Strong scaling comparison of Proto-based and Fortran-based elliptic operators

D. T. Graves

March 1, 2024

1 The point of all this

Chombo's Proto-based elliptic operators are still very new and need to be run through their paces with varying numbers of processors.

- All computations are run on the host (specfically, spencer.lbl.gov).
- All computations are run with 1, 2, 4, and 8 processors. More than that does not make sense for the very simple problems we are running (and 8 is too many in two dimensions) because the lowest level grid only has 2^D boxes.
- All runs have Dirichlet boundary conditions and very simple grids.
- All input files are pretty forgiving in terms of solvability and so on (coefficients set to reasonable constants and so on).
- The varying input files are really just making bigger grids

This framework should be fine for addresssing GPU issues but that will be a later document. Variable coefficient issues should be handled by adding cases.

2 Test description

There are eight operator classes we are considering.

- `AMRPoissonOp` (constant coefficient Helmholtz on host).
- `VCAMRPoissonOp2` (variable coefficient Helmholtz on host).
- `ViscousTensorOp` (viscous tensor on host).
- `ResistivityOp` (resistivity on host).
- `Proto_Helmholtz_Op` (constant coefficient Helmholtz on device).
- `Proto_Conductivity_Op` (variable coefficient Helmholtz on device).

- `Proto_Viscous Tensor_Op` (viscous tensor on device).
- `Proto_Resistivity_Op` (resistivity on device).

3 Results

Each class is run for both two and three dimensions. Each operator is run with input files for four separate cases. Each solve is done using 1, 2, 4, and 8 processors. For each case, all the operators L were used to solve

$$L\phi = 1$$

The cases used for this campaign are as follows.

- `case_0.inputs`: max level = 0, ncells = 32^D.
- `case_1.inputs`: max level = 1, ncells = 32^D.
- `case_2.inputs`: max level = 2, ncells = 32^D.
- `case_3.inputs`: max level = 2, ncells = 64^D.

With eight operators, five processor counts, and both two and three dimensions, this should amount to 320 runs.

4 Dataset from March 1, 2024

4.1 Summary of this round of data

Now that the multigrid convergence issues have been fixed, the data shows that proto-based operators are more competive with the fortran-based operators, occasionally even faster. For AMR problems, the proto-based coarse-fine interpolation is quite a bit simpler than the one used by the fortran-based operators. This might be the reason that the proto-based operators seem to look comparatively better on larger, deeper problems. Old Conductivity and old viscosity were not dumping timing info and the old resistivity operator is not working so I am going to just put Helmholtz for this data set.

Op	D	Case	N_p	Final $\|R\|$	Iter	Main Time
Proto_Helmholtz_Op	2	0	1	7.815970e-14	7	0.08826
Proto_Helmholtz_Op	2	0	2	7.815970e-14	7	0.07837
Proto_Helmholtz_Op	2	0	4	7.815970e-14	7	0.08190
Proto_Helmholtz_Op	2	0	8	DNF	DNF	DNF
AMRPoissonOp	2	0	1	6.672440e-14	7	0.05239
AMRPoissonOp	2	0	2	6.672440e-14	7	0.04826
AMRPoissonOp	2	0	4	6.672440e-14	7	0.05896
AMRPoissonOp	2	0	8	6.672440e-14	7	0.07726
Proto_Helmholtz_Op	2	1	1	9.476864e-13	7	0.09638
Proto_Helmholtz_Op	2	1	2	9.476864e-13	7	0.08375
Proto_Helmholtz_Op	2	1	4	9.476864e-13	7	0.08964
Proto_Helmholtz_Op	2	1	8	DNF	DNF	DNF
AMRPoissonOp	2	1	1	3.724798e-13	12	0.05891
AMRPoissonOp	2	1	2	3.724798e-13	12	0.05958
AMRPoissonOp	2	1	4	3.724798e-13	12	0.06769
AMRPoissonOp	2	1	8	3.724798e-13	12	0.09102
Proto_Helmholtz_Op	2	2	1	1.705303e-13	8	0.10884
Proto_Helmholtz_Op	2	2	2	1.705303e-13	8	0.09641
Proto_Helmholtz_Op	2	2	4	1.705303e-13	8	0.10481
Proto_Helmholtz_Op	2	2	8	DNF	DNF	DNF
AMRPoissonOp	2	2	1	9.429124e-13	12	0.05868
AMRPoissonOp	2	2	2	9.429124e-13	12	0.06301
AMRPoissonOp	2	2	4	9.429124e-13	12	0.06858
AMRPoissonOp	2	2	8	9.429124e-13	12	0.09020
Proto_Helmholtz_Op	2	3	1	3.126388e-13	8	0.12383
Proto_Helmholtz_Op	2	3	2	3.126388e-13	8	0.10778
Proto_Helmholtz_Op	2	3	4	3.126388e-13	8	0.10823
Proto_Helmholtz_Op	2	3	8	DNF	DNF	DNF
AMRPoissonOp	2	3	1	3.135270e-13	13	0.06820
AMRPoissonOp	2	3	2	3.135270e-13	13	0.07243
AMRPoissonOp	2	3	4	3.135270e-13	13	0.08025
AMRPoissonOp	2	3	8	3.135270e-13	13	0.10268

Table 1: Performance data for the solution of the Helmholtz equation in two dimensions. The data set is dated March 1, 2024.

Op	D	Case	N_p	Final $\|R\|$	Iter	Main Time
Proto_Helmholtz_Op	3	0	1	3.375078e-13	8	0.40241
Proto_Helmholtz_Op	3	0	2	3.339551e-13	8	0.25636
Proto_Helmholtz_Op	3	0	4	3.339551e-13	8	0.17792
Proto_Helmholtz_Op	3	0	8	3.339551e-13	8	0.18492
AMRPoissonOp	3	0	1	3.489431e-13	8	0.10812
AMRPoissonOp	3	0	2	3.367306e-13	8	0.08406
AMRPoissonOp	3	0	4	3.366196e-13	8	0.08065
AMRPoissonOp	3	0	8	3.366196e-13	8	0.10286
Proto_Helmholtz_Op	3	1	1	1.563194e-13	9	0.53076
Proto_Helmholtz_Op	3	1	2	1.847411e-13	9	0.35309
Proto_Helmholtz_Op	3	1	4	1.847411e-13	9	0.25635
Proto_Helmholtz_Op	3	1	8	1.847411e-13	9	0.24127
AMRPoissonOp	3	1	1	5.592193e-13	13	0.15747
AMRPoissonOp	3	1	2	5.591083e-13	13	0.12062
AMRPoissonOp	3	1	4	5.591083e-13	13	0.11600
AMRPoissonOp	3	1	8	5.591083e-13	13	0.13738
Proto_Helmholtz_Op	3	2	1	2.273737e-13	9	0.58475
Proto_Helmholtz_Op	3	2	2	2.557954e-13	9	0.40092
Proto_Helmholtz_Op	3	2	4	2.131628e-13	9	0.29642
Proto_Helmholtz_Op	3	2	8	2.131628e-13	9	0.29994
AMRPoissonOp	3	2	1	2.939871e-13	14	0.18959
AMRPoissonOp	3	2	2	3.027578e-13	14	0.13635
AMRPoissonOp	3	2	4	2.593481e-13	14	0.12618
AMRPoissonOp	3	2	8	2.593481e-13	14	0.15533
Proto_Helmholtz_Op	3	3	1	5.115908e-13	9	1.74798
Proto_Helmholtz_Op	3	3	2	5.115908e-13	9	0.99395
Proto_Helmholtz_Op	3	3	4	5.115908e-13	9	0.58095
Proto_Helmholtz_Op	3	3	8	5.115908e-13	9	0.43825
AMRPoissonOp	3	3	1	5.895284e-13	14	0.86169
AMRPoissonOp	3	3	2	6.024070e-13	14	0.52386
AMRPoissonOp	3	3	4	6.024070e-13	14	0.27940
AMRPoissonOp	3	3	8	6.024070e-13	14	0.25845

Table 2: Performance data for the solution of Helmholtz equation in three dimensions. The data set is dated March 1, 2024.