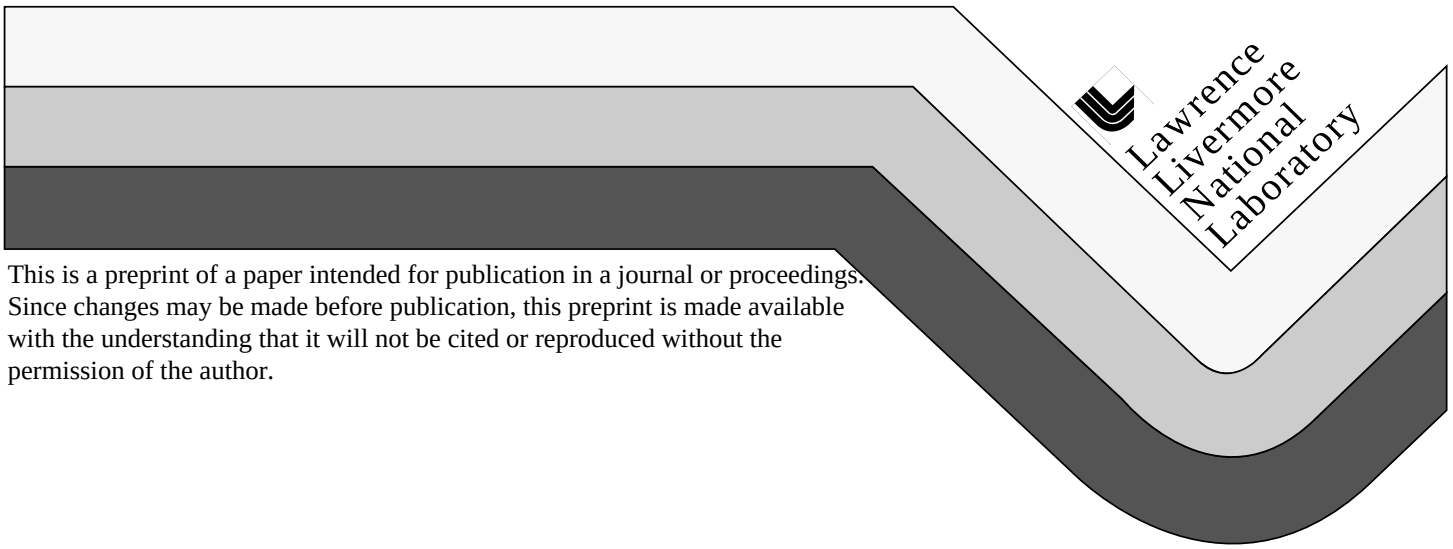
UCRL-JC-132006
Preprint

Performing Three-Dimensional Neutral Particle Transport Calculations on Tera Scale Computers

Peter N. Brown, Britton Chang, Milo R. Dorr,
Ulf R. Hanebutte and Carol S. Woodward

This paper was prepared for submittal to
High Performance Computing Symposium 99
San Diego, CA
April 11-15, 1999

January 12, 1999



PERFORMING THREE-DIMENSIONAL NEUTRAL PARTICLE TRANSPORT CALCULATIONS ON TERA SCALE COMPUTERS *

Peter N. Brown, Britton Chang, Milo R. Dorr, Ulf R. Hanebutte and Carol S. Woodward

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Box 808, L-561
Livermore, CA 94551

Abstract

A scalable, parallel code system to perform neutral particle transport calculations in three dimensions is presented. To utilize the hyper-cluster architecture of emerging tera scale computers, the parallel code successfully combines the MPI message passing and Pthreads paradigms. The code's capabilities are demonstrated by a shielding calculation containing over 14 billion unknowns. This calculation was accomplished on the IBM SP "ASCI-Blue-Pacific" computer located at Lawrence Livermore National Laboratory (LLNL).

1 Introduction

The ability to model the transport of neutral particles such as neutrons and photons through matter is of importance to many scientific and engineering activities. Among them are reactor and shielding design, development of medical radiation treatment and nuclear well logging applications, just to name a few. In this work we present a scalable, parallel code system to perform neutral particle transport calculations in three dimensions. We are pursuing two approaches to the simulation of these problems. First, solutions are sought to the linear steady-state Boltzmann transport equation (BTE)(Eq. 1), which is an integro-differential equation arising in deterministic models of neutral particle transport [7] given by,

$$\Omega \cdot \vec{\nabla}\psi(\vec{r},\Omega,E) + \sigma(\vec{r},E)\psi(\vec{r},\Omega,E) = \int_0^\infty \int_{S^2} \psi(\vec{r},\Omega',E')\sigma_s(\vec{r},E'\to E,\Omega\cdot\Omega')d\Omega' dE' + q(\vec{r},\Omega,E). \quad (1)$$

Here, $\sigma = \sigma_t$ and σ_s denote the total and scattering cross sections respectively. The discretization of the phase space variables consists of a discrete ordinates (S_n) collocation in angle, Petrov-Galerkin finite-element method in space and a multi-group approximation in energy.

The second approach is to seek solutions to a diffusion approximation to the Boltzmann transport equation and couple this approximation to a material temperature equation. This system of equations models transport of particles through optically thick regimes as well as the thermal coupling of energy between the particles and the material through which they move. This system is given by,

$$\frac{\partial \varepsilon}{\partial t} = \nabla \cdot \left(\frac{1}{3\sigma_T}\nabla\varepsilon\right) + \sigma_a\left(\frac{4\pi}{c}B - \varepsilon\right), \quad (2)$$

$$\frac{\partial (C_v \rho T)}{\partial t} = -\int_0^\infty \sigma_a \left(\frac{4\pi}{c}B - \varepsilon\right) d(h\nu), \quad (3)$$

where $\varepsilon = \varepsilon(\nu,\vec{r})$ is the energy density at frequency ν and position $\vec{r} = (x,y,z)$, $T = T(\vec{r})$ is the material temperature at position $\vec{r}$, t is time, c is the speed of light, $B = B(T,h\nu)$ is the Planck function at frequency ν and temperature T, ρ is the material

*This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

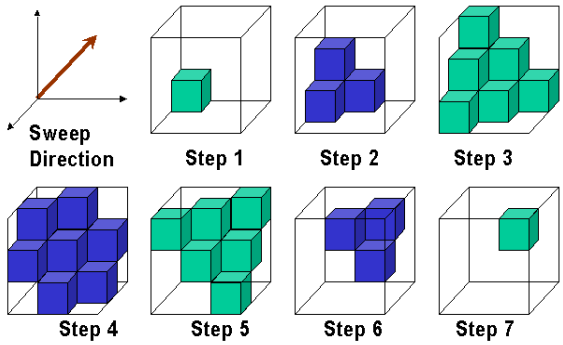


Figure 1: 3D wave front for one direction on a spatially distributed problem domain over $3^3 = 27$ processing nodes

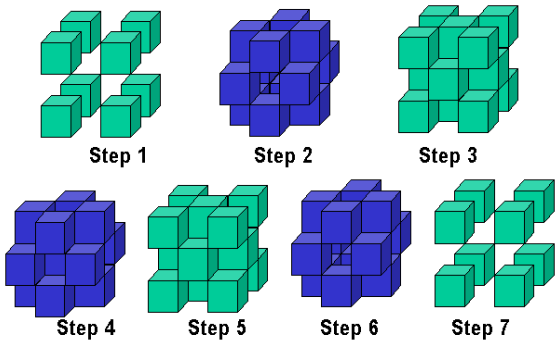


Figure 2: Simultaneous sweeps for all 8 octants on 27 processing nodes

density, C_v is the material specific heat, σ_T is the total opacity, σ_a is the absorption opacity and h is Planck's constant. Note that we are not allowing for scattering effects in this second model at this point (although we plan to add this physics later); therefore, here the total opacity and absorption opacity are equal. Discretization and solution methods for this system will be discussed in Section 5.

2 Overview of the Transport Solution Method

The transport code exploits concurrency with respect to all phase space variables represented by direction, position and energy [5]. The parallel execution and interprocessor communication is performed by calls to MPI library routines [6], which insures portability among computing platforms. Solutions to complex geometries can be obtained, since a powerful geometry package developed for the LLNL code COG [3] has been integrated with the code system. The geometry module produces structured Cartesian grids capable of non-uniform grid spacing. Both Dirichlet and specular reflection boundary conditions are supported. Therefore, one can take advantage of existing problem symmetries. For example, only one eighth of a domain has to be solved for problems that have reflection symmetry in all three directions, which is found in many engineering applications.

To obtain the steady state solution, either source iteration (Richardson Iteration) or a Krylov subspace method (the Biconjugate Gradient Stabilized, BICGSTAB, algorithm) is applied in conjunction with a sweeping algorithm to solve the discretized system. Diffusion Synthetic Acceleration (DSA) [1] is implemented with a parallel semicoarsening multigrid algorithm (SMG) [9] in a highly efficient manner, which improves the convergence behavior of the algorithm significantly in the optically thick regime. To remedy a fundamental shortcoming of the discrete ordinates approximation, known as ray effects, a harmonic projection method has been developed and implemented within the code system. This method allows one to obtain the quality of a spherical harmonics, or P_n solution, while exploiting the efficiency and better parallelizability of the S_n method.

3 3D Sweep Procedure

The sweep procedure is an important part of the overall solution algorithm and accounts for 50% of its runtime. Achieving good performance of a parallel computer is particularly difficult due to the fact that a single sweep is a sequential operation.

The phase space variables are dicretized in space by utilizing Cartesian grids and in angle through discrete ordinates (S_n) collocation. Dependent on boundary conditions of the problem, either all or subsets (i.e. all directions belonging to the same octant) of the angular directions can be swept concurrently.

Domain decomposition is applied to the spatial grid resulting in a single subgrid which resides on

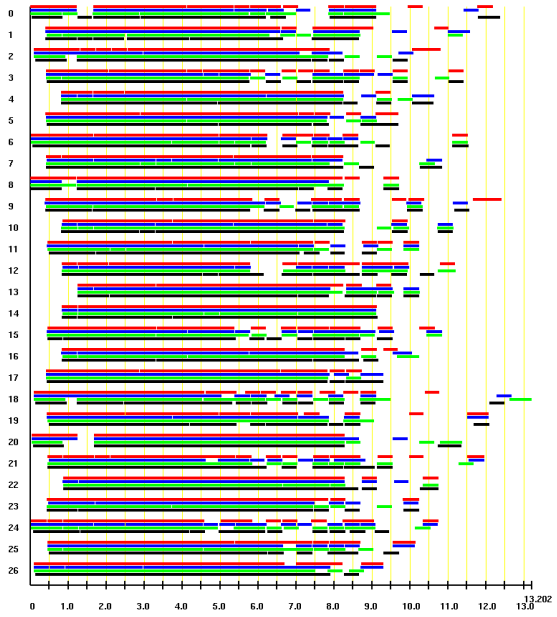


Figure 3: Measurements on $27 * 4 = 108$ processors of ASCI Blue

a single processing node. Fig. 1 depicts a single 3D wave front for one direction on a spatially distributed problem domain over $3^3 = 27$ processing nodes. As shown, seven steps are required to propagate the front from one corner of the domain to its opposite corner. The degree of concurrency can be increased by starting sweeps simultaneously (problem boundary conditions permitting) from all eight corners, i.e. all eight octants of the angular space. For 27 processing nodes this procedure is illustrated in Fig. 2.

For the simultaneous sweeps, the number of sweeps (i.e. the work load) that a processor has to perform during a single step is non-uniform. However, if the per node compute resources would be unlimited, the entire sweep process for all discrete directions combined would take exactly seven steps. Rather than enforcing a synchronize procedure which requires all processing nodes to synchronize at the end of each step, the code utilizes a data driven algorithm. This algorithm results in a better utilization of the computation and communication resources and leads ultimately to higher parallel efficiencies. Non-blocking send and receive operations provided by the MPI library are used to communicate data between neighboring processors during the sweep procedure.

To utilize hyper-cluster architectures, where each processing node is comprised of a set of processors, the transport code takes advantage of the multi-thread paradigm. Using Pthreads [8], each processing node is computing multiple sweeps concurrently. Assuming 4 processors per node, the code would start four independent threads at the beginning of the 3D sweep procedure. Successively, each thread is obtaining required initial data for any sweep direction that has not yet been swept, which may require waiting. As soon as a thread has received the data, it will start to calculate that particular sweep for which it received the initial data. After completion of the single sweep, the data is sent to the neighbor processing nodes and the thread is lined up to receive the next available work order.

In order to study the performance of the 3D sweeping procedure, a rudimentary simulation tool has been developed which focuses on task dependency, while communication costs are ignored. The simulation tool further assumes a uniform spatial distribution of work among compute nodes (i.e. equal work). Within these limitations, the tool provides performance predictions depending on the number of processing nodes, the layout of processing nodes, the number of threads per node as well as the order of the angular approximation. Additionally, various strategies for the sweep procedure can be compared. In order to validate the simulation tool, the transport code has been augmented with routines to collect time stamp information. Actual measurements obtained on the ASCI Blue-Pacific (Fig. 3) and simulation results (Fig. 4) compare favorably. The results are given for an angular discetization of order S_8, i.e. 80 angular directions. The spatial domain is aligned with a cubical processor layout, where 108 processors are arranged in $3^3 = 27$ clusters of 4 processors each.

The effect of higher S_n / P_n approximations on the parallel efficiency of the code for hyper-clusters of various processor/cluster configurations is given in Table 1. A transport calculation utilizing a P_{21} approximation with 968 angular directions has a predicted efficiency that is more than twice the efficiency of a calculation using a S_8 approximation, which contains 80 directions.

The simulations further show, that while domain decomposition in direction is supported by the transport code a directional decomposition is not advisable for computations on hyper-clusters. In order to obtain the highest efficiencies, each processing node

S_n / P_n Order	Threads	Steps	Efficiency
S_8	1	89	90%
S_8	4	27	74%
S_8	12	13	51%
S_8	16	12	42%
P_{21}	1	977	99%
P_{21}	4	249	97%
P_{21}	12	88	92%
P_{21}	16	67	90%

Table 1: Predicted efficiencies for $3^3 = 27$ processing nodes.

should be responsible for all angular directions.

4 Shielding Calculation of the Nova Target Chamber

The code's capabilities are demonstrated by a shielding calculation containing over 14 billion unknowns. This calculation was accomplished on the IBM SP "ASCI-Blue-Pacific" computer located at LLNL. The ASCI-Blue-Pacific computer manufactured by IBM [10] is a Hyper-Cluster of 1,464 "Silver" processing nodes. Each node is a four-way shared memory multi processor (SMPs) with either 1.5 or 2.5 gigabytes of memory for a total of 2.6 terabytes. Global and local disk space accounts to 62.5 and 17.3 terabytes respectively. The full machine consists of three 488-node sectors with a peak speed of 3.9 TFLOPS. The shielding calculation was carried out on two of the three sectors.

The purpose of this calculation is to simulate the flux of fusion neutrons that comes out of the Nova laser target chamber. The neutrons are produced by the fusion of Deuterium-Tritium gas. In order to protect the experimentalists from the harmful effects of the neutrons, building engineers need to know the distribution of neutron flux so that they can design the appropriate shielding structures into the building. The fusion neutrons can be very penetrating when they emerge from the target chamber because they are born at a very high energy (14.1 MeV). Even though the target chamber is packed with experimental equipment that can absorb the neutrons or scatter them into other directions, there are directions in the target chamber with large voids, and this allows

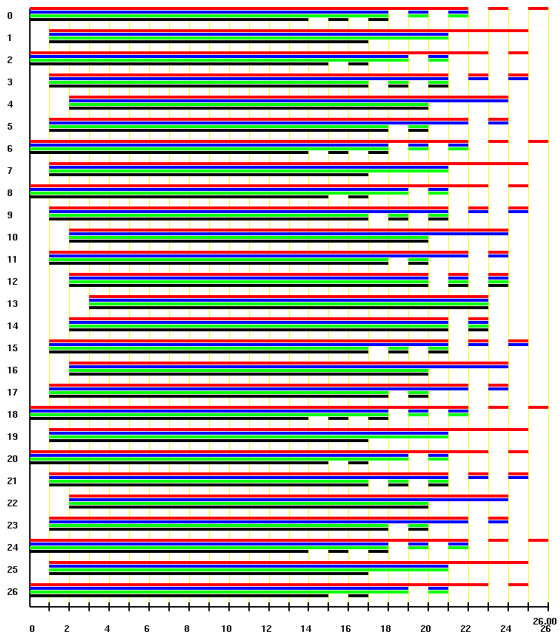


Figure 4: Simulation results for $27 * 4 = 108$ processors

the neutrons to pass through relatively freely. It is in these directions that the high-energy neutrons can be dangerous if not absorbed in a shield. These directions show up clearly in our calculations (Fig. 5).

5 Overview of the Solution Method for Diffusion

The solution method for the diffusion approximation is fully described in [2]. We briefly overview it here. The discretization of equations (2)–(3) consists of cell-centered differences in space and multi-group approximation in energy. Note that we have implmented parallelism in space. In order to allow large time steps, we use an implicit formulation of the equations and apply ODE time integration techniques. In particular, we use the PVODE package [4] developed at LLNL which implements a variant of the Backward Differentiation Formula for time integration.

With these discretizations, there is a coupled system of discrete, nonlinear equations which must be solved at each time step. We use an inexact Newton-

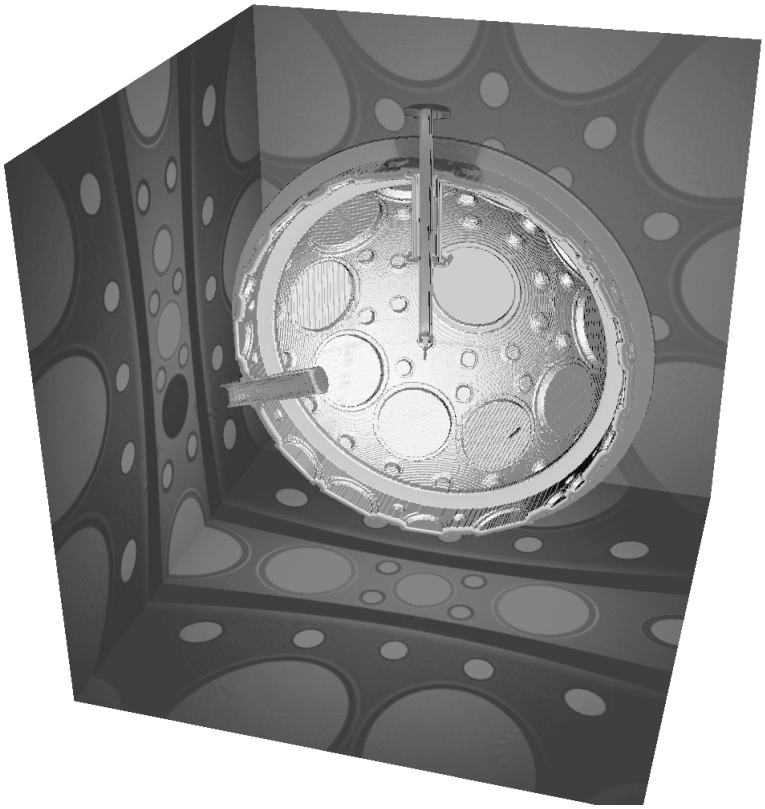


Figure 5: The Nova target chamber. Isosurfaces in the figure display a cut-away view of the interior of the chamber. In particular, the target assembly is shown along with the interior geometry of the chamber. The colors on the isosurfaces (reproduced here in grey-scale) represent neutron scalar flux values for the most energetic neutrons.

Krylov method in the context of ODE integrators to solve these systems. Newton's method is used here because of its fast, robust convergence for nonlinear systems. The Krylov method is used to solve the linear Jacobian systems which arise for each Newton iteration. Krylov methods are particularly useful in this situation since only the action of the Jacobian matrix on a vector is needed. This action can be approximated by taking differences of the nonlinear function representing the discrete nonlinear system as in,

$$F'(u)v \approx \frac{F(u+\theta v) - F(u)}{\theta}, \tag{4}$$

where θ is a scalar. Thus, only the implementation of the nonlinear function is necessary, and matrix entries need never be formed or stored.

As our Krylov solver, we use GMRES for its robust convergence. However, convergence with GMRES may be slow or stagnated at times, and so we precondition the Jacobian systems. We impose an ordering of unknowns first by energy and then by space within each energy group. Equations are ordered in a similar manner. Given this ordering, the Jacobian has diagonal blocks consisting of discrete diffusion operators and a contribution from local physics for the blocks corresponding to the energy equations and a diagonal block corresponding to the temperature equation. This last block is easily inverted. We apply a block Jacobi preconditioner to these Jacobian Systems where the blocks from the energy equations are inverted using the semicoarsening multigrid algorithm of Schaffer [9].

6 Scalability of the Diffusion Approximation

Our studies have shown that the solution method we employ for the diffuison approximation shows both algorithmic scalability as well as parallel scalability [2]. Tests on a one-dimensional Marshak problem show that the method requires a fairly constant number of nonlinear iterations per time step and a constant number of linear iterations per nonlinear iteration as the spatial mesh is increased from 200 to 25,600 points. Thus, as more unknowns are added, the solvers require about the same number of iterations, indicating algorithmic scalability. Also, as the number of unknowns is doubled, the total sequential compute time approximately doubles, indicating implementation scalability.

To test for parallel scalability, we looked at a three-dimensional version of the Marshak test problem and added more unknowns by keeping 50^3 spatial zones (250,000 unknowns) on a processor and going from 1 to 128 processors on the IBM SP2 (Technical Refresh) machine at LLNL. We saw a scaled efficiency of about 50% for 128 processors for the full run. Looking at the ratio of time for the linear iteration to the time for the full run, we saw a scaled efficiency of the linear iterations of 72%.

7 Conclusion

Large transport calculations are routinely performed on clusters of SUN workstations, DIGITAL Alpha Clusters and IBM SP computers. Previous implementations included the Meiko CS-2 and the Cray Research T3D. The transport code combines multithreading with massage passing, while preserving the efficient sweeping algorithm of the code, i.e. a data driven, 3D sweep procedure. A simulation tool has been developed to study performance issues. Comparison between actual measurements obtained with the threaded code on "ASCI-Blue-Pacific" and simulation results show a high level of agreement. This agreement enables us to predict code performance on future parallel architectures which will feature larger numbers of processors per node and further increases in the number of compute nodes. We have also shown both, algorithmic and parallel implementation scalability for a diffusion approximation. Currently, we are using time integration techniques from the diffusion approximation to extend our transport solution method to the time-dependent BTE.

References

[1] Brown, P.N. *A linear algebraic development of diffusion synthetic acceleration for 3-d transport equations.* SIAM J. Numer. Anal., 32 (1995), pp. 179–214.

[2] Brown, P.N., Chang, B., Graziani, F. and Woodward, C.S. *Implicit Solution of Large-Scale Radiation-Material Energy Transfer Problems.* Iterative Methods in Scientific Computation II, International Association for Mathematics and Computers in Simulations, 1999. Also available as LLNL Technical Report UCRL-JC-132831.

[3] Buck, R., Clark, D., Hadjimarkos, S. and Lent, E. *COG User's Manual.* Lawrence Livermore National Laboratory, July, 1994.

[4] Byrne, G.D. and Hindmarsh, A.C. *PVODE an ODE solver for parallel computers.* Tech. Rep. UCRL-JC-132361, Lawrence Livermore National Laboratory, 1998. Submitted.

[5] Dorr, M.R. and Still, C.H. *Concurrent source iteration in the solution of three-dimensional, multigroup, discrete ordinates neutron transport equations.* Nuclear Science and Engineering, 122, pp. 287-308, 1996. Tech. Rep. UCRL-JC-116694, Lawrence Livermore National Laboratory, July 1994.

[6] Gropp, W., Lusk, E., Skjellum, A. (1994). *Using MPI.* MIT Press, Cambridge, MA, 1994.

[7] Lewis, E.E., Miller Jr., W.F. *Computational Methods of Neutron Transport.* Wiley, New York, 1984.

[8] Nichols, B., Buttlar, D. and Proulx Farrell, J. *Pthreads Programming.* O'Reilly, 1996.

[9] Schaffer, S. *A Semi-coarsening mulitgrid method for elliptic partial differential equations with highly discontinous and anisotropic coefficients.* SIAM Journal on Scientific Computing, 20, 1, pp 228–242, 1998.

[10] www document *IBM Delivers Phase 1 of Blue-Pacific SST to Livermore.* http://www.llnl.gov /asci/news/phase1_SST.html, October, 3, 1998.