LA-UR- 96-4507

CONF-961249--3

Title: TIME-DEPENDENT 3-D DETERMINISTIC TRANSPORT ON PARALLEL ARCHITECTURES USING DANTSYS/MPI

Author(s): R.S. Baker
R.E. Alcouffe

Submitted to: OECD/NEA Meeting, 2 - 3 December 1996, Paris, France

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

TIME-DEPENDENT 3-D DETERMINISTIC TRANSPORT ON PARALLEL ARCHITECTURES USING DANTSYS/MPI

Randal S. Baker (rsb@lanl.gov) and Raymond E. Alcouffe (rea@lanl.gov)
Transport Methods Group, MS B226
Los Alamos National Laboratory
Los Alamos, NM 87545

Introduction

In addition to the ability to solve the static transport equation, we have also incorporated time dependence into our parallel 3-D S_N code DANTSYS/MPI. Using a semi-implicit scheme, DANTSYS/MPI is capable of performing time-dependent calculations for both fissioning and pure source driven problems. We have applied this to various types of problems such as nuclear well logging and prompt fission experiments. This paper describes the form of the time-dependent equations implemented, their solution strategies in DANTSYS/MPI including iteration acceleration, and the strategies used for time-step control. Results are presented for a model nuclear well logging calculation.

Time-Dependent Transport for Discrete Ordinates

The time-dependent form of the source iteration equation is

$$\frac{1}{v}\frac{\partial}{\partial t}\psi(\vec{r}, E, \vec{\Omega}, t) + \vec{\Omega} \bullet \vec{\nabla}\psi(\vec{r}, E, \vec{\Omega}, t) + \Sigma_T(\vec{r}, E, t)\psi(\vec{r}, E, \vec{\Omega}, t) = S(\vec{r}, E, \vec{\Omega}, t) \tag{1}$$

Time-dependence is included in the total cross section to reflect possible depletion. We choose to use a semi-implicit scheme to handle the time derivative in Eq. (1), i.e.,

$$\psi(\vec{r}, E, \vec{\Omega}, t^n) = \frac{1}{2}[\psi(\vec{r}, E, \vec{\Omega}, t^{n+1/2}) + \psi(\vec{r}, E, \vec{\Omega}, t^{n-1/2})] \tag{2}$$

where n is the time index. The semi-implicit scheme has the well-known advantage of being unconditionally stable as the mesh is refined, but does require iteration in time. Using this scheme, the time-differenced form of Eq. (1) then becomes

$$\begin{gathered}\vec{\Omega} \bullet \vec{\nabla}\psi^n(\vec{r}, E, \vec{\Omega}) + \left[\Sigma^n{}_T(\vec{r}, E) + \frac{2}{v\Delta t}\right]\psi^n(\vec{r}, E, \vec{\Omega}) = S^n(\vec{r}, E, \vec{\Omega}) + \frac{2}{v\Delta t}\psi^{n-1/2}(\vec{r}, E, \vec{\Omega}) \\ \psi^{n+1/2}(\vec{r}, E, \vec{\Omega}) = 2\psi^n(\vec{r}, E, \vec{\Omega}) - \psi^{n-1/2}(\vec{r}, E, \vec{\Omega})\end{gathered} \tag{3}$$

where the superscript n refers to the value at time t^n, and

$$\Delta t = t^{n+1/2} - t^{n-1/2} \tag{4}$$

The semi-implicit method is easily implemented in an existing static code. For each time step, simply modify the total transport cross section appropriately and add a time source on the right-hand side, then solve the transport equation as before. As the angular flux at time t^n is calculated, calculate (and store) the extrapolated angular flux for time $t^{n+1/2}$. Outer and inner iterations are performed identically to the static case, and multiple outers per time step are allowed although not required. When the outer and inner iterations have converged, the time-step controller is called to calculate the next Δt, and the calculation proceeds to the next time step. One complication, though, is whenever the flux at n+1/2 is extrapolated negative from Eq. 3, we must do a fixup to preserve the strict positivity of the method. A set-to-zero fixup in time has thus been incorporated into the sweeper in that $\psi^{n+1/2}(\vec{r}, E, \hat{\Omega}) = 0$ and Eq. 3 is suitably modified to preserve balance. This is consistent with the philosophy of maintaining positivity in the spatial variables by doing a fixup while preserving particle balance.

In time-dependent calculations, storage requirements are dramatically increased from the static case due to the need to store the angular fluxes. While the static case needs to store only the flux moments, i.e., $(N+1)^2 \times G$ words per cell, where N is the P_N order and G is the number of energy groups, a time-dependent calculation must store the angular fluxes at times $n+1/2$ and $n-1/2$, as well as the flux moments at time n, when performing calculations that require outer iterations. Thus, the time-dependent storage requirements are $2 \times M \times 8 \times G + (N+1)^2 \times G$ words per cell, where M is the number of angles per octant. For a 12 group, S_6 ($M = 9$), P_2 calculation, this requirement equals 1,836 words/cell, versus 108 for a static calculation. In the YMP version of the code, the core storage is greatly reduced by writing the time boundary fluxes (at $t^{n-1/2}$ and $t^{n+1/2}$) to disk. Thus only the current group's time boundary flux need be in core. This assumes that disk I/O is relatively cheap and that core memory is a scarce resource. This certainly is true for the YMP. For the parallel architectures this strategy assumes that parallel I/O is being efficiently done which should be true. It is not true for the T3D thus we keep the time boundary angular fluxes in core when using that machine.

Implementation in DANTSYS/MPI

The existing 3-D *X-Y-Z* sweepers in DANTSYS/MPI have been modified to solve the time-dependent form of the transport equation specified in Eq. (3). For static problems, the time absorption and source terms in Eq. (3) are simply set to zero. Additional storage has been allocated for storage of the angular fluxes at $t^{n+1/2}$ and $t^{n-1/2}$ for time-dependent calculations, but this additional storage is not allocated for static problems. Thus, we retain the ability to solve both static and time-dependent problems in a single code. The solution in time is implemented as an additional loop outside of the existing outer/inner iteration loops, and convergence checks within the outer/inner iterations have been appropriately modified where necessary. The input module has been enhanced to allow for the specification of linearly-varying in time fixed sources and cross sections.

The selection of time-step control criteria is a critical factor in the overall efficiency of the solution algorithm for time-dependent problems. The time-step criteria must be tight enough to ensure accuracy, but loose enough that they allow large time steps for smoothly varying solutions in time. Furthermore, they must be robust enough that they do not "crash", i.e., go to zero, for sudden changes in the source or cross sections. In DANTSYS/MPI, time step control is currently performed by looking at both the maximum change in the pointwise total reaction rate and the pointwise fission rate. If both the maximum change in the pointwise reaction rate and fission rate from the previous time step are less than some error criteria (typically 10% for the reaction rate and 5% for the fission rate), then the time step is increased by 20%. If the change in the reaction rate is greater than some error criteria (typically 50%), then Δt is reduced by a factor of twice the relative change in the reaction rate.

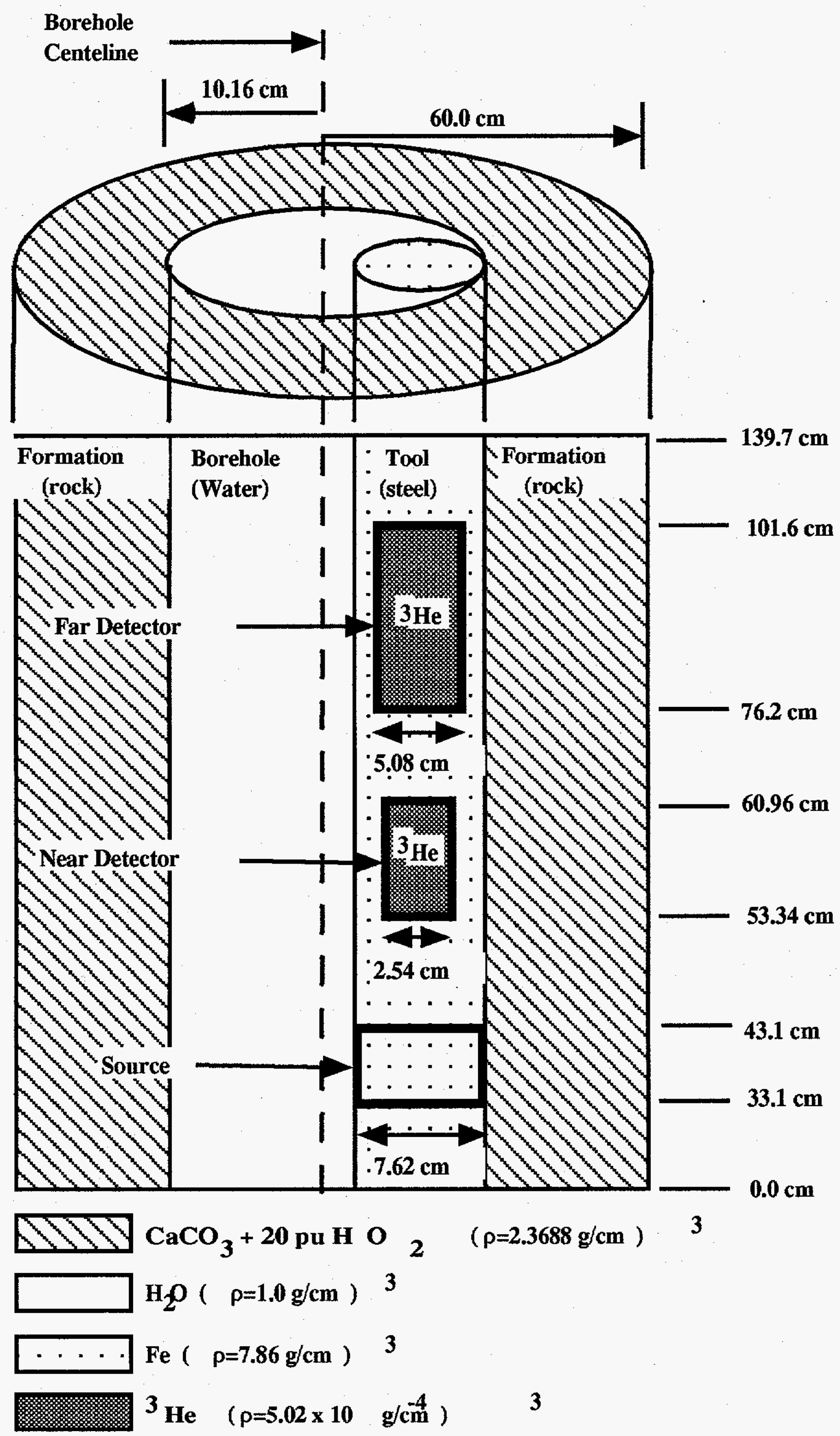
Our time-dependent form of the transport operator [Eq. (3)] is identical to that of the static operator, with the exception of an additional absorption and source term. Thus, the Diffusion Synthetic Acceleration (DSA) equations used to accelerate convergence of the transport operator for static problems are easily extended to time-dependent calculations by an appropriate modification of the absorption and source terms in the diffusion operator. Unlike the transport operator, there is no need to retain information on the diffusion fluxes from one time step to the next, so no additional storage is required for time-dependent DSA. Although DSA is not as important for time-dependent problems due to the time leakage term, our experience has been that it is still beneficial for large time steps.

Application to Nuclear Well Logging Problems

Nuclear well logging is a technique commonly used in the oil and gas industry to measure the porosity of formations. In nuclear well logging, a tool which emits neutrons or gamma rays is inserted down the borehole of a well, and pushed up against one side of the wall. The tool also includes detectors which are located at various distances from the source. The source in the tool is pulsed, radiation penetrates through the tool and out into the surrounding formation, and some of it returns to the detectors. By examining the time response of the radiation received back at the detectors from the pulsed source, important physical characteristics of the formation can be estimated. Due to the above conditions, the transport problem is inherently a time-dependent, 3-D one. Thus, Monte Carlo has traditionally been used. However, determining the detector responses through a drop of several orders of magnitude in the signal strength requires a large number of particles for adequate statistics, which makes the Monte Carlo method very expensive. Time-dependent, 3-D deterministic calculations are thus an attractive alternative.

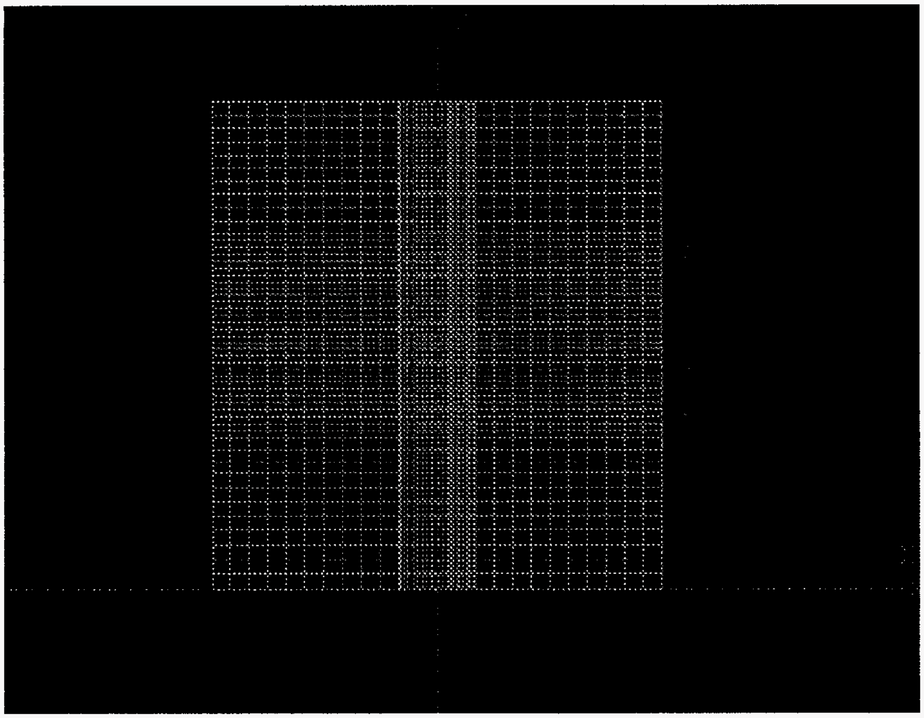
A Model Well Logging Problem

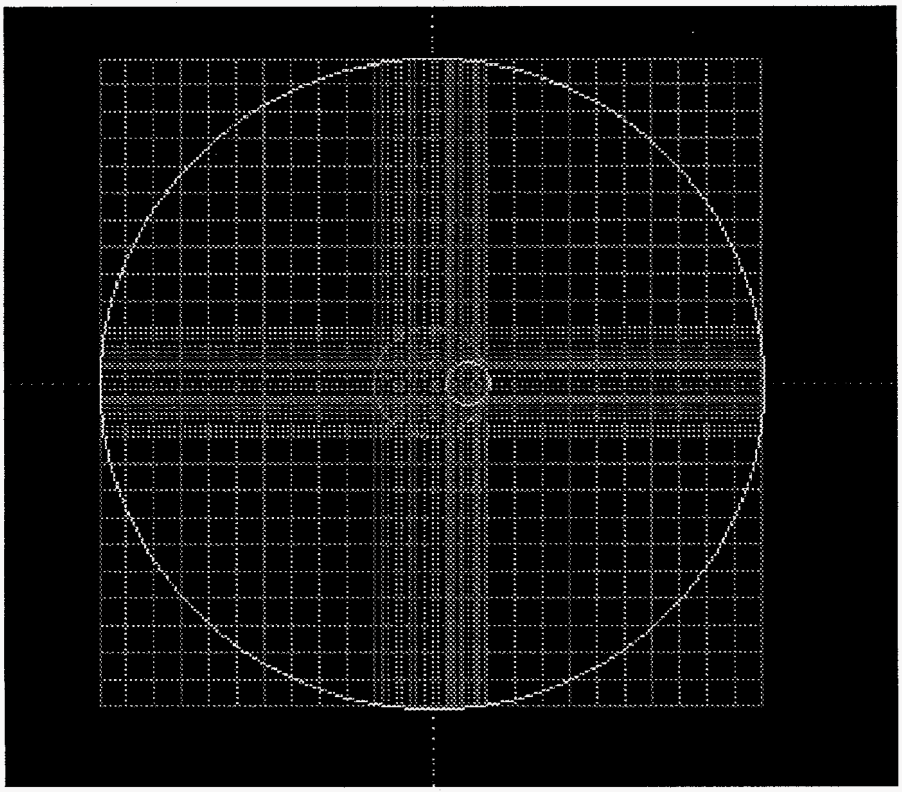
For the purposes of this paper we will examine the time response of a model well logging problem. This problem has been described in reference [1] and is represented here in Fig. 1. It is a very simple geometric model composed of a cylindrical bore hole in which is placed, off center, a cylindrical representation of a nuclear logging tool with a source region and two detector regions. In our model these are gamma detectors designated as near and far detectors. The geometry is modeled in DANTSYS/MPI using a Frac-In-The-Box body description of the system (much as would be done for a Monte Carlo code model). A 3D mesh is then superimposed upon this from which the Frac code computes the requisite volume fractions of the material in each mesh cell. An example of this mesh is shown in Fig. 2 where in the first plate is the XZ plane mesh and in the second is the XY plane mesh. The spatial mesh used for this problem is



not drawn to scale

thus 43x40x51. For this model we have developed a five group set of coupled neutron-gamma cross sec-

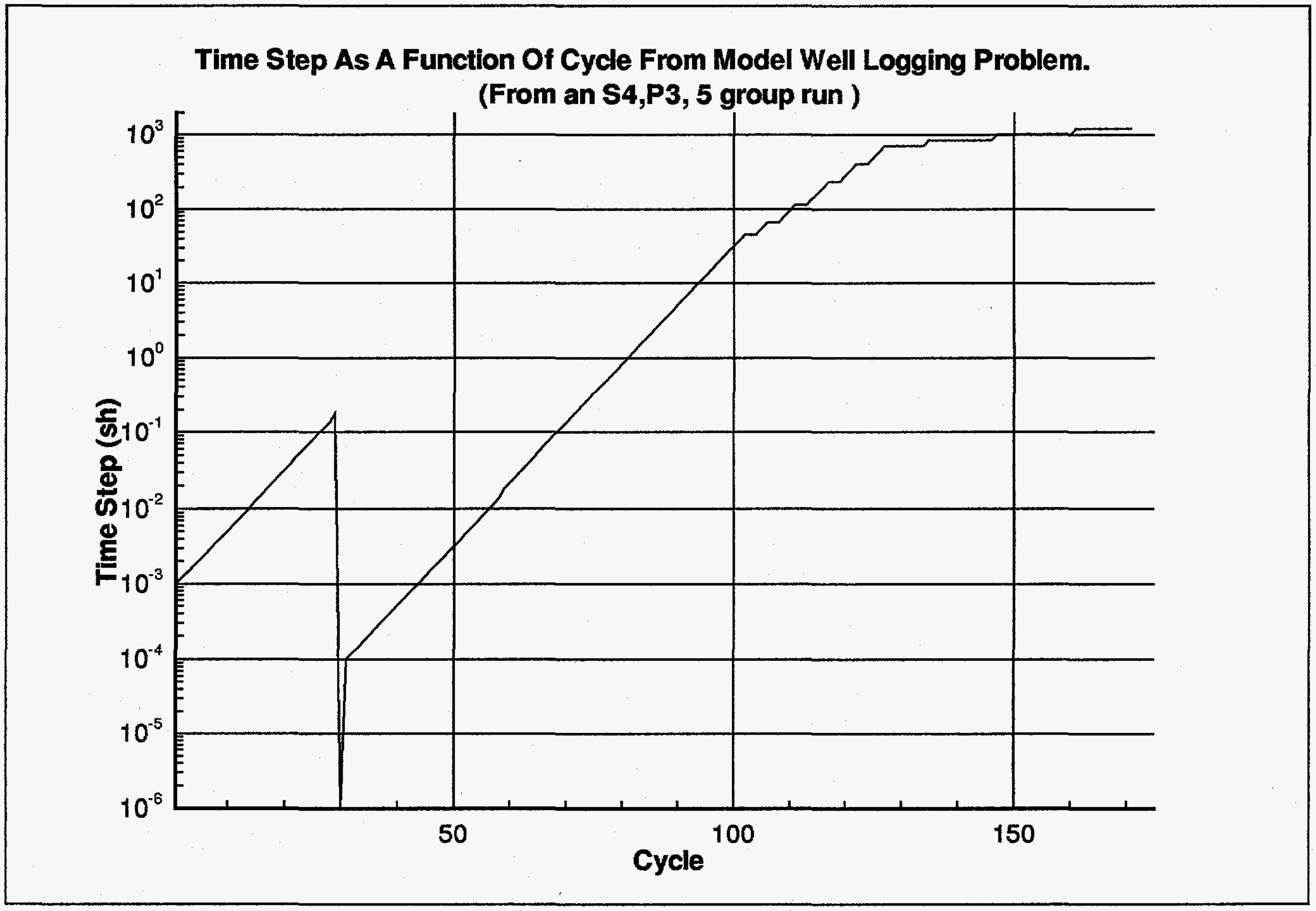


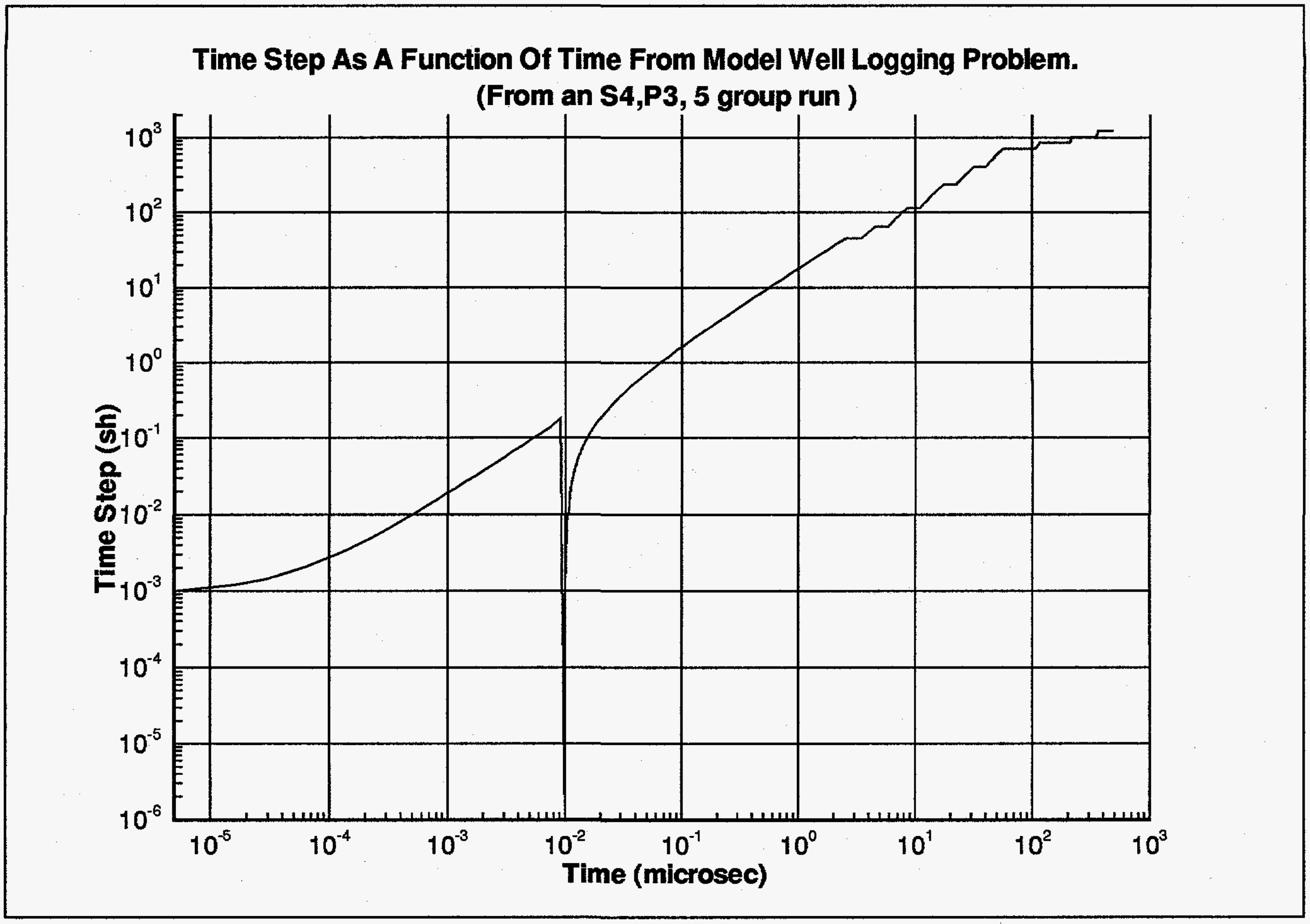


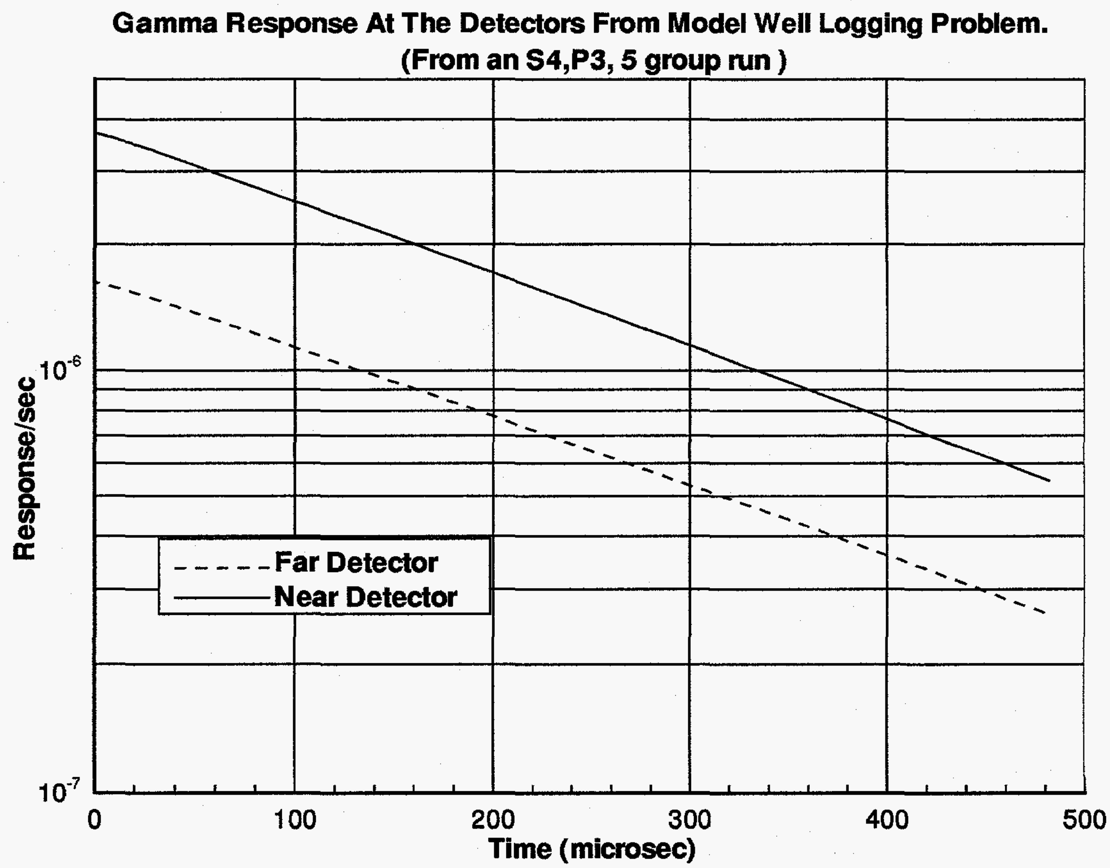
tions; the first 3 groups are neutrons and the last 2 are gammas. The problem was thus run as an S4s, P3, 5 group approximation. The source is placed in the source region of the problem and is assumed on for 0.01 microseconds and then switched off. The initial condition is that the source has been on for an infinite time. We then monitor the time response of the system by monitoring the gamma response of the two detectors.

Calculational Results

In the running of this problem, we chose an initial time step of 1.0E-5 microseconds and let the problem run until 500 microseconds with our strategy of an automatic choice of time step. In Fig. 3 we show the time step as a function of cycle to demonstrate how the time step increases as the problem progresses. We see clearly the effect of the source whose shutoff causes the time step to choose a minimum value of 1.0e-8 microseconds, but then increases steadily to about 12 microseconds at the end of the run. In the next figure we show the time step as a function of time. Finally in Fig. 5 we show the detector responses to the gammas as a function of time. This is a pure decay after the transient behavior from the source being switched off has been resolved. We ran this problem on the Cray T3D and it took 5100 seconds with 32 processors and 2755 seconds with 64 processors. For more information this problem required 173 time steps and a total of 5020 inner iterations whose convergence was accelerated by DSA in the code.




Time Step As A Function Of Time From Model Well Logging Problem.
(From an S4,P3, 5 group run)
Time Step (sh)
Time (microsec)
10^3
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
10^2
10^3



Conclusions

In developing our time dependent, three-dimensional, Sn transport module for parallel architectures, we have incorporated a successful, time step algorithm that makes for an efficient solver on these parallel machines. As can be seen from the times for a fairly realistic time-dependent problem, it is feasible to do these time-dependent calculations in a reasonable amount of machine time. This will have some valuable consequences for nuclear well logging for example and also for medical applications.