Sweeping Preconditioners for the Helmholtz Equation

Lexing Ying
Mathematics and ICES, Univ of Texas at Austin

Joint work with Bjorn Engquist
Supported by NSF and Sloan Foundation
Problem statement

- Helmholtz equation in 2D and 3D

\[ \Delta u(x) + \frac{\omega^2}{c^2(x)} u(x) = f(x) \text{ in } D = (0,1)^d \]

with Sommerfeld boundary condition
\( \omega : \text{frequency, large} \)
\( c(x) : \text{velocity field } O(1) \)

- Two main difficulties
  - Large number of unknowns
    - \( O(1) \) pts per wavelength \( \lambda \)
    - \( N = n^d = O(\omega^d) \)
  - The system is indefinite
    - Standard multiscale methods fail
Previous work  

Direct methods: multifrontal methods [George 73], [Duff-Reid 83]
- 2D: cost=O(N^{3/2}), default choice
- 3D: cost=O(N^2), too expensive

Iterative methods
- Multigrid-type methods
  - Wave-ray method [Brandt-Livshits 97], const coeff case
- Domain decomposition methods: large # of iterations
- Shifted Laplacian preconditioner: # of iters ≈ O(ω)
- ILU preconditioner: # of iters ≈ O(ω)

This talk
- New preconditioners: # of iters is indep of ω
- Essentially linear O(N) solver for the Helmholtz equation
Outline

- Sweeping factorization
- Approach 1
- Approach 2
- Conclusion
Perfectly matched layers

- Consider 2D first. Domain D=(0,1)^2
- PML for Sommerfeld B.C.
  - Analytically continue soln u(x)
  - Stretch coordinate to get decay
  - Truncate with zero B.C.

\[ \sigma(t) : \frac{1}{0 \eta} \times 1 \]

\[ s_1(x_1) = \left(1 + i \frac{\sigma(x_1)}{\omega} \right)^{-1} \]
\[ s_2(x_2) = \left(1 + i \frac{\sigma(x_2)}{\omega} \right)^{-1} \]
\[ \partial_1 \Rightarrow s_1(x_1) \cdot \partial_1 \]
\[ \partial_2 \Rightarrow s_2(x_2) \cdot \partial_2 \]

\[ \Delta u + \frac{\omega^2}{c^2(x)} u = f \quad \Rightarrow \quad \left( \partial_1 \left( \frac{s_1}{s_2} \partial_1 \right) + \partial_2 \left( \frac{s_2}{s_1} \partial_2 \right) + \frac{\omega^2}{s_1 s_2 \cdot c^2(x)} \right) u = f \]
Discretization

- Constant # of pts per wavelength
- # pts in each dim = \( n = O(\omega) \)
- # total pts \( N = O(n^2) = O(\omega^2) \)

- Order grid points row by row
- \( P_m \) = points in the m-th row
- \( f_m \): RHS at \( P_m \)
- \( u_m \): unknowns at \( P_m \)

- Discretize with 5-pt stencil for Laplacian

\[
Au = f \quad \iff \quad \begin{pmatrix}
A_{1,1} & A_{1,2} & & & \\
A_{2,1} & A_{2,2} & & & \\
& & \ddots & & \\
& & & A_{n-1,n} & \\
& & & & A_{n,n}
\end{pmatrix}
\begin{pmatrix}
u_1 \\
u_2 \\
\vdots \\
u_{n-1} \\
u_n
\end{pmatrix} =
\begin{pmatrix}
f_1 \\
f_2 \\
\vdots \\
f_{n-1} \\
f_n
\end{pmatrix}
\]
Schur complement

- For 2 by 2 block matrix

\[
\begin{pmatrix}
E & C \\
B & D
\end{pmatrix} = \begin{pmatrix}
I & 0 \\
BE^{-1} & I
\end{pmatrix} \begin{pmatrix}
E & 0 \\
0 & D - BE^{-1}C
\end{pmatrix} \begin{pmatrix}
I & E^{-1}C
\end{pmatrix}
\]

- Decouple (or eliminate) the unknowns
- Decouple the first layer $P_1$ from the rest unknowns

\[
A = \begin{pmatrix}
A_{1,1} & A_{1,2} & \cdots & \cdots & A_{1,n-1} & A_{1,n} \\
A_{2,1} & A_{2,2} & \cdots & \cdots & A_{n-1,1} & A_{2,n} \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
A_{n,n-1} & A_{n,n} & \cdots & \cdots & A_{n,n}
\end{pmatrix} = L_1 \begin{pmatrix}
S_1 \\
S_2 \\
\vdots \\
\vdots \\
S_n
\end{pmatrix} L_1^t,
\]

\[
L_1 = \begin{pmatrix}
I \\
A_{2,1} S_1^{-1} I \\
\vdots \\
I
\end{pmatrix}
\]

\[
S_1 = A_{1,1}, \quad S_2 = A_{2,2} - A_{2,1} S_1^{-1} A_{1,2}
\]
Sweeping factorization

- Eliminate unknowns layer by layer from the bottom layer

\[ A = L_1 L_2 \cdots L_{n-1} \begin{pmatrix} S_1 & & \\ & S_2 & \cdots \\ & & S_n \end{pmatrix} L_{t_{n-1}}^t \cdots L_{2t}^t L_{t_1}^t, \quad L_m = \begin{pmatrix} I \\ & I \\ & & A_{m+1,m} S_m^{-1} \cdots \end{pmatrix} \]

\[ S_1 = A_{1,1}, \quad S_m = A_{m,m} - A_{m,m-1} S_{m-1}^{-1} A_{m-1,m} \]

- Solve \( Au = f \) by inverting factorization

\[ u = (L_{t_1}^t)^{-1} \cdots (L_{t_{n-1}}^t)^{-1} \begin{pmatrix} S_1^{-1} \\ & S_2^{-1} \\ & & \cdots \\ & & & S_n^{-1} \end{pmatrix} L_{n-1}^{-t} \cdots L_1^{-t} \cdot f \]

Define \( T_m = S_m^{-1} \) (dense nxn matrices)
Algorithms

- Construction of the sweeping factorization
  - \( S_1 = A_{1,1} \) and \( T_1 = (S_1)^{-1} \)
  - For \( m=2,\ldots,n \)
    - \( S_m = A_{m,m} - A_{m,m-1}T_{m-1}A_{m-1,m} \) and \( T_m = (S_m)^{-1} \)

- Application of \( u = A^{-1} f \)
  - For \( m=1,\ldots,n \)
    - \( u_m = f_m \)
  - For \( m=1,\ldots,n-1 \)
    - \( u_{m+1} = u_{m+1} - A_{m+1,m}(T_mu_m) \)
  - For \( m=1,\ldots,n \)
    - \( u_m = T_mu_m \)
  - For \( m=n-1,\ldots,1 \)
    - \( u_m = u_m - T_m(A_{m,m+1}u_{m+1}) \)

Cost = \( O(n^4) = O(N^2) \)

Cost = \( O(n^3) = O(N^{3/2}) \)

Slower than multifrontal \( O(N^{3/2}) \) and \( O(N \log N) \). Computation of \( T_m \) is the bottleneck.
What is $T_m = (S_m)^{-1}$?

- Restrict to the first $m$ by $m$ blocks

$$
\begin{pmatrix}
A_{1,1} & A_{1,2} \\
A_{2,1} & A_{2,2} & \ddots & \ddots \\
\vdots & \ddots & \ddots & A_{m-1,m} \\
A_{m,m-1} & \ddots & \ddots & A_{m,m}
\end{pmatrix}
= 
L_1 L_2 \cdots L_{m-1}
\begin{pmatrix}
S_1 \\
S_2 \\
\vdots \\
S_{m-1} \\
S_m
\end{pmatrix}
$$

Half-plane Helmholtz with 0 BC at $x_2 = (m+1)h$

$$
\begin{pmatrix}
G_{1,1} & G_{1,2} & \cdots & G_{1,m} \\
G_{2,1} & G_{2,2} & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots \\
G_{m,1} & \cdots & G_{m,m-1} & G_{m,m}
\end{pmatrix}
= 
(L_1^t)^{-1} \cdots (L_{m-1}^t)^{-1}
\begin{pmatrix}
S_1^{-1} \\
S_2^{-1} \\
\vdots \\
S_{m-1}^{-1} \\
S_m^{-1}
\end{pmatrix}
$$

Preserved under product
What is $T_m = (S_m)^{-1}$?

- $T_m = (S_m)^{-1}$ is the discrete half-space Green’s function with zero BC at $x_2 = (m+1)h$, restricted to the line $x_2 = mh$

Goal: approximate and manipulate $T_m$ efficiently
Outline

- Sweeping factorization
- Approach 1
- Approach 2
- Conclusion
Approach 1

- Central idea
  - $T_m$ and $S_m$ are compressible with low-rank off-diag blocks

- Theorem [Engquist-Y. 10] For const $c(x)$, given $\varepsilon > 0$, each off-diag block of $T_m$ has an $\varepsilon$-accurate factorization with rank $O(\log(\omega) \log^2(1/\varepsilon))$.

\[
\frac{\omega}{2\pi} = 32 \\
n = 256 \\
m = 128
\]
Hierarchical matrices

- Operational matrix algebra
  - Uses ideas from tree-codes and FMM (Greengard-Rokhlin)
  - H-matrices, [Hackbusch et al]
  - HSS matrices, [Chandrasekaran-Gu-Li-Xia]
  - Direct solver [Martinsson-Rokhlin]

\[ P_m = J_0^{1} \]
\[ J_1^{1} \]
\[ J_2^{1} J_2^{2} J_3^{2} J_4^{2} \]

- Dense
- Factorized, low rank
Hierarchical matrices

- Matrix representation
  - $O(n \log n)$ cost
- Matrix-vector multiplication: $hmatvec(G,f)$
  - $O(n \log n)$ cost
- Matrix addition and subtraction: $hadd(G,H)$ and $hsub(G,H)$
  - $O(n \log n)$ cost
- Matrix multiplication and inversion: $hmul(G,H)$ and $hinv(G)$
  - $O(n \log^2 n)$ cost
- Multiplication with a diagonal matrix: $hdiagmul(G,D)$
  - $O(n \log n)$ cost

- Use this to represent and manipulate $T_m$ and $S_m$

- Related work: 1D spectral projection [Sandberg-Beylkin 09]
Algorithms using hierarchical matrix rep.

- Construction of the sweeping factorization
  - $S_1 = A_{1,1}$ and $T_1 = \text{hinv}(S_1)$
  - For $m=2,\ldots,n$
    - $S_m = \text{hsub}(A_{m,m}, \text{hdiagmul}(A_{m,m-1}, \text{hdiagmul}(T_{m-1}, A_{m-1,m})))$
    - $T_m = \text{hinv}(S_m)$
  
- Application of $u \approx A^{-1} f$
  - For $m=1,2,\ldots,n$
    - $u_m = f_m$
  - For $m=1,\ldots,n-1$
    - $u_{m+1} = u_{m+1} - A_{m+1,m} \text{hmatvec}(T_m, u_m)$
  - For $m=1,\ldots,n$
    - $u_m = \text{hmatvec}(T_m, u_m)$
  - For $m=n-1,\ldots,1$
    - $u_m = u_m - \text{hmatvec}(T_m, A_{m,m+1}u_{m+1})$

Cost = $O(n^2 \log^2 n) = O(N \log^2 N)$
Sweeping preconditioner

- Last algorithm defines an accurate approximate inverse
  \[ M : f = \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} \Rightarrow u = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \]

- Solve the preconditioned system using GMRES or TFQMR
  \[ MAu = Mf \]
  - The \# of iterations is very small and indep of \( \omega \)
  - Linear solver for the Helmholtz equation \( Au = f \)
Remarks

- Start sweeping from a PML side
  - Multifrontal method would fail ($T_m$ no longer compressible)
- How about variable $c(x)$?
  - Turning rays. Addressed by multiple sweeps (or iterations)
- Different sweeping directions
  - Result different # of iterations
  - Optimality = avoiding turning rays
- Works in general for systems with no trapped rays
Other boundary conditions

- Depth extrapolation in seismology

- Mixed boundary condition: Dirichlet plus absorbing

- Absorbing boundary condition (ABC), [Engquist-Majda 77]
Example: converging lens

<table>
<thead>
<tr>
<th>$\omega/(2\pi)$</th>
<th>$q$</th>
<th>$N = n^2$</th>
<th>$R$</th>
<th>$T_{\text{setup}}$</th>
<th>\textbf{Test 1}</th>
<th>\textbf{Test 2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>8</td>
<td>$128^2$</td>
<td>2</td>
<td>6.50e-01</td>
<td>2</td>
<td>5.00e-02</td>
</tr>
<tr>
<td>32</td>
<td>8</td>
<td>$256^2$</td>
<td>2</td>
<td>5.05e+00</td>
<td>2</td>
<td>2.50e-01</td>
</tr>
<tr>
<td>64</td>
<td>8</td>
<td>$512^2$</td>
<td>2</td>
<td>3.44e+01</td>
<td>3</td>
<td>1.45e+00</td>
</tr>
<tr>
<td>128</td>
<td>8</td>
<td>$1024^2$</td>
<td>2</td>
<td>2.16e+02</td>
<td>3</td>
<td>7.37e+00</td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>$2048^2$</td>
<td>2</td>
<td>1.24e+03</td>
<td>3</td>
<td>3.31e+01</td>
</tr>
</tbody>
</table>
Example: random media

<table>
<thead>
<tr>
<th>$\omega/(2\pi)$</th>
<th>$q$</th>
<th>$N = n^2$</th>
<th>$R$</th>
<th>$T_{\text{setup}}$</th>
<th>$N_{\text{iter}}$</th>
<th>$T_{\text{solve}}$</th>
<th>$N_{\text{iter}}$</th>
<th>$T_{\text{solve}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>8</td>
<td>$128^2$</td>
<td>2</td>
<td>6.50e-01</td>
<td>2</td>
<td>5.00e-02</td>
<td>2</td>
<td>5.00e-02</td>
</tr>
<tr>
<td>32</td>
<td>8</td>
<td>$256^2$</td>
<td>2</td>
<td>5.10e+00</td>
<td>2</td>
<td>2.50e-01</td>
<td>3</td>
<td>3.00e-01</td>
</tr>
<tr>
<td>64</td>
<td>8</td>
<td>$512^2$</td>
<td>2</td>
<td>3.48e+01</td>
<td>3</td>
<td>1.49e+00</td>
<td>3</td>
<td>1.48e+00</td>
</tr>
<tr>
<td>128</td>
<td>8</td>
<td>$1024^2$</td>
<td>2</td>
<td>2.16e+02</td>
<td>4</td>
<td>8.99e+00</td>
<td>3</td>
<td>7.37e+00</td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>$2048^2$</td>
<td>2</td>
<td>1.26e+03</td>
<td>5</td>
<td>4.64e+01</td>
<td>3</td>
<td>3.25e+01</td>
</tr>
</tbody>
</table>
3D case

- $T_m$ is the discrete half-space Green’s function with zero BC at $x_3=(m+1)h$, restricted to the plane $x_3=mh$

- $\varepsilon$-rank of off-diag block of $T_m \approx \sqrt{\text{block size}}$

- Since we aim for a preconditioner, we can still approximate $T_m$ with hierarchical matrix algebra
Example: wave guide

<table>
<thead>
<tr>
<th>$\omega/(2\pi)$</th>
<th>$q$</th>
<th>$N = n^3$</th>
<th>$R$</th>
<th>$T_{\text{setup}}$</th>
<th>$N_{\text{iter}}$</th>
<th>$T_{\text{solve}}$</th>
<th>$N_{\text{iter}}$</th>
<th>$T_{\text{solve}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>8</td>
<td>$40^3$</td>
<td>2</td>
<td>$8.95e+01$</td>
<td>3</td>
<td>$7.10e-01$</td>
<td>3</td>
<td>$7.00e-01$</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>$80^3$</td>
<td>3</td>
<td>$2.35e+03$</td>
<td>5</td>
<td>$1.40e+01$</td>
<td>5</td>
<td>$9.38e+00$</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>$160^3$</td>
<td>4</td>
<td>$4.73e+04$</td>
<td>4</td>
<td>$1.38e+02$</td>
<td>4</td>
<td>$1.34e+02$</td>
</tr>
</tbody>
</table>
Outline

- Sweeping factorization
- Approach 1
- Approach 2
- Conclusion
Approach 2

- \( T_m = (S_m)^{-1} \) is the discrete half-space Green’s function with zero BC at \( x_2 = (m+1)h \), restricted to the line \( x_2 = mh \)

- \( T_m: g_m \rightarrow v_m \)
  - Load external force \( g_m \) at \( x_2 = mh \)
  - Extract solution field \( v_m \) at \( x_2 = mh \)
  - Involve pts from prev \((m-1)\) layers. Costly

- Central idea
  - But domain of interest for this step is only \( \{ x_2 = mh \} \). Use PML.
  - Push the PML from \( x_2 = 0 \) very close (or right next) to \( x_2 = mh \)
Moving PML

- $G_m$: subgrid of moving PML
- $H_m$: discrete operator on $G_m$

\[
H_m \begin{pmatrix} * \\ * \\ \vdots \\ v_m \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ g_m \end{pmatrix}
\]

- Solving this gives an approx. to $T_m$: $g_m \rightarrow v_m$
  - Replace an infinite (or long) continued fraction with a short one

- $G_m$ is a quasi-1D grid
  - $H_m$ is a *banded* matrix with small band
  - LU with optimal ordering ($x_2$ first)
    - Construction time = $O(n)$; Solution time = $O(n)$
Algorithms using moving PML

- Construction of the sweeping factorization
  - For $m=1,...,n$
    - Compute LU decomp for $H_m$
    - Cost $= O(n^2) = O(N)$

- Application of $u \approx A^{-1} f$
  - For $m=1,2,...,n$
    - $u_m = f_m$
  - For $m=1,...,n-1$
    - $u_{m+1} = u_{m+1} - A_{m+1,m} (T_m u_m)$ using LU decomp of $H_m$
  - For $m=1,...,n$
    - $u_m = T_m u_m$ using LU decomp of $H_m$
  - For $m=n-1,...,1$
    - $u_m = u_m - T_m (A_{m,m+1} u_{m+1})$ using LU decomp of $H_m$
    - Cost $= O(n^2) = O(N)$
Sweeping preconditioner

- For better stability, build factorization for a slightly different eqn
  \[
  \Delta u(x) + \frac{(\omega + i\alpha)^2}{c^2(x)} u(x) = f(x), \quad \alpha = O(1) \quad \Rightarrow \quad A_\alpha u = f
  \]

- Applying algo to $A_\alpha u = f$ gives preconditioner for $Au = f$ as $\alpha$ is small
  \[
  M_\alpha : f = \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} \Rightarrow u = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}
  \]

- Solve the preconditioned system using GMRES or TFQMR
  \[
  M_\alpha Au = M_\alpha f
  \]

- # of iterations is very small and indep of frequency
- Linear solver for the Helmholtz equation $Au = f$
Example: converging lens

<table>
<thead>
<tr>
<th>$\omega/(2\pi)$</th>
<th>$q$</th>
<th>$N = n^2$</th>
<th>$T_{\text{setup}}$</th>
<th>$N_{\text{iter}}$</th>
<th>$T_{\text{solve}}$</th>
<th>$N_{\text{iter}}$</th>
<th>$T_{\text{solve}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>8</td>
<td>$127^2$</td>
<td>2.86e-01</td>
<td>14</td>
<td>4.73e-01</td>
<td>15</td>
<td>3.81e-01</td>
</tr>
<tr>
<td>32</td>
<td>8</td>
<td>$255^2$</td>
<td>8.95e-01</td>
<td>15</td>
<td>1.59e+00</td>
<td>15</td>
<td>1.57e+00</td>
</tr>
<tr>
<td>64</td>
<td>8</td>
<td>$511^2$</td>
<td>3.78e+00</td>
<td>15</td>
<td>7.14e+00</td>
<td>15</td>
<td>7.12e+00</td>
</tr>
<tr>
<td>128</td>
<td>8</td>
<td>$1023^2$</td>
<td>1.61e+01</td>
<td>15</td>
<td>2.90e+01</td>
<td>13</td>
<td>2.54e+01</td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>$2047^2$</td>
<td>6.85e+01</td>
<td>16</td>
<td>1.44e+02</td>
<td>11</td>
<td>9.42e+01</td>
</tr>
</tbody>
</table>
Example: random media

<table>
<thead>
<tr>
<th>$\omega/(2\pi)$</th>
<th>$q$</th>
<th>$N = n^2$</th>
<th>$T_{\text{setup}}$</th>
<th>$N_{\text{iter}}$</th>
<th>$T_{\text{solve}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>8</td>
<td>$127^2$</td>
<td>3.56e-01</td>
<td>18</td>
<td>4.91e-01</td>
</tr>
<tr>
<td>32</td>
<td>8</td>
<td>$255^2$</td>
<td>9.31e-01</td>
<td>18</td>
<td>2.42e+00</td>
</tr>
<tr>
<td>64</td>
<td>8</td>
<td>$511^2$</td>
<td>3.76e+00</td>
<td>17</td>
<td>8.66e+00</td>
</tr>
<tr>
<td>128</td>
<td>8</td>
<td>$1023^2$</td>
<td>1.60e+01</td>
<td>19</td>
<td>3.90e+01</td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>$2047^2$</td>
<td>6.82e+01</td>
<td>17</td>
<td>1.54e+02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test 1</th>
<th>Test 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{\text{iter}}$</td>
<td>$T_{\text{solve}}$</td>
</tr>
<tr>
<td>18</td>
<td>4.91e-01</td>
</tr>
<tr>
<td>18</td>
<td>2.42e+00</td>
</tr>
<tr>
<td>17</td>
<td>8.66e+00</td>
</tr>
<tr>
<td>19</td>
<td>3.90e+01</td>
</tr>
<tr>
<td>17</td>
<td>1.54e+02</td>
</tr>
</tbody>
</table>
Example: scattering problem

\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
\(\omega/(2\pi)\) & \(q\) & \(N\) & \(T_{\text{setup}}\) & Incident field 1 & & Incident field 2 \\
\hline
16 & 8 & 45 \times 403 & 6.11e-01 & \multicolumn{2}{c|}{N_{\text{iter}} \quad T_{\text{solve}}} \quad \multicolumn{2}{c|}{N_{\text{iter}} \quad T_{\text{solve}}} \\
32 & 8 & 90 \times 805 & 2.61e+00 & 7 & 3.61e-01 & 7 & 2.25e-01 \\
64 & 8 & 180 \times 1609 & 1.17e+01 & 7 & 1.11e+00 & 7 & 1.11e+00 \\
128 & 8 & 359 \times 3217 & 4.95e+01 & 7 & 4.92e+00 & 7 & 4.90e+00 \\
256 & 8 & 717 \times 6434 & 1.99e+02 & 7 & 2.10e+01 & 7 & 2.06e+01 \\
\hline
\end{tabular}
\end{table}
3D case

- \( T_m = (S_m)^{-1} \) is the discrete half-space Green’s function with zero BC at \( x_3 = (m+1)h \), restricted to \( x_3 = mh \)

- \( G_m \): the subgrid of moving PML
- \( H_m \): the discrete operator on \( G_m \)

\[
H_m \begin{pmatrix} * \\ * \\ \vdots \\ v_m \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ g_m \end{pmatrix}
\]

- Solving this gives an approx. to \( T_m \)

- \( G_m \) is a quasi-2D grid
  - Apply **multifrontal method** to get \( LDL^t \) factorization of \( H_m \)
  - Construction time = \( O(n^3) \); Solution time = \( O(n^2 \log n) \)
Algorithms using moving PML

- Construction of the sweeping factorization
  - For $m=1,...,n$
    - Compute multifrontal decomp for $H_m$

- Application of $u \approx A^{-1} f$
  - For $m=1,2,...,n$
    - $u_m = f_m$
  - For $m=1,...,n-1$
    - $u_{m+1} = u_{m+1} - A_{m+1,m}(T_m u_m)$ using multifrontal decomp of $H_m$
  - For $m=1,...,n$
    - $u_m = T_m u_m$ using multifrontal decomp of $H_m$
  - For $m=n-1,...,1$
    - $u_m = u_m - T_m(A_{m,m+1} u_{m+1})$ using multifrontal decomp of $H_m$

Cost = $O(n^4) = O(N^{4/3})$

Cost = $O(n^3 \log n) = O(N \log N)$

(N = $n^3$)
Example: wave guide

<table>
<thead>
<tr>
<th>$\omega/(2\pi)$</th>
<th>$q$</th>
<th>$N = n^3$</th>
<th>$T_{\text{setup}}$</th>
<th>$T_{\text{solve}}$</th>
<th>$N_{\text{iter}}$</th>
<th>$T_{\text{solve}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>8</td>
<td>$39^3$</td>
<td>4.83e+00</td>
<td>12</td>
<td>5.14e+00</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>$79^3$</td>
<td>6.76e+01</td>
<td>13</td>
<td>5.70e+01</td>
<td>12</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>$159^3$</td>
<td>8.24e+01</td>
<td>14</td>
<td>6.32e+02</td>
<td>11</td>
</tr>
</tbody>
</table>
Comparison

- Both based on sweeping idea and analytic interpretation of $T_m$
- Approach 1: Hierarchical matrix representation of $T_m$
  - Longer construction time
  - Short application time
  - A couple iterations
- Approach 2: Moving PML approximation of $T_m$
  - Shorter construction time
  - Short application time
  - Slightly more iterations (due to $\alpha$)
  - Flexible (unstructured & adaptive grids, curvilinear sweeping)

- # of iterations indep of $\omega$
- Essentially linear complexity for solving the Helmholtz equation
Conclusions

- Sweeping starting from a PML layer so rays do not get trapped
- Efficient representation of the integral operators $T_m$
- Dimension reduction
- A new paradigm for constructing preconditioners?
  - “Right” elimination order for LU
  - Efficient representation for the Schur complement matrices

- Preprints at http://www.math.utexas.edu/~lexing
  - Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation (arXiv:1007.4290)
  - Sweeping preconditioner for the Helmholtz equation: Moving perfectly matched layer (arXiv:1007.4291)

Thank you for your attention