Multi-Polynomial Monte Carlo for Trace Estimation in Lattice QCD

speaker: Walter Wilcox*
Paul Lashomb*
Ronald Morgan*
Travis Whyte+

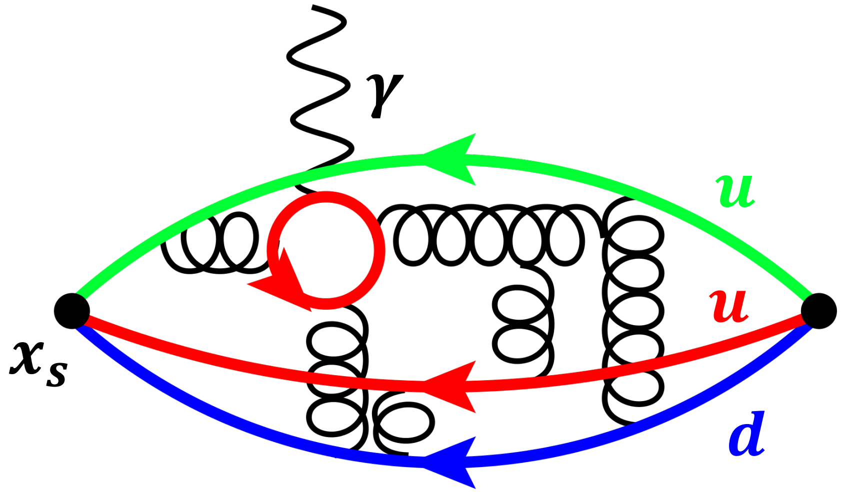
**Department of Physics Baylor University*
+DAMTP, Cambridge University

Disconnected Loops

- Physical quantities on lattice come in form of,

$$\langle\bar{\psi}(x)\Theta\psi(x)\rangle = -Tr(\Theta M^{-1})$$

- Hard to evaluate due to many matrix inversions needed to measure all the background fermionic degrees of freedom
- Treat the disconnected quark loops stochastically, through the use of noise vectors to project out operator contributions



Subtraction methods needed in order to reduce the variance of these noisy calculations

Noise Subtracted Trace Estimator

$$Tr(\Theta M^{-1}) = \frac{1}{N}\sum_{n=1}^{N} \eta^{\dagger(n)}\Theta(M^{-1} - \tilde{M}^{-1})\eta^{(n)} + Tr(\Theta \tilde{M}^{-1})$$

The only term that affects $Var[Tr(\Theta M^{-1})]$

The "trace correction"

- Each noise subtraction method uses a different matrix $\tilde{M}^{-1}$
- Will need to deal with "trace correction" later.

The GMRES Polynomial

- New GMRES Polynomial[5]
  - Implementation from factored roots,

$$\pi(\alpha) = \prod_{i=1}^{d} \left(1 - \frac{\alpha}{\tilde{\theta}_i}\right)$$

  - $\tilde{\theta}_i$ are the Leja ordered[6], harmonic Ritz values of the system
  - Much more stable

[5] J. A. Loe and R. B. Morgan, Numer. Linear Algebra Appl., 29:1–21, (2021).
[6] Z. Bai, D. Hu, and L. Reichel, IMA J. Numer. Anal. 14 (1994), pp. 563–581.

The GMRES Polynomial Cont.

- New GMRES Polynomial[5]
  - GMRES polynomial related to $p(A) \approx A^{-1}$ by $\pi(\alpha) = 1 - \alpha p(\alpha)$,

$$p(\alpha) = \sum_{k=1}^{d} u_k \qquad u_k = \frac{1}{\tilde{\theta}_k}\left(1 - \frac{\alpha}{\tilde{\theta}_1}\right)\left(1 - \frac{\alpha}{\tilde{\theta}_2}\right)\cdots\left(1 - \frac{\alpha}{\tilde{\theta}_{k-1}}\right)$$

  - For subtraction, $\tilde{M}^{-1}_{poly} := p(M)$

[5] J. A. Loe and R. B. Morgan, Numer. Linear Algebra Appl., 29:1–21, (2021).

Constructing the polynomial

- Run single cycle of GMRES(d)
- Extract d harmonic Ritz values
- Leja order these for numerical stability
- Apply polynomial, $p(\alpha) = \sum_{k=1}^{d} u_k(\alpha)$
- $p(\alpha)$ of degree $d-1$

Relative Variance

Relative Variance

$$\bar{\sigma}_R^2 = \frac{\bar{\sigma}_A^2}{\bar{\sigma}_{NS}^2}$$

➢ Lots of fluctuations across configurations. Need to *log-average*

Log-averaged Relative Variance

$$\bar{\sigma}_{R,log}^2 = 10^{\bar{\rho}_A^2 - \bar{\rho}_{NS}^2}$$

where, $\bar{\rho}_{R,log}^2 = \frac{1}{N}\sum_{j=1}^{N} \log_{10}[(\sigma_A^2)^{(j)}]$

Polynomial Degree and Lattice Volume

Lattice Settings:

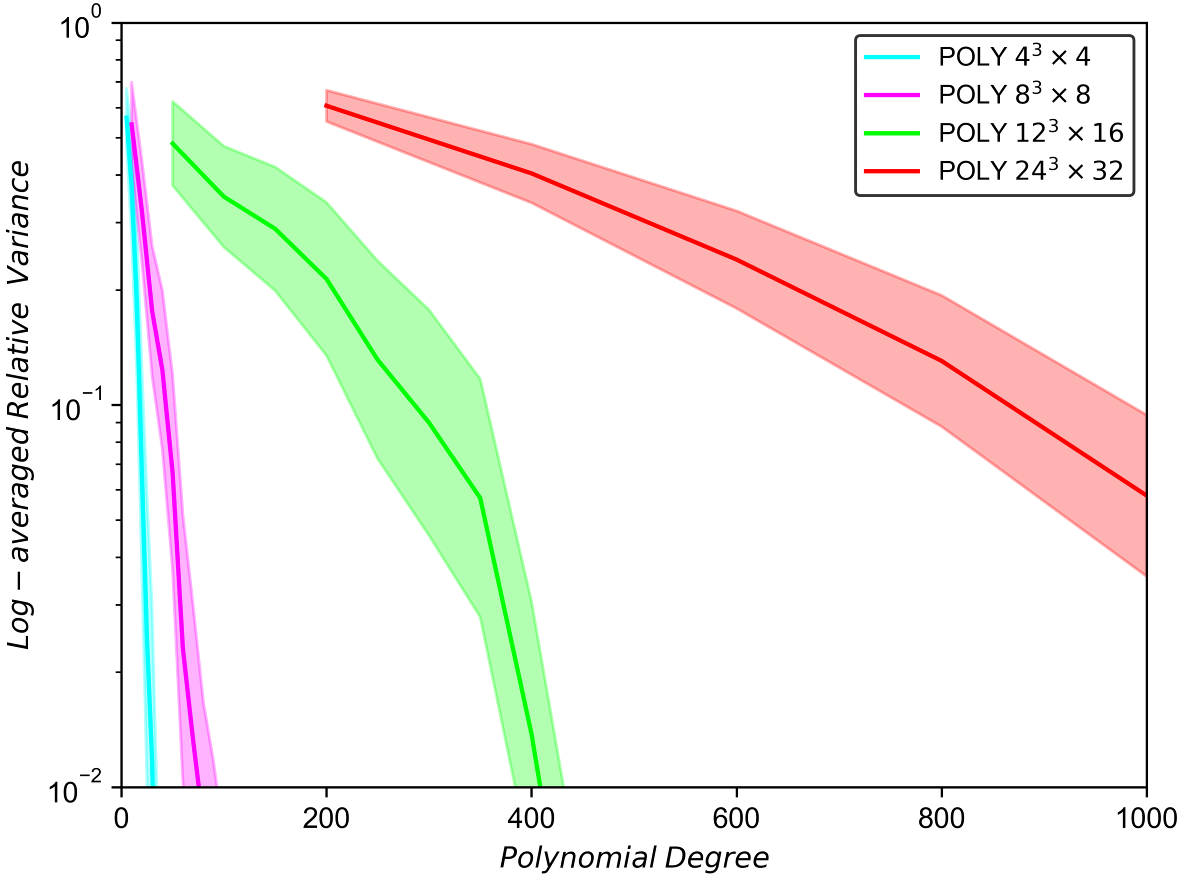
$$N_s^3 \times N_t = 4^3 \times 4,\ 8^3 \times 8,\ 12^3 \times 16,\ 24^3 \times 32$$

$$\beta = 6.0$$

$$\kappa = 0.1570$$

Scalar Operator:

$$\Theta = Re[\bar{\psi}(x)\psi(x)]$$



Multipolynomial Monte Carlo (MPMC)

$$Tr(\Theta M^{-1}) = \frac{1}{N}\sum_{n=1}^{N} \eta^{\dagger(n)}\Theta(M^{-1} - p(M))\eta^{(n)} + Tr(\Theta p(M))$$

The only term that affects $Var[Tr(\Theta M^{-1})]$

The "trace correction"

- Polynomial degree is too high to calculate $Tr(\Theta p(M))$ directly…

Multipolynomial Monte Carlo (MPMC)

$$Tr(\Theta M^{-1}) = \frac{1}{N}\sum_{n=1}^{N} \eta^{\dagger(n)}\Theta(M^{-1} - p_1(M))\eta^{(n)} + Tr(\Theta p_1(M))$$

Monte Carlo Estimation of Trace Correction

$$Tr(\Theta p_1(M)) = \frac{1}{Q}\sum_{q=1}^{Q} \xi^{\dagger(q)}\Theta(p_1(M) - p_2(M))\xi^{(q)}$$

$$+ \frac{1}{R}\sum_{r=1}^{R} \chi^{\dagger(r)}\Theta(p_2(M) - p_3(M))\chi^{(r)}$$

$$+ Tr(\Theta p_3(M))$$

Degree of $p_3(M)$ low enough to compute $Tr(\Theta p_3(M))$ directly!

Preconditioning and Double Polynomials

- Single polynomial preconditioning:
  - Right precondition $Ax = b$ using $p_{in}(A) \approx A^{-1}$,

Single Polynomial Preconditioning

$$\phi_{in}(A)y = b \qquad \phi_{in}(A) \equiv Ap_{in}(A)$$

- Double polynomial preconditioning[5]:
  - Preconditioner system a second time with $p_{out}(\phi_{in}(A)) \approx (\phi_{in}(A))^{-1}$,

Double Polynomial Preconditioning

$$\phi_{out}(\phi_{in}(A))z = b \qquad \phi_{out}(\phi_{in}(A)) \equiv \phi_{in}(A)p_{out}(\phi_{in}(A))$$

[5] J. A. Loe and R. B. Morgan, Numer. Linear Algebra Appl., 29:1–21, (2021).

Preconditioning and Double Polynomials

- Double polynomial preconditioner:
  - Combine preconditioners $p_{in}(A)$, $p_{out}(\phi_{in}(A))$ into one preconditioner,

Double Polynomial Preconditioner

$$p_{double}(A) \equiv p_{in}(A)p_{out}(\phi_{in}(A))$$

  - Orthonormal basis for Arnoldi is expensive to form and store
  - Much cheaper to form as it requires two smaller Krylov subspaces

Deflated Multipolynomial Monte Carlo

- Deflation can be used in three different ways:
  1. Deflation of Monte Carlo Levels (new)

Deflation in Monte Carlo Levels

$$Tr(p_j(M) - p_{j+1}(M)) \rightarrow Tr\left(p_j(M) - p_{j+1}(M) - \sum_k (p_j(\lambda_k) - p_{j+1}(\lambda_k)) z_k u_k^\dagger\right)$$

  2. Deflation in the main solves: $x^{(n)} = M^{-1}\eta^{(n)}$
  3. Deflation of the double polynomial $p_1(M)$ when forming (new)

- We will be making MPMC **and** Hutchinson more effective

Summary of Improvements to MPMC

Double Polynomial Subtraction

$$p_1(M) = p_{in}(M)p_{out}(\phi_{in}(M)) = p_{in}(M)p_{out}(Mp_{in}(M))$$

Polynomial Preconditioning of Linear Equations

$$Mp_{in}(M)y^{(n)} = \eta^{(n)}, \quad x^{(n)} = p_{in}(M)y^{(n)}$$

Three Different Applications for Eigenvalue Deflation

$$Tr(p_j(M) - p_{j+1}(M)) \rightarrow Tr\left(p_j(M) - p_{j+1}(M) - \sum_k (p_j(\lambda_k) - p_{j+1}(\lambda_k))z_k u_k^\dagger\right)$$

Trace for $12^3 \times 16$ with $\epsilon = 0.0005 * 12^3 * 16$

Improvements	Settings
Double poly	$deg = 30 * 31 - 1$
(Defl.) Double poly	$deg = 30 * 12 - 1$
Defl. of PP-GMRES	13 Eigenmodes
(Defl.) Double poly	$deg = 30 * 12 - 1$
Defl. of PP-GMRES	13 Eigenmodes
Defl. Of MC	

Method	Noise vectors	Time	MVP's
Hutchinson, GMRES(50)	888	3.49 days	$2.52 * 10^6$
1 Poly, $deg = 4$	694	2.71 days	$1.99 * 10^6$
2 Poly's, $deg = 929, 4$	2+1572= 1574	11.5 hours	$1.49 * 10^6$
3 Poly's, $deg = 929, 300, 4$	2+138+2367=2507	7.35 hours	$8.90 * 10^5$

Method	Noise vectors	Time	MVP's
Hutch., defl. PP(30)-G(50)	888	3.43 hours	$4.20 * 10^5$
1 Poly, $deg = 4$	694	2.71 hours	$3.31 * 10^5$
2 Poly's, $deg = 359, 4$	2+1572=1574	4.50 hours	$5.81 * 10^5$
3 Poly's, $deg = 359, 100, 4$	2+1107+1815=2924	5.50 hours	$7.01 * 10^5$

Method	Noise vectors	Time	MVP's
Hutch., defl. PP(30)-G(50)	888	3.43 hours	$4.20 * 10^5$
2 Poly's, $deg = 359, 4$	2+186=188	36.1 min's	$7.67 * 10^4$
3 Poly's, $deg = 359, 100, 4$	2+6+276=284	20.0 min's	$3.47 * 10^4$

Trace for $24^3 \times 24$ with $\epsilon = 0.0005 * 24^3 * 24$

Ave. # of Evals: 22.4 | Ave. # of Evals: 52.9

Matrix	Eigenvalues found from PP(50)-Arnoldi to $rn < 10^{-12}$: Noises, 3 M. Carlo's	Time, Hours	MVP's	Extra deflation, eigenval's from PP(70)-Arnoldi for 150 it's: Noises, 3 M. Carlo's	Time, Hours	MVP's
1	2, 3, 60	3.93	$3.72 * 10^4$	2, 3, 15	2.85	$2.59 * 10^4$
2	2, 3, 105	4.93	$4.52 * 10^4$	2, 3, 12	2.81	$2.56 * 10^4$
3	2, 3, 75	3.98	$3.82 * 10^4$	2, 3, 42	3.46	$3.18 * 10^4$
4	2, 6, 60	4.23	$4.04 * 10^4$	2, 3, 24	3.05	$2.81 * 10^4$
5	2, 66, 177	16.3	$1.68 * 10^5$	2, 3, 30	3.22	$2.88 * 10^4$
6	2, 6, 42	3.85	$3.57 * 10^4$	2, 3, 21	2.96	$2.68 * 10^4$
7	2, 39, 132	11.0	$1.04 * 10^5$	2, 3, 21	2.93	$2.70 * 10^4$
8	2, 3, 63	3.93	$3.62 * 10^4$	2, 3, 27	3.13	$2.82 * 10^4$
9	2, 39, 195	11.8	$1.20 * 10^5$	2, 3, 27	3.15	$2.87 * 10^4$
10	2, 3, 108	4.46	$4.35 * 10^4$	2, 3, 30	3.10	$2.93 * 10^4$

- Fluctuations across configurations reduced with extra deflation

Trace for $24^3 \times 24$ with $\epsilon = 0.0005 * 24^3 * 24$

Method	Noise vectors	Time	MVP's
Hutch., defl. PP(70)-GMRES	188	19.6 hours	$2.02 * 10^5$
3 Poly's, 10^{-12} for deflation	2, 17.1, 101	6.84 hours	$6.68 * 10^4$
3 Poly's, extra deflation	2, 3, 24.9	3.07 hours	$2.80 * 10^4$

- Extra deflation is an overall improvement
- Hutchinson would take **much** longer without deflation and preconditioning

Conclusions

- High-degree GMRES polynomials can be efficiently formed and used in lattice QCD for noise subtraction.
- The full trace estimator can be performed using our new Multipolynomial Monte Carlo method
- We can use double-polynomials, preconditioning, and eigenvalue deflation to make MPMC more effective
- Lots of variables that need further tuning, and many possibilities for future improvements