PROCEEDINGS OF SCIENCE

Algorithms for Domain Wall Fermions

Peter Boyle,[a,b,*] **Christopher Kelly**[c] **and Azusa Yamaguchi**[b]

[a] *Physics Department, Brookhaven National Laboratory, Upton, NY, USA*

[b] *School of Physics and Astronomy,*
University of Edinburgh, Edinburgh UK

E-mail: pboyle@bnl.gov, ckelly@bnl.gov, ayamaguc@staffmail.ed.ac.uk

We discuss algorithms for domain wall fermions focussing on accelerating Hybrid Monte Carlo sampling of gauge configurations. Firstly a new multigrid algorithm for domain wall solvers and secondly a Domain decomposed hybrid monte carlo approach applied to large subvolumes and optimised for GPU accelerated nodes.

The 38th International Symposium on Lattice Field Theory, LATTICE2021 26th-30th July, 2021
Zoom/Gather@Massachusetts Institute of Technology

[*] Speaker

1. Introduction

Domain wall fermions are a theoretically attractive but numerically expensive formulation of lattice QCD. In recent years valence analysis has become a largely solved problem with multiple propagator inversions accelerated by deflation [1–3] and able to run independently in a large system using MPI split communicators [4]. Current calculations use globally stored eigenvector deflation and solve thousands of (deflated) propagators independently in parallel limiting communication overhead

In contrast, gauge configuration sampling is serially dependent and a strong scaling problem. A single solution of the linear system is required for each quark mass and the serial dependence of these requires scaling a single problem to as many nodes as possible.

Some recent systems such as those installed at Juelich and Edinburgh are well balanced with communication and computation taking the same amount of time and delivering up to 10TF/s per node on multiple node jobs with a volume per GPU in the region of 32^4.

This balance will not be preserved in future systems, however. GPU performance is likely to increase by as much as ten fold, while network performance may only increase by 50% in the next few years. In order to accelerate HMC evolution new algorithms will be required and in this proceedings we document our plan to address this.

We plan to use a modified formulation of Domain Decomposed Hybrid Monte Carlo (DDHMC) and to combine this with multigrid deflation of the local solves. The difference will be to focus on *large* subdomains appropriate to the entire volume processed by a large, multi-GPU computing node.

2. Large volume domain decomposition

We follow the DDHMC algorithm[5–7] but aim to implement this rather differently.

Large blocks will be used, of $O(32^4)$ and no attempt will be made to keep the Dirichlet boundary condition block operators well conditioned. Rather we aim to implement a communication avoiding algorithm for multi-GPU nodes, rather than to precondition the HMC algorithm. Prior implementations[6–9] on conventional multi-core microprocessors have found substantial computational acceleration from smaller, cache resident cells processed by individual processor cores, reaching very high performance on the Fugaku computer in particular [10, 11].

As discussed by the original author of DDHMC, the locality factorisation of the determinant must eventually win in computing systems with penalties for non-locality. However, in highly parallel hardware optimising for cache locality is not possible as opportunities for sequential ordering of access is limited.

This proposed usage will keep the fraction of active links in the HMC at close to 100%, while the force for the boundary forces will be suppressed by the width of the bands of inactive links. This may decorrelate better than the original DDHMC. The speed gain will come solely from avoiding communication slow down, and will reflect the nature and cost of computing and data access in modern supercomputers.

The large cell gives the opportunity for larger bands of inactive links than the original implementation, and peak forces may be suppressed by the distance of propagation.

2.1 Cuboidal domains and the Dirac operator

We partition the lattice into two domains Ω and $\bar\Omega$. Their *exterior* boundaries haloes are ∂_Ω and $\partial_{\bar\Omega}$ such that,

$$\partial_\Omega \cap \Omega = \emptyset,$$

and

$$\partial_{\bar\Omega} \cap \bar\Omega = \emptyset,$$

respectively.

The Dirac operator, with an appropriate non-lexicographic ordering may then be written as

$$D = \begin{pmatrix} D_\Omega & D_\partial \\ D_{\bar\partial} & D_{\bar\Omega} \end{pmatrix},$$

where D_∂ are terms in the matrix that connect the exterior boundary ∂_Ω to Ω, and D_Ω are terms in the matrix that connect Ω with itself, and similar for the other terms. We write short hand,

$$\hat D_\Omega = \begin{pmatrix} D_\Omega & 0 \\ 0 & 0 \end{pmatrix} \quad \hat D_\partial = \begin{pmatrix} 0 & D_\partial \\ 0 & 0 \end{pmatrix}$$
$$\hat D_{\bar\partial} = \begin{pmatrix} 0 & 0 \\ D_{\bar\partial} & 0 \end{pmatrix} \quad \hat D_{\bar\Omega} = \begin{pmatrix} 0 & 0 \\ 0 & D_{\bar\Omega} \end{pmatrix}$$

The Dirac operator may then be Schur factored as:

$$\begin{pmatrix} D_\Omega & D_\partial \\ D_{\bar\partial} & D_{\bar\Omega} \end{pmatrix} = \begin{pmatrix} 1 & D_\partial D_{\bar\Omega}^{-1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} D_\Omega - D_\partial D_{\bar\Omega}^{-1} D_{\bar\partial} & 0 \\ 0 & D_{\bar\Omega} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ D_{\bar\Omega}^{-1} D_{\bar\partial} & 1 \end{pmatrix}. \tag{1}$$

We may then write the Fermion determinant as,

$$\det D = \det D_\Omega \det D_{\bar\Omega} \det \left\{ 1 - D_\Omega^{-1} D_\partial D_{\bar\Omega}^{-1} D_{\bar\partial} \right\},$$

where we identify

$$\chi = 1 - D_\Omega^{-1} D_\partial D_{\bar\Omega}^{-1} D_{\bar\partial}$$

Following Luscher, we introduce projectors $\hat{\mathbb{P}}_{\bar\partial}$ with both spinor and space structure projecting all spinor elements in Ω connected by $D_{\bar\partial}$ to $\bar\Omega$, and similarly $\hat{\mathbb{P}}_\partial$.

The matrix $\hat D_{\bar\partial}$ acts only non-trivially on this subset of spinor components fields in $\partial_{\bar\Omega}$,

$$\hat D_{\bar\partial} \hat{\mathbb{P}}_{\bar\partial} = \begin{pmatrix} 0 & 0 \\ D_{\bar\partial} & 0 \end{pmatrix} \begin{pmatrix} \mathbb{P}_{\bar\partial} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ D_{\bar\partial} & 0 \end{pmatrix},$$

and so we introduce the matrix,

$$\hat R = \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix} = \hat{\mathbb{P}}_{\bar\partial} - \hat{\mathbb{P}}_{\bar\partial} \hat D_\Omega^{-1} \hat D_\partial \hat D_{\bar\Omega}^{-1} \hat D_{\bar\partial} = \hat{\mathbb{P}}_{\bar\partial} - \begin{pmatrix} \mathbb{P}_{\bar\partial} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} D_\Omega^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & D_\partial \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & D_{\bar\Omega}^{-1} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ D_{\bar\partial} & 0 \end{pmatrix}.$$

Since in the right basis χ takes the form

$$\chi = \begin{pmatrix} 1 - X & 0 \\ Y & 1 \end{pmatrix}$$

we see that,

$$\det \chi = \det R = \det(1 - X).$$

We may therefore treat the determinant of χ via a usual pseudofermion integral *only* over those fields in the space projected by $\mathbb{P}_{\bar\partial}$, and call these fields $\phi_{\bar\partial}$. When R is taken as matrix *from this subspace to itself*, it is non-singular with an inverse we can compute:

$$\hat{R}^{-1} = \hat{\mathbb{P}}_{\bar\partial} - \hat{\mathbb{P}}_{\bar\partial} D^{-1} \hat{D}_{\bar\partial} = \begin{pmatrix} \mathbb{P}_{\bar\partial} & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} \mathbb{P}_{\bar\partial} & 0 \\ 0 & 0 \end{pmatrix} D^{-1} \begin{pmatrix} 0 & 0 \\ D_{\bar\partial} & 0 \end{pmatrix}.$$

This is most easily seen by inserting UDL decomposition of D^{-1}, showing that $\hat{R}\hat{R}^{-1} = \hat{\mathbb{P}}_{\bar\partial}$.

2.2 Domain shapes

The original algorithm selected Ω and $\bar\Omega$ to be sites located on even and odd cells with a hypercuboidal decomposition, figure 1. We seek to make the domains as large as possible. However, with a symmetrical block size for $Omega$ and $\bar\Omega$, one cannot increase the size of a domain to be close to an entire computing node. If the sub-domain were sized to a single computing node, then sequences involving only $\hat{D}_{\Omega}^{-1}$, for example, would not load balance on the machine and leave half the nodes idle.

Therefore to maximise the number of active links we need a different domain decomposition scheme. Communication avoidance suggests to make regions inactive only in the neighbourhood of the boundaries between nodes, with the size of the inactive region controlled to suppress molecular dynamics forces by however much is required to maintain a good suppression of communication. This encouraged us to adopt a non-standard domain pattern, figure 2, where Ω is the interior (non-boundary) cells of each node, while $\bar\Omega$ is union of all lattice sites on the surfaces of all computing nodes. The entire domain $\bar\Omega$ is left inactive in the HMC evolution, along with a controllable subset of links within Ω. This allows minimises the forces in Ω and makes evolution of $\bar\Omega$ irrelevant.

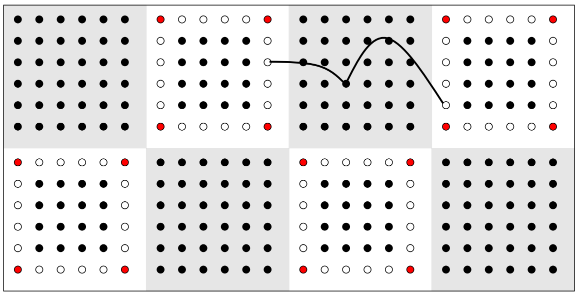

Figure 1: Original small domain DDMHC domain layout. Sites containing spin projected pseudofermion support are labelled with open circles while sites with all spin components in pseudofermions are labelled filled red. The force for the boundary determinant is suppressed by two factors of the quark propagator. By using large domains we have control over the level of suppression and can reduce this significantly using inactive links while retaining a good fraction of active links in the evolution.

2.3 Domain wall pseudofermion structure

The local two flavour determinants are formed in the usual way from Hasenbusch ratios up to the Pauli Villars mass. These solves can be performed independently on each node and decouple

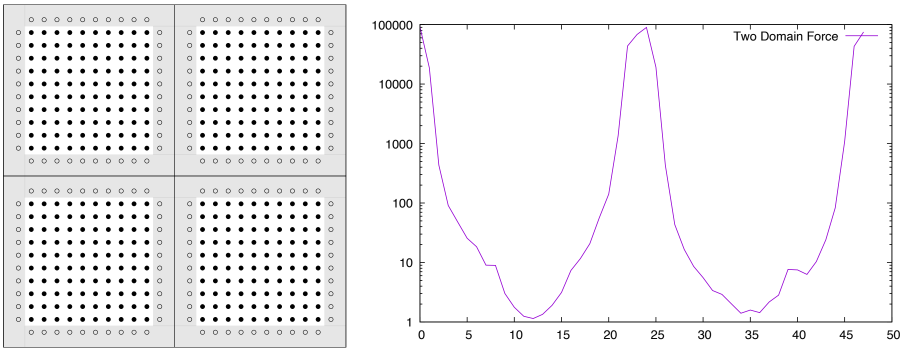

Figure 2: Left: alternate domain decomposition adopted in this work: domain $\bar{\Omega}$ is the union of all surface lattice sites on each computing node and is entirely inactive, while domain Ω is interior. The forces in Ω are somewhat reduced. Right: Force (norm squared) profile from the boundary determinant in a one dimensional domain decomposition on a $16^3 \times 48$ lattice. The hierarchy of force is visible and may be exploited.

communication on the machine. Within these, the usual red-black preconditioned HMC can be used. The domain $\bar{\Omega}$ has frozen link variables throughout a HMC trajectory has no change in determinant. Full sampling is restored with a random translation applied between trajectories.

The two flavour boundary pseudofermion action takes the form

$$\phi^\dagger_{\bar\partial}(RR^\dagger)^{-1}\phi_{\bar\partial}.$$

Where the boundary links are all inactive (so $D_{\bar\partial}$ is *not* differentiated) the force can be calculated as follows. The force term for the local determinant factor is standard but restricted to the local cell, and only active links. However all cells add together so the code implementation will be the standard one, and only the solver will differ.

For the pseudofermion derivative terms involving R, we have,

$$\delta R^{-1} = \mathbb{P}_{\bar\partial} D^{-1} \delta D D^{-1} D_{\bar\partial}.$$

The force is suppressed by quark propagation by the distance from the gauge link to the surface or plane of the domain boundary. This may further be suppressed if more gauge links are kept inactive, perhaps a band of some depth around the plane connecting subvolumes. This is a tunable parameter that pretty much guarantees we can obtain a reasonable ratio in the “size” of gauge forces.

A Hasenbusch ratio between matrices with differing mass R and P (e.g. Pauli Villars boundary) pseudofermion is formed with action,

$$\phi^\dagger_{\bar\partial} P^\dagger R^{-\dagger} R^{-1} P \phi_{\bar\partial}.$$

and we may differentiate this using

$$\delta R^{-1} = \mathbb{P}_{\bar\partial} D^{-1} \delta D D^{-1} D_{\bar\partial}.$$

and

$$\delta R = \mathbb{P}_{\bar\partial} D_\Omega^{-1} (\delta D_\Omega) D_\Omega^{-1} D_\partial D_{\bar\Omega}^{-1} D_{\bar\partial} + \mathbb{P}_{\bar\partial} D_\Omega^{-1} D_\partial D_{\bar\Omega}^{-1} (\delta D_{\bar\Omega}) D_{\bar\Omega}^{-1} D_{\bar\partial}.$$

The pseudofermion derivative term therefore involves four PV local solves, and two non-local light quark solves.

2.4 Results

We carried out an initial implementation of DDHMC by introducing a wrapper for all Grid[12] Fermion action objects that applies Dirichlet boundary conditions to its gauge field argument. The communications may be shut off for local domain solves. The present implemention has a restriction that the suppressed communication boundaries must align with GPU boundaries, however it is planned to generalise the code for multi-GPU nodes to align the domain boundaries at whole node boundaries.

The code was designed with flexibility and the domain shapes are in principle completely general. A projector to the domain is implemented and edges are detected with a mask and shift by one approach in each direction. This allows for algorithmic flexibility and rapid prototyping at the expense of lower efficiency in a performance non-critical part of the code.

A standard two flavour Pseudofermion ratio object is sufficient to simulate the local determinant factors, while a new domain decomposed two flavour ratio object was introduced for the boundary determinant.

We have evolved a two flavour $16^3 \times 48$ system with the Iwasaki gauge action and $\beta = 2.13$. This was subdivided into GPUs, of size $16^3 \times 24$ each and the interior cells of $16^3 \times 22$ evolved in each trajectory. The plaquette time history is displayed in figure 3.

During this evolution a timestep ratio of 4:1 between the boundary determinant and the local determinant was maintained with only three steps per trajectory in the boundary determinant and 12 for the local determinant using the Omelyan integrator.

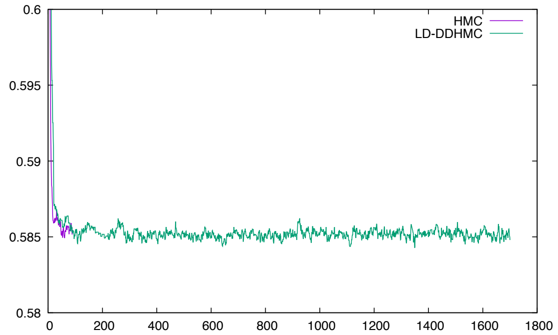

Figure 3: Plaquette log for two flavour DWF simulation on $16^3 \times 48$ with $m_f = 0.01$ and Iwasaki gauge action at $\beta = 2.13$. The HMC and DDHMC agree validating the implementation.

We have prototyped a DDHMC algorithm in the Grid library. It runs efficiently on GPU computing nodes, supporting CUDA, HIP, and SYCL for all large US supercomputer architectures. Multi-core CPUs are also supported but these are not presently the target of the software optimisation and algorithmic tuning.

Initial results suggest correct implementation and a substantially reduced level of communication is enabled.

Although multigrid for Domain Wall Fermions is less mature [16–19] than for Wilson fermions [13–15] we intend to combine the evolution with a recently developed approach to set up multigrid algorithms quickly using Chebyshev filters inside the HMC algorithm. We believe the filter func-

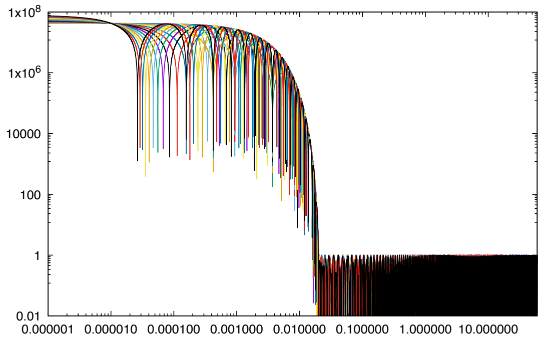

Figure 4: Filter functions obtained with Chebyshev filters for fast set up multigrid. These have been demonstrated to *both* set up and solve a multigrid algorithm faster than a single standard red-black preconditioned Krylov solver for the Shamir DWF case. This is appropriate to use for local domain solves inside DDHMC evolution.

tions, figure 4, (which can be obtained recursively) offer good subspace generation with lower cost than inverse iteration[19].

3. Acknowledgements

PB has been supported in part by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics under the Contract No. DE-SC-0012704 (BNL). P.B. has also received support from the Royal Society Wolfson Research Merit award WM/60035. A.Y. has been supported by Intel.

References

[1] “Non-Perturbative Renormalization and Low Mode Averaging with Domain Wall Fermions”, R. Arthur, PhD Thesis, University of Edinburgh.

[2] E. Shintani, R. Arthur, T. Blum, T. Izubuchi, C. Jung and C. Lehner, “Covariant approximation averaging,” Phys. Rev. D **91** (2015) no.11, 114511 doi:10.1103/PhysRevD.91.114511 [arXiv:1402.0244 [hep-lat]].

[3] M. A. Clark, C. Jung and C. Lehner, “Multi-Grid Lanczos,” EPJ Web Conf. **175** (2018), 14023 doi:10.1051/epjconf/201817514023 [arXiv:1710.06884 [hep-lat]].

[4] “Grid Documentation” https://github.com/paboyle/Grid/blob/develop/documentation/Grid.pdf

[5] M. Luscher, “Schwarz-preconditioned HMC algorithm for two-flavour lattice QCD,” Comput. Phys. Commun. **165** (2005), 199-220 doi:10.1016/j.cpc.2004.10.004 [arXiv:hep-lat/0409106 [hep-lat]].

[6] L. Del Debbio, L. Giusti, M. Luscher, R. Petronzio and N. Tantalo, “QCD with light Wilson quarks on fine lattices (I): First experiences and physics results,” JHEP **02** (2007), 056 doi:10.1088/1126-6708/2007/02/056 [arXiv:hep-lat/0610059 [hep-lat]].

[7] L. Del Debbio, L. Giusti, M. Luscher, R. Petronzio and N. Tantalo, “QCD with light Wilson quarks on fine lattices. II. DD-HMC simulations and data analysis,” JHEP **02** (2007), 082 doi:10.1088/1126-6708/2007/02/082 [arXiv:hep-lat/0701009 [hep-lat]].

[8] S. Aoki *et al.* [PACS-CS], “Physical Point Simulation in 2+1 Flavor Lattice QCD,” Phys. Rev. D **81** (2010), 074503 doi:10.1103/PhysRevD.81.074503 [arXiv:0911.2561 [hep-lat]].

[9] T. Boku, K. I. Ishikawa, Y. Kuramashi, K. Minami, Y. Nakamura, F. Shoji, D. Takahashi, M. Terai, A. Ukawa and T. Yoshie, “Multi-block/multi-core SSOR preconditioner for the QCD quark solver for K computer,” PoS **LATTICE2012** (2012), 188 doi:10.22323/1.164.0188 [arXiv:1210.7398 [hep-lat]].

[10] I. Kanamori, K. I. Ishikawa and H. Matsufuru, “Object-oriented implementation of algebraic multi-grid solver for lattice QCD on SIMD architectures and GPU clusters,” doi:10.1007/978-3-030-86976-2_15 [arXiv:2111.05012 [hep-lat]].

[11] K. I. Ishikawa, I. Kanamori, H. Matsufuru, I. Miyoshi, Y. Mukai, Y. Nakamura, K. Nitadori and M. Tsuji, “102 PFLOPS Lattice QCD quark solver on Fugaku,” [arXiv:2109.10687 [hep-lat]].

[12] P. A. Boyle, G. Cossu, A. Yamaguchi and A. Portelli, “Grid: A next generation data parallel C++ QCD library,” PoS **LATTICE2015** (2016), 023 doi:10.22323/1.251.0023

[13] M. Luscher, “Local coherence and deflation of the low quark modes in lattice QCD,” JHEP **07** (2007), 081 doi:10.1088/1126-6708/2007/07/081 [arXiv:0706.2298 [hep-lat]].

[14] J. Brannick, R. C. Brower, M. A. Clark, J. C. Osborn and C. Rebbi, “Adaptive Multigrid Algorithm for Lattice QCD,” Phys. Rev. Lett. **100** (2008), 041601 doi:10.1103/PhysRevLett.100.041601 [arXiv:0707.4018 [hep-lat]].

[15] R. Babich, J. Brannick, R. C. Brower, M. A. Clark, T. A. Manteuffel, S. F. McCormick, J. C. Osborn and C. Rebbi, “Adaptive multigrid algorithm for the lattice Wilson-Dirac operator,” Phys. Rev. Lett. **105** (2010), 201602 doi:10.1103/PhysRevLett.105.201602 [arXiv:1005.3043 [hep-lat]].

[16] S. D. Cohen, R. C. Brower, M. A. Clark and J. C. Osborn, PoS **LATTICE2011** (2011), 030 doi:10.22323/1.139.0030 [arXiv:1205.2933 [hep-lat]].

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

[18] R. C. Brower, M. A. Clark, D. Howarth and E. S. Weinberg, “Multigrid for chiral lattice fermions: Domain wall,” Phys. Rev. D **102** (2020) no.9, 094517 doi:10.1103/PhysRevD.102.094517 [arXiv:2004.07732 [hep-lat]].

[19] P. Boyle and A. Yamaguchi, “Comparison of Domain Wall Fermion Multigrid Methods,” [arXiv:2103.05034 [hep-lat]].