Solving the Helmholtz equation via row-projections
Tristan van Leeuwen (Univ. of BC, Canada)
Dan Gordon (Univ. of Haifa, Israel)
Rachel Gordon (Technion, Israel)
Felix J. Herrmann (Univ. of BC, Canada)
Modelling engine for 3D Frequency-domain FWI:

- work with few sources/frequencies at each iteration
- flexibility in type of equation
- robust
- parallel
3D Helmholtz equation:

- *large, sparse, indefinite system*
- *direct factorization not feasible*
- *`standard’ preconditioners often fail*
- *successful preconditioners often tailored to specific wave equation*

[Ernst & Martin, 2011]
fast, complicated, ...

VS.

simple, robust, ...
Overview

- Kaczmarz preconditioning
- Examples
- Parallelization
- 3D Benchmark
- Inversion
- Conclusions
The Kaczmarz method solves a system $Ax = b$ by successive row projections

$$x := x + \frac{\lambda_i}{\|a_i\|_2^2} (b_i - a_i^T x) a_i,$$

with relaxation parameter $0 < \lambda_i < 2$
The diagram illustrates the behavior of \( \lambda \) for different values:

- \( \lambda < 1 \)
- \( \lambda = 1 \)
- \( \lambda > 1 \)
Kaczmarz

rewrite:

\[ x := \left( I - \frac{\lambda_i}{\|a_i\|_2^2} a_i a_i^T \right) x + \frac{\lambda_i}{\|a_i\|_2^2} b_i a_i \]

a double sweep yields

\[ x := \left( Q_1 Q_2 \ldots Q_n Q_n \ldots Q_1 \right) x + \left( \ldots \right) b \]
Find a fixed point by solving

$$(I - Q)x = Rb$$

where $I - Q$ is symmetric and positive semidefinite, so we can use CG (CGMN).

[Bjork & Elfving, '79]
Kaczmarz

We never form the matrix explicitly, but compute its action:

Algorithm 1 DKSWP \((A, x, b, \lambda) = Qx + Rb\)

\text{forward sweep}
\begin{align*}
\text{for } i = 1 \text{ to } n \text{ do} \\
x := x + \lambda(b_i - a_i^T x)a_i/||a_i||_2^2 \\
\text{end for}
\end{align*}

\text{backward sweep}
\begin{align*}
\text{for } i = n \text{ to } 1 \text{ do} \\
x := x + \lambda(b_i - a_i^T x)a_i/||a_i||_2^2 \\
\text{end for}
\end{align*}

\text{return } x
Kaczmarz

- low complexity
- low memory (same as original matrix)
- no setup time
1D results

1D profile, varying k, 10 p/wavelength
1D results

eigenvalues  # of CG iterations

---

[Graph showing eigenvalues and number of CG iterations for different matrices and weighting factors.]
2D results

Marmousi, 304 x 1100, f=20, h=10

- CG + Kaczmarz (CGMN)
- BiCGstab + ILU(0)
- SQMR + ML [Bollhofer et al, '08]
2D results

solve \( Ar = 0 \) starting from random vector

\[
r_1 = (I - M^{-1}A)r_0
\]

ILU(0)  

ML  

Kaczmarz
2D results

<table>
<thead>
<tr>
<th>Method</th>
<th>Iterations</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG + Kaczmarz</td>
<td>5542</td>
<td>603</td>
</tr>
<tr>
<td>BiCGstab + ILU(0)</td>
<td>div.</td>
<td>div.</td>
</tr>
<tr>
<td>SQMR + ML</td>
<td>514</td>
<td>379</td>
</tr>
</tbody>
</table>
Parallelization

- divide domain in blocks
- Kaczmarz sweeps on blocks are done in parallel (CARP)
- average boundary elements between each sweep
- convergence guaranteed

[Gordon & Gordon, ’10]
SEG/EAGE salt

7-point stencil, ABC
SEG/EAGE salt

\begin{table}[h]
\begin{center}
\begin{tabular}{ccc}
\hline
\textbf{f} & \textbf{h} & \textbf{iterations} \\
\hline
1.25 & 160 & 310 \\
2.5 & 80 & 510 \\
5 & 40 & 760 \\
10 & 20 & 1780 \\
\hline
\end{tabular}
\end{center}
\end{table}

on 1 processor, $\epsilon = 10^{-4}$

on 12 processors
SEG/EAGE salt

grid: 105 x 338 x 338, h=40, f=5, $\epsilon = 10^{-4}$

<table>
<thead>
<tr>
<th>np</th>
<th>iter</th>
<th>time (s)</th>
<th>efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>621</td>
<td>4444.90</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>619</td>
<td>3091.10</td>
<td>0.72</td>
</tr>
<tr>
<td>4</td>
<td>593</td>
<td>1335.00</td>
<td>0.83</td>
</tr>
<tr>
<td>8</td>
<td>599</td>
<td>737.90</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Overthrust

27 point stencil (2$^{nd}$ order), PML
Overthrust

- Iteration
- Rel. residual

- $f=1.5$, $h=200$
- $f=3$, $h=100$
- $f=6$, $h=50$

on 1 processor
Overthrust

grid: 47x201x201, h=100, f=3 Hz, $\epsilon = 10^{-4}$

<table>
<thead>
<tr>
<th>np</th>
<th>iter</th>
<th>time</th>
<th>efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>659</td>
<td>20785.40</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>657</td>
<td>11306.90</td>
<td>0.92</td>
</tr>
<tr>
<td>4</td>
<td>596</td>
<td>4882.50</td>
<td>0.96</td>
</tr>
<tr>
<td>8</td>
<td>603</td>
<td>3960.10</td>
<td>0.60</td>
</tr>
</tbody>
</table>
Inversion

Camembert model in 2D...
Inversion

EDAM model in 3D!
Inversion

transmission setup, 9 sources, 3 frequencies
Conclusions

• simple, robust and generic preconditioner
• no overhead, cheap to apply
• easy to parallelize
Future plans

- efficient implementation of sweeps using multi-threading
- investigate BlockCG
- incorporate in inversion
- high-order schemes
This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (22R81254) and the Collaborative Research and Development Grant DNOISE II (375142-08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BP, Chevron, ConocoPhillips, Petrobras, Total SA, and WesternGeco.
References


