m2cpp – A MATLAB to C++ Code Translator with Traceability

Geir Y. Paulsen*, Bjørn Nordmoen, Xing Cai, Stuart Clark, Jonathan Feinberg and Hans Petter Dahle
*Corresponding author, email: geirpa@simula.no

Seismic Surveying: The Need for Optimised Code and Agile Development

Optimisation
• Huge Data Rates (> 1 Gb/s)
• Computing Power (> 2 Tflop)
• Network of Computers (>2000 cores)
• But it is difficult to write optimised code quickly for new algorithms

Agility
• MATLAB is familiar to seismologists
• It is easy to write new algorithms
• However, runtime is too slow for industrial execution!

The Idea: Legible Code Translation

Seismic Towing Configuration

Workflow

MATLAB Source
• Parse the code and create an Abstract Syntax Tree

C++ with Armadillo
• Armadillo allows C++ Syntax to be similar to MATLAB

Parallelisation and Vectorisation
• Currently supports parallelising loops using OpenMP* or TBB

[Optional] Hand-coded optimisation
• Since the code is readable, manual optimisation is possible

MATLAB to C++ Speed up (Single Core)

We tested the execution times and speed-up factor of the generated C++ from m2cpp vs the MATLAB on 5 examples (single core).

<table>
<thead>
<tr>
<th>Code Example</th>
<th>Problem Size</th>
<th>MATLAB Time (s)</th>
<th>C++ Time (s)</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>fx decon</td>
<td>3.2 \times 10^6</td>
<td>2.7</td>
<td>1.9</td>
<td>1.4</td>
</tr>
<tr>
<td>moveout</td>
<td>1.0 \times 10^7</td>
<td>3.0</td>
<td>0.9</td>
<td>3.3</td>
</tr>
<tr>
<td>parabolic</td>
<td>6.3 \times 10^6</td>
<td>268</td>
<td>141</td>
<td>1.9</td>
</tr>
<tr>
<td>radon decon</td>
<td>6.3 \times 10^6</td>
<td>34</td>
<td>27</td>
<td>1.3</td>
</tr>
<tr>
<td>va</td>
<td>2.5 \times 10^6</td>
<td>172</td>
<td>70</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Conclusion: m2cpp is a traceable translator that creates faster code

• Code in C++ appears similar to MATLAB code
• We obtain speed-up factors from 1.4 to over 3 on a single core
• Hand-coded optimisations are easier to implement because generated C++ code is readable
• Multi-core parallelisation available via OpenMP or TBB

Seismic Towing Configuration

Typical Configuration
Outer separation: 1.4 km
Streamer length: 10 km

Try it! m2cpp is on github:
https://github.com/emc2norway/m2cpp

MATLAB
B = M'*M;
beta = B(1,1)*mu/100;
ab = B + beta*eye(lf);
temp = [temp;zeros(lf,1)];

C++/Armadillo
B = arma::trans(M)*M;
beta = B(0,0)*mu/100.0;
ab = arma::solve((B+beta*arma::eye<cx_mat>(lf,lf)), arma::trans(M), solve_opts::fast)*y;
temp = arma::join_cols(temp, arma::zeros<cx_mat>(lf, 1));

%#TBB tells m2cpp to parallelise the for loop with TBB

Try it! m2cpp is on github:
https://github.com/emc2norway/m2cpp

Seismic Surveying: The Need for Optimised Code and Agile Development

Optimisation
• Huge Data Rates (> 1 Gb/s)
• Computing Power (> 2 Tflop)
• Network of Computers (>2000 cores)
• But it is difficult to write optimised code quickly for new algorithms

Agility
• MATLAB is familiar to seismologists
• It is easy to write new algorithms
• However, runtime is too slow for industrial execution!

The Idea: Legible Code Translation

Seismic Towing Configuration

Workflow

MATLAB Source
• Parse the code and create an Abstract Syntax Tree

C++ with Armadillo
• Armadillo allows C++ Syntax to be similar to MATLAB

Parallelisation and Vectorisation
• Currently supports parallelising loops using OpenMP* or TBB

[Optional] Hand-coded optimisation
• Since the code is readable, manual optimisation is possible

MATLAB to C++ Speed up (Single Core)

We tested the execution times and speed-up factor of the generated C++ from m2cpp vs the MATLAB on 5 examples (single core).

<table>
<thead>
<tr>
<th>Code Example</th>
<th>Problem Size</th>
<th>MATLAB Time (s)</th>
<th>C++ Time (s)</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>fx decon</td>
<td>3.2 \times 10^6</td>
<td>2.7</td>
<td>1.9</td>
<td>1.4</td>
</tr>
<tr>
<td>moveout</td>
<td>1.0 \times 10^7</td>
<td>3.0</td>
<td>0.9</td>
<td>3.3</td>
</tr>
<tr>
<td>parabolic</td>
<td>6.3 \times 10^6</td>
<td>268</td>
<td>141</td>
<td>1.9</td>
</tr>
<tr>
<td>radon decon</td>
<td>6.3 \times 10^6</td>
<td>34</td>
<td>27</td>
<td>1.3</td>
</tr>
<tr>
<td>va</td>
<td>2.5 \times 10^6</td>
<td>172</td>
<td>70</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Conclusion: m2cpp is a traceable translator that creates faster code

• Code in C++ appears similar to MATLAB code
• We obtain speed-up factors from 1.4 to over 3 on a single core
• Hand-coded optimisations are easier to implement because generated C++ code is readable
• Multi-core parallelisation available via OpenMP or TBB

Contributing Partners:

WesternGeco
ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

Geir Y. Paulsen*, Bjørn Nordmoen, Xing Cai, Stuart Clark, Jonathan Feinberg and Hans Petter Dahle
*Corresponding author, email: geirpa@simula.no

Seismic Surveying: The Need for Optimised Code and Agile Development

Optimisation
• Huge Data Rates (> 1 Gb/s)
• Computing Power (> 2 Tflop)
• Network of Computers (>2000 cores)
• But it is difficult to write optimised code quickly for new algorithms

Agility
• MATLAB is familiar to seismologists
• It is easy to write new algorithms
• However, runtime is too slow for industrial execution!

The Idea: Legible Code Translation

Seismic Towing Configuration

Workflow

MATLAB Source
• Parse the code and create an Abstract Syntax Tree

C++ with Armadillo
• Armadillo allows C++ Syntax to be similar to MATLAB

Parallelisation and Vectorisation
• Currently supports parallelising loops using OpenMP* or TBB

[Optional] Hand-coded optimisation
• Since the code is readable, manual optimisation is possible

MATLAB to C++ Speed up (Single Core)

We tested the execution times and speed-up factor of the generated C++ from m2cpp vs the MATLAB on 5 examples (single core).

<table>
<thead>
<tr>
<th>Code Example</th>
<th>Problem Size</th>
<th>MATLAB Time (s)</th>
<th>C++ Time (s)</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>fx decon</td>
<td>3.2 \times 10^6</td>
<td>2.7</td>
<td>1.9</td>
<td>1.4</td>
</tr>
<tr>
<td>moveout</td>
<td>1.0 \times 10^7</td>
<td>3.0</td>
<td>0.9</td>
<td>3.3</td>
</tr>
<tr>
<td>parabolic</td>
<td>6.3 \times 10^6</td>
<td>268</td>
<td>141</td>
<td>1.9</td>
</tr>
<tr>
<td>radon decon</td>
<td>6.3 \times 10^6</td>
<td>34</td>
<td>27</td>
<td>1.3</td>
</tr>
<tr>
<td>va</td>
<td>2.5 \times 10^6</td>
<td>172</td>
<td>70</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Conclusion: m2cpp is a traceable translator that creates faster code

• Code in C++ appears similar to MATLAB code
• We obtain speed-up factors from 1.4 to over 3 on a single core
• Hand-coded optimisations are easier to implement because generated C++ code is readable
• Multi-core parallelisation available via OpenMP or TBB

Contributing Partners:

WesternGeco
ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership