RHMC with Block Solvers and Multiple Pseudofermions

Liam Keegan, Philippe de Forcrand
keeganl@phys.ethz.ch, forcrand@phys.ethz.ch

Introduction

The main difficulty in lattice simulations of QCD is calculating the determinant of the Dirac operator, a very large and badly conditioned matrix.

In the RHMC this is stochastically estimated by inverting the matrix acting on a bosonic field of "pseudofermions" using a Krylov solver.

An old idea to improve this is to use **multiple pseudofermion fields**, which results in a smaller force term [1], but requires the inversion of multiple pseudofermion vectors.

Recently there has been renewed interest in another old idea: **block Krylov solvers** [2, 3], which invert the same matrix on multiple vectors simultaneously, and converge with fewer iterations than are required to solve each vector separately.

Here we combine these two ideas to speed up the RHMC algorithm.

Multiple Pseudofermions

Starting from the simple observation

$$\det M = [\det M^{1/n_{\mathrm{pf}}}]^{n_{\mathrm{pf}}}, \qquad (1)$$

one can construct an RHMC using n_{pf} pseudofermions, where for a given gauge field the resulting pseudofermion force term has an expectation value that is independent of n_{pf},

$$\overline{F^a_{x\mu}(n_{\mathrm{pf}})} = \mathrm{Tr}\left[M\frac{\partial M}{\partial U^a_{x\mu}}\right] \equiv \mathrm{Tr}[X], \qquad (2)$$

and a variance that is suppressed by n_{pf},

$$\left[\overline{F^a_{x\mu}(n_{\mathrm{pf}})^2}\right] - \left[\overline{F^a_{x\mu}(n_{\mathrm{pf}})}\right]^2 = \tfrac{2}{n_{\mathrm{pf}}}\mathrm{Tr}[X^2]. \qquad (3)$$

This variance dominates the rms force – see Fig.1 – which determines how large the step size can be in the integrator, so increasing n_{pf} allows a **larger step size** to be used.

Block Solvers

A Krylov solver iteratively solves $Ax = b$ for x given some b. Starting from some initial residual r, it constructs a solution $x^{(i)}$ after i iterations from the Krylov basis $\mathcal{K}_i = \{r, Ar, A^2r, \ldots, A^{i-1}r\}$.

If we want to solve for n_{pf} vectors b_j, we can form a block matrix B where the j-th column is b_j, and solve $AX = B$. The solution is now constructed from the much larger block-Krylov basis $\mathcal{K}_i = \{R, AR, A^2R, \ldots, A^{i-1}R\}$, potentially with significantly **fewer iterations** – see Fig.3.

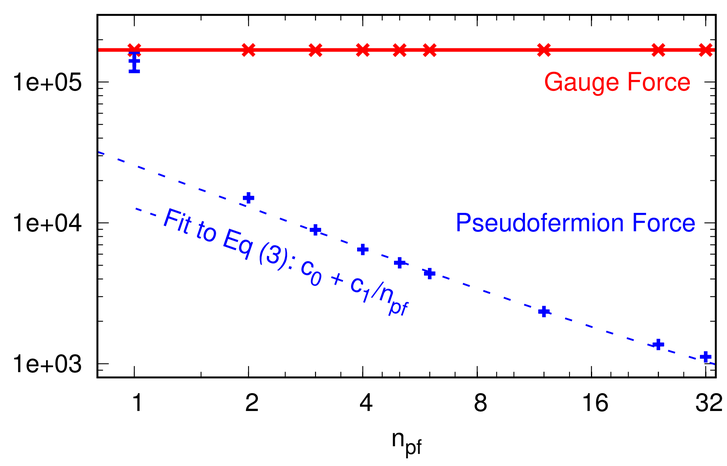
One complication is numerical stability, in particular if the matrix of residuals becomes badly conditioned the block algorithm can break down, but this issue was resolved by adding a QR decomposition of this residual matrix at each iteration [4]. Here we use a **multi–shift variant of block CG** with this QR decomposition of the residuals matrix [5].

Force Reduction

Increasing n_{pf} reduces both the size and the variance of the pseudofermion force term. The plot below shows the squared norm of the pseudofermion and gauge force terms versus n_{pf}, the former decreases by two orders of magnitude as n_{pf} is increased, allowing a substantial increase in the molecular dynamics step size.

Also shown is a fit to the form predicted by Eq. (3), which should be valid for large enough n_{pf}, and seems to provide a reasonable fit to the data for $n_{\mathrm{pf}} \gtrsim 2$.

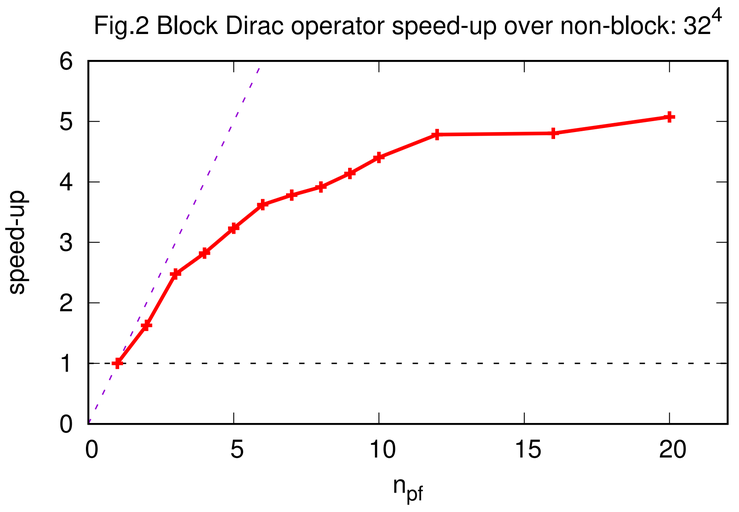
Fig.1 $|F|^2$ vs n_{pf}: 8^4, n_f=4, beta=5.12, m=0.002



Block Speed-up

Applying the Dirac operator to a block of pseudofermion fields amortises the cost of loading the gauge links from memory, and additionally provides the benefit of cache locality for the blocked pseudofermion fields, resulting in a significant (up to $\sim 5\times$) speed-up.

Fig.2 Block Dirac operator speed-up over non-block: 32^4



References

[1] M. A. Clark and A. D. Kennedy. Accelerating dynamical fermion computations using the rational hybrid Monte Carlo (RHMC) algorithm with multiple pseudofermion fields. *Phys. Rev. Lett.*, 98:051601, 2007, [hep-lat/0608015].

[2] H. Tadano, Y. Kuramashi, and T. Sakurai. Application of preconditioned block BiCGGR to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD. *Comput. Phys. Commun.*, 181:883, 2010, arXiv:0907.3261 [hep-lat].

[3] M. A. Clark, Alexei Strelchenko, Alejandro Vaquero, Mathias Wagner, and Evan Weinberg. Pushing Memory Bandwidth Limitations Through Efficient Implementations of Block-Krylov Space Solvers on GPUs. 2017, arXiv:1710.09745 [hep-lat].

[4] A. A. Dubrulle. Retooling the method of block conjugate gradients. *Electronic Trans. on Num. Anal*, pages 216–233, 2001, http://eudml.org/doc/121814.

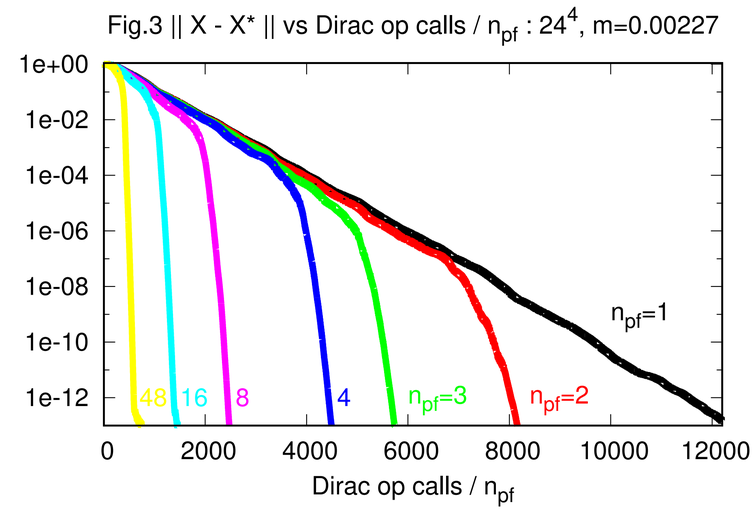
[5] Yasunori Futamura, Tetsuya Sakurai, Shinnosuke Furuya, and Jun-Ichi Iwata. Efficient algorithm for linear systems arising in solutions of eigenproblems and its application to electronic-structure calculations. In *High Performance Computing for Computational Science - VECPAR 2012*, pages 226–235, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Block Work Reduction

The block solver requires fewer iterations per solution to converge as n_{pf} is increased. The figure below shows the norm of the error of the solution versus the number of Dirac operator calls per righthand side.

In the example below, using a normal CG solver for $n_{\mathrm{pf}} = 8$ would require $\sim 8\times$ more iterations than $n_{\mathrm{pf}} = 1$, while with the block solver only $\sim 1.5\times$ more iterations are required.

Fig.3 || X - X* || vs Dirac op calls / n_{pf} : 24^4, m=0.00227



Block Overhead

The multishift block solver promotes all vector multiply–add operations to matrix multiply–add operations, leading the computational cost of these operations to scale with an additional factor $\mathcal{O}(n_{\mathrm{pf}})$ compared to the usual multi–shift CG. However it may be possible to overlap these operations with other communications to mitigate their cost.

Another issue is that all vectors need to be stored simultaneously for the block solver, which results in an extra factor n_{pf} in the memory required by the algorithm compared to the usual multi–shift CG.

For sufficiently large n_{pf} these costs can outweigh the gain from both the speed-up of the block Dirac operator and the reduction in the required number of Dirac operator calls.

Conclusions

RHMC with multiple pseudofermions and block solvers has several attractive features:

- More pseudofermions result in a smaller rms fermion force, allowing a **larger integrator step size** to be used.
- **Fewer matrix–vector** operations are required to invert the Dirac operator for each of these steps.
- The block Dirac operator has a **higher flop rate** due to caching of the gauge links and cache locality of the pseudofermion fields.
- Possibly **faster Monte Carlo dynamics** – under study.