

LA-UR- - 98 - 272

*Title:*

TIME-DEPENDENT DETERMINISTIC TRANSPORT ON PARALLEL  
ARCHITECTURES USING PARTISN

CONF-980403--

*Author(s):*

R.E. Alcouffe  
R.S. Baker

RECEIVED

JUL 01 1998

OSTI

MASTER

*Submitted to:*

1998 Radiation Protection and Shielding Division  
Topical Conference, April 19-23, 1998, Nashville, TN

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

**Los Alamos**  
NATIONAL LABORATORY



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

Form No. 836 R5  
ST 2629 10/91

### **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

## **DISCLAIMER**

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

# TIME-DEPENDENT DETERMINISTIC TRANSPORT ON PARALLEL ARCHITECTURES USING PARTISN

Raymond E. Alcouffe (rea@lanl.gov) and Randal S. Baker (rsb@lanl.gov)

Transport Methods Group, MS D409

Los Alamos National Laboratory

Los Alamos, NM 87545

## Abstract

In addition to the ability to solve the static transport equation, we have also incorporated time dependence into our parallel  $S_N$  code PARTISN. Using a semi-implicit scheme, PARTISN 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 shielding and prompt fission experiments. This paper describes the form of the time-dependent equations implemented, their solution strategies in PARTISN including iteration acceleration, and the strategies used for time-step control. Results are presented for a iron-water shielding calculation and a criticality excursion in a uranium solution configuration.

## 1. Introduction

The solution of the time-dependent transport equation has long thought to be out of the realm of practical computations especially in two and three dimensions. The main reason is not that solution techniques have been unknown, but because their use implies an enormous amount of computer resources both in CPU time and memory. However, with the dawn of new and practical parallel architectures, the necessary amount of resources is now available for specialized applications and will soon be in general use throughout the nuclear engineering community. Thus it is now useful to add the time dependent capability to general purpose transport codes. In this paper we describe such an implementation which uses the traditional approach in time discretization and that takes advantage of the efficient computational techniques used to solve the static transport equation. We have done this implementation in our parallel code, PARTISN, formerly known as DANTSYS/MPI<sup>1</sup>, which is a derivative of DANTSYS<sup>2</sup> already released to the community. In the following we present the time discretization method and the alterations to the source iteration technique that are needed for a successful solution. We also present some computational examples demonstrating their usefulness and their impact upon computational efficiency.

## 2. Time Discretization of the Transport Operator

The time-dependent form of the source iteration equation is

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

where the left hand side is simple the streaming and total collision operator and right-hand side stands for the scattering operator, the fission source and any extraneous source that may be present. Time dependence is included in the total cross section to reflect possible external changes to the cross sections. We choose to use a semi-implicit scheme to handle the time derivative in Eq. (1), i.e.,

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

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

$$\vec{\Omega} \cdot \nabla \psi^n(\vec{r}, E, \vec{\Omega}) + \left[ \Sigma_T^n(\vec{r}, E) + \frac{2}{v \Delta t} \right] \psi^n(\vec{r}, E, \vec{\Omega}) = S^n(\vec{r}, E, \vec{\Omega}) + \frac{2}{v \Delta t} \psi^{n-1/2}(\vec{r}, E, \vec{\Omega}) \quad (3)$$

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

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

$$\Delta t = t^{n+1/2} - t^{n-1/2} \quad (4)$$

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  $t^{n+1/2}$ . If the time-extrapolated flux from the second of equations (3) is negative, then we use a fixup strategy where that negative flux is set to zero and the balance equation is rewritten to conserve particles. This is detailed in the next section.

### 3. The Source Iteration Equations and Iteration Strategy

Equation (3) is rewritten as source iteration equation with the energy variable discretized in multigroup form to show the iteration strategy and requirements along with a possible fixup in time:

$$\vec{\Omega} \cdot \nabla \psi_g^n + \left( \Sigma_{t,g}^n(\vec{r}) + \frac{a(\vec{\Omega})}{v_g \Delta t} \right) \psi_g^n(\vec{r}, \vec{\Omega}) = \sum_{l=0}^L \frac{2l+1}{4\pi} \Sigma_{s,l,g \rightarrow g}^n \sum_{q=-l}^l Y_{lq}(\vec{\Omega}) \psi_{l,g}^{q,n}(\vec{r}) +$$

$$\frac{b(\vec{\Omega})}{v_g \Delta t} \psi_g^{n-1/2} + \sum_{g' \neq g}^G \sum_{l=0}^L \frac{2l+1}{4\pi} \Sigma_{s,l,g' \rightarrow g}^{n*} \sum_{q=-l}^l Y_{lq}(\vec{\Omega}) \psi_{l,g'}^{q,n}(\vec{r}) + \frac{\chi_g}{4\pi} \sum_{g'=1}^G (v \Sigma_f)_{g'}^{n*} \psi_{0,g'}^n(\vec{r})$$

$$g = 1, \dots, G$$

$$n = 0, \dots, \infty \quad (5)$$

In equation (5), the left-hand side is the transport or streaming operator, the first term on the right-hand side is the self scattering source with Legendre angular components to order  $L$ , the next is the time source, then the source due to scattering into group  $g$  from all other groups, and the last term is the source due to fissions. Also in writing this equation, we have assumed:  $\psi_g^{n+1/2}(\vec{r}, \vec{\Omega}) = 2\psi_g^n(\vec{r}, \vec{\Omega}) - \psi_g^{n-1/2}(\vec{r}, \vec{\Omega})$  when  $\psi_g^{n+1/2}(\vec{r}, \vec{\Omega})$  is  $\geq 0$ , and  $\psi_g^{n+1/2}(\vec{r}, \vec{\Omega}) = 0$  when  $\psi_g^{n+1/2}(\vec{r}, \vec{\Omega})$  is less than 0. This contingency sets the values of  $a$  and  $b$  in the above equation, i.e.,  $a(\vec{\Omega}) = b(\vec{\Omega}) = 2.0$  if  $\psi_g^{n+1/2}(\vec{r}, \vec{\Omega}) \geq 0$ , and  $a(\vec{\Omega}) = 0$ ,  $b(\vec{\Omega}) = 1$  if  $\psi_g^{n+1/2}(\vec{r}, \vec{\Omega}) < 0$ . This set-to-zero fixup of a negative extrapolation of the time flux is important for the stability and accuracy of the method in some cases because particle balance is preserved while nonphysical negative fluxes are prevented.

The inner iteration is performed on the first line of the above equation for each energy group to converge the self scattering source terms (those which couple all the angles). The outer iteration is then done over all the groups to converge the fission term and the upscatter term. Thus the 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  $\Delta t$ , and the calculation proceeds to the next time step.

#### 4. Diffusion Acceleration of the Source Iteration

In our static code we have found it to be extremely advantageous to employ an iteration acceleration technique called diffusion synthetic acceleration (DSA)<sup>3</sup>. In brief, this method uses a specially crafted diffusion equation to compute the scalar flux that goes into the evaluation of the scattering sources. This is slightly modified for the time discretized equation to take the following form:

$$-\nabla \cdot \mathbf{D}(\vec{r}) \cdot \nabla \phi_g^n(\vec{r}) + \left( \Sigma_{t,g}^n(\vec{r}) + \frac{\langle a \rangle}{v_g \Delta t} \right) \phi_g^n(\vec{r}) = \frac{\langle b \rangle}{v_g \Delta t} \phi_g^{n-1/2} + \sum_{g'=1}^G \Sigma_{s0,g' \rightarrow g}^n \phi_{g'}^n(\vec{r}) +$$

$$\chi_g \sum_{g'=1}^G (v \Sigma_f)_{g'}^n \phi_{g'}^n(\vec{r}) - (\nabla \cdot \mathbf{D}(\vec{r}) \cdot \nabla \psi_{0,g}^n(\vec{r}) + \nabla \cdot \mathbf{J}_g(\vec{r}))$$

$$g = 1, \dots, G \quad (6)$$

where the transport derived quantities are defined as:

$\psi_{0,g}^n(r) = \frac{1}{4\pi} \int \psi_g^n(r, \underline{\Omega}) d\underline{\Omega}$  and  $\mathbf{J}_g(r) = \frac{1}{4\pi} \int \underline{\Omega} \psi_g^n(r, \underline{\Omega}) d\underline{\Omega}$ ;  $\langle a(r) \rangle$  and  $\langle b(r) \rangle$  are the angular integrated values of  $a$  and  $b$  of the time discretized transport equation.

The DSA equation (6) is solved for each energy group to accelerate the convergence of the inner iterations and also, if necessary, as a multigroup equation to accelerate the convergence of the fission and upscattering sources. This of course can involve considerable computational effort especially for 3D problems. What is postulated is that the cost of this additional work is more than offset by the decrease in the number of transport iterations that need be performed to attain convergence in each time step. This point is shown in the results of the problems described below.

#### 5. Storage Considerations for Time Dependent Transport Problems

In time-dependent, three-dimensional 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.,  $(L+1)^2 \times G$  words per cell, where  $L$  is the  $P_L$  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  $2 \times M \times 8 \times G + (L+1)^2 \times G$  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. The core storage can be greatly reduced by writing the time boundary fluxes (at  $t^{n-1/2}$  and  $t^{n+1/2}$ ) to disk. In this way 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 which is certainly is true for machines such as the Cray/Y-MP. For parallel architectures this strategy assumes that parallel I/O is being efficiently done which may or may not be true. It is not true for the Cray T3D thus we keep the time boundary angular fluxes in core when using that machine.

#### 6. Implementation in PARTISN

The existing 3-D X-Y-Z sweepers in PARTISN have been modified to solve the time-dependent form of the transport equation specified in Eq. (5). For static problems, the time absorption and source terms in Eq. (5) 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 PARTISN, 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  $\Delta t$  is reduced by a factor of twice the relative change in the reaction rate.

## 7. Description of the Test Problems

### 7.1 Iron-Water Shield Problem

For the purposes of demonstrating some of the properties of our solution technique for the time dependent transport equation, we describe two simple test problems. The first is an iron-water shield problem. This is a three-dimensional, XYZ problem where the innermost region is 10x10x10 cm of water with a uniform source. This is surrounded by a 10 cm thick region of iron which is in turn surrounded by 30 cm of water. This is a type of shielding test problem which, because of the large amount of water, is a difficult transport calculation. The time dependence is driven by the source which is assumed to be on forever prior to its being shut off at 0.1 sh (1 sh=1.0e-8 secs). We then examine the total neutron population as a function of time as well as the net leakage for each of the energy groups. This problem is solved with 3 energy groups which represent roughly the 14 Mev region, the peak fission region, and the thermal region of energy space. We solve it using an S8/P0 ( $L=1$  in equation (5)) approximation with the adaptive weighted diamond difference (AWDD)<sup>4</sup> method for the spatial variables. The spatial mesh is uniform with 50x50x50 1 cm mesh size in each dimension. We used the Cray/T3D and 64 processors to do the calculation.

### 7.2 Critical Solution Problem

The second problem is a 3D fissioning problem with two regions. The first region is a solution of U-235 in water with dimensions 12.87x12.87x6.5 cm with reflecting boundaries on the left, bottom and front surfaces. This is surrounded by 15x15x6.5 cm of water. The system is initially critical when between time 0 to 10  $\mu$ s the U-235 concentration is doubled. It remains thus until 250  $\mu$ s when the U-235 concentration is gradually decreased to half its original value by 5000  $\mu$ s. The solution remains in this state until the problem is terminated at 12000  $\mu$ s. This problem is designed to represent a criticality type of accident. We monitor the total neutron population as a function of time as well as the total fission rate and leakage rate. The problem is solved with three energy groups which cover the peak fission, epithermal, and thermal regions respectively. For discretization we use an S4/P1 angular approximation with a 50x50x20 spatial mesh using diamond differencing with set-to-zero fixup. This was computed using one processor of the Cray Y-MP.

## 8. Computational Results

In presenting the computational results, we present the solution time assuming differing time step error criteria and the results of using or not using DSA acceleration of the iterations. We also present graphs of selected aspects of the solution which demonstrates the impact on accuracy of the differing options.

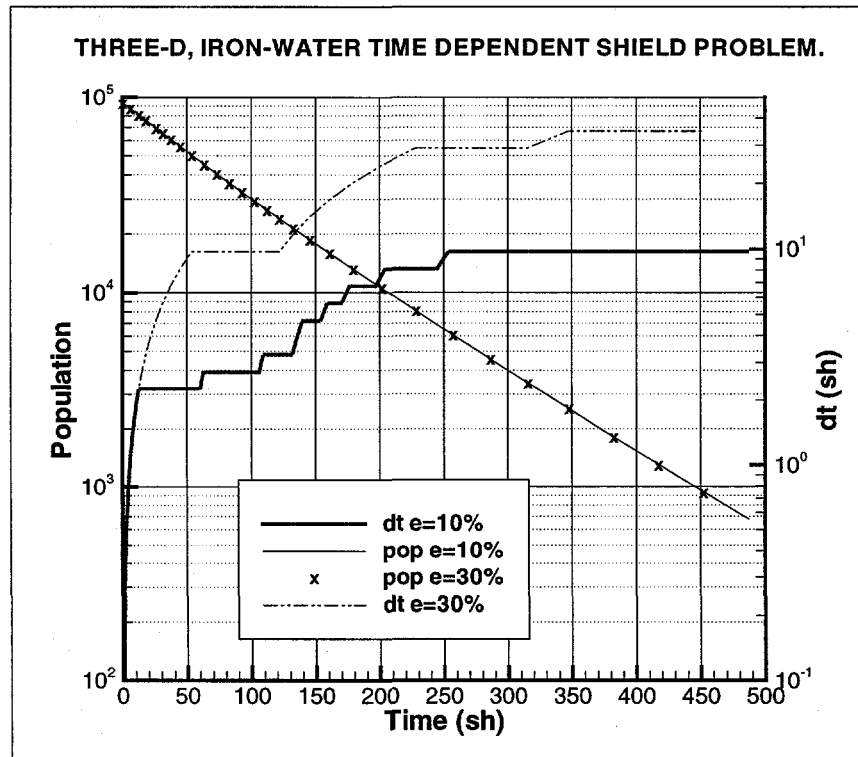
### 8.1 Iron-Water Shield Problem

For this problem we ran 4 cases: (1) DSA is on and the time step error criterion is 10%, (2) no DSA with the same error criterion as 1, (3) DSA is on with the time step error criterion at 30%, and (4) no DSA with the same error criterion as 3. In Table I we present the number of time steps taken, the number of inners performed, the total wall clock time as well as the time spent in the transport solver and the diffusion solver if applicable. Recall that these computations were done with 64 processors on the Cray T3D.

**Table 1: Selected Computational Results from the Iron-Water Shield Problem**

| Iteration accelerator | Error criterion | Number of time steps | Number of inners | Total CPU time (secs) | Transport time (secs) | Diffusion time (secs) |
|-----------------------|-----------------|----------------------|------------------|-----------------------|-----------------------|-----------------------|
| DSA                   | 10%             | 186                  | 2178             | 2690                  | 2377                  | 265                   |
| none                  | 10%             | 184                  | 6511             | 7057                  | 6955                  | -                     |
| DSA                   | 30%             | 104                  | 1142             | 1460                  | 1247                  | 187                   |
| none                  | 30%             | 104                  | 4850             | 4412                  | 4347                  | -                     |

These results show that the use of DSA on the iterations (only inner iterations in this case) is advantageous in both the reduction in the number of transport iterations and in the computation time. The effect is more pronounced for the 30% time step error criterion case because as the time step is increased, the iterations become more important to preserve the accuracy of the solution. To demonstrate the impact upon the DSA solution of the different error criteria, we present in Fig 1a comparison based upon the total neutron population.



**Fig 1. Comparison of the solution and time step for two error criteria using DSA.**

Also in this figure we show the time step (dt) selected as a function of time and thus the impact of the error criterion on the size of the step at later times. The population decreases exponentially as a function of time as expected from this type of problem. It is seen that the solution is essentially the same for the two time steps and thus from these



results it makes sense to use the larger time step as it significantly saves in computation time. The next figure shows the solution with and without DSA on the case with the 30% error criterion.

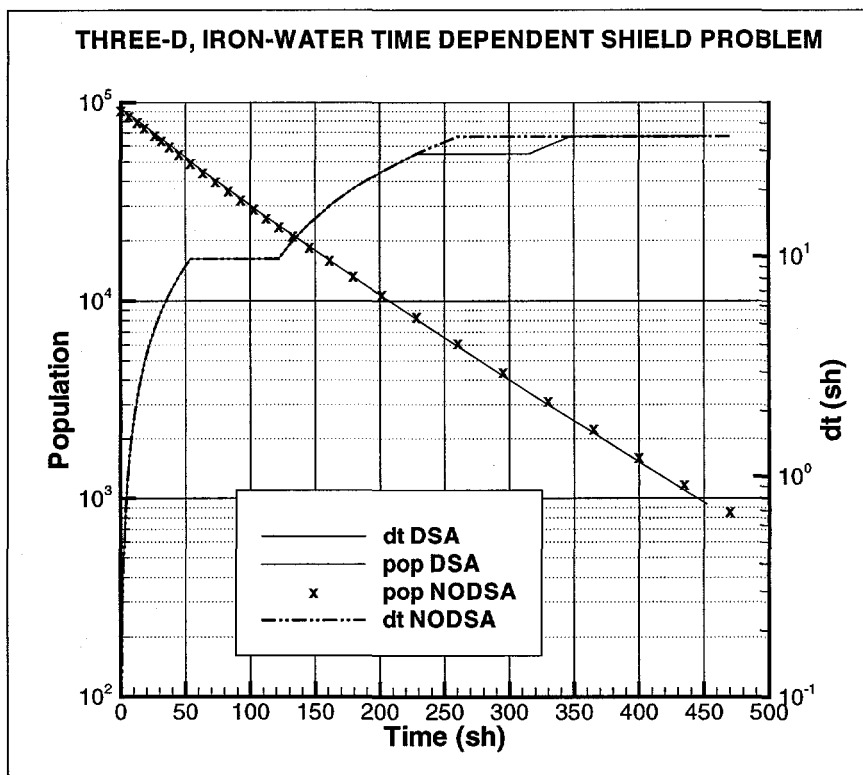


Fig 2. Comparison of solution and time step w/wo DSA for an error criterion of 30%.

There should be no difference in the solution with and without DSA which is nearly true from the results shown in Fig. 2. The time step chosen is also about the same. The differences that do exist is due to the weakness in iteration convergence for the no DSA case. Finally in Fig 3, we show that for the group dependent leakages even with DSA, the time step error criteria does make a difference. This is because the criterion is based upon a group averaged absorption rate which in this case is dominated by group 3. Thus the time step control necessary to obtain the correct solution for group 2 is not activated; we would need to generalize the criterion to monitor the groups we were interested in.

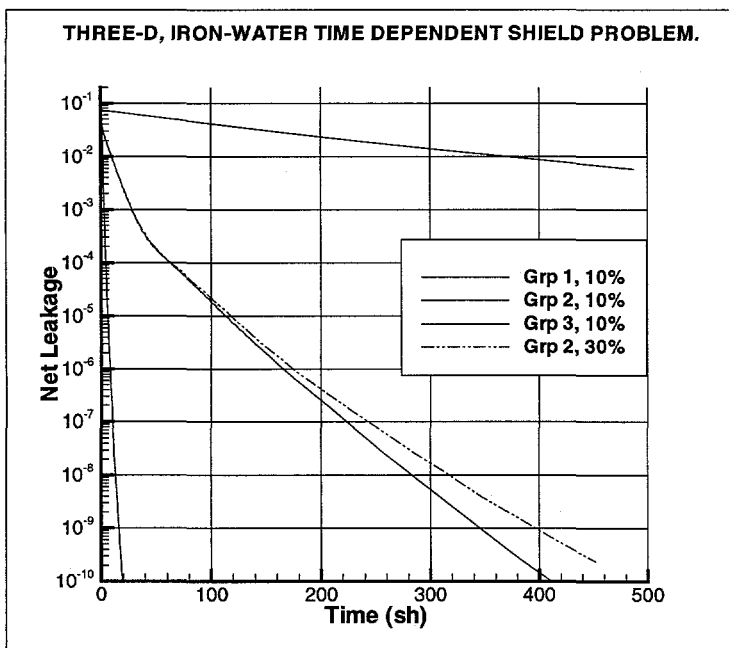


Fig 3. Group dependent net leakage comparing the impact of different error criteria.

## 8.2 Critical Solution Problem

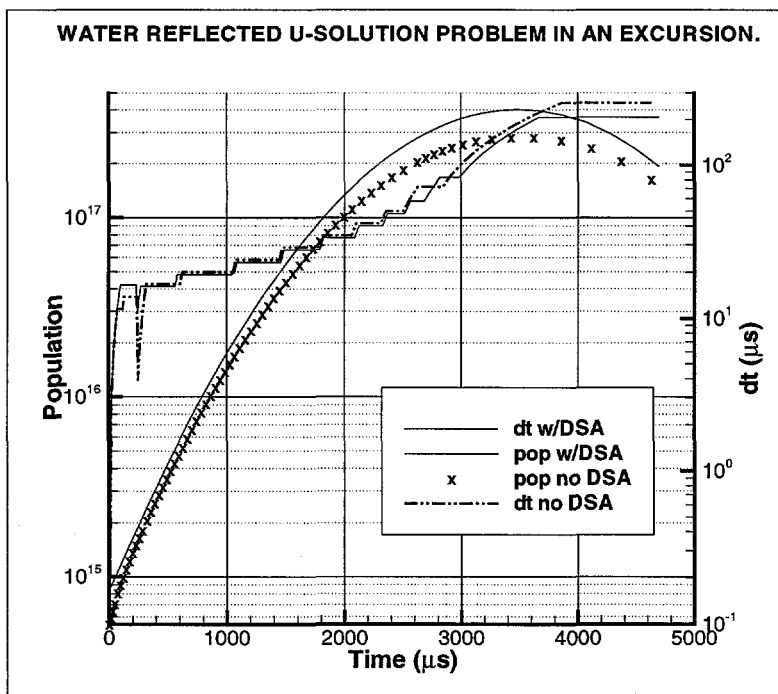
For this problem we ran 3 cases: (1) DSA is on with an error criterion of 10% and problem time of 5000  $\mu$ s, (2) DSA is not on but otherwise the same as 1, (3) the same as 1 except the problem time is 12000  $\mu$ s, and (4) the same as 3 with a 20% error criterion. The first two problems were run to show the computational impact of the DSA, and the next two to show that the excursion could be followed to a reasonable end state. In Table 2 we present the computational results in terms of the number of iterations and the computation time for each of the cases. The first column indicates the accelerator use and the problem time in ms. Recall that this was run on 1 processor of a Cray

**Table 2: Selected Computational Results from the Critical Solution Problem.**

| Iteration accelerator | Error criterion | No. of time steps | No. of outers | No. of inners | Total CPU time (secs) | Transport time (s) | Diffusion time (s) |
|-----------------------|-----------------|-------------------|---------------|---------------|-----------------------|--------------------|--------------------|
| DSA/5ms               | 10%             | 189               | 393           | 3267          | 4026                  | 2042               | 1824               |
| none/5ms              | 10%             | 190               | 1062          | 37484         | 23902                 | 22780              | -                  |
| DSA/12ms              | 10%             | 225               | 488           | 4413          | 5446                  | 2780               | 2457               |
| DSA/12ms              | 20%             | 153               | 340           | 3013          | 3810                  | 1889               | 1777               |

YMP.

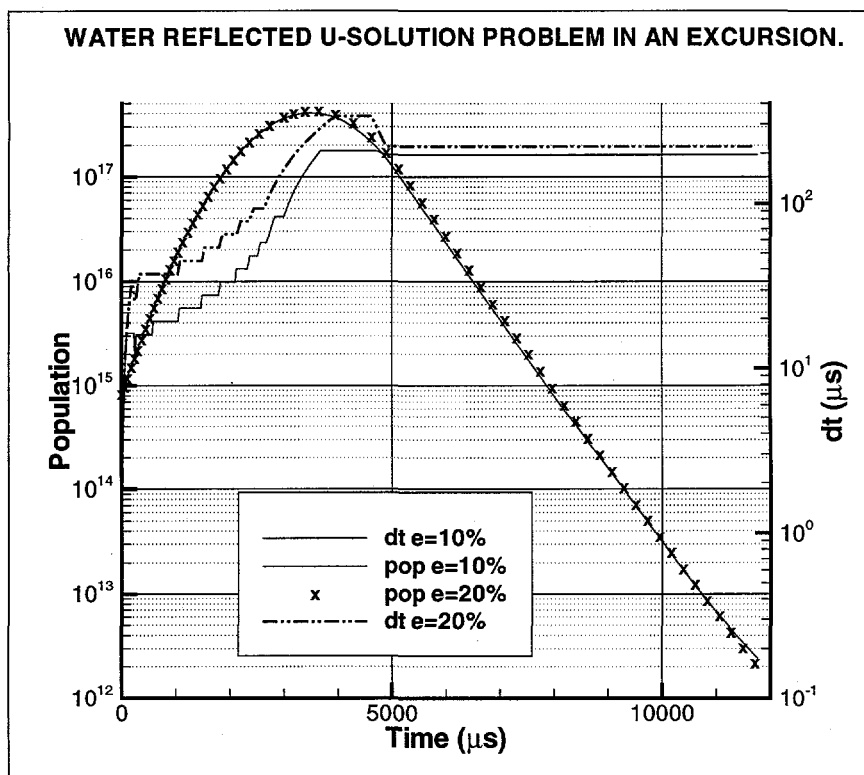
These results show that the DSA has a tremendous impact upon the calculational efficiency of the criticality problem.



**Fig 4. Comparison of DSA/noDSA Neutron Population and Time Step.**

In Fig 5 we present the solution over the entire range to 12 ms (12000  $\mu$ s) to illustrate the nature of the solution which is as expected. Note that at about 3500  $\mu$ s the time step goes to a value of about 200  $\mu$ s and remains at that value for the rest of the excursion. We also show a comparison with a solution computed using a 20% criterion.

The main reason is that outer iterations are involved and the DSA method is particularly effective in converging them. Thus there is an overall savings of more than an order of magnitude in the number of inners and a factor of 6 in computation time. Even with the increased computational cost, the computation with no DSA is not as accurate as with DSA. In Fig 5 we present the comparison of the two runs to 5 ms with and without DSA. It is seen that indeed the no DSA case does not match the DSA case. Part of the reason for this is that, for some time steps, the problem hit the outer iteration limit which was set to the default of 20. One could rerun this with a higher limit and the comparison would no doubt be better but we felt we could not afford to repeat this calculation. These problems were run with a time step error criterion of 10%.



## 9. Concluding Remarks

We have outlined the methods we use to compute time dependent transport problems in multidimensions. Our time difference method is a simple diamond with fixup approach imbedded in our previously static, parallel transport code, PARTISN. For the problems selected our DSA method for the convergence of the iterations involved in the solution is a crucial ingredient. This certainly is true for the fissioning system. We have demonstrated that time dependent transport computations even in three dimensions are becoming practical especially in the world of parallel computer architectures. The performance of the problems we show here will give an idea of the computational time of more spatially complex problems in that as the number of spatial mesh intervals needed to describe the problem increases, one need only add more processors and the time will be approximately the same as given here.

## REFERENCES

1. R. S. Baker, R. E. Alcouffe, "Parallel 3-D Sn Performance for DANTSYS/MPI on the Cray T3D," Proceedings of the Joint International Conference on Mathematical Methods and Supercomputing for Nuclear Applications, Saratoga, NY, 1997.
2. R. E. Alcouffe et al., "DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System," Los Alamos National Laboratory Manual LA-12969-M (1995).
3. R.E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Differenced, Discrete-Ordinates Equations", *Nucl. Sci. Eng.* **64**, 1977.
4. R.E. Alcouffe, "An Adaptive Weighted Diamond-Differenced Method for Three-Dimensional XYZ Geometry", *Trans. Am. Nucl. Soc.*, **68**, 205 (1993)