Implementation of the BiCGSTAB method for the Helmholtz Equation on a Maxeler Data Flow Machine

Delft University of Technology

Onno Leon Meijers
August 25, 2017
Outline

1. Concepts
2. Data Flow
3. Results
4. Conclusions
1 Concepts

2 Data Flow

3 Results

4 Conclusions
What is under the surface?
Ground research process
calculation process

Helmholtz equation:

\[ A_{k,\alpha} u(x) := -\Delta u(x) - (1 - \alpha i)k^2(x)u(x) = g(x) \]

\( u \) is the wave function and we have measurements.  
\( g \) is the source and is known.  
\( k \) is the wave number and is unknown.  
\( \alpha \) is the damping factor and is set.
calculation process

Helmholtz equation:

\[ A_{k,\alpha}u(x) := -\Delta u(x) - (1 - \alpha i)k^2(x)u(x) = g(x) \]

\( u \) is the wave function and we have measurements.
\( g \) is the source and is known.
\( k \) is the wave number and is unknown.
\( \alpha \) is the damping factor and is set.

\[ A_{k,\alpha}u = g \]

Solve for \( u \).
Algorithm 1 Pseudocode for the BiCGSTAB method

1: \( \mathbf{u} = \mathbf{v} = \mathbf{p} = 0; \ r_0 = r = g = \delta_{x_s,y_s,z_s}; \ \rho_{old} = \alpha = \omega = \rho_{new} = 1; \)
2: \textbf{for } i = 0, 1, 2, \ldots, maxit \ \textbf{do}
3: \ \ \ \beta = \frac{\rho_{new}}{\rho_{old}} \ \alpha \ \omega; \ \rho_{old} = \rho_{new};
4: \ \ \ \mathbf{p} = \mathbf{r} + \beta (\mathbf{p} - \omega \mathbf{v});
5: \ \ \ \mathbf{v} = A \mathbf{p};
6: \ \ \ \alpha = \frac{\rho_{old}}{\langle \mathbf{v}, \mathbf{r}_0 \rangle};
7: \ \ \ \mathbf{s} = \mathbf{r} - \alpha \mathbf{v};
8: \ \ \ \mathbf{t} = A \mathbf{s};
9: \ \ \ \omega = \frac{\langle \mathbf{t}, \mathbf{s} \rangle}{\langle \mathbf{t}, \mathbf{t} \rangle};
10: \ \ \ \mathbf{u} = \mathbf{u} + \alpha \mathbf{p} + \omega \mathbf{s};
11: \ \ \ \mathbf{r} = \mathbf{s} - \omega \mathbf{t};
12: \rho_{new} = \langle \mathbf{r}, \mathbf{r}_0 \rangle;
13: \ \textbf{if } \|\mathbf{r}\|_2 < 10^{-6} \ \textbf{then}
14: \ \ \ \text{quit ;}
15: \ \textbf{end if}
16: \ \textbf{end for}
Maxeler Data Flow Machine
Maxeler Data Flow Machine

CPU Application

SLiC
MaxelerOS

Interconnect

Memory

Fast Memory (FMem)

Dataflow Engine

Kernels

Large Memory (LMem)

Manager
1 Concepts

2 Data Flow

3 Results

4 Conclusions
Input node.

Output node.

Computation nodes.

Constant set by CPU.

Multiplexer (mux).

Counter.

Stream offsets.

Part specifier.
BiCGSTAB method
\[ u = v = p = 0; \]
\[ r_0 = r = g = \delta_{x_s, y_s, z_s}; \]
\[ \rho_{old} = \alpha = \omega = \rho_{new} = 1; \]
for \( i = 0, 1, 2, \ldots, \text{maxit} \)
\[ \beta = \frac{\rho_{new}}{\rho_{old}} \frac{\alpha}{\omega}; \quad \rho_{old} = \rho_{new}; \]
\[ p = r + \beta (p - \omega v); \]
\[ v = Ap; \]
\[ \alpha = \frac{\rho_{old}}{(v, r_0)}; \]
\[ s = r - \alpha v; \]
\[ t = As; \]
\[ \omega = \frac{(t, s)}{(t, t)}; \]
\[ u = u + \alpha p + \omega s; \]
\[ r = s - \omega t; \]
\[ \rho_{new} = (r, r_0); \]
If \( \|r\|_2 \) is small enough) then
quit
BiCGSTAB method

Part 1

\[ u = v = p = 0; \]
\[ r_0 = r = g = \delta x_s, y_s, z_s; \]
\[ \rho_{old} = \alpha = \omega = \rho_{new} = 1; \]
for \( i = 0, 1, 2, \ldots, maxit \) \n\[ \beta = \frac{\rho_{new}}{\rho_{old}} \frac{\alpha}{\omega}; \quad \rho_{old} = \rho_{new}; \]
\[ p = r + \beta(p - \omega v); \]
\[ v = Ap; \]
\[ \alpha = \frac{\rho_{old}}{(v, r_0)}; \]
\[ s = r - \alpha v; \]
\[ t = As; \]
\[ \omega = \frac{(t,s)}{(t,t)}; \]
\[ u = u + \alpha p + \omega s; \]
\[ r = s - \omega t; \]
\[ \rho_{new} = (r, r_0); \]
If \(||r||_2\) is small enough) then
quit
BiCGSTAB method

**Part 2**

\[ u = v = p = 0; \]
\[ r_0 = r = g = \delta x_s, y_s, z_s; \]
\[ \rho_{old} = \alpha = \omega = \rho_{new} = 1; \]
\[ \text{for}(i = 0, 1, 2, \ldots, \text{maxit}) \]
\[ \beta = \frac{\rho_{new}}{\rho_{old}} \frac{\alpha}{\omega}; \quad \rho_{old} = \rho_{new}; \]
\[ p = r + \beta(p - \omega v); \]
\[ v = Ap; \]
\[ \alpha = \frac{\rho_{old}}{(v, r_0)}; \]
\[ s = r - \alpha v; \]
\[ t = As; \]
\[ \omega = \frac{(t, s)}{(t, t)}; \]
\[ u = u + \alpha p + \omega s; \]
\[ r = s - \omega t; \]
\[ \rho_{new} = (r, r_0); \]

If \(|r|_2\) is small enough) then

\[ \text{quit} \]
BiCGSTAB method

**Part 3**

\[
\begin{align*}
u &= v = p = 0; \\
r_0 &= r = g = \delta_{x_s,y_s,z_s}; \\
\rho_{old} &= \alpha = \omega = \rho_{new} = 1; \\
\text{for} (i = 0, 1, 2, \ldots, \text{maxit}) \quad & \beta = \frac{\rho_{new}}{\rho_{old}} \frac{\alpha}{\omega}; \quad \rho_{old} = \rho_{new}; \\
p &= r + \beta(p - \omega v); \\
v &= Ap; \\
\alpha &= \frac{\rho_{old}}{(v,r_0)}; \\
s &= r - \alpha v; \\
t &= As; \\
\omega &= \frac{(t,s)}{(t,t)}; \\
u &= u + \alpha p + \omega s; \\
r &= s - \omega t; \\
\rho_{new} &= (r, r_0); \\
\text{If} (\|r\|_2 \text{ is small enough}) \quad & \text{then} \\
\quad & \text{quit}
\end{align*}
\]

Onno Leon Meijers (TU Delft)
\[ A_{k,\alpha}u := -\Delta u - (1 - \alpha i)k^2u \approx \]

\[
\frac{1}{h^2} \begin{bmatrix}
0 & 1 \\
1 & -4 + (1 - \alpha i)k^2h^2 & 0 \\
0 & 1 & 0
\end{bmatrix} u
\]

grid:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td></td>
</tr>
</tbody>
</table>
Real valued implementation

(a) normal scale.
(b) logarithmic scale.

Figure: Calculation times of 1 real valued BiCGSTAB iteration.
Complex valued problem.

Figure: Calculation times of 1 complex valued BiCGSTAB iteration.
Convergence pattern of BiCGSTAB of a grid of $324 \times 324$

![Plot](image)

- Iteration number $\times 10^4$
- $2$-norm $10^0$ to $10^{-6}$

Onno Leon Meijers (TU Delft)
A plot of the solution of the 2D Helmholtz equation with Dirichlet boundary conditions.
Note: $V_{p1} < V_{p2}$

Determine depth to rock layer, $z_n$

Source (Plate)

Vertical Geophones

$V_{p2}$

Soil

$V_{p1}$

Rock

$1$ https://www.omicsonline.org/articles-images/Geology-Geophysics-Show-the-path-seismic-waves-refraction-from-source-5-259-g008.png
Conclusions

1. The PCIe communication speed between the CPU and the DFE is not high enough to only do 1 calculation on the DFE, like the matrix vector multiplication, and send the results back.

2. The BiCGSTAB method can be implemented on the DFE. This is done by splitting the algorithm in 3 parts.

3. An improvement can be seen over the GPU implementation of HP Knibbe. This DFE implementation is 2.4 times faster for the BiCGSTAB method without preconditioner.

4. Finally, an increase in communication speed between the large memory and the DFE will result in better performance of this BiCGSTAB implementation. When the communication speed is doubled the calculation time will be halved.

HP Knibbe, Reduction of computing time for seismic applications based on the Helmholtz equation by Graphics Processing Units, PhD thesis, TU Delft, Delft University of Technology, 2015
Conclusions

2. The BiCGSTAB method can be implemented on the DFE. This is done by splitting the algorithm in 3 parts.
Conclusions

An improvement can be seen over the GPU implementation of HP Knibbe\(^2\). This DFE implementation is 2.4 times faster for the BiCGSTAB method without preconditioner.

---

\(^2\)HP Knibbe, Reduction of computing time for seismic applications based on the Helmholtz equation by Graphics Processing Units, PhD thesis, TU Delft, Delft University of Technology, 2015
Conclusions

Finally, an increase in communication speed between the large memory and the DFE will result in better performance of this BiCGSTAB implementation. When the communication speed is doubled the calculation time will be halved.
Conclusions

1. The PCIe communication speed between the CPU and the DFE is not high enough to only do 1 calculation on the DFE, like the matrix vector multiplication, and send the results back.

2. The BiCGSTAB method can be implemented on the DFE. This is done by splitting the algorithm in 3 parts.

3. An improvement can be seen over the GPU implementation of HP Knibbe\(^2\). This DFE implementation is 2.4 times faster for the BiCGSTAB method without preconditioner.

4. Finally, an increase in communication speed between the large memory and the DFE will result in better performance of this BiCGSTAB implementation. When the communication speed is doubled the calculation time will be halved.

\(^2\)HP Knibbe, Reduction of computing time for seismic applications based on the Helmholtz equation by Graphics Processing Units, PhD thesis, TU Delft, Delft University of Technology, 2015
Future research

1. Sparse Matrix vector product on DFE.
2. Calculate partial sums on DFE.
4. Implement the multi grid preconditioner on the DFE.
5. Make $k$ a variable.
6. Implement the 3D problem.