Experiments with Staggered Multigrid

Alexei Strelchenko

Fermilab, Batavia, IL

in collaboration with Richard Brower, Kate Clark, Evan Weinberg

QCDNA 2016, Edinburgh

Outline

- Introduction
- Designing MG for the staggered operator
- Adapting AMG to QUDA framework
- Results
- Conclusions and future work

Motivation

- Push to exascale enables increasingly accurate lattice calculations.
- Physical pion mass, finer lattices: critical slowing down.
  - MILC: $144^3 \times 288$, physical pion mass, single precision multi-mass solve to rel. resid. 10^{-6}: about 25,000 iterations.
- New measurements (disconnected diagrams) require more $\not{D}$ inversions.
- ... and this is a missing point in QUDA

$32^3 \times 48$ **HISQ** ($M_\pi = 131.0MeV$)

Mixed precision incremental eigCG performance, 800 eigenvectors, $32^3 \times 48$ HISQ $2+1+1$ configuration (6 nodes, 24 K40m)

Solver type	Inc. NRHS / Time	CG Iters	CG Time
inc. eigCG (full)	100 (1534)	790(x10.1)	1.48 (x7.8)
inc. eigCG (mixed)	46 (642)	1255 (x6.4)	2.13 (x5.4)
undef. CG (mixed)	N.A.	8000 (x1.0)	11.5 (x1.0)

The Staggered Operator

$$D_{xy} = \sum_{\mu} \eta_\mu(x) \left[U_\mu^\dagger(x)\delta_{y+\mu,x} - U_\mu(x)\delta_{x+\mu,y} \right] + m\delta_{x,y}$$

$$= iA + m\mathbf{I} \quad \leftarrow \text{Anti-Hermitian} + \text{Hermitian piece}$$

Schur decomposition:

$$\begin{bmatrix} m & D_{eo} \\ D_{oe} & m \end{bmatrix} = \begin{bmatrix} 1 & \frac{D_{eo}}{m} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \left(m^2 - D_{eo}D_{oe}\right) & 0 \\ 0 & m^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{D_{oe}}{m} & 1 \end{bmatrix}$$

The normal equation

Due to $D_{eo} = -D_{oe}^{\dagger}$,
the E/O preconditioned operator is Hermitian positive definite:
Rich theory of multigrid exists.

$$\left(m^2 - D_{eo}D_{oe}\right)\psi_e = mb_e - D_{eo}b_o; \qquad \psi_o = \frac{1}{m}\left(b_o - D_{oe}\psi_e\right)$$

Issues:

- Two link sparsity pattern is computationally inefficient.
- Further issue: Even and odd decouple in chiral limit.

Follow lead of Wilson-Clover MG:
Directly precondition $\not{D}$.

Adaptive Algebraic Multigrid

- Algorithm setup for 2-level V-Cycle:
  1. Relax on homogeneous problem $\not{D}\vec{x}_i = 0$ for N_{vec} random $\vec{x}_0$. (Each vector = extra degree of freedom per coarse site.)
  2. Cut vectors $\vec{x}_i$ into geometrically regular subsets to be aggregated (blocked)
  3. Block orthonormalize vectors $\vec{x}_i$
  4. This defines the prolongator such that $\left(1 - PP^\dagger\right)\vec{x}_i = 0$
  5. Define coarse grid operator $\not{D}_c = P^\dagger \not{D} P$

Designing Staggered AMG

$$D_{xy} = \sum_{\mu} \eta_\mu(x) \left[U_\mu^\dagger(x)\delta_{y+\mu,x} - U_\mu(x)\delta_{x+\mu,y} \right] + m\delta_{x,y}$$
$$= iA + m\mathbf{I} \quad \leftarrow \text{Anti-Hermitian} + \text{Hermitian piece}$$

The "Gamma-5" symmetry ; $D^\dagger = \gamma_5^{stag} D \gamma_5^{stag}$

Unique to staggered: $\gamma_5^{stag} \equiv \epsilon(x) = (-1)^{\sum_\mu x_\mu}$
$(1 \pm \gamma_5^{stagg})$ is just parity (even/odd) projection operator.

Parity Aggregation

Similar to Wilson, every right eigenv. v_λ corrsponds to a left eigenv. $\tilde{v}_{\bar{\lambda}}$:

$$\tilde{v}_{\bar{\lambda}} = \gamma_5^{stag} v_\lambda$$

$$v_\lambda = \begin{pmatrix} v_\lambda^e \\ v_\lambda^o \end{pmatrix} \rightarrow \tilde{v}_\lambda = \begin{pmatrix} v_\lambda^e \\ -v_\lambda^o \end{pmatrix}$$

I.e., the staggered γ_5-compitibile aggregation:

$$v_i = \begin{pmatrix} v_i^e \\ v_i^o \end{pmatrix} \rightarrow \begin{pmatrix} v_i^e \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ v_i^o \end{pmatrix}$$

Constructing intergrid operators

The staggered γ_5-compitibile aggregation:

$$P_{x,c;\hat{x},\hat{c},s} = \left[V_{\hat{x},\hat{c},s}(x,c)\right], \quad R = P^\dagger$$

On coarse grid we get spin-2 fileds:

$$\begin{pmatrix} w_i^e(x,c) \\ 0 \end{pmatrix} = P_{x,c;s=0}\hat{w}_{s=0}, \begin{pmatrix} 0 \\ w_i^o(x,c) \end{pmatrix} = P_{x,c;s=1}\hat{w}_{s=1}$$

$$P^\dagger \gamma_5^{stagg} = \sigma_3 P^\dagger$$

Coarse grid operator

Coarse-grid staggered operator has structure similar to Wilson case!

$$\begin{bmatrix} M^{ss'}_{\hat{e}\hat{e}} & D^{ss'}_{\hat{e}\hat{o}} \\ D^{ss'}_{\hat{o}\hat{e}} & M^{ss'}_{\hat{o}\hat{o}} \end{bmatrix} = \begin{bmatrix} m & M_{\hat{e}\hat{e}} & 0 & D^{01}_{\hat{e}\hat{o}} \\ M^{\dagger}_{\hat{e}\hat{e}} & m & D^{10}_{\hat{e}\hat{o}} & 0 \\ 0 & D^{01}_{\hat{o}\hat{e}} & m & M_{\hat{o}\hat{o}} \\ D^{10}_{\hat{o}\hat{e}} & 0 & M^{\dagger}_{\hat{o}\hat{o}} & m \end{bmatrix}$$

I.e., integration to QUDA MG framework is straightforward.

Testing Environment

- Use a simpler model *with similar physics* for initial testing:
- Two-flavor Schwinger model in two dimensions:

$$\mathcal{L} = \frac{1}{2}F^2 - i\sum_{f=1,2}\bar{\psi}_f\gamma^\mu\left(\partial_\mu - igA_\mu\right)\psi_f + m\sum_{f=1,2}\bar{\psi}_f\psi_f$$

  - Confinement
  - Chiral symmetry breaking
  - Vorticies (2 dimensional "instantons")
  - Topology
  - Two flavor theory has a "pion"-like state: $M_{gap}(m) = A_{gap}m^{2/3}g^{1/3}$ [Phys.Rev. D55 (1997)]
- Comparatively very inexpensive to look at large volumes: 128^2

Our Prescription

Remark: Our presecription isn't final—this just fixes our tests!

- Null vector generation:
  - Use (full precision) BiCGStab
  - Generate random vector $\vec{x_0}$.
  - Solve to tolerance 5×10^{-5}.
  - Generate N_{vec} null vectors.
- MG Solve:
  - One-level V cycle
  - Outer solver: GCR(24) to tolerance 5×10^{-7}
  - Block size: 4×4
  - Pre- and Post-Smoother: 6 iterations of GCR
  - Inner solver: GCR(64) to tolerance 10^{-3}

Outer Iterations: Free Case

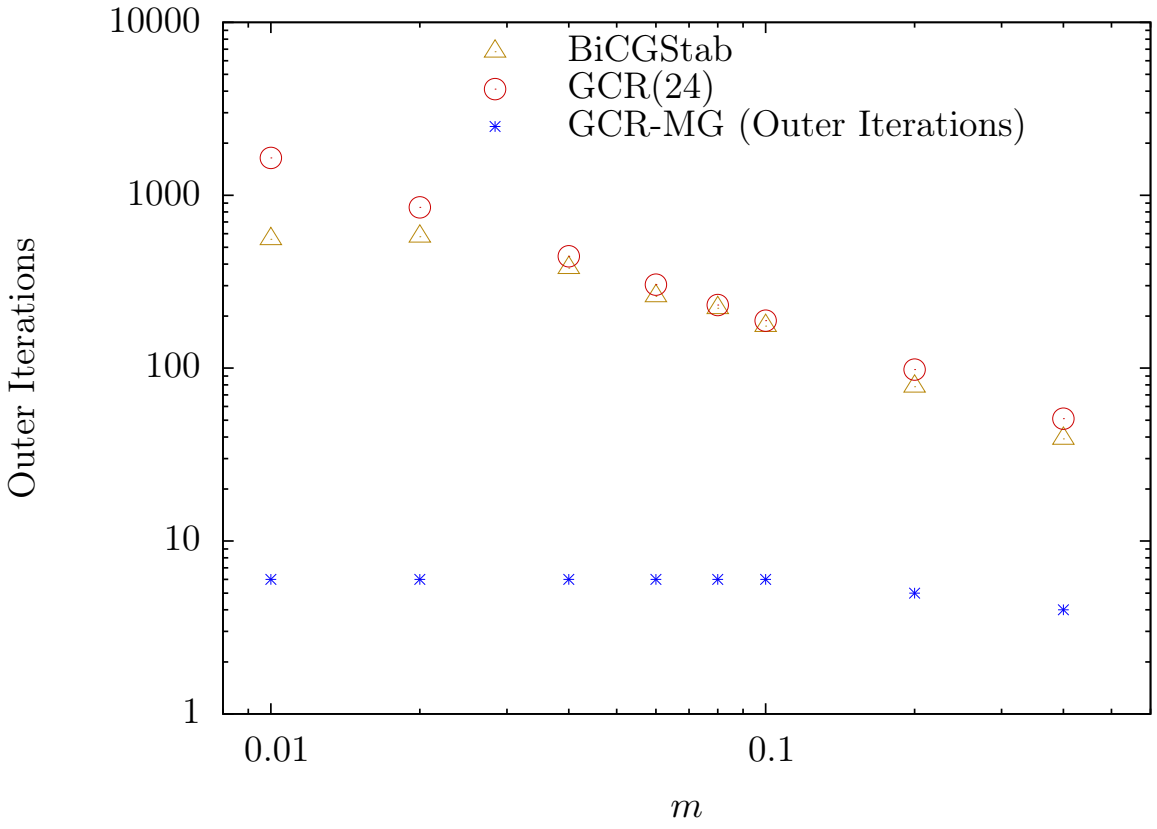

Figure : Outer Iterations for 128^2, free field

Demonstrates working (free algorithm), but not a "fair" comparison.

Fine $\not{D}$ Operators: Free Case

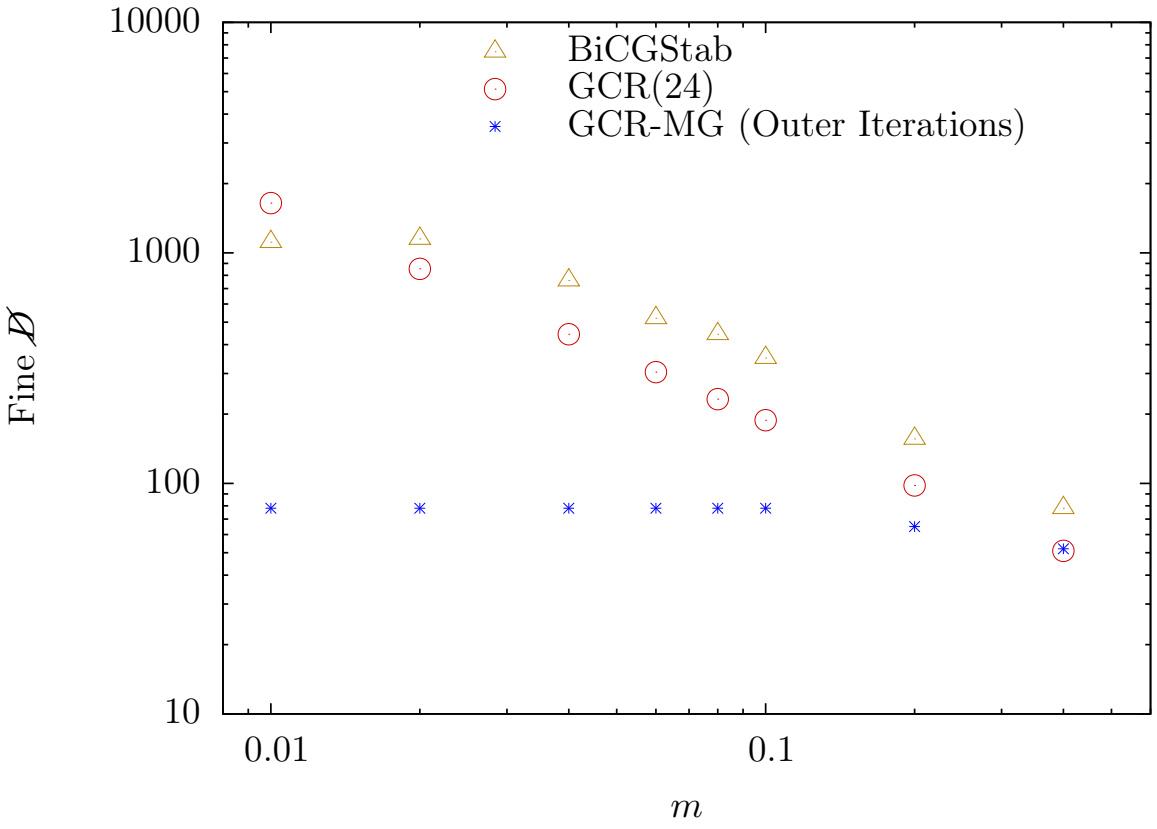

Figure : Fine $\not{D}$ for 128^2, free field

For large mass, MG doesn't make a difference...
There's no critical slowing down to remove!

Interacting Case

Quenched Schwinger model:
Non-Compact 2D $U(1)$ gauge action.

β	6.0	10.0	[Phys.Rev.Lett. 100 (2008): J. Brannick, R.C. Brower, M.A. Clark, J.C. Osborn, C. Rebbi]
l_σ	3.30	4.35	Gauge correlation length via Wilson loop: $W \approx e^{-A/l_\sigma^2}$
μ^{-1}	≈ 5.5	≈ 6.3	Pseudoscalar meson correlation length for $m = 0.01$

Interacting Case

Quenched Schwinger model:
Non-Compact 2D $U(1)$ gauge action.

β	6.0	10.0	[Phys.Rev.Lett. 100 (2008): J. Brannick, R.C. Brower, M.A. Clark, J.C. Osborn, C. Rebbi]
l_σ	3.30	4.35	Gauge correlation length via Wilson loop: $W \approx e^{-A/l_\sigma^2}$
μ^{-1}	≈ 5.5	≈ 6.3	Pseudoscalar meson correlation length for $m = 0.01$

As a representative ensemble, we will look in depth at $128^2, \beta = 10.0$.

Fine $\not{D}$: Interacting, $\beta = 10.0$

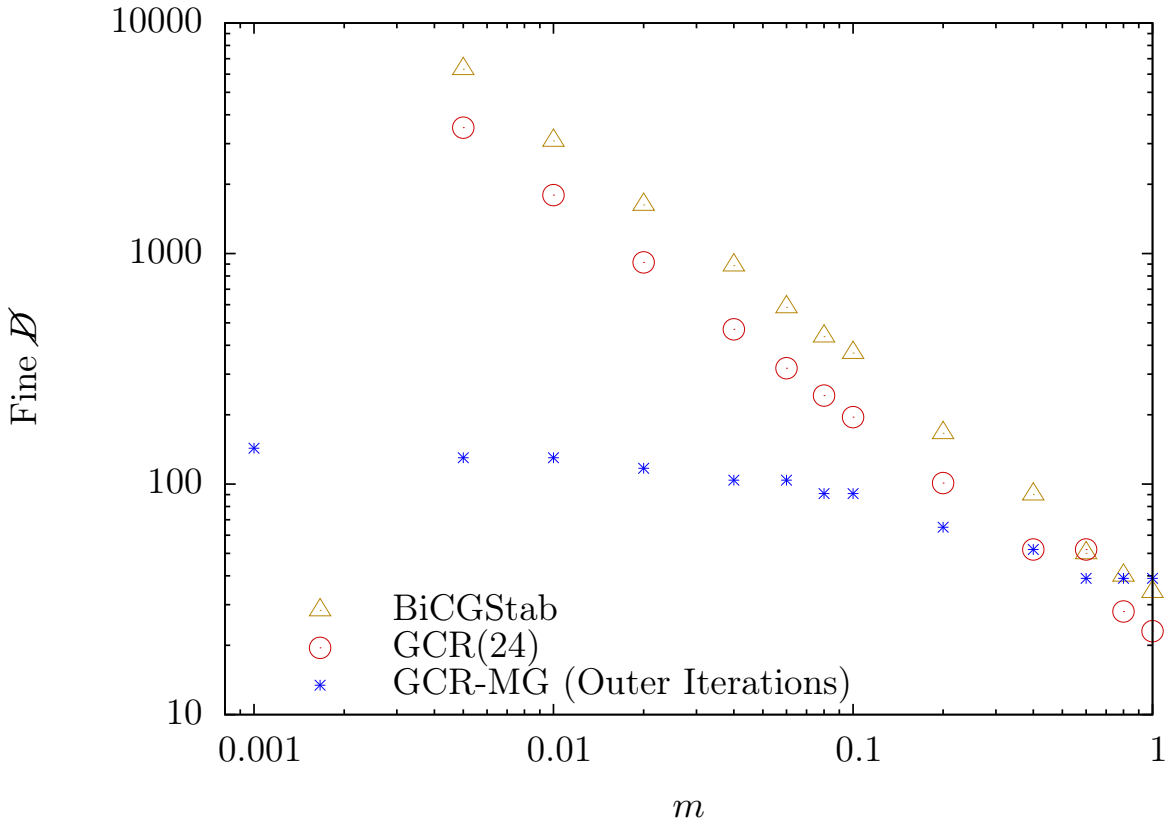

Figure : Fine $\not{D}$ for $128^2, \beta = 10.0, N_{vec} = 4$

For small m, critical slowing down is suppressed.

Residual per Iteration: Interacting, $\beta = 10.0$

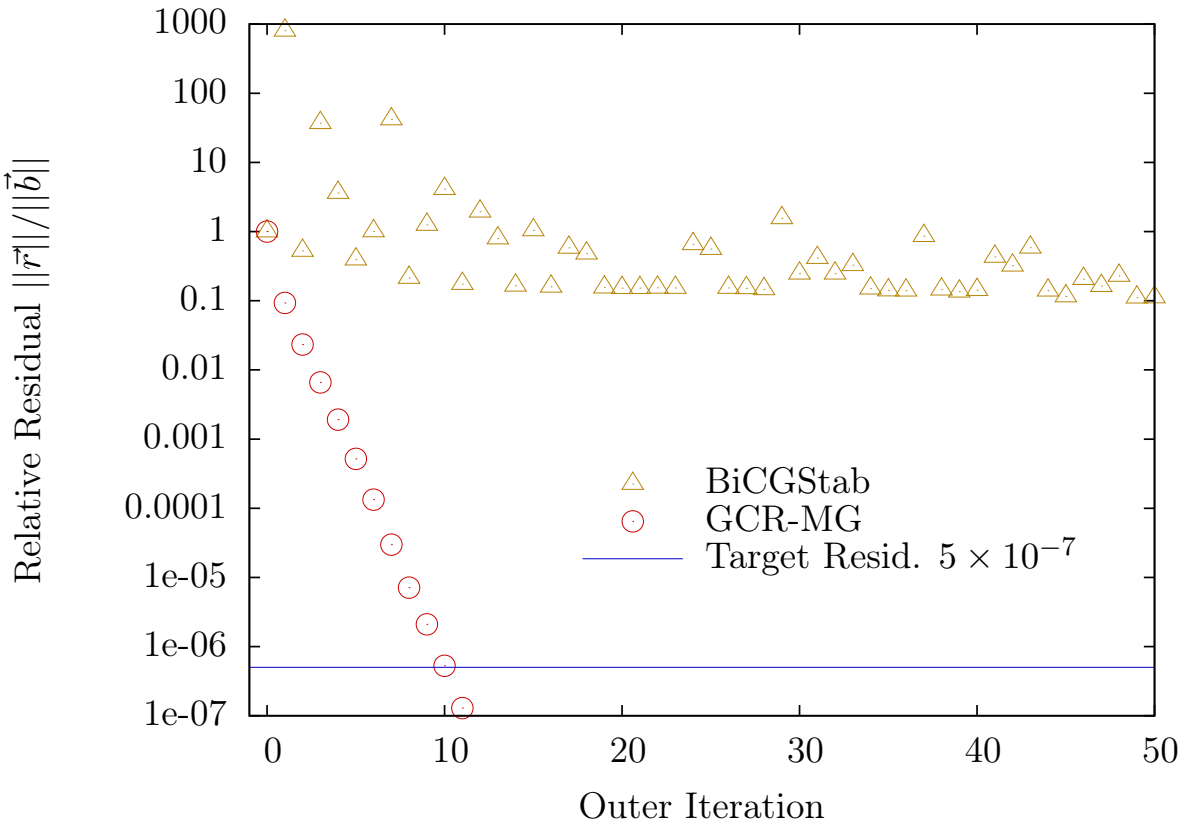

Figure : Relative Residual for $128^2, \beta = 10.0, N_{vec} = 4, m = 10^{-3}$

MG can stabilize otherwise unstable solves.

Number of Null Vectors: Interacting, $\beta = 10.0$

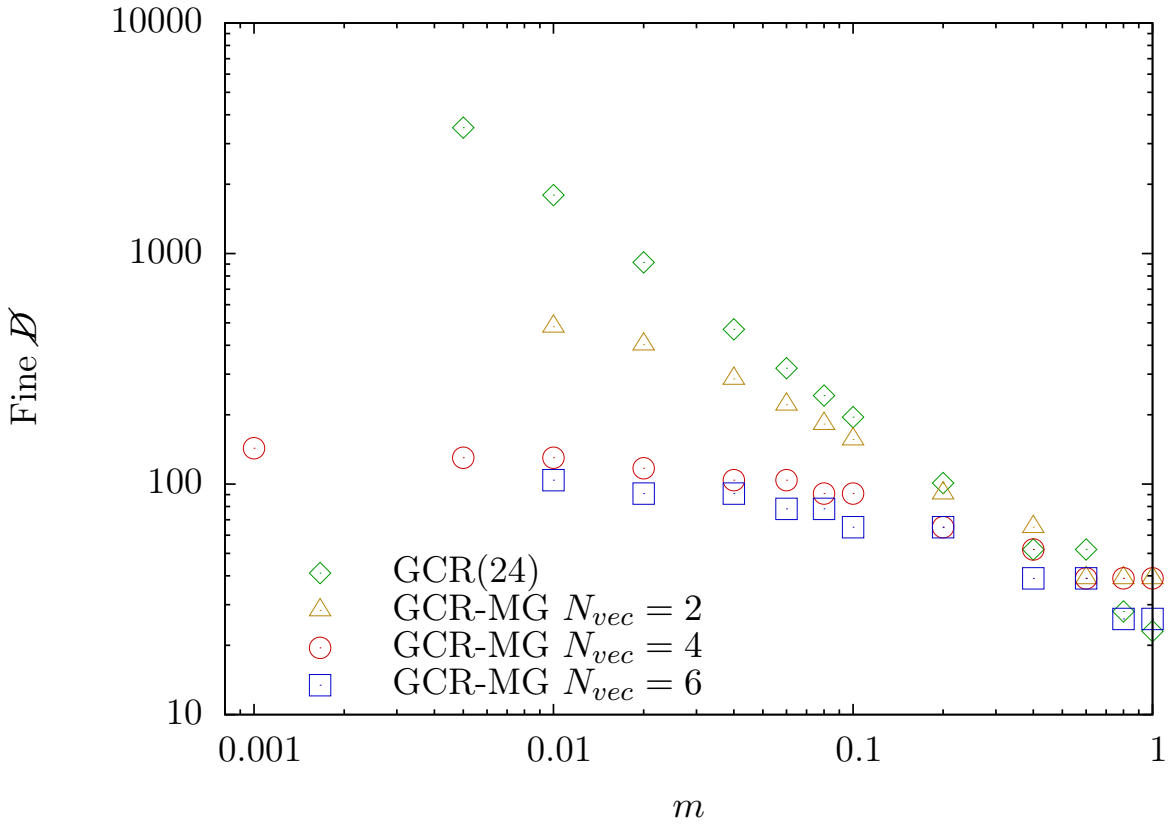

Figure : Comparison of different N_{vec} for $128^2, \beta = 10.0$

Diminishing returns with more N_{vec}: not a problem!

Coarse Solve: Interacting, $\beta = 10.0$

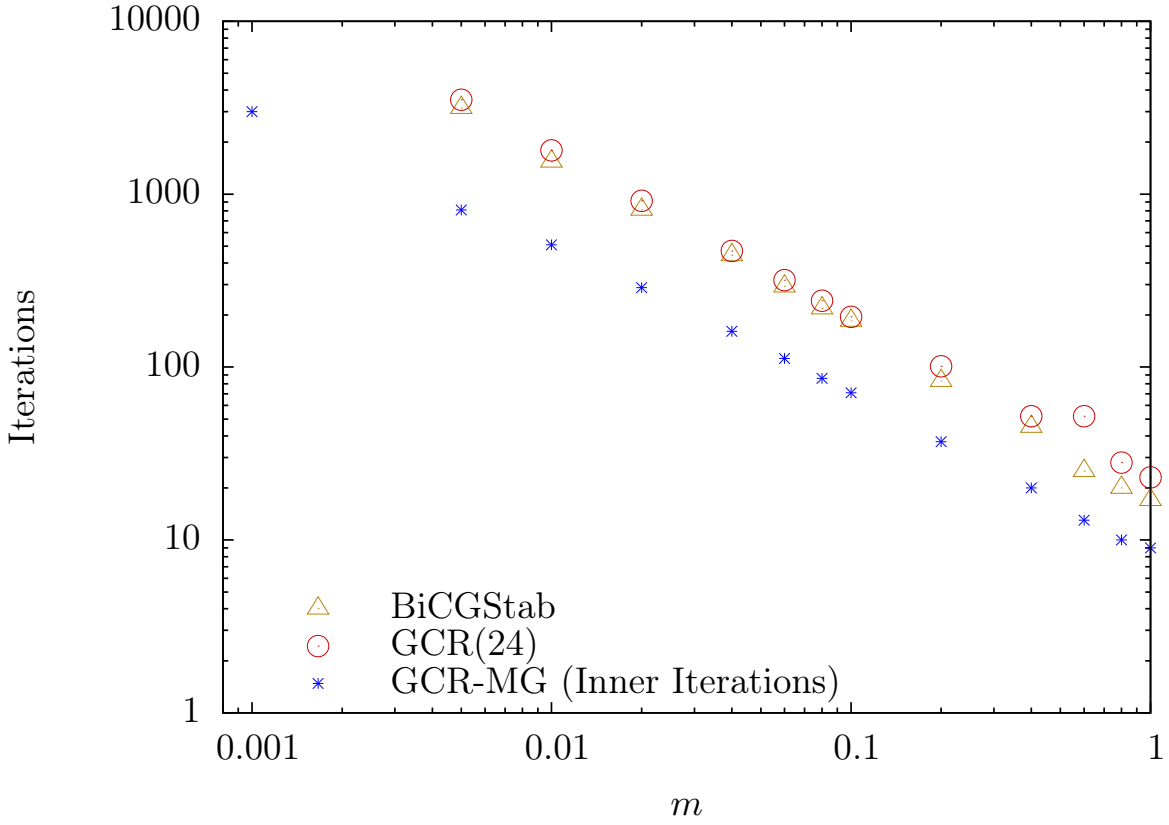

Figure : Inner Iterations for $128^2, \beta = 10.0, N_{vec} = 4$

Critical slowing down moves to coarse operator.
Still coarser levels will cure this.

Fine $\not{D}$: Interacting, $\beta = 6.0$

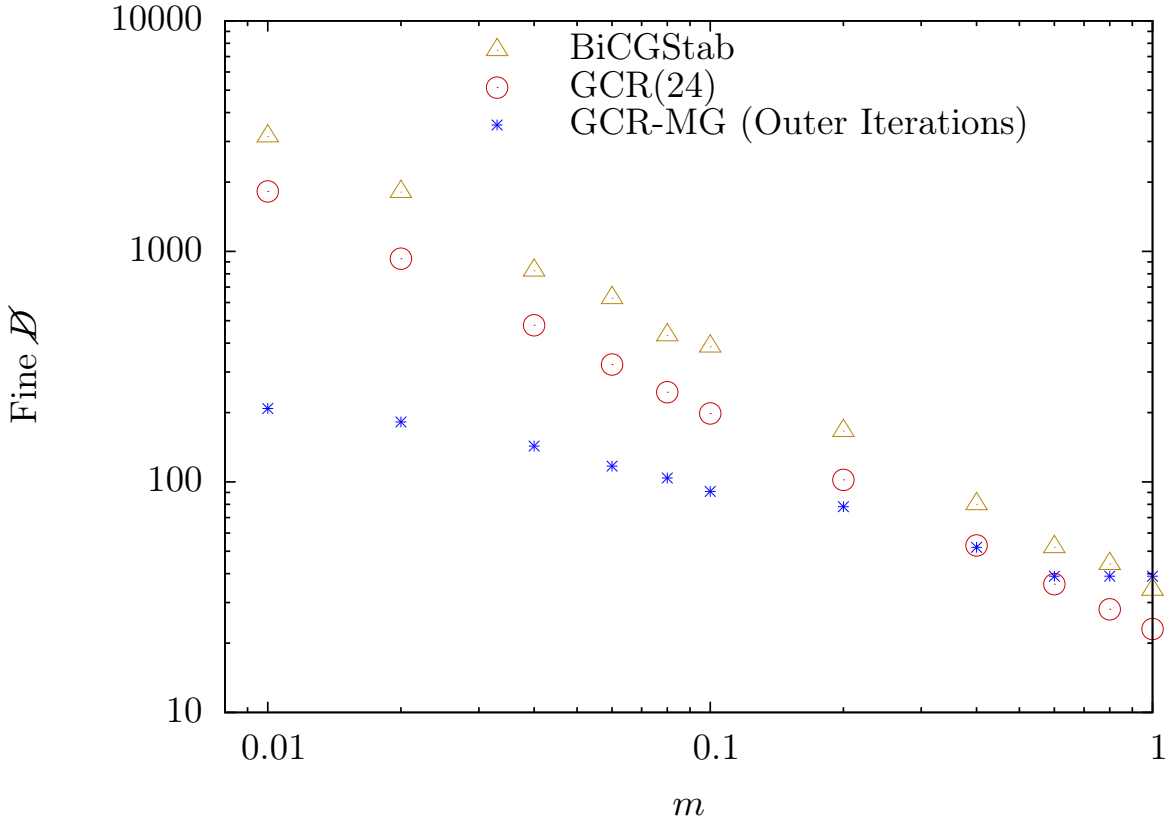

Figure : Fine $\not{D}$ for $128^2, \beta = 6.0, N_{vec} = 4$

MG still works for coarser β.

Overview and Future Work

Overview

- Showed a working MG algorithm for two-flavor Schwinger model

Future Work

- Investigate 4 dimensional staggered operator: naïve, HISQ
- Continue progress on an implementation in QUDA
  - Extended existing infrastructure for Wilson-Clover
  - Implementation there, further room for tuning and optimization.
  - `github.com/lattice/quda/tree/feature/staggered-multigrid`
- Experiment with alternative methods to generate null vectors
  - Can we use the normal equation to generate null vectors?
  - If we can, can we use approximate low vectors from EigCG instead?
    - This lets us do "real" solves while generating null vectors!
  - or we can use an eigensolver for the direct system?