Computational Framework for Surface-Wave Seismic Analysis Feasibility Study

Student: Isaac M. Smith
Course: PHY 386 Undergraduate Research I
Advisor: Dr. Sarah Morrison
Advisory Committee: Dr. Mike Reed, Dr. Kevin Mickus
Institution: Department of Physics, Missouri State University

Abstract

This feasibility study will be evaluating the proposal of a physics-based computational framework for processing seismic surface-wave data and estimating shallow subsurface shear-wave velocity profiles. The physical phenomenon for project is the dispersion of Rayleigh surface waves in layered elastic media, where frequency-dependent phase velocity can be used to infer subsurface shear-wave velocity structure. The project will combine signal processing, frequency wavenumber analysis, phase shift dispersion imaging, numerical inversion, and parallel processing in Python. Existing tools such as SurfSeis and Geopsy demonstrate the practical value of MASW and surface-wave inversion, but the proposed framework will be a transparent, modular, research tool suitable for computational physics investigation. This study concludes that the project is feasible if its scope is limited to a prototype capable of reading standard seismic formats, preprocessing multichannel records, generating dispersion images, performing basic curve picking, and testing a constrained inversion workflow on well-characterized sample data.

Introduction

Near-surface seismic methods are often used to estimate the structure of shallow subsurface materials. Multichannel analysis of surface waves, or MASW, is very valuable because it uses the dispersive behavior of Rayleigh surface waves to predict shear wave velocity, denoted as V_S. Park, Miller, and Xia describe MASW as a method that uses the frequency-dependent properties of Rayleigh waves to image and characterize the shallow subsurface (800). Since shear-wave velocity is directly related to the shear modulus by the equation

$$V_S = \sqrt{\frac{\mu}{\rho}},$$

where μ is the shear modulus and ρ is density, V_S provides a measure of subsurface stiffness.

The ability to estimate V_S is important in geophysics, earthquake engineering, groundwater studies, and environmental investigations. Xia, Miller, and Park also mention that the shear-wave velocity of near-surface materials strongly affect seismic-wave propagation and is integral in engineering and environmental applications (691). Surface-wave methods are preferable because they can be performed noninvasively and can provide useful estimates of subsurface stiffness without requiring boreholes.

There are several software tools that already support surface-wave processing. SurfSeis is widely known for MASW processing and dispersion-curve analysis, while Geopsy provides an open-source environment for seismic signal processing and surface-wave inversion. Wathelet et al. describes Geopsy as a mature open-source package for ambient vibration processing and site

characterization (1878). These programs show that the scientific and engineering problem is very well defined. Although, the purpose of this project is not to replace any professional software. Instead, the proposed framework is designed as a computational physics research project in which the mathematical, physical, and numerical steps of surface-wave analysis will be implemented.

This paper evaluates whether a Python-based surface-wave analysis framework can be finished within a realistic undergraduate research timeline. The main feasibility question is: Can a modular Python-based framework be designed and partially implemented to show surface-wave dispersion imaging and shear-wave velocity inversion while remaining computationally and scientifically manageable for an undergraduate physics project?

PHY 386 Feasibility Context and Project Scope

The complete vision for the project includes data import, preprocessing, dispersion imaging, manual or semi-automated dispersion picking, inversion, and graphical visualization. However, to make the undergraduate scope more feasible, the vision must be narrower. The minimum scope of the PHY 486 project should consist of the following:

1. Import seismic shot-gather data from at least one standard format, liekly SEG-2 or SEG-Y.
2. Apply basic preprocessing, including detrending, tapering, filtering, and normalization.
3. Generate dispersion images using either frequency-wavenumber analysis or the phase-shift method.
4. Implement a simple dispersion-curve picker or allow manual picking from the image.
5. Perform a constrained inversion or forward-model comparison for a layered V_S profile.
6. Benchmark runtime for serial and parallel versions of at least one computational step.
7. Compare results qualitatively with published methods or outputs from established tools where possible.

Some of the advanced goals like a fully polished graphical interface, field-data automation, or machine-learning-based picking, will be treated as stretch goals rather than required outcomes.

Background and Theory

Elastic Wave Propagation

The physical foundations of the proposed framework are in elastic wave propagation. In an isotropic linear elastic medium, displacement u satisfies the elastic wave equation

$$\rho \frac{\partial^2 u}{\partial t^2} = (\lambda + \mu)\nabla(\nabla \cdot u) + \mu \nabla^2 u,$$

where ρ is density and λ and μ are the Lamé parameters. This equation separates into compressional and shear wave behavior, with velocities

$$V_P = \sqrt{\frac{\lambda + 2\mu}{\rho}}$$

and

$$V_S = \sqrt{\frac{\mu}{\rho}}.$$

Surface-wave analysis is sensitive to V_S, this is why MASW and other similar methods are commonly used to infer shallow shear-wave velocity structure.

Rayleigh Waves and Dispersion

Rayleigh waves are surface waves that travel along the free surface of an elastic medium. In a homogeneous space, Rayleigh waves have nearly constant phase velocity. In a layered medium, their phase velocity becomes frequency dependent. This frequency dependence is called dispersion.

The reason for Rayleigh-wave dispersion is that different wavelengths sample different depths. Shorter wavelengths are more sensitive to shallow layers, while longer wavelengths sample deeper structure. The measured dispersion curve,

$$c = c(f),$$

where c is phase velocity and f is frequency, contains information about how elastic properties vary with depth. Foti et al. explains that surface-wave dispersion analysis is commonly used to infer a shear-wave velocity model of the subsoil through the solution of an inverse problem ("Application of Surface-Wave Methods for Seismic Site Characterization." 777).

Frequency-Wavenumber Analysis

A multichannel seismic record can be represented as a wavefield $u(x, t)$, where x is receiver offset and t is time. Frequency-wavenumber analysis transforms this wavefield into the spectral domain:

$$U(k, \omega) = \iint u(x, t) e^{-i(kx - \omega t)}\, dx\, dt,$$

where k is wavenumber and $\omega = 2\pi f$ is angular frequency. Dispersion energy appears as peaks in the fk domain. Since phase velocity is related to frequency and wavenumber by

$$c = \frac{\omega}{k},$$

the dispersion curve can be obtained by locating spectral maxima across frequency. Park, Miller, and Xia's dispersion-imaging work demonstrates that multichannel surface-wave records can be transformed into images of multimode dispersion curves (1377).

Phase-Shift Dispersion Imaging

An alternative to fk analysis is the phase-shift method. In this approach, traces are transformed into the frequency domain and then phase corrected for a range of trial velocities. For each frequency and trial velocity, energy is summed across offsets. The correct trial velocity produces constructive interference, creating a peak in the dispersion image. This method is well suited for MASW because it can produce high-resolution dispersion images even from practical field spreads with limited receiver offsets (Park, Miller, and Xia 1377).

Inversion and Non-Uniqueness

Once a dispersion curve has been measured, the next step is inversion. The goal is to find a layered-earth model

$$m = \{V_{S,i}, V_{P,i}, \rho_i, h_i\}$$

that produces a theoretical dispersion curve close to the observed curve. The function can be written as

$$\Phi(m) = \sum_i w_i \left[c_{\text{obs}}(f_i) - c_{\text{cal}}(f_i; m)\right]^2,$$

where c_{obs} is the observed phase velocity, c_{cal} is the calculated phase velocity for model m, and w_i are data weights. Xia, Miller, and Park show that Rayleigh-wave phase velocity is most strongly influenced by S-wave velocity in the high-frequency near-surface range, followed by layer thickness (691). This information supports a practical inversion strategy where V_S and layer thickness are emphasized, while V_P and density may be constrained or estimated.

However, surface-wave inversion is not unique. Foti et al. explains that different shear-wave velocity profiles can fit the same measured dispersion curve with uncertainty, creating an equivalence problem ("Non-Uniqueness" 982). Wathelet, Jongmans, and Ohrnberger also emphasize that dispersion-curve inversion is nonlinear and generally non-unique, making stochastic or direct-search methods useful alternatives to purely local inversion methods (211). Therefore, the proposed framework should not present a single V_S profile as a result. Instead, it should report model uncertainty, misfit, and sensitivity when possible.

Research Objectives

The proposed project has five primary objectives.

First, the proposed framework will provide a transparent computational workflow for surface-wave analysis. The software should make each stage visible: data import, preprocessing, transformation, dispersion imaging, picking, inversion, and plotting.

Second, the project will investigate how parallel processing can improve computational efficiency. Python's threading documentation notes that threads are useful for I/O-bound work but limited for CPU-bound computation because of the Global Interpreter Lock (Python Software Foundation, "threading"). Python's multiprocessing module can use subprocesses to bypass the Global Interpreter Lock and better use multiple processors (Python Software

Foundation, “multiprocessing”). Therefore, the project will use threading primarily for interface responsiveness and multiprocessing for CPU-heavy calculations.

Third, the framework will compare dispersion-imaging strategies, especially fk analysis and phase-shift imaging. Implementing more than one method will allow the project to compare numerical behavior and image quality.

Fourth, the project will explore dispersion-curve picking. The feasible baseline will be manual or deterministic ridge picking. Machine-learning-assisted picking may be investigated as a future extension, since recent work by Ren et al. demonstrates that convolutional neural networks can be used to automate multimode dispersion-curve picking (1173).

Fifth, the project will test a constrained inversion workflow for estimating shallow V_S profiles. The goal is not to create an industry inversion package, but to demonstrate the connection between measured dispersion and layered-earth shear-wave velocity.

Proposed Methods

Data Input and File Handling

The proposed framework will target SEG-2 and SEG-Y data because these are common seismic data formats. Pullan’s SEG-2 standard was developed for seismic and radar data files in the personal computer environment (1260). SEG-Y is also a standard seismic data-exchange format; the SEG Technical Standards Committee released SEG-Y Revision 2.0 in 2017 to extend the format’s flexibility and compatibility.

ObsPy is an appropriate Python library for the project because it supports seismological data handling. Beyreuther et al. describes ObsPy as a Python toolbox intended to bridge interactive seismological analysis and automated data processing (530). ObsPy documentation also includes SEG-2 read support and SEG-Y read/write support (ObsPy Development Team, “obspy.io.seg2”; ObsPy Development Team, “obspy.io.segy”).

A feasible implementation will begin with one known test file format. If both SEG-2 and SEG-Y prove too time-consuming, the project will prioritize whichever format is most relevant to available data.

Signal Preprocessing

The preprocessing module will prepare raw traces for dispersion imaging. Proposed steps include:

- demeaning and detrending,
- tapering to reduce edge effects,
- band-pass filtering,
- amplitude normalization,
- optional muting of unwanted arrivals,

- organization of traces by source-receiver offset.

The goal of preprocessing is not to change the physics of the wavefield, but to improve the visibility of coherent surface-wave energy. Since surface-wave methods depend strongly on phase information, amplitude normalization must be applied carefully.

Dispersion Imaging

The first imaging method will be a frequency-wavenumber transform. This method is conceptually direct and computationally feasible because it can be implemented using two-dimensional Fourier transforms. It also provides a useful way to visualize the relationship between frequency, wavenumber, and phase velocity.

The second possible method will be phase-shift imaging. For each frequency and trial phase velocity, the framework will apply phase corrections across traces and sum the result. Peaks in the summed energy indicate likely phase velocities. This approach is closely aligned with MASW dispersion imaging and is appropriate for testing literature (Park, Miller, and Xia 1377).

Inversion Strategy

The inversion stage will begin with a simplified layered-earth model. A feasible undergraduate implementation may use one of three approaches:

1. Compare observed dispersion curves against a grid of forward-modeled trial models.
2. Use a local least-squares method for a small number of layers.
3. Use a stochastic global method, such as simulated annealing or differential evolution.

The initial inversion will likely constrain density and V_P, while focusing on V_S and layer thickness. This is justified because Rayleigh-wave phase velocity is most sensitive to V_S in many near-surface MASW applications (Xia, Miller, and Park 691). The inversion module will report misfit values and, if possible, multiple acceptable models. This is important because non-uniqueness is a known limitation of surface-wave inversion (Foti et al., “Non-Uniqueness” 982).

Computational Implementation

The software framework will be written in Python. The proposed libraries include:

1. NumPy for arrays and Fourier transforms,
2. SciPy for optimization,
3. ObsPy for seismic data handling,
4. Matplotlib for plotting,
5. PyQt or PySide for a graphical interface.

The computational feasibility of the project depends on separating I/O-bound and CPU-bound tasks. Threading may be used to keep the interface responsive, but multiprocessing is more appropriate for CPU-heavy work such as repeated forward modeling or dispersion-image generation. Python's official documentation states that multiprocessing can bypass the Global Interpreter Lock by using subprocesses, allowing better use of multiple processors (Python Software Foundation, "multiprocessing").

Software Architecture

The proposed architecture will be modular:

1. Io for file import and metadata handling,
2. Preprocess for filtering and normalization,
3. Imaging for fk and phase-shift transforms,
4. Picking for manual or automated dispersion picking,
5. Inversion for model fitting,
6. Visualization for plots and figures,
7. Benchmarking for runtime and accuracy tests.

This architecture supports the PHY 486 timeline because each module can be developed and tested independently.

Feasibility Analysis

The project is feasible because the required physical theory, algorithms, and computational libraries are accessible at the undergraduate level. The physics is based on elastic-wave propagation, Fourier analysis, and numerical optimization, all of which are appropriate topics for a physics research project. The software tools are also available as open-source Python libraries.

This scope is realistic for an undergraduate research sequence. A full-featured commercial-quality program is not feasible within the available time, but a scientifically meaningful prototype is feasible.

The PHY 386 tasks are:

1. Complete literature review on MASW, dispersion imaging, and inversion.
2. Define the minimum viable computational workflow.
3. Identify available data sources.
4. Install and test Python libraries.
5. Write basic pseudocode for each software module.
6. Produce one preliminary plot from synthetic or example data.

7. Prepare the feasibility paper and oral presentation.

The proposed PHY 486 tasks are:

1. Implement data import and preprocessing.
2. Implement fk or phase-shift dispersion imaging.
3. Implement manual or semi-automated dispersion picking.
4. Implement a simple inversion or model-comparison routine.
5. Benchmark serial and parallel performance.
6. Compare results with literature examples or existing software where possible.
7. Prepare the final report in the required undergraduate research format.
8. Present the final project to the Advisory Committee.

Expected Results

The expected result of PHY 386 is a well-supported feasibility study showing that the proposed computational framework is scientifically justified and computationally possible. The expected result of PHY 486 is a working prototype rather than a complete professional software package.

A successful prototype should produce dispersion images that show clear energy trends for data. It should also demonstrate that a measured or selected dispersion curve can be connected to a plausible layered V_S model. If multiprocessing is successfully implemented, the project should show measurable runtime improvement for repeated imaging or inversion tasks, although the amount of speedup will depend on data size, processor count, and memory overhead.

The project is also expected to show limitations. It should find that inversion results are not unique and that automated picking is difficult when dispersion images are noisy or multimodal.

Conclusion

The proposed computational framework is a feasible PHY 386/486 project when framed as a research tool rather than a finished commercial software package. The project has a clear physical foundation in elastic-wave propagation and Rayleigh-wave dispersion. It has a clear computational foundation in Fourier analysis, numerical optimization, and parallel processing. It also has a realistic path from PHY 386 feasibility study to PHY 486 implementation.

The recommended project scope is to develop a modular Python program that imports or generates seismic shot gathers, preprocesses the data, creates dispersion images, extracts a dispersion curve, and tests a simple V_S inversion.

Works Cited

Beyreuther, Moritz, et al. “ObsPy: A Python Toolbox for Seismology.” *Seismological Research Letters*, vol. 81, no. 3, 2010, pp. 530–533.

Foti, Sebastiano, et al. “Application of Surface-Wave Methods for Seismic Site Characterization.” *Surveys in Geophysics*, vol. 32, no. 6, 2011, pp. 777–825.

Foti, Sebastiano, et al. “Non-Uniqueness in Surface-Wave Inversion and Consequences on Seismic Site Response Analyses.” *Soil Dynamics and Earthquake Engineering*, vol. 29, no. 6, 2009, pp. 982–993.

Hagelund, Rune, and Stewart A. Levin, editors. *SEG-Y_r2.0: SEG-Y Revision 2.0 Data Exchange Format*. SEG Technical Standards Committee, Jan. 2017.

Missouri State University Department of Physics, Astronomy, and Materials Science. *PHY 386/486 – Policies and Procedures*. Aug. 2019.

ObsPy Development Team. “obspy.io.seg2 – SEG-2 Read Support for ObsPy.” *ObsPy Documentation*, 2024.

ObsPy Development Team. “obspy.io.segy – SEG Y and SU Read and Write Support for ObsPy.” *ObsPy Documentation*, 2024.

Park, Choon B., Richard D. Miller, and Jianghai Xia. “Imaging Dispersion Curves of Surface Waves on Multi-Channel Record.” *SEG Technical Program Expanded Abstracts 1998*, Society of Exploration Geophysicists, 1998, pp. 1377–1380.

Park, Choon B., Richard D. Miller, and Jianghai Xia. “Multichannel Analysis of Surface Waves.” *Geophysics*, vol. 64, no. 3, 1999, pp. 800–808.

Pullan, S. E. “Recommended Standard for Seismic (/Radar) Data Files in the Personal Computer Environment.” *Geophysics*, vol. 55, no. 9, 1990, pp. 1260–1271.

Python Software Foundation. “multiprocessing — Process-Based Parallelism.” *Python 3 Documentation*, 2024.

Python Software Foundation. “threading — Thread-Based Parallelism.” *Python 3 Documentation*, 2024.

Ren, Li, et al. “Automated Dispersion Curve Picking Using Multi-Attribute Convolutional-Neural-Network Based Machine Learning.” *Geophysical Journal International*, vol. 232, no. 2, 2023, pp. 1173–1191.

Wathelet, Marc, Jean-Luc Chatelain, Cécile Cornou, Giuseppe Di Giulio, Bertrand Guillier, Matthias Ohrnberger, and Alexandros Savvaidis. “Geopsy: A User-Friendly Open-Source Tool Set for Ambient Vibration Processing.” *Seismological Research Letters*, vol. 91, no. 3, 2020, pp. 1878–1889.

Wathelet, Marc, Dominique Jongmans, and Matthias Ohrnberger. “Surface-Wave Inversion Using a Direct Search Algorithm and Its Application to Ambient Vibration Measurements.” *Near Surface Geophysics*, vol. 2, no. 4, 2004, pp. 211–221.

Xia, Jianghai, Richard D. Miller, and Choon B. Park. “Estimation of Near-Surface Shear-Wave Velocity by Inversion of Rayleigh Waves.” *Geophysics*, vol. 64, no. 3, 1999, pp. 691–700.