JOURNAL OF COMPUTATIONAL PHYSICS 83, 433–446 (1989)

Symbolic Implicit Monte Carlo*

EUGENE D. BROOKS III

University of California, Lawrence Livermore National Laboratory, Livermore, California 94550

Received March 18, 1988; revised October 21, 1988

We introduce a new implicit Monte Carlo technique for solving time dependent radiation transport problems involving spontaneous emission. In the usual implicit Monte Carlo procedure an effective scattering term in dictated by the requirement of self-consistency between the transport and implicitly differenced atomic populations equations. The effective scattering term, a source of inefficiency for optically thick problems, becomes an impasse for problems with gain where its sign is negative. In our new technique the effective scattering term does not occur and the execution time for the Monte Carlo portion of the algorithm is independent of opacity. We compare the performance and accuracy of the new symbolic implicit Monte Carlo technique to the usual effective scattering technique for the time dependent description of a two-level system in slab geometry. We also examine the possibility of effectively exploiting multiprocessors on the algorithm, obtaining supercomputer performance using shared memory multiprocessors based on cheap commodity microprocessor technology.

1. INTRODUCTION

The implicit Monte Carlo (IMC) technique for solving time dependent radiation transport was introduced in [1] for transport and material energy equations under LTE conditions and was extended to the case of non-LTE line transport in [2]. In this technique, the formal solution of the material energy or atomic population equations is substituted into the spontaneous emission term of the transport equation to obtain a consistent set of decoupled equations. This leads to an effective scattering term in the resulting transport equation. For optically thick media the

* Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

0021-9991/89 $3.00

effective scattering term dominates the Monte Carlo solution of the transport equation, causing the execution time to diverge as the optical thickness of the media is increased. Although the IMC technique is robust for problems of high opacity, its application to them is limited in practice due to the large amount of computer time consumed by the effective scattering process.

For line transport problems, where it is physically possible to have gain in the line, the situation can be worse than a simple loss of efficiency. When the problem crosses into the gain regime the effective scattering coefficient becomes negative. This can be handled, in principle, by introducing negative weight photon bundles. The distance to a scattering event is computed using the absolute value of the scattering coefficient, but just what happens at the scattering event depends upon the sign of the coefficient. If the sign is positive a normal scattering event occurs. If the sign is negative the weight of the incident bundle is doubled and a new bundle, with a negative weight equal in magnitude to the weight of the incident bundle, is generated to travel in the scattered direction. Using negative weight bundles results in poor statistical accuracy due to the subtractions occurring in the photon distribution. In addition to the poor statistical accuracy caused by subtractions, the *net* photon weight in a zone is no longer guaranteed to be positive. Such fluctuations can occur quite frequently in practice unless large sample sizes are used for the photon distribution.

A new Monte Carlo technique, which would remove the effective scattering term from the Monte Carlo solution of the transport equation while maintaining the same level of robustness inherent in IMC, would be a significant advance in the use of Monte Carlo to solve time dependent transport problems. With the effective scattering term gone, the computer time required for the Monte Carlo portion of the algorithm would be independent of optical thickness. The new technique would extend the practical application of Monte Carlo to very thick problems where IMC is currently a robust but inefficient method to solve time dependent transport problems. This is not the first attempt to improve the efficiency of implicit Monte Carlo for optically thick problems. A random walk procedure [3] which can be used for problems in LTE has been developed. Random walk is an approximation to the real solution of the transport equation that offers a limited performance increase over the standard IMC technique. We consider here a new implicit Monte Carlo technique, called symbolic implicit Monte Carlo (SIMC) for reasons explained below, which does not require any approximation in the solution of the transport equation to improve efficiency for optically thick problems. Our new technique is faster than IMC for optically thin systems as long as the Monte Carlo portion of the algorithm dominates the computation time. For optically thick systems the new technique maintains a computation time that is essentially independent of opacity.

The key to the new method is in realizing that you can track photon bundles and record their histories without knowing their weights. In the Monte Carlo used to provide an estimate of the radiation field all decisions regarding the disposition of a photon bundle are independent of its weight and this allows us to track and score

bundles with unknown weights. The spontaneous emission term in the transport equation can be treated implicitly, with respect to the atomic populations, by emitting bundles with unknown or *symbolic* weights that depend on the forward different atomic populations. Once the time histories of the bundles are known the spatially coupled equations governing the atomic populations may be solved. With the atomic populations computed the numeric weights of the symbolic photon bundles can be assigned. This process is repeated for each time step resulting in a robust time dependent solution of the coupled transport and atomic population equations.

The resulting computational method is as robust and accurate as the standard IMC technique and offers a startling speed improvement for optically thick problems. It may also be applied to line transport problems involving gain due to the absence of the effective scattering term. For problems without gain we have found the SIMC technique, like the tradiational IMC technique, to be unconditionally stable in practice. In the case of problems with gain we have found, again through practical numerical calculations, that one must limit the time step size to control instabilities generated by the stimulated emission term. The reasons for apparent stability in problems without gain and conditional stability for problems with gain are not clearly understood and remain an open issue.

In this paper we will consider the applications of the SIMC technique to a two-level line transport problem in slab geometry. The problem being considered is the same one considered in [2] and we compare both the accuracy and speed of the new algorithm with the one of the previous paper. The outline of the paper is as follows. In Section 2 we discuss the problem to be solved and the differencing scheme used in the numerical solution of the equation governing the atomic populations. This is essentially the same as in [2], except that the formal solution of the atomic populations is not substituted into the transport equation to generate an effective scattering term. In Section 3 we describe the SIMC technique, the heart of which is the tracking of photon bundles with *symbolic* weights. The linear system that must be solved to obtain the atomic population fractions is also discussed in this section. In Section 4 we consider a simple line trapping problem. We show that SIMC and IMC deliver equivalent results for sufficiently small time step sizes and that SIMC has slightly less overshoot for large time step sizes. In Section 5 we consider the execution speeds of both IMC and SIMC as the optical thickness of the problem is increased. The SIMC technique is faster than the IMC technique for optically thin problems as long as the Monte Carlo dominates the execution time. As the opacity is increased, the execution time for the IMC diverges while the execution time for SIMC remains essentially constant. In Section 6 we turn to the topic of multiprocessing the SIMC algorithm. We obtain supercomputer performance using a shared memory multiprocessor based on cheap commodity microprocessor technology. We end with a discussion in Section 7.

2. The Mathematical Method

As a test bed for the symbolic IMC technique, consider a two-level system including collisional, or external, pumping between the atomic levels. We will consider the system in 1-dimensional slab geometry. No further complications arise in higher dimensions except that the number of zones, and therefore the size of the linear system which occurs in SIMC, can rapidly get out of hand. The transport equation or such a system in slab geometry is

$$\frac{\partial f}{\partial t}+\mu c\frac{\partial f}{\partial x}=\frac{n_2}{2}A_{21}\phi-c(K_{12}n_1-K_{21}n_2)\,\phi f, \tag{2.1}$$

where c is the speed of light, x is the position in the slab, μ is the direction cosine of the radiation, ν is the frequency of the radiation, $f(\mu, \nu, x, t)$ is the photon number density distribution per unit atom density, $n_2(x, t)$ is the upper level atomic population fraction, $n_1(x, t)$ is the lower level atomic population fraction, A_{21} is the spontaneous emission rate, $\phi(\nu)$ is the line shape function normalized to unit integral and $K_{12}=\kappa N$, where κ is the lower state absorption cross section and N is the atom number density. The coefficient K_{21} is defined by

$$K_{21}=\frac{g_1}{g_2}K_{12}, \tag{2.2}$$

where g_1 and g_2 are the usual statistical weight factors for levels 1 and 2. As in [2], we consider the problem in the regime of complete redistribution and no physical scattering of photons. The accomodation of a physical scattering term is trivial, the generalization to partial redistribution and multi-line problems, for which the SIMC technique is essential, will be dealt with in a future paper.

The equations governing the atomic population fractions n_1 and n_2 are

$$\frac{dn_2}{dt}=C_{12}n_1-C_{21}n_2-A_{21}n_2+c(K_{12}n_1-K_{21}n_2)\int_{-1}^{1}d\mu\int_0^\infty d\nu\phi(\nu)f(\mu,\nu) \tag{2.3}$$

and

$$n_1+n_2=1, \tag{2.4}$$

where C_{12} and C_{21} are rate constants for the collisional transitions $1\to 2$ and $2\to 1$, respectively. One must add to the above equations suitable boundary conditions, for instance, a specification of the inwardly directed f on the left and right edges of the finite slab of thickness l in which our problem is defined and the values of f and the atomic population fractions inside the slab at time $t=0$.

Using (2.4) one can rewrite (2.1) and (2.3) as

$$\frac{\partial f}{\partial t}+c\mu\frac{\partial f}{\partial x}=\frac{n}{2}A_{21}\phi-c[K_{12}-(K_{21}+K_{12})\,n]\,\phi f \tag{2.5}$$

and

$$\frac{dn}{dt} = C_{12} - (C_{12} + C_{21} + A_{21})\, n + c[K_{12} - (K_{21} + K_{12})\, n]$$

$$\times \int_{-1}^{1} d\mu \int_{0}^{\infty} dv\, \phi(v)\, f(\mu, v), \tag{2.6}$$

respectively, where n is the upper level population fraction.

The scheme for generating a finite difference solution to (2.6) is the same one used in the standard IMC technique [2]. We integrate (2.6) from t_0 to $t_0 + \Delta t$, approximating $n(t)$ by $n(t_0 + \Delta t)$ in the spontaneous emission and collision terms and by $n(t_0)$ in the absorption term, obtaining

$$\begin{aligned} n(t_0 + \Delta t) = n(t_0) &+ [C_{12} - (C_{12} + C_{21} + A_{21})\, n(t_0 + \Delta t)]\, \Delta t \\ &+ c[K_{12} - (K_{21} + K_{12})\, n(t_0)] \\ &\times \int_{t_0}^{t_0 + \Delta t} dt \int_{-1}^{1} d\mu \int_{0}^{\infty} dv \phi(v)\, f(\mu, v, t). \end{aligned} \tag{2.7}$$

The reader will note that up to this point we have not done anything different that what was done in [2] to generate a prescription to compute $n(t_0 + \Delta t)$ given $n(t_0)$ and the time integral of the photon distribution over the time interval from t_0 to $t_0 + \Delta t$. It is at this point that we do something radically different.

3. The Symbolic Implicit Monte Carlo Technique

In order to use the prescription (2.7) to advance the atomic populations one time step we need the time integral of the photon distribution from t_0 to $t_0 + \Delta t$ specified in the last term of (2.7). To be consistent with (2.7) in the treatment of the spontaneous emission, our solution of (2.5) needs to use the forward differenced atomic fraction $n(t_0 + \Delta t)$ in the spontaneous emission term as we integrate (2.5) across the time step. In the standard IMC technique, we substitute the formal solution for $n(t_0 + \Delta t)$ into the spontaneous emission term of (2.5) and note that the resulting transport equation now contains an effective scattering term that can be dealt with via Monte Carlo. The Monte Carlo is run and the time integral of the photon distribution is used to compute $n(t_0 + \Delta t)$. The problem with this approach is that the effective scattering term dominates the Monte Carlo for optically thick problems, resulting in rather long execution times.

We would like to use Monte Carlo to provide a statistical estimate of the photon distribution without introducing an effective scattering term. At first sight, it seems impossible to create a Monte Carlo estimate for the integral of (2.5) without having $n(t_0 + \Delta t)$ in hand to assign weights to the spontaneously emitted photon bundles. It turns out to be quite trivial to accomplish once one notices that the decisions made in creating the time histories of the photon bundles do not depend on their

weights. Why not emit and track photon bundles with *symbolic* weights, that is weights that depend on the forward differenced atomic popultion fractions that have not yet been determined? By tracking the symbolic photon bundles, one creates a Monte Carlo estimate of the integral of the photon field which depends on the unknown forward differenced atomic population fractions. This estimate can now be used to solve (2.7) and results in a linear system in the atomic population fractions. The linear system can be solved and the atomic population fractions $n(t_0+\Delta t)$ can then be used to establish numeric values for the symbolic photon bundles.

To implement the symbolic IMC algorithm for our test problem, a slab of thickness l, we divide the region up into *nzones* zones of equal thickness. The atomic population fractions are assumed to be constant, as a function of x, within each zone. The first step of the algorithm is to create the statistical estimate of the photon distribution f. There are three sources of photon bundles: the census bundles from the previous time step, any bundles generated at the boundary due to boundary conditions, and any bundles that are spontaneously emitted within a zone during the time step. The procedure for generating the frequency distributions of emitted bundles and tracking them is the same as in [2], except for the lack of effective scattering, so we will not describe it here. The difference with the procedure in [2] is that the spontaneously emitted bundles now have a symbolic weight, which is proportional to the unknown forward differenced atomic population fraction $n(t_0+\Delta t)$ in the zone from which they were emitted. The symbolic weight is adjusted in flight just as the numeric weights of the census and boundary bundles are, but the scoring of a bundle with a symbolic weight is handled differently. When a bundle with a numeric weight makes a track in a zone, the time integral of the weight of the bundle is added to a single memory location associated with the zone. When a bundle with a symbolic weight, which was spontaneously emitted in zone i, makes track in zone j the ith element of an array associated with zone j is incremented with the time integral of the symbolic weight of the bundle. To get the time integral of $\int_{-1}^{1} d\mu \int_0^\infty dv\, \phi(v) f(\mu, v)$ within a zone, one must add the numeric value, which comes from tracking census and boundary bundles, to the sum of *symbolic* values from spontaneously emitted bundles after they have been scaled by the forward differenced atomic population fractions of their birth zones. The prescription (2.7) for the atomic population fraction update becomes

$$\begin{aligned} n(t_0+\Delta t)_i = n(t_0)_i &+ [C_{12} - (C_{12}+C_{21}+A_{21})\, n(t_0+\Delta t)_i]\, \Delta t \\ &+ c[K_{12} - (K_{21}+K_{12})\, n(t_0)_i] \\ &\times \left[FN_i + \sum_j FS_{ij}\, n(t_0+\Delta t)_j \right] \Big/ V_i, \end{aligned} \tag{3.1}$$

where FN_i is the contribution to

$$\int dx \int_{t_0}^{t_0+\Delta t} dt \int_{-1}^{1} d\mu \int_0^\infty dv\, \phi(v) f(\mu, x, v, t) \tag{3.2}$$

within zone i coming from bundles with numeric weights, $FS_{ij}n(t_0+\Delta t)_j$ is the contribution to the integral within zone i coming from bundles with symbolic weights that were born in zone j, $n(t_0)_i$ is the upper level atomic population fraction in zone i at the start of the time step, $n(t_0+\Delta t)_i$ is the upper atomic population fraction in zone i at the end of the time step, and V_i is the thickness of zone i.

Equation (3.1) specifies a linear system to be solved for the $n(t_0+\Delta t)_i$. The linear system is dense if the time step Δt is large enough to allow spontaneously emitted bundles from one zone to cross all the others. At first sight this seems like bad news as the solution of a linear system via Gauss elimination, for example, takes a time that grows like the cube of the dimension of the system, which in this case is the number of zones. It turns out that the matrix is strongly diagonally dominant and simple Jacobi iteration [4], using the atomic population fractions from the previous time step as a starting guess, does quite well with very few iterations required to obtain high accuracy solutions. Jacobi iteration gives a time for solution that scales like the number of zones squared and takes advantage of a good initial starting guess for the solution, the atomic population fractions from the previous time step. If the time step size is such that a symbolic bundle cannot cross too many zones during the time step, the linear system will be sparse and this can be exploited to good advantage in an iterative solver.

In practice, the time to solve the linear system is small compared to the time for the Monte Carlo and we have not had to be very sophisticated with the linear system solver. For problems with 15,000 census bundles and up to 500 zones the linear system solve has remained a small portion of the total computation time even though Gauss elimination is being used. This is particularly true on a supercomputer where the linear system solve is efficiently vectorized and the Monte Carlo is not. If we require the solution of problems with many thousands of zones we will clearly have to be more sophisticated with the linear system solve. Careful exploitation of sparseness and iterative techniques will make problems with large numbers of zones quite tractable.

4. Accuracy

We have performed numerous high statistics runs comparing the standard IMC technique [2] to the symbolic IMC technique introduced here. For reasonable time step sizes, a time step size which is small when compared to the characteristic time scales occurring in a problem, IMC and SIMC provide the same time dependent results. The two techniques have been critically compared for a variety of problem parameters and no differences, other than slightly different overshoot characteristics for large time step sizes, have been found. The agreement between IMC and SIMC is not surprising; the two techniques use the same differencing scheme to solve the coupled transport and atomic population equations. The only difference is in *how* we accomplish the forward differencing of the spontaneous emission term for the transport equation.

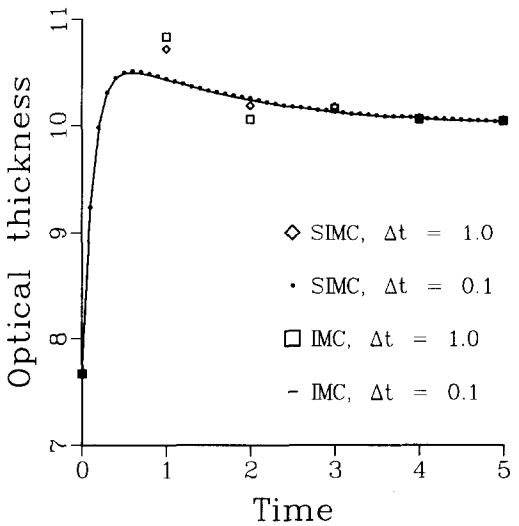

FIG. 1. Slab optical thickness as a function of time for different values of the integration time step Δt, for both IMC and SIMC.

As noted above, IMC and SIMC exhibit slightly different overshoot characteristics for large time step sizes. This is caused by the extra approximation which is made when deriving the effective scattering term of IMC. This approximation, discussed in Section 2 of [2], does not have to be made in SIMC where there is no effective scattering term. To illustrate this difference for large time step sizes, we show in Fig. 1 the slab optical thickness as a function of time using both techniques in a simple line trapping problem. The physical problem parameters for these calculations are given in Table I. The slab was divided into 32 zones of equal size, the length units are chosen so that the slab has unit length and the time units are chosen so that the light travel time across the slab is unity. For a time step size of 0.1 both IMC and SIMC give equivalent time dependent results. For a time step size of 1.0, which is quite large in comparison with the characteristic time scales in the problem, the SIMC technique has less overshoot than IMC, but both SIMC and IMC give the same steady state solution.

TABLE I

Physical Problem Parameters for Fig. 1

$n(x, t=0)$	0.25
$f(\mu, \nu, x, t=0)$	0
$f(\mu>0, \nu, x=0, t)$	0
$f(\mu<0, \nu, x=1, t)$	0
K_{21}	15.3422
K_{12}	15.3422
A_{21}	3.33564
C_{12}	0.245423
C_{21}	0.667128

5. Algorithm Performance

We consider in this section how the execution times for IMC and SIMC compare as we vary the optical thickness of the problem. To do this we take the problem configuration of Section 4 and vary the optical thickness of the slab by choosing one of the values 0.1, 1, 10, 100, or 1000 for the coefficients K_{12} and K_{21}. In Fig. 2 we plot the resulting equilibrium slab optical thickness versus the execution time to evolve the problem to $t = 20$ using a time step size of 1.0. For an optically thin problem the SIMC algorithm is about 30% faster than the IMC algorithm (provided the Monte Carlo dominates the execution time as is the case here). This is caused by the simpler Monte Carlo of SIMC that contains no scattering terms, saving the work of computing the distance to next collision. As the opacity of the problem is increased the divergence of execution time for IMC is clear while the SIMC algorithm maintains a flat execution time. This striking improvement in the execution time, while maintaining the same level of robustness, makes SIMC the clear choice for optically thick problems.

In Table II we show the execution times for SIMC on a sample problem, 20 time steps of the problem of Section 4, as the number of zones is changed. The timings in the column labeled *scalar* are execution times in seconds for a single processor on a *Symmetry* multiprocessor system, manufactured by Sequent Computer Systems Inc., using version 3.0.7 of their operating system. The Symmetry system was equipped with copy/back cache and the floating point accelerator option. The timings in the column labeled *vector* are execution times in seconds using a single processor of the *Cray X-MP 4/16*, manufactured by Cray Research Inc. The code

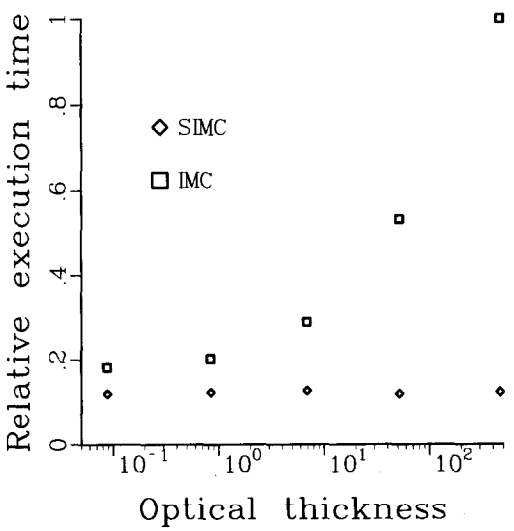

FIG. 2. Execution time for IMC and SIMC as the optical thickness of the slab is increased.

TABLE II

Execution Times for a Sample Problem as the Number of Zones is increased

*n*zones	Scalar	Vector
32	157.8	17.21
	(1.8)	(1.7)
64	283.3	28.89
	(2.0)	(1.8)
128	563.9	52.12
	(2.4)	(1.9)
256	1367.8	99.48
	(3.5)	(2.0)
512	4832.5	197.26
	(5.4)	
1024	26305.5	N.A.

was compiled for the Cray using version 2.3 of the LLNL C/Civic hybrid compiler. The numbers in parentheses are the ratios of execution times as the number of zones in the problem is doubled. The linear system solve is efficiently vectorized on the Cray machine, reducing its relative costs as compared to the Monte Carlo by about a factor of 10. The same source program, coded in the C programming language, was used on both machines and identical numerical results were obtained. The 1024 zone problem could not be run on the Cray machine due to memory limitations, the operating system enforced an eight million word memory limit.

Note that the ratio of execution times as the number of zones is doubled is slightly less than 2.0 for small numbers of zones on the scalar machine. This ratio rises to 3.5 as we double the number of zones from 256 to 512, and increases to 5.4 as we double from 512 to 1024 zones. The ratio of slightly less than 2.0 for small numbers of zones results when the Monte Carlo dominates execution time. As the zone width is halved the number of tracks made by a bundle for a large time step size is doubled. When the execution time is dominated by the linear system solve, the use of Gauss elimination results in a ratio of 8 for the execution times as the number of zones is doubled. The ratio of execution times as we double the number of zones from 512 to 1024, running on the scalar machine, indicates that the linear system solve is beginning to dominate. This is confirmed by an execution profile indicating that 60% of the time was spent in the linear system solve for the 1024 zone problem.

For the vector machine the ratio of execution times, as we double the number of zones, remains below 2.0 in Table II. The point at which the linear system solve dominates the execution time is postponed to a larger number of zones by the efficient vectorization of the linear system solve. This point has not been moved very far away, a speed increase by a factor of 10 through vectorization is quickly

consumed by the growing cost of the linear system solve that increases by a factor of 8 each time we double the number of zones.

6. Parallel Performance

Progress in the single processor performance of supercomputers has fallen into a rut in the recent decade with the scalar performance of these machines becoming almost static. The way out of this rut, which is gradually becoming generally accepted, is to use multiprocessing. It is interesting to consider whether multiprocessing can be effectively applied to the SIMC algorithm. To answer this question the code used as the workhorse for this paper was recoded using PCP [5], a parallel extension of the C programming language which is portable to several shared memory multiprocessors. Computer codes written in PCP can be compiled into efficient serial code as well, using a special option of the PCP compiler.

In the standard IMC algorithm the effective scattering term can cause the Monte Carlo to account for almost all of the problem run time, and as a result one only needs to consider multiprocessing the Monte Carlo algorithm. This is particularly true in an optically thick problem where effective scattering can raise the cost of the Monte Carlo by an order of magnitude or more. In the SIMC algorithm the Monte Carlo is not as costly and one must effectively multiprocess the Monte Carlo, the linear system solve and the gathering of data for edits in order to achieve good parallel performance. The result is a rather tightly coupled parallel algorithm for which microtasking [6, 7] was found to be cumbersome and ineffective. The use of the PCP language, which was developed in part by the author to make multiprocessing a code in a such a global manner more efficient, proved to be very effective and good parallel performance was achieved. Using the serial compilation option for the PCP compiler the code also runs efficiently on any machine with a standard C execution environment.

In Fig. 3 we show the speedup, the execution time for one processor divided by the execution time for N processors, for the SIMC code as a function of the number of processor used on the Sequent Symmetry multiprocessor. The Sequent machine was equipped with 30 processors having copy/back caches and floating point accelerators. The single processor runs were performed using code compiled for parallel execution which included the overhead of concurrency control code. This overhead added about 4% to the single cpu execution times and could have been removed by compiling for serial execution. Most of the overhead is attributed to the locking of zones to increment the integral (3.2) within a zone as a photon bundle makes a track.

As can be seen in Fig. 3 the parallel version of the SIMC code produces good speedup characteristics. A bus bandwidth limited speedup of 23.6 was obtained using 30 processors and detailed monitoring of the memory system indicated that most of the bus transactions were caused by simple cache spills. Increasing the per-

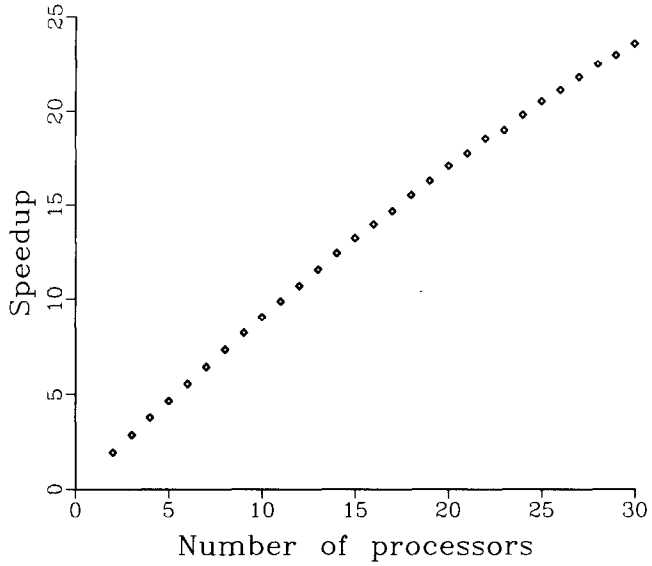

FIG. 3. The speedup of the parallel SIMC algorithm, as a function of the number of processors, for the Sequent Symmetry multiprocessor.

processor cache size so that the problem data set could be distributed in the caches without spilling would have improved performance.

When presenting speedup data for a parallel machine one is hiding what is probably the most important parameter, the absolute performance of the multiprocessor. To pin this down the same code was compiled using the serial compilation option for PCP and run on the Cray X-MP 4/16, a well-known supercomputer. About 10 min of execution time was required on the X-MP and the startling result that the 30 processor Symmetry system outperformed the Cray machine by a factor of 2.8 was obtained. The Monte Carlo code was not vectorized, putting the X-MP at a considerable disadvantage, but the aggregate performance of a multiprocessor based on the same microprocessors now appearing in personal computers is startling.

The same efficient parallel performance would probably be obtained using the Cray X-MP in multiprocessor mode, but the required PCP support does not yet exist for these machines. The cost of locking down a zone in order to score a bundle will be considerably higher on the Cray X-MP and will become a serious problem if the Monte Carlo is vectorized to improve performance. A vectorized test and set operation, using the addresses of zones for which lock attempts are to be made, would be very useful in reducing this overhead.

7. Discussion

We have developed a new technique, symbolic implicit Monte Carlo (SIMC), for the solution of radiation transport problems involving spontaneous emission. The new technique preserves the robustness of tradiational implicit Monte Carlo (IMC) while removing the effective scattering term which is a source of inefficiency for optically thick problems. We have demonstrated that the symbolic technique delivers the same results as traditional IMC, with slightly less overshoot for large time step sizes in a time dependent problem, and provides the same level of robustness.

The SIMC technique is slightly faster than the IMC technique for optically thin problems and provides an execution time which is independent of opacity. This is a startling speed improvement for optically thick problems. In the symbolic technique we have traded the effective scattering term of the IMC technique for a linear system solve to obtain the atomic population fractions at the end of the time step. A linear system solve can consume a great deal of computer time for a large number of zones. As problem sizes grow, sophisticated techniques will have to be exploited to minimize the time spent in the linear system solver. We have shown that problems with up to 1000 zones can be addressed, using Gauss elimination which would be the worse case linear system solver, before one needs to worry about the performance of the linear system solve. There are several sophisticated iterative techniques that can be used on larger problems and we are currently evaluating their performance.

In addition to introducing a new Monte Carlo algorithm, we have examined the possibility of effectively utilizing shared memory multiprocessors with modest numbers of processors on this type of code. The code had to be multiprocessed in a global sense, including the problem initialization, Monte Carlo, linear system solver, and the gathering of statistics for output, in order to achieve good parallel performance. Once the right programming model was found and used for the parallel version, the author was quite suprised at the ease with which a correctly running, efficient, and portable parallel version of the code was created. An implementation of the parallel version using microtasking was attempted but the resulting code was found to be cumbersome to maintain, inefficient, and lacking portability. A striking lesson learned in our multiprocessing studies is that the latest crop of cost effective shared memory multiprocessors, when effectively used to multiprocess a single job, are supercomputers if performance is the deciding criterion.

This paper is intended only as a vehicle to introduce the SIMC technique and concerns itself with a simple, almost trivial, line transport application. The idea of using symbolic weights in a time dependent Monte Carlo can almost certainly be applied to the LTE transport problem of [1]. The linear system solve of the transport case becomes a non-linear problem in the LTE case due to the T^4 dependence of spontaneous emission on the material temperature T. The SIMC technique was invented as a way around the problem of negative scattering coeffi-

cients which can arise in line transport IMC. Such negative coefficients arise when a line crosses into the gain regime for a simple two-level system and can arise in multiple line contexts even where no net gain is present in a line. We will publish the generalization of SIMC to multiple line problems in a future work.

The symbolic implicit Monte Carlo technique will likely play an important role in time dependent NLTE and LTE transport calculations. Banishing the effective scattering term provides a startling speed increase on problems with high opacities and opens up the possibility of treating problems with multiple lines including gain. The notion of using Monte Carlo particles with symbolic weights is not entirely new, a similar mechanism is employed by Dunn to solve the inverse scattering problem in [8]. The notion of using Monte Carlo particles with symbolic weights, with their numerical values to be determined after the track histories are recorded, may prove to be a valuable asset to other applications as well.

ACKNOWLEDGMENTS

The author thanks Joe Fleck who introduced him to the IMC concept through collaborative efforts on earlier work. This work stands on his shoulders. The author also thanks George Zimmerman for giving him the freedom to try out all the ideas that failed and the constant encouragement to find one that succeeded. A debt is also owed to Christopher Moore who developed the parallel version of the code using microtasking, pulling his hair out in the process, only to find that it did not perform very well. Chris also developed the initial version of the code that used PCP.

REFERENCES

1. J. A. FLECK, JR. AND J. D. CUMMINGS, *J. Comput. Phys.* **8**, 313 (1971).
2. E. D. BROOKS III AND J. A. FLECK, JR., *J. Comput. Phys.* **67**, 59 (1986).
3. J. A. FLECK, JR. AND E. H. CANFIELD, *J. Comput. Phys.* **54**, 508 (1984).
4. A. JENNINGS, *Matrix Computation for Engineers and Scientists* (Wiley-Interscience, New York, 1977), p. 182.
5. E. D. BROOKS III, *IEEE Software*, in press; Lawrence Livermore National Laboratory Report No. UCRL-99673, September 1988.
6. M. BOOTH AND K. MISEGADES, "Microtasking: A New Way to Harness Multiprocessors," *Cray channels*, Summer 1986, p. 24.
7. A. OSTERHAUG, "Guide to Parallel Programming on Sequent Computer Systems" (Sequent Computer systems Inc., Oregon, 1986).
8. W. L. DUNN, *J Comput. Phys.* **41**, 154 (1981).