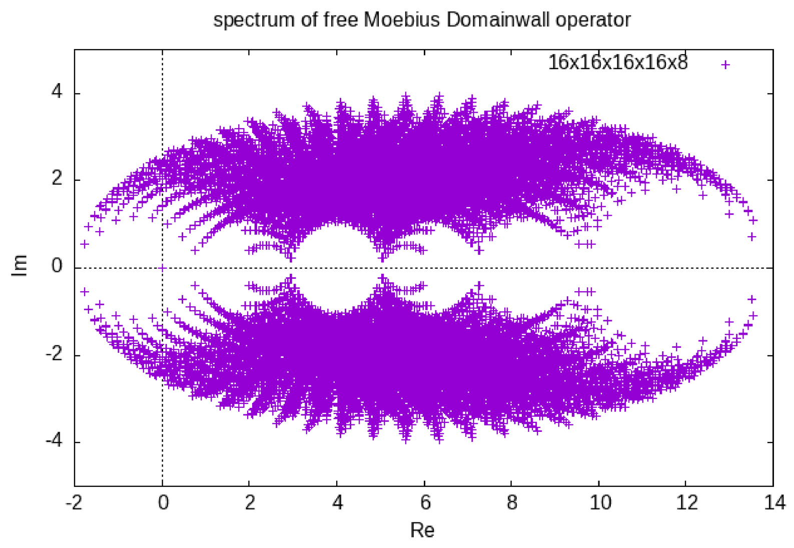
Spectrum of preconditioned Moebius domain-wall operators

Issaku Kanamori (RIKEN), Wei-Lun Chen (SOKENDAI) and Hideo Matsufuru (KEK)

Lattice QCD code Bridge++ http://bridge.kek.jp/Lattice-code/

(Moebius) domain-wall fermion

Spectrum is not "positive": the real part can be negative
(Wilson op. with large negative mass)



Most of the standard arguments for convergence of iterative solver fails:
CG solver with $D^\dagger D$
+ even-odd preconditioning
(+ mixed prec. scheme)

Any good precondtioner?

Physical degrees of freedom: $\underbrace{D(m=M_{PV})}_{D_{PV}}{}^{-1}D$ $\quad aM_{PV}=1$ Puali-Villars mass

Multigrid algorithms for domain-wall fermions: the gain is limited
cf. Boyle-Yamaguchi 2103.05034

Hierarchically deflated conjugate residual Yamaguchi-Boyle Lattice 2016
Preconditioning with Pauli-Villars oprator $D_{PV}^\dagger$
Brower et al. 2020 PRD 102 (2020) 9, 094517
We also did some trials and the all the attempt failed so far

How about the spectrum of the preconditioned operator?

$$Dx=b \quad \Leftrightarrow \quad LDRy=Lb,\ x=Ry$$

In this work, we investigate spectra of domain-wall/preconditioner operators, aiming to understand how the domain-wall solver can be improved from a view point of the spectra. It may also help to develop multigrid algorithms.

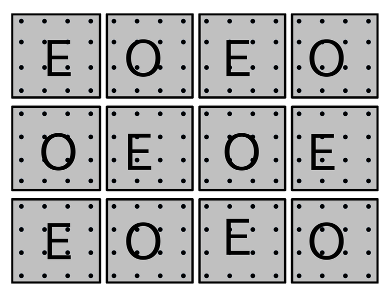
Preconditioning Operators

Site even-odd: with or without hopping C^{-1} can be easily obtained

$$D=\begin{pmatrix} D_{ee} & D_{eo} \\ D_{oe} & D_{oo}\end{pmatrix} = \underbrace{\begin{pmatrix} D_{ee} & 0 \\ 0 & D_{oo}\end{pmatrix}}_{\equiv C} \begin{pmatrix} 1 & D_{ee}^{-1}D_{eo} \\ D_{oo}^{-1}D_{oe} & 1\end{pmatrix} = \begin{pmatrix} 1 & D_{eo}D_{oo}^{-1} \\ D_{oe}D_{ee}^{-1} & 1\end{pmatrix} C$$

Block Even-Odd: with or without hopping between domains

$$D=\begin{pmatrix} D_{EE} & D_{EO} \\ D_{OE} & D_{OO}\end{pmatrix} = \underbrace{\begin{pmatrix} D_{EE} & 0 \\ 0 & D_{OO}\end{pmatrix}}_{\equiv B} + \begin{pmatrix} 0 & D_{EO} \\ D_{OE} & 0\end{pmatrix}$$

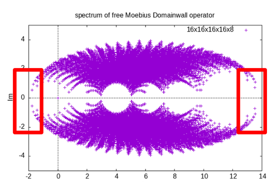


Precondition operators we tried:

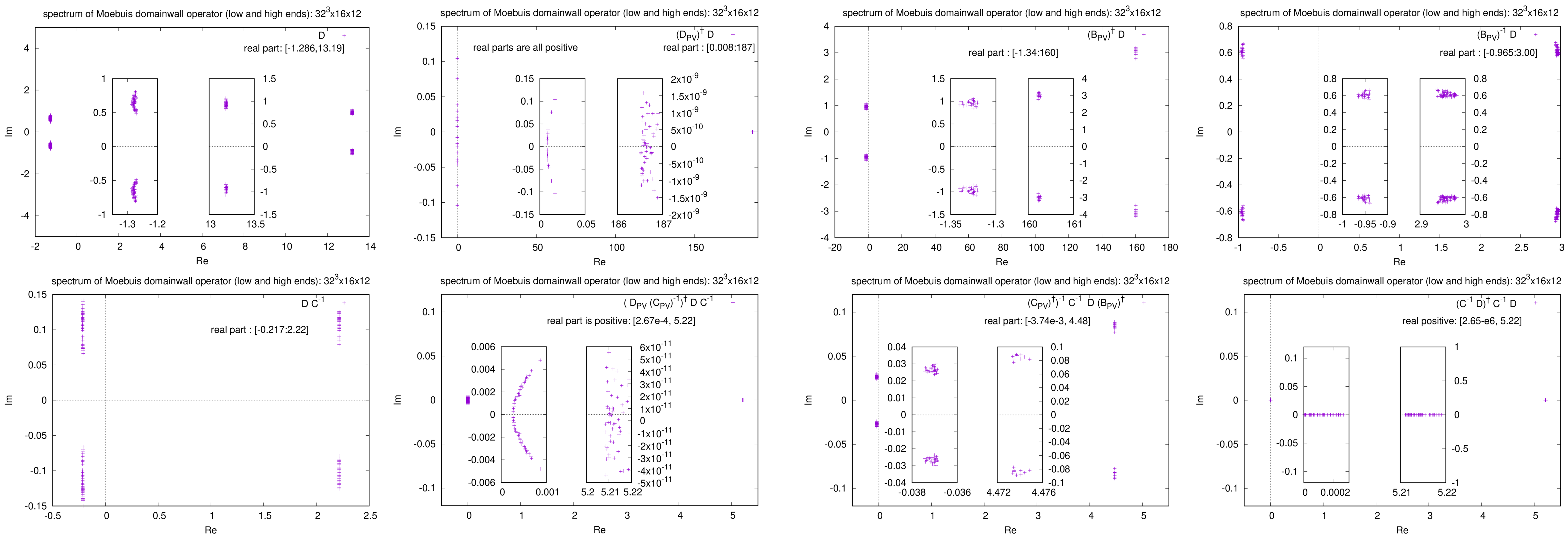
- $D_{PV}^\dagger$ as an approximation of D_{PV}^{-1} (cf. Brower et al.)
- $B_{PV}^\dagger$ as an approximation of D_{PV}^{-1}
- (loosely solved) B_{PV}^{-1}
- Combining C^{-1} to the above
- For a possible usage in multigrid algorithm, it is better to avoid 2-hop operation to simply the coarse grid operator

Spectra

Low lying/highest eigenvalue parts of various operators



The both end are calculated with implicit restarted Arnoldi algorithm implemented in Bridge++



Configuration: JLQCD finite-T ($T<T_c$), $m_l=0.1m_s$ cf. I.K., Lattice 2022

Code set: Bridge++

C++ object oriented framework
Portable, easy to read, and extendable keeping reasonable performance
Standard fermions, HMC, some measurements with test suite
Version 1.0 release: 2009 new architectures have appeared since then

Extended to enable implementations specific to recent architectures
Y.Akahoshi et al. J.Phys.Conf.Ser. 2207 (2022) 1, 012053

- GPU version with OpenACC/CUDA
  1 TFlop (OpenACC, fp32) and 2.3 TFlops (CUDA, fp32) on GH200
  Even-odd preconditioned domain-wall $D^\dagger D$ on 64x64x64x32 lattice
- SIMD version for A64FX (Fugaku, etc.) T.Aoyama et al., Lattice 2022,...
  380 Gflops/node (fp32) the same $D^\dagger D$ on 64x16x8x4 lat./proc, 4 proc/node
- another SIMD version for AVX-512 different data layout from that for A64FX
  I.K and H.Matsufuru, EPJ Web Conf 175 (2018) 09002; Lecture Notes in Computer Science, vol 10962 (2018) 456.

Homework

- Some combinations of operators seem to have smaller condition number: may improve iterative solver
- For application to larger systems, reducing communication is desired: efficiency of
  - SAP solver
  - SAP-like smoother for multi-grid algorithm

  are being tested.
- Which operator is the best for construction of corse grid operator?
- (Improving the performance on GPU)

Acknowledgments: This work is supported by JSPS KAKENHI (JP20K03961, JP23K22495), the MEXT as `Program for Promoting Researches on the Supercomputer Fugaku' (Simulation for basic science: from fundamental laws of particles to creation of nuclei, JPMXP1020230411) and Joint Institute for Computational Fundamental Science (JICFuS), and SOKENDAI Student Dispatch Program. Supercomputer "Fugaku" at RIKEN Center for Computational Science through Usability Research ra000001 was used for numerical calculations. Some parts of the code development were performed on Wisteria/BDEC-01 Odyssey provided by Multidisciplinary Cooperative Research Program in Center for Computational Sciences, University of Tsukuba.