Application of block Krylov subspace algorithms to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD

T. Sakurai[a], H. Tadano[a,b], Y. Kuramashi[c,b]

[a]*Department of Computer Science, University of Tsukuba,*
Tsukuba, Ibaraki 305-8573, Japan
[b]*Center for Computational Sciences, University of Tsukuba,*
Tsukuba, Ibaraki 305-8577, Japan
[c]*Graduate School of Pure and Applied Sciences, University of Tsukuba,*
Tsukuba, Ibaraki 305-8571, Japan

Abstract

It is well known that the block Krylov subspace solvers work efficiently for some cases of the solution of differential equations with multiple right-hand sides. In lattice QCD calculation of physical quantities on a given configuration demands us to solve the Dirac equation with multiple sources. We show that a new block Krylov subspace algorithm recently proposed by the authors reduces the computational cost significantly without loosing numerical accuracy for the solution of the $O(a)$-improved Wilson-Dirac equation.

Key words:
Lattice gauge theory, Lattice Dirac equation, multiple right-hand sides, block Krylov subspace

1. Introduction

In the last decade one of the primary issues in lattice QCD is to reduce the dynamical up and down quark masses toward the physical values. Most of our efforts have been devoted to reduce the computational cost for the configuration generation with light quark masses. Thanks to the algorithmic improvement together with rapid increase of the computational power we are now allowed to make a direct full QCD simulation on the physical up and down quark masses[1, 2]. On the other hand, we have been less concerned

1

with the algorithmic improvement for the calculation of physical quantities: its computational cost was negligibly smaller than that for the configuration generation until recently. At present, however, the latter is drastically reduced to be comparable to or smaller than the former.

The characteristic feature in the calculation of physical quantities is the solution of the Dirac equation with multiple sources: twelve in the simplest case and $O(10-100)$ for the stochastic technique. These are typical examples of the solution of differential equations with multiple right-hand sides. For this type of equations it is known that the block Krylov subspace solvers succeed in reducing the computational cost[4]. In this article we study the application of the block BiCGSTAB algorithm[5] and a new block Krylov subspace method proposed in Ref. [6] to the $O(a)$-improved Wilson-Dirac equation which is one of the popular fermion formulations in current lattice QCD simulations. For simplicity our numerical test is restricted on the quenched configuration with a fixed volume. We investigate the quark mass dependence of the algorithmic efficiency in detail.

This paper is organized as follows. In Sec. 2 we give the definition of the $O(a)$-improved Wilson-Dirac equation. The algorithmic details are described in Sec. 3. In Sec. 4 we present the results of the numerical tests after explaining the parameter choice and the machine specifications. Our conclusions are summarized in Sec. 5.

2. Wilson-Dirac equation

The lattice QCD is defined on a hypercubic four-dimensional lattice of finite extent being expressed as $L_x \times L_y \times L_z \times L_t$ with $L_{x,y,z}$ the three-dimensional spatial extent and L_t the temporal one. The lattice spacing is set to unity for notational convenience. The fields are defined on the sites n with periodic boundary conditions.

We define two types of fields on the lattice. One is the gauge field represented by $U_\mu(n, a, b)$ with $\mu = 1, 2, 3, 4$ and $a, b = 1, 2, 3$ which is a 3×3 SU(3) matrix assigned on each link. The other is the quark field $q(n, \alpha, a)$ which resides on each site carrying the Dirac index $\alpha = 1, 2, 3, 4$ and the color index $a = 1, 2, 3$. The $O(a)$-improved Wilson-Dirac operator is written as

$$
\begin{aligned}
D_{\mathrm{W}}(n, \alpha, a; m, \beta, b) \;=\; & \delta_{n,m}\delta_{\alpha,\beta}\delta_{a,b} \\
& -\kappa \sum_{\mu=1,\ldots,4} [\{\delta_{\alpha,\beta} - \gamma_\mu(\alpha, \beta)\} U_\mu(n, a, b)\delta_{m,n+\hat{\mu}}
\end{aligned}
$$

$$+\{\delta_{\alpha,\beta} + \gamma_\mu(\alpha,\beta)\}U_\mu^\dagger(n - \hat{\mu}, a, b)\delta_{m,n-\hat{\mu}}\big]$$

$$-\kappa \cdot c_{\text{SW}} \sum_{\mu,\nu=1,\dots,4} \frac{i}{2}\sigma_{\mu\nu}(\alpha,\beta)F_{\mu\nu}(n, a, b)\delta_{m,n}, \quad (1)$$

where $\hat{\mu}$ denotes the unit vector in the μ direction. The coefficient c_{SW} is a parameter to be adjusted for the $O(a)$ improvement. The Euclidean gamma matrices are defined in terms of the Minkowski ones in the Bjorken-Drell convention: $\gamma_j = -i\gamma_{BD}^j$ $(j = 1, 2, 3)$, $\gamma_4 = \gamma_{BD}^0$, $\gamma_5 = \gamma_{BD}^5$ and $\sigma_{\mu\nu} = \frac{1}{2}[\gamma_\mu, \gamma_\nu]$. The explicit representations for $\gamma_{1,2,3,4,5}$ are given by

$$\gamma_1 = \begin{pmatrix} & & & -i \\ & & -i & \\ & i & & \\ i & & & \end{pmatrix}, \quad (2)$$

$$\gamma_2 = \begin{pmatrix} & & & -1 \\ & & 1 & \\ & 1 & & \\ -1 & & & \end{pmatrix}, \quad (3)$$

$$\gamma_3 = \begin{pmatrix} & & -i & \\ & & & i \\ i & & & \\ & -i & & \end{pmatrix}, \quad (4)$$

$$\gamma_4 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix}, \quad (5)$$

$$\gamma_5 = \begin{pmatrix} & & 1 & \\ & & & 1 \\ 1 & & & \\ & 1 & & \end{pmatrix}, \quad (6)$$

where we list only nonzero elements. The field strength $F_{\mu\nu}$ in the last term of Eq. (1) is expressed as

$$F_{\mu\nu}(n, a, b) = \frac{1}{4}\sum_{i=1}^{4}\frac{1}{2i}\left(P_i(n, a, b) - P_i^\dagger(n, a, b)\right) \quad (7)$$

3

with

$$P_1(n,a,b) = \sum_{c,d,e} U_\mu(n,a,c)U_\nu(n+\hat{\mu},c,d)U_\nu^\dagger(n+\hat{\nu},d,e)U_\nu^\dagger(n,e,b), \quad (8)$$

$$P_2(n,a,b) = \sum_{c,d,e} U_\nu(n,a,c)U_\mu^\dagger(n-\hat{\mu}+\hat{\nu},c,d)$$
$$\times U_\nu^\dagger(n-\hat{\mu},d,e)U_\mu(n-\hat{\mu},e,b), \quad (9)$$

$$P_3(n,a,b) = \sum_{c,d,e} U_\mu^\dagger(n-\hat{\mu},a,c)U_\nu^\dagger(n-\hat{\mu}-\hat{\nu},c,d)$$
$$\times U_\mu(n-\hat{\mu}-\hat{\nu},d,e)U_\nu(n-\hat{\nu},e,b), \quad (10)$$

$$P_4(n,a,b) = \sum_{c,d,e} U_\nu^\dagger(n-\hat{\nu},a,c)U_\nu(n-\hat{\nu},c,d)$$
$$\times U_\nu(n+\hat{\mu}-\hat{\nu},d,e)U_\mu^\dagger(n,e,b). \quad (11)$$

The $O(a)$-improved Wilson-Dirac operator defined by Eq. (1) is a complex non-Hermitian square matrix, where only 51 out of $L_x \times L_y \times L_z \times L_t \times 3 \times 4$ entries in each row have nonzero values. The matrix becomes fairly sparse in current numerical simulations with $L_{x,y,z,t} \sim O(10)$.

The calculation of physical quantities requires the solution of the following linear equations:

$$\sum_{m,\beta,b} D_{\mathrm{W}}(n,\alpha,a;m,\beta,b)x^{(i)}(m,\beta,b) = s^{(i)}(n,\alpha,a), \quad (12)$$

where $s^{(i)}$ represents the i-th source vector. To illustrate the situation we consider the calculation of the hadron two-point function with the point source at the origin as a simplest example. In this case we need twelve source vectors expressed as

$$s^{(i)}(1,\alpha,a) = \begin{cases} 1 & i = a + 3(\alpha-1) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

with $a = 1,2,3$ and $\alpha = 1,2,3,4$. Another good example is the stochastic technique which usually requires $O(10-100)$ noise sources. Although we can think of a lot of other interesting examples, our numerical tests, which is explained later in Sec. 4, concentrate on the simplest one.

3. Block Krylov subspace methods

Before explaining block Krylov subspace algorithms it should be better to reformulate the problem in a generalized form. This help the readers easily understand the essence avoiding any complex notations specific to lattice QCD.

Our interest exists in the solution of linear systems with the multiple right-hand sides expressed as

$$AX = B, \tag{14}$$

where A is an $N \times N$ complex sparse non-Hermitian matrix. X and B are $N \times L$ complex rectangular matrices given by

$$X = \left(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(i)}, \ldots, \boldsymbol{x}^{(L)} \right), \tag{15}$$
$$B = \left(\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(i)}, \ldots, \boldsymbol{b}^{(L)} \right). \tag{16}$$

In the case of the Wilson-Dirac equation $N = L_x \times L_y \times L_z \times L_t \times 3 \times 4$ and L is the number of the source vectors.

For a preparative purpose we first write down the well-known BiCGSTAB algorithm for solving a single right-hand side linear system, where $\boldsymbol{x} = \boldsymbol{x}^{(1)}$ and $\boldsymbol{b} = \boldsymbol{b}^{(1)}$ in Eq. (14):

> $\boldsymbol{x}_0$ is an initial guess,
> **Compute $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,**
> **Set $\boldsymbol{p}_0 = \boldsymbol{r}_0$,**
> **Choose $\tilde{\boldsymbol{r}}_0$ such that $(\tilde{\boldsymbol{r}}_0, \boldsymbol{r}_0) \neq 0$,**
> **For $k = 0, 1, \ldots$, until $\|\boldsymbol{r}_k\|_2 / \|\boldsymbol{b}\|_2 \leq \epsilon$ do:**
> $$\begin{aligned} \alpha_k &= (\tilde{\boldsymbol{r}}_0, \boldsymbol{r}_k)/(\tilde{\boldsymbol{r}}_0, A\boldsymbol{p}_k), \\ \boldsymbol{t}_k &= \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k, \\ \zeta_k &= (A\boldsymbol{t}_k, \boldsymbol{t}_k)/(A\boldsymbol{t}_k, A\boldsymbol{t}_k), \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k + \zeta_k \boldsymbol{t}_k, \\ \boldsymbol{r}_{k+1} &= \boldsymbol{t}_k - \zeta_k A\boldsymbol{t}_k, \\ \beta_k &= (\alpha_k/\zeta_k) \cdot (\tilde{\boldsymbol{r}}_0, \boldsymbol{r}_{k+1})/(\tilde{\boldsymbol{r}}_0, \boldsymbol{r}_k), \\ \boldsymbol{p}_{k+1} &= \boldsymbol{r}_{k+1} + \beta_k (\boldsymbol{p}_k - \zeta_k A\boldsymbol{p}_k), \end{aligned}$$
> **End for.**

It is rather straightforward to extend the algorithm to the blocked version for solving multiple right-hand sides linear system:

$X_0 \in \mathbb{C}^{N \times L}$ is an initial guess,
Compute $R_0 = B - AX_0$,
Set $P_0 = R_0$,
Choose $\tilde{R}_0 \in \mathbb{C}^{N \times L}$,
For $k = 0, 1, \ldots,$ **until** $\max_i(\|\boldsymbol{r}_k^{(i)}\|_2 / \|\boldsymbol{b}^{(i)}\|_2) \leq \epsilon$ **do:**

$$
\begin{aligned}
V_k &= AP_k, \\
\text{\textbf{Solve} } &(\tilde{R}_0^{\mathrm{H}} V_k)\alpha_k = \tilde{R}_0^{\mathrm{H}} R_k \text{ for } \alpha_k, \\
T_k &= R_k - V_k \alpha_k, \\
Z_k &= AT_k, \\
\zeta_k &= \mathrm{Tr}\left(Z_k^{\mathrm{H}} T_k\right) / \mathrm{Tr}\left(Z_k^{\mathrm{H}} Z_k\right), \\
X_{k+1} &= X_k + P_k \alpha_k + \zeta_k T_k, \\
R_{k+1} &= T_k - \zeta_k Z_k, \\
\text{\textbf{Solve} } &(\tilde{R}_0^{\mathrm{H}} V_k)\beta_k = -\tilde{R}_0^{\mathrm{H}} Z_k \text{ for } \beta_k, \\
P_{k+1} &= R_{k+1} + (P_k - \zeta_k V_k)\beta_k,
\end{aligned}
$$

End for,

where R_k, P_k, T_k are $N \times L$ complex rectangular matrices and α_k, β_k are $L \times L$ complex square ones. At the k-th iteration in the block BiCGSTAB algorithm we find

$$
\begin{aligned}
X_k &\in X_0 + \mathcal{K}_k^{\blacksquare}(A; R_0), &(17) \\
R_k &\in \mathcal{K}_{k+1}^{\blacksquare}(A; R_0), &(18)
\end{aligned}
$$

where $\mathcal{K}_k^{\blacksquare}(A; R_0)$ is a block Krylov subspace defined by

$$
\mathcal{K}_k^{\blacksquare}(A; R_0) \equiv \left\{ \sum_{j=0}^{k-1} A^j R_0 \xi_j \; ; \; \xi_j \in \mathbb{C}^{L \times L} \right\}.
$$

This yields an essential difference from the consecutive application of the BiCGSTAB algorithm to $A\boldsymbol{x}^{(i)} = \boldsymbol{b}^{(i)}$ for $i = 1, \ldots, L$: The blocked version searches the solution vectors with the enlarged Krylov subspace.

In Ref. [6] a new block Krylov subspace method is proposed. This method improves the numerical accuracy of the block BiCGSTAB method where multiplication of the matrix α_k yields contaminations of the rounding error on the solution[6]. The algorithm is as follows:

$X_0 \in \mathbb{C}^{n \times L}$ is an initial guess,
Compute $R_0 = B - AX_0$,

Set $P_0 = R_0$ and $V_0 = W_0 = AR_0$,
Choose $\tilde{R}_0 \in \mathbb{C}^{n \times L}$,
For $k = 0, 1, \ldots,$ **until** $\max_i(\|\boldsymbol{r}_k^{(i)}\|_2/\|\boldsymbol{b}^{(i)}\|_2) \leq \epsilon$ **do:**
 Solve $(\tilde{R}_0^{\mathrm{H}} V_k)\alpha_k = \tilde{R}_0^{\mathrm{H}} R_k$ for α_k,

$$
\begin{aligned}
\zeta_k &= \mathrm{Tr}\left[W_k^{\mathrm{H}} R_k\right]/\mathrm{Tr}\left[W_k^{\mathrm{H}} W_k\right], \\
S_k &= P_k - \zeta_k V_k, \\
U_k &= S_k \alpha_k, \\
Y_k &= AU_k, \\
X_{k+1} &= X_k + \zeta_k R_k + U_k, \\
R_{k+1} &= R_k - \zeta_k W_k - Y_k, \\
W_{k+1} &= AR_{k+1},
\end{aligned}
$$

 Solve $(\tilde{R}_0^{\mathrm{H}} R_k)\gamma_k = \tilde{R}_0^{\mathrm{H}} R_{k+1}/\zeta_k$ for γ_k,

$$
\begin{aligned}
P_{k+1} &= R_{k+1} + U_k \gamma_k, \\
V_{k+1} &= W_{k+1} + Y_k \gamma_k,
\end{aligned}
$$

End

This algorithm is constructed to avoid the rounding error problem observed in the block BiCGSTAB method.

4. Numerical tests

4.1. Choice of parameters

The L dependence of the efficiency of the block Krylov algorithms, in which we are most interested, should be investigated on single CPU avoiding contaminations due to the communication overhead. The memory requirement forces the lattice size to be moderate. Our numerical tests are performed with samples of 10 statistically independent gauge field configurations on a $16^3 \times 32$ lattice generated by the Iwasaki gauge action at $\beta = 2.575$ in quenched approximation, which was employed in Ref. [3]. We solve the Wilson-Dirac equation with the local source at the origin choosing four hopping parameters $\kappa = 0.1359, 0.1357, 0.1355, 0.1300$ with the improvement coefficient $c_{\mathrm{SW}} = 1.345$. The bare quark mass is given by $m_q = (1/\kappa - 1/\kappa_c)/2$ with $\kappa_c = 0.136116(8)$, which increases as κ decreases. In Table 1 we list the pion mass in physical unit and the m_π/m_ρ ratio at each hopping parameter following Ref. [3]. Although the pion mass at $\kappa = 0.1300$ is extremely large from a view point of physical interest, we employ it for a comparative purpose. The lattice spacing estimated by m_ρ is $a = 0.1130$ fm[3].

Table 1: Pion mass and m_π/m_ρ ratio at each hopping parameter.

κ	$\kappa_c = 0.136116(8)$	0.1359	0.1357	0.1355	0.1300
m_π [MeV]	0	221	307	375	1282
m_π/m_ρ	0	0.28	0.39	0.46	0.87

4.2. Test environment

Numerical tests are carried out on single node of T2K-Tsukuba which is a large-scale cluster system 648 compute nodes providing 95.4Tflops of computing capability. Each node consists of quad-socket, 2.3GHz Quad-Core AMD Opteron Model 8356 processors whose on-chip cache sizes are 64KBytes/core, 512KBytes/core, 2MB/chip for L1, L2, L3, respectively. Each processor has a direct connect memory interface to an 8GBytes DDR2-667 memory and three hypertransport links to connect other processors. All the nodes in the system are connected through a full-bisectional fat-tree network consisting of four interconnection links of 8GBytes/sec aggregate bandwidth with Infiniband.

4.3. Results

Table 2 shows the L dependence of the computational cost to solve the Wilson-Dirac equation with the block BiCGSTAB algorithm imposing rather stringent tolerance $\epsilon = 10^{-14}$ for the relative residual. We present averaged values over 10 configuration samples for the number of iteration and the execution time. The true residual, which is defined by $\max_i \|Ax^{(i)} - b^{(i)}\|_2 / \|b^{(i)}\|_2$, is evaluated after the relative residual reaches the tolerance. The maximum and minimum values among 10 configuration samples are listed. Although the computational cost divided by L is considerably reduced as L increases at $\kappa = 0.1359, 0.1357, 0.1355$, there exist two concerns: One is the discrepancy between the relative residual and the true one which is enhanced as L increases. The other is the convergence failure of the relative residual, which is found for 4 samples out of 10 at $\kappa = 0.1359$ with $L = 4$. This flaw is also observed at $\kappa = 0.1355$ and 0.1357 once we go beyond $L = 4$.

In order to remove the discrepancy of the relative residual and true one, we proposed a new algorithm whose details are presented in Ref. [6]. The effectiveness of the new method is observed in Table 3, where the deviation between the true and the relative residuals essentially vanishes. Figure 1 plots the number of iteration and the execution time divided L as a function of L. We observe two important features. One is the acceleration of the

cost reduction at the light quark masses where the physical interest exists: At $\kappa = 0.1357$ the number of iteration is almost cut in half from $L = 1$ to 4, which should be compared to the case of $\kappa = 0.1300$. The other is that the executional time divided by L decreases faster than the number of iteration as L increases: The former is reduced by more than 60% from $L = 1$ to 4 at $\kappa = 0.1357$. This fact demonstrates the efficiency of the cache-aware implementation for the matrix multiplication on the multiple vectors. In Fig. 2 we show a representative behavior of the relative residual as a function of the number of iteration at each hopping parameter. We observe little L dependence up to 700 iterations, beyond which the convergence speed is accelerated as L increases at $\kappa = 0.1359, 0.1357, 0.1355$.

Table 2: L dependence of the number of iteration and the execution time to solve the Wilson-Dirac equations with the block BiCGSTAB method. True residual is evaluated after the relative residual reaches the tolerance $\epsilon = 10^{-14}$.

κ	L	#iteration	time [sec]	time/L [sec]	true residual max	true residual min
0.1359	1	2432.6	1439.3	1439.3	1.57×10^{-14}	7.99×10^{-15}
	2	1696.8	1316.0	658.0	2.91×10^{-11}	2.36×10^{-13}
	4	—	—	—	—	—
0.1357	1	1999.0	1162.4	1162.4	1.08×10^{-14}	5.98×10^{-15}
	2	1410.1	1092.4	546.2	2.62×10^{-12}	3.75×10^{-14}
	4	1100.0	1633.3	408.3	1.36×10^{-11}	6.33×10^{-13}
0.1355	1	1518.5	884.4	884.4	1.09×10^{-14}	5.25×10^{-15}
	2	1264.2	979.6	489.8	2.55×10^{-12}	1.49×10^{-14}
	4	961.7	1430.0	357.5	1.30×10^{-11}	2.30×10^{-13}
0.1300	1	165.3	96.4	96.4	9.29×10^{-15}	3.68×10^{-15}
	2	172.8	134.3	67.2	1.57×10^{-14}	6.14×10^{-15}
	4	181.0	272.7	68.2	3.42×10^{-13}	7.99×10^{-15}

5. Conclusions

In this paper we present the first example for the efficiency of the block Krylov subspace methods in lattice QCD to solve the $O(a)$-improved Wilson-Dirac equation with multiple local sources. We find remarkable cost reductions for the light quark masses. Roughly speaking, the solver performance normalized by L is doubled by increasing L from one to four at the light

Table 3: L dependence of the number of iteration and the execution time to solve the Wilson-Dirac equations with the new method. True residual is evaluated after the relative residual reaches the tolerance $\epsilon = 10^{-14}$.

κ	L	#iteration	time [sec]	time/L [sec]	true residual max	true residual min
0.1359	1	2440.3	1398.2	1398.2	1.53×10^{-14}	8.57×10^{-15}
	2	1701.5	1308.4	654.2	2.92×10^{-13}	6.65×10^{-15}
	4	−	−	−	−	−
0.1357	1	1986.6	1138.1	1138.1	1.14×10^{-14}	6.81×10^{-15}
	2	1417.0	1090.1	545.1	1.08×10^{-14}	8.51×10^{-15}
	4	1063.6	1556.6	389.2	1.51×10^{-14}	6.72×10^{-15}
0.1355	1	1519.0	870.0	870.0	1.18×10^{-14}	9.03×10^{-15}
	2	1252.9	962.9	481.5	1.08×10^{-14}	7.68×10^{-15}
	4	975.7	1427.1	356.8	1.29×10^{-14}	7.23×10^{-15}
0.1300	1	165.4	95.3	95.3	9.34×10^{-15}	4.02×10^{-15}
	2	173.1	133.6	66.8	9.79×10^{-15}	5.15×10^{-15}
	4	181.7	266.1	66.5	9.10×10^{-15}	6.79×10^{-15}

quark masses. The block Krylov subspace methods have two advantageous points. Firstly we can easily implement the method by extending the conventional solver for a single right hand side. The deviation between the relative residual and the true one observed in the block BiCGSTAB algorithm is successfully removed by the new algorithm proposed in Ref. [6]. Although our numerical tests are carried out on a single CPU, it is obvious that there is no difficulty in parallelization. Secondly the multiplication of the Wilson-Dirac matrix on the multiple vectors allow us an effective use of the cache, where the link variables can be retained during the operation. The cost reduction is achieved by not only the algorithmic efficiency but also the implementation technique. One concern about the methods is that the increase of L makes difficult the convergence of the relative residuals at light quark masses. We are now investigating its origin and possible improvements.
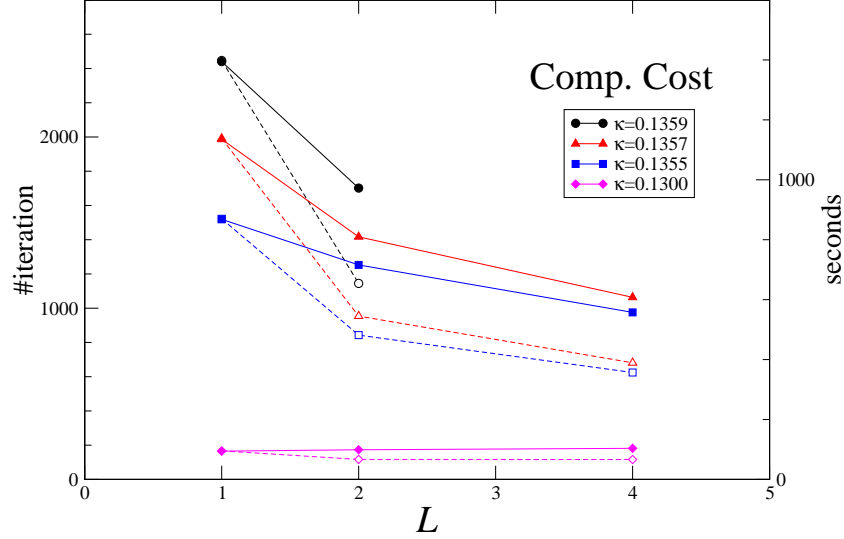
Acknowledgments

Figure 1: L dependence of the number of iteration (solid) and the execution time divided by L (open) for the new method. All the results are averaged over 10 configuration samples.

References

[1] PACS-CS Collaboration, S. Aoki *et al.*, 2+1 Flavor Lattice QCD toward the Physical Point, Phys. Rev. **D 79**, 034503 (2009).

[2] Y. Kuramashi, PACS-CS Results for 2+1 Flavor Lattice QCD Simulation on and off the Physical Point, PoS **LATTICE 2008** (2008) 018.

[3] CP-PACS Collaboration, A. Ali Khan *et al.*, Dynamical Quark Effects on Light Quark Masses, Phys. Rev. Lett. **85** (2000) 4674; Light Hadron Spectroscopy with Two Flavors of Dynamical Quarks on the Lattice, Phys. Rev. **D65** (2002) 054505.

[4] A. A. Nikishin, A. Yu. Yeremin, Variable Block CG Algorithms for Solving Large Sparse Symmetric Positive Definite Linear Systems on Parallel
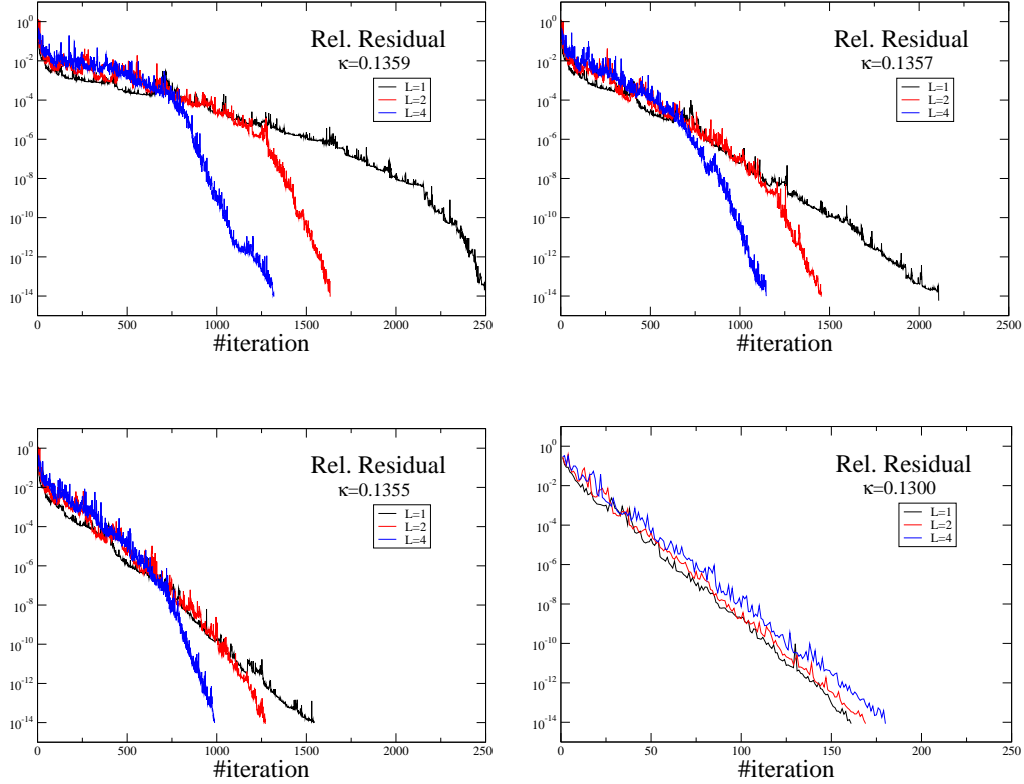
Figure 2: Representative behaviors of the relative residual as a function of the number of iteration. All the measurements are performed on the same configuration.

Computers, I: General Iterative Scheme, SIAM J. Matrix Anal. Appl. **16** (1995) 1135.

[5] A. El Guennouni, K. Jbilou, H. Sadok, A Block Version of BiCGSTAB for Linear Systems with Multiple Right-Hand Sides, Elec. Trans. Numer. Anal. **16** (2003) 129.

[6] H. Tadano, T. Sakurai, Y. Kuramashi, A New Block Kryrov Subspace Method for Computing High Accuracy Solutions, preprint, CS-TR-08-16 (2009).