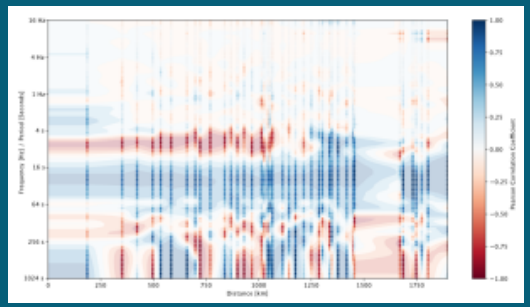
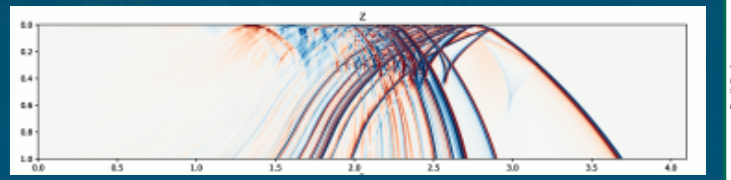
This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

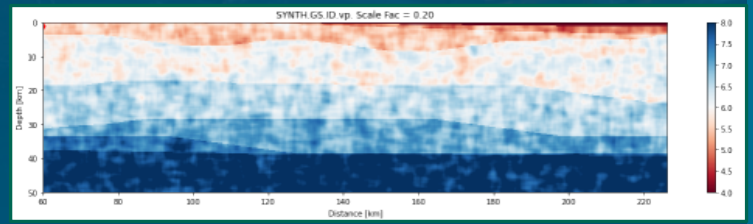
SAND2021-7902C

Sandia National Laboratories

Local to regional scale simulation of small events with 2D finite differences.







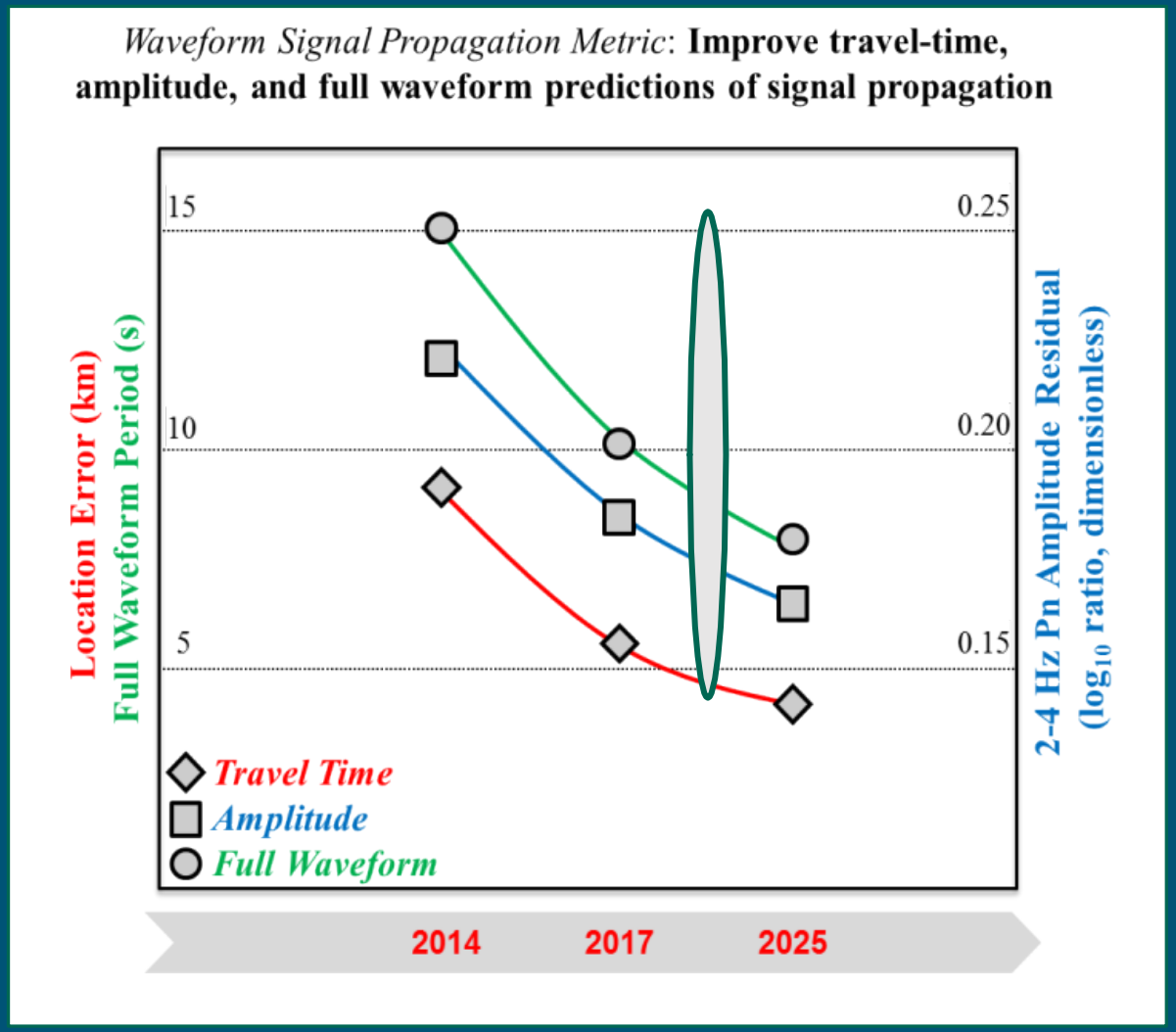
Rob Porritt

Lisa Linville, Andrea Conley, Thomas Catanach, Rigobert Tibi, John Merchant, Nathan Downey, and Chris Young.

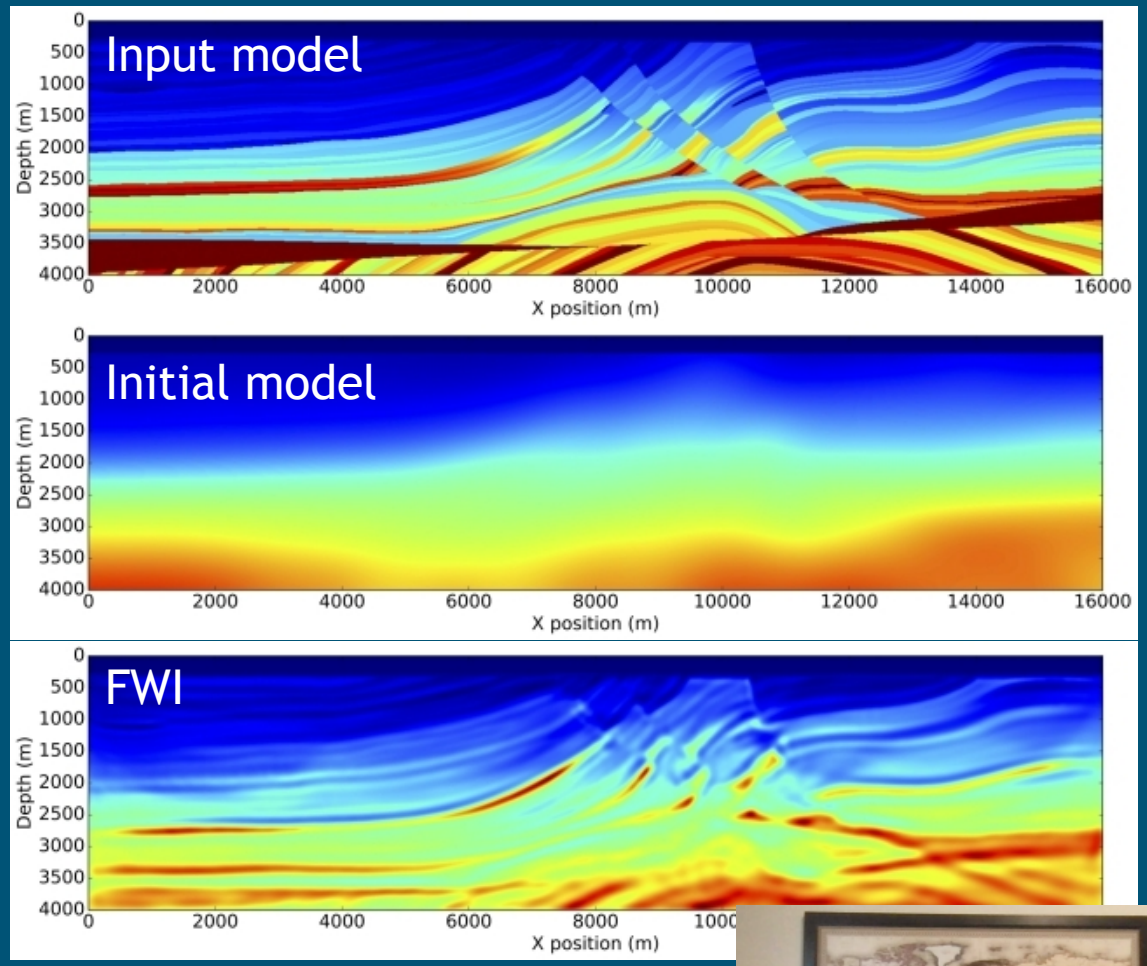
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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.

Future of waveform modeling



Aspirational chart pulled from Steve Myers 2021 NEM



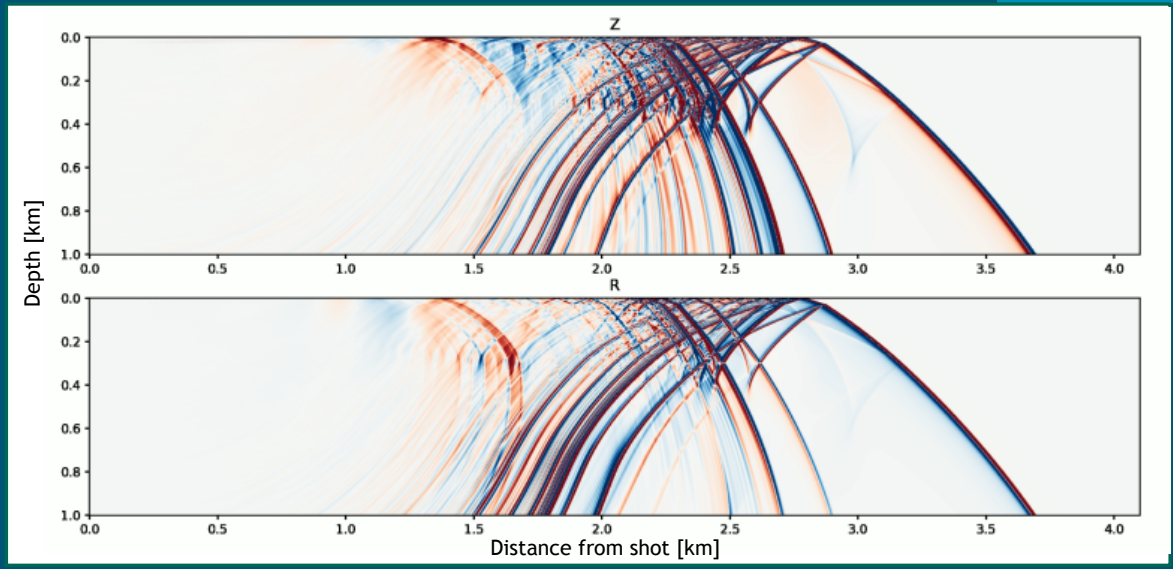
Example full-waveform inversion from Devito project

CUDA 2D Simulator

- A GPU implementation of the 2D Finite Differences code used by Caltech (Helmberger, Clayton, Vidale, et al.*)
  - Pushing the simulation to GPUs provides fast calculation and natural parallelization.
  - Simulation only considers Vp, Vs, and density (i.e. it does not account for intrinsic attenuation or lattice preferred orientation anisotropy).
  - FD is 8th-order in space, 2nd order in time
  - 2D significantly reduces the computational domain.
    - Higher spatial and temporal resolution than 3D versions.
    - P-SV and SH systems have to be run separately.
    - Out of plane scattering and attenuation is not fully accounted for.
- Simulations organized through a Python wrapper (Nathan Downey)
  - The interface script facilitates a natural method to define synthetic sources and receivers.
  - Methods are included to test 1D models or path-specific 2D cross-section models extracted from 3D models, such as CRUST1.0.
  - Functions allow the user to manipulate the 2D models with additions such as stochastic structural perturbations or overlaying parts of one model onto the existing model.
  - Built-in sanity checks prevent runs with unstable velocity models or source parameters.

Li, Helmberger, Clayton, Sun, 2014, GJI

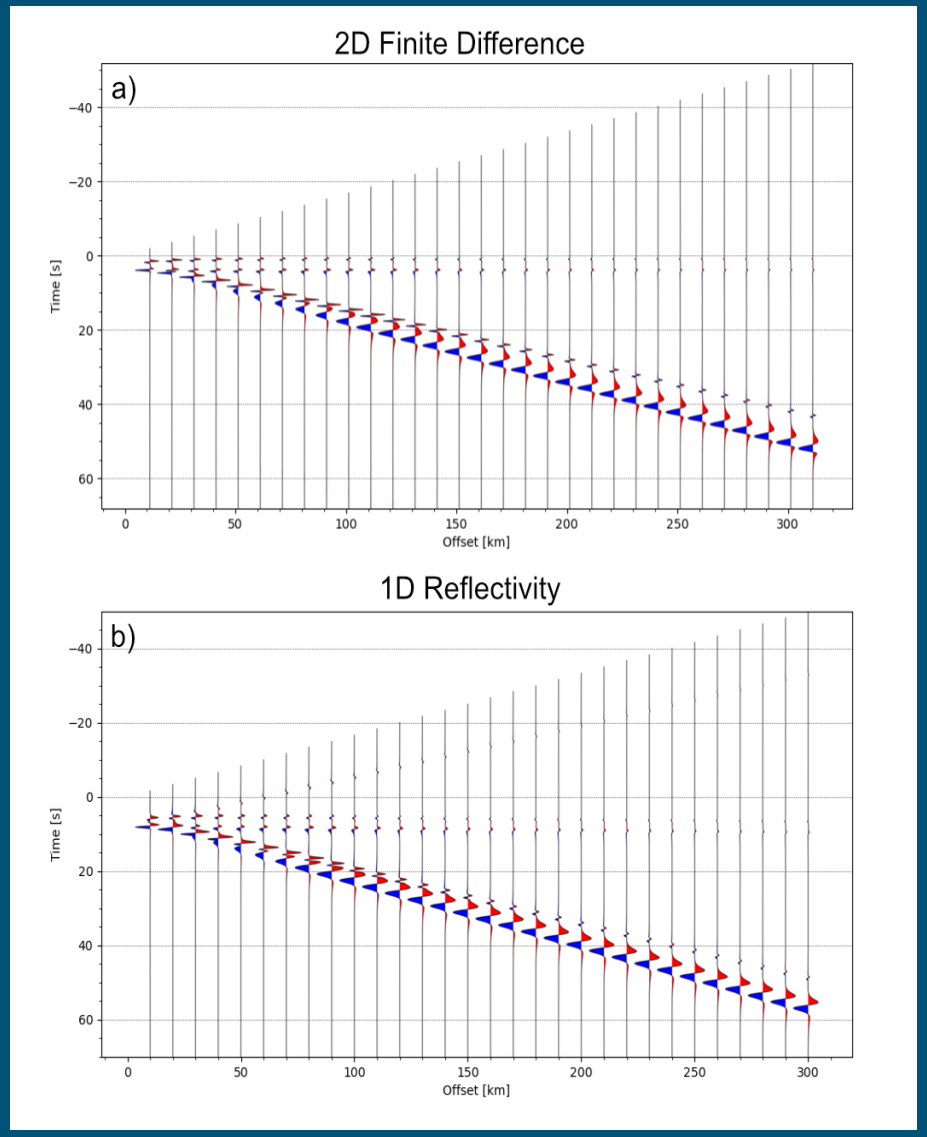
Shallow explosive source recorded in the near-field




```
def setupJobTomoLE(receiverTuple):
    receiverIndex = receiverTuple[0]
    receiverCoord = receiverTuple[1]
    receiverOutputName = "SYNTH."+receiverCoord[3]+"."+receiverCoord[4]
    thisName = os.path.join(outputdir, "TOMO_LE", receiverOutputName)
    sim = simulator.CUDA2DSimulation(simDir=thisName,numGPUs=1,simTypes=['PSV', 'SH'])

    rec = simulator.Receiver(receiverCoord,stationName = receiverCoord[4], network = receiverCoord[3], location = receiverCoord[5])
    sim.setReceiver(rec)

    sim.setSource(srcLE)

    model = simulator.TomographyModel(beginPad=beginpad, endPad=endpad, h=h, minDepth=mindepth, maxDepth=maxdepth,
fname='tomo_models/PrelimModel_UTAH_GNEM.cdf',modType='PandS', vpvs=1.8)
    model.readNetCDFModel()
    layervels = np.arange(2.5,8.5,0.5)
    #model.convertModelToLayerCake(layervels, velType='P', xsmooth=10, ysmooth=10)
    #model.gaussian3DSmoothModel(xnodes=15, ynodes=15, znodes=15, sigma=0.8)
    model.setGeometryFromSim(sim)
    model.getProfile()
    model.mask_with_crust1(layer=0, replaceMantle=False)
    model.mask_with_crust1(layer=1, replaceMantle=False)
    model.mask_with_crust1(layer=2, replaceMantle=False)
    model.mask_with_crust1(layer=3, replaceMantle=True)
    #model.slowTheMantle(mv=0, vred=0.25, minvel=1.0)
    model.convertToLayerCake(layervels, velType='P', xsmooth=10)
    model.add_static_shift(velshift=0.3, veltype='vs')
    model.addPerturbations(corrL=[corrLx, corrLy], scalefac=scalefac, seed=seed)
    #model.addLayerDependentPerturbations(layervels=layervels, veltype="P", corrL
    model.outputProfile(flatten=True)
    model.imageProfile(showPlots=True, flatten=True, amplitudeScales='manual')
    sim.setModel(model)
    sim.assertStability(dt=dt)
    sim.okaya_checks(dt=dt, targetCFL=0.5, f_max=5, dispersionConditionMin=2.8)
    sim.setSimulationParameters(nt=nt, dt=dt, itrecord=1, itprint=100)

    return sim
```

Eye test showing how easy it is to setup a simulation

Code validation: other synthetic methods

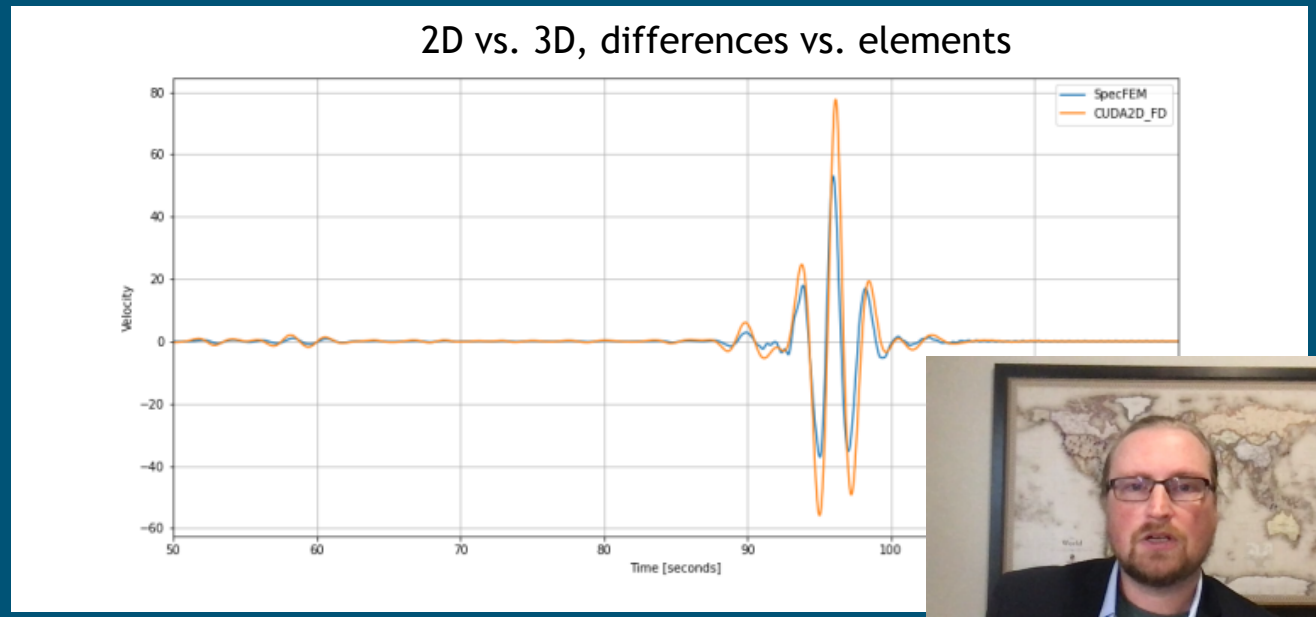


Single homogeneous half-space

The 1D Reflectivity code [Kind, Tibi] is fast, but struggles with complex media, particularly low velocity zones.

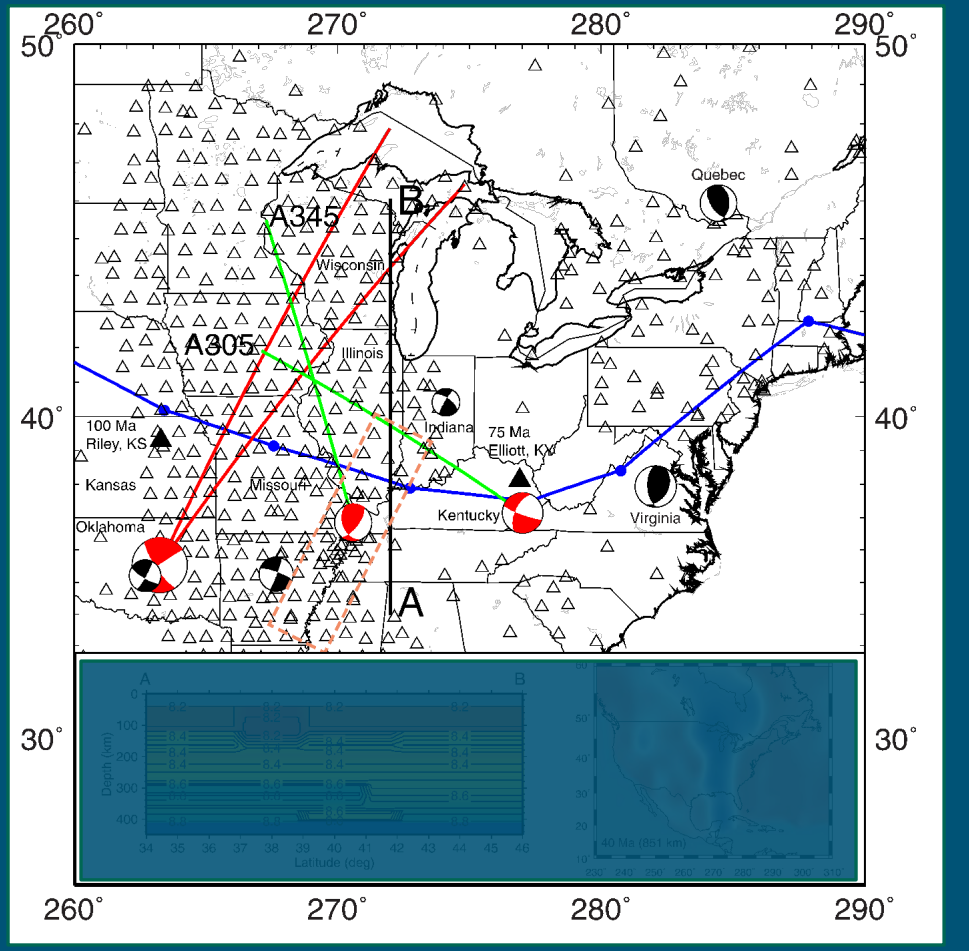
SpecFEM3D is the academic world standard for global to regional scales. However, it becomes computationally intractable for local, high frequency cases.

As with any engineering problem, it comes down to optimizing between three competing preferences. In this case, speed, accuracy, and medium complexity.

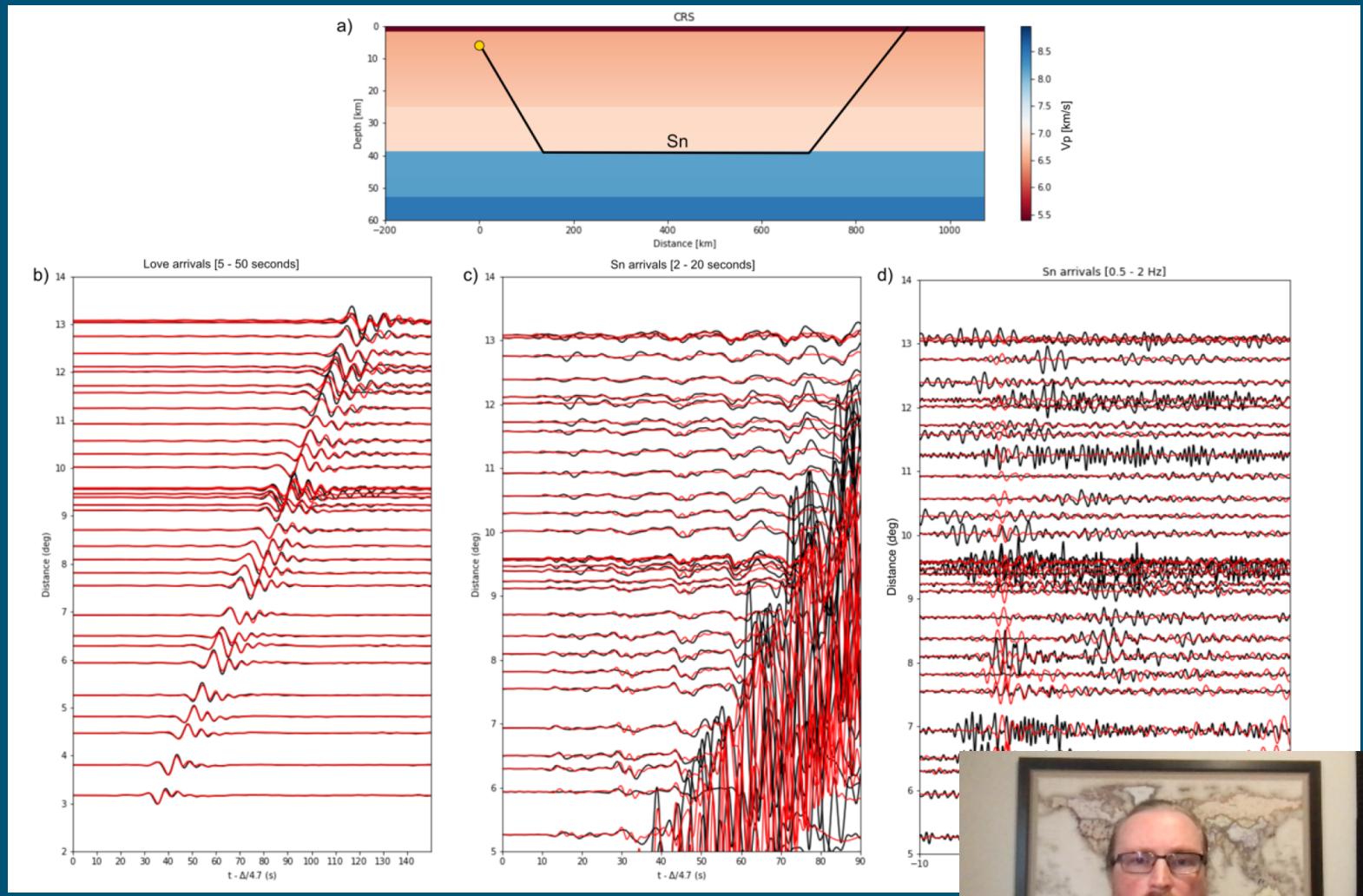


Single homogeneous half-space, dirac comb STF

Code validation: published Oklahoma event

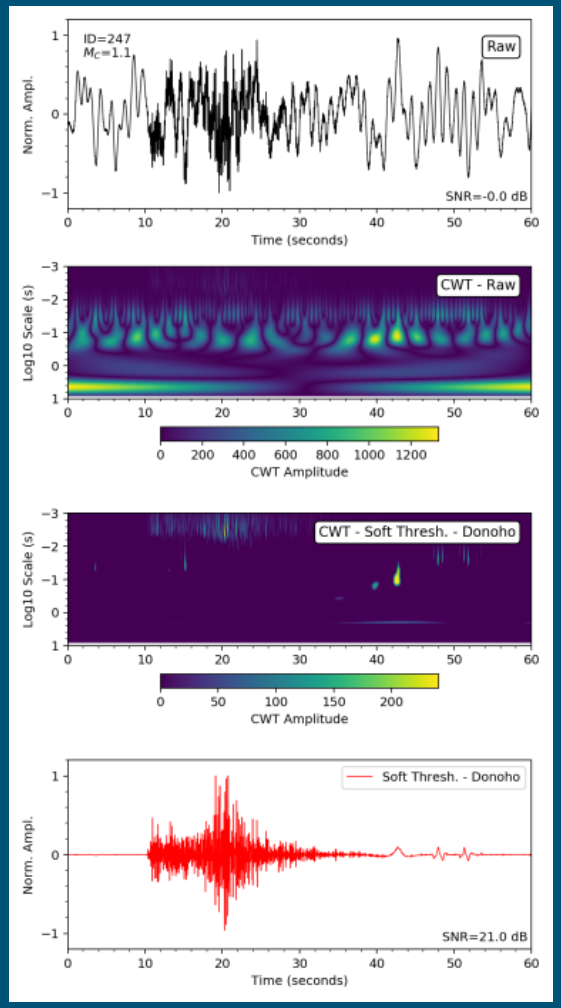


Chu and Helmberger, 2014, G³



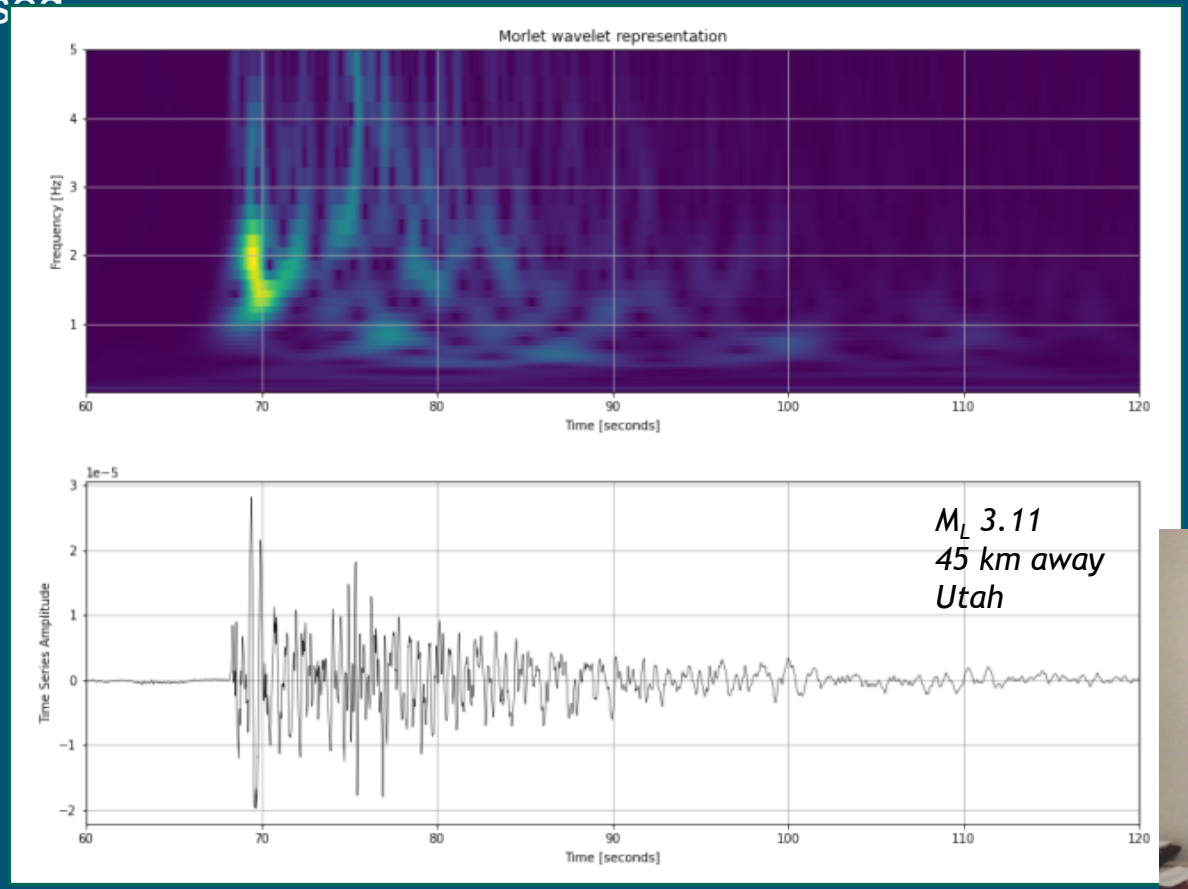
Continuous Wavelet Transform

Following Mousavi and Langston's demonstration of the CWT for denoising, we've translated their Matlab software into Python for adaption into our workflows and experimentation.

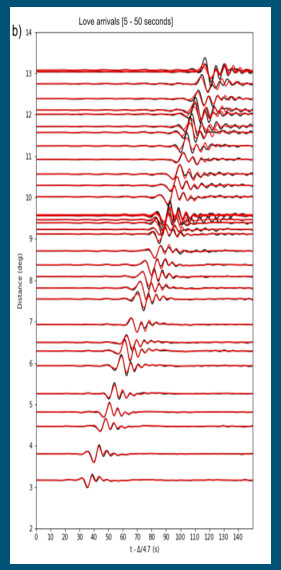


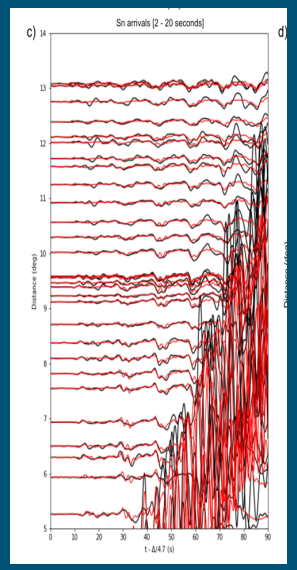
Tibi, Koch, in progress

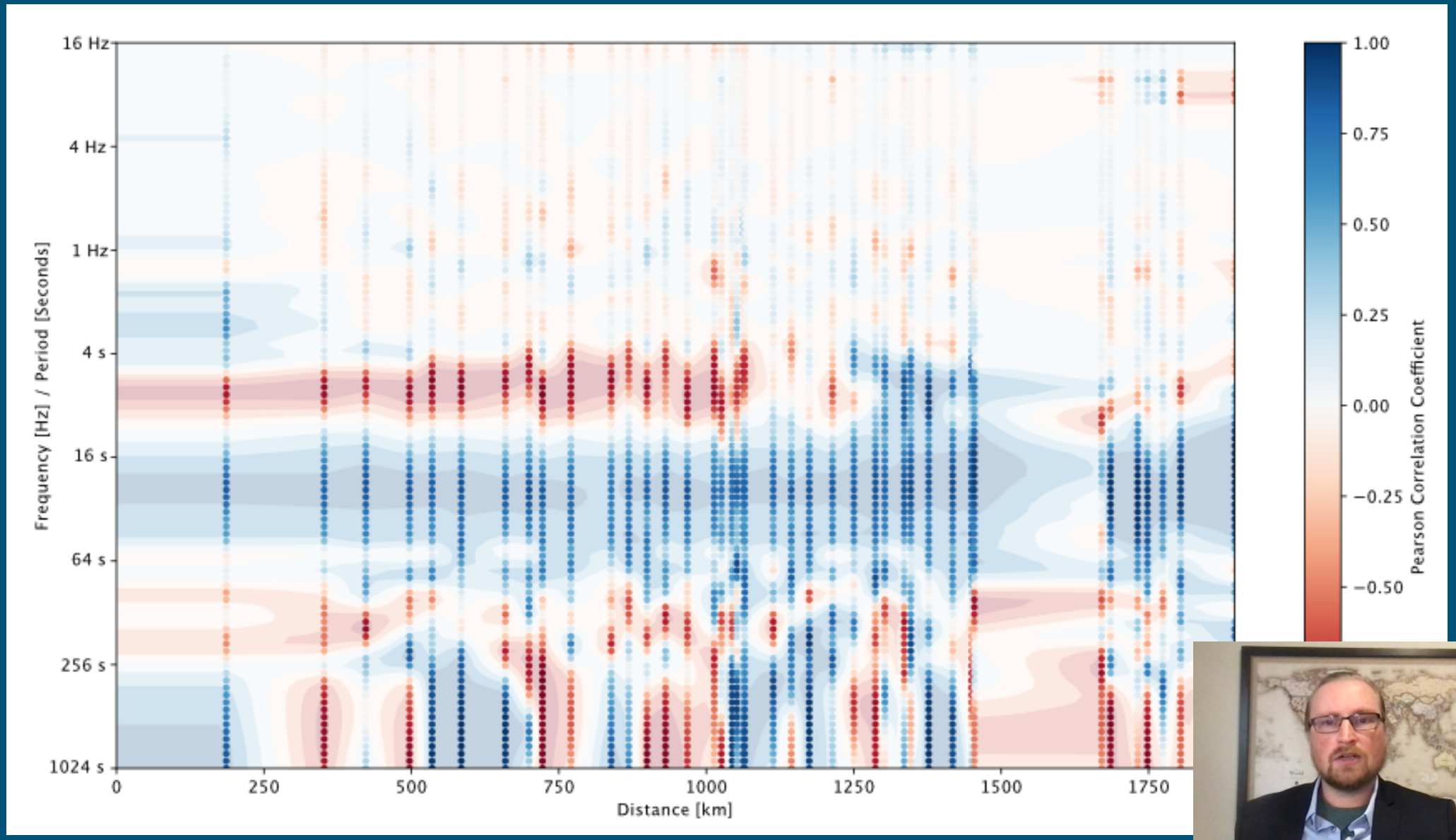
Surface level advantage is that it decomposes a waveform into time-frequency space without short-time windowing needed for standard spectrograms. Deeper level advantages may stem from changing which mother wavelet(s) are used



Observed to synthetic correlation



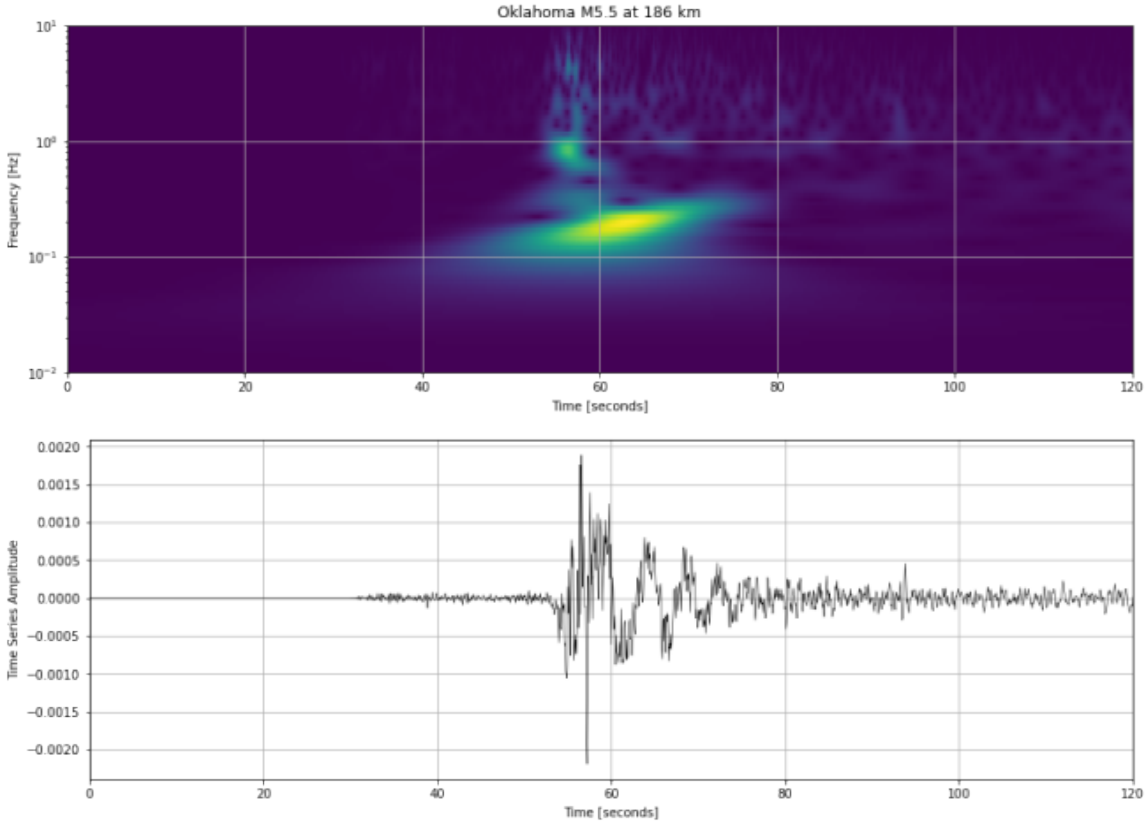




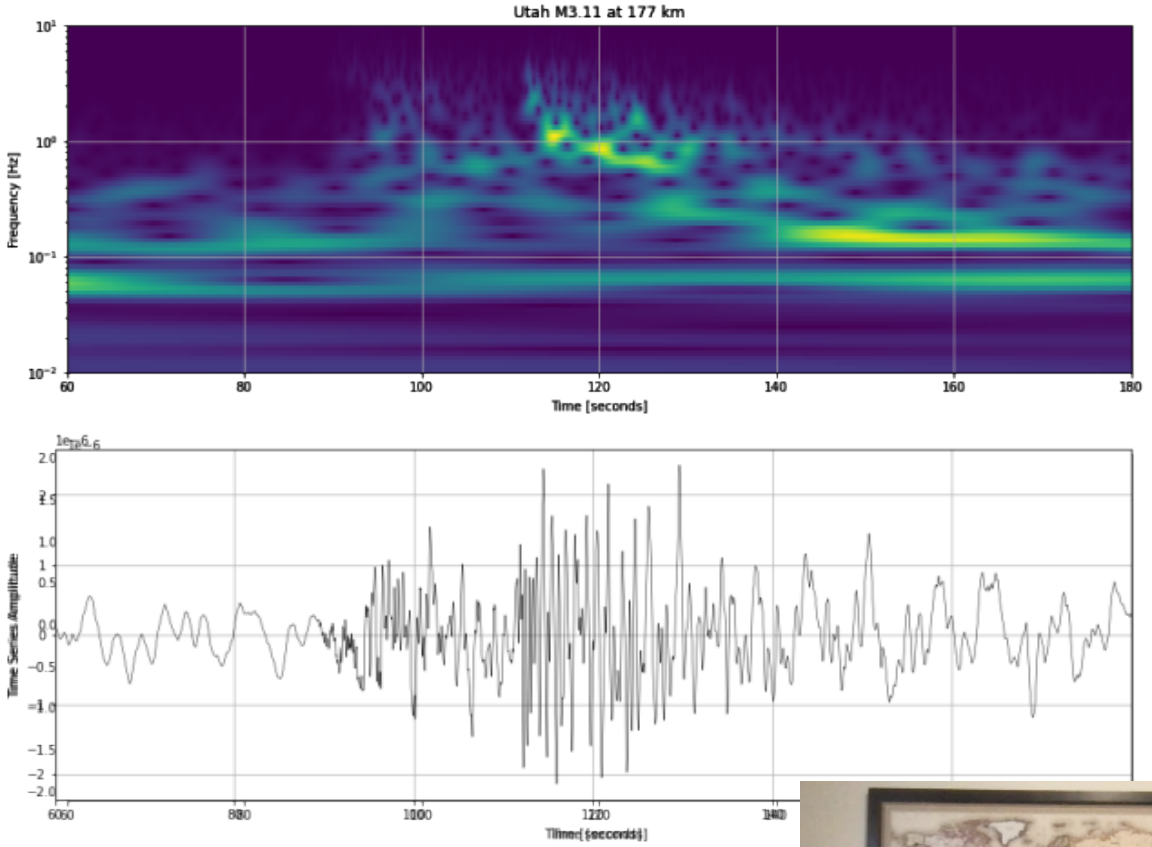
Oklahoma M5.5 as used in Chu and Helmberger, 2014

Oklahoma M5.5 vs. Utah local M3.11

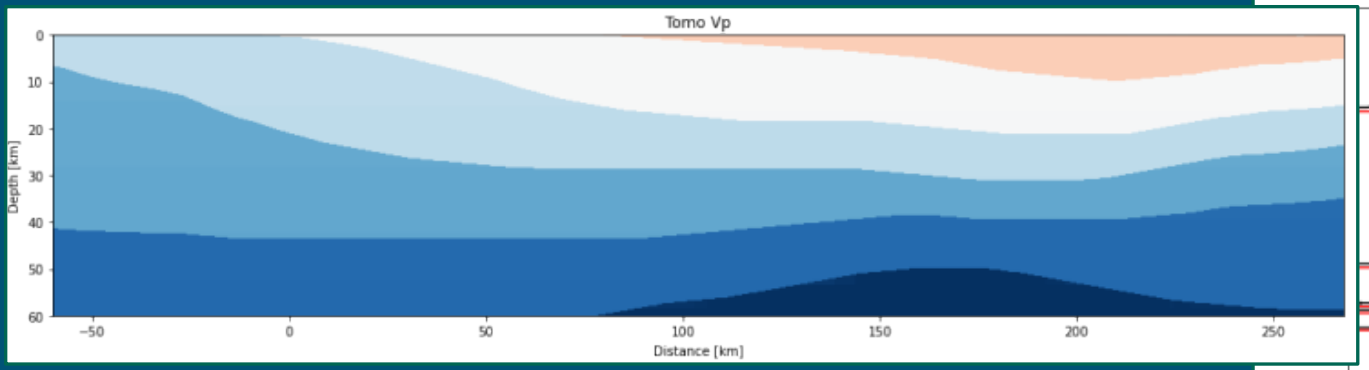
Dominated by Love wave. 5.5 km depth

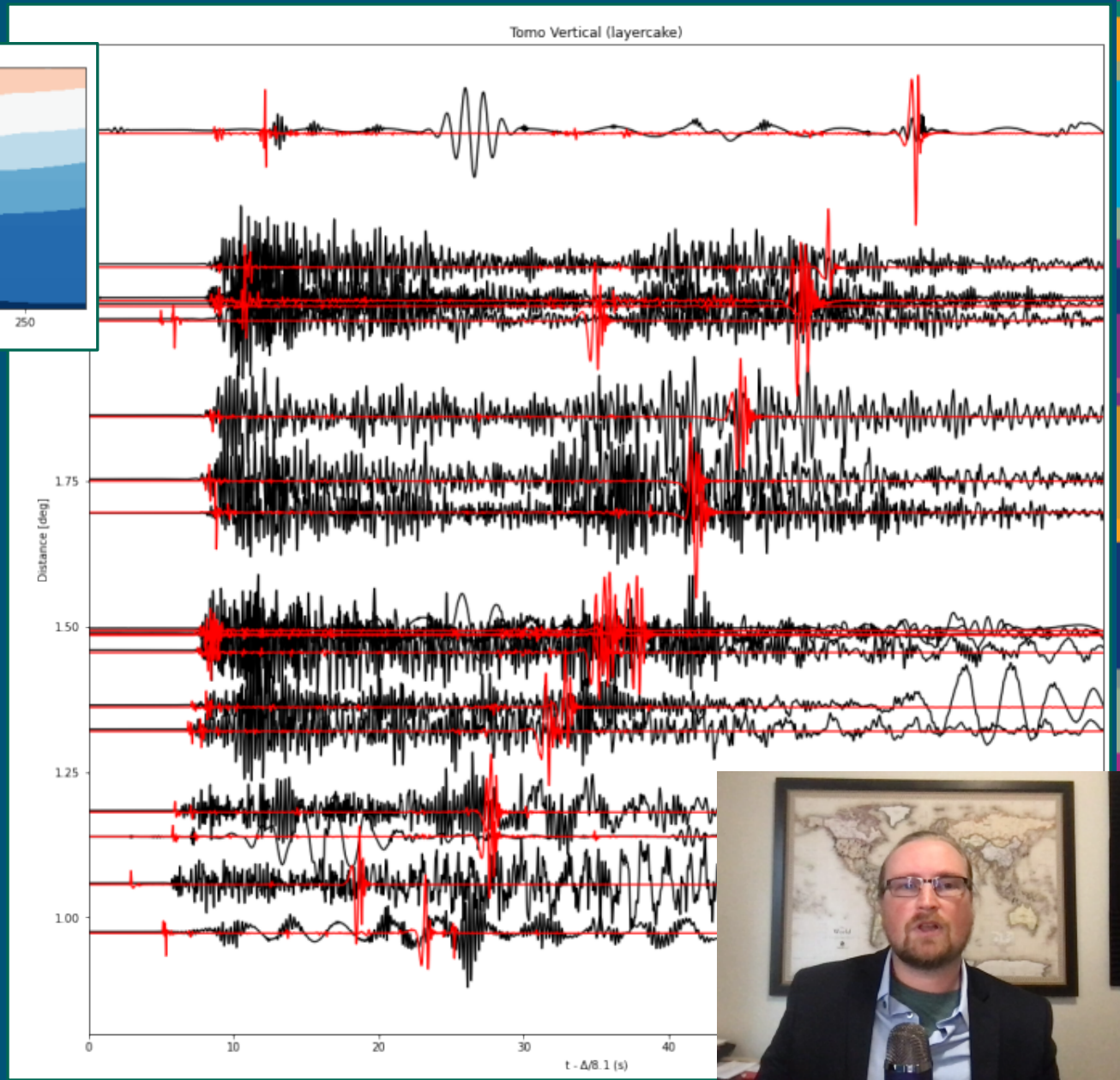


Dominated by S coda. 1 km depth



Simulations with a Utah local 3D tomography model

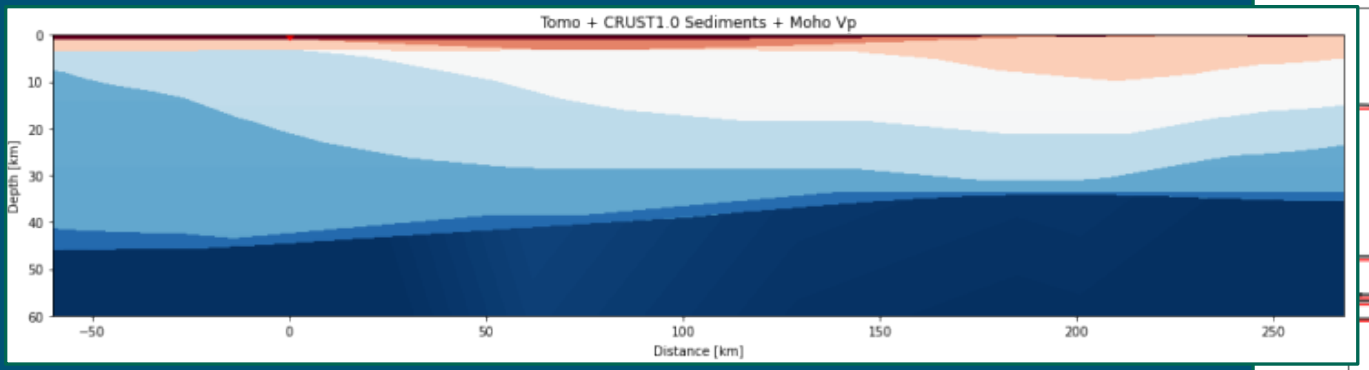


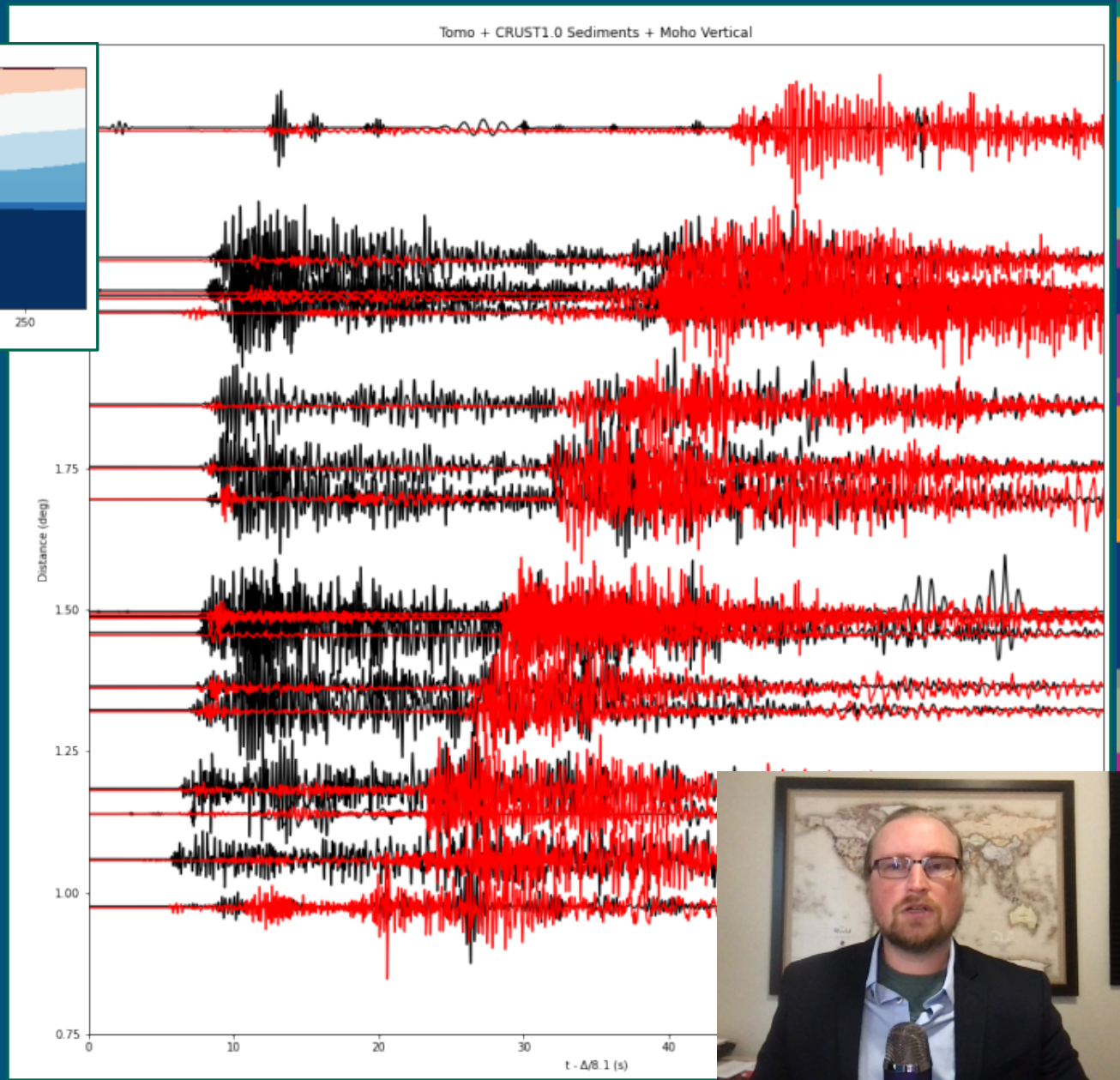


Synthetics (red) based on the above tomography model show approximately close arrival times but the coda amplitude is completely missing.

Wavelet domain denoising does cleanup the observed (black) fairly well pre-arrival, but the ‘signal-generated noise’ is still significant.

Simulations with a Utah local 3D tomography model and CRUST1.0

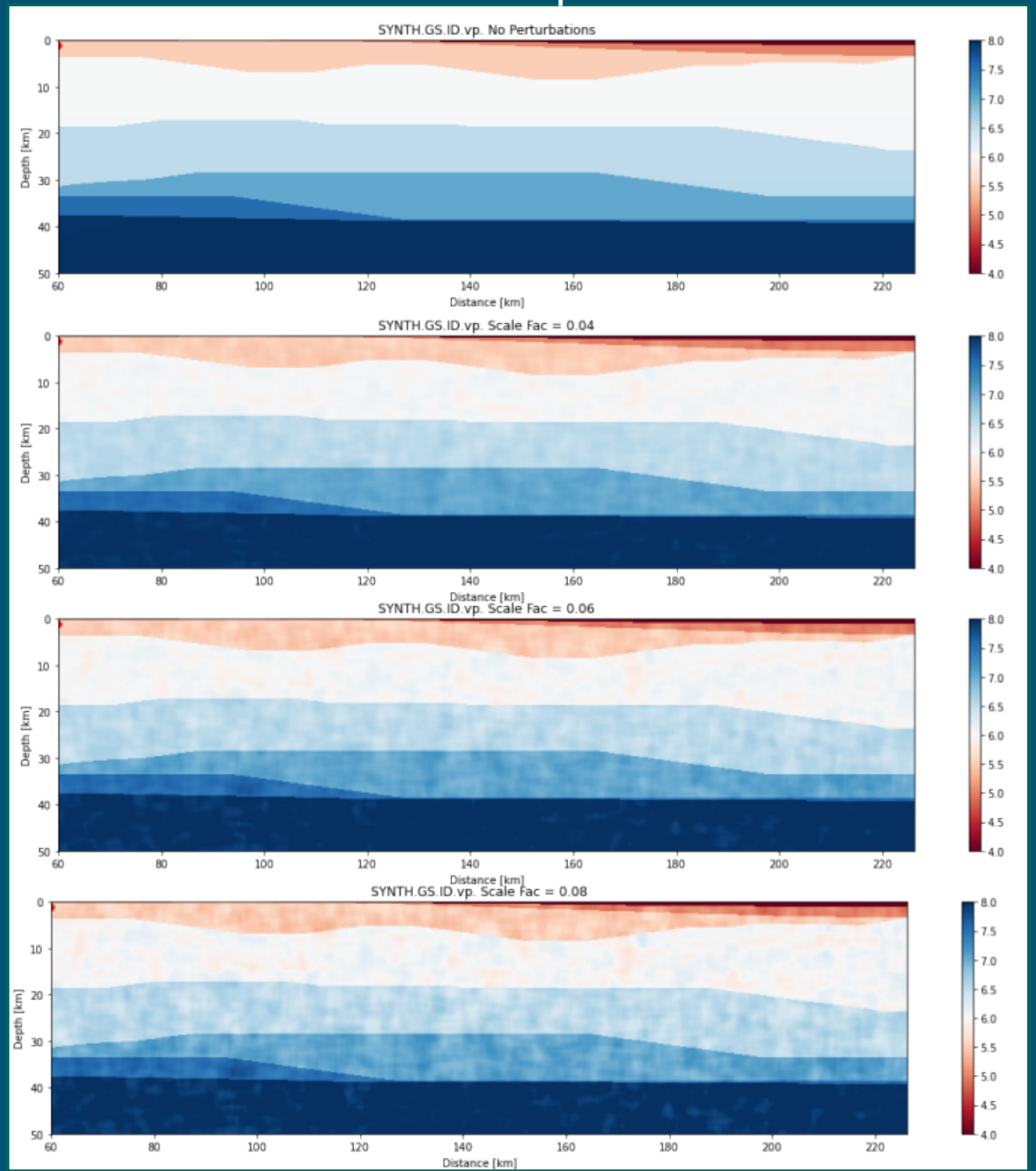


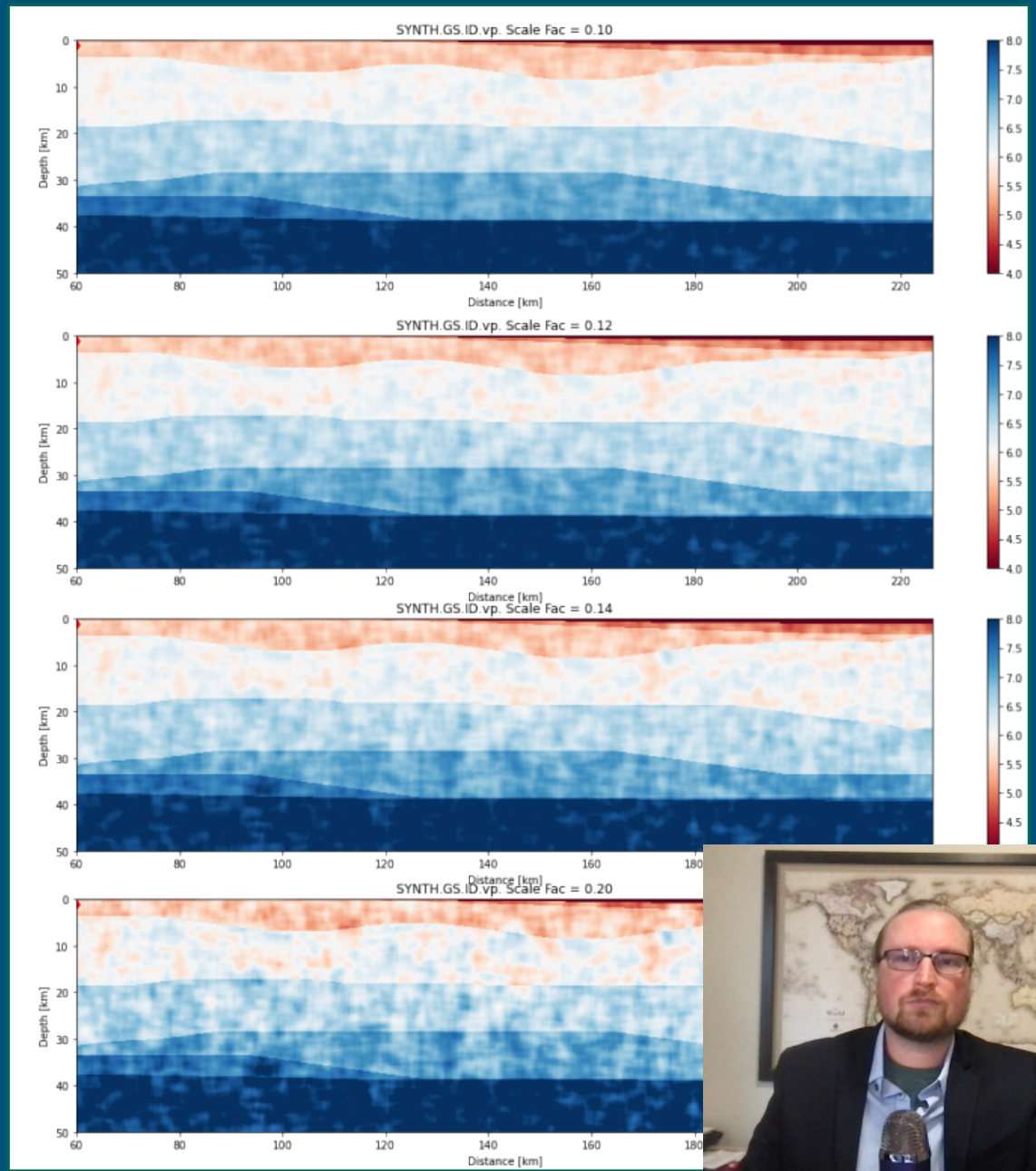


Adding the sedimentary layers from CRUST1.0 creates a significantly more realistic surface wave train as well as the P wave coda.

While the current example also includes the Moho from CRUST1.0, it has a negligible effect compared with the sedimentary layer.

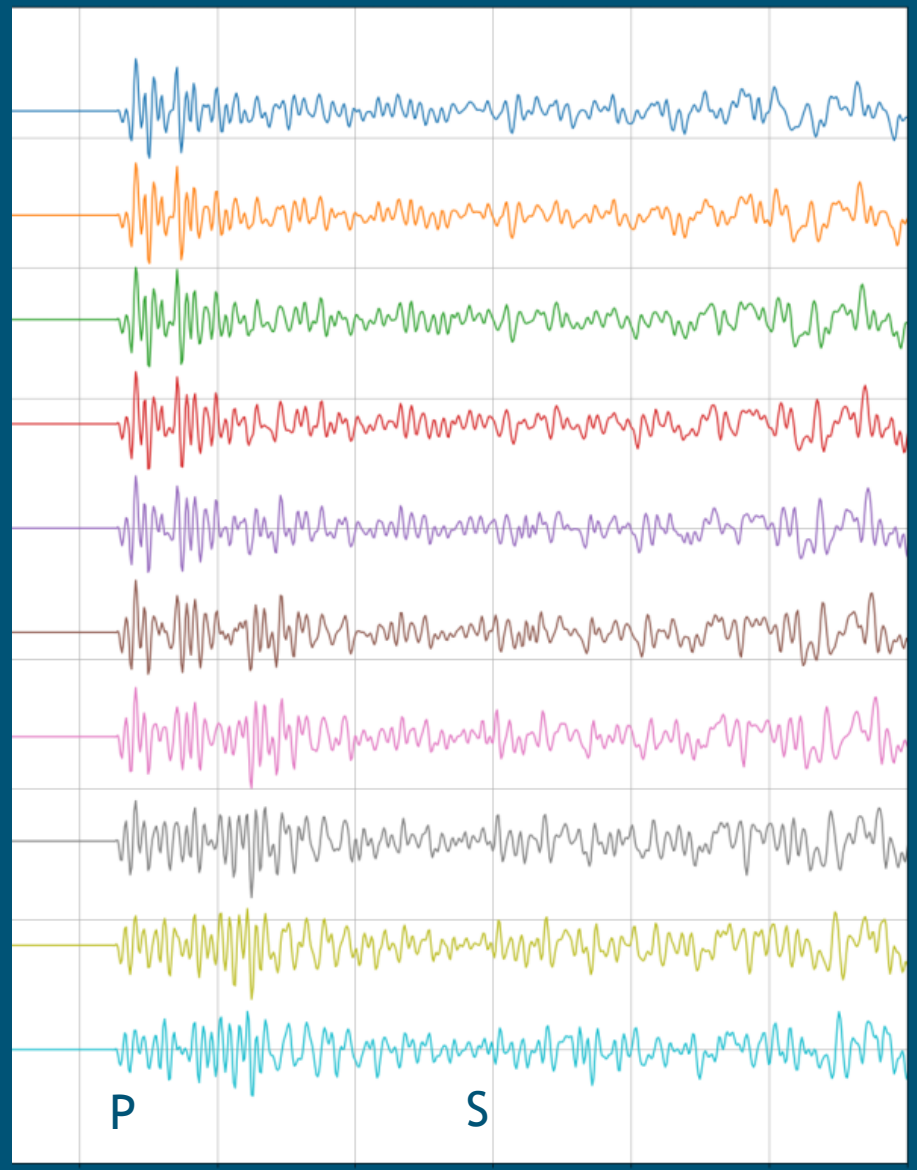
Effect of stochastic perturbations



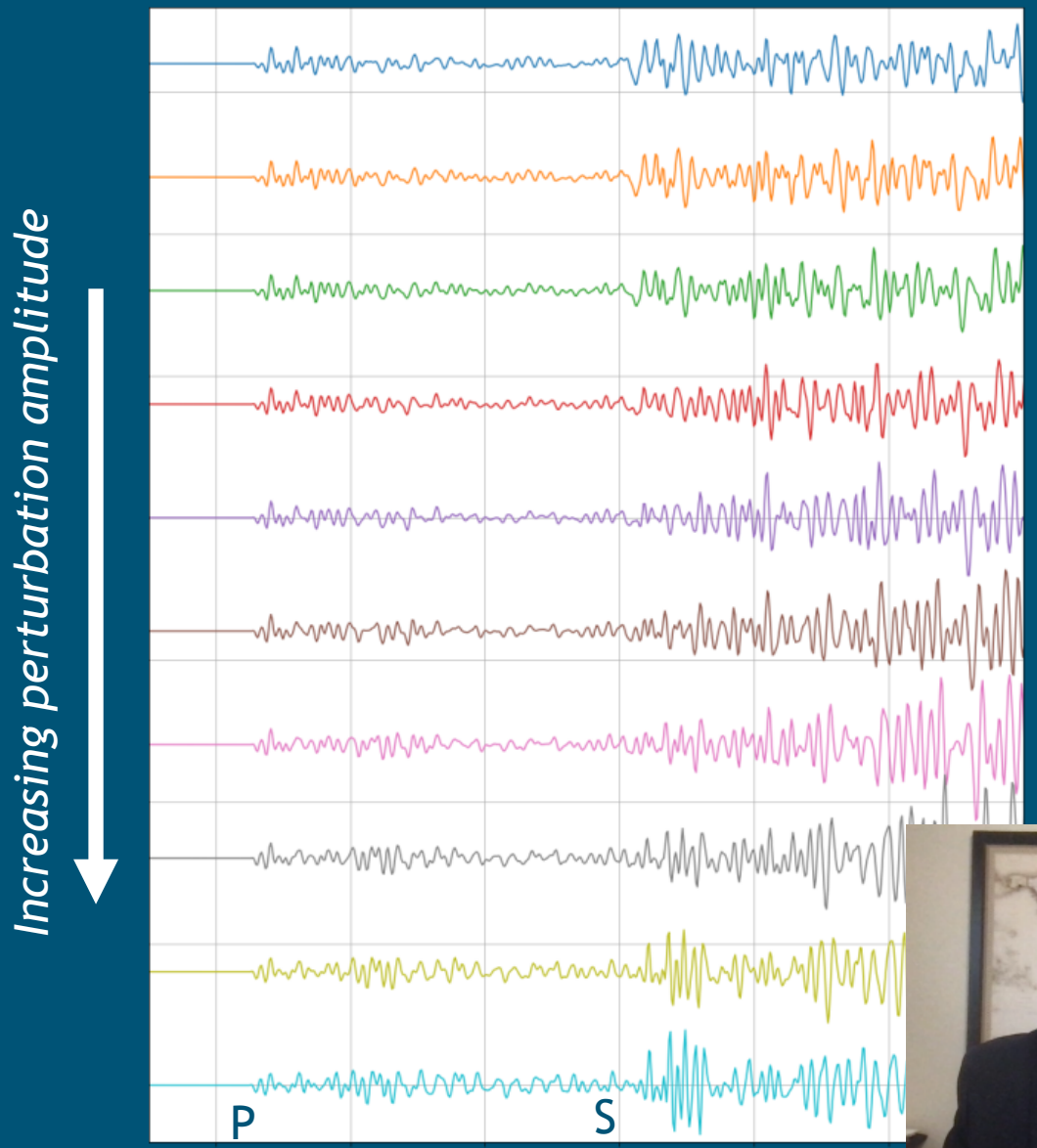


Resulting waveforms

Vertical, explosion source



Vertical, double-couple source



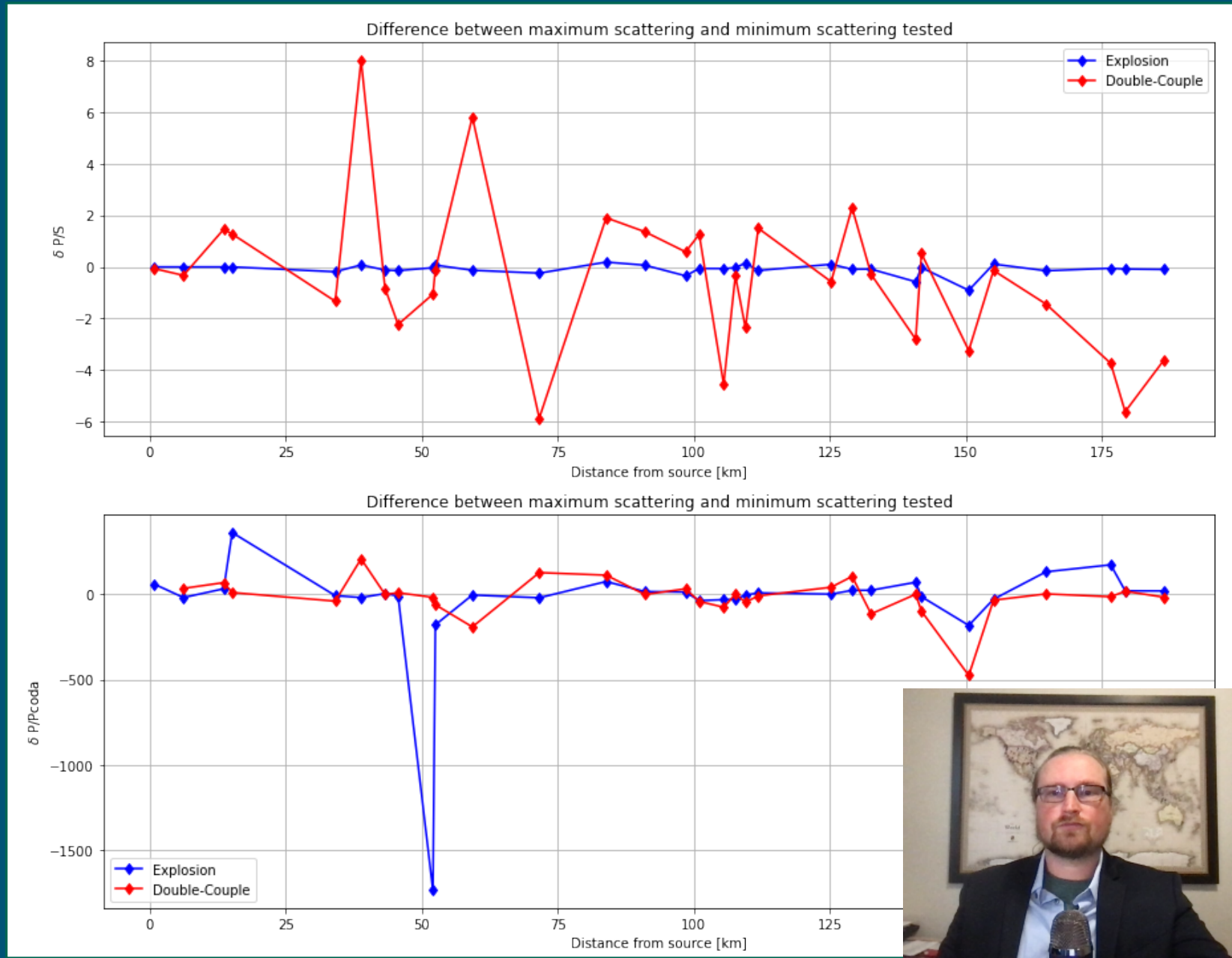
Quantifying P/S and P/Pcoda

These plots show the difference between the maximum perturbation value tested and the minimum perturbation value tested.

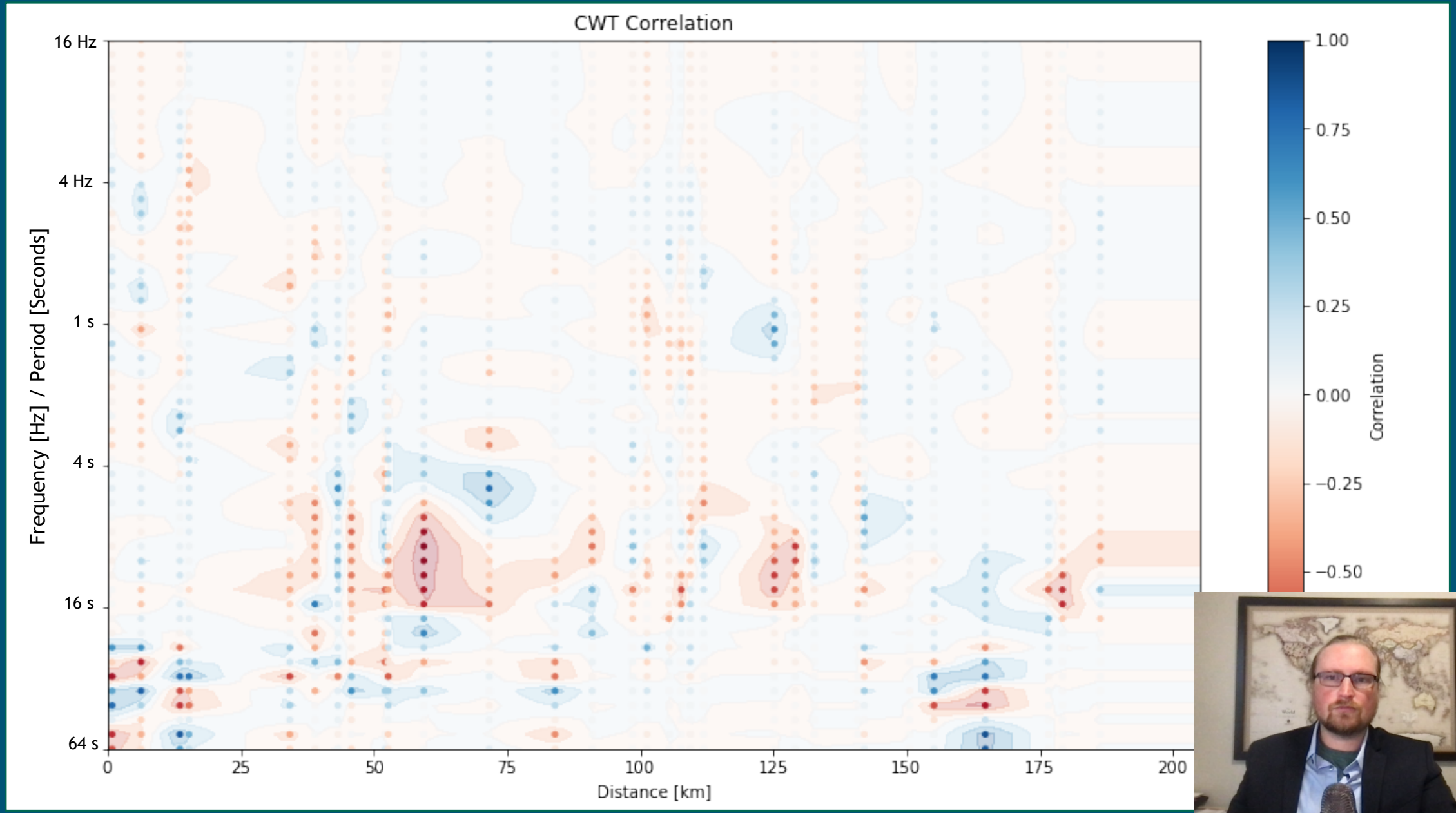
Variability is dominant, but may reflect variable background structures as this is not a true record section along one structural swath.

Nonetheless, we do see for the earthquake case, large P/S ratio for large scattering near the source and the reverse is the case for more distant stations.

Despite our visual inference previously, there is no quantifiable difference in P/P coda ratio.



CWT Correlation for M3.11 in Utah



Next Steps

- Develop and test more metrics for parameter similarity.
- Continue to explore signals in the wavelet domain.
- Test envelope functions.
- Look to fit large events at short periods to push towards higher frequencies and smaller events.
- Quantify trade-offs with 3D codes.
- Explore Full-Waveform Inversion for regional scales.

Summary

- We’re working with an efficient 2D finite differences simulator which has been validated against alternative 1D and 3D methods and real-world earthquakes.
- Decomposing the waveforms into the wavelet domain provides a method of quantifying data fit as a function of frequency.
- Shallow, low velocity structure and stochastic structural perturbations have significant effects on the surface wave and coda amplitudes at high frequencies.
- Validating synthetic waveforms at frequencies above ~1 Hz and for small events may require a focus on matching waveform based measurements, rather than wiggle for wiggle replication.