RHMC with Multiple Pseudofermions and Block Solvers [1808.01829]

Liam Keegan and Philippe de Forcrand
ETH Zurich

27/09/2018

Introduction

Dominant cost of RHMC lattice simulations: calculation of fermion force term.

- **Multiple pseudofermions** reduce the size of this force: allowing a larger molecular dynamics integrator step-size, at the cost of having to invert the Dirac operator acting on each pseudofermion field. [Clark, Kennedy 2007]

Having multiple vectors to invert allows use of a more efficient class of solver.

- **Block Krylov solvers** invert a matrix acting on multiple vectors simultaneously: can converge after significantly fewer iterations than when solving each vector separately. [O'Leary 1980]

We combine these two ideas to speed up RHMC simulations [1808.01829].

Hybrid Monte Carlo

We want to sample the partition function

$$\mathcal{Z} = \int dU e^{-S_\mathrm{g}} \det \left[M^\dagger M \right]^{N_\mathrm{f}/2} = \int dU e^{-S_\mathrm{g} - S_\mathrm{f}^\mathrm{eff}}. \tag{1}$$

Non-local action, so we use a non-local update, Hybrid Monte Carlo: Add gaussian field P, solve classical equations of motion:

$$\frac{dP^a_{x\mu}}{d\tau} = -\frac{\partial H}{\partial U^a_{x\mu}} = -\frac{\partial}{\partial U^a_{x\mu}} \left(S_\mathrm{f}^\mathrm{eff} + S_\mathrm{g} \right) \tag{2}$$

$$\frac{dU^a_{x\mu}}{d\tau} = \frac{\partial H}{\partial P^a_{x\mu}} = P^a_{x\mu} \tag{3}$$

where a is the color index, x the site index, and μ the direction index.

Numerical Integration

A HMC trajectory consists of:

- Generate random gaussian field P.
- Solve equations of motion numerically to evolve (U, P) by "time" τ.
- Use a Symplectic integrator (area preserving and reversible)
- Remove residual integration error with an accept/reject Metropolis step.

Fermion Force

Expensive part of HMC update is the calculation of the fermionic force term,

$$F^a_{x\mu} = -\frac{\partial S_{\mathrm{f}}^{\mathrm{eff}}}{\partial U^a_{x\mu}} = \mathrm{Tr}\left[\left(M^\dagger M\right)^{-\frac{N_{\mathrm{f}}}{2}} \frac{\partial \left(M^\dagger M\right)^{\frac{N_{\mathrm{f}}}{2}}}{\partial U^a_{x\mu}}\right], \tag{4}$$

where a is the color index, x the site index, and μ the direction index.

This would require inversion of whole Dirac operator - prohibitively expensive.

Pseudofermions

Starting from the gaussian integral $\int_0^\infty dr\, re^{-ar^2} = a^{-1}$ one can show

$$\det \left[M^\dagger M\right]^{N_\mathrm{f}/2} \propto \int d\phi d\phi^\dagger e^{-\phi^\dagger [M^\dagger M]^{-N_\mathrm{f}/2}\phi}, \tag{5}$$

and hence we can study the equivalent partition function

$$\mathcal{Z} = \int dUd\phi d\phi^\dagger e^{-S_\mathrm{g}-\phi^\dagger [M^\dagger M]^{-\frac{N_\mathrm{f}}{2}}\phi}, \tag{6}$$

where pseudofermions with the desired distribution can be generated by first sampling η from a gaussian distribution $p(\eta) \propto e^{-\eta^\dagger \eta}$, then constructing $\phi = \left[M^\dagger M\right]^{N_\mathrm{f}/4} \eta$.

Multiple Pseudofermions

This approach can be trivially extended to multiple pseudofermions, using

$$\det\left[M^\dagger M\right] = \det\left[\left(M^\dagger M\right)^{\frac{1}{n_{\mathrm{pf}}}}\right]^{n_{\mathrm{pf}}}, \tag{7}$$

the partition function can instead be written as

$$\mathcal{Z} = \int dU \prod_{i=1}^{n_{\mathrm{pf}}} \left(d\phi_i d\phi_i^\dagger\right) e^{-S_{\mathrm{g}} - \sum_{i=1}^{n_{\mathrm{pf}}} \phi_i^\dagger [M^\dagger M]^{-\frac{N_{\mathrm{f}}}{2n_{\mathrm{pf}}}} \phi_i}, \tag{8}$$

where η_i are again sampled from a normal distribution, and $\phi_i = \left[M^\dagger M\right]^{\frac{N_{\mathrm{f}}}{4n_{\mathrm{pf}}}} \eta_i$.

Force term

The force term for $n_{\rm pf}$ pseudofermions is given by

$$F^a_{x\mu}(\phi_i, U, n_{\rm pf}) = \sum_{i=1}^{n_{\rm pf}} \phi_i^\dagger \frac{\partial \left[M^\dagger M\right]^{-\frac{N_{\rm f}}{2n_{\rm pf}}}}{\partial U^a_{x\mu}} \phi_i. \tag{9}$$

Integrating over the pseudofermions we recover the exact $n_{\rm pf}$-independent force term:

$$\begin{aligned} \overline{F^a_{x\mu}(U, n_{\rm pf})} &\equiv \int \prod_{i=1}^{n_{\rm pf}} \left(p(\eta_i) d\eta_i\right) F^a_{x\mu}\left(\left[M^\dagger M\right]^{\frac{N_{\rm f}}{4n_{\rm pf}}} \eta_i, U, n_{\rm pf}\right) \qquad (10) \\ &= \mathrm{Tr}\left[\left(M^\dagger M\right)^{-\frac{N_{\rm f}}{2}} \frac{\partial \left(M^\dagger M\right)^{\frac{N_{\rm f}}{2}}}{\partial U^a_{x\mu}}\right], \end{aligned}$$

Force term variance

But the variance (and higher order cumulants) *do* depend on n_{pf}:

$$\left[\overline{F^a_{x\mu}(U,n_{\text{pf}})^2}\right]-\left[\overline{F^a_{x\mu}(U,n_{\text{pf}})}\right]^2=\frac{c_1}{n_{\text{pf}}}+\mathcal{O}(n_{\text{pf}}^{-2}). \tag{11}$$

Larger variance of F implies a smaller integrator step size to maintain a fixed acceptance rate. In simulations we can easily measure the expectation value of the norm F^2 of this pseudofermion force,

$$F^2(n_{\text{pf}})=\left\langle\sum_{ax\mu}\tfrac{1}{2}\left[F^a_{x\mu}(\phi_i,U,n_{\text{pf}})\right]^2\right\rangle=c_0+\frac{c_1}{n_{\text{pf}}}+\mathcal{O}(n_{\text{pf}}^{-2}). \tag{12}$$

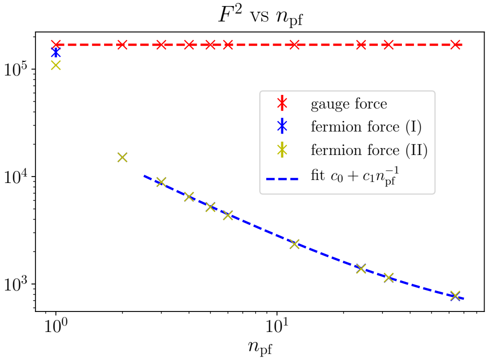
Numerical Simulations

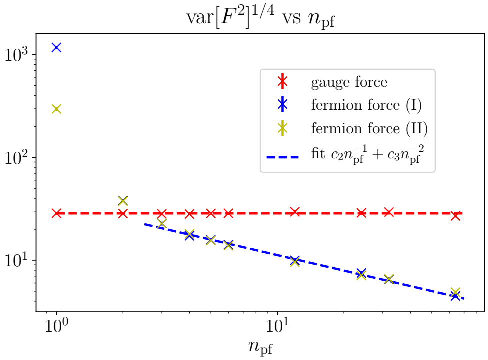
Simple, cheap setup for initial numerical study: $N_\mathrm{f} = 4$ QCD.

- Staggered fermions
  - no rooting required
  - unimproved
  - even-odd preconditioning
- Wilson gauge action
- 8^4 lattice
- $\beta = 5.12$
- $am = 0.002$
- HMC ($n_\mathrm{pf} = 1$)

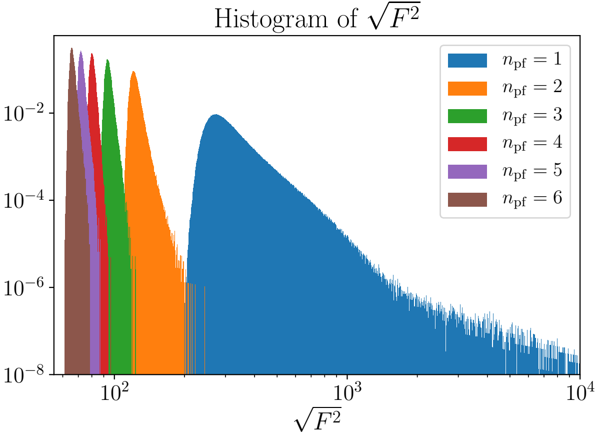
- RHMC ($n_\mathrm{pf} = 2 - 64$)
- Rational approx relative error
  - Molecular Dynamics: $|r|/|r_0| < 10^{-7}$ ($N_\mathrm{shifts} \simeq 15$)
  - Accept/reject step: $|r|/|r_0| < 10^{-15}$ ($N_\mathrm{shifts} \simeq 30$)
- Solver stopping criterion
  - Molecular Dynamics: $|r|/|r_0| < 10^{-7}$
  - Accept/reject step: $|r|/|r_0| < 10^{-14}$
- OMF2 integrator setting $\lambda = 1/6$

Multiple Pseudofermions: large n_{pf} fits

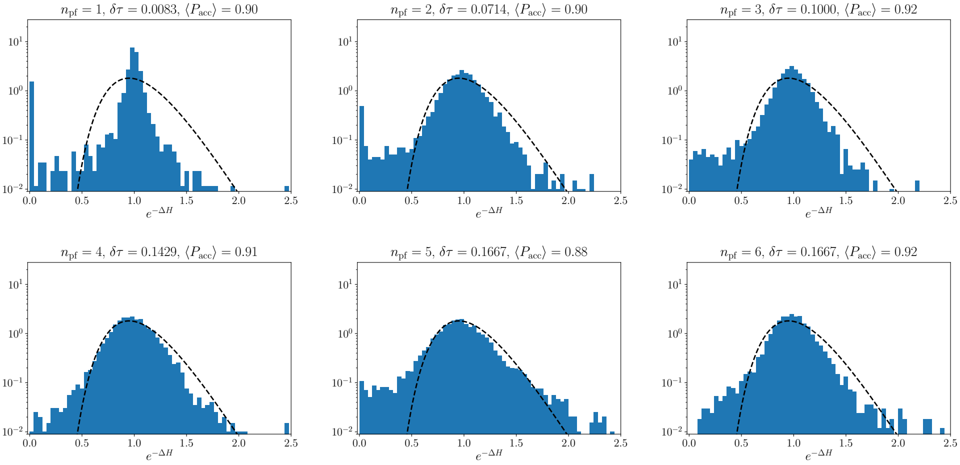




Multiple Pseudofermions: Force distribution



Multiple Pseudofermions: $\exp(-\Delta H)$ distribution



Predicting the acceptance rate

- Symplectic integrators *exactly* preserve a nearby "shadow" Hamiltonian [Kennedy et. al. 2012]
- Difference from actual Hamiltonian can be expanded in Poisson brackets
- Special case: 2nd order Omelyan integrator with $\lambda = 1/6$: [Bussone et. al. 2018]

$$\mathrm{var}\left[\Delta H\right] = 8 \left(\frac{\delta\tau}{12}\right)^4 \mathrm{var}\left[F^2(n_{\mathrm{pf}})\right] + \mathcal{O}(\delta\tau^6). \tag{13}$$

- Variance related to the acceptance via Creutz acceptance formula:

$$P_{\mathrm{acc}}(\Delta H) = \mathrm{erfc}(\sqrt{\mathrm{var}\left[\Delta H\right]/8}), \tag{14}$$

Cost estimates

Assuming that the total trajectory cost is dominated by the force term inversions, trajectory cost $\sim$ force term inversion cost × number of inversions:

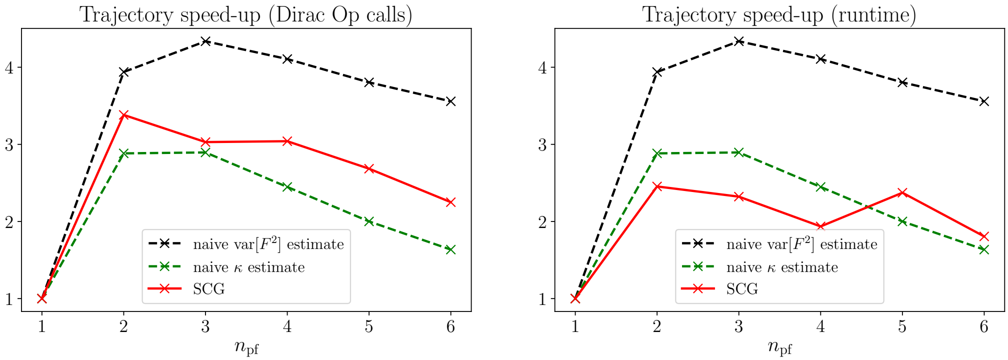
$$C(n_{\mathrm{pf}}) \propto n_{\mathrm{pf}}/\delta\tau \propto n_{\mathrm{pf}} \left(\mathrm{var}\left[F^2(n_{\mathrm{pf}})\right]\right)^{1/4}. \qquad (15)$$

Another estimate for the cost is given by [Clark, Kennedy 2007]:

$$C(n_{\mathrm{pf}}) \propto n_{\mathrm{pf}}^2 \kappa^{\frac{1}{n_{\mathrm{pf}}}}, \qquad (16)$$

where κ is the condition number of the Dirac operator.

Multiple Pseudofermions Speed-up



Krylov Solvers: Conjugate Gradient

Iteratively solve the system $Ax = b$ for the vector x given some vector b.
Here we take A to be a hermitian positive definite matrix.

Conjugate Gradient (CG):

- Start from some initial guess $x^{(0)}$ with residual $r = b - Ax^{(0)}$
- Construct solution $x^{(k)}$ from Krylov basis $\mathcal{K}_k = \left\{r, Ar, A^2 r, \ldots, A^{k-1} r\right\}$
- Solution minimises the error norm $|e_k|_A \equiv (x^{(k)} - x^*)^\dagger A(x^{(k)} - x^*)$

Block Krylov Solvers: Block Conjugate Gradient

For n_{pf} vectors b_j, where $j = 1, 2, \ldots, n_{\mathrm{pf}}$, with the *same* Dirac matrix for each vector, we can form a block matrix B whose j-th column is b_j, and solve the system $AX = B$.

Block Conjugate Gradient (BCG) [O'Leary 1980]:

- Start from some initial guess $X^{(0)}$ with residual $R = B - AX^{(0)}$
- Construct solution $X^{(k)}$ from Krylov basis $\mathcal{K}_k = \left\{R, AR, A^2R, \ldots, A^{k-1}R\right\}$
- Solution minimises the error norm $\mathrm{Tr}\left[(X^{(i)} - X^*)^\dagger A (X^{(i)} - X^*)\right]$

Block Solvers: convergence

There is an upper bound on the relative error of the BCG solution after k steps,

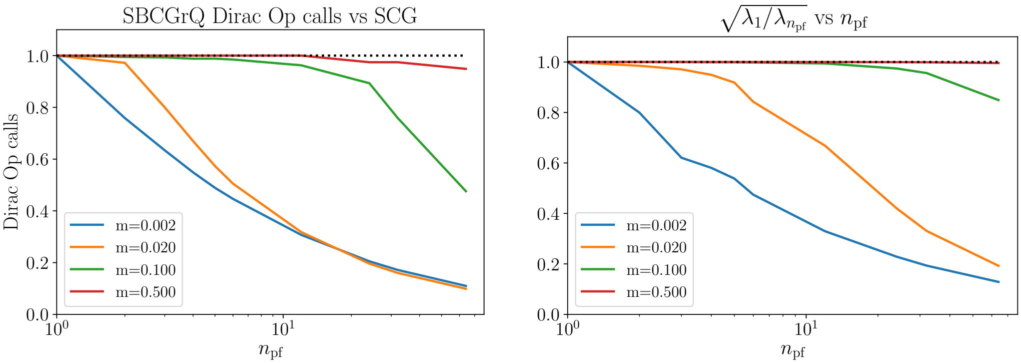
$$\frac{|e_k|_A}{|e_0|_A} \leq c_1(n_{\mathrm{pf}}) \left(\frac{1 - \sqrt{\lambda_{n_{\mathrm{pf}}}/\lambda_{\max}}}{1 + \sqrt{\lambda_{n_{\mathrm{pf}}}/\lambda_{\max}}} \right)^{2k} \tag{17}$$

where where $c_1(1) = 4$ and the eigenvalues of A in ascending order are given by $\{\lambda_1, \lambda_2, \ldots, \lambda_{n_{\mathrm{pf}}}, \ldots, \lambda_{\max}\}$.

Expanding in powers of $\sqrt{\lambda_{n_{\mathrm{pf}}}/\lambda_{\max}}$ this can be written as

$$\frac{|e_k|_A}{|e_0|_A} \leq c_1(n_{\mathrm{pf}}) e^{-4k\sqrt{\lambda_{n_{\mathrm{pf}}}/\lambda_{\max}}} + \mathcal{O}(k(\lambda_{n_{\mathrm{pf}}}/\lambda_{\max})^{3/2}), \tag{18}$$

Convergence



Block Solvers: QR-stabilisation and multishift

BCG**rQ** Algorithm [Dubrulle 2008]

- BCG residuals matrix can become badly conditioned: solver fails to converge.
- Issue resolved by QR-orthogonalisation of residuals matrix at each step

SBCGrQ Algorithm [Futamura et. al. 2012]

- For RHMC we need a multishift (i.e. $A + \sigma$) version of the solver
- Block Krylov basis is shift invariant
  (with shift invariant initial condition e.g. $X_0 = 0$).
- Can relate shifted residuals to unshifted ones to find shifted solutions without additional Dirac operator calls

SBCGrQ Algorithm

```
$X^{(s)}, P^{(s)}, Q, \in \mathcal{C}^{L\times n_{\mathrm{pf}}}$; $\alpha, \rho, \delta, \alpha^{(s)}, \beta^{(s)} \in \mathcal{C}^{n_{\mathrm{pf}}\times n_{\mathrm{pf}}}$
$X_0^{(s)} = 0$, $\{Q_0, \delta_0\} = \mathrm{qr}(B)$, $P_0^{(s)} = Q_0$;
  $\rho_0 = \delta_0, \alpha_0 = \alpha_0^{(s)} = \beta_0^{(s)} = 1$
for $k = 1, 2, \ldots$ until $\sqrt{\sum_j \delta_k(i,j) / \sum_j \delta_0(i,j)} < \epsilon \,\forall i$ do
    $\alpha_k \leftarrow (P_{k-1}^{(0)\dagger}(A+\sigma_0)P_{k-1}^{(0)})^{-1}$
    $\{Q_k, \rho_k\} \leftarrow \mathrm{qr}(Q_{k-1} - (A+\sigma_0)P_{k-1}^{(0)}\alpha_k)$
    $X_k^{(0)} \leftarrow X_{k-1}^{(0)} + P_{k-1}^{(0)}\alpha_k\delta_{k-1}$
    $P_k^{(0)} \leftarrow Q_k + P_{k-1}^{(0)}\rho_k^\dagger$
    $\delta_k \leftarrow \rho_k\delta_{k-1}$
    for $s = 1, \ldots, N_{\mathrm{shifts}} - 1$ do
        $\beta_k^{(s)} \leftarrow \left(1 + (\sigma_s - \sigma_0)\alpha_k + \alpha_k\rho_{k-1}\alpha_{k-1}^{-1}(1-\beta_{k-1}^{(s)})\rho_{k-1}^\dagger\right)^{-1}$
        $\alpha_k^{(s)} \leftarrow \beta_k^{(s)}\alpha_k\rho_{k-1}\alpha_{k-1}^{-1}\alpha_{k-1}^{(s)}$
        $X_k^{(s)} \leftarrow X_{k-1}^{(s)} + P_{k-1}^{(s)}\alpha_k^{(s)}$
        $P_k^{(s)} \leftarrow Q_k + P_{k-1}^{(s)}\beta_k^{(s)}\rho_k^\dagger$
    end for
end for
```

- SBCGrQ [Futamura et. al. 2012]:
  - n_{pf} RHS vectors
  - N_{shifts} shifts
- BCGrQ [Dubrulle 2008]:
  - $N_{\mathrm{shifts}} = 1$
- SCG [Jegerlehner 1996]:
  - $n_{\mathrm{pf}} = 1$
- CG:
  - $n_{\mathrm{pf}} = N_{\mathrm{shifts}} = 1$

Block Solvers: convergence of shifted solutions

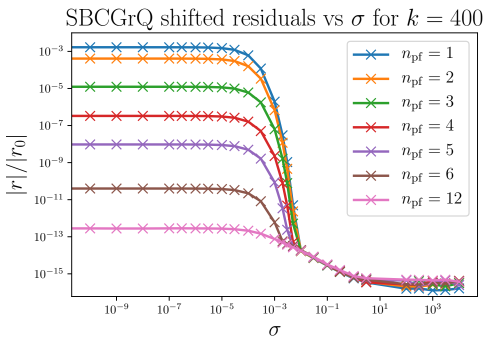
For the shifted matrix $A+\sigma$, for large shifts $\sigma \gg \lambda_{n_{\rm pf}}$, the bound on the error becomes

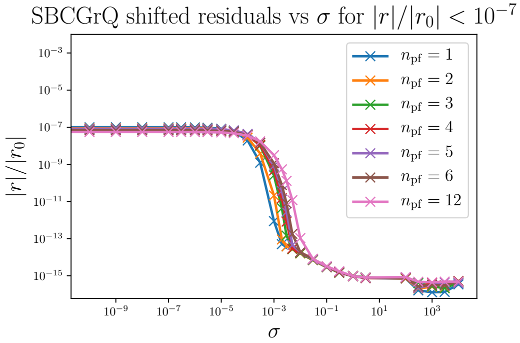
$$\begin{aligned}\frac{|e_k|_{A+\sigma}}{|e_0|_{A+\sigma}} &\lesssim c_1(n_{\rm pf})e^{-4k\sqrt{(\sigma+\lambda_{n_{\rm pf}})/(\sigma+\lambda_{\rm max})}} \\ &\lesssim c_1(n_{\rm pf})e^{-4k\sqrt{\sigma/(\sigma+\lambda_{\rm max})}\left[1+\mathcal{O}(\lambda_{n_{\rm pf}}/\sigma)\right]}. \end{aligned} \tag{19}$$

To leading order the convergence rate does not depend on $\lambda_{n_{\rm pf}}$, but only on

- σ: the size of the shift
- k: the number of solver iterations

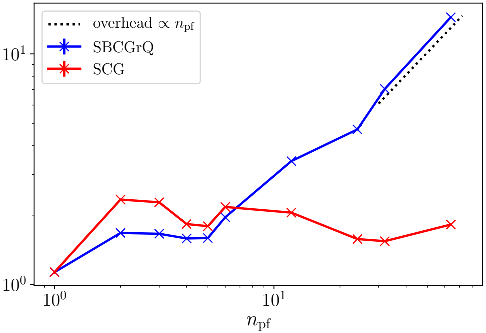
Block Solvers: multishift convergence





Block Solvers: multishift overhead

Solver runtime per Dirac Op call vs n_{pf}

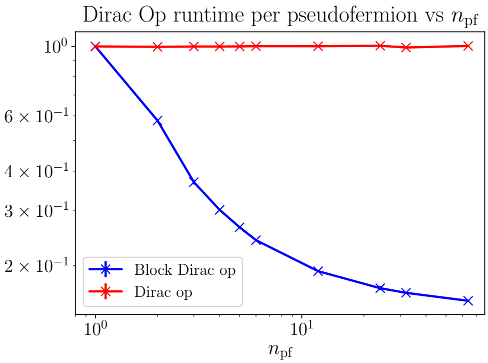


Multishift block overhead per pseudofermion grows $\propto n_{\mathrm{pf}}$ compared to non–block multishift:

- $1 \times \mathcal{O}(V)$ Dirac operator
- $N_{\mathrm{shifts}} \times \mathcal{O}(V n_{\mathrm{pf}})$ Vector multiply–adds
- $N_{\mathrm{shifts}} \times \mathcal{O}(n_{\mathrm{pf}}^2)$ Dense matrix ops

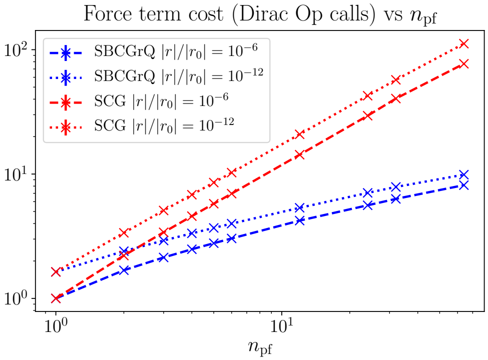
But this overhead is not prohibitive for region of interest ($n_{\mathrm{pf}} \lesssim 6$)

Block Fields: faster Dirac operator

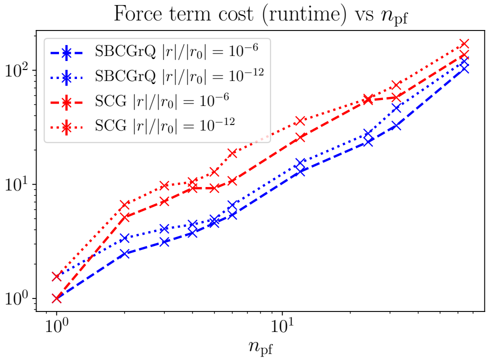


- Applying the Dirac operator to a block of pseudofermion vectors at each site is more computationally efficient:
- Higher computational intensity (flops/bytes): only need to load gauge links once per n_{pf} vectors.
- Contiguous vectors within block allow better use of cache.

Force term cost

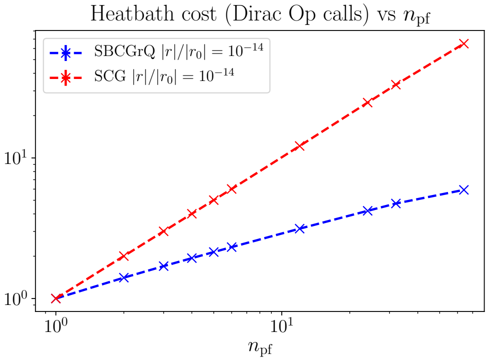


Force term cost (Dirac Op calls) vs n_{pf}

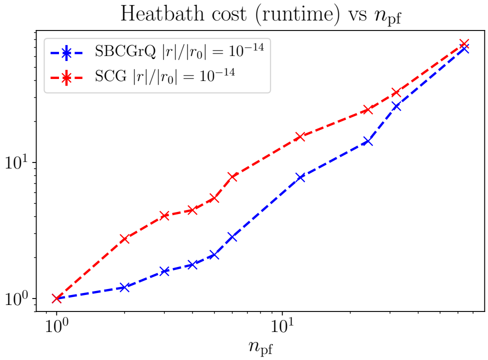


Force term cost (runtime) vs n_{pf}

Heat bath term cost

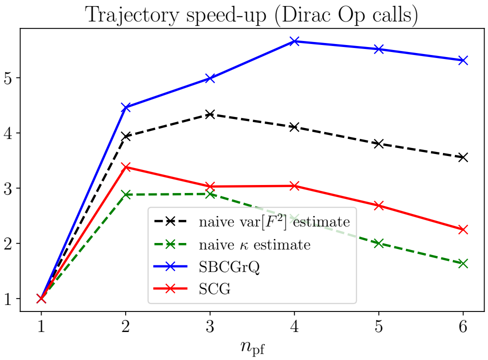


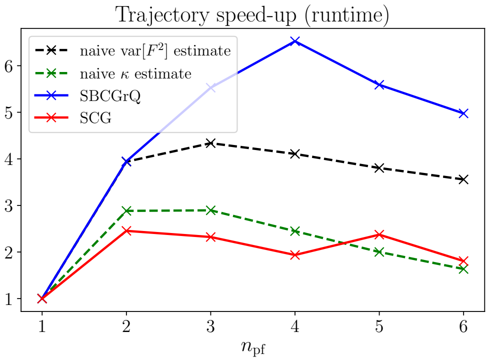
Heatbath cost (Dirac Op calls) vs n_{pf}



Heatbath cost (runtime) vs n_{pf}

Multiple Pseudofermions + Block Solver Speed-up


Trajectory speed-up (Dirac Op calls)


Trajectory speed-up (runtime)

Conclusion

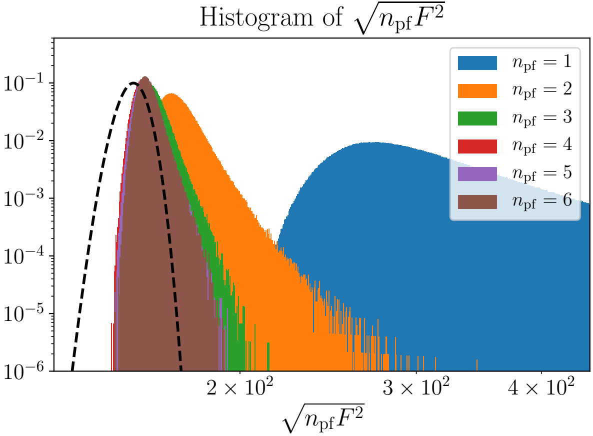
RHMC with multiple pseudofermions and block solvers has three cumulative advantages over plain RHMC:

- Multiple n_{pf}: **larger** integrator step size ⇒ fewer inversions
- Block CG: **fewer** Dirac operator calls per inversion
- Block vectors: **faster** Dirac operator calls

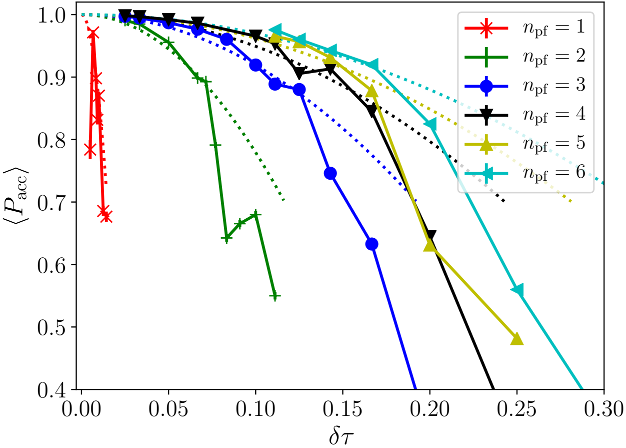
Additionally, multiple pseudofermions may make it easier to cross high energy barriers (smaller force, larger step size) and help to decorrelate topological charge.

Next step: implement in production code, see how it scales with volume, compare to other methods (mass–preconditioning, deflation, multigrid, etc.)

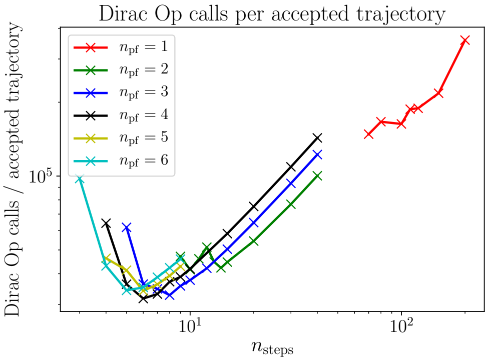
Force histogram scaling

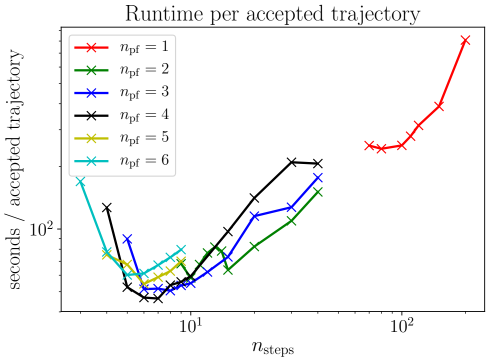


Simulation acceptance vs prediction



Simulation cost vs number of integrator steps





Long Numerical Simulations

n_{pf}	n_{steps}	$\langle P_{\text{acc}} \rangle$	$\langle e^{-\Delta H} \rangle$	$\langle \text{plaq} \rangle$	τ_{int}	$n_{\text{trajectories}}$
1	250	0.961(11)	0.9701(100)	0.52268(14)	5	5×10^3
2	16	0.942(5)	0.9920(28)	0.52283(6)	4	28×10^3
3	11	0.965(1)	0.9998(6)	0.52288(8)	5	33×10^3
4	9	0.966(1)	1.0005(5)	0.52297(6)	4	26×10^3
5	8	0.960(1)	0.9994(7)	0.52272(8)	5	25×10^3
6	7	0.954(2)	1.0006(8)	0.52277(10)	6	21×10^3

Table: Run parameters for the longer simulations, with n_{steps} tuned such that $\langle P_{\text{acc}} \rangle \simeq 0.96$. The integrated autocorrelation time of the plaquette does not appear to depend on n_{pf}.