

Pure Gauge SU(3) Lattice Theory on an Array of Computers

E. Brooks, III,^(a) G. Fox, M. Johnson, S. Otto, and P. Stolorz

High Energy Physics Department, California Institute of Technology, Pasadena, California 91125

and

W. Athas, E. DeBenedictis,^(b) R. Faucette, and C. Seitz

Computer Science Department, California Institute of Technology, Pasadena, California 91125

and

J. Stack

Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801

(Received 18 April 1984)

We report on a pure gauge SU(3) lattice theory computation performed on an array of 64 microprocessors interconnected as a 2^6 hypercube and having the cumulative power of approximately eight VAX11/780's. The availability of a substantial number of computer cycles, coupled with an improvement in the algorithm, made possible a high-statistics determination of the heavy-quark potential on a $12^3 \times 16$ lattice.

PACS numbers: 11.15.Ha, 02.70.+d, 12.35.Ht, 12.40.Qq

The numerical approach to quantum field theories via Monte Carlo lattice calculations appears to hold much promise for the eventual solution of these theories.¹ However, before truly accurate, quantitative results can be achieved, much faster and larger (i.e., more memory) computers will be needed. Factors contributing to the intense computational requirements include the large number of degrees of freedom for even a seemingly modest four-dimensional lattice; the statistical nature of the calculation, leading to slow convergence of physical observables; the need to estimate exponentially small, long-distance correlations; and the critical slowing down of the Monte Carlo procedure as the continuum limit is approached.

Fundamental limits in very-large-scale-integration technologies suggest that significant increases in computer performance will come not from pushing current designs yet further, but instead from new computer architectures utilizing many computers executing in parallel.² A simple design for such a computer is a "homogeneous machine": a regular array of (independent) processors with a small number of interconnections per processor.³ Such a machine has been built at Caltech and consists of 64 processors (based on the Intel 8086-8087) wired as a 2^6 hypercube, with a cumulative power of approximately eight VAX11/780's (or up to 3.2 million floating point operations per second) and a total storage capacity of 8 Mbyte. A hypercube interconnection scheme has advantages over two- or three-dimensional mesh connections, though it does contain these. The current 2^6 machine is isomorphic to

a $4 \times 4 \times 4$ cube with periodic faces, but it is not restricted to this; for example, it can also be used as $4 \times 4 \times 2 \times 2$ or 8×8 meshes (ignoring some communication channels in the latter case), or a 64-processor ring. This structure allows flexibility in the choice of decomposition of the application onto the computer. Beyond meshes, the hypercube is natural for the important fast Fourier transform algorithm, and also, the maximum distance between processors grows only logarithmically with the total number of processors, allowing long-distance communications to proceed rapidly.⁴

This 64-processor machine should be regarded as an experimental proving ground towards the construction of much larger and faster multicomputer systems. Other physics-related projects include the successful Ising-model processor built by Pearson, Richardson, and Toussaint,⁵ while Christ and Terzano are building a two-dimensional mesh computer applicable to lattice gauge theories and capable of much higher performance.⁶ We are also presently designing and building more advanced systems. Calculations with a prototype four-node machine and further details of our techniques have been given elsewhere.⁷

In this Letter, we discuss the application of the 64-node machine to SU(3) pure gauge theory, for which the heavy-quark potential has been calculated. Because of the large number of computational cycles available (~ 2500 h) and also the improvement in the method of obtaining the potential, we have achieved a high-statistics, self-consistent (scales for sufficiently small lattice spacing) result.

Here, issues related to the novel computer and algorithm are discussed; a detailed interpretation of the results is given in a companion Letter.⁸

The calculation was performed on a $12^3 \times 16$ lattice and was decomposed onto the computer in three dimensions, i.e., each processor contained a $3^3 \times 16$ sublattice. The action employed was the simple plaquette or Wilson form. The update algorithm used was a variation of the subgroup-heat-bath method,⁹ in which a heat bath in all three SU(2) subgroups of SU(3) is performed before moving on to the next link. To save on storage, only two columns of a SU(3) link matrix were actually stored, with the third reconstructed when necessary. This turns out not only to save on memory, but also to speed up the calculation, since when a matrix product is computed, the third column need not be found. To further reduce memory requirements, the matrices were stored in a fixed-point, sixteen-bit format. This is possible since the elements of the matrix (real and imaginary parts) vary between -1 and 1. In comparison with higher accuracy representations, this was found to have little effect for the multiplication of two such stored matrices. However, as the Monte Carlo calculation ran, it was observed that the matrices drifted off the SU(3) group manifold rapidly and this effect was corrected by projecting back onto the manifold after every two sweeps.

Programming a four-dimensional gauge theory on a mesh-connected computer can be a horrendous task if not approached correctly. The fundamental quantities, Wilson loops of arbitrary shape, can intersect many processors, and explicitly keeping track of all necessary interprocessor communications is difficult. However, a simple, recursive algorithm was constructed which takes a list of numbers as input (each number giving the direction of one step in the loop) and literally travels around the loop. This is easy to implement on the multi-computer with the one restriction that the same shape loop is calculated by all processors simultaneously (which is the typical situation).⁷ With this method, it was straightforward to implement the update of the lattice and the measurement of loops of arbitrary shape and size.

The efficiency of this algorithm has been found to be 0.97 (during update) and 0.95 (during loop measurement). The efficiency is the fraction of the time that the computer is doing useful work, as opposed to interprocessor communication. Another way of putting it is in terms of speedup: During update, the 64-processor machine runs $0.97 \times 64 = 62$ times faster than a single processor. Such high effi-

ciencies are not due to the interaction being short ranged—in fact, most of the links in each $3^3 \times 16$ sublattice are communicated at least once during an update sweep. Instead, they are a result of the large amount of computation done for each matrix passed. To communicate the matrix requires exchanging three packets of 64 bits, each taking 160 μ s, while use of that information in a matrix multiplication (the typical operation) consists of 175 floating point operations, each taking effectively 30 μ s. This gives, as an estimate of the efficiency, $\frac{5250}{5730} = 0.92$. The actual efficiencies are higher because every multiplied matrix does not need to be communicated and, during update, part of the time is spent in the heat bath application, requiring no communication. The conclusion is that we are not limited to short-ranged interactions; more nonlocal actions (such as the very important renormalization-group improved versions) could be used with little, if any, degradation in efficiency.

A quantity fundamental to the understanding of quantum chromodynamics is the heavy-quark potential. With use of the technique developed by Stack,¹⁰ the potential is found by computing $R \times T$ Wilson loops and extracting the potential, V , from

$$W(R, T) \sim e^{-V(R)T}, \quad (1)$$

for sufficiently large T . Our calculation of $W(R, T)$ was significantly improved by using an idea of Parisi, Petronzio, and Rapuano.¹¹ This method consists of completely integrating out those links to which the loop is most sensitive. Figure 1 illustrates the method for $W(R, T)$. If one denotes by \bar{U} the integral,

$$\bar{U} = \int dU U e^{\beta S(U)}, \quad (2)$$

with the links surrounding U held fixed, then (for

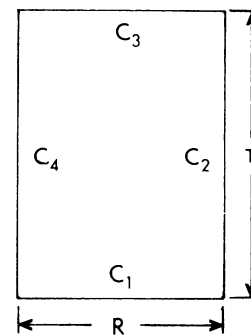


FIG. 1. The Wilson loop, $W(R, T)$. The labeling of the path by C_1, \dots, C_4 refers to Eq. (3).

the simple plaquette action)

$$\langle W(R, T) \rangle = \langle \prod_{c_1} U \prod_{c_2} \bar{U} \prod_{c_3} U^{-1} \prod_{c_4} \bar{U}^{-1} \rangle, \quad (3)$$

for $R \geq 2$. Though the integral in (2) can be found analytically,¹² it is very complex, and it is, perhaps, more practical to estimate the integral by simply updating the link in question many times:

$$\bar{U} = N^{-1} \sum_{i=1}^N U_i, \quad (4)$$

with the index, i , labeling the updates. Despite the fact that this is only an estimate for (2), no systematic error is introduced by using this expression for \bar{U} . This can be seen by inserting (4) into (3) and expanding out the products of sums over updates. Each term then corresponds to an update of the lattice which satisfies detailed balance.

Choosing the N of Eq. (3) to be 15, we find that the improved method reduces the statistical error by approximately a factor of 10, corresponding to an effective gain in computation time of 100. This coupled with the relatively large amount of computer time has meant that even large loops can be found reliably, e.g., the 6×8 loop at $\beta (= 6/g^2) = 6.0$ is measured with a 15% accuracy.

The heavy-quark potential extracted from these loop values is shown in Fig. 2. Also shown is a

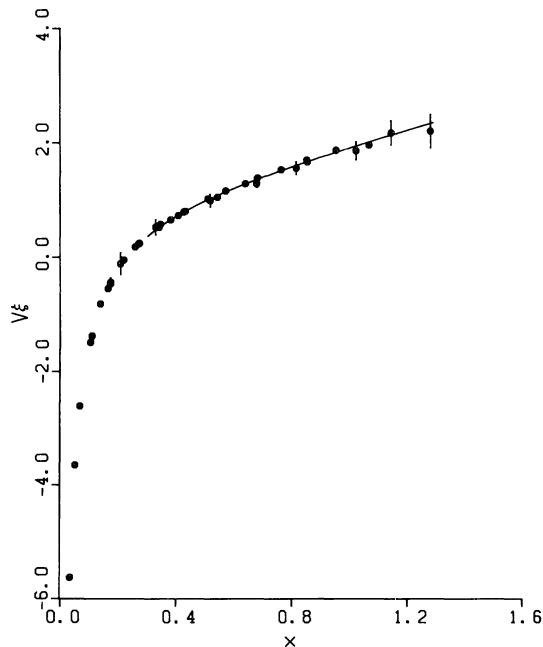


FIG. 2. The heavy-quark potential. Defining the correlation length, ξ , as $\xi = 0.011/\Lambda_L$, the figure shows $V\xi$ vs the distance $x = R/\xi$. Where error bars are not drawn, they are smaller than the symbol size.

(linear plus Coulomb) fit to the large-distance region. We see clearly both the linear (confining) behavior at large distances and the perturbative (Coulomb) behavior at shorter distances. Detailed analysis of this result, such as discussion of the fits and sensitivity to the lattice spacing, will be given in the companion Letter.⁸

As for further applications of the 64-node machine to lattice gauge theories, we are currently using an extension of the methods mentioned here to improve the statistics in a SU(3) glueball spectrum calculation and are preparing renormalization-group [for SU(2)] and dynamical-quark [for SU(3)] applications. A two-dimensional Coulomb gas calculation, with the use of an identical machine with eight nodes, has recently been completed.¹³ Projects in other scientific fields which are being implemented on the 64-node machine include early universe evolution, galactic dynamics, two-dimensional melting, nonlinear wave propagation for geophysics, chemical reaction dynamics, granular flow, and electronic circuit simulation.⁴

We have recently learned of similar methods [to Eq. (3)] which are being applied to the SU(2) case.¹⁴

We would like to thank the Department of Energy, the Defense Advanced Research Projects Agency, the Intel Corporation, the Digital Equipment Corporation, the Ralph M. Parsons Foundation, and the System Development Foundation for much valuable support. This work was supported in part by the U. S. Department of Energy under Agreements No. DE-AC03-81-ER40050 and DE-AT03-83-ER13118, and in part by the National Science Foundation under Grant No. NSF-PHY-81-09494.

^(a)Present address: L-71, Lawrence Livermore National Laboratory, Box 808, Livermore, Calif. 94550.

^(b)Present address: Bell Laboratories, 600 Mountain Ave., Murray Hill, N.J. 07974.

¹M. Creutz, L. Jacobs, and C. Rebbi, Phys. Rep. **95C**, 201 (1983); J. Kogut, Rev. Mod. Phys. **55**, 775 (1983); K. Wilson, Cornell University Report No. CLNS/80/442, 1980 (unpublished).

²C. Seitz and J. Matisoo, Phys. Today, **37** (5), 38 (1984).

³C. Seitz, Journal of VLSI and Computer Systems (to be published).

⁴F. Fox and S. Otto, Phys. Today **37**, (5), 50 (1984).

⁵R. Pearson, J. Richardson, and D. Toussaint, Institute of Theoretical Physics, University of California, Santa Barbara, Report No. NSF-ITP-81-139, 1981 (unpublished).

⁶N. Christ and A. Terrano, Columbia University Re-

port No. CU-TP-261, 1983 (to be published).

⁷E. Brooks, III, *et al.*, Nucl. Phys. **B220** [FS8], 383 (1983).

⁸S. Otto and J. Stack, California Institute of Technology, Report No. CALT-68-1113, 1984 (to be published).

⁹N. Cabibbo and E. Marinari, Phys.Lett. **119B**, 387 (1982).

¹⁰J. Stack, Phys. Rev. D **27**, 412 (1983), and **29**, 1213 (1984); E. Kovacs, Phys. Rev. D **25**, 3312 (1982).

¹¹G. Parisi, R. Petronzio, and F. Rapuano, Phys. Lett. **128B**, 418 (1983).

¹²R. Brower, P. Rossi, and C.-I Tan, Nucl. Phys. **B190** [FS3], 699 (1981).

¹³F. Fucito and S. Solomon, California Institute of Technology Report No. CALT-68-1114, 1984 (to be published).

¹⁴F. Karsch and C. Lang, CERN Report No. TH.3789-CERN, 1983 (to be published).