Making Numerics Better and Worse

Walter Landry
wlandry@caltech.edu

https://groups.google.com/forum/#!forum/bootstrap-collaboration-software

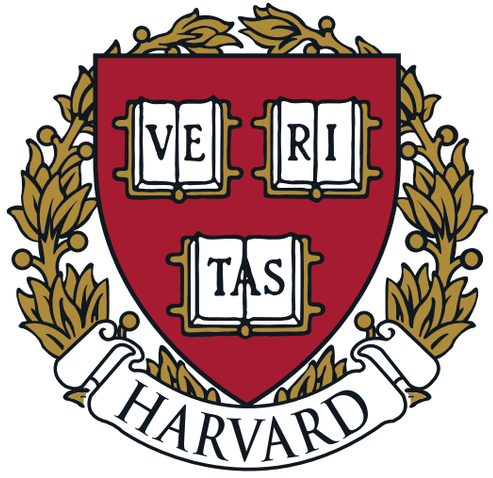
Caltech

Speeding up SDPB

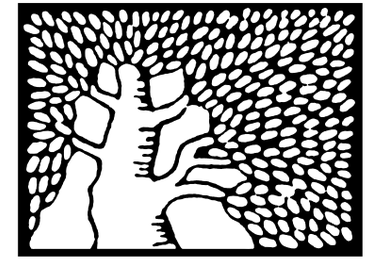
- Large parts of SDPB involves operations on independent blocks of varying size and structure.
- There are also some collective operations on a matrix Q.
- I ended up using MPI with the Elemental library.

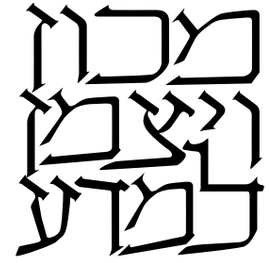
Fully Operational

- Produces the same results for a full run of the 3D Ising model up to the duality gap $\left(10^{-80}\right)$
- Installed at some sites



Caltech

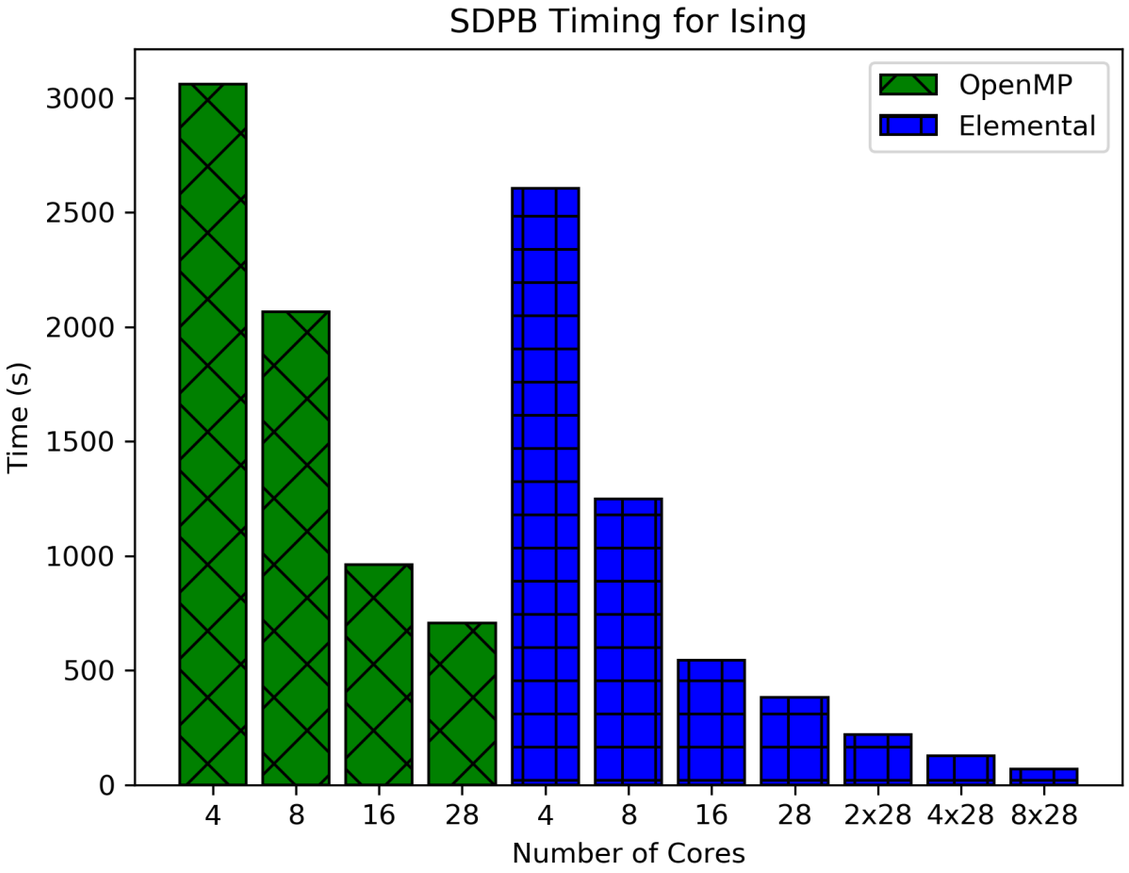


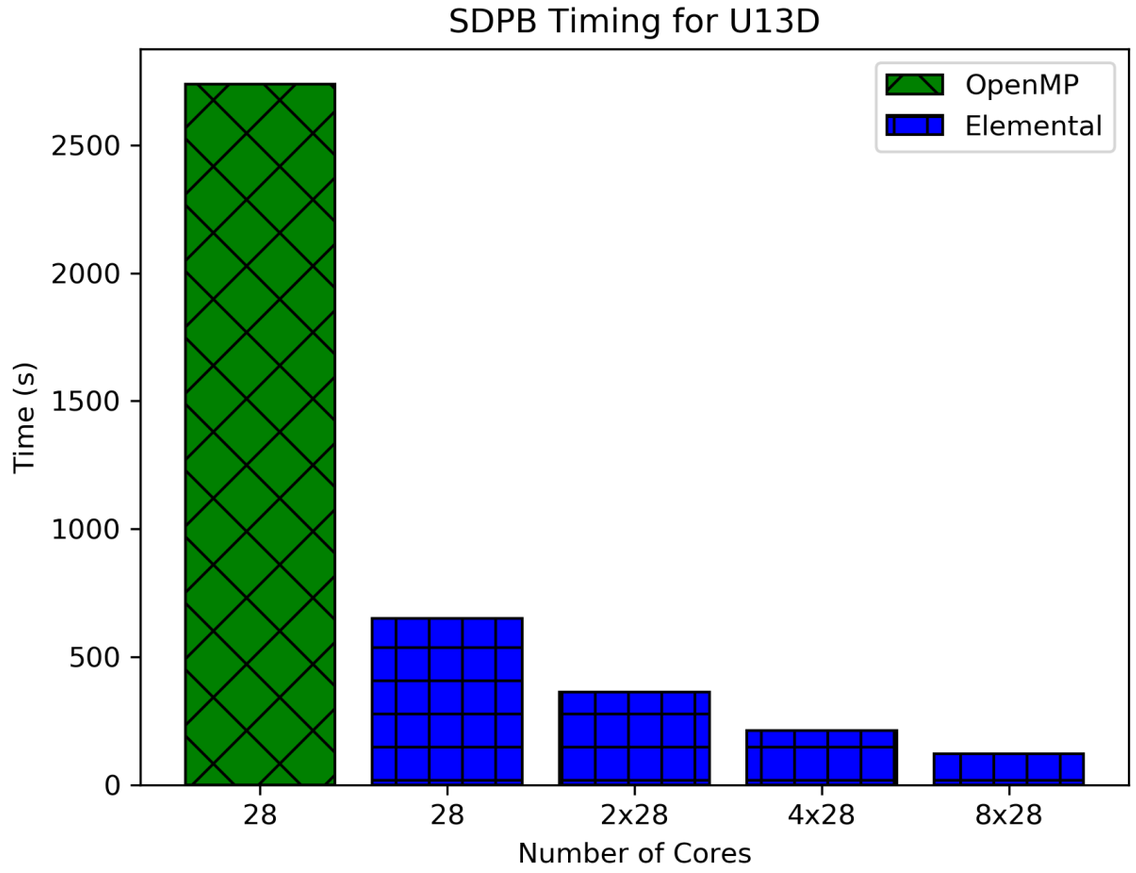


- Distributed through github

`https://github.com/davidsd/sdpb/tree/elemental`

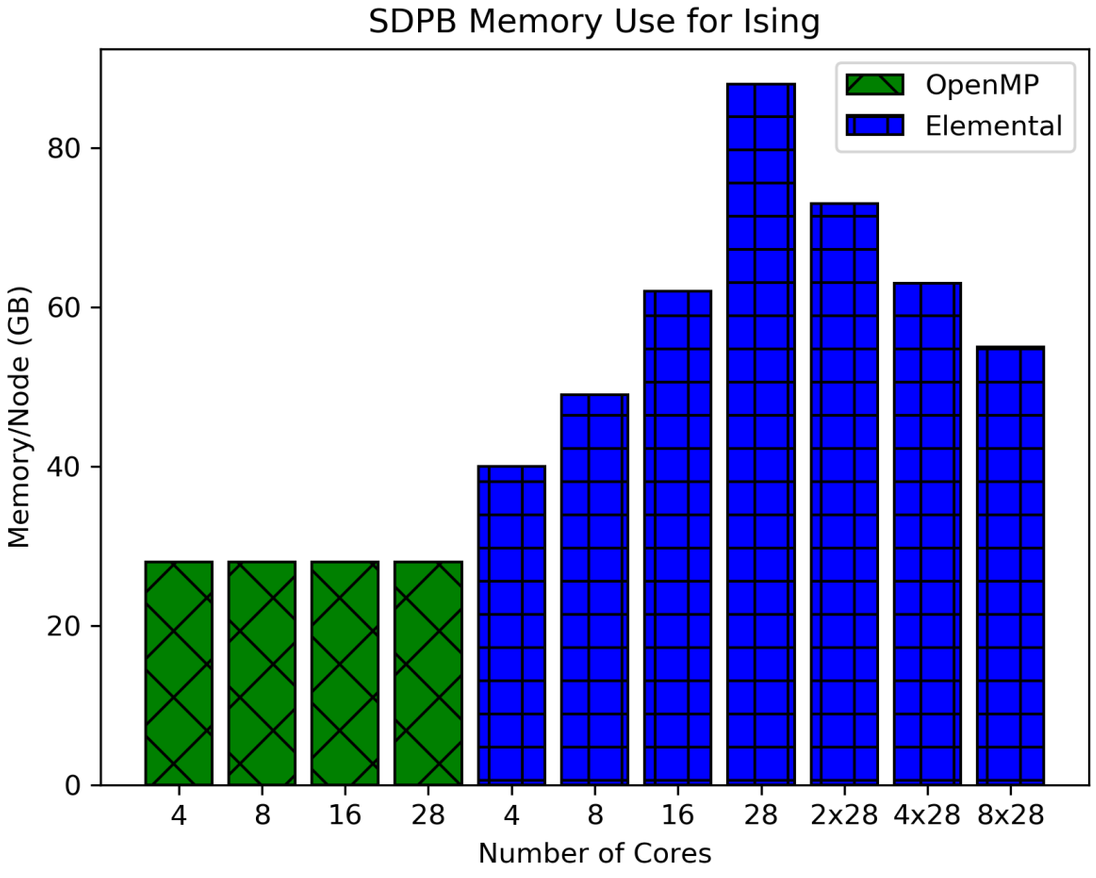
Faster

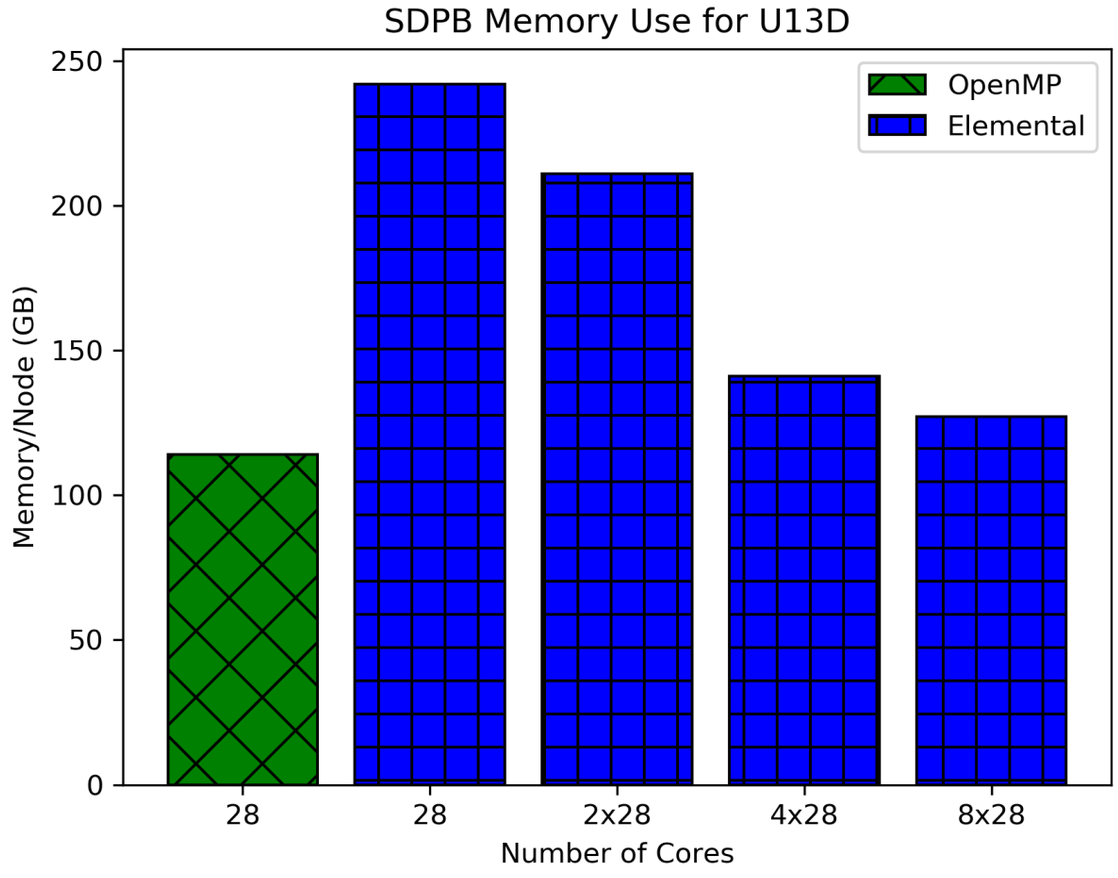




- 2-3 times faster on a single node
- 10-25 faster with more nodes

Uses Much More Memory





- You may need to run jobs on more nodes.

Convert XML Input

- First you have to convert XML input files into a format that is faster and uses less memory to load.
- You have to specify a precision during conversion. The actual calculation can run at that precision or less.

Block Timing

- For large runs, you should (but do not have to) perform a timing run to measure the cost to compute each block.
- This is necessary for accurately balancing the load between cores.
- It should be exactly two steps
  - The first step often has lots of zeroes, which are faster to add and multiply.

Block Timing

- It does not have to be exactly the same number of cores as the real runs.
  - It will use round-robin to allocate blocks.
  - If you have more cores than blocks, then some blocks will be mostly idle.
- If you are running the same model with different numbers, you should be able to reuse the timings file.

Harder to Install

- If everything works perfectly on the cluster, I can install it in a day.

Harder to Install

- If everything works perfectly on the cluster, I can install it in a day.
- I have yet to experience a cluster that works perfectly. I have run into problems with:
  - Compilers
  - MPI
  - Support libraries (GMP, Boost)
- Debugging these issues can take weeks.

Harder to Install

- If everything works perfectly on the cluster, I can install it in a day.
- I have yet to experience a cluster that works perfectly. I have run into problems with:
  - Compilers
  - MPI
  - Support libraries (GMP, Boost)
- Debugging these issues can take weeks.
- It might take longer if you do it yourself.

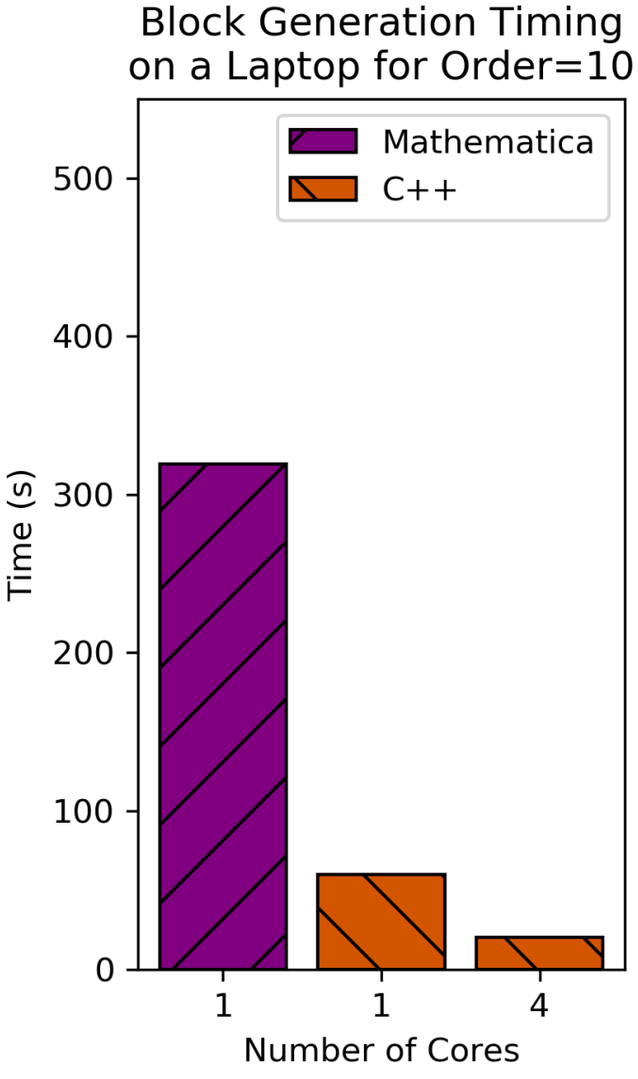
Block Generation

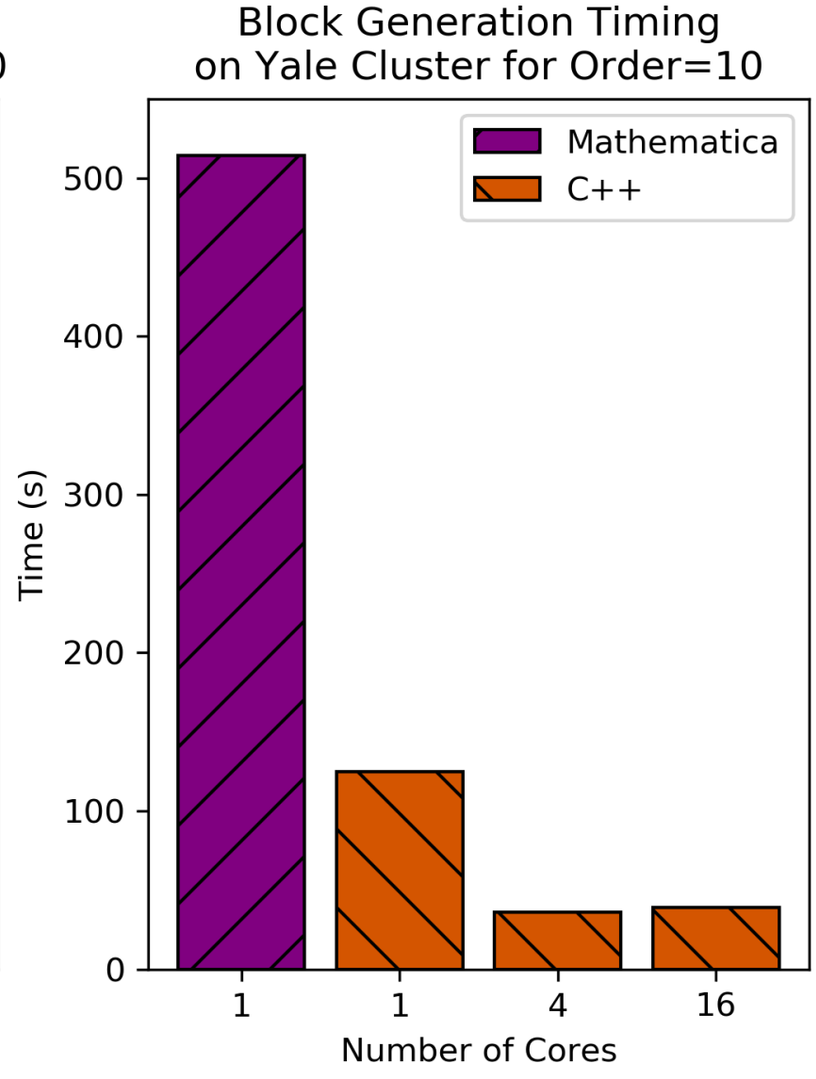
- Existing Mathematica script to generate conformal blocks.
- It can take longer than you might like.
  - Interpreted
  - Single threaded

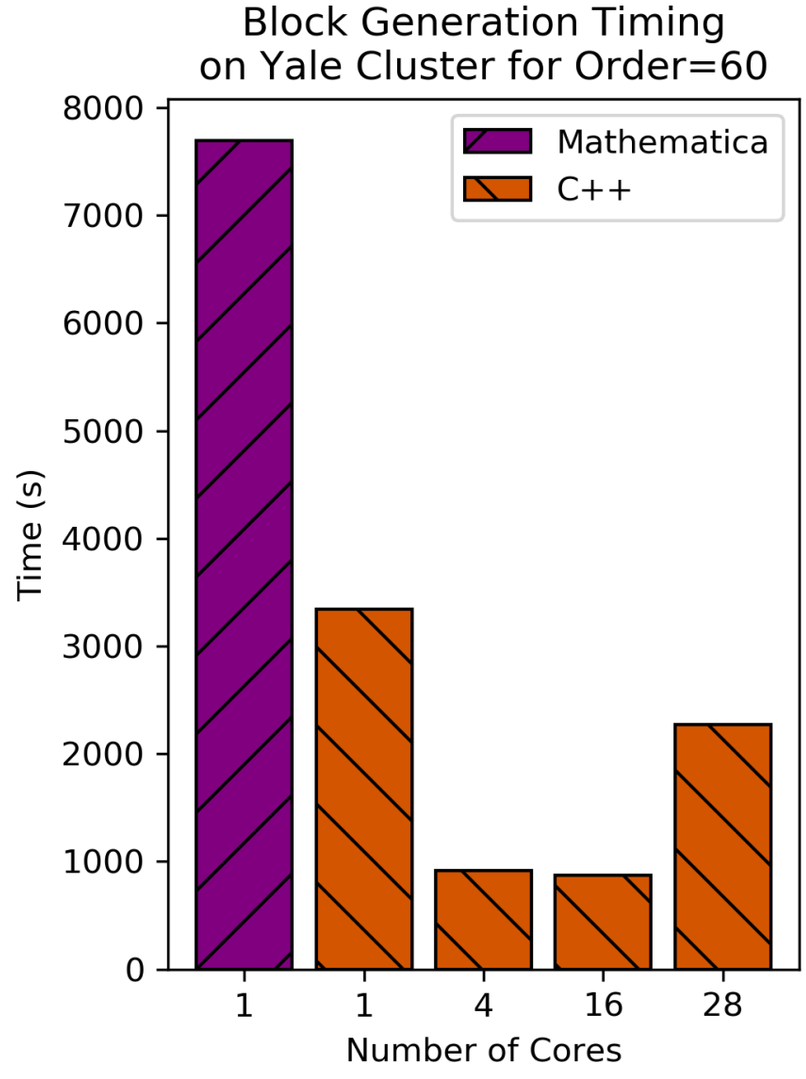
C++ Reimplementation

- Identical results up to 180+ digits
- Still has many rough edges.
- `https://gitlab.com/bootstrapcollaboration/scalar_blocks`

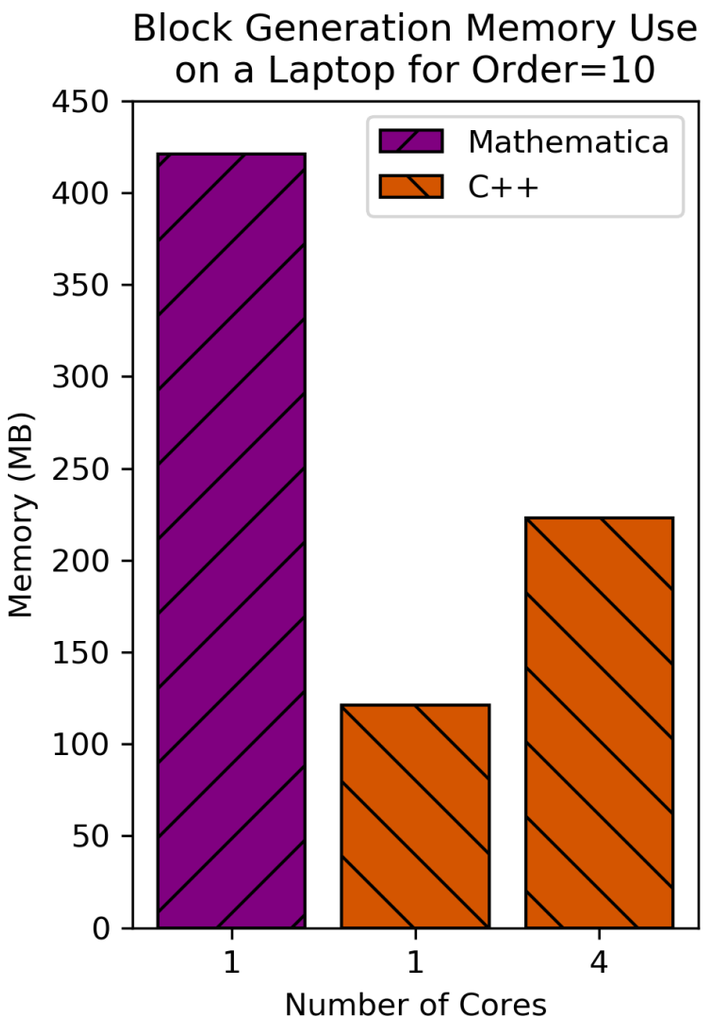
Faster

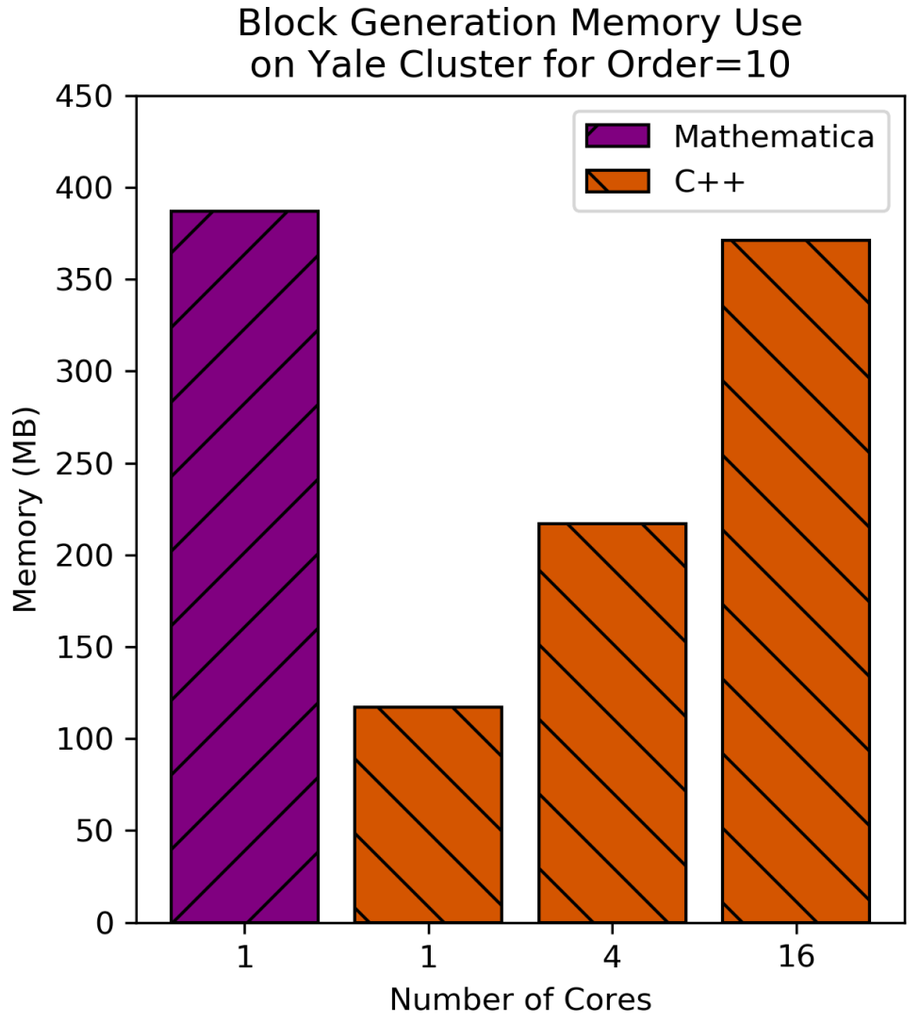

Block Generation Timing
on a Laptop for Order=10
Mathematica
C++
Time (s)
0
100
200
300
400
500
1
1
4
Number of Cores

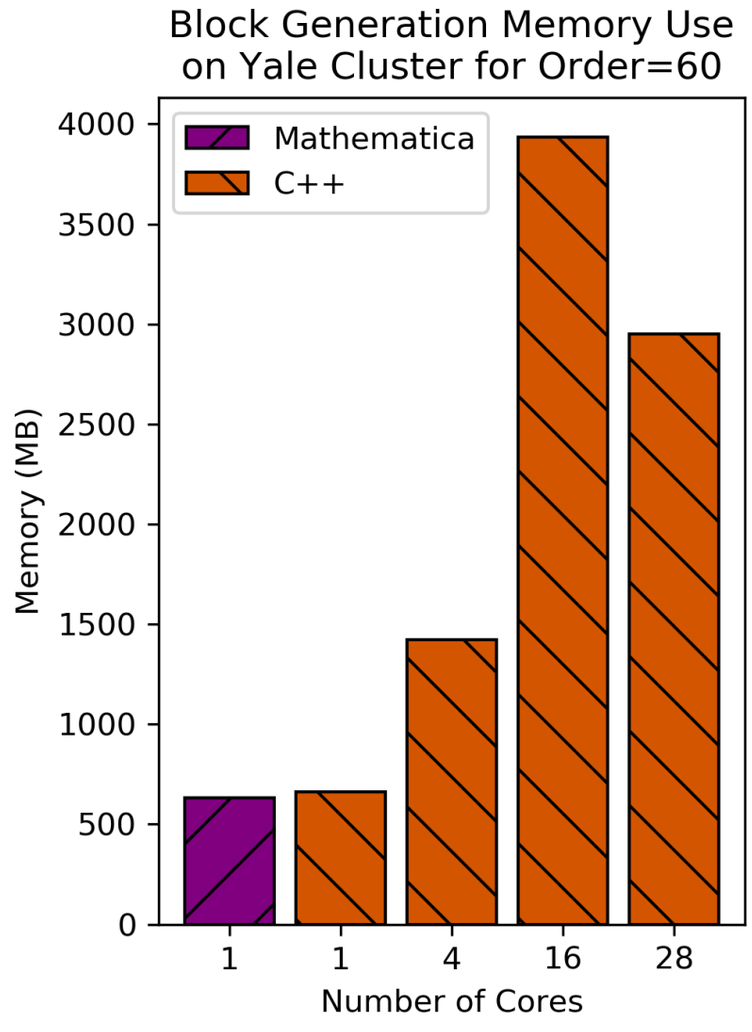

Block Generation Timing
on Yale Cluster for Order=10
Mathematica
C++
Time (s)
0
100
200
300
400
500
1
1
4
16
Number of Cores


Block Generation Timing
on Yale Cluster for Order=60
Mathematica
C++
Time (s)
0
1000
2000
3000
4000
5000
6000
7000
8000
1
1
4
16
28
Number of Cores

Less Memory?


Block Generation Memory Use
on a Laptop for Order=10
Mathematica
C++
Memory (MB)
0
50
100
150
200
250
300
350
400
450
1
1
4
Number of Cores


Block Generation Memory Use
on Yale Cluster for Order=10
Mathematica
C++
Memory (MB)
0
50
100
150
200
250
300
350
400
450
1
1
4
16
Number of Cores


Block Generation Memory Use
on Yale Cluster for Order=60
Mathematica
C++
Memory (MB)
0
500
1000
1500
2000
2500
3000
3500
4000
1
1
4
16
28
Number of Cores

Does not Require Mathematica

Does not Require Mathematica

Requires a C++ compiler

Not nearly as difficult to install as SDPB-Elemental

TODO

- Install SDPB-Elemental in more places
- Docker images for laptops and desktops?
- Investigate how precision is used in SDPB.
- Block generation polish and documentation
- Block generation optimization
  - More algorithms? (e.g. Casimir Equation)
  - More profiling and better scaling