A Preconditioner for Improved Fermion Actions

Wolfgang Bietenholz[1], Norbert Eicker[1], Andreas Frommer[3], Thomas Lippert[2], Björn Medeke[3], and Klaus Schilling[1,2]

[1] NIC, c/o Research Center Jülich, 52425 Jülich, Germany
[2] Department of Physics, University of Wuppertal, 42097 Wuppertal, Germany
[3] Department of Mathematics, University of Wuppertal, 42097 Wuppertal, Germany

Abstract. SSOR preconditioning of fermion matrix inversions which is parallelized using a *locally lexicographic* lattice sub-division has been shown to be very efficient for standard Wilson fermions. We demonstrate here the power of this method for the Sheikholeslami-Wohlert improved fermion action.

1 Introduction

Recently, the SSOR preconditioner turned out to be parallelizable by means of the *locally lexicographic* (*ll*) ordering technique [1]. In this way, SSOR preconditioning has been made applicable to the acceleration of standard Wilson fermion inversions on high performance massively parallel systems and it outperforms odd-even preconditioning.

It appears intriguing to extend the range of *ll*-SSOR preconditioners such as to accelerate the inversion of improved fermionic actions, which became very popular in the recent years.

In Symanzik's on-shell improvement program [2], counter terms are added to both, lattice action and composite operators in order to reduce $\mathcal{O}(a)$ artifacts which spoil results in the instance of the Wilson fermion formulation. In the approach of Sheikholeslami and Wohlert (SWA) [3], the Wilson action is modified by adding a diagonal term, the so-called clover term with a new free parameter c_{SW}. The generic form of SWA is given by

$$M = D + A.$$

D represents diagonal blocks (containing 12×12 sub-blocks) and A is a nearest-neighbor hopping term. In the following, we will show that the *ll*-SSOR scheme applies not only to the couplings in A but also to the internal spin and color degrees of freedom of the block diagonal term D.

2 SWA

SWA is composed of A (Wilson hopping term) and D (SW diagonal),

$$D_{SW}(x,y) = \left(I + \frac{c_{SW}}{2} \kappa \sum_{\mu,\nu} \sigma_{\mu\nu} \otimes F_{\mu\nu}(x) \right) \delta_{x,y},$$

P. Amestoy et al. (Eds.): Euro-Par'99, LNCS 1685, pp. 1040–1043, 1999.

$$A_{SW}(x,y) = -\kappa \left(\sum_\mu \left((I-\gamma_\mu)\otimes U_\mu(x)\right)\delta_{x,y-e_\mu} \right.$$
$$\left. + \sum_\mu \left((I+\gamma_\mu)\otimes U_\mu^H(x-e_\mu)\right)\delta_{x,y+e_\mu} \right),$$

where κ is the Wilson hopping parameter, c_{SW} couples the SW clover operator. This parameter is tuned to optimize $\mathcal{O}(a)$ cancellations. The local clover term $F_{\mu\nu}(x)$ consists of 12×12 diagonal blocks. Its explicit structure in Dirac space is given in Ref. [6].

3 Block SSOR Preconditioning

The preconditioned system is modified by two matrices V_1 and V_2,

$$V_1^{-1} M V_2^{-1} \tilde{\psi} = \tilde{\phi},\ \tilde{\phi} = V_1^{-1}\phi,\ \tilde{\psi} = V_2\psi.$$

Let $M = D - L - U$ be the decomposition of M into its block diagonal part D, its (block) lower triangular part $-L$ and its (block) upper triangular part $-U$. Block SSOR preconditioning is defined through the choice

$$V_1 = \left(\frac{1}{\omega}D - L\right)\left(\frac{1}{\omega}D\right)^{-1},\ V_2 = \frac{1}{\omega}D - U\ .$$

The *Eisenstat trick* [8] reduces the costs by a factor 2. It is based on the identity:

$$V_1^{-1}(D-L-U)V_2^{-1} =$$
$$\left(I-\omega L D^{-1}\right)^{-1}\left[I + (\omega-2)\left(I-\omega U D^{-1}\right)^{-1}\right] + \left(I-\omega U D^{-1}\right)^{-1}.$$

The preconditioned matrix-vector product, $z = V_1^{-1} M V_2^{-1} x$, is given by:

```
solve $(I - \omega U D^{-1})y = x$
compute $w = x + (\omega - 2)y$
solve $(I - \omega L D^{-1})v = w$
compute $z = v + y$
```

The "solve" is just a simple forward (backward) substitution process due to the triangular structure:

```
for $i = 1$ to $N$
    $v_i = w_i + \sum_{j=1}^{i-1} L_{ij} s_j$
    $s_i = \omega D_{ii}^{-1} v_i$
```

Options for D of SWA take each block D_{ii} to be of dimension 12 ($D^{(12)}$), 6 ($D^{(6)}$), 3 ($D^{(3)}$) or 1 ($D^{(1)}$), as suggested by the structure of D. The blocks have to be pre-inverted, the cost depends on the block size [6].

Parallelism can be achieved by *locally lexicographic* ordering [1]. “Coloring” is the decomposition of all lattice points into mutually disjoint sets $C_1, \dots, C_k$ (with respect to the matrix M), if for any $l \in \{1, \dots, k\}$ the property $x \in C_l \Rightarrow y \notin C_l$ for all $y \in n(x)$ holds. $n(x)$ denotes the set of sites $\neq x$ coupled to x. A suitable ordering first numbers all x with color C_1, then all with C_2 etc. Thus, each lattice point couples with lattice points of different colors only. The computation of v_x for all x of a given color C_l can be done in parallel, since terms like $\sum_{y \in n(x),\, y \leq_o x}$ involve only lattice points of the preceding colors $C_1, \dots, C_{l-1}$, with $x \leq_o y$ meaning that x has been numbered before y with respect to the ordering o.

Let the lattice blocks be of size $n^{loc} = n_1^{loc} \times n_2^{loc} \times n_3^{loc} \times n_4^{loc}$. A different color is associated with each of the sites of the n^{loc} groups. A *locally lexicographic* (*ll*) ordering is defined to be the color ordering, where all points of a given color are ordered after all points with colors, which correspond to lattice positions on the local grid that are lexicographically preceding the given color. The parallel forward substitution reads:

```
for all colors $C_i$, $i = 1, \dots, \frac{n}{p}$, $\frac{n}{p} \in \mathbf{N}$
    for all processors $j = 1, \dots, p$
        $x :=$ grid point of color $C_i$ on processor $j$
        $v_x = w_x + \sum_{y \in n(x),\, y \leq_{ll} x} L_{xy} s_y$
        $s_x = \omega D_{xx}^{-1} v_x$
```

If the lattice point x is close to the boundary of the local lattice, then the set $n(x)$ will contain grid points y residing in neighboring processors. Therefore, some of the quantities s_y will have to be communicated from those neighboring processors. For SWA, up to 8 neighbors may be involved on the 4-d grid. The detailed communication scheme for this case was given in Ref. [1].

4 Improvement

The SWA has been implemented on an APE100 equipped with $p = 32$ processors. We use a de-correlated set of 10 quenched gauge configurations generated on a 16^4 lattice at $\beta = 6.0$ at 3 values of c_{SW}, 0, 1.0 and 1.769. We have applied BiCGStab as iterative solver. The stopping criterion has been chosen as $\|MX - \phi\| \leq 10^{-6} \|X\|$, with X being the solution. We used a local source ϕ.

We have determined the optimal over-relaxation parameter to be about $\omega = 1.4$ for all block sizes and c_{SW}. In Fig. 1, the results from three diagonal block sizes are overlaid, the 1×1, 3×3, and 6×6 blocks.

We plot the ratio of iteration numbers of the odd-even procedure vs. *ll*-SSOR as function of κ in Fig. 2. A gain factor up to 2.5 in iteration numbers can be found.

There is no dependence on c_{SW} or on the block size of D and only 10 % on the local lattice size. As to real CPU costs on APE100, the optimal block size of D is a 3×3 block whereas on a scalar system, the optimum is found for a 1×1 diagonal.

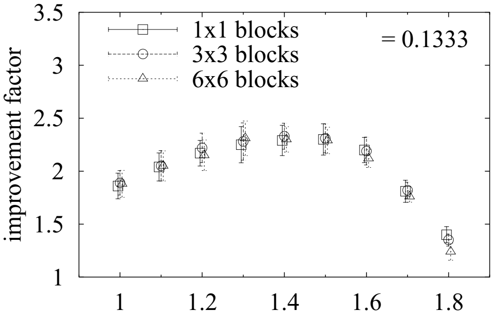

Fig. 1. Gain of ll-SSOR over odd-even preconditioning vs. ω for $c_{SW} = 1.769$.

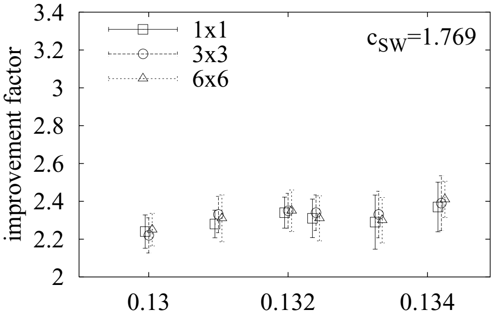

Fig. 2. Gain of ll-SSOR over odd-even preconditioning vs. κ.

References

[1] S. Fischer, A. Frommer, U. Glässner, Th. Lippert, K. Schilling: Comp. Phys. Comm. **98** (1996) 20

[2] K. Symanzik: Nucl. Phys. B **212** (1983) 1

[3] B. Sheikholeslami, R. Wohlert: Nucl. Phys. B **259** (1985) 572

[4] P. Hasenfratz, F. Niedermayer: Nucl. Phys. B **414** (1994) 785

[5] W. Bietenholz, R. Brower, S. Chandrasekharan, U.-J. Wiese: Nucl. Phys. B (Proc. Suppl.) **53** (1997) 921.

[6] W. Bietenholz, N. Eicker, A. Frommer, Th. Lippert, B. Medeke, K. Schilling: hep-lat 9807013, submitted to CPC

[7] W. Bietenholz, U.-J. Wiese: Nucl. Phys. B **464** (1996) 319

[8] S. Eisenstat: SIAM J. Sci. Stat. Comp. **2** (1981) 319