Transverse Lattice QCD

- Proposer/Project Leader:
  Dr. Brett van de Sande
  Geneva College
  3200 College Ave.
  Beaver Falls, PA 15010

- E-mail: bvds@geneva.edu

- Research field: Physics

- Underlying theory: Quantum Chromodynamics (QCD)

Summary of proposed research, including objectives/goals

The transverse lattice is a non-perturbative procedure for solving the bound state problem of SU(N) gauge theories like QCD. The goal of this work is to calculate various properties of the hadrons, such as structure functions, which are of great experimental interest but are difficult or impossible to calculate using other approaches (such as Euclidean lattice Monte Carlo).

The construction of the transverse lattice is as follows: In the two "transverse" space dimensions a lattice regulator is employed while light-front Hamiltonian techniques are used in the remaining two space-time dimensions. We introduce "link fields" that lie on each link of the transverse lattice. Calculationally, we use a Hamiltonian approach: we create a basis of link fields joined together to form a string. Next, we calculate the Hamiltonian matrix in this basis. Since the Hamiltonian matrix is quite sparse, we can use standard Lanczos techniques to calculate the lowest eigenvalues of the Hamiltonian.

Our Hamiltonian has an effective potential and several couplings must be determined in some way. Our technique is to demand rotational and Lorentz invariance for observables. We "tune" the various coupling constants until the glueball spectrum is rotationally invariant and Lorentz invariant. An essential test of the validity of our approach is to demonstrate the existence and uniqueness of an approximate "scaling trajectory" in coupling constant space.

In 1998, we finished extensive studies for pure gauge theory (no quarks) in 3 space-time dimensions. These results compare favorably with state-of-the-art Euclidean Lattice Monte Carlo results. The numerical portion of this study was completed on PC's running in roughly 4 CPU-weeks.

At the end of 1998, we turned our attention to calculations in 4 space-time dimensions. The initial results for glueballs are competitive (and in agreement with) state-of-the-art lattice glueball calculations. Thus, the major goal of this project is to demonstrate that our results remain invariant if we change lattice spacing of the transverse lattice. In other words, we must demonstrate the existence and uniqueness of an approximate "scaling trajectory" in coupling constant space. We were able to demonstrate this in 3 space-time dimensions and it is essential to demonstrate the same in 4 space-time dimensions. This is necessary to establish the credibility of our results.

In addition to the immediate goal mentioned above, we would like begin development work on future calculations. Future work includes:

- Developing a version of our code with parallelized sparse matrix diagonalization.
- Studies of excited states of the heavy quark potential.
- Include Fermions and calculate the meson spectrum. This will be calculationally challenging due to the increased number of degrees of freedom in the system.
- Studies of supersymmetric field theories in two space-time dimensions.

Computational Methodology/Algorithms

In the following, let us define a "run" to be the determination of the Lorentz invariant Hamiltonian for a given lattice spacing. Essentially, the computing task for each "run" is as follows:

1. For a given lattice spacing, choose values for the couplings in the Hamiltonian (eight in all).
2. (a) Construct various Hamiltonian matrices and
   (b) calculate the lowest eigenvalues using Lanczos. There are about 120 different matrices to be calculated.
3. Determine how well various criteria for rotational invariance and Lorentz invariance are fulfilled.
4. Choose new values of the couplings and goto 2.
5. The process is completed when there is no further improvement to the criteria.

Each iteration of step 2 requires approximately 6 CPU-hours. It takes 50 to 100 iterations to find the best values for the couplings. Thus I need roughly 600 CPU-hours per run.

The search of coupling constant space is performed using a standard χ^2 minimization subroutine LMDIF (available from GAMS).

I use the package DNLASO (available from GAMS) to perform the Lanczos diagonalization. I also tried ARPACK, but the performance was somewhat poorer. The 120 matrices vary widely in size, but the largest are 53,000 by 53,000 with about 2.8 million nonzero matrix elements stored in memory.

Preliminary Results of Progress to Date

Although the transverse lattice was first suggested back in the 1970's, this approach to relativistic field theory has only been explored during the last few years. Some recent publications are:

"Transverse Lattice Approach to Light-Front Hamiltonian QCD," with S. Dalley, in Phys. Rev. D **59**, 065008 (1999).

"Glueball calculations in large-N_c gauge theory," with S. Dalley, in Phys. Rev. Lett. **82**, 1088 (1999).

S. Dalley and B. van de Sande, "Color-Dielectric Gauge theory on a Transverse Lattice", Phys. Rev. D **56**, 7917 (1997).

Our preliminary results for 4 space-time dimensions are shown in the enclosed paper, "Glueball calculations in large-N_c gauge theory." This is my first grant application at a NSF supercomputing center.

Justification of Resources Requested

In order to demonstrate that our technique works, one must show that the spectrum remains invariant as the transverse lattice spacing is changed. Based on our experience in 2+1 dimensions, this will take at least 20 runs (the runs are defined above). (This is a somewhat optimistic estimate; it assumes that nothing goes wrong!) We believe that 33 would give us a sufficient margin of safety.

Thus, we need an absolute minimum of 12000 CPU-hours to complete the project, 20000 CPU-hours would give us a reasonable margin of safety.

Time frame and workplan

I hope to start work on the project in August 1999 and complete the 4 dimensional results within the course of a year. I have already tested and tuned the code on other IBM SP2 machines (at LRZ-Mnchen and at Argonne) as well as on the CRAY T3E at the Ohio Supercomputing Center. Thus, the bulk of the project will be running the code.

Memory use

The only significant memory user is the memory used to store the nonzero matrix elements of the Hamiltonian matrix. Since the matrices are sparse and symmetric, only nonzero elements are stored using "symmetric column pointer format." The total memory to store the matrix is about 50 Megabytes. The total memory needed per processor is about 70 Mbytes. Future work will be more memory limited, forcing one to go to parallelized generation and storage of the Hamiltonian matrix.

Parallelization

Our present work does not depend critically on inter-processor communication, but future work will require that we parallelize the code for producing and diagonalizing the Hamiltonian matrices. This is particularly true for theories including Fermions. In that case, we will need to have the fast inter-processor communication provided for by a machine like the CRAY T3E or IBM SP2.

Processor efficiency

The only significant time consumers in our computations, each taking roughly half the CPU time, are the routines to construct the Hamiltonian matrix and the calculation of sparse-matrix-vector products in the Lanczos routine.

Below, we list the time needed for generating and then diagonalizing a Hamiltonian matrix in a test case (single processor performance):

Platform	generate sec.	diagonalize sec.
Power PC's with LINUX and GNU C	35	70
IBM SP2	31	21
Convex SPP	52	62
Hitachi	60	69
Fujitsu vpp 300	82	20
Cray T3E	25	42

In this test case, the matrix dimension was 14439 by 14439 and had 639674 nonzero matrix elements in the upper triangle. The lowest five eigenvalues were computed via the standard Lanczos routine DNLASO with 306 matrix-vector multiplies. This corresponds to 391 million floating point multiplies or 20 Mflops/sec. on the IBM SP2.

In the full calculations in 4 space-time dimensions, the Hamiltonian matrices vary widely in size, but the largest are 53,000 by 53,000 with about 2.8 million nonzero matrix elements stored.

We have worked very hard to improve the efficiency of the code. The generation of the Hamiltonian matrix contains searches in sorted lists, bit manipulations and compare operations. There is little floating-point computation. Speed is maximized by using "divide-and-conquer" algorithms to perform searching. Further speed improvements included avoiding integer mod operations and replacing `memcmp()` routines with explicit C code. From comparing different platforms, it is clear that the quality of the C compiler is critical for good performance.

The sparse matrix-vector multiply is limited by memory access speed: one must perform non-contiguous memory gather and scatter operations. In our case, the nonzero elements of the Hamiltonian are scattered around the matrix, with no perceivable block or band structure. On the T3E, we tried a number of strategies suggested in the CRAY manual for improving single processor performance (`http://www.sgi.com/t3e/performance.html`). We tried including a CACHE_BYPASS compiler directive; this actually degraded performance. However an IVDEP (vectorization) compiler directive improved the performance by a factor of two (evidently, this allows the noncontiguous gather and scatter operations to be done more efficiently). Another possibility is to use Library E-register routines, but we have not pursued this yet. Our measured performance of 20 Mflops/sec on the IBM SP2 compares favorably with the benchmarks produced by Trey White and P. Sadayappan, "Evaluation of Data Structures for High-Performance Sparse Matrix-Vector Multiplication" (Ohio Supercomputer Center Report). They were able to achieve 20 Mflops/sec for a sparse vector-matrix multiply.

Local computing environment

Local computing will consist of a PC running LINUX. I will analyze and visualize the results on this machine using Mathematica.

Other supercomputer support

I am also a member of a collaboration with Prof. Pinsky at the Ohio State University, project title "Transverse lattice approach to QCD." We have applied for time (20000 CPU-hours on the T3E) at the Ohio Supercomputer Center. This application has not been approved. However, the project at the Ohio supercomputer center will focus on supersymmetric models in two space-time dimensions: this is the main interest of the PI, Prof. Pinsky, and the other collaborators.

Project Team Qualifications

Qualifications are summarized in the enclosed curriculum vitæ(enclosed). During the last four years, I have been working on the transverse lattice in close collaboration with Simon Dalley at Cambridge University. Presently, Simon Dalley is not involved in the development or running of the computer code.

I have experience, debugging, running, and optimizing code, on a wide variety of high-performance machines:

- Cray T3E at the Ohio Supercomputer Center.
- IBM SP at Argonne and the Universität Karlsruhe, Germany.
- Fujitsu VPP300 at the Universität Erlangen, Germany.
- Convex SPP, at the Universität Erlangen, Germany.
- Hitachi SR 2201, Cambridge, England.