Direct solvers based on discrete scattering matrices

Scott Weady

Fast Solvers
November 18, 2020

Overview

- We want to solve a dense $N \times N$ linear system

$$\boldsymbol{A}\boldsymbol{q} = \boldsymbol{f}$$

  that comes from the discretization of an integral equation.
- We'll use concepts familiar from the FMM, but this time we need to both invert a matrix and apply it.
- To apply the inverse we'll use *discrete scattering matrices*.

Hierarchically block separable (HBS) matrices

Definition (HBS matrices)

- (*Top level*) For each pair of distinct leaf nodes τ and τ', let $\boldsymbol{A}_{\tau,\tau'} = \boldsymbol{A}(I_\tau, I_{\tau'})$. We require each such matrix has rank at most k, and that there exist basis matrices $\{\boldsymbol{U}_\tau\}$ and $\{\boldsymbol{V}_\tau\}$ such that for all pairs of distinct leaf nodes $\{\tau, \tau'\}$ we have

$$\boldsymbol{A}_{\tau,\tau'} = \boldsymbol{U}_\tau \tilde{\boldsymbol{A}}_{\tau,\tau'} \boldsymbol{V}_{\tau'}^*.$$

- (*Level ℓ*) For any distinct nodes τ and τ' on level ℓ with children α, β and α', β', respectively, define

$$\boldsymbol{A}_{\tau,\tau'} = \begin{bmatrix} \tilde{\boldsymbol{A}}_{\alpha,\alpha'} & \tilde{\boldsymbol{A}}_{\alpha,\beta'} \\ \tilde{\boldsymbol{A}}_{\beta,\alpha'} & \tilde{\boldsymbol{A}}_{\beta,\beta'} \end{bmatrix}.$$

  We require each such matrix has rank at most k, and that there exists basis matrices as above.

Incoming and outgoing expansions

- Recall the recursive definitions

$$\boldsymbol{U}_\tau^{\text{long}} = \begin{bmatrix} \boldsymbol{U}_\alpha^{\text{long}} & 0 \\ 0 & \boldsymbol{U}_\beta^{\text{long}} \end{bmatrix} \boldsymbol{U}_\tau, \qquad \boldsymbol{V}_\tau^{\text{long}} = \begin{bmatrix} \boldsymbol{V}_\alpha^{\text{long}} & 0 \\ 0 & \boldsymbol{V}_\beta^{\text{long}} \end{bmatrix} \boldsymbol{V}_\tau.$$

- Define

Incoming expansion: $\tilde{\boldsymbol{c}}_\tau = (\boldsymbol{U}_\tau^{\text{long}})^\dagger \boldsymbol{A}(I_\tau, I_\tau^c)\boldsymbol{q}(I_\tau^c)$

Outgoing expansion: $\tilde{\boldsymbol{q}}_\tau = (\boldsymbol{V}_\tau^{\text{long}})^* \boldsymbol{q}(I_\tau).$

- Note that we can also express the incoming expansion in terms of local variables,

$$\tilde{\boldsymbol{c}}_\tau = (\boldsymbol{U}_\tau^{\text{long}})^\dagger (\boldsymbol{f}(I_\tau) - \boldsymbol{A}(I_\tau, I_\tau)\boldsymbol{q}(I_\tau)).$$

Relating the incoming and outgoing expansions

Lemma 18.1

Let τ be a node with children α and β. Then

$$\tilde{\boldsymbol{q}}_\tau = \boldsymbol{V}_\tau^* \begin{bmatrix} \tilde{\boldsymbol{q}}_\alpha \\ \tilde{\boldsymbol{q}}_\beta \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} \tilde{\boldsymbol{c}}_\alpha \\ \tilde{\boldsymbol{c}}_\beta \end{bmatrix} = \boldsymbol{U}_\tau \tilde{\boldsymbol{c}}_\tau.$$

Moreover,

$$\begin{bmatrix} \tilde{\boldsymbol{c}}_\alpha \\ \tilde{\boldsymbol{c}}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \tilde{\boldsymbol{A}}_{\alpha,\beta} \\ \tilde{\boldsymbol{A}}_{\alpha,\beta} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{q}}_\alpha \\ \tilde{\boldsymbol{q}}_\beta \end{bmatrix} + \boldsymbol{U}_\tau \tilde{\boldsymbol{c}}_\tau.$$

- The first two equations follow from the recursive definition of $\boldsymbol{U}_\tau^{\text{long}}$ and $\boldsymbol{V}_\tau^{\text{long}}$.
- The second follows from the definition of the incoming expansion

$$\begin{bmatrix} \boldsymbol{c}_\alpha \\ \boldsymbol{c}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \boldsymbol{A}_{\alpha,\beta} \\ \boldsymbol{A}_{\alpha,\beta} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_\alpha \\ \boldsymbol{q}_\beta \end{bmatrix} + \boldsymbol{U}_\tau^{\text{long}} \tilde{\boldsymbol{c}}_\tau.$$

Outline of the direct solver

(1) Upwards pass (small $\to$ large) in which we build *load vectors* that encode information about $\boldsymbol{f}$, and *discrete scattering matrices* which encode information about the patch being compressed.

(2) Downwards pass (large $\to$ small) in which we build compressed representations of the solution $\boldsymbol{q}$.

(3) One the downwards pass reaches the leaf node, we construct the full solution.

An important feature of this algorithm is the matrices from the build stage can be reused for multiple right hand sides $\boldsymbol{f}$.

Leaf box

- For a leaf node τ, the incoming expansion is

$$\tilde{\boldsymbol{c}}_\tau = (\boldsymbol{U}_\tau)^\dagger(\boldsymbol{f}(I_\tau) - \boldsymbol{A}(I_\tau, I_\tau)\boldsymbol{q}(I_\tau)).$$

  Left multiplying by $\boldsymbol{U}_\tau$ gives

$$\boldsymbol{A}_{\tau,\tau}\boldsymbol{q}_\tau + \boldsymbol{U}_\tau\tilde{\boldsymbol{c}}_\tau = \boldsymbol{f}_\tau.$$

- Left multiplying again by $\boldsymbol{V}^*\boldsymbol{A}_{\tau,\tau}^{-1}$ we get

$$\boxed{\tilde{\boldsymbol{q}}_\tau + \boldsymbol{S}_\tau\tilde{\boldsymbol{c}}_\tau = \tilde{\boldsymbol{y}}_\tau,} \qquad \begin{aligned} \boldsymbol{S}_\tau &:= (\boldsymbol{V}_\tau^*\boldsymbol{A}_{\tau,\tau}^{-1})\boldsymbol{U}_\tau \\ \tilde{\boldsymbol{y}}_\tau &:= (\boldsymbol{V}_\tau^*\boldsymbol{A}_{\tau,\tau}^{-1})\boldsymbol{f}_\tau. \end{aligned}$$

- The matrix $\boldsymbol{S}_\tau$ is the **discrete scattering matrix**.

Parent compression

- Let τ be a parent whose children α and β are leaves in the tree,

$$\begin{bmatrix} \tilde{\boldsymbol{q}}_\alpha \\ \tilde{\boldsymbol{q}}_\beta \end{bmatrix} + \begin{bmatrix} \boldsymbol{S}_\alpha & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_\beta \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{c}}_\alpha \\ \tilde{\boldsymbol{c}}_\beta \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{y}}_\alpha \\ \tilde{\boldsymbol{y}}_\beta \end{bmatrix}.$$

- Using Lemma 18.1 to replace $(\tilde{\boldsymbol{c}}_\alpha, \tilde{\boldsymbol{c}}_\beta)^T$, we can also show

$$\boxed{\tilde{\boldsymbol{q}}_\tau + \boldsymbol{S}_\tau \tilde{\boldsymbol{c}}_\tau = \tilde{\boldsymbol{y}}_\tau.}$$

- This time

$$\boldsymbol{S}_\tau := \boldsymbol{V}_\tau^* \boldsymbol{Z}_\tau \begin{bmatrix} \boldsymbol{S}_\alpha & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_\beta \end{bmatrix} \boldsymbol{U}_\tau \quad \text{and} \quad \tilde{\boldsymbol{y}}_\tau := \boldsymbol{V}_\tau^* \boldsymbol{Z}_\tau \begin{bmatrix} \tilde{\boldsymbol{y}}_\alpha \\ \tilde{\boldsymbol{y}}_\beta \end{bmatrix},$$

  with

$$\boldsymbol{Z}_\tau := \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_\alpha \tilde{\boldsymbol{A}}_{\alpha,\beta} \\ \boldsymbol{S}_\beta \tilde{\boldsymbol{A}}_{\beta,\alpha} & \boldsymbol{I} \end{bmatrix}^{-1}.$$

Top-level solve

- For the top level node $\tau = 1$ with children $\alpha = 2$ and $\beta = 3$, we have

$$\begin{bmatrix} \tilde{\boldsymbol{q}}_2 \\ \tilde{\boldsymbol{q}}_3 \end{bmatrix} + \begin{bmatrix} \boldsymbol{S}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_3 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{c}}_2 \\ \tilde{\boldsymbol{c}}_3 \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{y}}_2 \\ \tilde{\boldsymbol{y}}_3 \end{bmatrix}.$$

- The incoming expansion is $\tilde{\boldsymbol{c}}_1 = 0$, so using Lemma 18.1 we get

$$\begin{bmatrix} \tilde{\boldsymbol{c}}_2 \\ \tilde{\boldsymbol{c}}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \tilde{\boldsymbol{A}}_{2,3} \\ \tilde{\boldsymbol{A}}_{3,2} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_2 \tilde{\boldsymbol{A}}_{2,3} \\ \boldsymbol{S}_3 \tilde{\boldsymbol{A}}_{3,2} & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{\boldsymbol{y}}_2 \\ \tilde{\boldsymbol{y}}_3 \end{bmatrix},$$

  which completes the upwards pass.

Summary

Build stage

for $\tau = N_{\text{boxes}} : (-1) : 2$
  if τ is a leaf
    $\boldsymbol{Z}_\tau = \boldsymbol{A}_{\tau,\tau}^{-1}$
    $\boldsymbol{S}_\tau = \boldsymbol{V}_\tau^* \boldsymbol{Z}_\tau \boldsymbol{U}_\tau$
  else
    Let α and β denote the children of τ.

$$\boldsymbol{Z}_\tau = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_\alpha \tilde{\boldsymbol{A}}_{\alpha,\beta} \\ \boldsymbol{S}_\beta \tilde{\boldsymbol{A}}_{\beta,\alpha} & \boldsymbol{I} \end{bmatrix}^{-1}$$

$$\boldsymbol{S}_\tau = \boldsymbol{V}_\tau^* \boldsymbol{Z}_\tau \begin{bmatrix} \boldsymbol{S}_\alpha & 0 \\ 0 & \boldsymbol{S}_\beta \end{bmatrix} \boldsymbol{U}_\tau$$

  end if
end for

$$\boldsymbol{Z}_1 = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_2 \tilde{\boldsymbol{A}}_{2,3} \\ \boldsymbol{S}_3 \tilde{\boldsymbol{A}}_{3,2} & \boldsymbol{I} \end{bmatrix}^{-1}$$

Summary

Solve stage

Upwards pass:

for $\tau = N_{\text{boxes}} : (-1) : 2$
  if τ is a leaf
    $\tilde{\boldsymbol{y}}_\tau = \boldsymbol{V}_\tau^* \boldsymbol{Z}_\tau \boldsymbol{f}(I_\tau)$
  else
    Let α and β denote the children of τ.
    $\tilde{\boldsymbol{y}}_\tau = \boldsymbol{V}_\tau^* \boldsymbol{Z}_\tau \begin{bmatrix} \tilde{\boldsymbol{y}}_\alpha \\ \tilde{\boldsymbol{y}}_\beta \end{bmatrix}.$
  end if
end for

Top-level solve:

$$\begin{bmatrix} \tilde{\boldsymbol{c}}_2 \\ \tilde{\boldsymbol{c}}_3 \end{bmatrix} = \begin{bmatrix} 0 & \tilde{\boldsymbol{A}}_{2,3} \\ \tilde{\boldsymbol{A}}_{3,2} & 0 \end{bmatrix} \boldsymbol{Z}_1 \begin{bmatrix} \tilde{\boldsymbol{y}}_2 \\ \tilde{\boldsymbol{y}}_3 \end{bmatrix}.$$

Parent solve

- Let τ denote a node that has not yet been processed, but whose parent has been processed, and let α, β be its children.
- We need the outgoing expansions $\{\tilde{\boldsymbol{q}}_\alpha, \tilde{\boldsymbol{q}}_\beta\}$ and the incoming expansions $\{\tilde{\boldsymbol{c}}_\alpha, \tilde{\boldsymbol{c}}_\beta\}$. From the compressed parent representation,

$$\begin{bmatrix} \tilde{\boldsymbol{c}}_\alpha \\ \tilde{\boldsymbol{c}}_\beta \end{bmatrix} = \begin{bmatrix} 0 & \tilde{\boldsymbol{A}}_{\alpha,\beta} \\ \tilde{\boldsymbol{A}}_{\beta,\alpha} & 0 \end{bmatrix} \boldsymbol{Z}_\tau \begin{bmatrix} \tilde{\boldsymbol{y}}_\alpha \\ \tilde{\boldsymbol{y}}_\beta \end{bmatrix} + \left(\boldsymbol{I} - \begin{bmatrix} 0 & \tilde{\boldsymbol{A}}_{\alpha,\beta} \\ \tilde{\boldsymbol{A}}_{\beta,\alpha} & 0 \end{bmatrix} \boldsymbol{Z}_\tau \begin{bmatrix} \boldsymbol{S}_\alpha & 0 \\ 0 & \boldsymbol{S}_\beta \end{bmatrix} \right) \boldsymbol{U}_\tau \tilde{\boldsymbol{c}}_\tau$$

- The load vectors $\tilde{\boldsymbol{y}}_\alpha$ and $\tilde{\boldsymbol{y}}_\beta$ were constructed in the upwards pass, and the incoming expansion $\tilde{\boldsymbol{c}}_\tau$ is known from the previous stage of the downwards pass.

Leaf solve

- Once we get to the leaf τ, we know the incoming expansion $\tilde{\boldsymbol{c}}_\tau$ and the load vector $\boldsymbol{f}$, so the solution at the leaf node is

$$\boldsymbol{q}(I_\tau) = \boldsymbol{A}_{\tau,\tau}^{-1}(\boldsymbol{f}(I_\tau) - \boldsymbol{U}_\tau \tilde{\boldsymbol{c}}_\tau),$$

which completes the downwards pass.

Summary

Solve stage

Downwards pass:

for $\tau = 2 : N_{\text{boxes}}$
  if τ is a parent
    Let α and β denote the children of τ.

$$\begin{bmatrix} \tilde{\boldsymbol{c}}_\alpha \\ \tilde{\boldsymbol{c}}_\beta \end{bmatrix} = \begin{bmatrix} 0 & \tilde{\boldsymbol{A}}_{\alpha,\beta} \\ \tilde{\boldsymbol{A}}_{\beta,\alpha} & 0 \end{bmatrix} \boldsymbol{Z}_\tau \begin{bmatrix} \tilde{\boldsymbol{y}}_\alpha \\ \tilde{\boldsymbol{y}}_\beta \end{bmatrix} + \left(\boldsymbol{I} - \begin{bmatrix} 0 & \tilde{\boldsymbol{A}}_{\alpha,\beta} \\ \tilde{\boldsymbol{A}}_{\beta,\alpha} & 0 \end{bmatrix} \boldsymbol{Z}_\tau \begin{bmatrix} \boldsymbol{S}_\alpha & 0 \\ 0 & \boldsymbol{S}_\beta \end{bmatrix} \right) \boldsymbol{U}_\tau \tilde{\boldsymbol{c}}_\tau.$$

  else
    $\boldsymbol{q}(I_\tau) = \boldsymbol{Z}_\tau (f(I_\tau) - \boldsymbol{U}_\tau \tilde{\boldsymbol{c}}_\tau)$.
  end if
end for

Comparison with the original multilevel scheme

Build stage (Chapter 14)

loop over all levels, finest to coarsest, $\ell = L, L-1, \ldots, 1$
loop over all boxes τ on level ℓ,
if τ is a leaf node
$\tilde{\boldsymbol{D}}_\tau = \boldsymbol{D}_\tau$.
else
Let α and β denote the children of τ.
$\tilde{\boldsymbol{D}}_\tau = \begin{bmatrix} \hat{\boldsymbol{D}}_\alpha & \tilde{\boldsymbol{A}}_{\alpha,\beta} \\ \tilde{\boldsymbol{A}}_{\beta,\alpha} & \hat{\boldsymbol{D}}_\beta \end{bmatrix}$.
end if
$\hat{\boldsymbol{D}}_\tau = (\boldsymbol{V}_\tau^* \tilde{\boldsymbol{D}}_\tau^{-1} \boldsymbol{U}_\tau)^{-1}$.
$\boldsymbol{E}_\tau = \tilde{\boldsymbol{D}}_\tau^{-1} \boldsymbol{U}_\tau \hat{\boldsymbol{D}}_\tau$.
$\boldsymbol{F}_\tau^* = \hat{\boldsymbol{D}}_\tau \boldsymbol{V}_\tau^* \tilde{\boldsymbol{D}}_\tau^{-1}$.
$\boldsymbol{G}_\tau = \tilde{\boldsymbol{D}}_\tau - \tilde{\boldsymbol{D}}_\tau^{-1} \boldsymbol{U}_\tau \hat{\boldsymbol{D}}_\tau \boldsymbol{V}_\tau^* \tilde{\boldsymbol{D}}_\tau^{-1}$.
end loop
end loop

$$\boldsymbol{G}_1 = \begin{bmatrix} \hat{\boldsymbol{D}}_2 & \tilde{\boldsymbol{A}}_{2,3} \\ \tilde{\boldsymbol{A}}_{3,2} & \hat{\boldsymbol{D}}_3 \end{bmatrix}^{-1}.$$

Comparison with the original multilevel scheme

- For each build stage, we need an HBS matrix $\boldsymbol{A}$ stored in the compressed form of its basis matrices $\{\boldsymbol{U}_\tau, \boldsymbol{V}_\tau\}$ and its sibling interaction matrices $\{\tilde{\boldsymbol{A}}_{\alpha,\beta}\}$ such that

$$\boldsymbol{A}_{\alpha,\beta} = \boldsymbol{U}_\alpha \tilde{\boldsymbol{A}}_{\alpha,\beta} \boldsymbol{V}_\beta^*.$$

- For the scattering direct solver, the outputs of the build stage are the scattering matrices $\{\boldsymbol{S}_\tau\}$ and the "dual operators" $\{\boldsymbol{Z}_\tau\}$.
- The algorithm from Chapter 14 outputs four sets of matrices $\{\boldsymbol{E}_\tau, \boldsymbol{F}_\tau, \boldsymbol{G}_\tau, \hat{\boldsymbol{D}}_\tau\}$.
- Moreover, the build stage for the scattering matrices only requires one matrix inversion compared to two from the other scheme.

More on scattering matrices

Lemma 18.2

Let $\boldsymbol{A}$ be an HBS matrix whose diagonal blocks are all invertible, and let $\boldsymbol{f} \in \mathbb{R}^N$ be an arbitrary vector. Define for any node τ the following objects:

Scattering matrix: $\boldsymbol{S}_\tau = (\boldsymbol{V}_\tau^{\text{long}})^*(\boldsymbol{A}(I_\tau, I_\tau))^{-1}(\boldsymbol{U}_\tau^{\text{long}}),$

Effective charges: $\tilde{\boldsymbol{y}}_\tau = (\boldsymbol{V}_\tau^{\text{long}})^*(\boldsymbol{A}(I_\tau, I_\tau))^{-1}\boldsymbol{f}(I_\tau).$

Then, for any parent node τ with children α and β, we have

$$\boldsymbol{S}_\tau = \boldsymbol{V}_\tau^* \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_\alpha \tilde{\boldsymbol{A}}_{\alpha,\beta} \\ \boldsymbol{S}_\beta \tilde{\boldsymbol{A}}_{\beta,\alpha} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{S}_\alpha & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_\beta \end{bmatrix} \boldsymbol{U}_\tau,$$

$$\tilde{\boldsymbol{y}}_\tau = \boldsymbol{V}_\tau^* \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_\alpha \tilde{\boldsymbol{A}}_{\alpha,\beta} \\ \boldsymbol{S}_\beta \tilde{\boldsymbol{A}}_{\beta,\alpha} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{y}}_\alpha \\ \tilde{\boldsymbol{y}}_\beta \end{bmatrix}.$$

More on scattering matrices

- The point of Lemma 18.2 is that we can use the *nonrecursive* definitions

$$\begin{aligned} \boldsymbol{S}_\tau &= (\boldsymbol{V}_\tau^{\text{long}})^*(\boldsymbol{A}(I_\tau, I_\tau))^{-1}(\boldsymbol{U}_\tau^{\text{long}}), \\ \tilde{\boldsymbol{y}}_\tau &= (\boldsymbol{V}_\tau^{\text{long}})^*(\boldsymbol{A}(I_\tau, I_\tau))^{-1}\boldsymbol{f}(I_\tau) \end{aligned}$$

  to define the scattering matrices and load vectors.

- This form looks the same as that for the leaf nodes, just with $\boldsymbol{U}_\tau, \boldsymbol{V}_\tau$ replaced by $\boldsymbol{U}_\tau^{\text{long}}, \boldsymbol{V}_\tau^{\text{long}}$ which makes analysis of the algorithm a bit cleaner.

Inversion of the diagonal blocks

Theorem 18.3

Let $\boldsymbol{A}$ be an invertible HBS matrix and let $\{\hat{\boldsymbol{D}}_\tau\}_\tau$ be the compressed diagonal blocks from the inversion algorithm of Chapter 14. As long as no singular matrices are encountered, we have

$$\hat{\boldsymbol{D}}_\tau = \boldsymbol{S}_\tau^{-1}.$$

- This shows the inversion algorithm based on scattering matrices is mathematically equivalent to the HBS scheme from Chapter 14.
- The main difference is the central object here is the discrete scattering matrix $\boldsymbol{S}_\tau$ rather than its inverse, which makes the solver more stable.

Interpretation of the scattering matrix

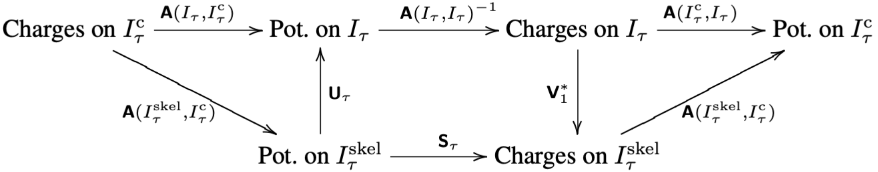
- Consider the interpolative decomposition:

$$\boldsymbol{A}(I_\tau, I_\tau^c) \approx \boldsymbol{U}_\tau \boldsymbol{A}(I_\tau^{\mathsf{skel}}, I_\tau^c) \quad \text{and} \quad \boldsymbol{A}(I_\tau^c, I_\tau) \approx \boldsymbol{A}(I_\tau^c, I_\tau^{\mathsf{skel}})\boldsymbol{V}_\tau^*.$$

- When evaluating the Schur complement, we need

$$\begin{aligned}\boldsymbol{A}(I_\tau^c, I_\tau)(\boldsymbol{A}(I_\tau, I_\tau))^{-1}\boldsymbol{A}(I_\tau, I_\tau^c) &\approx \boldsymbol{A}(I_\tau^c, I_\tau^{\mathsf{skel}})\boldsymbol{V}_\tau^*(A(I_\tau, I_\tau))^{-1}\boldsymbol{U}_\tau \boldsymbol{A}(I_\tau^{\mathsf{skel}}, I_\tau^c) \\ &= \boldsymbol{A}(I_\tau^c, I_\tau^{\mathsf{skel}})\boldsymbol{S}_\tau \boldsymbol{A}(I_\tau^{\mathsf{skel}}, I_\tau^c)\end{aligned}$$

- We can summarize th through the following diagram:



Application: the Lippmann-Schwinger equation

- The Lippmann-Schwinger equation models acoustic wave propogation in a medium with variable wave speed.
- The equation in integral form is

$$\sigma(\boldsymbol{x}) + \kappa^2 b(\boldsymbol{x}) \int_\Omega G_\kappa(\boldsymbol{x}, \boldsymbol{y}) \sigma(\boldsymbol{y}) d\boldsymbol{y} = -\kappa^2 b(\boldsymbol{x}) u_{\text{in}}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega.$$

where

$$G_\kappa(\boldsymbol{x}, \boldsymbol{x}') = \frac{i}{4} H_0(\kappa |\boldsymbol{x} - \boldsymbol{y}|).$$

- The coefficient matrix $\boldsymbol{A}$ takes the form

$$\boldsymbol{A} = \boldsymbol{I} + \boldsymbol{B}\boldsymbol{G}.$$

where $\boldsymbol{B}$ is diagonal and

$$\boldsymbol{G}_{i,j} = G_\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{for } i \neq j.$$

Outline of the direct solver

- We represent $\boldsymbol{G}$ by its low-rank decomposition,

$$\boldsymbol{G}(I_\alpha, I_\beta) = \boldsymbol{U}_\tau^{\text{long}} \tilde{\boldsymbol{G}}_{\alpha,\beta} (\boldsymbol{V}_\tau^{\text{long}})^*.$$

- The outgoing and incoming expansions are

$$\textbf{Outgoing expansion:} \qquad \tilde{\boldsymbol{q}}_\tau = (\boldsymbol{V}_\tau^{\text{long}})^* \boldsymbol{q}(I_\tau),$$

$$\textbf{Incoming expansion:} \qquad \tilde{\boldsymbol{c}}_\tau = (\boldsymbol{U}_\tau^{\text{long}})^\dagger \boldsymbol{G}(I_\tau, I_\tau^c) \boldsymbol{q}(I_\tau^c).$$

- The sibling exchange relation is

$$\begin{bmatrix} \tilde{\boldsymbol{c}}_\alpha \\ \tilde{\boldsymbol{c}}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \tilde{\boldsymbol{G}}_{\alpha,\beta} \\ \tilde{\boldsymbol{G}}_{\alpha,\beta} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{q}}_\alpha \\ \tilde{\boldsymbol{q}}_\beta \end{bmatrix} + \boldsymbol{U}_\tau \tilde{\boldsymbol{c}}_\tau.$$

Upward pass

Leaf node:

$$\tilde{\boldsymbol{q}}_\tau + \boldsymbol{S}_\tau \tilde{\boldsymbol{c}}_\tau = \tilde{\boldsymbol{y}}_\tau, \qquad \begin{aligned} \boldsymbol{Z}_\tau &= (\boldsymbol{I} + \boldsymbol{B}_\tau \boldsymbol{G}_\tau)^{-1} \\ \boldsymbol{S}_\tau &= \boldsymbol{V}_\tau^* \boldsymbol{Z}_\tau \boldsymbol{B}_\tau \boldsymbol{U}_\tau \\ \tilde{\boldsymbol{y}}_\tau &= \boldsymbol{V}_\tau^* \boldsymbol{Z}_\tau \boldsymbol{f}_\tau. \end{aligned}$$

Parent node:

$$\tilde{\boldsymbol{q}}_\tau + \boldsymbol{S}_\tau \tilde{\boldsymbol{c}}_\tau = \tilde{\boldsymbol{y}}_\tau, \qquad \begin{aligned} \boldsymbol{Z}_\tau &:= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_\alpha \tilde{\boldsymbol{G}}_{\alpha,\beta} \\ \boldsymbol{S}_\beta \tilde{\boldsymbol{G}}_{\beta,\alpha} \boldsymbol{I} & \end{bmatrix}^{-1}, \\ \boldsymbol{S}_\tau &:= \boldsymbol{V}_\tau^* \boldsymbol{Z}_\tau \begin{bmatrix} \boldsymbol{S}_\alpha & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_\beta \end{bmatrix} \boldsymbol{U}_\tau, \\ \tilde{\boldsymbol{y}}_\tau &:= \boldsymbol{V}_\tau^* \boldsymbol{Z}_\tau \begin{bmatrix} \tilde{\boldsymbol{y}}_\alpha \\ \tilde{\boldsymbol{y}}_\beta \end{bmatrix}. \end{aligned}$$

Top-level solve:

$$\begin{bmatrix} \tilde{\boldsymbol{q}}_\alpha \\ \tilde{\boldsymbol{q}}_\beta \end{bmatrix} = \boldsymbol{Z}_\tau \begin{bmatrix} \tilde{\boldsymbol{y}}_\alpha \\ \tilde{\boldsymbol{y}}_\beta \end{bmatrix}, \qquad \begin{bmatrix} \tilde{\boldsymbol{c}}_\alpha \\ \tilde{\boldsymbol{c}}_\beta \end{bmatrix} = \begin{bmatrix} 0 & \tilde{\boldsymbol{G}}_{\alpha,\beta} \\ \tilde{\boldsymbol{G}}_{\beta,\alpha} & 0 \end{bmatrix} \boldsymbol{Z}_\tau \begin{bmatrix} \tilde{\boldsymbol{y}}_\alpha \\ \tilde{\boldsymbol{y}}_\beta \end{bmatrix}.$$

Downwards pass

Parent solve:

$$\begin{bmatrix}\tilde{\boldsymbol{c}}_\alpha\\ \tilde{\boldsymbol{c}}_\beta\end{bmatrix} = \begin{bmatrix}0 & \tilde{\boldsymbol{G}}_{\alpha,\beta}\\ \tilde{\boldsymbol{G}}_{\beta,\alpha} & 0\end{bmatrix} \boldsymbol{Z}_\tau \begin{bmatrix}\tilde{\boldsymbol{y}}_\alpha\\ \tilde{\boldsymbol{y}}_\beta\end{bmatrix} + \Big(\boldsymbol{I} - \begin{bmatrix}0 & \tilde{\boldsymbol{G}}_{\alpha,\beta}\\ \tilde{\boldsymbol{G}}_{\beta,\alpha} & 0\end{bmatrix} \boldsymbol{Z}_\tau \begin{bmatrix}\boldsymbol{S}_\alpha & 0\\ 0 & \boldsymbol{S}_\beta\end{bmatrix}\Big)\boldsymbol{U}_\tau \tilde{\boldsymbol{c}}_\tau.$$

Leaf solve:

$$\boldsymbol{q}_\tau = \boldsymbol{Z}_\tau(\boldsymbol{f}_\tau - \boldsymbol{B}_\tau \boldsymbol{U}_\tau \boldsymbol{c}_\tau).$$