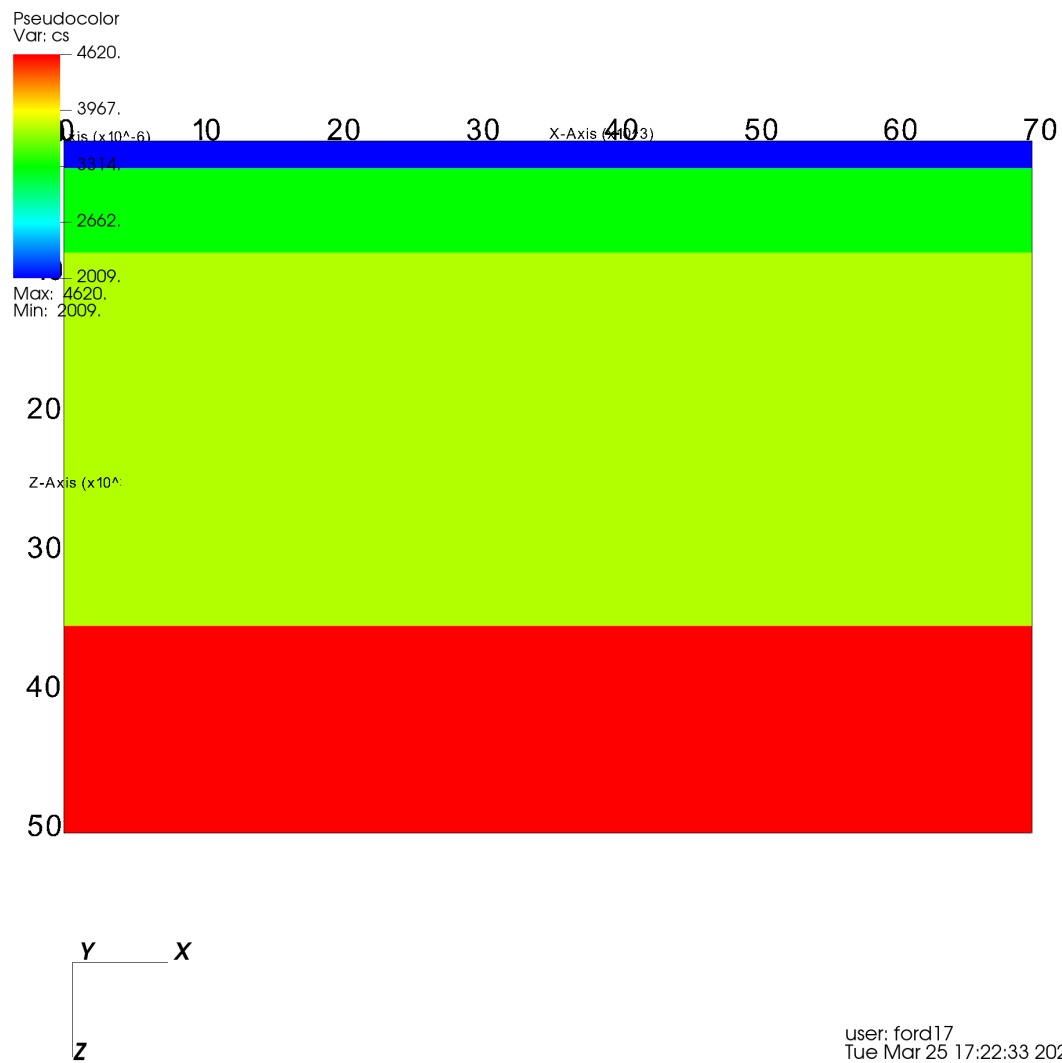


Figure 1: Shear wave velocity model plotted with [VisIt](#)

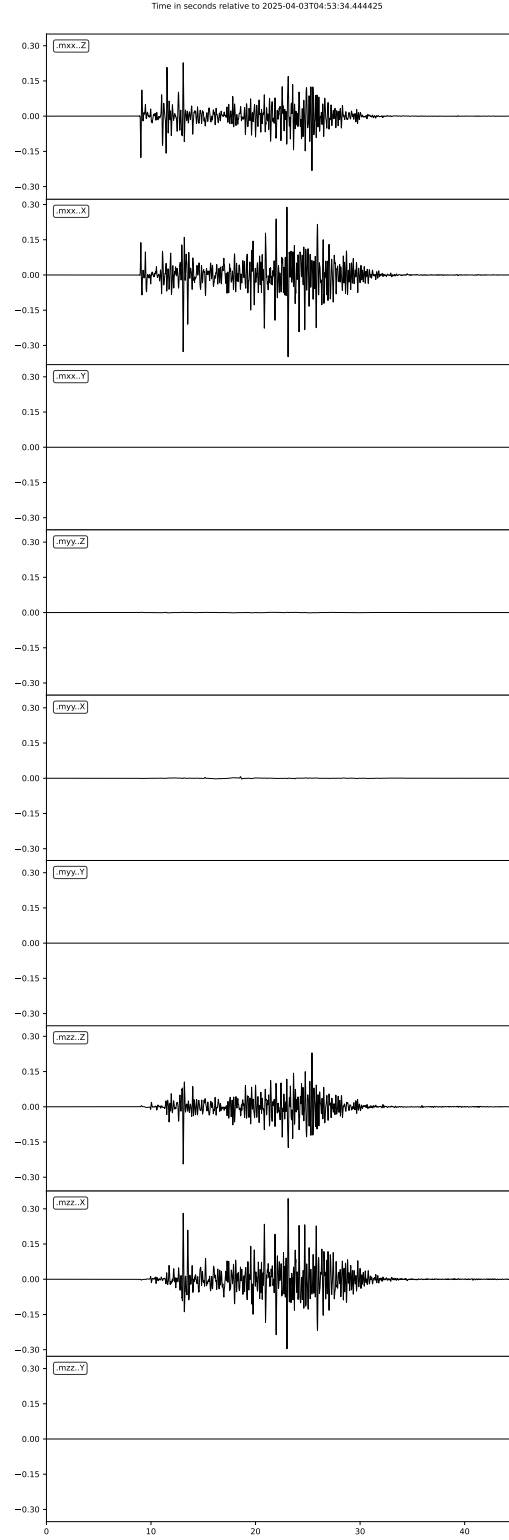


Figure 2: Calculated displacements low passed at 5 Hz plotted with [ObsPy](#).

```

#!/usr/bin/env bash

## Any layered model
cat > WUSnoQ.mod << FIN
MODEL.01
Model after      8 iterations
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
  H(KM) VP(KM/S) VS(KM/S) RHO(GM/CC) QP QS ETAP ETAS FREFP FREFS
  1.9000  3.4065  2.0089  2.2150  0  0 0.00 0.00  1.00  1.00
  6.1000  5.5445  3.2953  2.6089  0  0 0.00 0.00  1.00  1.00
 13.0000  6.2708  3.7396  2.7812  0  0 0.00 0.00  1.00  1.00
 19.0000  6.4075  3.7680  2.8223  0  0 0.00 0.00  1.00  1.00
  0.0000  7.9000  4.6200  3.2760  0  0 0.00 0.00  1.00  1.00
FIN

## Recording at 50 km with Nyquist above 5 Hz
# DIST DT NPTS TO VRED
echo "50 0.01 4096 0 0" > dfile

## Run
rm -fr hspec96.*
hprep96 -M WUSnoQ.mod -d dfile -HS 10
hspec96

## Output
# Triangle is 2 * L * dt = 2 * 25 * 0.02 = 1
# hpulse96 -V -t -l 25 | f96tosac -G
# Impulse
hpulse96 -V -i | f96tosac -G

## Put in mechanism
# 1e18 nt-m = 1e25 dyne-cm
MOM=1.0e+25
ZSSFILE=`ls *.ZSS | head -1`
PROTO=`basename $ZSSFILE .ZSS`

```

```

for MIJ in MXX MXY MXZ MYY MYZ MZZ
do
gsac << EOF
mt to ZNE AZ 0 BAZ 180 ${MIJ} ${MOM} FILE ${PROTO}
w
mv T.Z ${MIJ}_${PROTO}_Z
mv T.N ${MIJ}_${PROTO}_N
mv T.E ${MIJ}_${PROTO}_E
q
EOF
done
FORCE=1.0e+20
for F in FN FE FD
do
gsac << EOF
mt to ZNE AZ 0 BAZ 180 ${F} ${FORCE} FILE ${PROTO}
w
mv T.Z ${F}_${PROTO}_Z
mv T.N ${F}_${PROTO}_N
mv T.E ${F}_${PROTO}_E
q
EOF
done

```

We also calculated the Green's functions using the **strains** calculation from SW4. This is obtained via reciprocity where strains at the source location are calculated due to a unit force at the surface. Such an input file for the vertical component can be calculated using the following input file:

```

fileio pfs=1 nwriters=16 path=fz.dir printcycle=1000
grid x=70e3 y=20e3 z=50e3 h=50
time t=45
block vp=7900.0 vs=4620.0 rho=3276.0 qp=60976 qs=27027
block vp=6407.5 vs=3768.0 rho=2822.3 qp=901 qs=402 z2=40000
block vp=6270.8 vs=3739.6 rho=2781.2 qp=658 qs=293 z2=21000
block vp=5544.5 vs=3295.3 rho=2608.9 qp=287 qs=128 z2=8000
block vp=3406.5 vs=2008.9 rho=2215.0 qp=331 qs=147 z2=1900
source x=60e3 y=10e3 z=0 fz=1e15 type=Dirac t0=0
#source x=60e3 y=10e3 z=0 fz=1e15 type=Triangle t0=0 freq=1
#source x=60e3 y=10e3 z=0 fz=1e15 type=C6SmoothBump t0=0 freq=1
rec x=10e3 y=10e3 z=10e3 file=rec50 sacformat=1 variables=strains
image mode=s y=10e3 cycle=1

```

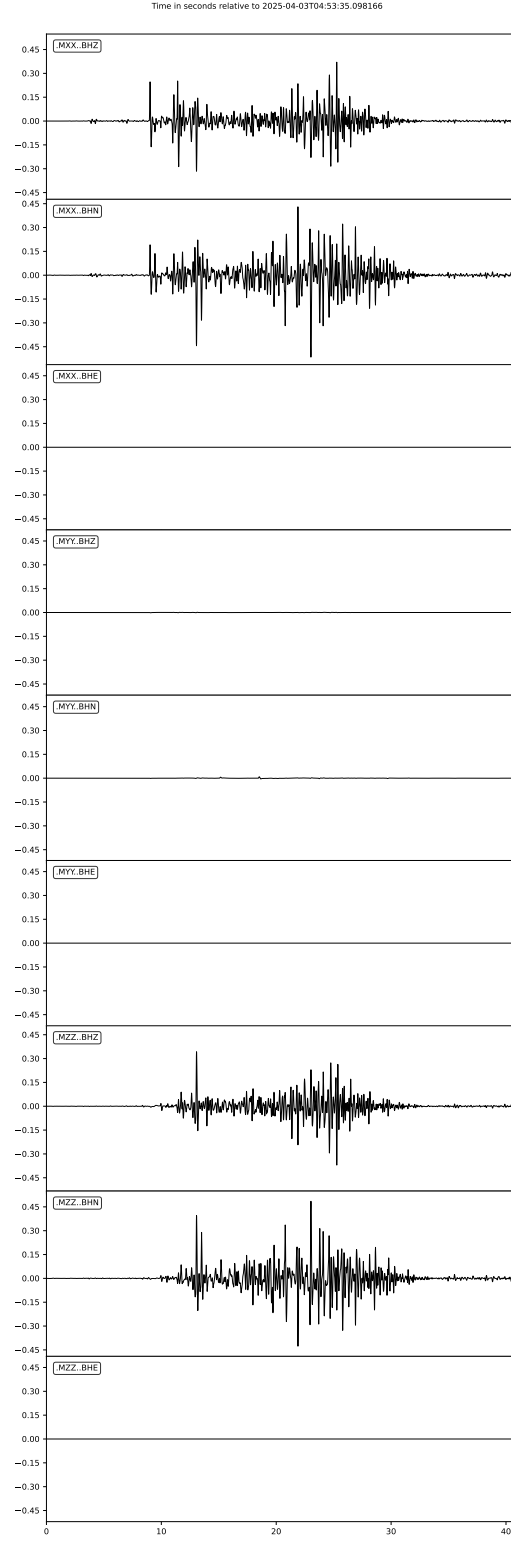


Figure 3: Calculated displacements using CPS low passed at 5 Hz plotted with [ObsPy](#).

5 Conclusions

SW4 and CPS are excellent tools for wavefield calculations.