Linear Solvers

For jcer matrices

Goal: Solve Ax=b as quickly as possible

Input 4 sets of matrices (called seq), with 3 A matrices each and the corresponding b vectors

Libraries used
AMGCL: https://github.com/ddemidov/amgcl MIT license
Eigen(iterative_solver_branch):
https://gitlab.com/jenswehner/eigen/-/tree/linear_solvers

Click to add text

Benchmark Code: https://github.com/NLESC-JCER/sparse_solver

Machines used: CPU benchmark Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 4 hardware threads
GCC 7.5 -O3 -DNDEBUG -march=native -mtune=native

GPU benchmark CUDA Version: 10.2 GeForce GTX 980 Intel(R) Xeon(R) CPU E5-1630 v4 @ 3.70GHz

Setup

Tolerance for solution is 1e-14

Initial guess was the zero vector. All libraries allow the specification of an initial guess, which typically lowers the iterations count by 10% for a tolerance of 1e-14

All solvers converged for all problems within the limit of 1000 iterations. Each seq was profiled independently. Timings were averaged over the 3 matrices in set.

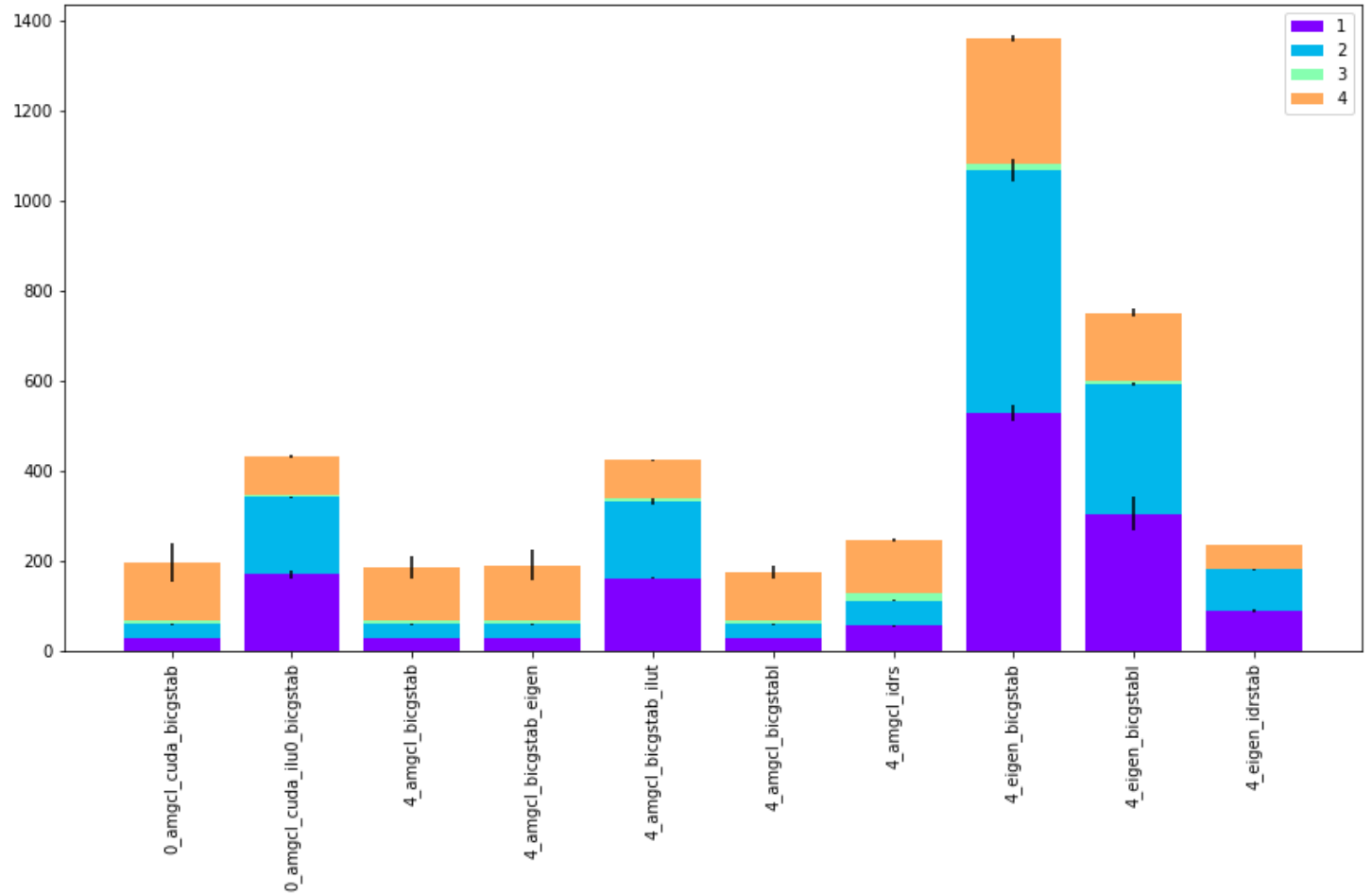
Solvers used: Eigen: BICGSTAB, BICGSTABL, IDRSTAB with diagonal preconditioner
(the Eigen::IncompleteLUT took longer to setup than the whole benchmark ran)

AMGCL: BICGSTAB, BICGSTABL, IDRS with AMG preconditioner
and with ilu0/ilut preconditioner

AMGCL can use different backends for the mat-vec product:
The CUDA, Eigen and the internal backend where used.

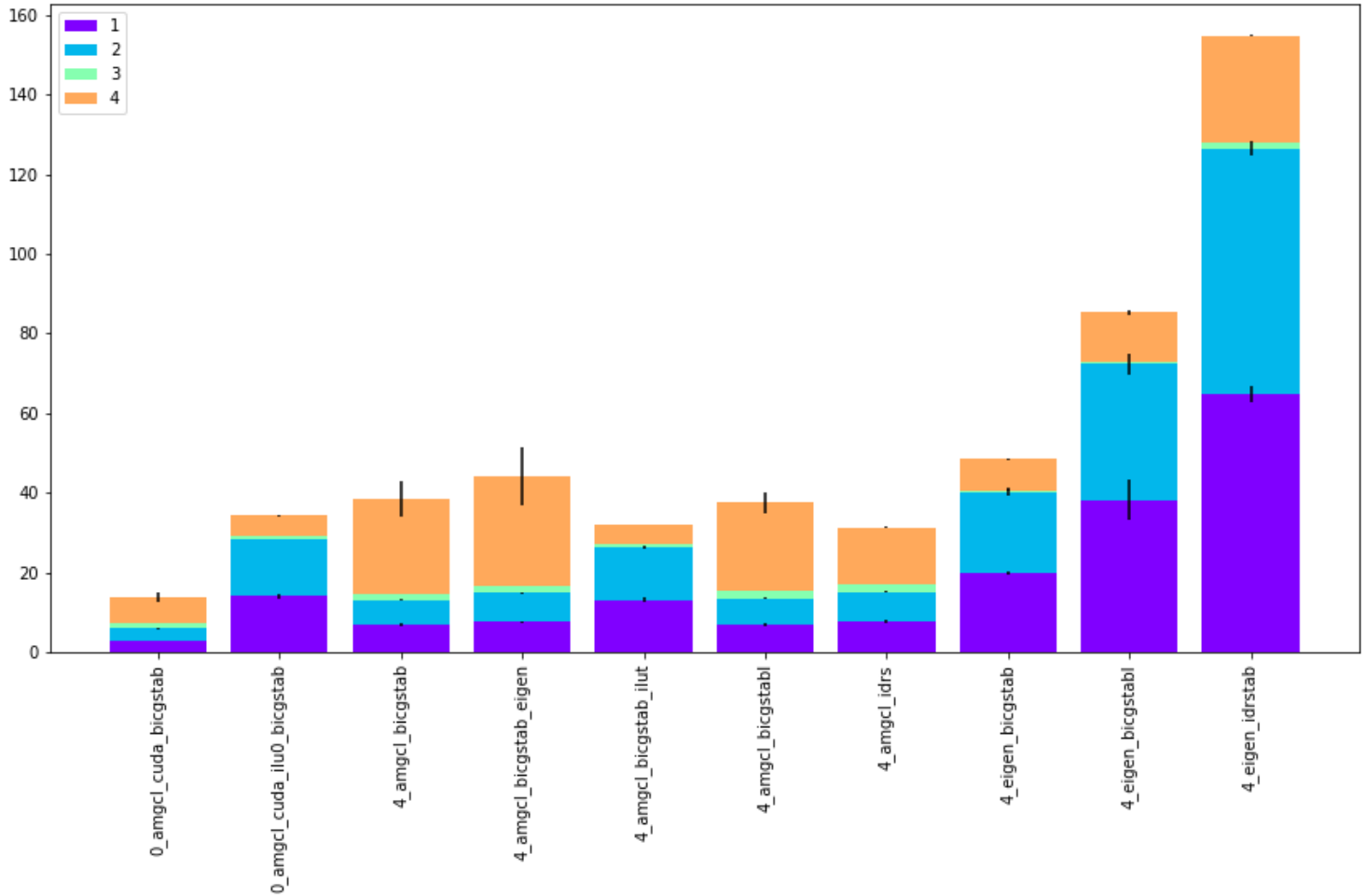
CPU benchmarks, where run with 1,2,4 threads. CUDA benchmarks are given with 0 threads

Iteration count



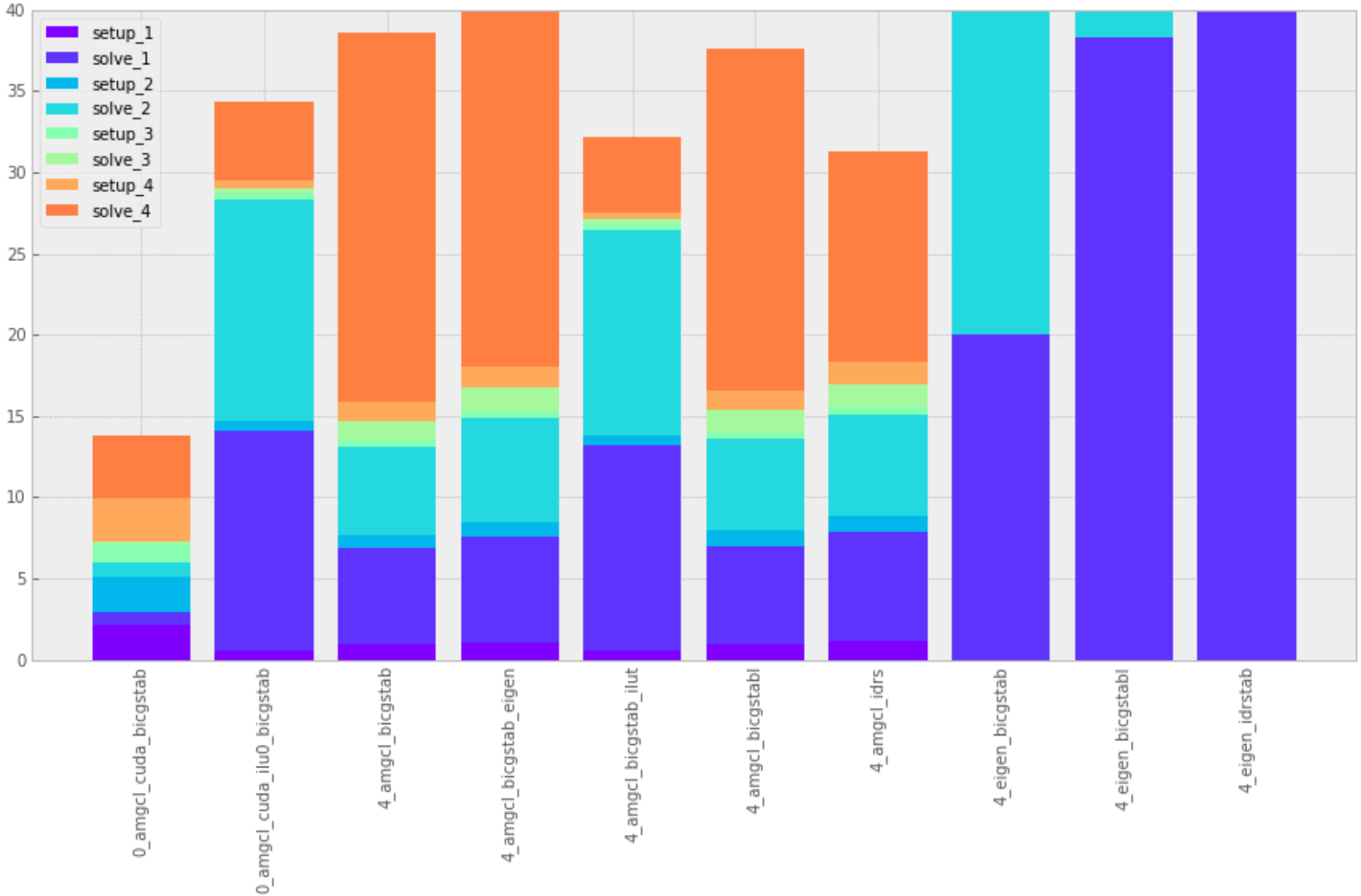
AMG preconditioner needs a lot less iterations then other preconditioners for seq1 and seq2.
Backend does not change iteration count, which is good.
For seq3 and seq4 ilu0/ilut preconditioner seems better.

Timings



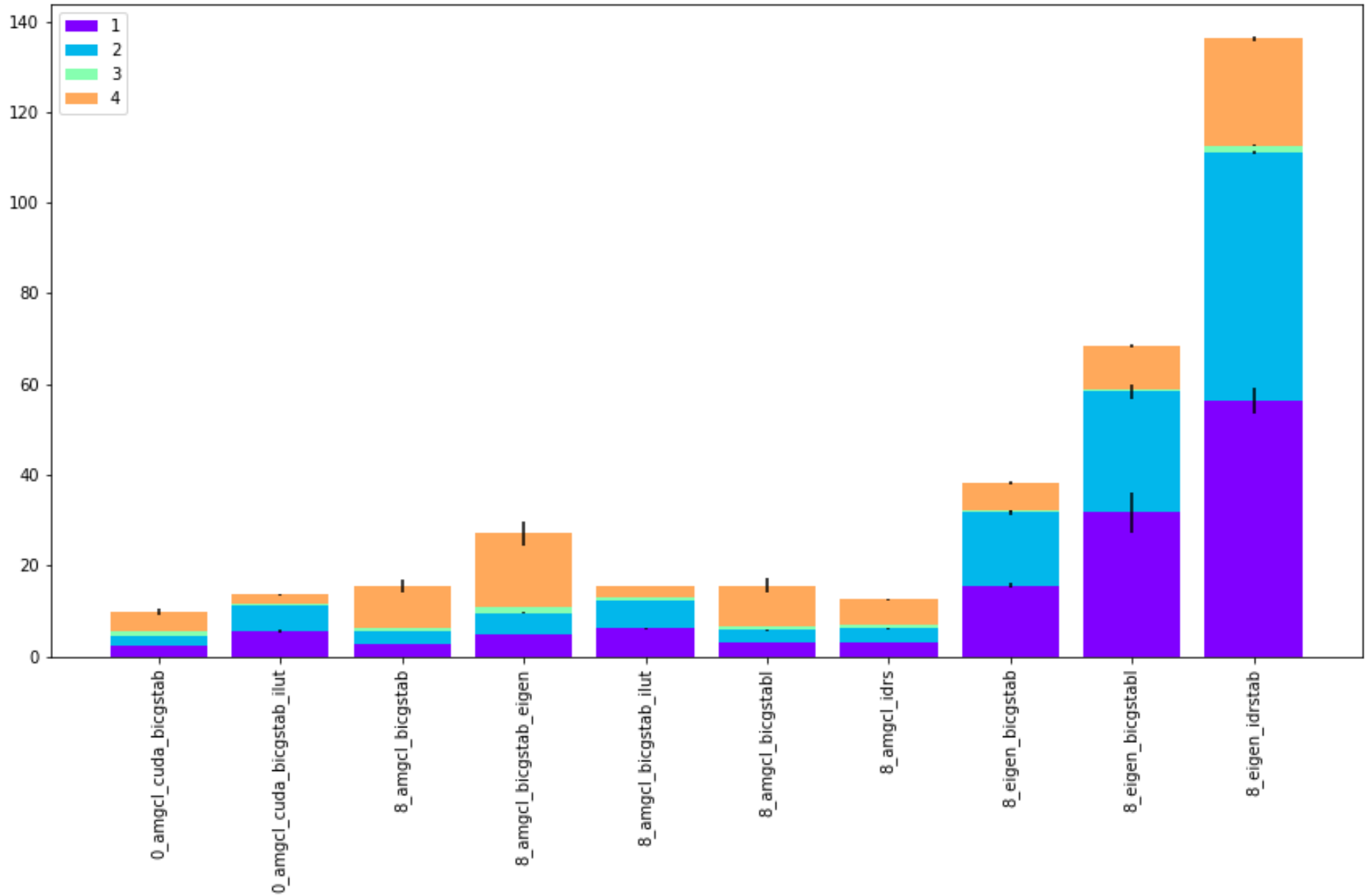
CUDA backend is fastest, AMG preconditioner is good for seq1 and seq2. Ilu0 might be better for seq3 and seq4

Setup vs solve

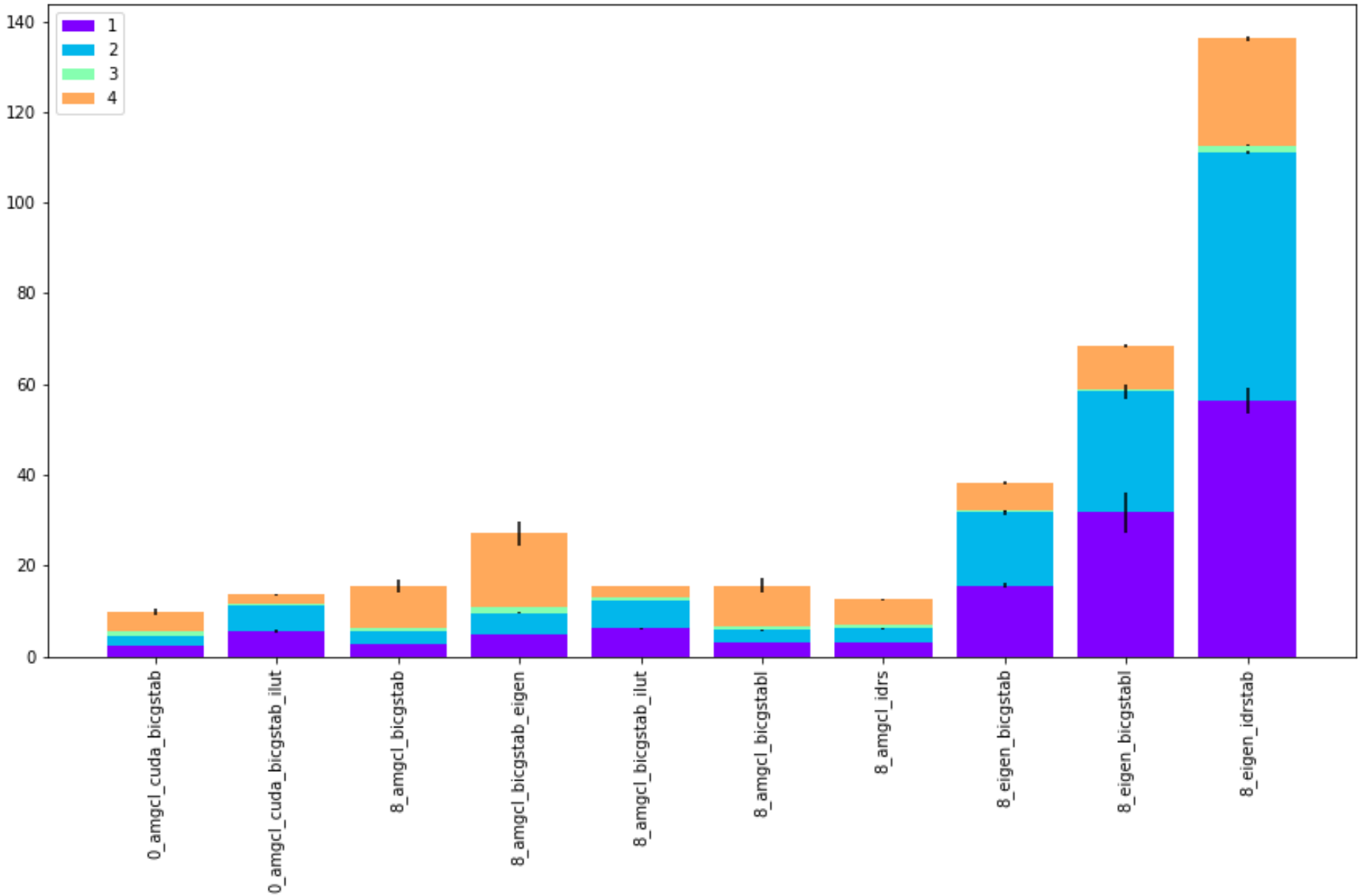


TU/e gum

iterations

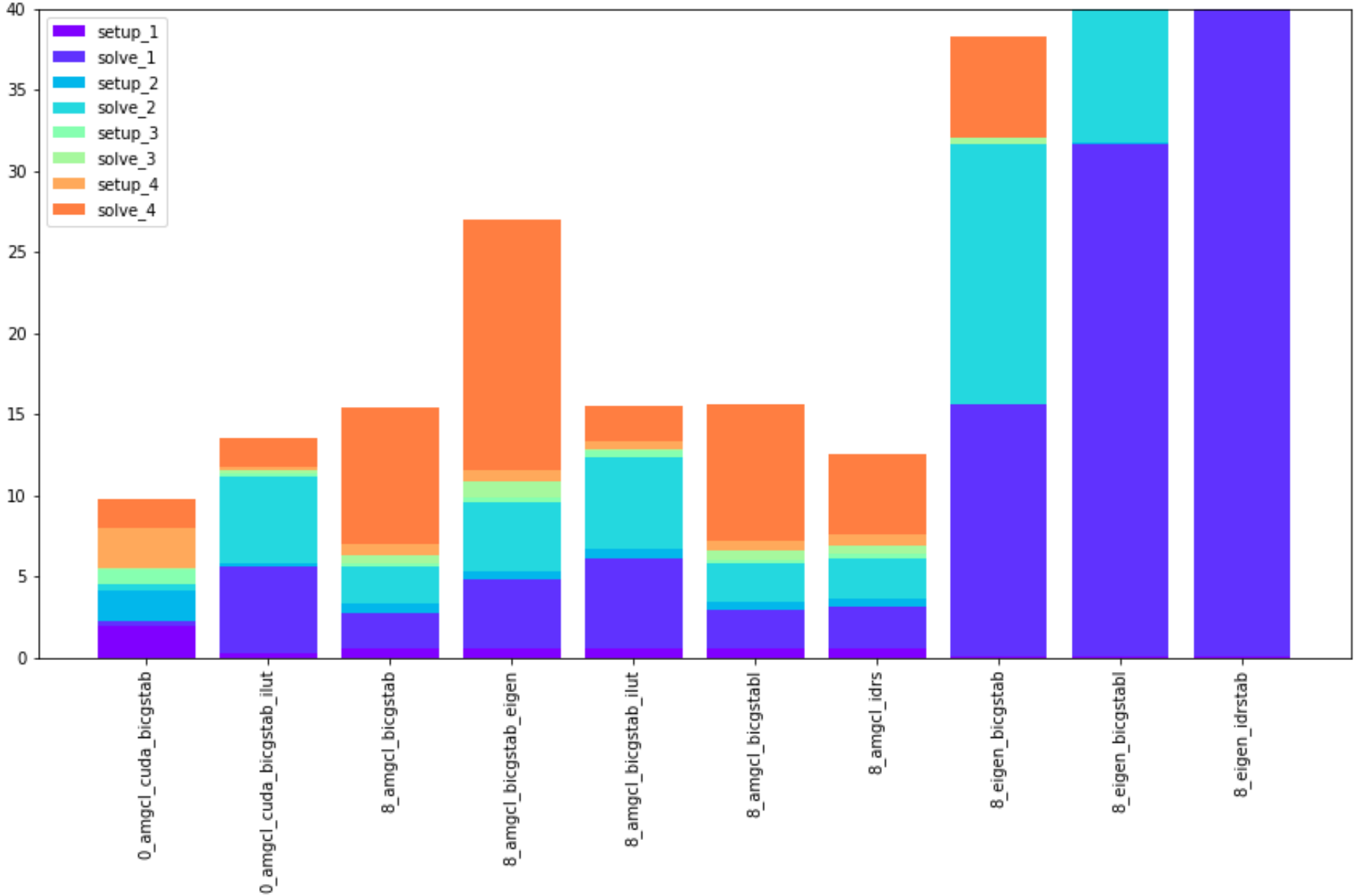

1
2
3
4
140
120
100
80
60
40
20
0
0_amgcl_cuda_bicgstab
0_amgcl_cuda_bicgstab_ilut
8_amgcl_bicgstab
8_amgcl_bicgstab_eigen
8_amgcl_bicgstab_ilut
8_amgcl_bicgstabl
8_amgcl_idrs
8_eigen_bicgstab
8_eigen_bicgstabl
8_eigen_idrstab

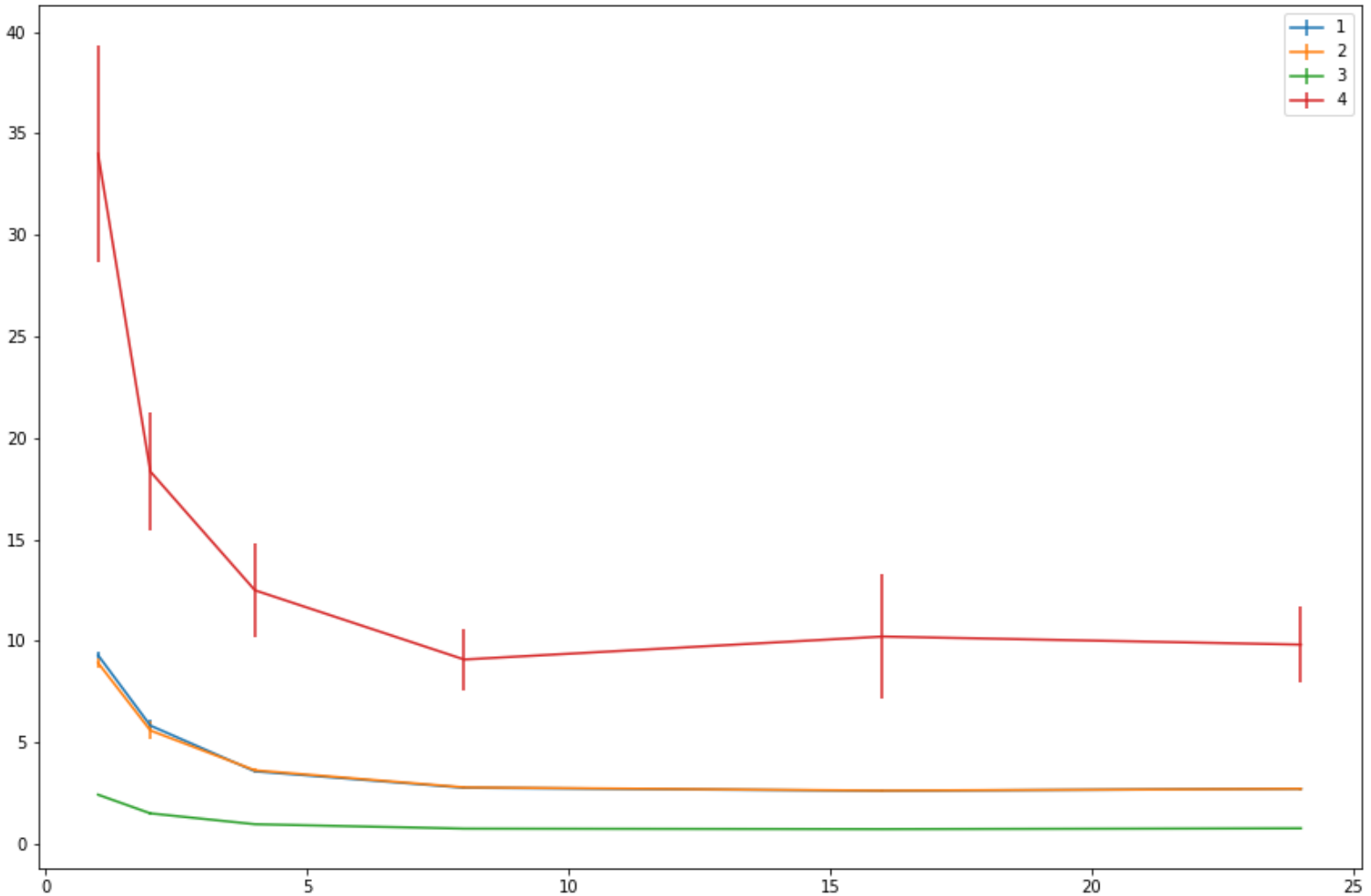
timings



For seq4 ilut is better than amgcl as a preconditioner

Timings setup vs solve





omp threads

No benefits after roughly 8 threads