Scalable Parallel SSOR Preconditioning for Lattice Computations in Gauge Theories

A. Frommer[1], Th. Lippert[2*] and K. Schilling[2]

[1]Department of Mathematics, University of Wuppertal, 42097 Wuppertal, Germany
[2]HLRZ, c/o KFA-Jülich, D-52425 Jülich, Germany and DESY, Hamburg, Germany

Abstract. We discuss a parallelization scheme for SSOR preconditioning of Krylov subspace solvers as applied in lattice gauge theory computations. Our preconditioner is based on a locally lexicographic ordering of the lattice points leading to a parallelism adapted to the parallel system's size. By exploitation of the 'Eisenstat-trick' within the bi-conjugate gradient stabilized iterative solver, we achieve a gain factor of about 2 in the number of iterations compared to conventional state-of-the-art odd-even preconditioning. We describe the implementation of the scheme on the APE100/Quadrics SIMD parallel computer in the realistic setting of a large scale lattice quantum chromodynamics simulation.

1 Introduction

Lattice gauge theory (LGT) deals with the controlled numerical evaluation of gauge theories like quantum chromodynamics (QCD) on a 4-dimensional space-time-grid. QCD [1] is considered as the fundamental theory of the strong forces that bind quarks with gluons to form the known hadrons like the proton or neutron. In the low energy regime, QCD cannot be solved by non-perturbative analytical methods. Therefore, numerical simulations become more and more important to provide theoretical input for current and future accelerator experiments that attempt to observe new physics beyond the *Standard Model* of elementary particle physics [2].

The heavy computational demands in LGT are due to repeated solution of a huge system of linear equations,

$$M\psi = (\mathbf{1} - \kappa D)\psi = \phi, \tag{1}$$

with M being the so-called fermion (quark) matrix (that can be considered as analogous to a discretised Laplace equation) of dimension $r = 3 \times 4 \times V$. V is the volume of the underlying 4-dimensional space-time lattice, D contains non-diagonal elements only. The solution ψ of (1), a Green's function, describes the time behavior of the quarks [3]. This Green's function is related to both the simulation of QCD with respect to dynamical quark-gluon interaction and the extraction of physical observables like hadron masses. The size of the solution vector is of order $O(10^7)$ elements in today's state-of-the-art simulations.

* Presented by Th. Lippert

Recently, the BiCGstab method [5] has been established as nearly optimal Krylov subspace method in lattice QCD applications [6], requring only slightly more less iterations than the full GMRES method. This quasi-optimality suggests to turn attention on multigrid methods and/or preconditioning in order to achieve further speed up in numerical methods for solving (1). The application of multigrid techniques, up to now, is impractical, however, due to the gauge noise of the gluonic background field entering the fermion matrix in form of matrix coefficients of the discrete differential operator. This fields represents the gluons in lattice QCD (particles analogous to photons in electrodynamics). Thus preconditioning techniques, *i. e.* methods to decrease the condition of M appear to be the only promising path to further accelerate Krylov subspace solvers like BiCGstab.

A widely used parallelizable preconditioning approach in lattice gauge computations rests upon an odd-even decomposition of the matrix M [7]. It yields an efficiency gain by a factor of 2 when solving (1). In the mid-eighties, Oyanagi [8] proposed to use a certain incomplete LU (ILU) factorization of M as a preconditioner. As it stands, Oyanagi's method works satisfactorily on vector machines. However, on local memory or grid-oriented parallel computers, this preconditioner can hardly be implemented efficiently.

In the present paper we will discuss the parallel aspects of a new SSOR preconditioner for Lattice QCD. We call it the *locally lexicographic SSOR preconditioner* (LL-SSOR). As opposed to multicolor preconditioners (like the odd-even preconditioner) which lead to a decoupling of variables on a very fine grain level, the LL-SSOR approach reduces the decoupling to the minimum which is necessary to achieve a given parallelism. As for any SSOR preconditioner, the Eisenstat Trick [9] is crucial to its efficient implementation.

Our numerical experiments show that LL-SSOR leads to the fastest known solution method on current parallel computers [10], if M represents the widely used standard Wilson fermion matrix. The SSOR preconditioner is applied on the Italian supercomputer APE100/Quadrics and on the Cray T3E within the large scale simulation project SESAM [11, 12].

2 Preconditioning

2.1 Symmetric Gauss-Seidel method

Generally, the preconditioning of (1) proceeds by application of two non-singular matrices V_1 and V_2. They play the role of a left and a right preconditioner, respectively:

$$V_1^{-1} M V_2^{-1} \tilde{\psi} = \tilde{\phi}, \text{ where } \tilde{\phi} = V_1^{-1}\phi, \ \tilde{\psi} = V_2\psi. \tag{2}$$

Any Krylov subspace method could now be applied directly to (2), replacing each occurrence of M and ϕ by $V_1^{-1} M V_2^{-1}$ and $\tilde{\phi}$, respectively.

However, this would yield only the preconditioned iterates $\tilde{\psi}^k$ and preconditioned residuals. Therefore, one usually reformulates the algorithms incorporating an implicit transformation back to the unpreconditioned quantities. For

BiCGstab the resulting algorithm then requires two additional systems with matrix $V = V_1 V_2$ and two systems with matrix V_1 to be solved in each iterative step (see Ref. [5]).

We consider symmetric Gauß-Seidel (SSOR) preconditioning. Assuming that M is scaled to unit diagonal and denoting $M = I - L - U$ with L strictly lower and U strictly upper triangular, the SSOR preconditioner is specified by $V_1 = I - L,\ V_2 = I - U$.

2.2 Eisenstat-Trick

For the SSOR preconditioner the simple identity $V_1 + V_2 - M = I$ holds. This important relation allows to apply the so-called 'Eisenstat-trick' [9]: We can write $V_1^{-1} M V_2^{-1} = V_2^{-1} + V_1^{-1}(I - V_2^{-1})$. Thus the matrix vector product $w = V_1^{-1} M V_2^{-1} r$ amounts to a 2-step solve

$$v = V_2^{-1} r,\ \ u = V_1^{-1}(r - v),\ \ w = v + u. \tag{3}$$

Since the matrices $V_1 = I - L$ and $V_2 = I - U$ are triangular, the solves can be done directly via forward or backward substitution, respectively. In terms of computational cost, a forward followed by a backward solve is approximately as expensive as a multiplication with M (required in the unpreconditioned method).

As a result, the SSOR-preconditioned BiCGstab-method (as well as other Krylov subspace methods) can be implemented with basically the same amount of work per iteration as in the unpreconditioned method. Any occurence of a matrix vector multiply in the unpreconditioned method is replaced by a forward and a backward solve in the preconditioned method, see Algorithm 1.

2.3 Ordering

The generic form of the Wilson fermion matrix M can be seen from the relation

$$(M\psi)_x = \psi_x - \kappa \left(\sum_{\mu=1}^{4} m^{+}_{x,x-\mu} \psi_{x-\mu} m^{-}_{x,x+\mu} \psi_{x+\mu} \right), \tag{4}$$

where x is an index for the (4-dimensional) grid coordinate, and $x \pm \mu$, $\mu = 1, \ldots, 4$ stands for the nearest neighbors in dimension μ on the grid (periodic at the boundaries). At each grid point x we have 12 variables, i.e. $\psi_x \in \mathbf{C}^{12}$, and the coupling matrices $m^{-}_{x,x-\mu}$ and $m^{+}_{x,x+\mu}$ are of the form

$$m^{-}_{x,x-\mu} = (I + \gamma_\mu) \otimes U^{H}_{\mu}(x + \mu), \quad m^{+}_{x,x-\mu} = (I - \gamma_\mu) \otimes U_\mu(x).$$

Here, the $U_\mu(x)$ are 3×3 matrices from SU(3) which represent the gluonic degrees of freedom on the lattice. The matrices γ_μ are 4×4 Dirac matrices.

We have the freedom to choose any ordering scheme for the lattice points x. Different orderings yield different matrices M, permutationally similar to each other. The efficiency of the SSOR preconditioner depends on the ordering scheme chosen. If we assume an arbitrary numbering (ordering) of the lattice

{ initialization }
choose ψ_0, set $r_0 = \phi - M\psi^0$
solve $(I - L)\tilde{r}_0 = r_0$ to get $\tilde{r}_0$ { forward solve }
$\tilde{\hat{r}}_0 = \tilde{r}_0$
set $\rho_0 = \rho_1 = \alpha_0 = \omega_0 = 1$
set $\tilde{v}_0 = \tilde{p}_0 = 0$
{ iteration }
for $i = 1, 2, \ldots$
$\rho_i = \tilde{\hat{r}}_0^\dagger \tilde{r}_{i-1}$
$\gamma_i = (\rho_i/\rho_{i-1})(\alpha_{i-1}/\omega_{i-1})$
$\tilde{p}_i = \tilde{r}_{i-1} + \gamma_i(\tilde{p}_{i-1} - \omega_{i-1}\tilde{v}_{i-1})$
solve $(I - U)z_i = \tilde{p}_i$ to get z_i { backward solve }
solve $(I - L)\tilde{w}_i = \tilde{p}_i - z_i$ to get $\tilde{w}_i$ { forward solve }
$\tilde{v}_i = z_i + \tilde{w}_i$
$\alpha_i = \rho_i/\tilde{\hat{r}}_0^\dagger \tilde{v}_i$
$\tilde{s}_i = \tilde{r}_{i-1} - \alpha_i \tilde{v}_i$
solve $(I - U)y_i = \tilde{s}_i$ to get y_i { backward solve }
solve $(I - L)\tilde{u}_i = s_i - y_i$ to get $\tilde{u}_i$ { forward solve }
$\tilde{t}_i = y_i + \tilde{u}_i$
$\omega_i = \tilde{t}_i^\dagger \tilde{s}_i / \tilde{t}_i^\dagger \tilde{t}_i$
$\psi_i = \psi_{i-1} + \omega_i y_i + \alpha_i z_i$
$\tilde{r}_i = \tilde{s}_i - \omega_i \tilde{t}_i$
end for

Algorithm 1 SSOR preconditioned BiCGstab. Due to the Eisenstat trick, only backward and forward solves are present, no matrix vector multiplications

points, then, considering a given grid point x, we find that the corresponding row in the matrix L or U contains exactly the coupling coefficients of those nearest neighbors of x which have been numbered before or after x, respectively.

Therefore, a generic formulation of the forward solve for this ordering is given by Algorithm 2. The backward solves are done similarly, now running through the grid points in *reverse* order and taking those grid points $x \pm \mu$ which were numbered *after* (instead of *before*) x.

2.4 Odd-even preconditioning

A particular ordering for M is the odd-even scheme where lattice sites are collected in two groups according to their color in a checkerboard-like coloring. The SSOR preconditioner for this scheme hitherto was considered as the only successful preconditioner for lattice QCD in a parallel computing environment. For this particular ordering the inverses of $I - L$ and $I - U$ can be determined

for all grid points x in the given order
  { update y_x }
  $y_x = p_x$
  for $\mu = 1, \ldots, 4$
    if $x - \mu$ was numbered before x then
      $y_x = y_x + \kappa \cdot m^+_{x,x-\mu} y_{x-\mu}$
  for $\mu = 1, \ldots, 4$
    if $x + \mu$ was numbered before x then
      $y_x = y_x + \kappa \cdot m^-_{x,x+\mu} y_{x+\mu}$

for all colors in lexicographic order
  for all processors
    $x :=$ site of that color
    { update y_x }
    $y_x = p_x +$
    $\kappa \left(\sum_{\mu,\, x-\mu \leq_{ll} x} m^+_{x,x-\mu} y_{x-\mu} \right.$
    $\left. + \sum_{\mu,\, x+\mu \leq_{ll} x} m^-_{x,x+\mu} y_{x+\mu} \right)$

Algorithm 2 Generic forward solve and *ll*-forward solve

directly. With the odd-even ordering, the matrix M has the form

$$M = \begin{pmatrix} I & -\kappa D_{oe} \\ -\kappa D_{eo} & I \end{pmatrix} \tag{5}$$

so that

$$I - L = \begin{pmatrix} I & 0 \\ -\kappa D_{eo} & I \end{pmatrix}, \text{ therefore } (I - L)^{-1} = \begin{pmatrix} I & 0 \\ \kappa D_{eo} & I \end{pmatrix}$$

and

$$I - U = \begin{pmatrix} I & -\kappa D_{oe} \\ 0 & I \end{pmatrix}, \text{ therefore } (I - U)^{-1} = \begin{pmatrix} I & \kappa D_{oe} \\ 0 & I \end{pmatrix}.$$

Hence

$$(I - L)^{-1} M (I - U)^{-1} = \begin{pmatrix} I & 0 \\ 0 & I - \kappa^2 D_{eo} D_{oe} \end{pmatrix},$$

where $I - \kappa^2 D_{eo} D_{oe}$ is called the matrix of the odd-even reduced system.

2.5 Lexicographic ordering

The Oyanagi preconditioner considers M to be given with respect to the natural (lexicographic) ordering of the lattice points. This means that grid point $x = (i_1, i_2, i_3, i_4)$ is numbered before $x' = (i'_1, i'_2, i'_3, i'_4)$ if and only if $(i_4 < i'_4)$ or $(i_4 = i'_4$ and $i_3 < i'_3)$ or $(i_4 = i'_4, i_3 = i'_3$ and $i_2 < i'_2)$ or $(i_4 = i'_4, i_3 = i'_3, i_2 = i'_2$ and $i_1 < i'_1)$. Oyanagi [8] showed that SSOR preconditioning for the lexicographic ordering yields a further improvement over odd-even preconditioning as far as the number of iterations is concerned[2]. Unfortunately, the parallel implementation of Oyanagi's method on local memory machines is very difficult and not efficient.

[2] Actually, he employed ILU preconditioning. This turns out to be identical to SSOR in our case, however.

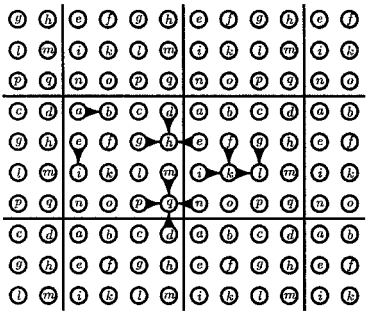

Fig. 1. *Locally lexicographic ordering* and forward solve in 2 dimensions

3 Parallel SSOR Preconditioning

The novel type of ordering we propose is adapted to the parallel computer used to solve equation (1). We assume that the processor connectivity is as a $p_1 \times p_2 \times p_3 \times p_4$ 4-dimensional grid (allowing $p_i = 1$ for certain i). The space-time lattice can be matched to the processor grid in an obvious natural manner, producing a local lattice of size $n_1^{loc} \times n_2^{loc} \times n_3^{loc} \times n_4^{loc}$ with $n_i^{loc} = n_i/p_i$ on each processor. The whole lattice is divided into $n^{loc} = n_1^{loc} n_2^{loc} n_3^{loc} n_4^{loc}$ groups. Each group corresponds to a fixed position of a site in the local grid, and a different color is associated with each of the groups, see Figure 1.

Let us consider the alphabetic ordering of the colors $a - q$ in Figure 1. Such an ordering is termed *locally lexicographic*. All nearest neighbors of a given grid point have colors different from that point. This implies that when performing the forward and backward solves in Algorithm 1, grid points having the same color can be worked upon in parallel, thus yielding an optimal parallelism of p, the number of processors.

A formulation of the *ll*-forward solve is given as Algorithm 2. Here, we use '$\leq_{ll}$' as a symbol for '*ll*-less than'. For grid points lying in the 'interior' of each local grid, we have $x - \mu \leq_{ll} x \leq_{ll} x + \mu$ for $\mu = 1, \ldots, 4$. The update in the forward solve $(I - L)y = p$ thus becomes

$$y_x = p_x + \kappa \left(\sum_{\mu=1}^{4} m^{+}_{x,x-\mu} y_{x-\mu} \right),$$

whereas on the 'local boundaries' we will have between 0 (for the *ll*-first point) and 8 (for the *ll*-last point) summands to add to p_x.

The *ll*-forward and *ll*-backward solves can be carried out in parallel synchronously for the members of each color. The parallelism achieved is p and thus less than with the odd-even ordering but it is optimal since we have p processors. If we change the number of processors, the *ll*-ordering, and consequently the properties of the corresponding LL-SSOR preconditioner will change, too.

4 Results

Our numerical tests of the locally lexicographic SSOR preconditioner (LL-SSOR) have been performed on APE100/Quadrics machines, a SIMD parallel architecture with next-neighbor connectivity. We had access to a 32-node Quadrics Q4 and a 512-node Quadrics QH4.

In Fig. 2a, three different iteration numbers for a given thermalised QCD background field configuration corresponding to 3 different quark masses are given. The standard BiCGstab solution of (1) is compared with the odd-even preconditioned method and the new locally lexicographic ordering scheme. The measurements are carried out on a lattice of size $8^3 \times 16$, hence the computational problem is of granularity $g = N/p = 256$ on the Quadrics Q4 with 32 nodes. The gain in iterations is more than a factor of 2 for the κ-parameter values cho-

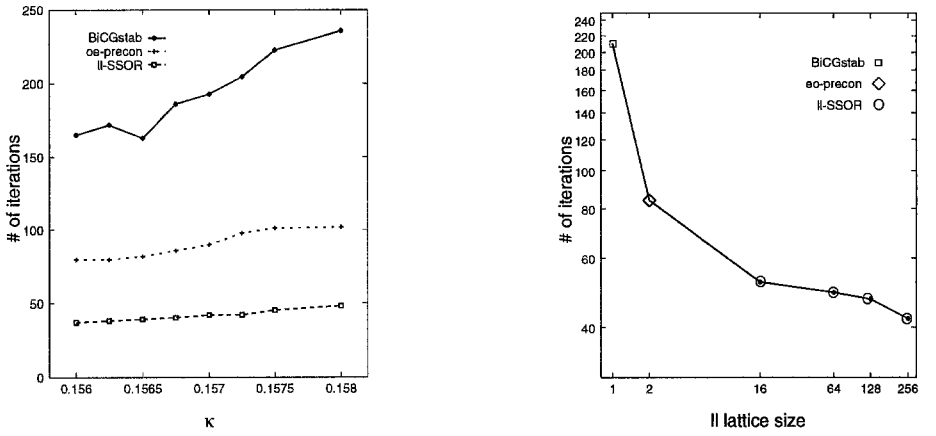

Fig. 2. Iteration numbers of unpreconditioned BiCGstab •, the odd-even- + and the *ll*-preconditioned □ version for a series of condition numbers on a $8^3 \times 16$ lattice. The second plot shows the dependence of iteration numbers on the local lattice size.

sen from a realistic setting and this translates into slighly smaller but similar gains in run time on the Q4. Compared to the unpreconditioned BiCGstab the gain is even a factor of 4. The convergence stopping criterion was to test for $||r||_2/||\phi||_2 < \epsilon$, where r is the *preconditioned* residual in the case of the preconditionmed methods. In either method we took $\epsilon = 10^{-8}$, for the preconditioned methods we also verified that the unpreconditioned residuals $Mx - \phi$, calculated explicitly from the solution x, also satisfied the stopping criterion.

The dependence of the iteration numbers on the local lattice size is depicted in Fig. 2b. Here the hopping parameter (controlling the condition number and the quark mass) of the matrix M is $\kappa = 0.1575$, closer to the critical value for which M becomes singular. The smallest local lattice size we can investigate within LL-SSOR is a 2^4 lattice. The unpreconditioned case can be interpreted as

having local lattice size 1 and the odd-even preconditioning having local lattice size 2. As expected, smaller local lattices lead to a less efficient preconditioning. Eventually, for local= global lattice size, Oyanagi's results can be recovered.

5 Outlook

The parallel preconditioning scheme presented here has been proven very efficient in a large scale lattice QCD simulation, the SESAM project. Currently, we are investigating the application of LL-SSOR preconditioning to so-called improved actions, *i. e.*, new discretisation schemes that are able to improve lattice discretisations on the quantum level.

Acknowledgments. The work of Thomas Lippert is supported by the Deutsche Forschungsgemeinschaft DFG under grant No. Schi 257/5-1.

References

1. P. M. Zerwas and H. A. Kastrup (edts.), *QCD 20 years later* (World Scientific, Singapore, 1992).
2. Particle data group, Phys. Rev **D54** (1996) 1.
3. Th. Lippert, K. Schilling, and N. Petkov, 'Quark Propagator on the Connection Machine', Parallel Computing **18** (1992) 1291.
4. A. Frommer, V. Hannemann, Th. Lippert, B. Nöckel, and K. Schilling, 'Accelerating Wilson Fermion Matrix Inversions by Means of the Stabilized Biconjugate Gradient Algorithm' Int. J. Mod. Phys. **C5** (1994) 1073.
5. van der Vorst, H.: BiCGstab: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Non-symmetric Linear Systems, SIAM J. Sc. Stat. Comp. **13** (1992) 631–644.
6. Frommer, A., Hannemann, V., Nöckel, B., Lippert, Th. and Schilling, K.: Accelerating Wilson Fermion Matrix Inversions by Means of the Stabilized Bi-conjugate Gradient Algorithm, Int. J. of Mod. Phys. **C5** (1994) 1073–1088.
7. T. DeGrand and P. Rossi, Comp. Phys. Comm. **60** (1990) 211.
8. Oyanagi, Y.: An Incomplete LDU Decomposition of Lattice Fermions and its Application to Conjugate Residual Methods, Comp. Phys. Comm. **42** (1986) 333.
9. S. Eisenstat, 'Efficient Implementation of a Class of Preconditioned Conjugate Gradient Methods', SIAM J. Sci. Stat. Comput. **2** (1981) 1.
10. S. Fischer, A. Frommer, U. Glässner, S. Güsken, H. Hoeber, Th. Lippert, G. Ritzenhöfer, K. Schilling, G. Siegert, A. Spitz, 'A Parallel SSOR Preconditioner for Lattice QCD', in C. Bernard, M. Golterman, M. Ogilvie, and Jean Potvin, (edts.): Proceedings of *Lattice '96*, Nucl. Phys. **B** (Proc. Suppl.), **53** (1997) 990-992.
11. N. Attig, S. Güsken, P. Lacock, Th. Lippert, K. Schilling, P. Ueberholz and J. Viehoff, 'Highly Optimized Code for Lattice Quantum Chromodynamics on Cray T3E', accepted for Parco 97.
12. SESAM-Collaboration: U. Glässner, S. Güsken, H. Hoeber, Th. Lippert, X. Luo, G. Ritzenhöfer K. Schilling and G. Siegert: 'QCD with Dynamical Wilson Fermions – First Results from SESAM', in T. D. Kieu, B. H. J. McKellar, and A. J. Guttmann, (edts.): Proceedings of *Lattice '95*, Nucl. Phys. **B** (Proc. Suppl.) **47** (1996) 386-393.