A numerical scheme to solve the Lippmann-Schwinger equation (LSE) for a linearly varying background

Rahul Sarkar[1] & Biondo Biondi[1,2]

[1]Institute for Computational & Mathematical Engineering
Stanford University

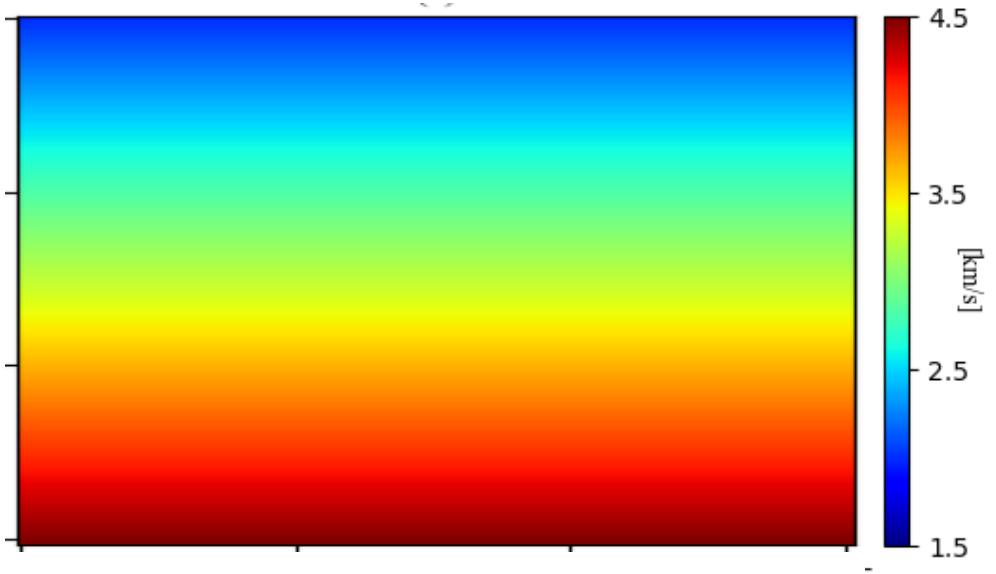
[2]Department of Geophysics
Stanford University

September 29, IMAGE 2021, Denver

Motivation

- To a first approximation, the sound speed profile in the Earth is linearly varying with depth (theoretical seismology often uses this assumption).

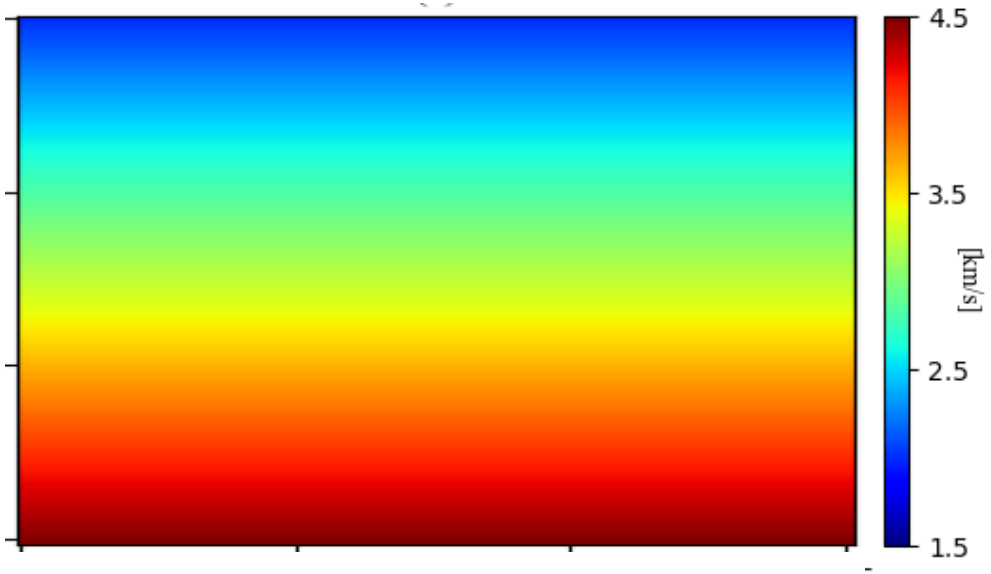
Linearly Varying Sound Speed Profile



Motivation

- To a first approximation, the sound speed profile in the Earth is linearly varying with depth (theoretical seismology often uses this assumption).
- Linearly varying sound speed is used as a ***starting model for FWI at low frequencies***, where there is no apriori information of the velocity.

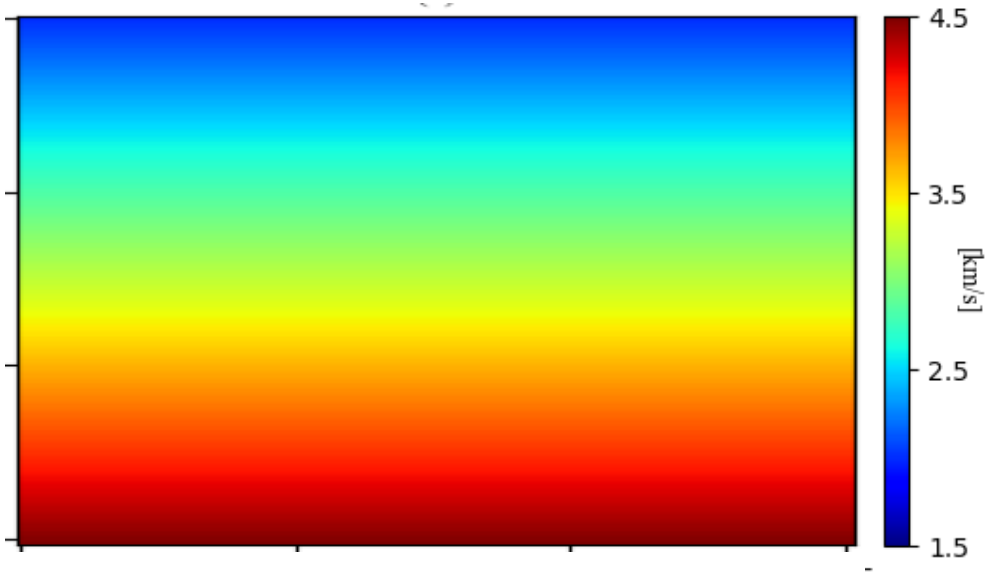
Linearly Varying Sound Speed Profile



Motivation

- To a first approximation, the sound speed profile in the Earth is linearly varying with depth (theoretical seismology often uses this assumption).
- Linearly varying sound speed is used as a ***starting model for FWI at low frequencies***, where there is no apriori information of the velocity.
- Waveform inversion like WRI (wavefield reconstruction inversion) with LSE.

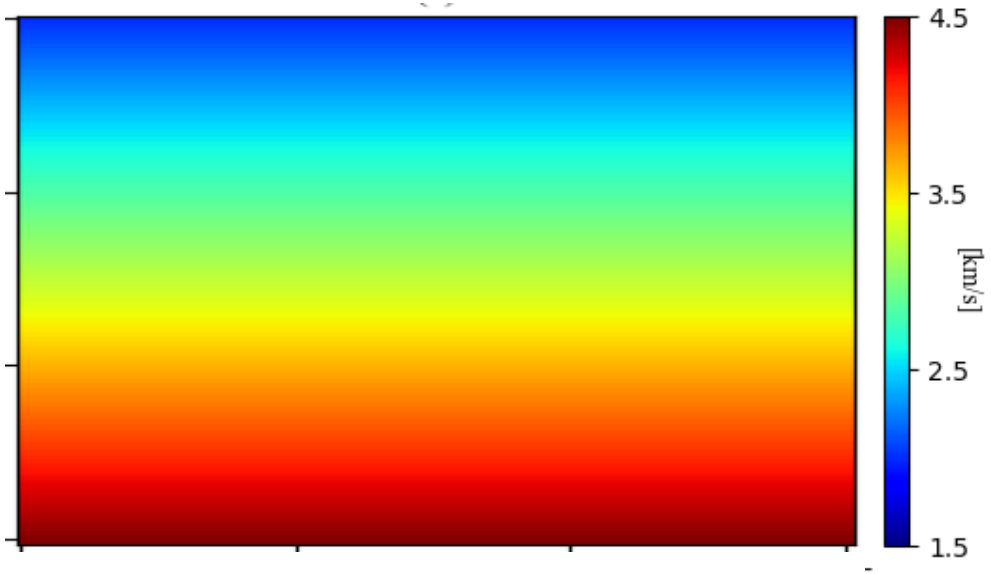
Linearly Varying Sound Speed Profile



Motivation

- To a first approximation, the sound speed profile in the Earth is linearly varying with depth (theoretical seismology often uses this assumption).
- Linearly varying sound speed is used as a ***starting model for FWI at low frequencies***, where there is no apriori information of the velocity.
- Waveform inversion like WRI (wavefield reconstruction inversion) with LSE.
- Previous work on LSE uses the constant background assumption.

Linearly Varying Sound Speed Profile



Use cases of our work

- Low frequency 3D FWI (<5 Hz)
  - FWI with marine resonators like Wolfspar
  - Sources generate data at a few discrete set of frequencies.
  - Our method will have an advantage over time domain methods.

Wolfspar Marine Seismic Source

Source: “Wolfspar®, an “FWI-friendly” ultra-low-frequency marine seismic source”, SEG 2016, Dellinger et.al.

Use cases of our work

- Low frequency 3D FWI (<5 Hz)
  - FWI with marine resonators like Wolfspar
  - Sources generate data at a few discrete set of frequencies.
  - Our method will have an advantage over time domain methods.

- Low frequency time lapse FWI, because velocity changes are small with time.

Wolfspar Marine Seismic Source

Source: “Wolfspar®, an “FWI-friendly” ultra-low-frequency marine seismic source”, SEG 2016, Dellinger et.al.

Use cases of our work

- Low frequency 3D FWI (<5 Hz)
  - FWI with marine resonators like Wolfspar
  - Sources generate data at a few discrete set of frequencies.
  - Our method will have an advantage over time domain methods.
- Low frequency time lapse FWI, because velocity changes are small with time.
- All at once waveform inversion using the LSE as the PDE constraint, instead of the Helmholtz.

Wolfspar Marine Seismic Source

Source: “Wolfspar®, an “FWI-friendly” ultra-low-frequency marine seismic source”, SEG 2016, Dellinger et.al.

Prior work on LSE

- In seismic imaging, LSE intimately connected with forward and inverse scattering problems.

Prior work:

- How to overcome slow convergence (even divergence) of the associated Born scattering series, using ***renormalization*** and ***partial summation techniques***.
  - De-Wolfe series (Wu and Zheng, 2014; Jakobsen and Wu, 2015; Huang et al., 2019)
  - Volterra renormalization (Lesage et al., 2014; Yao et al., 2015)
  - Homotopy analysis method (Jakobsen et al., 2020)

Prior work on LSE

- In seismic imaging, LSE intimately connected with forward and inverse scattering problems.

Prior work:

- How to overcome slow convergence (even divergence) of the associated Born scattering series, using ***renormalization*** and ***partial summation techniques***.
  - De-Wolfe series (Wu and Zheng, 2014; Jakobsen and Wu, 2015; Huang et al., 2019)
  - Volterra renormalization (Lesage et al., 2014; Yao et al., 2015)
  - Homotopy analysis method (Jakobsen et al., 2020)

- Inverse scattering (Weglein et al., 2001; Zhang and Weglein, 2009; Zou et al., 2019)
  - Imaging and inversion using inverse scattering series
  - Multiple removal

Our contributions

- We developed a numerical scheme to solve the LSE for a linearly varying sound speed profile based on the truncated kernel method.

In this talk

- Briefly introduce the problem setup, and recap the LSE
- Brief description of the truncated kernel method (originally developed for the constant background LSE by Vico et al.) and how we adapt it to our problem.
- Complexity analysis of operator application cost
- Some numerical examples
- Extending the method to general v(z) case

Hierarchy of difficulty

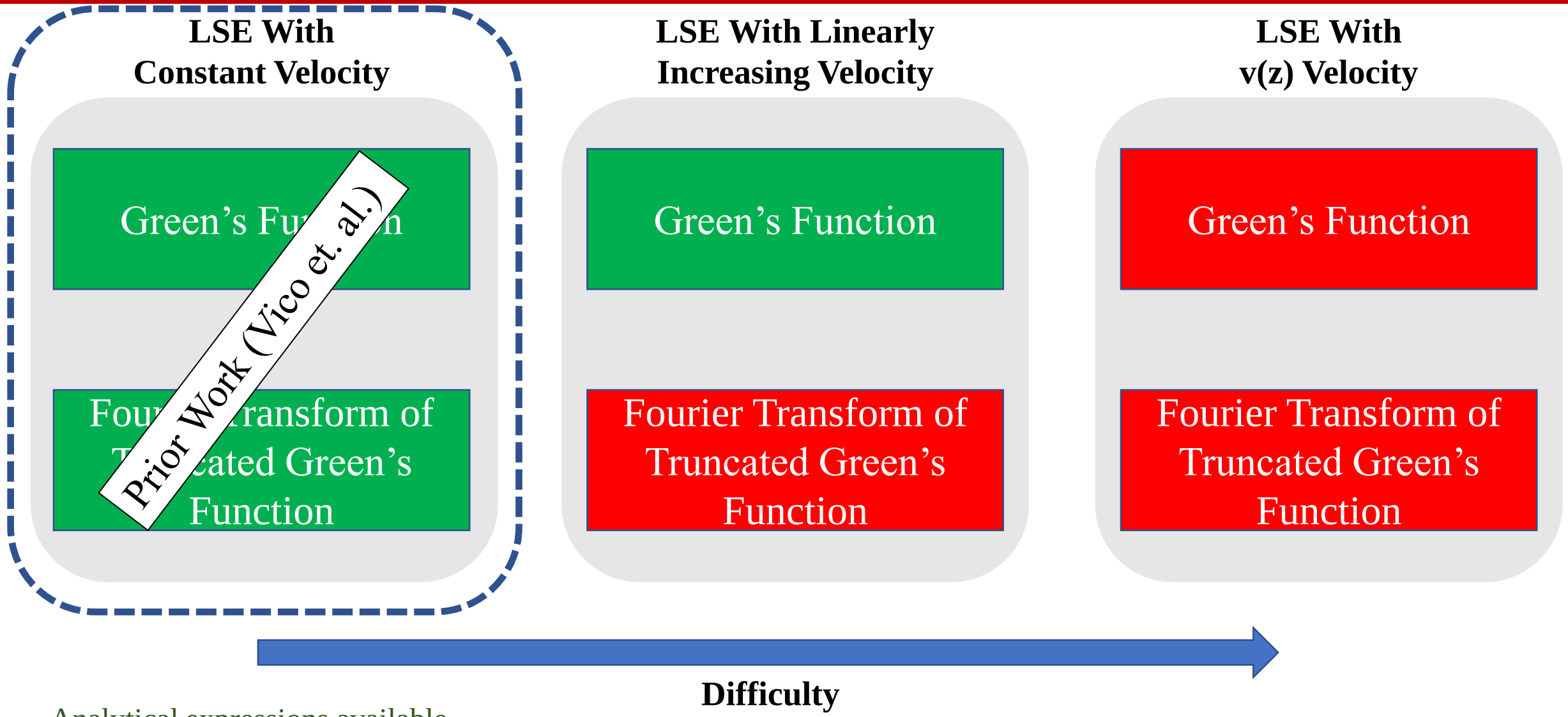
LSE With Constant Velocity	LSE With Linearly Increasing Velocity	LSE With v(z) Velocity
Green's Function	Green's Function	Green's Function
Fourier Transform of Truncated Green's Function	Fourier Transform of Truncated Green's Function	Fourier Transform of Truncated Green's Function

Difficulty

Analytical expressions available.

Analytical expressions not available. Needs to be computed.

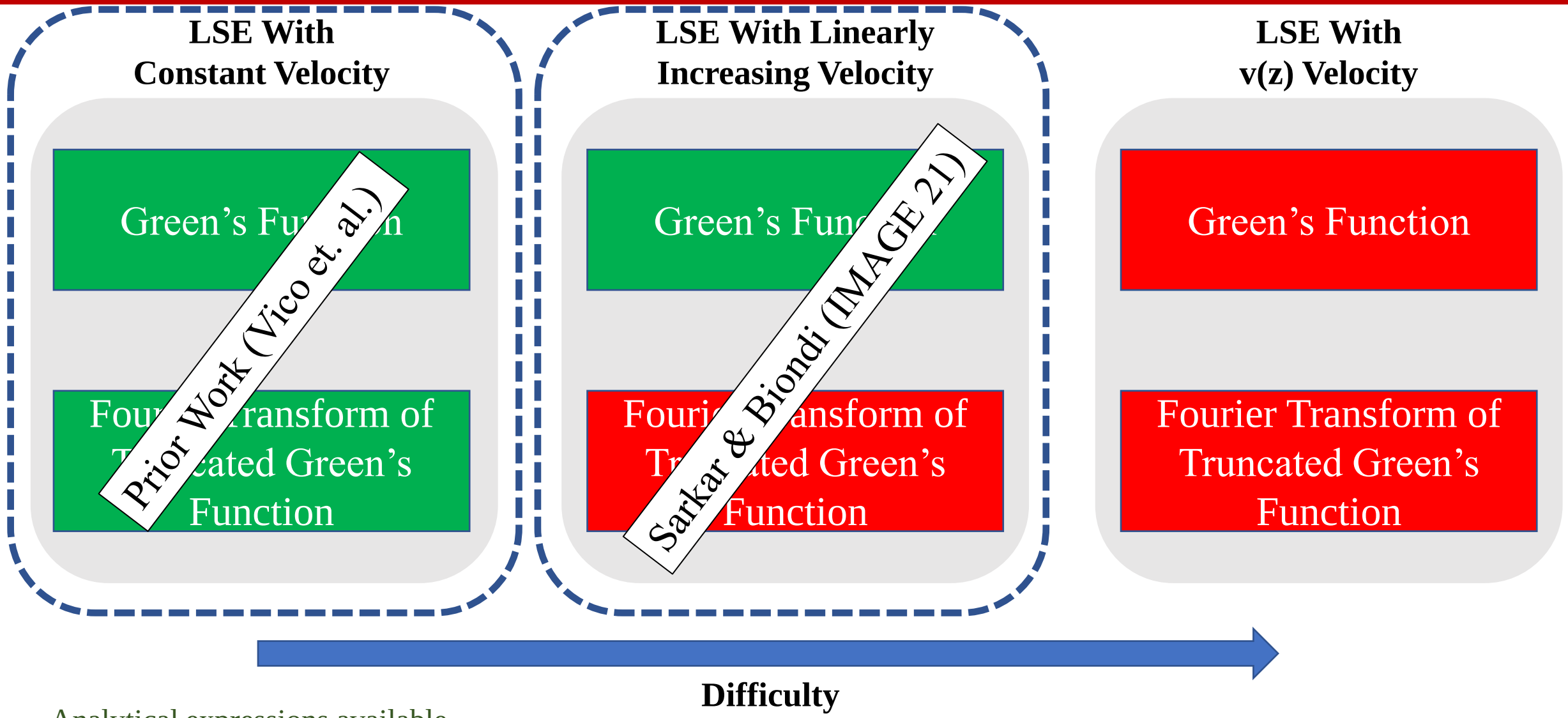
Hierarchy of difficulty



Analytical expressions available.

Analytical expressions not available. Needs to be computed.

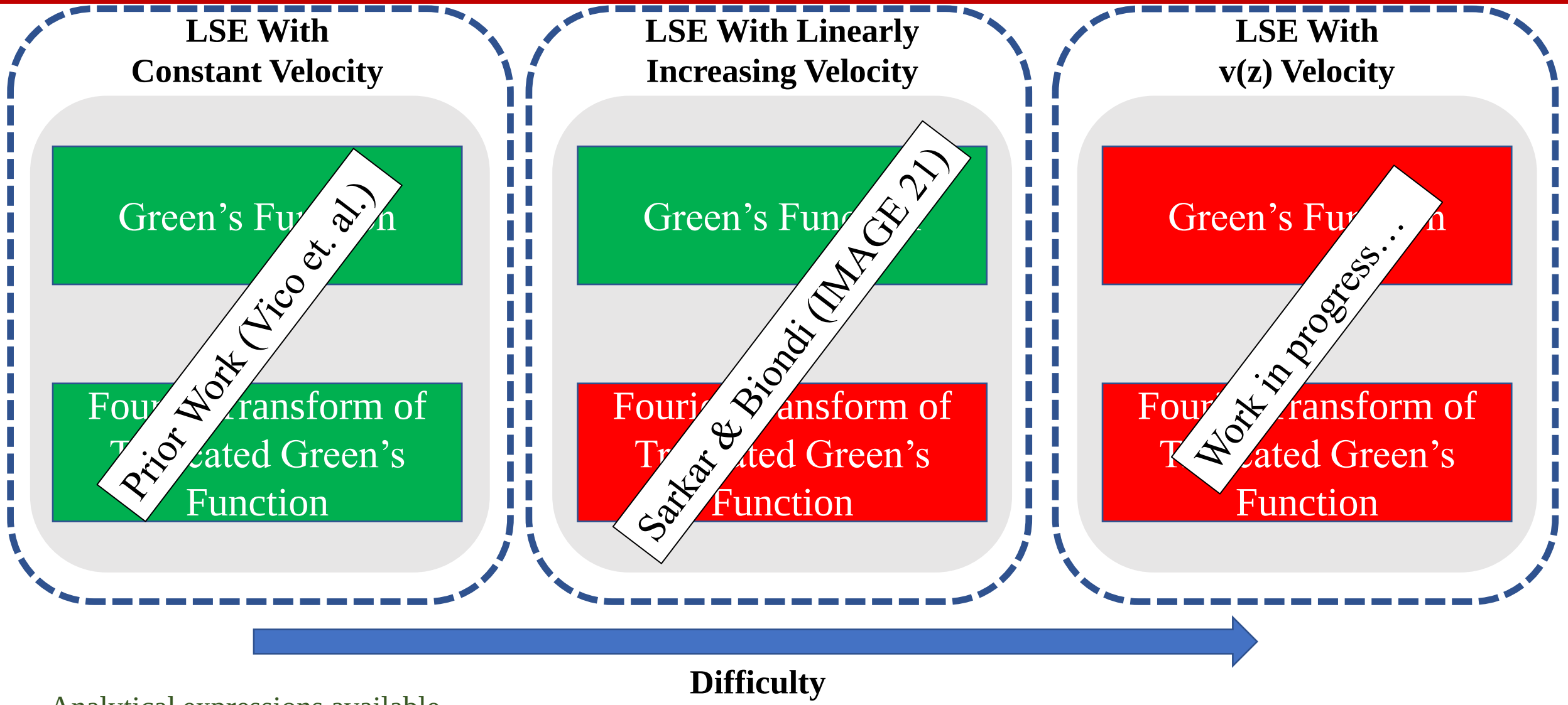
Hierarchy of difficulty



Analytical expressions available.

Analytical expressions not available. Needs to be computed.

Hierarchy of difficulty



Analytical expressions available.

Analytical expressions not available. Needs to be computed.

Helmholtz equation: Our setup

Consider now the Helmholtz equation in the half-space $\mathbb{H} = \{(x, z) : x \in \mathbb{R}^d, z > 0\}$, for $d = 1, 2$ (here x, z denotes horizontal and vertical coordinates respectively)

$$\left(\Delta + \frac{\omega^2}{c(x,z)^2}\right)\phi = f, \tag{3}$$

perturbation

where $\Delta = \frac{\partial^2}{\partial z^2} + \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}$, $c(x, z) = \alpha z + w(x, z) > 0$, $\alpha > 0$, and $w \in C_0^\infty(\mathbb{H})$.

Thus the **background component** of $c(x, z)$ is now the **linearly varying** function αz. If $w = 0$ and $f = \delta(x - x_0, z - z_0)$, the solution ϕ to (3) (again called the Green's function) is translation invariant in the horizontal coordinates only, which we now denote $G_\alpha(x - x_0, z, z_0)$.

The equivalent Lippmann-Schwinger equation

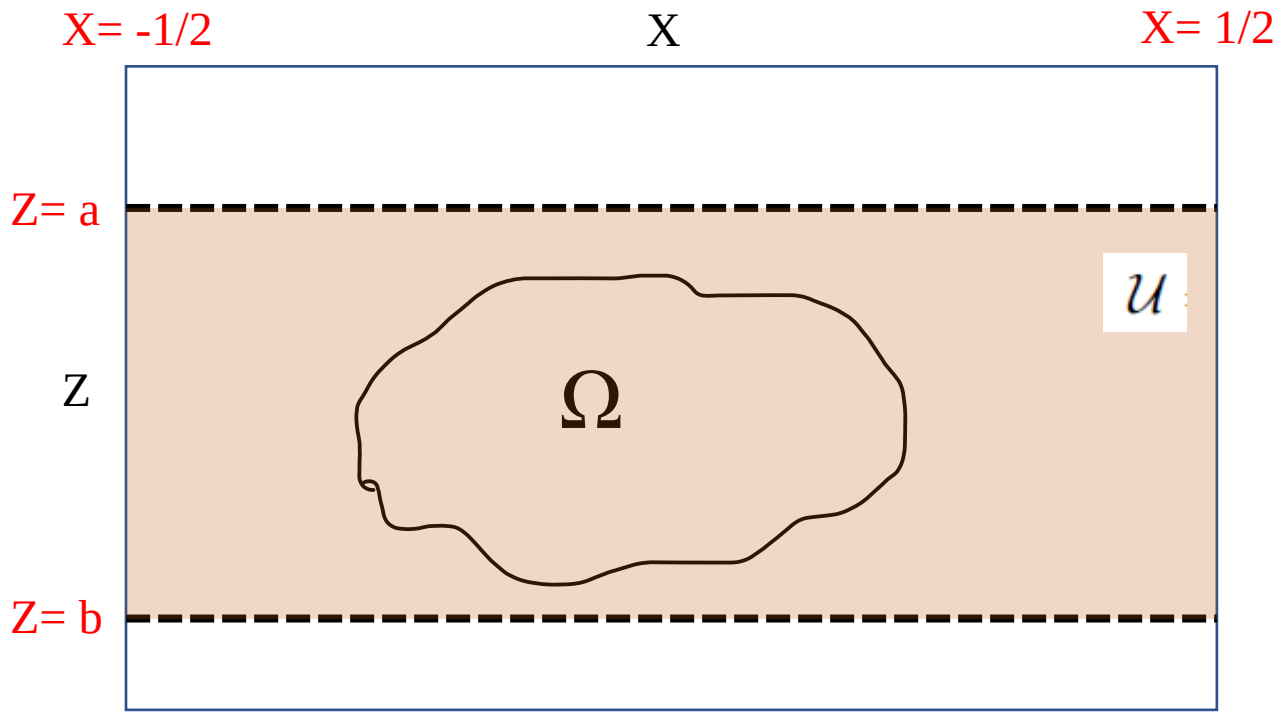
Defining $\psi(x,z) = \left(\frac{1}{\alpha^2 z^2} - \frac{1}{c(x,z)^2}\right)$, the solution ϕ to (3) satisfies an integral equation similar to (2) obtained by rearranging (3), multiplying both sides by G_α, and integrating over $\mathbb{H}$

$$\begin{aligned}\phi(x,z) = &\int G_\alpha(x-x',z,z')f(x',z')\,dx'dz' \\ &+ \omega^2 \int G_\alpha(x-x',z,z')\psi(x',z')\phi(x',z')\,dx'dz'.\end{aligned} \tag{5}$$

- Our goal in this work is to design an efficient finite difference based numerical scheme to solve (5).
- Discretization of (2) leads to a dense linear system which is then solved using iterative methods like GMRES, where in each iteration the cost of forming the matrix-vector product is dominated by the cost of computing the second integral in the right-hand side of (2).

Simplified domain reduction for computation

- Let Ω be an open subset of $\mathbb{H}$ with compact closure, and suppose $w, f \in C_0^\infty(\Omega)$.
- Then it is clear from definition that $\psi \in C_0^\infty(\Omega)$.
- Let us also assume without loss of generality, that Ω is a subset of $\mathcal{U} = \left[-\frac{1}{2}, \frac{1}{2}\right]^d \times [a, b]$, for some $0 < a < b$.



One can always reduce the LSE to be solved to this setting by an isotropic scaling of the coordinates, where a and b depend on the problem.

The LSE to discretize

Denote by $\mathcal{A}$ the map

$$C_0^\infty(\mathcal{U}) \ni v \mapsto \int_{\mathcal{U}} G_\alpha(x - x', z, z')v(x', z')\, dx'\, dz', \tag{6}$$

where the output is also restricted to $\mathcal{U}$, then from the support conditions on w and f, we find that (5) can be written compactly as

$$\phi - \omega^2 \mathcal{A}(\psi\phi) = \mathcal{A}(f) \quad \text{on } \mathcal{U}. \tag{7}$$

Evaluating the integral

Break up the X and Z integrals: first integrate in X and then integrate in Z

One issue in computing (6) is that $G_\alpha(x - x', z, z')$ has a singularity whenever $(x, z) = (x', z')$. We start by splitting the integral into an integration over the x coordinate, followed by an integration over the z coordinate. Thus for any $(x, z) \in \mathcal{U}$ and $z' \in [a, b]$, denoting $\mathcal{U}_x = \left[-\frac{1}{2}, \frac{1}{2}\right]^d$ and

$$(\mathcal{L}v)(x, z, z') = \int_{\mathcal{U}_x} G_\alpha(x - x', z, z')v(x', z')\, dx', \tag{8}$$

one obtains from (6) that $(\mathcal{A}v)(x, z) = \int_a^b (\mathcal{L}v)(x, z, z')dz'$, for all $v \in C_0^\infty(\Omega)$.

Evaluating the integral

Break up the X and Z integrals: first integrate in X and then integrate in Z

One issue in computing (6) is that $G_\alpha(x - x', z, z')$ has a singularity whenever $(x, z) = (x', z')$. We start by splitting the integral into an integration over the x coordinate, followed by an integration over the z coordinate. Thus for any $(x, z) \in \mathcal{U}$ and $z' \in [a, b]$, denoting $\mathcal{U}_x = \left[-\frac{1}{2}, \frac{1}{2}\right]^d$ and

$$(\mathcal{L}v)(x, z, z') = \int_{\mathcal{U}_x} G_\alpha(x - x', z, z')v(x', z')\, dx', \tag{8}$$

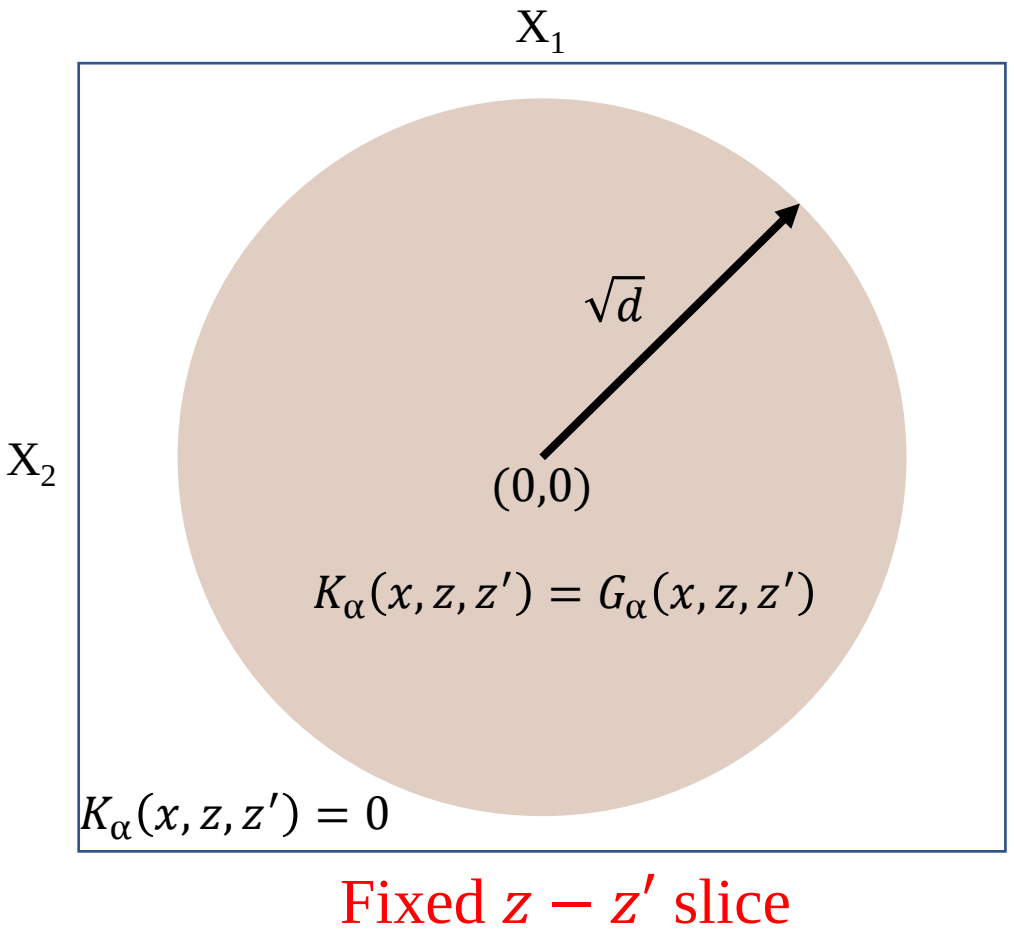
one obtains from (6) that $(\mathcal{A}v)(x, z) = \int_a^b (\mathcal{L}v)(x, z, z')dz'$, for all $v \in C_0^\infty(\Omega)$.

- The integral in (8) is a convolution which can be handled using the truncated kernel method, which also allows us to handle the presence of singularities in the integrand.

Truncated kernel method

Let us define for all $x \in \mathbb{R}^d$ and $z, z' \in (0, \infty)$ the "truncated Green's function"

$$K_\alpha(x, z, z') = G_\alpha(x, z, z')\mathbb{I}_{B(0,\sqrt{d})}(x),$$



Fixed $z - z'$ slice

Truncated Green's function = Original Green's function inside ball $B(0, \sqrt{d})$

Truncated Green's function = 0 outside ball $B(0, \sqrt{d})$

$d = 2$ for 3D case

Truncated kernel method

Notation: Let $\mathcal{F}, \mathcal{F}^{-1}$ denote the partial Fourier transform and partial inverse Fourier transform respectively in the x coordinate. We will use a "hat" to denote Fourier transformed quantities.

Then one can prove that for fixed z, z', $(\mathcal{L}v)(\cdot, z, z') \in C^\infty(\mathcal{U}_x)$ given by

Main result:
$$(\mathcal{L}v)(x, z, z') = \mathcal{F}^{-1}\left(\widehat{K}_\alpha(\cdot, z, z')\widehat{v}(\cdot, z')\right)(x), \tag{10}$$

and moreover $\widehat{K}_\alpha(\xi, z, z')\widehat{v}(\xi, z')$ is **smooth and rapidly decaying** as $|\xi| \to \infty$. This allows for the computation of (10) efficiently using the fast Fourier transform (FFT).

Truncated kernel method

Notation: Let $\mathcal{F}, \mathcal{F}^{-1}$ denote the partial Fourier transform and partial inverse Fourier transform respectively in the x coordinate. We will use a "hat" to denote Fourier transformed quantities.

Then one can prove that for fixed z, z', $(\mathcal{L}v)(\cdot, z, z') \in C^\infty(\mathcal{U}_x)$ given by

$$(\mathcal{L}v)(x, z, z') = \mathcal{F}^{-1}\left(\widehat{K}_\alpha(\cdot, z, z')\widehat{v}(\cdot, z')\right)(x), \tag{10}$$

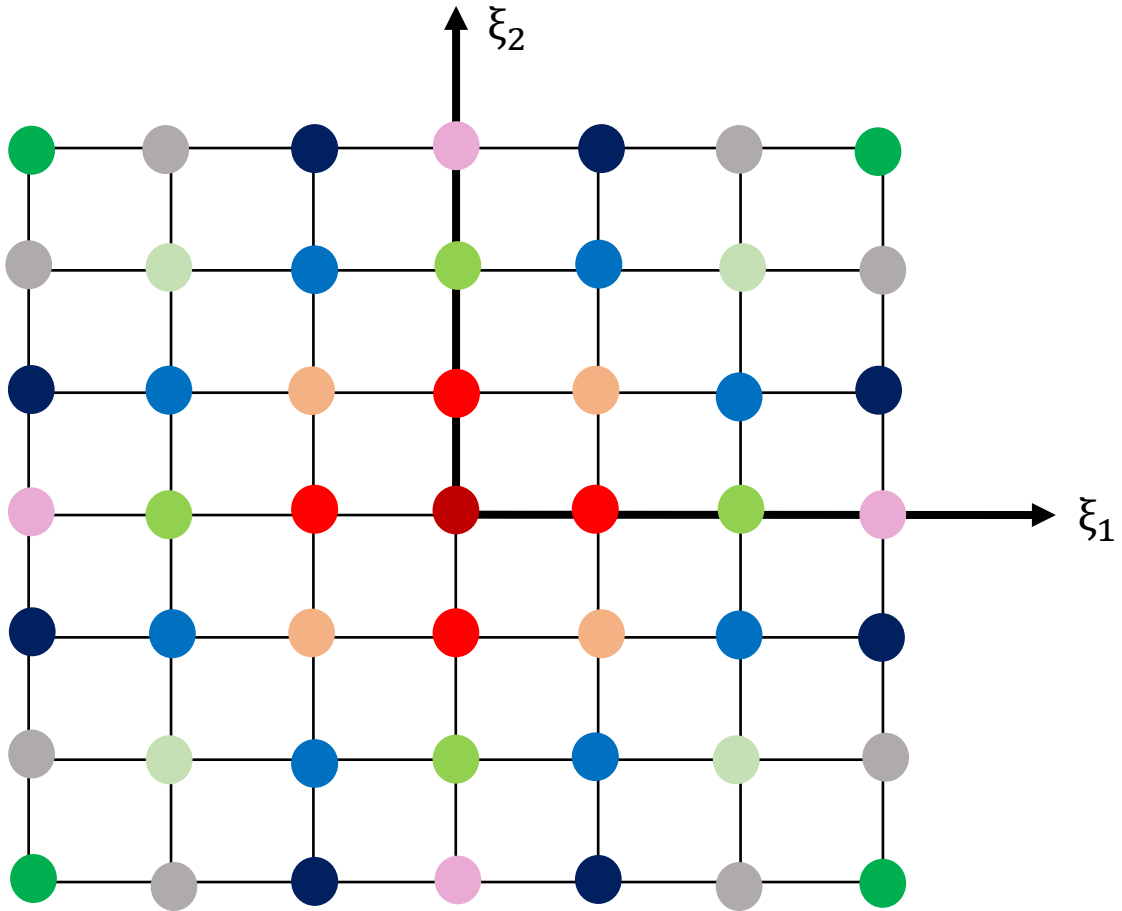
and moreover $\widehat{K}_\alpha(\xi, z, z')\widehat{v}(\xi, z')$ is **smooth and rapidly decaying** as $|\xi| \to \infty$. This allows for the computation of (10) efficiently using the fast Fourier transform (FFT).

This can be precomputed.

Fast precomputation (Trick #1)

Symmetries:

- **Z-reciprocity:** $\widehat{K}_\alpha(\xi, z, z') = \widehat{K}_\alpha(\xi, z', z)$
- **Radial symmetry:** $\widehat{K}_\alpha(\xi, z, z') = \widehat{K}_\alpha(|\xi|, z, z')$

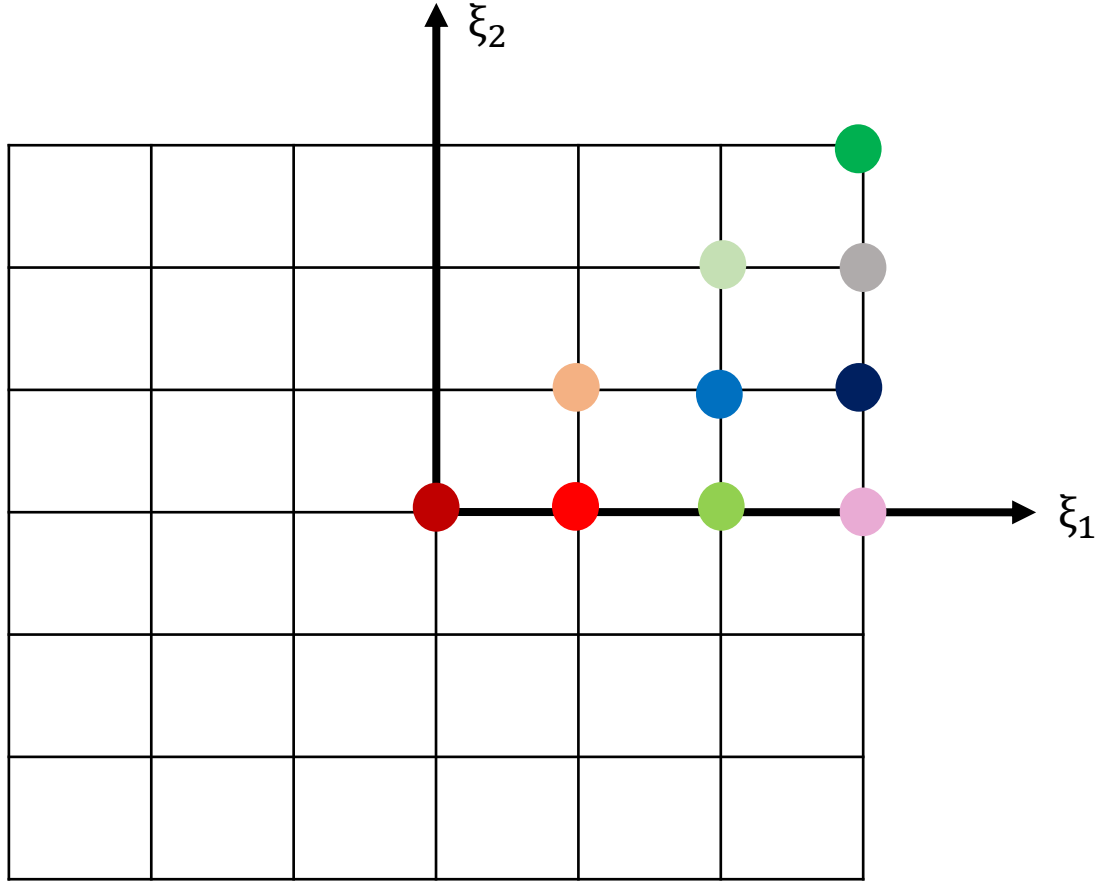


Main Goals:

- Have a method that can be vectorized and parallelized.
- Compatible with numpy vectorization requirements.

Fast precomputation (Trick #1)

- Store only $z \geq z'$.
- Store only unique values of $|\xi|$.
  - ❖ Assume spacing along ξ coordinates is $\Delta\xi$.
  - ❖ Then $|\xi|_{max} = k_{max}\Delta\xi$.
  - ❖ $(\xi_1, \xi_2) = (k_1, k_2)\Delta\xi$.
  - ❖ For each k_1, k_2
    - Evaluate $k_1^2 + k_2^2$ and keep unique values
    - Sort by these values (optional)



- Only need to do the precomputation for these values of $z, z', |\xi|$.
- Parallelize computation over (z, z') pairs or **sorted axis** of $|\xi|$ values.

Fast precomputation (Trick #2)

Let's look at the 3D case.

Need to compute the field:

$$\widehat{K}_\alpha(\xi, z, z') = \int_{B(0,\sqrt{d})} G_\alpha(x, z, z') e^{-i\xi\cdot x} \, dx.$$

4D object but has symmetries

2D integral

Fast precomputation (Trick #2)

Let's look at the 3D case.

Need to compute the field:

$$\widehat{K}_\alpha(\xi, z, z') = \int_{B(0,\sqrt{d})} G_\alpha(x, z, z') e^{-i\xi\cdot x} \; dx.$$

Polar coordinate trick:

$$\widehat{K}_\alpha(\xi, z, z') = \int_0^{\sqrt{2}} \int_0^{2\pi} G_\alpha(r, z, z') e^{-ir|\xi|\cos(\theta)} r \, d\theta dr$$

1D integral

$$= 4\pi \int_0^1 r G_\alpha(r\sqrt{2}, z, z') J_0(r|\xi|\sqrt{2}) \, dr,$$

Fast precomputation (Trick #2)

Let's look at the 3D case.

Need to compute the field:

$$\widehat{K}_\alpha(\xi, z, z') = \int_{B(0,\sqrt{d})} G_\alpha(x, z, z') e^{-i\xi\cdot x} \, dx.$$

Polar coordinate trick:

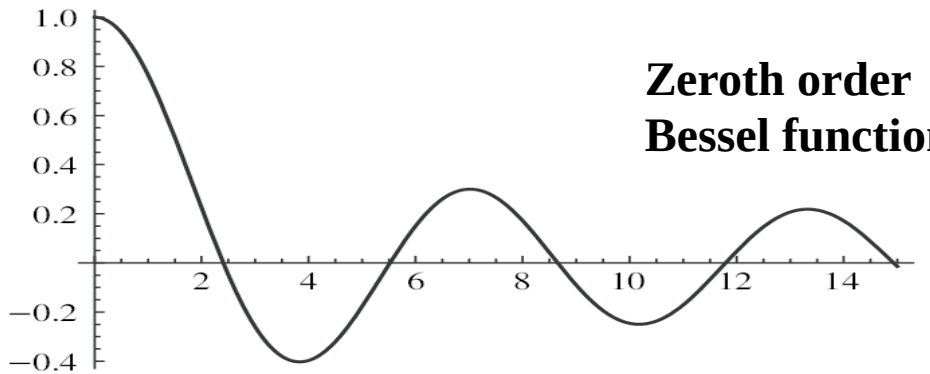
$$\widehat{K}_\alpha(\xi, z, z') = \int_0^{\sqrt{2}} \int_0^{2\pi} G_\alpha(r, z, z') e^{-ir|\xi|\cos(\theta)} r \, d\theta dr$$

1D integral

$$= 4\pi \int_0^1 r G_\alpha(r\sqrt{2}, z, z') J_0(r|\xi|\sqrt{2}) \, dr,$$

Riemann sum:

$$\overline{\widehat{K}}_\alpha(\xi, z, z') = \frac{4\pi}{M^2} \sum_{\ell=0}^{M-1} \ell G_\alpha(\ell\sqrt{2}/M, z, z') J_0(\ell|\xi|\sqrt{2}/M)$$



Zeroth order Bessel function

[0,1] discretized into **M** intervals which should be fine enough to sample the Bessel function

Kernel application algorithm and complexity

Algorithm 1 Compute $\mathscr{S}(\mathscr{A}v)$ in 3D case

procedure KERNEL_APPLY($\mathbf{v} \in \mathbb{C}^N$)
  // Compute $\widehat{K}_\alpha(\xi, z, z')$
  for $(k_1, k_2, j) \in \bar{P}_\xi$ and $j' \in \{0, \ldots, n_z - 1\}$ **do**
    $\widehat{\mathbf{K}}_\alpha(k_1, k_2, j, j') \leftarrow$ Compute $\widehat{K}_\alpha(\xi_{k_1,k_2}, z_j, z'_{j'})$
  $\bar{\mathbf{v}} \leftarrow$ Zero-pad $\mathbf{v}$ on $\bar{P} \setminus P$
  $\widehat{\bar{\mathbf{v}}} \leftarrow$ FFT(FFT_Shift($\bar{\mathbf{v}}$)) ▷ Fourier transform in x
  for $j \in \{0, \ldots, n_z - 1\}$ **do** ▷ Loop over z slices
    for $(k_1, k_2, j') \in \bar{P}_\xi$ **do**
      $\widehat{\mathbf{p}}(k_1, k_2, j') \leftarrow \widehat{\mathbf{K}}_\alpha(k_1, k_2, j, j')\widehat{\bar{\mathbf{v}}}(k_1, k_2, j')$
    for $k_1, k_2 \in \{0, \ldots, 2\beta n\}$ **do**
      $\widehat{\bar{\mathbf{q}}}(k_1, k_2, j) \leftarrow \sum_{j'=1}^{n_z-1} \mu_{j'} \widehat{\mathbf{p}}(k_1, k_2, j')$
  $\bar{\mathbf{q}} \leftarrow$ FFT_Shift(IFFT($\widehat{\bar{\mathbf{q}}}$))
  $\mathbf{q} \leftarrow$ Extract $\bar{\mathbf{q}}$ on P
  return q

n: number of gridpoints in X direction
n_z: number of gridpoints in Z direction
$N = (n+1)^d n_z$

Complexity of kernel application step:

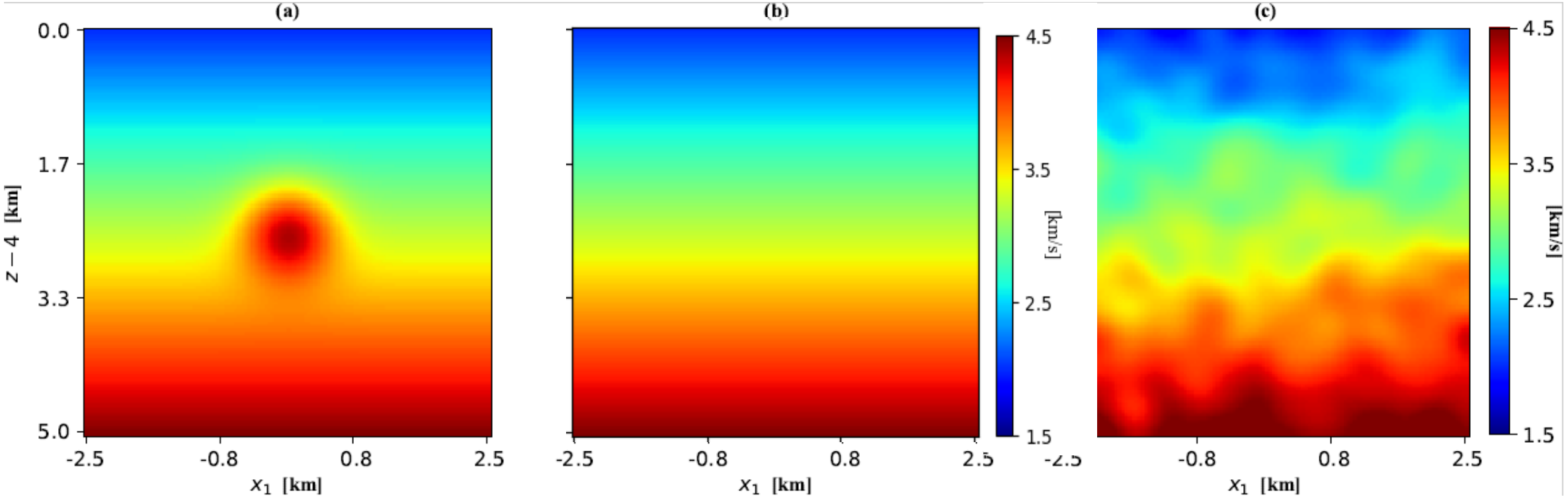
$O(Nn_z + N\log(n))$

Complexity of kernel application step for constant background LSE:

$O(N\log(N))$

Numerical examples (3D): velocity perturbations

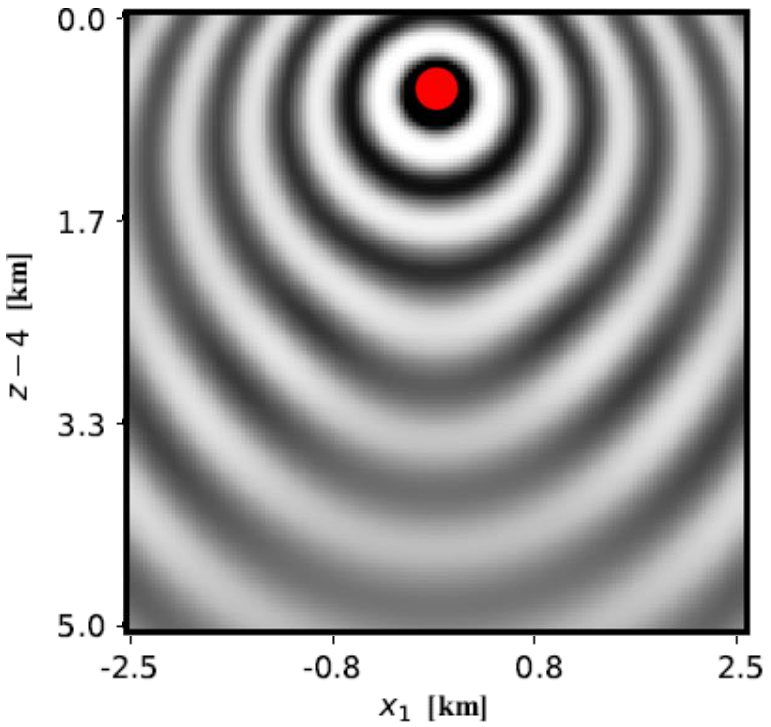
Background velocity



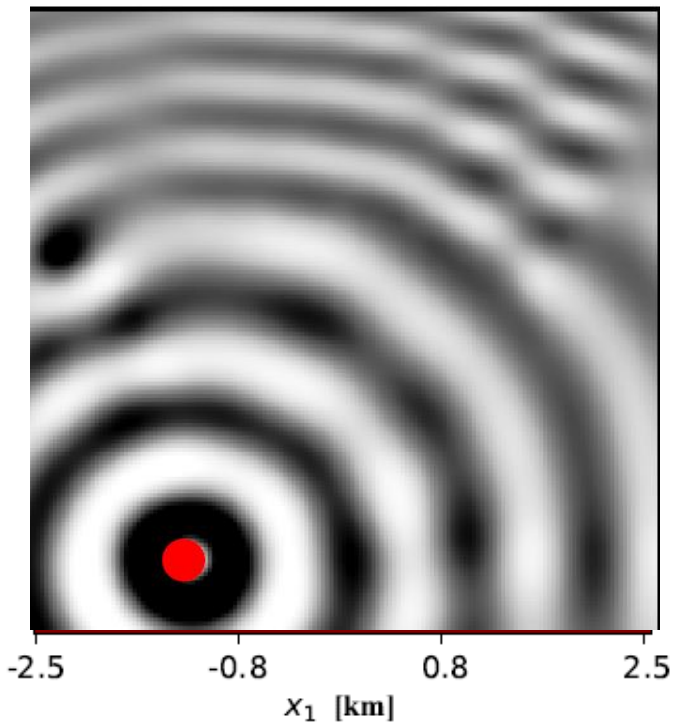
Gaussian anomaly | **Salt perturbation** | **Random perturbation**

Plot of $v(x_1, 0, z)$

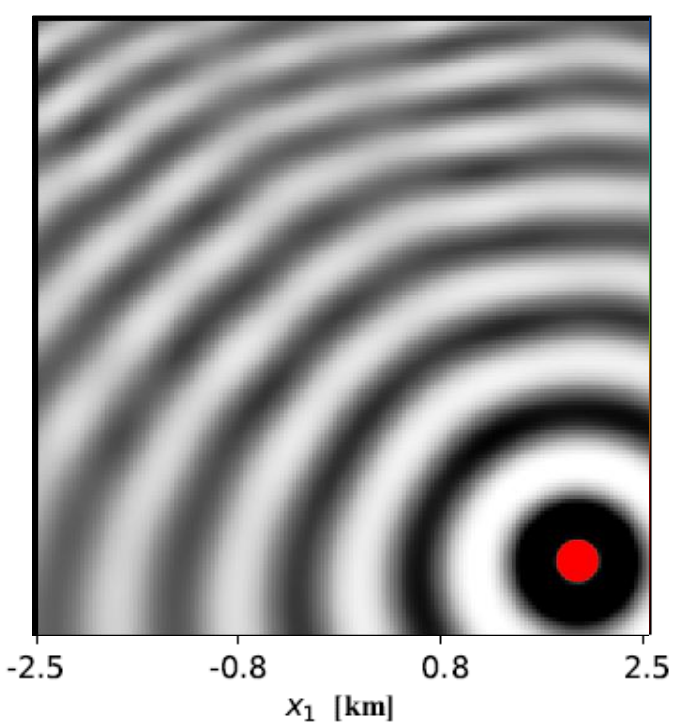
Numerical examples (3D): solutions



Gaussian anomaly



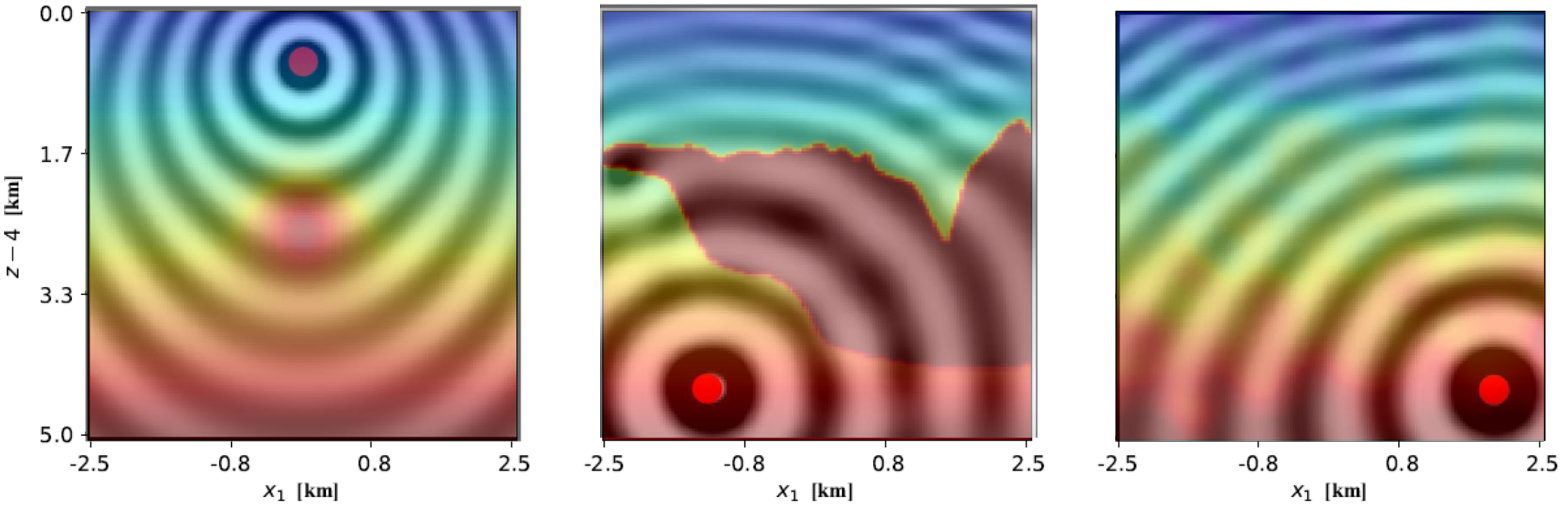
Salt perturbation



Random perturbation

(Data modeled at 5 Hz)

Numerical examples (3D): overlay

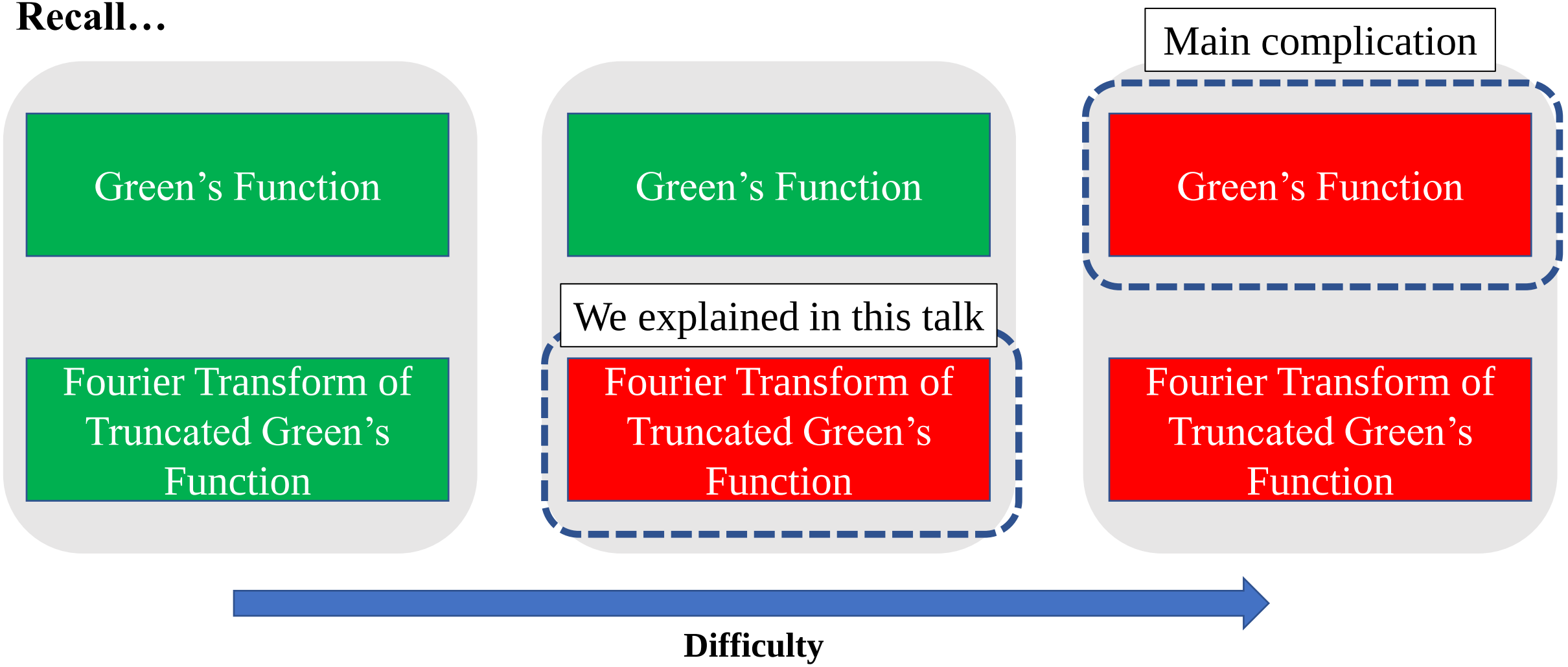


Gaussian anomaly **Salt perturbation** **Random perturbation**

(Data modeled at 5 Hz)

Generalization to arbitrary v(z) background

Recall…

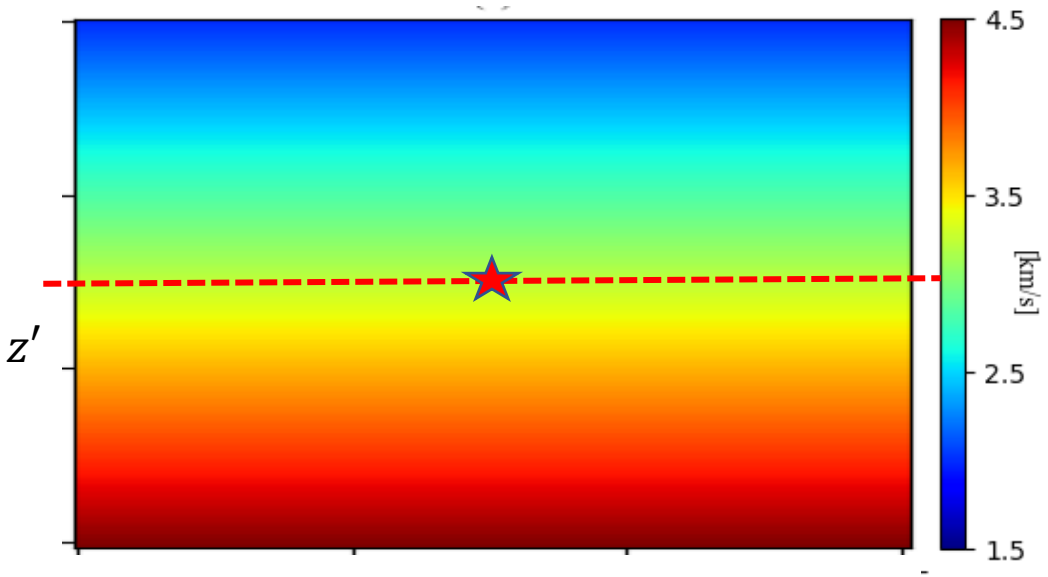


Analytical expressions available.

Analytical expressions not available. Needs to be computed.

Why computing Green's functions expensive?

Illustration of 2D Case



Nz = # depth slices

Need to solve Helmholtz equation for each depth slice and frequency.

Need to compute

$$G_\alpha(x, z, z')$$

Vary location of Dirac delta source(some approximation) at depth z'

- Solve 2D Helmholtz Equation with Dirac delta source as RHS to get the Green's function approximately.
- This is even more expensive in 3D setting. However 3D problem reduces to a 2D problem by using horizontal symmetry.

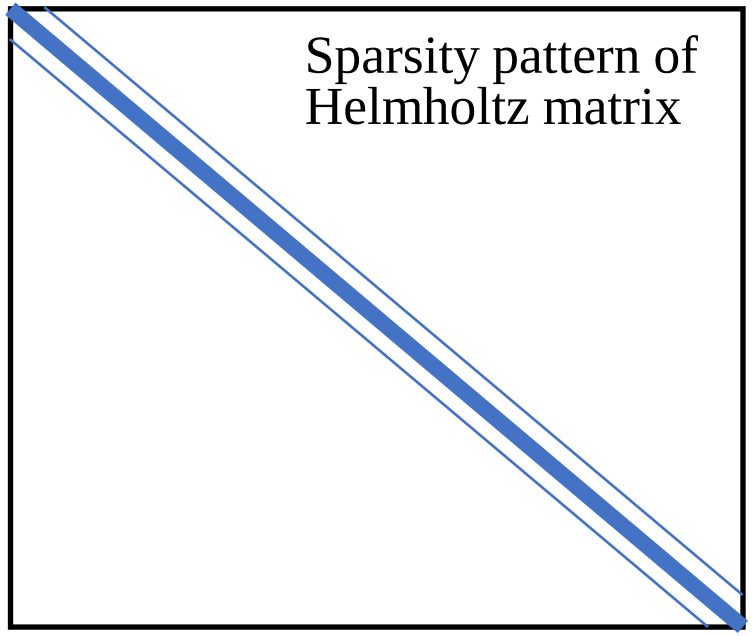
How to make this more efficient?

2D nature of problem + Sparse Linear Algebra

- Since this is a 2D problem, sparse matrix techniques can be used
- Specifically suppose you need to solve

$$HG(x, z, z') = f, \qquad where\ H\ is\ Helmholtz\ operator$$

- Then one can form LU factorization of $\boldsymbol{H = LU}$
- The factors $\boldsymbol{L, U}$ do not have a lot of fill in (because of 2D nature of problem)
- For each RHS f, one obtains $G(x, z, z')$ by one forward solve and backsolve (fast using sparse matrix techniques).



Final remarks and conclusions

- We have developed a new numerical scheme to solve the LSE with a linearly varying background sound speed profile.
- The method solves the LSE using an iterative linear equation solver like GMRES which is a Krylov subspace method, and not by trying to sum a Born scattering series. Thus the method is guaranteed to converge to the correct answer irrespective of the strength of the perturbation.
- Iterative solver converges in fewer iterations at lower frequencies and also when the strength of the perturbation is small.

Acknowledgments: We thank the sponsors of Stanford Exploration Project for financial support.

For questions: **rsarkar@stanford.edu**