Deep Learning-Based Inversion of Surface Wave Effective Dispersion Curves

WaveDisp

Ali Vaziri

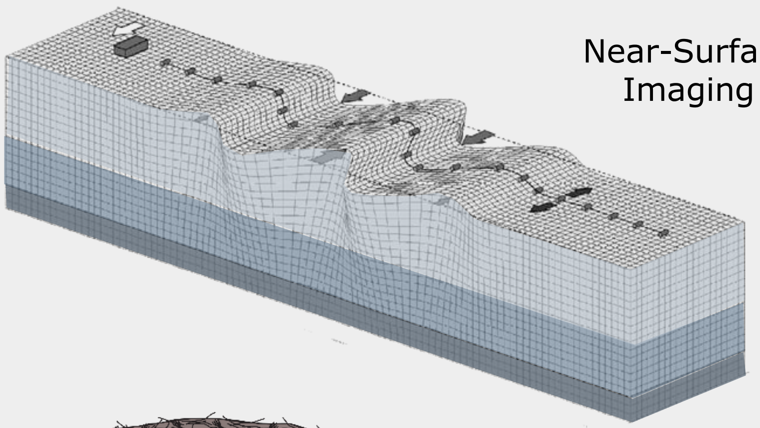
https://github.com/github-ava/WaveDisp

Outline

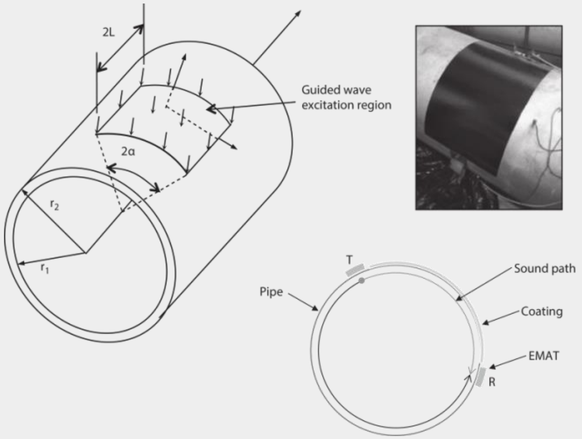
- Motivation
- Guided Wave Propagation
- Forward Modeling
- Inversion
- Applications
- Conclusions

Motivation

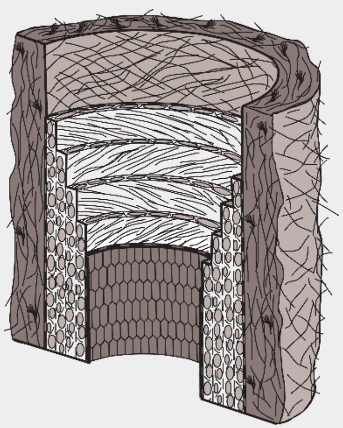
- Characterization of Multilayered Waveguides



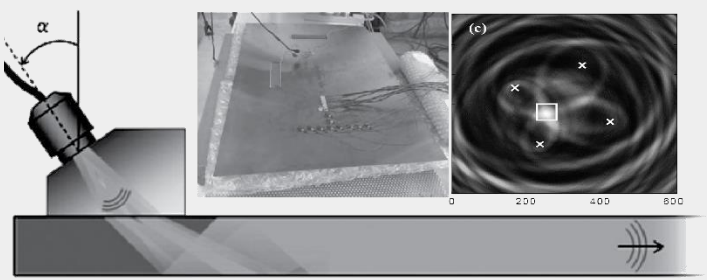
Near-Surface Imaging



Nondestructive Testing



Ultrasound Elastography

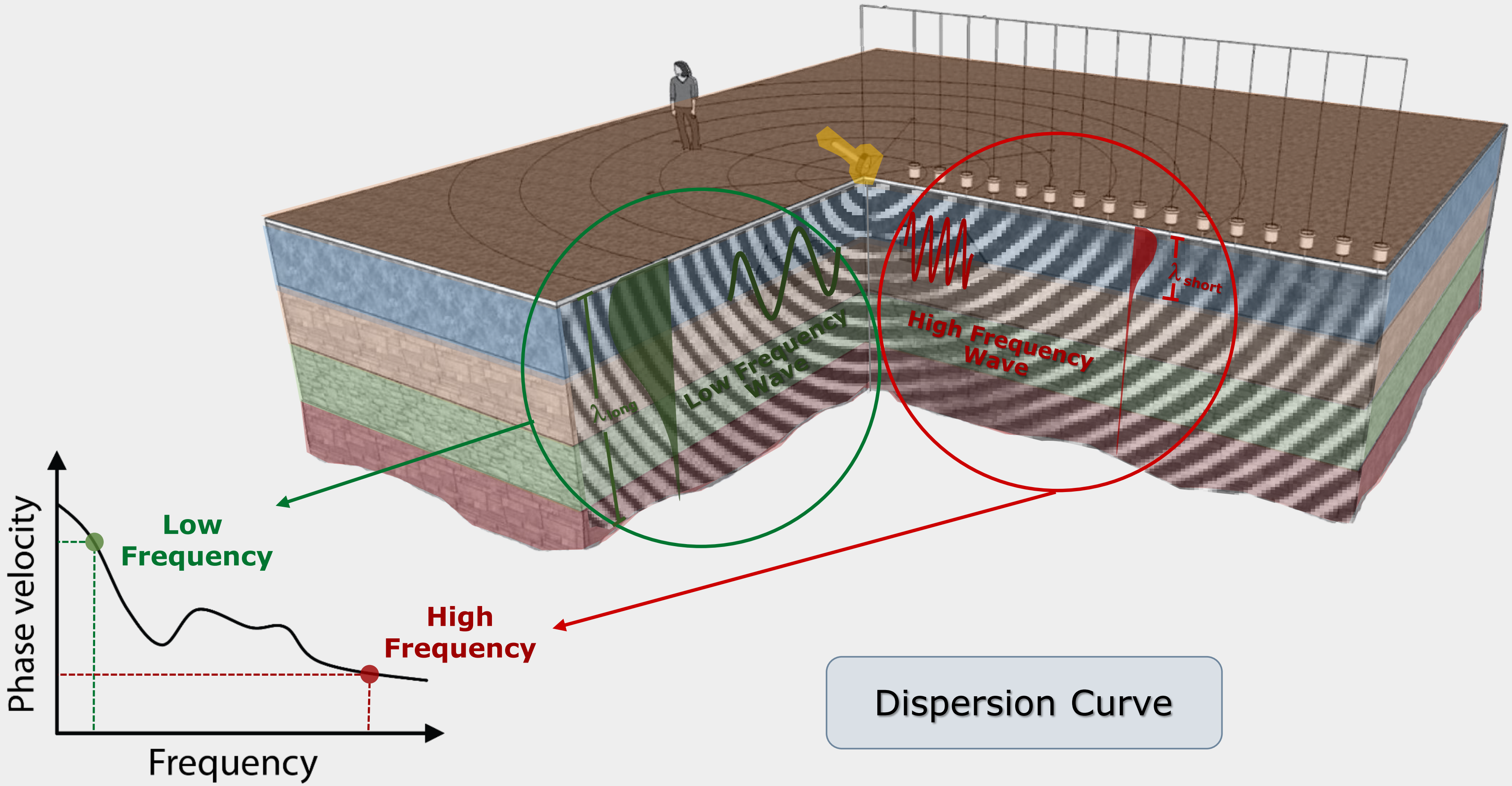


Top left image: Foti S *et al.*, CRC Press, 2014.
Top right image: Rose JL, Cambridge University Press, 2004.

Outline

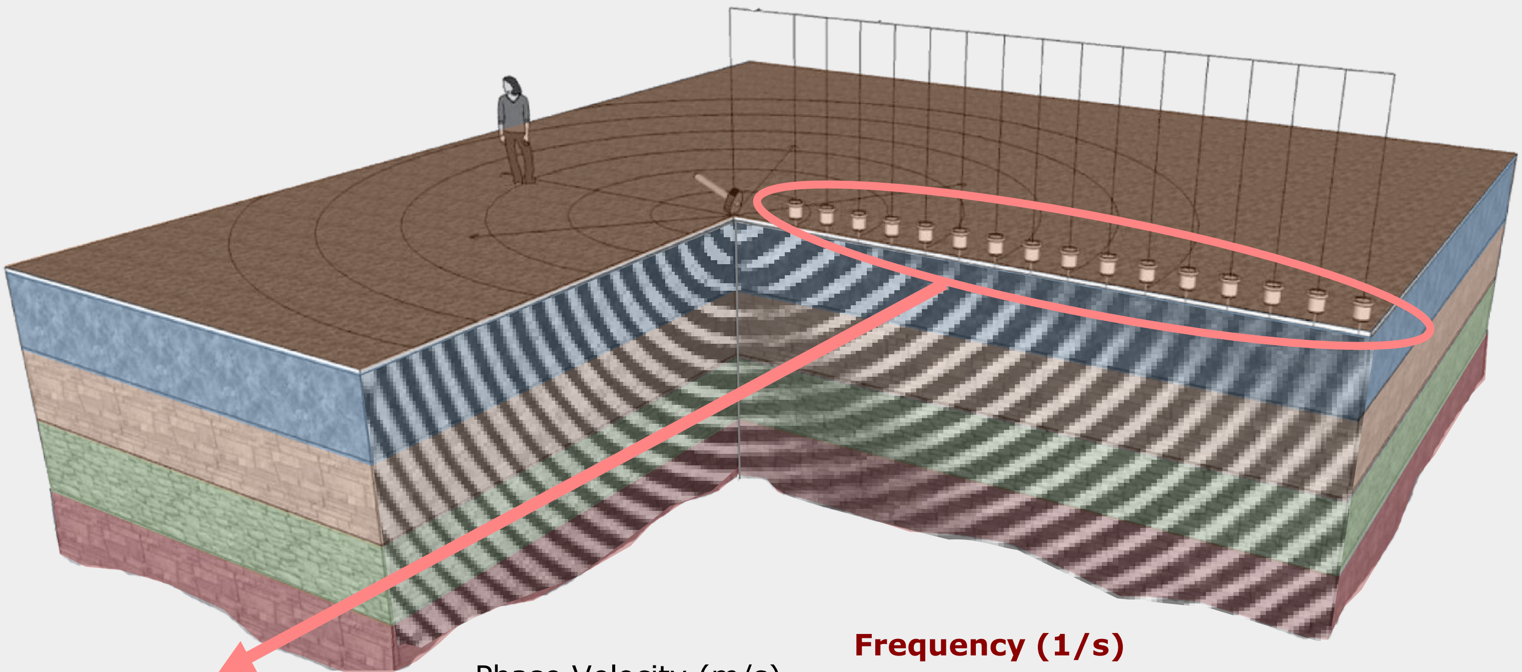
- Motivation
- Guided Wave Propagation
- Forward Modeling
- Inversion
- Applications
- Conclusions

Guided Wave Dispersion



Dispersion Curve

Spectral Analysis



$$\text{Phase Velocity (m/s)} = \frac{\textbf{Frequency (1/s)}}{\textbf{Wavenumber (1/m)}}$$

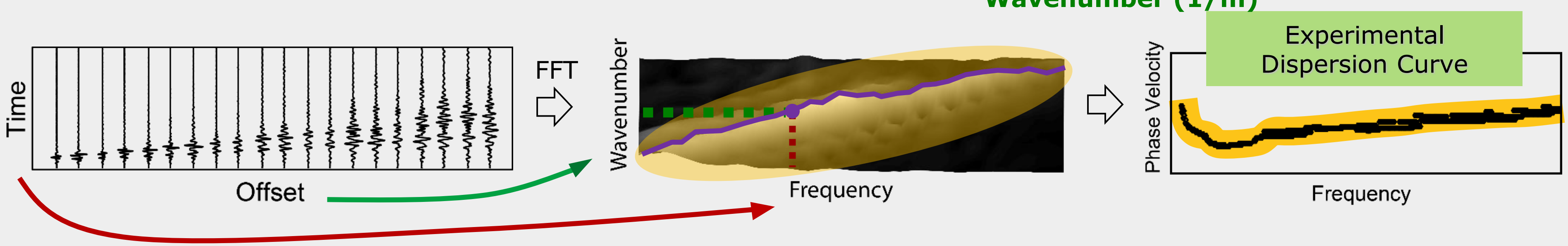
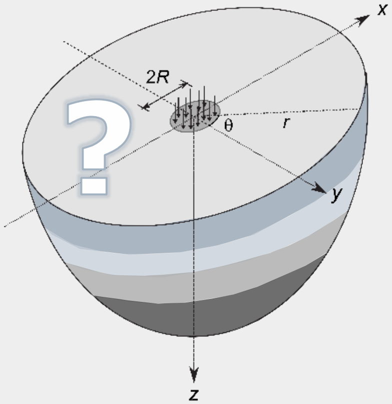
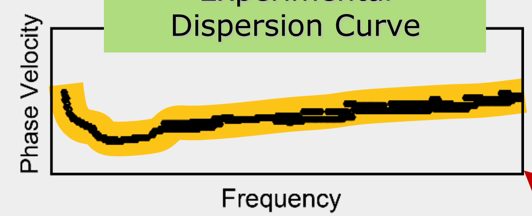


Image: Foti S. et al. Surface Wave Methods for Near-Surface Site Characterization, CRC Press (2014)

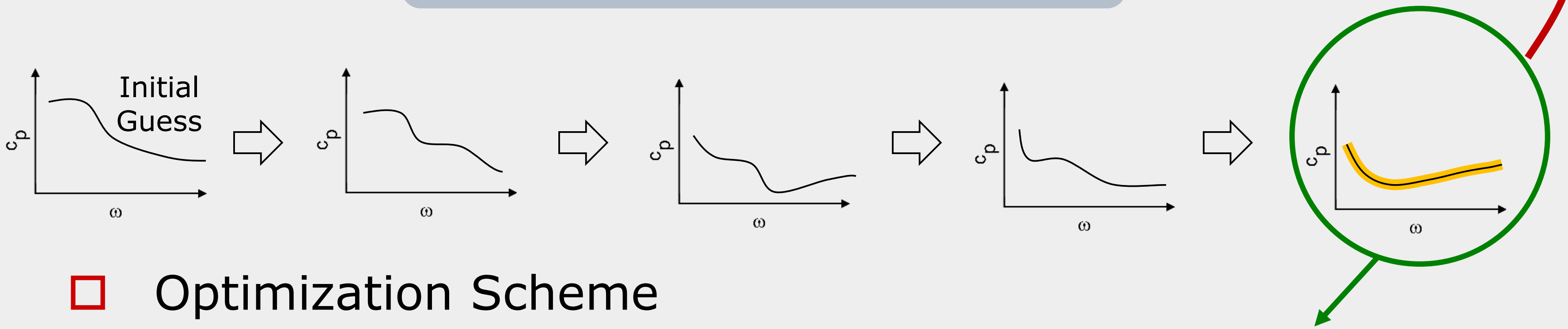
Medium Characterization



Inverse Identification



Iteratively Minimize $E = \sum_{i=1}^{N} (c_i^{\text{experimantal}} - c_i^{\text{predicted}})^2$



Forward Problem:
Predicted Dispersion Curve

☐ Optimization Scheme

- Gradient Based, e.g. Newton-like Methods
- Global Search, e.g. Genetic Algorithm
- Deep Learning

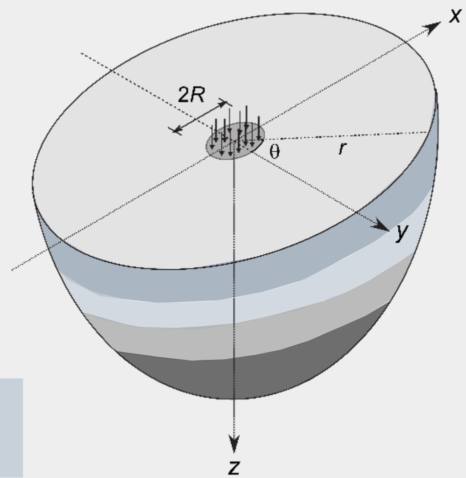
Outline

- Motivation
- Guided Wave Propagation
- Forward Modeling
- Inversion
- Applications
- Conclusions

Forward Modeling – State of the Art

$$-\frac{1}{r}\frac{\partial}{\partial r}\left(\mathbf{D}_{rr}\, r\frac{\partial \mathbf{u}}{\partial r}+\mathbf{D}_{rz}\, r\frac{\partial\mathbf{u}}{\partial z}+\mathbf{D}_{ro}\mathbf{u}\right)-\frac{\partial}{\partial z}\left(\mathbf{D}^T_{rz}\frac{\partial \mathbf{u}}{\partial r}+\mathbf{D}_{zz}\frac{\partial\mathbf{u}}{\partial z}+\mathbf{D}_{zo}\frac{1}{r}\mathbf{u}\right)$$

$$-\frac{1}{r}\left(-\mathbf{D}^T_{ro}\frac{\partial\mathbf{u}}{\partial r}-\mathbf{D}^T_{zo}\frac{\partial \mathbf{u}}{\partial z}-\frac{1}{r}\mathbf{D}_{oo}\mathbf{u}\right)-\left(\rho\omega^2\mathbf{I}\right)\mathbf{u}=\mathbf{0}$$

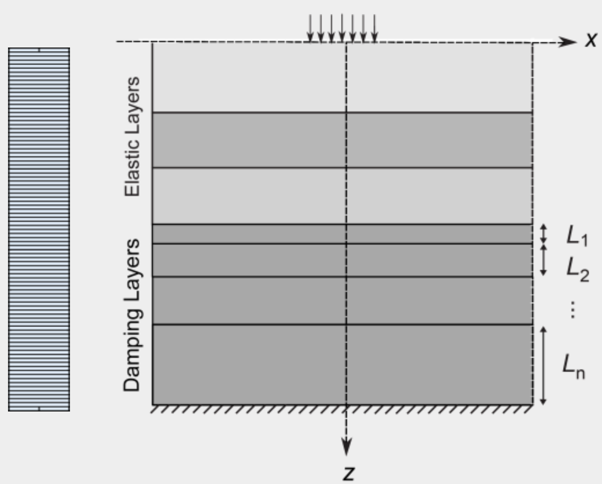


Discretize
z direction

Hankel Transform
r direction

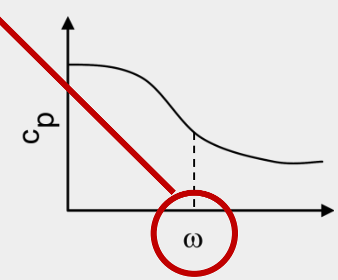
⇨

Quadratic
Eigenvalue Problem



$$\overbrace{\left(k^2\,\mathbf{A}+k\;\mathbf{B}+\mathbf{C}-\omega^2\;\mathbf{D}\right)}^{\mathbf{K}}\mathbf{u}=\mathbf{0}$$

$$k \longrightarrow c_p=\frac{\omega}{k}$$



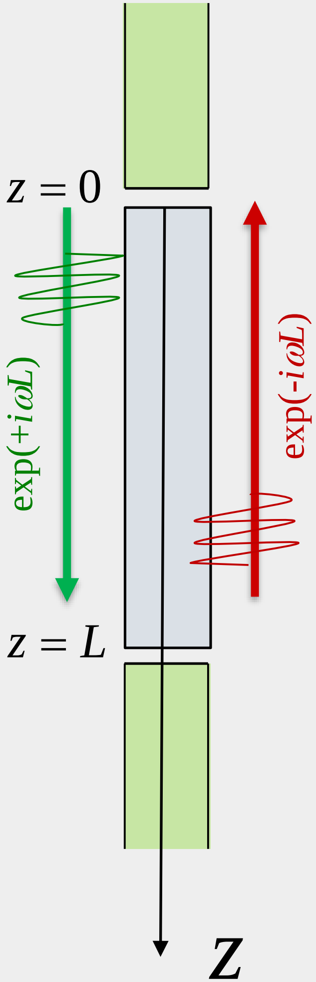
Challenge

- Eigenvalue Problem Complexity: $\mathcal{O}(n^3)$
- Conventional FEM
  - Algebraic convergence but sparse system
- Spectral FEM
  - Exponential convergence but dense system

Exponential convergence + Sparse system

- Complex-Length FEM (CFEM)
  - **Exponential convergence with linear elements**

1D Helmholtz Equation



$$-\frac{\partial^2 u}{\partial z^2} - \omega^2 u = 0$$

$$\begin{Bmatrix} -\partial u_0 / \partial x \\ \partial u_L / \partial x \end{Bmatrix} = \overbrace{i\omega \begin{bmatrix} \coth(i\omega L) & -\operatorname{csch}(i\omega L) \\ -\operatorname{csch}(i\omega L) & \coth(i\omega L) \end{bmatrix}}^{\mathbf{K}^{\text{exact}}} \begin{Bmatrix} u_0 \\ u_L \end{Bmatrix}$$

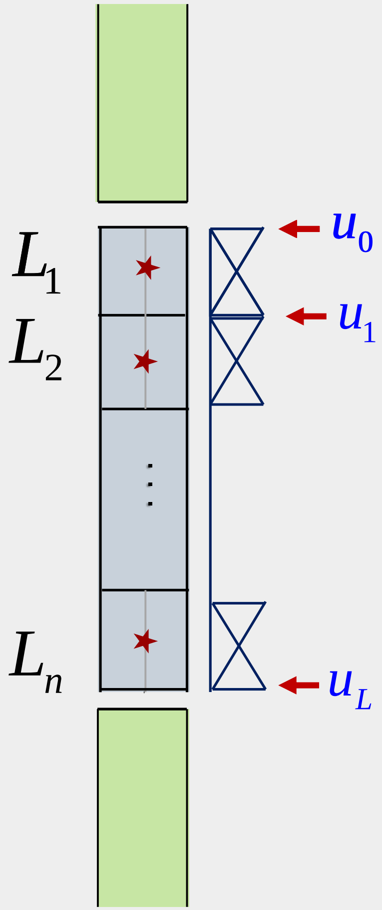
1st Order Form

$$v = \partial u / \partial z$$

$$\frac{\partial}{\partial z} \begin{Bmatrix} u \\ v / i\omega \end{Bmatrix} = \begin{bmatrix} 0 & i\omega \\ i\omega & 0 \end{bmatrix} \begin{Bmatrix} i\omega u \\ v / i\omega \end{Bmatrix}$$

$$\begin{Bmatrix} u_L \\ \partial u_L / \partial z \end{Bmatrix} = \overbrace{\begin{bmatrix} \exp(i\omega L) & 0 \\ 0 & \exp(i\omega L) \end{bmatrix}}^{\mathbf{P}^{\text{exact}}} \begin{Bmatrix} u_0 \\ \partial u_0 / \partial z \end{Bmatrix}$$

Propagator Approximation



$$\begin{Bmatrix} u_1 \\ \partial u_1/\partial z \end{Bmatrix} = \overbrace{\begin{bmatrix} \left(\dfrac{1+i\omega L_1/2}{1-i\omega L_1/2}\right) & 0 \\ 0 & \left(\dfrac{1+i\omega L_1/2}{1-i\omega L_1/2}\right) \end{bmatrix}}^{\mathbf{P}_1^{\text{approximate}}} \begin{Bmatrix} u_0 \\ \partial u_0/\partial z \end{Bmatrix}$$

$$\frac{1}{L_1}\begin{Bmatrix} u_1 - u_0 \\ \bar{v}_1 - \bar{v}_0 \end{Bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & i\omega \\ i\omega & 0 \end{bmatrix}\begin{Bmatrix} u_1 + u_0 \\ \bar{v}_1 + \bar{v}_0 \end{Bmatrix}$$

Crank-Nicolson

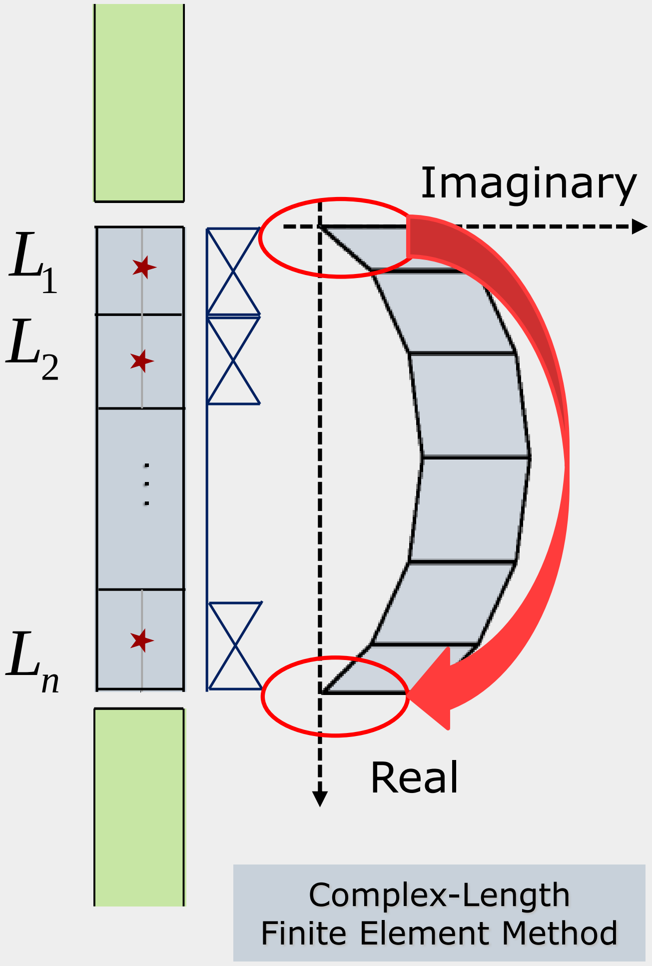
1st Order Form

$$\frac{\partial}{\partial z}\begin{Bmatrix} u \\ v/i\omega \end{Bmatrix} = \begin{bmatrix} 0 & i\omega \\ i\omega & 0 \end{bmatrix}\begin{Bmatrix} i\omega u \\ v/i\omega \end{Bmatrix}$$

Crank-Nicolson

$$\begin{Bmatrix} u_L \\ \partial u_L/\partial z \end{Bmatrix} = \overbrace{\begin{bmatrix} \exp(i\omega L) & 0 \\ 0 & \exp(i\omega L) \end{bmatrix}}^{\mathbf{P}^{\text{exact}}} \begin{Bmatrix} u_0 \\ \partial u_0/\partial z \end{Bmatrix}$$

Padé Approximation



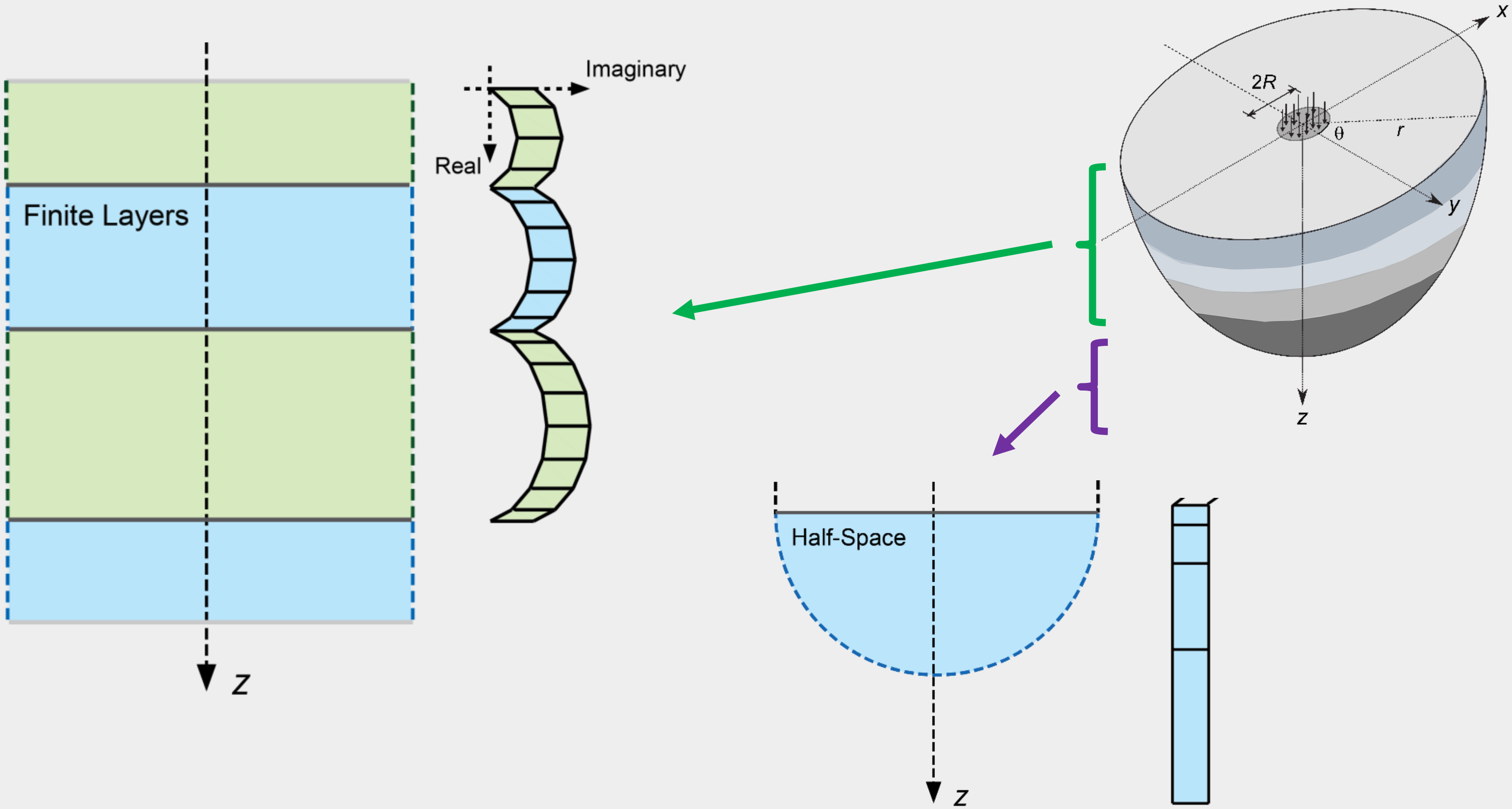
$$\exp(\alpha L) \approx \overbrace{\prod_{j=1}^{n}\left(\frac{1+\alpha L_j/2}{1-\alpha L_j/2}\right)}^{P_{\text{Padé}}}$$

$$\left.\frac{d^j\left(\exp(\alpha L)\right)}{d\alpha^j}\right|_{\alpha=0} = \left.\frac{d^j P_{\text{Padé}}}{d\alpha^j}\right|_{\alpha=0}$$

$$(j = 0, \cdots, 2n)$$

$$\sum_{j=0}^{n}\frac{(2n-j)!}{j!\,(n-j)!}(-x)^j = 0 \quad \rightarrow \quad L_j = 2L/x_j$$

Forward Modeling: Complex-Length FEM

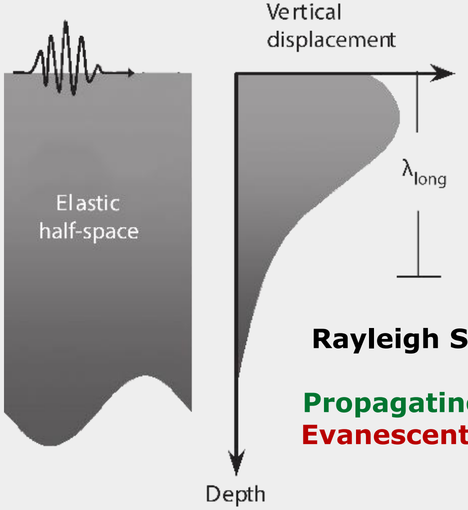


Complex-Length FEM
(Finite Layers)

Perfectly Matched *discrete* Layers
(Infinite Layer)

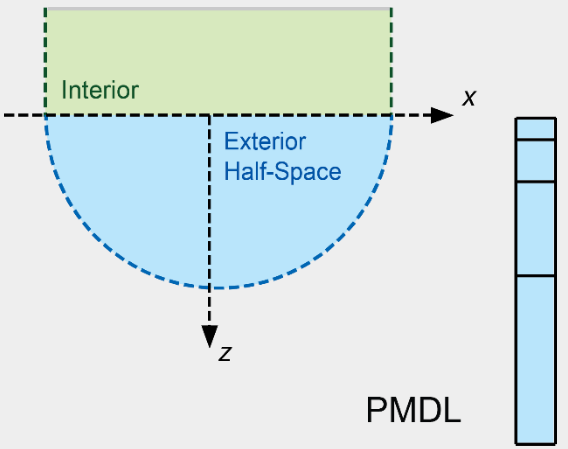
Infinite Half-Space

- Damping Layers for Rayleigh Surface Waves



Rayleigh Surface Waves:

Propagating in x direction
Evanescent in z direction



$$L_n = L_1\ \alpha^{n-1}, \qquad L_1 = \frac{2}{\sqrt{\alpha^{n-1} k_z^{\min} k_z^{\max}}}$$

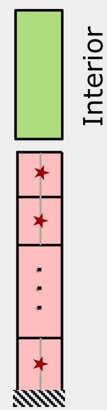
Damping Layers with Real Lengths

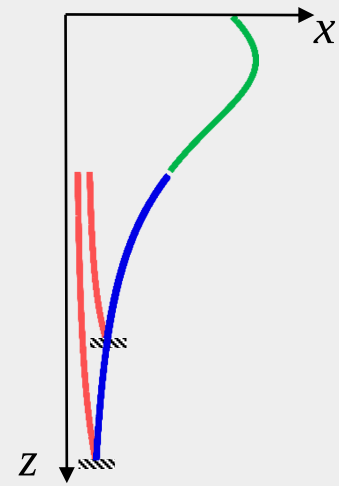
2 parameters: *n*, *α*

Perfectly Matched Discrete Layer (PMDL)†

Truncation Effect

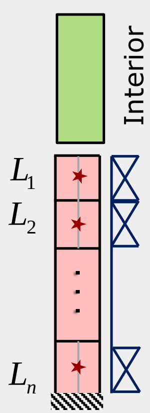
reflections





Discretization Effect

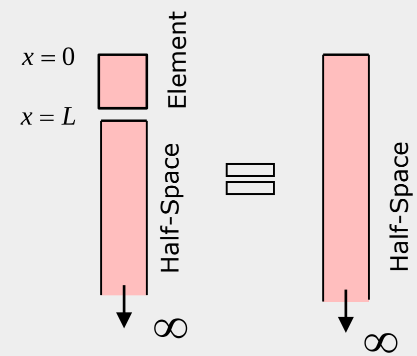
$$\mathbf{K}_j = \begin{bmatrix} \left(\frac{1}{L_j} + \frac{k^2 L_j}{4}\right) & \left(-\frac{1}{L_j} + \frac{k^2 L_j}{4}\right) \\ \left(-\frac{1}{L_j} + \frac{k^2 L_j}{4}\right) & \left(\frac{1}{L_j} + \frac{k^2 L_j}{4}\right) \end{bmatrix}$$



$$\begin{bmatrix} k_{11} & k_{12} \\ k_{12} & k_{11} + K_{\text{Half-Space}} \end{bmatrix} \begin{Bmatrix} u_0 \\ u_L \end{Bmatrix} = \begin{Bmatrix} K_{\text{Half-Space}}\, u_0 \\ 0 \end{Bmatrix}$$

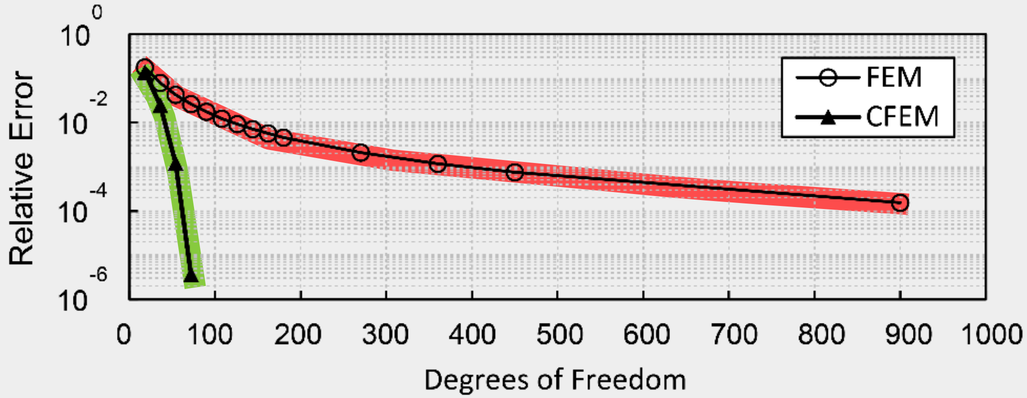
$$K_{\text{Half-Space}} = k$$

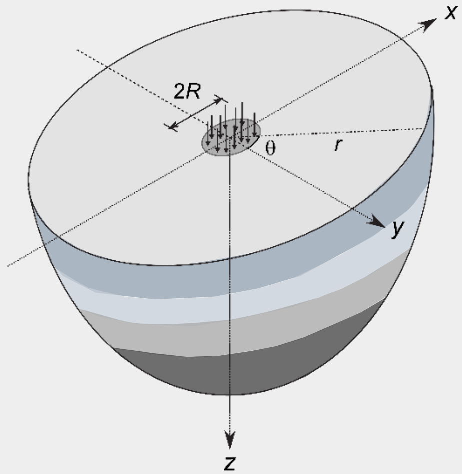
No discretization error



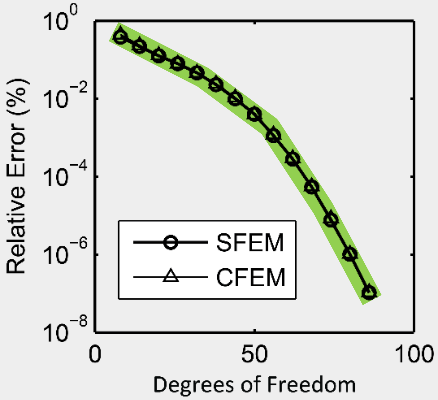
Dispersion Curve Convergence

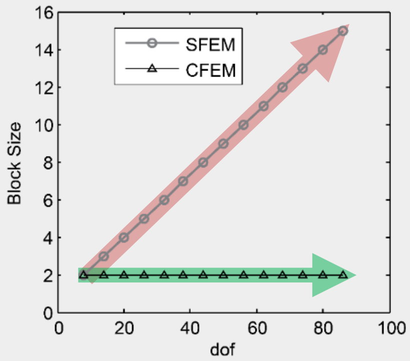
- CFEM vs. FEM (Eigenvalue Problem)





- CFEM vs. SFEM (Eigenvalue Problem)

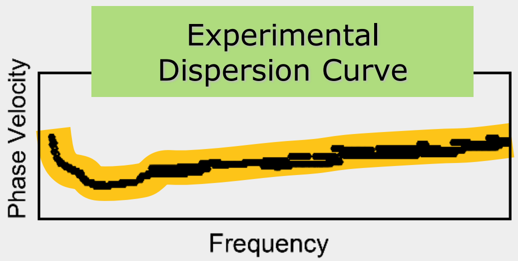




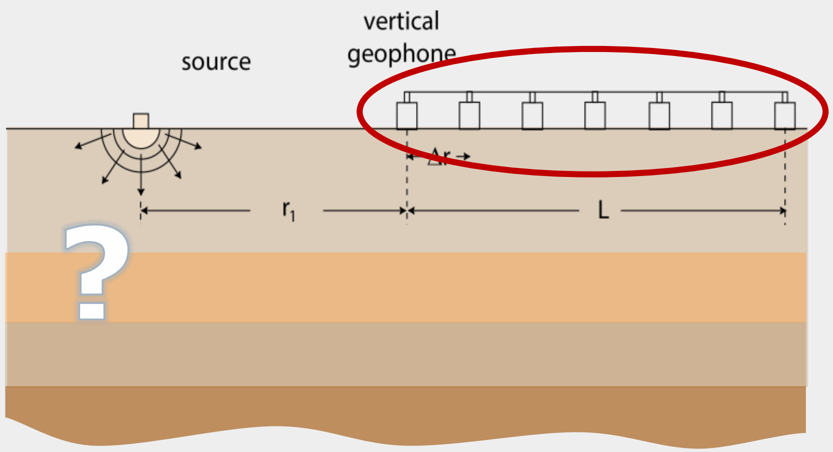
Outline

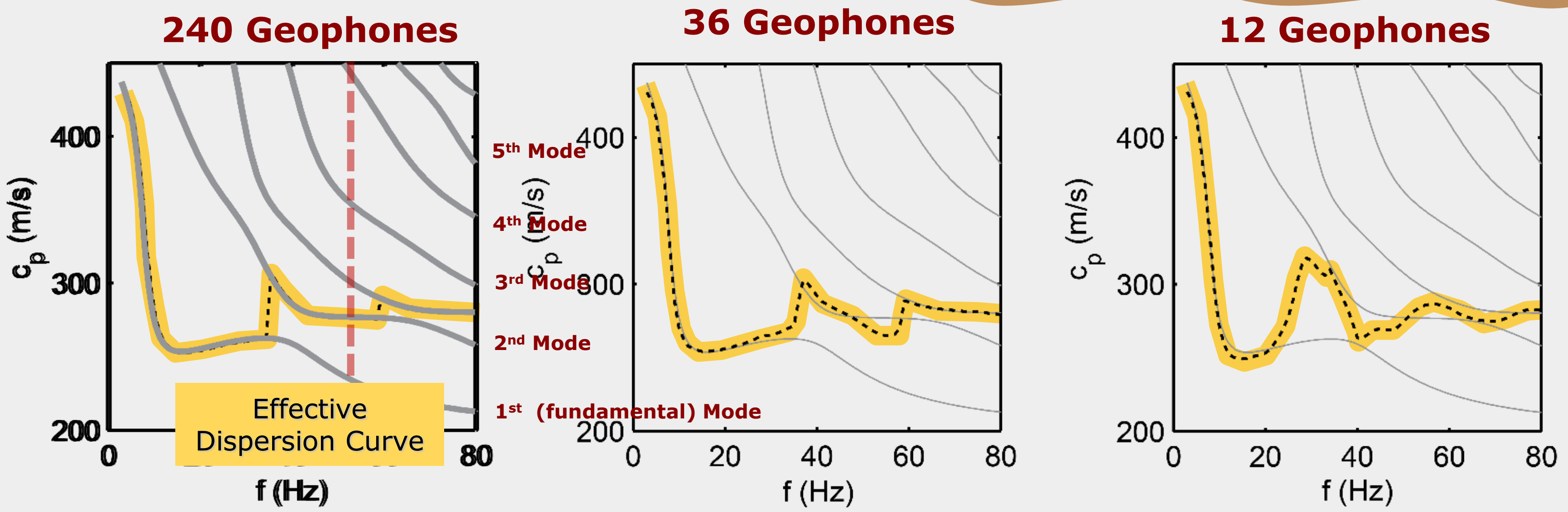
- Motivation
- Guided Wave Propagation
- Forward Modeling
- Inversion
- Applications
- Conclusions

Effective Dispersion Curve



Inverse Identification



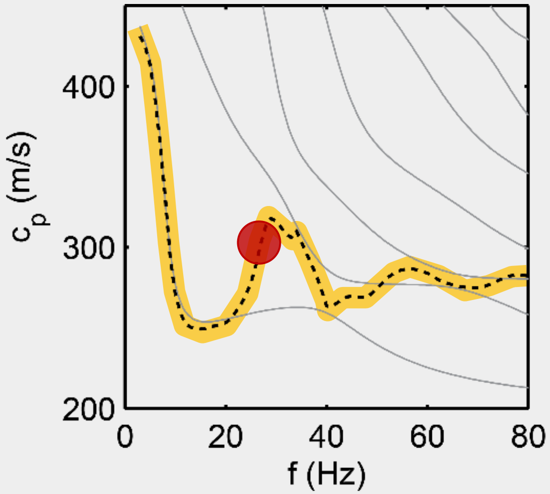


Challenge

- No analytical derivative

- Rough misfit function

$$\text{Minimize} \quad E = \sum_{i=1}^{N} \left(c_i^{\text{experimantal}} - c_i^{\text{predicted}} \right)^2$$



- Existing approach: Finite Difference Method (FDM)
  - Expensive: Multiple computations of dispersion curve
  - Slow convergence: Oscillatory gradient

Deep Learning Inversion

Neural Network Architecture

- Input Layer:
- Shape determined by the length of the input (effective curve-frequency)
- Hidden Layers:
- Dense Layer with Custom Leaky ReLU Activation and Batch Normalization
- Dense Layer with Custom Leaky ReLU Activation and Batch Normalization
- Dense Layer with Custom Leaky ReLU Activation and Batch Normalization
- Dense Layer with Custom Leaky ReLU Activation and Batch Normalization
- Dense Layer with Custom Leaky ReLU Activation and Batch Normalization
- Dense Layer with Linear Activation (Output Layer)

Deep Learning Inversion

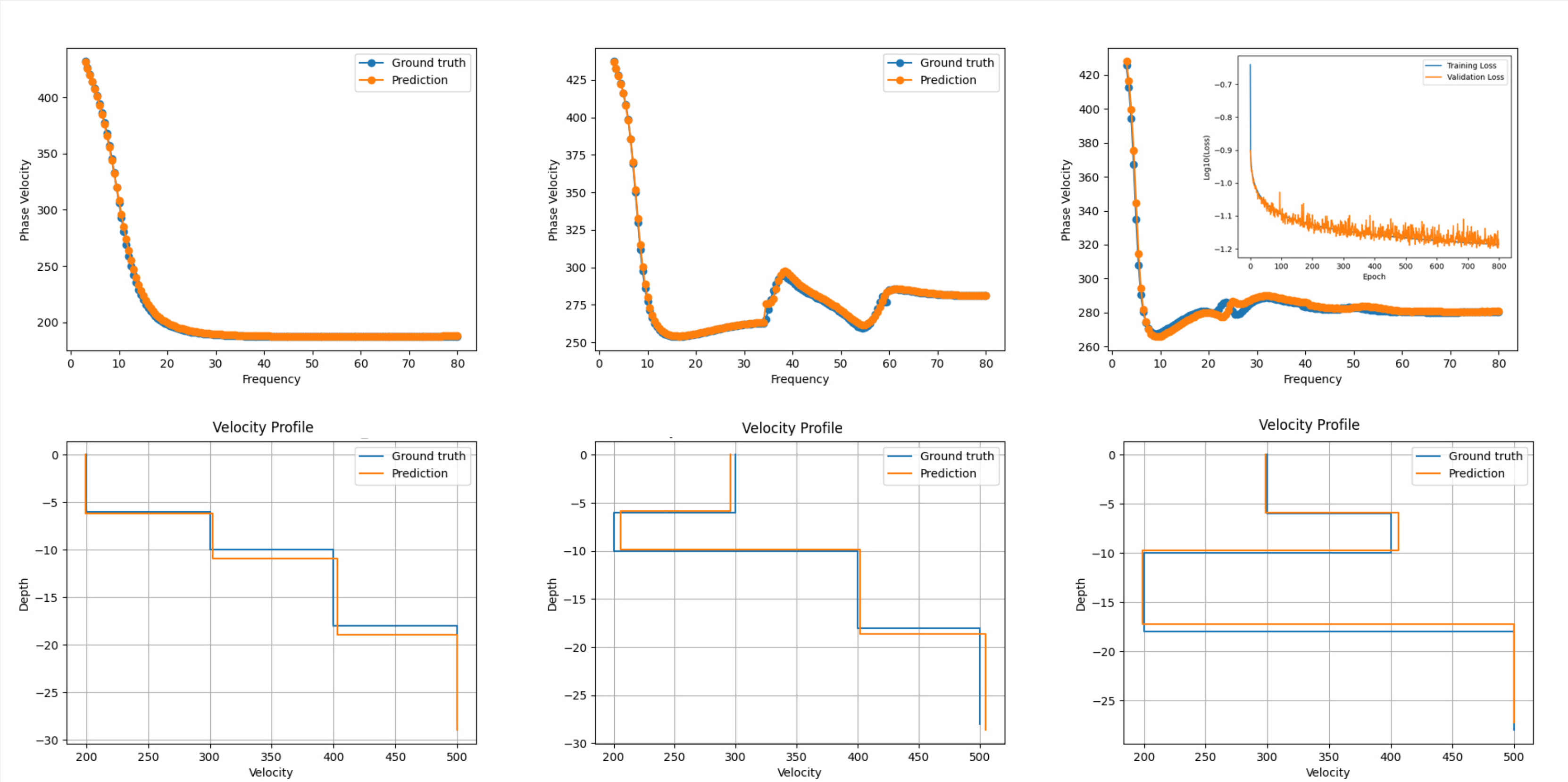
Data Preprocessing

- Input data is normalized using Min-Max scaling.
- Output data is also normalized using Min-Max scaling.
- The data is split into training, validation, and testing sets based on the provided ratios.
- Early stopping and model checkpointing callbacks are used during training for optimization.

Training Process

- The model is compiled using Nadam optimizer and Mean Absolute Error loss function.
- The training process includes early stopping based on validation loss and model checkpointing.
- Training progress is saved periodically, and training time is recorded.
- After training, the model is evaluated on the test set, and the final losses are reported.

Inversion Results: Synthetic Examples

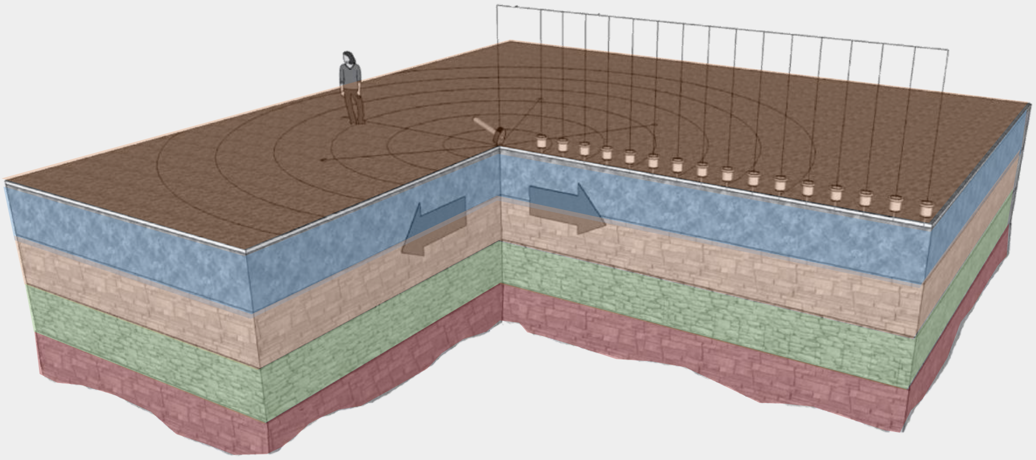


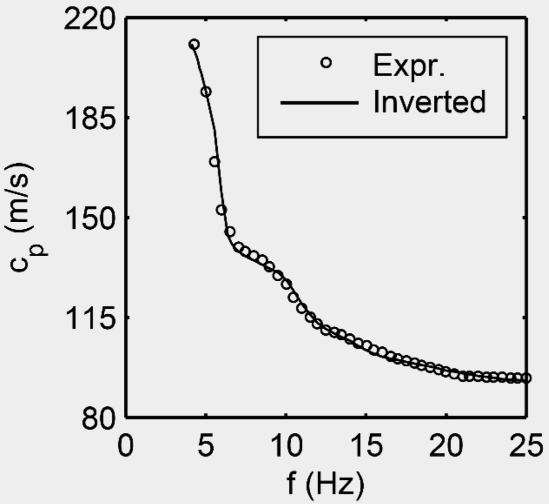
Outline

- Motivation
- Guided Wave Propagation
- Forward Modeling
- Inversion
- Applications
- Conclusions

Application: Near-Surface Imaging†

- Characterizing Multi-Layer Soil Profile





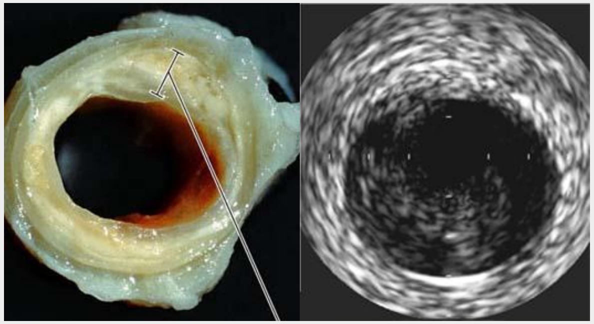
- Parameters
  - Layer Shear Wave Velocity $C_s = \sqrt{G/\rho}$
  - Layer Thickness

Application: Nondestructive Testing

Plates-Like
Waveguides

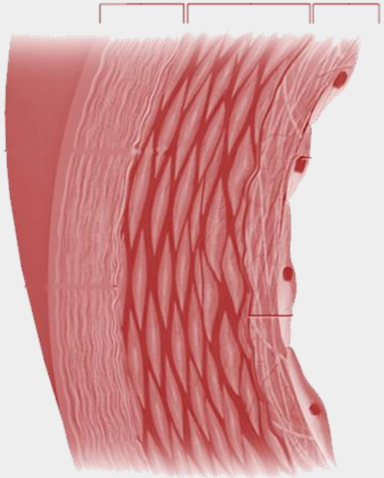
Cylindrical
Waveguides

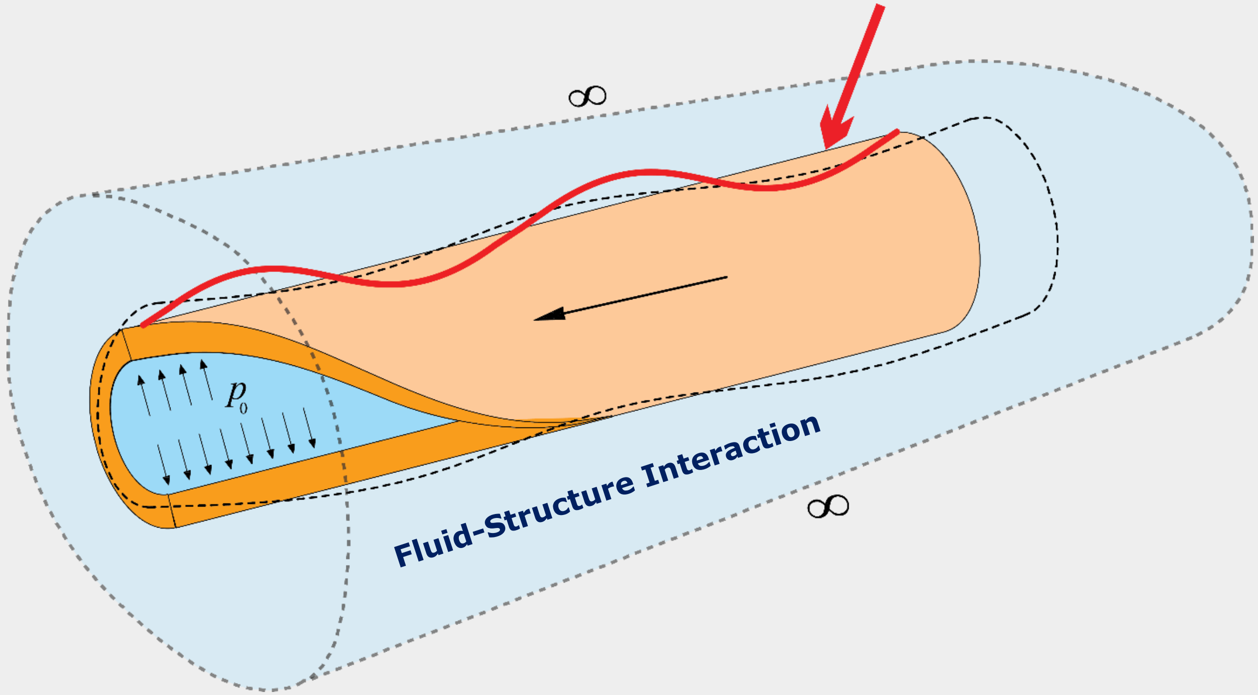
Generic
Cross-Section
Waveguides



Application: Biomedical Imaging

- Arterial Wall Stiffness





Outline

- Motivation
- Guided Wave Propagation
- Forward Modeling
- Inversion
- Applications
- Conclusions

Conclusions

- Forward Modeling
  - CFEM: Exponential convergence of dispersion curves
  - Reduction of eigenproblem size by an order magnitude
  - Minor modifications of existing FEM software
- Inverse Problem
  - Deep learning based inversion of effective curves
- Applications
  - Near-surface geotechnical characterization
  - Nondestructive testing
  - Biomedical imaging