Preconditioning the Kogut-Susskind fermion matrix

J. F. McCarthy
Department of Physics, Indiana University, Bloomington, Indiana 47405

(Received 22 September 1988; revised manuscript received 12 December 1988)

Preconditioning has been successfully used to accelerate the conjugate-gradient algorithm for inverting the lattice Dirac operator in the Wilson formulation. However, technical problems have arisen when trying to apply it in the Kogut-Susskind formulation. We discuss attempts to circumvent the technical problems and to find a successful way to precondition the Kogut-Susskind fermion matrix.

I. INTRODUCTION

The most time-consuming part of numerical simulations of lattice gauge theory involves solving systems of equations of the form

$$M^{\dagger}Mx=M^{\dagger}b \ ,$$

where M is the Dirac operator for x, and b is a source vector. This is usually done using the conjugate-gradient (CG) algorithm. Recently, attempts have been made to accelerate the basic CG algorithm by means of "preconditioning."[1-5] The idea is to find a matrix $\tilde{M}$ which is a reasonable approximation to M but which is easy to invert and then to solve the system of equations

$$(\tilde{M}^{\dagger}\tilde{M})^{-1}M^{\dagger}Mx=(\tilde{M}^{\dagger}\tilde{M})^{-1}M^{\dagger}b \ .$$

If $\tilde{M}$ really is a good approximation to M then $(\tilde{M}^{\dagger}\tilde{M})^{-1}M^{\dagger}M$ will be nearly diagonal and its condition will be much better than that of $M^{\dagger}M$ so the CG algorithm will converge in far fewer iterations.

One way to find a good preconditioning matrix for M is by a method called incomplete Cholesky decomposition. An incomplete Cholesky decomposition of a sparse matrix M is a decomposition into lower and upper triangular matrices, L and U, which are themselves sparse but whose product is only an approximation to M. A simple choice for L is to let it have the same sparsity pattern as the lower part of M; indeed, to give it the same elements as M below the diagonal and just to change the diagonal elements in such a way that the product LU will have the desired diagonal. It is this form of L which has so far been used in preconditioning the Wilson fermion matrix. A problem arises when preconditioning the Kogut-Susskind matrix because it is not diagonally dominated. Its diagonal elements are equal to $2m$, where m is the fermion mass, while its off-diagonal elements are SU(3) matrices. This leads to the incomplete Cholesky method becoming unstable.

Now, consider the matrix $\Theta\equiv M^{\dagger}M$. Its diagonal elements are equal to $8+4m^2$ and its off-diagonal elements are products of two SU(3) matrices, so it is diagonally dominated and we can try to precondition it directly.

This Brief Report describes various attempts to find a good preconditioning matrix for Θ. As in the Wilson fermion case, it proved to be possible to find a preconditioning matrix $\tilde{\Theta}$, which would reduce the number of iterations required in a typical implementation of the conjugate-gradient algorithm by a factor of about 3. However, the computational overhead per iteration was such as to reduce the advantage gained to a factor of about 10%.

II. PRECONDITIONING

To define M and Θ more precisely we must first assign an ordering to the sites of the lattice. For example, if we let the site i have components $i=(i_x,i_y,i_z,i_t)$ then we can adopt the usual convention of writing

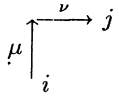
$$i=\{[(i_t-1)n_z+i_z-1]n_y+i_y-1\}n_x+i_x \ , \tag{1}$$

where n_x, n_y, n_z, and n_t are the linear extensions of the lattice in the x, y, z, and t directions. M can be defined in terms of 3×3 matrices connecting sites i and j of the lattice:

$$\begin{aligned} &M_{ij}=2mI \quad \text{if } j=i \ , \\ &M_{ij}=\pm(-1)^{n_{i\nu}}U_{ij} \quad \text{if } j=i\pm\hat{\nu} \ , \\ &M_{ij}=0 \quad \text{otherwise} \ , \end{aligned} \tag{2}$$

where m is the fermion mass, I is the 3×3 unit matrix, U_{ij} is an SU(3) matrix, and $\eta_{i\nu}$ gives the usual Kogut-Susskind phase at the site i in the $\hat{\nu}$ direction.

Clearly M has 8 off-diagonal nonzero elements per row. Θ, on the other hand, has 32 off-diagonal nonzero elements per row. These can be graphically represented as

(i) 

$$(-1)^{\eta_{i\mu}}(-1)^{\eta_{j\nu}}U_{i,\hat{\mu}}U_{i+\hat{\mu},\hat{\nu}}\begin{cases}-1 & \text{if } \hat{\mu},\hat{\nu}=++ \text{ or } -- \\ +1 & \text{if } \hat{\mu},\hat{\nu}=+- \text{ or } - \ ,\end{cases}$$

and

$$\text{(ii)}\quad i \xrightarrow{\mu}\xrightarrow{\mu} j \qquad -(-1)^{\eta_{i\mu}}(-1)^{\eta_{j\mu}} U_{i,\hat{\mu}} U_{i+\hat{\mu},\hat{\mu}} \ .$$

The diagonal elements of Θ are $(8+4m^2)I$.

Unlike M, Θ contains next-nearest-neighbor terms. Among other things, this means that if we adopt the usual procedure for computing the lower triangular matrix L of the incomplete Cholesky decomposition then the resulting preconditioning matrix, call it $\tilde{\Theta}$, will no longer have the same elements as Θ in the positions where $\Theta_{ij} \neq 0$. Also, $\tilde{\Theta}$ will have more fill-in in places where $\Theta_{ij}=0$. However, since the correction terms and the fill-in are small, we will still obtain a reasonably good approximation to Θ and more sophisticated preconditioning procedures can be tried to see what effect they have.

Let $\tilde{\Theta}=LDU$ be an incomplete Cholesky decomposition of Θ, where D is a diagonal matrix introduced for convenience. Since Θ is Hermitian, we have $\Theta_{ik}^{\dagger}=\Theta_{ki}$ and $U=L^{\dagger}$. As a first approximation for L we could try

$$L_{ij}=\Theta_{ij} \quad \text{for } i>j \ , \tag{3}$$

$$L_{ii}=\Theta_{ii}-\sum_{k=0}^{i-1} \Theta_{ik}\Theta_{ki}D_k \ , \qquad D_i=(L_{ii})^{-1} \ . \tag{4}$$

But consider the product $\Theta_{ik}\Theta_{ki}$ in the expression for computing the diagonal elements. There are two cases: (i) if Θ_{ik} is of the form $\rightarrow\!\rightarrow$ then

$$\Theta_{ik}\Theta_{ki}=I \ ;$$

(ii) if Θ_{ik} is of the form $\uparrow\!\!\rightarrow + \rightarrow\!\!\uparrow$ then

$$\Theta_{ik}\Theta_{ki}=2I+2\,\square \ .$$

In case (ii) the result is not a multiple of the unit matrix so, in accordance with our prescription that L have the same sparsity pattern as Θ, we would have to discard the off-diagonal elements. However, this would break gauge invariance so we choose instead to average over the diagonal elements: i.e.,

$$\Theta_{ik}\Theta_{ki} \rightarrow 2I\left(1+\tfrac{1}{3}\mathrm{Re}\,\mathrm{Tr}\ \square\right) \ .$$

Alternatively we could use a global parameter for the plaquette term. This introduces a tunable parameter into the system, which has proven to be of great benefit in previous calculations of this kind. The above prescription for the preconditioning matrix is called LDU_BAS (for "basic") throughout our paper.

As an improvement on LDU_BAS we have also tried using a preconditioning matrix which includes some correction for the next-nearest-neighbor terms. This is achieved by replacing (3) and (4) above with

$$L_{ij}=\Theta_{ij}-\sum_{k=0}^{j-1} L_{ij}D_k U_{kj} \quad \text{for } i \geq j \ . \tag{3'}$$

We call this LDU_CORR (for "corrected").

III. COMPUTATIONAL PROCEDURE

We used lattices of size $6^3 \times 8$ and $8^3 \times 12$. The external gauge field configurations were generated using the hybrid-molecular-dynamic algorithm with four Kogut-Susskind flavors, periodic boundary conditions for the gauge field and periodic in space and antiperiodic in Euclidean time for the fermions. The mass of the dynamical fermions, for all the configurations on which we tried our inverter, was always $m=0.1$.

For every iteration we computed the residual $R_n=b-\Theta x_n$, where x_n is the nth iterate of the conjugate-gradient algorithm and b is the source vector. For b we used a delta function in Dirac and color space. Our condition for convergence was

$$\epsilon_n \equiv (R_n^{\dagger}R_n)^{1/2} < 0.000\,05 \ .$$

We present our results as graphs of ϵ_n vs n. This way to compare different algorithms is as good or bad as any other one. One would think that the bottom line is the amount of computer time, but such a number is strictly hardware dependent and very much a function of the details of the program. Since in this respect we did not attempt any optimization, CPU's comparison would be meaningless. We are particularly interested in the ratio of the total number of iterations required for convergence of the preconditioned and unpreconditioned algorithms. We observe how this depends on the volume V, on the coupling β, on the valence-quark mass m (note that we change the value of m on the diagonal of M but keep the gauge configuration, and hence the *dynamical* quark mass, fixed), and on the type of preconditioning used.

IV. RESULTS

Our most extensive computations were done on $8^3 \times 12$ lattices. The results for $6^3 \times 8$ lattices were very similar and we do not report them. A typical graph which we

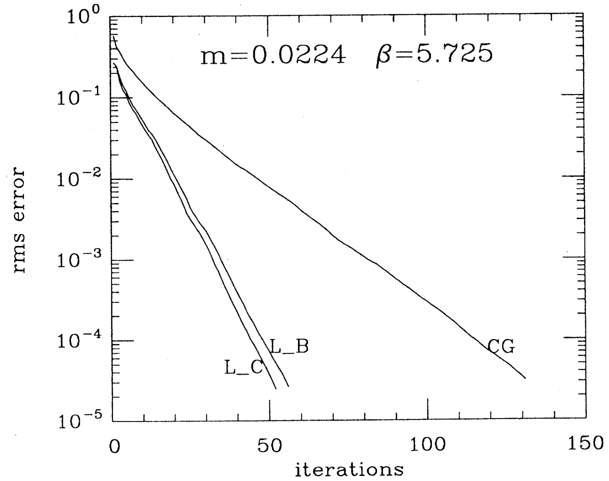


FIG. 1. ϵ_n vs n for CG, LDU_BAS, and LDU_CORR at $\beta=5.725$ and $m=0.0224$.

TABLE I. Total number of iterations required for convergence by CG and LDU_CORR on gauge configurations at various β and $m=0.0224$, and the ratio of total iterations.

β	N_{CG}	$N_{\mathrm{L\_C}}$	Ratio
5.000	533	189	2.820
5.100	522	183	2.852
5.200	502	176	2.852
5.300	467	162	2.883
5.375	460	161	2.857
5.425	446	158	2.823
5.475	311	110	2.827
5.525	344	120	2.867
5.725	148	52	2.846

TABLE II. Minimum eigenvalue of Θ for different β with $m=0.0224$.

β	$\lambda_{\min}$
5.200	0.004 12
5.375	0.004 44
5.475	0.004 62
5.525	0.020 6
5.575	0.038 6
5.725	0.053 4

obtained is shown in Fig. 1. It shows a comparison between the unpreconditioned conjugate-gradient algorithm, CG, and the preconditioned algorithms, LDU_BAS and LDU_CORR, for $\beta=5.725$ and $m=0.0224$. Clearly, LDU_CORR takes slightly fewer iterations than LDU_BAS and about a factor of 3 times fewer than CG.

Table I shows the total number of iterations required for convergence by CG and LDU_CORR and the ratio of these quantities for different β at $m=0.0224$. Note how the total number of iterations required by both algorithms decreases steadily as β crosses the finite-temperature phase transition, which we estimate to occur around $\beta=5.5$. However, the ratio of iterations remains remarkably constant above and below the phase transition.

We also investigated the effect of varying the mass with a fixed gauge configuration at $\beta=5.30$. As expected, the total number of iterations required decreases as the mass increases but still the ratio appears to change very little.

The rate of convergence of the conjugate-gradient algorithm is determined by the eigenvalue spectrum of the matrix one is trying to invert. The whole point of preconditioning is to try to reduce the condition number of the matrix (i.e., the ratio of highest to lowest eigenvalues) and to make its eigenvalues more clustered. To investigate the effect of preconditioning on the condition number, we have calculated the highest and lowest eigenvalues of Θ and $\tilde{\Theta}^{-1}\Theta$ for gauge configurations at different β and m using the power method (see, e.g., Ref. 6).

For Θ, we found that the highest eigenvalue stayed constant at around 20.0±0.1 for all β. Table II shows our results for the value of the lowest eigenvalue to three significant figures. Notice how the lowest eigenvalue rapidly increases as β crosses the transition point. This means that the condition number decreases, and hence the conjugate-gradient algorithm converges in fewer iterations.

For $\tilde{\Theta}^{-1}\Theta$, the highest eigenvalue is about 1.0±0.05 in all cases. We found that the lowest eigenvalue varied in such a way that the ratio of condition numbers for Θ to $\tilde{\Theta}^{-1}\Theta$ stayed between the bounds of 10.0±0.8. This corroborates the remarkable constancy of the value of the ratio of iterations required for convergence by the unpreconditioned and preconditioned algorithms at different β and m.

V. CONCLUSION

The most time-consuming part of one conjugate-gradient iteration is the matrix-vector multiplication. The multiplication $M^{\dagger}(Mp_n)$ involves $2\times9\times3\times(V/2)$ floating point operations per row. For the preconditioned algorithm, $\tilde{\Theta}$ has 33 nonzeros per row so the corresponding figure for $\tilde{\Theta}^{-1}r_n$ is $33\times3\times(V/2)$. In addition, there are two vector-vector multiplications per iteration. So we can say that the preconditioned algorithm takes slightly less than $(33+18+2)/(18+2)$ or about 2.6 times as long per iteration as the unpreconditioned algorithm. The value of the ratio of iterations required for convergence by the unpreconditioned and preconditioned algorithms is about 2.85 for $8^3\times12$ lattice. For the $6^3\times8$ lattices it was closer to 2.6. Hence, there is some small improvement with increasing lattice size but still the net gain is negligible.

In conclusion, we have tried several different ways of preconditioning the Kogut-Susskind function matrix and all seem to suffer from the same fundamental problem—what is gained in a decrease in conjugate-gradient iterations is paid for by the necessity of performing extra operations per iteration—a defect which we do not believe can be overcome by more efficient programming. The effects of varying the coupling, mass, and lattice size are negligible.

ACKNOWLEDGMENTS

I would like to thank S. Gottlieb for his considerable help, R. Sugar for suggesting the project to me, and D. Toussaint, W. Liu, and R. Renken for discussions. This research was supported in part by the United States Department of Energy under Contract No. DE-AC02-84ER40125.

[1] S. A. Gottlieb, W. Liu, R. L. Renken, R. L. Sugar, and D. Toussaint, Phys. Rev. D **35**, 2531 (1987).

[2] For a recent review, see *Field Theory on the Lattice,* proceedings of the International Symposium, Seillac, France, 1987, edited by A. Billoire *et al.* [Nucl. Phys. B, Proc. Suppl. **4** (1988)].

[3] Y. Oyanagi, Comput. Phys. Commun. **42**, 333 (1986).

[4] P. Rossi, C. T. Davies, and G. P. Lepage, Nucl. Phys. **B297**, 287 (1988).

[5] T. A. DeGrand, University of Colorado Report No. COLO-HEP-176, 1988 (unpublished).

[6] V. N. Fadeeva, *Computational Methods of Linear Algebra* (Dover, New York, 1959).