

Rational Domain-Decomposed HMC

Yoshifumi Nakamura*

Institut für Theoretische Physik, Universität Regensburg, 93040 Regensburg, Germany

E-mail: yoshifumi.nakamura@physik.uni-regensburg.de

We study new HMC algorithm combined the domain-decomposed HMC algorithm and the rational HMC algorithm. We perform numerical tests with the standard Wilson gauge action and 2 flavours of the standard Wilson fermions and compare with HMC algorithm of even-odd preconditioning with chronological inverter. We could not find gain with Rational Domain-Decomposed HMC on 8^4 lattices.

The XXVIII International Symposium on Lattice Field Theory

June 14-19, 2010

Villasimius, Sardinia Italy

*Speaker.

1. Introduction

By recent algorithmic and computational developments the lattice QCD simulations around the physical point became possible. Especially, the Rational Hybrid Monte Carlo (RHMC) algorithm [1, 2] and the domain-decomposed HMC (DDHMC) algorithm [3] led to remarkable cost reduction.

The RHMC algorithm, which is an exact algorithm for any numbers of dynamical fermions, has been used widely over several years. It was originally proposed for rooted kernel since R algorithm is inexact and PHMC needs extra calculation to correct metropolis check. RHMC is also useful for simulations with non-rooted kernel, e.g. 2 flavors of Wilson fermions. It achieves good algorithmic performance by combining some acceleration methods, the n^{th} root trick and relaxation of solver's tolerance for each fraction, or separating fractions to large force and small force part and putting them at different time scale [2]. It is also good point that programming is easy. Once remez algorithm to calculate coefficients for rational approximation and multi-shift solver is implemented, one does not need non-trivial programming.

The DDHMC algorithm makes a geometric separation of fermionic determinant into determinant for small blocks and *Schur complement*. In DDHMC, inverse matrix of the Wilson Dirac operator is solved effectively by using Schwarz alternating procedure. In Schwarz alternating procedure preconditioning, it is possible to use single precision acceleration. In order to suppress forces and minimize communication overhead, the gauge links connecting domains and their staple links are fixed ("dead/alive link method") during molecular dynamics (MD) step was also proposed. When dead/alive link method is employed, to make sure that all gauge links are updated with (almost) same rate, a method such like parallel translation at every trajectory is necessary. For high performance computing, multi nodes/cores architecture is absolutely imperative. Here communication would be a bottleneck. DDHMC is reasonable choice in the sight of this because its communication overhead is less than the others. It is possible to get better performance out of them by domain-decomposition on multiple cores architectures such as Cell Cell/B.E. and GPGPU [7, 8].

In this proceeding we describe new algorithm combined RHMC and DDHMC and report test results for feasibility and efficiency on small lattices.

2. RDDHMC Algorithm

Rational domain-decomposed HMC algorithm is of extension from DDHMC. First we perform a domain decomposition of the Wilson Dirac operator and even odd preconditioning.

$$\begin{aligned} \det D &= \det \begin{pmatrix} D_{EE} & D_{EO} \\ D_{OE} & D_{OO} \end{pmatrix} = \det \begin{pmatrix} D_{EE} & D_{EO} D_{OO}^{-1} \\ 0 & 1 \end{pmatrix} \det \begin{pmatrix} \hat{D}_{EE} & 0 \\ 0 & 1 \end{pmatrix} \det \begin{pmatrix} 1 & 0 \\ D_{OE} & D_{OO} \end{pmatrix}, \\ &= \det(1 - D_{eo} D_{oe})_{EE} \det \hat{D}_{EE} \det(1 - D_{eo} D_{oe})_{OO}, \\ &= \det \tilde{D} \det \hat{D}, \end{aligned} \tag{2.1}$$

where $\hat{D} = 1 - D_{EE}^{-1} D_{EO} D_{OO}^{-1} D_{OE}$ and $\tilde{D} = \det(1 - D_{eo} D_{oe})_{EE} \det(1 - D_{eo} D_{oe})_{OO}$. Note we write e/o as even or odd site and E/O as even or odd domain. By using γ_5 Hermiticity and Eq. (2.1), $\det D$ for degenerate quark masses is expressed as

$$(\det D)^{n_f} = \det(\tilde{D}^\dagger \tilde{D})^{\frac{n_f}{2}} \det(\hat{D}^\dagger \hat{D})^{\frac{n_f}{2}}, \quad (2.2)$$

where n_f is number of flavours. Even when n_f is even, one could calculate the determinant by using the rational approximation like following,

$$\begin{aligned} (\det D)^{n_f} &= [\det(\tilde{D}^\dagger \tilde{D})^{\frac{n_f}{2j_1}}]^{j_1} [\det(\hat{D}^\dagger \hat{D})^{\frac{n_f}{2j_2}}]^{j_2}, \\ X^{\frac{n_f}{2j}} &\approx a_0 + \sum_{i=1} \frac{a_i}{X + b_i}, \end{aligned} \quad (2.3)$$

where X is $\tilde{D}^\dagger \tilde{D}$ or $\hat{D}^\dagger \hat{D}$. The partition function for $n_f = 2$ is

$$Z = \int DU [\det(\tilde{D}^\dagger \tilde{D})^{\frac{1}{j_1}}]^{j_1} [\det(\hat{D}^\dagger \hat{D})^{\frac{1}{j_2}}]^{j_2} e^{-S_g}, \quad (2.4)$$

for $n_f = 2 + 1$

$$\begin{aligned} Z &= \int DU [\det(\tilde{D}^{l\dagger} \tilde{D}^l)^{\frac{1}{j_1}}]^{j_1} [\det(\hat{D}^{l\dagger} \hat{D}^l)^{\frac{1}{j_2}}]^{j_2} \times \\ &\quad [\det(\tilde{D}^{s\dagger} \tilde{D}^s)^{\frac{1}{2k_1}}]^{k_1} [\det(\hat{D}^{s\dagger} \hat{D}^s)^{\frac{1}{2k_2}}]^{k_2} e^{-S_g}. \end{aligned} \quad (2.5)$$

As usual cost reduction methods, multi timescales [4], OMF [5, 6] integrator and relaxing solver's tolerance with keeping reversibility during MD steps are applicable. The single precision acceleration can be used for multiplication of $D_{EE/00}^{-1}$. It, however, is difficult to use chronological guess as an initial guess for solver since high precision is required for restarting multi-shift solver when there are many shifts [9].

2.1 Dead/Alive links

"Dead/Alive link method" and a translation method are introduced by Lüscher. Communication overhead could be minimized by this method. Generally one could make any links non-active during MD steps. We consider 3 types for "Dead/Alive link method" as following.

- Active link method: All links are alive during MD.
- Level 1 non-active link method: Links connecting other domain are dead.
- Level 2 non-active link method: Staples around Level 1 non-active link are also dead.

When two or more domains are put in a node, like left figure of Fig. (1), communication is needed only for multiplication of D_{OE} and D_{EO} in \hat{D} . But, when forces for gauge action and clover term are calculated in case of level 1 non-active link method, data transfer is necessary since derivative of plaquette term of gauge action and clover term at domain surface depend on link variables which are in other domains and change during MD steps. In case of level 2 non-active link method, one does not need data transfer for rectangular term, 1 level smeared hopping term, e.g. SLiNC fermion action, besides plaquette term and clover term. For smeared clover term and more smeared hopping term, further non-active level is required to avoid communication.

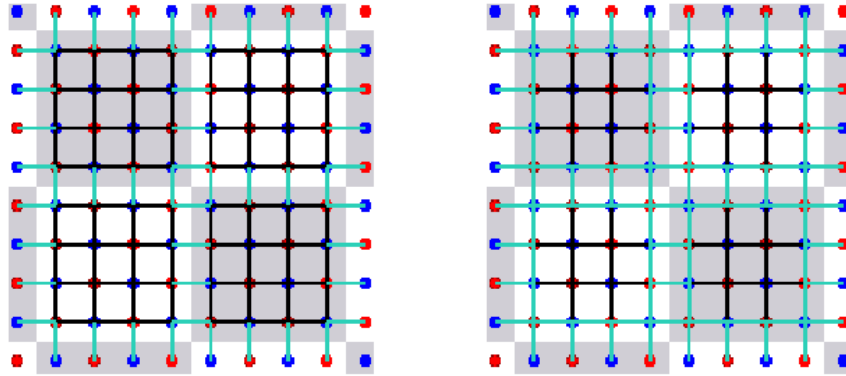


Figure 1: (left) Domain decomposition in the case of $4 \times 4 \times 4 \times 4$ block size. Light blue links denote non-active link, level 1 non-active link method. Red and blue points and wight and gray domain are even-odd sites and even/odd domains, respectively. (right) Level 2 non-active link method, additionally links around blue links of left figure are non-active

2.2 Tuning rational fraction part

Usually to avoid generating coefficients for the rational approximation every time we implement/calculate specific sets of coefficients in advance. If approximation range of generated coefficients is wider than the condition number of X , one can use them by shifting as following, when range does not cover actual $[\min, \max]$ of X .

$$\begin{aligned} X^\alpha &= \beta^{-\alpha} (\beta X)^\alpha \\ &\approx \beta^{-\alpha} \left[c_0 + \sum_{i=1} \frac{c_i}{\beta X + d_i} \right] \end{aligned} \quad (2.6)$$

where β is inverse of the minimum eigenvalue of X and c_i, d_i is generated by remez algorithm with range for one to the condition number of X , $[1, C(X)]$. When the condition number is big and an accurate approximation is required, CPU time to generate new coefficients a number of times is unignorable except for parallelizing the remaz algorithm.

Fig. (2) is an example at some trajectory for relation between the number of averaged iteration and maximum force over MD step for each partial fraction of $(\hat{D}^\dagger \hat{D})^{-1/4}$ in case that approximation range is set as $[\min/\sqrt{2}, \sqrt{2} \max]$, block size is $4 \times 6 \times 6 \times 4$ and non-active link level is 1. When a multi-timescale integrator is used, the term that force is small (big) and numerical cost is expensive (cheap) are generally put at coarser (finer) timescale. While it is possible to put partial fractions at different timescales, it, however, is easier to reduce cost that relaxing tolerances of a few first fractions since their forces are small and all forces are summed up.

3. Numerical tests

We perform numerical tests with the standard Wilson action at $\beta = 5.00$ and 2 flavours of standard Wilson fermions for $\kappa = 0.100$ to 0.17 (cf. $\kappa_c = 0.187(1)$) on 8^4 lattices. To check algorithm

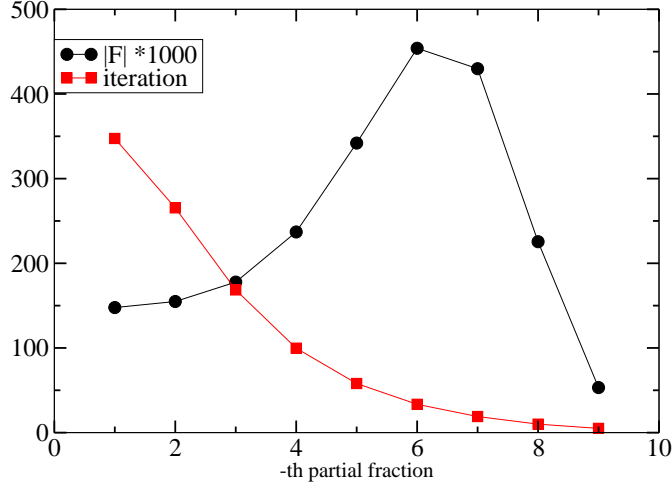


Figure 2: The number of iteration and maximum force for each partial fraction of $(\hat{D}^\dagger \hat{D})^{-1/4}$ on $24^3 \times 32$ lattice for the standard Wilson gauge action at $\beta = 5.60$ and $\kappa = 0.1575$ for $n_f = 2$ standard Wilson fermions.

correctness we compare results with HMC, which is accelerated by even-odd preconditioning, chronological inverter and multi timescale Omelyan integrator.

For RDDHMC, chronological inverter is not used since multi-shift solver with the double precision does not work when there are many shifts, although multi timescale Omelyan integrator is used. The block size of domain is $4 \times 4 \times 4 \times 4$ and both active and level 1 non-active link method are tested. Level 2 non-active link method is not tested because 90% of links become non-active in case of such small domain size. For rational approximation Eq. (2.4), $j_1 = j_2 = 2$ and degree of approximation is 20 and 25 for force and action calculation, respectively. To solve multi-shifted system, multi-shift CG is used and GMRES is used to solve D_{EE}^{-1} and D_{OO}^{-1} .

Fig. (3) shows plaquette values and $e^{-\Delta H}$ for each κ for different algorithms. Statistics is $O(2000)$ and first 1000 trajectories are discarded for thermalisation. The plaquette values are consistent and $e^{-\Delta H}$ is 1 within error. In fig. (4) we plot gauge force and fermionic force as a function of κ . The forces become large as increasing κ . Fermionic force of HMC is separated to two parts, which are of $\hat{D}^\dagger \hat{D}$ and $\tilde{D}^\dagger \tilde{D}$ by changing to RDDHMC algorithm. When non-active link method is used, forces of gauge action and $\phi^\dagger (\hat{D}^\dagger \hat{D})^{-1} \phi$ are suppressed. On the other hand, force of $\phi^\dagger (\tilde{D}^\dagger \tilde{D})^{-1} \phi$ is not suppressed because \tilde{D} both for active and level 1 non-active link method are same.

In this test, numerical cost for RDDHMC is much more expensive than HMC (see Fig (5)). The multiplication of $\hat{D}^\dagger \hat{D}$, which includes four $D_{EE/OO}^{-1}$, with the double precision is performed to solve $(\hat{D}^\dagger \hat{D} + \sigma_i)x_i = b_i$ at RDDHMC. Because the condition number of $\hat{D}^\dagger \hat{D}$ is as large as one of $D^\dagger D$, multiplication of $D_{EE/OO}^{-1}$ is calculated many times inside of multi-shift solver. In fact, most of the time is spent for this. Therefore optimization for multiplication of $D_{EE/OO}$ and cost reduction at computing multiplication of $D_{EE/OO}^{-1}$, for example by deflation, are important.

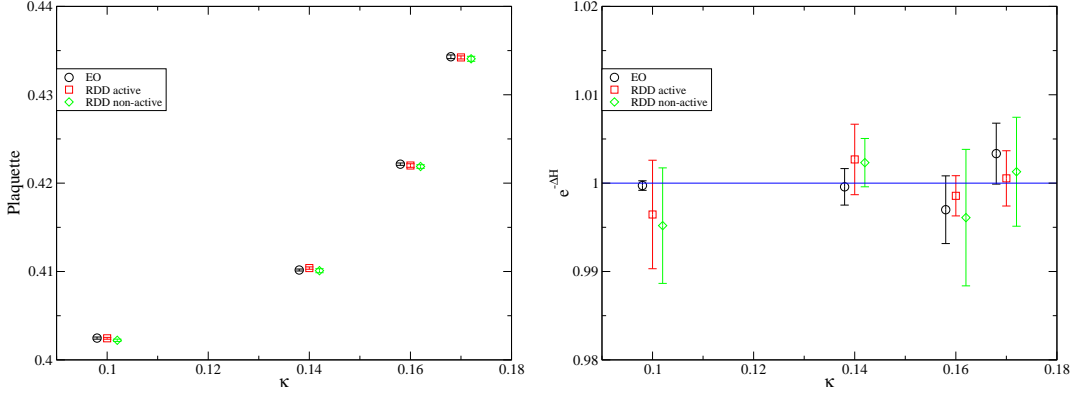


Figure 3: The plaquette values(left) and $e^{-\Delta H}$ (right) at $\beta = 5.00$ on 8^8 for ordinal even-odd preconditioned HMC, RDDHMC(active) RDDHMC (level 1 non-active link method) algorithm.

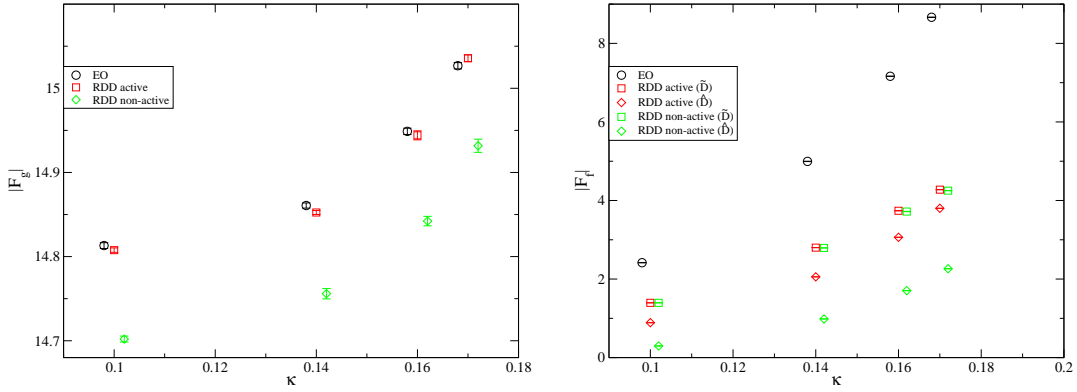


Figure 4: Norm of gauge force (left) and fermionic force (right) with same algorithmic parameters as Fig. (3).

4. Summary

We have been described RDDHMC algorithm and tested it on small lattices. The values for plaquette are consistent with standard HMC and $e^{-\Delta H}$ is equal to 1 within error. So RDDHMC algorithm seems to be correct. Unfortunately, we could not see any gain in this test. One reason is that inversion for $\hat{D}^\dagger \hat{D}$, which is needed for both action and force calculations is expensive since $\hat{D}^\dagger \hat{D}$ contains $D_{EE/OO}^{-1}$. Besides this, force of $\phi^\dagger (\hat{D}^\dagger \hat{D})^{-1} \phi$ is small even in case of level 1 non-active link method. On large lattices we expect better situation since force of $\phi^\dagger (\hat{D}^\dagger \hat{D})^{-1} \phi$ is suppressed further by level 2 non-active link method with bigger block size.

5. Acknowledgements

We would like to thank Andrea Nobile, Dirk Pleiter and Hinnerk Stüben for useful discussions. Numerical simulations were done on PC-Clusters at University of Regensburg.

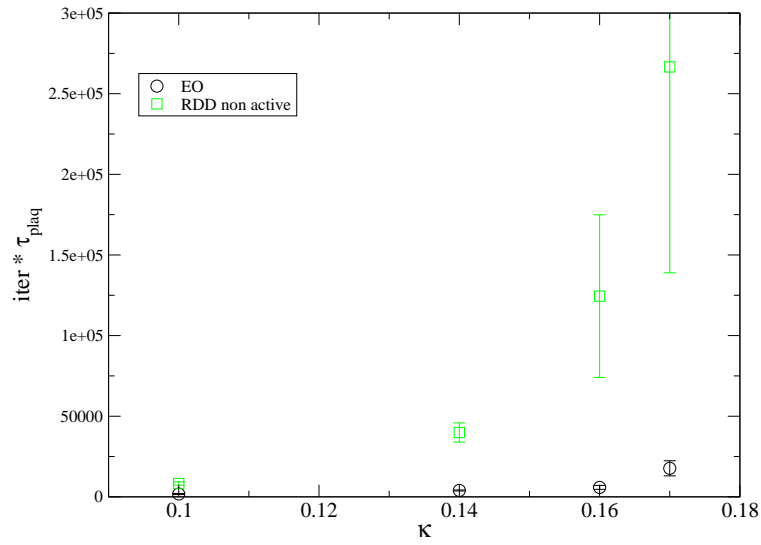


Figure 5: The cost for HMC and RDDHMC (level 1 non-active link method) at $\beta = 5.00$ on 8^8 .

References

- [1] M. A. Clark and A. D. Kennedy, Nucl.Phys.Proc.Suppl. 129 (2004) 850-852 [hep-lat/0309084], M. A. Clark, PoS (LAT2006) 004.
- [2] M. A. Clark and A. D. Kennedy, Phys. Rev. Lett. **98** 051601, (2007).
- [3] M. Lüscher, JHEP **0305**, 052 (2003); Comput. Phys. Commun. **165**, 199 (2005).
- [4] J. C. Sexton and D. H. Weingarten, Nucl. Phys. **B380**, 665-678 (1992).
- [5] I. P. Omelyan, I. M. Mryglod, and R. Folk, Comput. Phys. Commun. **151**, 272 (2003).
- [6] T. Takaishi and P. de Forcrand, Phys. Rev. **E73**, 036706 (2006).
- [7] A. Nobile, PoS (LATTICE 2010) 034.
- [8] Y. Osaki and K.-I. Ishikawa, PoS (LATTICE 2010) 034.
- [9] J. C. Osborn, PoS (LATTICE 2008) 029.