Accelerating an Iterative Helmholtz Solver with FPGAs

Art Petrenko¹, Tristan van Leeuwen², Diego Oriato³, Simon Tilbury³ and Felix J. Herrmann¹
June 18, 2014

¹University of British Columbia (UBC)
²Centrum Wiskunde & Informatica, formerly with UBC
³Maxeler Technologies
Oh by the way: I have a stutter.
Seismic Wave Simulation
Full-waveform Inversion

Seismic Wavefield ($\mathbf{u}$)

Earth model ($\mathbf{m}$)
The Accelerators Have Arrived

# Top 10 of “Top 500” Supercomputers

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>System</th>
<th>Corea</th>
<th>Rmax (TFlop/s)</th>
<th>Rpeak (TFlop/s)</th>
<th>Power (KW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>National Super Computer Center in Guangzhou, China</td>
<td>Tianhe-2 (MilkyWay-2) - TH-YB-PEP Cluster, Intel Xeon E5-2661 23C 2.000GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT</td>
<td>3,120,000</td>
<td>33,662.2</td>
<td>54,902.4</td>
<td>17,808</td>
</tr>
<tr>
<td>2</td>
<td>DOE/SCOak Ridge National Laboratory, United States</td>
<td>Titan - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnected, NVIDIA K20x Cray Inc.</td>
<td>600,040</td>
<td>17,590.0</td>
<td>27,112.5</td>
<td>8,209</td>
</tr>
<tr>
<td>3</td>
<td>DOE/ANSALNL, United States</td>
<td>Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM</td>
<td>1,072,864</td>
<td>17,113.2</td>
<td>20,132.7</td>
<td>7,890</td>
</tr>
<tr>
<td>4</td>
<td>RIKEN Advanced Institute for Computational Science (AICS), Japan</td>
<td>K computer, SPARC64 VIII IF, Tofu interconnect Fujitsu</td>
<td>705,024</td>
<td>10,510.0</td>
<td>11,280.4</td>
<td>12,680</td>
</tr>
<tr>
<td>5</td>
<td>DOE/SC/Argonne National Laboratory, United States</td>
<td>Mira - BlueGene/Q, Power BQC 16C 1.65GHz, Custom IBM</td>
<td>786,432</td>
<td>8,596.6</td>
<td>10,068.3</td>
<td>3,945</td>
</tr>
<tr>
<td>6</td>
<td>Swiss National Supercomputing Centre (CSCS), Switzerland</td>
<td>Piz Daint - Cray XC30, Xeon E5-2670 8C 2.60GHz, Aries interconnected, NVIDIA K20x Cray Inc.</td>
<td>115,984</td>
<td>6,271.0</td>
<td>7,788.9</td>
<td>2,325</td>
</tr>
<tr>
<td>7</td>
<td>Texas Advanced Computing Center/Univ. of Texas, United States</td>
<td>Stampede - PowerEdge C8200, Xeon E5-2680 8C 2.70GHz, Infiniband FDR, Intel Xeon Phi 31S1P Dell</td>
<td>462,492</td>
<td>5,108.1</td>
<td>6,020.1</td>
<td>4,510</td>
</tr>
<tr>
<td>8</td>
<td>Forschungszentrum Juelich (FZJ), Germany</td>
<td>Juqueen - BlueGene/Q, Power BQC 16C 1.60GHz, Custom interconnected IBM</td>
<td>450,752</td>
<td>5,006.9</td>
<td>5,872.0</td>
<td>2,301</td>
</tr>
<tr>
<td>9</td>
<td>DOE/ANSALNL, United States</td>
<td>Vulcan - BlueGene/Q, Power BQC 16C 1.60GHz, Custom interconnected IBM</td>
<td>393,216</td>
<td>4,292.3</td>
<td>5,032.0</td>
<td>1,972</td>
</tr>
<tr>
<td>10</td>
<td>Leibniz Rechenzentrum, Germany</td>
<td>SuperMUC - iDataPlex DX3800M4, Xeon E5-2690 8C 2.70GHz, Infiniband FDR IBM</td>
<td>147,456</td>
<td>2,857.0</td>
<td>3,185.1</td>
<td>3,423</td>
</tr>
</tbody>
</table>
FPGAs: Reconfigurable Hardware Accelerators
The Punchline
Modelling Seismic Waves
Mathematical Formulation
When simulating waves in the frequency domain, the PDE that describes the motion of the wave through a heterogeneous medium can be written as:

$$\left(\omega^2 m + \Delta\right) u = q$$

and is known as the Helmholtz equation. As written above, the Helmholtz equation represents the special case of a constant density isotropic medium which only supports acoustic waves. Damping effects of viscosity are modelled heuristically by allowing $m$ to be complex-valued. I ignore the case of elastic and anisotropic media to keep the resulting implementation relatively simple. The symbol $\Delta$ represents the Laplacian operator. I take the subsurface earth model to be the slowness squared, $m = \frac{1}{v^2}$, where $v$ is the sound speed of the medium. $u$ is the (complex) Fourier transform of the pressure with respect to time, and $q$ is the amplitude of the source at angular frequency $\omega$. The Laplacian operator here also implements perfectly matched layer (PML) boundary conditions that eliminate reflection artefacts from the boundaries of the domain by setting a damping layer consisting of complex velocities. (See Equation (2) in the work by Operto et al. [42].) In the frequency domain, Equation 2.1 must be solved for each frequency $\omega$ that is to contribute to the final wavefield $u_s$ for a given source.
Modelling Seismic Waves: Discretization
[Operto, 2007]

Cartesian grid with $N$ grid-points, Equation 2.1 can be represented as a large system of linear equations and succinctly written in matrix notation,

$$A(m, \omega)u = q$$

where $A$ is the $N \times N$ Helmholtz matrix. The elements of $A$ are calculated by the method of finite differences, following the strategy of Operto et al. [42], which consists of two main parts.

First, the Laplacian operator is discretized using a $3 \times 3 \times 3$ cube stencil. This 27-point stencil is a weighted average of eight different second-order 7-point star stencils, each on their own coordinate grid. The coordinate grids are rotated and scaled versions of the original Cartesian grid. The grids are designed in such a way that even though their axes are not parallel to each other, the grid-point locations of all eight grids coincide. This allows the cube stencil to use all 27 points in the three-dimensional neighbourhood of a given central point. The weighting coefficients are tuned to minimize numerical anisotropy, as described by Operto et al. [42].

Second, the value of the earth model $m$ at each grid-point is re-distributed to the 27 neighbouring points that make up the cube stencil for that point, a process known as mass-averaging. Mass-averaging is done using a second set of weighting coefficients, and results in a matrix with the same pattern of non-zero entries as the discretized Laplacian. By choosing optimal values for the mass-averaging coefficients, Operto et al. [42] showed that numerical dispersion of the stencil is minimized, which enhances (by a constant factor) the stencil's accuracy. This allows to use as little as 4 grid-points per wavelength [42], although 6 grid points per wavelength are used in this work. The need for accuracy by using finer grid spacings (more grid-points per wavelength) must be balanced against the computational expense of simulating on a larger grid. As noted by Operto et al. [42], 4 grid-points per wavelength is the limit at which the modelling step is accurate, without modelling wavefield features that are in any case too small to be of use to full-waveform inversion in resolving the earth model. A further advantage of the stencil introduced by Operto et al. [42] is that it is compact. A stencil with a large extent in the last dimension (for example as in 5-point star stencil) implies that more intervening grid-points need to be buffered in short-term memory (see Section 3.3 for details).

The mass matrix and the discretized Laplacian are added together to make the Helmholtz matrix $A$, which is very sparse: while $N$ is at least $10^7$ for a realistic model, the number of non-zeros per row, determined by the finite difference stencil, is at most only 27. $A$ is also very structured: its non-zeros are arranged in 27 diagonals. This means that the locations of the non-zero elements, taken together, of each matrix row, do not repeat. In other words, the support of each matrix row is unique (although the support of rows that correspond to grid points adjacent to the edge of the grid is a subset of the support of the rows that correspond to adjacent points). These special properties mean that the rows of the Helmholtz matrix are linearly independent and hence theoretically (disregarding round-off errors) $A$ is invertible.
Solving the Helmholtz System
Adapted from conjugate gradients: The CGMN algorithm

Despite the advantages mentioned in the last section, the Kaczmarz algorithm (SSOR-NE) converges slowly, thus it is not suitable for direct application to the Helmholtz system (Equation 2.2). Instead, Björck and Elfving [4] showed that it can be used to accelerate the method of conjugate gradients, calling the resulting algorithm CGMN. Recent studies of how CGMN fares in solving the Helmholtz equation include work by van Leeuwen [57] and Gordon and Gordon [18]. In the latter case, CGMN is equivalent to the sequential (non-parallel, running on only one processor core) version of the algorithm CARP-CG, introduced by Gordon and Gordon [17].

I now describe the CGMN algorithm.

First, it is useful to represent the Kaczmarz sweeps in matrix notation. Following Tanabe [55], let $Q_i$ be the projection matrix onto the hyperplane defined by $a_i.x = 0$:

$$Q_i = I - \frac{1}{a_i a_i^T} a_i a_i^T.$$

The double sweep can then be written as

$$DKSWP(A, u, q, \lambda) = Q_1 \cdots Q_N Q_N \cdots Q_1 u + R_q. \quad (2.5)$$

Since $A$ is invertible, SSOR-NE will converge to the solution of Equation 2.2, as mentioned by Björck and Elfving [4]. At that point, the iterate $u$ will be a fixed point of the relation 2.4, which means that Equation 2.5 can be re-written as a linear system:

$$(I - Q)u = R_q, \quad (2.6)$$

where $I$ is the identity matrix. As mentioned by Björck and Elfving [4] and proved by, for example, Gordon and Gordon [17], the system in Equation 2.6 is consistent, symmetric and positive semi-definite. Björck and Elfving [4] show in their Lemma 5.1 that this is sufficient for CG to converge to the pseudoinverse (minimum $\ell_2$-norm) solution of Equation 2.6, which is the same as the solution of the original system (Equation 2.2). Note that the matrices $Q$ and $R$ do not have to be formed explicitly, as their action on a vector is calculated using a double Kaczmarz sweep, as in Equation 2.5.

Thus, CGMN is the use of the method of conjugate gradients to solve the SSOR-NE iteration system (Equation 2.6) for the fixed point of that iteration. SSOR-NE is implemented efficiently using Kaczmarz row projections. Björck and Elfving [4] also note that it is possible to view CGMN as solving a variant of the Helmholtz system (Equation 2.2), preconditioned from the left by a matrix derived from a decomposition of $A A^T$.

Pseudo-code for the CGMN algorithm is given below. Note that the double Kaczmarz sweep on line 1 of the algorithm is performed with an initial guess of zero because only the action

\[10\]
The Kaczmarz Algorithm: Equivalent to SSOR-NE

[Björck and Elfving, 1979]

Double Kaczmarz sweep on the original system:

\[ Au = q \]

One iteration of SSOR on the normal equations:

\[ AA^*y = q \]
\[ A^*y = u \]

Both are computed as:

\[ u_{k+1} = u_k + \lambda (b_i - \langle a_i, u_k \rangle) \frac{a_i^*}{\|a_i\|^2} \]

\[ k : 1 \rightarrow 2N \]
\[ i : 1 \rightarrow N, N \rightarrow 1 \]
Kaczmarz + CG = CGMN
[Björck & Elfving 1979]
CGMN: Solves for Fixed Point of Kaczmarz Row Projections

\[
\text{DKSWP}(A, u, q, \lambda) = Q_1 \cdots Q_N Q_N \cdots Q_1 u + Rq \\
= Qu + Rq.
\]

Assume \( u \) is a solution and re-arrange:

\[
(I - Q)u = Rq
\]
Block parallelization of Kaczmarz + CG = CARP-CG
[Gordon & Gordon, 2010]
Contribution of This Work
Compute Node Overview
[Maxeler Technologies, 2011]

Algorithm 1 CGMN (Björck and Elfving [4])

Input: $A$, $u$, $q$, $\lambda$

1: $Rq \leftarrow$ DKSWP($A$, $0$, $q$, $\lambda$)
2: $r \leftarrow Rq - u +$ DKSWP($A$, $u$, $0$, $\lambda$)
3: $p \leftarrow r$
4: while $\|r\|^2 > tol$ do
5: $s \leftarrow (I - Q)p = p -$ DKSWP($A$, $p$, $0$, $\lambda$)
6: $\alpha \leftarrow \|r\|^2 / \langle p, s \rangle$
7: $u \leftarrow u + \alpha p$
8: $r \leftarrow r - \alpha s$
9: $\beta \leftarrow \|r\|_\text{curr}^2 / \|r\|_\text{prev}^2$
10: $\|r\|_\text{prev} \leftarrow \|r\|_\text{curr}^2$
11: $p \leftarrow r + \beta p$
12: end while

Output: $u$
Implementation Details
Layout of 3D Wavefields in 1D Memory

3D layout

fast  medium  slow

-31  -26  -21  -6  -1  4  19
-30  -25  -20  0  5  25  30
-29  -24  -19  1  6  26  31

Linear layout (for 5 x 5 x 5 system)

-30  -25  -20  -5  0  5  20  25  30

increasing memory addresses
Buffering: Overcoming Latency of Memory Access

Memory (24 GB)

On-chip Buffers (BRAM)
4.6 MB maximum
Pipelining: Overcoming Latency of Computation
Pipelining: Overcoming Latency of Computation
Pipelining: Overcoming Latency of Computation

The diagram illustrates the concept of pipelining, which helps in overcoming the latency of computation. It shows different stages of processing, marked as 'slow', 'medium', and 'fast'. The diagram includes numbers 9, 10, 11, and 12, indicating the progression through these stages.
Granular Memory Access: 384 bytes / burst

<table>
<thead>
<tr>
<th>Number of bits in a real number</th>
<th>Number of bits in a complex number</th>
<th>Complex numbers per burst</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>48</td>
<td>64</td>
</tr>
<tr>
<td>32 (single precision)</td>
<td>64</td>
<td>48</td>
</tr>
<tr>
<td>48</td>
<td>96</td>
<td>32</td>
</tr>
<tr>
<td>64 (double precision)</td>
<td>128</td>
<td>24</td>
</tr>
</tbody>
</table>
Matrix Coefficient Representation

Number of bits in a real number

Matrix row storage efficiency

32 bits used in this work
Results & Discussion
What is being compared?

**Reference implementation:**
- Solution Algorithm
  - CARP-CG
  - written in MATLAB
  - single-threaded
  - double precision
  - running on Intel Xeon E5-2670
- Computational Kernel
  - CARP sweeps
  - written in C
  - 32 threads
  - double precision
  - running on Intel Xeon E5-2670

**Accelerator implementation:**
- Solution Algorithm
  - CGMN
  - written in MATLAB
  - single-threaded
  - single precision
  - running on Intel Xeon E5-2670
- Computational Kernel
  - Kaczmarz sweeps
  - single precision
  - running on Maxeler Vectis accelerator at 100 MHz
  - Memory clock at 303 MHz
Experimental Set-up

- Solve part of the SEG/EAGE Overthrust System: 432 x 500 x Z.
- Point source.
- Zero initial guess.
- Run 100 CARP-CG/CGMN iterations.
Throughput of Kaczmarz/CARP sweeps

Double Kaczmarz/CARP sweeps: Throughput

Throughput (millions of grid points/s)

Number of grid points (millions)

- Intel Xeon E5-2670 (16 cores)
- Maxeler Vectis accelerator
CARP-CG: End-to-end execution time

Lots of overhead!

Kaczmarz sweeps are 39% of run time.

Future Solution: Port all of CGMN to accelerator.
Avoiding Future Communication Bottlenecks

- 38.4 GB/s: memory bandwidth limit
- 29 GB/s: bandwidth limit in current work
- 2 GB/s: PCIe bandwidth limit

Diagram showing the relationship between FPGA frequency (MHz) and LMem and PCIe bandwidth (GB/s). The diagram includes lines for current accelerated, CGMN on FPGA efficient matrix storage, CGMN on the FPGA, matrix on the fly, and PCIe requirement for current accelerated version.
FPGA Resource Usage: Room for parallelism?

inner product: $\langle a_i, x_k \rangle$

row projection component: $\times a_i$

updating the iterate: $x_k + \ldots$

kernel (other)

LUTs

FFs

BRAMs

DSPs

$432 \times 500 \times z$

buffers

memory controller

Fraction of available resources
Effect of matrix row ordering on CGMN convergence

![Graph showing the effect of matrix row ordering on CGMN convergence. The graph plots the relative residual norm against iteration for different matrix row ordering schemes: 1 to N (sequential), Accelerator ordering, wavefield source, and point source.]
Conclusion

Have implemented frequency-domain wave simulation using reconfigurable hardware. More work needed to realize full potential of accelerator system.
Acknowledgements

Thank you to:

- Felix Herrmann, Henryk Modzelewski, Diego Oriato, Simon Tilbury, Tristan van Leeuwen, Eddie Hung, Rafael Lago, Maxeler Technologies, and everyone in the SLIM group!

This work was financially supported in part by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (RGPIN 261641-06) and the Collaborative Research and Development Grant DNOISE II (CDRP J 375142-08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BGP, BP, Chevron, ConocoPhillips, CGG, ION GXT, Petrobras, PGS, Statoil, Total SA, WesternGeco, Woodside.
References


F. Grüll, M. Kunz, M. Hausmann, and U. Kebschull. An implementation of 3D electron tomography on FPGAs. In Reconfigurable Computing and FPGAs (ReConFig), 2012 International Conference on, pages 1–5, 2012. doi: 10.1109/ReConFig.2012.6416732.


www.slim.eos.ubc.ca