High-performance computation of correlation functions

Eloy Romero Alcalde

JLab, USA

SciDAC-5 PI Meeting
September 17 2025

Big pic

1. Gauge generation
computationally intensive

2. Compute observables
computationally intensive

3. Analysis
meh

Dirac matrix, D:

- 4D (space-time) lattice object
- the links are 12×12 matrices from the configuration gauge links

In computing observables, the time is dominated by:

- Solving linear systems $D\mathbf{u} = \mathbf{v}$ iteratively
- Tensor contractions

Calculating correlation functions with distillation

One needs to compute:

(cheap)

- Distillation basis:
  $V_t \approx \; k$ eigenvecs of $D_{t,t}$

(expensive)

- Propagators: $V_{t_1}^{\dagger} \; D^{-1} \; V_{t_0}$
- Generalized propagators:
  $V_{t_2}^{\dagger} \; D^{-1} \; I_{t_1} \; \Gamma \; D^{-1} \; V_{t_0}$
- Disconnected loops:
  trace$[\; \Gamma \; [D^{-1}]_{t,t} \;]$

(cheap)

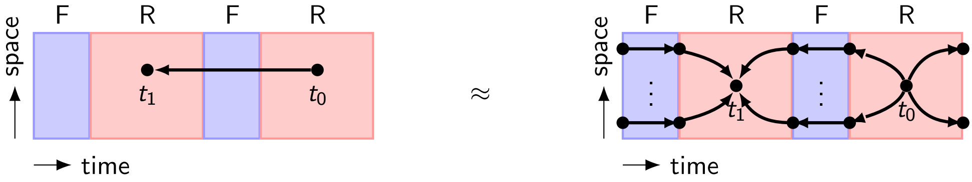
- Meson: $V_t^{\dagger} \; \Gamma \; V_t$
- Baryons: $< V_t, V_t, V_t >$

(expensive)

- The correlation functions:
  $f_i = \sum_j f_{i,j}$, where $f_{i,j} = \prod_k M^{(i,j,k)}_{\mathcal{I}(i,j,k)} \in \mathbb{C}$
  and $M^{(i,j,k)}_{\mathcal{I}(i,j,k)}$ are mesons, baryons, propagators, or generalized propagators

Efforts in the Dirac equation

Domain Decomposition [K. Orginos, B. Slimmer, W&M]



- Divide the domain and image of a matrix D into two non-overlapping domains F (frozen sites) and R (disconnected domains)
- We are interested in D^{-1} restricted to R: $\left[\begin{array}{c|c} D_F & D_{FR} \\ \hline D_{RF} & D_R \end{array}\right]^{-1} = \left[\begin{array}{c|c} \cdots & \cdots \\ \hline \cdots & D^{-1}_{(R,R)} \end{array}\right]$
- Approximate perambulator:

$$V_{t_1}^{\dagger} D^{-1}_{(R,R)} V_{t_0} \approx V_{t_1}^{\dagger} (D_R^{-1} + D_R^{-1} D_{RF} D_F^{-1} D_{FR} D_R^{-1}) V_{t_0}$$

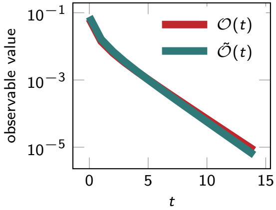
Efforts in the Dirac equation

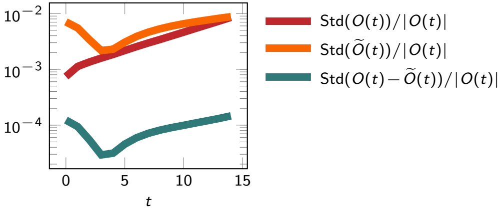
Domain Decomposition [K. Orginos, B. Slimmer, W&M]

- Divide the domain and image of a matrix D into two non-overlapping domains F (frozen sites) and R (disconnected domains)
- We are interested in D^{-1} restricted to R: $\left[\begin{array}{c|c} D_F & D_{FR} \\ \hline D_{RF} & D_R \end{array}\right]^{-1} = \left[\begin{array}{c|c} \cdots & \cdots \\ \hline \cdots & D^{-1}_{(R,R)} \end{array}\right]$
- Approximate perambulator:

$$V_{t_1}^\dagger D^{-1}_{(R,R)} V_{t_0} \approx V_{t_1}^\dagger (D_R^{-1} + D_R^{-1} D_{RF} D_F^{-1} D_{FR} D_R^{-1}) V_{t_0}$$

Comparison of a nucleon 2 point correlation function with the full perambulator, $O(t) = \langle \mathcal{N}(t)\overline{\mathcal{N}}(0)\rangle$, and with the cheap perambulator, $\widetilde{O}(t)$:





Efforts in the Dirac equation

Multigrid [T. Whyte, Jülich Supercomputer Ctr, A. Stathopoulos, W&M]

New **Chiral SVD**† prolongators:

1. Get K null vectors, that is, solving $D\psi_i = 0$ with K random initial guesses
2. Take each chiral component, $\Psi_0 = (1+\gamma_5)\Psi$ and $\Psi_1 = (1-\gamma_5)\Psi$
3. Divide the lattice into d small domains (blocking), Λ_j, and select the $k \leq K$ most frequent directions on Ψ_i on each Λ_j, $\mathrm{SVD}_k(\Psi_i(\Lambda_j))$
4. Construct the prolongator as $[\ \mathrm{SVD}_k(\Psi_i(\Lambda_j))$ for $i \in \{0,1\}, 1 \leq j \leq d\ ]$

† T. Whyte et al., Accelerating multigrid with streaming chiral SVD for Wilson Fermions in lattice QCD. `https://arxiv.org/abs/2505.14399`

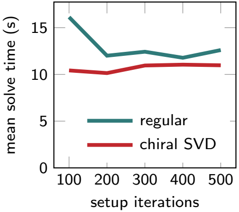
Efforts in the Dirac equation

Multigrid [T. Whyte, Jülich Supercomputer Ctr, A. Stathopoulos, W&M]

- The Chiral SVD prolongators are equivalent to the regular prolongators when $k = K$
- Their application cost is the same as the regular prolongators generated with k null vectors,
- but they perform closer to prolongators generated with K null vectors

Average number of iterations for each multigrid level on a $32^3 \times 64$ lattice with $K = 2k$ for all levels:

MG Level	Regular Prolong.	CSVD Prolong.
0	12.0	8.0
1	34.0	16.0
2	115.2	58.1



Efforts in the Dirac equation

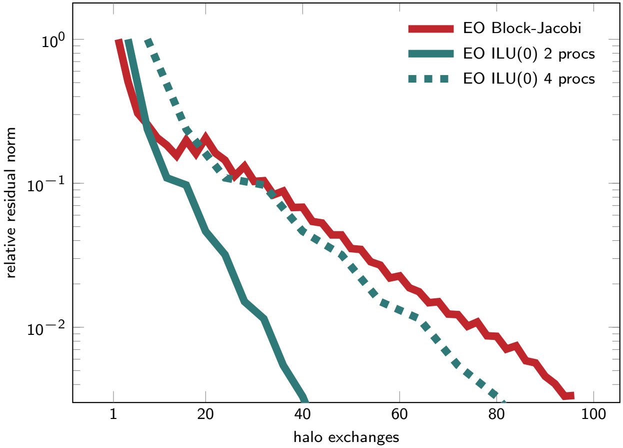
ILU(0) [H. Liu, A. Stathopoulos, W&M, X. S. Li, Y. Liu, LBNL]

- We are exploring the potential of using Incomplete LU (ILU) factorizations as preconditioners as complement for multigrid
- Despite the fact that solving the triangular factors is difficult to parallelize, ILU may be useful:
  - as part of the domain decomposition based preconditioners
  - for solving the coarse operator with a small number of processes
- We are planning on taking advantage of SuperLU optimized sparse triangular solver kernels

Efforts in the Dirac equation

ILU(0) [H. Liu, A. Stathopoulos, W&M, X. S. Li, Y. Liu, LBNL]

Preliminary study of several preconditioners for accelerating the solution of the coarsest operator on a typical 2-level mutltigrid for a lattice of $32^4 \times 64$.



Efforts in contractions (`redstar`)

J. Chen, JLab

- The correlation functions: $f_i = \sum_j f_{i,j}$, where $f_{i,j} = \prod_k M^{(i,j,k)}_{\mathcal{I}(i,j,k)} \in \mathbb{C}$ and $M^{(i,j,k)}_{\mathcal{I}(i,j,k)}$ are mesons, baryons, propagators, or generalized propagators

- Examples:
$$\begin{aligned} f_0 &= M^{(0,0,0)}_{0,1} M^{(0,0,1)}_{0,1} + M^{(0,1,0)}_{0,1} M^{(0,1,1)}_{1,2} M^{(0,1,2)}_{2,0} \\ f_1 &= M^{(1,0,0)}_{0,1,2} M^{(1,0,0)}_{0,3} M^{(1,0,0)}_{1,2,3} \end{aligned}$$

- $M^{(i,j,k)}_{\mathcal{I}(i,j,k)}$ are read from disk, they repeat, but they don't fit all in memory

- There are common sub-sequences of tensor products across several f_i

	Number of $M^{(i,j,k)}_{\mathcal{I}(i,j,k)}$		Read			Speedup from
Test	Total	Unique	from disk	TFLOPs	Time	prev. version
roper	1k	334	1 TiB	514	207 s	10x
nucleon 3 pt	71M	2k	1.8 TiB	1100	900 s	20x

Questions?