

Title: DANTSYS - A SYSTEM FOR DETERMINISTIC, NEUTRAL PARTICLE TRANSPORT CALCULATIONS

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

RECEIVED  
FEB 14 1997  
OSTI

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

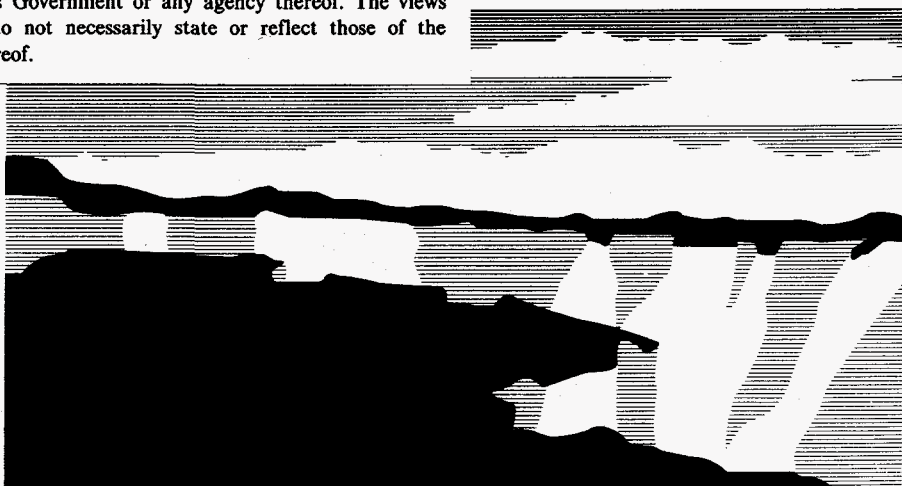
Submitted to: OECD/NEA Meeting, 2 - 3 December 1996, Paris, France

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.

MASTER

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.

**DISCLAIMER**

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

## DANTSYS - A SYSTEM FOR DETERMINISTIC, NEUTRAL PARTICLE TRANSPORT CALCULATIONS

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

### Introduction

The THREEDANT code is the latest addition to our system of codes, DANTSYS<sup>1</sup>, which perform neutral particle transport computations on a given system of interest. The system of codes is distinguished by geometrical or symmetry considerations. For example, ONEDANT and TWODANT<sup>1</sup> are designed for one and two dimensional geometries respectively. We have TWOHEX<sup>2</sup> for hexagonal geometries, TWODANT/GQ for arbitrary quadrilaterals in XY and RZ geometry, and THREEDANT for three-dimensional geometries. The design of this system of codes is such that they share the same input and edit module and hence the input and output is uniform for all the codes (with the obvious additions needed to specify each type of geometry). The codes in this system are also designed to be general purpose solving both eigenvalue and source driven problems.

In this paper we concentrate on the THREEDANT module since there are special considerations that need to be taken into account when designing such a module. The main issues that need to be addressed in a three-dimensional transport solver are those of the computational time needed to solve a problem and the amount of storage needed to accomplish that solution. Of course both these issues are directly related to the number of spatial mesh cells required to obtain a solution to a specified accuracy, but is also related to the spatial discretization method chosen and the requirements of the iteration acceleration scheme employed as will be noted below. Another related consideration is the robustness of the resulting algorithms as implemented; because insistence on complete robustness has a significant impact upon the computation time. We address each of these issues in the following through which we give reasons for the choices we have made in our approach to this code. And this is useful in outlining how the code is evolving to better address the shortcomings that presently exist.

## General Solution Approach In DANTSYS

DANTSYS employs the multigroup, discrete ordinates method to discretize the momentum variables in the transport equation. This leads to a coupled set of linear partial differential equations in the space variables where the coupling is through the collision source term and the fission term. To illustrate the form of the discretized equation and the source iteration process, we write the system of equations as:

$$\underline{\Omega}_m \cdot \nabla \psi_{g,m}^{k+1/2} + \sigma_{t,g}(\underline{r}) \psi_{g,m}^{k+1/2} = \sum_{l=1}^L \sum_{q=-l}^l \sigma_{s,l,g \rightarrow g} \phi_{g,l,q}^{k+1/2} Y_{l,q}(\underline{\Omega}_m) +$$

$$\sum_{g'=1}^{g-1} \sum_{l=1}^L \sum_{q=-l}^l \sigma_{s,l,g' \rightarrow g} \phi_{g',l,q}^{k+1/2} Y_{l,q}(\underline{\Omega}_m) + \frac{\chi_g}{k_{eff}} \sum_{g'=1}^G (\nu \sigma_f)_{g'} \phi_{g'}^k +$$

$$\sum_{g'=g+1}^G \sum_{l=1}^L \sum_{q=-l}^l \sigma_{s,l,g' \rightarrow g} \phi_{g',l,q}^k Y_{l,q}(\underline{\Omega}_m) + Q_{g,m}(\underline{\Omega}_m)$$

$$g = 1, \dots, G; \quad m = 1, \dots, M$$

Eq. 1

where:

$$\phi_g(\underline{r}) = \sum_{m=1}^M w_m \psi_{g,m}(\underline{r})$$

$$\phi_{g,l,q}(\underline{r}) = \sum_{m=1}^M w_m \psi_{g,m}(\underline{r}) Y_{l,q}(\underline{\Omega}_m)$$

In Eqn 1,  $g$  is the energy group index,  $m$  is the angle index,  $Y_{l,q}(\underline{\Omega}_m)$  are the spherical harmonics, and  $\underline{\Omega}_m$  is the discrete direction of particle travel. Thus it is seen that the scattering function is expanded in Legendre polynomials which requires the flux be expanded in spherical harmonics. The terms on the right hand side of Eqn 1 are the within group scattering source, the down scatter, the source from fissions and the upscatter source. Since the order of computation is from higher energy to lower ( $g=1$  corresponds to the highest energy group), the down scatter term is known. The superscript  $k$  refers to the outer iteration index; thus  $k+1/2$  is the current level of iteration and  $k$  indicates the previous level. The within group scatter is indicated at level  $k+1/2$  which means that an inner iteration process for each group must be employed to do this. Thus, the source iteration process for solving the transport equation consists of inverting the left hand side of Eqn 1 onto the source (right hand side) which has been evaluated from the previous iteration. We

write the source iteration in this fashion because we employ iteration acceleration methods to raise the iteration level to  $k+1$ . In the simplest case of no iteration, we simply set  $\phi_g^{k+1} = \phi_g^{k+1/2}$ . When the DSA<sup>3,4</sup> acceleration method is used, then  $\phi_g^{k+1}$  is the solution of a specially formulated diffusion-like equation. The use of this DSA technique greatly speeds the convergence of the source iteration process especially for eigenvalue calculations.

We recapitulate the solution process for the discrete ordinates transport equation as follows:

1. Evaluate the source from a flux guess or the previous iterate.
2. Invert the transport operator (the right hand side of Eqn 1) onto this source.
3. Evaluate the spherical harmonic moments of the flux from the angular flux.
4. Evaluate the scalar flux from the angular flux or for DSA compute the elements of the DSA equation and solve it for the scalar flux in each group.
5. Check for convergence depending upon the convergence criteria specified.
6. If the solution has not met the requirement, return to step 1.

Thus from a computational point of view, the main work is in inverting the transport operator and in solving the diffusion equation if DSA is invoked. The effort in obtaining the spherical harmonic expansion of the flux and computing the various sources is usually less than 10% of the total computation time on PVPs. In the section where we discuss solution strategy, we give an indication of how we measure the time to invert the transport and diffusion operators and its impact upon the default solution strategy. It turns out that this is an important effect on the performance of the code and on the robustness of the solution method.

### Combinatorial Geometry Capability In DANTSYS

When presented with a capability to do two- and three-dimensional calculations, the user has an automatic desire to be able to model the 'real' geometry that is to be calculated rather than to make an idealized model which adds some unknown degree of uncertainty to the solution obtained. This natural desire has been addressed to some extent in the TWODANT and THREEDANT modules in the sense that the user has the option to describe his geometry in terms of bodies much as is done in Monte Carlo codes. A calculational mesh is then overlaid on the geometry and the volume fractions of the intersection of the bodies with each mesh cell is computed. This volume fraction method is formulated so as to preserve the mass in the original body described geometry. The code we are using to set up the body described geometry and to compute the volume fractions on a prescribed XY or XYZ mesh is Frac-in-the box<sup>5</sup>. This geometry can be setup by hand instructions or through the use of our graphical users interface, JUSTINE. JUSTINE allows us to set up problems with a common geometry for THREEDANT and our Monte Carlo code, MCNP<sup>6</sup>. This GUI also allows us to couple Monte Carlo calculations and deterministic calculations through a system we call AVATAR. The linking is done by doing an adjoint calculation on the problem geometry with THREEDANT and then generating importances as a function of space, angle and energy group for the weight window mechanism in MCNP. This has allowed a significant speedup of some MCNP calculations.

In using the volume fraction method for the deterministic calculations of geometries described by bodies in 3D space, there is a source of error due to a 'fuzzing' of the boundaries of the various regions. This puts an added requirement on the spatial meshing of the problem in that these boundary cells should be less than

one mean free path around the boundaries for acceptable accuracy. We have demonstrated that if this is done, then accurate pointwise solutions result from this technique.

### Issues In Computational Efficiency

As pointed out above, computational efficiency is affected by many ingredients. The main considerations we address here and which are addressed in THREEDANT involve: (1) the CPU time per phase space cell to invert the transport operator, (2) the amount of time needed to solve the DSA equations, (3) the spatial differencing scheme, and (4) vectorization, parallelization and super scalar considerations. Efficiency also includes the memory required to implement the methods used and is affected by the type of calculation to be done, eigenvalue or source driven. Note that item (4) above is an ingredient in all issues of CPU time to do a computation and in the amount of memory required because of the way the algorithms are coded to take advantage of vectorization, super scalar and cash management and especially parallelization. The parallelization issues are treated in a separate companion paper so here we consider only the first three items in turn and indicate how they are handled in the code

#### *CPU Time to Invert Transport Operator.*

This time is governed by the computing platform type, of course. But on each platform certain optimizations can be done to reduce the amount of time spent in inverting the transport operator. Another consideration is the type of spatial differencing scheme used to solve the transport equation. The more elaborate the differencing method the more operations that need be performed and hence the more costly it is to use. We assess the time to invert the transport operator by quoting the average CPU time per phase space cell; i.e., the time per energy group per angle per spatial mesh cell. The spatial differencing schemes that are used in THREEDANT are the diamond differencing method with set-to-zero fixup (DD/STZ), the adaptive weighted diamond method (AWDD), and the nodal method both constant-linear and linear-linear<sup>8</sup>. Vectorization and super scalar transport sweepers use the same method for all differencing schemes. That is, we invert the transport operator a Z-plane at a time using diagonal line sweeps in the XY plane, following the direction of travel of the particles resulting in a variable vector length equal to the length of the diagonal in the XY plane. In the current XYZ solver, we make this somewhat more efficient by including all the angles in an octant in the sweep so that a gather-scatter operation is done. This is still vectorizable and is quite efficient on the Cray and the IBM. In Table 1, we present an average of the cell time for a variety of computing platforms and the spatial differencing schemes in THREEDANT gathered from representative problems.

**TABLE 1. Cell Time in Microseconds for a Variety of Platforms and Differencing Methods.**

Platform (1 processor)	Diamond	Diamond (STZ)	AWDD	CL Nodal	LL Nodal
Cray/YMP	0.25	0.5	0.5	1.25	2.2
Cray/T3D	2.9	5.0	5.0	-	-
IBM/RS6000/590	1.3	2.5	2.5	-	-
Sun/Sparc10	12.5	25.0	25.0	-	-

This table shows that the transport sweeper performs quite well on these platforms and that, as mentioned above, the spatial differencing scheme does influence greatly the amount of time needed to invert the transport operator.

### *Time Needed to Solve the DSA Equations.*

If the source iteration process were rapidly convergent, then the bulk of the computation time would just be that required to invert the transport operator. However, the source iteration process is frequently very slowly convergent so some means of iteration convergence acceleration must be used to obtain optimal efficiency in the transport calculations. This paper assumes that the reader is familiar with the concept of diffusion synthetic acceleration of the source iterations. Suffice it to say here that in DANTSYS, the DSA method involves the solution of a diffusion equation for each of the groups in the inner iterations and the solution of the multigroup diffusion equation for the outer iterations. To define inner and outer iterations, we refer back to Eqn 1. The first line of Eqn. 1 represents the inner iteration which is the source iteration process that accounts for the coupling of the angular flux through the within-group scattering source. In this iteration process, the other scattering terms and the fissions are assumed known from the previous iteration or the previous groups and are thus held fixed. The outer iteration process accounts for the coupling of the group fluxes through the fissions and upscattering processes. The DSA procedure can be used (and is used in DANTSYS) to accelerate the convergence of each of these processes. In the DSA process the main assumption is that we can replace transport iterations by diffusion (or some other similar low order operator) iterations and achieve a more computationally efficient solution. This being said, we must somehow measure the efficiency of the acceleration process against the CPU time required to use it to ascertain what the overall benefit will be.

Now the three-dimensional diffusion operator is by no means a trivial operator to invert; the inversion process is itself an iterative procedure. In THREEDANT we have four methods to invert the diffusion operator: (1) multigrid with line relaxation<sup>9</sup>, (2) conjugant gradient with 3-line sweep preconditioner, (3) conjugate gradient with 1-line sweep preconditioner, and (4) successive line relaxation. In general, multigrid is the most efficient and thus is the default. As a rule of thumb, the time to invert the diffusion operator is about the same time as that required to invert the transport operator using S4 quadrature. Thus in order for the DSA procedure to be efficient, it must be effective enough to reduce the number of iterations by at least a factor of three. In order to demonstrate some of the effectiveness of the DSA process and to assess the impact of the inversion of the diffusion operator we present three problems which have been described in detail elsewhere<sup>10</sup>. Two of the problems are eigenvalue problems and the third is a shielding problem. The first eigenvalue problem is a two group LWR model with a spatial mesh of 25x25x25 run in S8; and the second is a four group fast reactor assembly with a 32x32x20 spatial mesh and S8 quadrature. The shielding problem is the iron-water shield of ref 8 expanded to three dimensions with a spatial mesh of 50x50x50 and using S8 quadrature. In Table 2 we present some timing and iteration results on the Cray/YMP. These same runs on the IBM/RS600/590 took about 5 times longer in CPU time

TABLE 2. Timing and Iteration Data for Three Problems from THREEDANT/YMP.

Problem	Accelerator	Keff	Total CPU time(s)	# inners	# outers	trans time(s)	Diffusion time (s)	Work units <sup>a</sup>
LWR	none	0.96239	228.3	580	82	228.3	-	-
	DSA/MG	0.96236	10.4	15	7	5.1	5.1	441
	DSA/CG3L	0.96236	11.2	17	8	5.8	5.0	774
	DSA/CG1L	0.96236	17.2	16	7	5.5	11.4	533
	DSA/SR	0.96236	29.2	15	7	5.7	23.0	3390
FBRL	none	0.96973	905.7	1960	88	905.7	-	-
	DSA/MG	0.96999	48.3	53	10	24.8	21.7	1982
	DSA/CG3L	0.96999	46.7	54	10	25.3	19.6	2955
	DSA/CG1L	0.96999	70.3	52	10	24.4	44.1	1765
	DSA/SR	0.96999	69.3	53	10	24.6	42.9	6483
3DFEH2O	none	-	716.5	233	1	716.5	-	-
	DSA/MG	-	152.0	58	1	123.2	24.4	483
	DSA/CG3L	-	155.7	58	1	123.3	28.0	1062
	DSA/CG1L	-	197.3	58	1	123.3	69.6	687
	DSA/SR	-	301.8	52	1	160.2	138.7	6396

a. A work unit is a line relaxation through the entire 3D mesh.

In the above MG refers to multigrid, CG refers to conjugate gradient with either 3 line or 1 line preconditioning, and SR refers to successive line relaxation. It is seen from these examples that the acceleration is certainly an efficient way to go to reduce the number of iterations by nearly a factor of 60 for the eigenvalue problems and a factor of 4 in the source driven shielding problem. This translates into a savings of about a factor of 20 in CPU time in the eigenvalues and a factor of nearly 4 for the shield. We also see that from the work unit point of view, the MG solver is the most efficient giving a slight edge in computational time for the diffusion compared to CG and a substantial edge compared to SR.

### Solution Strategy

Mostly in the name of computational efficiency, it is expedient to devise a solution strategy to solve three-dimensional transport problems. The strategy is different for source driven calculations from that used in eigenvalue calculations. Both involve the assumption that the iteration accelerator is effective in reducing the error in the solution over that of the unaccelerated case. We outline each of these strategies in the following:

#### Source Driven Problems

In source driven problems without fissions, the strategy is to converge the inner iterations for each group according to the input convergence criterion. This criterion is such that the change in the scalar flux in each spatial mesh cell must be less than the input error, eg. less than 0.1%. If there is no upscatter, then the problem is solved. If there is upscatter, then outer iterations must be done and so the inner iteration procedure is repeated with the updated scattering source. When this source has been converged to the convergence crite-



tion, the problem is then complete. If there are fissions in the problem, then the strategy switches to that used for eigenvalue problems.

### ***Eigenvalue Problems***

In this case it is of little use to converge the inner iterations until the source has been converged. Thus when the DSA is effective, the strategy is to do only one inner iteration per outer for each group. The source is then iterated using the multigroup DSA equation with the transport information from the one inner. With the new source, then another inner is done for each group and the source is then recomputed using the multigroup DSA. This process continues until the source is converged with the supplied transport information. The inners are then converged for each group. It is very rare that another source iteration will need be done as the new source using the converged inner information is usually within the convergence criterion of the old source. This strategy greatly reduces the number of times that the transport operator need be inverted and hence reduces the CPU time for solution. It is crucial though that the inner acceleration process be effective, or else this procedure will not be efficient. The code will detect these situations for the most part and turn off the DSA for the affected groups. In these cases the multigroup DSA will be used with some of the groups not participating and the overall strategy is still effective. If not, we wind up in the worst case with solving the problem with no acceleration; but for a properly posed problem, this is very rare.

### ***Quality of Solution Aids***

With the complications of the iteration solution process outlined above and the use of spatial differencing schemes that require some sort of fixup, it is prudent to provide an indication of the quality of the solution obtained and to indicate iteration convergence problems. One of the most important aids is a balance table. This is data derived from the solution which gives the quantities in the angle-integrated balance equation derived from the conservative form of the discretized transport equation. Thus terms appear such as the absorption rate, the self-scattering rate, the in-scattering rate, the out-scattering rate, the source rate, the fission rate, and the net leakage rate for each group. Also computed is the error in the balance between the sources and losses in each group and the sum of the groups. Since this is a spatially integrated balance it should be less than the convergence criterion specified. This table is valuable because it expresses the physics of the problem being solved; and if there is something amiss, this table will show it.

Another aid is the iteration monitor which records the history of the error in the solution process and the number of iterations required for solution. The code also monitors the convergence rate and compares it with the expected rate of convergence. If the computed rate is far below the expected rate, the code prints out this information. The slower than expected rate of convergence is usually due to nonlinearities in the spatial discretization which come from the fixups. This in turn is due to the mesh being too large. Hence improvements in the iteration convergence performance can be seen if the mesh is made finer in most cases. Of course the user may know from this information which groups are giving problems, but there is nothing to say which spatial region. A way to provide this information is to provide the number of fixups used in each coarse mesh spatial region of the problem as is done in TWODANT. This will be provided in some form in the THREEDANT code in the future.

## Other Capabilities In DANTSYS

We have included other useful capabilities in the DANTSYS code to solve particular problems. Below is a partial list but indicates some of the directions we have taken.

### *Boundary Conditions*

Besides vacuum boundary conditions, we have also included reflective boundary conditions on all faces as needed, albedo boundaries, periodic boundaries and white boundaries. This allows the modelling of many types of cell problems and other related phenomena.

### *Group Dependent Sn*

The Sn order can be input as group dependent. Thus some groups which require it can use a high order angular quadrature while others use a lower order as needed. This is for the sake of efficiency in that the computing time is roughly proportional to the number of angles used (depending on the amount of vectorization). This option can even be used to specify some groups be solved by diffusion theory only.

### *First Collision Source*

In some problems of interest, the source is in a spatially very localized region. If the scattering is also relatively weak, serious ray effects will develop in the solution. Ray effects are the unphysical spatial oscillations in the solution due to the finite number of angles being used. One way to alleviate this, is to use a first collision source method. In DANTSYS we use a stochastic ray tracing option that does an analytic ray trace from the source region through all other regions of the problem. From this is constructed a first collision source which is spread throughout the problem region. Another transport problem is then solved using discrete ordinates and this first collision source. The solution to the original problem is then the sum of the ray tracing solution and the Sn solution. We also have a semi-analytic point source option which is similar to the above except that an analytic solution from the point source is used. This technique greatly improves the accuracy of the solution to such problems.

## Conclusions

DANTSYS is a code still actively under development but is also a mature code that has capabilities to solve many multi-dimensional transport problems in computationally efficient manner. The flexible geometry options are particularly attractive as well as the attention paid to solving eigenvalue problems. It is up and running on a variety of computing platforms and takes advantage of the vector, super scalar or parallel opportunities that are possible on each one. Not too much has been mentioned on memory management in this paper because not a whole lot of effort has gone into a modern memory management capability for the code. This is the weakest part of the code and we are addressing it now by converting the code to a FORTRAN90 structure. Also we intend to implement other more advanced spatial differencing schemes such as exponential discontinuous which, from research being done at Los Alamos, looks promising for deep penetration problems.

## References

1. R. E. Alcouffe, R.S. Baker, F. W. Brinkley, D. R. Marr, R. D. O'Dell, and W. F. Walters, "DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System", Los Alamos National Laboratory report LA-12969-M, (June 1995).
2. W. F. Walters, F. W. Brinkley, and D. R. Marr, "User's Guide for TWOHEX: A Code Package for Two-Dimensional, Neutral Particle Transport in Equilateral Triangular Meshes", Los Alamos National Laboratory report LA-10258-M, (Oct 1984).
3. R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Difference Discrete Ordinates Equations", *Nucl. Sci. Eng.* 64, 344 (1977)
4. E. W. Larsen, "Unconditionally Stable Diffusion-Synthetic Acceleration Methods for the Slab Geometry Discrete Ordinates Equations. Part I: Theory", *Nucl. Sci. Eng.* 82, 47(1982)
5. D. G. Collins and J. T. West, "FRAC-IN-THE-BOX Utilization-Revision 1", Los Alamos National Laboratory report LA-11606-MS, Rev 1, (Oct 1984).
6. J. F. Briesmeister, Editor, "MCMP<sup>TM</sup>- A General Monte Carlo, N-Particle Transport Code", Los Alamos National Laboratory report LA-12625-M, (Nov 1993).
7. R. D. O'dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analyses: Theory and Guidelines for Effective Use of Transport Codes", Los Alamos National Laboratory report LA-10983-MS, (Sep 1987).
8. W. F. Walters and R. D. O'Dell, "Nodal Methods for Discrete-Ordinates Transport Problems in X-Y Geometry," Proc. Am. Nucl. Soc. Top. Meeting on Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems, Munich, Germany, April 27-29, 1981, Vol. 1, pp 3-16
9. R. E. Alcouffe, Achi Brandt, J. E. Dendy Jr., and J. W. Painter, "The Multigrid Method for the Diffusion Equation with Strongly Discontinuous Coefficients", *SIAM J. Sci. Stat. Comput.* Vol 2, no. 4 (Dec 1981).
10. T. Takeda and H. Ikeda, "3-D Neutron Transport Benchmarks", OECD/NEA Committee on Reactor Physics report NEACRP -L-330 (Mar 1991).
11. R.E.Alcouffe, "An Adaptive Weighted Diamond-Differenced Method for Three-Dimensional XYZ Geometry", *Trans. Am. Nucl. Soc.*, 68, 205 (1993)