PROCEEDINGS OF SCIENCE

Algorithmic improvements for weak coupling simulations of domain wall fermions

Greg McGlynn[*]
Columbia University
E-mail: gem2128@columbia.edu

We discuss algorithmic improvements being used in the evolution of new 2+1+1 flavor lattices by the RBC/UKQCD collaboration. A technique similar to Hasenbusch mass splitting and suggested previously by Brower, Neff, and Orginos allows the light quark action to be split into multiple parts with different values of L_s, the size of the fifth dimension. This allows us to reduce chiral symmetry breaking by making L_s large while most inversions are actually done at a cheaper, smaller value of L_s.

The 33rd International Symposium on Lattice Field Theory
14 -18 July 2015
Kobe International Conference Center, Kobe, Japan

[*] Speaker.

PoS(LATTICE 2015)019

1. Introduction

Domain wall fermions have good chiral symmetry properties but are expensive to simulate. Domain wall fermions introduce an unphysical fifth dimension with size L_s. Larger values of L_s give smaller chiral symmetry breaking but the cost of simulating the quark determinant is proportional to L_s. In this talk we discuss two attempts to achieve the good chiral symmetry properties of a large-L_s domain wall simulation without paying the full cost. We focus on speeding up the hybrid Monte Carlo (HMC) evolution.

2. HDCG in HMC evolution

The first technique we discuss is deflation of the light quark solves in the evolution. Most of the computational effort in the evolution is spent on inverting the light quark Dirac operator. A standard technique for speeding up such inversions is deflation. Deflation removes the lowest eigenmodes of the Dirac operator from the problem, speeding up the inversions by reducing the condition number. To apply deflation we have to spend some time actually calculating these low modes. The cost of this calculation can easily be amortized if we are doing many inversions on a single gauge field configuration. However in an evolution we only do a single inversion on each configuration. It may therefore be difficult to amortize the setup cost. We can try to keep old low modes around and use them to deflate solves on new, changed gauge configurations, but the low modes will presumably grow gradually “stale” as the gauge configuration changes.

Deflation has been applied to HMC evolution with Wilson fermions in [1]. We attempted to apply deflation to the evolution of a 2+1+1 flavor domain wall ensemble on a $32^3 \times 64$ lattice with $L_s = 24$ and $1/a \approx 3$ GeV. We used Peter Boyle’s hierarchically deflated conjugate gradient (HDCG) algorithm, a multigrid-like deflation technique designed for domain wall fermions [2].

As with all deflation methods, there is a tradeoff between the setup time and the solve time. Spending more time on setup increases the number and quality of low modes calculated and therefore decreases the time for later solves. One must experiment to determine the optimal amount of time to spend on setup. A useful technique is that the setup does not need to be re-done from scratch each time; rather, we can start from the previous low-mode subspace and “refine” the old low modes to get low modes on the new, changed gauge configuration. This makes setup faster and so allows us to run it more often.

In our case, we were unable to achieve a speedup by including HDCG in the evolution. While it was possible to tune the algorithm parameters so that the deflated evolution broke even with the original evolution, we did not achieve a speedup. There were two reasons for this. One is that it was not possible to amortize the setup time over enough solves. As Figure 1 shows, the low modes grow rapidly stale, so that after only 0.1 MD time units deflated solves are no faster than undeflated solves. Given our time step, this meant that we had only about 4 solves over which to amortize the setup cost. The second problem was that the HDCG algorithm requires us to calculate the projection of the Dirac operator to the low mode subspace every time the gauge field changes. This is a nontrivial cost which must be paid before each inversion.

It is possible that future work will overcome these difficulties. One concern that we did not investigate is that the use of deflation presumably makes the evolution non-reversible unless the

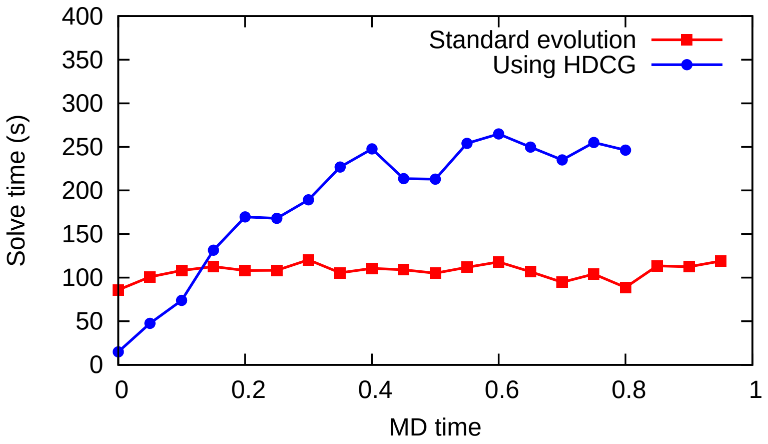

Figure 1: Times for light quark inversions during an HMC trajectory. In the HDCG evolution, the low mode subspace is computed at time zero and these low modes are used throughout the trajectory. Solve time increases as the gauge field changes and these low modes grow "stale." Deflation only helps for the first 0.1 units of MD time. Therefore the setup must actually be rerun every 0.1 MD time units.

Dirac operator inversions are done to high precision. Since reversibility is important to the correctness of the simulation, this may be another obstacle to using deflation. Reversibility can be regained by tightening the stopping conditions for the solves, but this will increase the solve time, which might wipe out the gains of deflation.

3. L_s-splitting of light quark determinant

We now turn to another technique. In [3] it is suggested that domain wall evolutions may be sped up by splitting the light quark determinant into factors with different L_s.

This technique relies on the domain wall-overlap correspondence (see [3]). Consider a domain wall Dirac operator $D_{DW}(m,L_s)$ with quark mass m and fifth dimension extent L_s. This operator acts on five-dimensional fermion fields. Corresponding to each such operator there is a four-dimensional "effective overlap operator" $D_{ov}(L_s)$ such that

$$\det\left[\frac{D_{DW}(m,Ls)}{D_{DW}(1,L_s)}\right]_{5D} = \det[D_{ov}(L_s)]_{4D} \tag{3.1}$$

In the $L_s \to \infty$ limit, $D_{ov}(L_s)$ becomes the true overlap operator [4]. D_{ov} is easily implemented explicitly.

This four-dimensional operator lets us manipulate the domain wall fermion determinant as follows. Pick a new L_s' smaller than the original L_s and write

$$\begin{aligned}
\det\left[\frac{D_{DW}(m,L_s)}{D_{DW}(1,L_s)}\right]_{5D} &= \det[D_{ov}(L_s)]_{4D} \\
&= \det[D_{ov}(L'_s)]_{4D} \times \det\left[\frac{D_{ov}(L_s)}{D_{ov}(L'_s)}\right]_{4D} \\
&= \det\left[\frac{D_{DW}(m,L'_s)}{D_{DW}(1,L'_s)}\right]_{5D} \times \det\left[\frac{D_{ov}(L_s)}{D_{ov}(L'_s)}\right]_{4D}
\end{aligned} \tag{3.2}$$

The first term of the final expression is a regular 5D domain wall determinant at a smaller, and thus cheaper, L'_s. The second term is a 4D correction determinant which accounts for the different between L'_s and the original L_s.

If $D_{ov}(L'_s) \approx D_{ov}(L_s)$, then the second determinant is always close to 1. The molecular dynamics forces corresponding to such a determinant are small and the correction determinant can be simulated on a large time step and therefore at relatively low cost. If the cost of the correction determinant is low enough then the above splitting of the original determinant gives a net speedup.

It is important to realize that we have not changed the fermion action, only our strategy for simulating it. To carry out this strategy we need to find a $D_{ov}(L'_s)$ with $L'_s < L_s$ such that the reduced-L_s effective overlap operator closely approximates the original effective overlap operator. In some sense this is true for any possible $D_{ov}(L'_s)$. After all, any effective overlap operator is an approximation to the true overlap operator, so any two effective overlap operators should be fairly good approximations to each other.

4. zMöbius

It is possible to do better, however: we can tune the reduced-L_s operator to more closely match the original operator. To see how to do this we recall that the effective overlap operator for domain wall fermions can be written

$$D_{ov} = \frac{1}{2}[1+m+(1-m)\varepsilon(H)] \tag{4.1}$$

where H, the "Shamir kernel", is defined in terms of the Wilson Dirac operator D_W with a large negative mass $-M_5$:

$$H = \frac{\gamma_5 D_W(-M_5)}{2+D_W(-M_5)} \tag{4.2}$$

Above, $\varepsilon(x)$ is an approximation to the sign function $|x|/x$. As D_{ov} converges to the true overlap operator, the $\varepsilon(x)$ converges to the true sign function. From this we can see that matching two effective overlap operators means matching their sign functions.

For "plain" domain wall fermions the sign function approximation is

$$\varepsilon(x) = \frac{(1+x)^{L_s}-(1-x)^{L_s}}{(1+x)^{L_s}+(1-x)^{L_s}} \tag{4.3}$$

For the Möbius generalization of domain wall fermions, the sign function becomes

$$\varepsilon(x) = \frac{f(x) - f(-x)}{f(x) + f(-x)} \tag{4.4}$$

where

$$f(x) = \prod_{i=1}^{L_s} (\omega_i + x) \tag{4.5}$$

The ω_i are complex numbers that can be chosen arbitrarily by adjusting coefficients in the definition of the Möbius domain wall fermion operator.

We can use the Remez algorithm to tune the ω_i to optimally approximate any original sign function. That is, given a domain wall operator with some L_s with sign function $\varepsilon^{(L_s)}(x)$, we can find a reduced-L_s Möbius operator with sign function $\varepsilon^{(L_s')}(x)$ such that

$$\varepsilon^{(L_s')}(x) \approx \varepsilon^{(L_s)}(x) \tag{4.6}$$

This is the “zMöbius” technique; it was developed in [5].[1]

This approximation can often be made quite precise. In Figure 2 we show the absolute difference between an $L_s = 24$ sign function and and $L_s' = 12$ sign function tuned to approximate it. The difference is everywhere less than 10^{-5}. If desired the accuracy of the approximation can be concentrated in the region of highest eigenvalue density for the kernel operator H. We show an example of this in Figure 3.

Given that we can approximate a large-L_s sign function $\varepsilon^{(L_s)}(x)$ with a reduced-L_s sign function $\varepsilon^{(L_s')}(x)$, might it be better to just simulate at the reduced L_s directly? Our strategy has several advantages. One is that we can adjust the reduced-L_s sign function according to the observed eigenvalue distribution of the kernel without actually changing the true action. Another advantage is that if we wish to simulate several ensembles along a scaling trajectory, we can use the same overall action on each while adjusting the underlying reduced-L_s operator according to the kernel eigenvalue distribution at each a. Finally, this technique allows us to use a simple true action which is easily specified by a few parameters, while underneath we accelerate the simulation using a complicated action with many adjustable parameters (L_s' and the $\{\omega_i\}$).

5. Simulation strategy and results

The L_s-splitting strategy described above splits the fermion determinant into a regular 5D domain wall determinant at reduced L_s and a 4D correction determinant. When we apply this strategy to the two-flavor light quark determinant the two-flavor correction determinant is

$$\det \left[\frac{D_{ov}(L_s)^\dagger D_{ov}(L_s)}{D_{ov}(L_s')^\dagger D_{ov}(L_s')} \right]_{4D} \propto \int D\phi^\dagger D\phi \exp\left(-\phi^\dagger \left[\frac{D_{ov}(L_s)^\dagger D_{ov}(L_s)}{D_{ov}(L_s')^\dagger D_{ov}(L_s')} \right]^{-1} \phi \right) \tag{5.1}$$

The above equation shows how we simulate the correction determinant using a single 4D pseudofermion field ϕ.

[1] The “z” indicates the fact that the optimal ω_i maybe actually be complex.

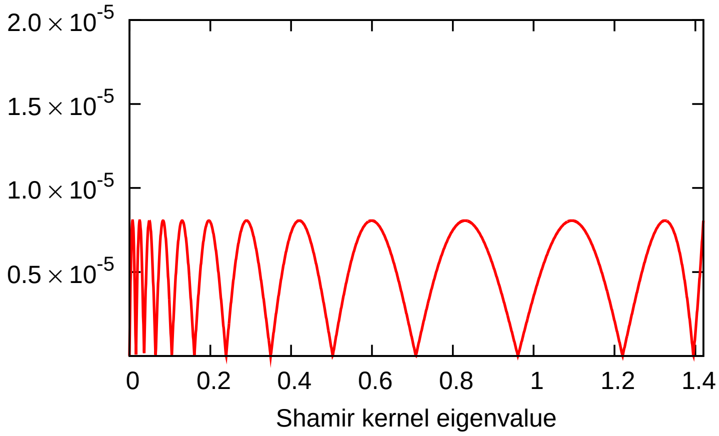

Figure 2: The absolute difference $|\varepsilon^{(L_s=24)}(x)-\varepsilon^{(L_s'=12)}(x)|$ between two approximate sign functions. The $L_s'=12$ sign function has been tuned using the Remez algorithm to approximate the $L_s=24$ sign function as closely as possible. Quite good accuracy is achieved: the absolute difference is everywhere less than 10^{-5}. The range of x values plotted corresponds to the range of eigenvalues of the Shamir kernel.

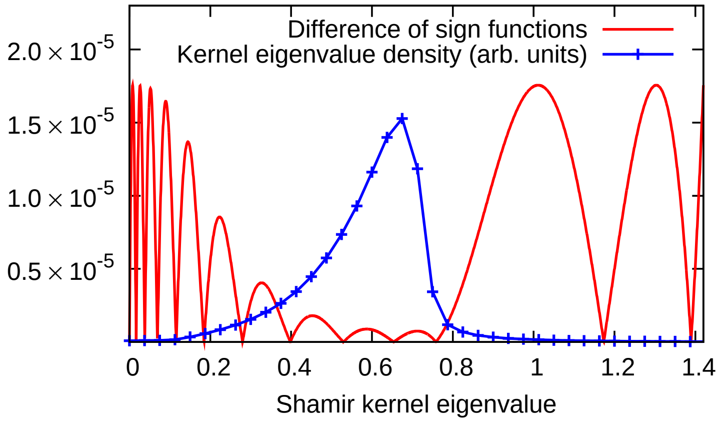

Figure 3: The absolute difference $|\varepsilon^{(L_s=24)}(x)-\varepsilon^{(L_s'=12)}(x)|$ between two approximate sign functions. Here we adjust the Remez algorithm to concentrate the accuracy of the approximation on the regions of high eigenvalue density of the Shamir kernel. In addition to the difference of sign functions we plot the eigenvalue density of the Shamir kernel. Compared to Figure 2, the sign function difference has been made much smaller (order 10^{-6}) in the region of highest eigenvalue density, at the cost of slightly larger errors in regions of lower eigenvalue density.

The zMöbius approximation to the original sign function is extremely good, meaning that the molecular dynamics forces coming from this determinant are tiny. In fact we can get away with neglecting this force completely and only including the correction determinant in the accept/reject step of the HMC. This is feasible because the typical variation in the correction determinant pseudfermion action over the course of a trajectory is typically much smaller than one, even on large lattice volumes.

We have observed a 30% reduction in the cost of the light quark determinant by using this strategy with $L_s=24$ and $L_s'=12$ on a $32^3\times 64$ 2+1+1 flavor domain wall ensemble with $1/a=3$

GeV. Naively we might expect a 50% cost reduction because L_s is being decreased by 50%. However we observe that the iteration count of the conjugate gradient inversions increases somewhat, and we end up with only a 30% cost reduction. In fact to avoid an even larger increase in the CG iteration count we need to adjust the preconditioning scheme used in these inversions (see [5]).

We are now using this strategy for the light quark determinant in a large-scale RBC/UKQCD physical point 2+1+1 flavor domain wall ensemble with $1/a \approx 3$ GeV and lattice size $80 \times 80 \times 96 \times 192$. In this run we use $L_s = 32$ and $L_s' = 14$.[2] Even on this large volume, our strategy of neglecting the force from the correction determinant works well. The fluctuations of the correction pseudofermion action along a single trajectory are of order 3×10^{-2} which is much less than 1. This means that neglecting the molecular dynamics force of the correction determinant is not significantly affecting the acceptance rate.

6. Conclusion

We have described two attempts to speed up HMC evolutions of domain wall fermion ensembles. The first attempt, deflation in the evolution using the HDCG algorithm, did not succeed because the low modes we computed went “stale” too quickly as the gauge field evolved. The second attempt, L_s-splitting of the light quark determinant along the lines suggested by Brower et al. in [3], did produce a 30% reduction in the cost of the light quark determinant and is now in production.

References

[1] M. Lüscher, “Deflation acceleration of lattice QCD simulations,” JHEP **0712**, 011 (2007) [arXiv:0710.5417 [hep-lat]].

[2] P. A. Boyle, “Hierarchically deflated conjugate gradient,” arXiv:1402.2585 [hep-lat].

[3] R. C. Brower, H. Neff and K. Orginos, “The Móbius Domain Wall Fermion Algorithm,” arXiv:1206.5214 [hep-lat].

[4] H. Neuberger, “Exactly massless quarks on the lattice,” Phys. Lett. B **417**, 141 (1998) [hep-lat/9707022].

[5] T. Izubuchi, C. Jung, S. Syritsyn and T. Blum, “zMobius and other recent developments on Domain Wall Fermions,” these proceedings.

[2] L_s is large here because we have included a “dislocation-enhancing determinant” to promote topology change. Dislocations increase the residual chiral symmetry breaking of domain wall fermions and we use large L_s to compensate.