Parallel Supercomputers for Lattice Gauge Theory

FRANK R. BROWN AND NORMAN H. CHRIST

During the past 10 years, particle physicists have increasingly employed numerical simulation to answer fundamental theoretical questions about the properties of quarks and gluons. The enormous computer resources required by quantum chromodynamic calculations have inspired the design and construction of very powerful, highly parallel, dedicated computers optimized for this work. This article gives a brief description of the numerical structure and current status of these large-scale lattice gauge theory calculations, with emphasis on the computational demands they make. The architecture, present state, and potential of these special-purpose supercomputers is described. It is argued that a numerical solution of low energy quantum chromodynamics may well be achieved by these machines.

DURING THE PAST DECADE, NUMERICAL METHODS HAVE come to be recognized as an essential tool for research in even the most fundamental aspects of elementary particle physics. The quantum chromodynamic (QCD) (*1–3*) theory of the strongly interacting particles has upset the notion that, on the deepest level, the laws of nature must be sufficiently simple and symmetrical as to permit analytical treatment. This theory describes the forces that bind together the quarks making up the neutron, proton, and other subnuclear particles as arising from the exchange of further unseen particles called "gluons." The theory has an extremely elegant geometrical foundation (*4*) but contains strong nonlinearities that have frustrated all analytical approaches tried to date.

However, numerical simulation of QCD with the lattice approximation introduced by Wilson (*5*) has begun to provide important information about the properties of the theory. Unfortunately, vast computer resources are required for this work. Certainly a significant fraction of the supercomputer time now available for academic research is being devoted to these studies. In fact, the computer requirements are so large, and the scientific potential so great, that a number of groups (*6–8*) have begun the construction of special-purpose computers, optimized for these calculations. By exploiting the parallelism inherent in this physics problem, these machines have already achieved a computation rate of 1 gigaflop (Gflop; 10^9 floating point operations per second) and a ratio of cost to performance that is two orders of magnitude better than that of commercial supercomputers. The smallest versions of these devices have been yielding interesting physics results for the past 2 years (*9, 10*).

The authors are in the Department of Physics, Columbia University, New York, NY 10027.

Quantum Chromodynamics—Formulation

The Yang-Mills theory of the strong interactions is an appealing generalization of the very successful field theory of the quantum mechanical interaction of light and electrons: quantum electrodynamics (QED). In QED structureless fundamental particles (electrons) experience electromagnetic forces by absorbing and emitting intermediate particles (photons) which themselves constitute the electromagnetic field. In QCD (by analogy to QED) the electron is replaced by a multiplet of three equal-mass quarks and the single photon by a family of eight spin-one gluons. The familiar invariance of electromagnetism under a change of gauge is generalized to an "$SU(3)$" gauge invariance that fixes the form of the quark-gluon interactions. [Here $SU(3)$ refers to a three-dimensional unitary rotation symmetry among the triplet of quarks.] Just as there are many species of electrically charged particles, there are also many such triplets of quarks distinguished by their masses and other additive quantum numbers conserved by the QCD couplings. In this way one distinguishes the various "flavors" of quarks: up, down, strange, charm, bottom, and presumably top. QCD provides a fundamental theoretical basis for the phenomenologically deduced quark model (*11, 12*).

A serious difficulty facing all relativistic quantum field theories is the problem of divergences—certain of the integrals that need to be calculated go to infinity. Like QED, QCD requires that these infinities be removed by some method of "regularization." This can be done by introducing a regulator into the calculations, which might be, for example, a minimum distance for interaction. Let a denote the length scale at which this regularization is performed. If one is to obtain well-behaved physical quantities when the regulator is removed from the theory, that is as the regulator length, a, is taken to zero, then just as in QED, the coupling strength between quarks and gluons, $g_0(a)$, must be varied as an appropriate function of a as $a \to 0$. For QED the appropriate $a \to 0$ variation of the bare charge, $e_0(a)$ (the QED analog of g_0), is not known, and may not even exist. However for QCD one can show that $g_0^2(a) \approx 1/\log(a^{-1})$ as $a \to 0$ (*2, 13, 14*). Note that $g_0(a)$ vanishes in the continuum limit, making a selective use of perturbation theory—a small g_0 approximation—possible.

This weakening of the effective coupling at short distances, "asymptotic freedom," allows analytical predictions to be made for a number of high-energy phenomena (*15*); in particular the very successful calculation of the ratio R of hadron to muon production in high-energy electron-positron annihilation (*16*). The correct prediction of this ratio as simply the sum of the charges squared of all those quarks with mass lying below the production threshold gives important evidence for the validity of the whole picture, down to the fractional assignment of charges and the triplet character of the quarks' $SU(3)$ representation.

This property of asymptotic freedom has an important implica-

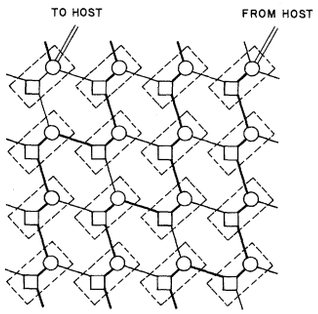

Fig. 1. Our two-dimensional interconnection scheme shown for an array of 16 processors. For the 16-node machine the communication paths are joined from left to right and top to bottom to realize the topology of a torus and to implement the desired periodic boundary condition. The squares represent memory elements and the circles represent processors. The double or darkened paths indicate the data path followed during communication with the host computer.

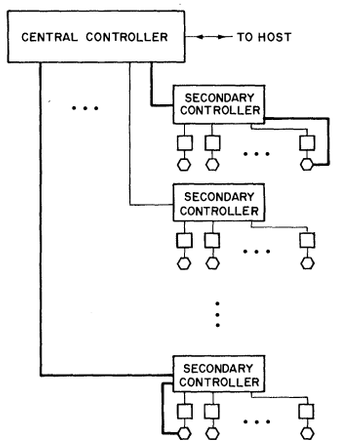

Fig. 2. The elements of the architecture that provide limited central control of the array of processors. In the 64-node machine the central controller drives eight secondary controllers each of which in turn drive eight boards. The individual processor boards are shown as squares and the separate Multibus joined to each board is represented as a hexagon. Again the darkened lines also function to transmit data to and from the host.

tion for lattice gauge theory, a nonperturbative formulation of QCD introduced by Wilson (*5*), which is amenable to numerical study. Here the short distance regularization is provided by a regular, hypercubic lattice in space-time. The short-distance cut off length, a, is now the distance between adjacent lattice sites. As argued by Wilson, the continuum limit is obtained at a second-order critical point of the lattice theory, where physical correlation lengths diverge when measured in lattice units. Asymptotic freedom guarantees that the $g_0 \to 0$ limit is precisely such a critical point. That is, as the lattice coupling constant g_0 is taken to zero, the lattice spacing a also goes to zero when measured in physical units of length. This is, of course, exactly the limit required for the effects of a grid of lattice points, introduced to allow numerical analysis, to become unimportant.

Achieving a sufficiently small lattice spacing is the central computational challenge in lattice QCD. The predictions of perturbation theory for the scaling behavior of physical quantities as $g_0 \to 0$ offer an important consistency check that finite lattice spacing errors are under control. The deviation of a calculation from the expected continuum scaling measures the effects of finite lattice spacing.

How is such a theory formulated? The quark wave functions, $\psi(x)$ (or better, second-quantized field operators), are simply interpreted as taking values on the discrete set of sites x_n making up the space-time lattice. In order to construct gauge covariant differences between quark wave functions on neighboring sites—in analogy with the gauge covariant derivatives of the continuum theory—one introduces the gluonic degrees of freedom as $SU(3)$ matrices U_l associated with each link l in the lattice joining neighboring sites. Thus each U_l is a 3×3 unitary matrix with unit determinant. Finally the theory is made quantum mechanical by writing it as a Feynman sum over histories (*17*).

Because of the intended numerical application, it is important to formulate the theory in thermodynamic terms, that is, as quantum mechanics with Euclidean or imaginary time. For example, the average value of a quantum mechanical observable O at a temperature T is given by the high-dimensional integral

$$\mathrm{tr}[Oe^{-H/kT}] = \int \prod_l dU_l e^{-\mathcal{A}_G} \det(\mathcal{D}) O \qquad (1)$$

The left-hand side of this equation represents the usual quantum mechanical Boltzmann average where H is the quantum Hamiltonian for the system and k is Boltzmann's constant. The right-hand side is a discrete Feynman path integral, an integral over all space-time configurations of the gluon degrees of freedom $\{U_l\}$. The gluon self-interactions are described by the classical gluonic action $\mathcal{A}_G$, a local function of the link variables U_l that depends on a single parameter $\beta = 6/g^2$ specifying the strength of the nonlinearities. The effect of the quark degrees of freedom is represented by the determinant of the lattice Dirac operator, $\det(\mathcal{D})$. This determinant depends on the quark masses and is a highly nonlocal function of the link variables. The lattice of sites x_n and links l on which these variables are defined has a spatial volume that corresponds to that of the quantum system and a temporal extent τ given by $\tau = 1/kT$. By choosing to work with the thermodynamic quantity $\exp(-H/kT)$ we ensure the absolute convergence of the integration in Eq. 1 but lose the ability to directly discuss evolution in physical time. However, approximate knowledge of the Boltzmann operator $\exp(-H/kT)$ can tell us much about the low-energy eigenstates of the quantum mechanical Hamiltonian H. In particular, the properties of the QCD vacuum and single-particle excitations can be directly deduced from the path integral in Eq. 1.

The numerical problem of evaluating the integral Eq. 1 is well defined but truly demanding. It has been approached in two stages. First the so-called "quenched" approximation (18) is made in which the quark determinant $\det(\mathcal{D})$ is replaced by unity—the gluon degrees of freedom are treated as static or quenched while the statistical quark average is performed. The Boltzmann ensemble of gluon configurations is not affected by the quark dynamics. Even with this ad hoc approximation the problem is still quite hard. For example recent calculations (*19*) have used space-time lattices as large as $24^3 \times 48$. With four links per site and eight degrees of freedom in each $SU(3)$ matrix, this implies a gauge integral involving over 10 million variables.

There are well-developed Monte Carlo techniques for numerically performing large-dimensional averages over a positive weight that are commonly used in statistical physics and chemistry (*20*). In these methods the integral is replaced by an average over a finite ensemble of integration points (or configurations) that are distributed according to the weight $e^{-\mathcal{A}}$ where $\mathcal{A}$ is the action. When the gluon action is local (as is the case when the quark determinant is neglected), a new configuration is usually generated from the previous one by sequentially updating each link variable until all link variables have been modified. A variety of updating procedures are effective. For example, in the heat-bath method the update simply thermalizes the link with respect to its environment (the neighboring links with which it interacts are temporarily held fixed) with $e^{-\mathcal{A}}$ as a Boltzmann weight.

Each such update may take approximately 4,000 floating point additions and multiplications. A useful statistical ensemble of 50,000 configurations then requires 2×10^{15} floating point operations for its generation. This is 6 weeks of continuous computation on a 1-Gflop supercomputer (a Cray X-MP/48, for example) running at 50% efficiency for each set of parameters studied. If one wishes to compute the properties of hadrons additional inverse matrices $\mathcal{D}^{-1}$ must be evaluated, typically for one gluon configuration out of 100. Although these matrix inversions can be carried out using efficient techniques appropriate for sparse matrices, an equivalent amount of computer time is required.

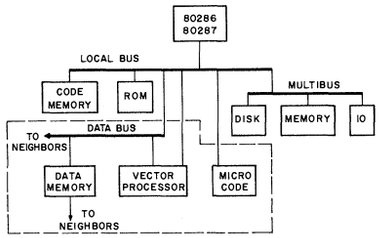

Fig. 3. The configuration of a single node. The data bus is a pair of 16-bit busses for the 16- and 64-node machines, whereas for the 256-node machine it is a single 32-bit bus. The connection of the secondary controller to the Multibus shown here is actually used only by the first and last nodes in the linear chain for data transfer to and from the host computer.

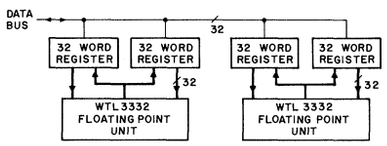

Fig. 4. The architecture of the vector processor for the 256-node machine. The bandwidth between the memory and registers is 32 Mbyte/sec, while the three busses joining one pair of registers with the WTL 3332 floating point unit have a bandwidth of 192 Mbyte/sec. Except for data transfers between the four registers and memory, all other operations of the two paired vector processors must be identical.

The second stage of this program attempts to include the effects of the quark dynamics on the gluon degrees of freedom; the quark determinant $\det(\mathcal{D})$ is not neglected. Of course, exact evaluation of this determinant is hopelessly impractical. However, a number of approximate methods have been developed. Among these are the pseudofermion (*21*), the Langevin (*22*), and the hybrid (*23*) methods. All of these approaches introduce finite step size errors that must be studied numerically and controlled. These calculations are probably two orders of magnitude harder than those in which the quark determinant is neglected.

After stressing the enormous computer requirements of these calculations, we emphasize what makes this problem so attractive: fundamental questions about elementary particle physics are being addressed and all the approximations involved can be well controlled. The relatively simple physical system being studied contains explicitly N_f independent triplets of quark species and eight gluons where typically $N_f = 2$ or 3. There are, however, other particles and interactions that are being neglected.

The omitted electro-weak interactions are suppressed by a factor of the fine-structure constant $\alpha = 1/137$; its omission gives 1% errors. The presence of heavier quarks omitted from the simulation should introduce errors $\approx (M_{\text{light}}/M_{\text{heavy}})^2$. This correction may be as large as 10% for the case of $M_{\text{strange}} \approx 0.5$ GeV and $M_{\text{charmed}} \approx 1.5$ GeV. However, the effect of heavier quarks can be systematically studied by adding them to the calculation. The remaining systematic errors in these calculations all derive from the numerical lattice gauge theory approach: limited Monte Carlo statistics, coarse grid spacing, small physical volume, and finite step size when, for example, performing Langevin integration (*22*). Although these are very serious sources of error for which theoretical bounds are difficult to obtain, they are subject to direct numerical study, and in each case can be made arbitrarily small—at the cost of additional computer time. Thus one hopes to predict within a few percent the properties of the low-energy hadrons in terms of only three parameters: the average of the up and down quark masses, the strange quark mass, and the size of the lattice unit a, which should vary with the coupling g_0 in a known way.

There are, of course, a number of important physical systems for which lattice gauge theory techniques are not well suited. Although Monte Carlo techniques can in principle be applied to systems with a complex action, no practical algorithms exist for cases where the imaginary part of the action is in any sense significant. Such a situation occurs both when QCD is given a nonzero vacuum angle (also called the θ parameter—nearly zero in the real world) and in the presence of a finite chemical potential. The latter situation is unfortunate because finite baryon number density is expected to play an important role in the thermodynamics of heavy-ion collisions. The so-called fermion doubling problem (*24*), the fact that lattice equations describe a system with 16 species of quark, rather than only one, is also a serious difficulty. This problem can be surmounted in cases such as QCD where the fermions occur in matched right-handed–left-handed pairs. However, no solution is known (*25*) for the weak interactions, which involve explicitly left-handed neutrinos, even though it is a very interesting physical system that one would like to study numerically.

Specialized Computers

Given the great demands for computer resources created by these lattice gauge theory simulations, it is natural that the construction of specialized, dedicated machines has become the subject of serious investigation. By exploiting special features of the calculations one expects to gain significant savings. The most important possibility is the incorporation of parallelism. The locality and homogeneity of this physical problem allow the calculation to be easily divided among a large number of processors with relatively simple intercommunication. The size of the computer capable of this work is essentially set by the ~100-Mbyte size of the memory required to hold the 10 million variables referred to above. With parallel construction one has the opportunity to vary the speed and number of individual processors joined to this memory while keeping the computational speed of the entire machine constant.

The ability to optimize by varying the computational power per elementary processor is quite significant. For example a $10-million Cray X-MP/48 with a speed of 1 Gflop gives a performance of 0.1 kflop per dollar, whereas a $500, 16-MHz Weitek WTL 3332 chip with a speed of 32 Mflops yields 64 kflops per dollar. Of course comparing a computer with an isolated component is not very meaningful—more reasonable comparisons are given below.

The utility of constructing special-purpose machines for physics calculations has been amply demonstrated by the three machines built to perform Ising model calculations in the past 5 years (*26–28*). The Ising machine built at AT&T Bell Laboratories by J. Condon and A. Ogielski (*28*) has been especially effective. Lattice gauge theory calculations were first carried out with highly parallel machines by a group at Edinburgh using the Distributed Array Processor (*29*), a 4096-node machine, and by G. Fox, C. Seitz and collaborators on the 64-node Caltech Cosmic Cube (*30*). These calculations demonstrated the ease with which parallelism could be implemented in this work. [A significantly enhanced version of Caltech hypercube machine is now manufactured commercially by Intel as the iSBC-VX (*31*).]

At present there are three groups (*6–8*) constructing parallel machines for these lattice gauge theory calculations:

1) The group at Columbia University (*6*) has constructed a 16-node, 0.25-Gflop machine (operating since 1985) and a 64-node, 1-Gflop machine (completed in February 1987). We are currently testing the first five boards for a 256-node, 16-Gflop machine.

2) A Rome, CERN, Piza, Bologna, Padova collaboration (*8*) completed a 4-node, 0.25-Gflop machine in the fall of 1986 and is currently constructing a 1-Gflop, 16-node machine.

3) A group at IBM Research, Yorktown Heights (*7*) is assembling and testing an 11-Gflop, 576-node machine.

The final version of each of these machines will have on the order of 1 Gbyte of memory.

Probably the most significant feature of our machines at Columbia is their global architecture; each is an $N \times N$ mesh of processors with the simple nearest-neighbor coupling shown in Fig. 1. At present, the two operating machines have $N = 4$ and 8, whereas the 256-node machine will have $N = 16$ as well as the option to be configured with more nodes as a 12×24 mesh. A physical problem posed on an $N_x \times N_y \times N_z \times N_t$ array of grid points can be easily mapped onto such a two-dimensional mesh provided, for example, that N_x and N_y are each multiples of N. Each processor stores the variables and does the computations related to the mesh points in $(N_x/N) \times (N_y/N)$ planes that extend in the z and t directions. We chose two as the smallest number of dimensions that would contain a sufficient number of processors and yet have a linear size no larger than that of the numerical grids we intend to employ. Note that with a large number of processors, the constraints imposed by a particular linear size can be significant. For example, our 16×16, 256-node machine will be naturally suited to simulations on $16^3 \times N_t$ and $32^3 \times N_t$ lattices. It may well be that $16^3 \times N_t$ is too small for definitive results, whereas $32^3 \times N_t$ is too numerically demanding. We have therefore made a point of retaining the 12×24 processor grid as an option, which would permit convenient simulation of $24^3 \times N_t$ lattices.

Figure 1 also shows the distinction between processor and memory elements. These are interconnected so that for any pair of nearest-neighbor memories (shown as boxes) both can be accessed by a single processor (shown as circles). Each processor views the three memories to which it is connected as lying in a continuous address space and it can read from or write to each in the normal way. Thus any computation requiring neighboring operands can be carried out by a single processor without any additional data transfer steps.

The darkened connections shown in Fig. 1 indicate the subset of interprocessor links that are used for communication with the host computer. The first processor in this linear chain reads data written to a central controller by the host computer while the final node in the chain writes data to that same controller which can then be read by the host.

The special association between each processor and one of the memories indicated in Fig. 1 by the dotted boxes has two aspects. First, each paired memory and processor are physically placed on the same printed circuit board. Second, each processor's address lines are routed only to the adjoining memory unit. Thus access to memory of a neighboring node must rely on the processor on the neighboring node to properly address the desired location. This cooperation is easily achieved by having all the processors operating in lockstep, performing symmetrical memory references when off-board data are used. For instance, each processor might simultaneously read data from its northern neighbor while addressing the data in its own memory to be read by the processor to its south. This strategy reflects, of course, the homogeneity inherent in lattice gauge theory calculations.

The second aspect of our global architecture is shown in Fig. 2. Here we show our central controller driving a number of secondary controllers that finally drive the individual nodes. In addition to performing the obvious functions of distributing synchronous clock, reset, and interrupt signals and watching for finish and error signals, this element of central control also permits the resynchronization of the nodes necessary for the synchronous data transfer. After execution of asynchronous data- or node-dependent code, each processor sends a request for resynchronization signal to the central controller that, after all nodes have responded, resynchronizes the machine. Thus our parallelism is achieved through a collection of independent processors that are capable of synchronizing themselves before lockstep synchronous calculation or internode communication.

This very simple intercommunication scheme is the major benefit that comes from focusing on our specialized application. All of the algorithms currently in use in lattice gauge theory calculations can be very efficiently implemented on this architecture. There is essentially no overhead associated with off-board communication, and the 32 Mbyte/sec bandwidth simultaneously sustainable on two of the four cables joined to each node is achieved by inexpensive components representing 5% of the space and 1% of the cost of each board. Although this global architecture is far more restrictive than that implemented on a commercial multiprocessor supercomputer or hypercube-style parallel computer, it is applicable for many physical problems where the nearest-neighbor communication matches the local interactions that underlie all physical processes.

Next, let us turn to the configuration of a single node. As can be seen from Fig. 3, each node is very much like a personal computer or workstation. Controlled by an Intel 80286 microprocessor, augmented by a high-precision 80287 floating point coprocessor, and provided with an industry standard Multibus port, each node can be programmed and enhanced with peripheral memory, hard disk controllers, and so forth, just as is possible for a personal computer. The code memory shown in Fig. 3 varies from 32 kbytes per node on the 16-node machine up to 512 kbytes per node on the 256-node machine. Of course an array of 256 personal computers would miss our computational requirements by three orders of magnitude; the

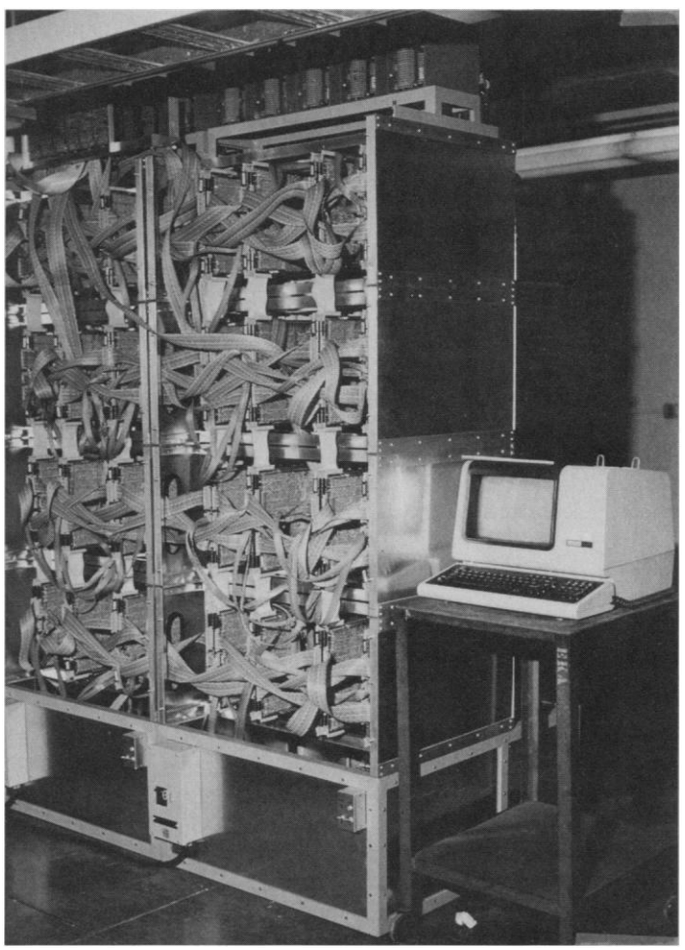

Fig. 5. Photograph of the 64-node Columbia machine. Each wire-wrap board is one of 64 identical processors. The wide ribbon cables provide data paths between neighboring nodes, while the narrow ribbon cables carry control signals between the nodes and the secondary controllers. This machine has the power of roughly six Cray-1 supercomputers and costs about $400,000.

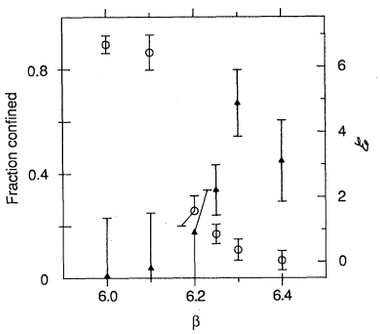

Fig. 6. A $16^3 \times 10$ pure gauge deconfinement calculation performed on the 16-node Columbia machine. The sharp crossover in the fraction confined and $\mathscr{E}$ (a quantity related to the internal energy) demonstrates the presence of a first-order phase transition. The jump in $\mathscr{E}$ across the phase transition gives the latent heat of the transition.

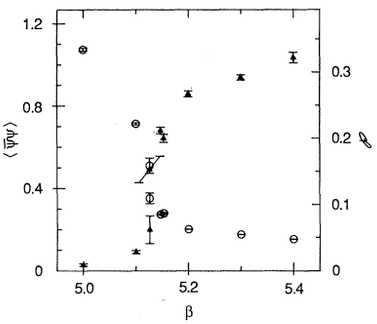

Fig. 7. Data from (*52*) showing a finite-temperature phase transition in the presence of dynamical quarks. The sharp crossover displayed by the order parameter $\mathscr{P}$ and $\langle\bar{\psi}\psi\rangle$ suggests that the transition is first order. The calculations were performed on a $10^3 \times 6$ lattice for four flavors of quark with a mass (in lattice units) of $m = 0.025$.

vector processor shown in Fig. 3 is the essential element in achieving the necessary performance for floating point calculations.

The vector processor incorporated in our 256-node design is shown in Fig. 4. It is composed of a pair of units each of which is made up of a very high bandwidth register file (two Weitek WTL 1066 chips) holding 64 32-bit words and a floating point chip (a Weitek WTL 3332). This latter component contains an adder, multiplier, and an additional 32-word register file, as well as lookup tables to assist in division. Each of the units in Fig. 4 is composed of many stages and run in a standard pipelined mode. In each 62.5-nsec clock cycle, each stage completes a portion of a different calculation taking as its input the output of a different stage generated in the previous cycle. As is common in such designs, the specification of the operation to be performed by each stage and the source and destination for the operands of each stage are determined by a microcode instruction word supplied anew each cycle from a microcode memory.

In typical operation the vector processor and microprocessor operate concurrently. The microprocessor determines the next vector processor routine to be executed and calculates the addresses of the operands that will be required while the vector processor finishes the current routine. Efficient operation is obtained if the microprocessor finishes its preparation for the next vector processor routine and waits for the vector processor to finish its present work. Since the actual restarting of the vector processor takes a fraction of a microsecond, much less than the normal 20-μsec vector processor routine, the vector processor can be made to operate essentially continuously.

The programming of such a special-purpose machine poses a number of different challenges. On the lowest level, we must provide operating software to load data and programs and examine and unload the results of a calculation. This code appears in programmable read-only memory (PROM) on each node and on the host computer, a VAX 11/780. On startup, the PROM code exchanges test data between nodes to verify all cabling, determines the size of the array of processors, and computes the particular input/output path though the array indicated by the darkened connections in Fig. 1. Next, the VAX-resident program sends a series of packets to the machine that are passed from neighbor to neighbor along the indicated linear chain, each processor either copying from, filling, or simply transmitting the passing packet.

This transfer procedure can be invoked by a user at a VAX terminal attempting either to load or extract data or to load and run a program. It can also be used by a program running on the parallel machine that wishes to transfer data to the VAX. Although at first quite simple, this "operating system" has become increasingly sophisticated, absorbing at least one man-year of effort.

The Intel 80286/80287 microprocessor is the most easily programmed part of the machine. Well-documented and robust development software is available that runs on the VAX and produces machine instructions that we load into our array. The majority of these programs are now written in C, although the less convenient language PL/M is used to implement some lower level operating system functions. These C programs make up the higher level parts of our applications programs as well as programs to test both the hardware and the physics programs.

As befits an application that is inherently parallel, the multinode nature of our application programs causes essentially no difficulties. In fact, we typically program our machines in a style that does not depend on the number of nodes. The program currently running on our 64-node machine (*32*), if viewed as a program for a single node, is written for a $3 \times 3 \times N_z \times N_t$ lattice. The parameters N_z and N_t are set at run time with N_z required to be a multiple of six because of vectorization. When run on a single node (with its communication ports connected back on themselves), this is exactly the size of the lattice that is computed. However, if run on the full 8×8 64-node machine with different random number seeds loaded into each node, this same program performs a calculation on a $24 \times 24 \times N_z \times N_t$ lattice.

Some applications deviate from this extremely regular parallelism in a trivial way. A conjugate gradient matrix inversion, for example, requires code to pass around the accumuland for a few nonlocal dot products. Such a routine must consult a data structure set up by the startup PROM program to determine the size of the array. In other cases local communication is a more significant restriction. Fourier transforms, for example, figure prominently in a class of algorithms now under investigation (*33*), and would require careful programming to keep the inevitable loss of efficiency associated with the required nonlocal communication to an acceptable minimum.

By far the greatest programming difficulty lies in programming the vector processor, and especially the vector processor—microprocessor interface. Although the pipelined architecture is similar to that of the floating point units found in commercial supercomputers, we have not developed the analog of the optimizing Fortran compilers available on such machines, which permit applications to be written in a single high-level language. At present the most numerically demanding components of our physics calculations are programmed as follows: First, the calculation must be divided into subroutines. Those that are best done by floating point arithmetic are assigned to the vector processor, whereas tasks that are best done with tables or comparisons are coded in assembly language for the 80286. Second, the routines to be executed by the vector processor are written in a specially developed microcode assembly language and require a companion piece of assembly code used by 80286 to start the vector processor. These 80286 assembly language routines are designed to read the addresses of arguments and inter-subroutine linkage information from a tabulated list of instructions that specify a sequence of microcode subroutines to be executed. The general format and conventions used in this table are reasonably standardized but still must be adjusted somewhat for the application at hand. The third and final step is the actual construction of the table. For a relatively short and straightforward sequence of opera-

tions, such as performing a three-subgroup pseudo-heatbath update (*34*) on a vetor of $SU(3)$ links, the series of random number generations, sine, cosine, square root evaluations, and table lookups can be simply listed by hand. A more complex sequence such as stepping though the links l in memory and identifying the links that make up the six groups needed to update each link l is normally created by a C program run once on the VAX. This multi-layered programming strategy is rather complicated and time-consuming. Nonetheless, each element is quite straightforward, so that with careful planning we routinely design programs that make efficient use of our hardware.

The actual configuration of the 64-node machine can be seen in the photograph in Fig. 5. Fans mounted at the bottom force cooling air upward through the processor boards. Each board measures 12 by 18 inches and consumes 75 W. The entire machine dissipates 10,000 W. The processor boards of our 16- and 64-node machines are constructed with wire-wrap interconnection and cost under $3,000 each. The 256-node boards are eight-layer printed circuit boards costing $4,000 apiece. The total construction costs of the 16-, 64-, and 256-node machines are $150,000, $400,000, and $1.4 million, respectively. (These costs do not include the salaries of the physicists involved in the design, testing, and programming of these machines.)

Present Lattice Gauge Theory Results

Much work remains to be done before the Monte Carlo study of QCD outlined in the first section can be expected to yield physically realistic results. Nonetheless, a number of encouraging successes have been achieved. Our discussion is necessarily incomplete, emphasizing the work with which we have been most involved. Exploratory calculations—some quite ambitious—have been carried out for a large number of physical quantities; noteworthy are hadron mass spectrum studies, both quenched (*18, 19, 35*) and unquenched (*36*), and weak matrix element calculations (*37*). Summaries of current results, as well as many further references, can be found in two recent reviews (*38, 39*).

One of the most striking features of the quark model is the fact that the basic constituents of the model, the quarks and gluons, are not directly observed. This is the phenomenon called confinement. Just as QCD interactions weaken at short distances, they might strengthen at large distances, perhaps sufficiently so that infinite energy would be required to separate a nucleon or meson into the quarks of which it is made. Such behavior can be investigated in the quenched approximation by studying the potential between static quarks.

An early success of numerical lattice gauge theory, originating with Creutz's pioneering $SU(2)$ studies (*40*), was the demonstration of confinement in QCD without quarks ("pure" gauge theory), a result that has become increasingly compelling as larger and larger calculations have been carried out. Confinement can be shown to hold analytically in the strong coupling (large lattice spacing) limit of lattice gauge theory. Numerical calculations provide strong evidence that no phase transition separates the strong coupling limit from the small lattice spacing regime that approximates the continuum limit, and hence that the continuum limit has the same confining behavior seen at strong coupling.

One physically important aspect of QCD—its behavior at finite temperature (*41*)—has been investigated numerically by many groups. The QCD vacuum is expected to undergo a phase transition at finite temperature, above which a new state of matter, the quark-gluon plasma, is produced. This area of study is attractive for two reasons. Euclidean Monte Carlo techniques are inherently well suited to the study of finite temperature quantum field theories, and the quark-gluon plasma may be experimentally accessible in ultrarelativistic heavy ion collisions (*42*). Finite temperature is easily achieved numerically; a simulation performed on a lattice that is periodic in the (Euclidean) time direction with temporal extent $\tau = N_t a$ yields results for the system at a temperature of $T = 1/k\tau$. The finite size of the lattice (in the temporal direction), which is normally regarded as a drawback, is thereby put to good use. A variety of arguments (*41*) suggest that pure gauge QCD will undergo a phase transition at some critical temperature, T_c, above which the theory no longer confines. It is largely to the study of this deconfining phase transition that the first generation (16-node) Columbia machine has been devoted (*9*).

Figure 6 shows results (*9, 44, 45*) typical of recent large-scale confinement calculations. Calculations were performed on a $16^3 \times 10$ lattice for several values of the interaction strength β. By varying β we vary the lattice spacing a measured in physical units and hence the temperature $T = 1/ka(\beta)N_t$. The sharp crossover in the fraction confined (*9*) yields a critical coupling for the phase transition of $\beta_c = 6.160(7)$. The quantity $\mathcal{C}$, which probes the latent heat of the transition (*46*), also jumps abruptly as β is varied. Because this calculation favored a small lattice spacing over a large physical volume, the transition is not as sharp as those in (*43*). Not only is the transition broadened by finite volume effects, also the internal energy jumps at a slightly larger value of β than the fraction confined. Such effects must be controlled by comparing calculations performed for different spatial volumes.

By repeating such large-scale calculations for different values of N_t one is able to demonstrate that the lattice spacings used are small enough to well approximate the continuum. Combining the results of many calculations (*9, 44, 45, 47–50*), we can compare the computed dependence of the critical value of beta β_c on N_t with the theoretical prediction of the continuum theory. Satisfactory agreement is seen for values of N_t between 10 and 16, the largest value studied so far, suggesting that finite lattice spacing errors are reasonably well controlled.

Given that months of 16-node computer time were used by the $16^3 \times N_t$ calculations, it is worth discussing the feasibility of $24^3 \times N_t$ calculations now under way on our 64-node machine. Making the reasonable guess that the cost of such calculations scales like the sixth power of the linear extent of the lattice, we expect the $24^3 \times N_t$ calculations to be an order of magnitude more costly $[(24/16)^6 \simeq 11]$. On the other hand, the 64-node machine offers a factor of 4 increase in power, more efficient programming practices yield an additional twofold increase in the sustained throughput of the machine, while an improved pure gauge Monte Carlo algorithm (*51*) is expected to increase the efficiency of the calculation by at least another factor of 2. Taken together, these improvements in hardware, programming, and algorithms represent a 16-fold increase in performance, leading to the conclusion that the 64-node $24^3 \times N_t$ calculations will be less demanding than were the $16^3 \times N_t$ calculations performed on the 16-node machine.

In the physically realistic situation where light dynamical quarks are coupled to the gluon field the issue of confinement becomes significantly more difficult. First, when separating two colored sources in an attempt to probe the linearly rising, confining potential, at some point it becomes energetically favorable to pair-produce a dynamical quark and antiquark from the vacuum, which then serve to cancel the original color charges, permitting the now colorless composite objects to be separated at no further cost in energy.

Second, the phase transition observed in the absence of quarks need not persist for all values of the quark mass. There is evidence in the literature for the existence of a path in the temperature versus quark mass plane that connects the zero temperature, infinite quark

mass (that is, pure gauge) confined phase with the high-temperature, infinite mass deconfined phase, and yet never passes through a phase transition. Furthermore, for zero quark mass, the theory possesses a so-called "chiral symmetry" (implying the conservation of the quark helicity) that is broken at zero temperature and restored at high temperature. Should a single-phase transition be found in the presence of massless or light quarks, it is best thought of as a chiral symmetry phase transition, and its relation to deconfinement is not a priori clear. Of course the rich structure displayed by the theory in the presence of dynamical quarks makes full-fledged QCD a fertile ground for numerical study.

Finally, as emphasized above, calculations that correctly incorporate the effects of light quarks place vastly greater demands on computational resources than do their pure gauge counterparts. The lighter the quark, the greater its influence on low energy physics, and, not surprisingly, the more costly it makes reliable calculations. Thus early calculations yielded results for relatively heavy quarks, whereas current efforts are directed at pushing the quark mass down low enough to be relevant to the real world in which at least two flavors are to first approximation massless.

Two recent finite temperature studies (*52, 53*) exemplify the current state of dynamical fermion calculations, and serve to illustrate what can be achieved with the commercial supercomputers currently available. Kovacs *et al.* (*52*) and Fukugita *et al.* (*53*) present evidence suggesting the existence of a first-order phase transition for sufficiently small quark mass, and support the same qualitative picture, although different systems were studied. Kovacs *et al.* (*52*) treat four flavors of quark with a mass (in lattice units) of $m = 0.025$ on a $10^3 \times 6$ lattice using the hybrid method, whereas the calculations in (*53*) are for two flavors with mass down to $m = 0.05$ on an $8^3 \times 4$ lattice with the Langevin method.

In Fig. 7, the dynamical fermion analog of Fig. 6, we reproduce data presented in (*52*) for an order parameter $\mathcal{P}$ related to the confined fraction plotted in Fig. 6 and $\langle\bar{\psi}\psi\rangle$, the natural order parameter for chiral symmetry breaking. Both change abruptly, indicating the presence of a phase transition, quite possibly first order. Furthermore, they change together, so that one may picture this phase transition as a single transition at which deconfinement and the restoration of chiral symmetry occur simultaneously.

The work of Kovacs *et al.* (*52*) represents the most ambitious dynamical fermion calculation reported to date. Experience with pure gauge calculations suggests that the $10^3 \times 6$ lattice on which it was performed is sufficiently large to provide a qualitatively reasonable picture of QCD. The same reasoning implies, however, that significantly larger lattices will be necessary to obtain quantitatively accurate predictions for full-fledged QCD. Kovacs *et al.* report that this calculation consumed 5000 hours of CPU time on an ST-100 array processor running with a sustained performance of 50 Mflops. That is, such a calculation would take 3 months on the 16-node and 3 weeks on the 64-node Columbia machine. To duplicate such a calculation on a $16^3 \times 10$ lattice would require (assuming the cost goes like the sixth power of the linear extent of the lattice, and making the pessimistic assumption that significant improvements in algorithms will not be forthcoming) about a year of running on the 64-node Columbia machine, or between 1 and 2 months on a machine with a sustained performance of 5 Gflops, such as may well be achieved by the IBM project (*7*) or the third-generation Columbia machine.

Bearing in mind that such bulk thermodynamic studies are only a relatively easy subset of the calculations one would like to perform, and that larger lattices will be necessary at least for occasional consistency checks, it is apparent that the numerical solution of QCD will easily saturate the powerful parallel supercomputers currently being constructed for this purpose for years to come.

Future Prospects

In the preceding we have described what we expect will become a significant new direction in theoretical physics: the incorporation of advanced computer technology into specially designed machines capable of making real progress on important questions in relativistic quantum field theory. We have emphasized a particular application, the computation from the known QCD action of the low-energy properties of the strongly interacting particles (and the quarks and gluons out of which they are made). Such an endeavor will provide an important quantitative test of QCD, and should serve to establish the utility of lattice gauge theory and the large-scale computation strategy presented here.

Perhaps even more important will be the study of theoretical models whose properties, even qualitatively, are not well understood. For example, the spontaneous symmetry breaking that is believed to distinguish the electromagnetic and weak interactions giving the intermediate $W^{\pm}$ and Z^0 bosons their mass may well occur as a result of a new strongly coupled non-Abelian gauge theory (*54*). Similarly, attempts to understand the increasingly rich spectrum of "fundamental" quarks and leptons often postulate a new, high energy strong interaction whose bound states (*55*) form the known families of quarks and leptons. Both of these speculations posit new non-Abelian gauge couplings that possess the same strong nonlinearities that have plagued all analytic approaches to QCD. These attempts to gain insight into aspects of elementary particle physics that are not understood are just two examples of important but nearly intractable models whose properties are best studied with the techniques discussed here. Thus it appears that even more interesting and difficult problems await the techniques now being developed to study QCD.

REFERENCES AND NOTES

1. H. Fritzsch and M. Gell-Mann, in *Sixteenth International Conference on High Energy Physics*, J. D. Jackson and A. Roberts, Eds. (NAL, Batavia, IL, 1973), vol. 2, p. 135; H. Fritzsch, M. Gell-Mann, H. Leutwyler, *Phys. Lett.* **47B**, 365 (1973).
2. D. J. Gross and F. Wilczek, *Phys. Rev. Lett.* **30**, 1343 (1973); H. D. Politzer, *ibid.*, p. 1346.
3. W. Marciano and H. Pagels, *Phys. Rep.* **36C**, 137 (1978).
4. C. N. Yang and R. Mills, *Phys. Rev.* **96**, 191 (1954).
5. K. G. Wilson, *Phys. Dev. D* **10**, 2445 (1974).
6. N. H. Christ and A. E. Terrano, *IEEE Trans. Comput.* **33**, 344 (1984); *Byte* **11**, 145 (April 1986).
7. J. Beetem, M. Denneau, D. Weingarten, in *The 12th Annual International Symposium on Computer Architecture* (IEEE Computer Society Press, Silver Spring, MD, 1985), p. 108.
8. P. Bacilieri *et al.*, in *Computing in High Energy Physics*, L. O. Hertzberger and W. Hoogland, Eds. (North-Holland, Amsterdam, 1986), p. 330.
9. N. H. Christ and A. E. Terrano, *Phys. Rev. Lett.* **56**, 111 (1986).
10. APE Collab., M. Albanese *et al.*, *Phys. Lett.* **192B**, 163 (1987).
11. M. Gell-Mann, *ibid.* **8**, 214 (1964); G. Zweig, in *Symmetries in Elementary Particle Physics*, A. Zichichi, Ed. (Academic Press, New York, 1965), p. 192.
12. H. J. Lipkin, *Phys. Rep.* **8C**, 173 (1973); F. E. Close, *An Introduction to Quarks and Partons* (Academic Press, London, 1979).
13. S. Coleman and D. Gross, *Phys. Rev. Lett.* **31**, 851 (1973).
14. D. J. Gross, in *Methods in Field Theory*, R. Balian and J. Zinn-Justin, Eds. (North-Holland, Amsterdam, 1976), p. 141.
15. A. H. Mueller, *Phys. Rep.* **73**, 237 (1981).
16. TASSO Collab., M. Althoff *et al.*, *Phys. Lett.* **138B**, 441 (1984).
17. R. P. Feynman and A. R. Hibbs, *Quantum Mechanics and Path Integrals* (McGraw-Hill, New York, 1965).
18. H. Hamber and G. Parisi, *Phys. Rev. Lett.* **47**, 1792 (1981); E. Marinari, G. Parisi, C. Rebbi, *ibid.*, p. 1795; D. Weingarten, *Phys. Lett.* **109B**, 57 (1982).
19. P. de Forcrand *et al.*, preprint WU-B-86-12 [University of Wuppertal, Wuppertal, West Germany, 1986 (unpublished)].
20. *Monte Carlo Methods in Statistical Physics*, K. Binder, Ed. (Springer, Berlin, 1979); *Applications of Monte Carlo Methods in Statistical Physics*, K. Binder, Ed. (Springer, Berlin, 1984).
21. F. Fucito *et al.*, *Nucl. Phys.* **B180[FS2]**, 369 (1981).
22. A. Ukawa and M. Fukugita, *Phys. Rev. Lett.* **55**, 1854 (1985); G. G. Batrouni *et al.*, *Phys. Rev. D* **32**, 2736 (1985).
23. S. Duane, *Nucl. Phys.* **B257[FS14]**, 652 (1985).
24. K. G. Wilson, in *New Phenomena in Subnuclear Physics*, A. Zichichi, Ed. (Plenum, New York, 1977), p. 13.
25. H. B. Nielsen and M. Ninomiya, *Nucl. Phys.* **B185**, 20 (1981); *ibid.* **B195**, 541 (1982); *Phys. Rev. Lett.* **58**, 2515 (1987).

26. R. B. Pearson, J. L. Richardson, D. Toussaint, *J. Comput. Phys.* **51**, 241 (1983).
27. A. Hoogland *et al.*, *ibid.*, p. 250.
28. J. H. Condon and A. T. Ogielski, *Rev. Sci. Instrum.* **56**, 1691 (1985).
29. D. J. Wallace, *Phys. Rep.* **103**, 191 (1984).
30. E. Brooks III *et al.*, *Phys. Rev. Lett.* **52**, 2324 (1984).
31. "Vector processing boosts hypercube's performance," *Electronics*, 14 April 1986, p. 30.
32. Y. Deng, in *Lattice Gauge Theory Using Parallel Processors*, X. Y. Li, Z. M. Qiu, H. C. Ren, Eds. (Gordon and Breach, New York, 1987), p. 419; M. Gao, *ibid.*, p. 369.
33. G. G. Batrouni *et al.*, *Phys. Rev. D* **10**, 2736 (1985).
34. N. Cabibbo and E. Marinari, *Phys. Lett.* **119B**, 387 (1982).
35. S. Itoh, Y. Iwasaki, T. Yoshie, *ibid.* **183B**, 351 (1987); K. C. Bowler *et al.*, *Nucl. Phys.* **B284**, 299 (1987).
36. A. Billoire and E. Marinari, *Phys. Lett.* **184B**, 381 (1987); M. Fukugita *et al.*, *ibid.* **191B**, 164 (1987).
37. C. Bernard *et al.*, *Phys. Rev. Lett.* **55**, 2770 (1985); L. Maiani and G. Martinelli, *Phys. Lett.* **181B**, 344 (1986); S. R. Sharpe *et al.*, *Phys. Lett.* **192B**, 149 (1987).
38. P. Hasenfratz, in *Proceedings of the XXIII International Conference on High Energy Physics*, S. C. Loken , Ed. (World Scientific, Singapore, 1987), vol. 1, p. 169.
39. M. Fukugita, in *Lattice Gauge Theory Using Parallel Processors*, X. Y. Li, Z. M. Qui, H. C. Ren, Eds. (Gordon and Breach, New York, 1987), p. 195.
40. M. Creutz, *Phys. Rev. Lett.* **43**, 553 (1979).
41. D. J. Gross, R. D. Pisarski, L. G. Yaffe, *Rev. Mod. Phys.* **53**, 43 (1981).
42. E. V. Shuryak, *Phys. Rep.* **115**, 151 (1984), chap. 7.
43. J. Kogut *et al.*, *Phys. Rev. Lett.* **50**, 393 (1983); T. Çelik, J. Engels, H. Satz, *Phys. Lett.* **125B**, 411 (1983).
44. N. H. Christ, in *Nonperturbative Methods in Field Theory*, H. W. Hamber, Ed. (North-Holland, Amsterdam, in press).
45. H.-Q. Ding, thesis, Columbia University (1987).
46. J. Engles *et al.*, *Nucl Phys.* **B205[FS]**, 545 (1982); B. Svetitsky and F. Fucito, *Phys. Lett.* **131B**, 165 (1983); D. Toussaint, private communication.
47. A. D. Kennedy *et al.*, *Phys. Rev. Lett.* **54**, 87 (1985).
48. S. A. Gottlieb *et al.*, *ibid.* **55**, 1958 (1985).
49. D. Toussaint *et al.*, in *Lattice Gauge Theory*, H. Satz, Ed. (Plenum, New York, in press).
50. N. H. Christ, in *Proceedings of the International Symposium on Field Theory of the Lattice*, A. Billoire *et al.*, Eds. (*Nucl. Phys.*, in press).
51. F. R. Brown and T. J. Woch, *Phys. Rev. Lett.* **58**, 2394 (1987).
52. E. V. E. Kovacs, D. K. Sinclair, J. B. Kogut, *ibid.*, p. 751.
53. M. Fukugita *et al.*, *ibid.*, p. 2515.
54. E. Farhi and L. Susskind, *Phys. Rep.* **74**, 277 (1981).
55. H. Harari, *ibid.* **104**, 159 (1984).
56. The project described here represents the efforts of a relatively small group of faculty, postdoctoral researchers, and graduate students in the Physics Department at Columbia: A. E. Terrano (now at Rutgers University) and K. Barad (at Barnard College), H.-Q. Ding (now at Caltech), and F. P. Butler, H. Chen, Y. F. Deng, M. S. Gao, P. F. Hsieh, H. Shi (in the Electrical Engineering Department), L. I. Unger, and T. J. Woch in addition to the authors. We thank the Intel Corporation and the U.S. Department of Energy for their support.

Research Articles

Activation of Cell-Specific Expression of Rat Growth Hormone and Prolactin Genes by a Common Transcription Factor

CHRISTIAN NELSON, VIVIAN R. ALBERT, HARRY P. ELSHOLTZ, LESLIE I.-W. LU, MICHAEL G. ROSENFELD

In the anterior pituitary gland, there are five phenotypically distinct cell types, including cells that produce either prolactin (lactotrophs) or growth hormone (somatotrophs). Multiple, related cis-active elements that exhibit synergistic interactions appear to be the critical determinants of the transcriptional activation of the rat prolactin and growth hormone genes. A common positive tissue-specific transcription factor, referred to as Pit-1, appears to bind to all the cell-specific elements in each gene and to be required for the activation of both the prolactin and growth hormone genes. The data suggest that, in the course of development, a single tissue-specific factor activates sets of genes that ultimately exhibit restricted cell-specific expression and define cellular phenotype.

EUKARYOTIC GENES ARE TRANSCRIPTIONALLY REGULATED by protein factors that bind cis-acting promoter and enhancer elements (*1*), some of which exert their actions in a tissue-specific manner (*2*). During the developmental program of organogenesis, there is a serial appearance of phenotypically distinct cell types that exhibit selective patterns of gene expression. Understanding the mechanisms determining the sequential activation of these differentiated states requires the elucidation of factors governing the cell type–specific expression of genes. The expression of two evolutionarily related genes, prolactin and growth hormone (GH), in two phenotypically distinct cell types (lactotrophs and somatotrophs, respectively) of the anterior pituitary gland (*3*) provides a model system for the analysis of cell type–specific gene expression within an organ. During pituitary development the appearance of somatotrophs temporally precedes that of lactotrophs (*4*). The transient coexpression of growth hormone in more than 95 percent of prolactin-producing cells before the appearance of mature lactotrophs (*4*) raises the possibility that these two genes may share a common developmental signal for activation. We now provide evidence that a common tissue-specific transcription factor is required for activation of these two genes expressed in phenotypically distinct cell types.

A common cell-specific factor binds to the prolactin and growth hormone enhancer elements. Tissue-specific enhancers in the 5′ flanking regions of both the prolactin and growth hormone genes appear to dictate their pituitary-specific expression (*5*). We have used deletion mapping and protection from digestion by deoxyribonuclease I (DNase 1) by binding of nuclear proteins (DNase I footprinting analysis) to identify prolactin enhancer

The authors are in the Eukaryotic Regulatory Biology Program, Center for Molecular Genetics, University of California, San Diego, School of Medicine, San Diego, CA 92093. In addition, H. P. Elsholtz and M. G. Rosenfeld are in the Howard Hughes Medical Institute, University of California, San Diego, School of Medicine, and C. Nelson is a graduate student in the Department of Biology, University of California, San Diego.