Partitioned Low Rank Compression of Absorbing Boundary Conditions for the Helmholtz equation

Rosalie Bélanger-Rioux, Harvard University

Laurent Demanet, Massachusetts Institute of Technology
Imaging and Computing Group

CSE 15
March 15, 2015
Helmholtz equation in unbounded domain

2D Helmholtz equation

$$\Delta u(x) + \frac{\omega^2}{c^2(x)} u(x) = f(x), \quad x = (x_1, x_2) \in \mathbb{R}^2.$$ 

Solution $u$, frequency $\omega$, medium $c(x)$, source $f(x)$.

Many sources!

Select **outgoing waves** using the Sommerfeld Radiation Condition

$$\lim_{r \to \infty} r^{1/2} \left( \frac{\partial u}{\partial r} - iku \right) = 0, \quad k = \frac{\omega}{c},$$

where $r$ is the radial coordinate.
Applications

- Wave-based imaging, an inverse problem.
  - Seismic imaging: for rock formations.
  - Ultrasonic testing: non-destructive testing of objects for defects.
  - Ultrasonic imaging: visualizing a fetus, muscle, tendon or organ.
  - Synthetic-aperture radar imaging: visualizing a scene or detecting the presence of an object far away or through clouds, foliage.

- Photonics: studying the optical properties of crystals.

- Speeding up Domain Decomposition Methods.
Absorbing Boundary Conditions (ABCs) and Layers (ALs)

\[ \Delta u(x) + \frac{\omega^2}{c^2(x)} u(x) = f(x), \quad k = \frac{\omega}{c(x)}, \quad x \in \Omega. \]

- Close system using Absorbing Boundary Condition (ABC) or Absorbing Layer (AL).
- \( N \) pts per dimension, \( h = 1/N \).

**Issue:** absorbing layers tend to get thick in heterogeneous media.
Absorbing Layers in heterogeneous media

Physical width $L > 1$ or width in number of points $w > N$. 

\[ \begin{align*} 
\Omega & \quad \text{\large \text{f.}} \\
\end{align*} \]
Our numerical scheme

**Goal:** Compress costly ABC or AL to speed up Helmholtz solver

**Step 1:** Obtain the exterior Dirichlet-to-Neumann (DtN) map $D$
- Matrix probing with solves of exterior problem

**Step 2:** Obtain a fast algorithm for matrix-vector products of $D$
- Partitioned low-rank (PLR) matrices, compress off-diagonal blocks

\[
D \xrightarrow{\text{probing}} \tilde{D} \xrightarrow{\text{PLR}} \overline{D}
\]
Step 1: Obtain the exterior DtN map $D$

\[
D \xrightarrow{\text{probing}} \tilde{D} \xrightarrow{\text{PLR}} \overline{D}
\]
The exterior problem to obtain the exterior DtN map

\[ \Delta u(x) + \frac{\omega^2}{c^2(x)} u(x) = f(x), \quad x \in \mathbb{R}^2 \setminus \Omega \]

- \( u(x) = g(x), \quad x \in \partial \Omega. \)
- Use ABC or AL.
- Solution \( u_1 \) on 1\(^{st} \) layer outside \( \Omega. \)
- Obtain product of \( D \) with \( g \):

\[ Dg = \frac{u_1 - g}{h}. \]

- Use \( D \) in a Helmholtz solver instead of ABC or AL.
Matrix probing

\[ M \in \mathbb{C}^{N \times N}, \quad \text{single random vector } z \]

- **Given:** \( z \) and \( Mz \)
- **Problem:** recover \( M \)
- **Model:** there exist \( B_1, \ldots, B_p \) (fixed, given) such that

\[ M = \sum_{j=1}^{p} c_j B_j \]

\( \Rightarrow \) find \( c_j \).
Matrix probing questions

- How to recover \( \mathbf{c} \)?

\[
M \mathbf{z} = \sum_{j=1}^{p} c_j B_j \mathbf{z} = \Psi_{\mathbf{z}} \mathbf{c}
\]

- 1 random realization: \( \Psi_{\mathbf{z}} \) has dimension \( N \) by \( p \).
- \( q > 1 \) random realizations: \( \Psi_{\mathbf{z}} \) has dimension \( Nq \) by \( p \).

- How large can \( p \) get?

- Which \( B_j \)?

Steps of matrix probing and their complexities

Steps of matrix probing:

- Orthonormalize $B_j$’s (QR).
- Build $\Psi_z$ from products $B_jz$.
- Obtain $Mz$.
- Apply pseudoinverse of $\Psi_z$.

Complexity:

- $N^2p^2$.
- $N^2pq$.
- $q$ solves of exterior problem.
- $Np^2q$. 
Media considered (plots of $c(x)$)

Figure: Uniform.

Figure: Slow disk.

Figure: Diagonal fault.

Figure: Waveguide.

Figure: Vertical fault.

Figure: Periodic.
Real part of solutions $u, \omega = 51.2, N = 1024.$

**Figure:** Waveguide.

**Figure:** Vertical, left.

**Figure:** Diagonal fault.

**Figure:** Slow disk.

**Figure:** Vertical, right.

**Figure:** Periodic.
Probing results

\[ D \xrightarrow{\text{probing}} \tilde{D} \xrightarrow{\text{PLR}} \overline{D} \]

- Number of basis matrices \( p \sim N^{0.2} \) at worst.

- Number of exterior solves \( q \) constant as \( N \) grows.

- Probing approximation does not degrade with grazing waves.

Limitations:

- Easier for smooth media;

- Careful design of basis matrices needed.
Step 2: Obtain a fast algorithm for matrix-vector products of $D$

\[ D \xrightarrow{\text{probing expansion}} \tilde{D} \xrightarrow{\text{PLR compression}} \overline{D} \]
Intuition: $D_{\text{half}}$ numerically low-rank away from singularity

Kernel of the uniform half-space DtN map: $K(r) = \frac{ik^2 H_{1}^{(1)}(kr)}{2kr}$.

Theorem (RBR, Demanet)

Let $0 < \epsilon \leq 1/2$, and $0 < r_0 < 1$, $r_0 = \Theta(1/k)$. There exists an integer $J$, functions $\{\Phi_j, \chi_j\}_{j=1}^J$ and a number $C$ such that we can approximate $K(|x - y|)$ for $r_0 \leq |x - y| \leq 1$:

$$K(|x - y|) = \sum_{j=1}^{J} \Phi_j(x)\chi_j(y) + E(x, y)$$

where $|E(x, y)| \leq \epsilon$, and $J \leq C (\log k \max(|\log \epsilon|, \log k))^2$ with $C$ which does not depend on $k$ or $\epsilon$. 
Numerically low-rank $\Rightarrow$ low-rank matrix block

Function

$$K(|x - y|) = \sum_{j=1}^{J} \Phi_j(x) \chi_j(y),$$

$$K(|x_i - y_\ell|) = \sum_{j=1}^{J} \Phi_j(x_i) \chi_j(y_\ell).$$

Matrix $K_{i\ell} = K(|x_i - y_\ell|)$:

$$K = \sum_{j=1}^{J} \vec{\Phi}_j \vec{\chi}_j^* = \Phi \chi^*$$

with $\vec{\Phi}_j$, $\vec{\chi}_j$ the $j^{th}$ columns of matrices $\Phi$, $\chi$.

This is almost the Singular Value Decomposition (SVD) of matrix $K_{i\ell}$. 
Proof: $D_{\text{half}}$ numerically low-rank away from singularity

Kernel $K(r) = \frac{ik^2 H_1^{(1)}(kr)}{2kr}$ for uniform half-space DtN map.
Proof: $D_{\text{half}}$ numerically low-rank away from singularity

Kernel $K(r) = \frac{ik^2 H_1^{(1)}(kr)}{2kr}$ for uniform half-space DtN map.

$$
\frac{1}{kr} = \int_0^\infty e^{-krt} dt \approx \int_0^T e^{-krt} dt
$$

with error $\int_T^\infty e^{-krt} dt \leq \epsilon$ for $T = O(\lvert \log \epsilon \rvert)$. 

![Graph showing exponential decay of $e^{-krt}$]
Proof: $D_{\text{half}}$ numerically low-rank away from singularity

Use a Gaussian quadrature

$$\frac{1}{kr} \approx \int_0^T e^{-krt} dt \approx \sum_{j=1}^{n} w_j e^{-krt_j} = \sum_{j=1}^{n} w_j e^{-kx^j} e^{ky^j} \quad x > y$$
Proof: $D_{\text{half}}$ numerically low-rank away from singularity

Use a Gaussian quadrature

\[
\frac{1}{kr} \approx \int_0^T e^{-krt} dt \approx \sum_{j=1}^n w_j e^{-krt_j} = \sum_{j=1}^n w_j e^{-kxt_j} e^{kyt_j} \quad x > y
\]

but need a dyadic partition of the interval for convergence.
Proof: $D_{\text{half}}$ numerically low-rank away from singularity

- Kernel $K(r) = \frac{i k^2 H_1^{(1)}(kr)}{2kr}$ of uniform half-space DtN map.

- Use Gaussian quadratures for $1/kr$ on dyadic partition of interval:
  
  \[ \log k \text{ subintervals, } |\log \epsilon| \text{ pts each} \]

- Treat integral form of Hankel function same way (Martinsson-Rokhlin 2007).

- Multiply $1/kr$ with $H_1^{(1)}$, total number of quad. pts:
  
  \[ J \approx (\log k |\log \epsilon|)^2. \]
Partitioned low-rank (PLR) matrices

Adaptively, recursively divide blocks of matrix.

Stop when numerical rank $\leq R_{\text{max}}$, with tolerance $\epsilon$. 
Partitioned low-rank (PLR) matrices

Adaptively, recursively divide blocks of matrix.

Stop when numerical rank $\leq R_{\text{max}}$, with tolerance $\epsilon$. 

\[ ? = \text{\begin{figure} \end{figure}} \]
Partitioned low-rank (PLR) matrices

Adaptively, recursively divide blocks of matrix.

Stop when numerical rank $\leq R_{\text{max}}$, with tolerance $\epsilon \Rightarrow \text{“leaf”}$. 
Partitioned low-rank (PLR) matrices

Adaptively, recursively divide blocks of matrix.

Stop when numerical rank $\leq R_{\text{max}}$, with tolerance $\epsilon \Rightarrow \text{“leaf”}$.

Figure: $\frac{N}{R_{\text{max}}} = 8$, weak hierarchical structure.

Figure: $\frac{N}{R_{\text{max}}} = 16$, strong hierarchical structure.

Figure: $\frac{N}{R_{\text{max}}} = 8$, corner PLR structure.
Complexity of compression: PLR matrices

Cost per block $B$ dominated by (randomized) SVD: $O(N B R_{\text{max}}^2)$.

Figure: $\frac{N}{R_{\text{max}}} = 8$, weak h. structure.

Figure: $\frac{N}{R_{\text{max}}} = 16$, strong h. structure.

Figure: $\frac{N}{R_{\text{max}}} = 8$, corner PLR structure.

Total complexity:

$O(N R_{\text{max}}^2 \log \frac{N}{R_{\text{max}}})$  $O(N R_{\text{max}}^2 \log \frac{N}{R_{\text{max}}})$  $O(N R_{\text{max}}^2)$
Complexity of matrix-vector products: PLR matrices

Cost per leaf $B$: $O(N_B R_{\text{max}})$.

Figure: $\frac{N}{R_{\text{max}}} = 8$, weak h. structure.

Figure: $\frac{N}{R_{\text{max}}} = 16$, strong h. structure.

Figure: $\frac{N}{R_{\text{max}}} = 8$, corner PLR structure.

Total complexity:

$O(N R_{\text{max}} \log \frac{N}{R_{\text{max}}})$  $O(N R_{\text{max}} \log \frac{N}{R_{\text{max}}})$  $O(N R_{\text{max}})$
Results of PLR compression after probing

- In general, ask for PLR tolerance

\[ \epsilon = \frac{1}{25} \frac{\|D - \tilde{D}\|_F}{\|D\|_F}. \]

Table: \( c \equiv 1 \)

<table>
<thead>
<tr>
<th>( R_{\text{max}} )</th>
<th>( \epsilon )</th>
<th>( \frac{|D - \bar{D}|_F}{|D|_F} )</th>
<th>( \frac{|u - \bar{u}|_F}{|u|_F} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.6850e − 02</td>
<td>4.2126e − 01</td>
<td>6.5938e − 01</td>
</tr>
<tr>
<td>2</td>
<td>1.6802e − 03</td>
<td>4.2004e − 02</td>
<td>7.3655e − 02</td>
</tr>
<tr>
<td>2</td>
<td>5.0068e − 05</td>
<td>1.2517e − 03</td>
<td>2.4232e − 03</td>
</tr>
<tr>
<td>4</td>
<td>4.4840e − 06</td>
<td>1.1210e − 04</td>
<td>4.0003e − 04</td>
</tr>
<tr>
<td>8</td>
<td>4.3176e − 07</td>
<td>1.0794e − 05</td>
<td>1.4305e − 05</td>
</tr>
<tr>
<td>8</td>
<td>2.6198e − 08</td>
<td>6.5496e − 07</td>
<td>2.1741e − 06</td>
</tr>
</tbody>
</table>
Results of PLR compression after probing

**Table:** \( c \) is the diagonal fault.

<table>
<thead>
<tr>
<th>( R_{\text{max}} )</th>
<th>( \epsilon )</th>
<th>( | D - \overline{D} |_F / | D |_F )</th>
<th>( | u - \overline{u} |_F / | u |_F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>( 5.7124e - 03 )</td>
<td>( 1.4281e - 01 )</td>
<td>( 5.3553e - 01 )</td>
</tr>
<tr>
<td>2</td>
<td>( 7.6432e - 04 )</td>
<td>( 1.9108e - 02 )</td>
<td>( 7.8969e - 02 )</td>
</tr>
<tr>
<td>4</td>
<td>( 1.0241e - 04 )</td>
<td>( 2.5602e - 03 )</td>
<td>( 8.7235e - 03 )</td>
</tr>
</tbody>
</table>

**Table:** \( c \) is the periodic medium.

<table>
<thead>
<tr>
<th>( R_{\text{max}} )</th>
<th>( \epsilon )</th>
<th>( | D - \overline{D} |_F / | D |_F )</th>
<th>( | u - \overline{u} |_F / | u |_F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>( 5.1868e - 03 )</td>
<td>( 1.2967e - 01 )</td>
<td>( 2.1162e - 01 )</td>
</tr>
<tr>
<td>2</td>
<td>( 1.2242e - 03 )</td>
<td>( 3.0606e - 02 )</td>
<td>( 5.9562e - 02 )</td>
</tr>
<tr>
<td>8</td>
<td>( 3.6273e - 04 )</td>
<td>( 9.0682e - 03 )</td>
<td>( 2.6485e - 02 )</td>
</tr>
</tbody>
</table>
PLR compression after probing

\[ D \xrightarrow{\text{probing}} \tilde{D} \xrightarrow{\text{PLR compression}} \overline{D} \]

- Small \( R_{\text{max}} \) needed in practice, \( R_{\text{max}} \leq 8 \).
- Nearly linear matrix-vector product even in heterogeneous media.
- PLR compression is very flexible, “one size fits all”.

Bélanger-Rioux (Harvard)  Compressed ABC for Helmholtz

3/15/15   28 / 31
Conclusion – so far

- Insights from half-space DtN map to expand then compress exterior DtN map
- Handful of PDE solves $\Rightarrow$ exterior DtN map to good accuracy $\Rightarrow$ HE solution to good accuracy
- Compressed DtN map $\Rightarrow$ fast matrix-vector products
Conclusion – complexities

Constructing $D$:
- Matrix probing expansion, assuming fast solver.
- PLR compression.

Complexity:
- $\sim q(N + w)^2$, $q \leq 50$.
- $\sim NR_{\text{max}}^2$, $R_{\text{max}} \leq 8$.

Applying $D$:
- Dense matrix-vector product.
- PLR matrix-vector product.

Complexity:
- $\sim 16N^2$.
- $\sim 4NR_{\text{max}} \log \frac{N}{R_{\text{max}}} + 12NR_{\text{max}}$. 

Bélanger-Rioux (Harvard)  Compressed ABC for Helmholtz  3/15/15  30 / 31
Conclusion – outlook

- 3D
- Probe (and compress) entire structure of the Green’s function?
- Integrate in Domain Decomposition Methods