2+1 Flavor Domain Wall Fermion QCD Lattices: Ensemble Production and (some) Properties

Lattice 2019
June 19, 2019
Wuhan, China

Robert Mawhinney
Columbia University
RIKEN-BNL-Columbia Collaboration

Major contributions to this work by Chulwoo Jung, David Murphy and Jiqun Tu.
Substantial support for QUDA implementation from Kate Clark (NVIDIA)

The RBC & UKQCD collaborations

BNL and BNL/RBRC

Yasumichi Aoki (KEK)
Taku Izubuchi
Yong-Chull Jang
Chulwoo Jung
Meifeng Lin
Aaron Meyer
Hiroshi Ohki
Shigemi Ohta (KEK)
Amarjit Soni

UC Boulder

Oliver Witzel

CERN

Mattia Bruno

Columbia University

Ryan Abbot
Norman Christ
Duo Guo
Christopher Kelly
Bob Mawhinney
Masaaki Tomii
Jiqun Tu
Bigeng Wang
Tianle Wang
Yidi Zhao

University of Connecticut

Tom Blum
Dan Hoying (BNL)
Luchang Jin (RBRC)
Cheng Tu

Edinburgh University

Peter Boyle
Luigi Del Debbio
Felix Erben
Vera Gülpers
Tadeusz Janowski
Julia Kettle
Michael Marshall
Fionn Ó hÓgáin
Antonin Portelli
Tobias Tsang
Andrew Yong
Azusa Yamaguchi

KEK

Julien Frison

University of Liverpool

Nicolas Garron

MIT

David Murphy

Peking University

Xu Feng

University of Regensburg

Christoph Lehner (BNL)

University of Southampton

Nils Asmussen
Jonathan Flynn
Ryan Hill
Andreas Jüttner
James Richings
Chris Sachrajda

Stony Brook University

Jun-Sik Yoo
Sergey Syritsyn (RBRC)

RBC/UKQCD Ensembles with 2+1 flavor (M)DWF + Iwasaki Gauge Action

Action (F+G)	$1/a$ (GeV)	Lattice volume	m_l	m_s	$m_{\rm res}$	m_π (MeV)	m_K (MeV)	Size (fm)
			(in lattice units)					
DWF+I	1.785(5)	$24^3\times64\times16$	0.03	0.04	0.00308	693		2.6
DWF+I	1.785(5)	$24^3\times64\times16$	0.02	0.04	0.00308	576		2.6
DWF+I	1.785(5)	$24^3\times64\times16$	0.01	0.04	0.00308	432	626	2.6
DWF+I	1.785(5)	$24^3\times64\times16$	0.005	0.04	0.00308	340	593	2.6
MDWF+I	1.730(4)	$48^3\times96\times24$	0.00078	0.0362	0.000614	139	499	5.5
DWF+I	2.383(9)	$32^3\times64\times16$	0.008	0.03	0.000664	412	615	2.6
DWF+I	2.383(9)	$32^3\times64\times16$	0.006	0.03	0.000664	360	596	2.6
DWF+I	2.383(9)	$32^3\times64\times16$	0.004	0.03	0.000664	302	579	2.6
MDWF+I	2.359(7)	$64^3\times128\times12$	0.000678	0.02661	0.000314	139	508	5.4
MDWF+I	2.774(10)	$48^3\times96\times12$	0.002144	0.02144	0.000229	234	516	3.5
MDWF+I	2.774(10)	$96^3\times192\times12$	0.000541	0.0213	0.000229	135	495	6.9
DWF+I	3.15(2)	$32^3\times64\times12$	0.0047	0.0186	0.000631	371	558	2.0

- Shamir Domain wall and Mobius Domain Wall fermions used.
- 30-40% of physical light quark mass comes from residual mass.

Exact One Flavor Algorithm - EOFA

- RHMC: Form rational approximation $x^{1/2} \simeq \alpha_0 + \sum_k \alpha_k/(x+\beta_k)$ and compute

$$\det\left[\frac{D(m_1)}{D(m_2)}\right] = \left\{\det\left[\frac{D^\dagger D(m_1)}{D^\dagger D(m_2)}\right]\right\}^{1/2}$$

  Multishift CG allows $D^\dagger D + \beta_k$ to be inverted for all k simultaneously

- EOFA: Use Schur determinant identity applied to spin structure of D to factorize

$$\det\left[\frac{D(m_1)}{D(m_2)}\right] = \frac{1}{\det\left(H_1\right)} \cdot \frac{1}{\det\left(H_2\right)}$$

  with H_1 and H_2 Hermitian and positive-definite [TWQCD, arXiv:1403.1683]

- No rational approximation, so no need for multishift CG
- Expect [TWQCD, arXiv:1412.0819]:
  - Reduced memory footprint
  - Faster algorithm since we avoid extra linear algebra overhead

EOFA Mathematical Formalism

EOFA Action [TWQCD, arXiv:1403.1683]

$$S_{\text{EOFA}} = \phi^{\dagger} \Big[\underbrace{\mathbb{1} - k P_{-} \Omega_{-}^{\dagger} \left[H(m_1) \right]^{-1} \Omega_{-} P_{-} + k P_{+} \Omega_{+}^{\dagger} \left[H(m_2) - \Delta_{+} P_{+} \right]^{-1} \Omega_{+} P_{+}}_{\equiv \mathcal{M}_{\text{EOFA}}} \Big] \phi$$

- $\boldsymbol{D_{\text{DWF}}}$ (standard DWF Dirac operator) and $\boldsymbol{D_{\text{EOFA}}}$ (TWQCD's EOFA Dirac operator) are related by

$$\boxed{\boldsymbol{D_{\text{DWF}} = D_{\text{EOFA}} \cdot \widetilde{D}}}$$

  where $\boldsymbol{\widetilde{D} = \delta_{xx'} \delta_{\alpha\alpha'} \delta_{aa'} (\widetilde{D})_{ss'}}$ is block diagonal in 4D spacetime, spin, and color indices, and has no gauge field dependence
- $\boldsymbol{H \equiv \gamma_5 R_5 D_{\text{EOFA}}}$ is Hermitian EOFA Dirac operator
- Introducing D_{EOFA} trades explicit γ_5-Hermiticity for any choice of parameters for dense $L_s \times L_s$ block structure in fifth dimension: D_{DWF} has tridiagonal ss' stencil but $(\gamma_5 R_5 D_{\text{DWF}})^{\dagger} \neq \gamma_5 R_5 D_{\text{DWF}}$ in general
- Evaluating Hamiltonian or pseudofermion force requires two (ordinary) CG inversions
- Heatbath requires evaluating $\phi = \mathcal{M}_{\text{EOFA}}^{-1/2} \eta$ (still need rational approximation!)

EOFA Cayley Preconditioning

Generic EOFA Linear System

$$\left(\gamma_5 R_5 D_{\mathrm{EOFA}} + \beta \Delta_{\pm} P_{\pm}\right)\psi = \phi$$

- Möbius D_{EOFA} is dense in ss' and thus expensive compared to (tridiagonal) D_{DWF}
- Can use $\widetilde{D}^{-1}$ as a preconditioner to write EOFA system in terms of D_{DWF} for $\beta = 0$:

$$\boxed{\gamma_5 R_5 D_{\mathrm{EOFA}} \psi = \phi} \iff \underbrace{D_{\mathrm{EOFA}} \cdot \widetilde{D}}_{= D_{\mathrm{DWF}}} \cdot \underbrace{\widetilde{D}^{-1} \psi}_{\equiv \psi'} = \gamma_5 R_5 \phi \iff \boxed{D_{\mathrm{DWF}} \psi' = \phi'}$$

- $\beta \neq 0$ system can be treated as slight generalization of D_{DWF} to a four-point stencil in ss' by noticing that $\Delta_{\pm} \widetilde{D} = \vec{u} \otimes \vec{v}$ is rank-one [Jung et al., arXiv:1706.05843]
- **"Cayley preconditioning":** solve substantially cheaper preconditioned system for ψ' and recover ψ for one additional matrix multiplication by $\widetilde{D}$
- Allows simple EOFA implementation reusing existing high-performance D_{DWF} code

EOFA, QUDA and Summit

- EOFA has been in use in production of G-parity ensembles on BGQ
- For G-parity case, need the 1/2 power of the light quark determinant, so RHMC -> EOFA was expected to help substantially.
- G-parity EOFA plus retuning HMC gave 4.5× speed-up (Murphy)
- For this ensemble RHMC -> EOFA for the strange quark
- Required adding some GPU code to CPS for EOFA, for example for heat bath (Murphy). Generalized QUDA Mobius Dslash code to implement generic EOFA linear system of previous slide (Tu).
- Time for a trajectory went down by 30% when switching from RHMC -> EOFA
  - Time in solver is similar for RHMC and EOFA
  - Calculation of fermion force is reduced from the number of RHMC poles to one, which is the source of much of the speed-up.
- RHMC was in double precision (restarting not possible) and putting in Hasenbush preconditioning masses is costly for RHMC, so was not done.
- EOFA for 1/a = 2.8 GeV ensemble has one intermediate Hasenbusch mass, tuned to make EOFA forces similar to 2 flavor quotient forces (more below).

DWF Conjugate Gradient Performance on Summit Spring 2018

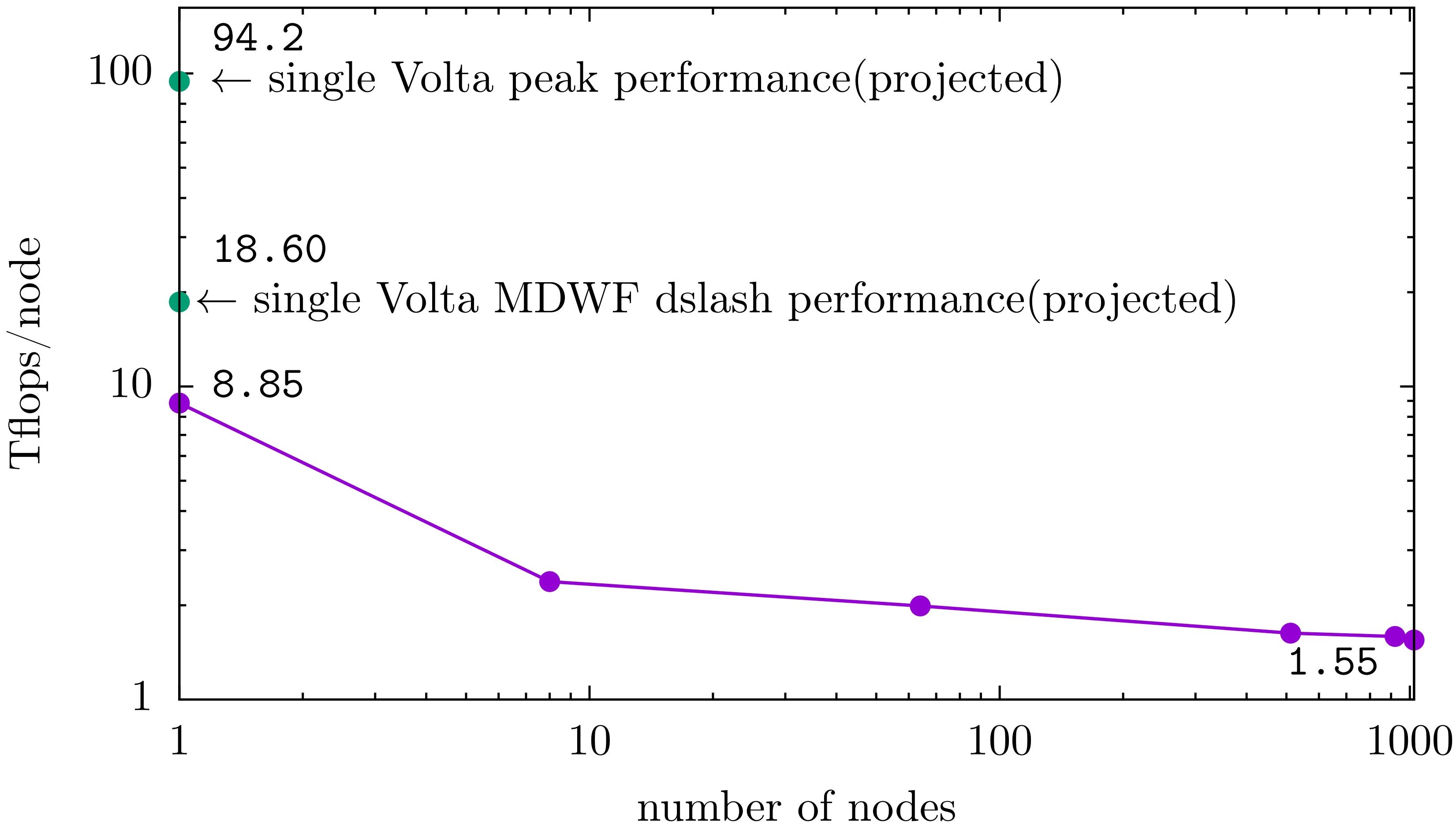
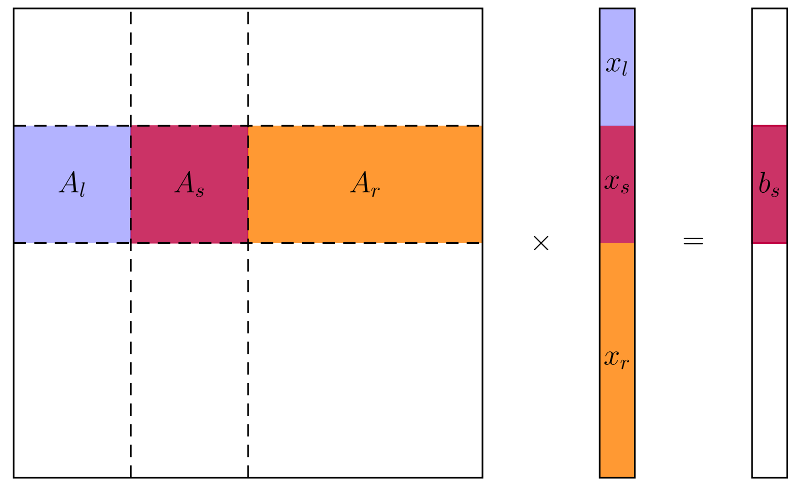


Figure 2: Half precision Möbius domain wall fermion CG weak scaling with local volume of $16 \times 12^3 \times 12$. 6 NVIDIA Volta GPUs on each compute node. Numbers provided by Chulwoo Jung.

Multisplitting Preconditioned CG (MSPCG)

$$Ax = b : A_l x_l + A_s x_s + A_r x_r = b_s$$



Solve

$$A_l x_l + A_s x_s + A_r x_r = b_s,$$

Rearrange into an iterative form

$$\begin{aligned} A_s x_s^{(k+1)} &= b_s - A_l x_l^{(k)} - A_r x_r^{(k)} \\ &= b_s - \left(Ax^{(k)} - A_s x_s^{(k)}\right) \\ &= r^{(k)} + A_s x_s^{(k)} \equiv \hat{b}_s^{(k)} \end{aligned}$$

For each cycle,

use communication to calculate the right-hand-side $\hat{b}_s$.
solve $A_s x_s^{(k+1)} = \hat{b}_s^{(k)}$ locally.
the updated solution $x_s^{(k+1)}$ will be used to ready the next cycle.

Get A_s for each node by chopping off all off-block-diagonal terms: applying zero Dirichlet boundary condition.

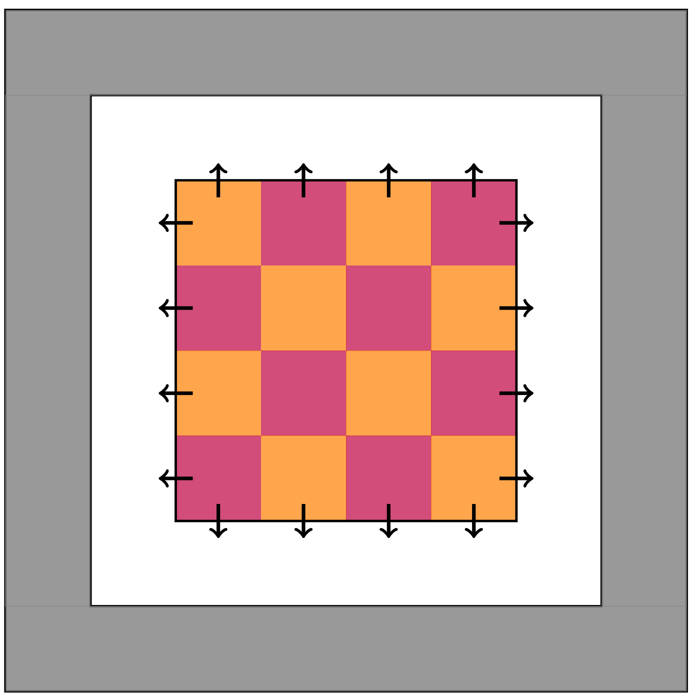
Implementation for Preconditioned Normal Operator

- For DWF, we are inverting

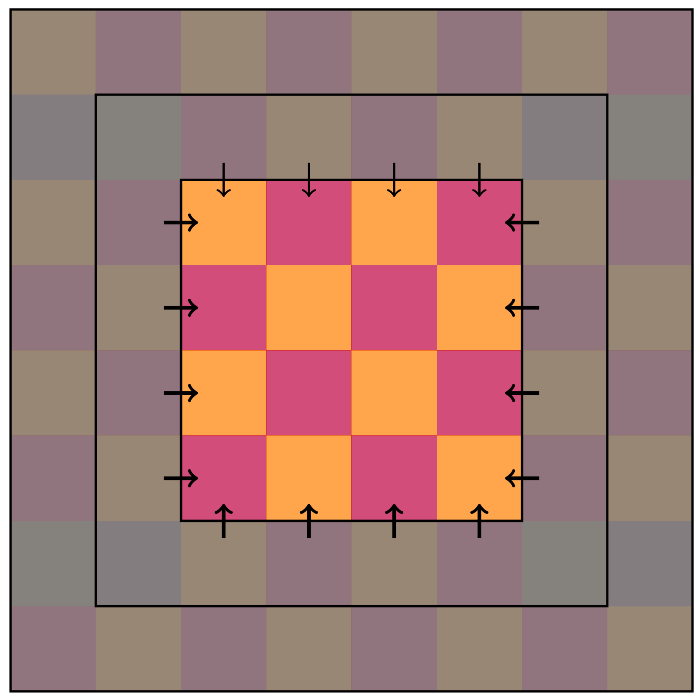
$$A = D_{PC}^\dagger D_{PC}$$
$$= (M_5 - \kappa_b^2 M_{eo}^4 M_5^{-1} M_{oe}^4)^\dagger (M_5 - \kappa_b^2 M_{eo}^4 M_5^{-1} M_{oe}^4)$$

- Need Dirichlet boundary conditions for this operator
- Implementing the Additive Schwartz method for this four-hop operator

before 1st hopping term



before 4th hopping term



MSPCG

- Multisplitting (additive Schwartz) converges, but slowly
- Try using as a preconditioner in a conventional preconditioned CG

$r_0 = b - Ax_0$
$z_0 = M^{-1} r_0$
$p_0 = z_0$
$k = 0$
while have not converged **do**
  $\alpha_k = \langle r_k, z_k \rangle / \langle p_k, A p_k \rangle$
  $x_{k+1} = x_k + \alpha_k p_k$
  $r_{k+1} = r_k - \alpha_k A p_k$
  $z_{k+1} = M^{-1} r_{k+1} \leftarrow A_s x_s^{(k+1)} = r^{(k)} + A_s x_s^{(k)}$
  only first cycle, zero initial guess, iterate a fixed number of times
  $\beta_k = \langle z_{k+1}, r_{k+1} \rangle / \langle z_k, r_k \rangle$
  $p_{k+1} = z_{k+1} + \beta_k p_k$
  $k = k + 1$
end while

- Implement as an inner CG (local)
- Preconditioner runs on-node
- Burns local Flops
- Avoids internode communication

MSPCG Effect on Outer CG Iterations

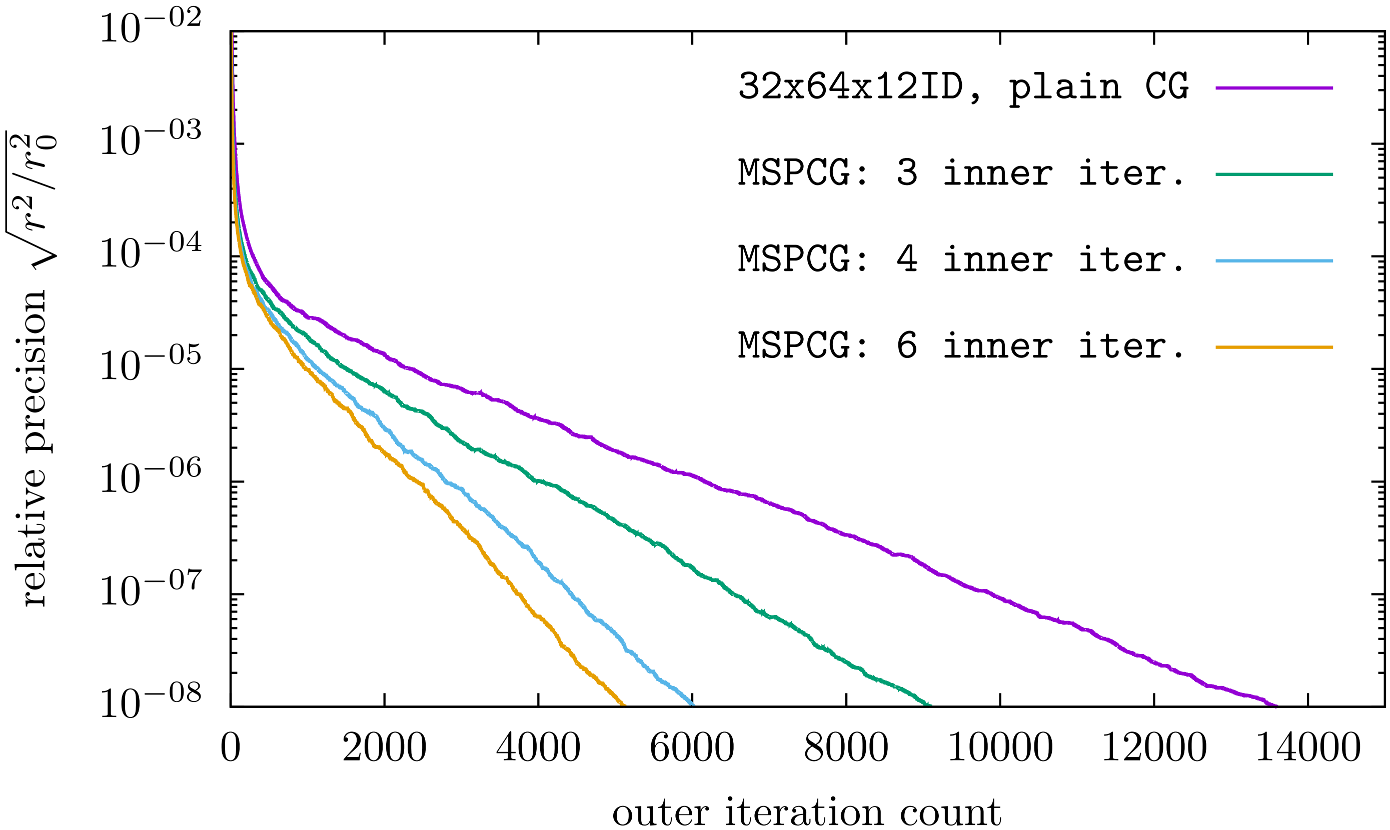


Figure 3: MSPCG solve on a $32^3 \times 64$ lattice $(a^{-1} = 1.37\ \mathrm{GeV})$ with physical pion mass. Test performed on CORI at NERSC on 128 KNL nodes.

Jiqun Tu

- CG converges stably even when only a few preconditioning steps are done
- Converging preconditioner fully reduces outer CG iteration, which requires internode communication, by about a factor of 3.

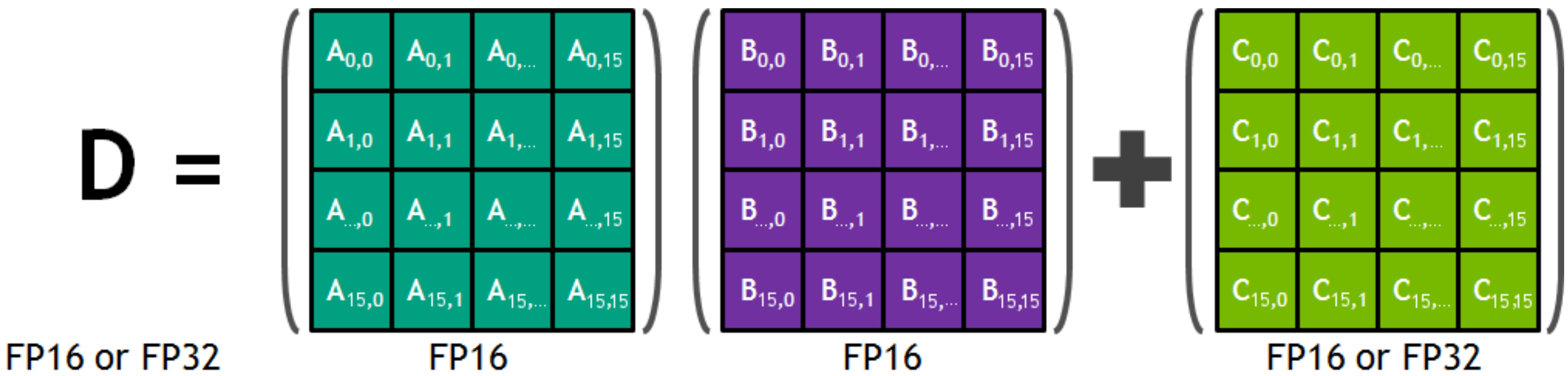
MSPCG Implementation on V100

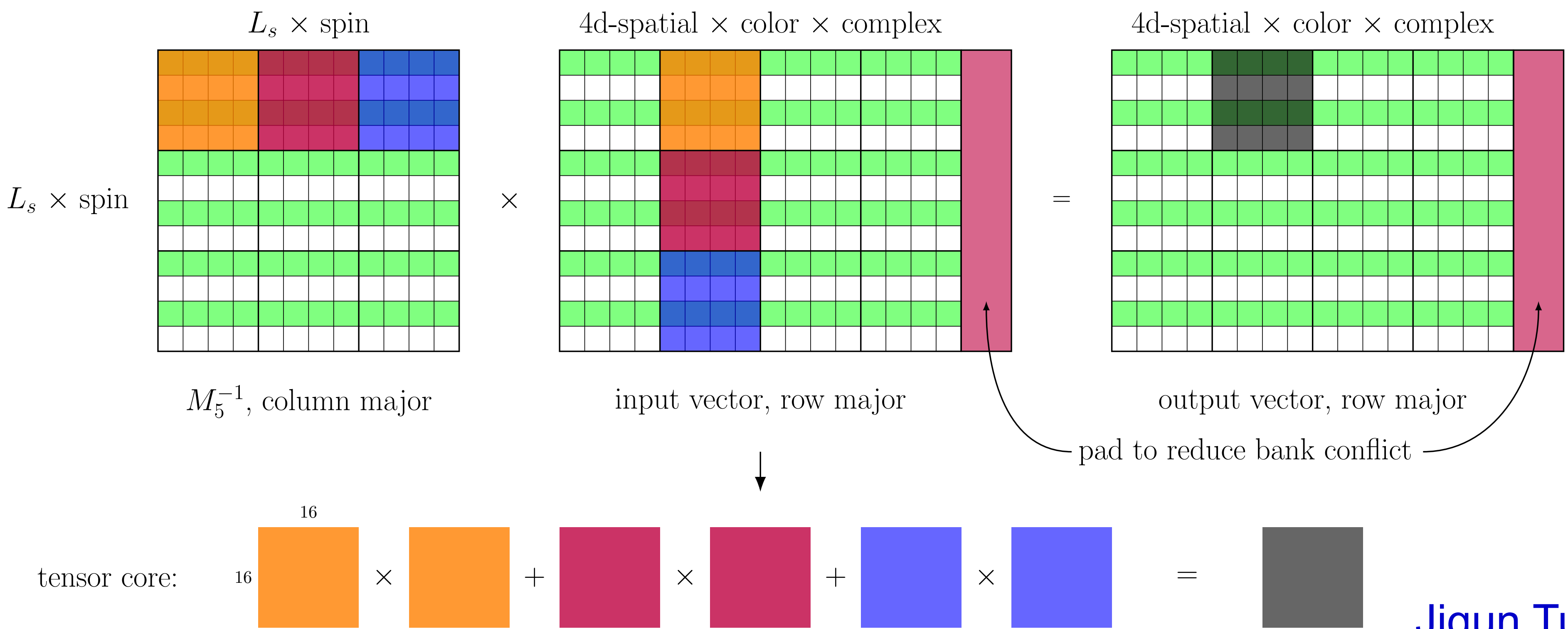
- Built on QUDA CG by adding a new preconditioner.
- Memory bandwidth for on-node calculations is limited
- Fusing 5-d operations with 4d-spatial operations allows reuse of data fetched from memory.

$$\left[1-\kappa_b^2 \underbrace{M_\phi^\dagger D_w^\dagger}_{\text{fuse}} \underbrace{M_5^{-\dagger} M_\phi^\dagger D_w^\dagger}_{\text{fuse}} \underbrace{M_5^{-\dagger}\right]\left[1-\kappa_b^2 M_5^{-1} D_w}_{\text{fuse}} \underbrace{M_\phi M_5^{-1} D_w}_{\text{fuse}} M_\phi\right]$$

- Since data is already in memory, can use the tensor cores to do this work.
- Enormous floating point power available (90 TFlops in half-precision per V100), but loading data just to use tensor cores would be too slow.
- Implemented by Jiqun (Jackson) Tu, including scaling input data to handle limited dynamic range of half-precision.
- Much of underlying framework code for implementation of M5 in QUDA written by Kate Clark of NVIDIA.

MSPCG and Tensor Cores





Jiqun Tu

Evolution on Summit

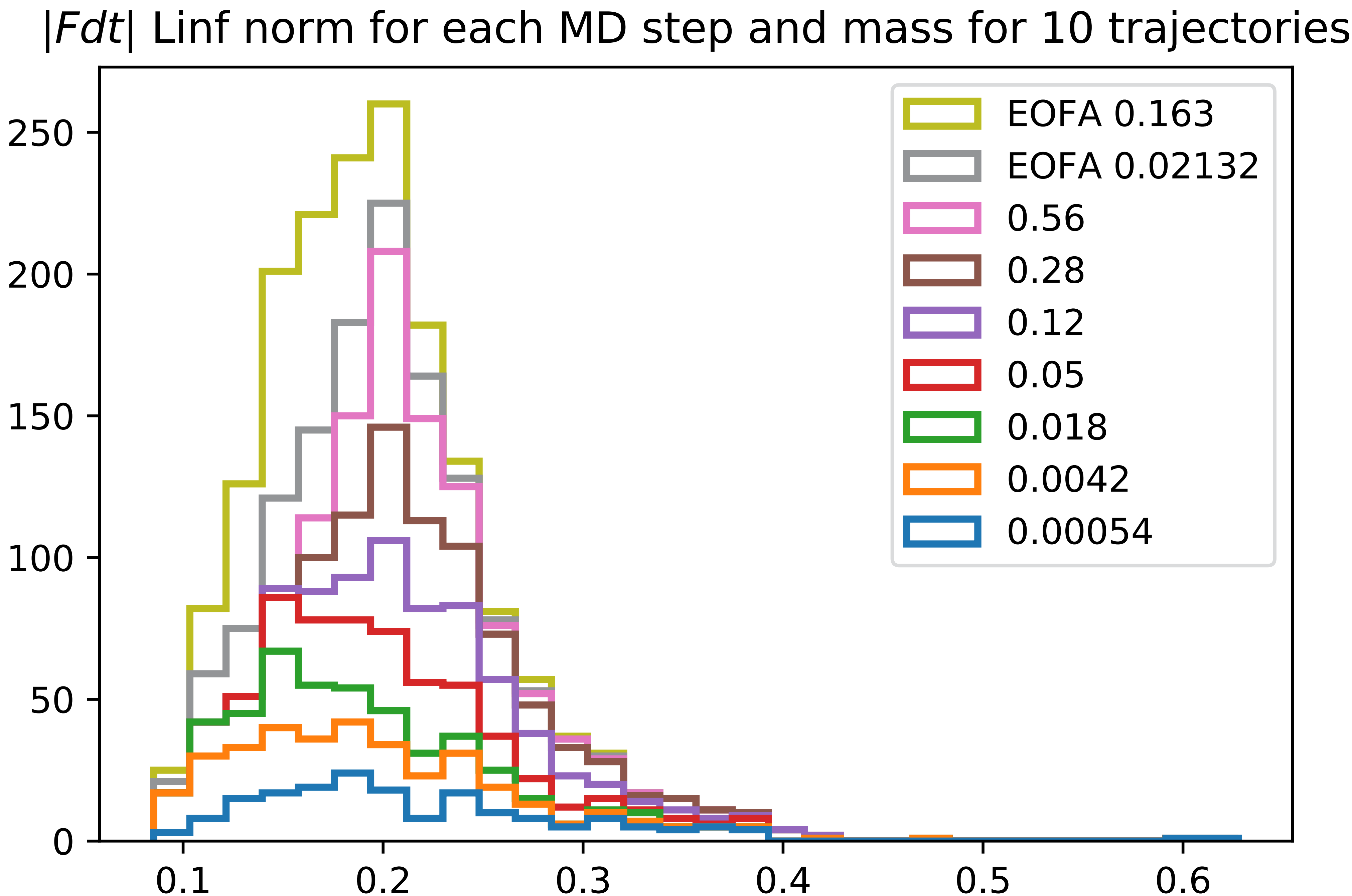
- Thermalized a $32^3 \times 64 \times 12$ lattice, with the desired input quark masses, on BGL at Mira.
    - * Provided a chance to tune Hasenbusch masses and step sizes
    - * Rough check of pion and kaon masses on a too-small volume
- Replicated 3 times in 4 dimensions (81 replicas overall) as initial lattice.
- Still needs thermalization, but allows tuning of HMC/EOFA while thermalizing.
- Have 350 trajectories and counting.
- Hasenbush masses for two flavor determinants

$$\frac{D(0.00054)}{D(0.0042)}\ \frac{D(0.0042)}{D(0.016)}\ \frac{D(0.016)}{D(0.05)}\ \frac{D(0.05)}{D(0.12)}\ \frac{D(0.12)}{D(0.28)}\ \frac{D(0.28)}{D(0.56)}\ \frac{D(0.56)}{D(1.0)}$$

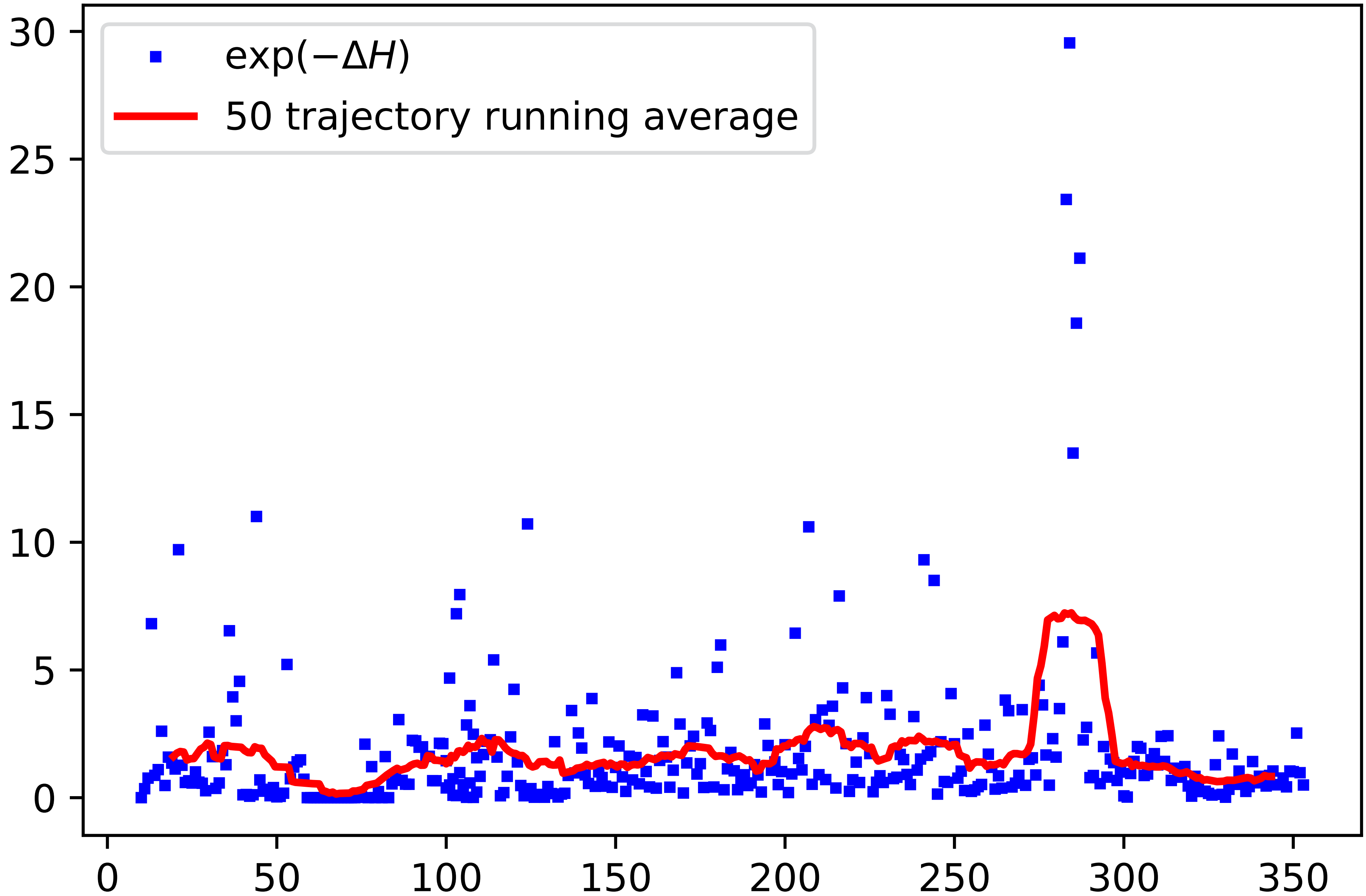
- Current trajectory time on 512 nodes, excluding I/O: 7929 seconds

Chulwoo Jung

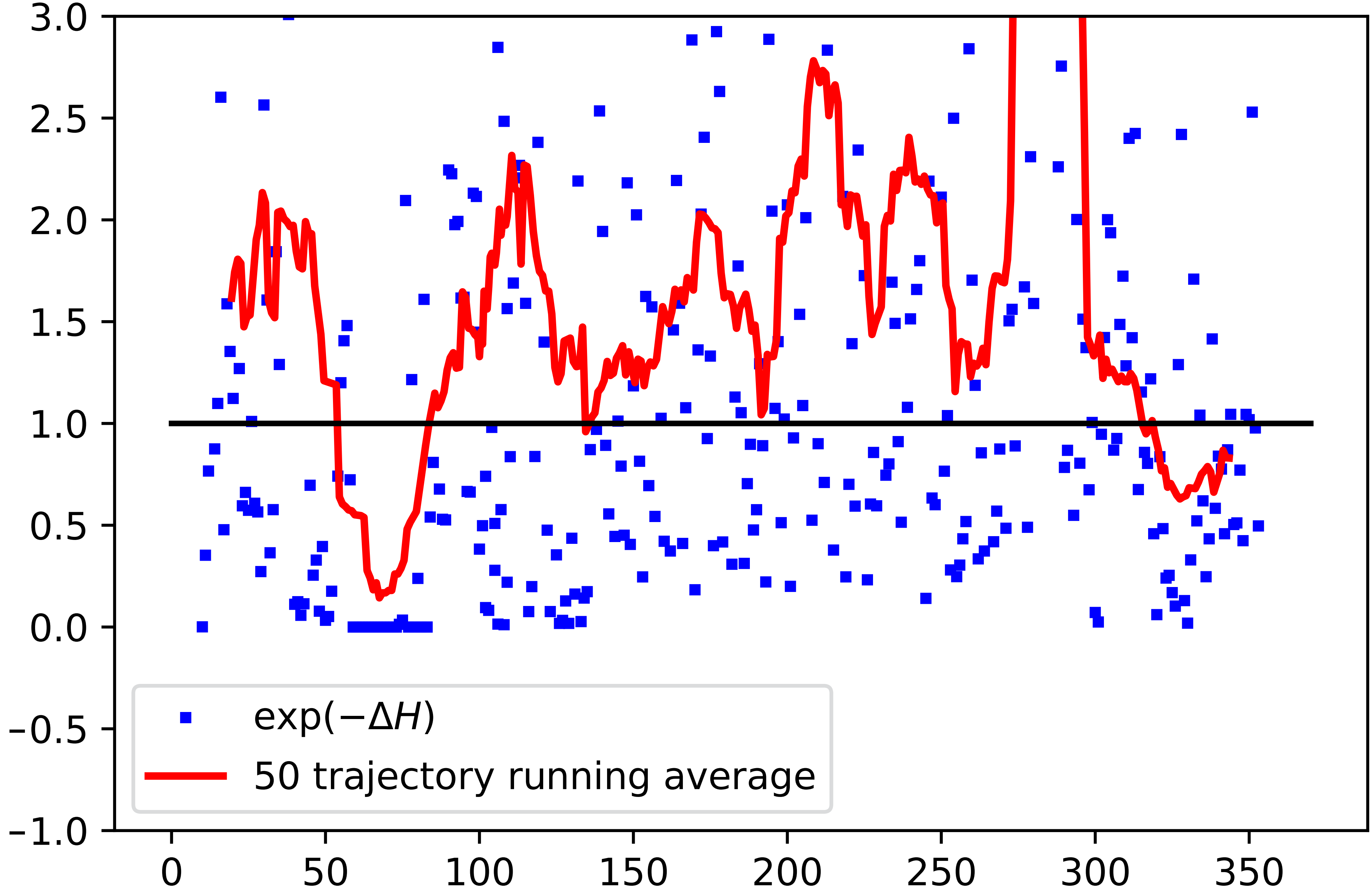
Histogram of Forces in Molecular Dynamics



Time History of exp(-ΔH)



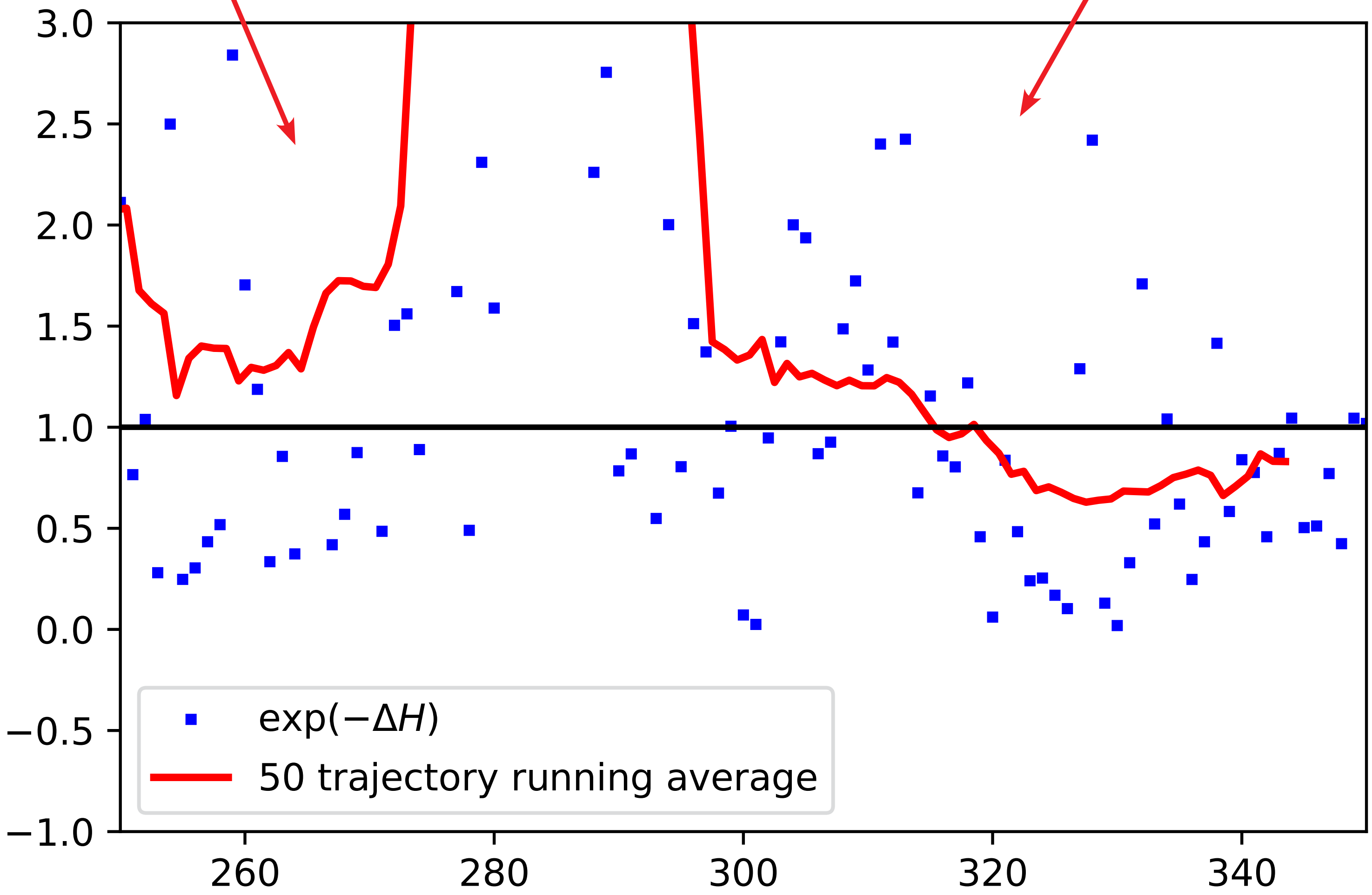
Zoomed Time History of exp(-ΔH)

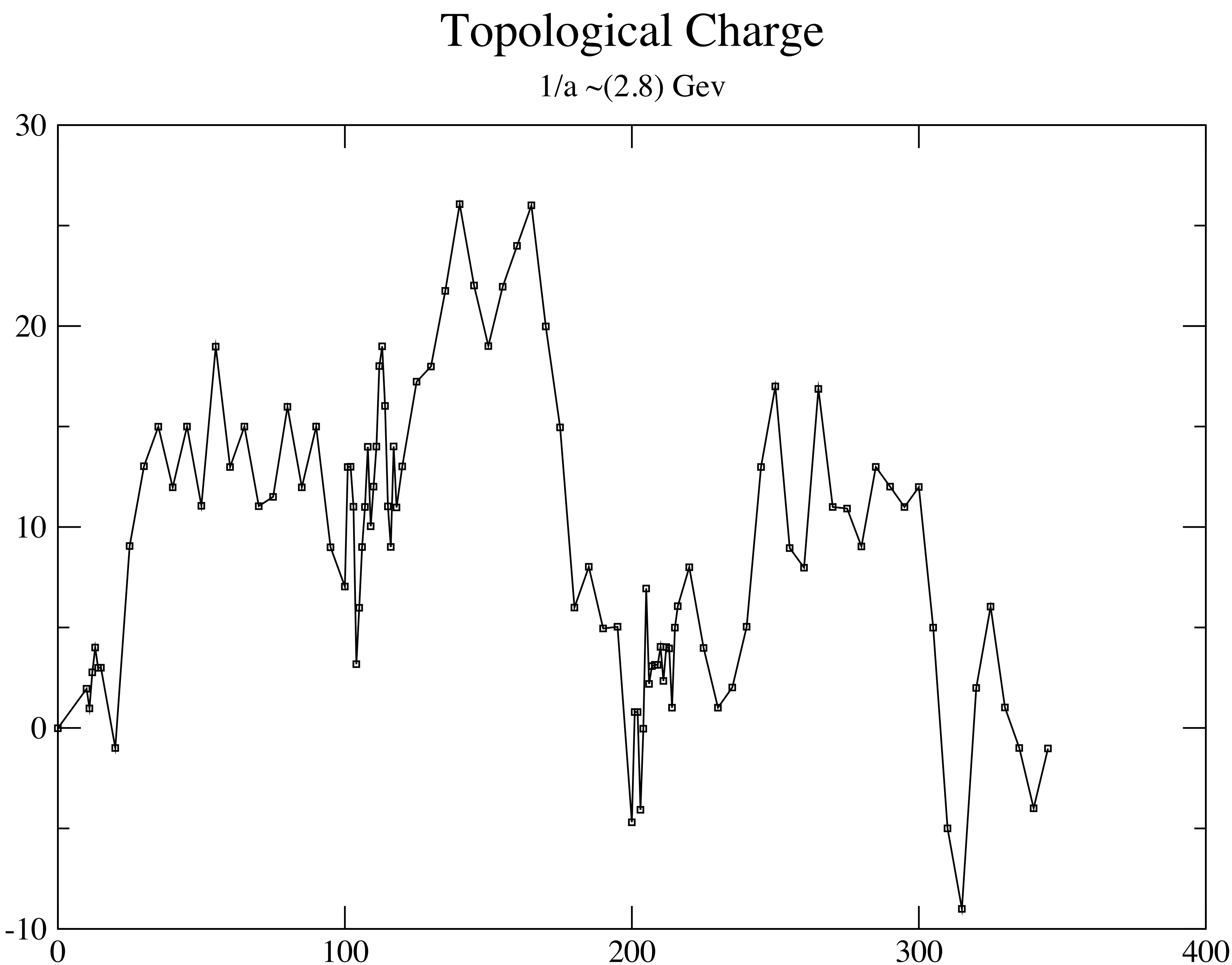


Last 50 Trajectories of Time History of exp(-ΔH)

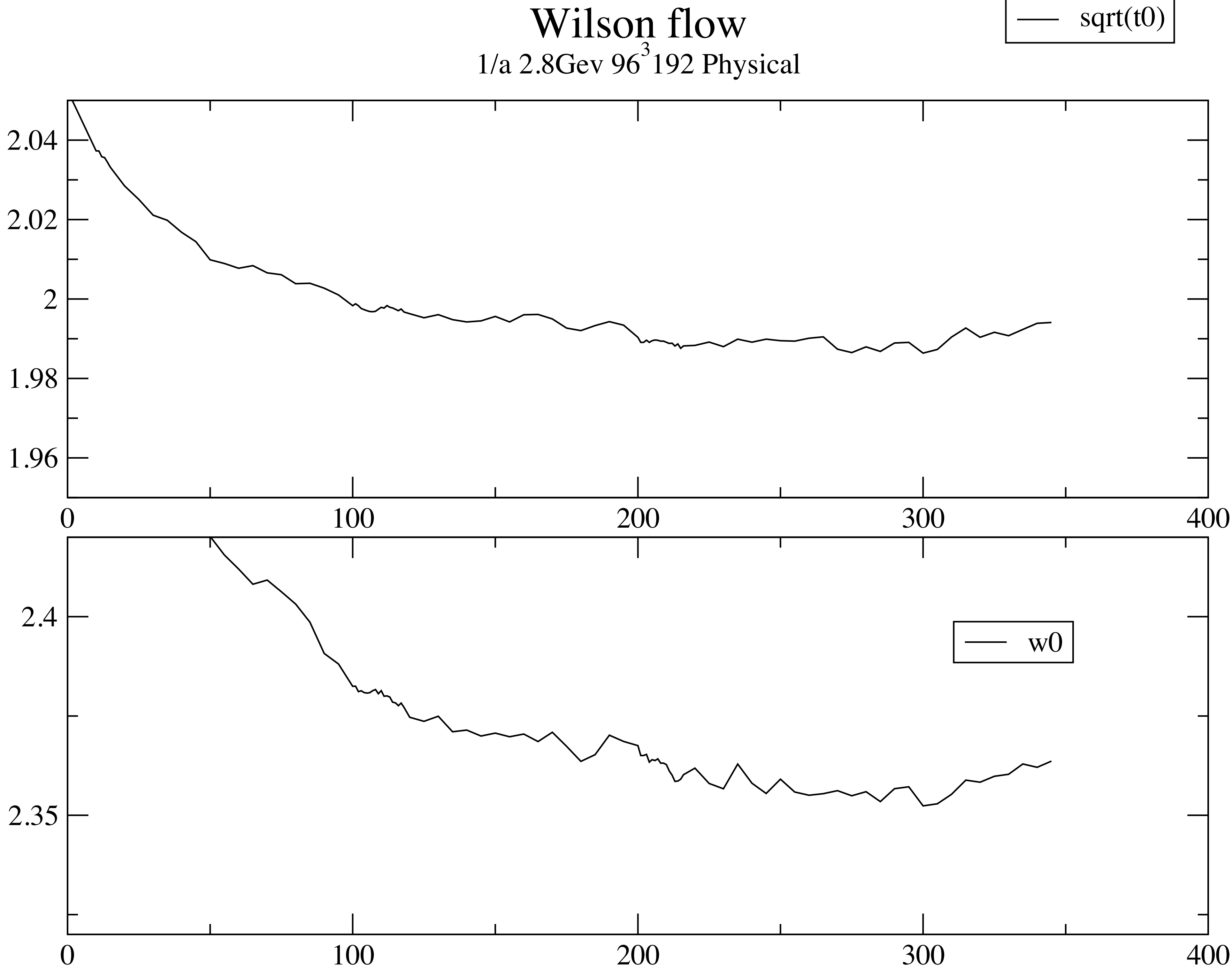
- Increased rational approximation (RA) bounds for EOFA heat bath
- This decreased RA precision.
- Tightened stopping condition for EOFA heat bath

- Increased rational approximation precision for EOFA heat bath.
- Tight stopping condition for EOFA heat bath





Topological Charge
1/a ~(2.8) Gev
30
20
10
0
-10
0
100
200
300
400



Wilson flow
1/a 2.8Gev 96^3 192 Physical
sqrt(t0)
2.04
2.02
2
1.98
1.96
0
100
200
300
400
w0
2.4
2.35
0
100
200
300
400

Summary

- 2 light flavor HMC + strange quark EOFA MDWF evolution underway on Summit with $1/a \approx 2.8$ GeV and 6.9 fm^3 volume
- Performance of MSPCG for lightest Hasenbusch ratio

  6377 iterations(with 06 inner_iterations each) with 076 reliable updates.
  True residual/target_r2: 1.7601e-06/1.7657e-06.
  Performance precise: 638.08 TFLOPS or 1.25 TFlops/node
  Performance outer: 2370.09 TFLOPS or 4.63 TFlops/node
  Performance preconditioner: 18709.41 TFLOPS or 36.5 TFlops/node
  Note these Flop counts are from QUDA, which uses more Flops in the 5d dense matrix parts of the code than other DWF codes.

- Performance for full (non-local) CG has improved, largely due to improved network performance.
- Increased performance of outer CG has decreased the relative improvement of MSPCG, but has decreased overall time to solution.
- MSPCG giving ~20% speed-up in current environment. 37% of time in outer CG, 63% in preconditioner. This is adjustable.
- Continuing to tune and look for improvements in performance.