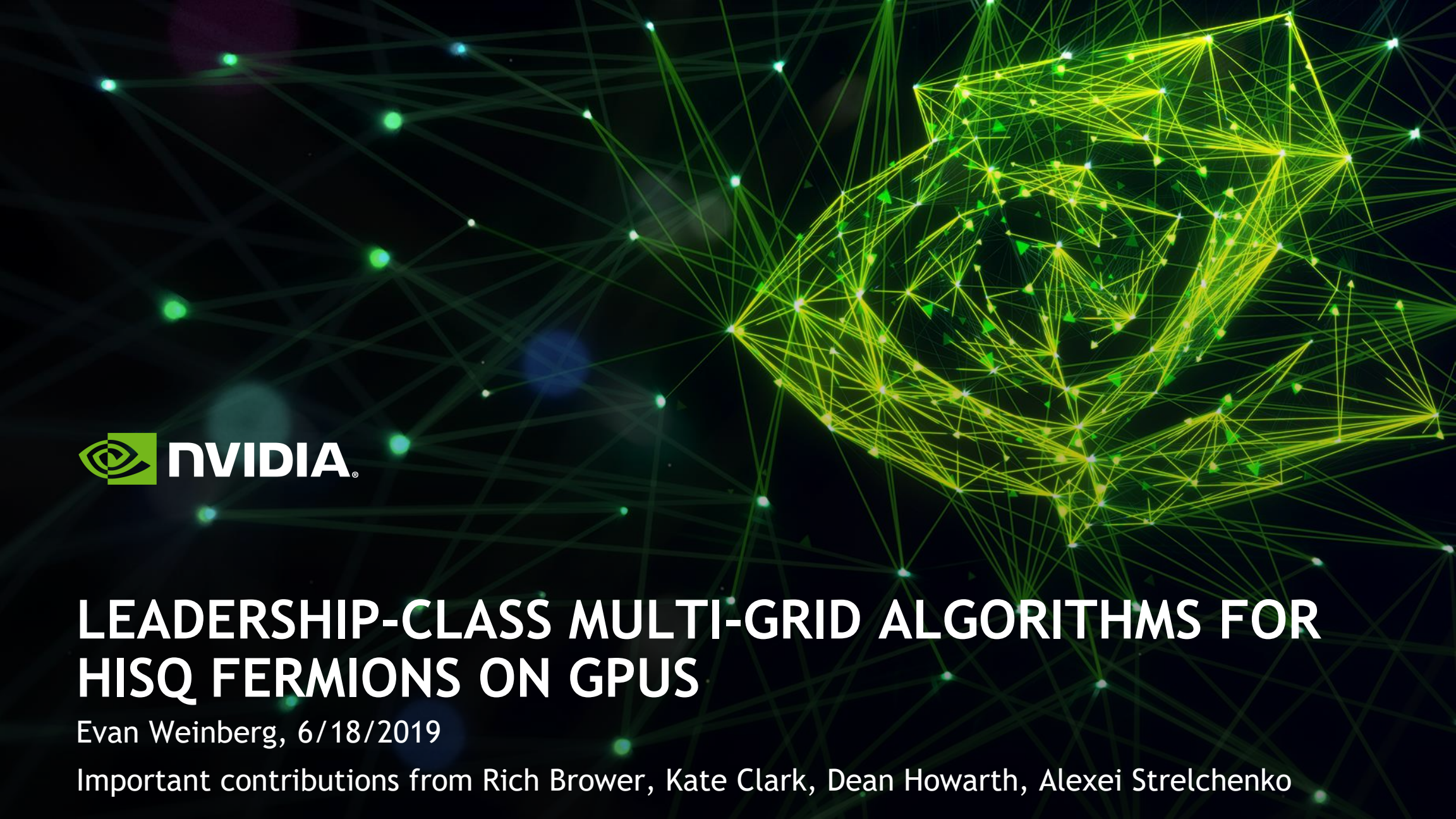


LEADERSHIP-CLASS MULTI-GRID ALGORITHMS FOR HISQ FERMIONS ON GPUS

Evan Weinberg, 6/18/2019

Important contributions from Rich Brower, Kate Clark, Dean Howarth, Alexei Strelchenko

AGENDA

Staggered MG Formalism: arXiv:1801.07823

Results from Physical Point HISQ Configurations


STAGGERED MG FORMALISM

STAGGERED OPERATOR

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

$$\eta_{\mu}(x) = (-1)^{\sum_{\nu<\mu} n_{\nu}}$$

D_{xy} is anti-Hermitian indefinite, m is a real shift

Chirality: $\varepsilon(x) = (-1)^{\sum_{\mu} n_{\mu}}$, $\varepsilon(x)D_{xy}$ is Hermitian indefinite

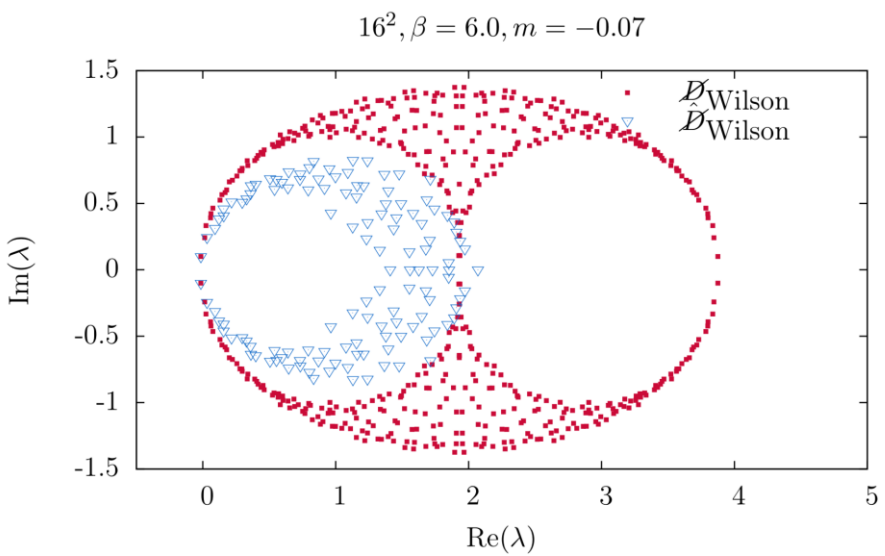
Remark: The HISQ operator adds a Naik (distance-three) term.

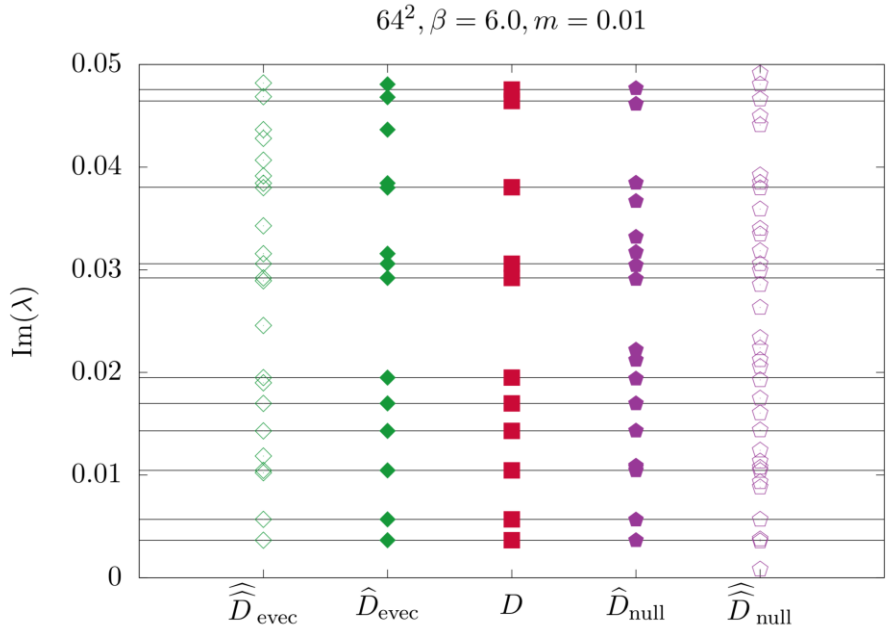
WHY IS STAGGERED MG HARD?

Naïve Galerkin projection does not work

Spurious low modes on coarse grids

System gets worse conditioned as we progressively coarsen





Wilson: high modes are gapped in the real direction

Coarsen: collapses, but not to the complex origin.

arXiv:1801.07823

NVIDIA

OBSERVATION: KAHLER-DIRAC EQUIVALENCE

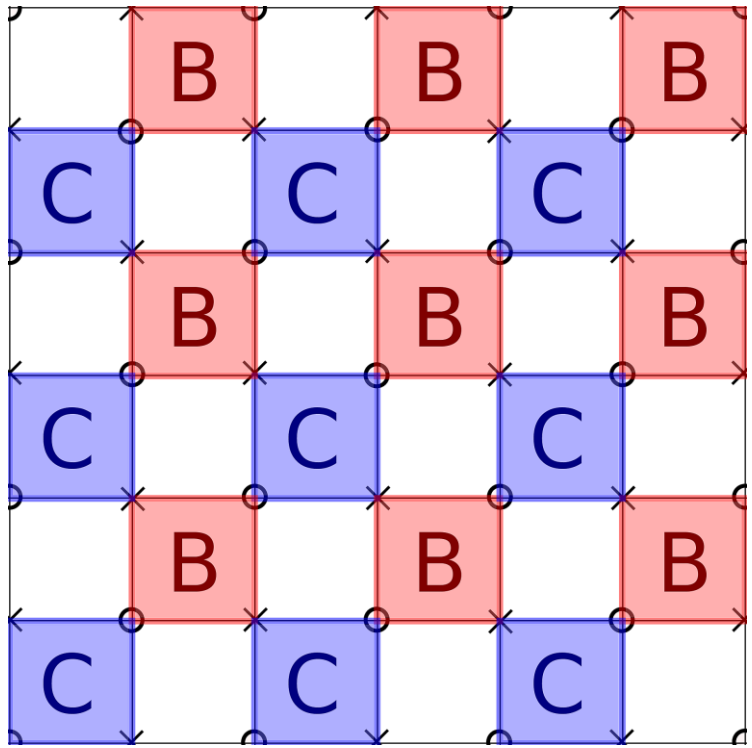
Staggered fermions distribute 2^d degrees of freedom over a 2^d hypercube of sites

Lattice Kähler-Dirac fermion has 2^d dof per site: free field equivalence (timeless, but well explained in arXiv:0509026 Dürr, among many others)

What can this equivalence teach us?

MATHEMATICAL MOTIVATION

- Consider a dual decomposition of the lattice
- B: hopping terms within a 2^d block
- C: hopping terms across 2^d blocks
- $D^{stag} = B + C + m$
- Up to a scaling factor, $B^2 \sim |B + m|^2 \sim C^2 \sim I$, implies $\sim \pm 1$ eigenvalues



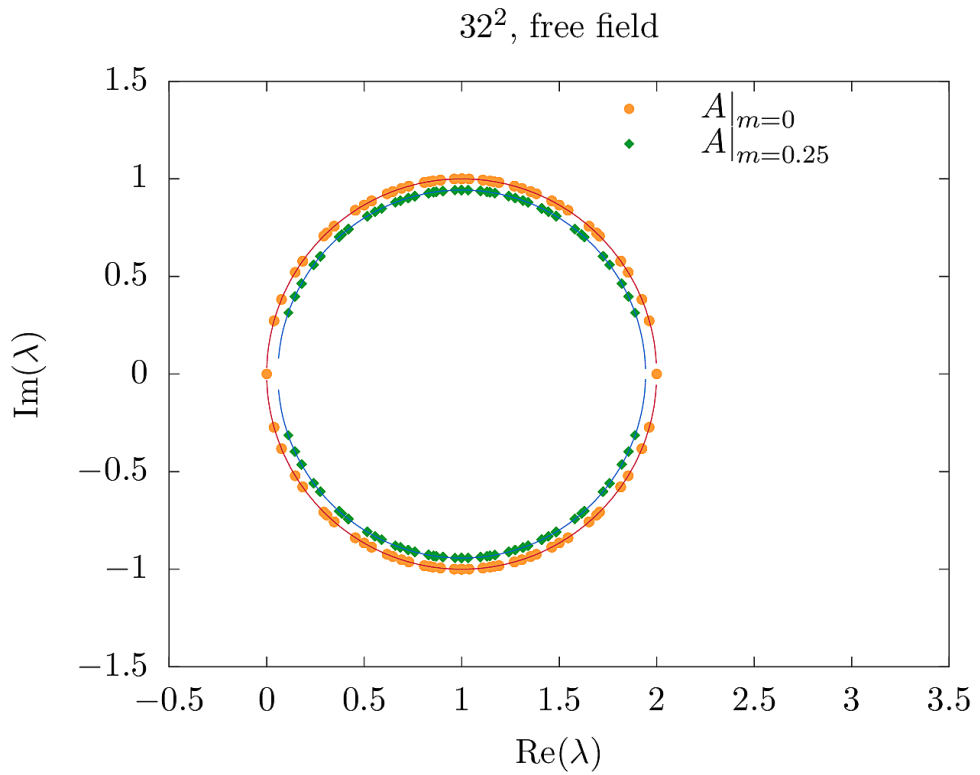
MATHEMATICAL MOTIVATION

- Consider a dual decomposition of the lattice
- B: hopping terms within a 2^d block
- C: hopping terms across 2^d blocks
- $D^{stag} = B + C + m$
- Up to a scaling factor, $B^2 \sim |B + m|^2 \sim C^2 \sim I$, implies $\sim \pm 1$ eigenvalues

$$A = (B + m)^{-1}(C + B + m)$$

$$= \underbrace{(B + m)^{-1}C}_{\sqrt{\frac{d}{d+m^2}}U} + I$$

Unitary!

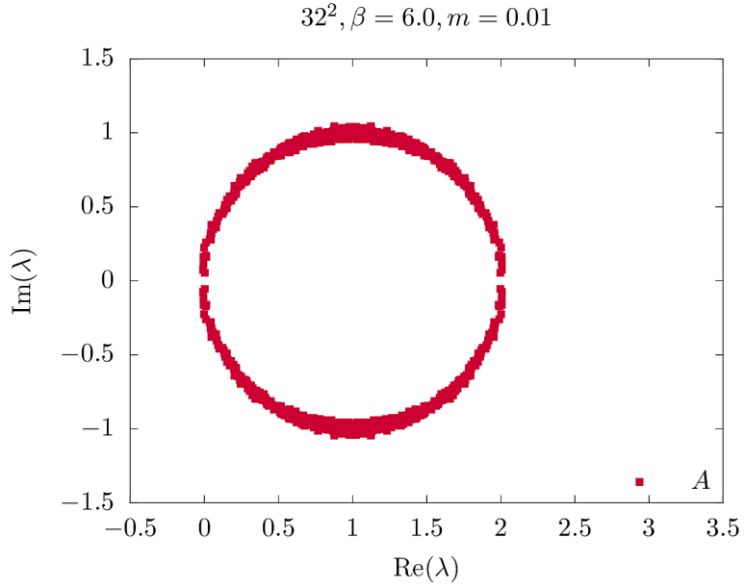


ALGORITHMIC IMPLEMENTATION

- In practice, $(B + m)^{-1} D^{stag}$ is a **block-Jacobi** preconditioning of the staggered operator by the Kähler-Dirac block
- Literally form the Kähler-Dirac operator: one site with $N_c \, 2^d$ degrees of freedom
- Implementation: MG "coarsen" a 2^d aggregate with $N_c \, 2^{d-1}$ near-null vectors (times 2 for chirality)
- $B + m$ is the "coarse clover", C is the "coarse hopping term"
- Block-Jacobi precondition, then run MG from there!

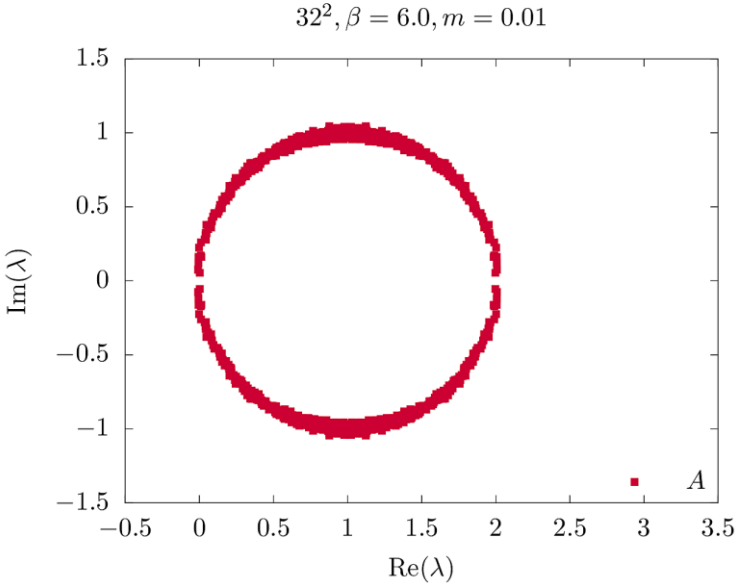
INTERACTING CASE

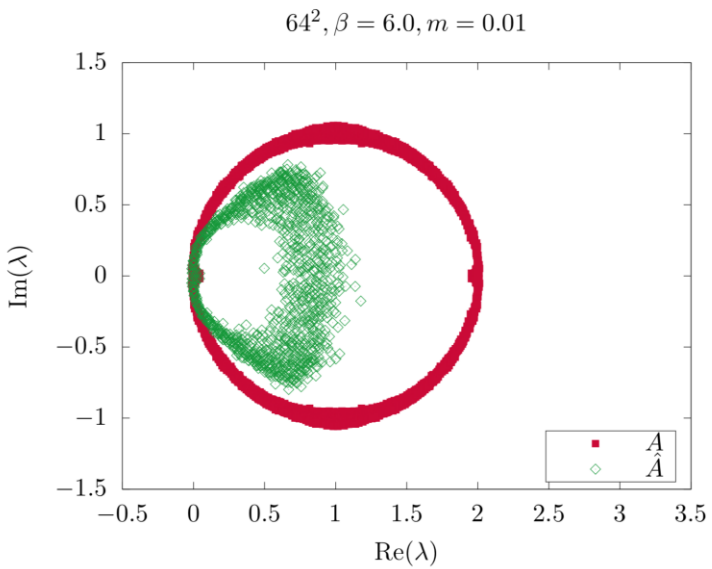
- After gauging the links, $B^2 \neq 1, C^2 \neq 1$. Press on anyway: spectrum becomes a “fuzzed” circle.

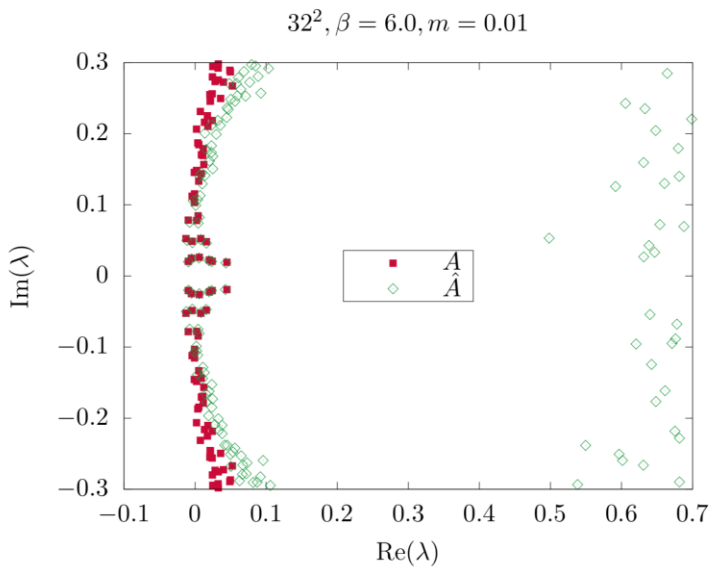


INTERACTING CASE

- After gauging the links, $B^2 \neq 1, C^2 \neq 1$. Press on anyway: spectrum becomes a “fuzzed” circle.
- Generate near-null vectors, coarsen the K-D operator… it worked! (In 2 dimensions…)







RESULTS FROM PHYSICAL POINT HISQ CONFIGURATIONS

QUDA

- "QCD on CUDA" – http://lattice.github.com/quda (open source, BSD license)
- Effort started at Boston University in 2008, now in wide use as the GPU backend for BQCD, Chroma, CPS, MILC, TIFR, etc.
- Provides:
  - Various solvers for all major fermionic discretizations, with multi-GPU support
  - Additional performance-critical routines needed for gauge-field generation
- Maximize performance
  - Exploit physical symmetries to minimize memory traffic
  - Mixed-precision methods
  - Autotuning for high performance on all CUDA-capable architectures
  - Domain-decomposed (Schwarz) preconditioners for strong scaling
  - Eigenvector and deflated solvers (Lanczos, EigCG, GMRES-DR)
  - Multi-source solvers
  - ***Multigrid solvers for optimal convergence***
- A research tool for how to reach the exascale

FOUR DIMENSIONAL TESTS

- **Physical pion mass** configurations courtesy of Carleton DeTar (MILC collaboration)
  - 2+1+1 HISQ + Symanzik configurations
- Test location: Summit --- 6 V100-16GB per node, all tests in QUDA

Volume	ß	a	m_l	m_s	m_c	# nodes
64^3x96	6.30	0.09	0.0012	0.0363	0.432	16
96^3x192	6.72	0.06	0.0008	0.022	0.260	72
144^3x288	7.00	0.0042	0.000569	0.01555	0.1827	432

- Numerical test: Solve $D^{HISQ}\vec{x} = \vec{b}$, $\vec{b}$ a random source, to tolerance 10^{-10}
  - Schur system: $\left(m^2 - D_{eo}^{HISQ} D_{oe}^{HISQ}\right)\overrightarrow{x_e} = m\overrightarrow{b_e} - D_{eo}^{HISQ}\overrightarrow{b_o}$ to tolerance $m10^{-10}$

FOUR DIMENSIONAL TESTS

- **Physical pion mass** configurations courtesy of Carleton DeTar (MILC collaboration)
  - 2+1+1 HISQ + Symanzik configurations
- Test location: Summit --- 6 V100-16GB per node, all tests in QUDA

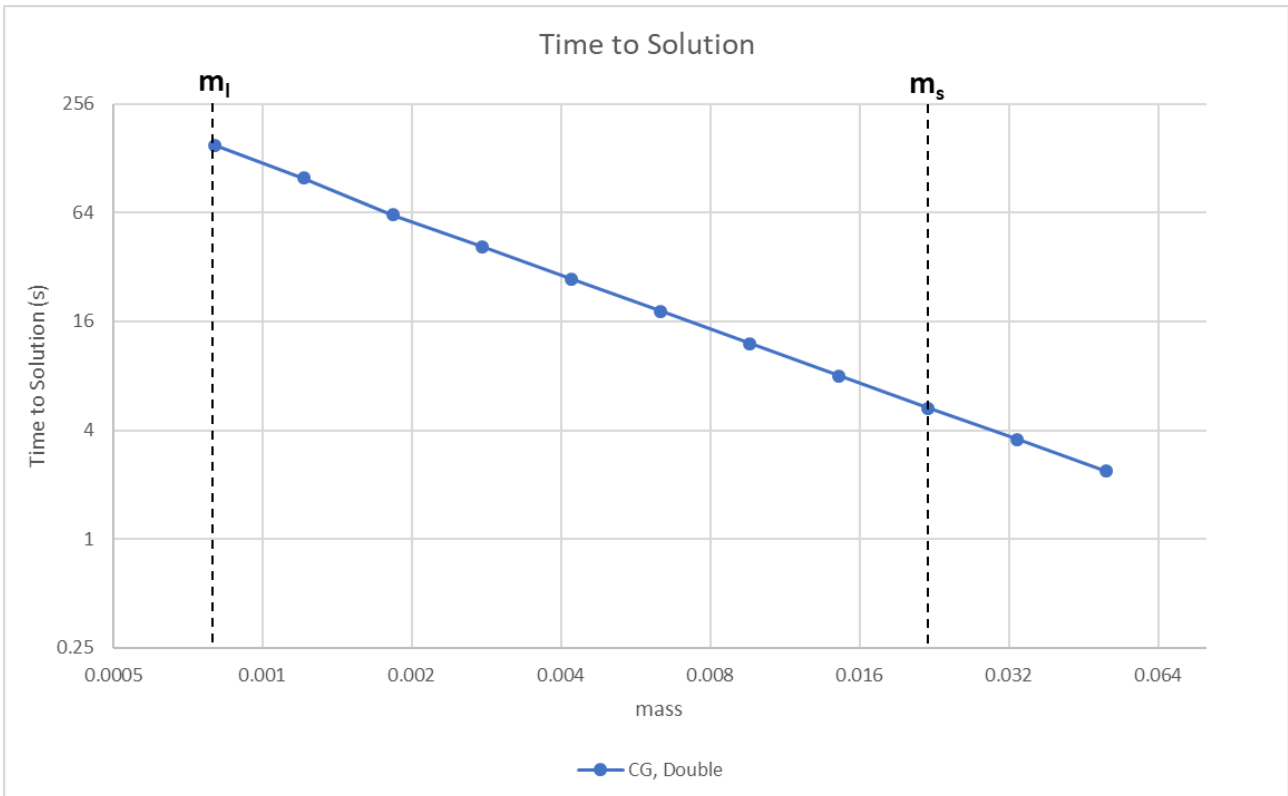
Volume	ß	a	m_l	m_s	m_c	# nodes
64^3x96	6.30	0.09	0.0012	0.0363	0.432	16
96^3x192	6.72	0.06	0.0008	0.022	0.260	72
144^3x288	7.00	0.0042	0.000569	0.01555	0.1827	432

- Numerical test: Solve $D^{HISQ}\vec{x} = \vec{b}$, $\vec{b}$ a random source, to tolerance 10^{-10}
  - Schur system: $\left(m^2 - D_{eo}^{HISQ} D_{oe}^{HISQ}\right)\overrightarrow{x_e} = m\overrightarrow{b_e} - D_{eo}^{HISQ}\overrightarrow{b_o}$ to tolerance $m10^{-10}$

OLD SCHOOL: CG

Schur system: $\left(m^2 - D_{eo}^{HISQ} D_{oe}^{HISQ}\right)\overrightarrow{x_e} = m\overrightarrow{b_e} - D_{eo}^{HISQ}\overrightarrow{b_o}$ to tolerance $m10^{-10}$

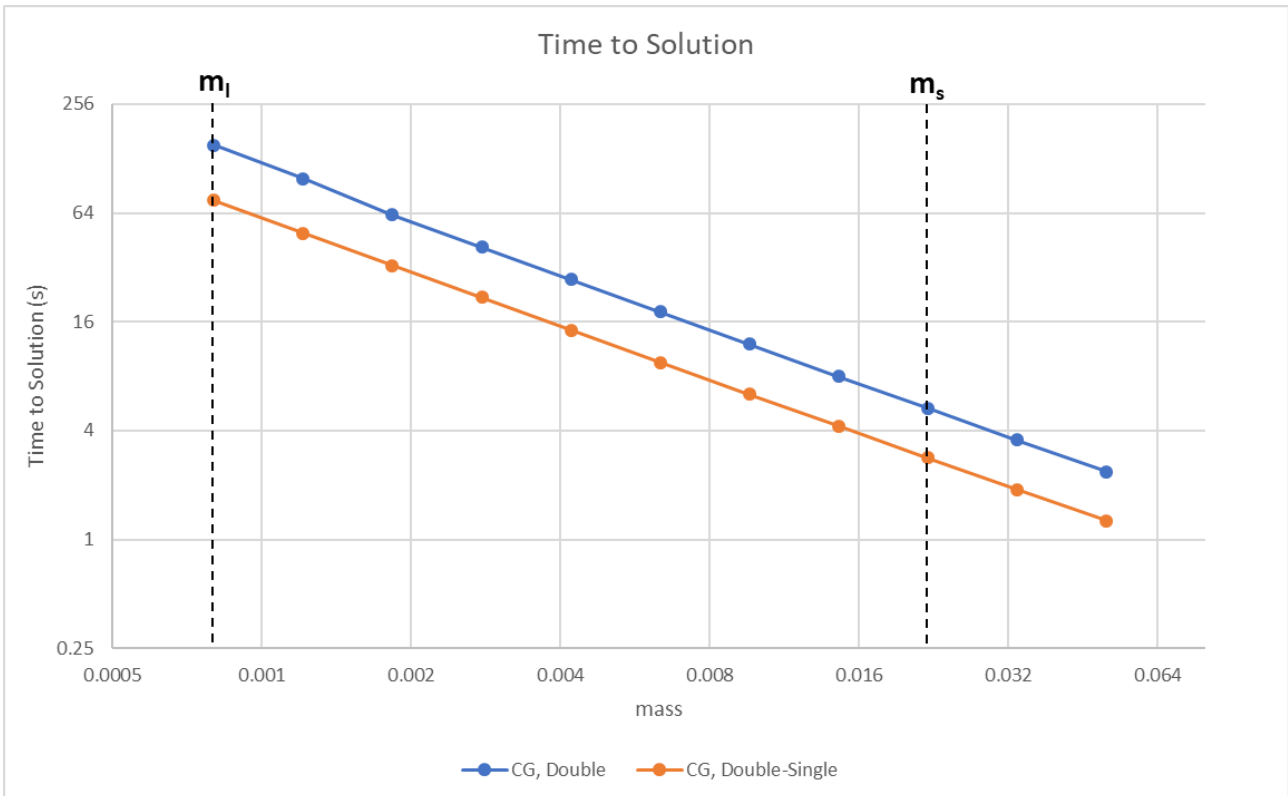
Pure double precision solve, reconstruct-9 (long links can be encoded by 9 numbers)



OLD SCHOOL: CG

Schur system: $\left(m^2 - D_{eo}^{HISQ} D_{oe}^{HISQ}\right)\overrightarrow{x_e} = m\overrightarrow{b_e} - D_{eo}^{HISQ}\overrightarrow{b_o}$ to tolerance $m10^{-10}$

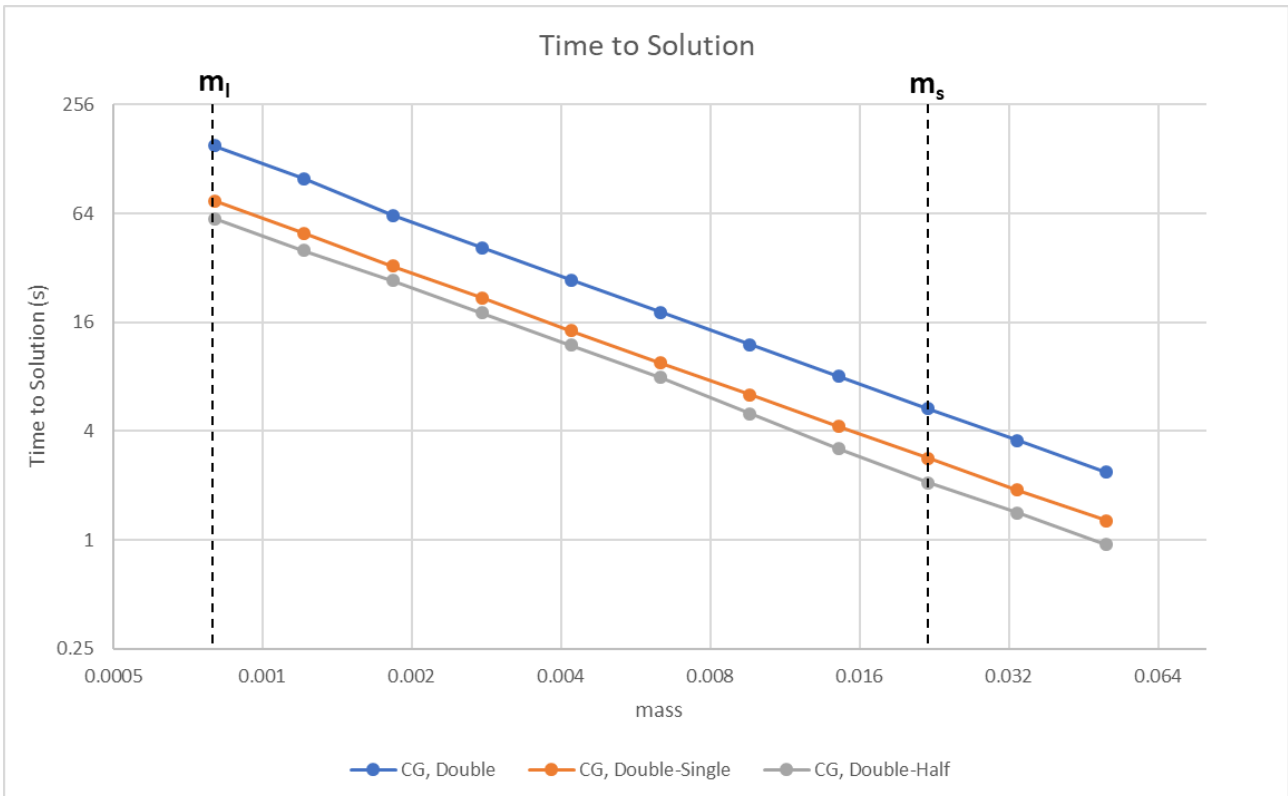
Use mixed precision instead: double-single



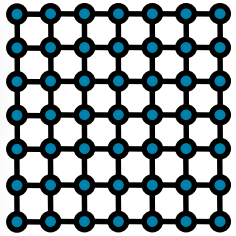
OLD SCHOOL: CG

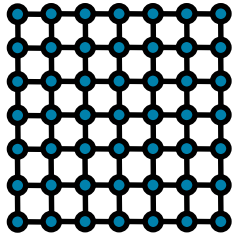
Schur system: $\left(m^2 - D_{eo}^{HISQ} D_{oe}^{HISQ}\right)\overrightarrow{x_e} = m\overrightarrow{b_e} - D_{eo}^{HISQ}\overrightarrow{b_o}$ to tolerance $m10^{-10}$

Use mixed precision instead: double-single, **double-half**

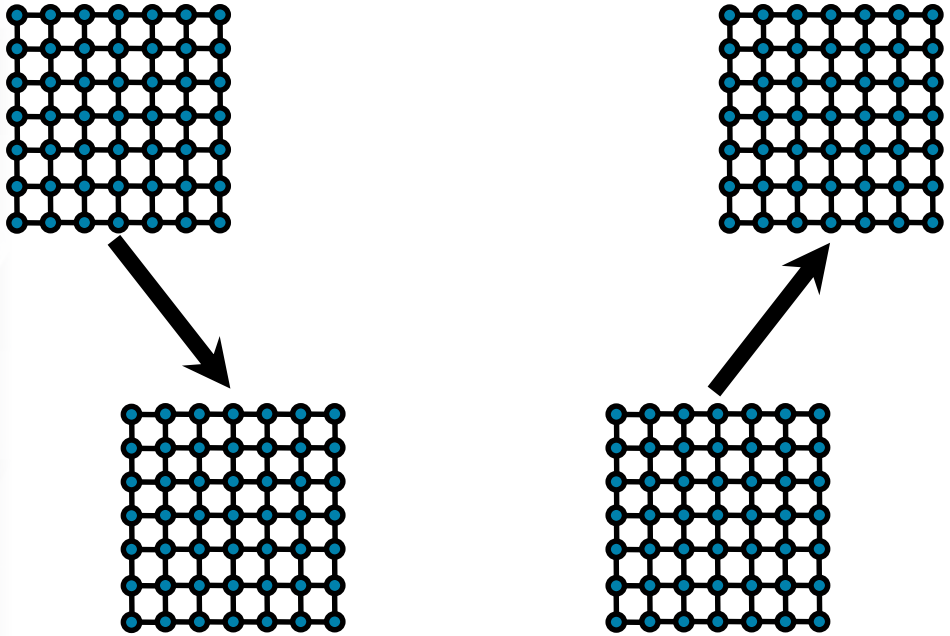


NVIDIA





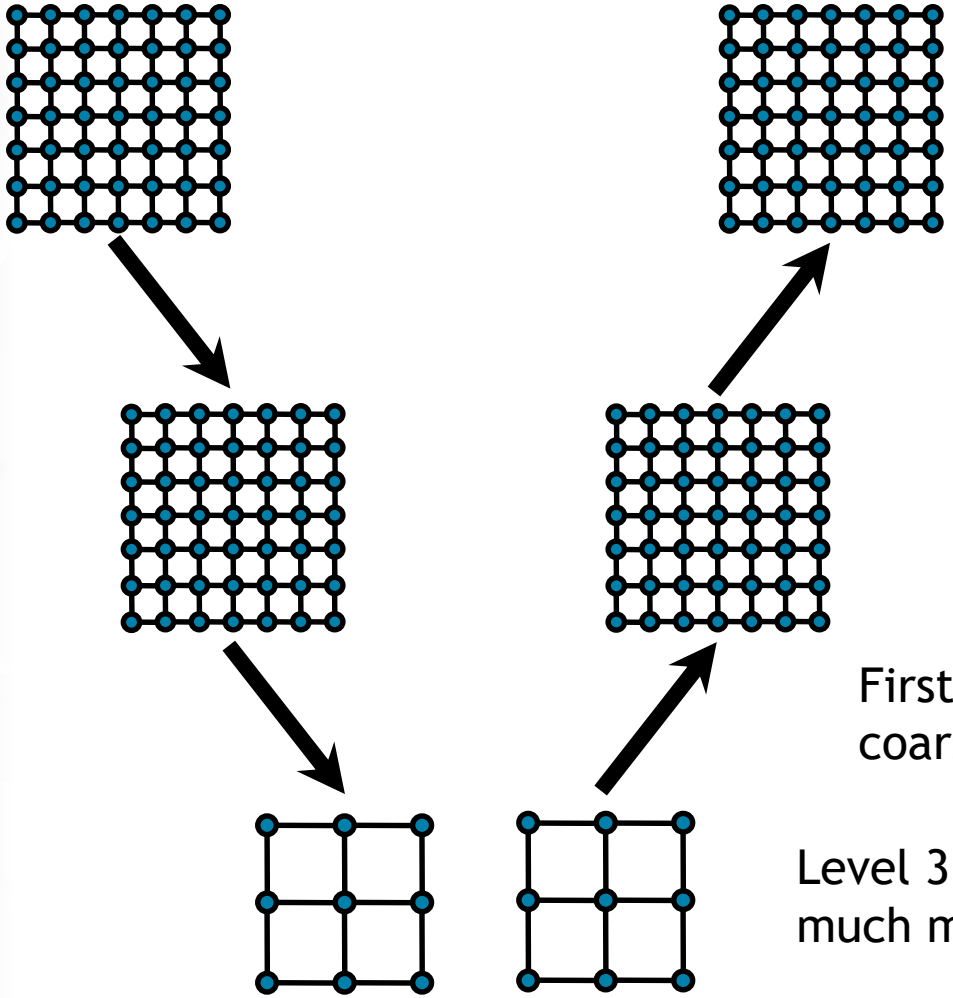
Level 1: “Fine level”, 3 dof per site.
Full HISQ stencil.



Level 1: “Fine level”, 3 dof per site.
Full HISQ stencil.

First “coarsening”: **truncate away Naïk term**,
go to block-preconditioned system

Level 2: “Pseudo-fine level”, 48 dof per site,
global volume of dof is constant



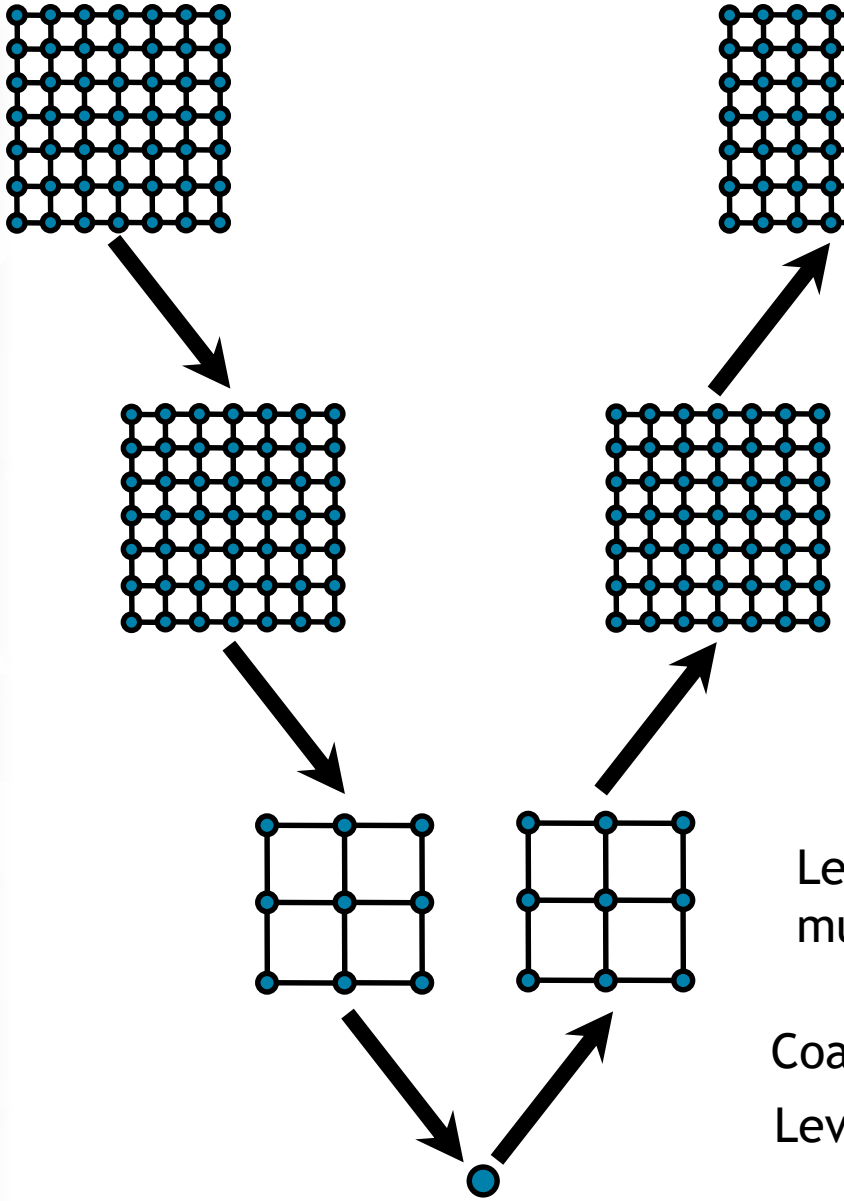
Level 1: “Fine level”, 3 dof per site.
Full HISQ stencil.

First “coarsening”: **truncate away Naïk term**,
go to block-preconditioned system

Level 2: “Pseudo-fine level”, 48 dof per site,
global volume of dof is constant

First “real” coarsening, i.e., reduction in global dof,
coarsen block-preconditioned system

Level 3: “Intermediate level”, **128 dof per site:**
much more than Wilson-clover!



Level 1: “Fine level”, 3 dof per site.
Full HISQ stencil.

First “coarsening”: **truncate away Naïk term**,
go to block-preconditioned system

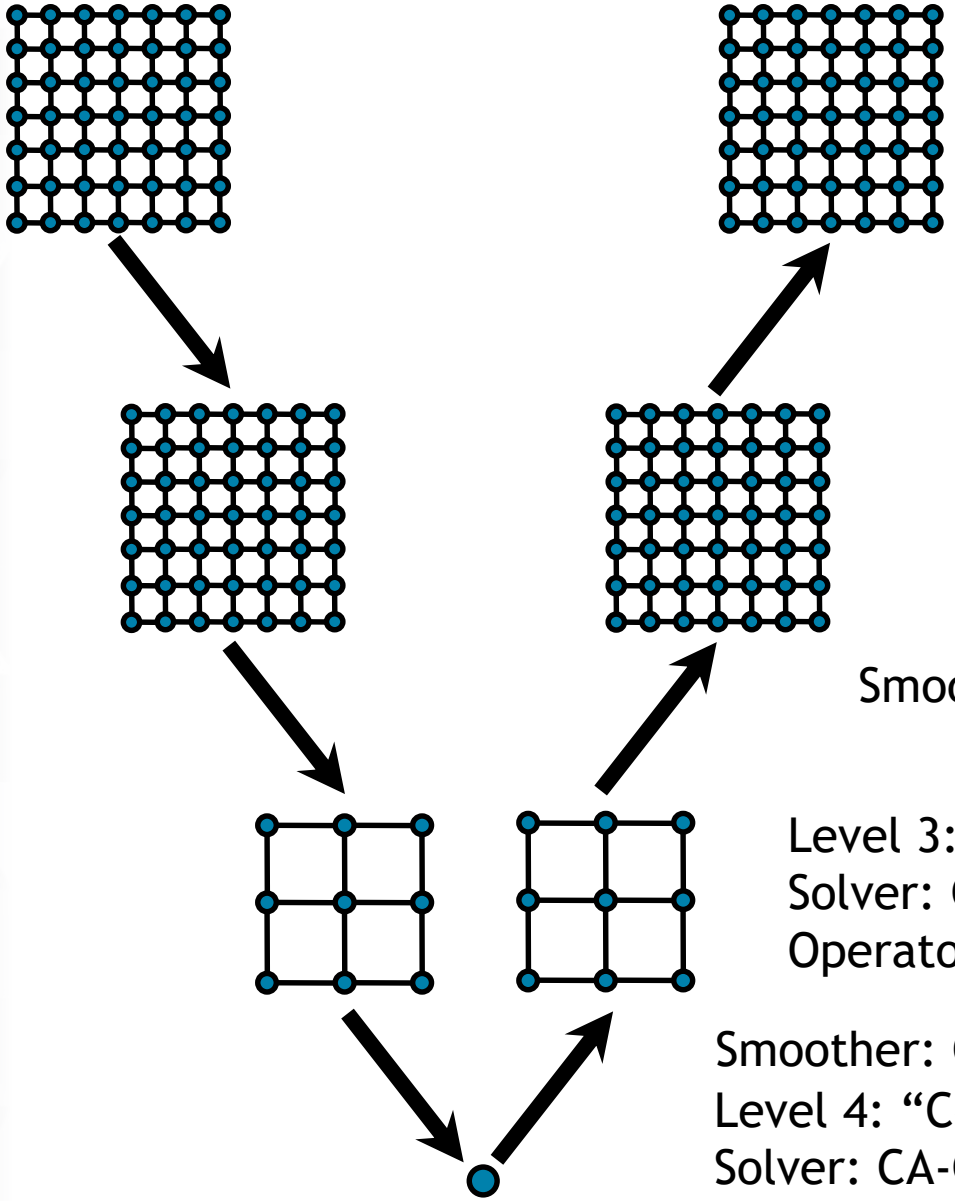
Level 2: “Pseudo-fine level”, 48 dof per site,
global volume of dof is constant

First “real” coarsening, i.e., reduction in global dof,
coarsen block-preconditioned system

Level 3: “Intermediate level”, **128 dof per site:**
much more than Wilson-clover!

Coarsen block-preconditioned system again…

Level 4: “Coarsest level”, **192 dof per site**.



Level 1: “Fine level”, 3 dof per site.
Solver: GCR, tolerance 10^{-10}

Smoother: CA-GCR(0,2)

Level 2: “Pseudo-fine level”, 48 dof per site.
Solver: GCR, tolerance 0.25, max 16 iterations
Operator: Left-block Schur, 16-bit precision

Smoother: CA-GCR(0,2)

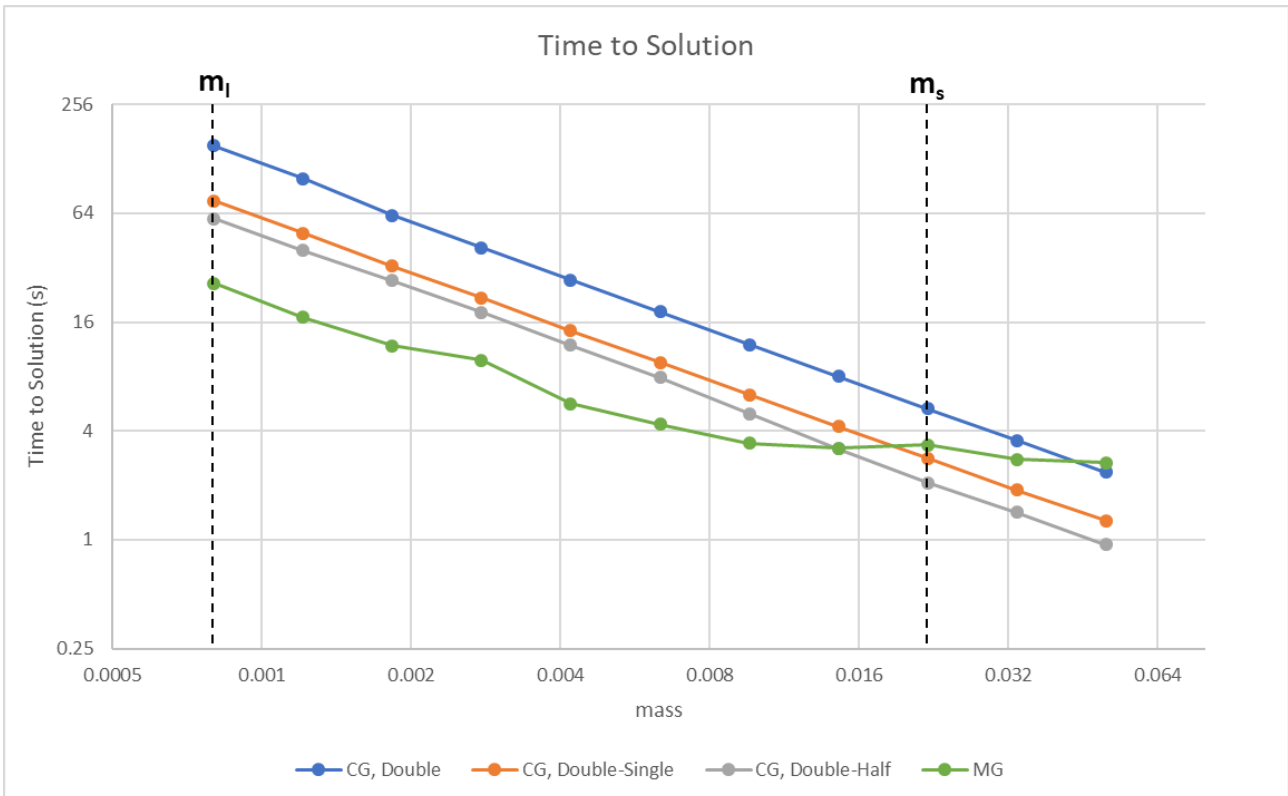
Level 3: “Intermediate level”, **128 dof per site.**
Solver: GCR, tolerance 0.25, max 16 iterations
Operator: Left-block Schur, 16-bit precision

Smoother: CA-GCR(0,2)
Level 4: “Coarsest level”, **192 dof per site.**
Solver: CA-GCR(16)
Operator: Left-block Schur, 16-bit precision

RESULTS WITH MG

Schur system: $\left(m^2 - D_{eo}^{stag} D_{oe}^{stag}\right)\overrightarrow{x_e} = m\overrightarrow{b_e} - D_{eo}\overrightarrow{b_o}$ to tolerance $m10^{-10}$

Note: re-uses near-null vectors generated at m_l for all masses

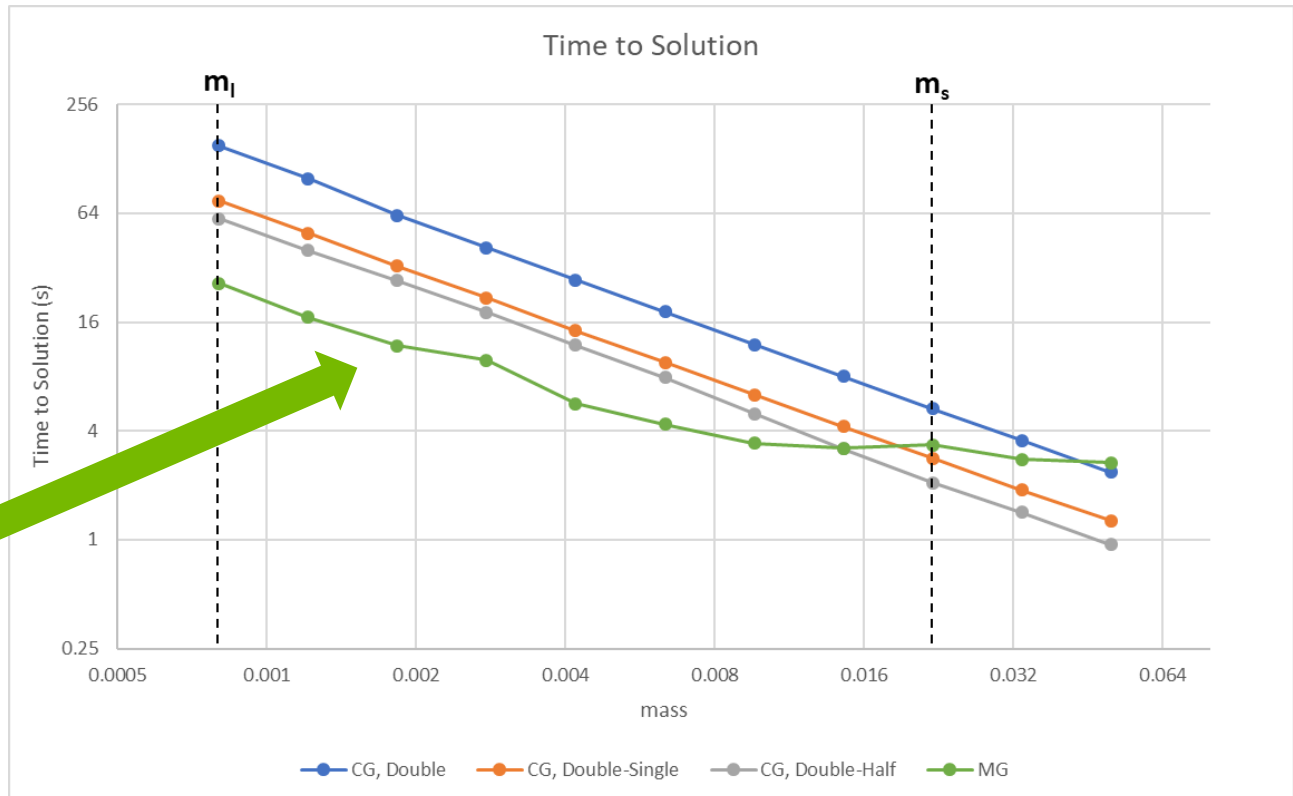


NVIDIA.

RESULTS WITH MG

Schur system: $\left(m^2 - D_{eo}^{stag} D_{oe}^{stag}\right)\overrightarrow{x_e} = m\overrightarrow{b_e} - D_{eo}\overrightarrow{b_o}$ to tolerance $m10^{-10}$

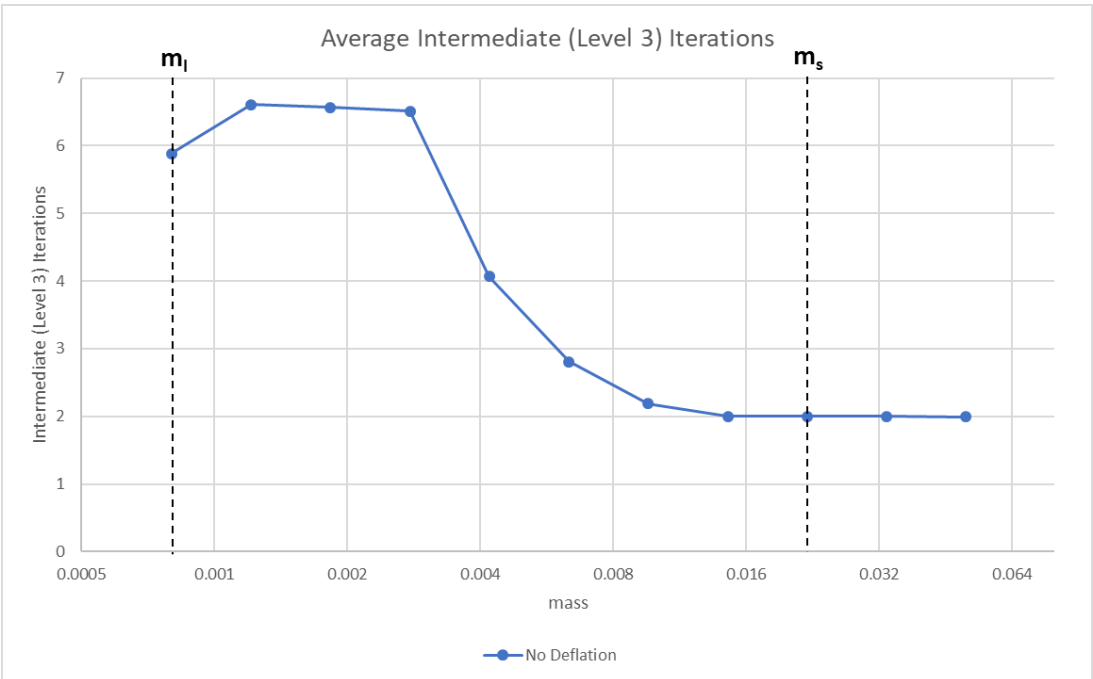
Note: re-uses near-null vectors generated at m_l for all masses



Still critical slowing down!

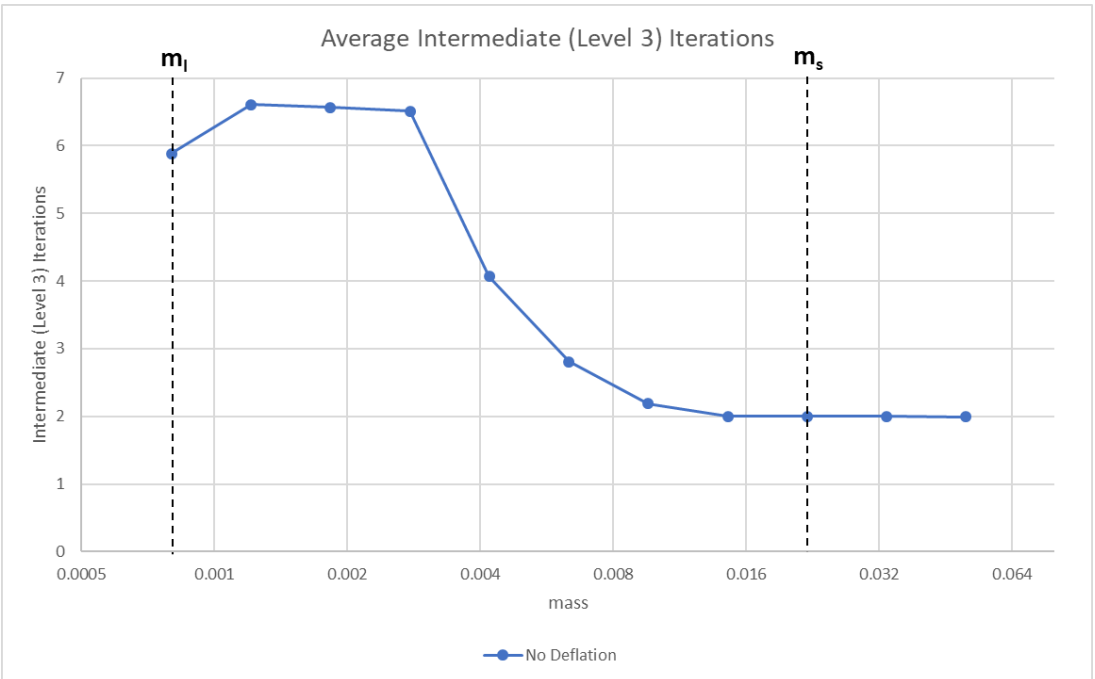
NVIDIA.

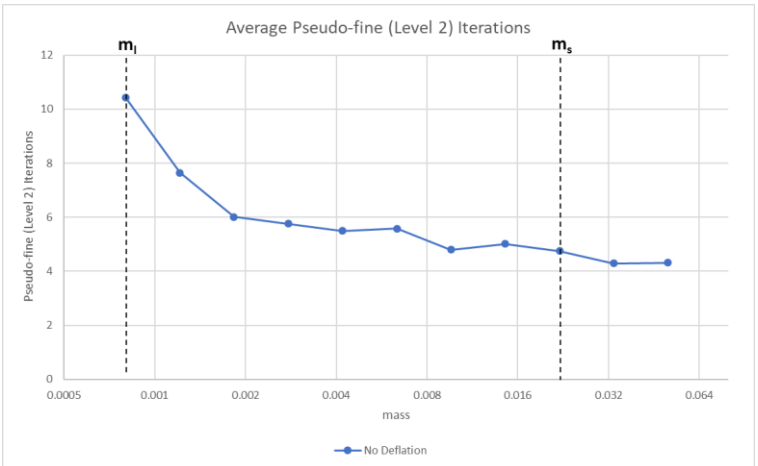
WHAT'S HAPPENING?

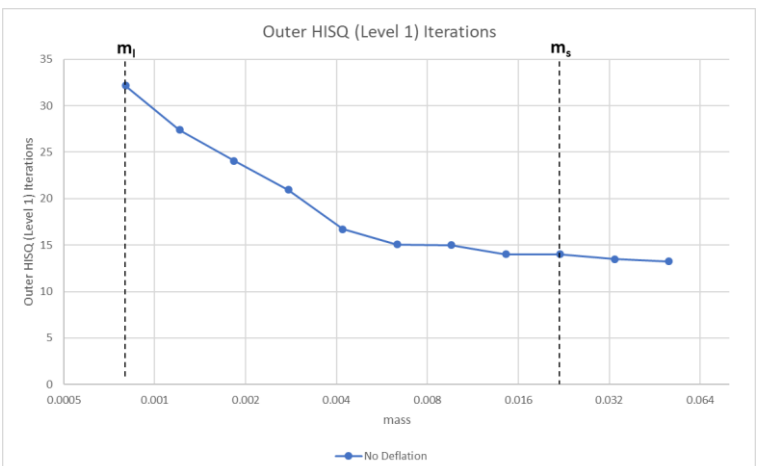


- We aren't doing a good job solving the coarsest level

WHAT'S HAPPENING?

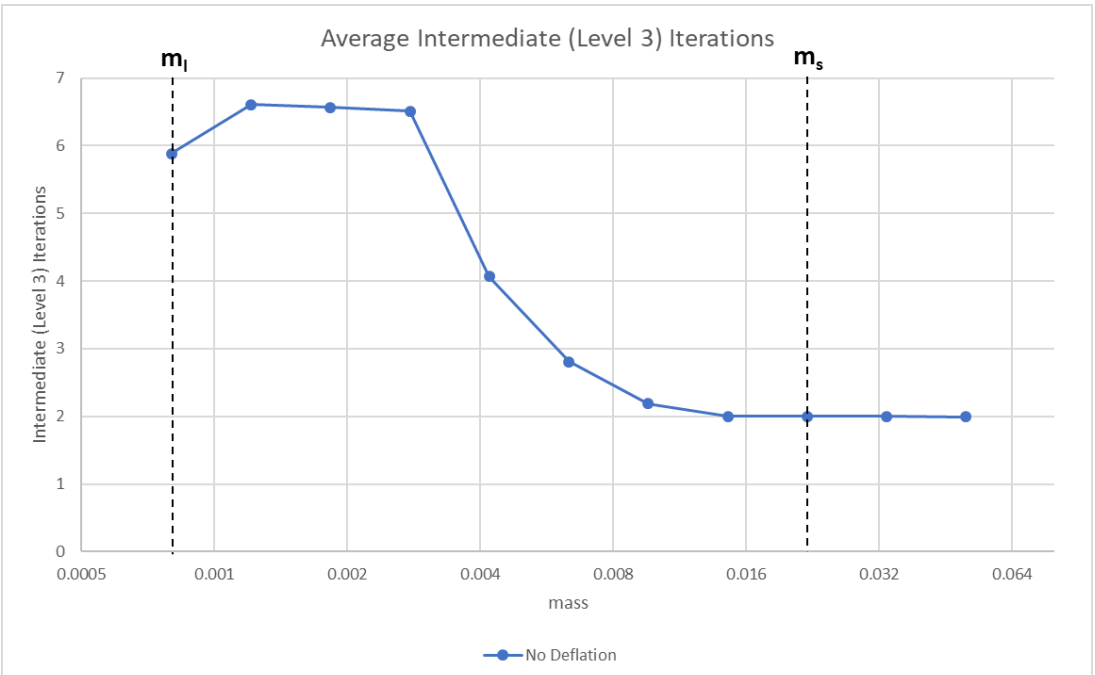


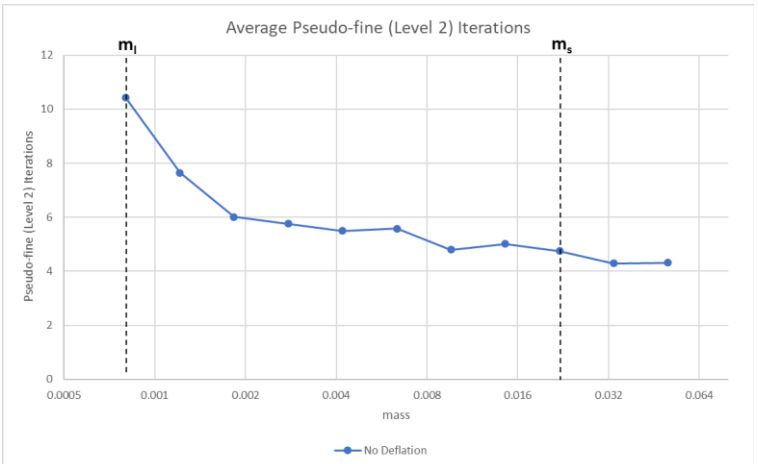




- We aren't doing a good job solving the coarsest level

WHAT'S HAPPENING?





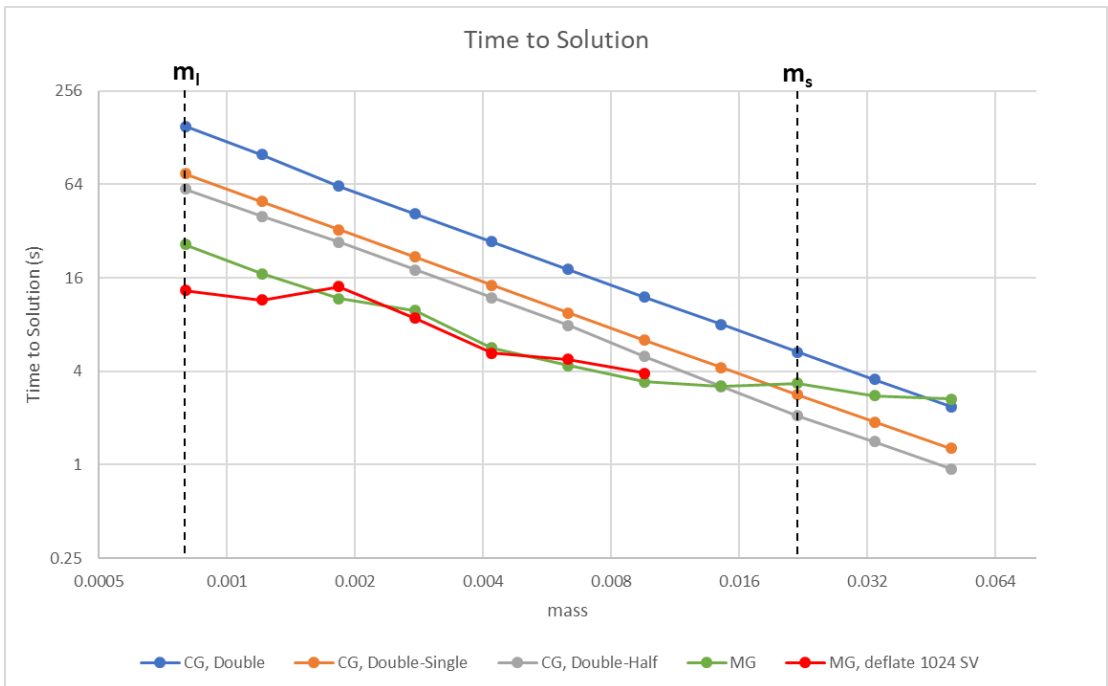
- We aren't doing a good job solving the coarsest level
- One solution: grind further on the coarsest level
  - **Far more expensive.**
- Better solution: deflate the coarsest level!

DEFLATION

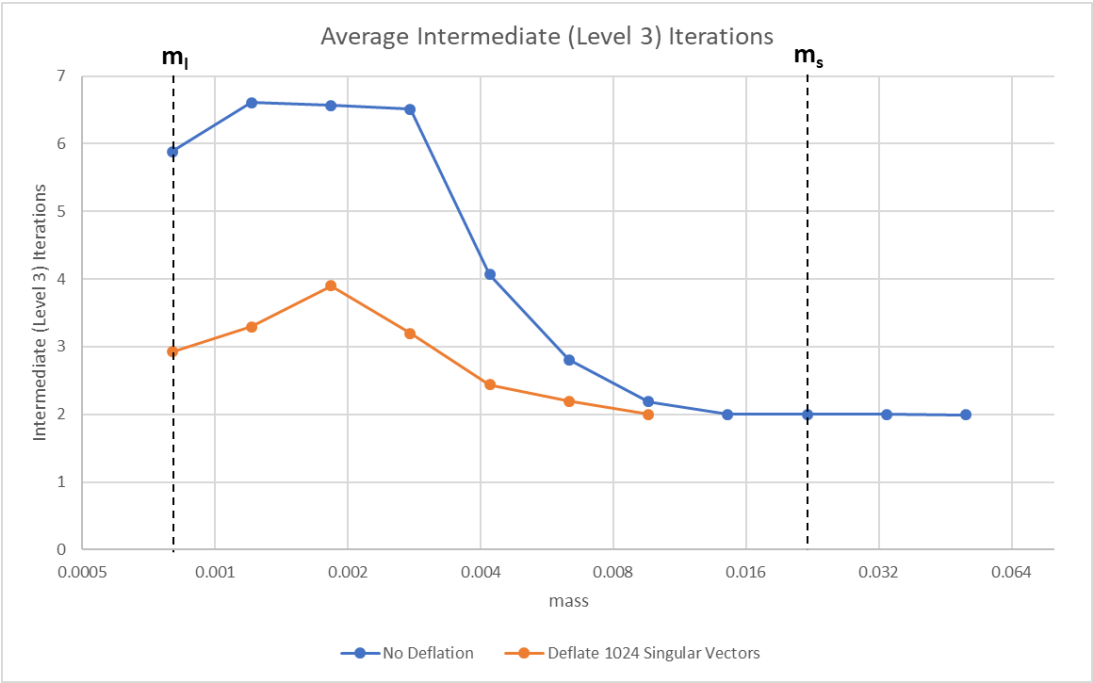
- **None** of this would be possible without hours and hours of work from Dean Howarth (BU)

DEFLATION

- **None** of this would be possible without hours and hours of work from Dean Howarth (BU)
- Perform an SVD deflation of 1024 vectors on the coarsest level

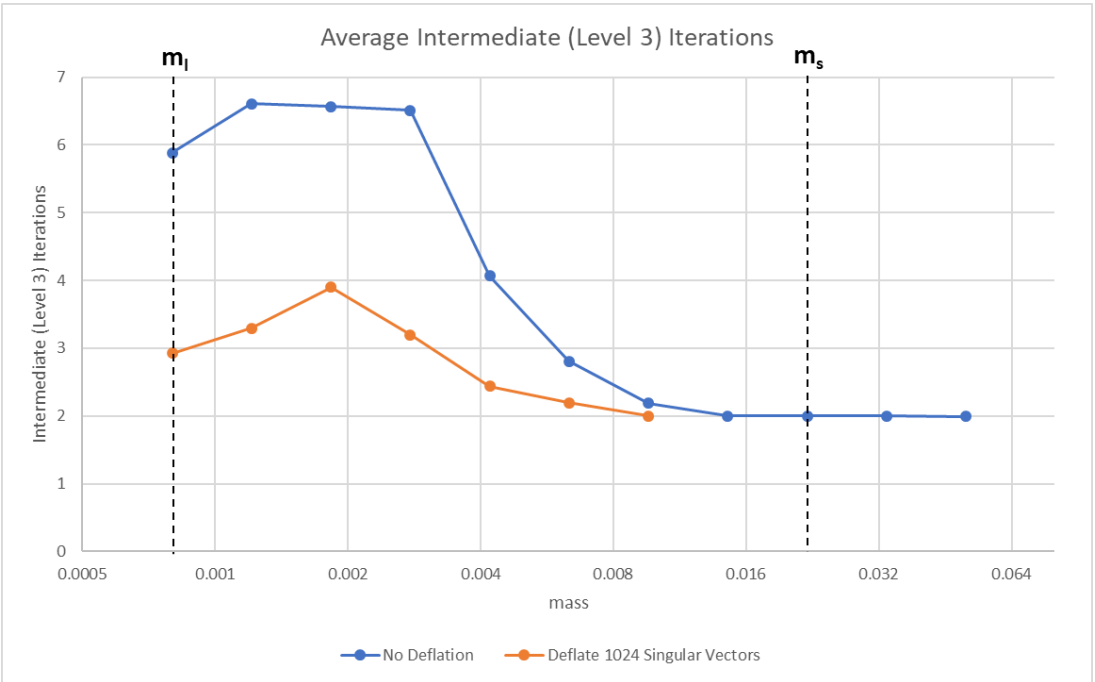


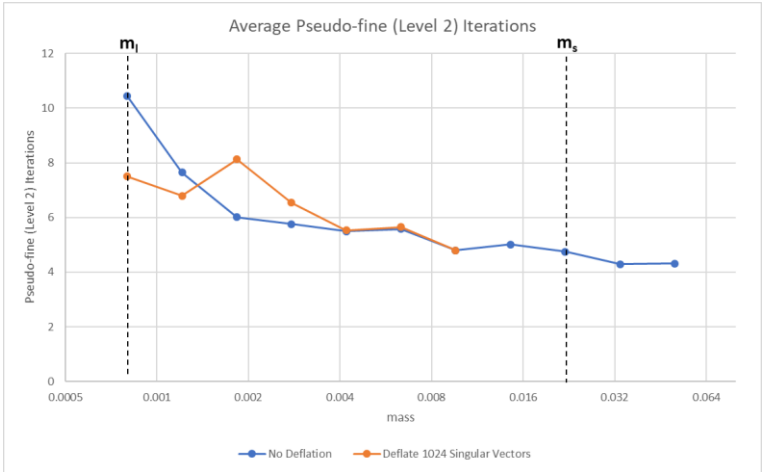
GETTING INTO THE WEEDS



- Deflation improves the intermediate solve

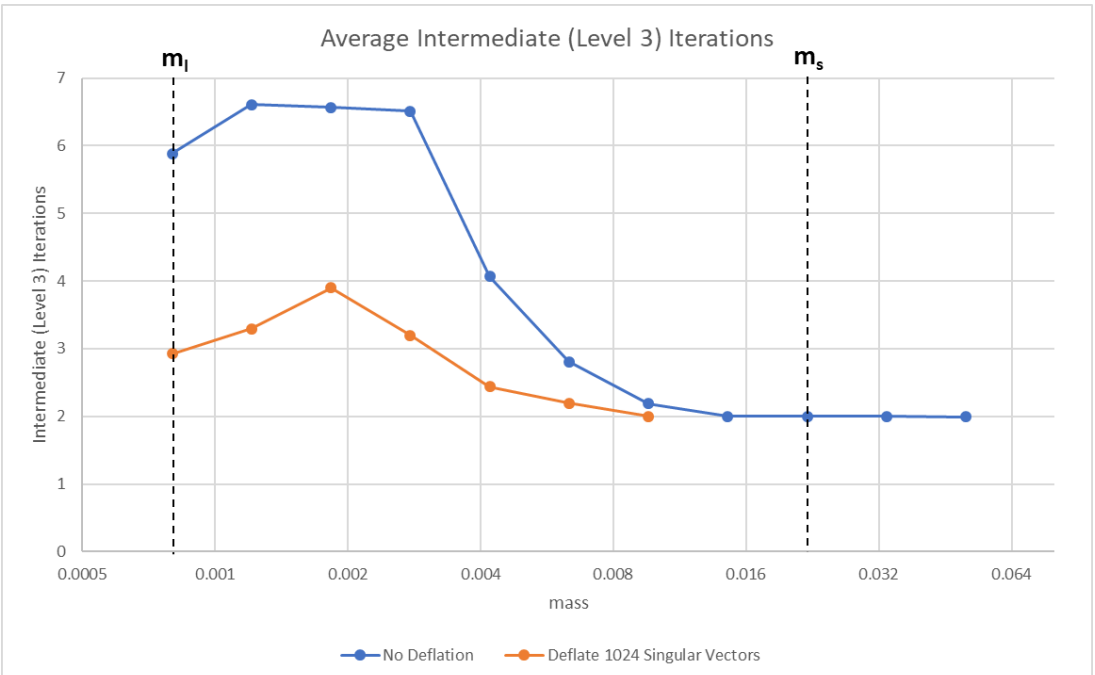
GETTING INTO THE WEEDS

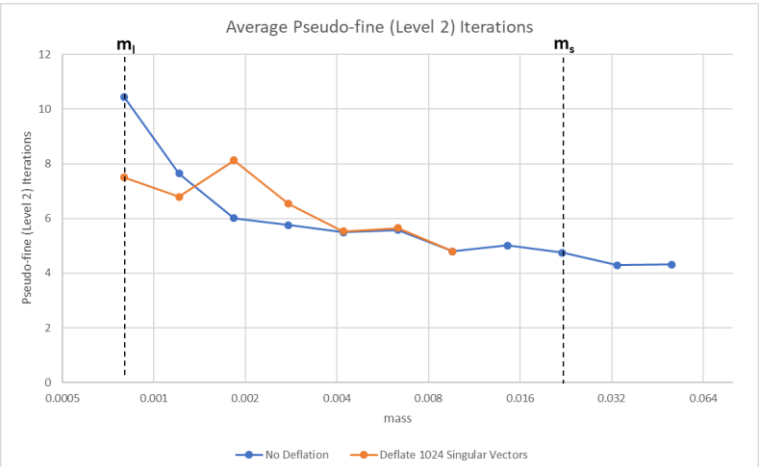


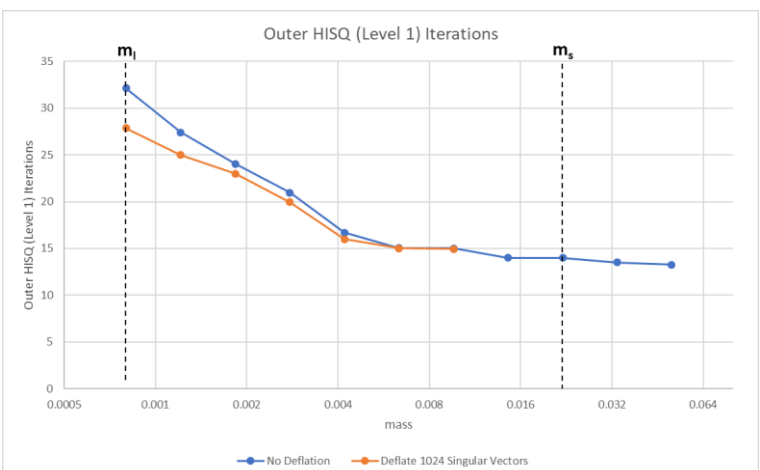


- Deflation improves the intermediate solve
- Coarsest level deflation has compounding benefits

GETTING INTO THE WEEDS

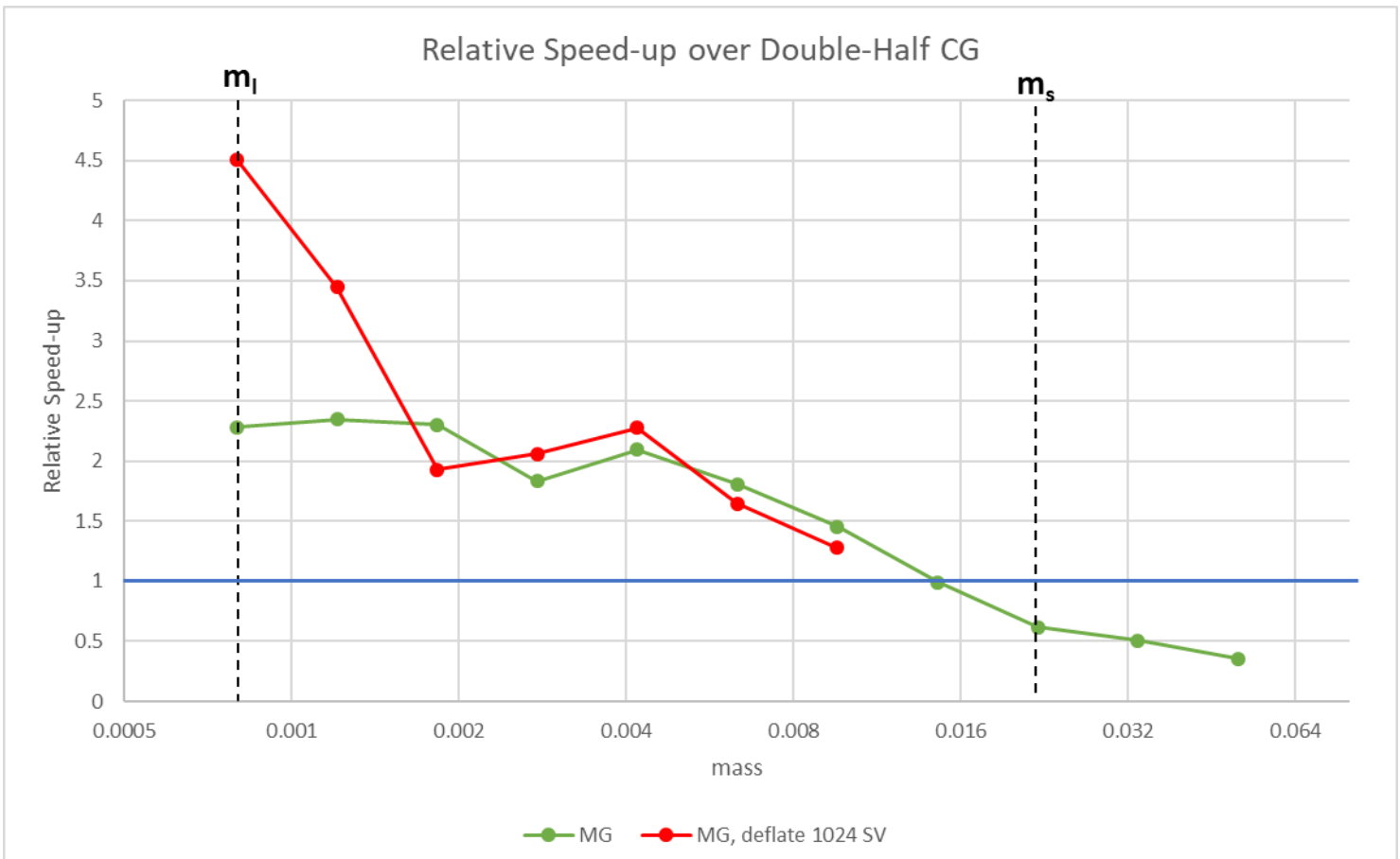




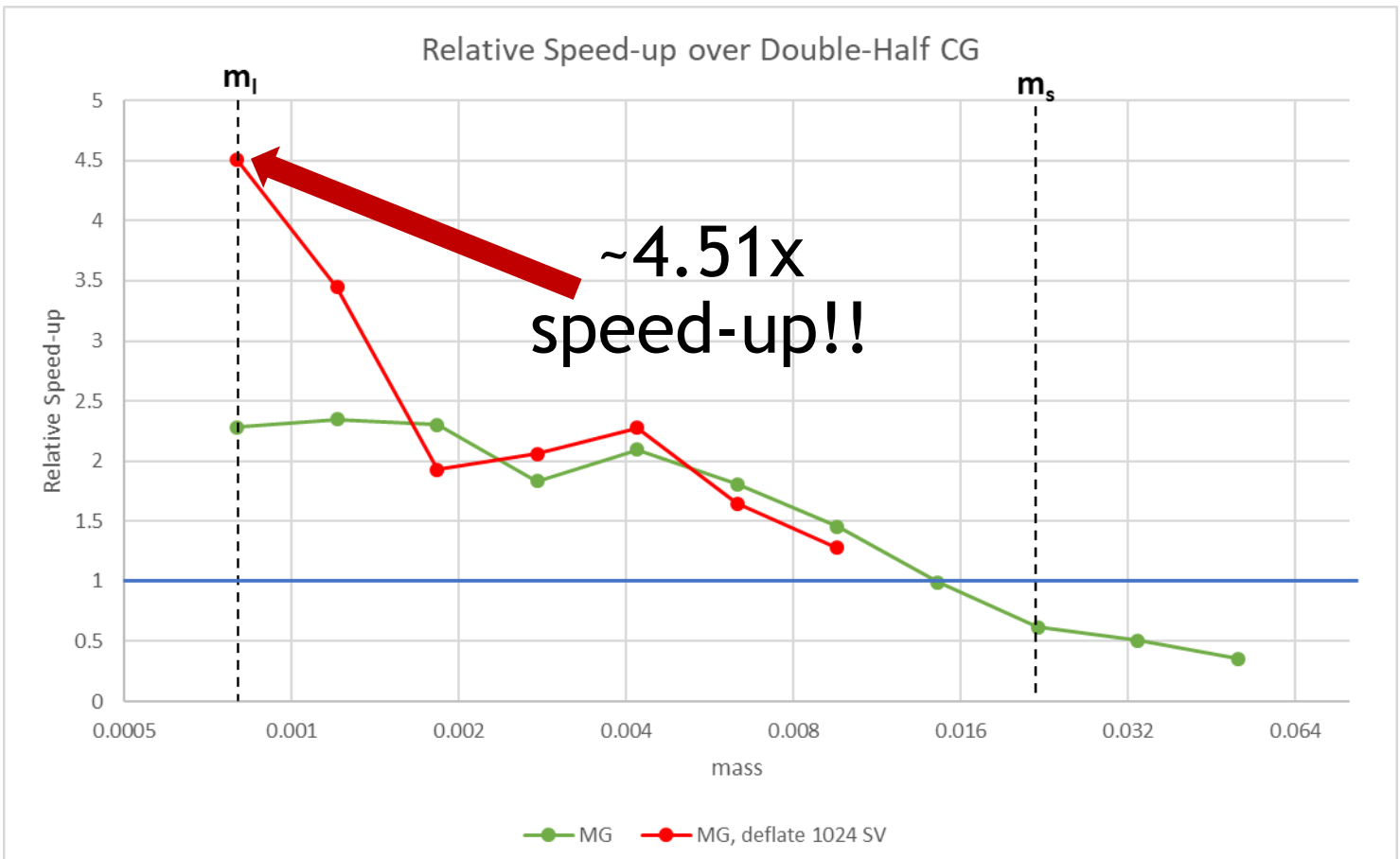


- Deflation improves the intermediate solve
- Coarsest level deflation has compounding benefits
- Remaining issue: the pseudo-fine level isn't doing a good job preconditioning the fine level! But nonetheless…

PERFORMANCE GAINS



PERFORMANCE GAINS



DATA COLLECTION ONGOING…

- MG on the 64^3x96 configuration currently isn't working---too coarse ***with the current MG setup parameters***
  - There's a large phase space to explore!
- MG on the 144^3x288 configuration at the physical light quark mass ***is*** working!! (With a modified prescription: CA-GCR(12) w/restarts until 0.25 tolerance.)
  - Double-half CG: 90.7 seconds
  - MG without deflation: 38.72 seconds – **2.34x speed-up!**
  - MG with deflation: …not sure yet, blame Summit :(

SUMMARY

- Using HISQ MG, we are seeing a speed-up in time to solution on **physical point HISQ configurations:**
  - **4.51x with deflation on 96^3x192!!**
  - **2.34x w/out deflation on 144^3x288!!**
- Coarser configurations are currently causing some woes… but the continuum limit is what really matters!
- Deflation on the coarsest level of an MG solve has compounding benefits for the pseudo-fine and intermediate levels

SUMMARY

- Using HISQ MG, we are seeing a speed-up in time to solution on **physical point HISQ configurations:**
  - **4.51x with deflation on 96^3x192!!**
  - **2.34x w/out deflation on 144^3x288!!**
- Coarser configurations are currently causing some woes… but the continuum limit is what really matters!
- Deflation on the coarsest level of an MG solve has compounding benefits for the pseudo-fine and intermediate levels

ROADMAP

- Further optimizing deflation: kernel fusion
- Algorithmic improvements: the pseudo-fine operator is breaking down as a preconditioner for the HISQ stencil
- Plugging HISQ MG into MILC
- MG preconditioning the Schur operator…
  - …as a necessary prerequisite for HMC
- Further optimizing the coarse operator, reducing memory use, etc…


NVIDIA

BACKUP

KAHLER-DIRAC EQUATION

$b = 2a$ --- K-D lattice spacing = 2x staggered lattice spacing

$$D^{KD} = \sum_{\mu} \left[\nabla_{\mu} \left(\gamma_{\mu} \otimes 1 \right) - \frac{b}{2} \Delta_{\mu} \left(\gamma_5 \otimes \tau_{\mu} \tau_5 \right) \right] + m(1 \otimes 1)$$

KAHLER-DIRAC EQUATION

$b = 2a$ --- K-D lattice spacing = 2x staggered lattice spacing

$$D^{KD} = \sum_{\mu} \left[\nabla_{\mu} \left(\gamma_{\mu} \otimes 1 \right) - \frac{b}{2} \Delta_{\mu} \left(\gamma_5 \otimes \tau_{\mu} \tau_5 \right) \right] + m(1 \otimes 1)$$

$$= \underbrace{\sum_{\mu} \left[\nabla_{\mu} \left(\gamma_{\mu} \otimes 1 \right) - \frac{b}{2} \Delta_{\mu} \left(\gamma_5 \otimes \tau_{\mu} \tau_5 \right) \right] - \frac{d}{b} \left(\gamma_5 \otimes \tau_{\mu} \tau_5 \right)}_{C} + \underbrace{\frac{d}{b} \left(\gamma_5 \otimes \tau_{\mu} \tau_5 \right) + m(1 \otimes 1)}_{B + m}$$

KAHLER-DIRAC EQUATION

$b = 2a$ --- K-D lattice spacing = 2x staggered lattice spacing

$$D^{KD} = \sum_{\mu} \left[\nabla_{\mu} \left(\gamma_{\mu} \otimes 1 \right) - \frac{b}{2} \Delta_{\mu} \left(\gamma_5 \otimes \tau_{\mu} \tau_5 \right) \right] + m(1 \otimes 1)$$

$$= \underbrace{\sum_{\mu} \left[\nabla_{\mu} \left(\gamma_{\mu} \otimes 1 \right) - \frac{b}{2} \Delta_{\mu} \left(\gamma_5 \otimes \tau_{\mu} \tau_5 \right) \right] - \frac{d}{b} \left(\gamma_5 \otimes \tau_{\mu} \tau_5 \right)}_{C} + \underbrace{\frac{d}{b} \left(\gamma_5 \otimes \tau_{\mu} \tau_5 \right) + m(1 \otimes 1)}_{B + m}$$

$$C^2 \sim C^{\dagger} C \sim I \qquad B^2 \sim B^{\dagger} B \sim I,$$

$$(B + m)^{\dagger} (B + m) \sim (B + m)^{-1} (B + m) \sim I$$

“KAHLER-DIRAC OPERATOR”

$$D^{KD} = \underbrace{\sum_{\mu}\left[\nabla_{\mu}\left(\gamma_{\mu}\otimes 1\right) - \frac{b}{2}\Delta_{\mu}\left(\gamma_5\otimes\tau_{\mu}\tau_5\right)\right] - \frac{d}{b}\left(\gamma_5\otimes\tau_{\mu}\tau_5\right)}_{C} + \underbrace{\frac{d}{b}\left(\gamma_5\otimes\tau_{\mu}\tau_5\right) + m(1\otimes 1)}_{B+m}$$

$$A = (B+m)^{-1}(C+B+m)$$

$$= \underbrace{(B+m)^{-1}C}_{\sqrt{\frac{d}{d+m^2}}U} + I$$

Unitary!

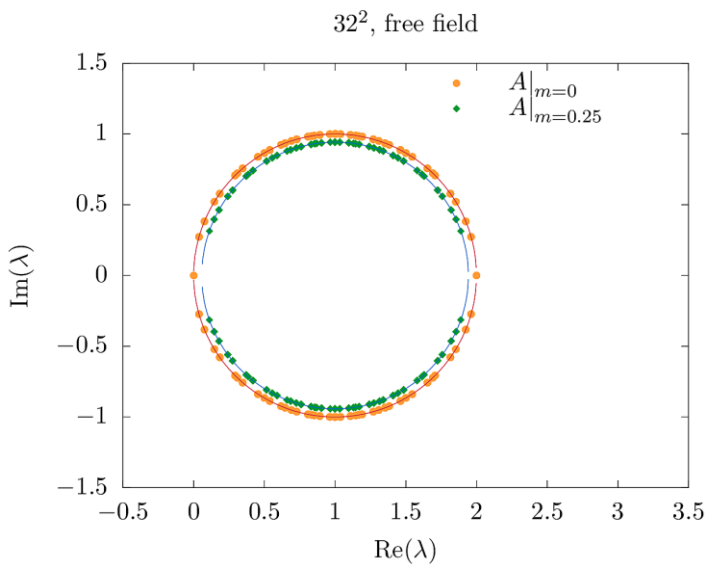
“KAHLER-DIRAC OPERATOR”

$$D^{KD} = \underbrace{\sum_{\mu}\left[\nabla_{\mu}\left(\gamma_{\mu}\otimes 1\right) - \frac{b}{2}\Delta_{\mu}\left(\gamma_5\otimes\tau_{\mu}\tau_5\right)\right] - \frac{d}{b}\left(\gamma_5\otimes\tau_{\mu}\tau_5\right)}_{C} + \underbrace{\frac{d}{b}\left(\gamma_5\otimes\tau_{\mu}\tau_5\right) + m(1\otimes 1)}_{B+m}$$

$$A = (B+m)^{-1}(C+B+m)$$

$$= \underbrace{(B+m)^{-1}C}_{\sqrt{\frac{d}{d+m^2}}U} + I$$

Unitary!



NVIDIA.